



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**BİLGİSAYAR TÜMLEŞİK ESNEK İMALAT
SİSTEMİNİN (CIM/FMS) MEKATRONİK
EĞİTİMİNE ENTEGRASYONU VE
MÜFREDAT GELİŞTİRİLMESİ**

BARIŞ DOĞAN

DOKTORA TEZİ

Elektronik-Bilgisayar Eğitimi Anabilim Dalı
Bilgisayar-Kontrol Eğitimi Programı

DANIŞMAN

Prof. Dr. Hasan ERDAL

İSTANBUL, 2018



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



BİLGİSAYAR TÜMLEŞİK ESNEK İMALAT SİSTEMİNİN (CIM/FMS) MEKATRONİK EĞİTİMİNE ENTEGRASYONU VE MÜFREDAT GELİŞTİRİLMESİ

BARIŞ DOĞAN

141200420070097

DOKTORA TEZİ

Elektronik-Bilgisayar Eğitimi Anabilim Dalı
Bilgisayar-Kontrol Eğitimi Programı

DANIŞMAN

Prof. Dr. Hasan ERDAL

İSTANBUL, 2018

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Marmara Üniversitesi Fen Bilimleri Enstitüsü Doktora Öğrencisi Barış DOĞAN'ın
“Bilgisayar Tümlşik Esnek İmalat Sisteminin (CIM/FMS) Mekatronik Eğitime
Entegrasyonu Ve Müfredat Geliştirilmesi” başlıklı tez çalışması, 09 Şubat 2018
tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Prof. Dr. Hasan ERDAL

(Danışman)

Marmara Üniversitesi

Prof. Dr. Nihat AKKUŞ

(Üye)

Marmara Üniversitesi

Prof. Dr. B. Koray TUNÇALP

(Üye)

Haliç Üniversitesi

Prof. Dr. Onur TUNCER

(Üye)

İstanbul Teknik Üniversitesi

Doç. Dr. Mustafa ONAT

(Üye)

Marmara Üniversitesi

ONAY

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 21/02/2018 tarih ve
2018/13-02 sayılı kararı ile Barış DOĞAN'ın Elektronik - Bilgisayar Eğitimi Anabilim
Dalı Bilgisayar - Kontrol Programında Doktora derecesi alması onanmıştır.

Fen Bilimleri Enstitüsü Müdürü
Prof. Dr. Bülent EKİCİ



ÖNSÖZ

2007 yılı Eylül ayında kabul edildiğim doktora programı kapsamında, 2009 yılı Ekim ayında, sunulmakta olan bu tez çalışmasına başladım. Epey uzun olan bu süreç boyunca beni ve çalışmalarımın seyrini etkileyen birçok olayla karşılaştım. Önce Araştırma Görevlisi olarak çalıştığım Teknik Eğitim Fakültesi kapanma sürecine girdi, yeni öğrenci alımını durdurdu. Sonrasında Teknoloji Fakültesi kuruldu. Başlarda Teknik Eğitim Fakültesi çalışanlarının akıbeti konusunda ve yönetimde belirsizlikler vardı. Teknik Eğitim Fakültesi'nde çalışıp da kariyerine Teknoloji Fakültesi'nde devam edecek personele “keyfi” kıstaslar getirilerek geçişler zorlaştırılmaya çalışıldı. Bu dönemde Teknik Eğitim Fakültesi'nde çalışmakta olan teknik altyapısı kuvvetli ve deneyimli birçok akademisyen başka üniversitelere yönelmek durumunda kaldı. Ayrıca “devlet politikası” gibi görünen sebeplerle akademik personelin, özellikle araştırma görevlilerinin maaşları da çok düşük seviyelerdeydi. Bu dönemde yeni personel alımları azaldığı gibi mevcut personel üzerine binen yük de artmış oldu.

Yukarıda saydığım sebepler, kendi kişisel merakım ve çıkış arama içgüdülerimle, 2011 yılında damperli araçların yük boşaltırken devrilmesini önlemeye yönelik “Elektronik Denge Kontrol Modülü - EDKM” geliştirme çalışmalarına başladım. Aynı yıl dosyasını kendim hazırlayarak TPE nezdinde incelemeli patent başvurusunda bulundum. 2012 yılında T.C. Bilim Sanayi ve Teknoloji Bakanlığı'nın vermiş olduğu Teknogirişim Sermaye Desteği hibe programına kabul edildim. Program gereği Mayıs 2012 de bir limited şirket kurdum. Bir sene boyunca EDKM geliştirme işleri ile şirket yönetmekle uğraştım. Bu sırada, aralarında ulusal kanallarda yayınlanan TV programları da olan fuar ve bilim şenliği gibi birçok etkinliğe katılarak hem projemi tanıttım, hem de üniversitemi temsil ettim. En son 2015 yılı Şubat ayında TÜBİTAK 2239 programı kapsamında, Amerika Birleşik Devletleri San Jose şehrinde bulunan “Silikon Vadisi” nde şirketleşme, Global ve teknolojik ürün geliştirmeye yönelik 1 aylık eğitim programına katıldım.

Sunmakta olduğum bu çalışmayı üzerinde yürüttüğüm FMS100 sistemi, 1990'lı yılların sonlarına doğru tasarlanmıştır. FMS100 2002 yılında Mekatronik Eğitimi bölümüyle birlikte faaliyete geçmiştir. Bölümün ilk kuruluş zamanından süre gelen elektrik altyapısındaki eksiklikler nedeniyle, FMS100 bünyesindeki cihazlarda zamanla çeşitli arızalar meydana gelmişti. Laboratuvardaki bilgisayarların güç kaynakları ve ana kartları sık sık arızalanıyordu. Bunların dışında FMS100 bünyesindeki imalat hücrelerinde

bulunan otomasyon denetleyicilerin (PLC, IPC) haberleşme kartı, güç kaynağı gibi donanımlarıyla; robot hafıza pillerinin bitmesi, robot hafızasındaki enkoder konumlarının silinmesi, robot eksen motoru ve kontrolör sürücü kartları arızası gibi birçok cihazda aksaklıklar çıkmıştı. Tüm bu aksaklıklar nedeniyle FMS100 bir bütün olarak kullanılamıyordu.

Öncelikle laboratuvar elektrik altyapısındaki eksikliklerin giderilmesi için bürokrasiyle çok uğraştım. Bir bilgisayarıcı ve kontrolcü olmama rağmen elektrik tesisatı ile ilgili birçok sorunla boğuşmak zorunda kaldım. Fakat ciddi deneyim sahibi oldum denebilir. Üniversitemiz Yapı İşleri Daire Başkanlığı'na yaptığım başvurular neticesinde, anca üç farklı ihale süreci sonucunda, elektrik altyapısındaki aksaklıkların giderilebilmesini sağladım. Sonrasında FMS100 bünyesinde zaman içerisinde oluşmuş olan diğer arıza ve aksaklıkların giderilmesi için; gerek şahsen (çoğunlukla), gerek araştırmacısı olduğumuz BAP projeleri kapsamında hizmet alımı yaparak, onarım çalışmaları yürütülmesini sağladım. Tüm bu zaruri işler, benim hem zamanımı, hem projeyi tamamlama enerjimi ve hepsinden önemlisi arzu ve isteğimi azalttı. Fakat süreç içinde epey emek vermiş olduğum çalışmadan da bir türlü kopamadım. Emeklerime ve hepsinden önemlisi büyük bedeller ile üniversitemize kazandırılmış olan FMS100'e de kıyamadım.

Görüldüğü üzere, oldukça uzun olan bu tez çalışması süresince farklı alanlarda birçok işle uğraşmak durumunda kaldım. Uğrunda ömrümün en güzel yıllarını ve sayısız emeğimi harcadığım bu çalışma sürecinde, tez kapsamında ve haricinde her tür konuda ilerleyebilmem için beni büyük bir sabırla destekleyen sevgili danışmanım Prof. Dr. Hasan ERDAL'a; Bölüm Başkan'ım Prof. Dr. Nihat AKKUŞ'a; eski Bölüm Başkan'larım Prof. Dr. Haluk KÜÇÜK, Prof. Dr. Koray TUNÇALP ve Doç. Dr. Erkan KAPLANOĞLU'na; arızaların giderilmesi ve özellikle robotların devreye alınması sırasında yardımlarını esirgemeyen öğrencim Ferit TİRYAKİ'ye ve adını sayamadığım diğer tüm öğrencilerime; iş arkadaşlarım Dr. Ersin TOPTAŞ ve Dr. Barkın BAKIR'a, Festo Firması çalışanı Yasin ARDIÇ ve Mitsubishi Elektrik firması çalışanı Necmi ÖMERDEDEOĞLU'na; sevgili meslektaşlarım Burak GÜL (eski öğrencim), Gönülkırılmaz BÜYÜK, Abdurrahman KAVUN'a; ülkemizdeki başarılı örneklerden biri olduğuna inandığım ve Esnek İmalat Sistemi eğitiminin verildiği İzmir Mazhar Zorlu Öğretmen Eğitim Merkezi'nde çalışan değerli teknik öğretmen meslektaşlarıma; hepsinden önemlisi maddi ve manevi destekleriyle, büyük bir sabırla bugünlere

gelebilmemi saęlayan sevgili Ailem'e sonsuz teőekkür eder; alıőma sonularının lkeme ve milletime faydalı olmasını dilerim.

Barıő DOĖAN

İstanbul, 2018

İÇİNDEKİLER

SAYFA

ÖZET	vii
ABSTRACT	ix
YENİLİK BEYANI.....	xi
CLAIM OF ORIGINALITY	xii
SEMBOLLER	xiii
KISALTMALAR	xiv
ŞEKİL LİSTESİ	xviii
TABLO LİSTESİ	xxix
1. GİRİŞ.....	1
1.1. Amaç ve Gerekçe.....	11
1.2. Erişilmesi Planlanan Hedefler	12
1.3. Literatür Araştırması.....	13
1.3.1. YÖK veritabanında “CIM”, “bilgisayar bütünleşik imalat” ve “bilgisayar tümleşik imalat” kelimeleri tarama sonuçlarında listelenen doktora ve yüksek lisans tez çalışmalarından, sunulmakta olan çalışma ile alakalı olanlar şunlardır: 13	
1.3.2. YÖK veritabanında “esnek imalat” ve “FMS” kelimeleri tarama sonuçlarında listelenen doktora ve yüksek lisans tez çalışmalarından, sunulmakta olan çalışma ile alakalı olanlar şunlardır:	15
1.3.3. YÖK veritabanında “esnek üretim” kelimesi tarama sonuçlarında listelenen doktora ve yüksek lisans tez çalışmalarından, sunulmakta olan çalışma ile alakalı olanlar şunlardır:	22
1.3.4. Literatür araştırma sonuçlarının yorumlanması:.....	24
2. MATERYAL VE YÖNTEM	26
2.1. Materyal.....	26
2.1.1. FMS100 Bilgisayar Destekli Esnek İmalat Sistemi	26
2.1.1.1. Taşıma Sistemi (Transport System)	28
2.1.1.2. Stok Hücresi (AS-RS)	29
2.1.1.3. Lehimleme Hücresi (Soldering)	30
2.1.1.4. Montaj Hücresi (Assembly)	31
2.1.1.5. Hidrolik-Baskı Hücresi (Hydraulic).....	31
2.1.1.6. CNC-Çifti Hücresi (CNC-Coupling)	32

2.1.1.7.	Görüntü İşleme Hücresi (Vision)	33
2.1.1.8.	Ana-Palet.....	34
2.1.1.9.	Yardımcı Palet.....	35
2.1.1.10.	Haberleşme Şeması.....	37
2.1.1.11.	İmalat Hücresini Devreye Alma ve Devreden Çıkartma	39
2.1.1.11.1.	Genel Olarak Yapılması Gerekli İşlemler	39
2.1.1.11.2.	Stok Hücresinin Devreye Alınması	41
2.1.1.11.3.	Lehimleme Hücresinin Devreye Alınması.....	44
2.1.1.11.4.	Montaj Hücresinin Devreye Alınması	44
2.1.1.11.5.	Hidrolik-Baskı Hücresinin Devreye Alınması.....	44
2.1.1.11.6.	CNC-Çifti Hücresinin Devreye Alınması.....	45
2.1.1.11.7.	Görüntü İşleme Hücresinin Devreye Alınması.....	47
2.1.1.11.8.	Taşıma Sisteminin Devreye Alınması	47
2.1.1.11.9.	COSIMIR FACTORY CONTROL Yazılımının Başlatılması ..	48
2.1.1.11.10.	FMS100'ü Devreden Çıkarma.....	48
2.1.2.	FMS100 Kullanımı	48
2.1.2.1.	Laboratuvar Davranış ve FMS100 Kullanım Kuralları.....	49
2.1.2.2.	Genel İmalat Sürecinin Tanımlanması	49
2.1.2.3.	Genel Alt İmalat Süreçlerinin Tanımlanması.....	50
2.1.2.4.	Alt-süreç 401 (Üst Kapak Frezeleme).....	51
2.1.2.5.	Alt-süreç 402 (Alt Kapak Frezeleme)	53
2.1.2.6.	Alt-süreç 403 (Metal Pin Tornalama)	54
2.1.2.7.	Alt-süreç 404 (Baskı Devre Lehimleme)	56
2.1.2.8.	Alt-süreç 405 (Baskı Devre Test).....	58
2.1.2.9.	Alt-süreç 406 (Elle Montaj)	59
2.1.2.10.	Alt-süreç 407 (Otomatik Montaj)	61
2.1.2.11.	Alt-süreç 408 (Hidrolik baskılama)	64
2.1.2.12.	SCADA Yazılımı (COSIMIR Factory Control)	66
2.1.2.12.1.	Yazılımın Çalıştırılması.....	66
2.1.2.12.2.	Kurulum Moduna Geçilmesi	67
2.1.2.12.3.	Üretim Moduna Geçilmesi.....	71
2.1.2.12.4.	Üretim Sürecinin Başlatılması	71

2.1.2.12.5. Üretim Sürecinin Sonlandırılması	74
2.1.2.13. Stok Yönetim Ekranı	74
2.1.2.13.1. Raf Seçme ve Düzenleme İşlemi	76
2.1.2.13.2. Getir İşlevi	76
2.1.2.13.3. Sakla İşlevi.....	78
2.1.2.13.4. Taşıma İşlevi.....	79
2.1.2.13.5. İlk Kez Başlatma İşlemi.....	80
2.1.2.13.6. Ürün Listesi ve Düzenlenmesi	80
2.1.2.13.7. Yazılıma Ait Uyarı ve Hata Mesajları	81
2.1.2.14. İmalat Hücresi Kontrolü	82
2.1.2.15. FMS100 Bünyesindeki Parça Numaraları	82
2.2. Çalışmada Uygulanan Eğitim Yöntemi.....	83
3. BULGULAR VE TARTIŞMA.....	85
3.1. Hazırlanan Ders Notları.....	85
3.1.1. Ürün, Üretim ve İmalat Kavramlarına Giriş.....	85
3.1.1.1. Sürekli Üretim.....	87
3.1.1.2. Ayrık Üretim	88
3.1.1.3. Yığın (Batch, Parti) Üretim.....	89
3.1.2. Bilgisayar Destekli İmalat ve Türevleri.....	90
3.1.3. Esnek İmalat Sistemlerinin Temel Bileşenleri	92
3.1.3.1. Hammadde, İş-Parçası ve Ürün.....	93
3.1.3.2. İstasyon ve Hücre Kavramları.....	93
3.1.3.3. Taşıma Sistemi	93
3.1.3.4. Depolama Sistemi (AS-RS)	93
3.1.3.5. Eyleyiciler	94
3.1.3.6. Algılayıcılar.....	94
3.1.3.7. Programlanabilir Denetleyiciler	96
3.1.3.8. Endüstriyel Haberleşme Sistemleri	96
3.1.4. Sıralı Programlama Yaklaşımı	97
3.1.4.1. Sıralı Durum Grafiği Yapısı.....	102
3.1.4.2. SDG Gösterimi Kuralları	106
3.1.4.3. SDG Yaklaşımı ile Programlama.....	107

3.1.4.3.1. SDG Gösterimi ile Sadece Algoritma Çıkarma Yöntemi	107
3.1.5. AS-i Endüstriyel Haberleşme Protokolü ve Cihazları	118
3.1.5.1. Donanım Yapısı ve Bileşenleri	120
3.1.5.1.1. Kablo:.....	120
3.1.6. Bilgisayar Destekli Talaşlı İmalat	121
3.1.6.1. CNC Tezgahı Keserek İlerleme ve İş Mili Devir Hesabı.....	123
3.1.6.2. Temel G Kodlar.....	126
3.1.6.3. Temel M Kodlar	131
3.1.6.4. NC Programlama Mantığı.....	131
3.1.6.5. Turn55 ve Mil55 CNC Tezgahları Tanıtımı	133
3.1.6.6. Turn55 ve Mill55 CNC Tezgahlarının Kullanımı.....	137
3.1.6.6.1. Tezgah Eksenlerini Sıfırlamak:.....	137
3.1.6.6.2. Kesici Takım Boy/Çap Telafı Ayarlarını Yapmak:	138
3.1.6.6.3. Kesici Takımı Eksenlerde Serbest Hareket Ettirmek (JOG):	138
3.1.6.6.4. Tezgaha NC Programı Yükleme:.....	138
3.1.6.6.5. Yeni Bir NC Programı Oluşturmak	138
3.1.6.6.6. NC Programı Görüntülemek ve Değiştirmek (EDIT):	139
3.1.6.6.7. NC Programın Başlangıç Satırını Seçmek:.....	139
3.1.6.6.8. Kesici Takım İlerleme/Dönme Hız Oranını Değiştirmek:.....	140
3.1.6.6.9. NC Programı Satır Satır Çalıştırmak:	141
3.1.6.6.10. NC Programı “Kuru” Çalıştırmak:	141
3.1.6.6.11. NC Programı Çalıştırmak/Durdurmak:.....	142
3.1.6.6.12. Serbest Kod Çalıştırmak:	142
3.1.6.6.13. Tezgaha Parça Bağlamak:.....	142
3.1.6.6.14. Parça Üzerinde Program Başlangıç Konumunu Belirlemek:...	143
3.1.6.6.15. Soğutma Havasını Açmak/Kapamak:	143
3.1.7. Endüstriyel Robot Kullanımı ve Programlanması.....	144
3.1.7.1. PLC ile Robot Kontrol Arayüzü Bağlantısı	150
3.1.7.2. Movemaster Komut Kümesi	156
3.1.7.2.1. Program Akış Denetim Komutları	156
3.1.7.2.2. Hareket Komutları	163
3.1.7.2.3. Sayısal Giriş/Çıkış Komutları	168

3.1.7.3.	COSIMIR Industrial Yazılımı ve Kullanımı.....	171
3.1.7.3.1.	Yeni Bir Proje Oluşturma	172
3.1.7.3.2.	Bilgisayar ile Robot Arası Bağlantı Kurma	175
3.1.7.3.3.	JOG İşlemi İle Robotu Hareket Ettirme	176
3.1.7.3.4.	Robot Programlama	178
3.1.7.3.5.	Program Benzetimi Yapma.....	184
3.1.7.3.6.	Projeyi Bilgisayardan Robota Yükleme	186
3.1.7.3.7.	Programı Başlatma/Yeniden Başlatma/Durdurma.....	188
3.1.7.3.8.	Tüm Programları Görüntüleme ve Bilgisayara Yükleme	189
3.1.8.	Fabrika Otomasyonu Hiyerarşik Kontrol Yapısı.....	190
3.1.8.1.	COSIMIR Factory Yazılımı	192
3.1.8.1.1.	COSIMIR Factory Yazılımında İmalat Hattı.....	194
3.1.8.1.2.	İmalat Hattındaki Malzeme Taşıma Sistemi.....	195
3.1.8.1.3.	İmalat Hattındaki Stok Yönetimi	196
3.1.8.1.4.	Otomasyon ve İmalat Cihazları Programları	197
3.1.8.1.5.	İmalat Veri Tabanı	198
3.1.8.2.	COSIMIR Factory Yazılımında İmalat Kontrol Sistemi (İKS)	199
3.1.8.3.	EİS ile Endüstri 4.0 Yaklaşımı Karşılaştırması.....	205
3.1.8.4.	COSIMIR Factory Control Yazılımı ve Kullanımı.....	207
3.1.8.4.1.	Kullanıcı Arayüzü	207
3.1.8.4.2.	Yeni Bir İmalat Projesi Oluşturma	208
3.1.8.4.3.	İmalatı Başlatma	216
3.1.8.4.4.	İleri Seviye Programlama ve Kullanım	218
3.1.8.4.5.	İmalat Süreci Projeleri İçin Görsel Arayüz Oluşturulması	238
3.2.	Hazırlanan Deney Föyleri.....	245
3.2.1.	Deney Adı: CNC Freze Tezgahı Kullanımı ve Program Yazma/Çalıştırma (CNC-Çifti Hücresi)	247
3.2.1.1.	Deney Başı Kontrol Listesi	247
3.2.1.2.	Deney İşlem Adımları	248
3.2.1.3.	Deney Sonu Kontrol Listesi	249
3.2.1.4.	Deney Sonuçları	250
3.2.2.	Deney Adı: RV-E2 Model Endüstriyel Robot Programlama Uygulaması (Lehimleme, Montaj ve Görüntü İşleme Hücreleri).....	251

3.2.2.1.	Deney Başı Kontrol Listesi	251
3.2.2.2.	Deney İşlem Adımları	252
3.2.2.3.	Deney Sonu Kontrol Listesi	254
3.2.2.4.	Deney Sonuçları	255
3.2.3.	Deney Adı: COSIMIR Factory Control Yazılımında Yeni Bir Süreç Planı Oluşturma – Temel İşlevler	256
3.2.3.1.	Deney Başı Kontrol Listesi	256
3.2.3.2.	Deney İşlem Adımları	258
3.2.3.3.	Deney Sonu Kontrol Listesi	261
3.2.3.4.	Deney Sonuçları	262
3.3.	Tartışma	263
4.	SONUÇLAR.....	267
	KAYNAKLAR.....	269
	EKLER	275
	FMS100 Bünyesinde Yürütülen Bakım ve Onarım İşleri	275
	CIM Laboratuvarı elektrik tesisatında giderilen sorunlar:	275
	CIM Laboratuvarı basınçlı hava tesisatında giderilen sorunlar:.....	276
	Profibus FMS kartında giderilen sorunlar:	276
	İmalat hücrelerindeki güç kaynaklarında giderilen sorunlar:	277
	Depolama hücresi Kartezyen konumlama sistemi hatası:	278
	Robot tutucu ellerinde görülen hatalar:	279
	CNC-Çifti hücrendeki robotun üçüncü ekseninde görülen sorunlar:	281
	Robotlar ve CNC tezgahları için yapılan bakım çalışmaları:	282
	Robotik havya için yapılan bakım ve güncelleme çalışmaları:	283
	ÖZGEÇMİŞ.....	285

ÖZET

BİLGİSAYAR TÜMLEŞİK ESNEK İMALAT SİSTEMİNİN (CIM/FMS) MEKATRONİK EĞİTİMİNE ENTEGRASYONU VE MÜFREDAT GELİŞTİRİLMESİ

Mekatronik kelimesinin kökleri mekanizma ile elektronik kelimelerinin birleşiminden meydana gelir. Sırf ismi bile göz önüne alındığında, Mekatronik birçok disiplinin bir araya geldiği karma bir bölümdür. Dolayısıyla Mekatronik Mühendisliği bölümü öğrencileri eğitim hayatları boyunca makina, elektronik, kontrol ve bilgisayar teknolojileri alanında eğitim almaktadırlar. Marmara Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği bölümü öğrencileri müfredat gereği endüstriyel algılayıcılar, elektro-hidrolik ve elektro-pnömatik, programlanabilir mantıksal denetleyiciler (PLC), gömülü sistemler, uygulamalı CAD ve CAM ve otomatik kontrol gibi geniş alanlardaki dersleri almaktadırlar. Öğrencilerin mezun olana kadar parça parça alınmış oldukları bu derslerin uygulamalı olarak birleştirilmiş hali Esnek İmalat Sistemleri dersidir.

Bu çalışmada eğitim amaçlı olarak üretilmiş fabrika otomasyon sistemleri arasında en büyüklerden biri olan FMS100 esnek imalat sisteminin bileşenleri ve hazırlanmış olan eğitim modülleri anlatılmıştır. FMS100 dördü robotik beslemeli olmak üzere toplam altı adet imalat hücresinden oluşan bir fabrika otomasyon sistemi örneğidir. Sahip olduğu CNC torna ve freze tezgahları ile mekanik imalat; robot uyumlu havya ile elektronik imalat yapma özelliğine sahiptir. Tüm parçaların bir araya getirildiği montaj hücresi ve hidrolik baskı ünitesinden çıkan ürünler, robotik beslemeli görüntü işleme ve kalite kontrol hücresinde test edilmektedir. 48 adet raftan oluşan stok hücresi ise

Bölüm 1’de mekatronik, üretim ve imalat, çıktılarına göre imalat çeşitleri ve esnek imalat konuları hakkında tanıtıcı kavramsal bilgiler verilmiş; yapılan bu çalışmanın amaçları ve hedefleri vurgulanmış; bu çalışma konularında literatürde yapılmış yayınlar ile yüksek lisans ve doktora tezleri incelenerek özetlenmiştir.

Bölüm 2’de bölümde tez konusu olan bilgisayar destekli esnek imalat sistemi olan FMS100 hakkında genel bilgiler; sistemi oluşturan hücrelar ve bünyelerinde bulunan otomasyon donanımları hakkında teknik bilgiler ve kullanım bilgileri verilmiştir. FMS100 sistemi hakkındaki bilgiler, üretici firmanın her bir sitasyon için İngilizce

hazırladığı 8 adet el kitapçığından birebir çevrilmiştir. Gerektiğinde, dilimize uyumu ve cümlelerin anlaşılmasını sağlamak için çevirilerde değişiklikler; cümle, kelime ve paragraf ilaveleri yapılmıştır. Bölümde sadece “FMS100 Manual” [79] isimli el kitabının çevisi yer almaktadır. FMS100’ü oluşturan diğer 7 bölüme ait dökümanların da dilimize çevirileri yapılmış ve Ekler bölümünde sunulmuştur. Ayrıca FMS100 sistemi üzerinde hangi yöntemlerle eğitim verildiği anlatılmaktadır.

Bölüm 3’de ise yoğunlukla FMS100 sisteminin eğitimi için tarafımızdan hazırlanan ders notları ve deney föyleri yer almaktadır. Ayrıca bölüm sonunda FMS100 sisteminin eğitim yönünden yeterliliği tartışılmıştır. Eğitim notları harzırlanırken 2. Bölümde incelenen ve çevirileri yapılan el kitapçıklarından faydalanılmıştır. Bunun dışında özellikle PLC programlama işlerini kolaylaştıracak ve belirli bir metodolojiye oturmasını sağlayacak Sıralı Durum Grafiği (SDG) yöntemi geliştirilmiştir. Bu bölümde hazırlanan eğitim materyalleri öncelikle FMS100 sistemiyle birlikte verilecek eğitimde kullanılacaktır. Bu nedenle bölüm içerisinde verilecek örnek kodlar hazırlanırken, FMS100 sisteminde bulunan donanımlar göz önüne alınmıştır. Örneğin PLC ile alakalı örnek kodlar açıklanırken S7-300 model PLC’ye göre hazırlanmış kodlar kullanılmıştır. Buna rağmen hazırlanan eğitim materyalleri içeriğinde, diğer üniversiteler ve endüstriyel okullar bünyesinde kullanılabilecek farklı model donanımlar da göz önüne alınmıştır. Yeri geldiğinde, örneğin S7-300 için verilmiş örnek uygulamanın açıklamasında, sıklıkla kullanılabilen Mitsubishi veya Fatek gibi firmaların PLC kodlarına göre nasıl olacağı hakkında kısa bilgiler verilmiştir.

Bölüm 4’de ise çalışmanın sonuçları değerlendirilmiştir.

ABSTRACT

INTEGRATING COMPUTER INTEGRATED FLEXIBLE MANUFACTURING SYSTEM (CIM/FMS) INTO MECHATRONICS EDUCATION AND CURRICULUM DEVELOPMENT

The roots of the mechatronic phrase come from the combination of the mechanism and electronic words. When even the name is taken into account, the Mechatronics is a mixed section where many disciplines come together. Therefore, the students of the Department of Mechanics Engineering are trained in the fields of machinery, electronics, control and computer technology throughout their education life. Marmara University Faculty of Technology Mechatronics Engineering department takes courses in a wide range of subjects such as industrial perceptrons, electro-hydraulic and electro-pneumatic, programmable logic controllers (PLC), embedded systems, applied CAD and CAM and automatic control for the students curriculum. The practical combination of these courses, where students are taken in pieces until they graduate, is the Flexible Manufacturing Systems course.

In this study, the components of the FMS100 flexible manufacturing system, one of the largest among factory automation systems produced for training purposes, and the prepared training modules are described. FMS100 is an example of a factory automation system consisting of a total of six manufacturing cells, four of which are robotic feeds. CNC lathe and milling machines with mechanical manufacturing; It has electronic manufacturing capability with robot-compatible soldering iron. The products from the assembly cell and the hydraulic printing unit, where all parts are assembled, are tested on the robotic feed image processing and quality control cell. The stock cell consisting of 48 rafts

In Chapter 1, introductory conceptual information is given about mechatronics, production and manufacturing, manufacturing varieties according to outputs and flexible manufacturing subjects; the aims and objectives of this work were emphasized; the publications in the literature and the master and doctorate theses in these study subjects are summarized.

General information about FMS100, a computer aided flexible manufacturing system, which is the subject of thesis in chapter 2; technical information and usage information about the cells forming the system and the automation equipment in their structures are given. Information about the FMS100 system has been translated into 8 manuals prepared by the manufacturer company for each site in English. When necessary, changes are made to translate in order to ensure compatibility with our language and understanding of the cues; sentences, words and paragraphs were added. Only the section "FMS100 Manual" [79] contains the translation of the handbook. The documents of the other 7 divisions forming FMS100 have also been translated into slices and presented in the Annexes. In addition, the methods of training on the FMS100 system are explained.

In Chapter 3, there are lecture notes and test papers prepared by us for intensive training of FMS100 system. Furthermore, the adequacy of the FMS100 system in terms of education has been discussed at the end of the chapter. While the training notes were being written, the handbooks that were reviewed and translated in Chapter 2 were utilized. In addition, the Sequential Statechart (SDG) method has been developed, which will facilitate PLC programming tasks and ensure that a specific methodology is available. The training materials prepared in this section will primarily be used in training with the FMS100 system. For this reason, while preparing the sample codes to be given in the section, the hardware included in the FMS100 system is taken into consideration. For example, while the sample codes related to the PLC are explained, the codes prepared according to the S7-300 model PLC are used. Nonetheless, in the context of the educational materials prepared, different models of equipment that can be used in other universities and industrial schools are also taken into consideration. For example, in the description of the example application for the S7-300, brief information is given about how commonly used companies such as Mitsubishi or Fatek can be used according to the PLC codes.

In Chapter 4, the results of the study are evaluated.

YENİLİK BEYANI

Marmara Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü bünyesindeki CIM laboratuvarında, eğitim amaçlı kullanılmak üzere kurulmuş bir Esnek İmalat Sistemi olan FMS100 bulunmaktadır. Sahip olduğu üretim istasyonu çeşidi ve teknolojisi bakımından FMS100, kendi alanında dünya genelindeki en büyük eğitim amaçlı fabrika otomasyon sistemlerinden biridir. Bu sistem, DPT projesi kapsamında 2002 yılında yaklaşık 8 milyon dolarlık bir bedele malolmuştur. Ancak, laboratuvar elektrik altyapısındaki yetersizlikler, yetişmiş öğretim elemanı eksikliği ve daha da önemlisi iyi bir eğitim için gerekli müfredatın olmaması gibi nedenlerle FMS100 bir bütün olarak gerektiği şekilde kullanılamamıştır. FMS100, Mekatronik Mühendisliği Bölümünde verilen; mekanik, elektrik, elektronik, robotik, kontrol, yazılım gibi alanlara ait tüm uygulamaların birlikte gerçekleştirildiği önemli bir eğitim sistemidir. Öğrencilerin eğitimler boyunca almış oldukları bilgileri mezun olmadan önce pekiştirdikleri bir eğitim ortamıdır. FMS100'ün gerektiği gibi kullanılamaması, bölümde yetişen öğrencilerin de eksik olarak mezun olmalarına sebep olmaktadır. Bu da hem üniversitemiz hem de ülkemiz açısından maddi ve manevi bir kayıptır.

FMS100 sistemine ait üretici firma tarafından sunulmuş tüm eğitim materyalleri ve el kitapları İngilizce'dir. Bazı cihazların Almanca'dan başka dokümanı da yoktur. Yapılan bu çalışma sayesinde, FMS100 sistemi için Türkçe müfredat geliştirilmiştir. Müfredat geliştirme kapsamında FMS100 sistemi bünyesindeki imalat hücreleri ve teknolojiler hakkında ders notları, deney föyleri, sınav soruları ve cevap anahtarları hazırlanmıştır. Hazırlanan materyal 2015-2016 ve 2016-2017 öğretim yılları bahar yarıyılında, Esnek İmalat Sistemleri isimli ders kapsamında 4. sınıf öğrencileri üzerinde uygulanmıştır. Bu yönüyle, yapılan bu çalışma kendi alanında özgündür ve ilk olma özelliği taşımaktadır.

Ayrıca bu çalışma sayesinde, büyük bedeller ödenerek kurulmuş olan FMS100'ün yazılım ve donanım aksaklıkları giderilmiş, müfredat geliştirilmiş; böylece FMS100 Mekatronik Mühendisliği lisans ve lisansüstü eğitim çalışmalarına daha verimli katkı sağlayacak hale getirilmiştir.

Yapılan bu tez çalışması sonuçlarının ileride ders kitabı haline getirilerek Mekatronik Mühendisliği eğitime katkısı daha da arttırılması planlanmaktadır.

CLAIM OF ORIGINALITY

Flexible Manufacturing System FMS100, which was established for educational purposes, is present in Marmara University Faculty of Technology Mechatronics Department CIM laboratory. By having a plenty of manufacturing systems and technologies, FMS100 is one of the largest educational fabric automation system around the World. This system costed about 8 million USD in year 2002. However, the lack of electricity infrastructure in CIM laboratory, lack of qualified teaching staff and, more importantly, a good FMS100 reasons such as the lack of the required curriculum for training as a whole could not be used as needed. FMS100, given in Mechatronics Engineering Department; mechanical, electrical, electronics, robotics, control is an important education system was carried out with all applications for areas such as software. Can not be used as part of this system, the students also grown in part is caused to be graduated as missing. This material both universities have moral losses in terms of both our countries.

Thanks to the work done in the Turkish language curriculum has been developed for FMS100 system. FMS100 system in manufacturing cells within the curriculum development and technology on grades, test sheets, exam questions and answer keys are prepared. The prepared material is applied on the 2015-2016 academic year the 4th grade students in the spring semester. In this respect, this study is unique in its field and carries the distinction of being first.

Thanks to this study, which established large fees paid fms100 software and hardware defects have been corrected, improved curriculum; so the Mechatronic Engineering undergraduate and postgraduate education has been made to contribute more effectively.

This thesis study and its products are planned to become a textbook as a future study by increasing its contribution to Mechatronics Engineering education.

SEMBOLLER

a	: Talaş derinliği [mm]
D	: Malzeme sabiti
E	: Elastisite modülü [Gpa]
F	: İlerleme [mm/rev]
F	: Talaş yüzeyindeki sürtünme kuvveti
F _c	: Asıl kesme kuvveti [N]
F _a	: İlerleme kuvveti [N]
F _n	: Kayma düzlemindeki normal kuvvet [N]
F _p	: Pasif kuvveti [N]
F _s	: Kayma düzlemine paralel kuvvet [N]
F _t	: İtme kuvveti [N]
l _c	: Talaş-takım temas boyu [mm]
K	: Direngenlik matrisi
M ₁	: Sönüm matrisi
M ₂	: Kütle matrisi
N	: Takım talaş yüzeyindeki kuvvet [N]
R	: Talaş kırıcı yiv radyüsü [mm]
R _a	: Yüzey pürüzlülük değeri [μm]

KISALTMALAR

AC	Alternating Current
ACK	Acknowledgement
AHP	Analitik Hiyerarşı Prosesi
AİS	Adanmış İmalat Sistemi
ARGE	Araştırma Geliştirme
AS-i	Actuator Sensor Interface
AS-RS	Automatic Storage Retrieve System
AUTO	Automatic
BAHP	Bulanık Mantık Analitik Hiyerarşı
BBİ	Bilgisayar Bütünleşik İmalat
BCD	Binary Coded Decimal
BTÜ	Bilgisayar Tümüleşik Üretim
CAM	Computer Aided Manufacturing
CIM	Computer Integrated Manufacturing
CIP	Continuous Integrated Process
CNC	Computer Numerical Control
CP	Communication Procesor
CPU	Central Processing Unit
CTS	Clear to Send
DC	Direct Current
DNA	Distributed Network Applications
DNC	Direct/Distributed Numarical Control
DP	Decentralised Peripherals
DTR	Data Terminal Ready

ECU	Engine Control Unit
EDKM	Elektronik Denge Kontrol Modülü
EIA	Electronic Industries Alliance
EİS	Esnek İmalat Sistemi
EİYS	Esnek İmalat Yönetim Sistemi
ERP	Enterprise Resource Planning
FBD	Functional Block Diagram
FM	Function Module
FMS	Flexible Manufacturing System
GA	Genetik Algoritma
HS	Handshake
HW	Hardware
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPC	Industrial Personal Computer
ISA	International Standards for Automation
İGTS	İmalat Görüntüleme ve Takip Sistemi
İKS	İmalat Kontrol Sistemi
İKY	İşletme Kaynak Yönetimi
KOBİ	Küçük ve Orta Büyüklükteki İşletmeler
LAD	Ladder Diagram
MAN	Manual
MDI	MultipleDocument Interface
MEMS	Micro Electro Mechanical Systems

METU	Middle East Technical University
MİP	Malzeme İhtiyaç Planlaması
MPI	Multi Point Interface
MPS	Modular Production System
MRP	Manufacturing Resource(s) Planning
NC	Numeric Control
NPN	Yarı iletkenlerin birleşiminden oluşan yapı, transistör tipi
ODA	Ortalama Değer Analizi
ODTÜ	Orta Doğu Teknik Üniversitesi
OF	Override Feedrate
OS	Override Spindle
PC	Personal Computer
PLC	Programmable Logic Controller
PNP	Yarı iletkenlerin birleşiminden oluşan yapı, transistör tipi
PS	Power Supply
PVC	Polyvinly Chloride
RAM	Random Access Memory
RFID	Radio Frequency Identification
RS	Reset Set
RTS	Request to Send
SBL	Single Block Operation
SCM	Supply Chain Management
SD	Sayısal Denetimli
SDG	Sıralı Durum Grafiği
SFC	Sequential Functional Chart
SM	Signal Module

SQL	Structured Query Language
SR	Set Reset
STL	Statement List
TCP	Transmission Control Protocol
TIA	Telecommunications Industry Association
TPE	Türk Patent Enstitüsü
TZÜ	Tam Zamanlı Üretim
TZY	Tedarik Zinciri Yönetimi
UML	Unified Modelling Language
YÖK	Yüksek Öğretim Kurumu
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

	SAYFA
Şekil 1-1. Mekatronik teknolojisinin bilim alanları ile ilişkisi.....	2
Şekil 1-2. Modern bir benzinli araç motoru elemanları [6].....	3
Şekil 1-3. CAM kodları ile ahşap üzerinde kesme işi (freze) yapan robot görüntüsü.....	4
Şekil 1-4. Zeytinyağı üretimi aşamaları.	5
Şekil 1-5. İş parçasından ürüne ayırık imalat süreci, örnek bir araba imalatı.	6
Şekil 1-6. Ürün çeşidi ve adedine göre AİS ve EİS arasındaki ilişki grafiği.	8
Şekil 1-7. Ebu'l İz El Cezeri tarafından tasarlanmış otomatik çalışan su makinasına ait çizim.	9
Şekil 1-8. Universidad Politecnica de Madrid'de bulunan topuzlu mekanik governöre sahip buhar makinası.	9
Şekil 1-9. Tarihsel endüstriyel devrim aşamaları.	11
Şekil 2-1. FMS100 genel görünüşe ait bilgisayar çizimi görüntüsü [79].....	27
Şekil 2-2. FMS100'ün ürettiği örnek ürün olan elektronik rulete ait bilgisayar çizimi (a), ruleti oluşturan iş-parçalarının bilgisayar çizimi (b).	28
Şekil 2-3. İmalat hücreleri arası iş-parçalarının taşınmasını sağlayan konveyör hattının bilgisayar çizimi [79].....	28
Şekil 2-4. Yardımcı paletlerin depolandığı AS-RS sistemine ait bilgisayar çizimi [79].29	
Şekil 2-5. Lehimleme hücresi bilgisayar çizimi [79].	30
Şekil 2-6. Montaj hücresi bilgisayar çizimi [79].	31
Şekil 2-7. Hidrolik Baskı hücresi bilgisayar çizimi [79].	32
Şekil 2-8. CNC-Çifti hücresi bilgisayar çizimi [79].....	33
Şekil 2-9. Görüntü İşleme istasyonu bilgisayar çizimi [79].	34
Şekil 2-10. Ana palet bilgisayar çizimi [79].....	34

Şekil 2-11. Ana-Palet altındaki manyetik etiket ve her istasyonda bulunan etiket okuyucu görüntüleri.....	35
Şekil 2-12. Yardımcı palet üzerindeki optik yansıtıcı görüntüsü.	36
Şekil 2-13. Palet 2ED isimli yardımcı paletin bilgisayar çizimi [79].....	36
Şekil 2-14. Palet 2PCB isimli yardımcı paletin bilgisayar çizimi [79].	37
Şekil 2-15. Palet 8PIN isimli yardımcı paletin bilgisayar çizimi [79].	37
Şekil 2-16. FMS100 endüstriyel haberleşme şeması [79].	39
Şekil 2-17. CNC-çifti hücreesine ait boş palet kabul alanları görüntüsü.	39
Şekil 2-18. Konveyör hareket yönüne göre yerleştirilmiş Ana-Palet’lerin görüntüsü. ..	40
Şekil 2-19. İmalat hücreleri otomatik ve elle çalışma tipi anahtarlarının görüntüsü: Stok hücresi (a), diğer hücreler (b).	40
Şekil 2-20. Stok yönetim ekranı “Stok Administration” görüntüsü.	43
Şekil 2-21. “First-time-init” butonu uyarı mesajı.	43
Şekil 2-22. Hidrolik pompa çalıştırma (yeşil) ve durdurma (kırmızı) butonları.	45
Şekil 2-23. CR1 robot denetleyici önden görünüşü.....	45
Şekil 2-24. CNC-Çifti istasyonu kontrol paneli görüntüsü.	46
Şekil 2-25. CNC cihazlarının yan tarafında bulunan güç ve çalışma tipi seçim anahtarları görüntüsü.	46
Şekil 2-26. Her istasyonda bulunan ve pnömatik vana kontrolünü yapan AS-i çalışanı görüntüsü.	47
Şekil 2-27. COSIMIR Factory Control yazılımının masaüstü kısayol simgesi (a), “Marmara1.jpg” dosyası seçme penceresi (b).	66
Şekil 2-28. COSIMIR Factory Control yazılımı ekran görüntüsü.	67
Şekil 2-29. COSIMIR Factory Control yazılımı üretim ve kurulum çalışma tipi seçim butonları.....	67
Şekil 2-30. COSIMIR Factory Control yazılımı proje menüsü görüntüsü.....	68
Şekil 2-31. Proje penceresi “Devices” sekmesi içeriği görünümü.	69
Şekil 2-32. Cihaz iletişim protokolü değişim penceresi.	69

Şekil 2-33. COSIMIR Factory Control yazılımında “Marmara1.lpj” proje dosyasında tanımlanmış süreç planları.....	70
Şekil 2-34. COSIMIR Factory Control yazılımı üretim modunda başlatma ekran görüntüsü.	71
Şekil 2-35. COSIMIR Factory Control yazılımının üretim modunda açıldığında ekran görüntüsü.	72
Şekil 2-36. Üretim ekranında iletişim kurulabilmiş (sarı) ve kurulamamış (mor) hücrelerin görüntülenmesi.	72
Şekil 2-37. Üretim ekranında aktif olan hücre ve görevlerin görüntülenmesi.	73
Şekil 2-38. Üretim ekranı taşıma sisteminde tanımlanmış ve hareket halindeki Ana-Paletlerin görüntüsü.....	73
Şekil 2-39. Üretim sonlandırma (finish production) ekranı görüntüsü.	74
Şekil 2-40. Stok yönetim ekranı görüntüsü (stok administration).....	75
Şekil 2-41. Stok yönetim ekranı hücre bilgileri düzenleme işlemi ekran görüntüsü.....	76
Şekil 2-42. Stok yönetim ekranında seçili duruma getirilmiş rafa ait ekran görüntüsü. 76	
Şekil 2-43. Stok yönetim ekranı üzerindeki getir (retrieve) işlevi bölümü görüntüsü. ..	77
Şekil 2-44. Stok yönetim ekranı üzerindeki getir (retrieve) işlevi “direk raf numarası ile getir” bölümü görüntüsü.....	77
Şekil 2-45. Stok yönetim ekranı üzerindeki sakla (store) işlevi “boş raf arama modu” bölümü görüntüsü.	78
Şekil 2-46. Stok yönetim ekranı üzerindeki sakla (store) işlevi “direk raf numarası ile depola” bölümü görüntüsü.	79
Şekil 2-47. Stok yönetim ekranı üzerindeki taşı (move) işlevi görüntüsü.....	79
Şekil 2-48. Stok yönetim ekranı üzerindeki “First-time-init” ve “Product List” butonları görüntüsü.	80
Şekil 2-49. Veri tabanı kayıtlarının değişebileceğine dair uyarı mesajı görüntüsü.....	80
Şekil 2-50. “Product List” penceresi görünümü.	81
Şekil 3-1. Bir üretim sistemindeki hammadde, ürün, ara-ürün ve atık ürün ilişkisi.....	86
Şekil 3-2. Zeytinyağı üretim sistemi ve aşamaları.	88

Şekil 3-3. İş parçasından ürüne ayırık imalat süreci, örnek bir araba imalatı.	89
Şekil 3-4. Örnek bir yığın üretim sistemi, meyve suyu üretimi.	89
Şekil 3-5. Algılayıcıların çıkış tiplerine göre sınıflandırılması.	95
Şekil 3-6. EİS bünyesinde kullanılabilecek örnek haberleşme protokolleri.....	97
Şekil 3-7. Aynı görevi yapan iki farklı adım sıralama örneği.	98
Şekil 3-8. “Sıralama 1” için mikro-denetleyici C kodu örneği.	99
Şekil 3-9. PLC çalışması ve çevrimi yapısı.	100
Şekil 3-10. Aynı süreç için hazırlanmış standart programlama akış şeması (a) ile SFC gösterimi (b) karşılaştırması.	101
Şekil 3-11. SDG yapısında basit adım (a), aktif adım (b), başlangıç adımı (c) ve aktif başlangıç adımı (d) gösterimi.	102
Şekil 3-12. SDG’de her adımda yürütülecek görevlerin gösterimi.	103
Şekil 3-13. SDG’de geçiş elemanları ve koşullarının gösterimi.	103
Şekil 3-14. Basit geçiş (a), birleşim geçişi (b), ayırım geçişi (c) ile birleşim ve ayırım geçişi (d)	104
Şekil 3-15. Koşullu dallanma gösterimi.	105
Şekil 3-16. Paralel dallanma gösterimi.	105
Şekil 3-17. Atlama gösterimi.	106
Şekil 3-18. Örnek uygulamaya ait elektro-pnömatik bağlantı şeması.	108
Şekil 3-19. Örnek uygulamaya ait SDG gösterimi.	109
Şekil 3-20. İlk vuruş kontağı için OB100 bloğuna ait PLC kodu.	110
Şekil 3-21. 0 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b).	110
Şekil 3-22. 20 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b)...	111
Şekil 3-23. 21 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b)...	112
Şekil 3-24. 22 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b)...	113
Şekil 3-25. 23 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b)...	114
Şekil 3-26. 24 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b)...	114

Şekil 3-27. 0 numaralı geçişe ait PLC kodu (a), sıralı durum grafiği gösterimi (b).....	115
Şekil 3-28. Gerçek sisteme çıkış işlemlerine ait PLC kodu.	116
Şekil 3-29. SR elemanı kullanım örneği (a), RS elemanı kullanım örneği (b)	117
Şekil 3-30. Örnek 1'e ait 4, 5, 6 ve 7. networklerin SR ve RS elemanları ile yazılmış kodları.....	117
Şekil 3-31. Bilgisayar destekli talaşlı imalat aşamaları.	122
Şekil 3-32. Bir freze kesici takımına (freze çakısı) ait katalog bilgileri.....	125
Şekil 3-33. G03 komutuyla R1 ve R2 yarıçaplı dairesel takım yörüngeleri oluşturulması.	127
Şekil 3-34. G02 komutuyla m1 ve m2 merkezli dairesel takım yörüngeleri oluşturulması.	128
Şekil 3-35. G41 koduyla takım yolu telafisi.....	129
Şekil 3-36. G42 koduyla takım yolu telafisi.....	130
Şekil 3-37. Bakır plaketten 40 x 80 mm ebatlarında parça kesimi örneği.....	132
Şekil 3-38. FMS100 bünyesindeki Mill55 tezgahı: dıştan görünüş (a), içten görünüş (b).	133
Şekil 3-39. FMS100 bünyesindeki Turn55 tezgahı: dıştan görünüş (a), içten görünüş (b).	134
Şekil 3-40. Turn55 tezgahının sahip olduğu otomatik takım değiştirme sistemi.	134
Şekil 3-41. FMS100 bünyesindeki CNC tezgahlarına ait bilgisayar ve kontrol klavyesi görünümü.....	135
Şekil 3-42. FMS100 bünyesindeki CNC tezgahlarında bulunan tezgah ve FMS modu açma/kapama anahtarları görünümü.....	136
Şekil 3-43. FMS100 bünyesindeki CNC tezgahlarında kullanılan klavyelerin görünümü: yatay ana kontrol klavyesi (a), dikey programlama klavyesi (b).	136
Şekil 3-44. CNC kontrol klavyesi çalışma tipi seçim anahtarı görünümü.	137
Şekil 3-45. CNC programlama klavyesinin sık kullanılan tuşları görünümü.	137
Şekil 3-46. Kontrol klavyesi üzerindeki eksen tuşları: Z (1), Y (2), X (3).....	138
Şekil 3-47. Mill55 tezgahında yüklü "O5" isimli NC programı ekran görüntüsü.....	139

Şekil 3-48. Kontrol klavyesi program kontrol tuşları görünümü.	140
Şekil 3-49. Kesici takım hız oranı ayarları görünümü: ilerleme (a), dönme (b).	141
Şekil 3-50. CNC kontrol klavyesindeki program başlat/durdur tuşları.	141
Şekil 3-51. CNC kontrol klavyesindeki tezgah kapısı açma/kapama tuşları.....	142
Şekil 3-52. Tezgahlardaki mengene ve ayna ile kontrol tuşları görüntüsü: mengene (a), mengene açma/kapama tuşu (b), ayna (c), ayna açma/kapama tuşu (d).....	143
Şekil 3-53. CNC kontrol klavyesindeki soğutma havası açma/kapama tuşu.	144
Şekil 3-54. Parçayı sıkıştırarak tutmak için tasarlanmış basit bir kaynak fikstürü [82] (a); pnömatik tutuculara sahip gelişmiş bir kaynak fikstürü görüntüleri [83]......	145
Şekil 3-55. Yeni kutu açılışı yapılmış olan RV4-FLM model robot görüntüsü.	145
Şekil 3-56. FMS100 bünyesindeki robotlarda kullanılan üç farklı el örneği görüntüleri; elektrik tahrikli tutucu el (a), pnömatik tahrikli tutucu el (b) ve lehimleme eli (c).....	146
Şekil 3-57. Kaynak robotu ve gaz altı kaynak eli (takımı) görüntüsü (a); araba camı taşımak için tasarlanmış vakumlu vantuzlu tutucu el görüntüsü (b).	146
Şekil 3-58. FMS100 Montaj hücresine robotun montaj görüntüleri: robotun üzerine sabitlenen masa (a), robotun masa üzerine sabitleniş biçimi (b).	147
Şekil 3-59. FMS100 sistemindeki CNC-Çifti istasyonunda bulunan ve doğrusal bir kızak üzerinde 3.000 mm strok ile hareket edebilen robot görüntüsü.	148
Şekil 3-60. Mitsubishi marka robotları programlamak için kullanılan RT Toolbox2 isimli yazılıma ait ekran görüntüsü.	149
Şekil 3-61. Montaj istasyonundaki robot ve çevre birimleri arası bağlantı şeması.	151
Şekil 3-62. Montaj istasyonu 6 numaralı robot programı algoritması gösterini (a), IPC programının 6 numaralı robot programı ile etkileşimli olan bölümü SDG gösterimi (b).	155
Şekil 3-63. GS komutu ile alt programa dallanma uygulamalarına örnek: aynı program içinde dallanma (a), farklı programlar arası dallanma (b).	161
Şekil 3-64. Örnek olarak doldurulmuş proje sihirbazı penceresi görüntüsü.	173
Şekil 3-65. Montaj istasyonu parametrelerini içeren proje sihirbazı penceresi görüntüsü.	174
Şekil 3-66. Boş bir COSIMIR Industrial yazılımı proje penceresi.....	175

Şekil 3-67. COSIMIR Industrial iletişim ayarları penceresi.	176
Şekil 3-68. Robot ile bağlantı başarıyla kurulduğunda ekrana gelen bilgi mesajı.	176
Şekil 3-69. “JOINT Jog” tipi hareket penceresi.	177
Şekil 3-70. “XYZ Jog” tipi hareket penceresi.	177
Şekil 3-71. “TOOL Jog” tipi hareket penceresi.....	178
Şekil 3-72. Robot uç noktasının A noktasından B noktasına hareketi.	179
Şekil 3-73. Robot eklem koordinat bilgilerinin COSIMIR yazılıma aktarılması.....	180
Şekil 3-74. COSIMIR Industrial projesindeki konum penceresi.....	180
Şekil 3-75. Örnek bir konum bilgisi penceresi.	182
Şekil 3-76. Bir engeli aşarak iki nokta arası robot hareketine örnek uygulama.	183
Şekil 3-77. Örnek uygulama için Movemaster Command dili ile geliştirilmiş robot programı.	184
Şekil 3-78. COSIMIR Industrial yazılımı ile robot program benzetimi menüsü.	186
Şekil 3-79. Robot projesinin program dosyasını robota yükleme penceresi.	187
Şekil 3-80. Robot projesinin konum dosyasını robota yükleme penceresi.	188
Şekil 3-81. RV-E2 model robot denetleyicinin ön taraftan görüntüsü.	189
Şekil 3-82. Program klasörü (Program Directory) penceresi görünümü.....	190
Şekil 3-83. Fabrika otomasyonu yapısal piramidi ve katmanları.	191
Şekil 3-84. COSIMIR Factory yazılımının alt yazılımları ve birbirleriyle ilişkileri. ...	194
Şekil 3-85. COSIMIR Factory yazılımında oluşturulabilecek örnek bir imalat hattı bileşenleri.	194
Şekil 3-86. BMW araba fabrikasında çalışan otonom taşıma sistemi [85].	195
Şekil 3-87. İmalat hücrelerinde iş parçaları ve paletleri taşımak için kullanılan robotik sistemler: 6 eksen endüstriyel robot (a), 2 eksenli ve 8 serbestlik dereceli Kartezyen tut/bırak sistemi (b).....	196
Şekil 3-88. İmalat kontrol sistemi bileşenleri.....	201
Şekil 3-89. COSIMIR Factory Control yazılımındaki kontrol yapısı.	204

Şekil 3-90. Endüstri 4.0 yaklaşımında kontrol yapısı.....	206
Şekil 3-91. COSIMIR Factory Control yazılımı geliştirme ortamı (IDE) ve bileşenleri.	208
Şekil 3-92. SIMULA sürücüsü ayar penceresi.	210
Şekil 3-93. COSIMIR Factory Control yazılımı boş süreç planı penceresi görünümü.	211
Şekil 3-94. Birden fazla başlangıç koşuluna sahip satırların görünümü.	212
Şekil 3-95. Örnek uygulamadaki “Mill_Workpiece” isimli süreç planı.	212
Şekil 3-96. Süreç planlarının yapısal durumu ve kontrol edilmesi.	213
Şekil 3-97. Süreç planında yer alan hataların “Mesaj Penceresi”nde gösterilmesi.	213
Şekil 3-98. Yerel ve yapısal değişken tanımlama görüntüsü.	214
Şekil 3-99. “View/Task Tool” (“F3” fonksiyon tuşu) penceresi görünümü.	215
Şekil 3-100. Süreç planının Proje Penceresi üzerinden başlatılması.....	216
Şekil 3-101. Önceden başlatılmış süreç planı ve aktif olan süreçlerin takibi.....	217
Şekil 3-102. Süreç planına hata ayıklama noktası (durak noktası - breakpoint) ekleme.	217
Şekil 3-103. Sürece ait yerel değişken değerlerinin değiştirilmesi.	218
Şekil 3-104. ABB marka robot ile RS232 haberleşme süreç planının bir bölümü.....	219
Şekil 3-105. COSIMIR Factory Control yazılımı projesine “Info” sürücüsü ekleme.	220
Şekil 3-106. “UserOutput” isimli cihaz ile “Info” sürücüsünün ilişkilendirilmesi.	220
Şekil 3-107. “HS_Milling” isimli süreç planı içeriği.	221
Şekil 3-108. “ChooseRadio” isimli kullanıcı arayüzünün görüntüsü.....	221
Şekil 3-109. “Simula” sürücüsünde, “Mill” cihazının “Milling” isimli görevine “HS_Milling” süreç planının atanarak haberleşme altyapısının oluşturulması.....	222
Şekil 3-110. “Mill_Workpace” süreç planında yapılacak değişiklikler.	222
Şekil 3-111. “Handshake Process Plan” penceresinden cihazı başlatma ve sonlandırma süreçler ile ilgili ayarların yapılması.	224
Şekil 3-112. Proje Penceresi üzerinden herhangi bir cihazın sonlandırılıp başlatılması.	224

Şekil 3-113. “.DISPLAY” fonksiyonu kullanımına örnek bir uygulama.....	227
Şekil 3-114. Hata mesajı penceresi.	228
Şekil 3-115. “.CALC” fonksiyonu kullanımına örnek kod satırları.	228
Şekil 3-116. “.CALC” fonksiyonu kullanımına örnek bir uygulama: asal sayı bulmaca.	229
Şekil 3-117. “.START” ve “.CALL” fonksiyonları kullanımına örnek süreç planı.....	229
Şekil 3-118. “.KILL” fonksiyonu kullanımına örnek süreç planı.	230
Şekil 3-119. “.CLAIM” ve “.RELEASE” fonksiyonları kullanımına örnek süreç planı.	231
Şekil 3-120. Süreç planı içerisindeki satır etiketlerinin sürecin sıralamasına etkisi. ...	232
Şekil 3-121. Satırlarda yürütülen görevlerin geri dönüş değerlerine göre dallanma örneği.	233
Şekil 3-122. “.CALL” fonksiyonu ile çağrılan bir süreç planından dönen değerin okunması.....	234
Şekil 3-123. “.CALL” fonksiyonu ile çağrılan bir süreç planından dönen değerin \$RES sistem değişkeni ile okunması.	234
Şekil 3-124. Bir süreç planına 2 adet parametre gönderilmesine yönelik kalıp örneği.	235
Şekil 3-125. “.WAITUNTIL” sistem fonksiyonu kullanımına örnek süreç planı.....	235
Şekil 3-126. “.WAITCHANGE” sistem fonksiyonu kullanımına örnek süreç planı. ...	236
Şekil 3-127. Süreç planının otomatik çalıştırılması ayarları.	237
Şekil 3-128. “.SPLIT” ve “.MERGE” sistem fonksiyonları kullanımına örnek süreç planı.	237
Şekil 3-129. Süreç planına görsel arayüz için “\$ProcVis” sürücüsünün eklenmesi. ...	238
Şekil 3-130. Boş görsel arayüze metin kutusu eklenmesi.	239
Şekil 3-131. Metin kutusuna ait özellikler penceresi görünümü.	239
Şekil 3-132. Görsel arayüz üzerinde metin kutusu görüntüsü.....	240
Şekil 3-133. Görsel arayüz üzerindeki metin kutusu içeriği ve değişken içeriği görüntüsü.	240
Şekil 3-134. Butona ait özellikler penceresi görünümü.	241

Şekil 3-135. Süreç planı içerisinde değişken değeri değiştirmeye yönelik örnek.	241
Şekil 3-136. “Example.lpj” isimli örnek proje açıldığında görüntülenen uygulamalar.	242
Şekil 3-137. “ProcVis Examples” isimli süreç planı ve yapılan değişikliklerin görüntüsü.	242
Şekil 3-138. Örnek uygulamaların çalıştırılabilmesini sağlayan kontrol penceresi.	243
Şekil 3-139. “Control” penceresi üzerindeki “I” butonuna ait özellikler.	243
Şekil 3-140. “ProcVis Examples” isimli süreç planı seçim bölümü.	244
Şekil 3-141. “Example_1” isimli süreç planı içeriği.	244
Şekil 3-142. Örnek uygulamalara ait görsel arayüz tasarım dosyalarının açılması.....	245
Şekil 3-143. Mill55 freze tezgahında işlenecek parçaya ait teknik çizim.	248
Şekil 3-144. RV-E2 model robotun merkezi koordinat sistemindeki (XYZ modu) hareketine yönelik örnek robot programı.	253
Şekil 3-145. “StartSubProc4” süreç planı görüntüsü.	258
Şekil 4-1. Bölüme basınçlı hava sağlayan eski kompresör görüntüsü (a), yerine getirilen yeni kompresör görüntüsü (b).	276
Şekil 4-2. Lehimleme hücresinde arızalandığı için yenisiyle değiştirilen Profibus FMS modülü görüntüsü.	277
Şekil 4-3. Lehimleme hücresinde arızalanan güç kaynağı (a), yerine takılan (b); Hidrolik-Baskı hücresinde arızalanan güç kaynağı (c), yerine takılan (d).	278
Şekil 4-4. Stok hücresi denetleyicisi olan IPC altına yerleştirilmiş soğutma fanları. ..	279
Şekil 4-5. Montaj hücresindeki robotta takılı durumdaki tutucu el görüntüsü (a), yerinden sökülmüş durumdaki tutucu el görüntüsü (b).	279
Şekil 4-6. 1E-HM01 model robotik tutucu elin sökülmüş haldeki görüntüsü.	280
Şekil 4-7. 1E-HM01 model robotik tutucu ele güç veren DC motorun iç görüntüsü (a); tutucu elin boyutlarını ifade eden dış görüntüsü (b).	280
Şekil 4-8. CNC-Çifti hücresindeki robotun 3. eksen servo motoru ve enkoder bölümü (a), enkoder bölümü içerisindeki kırık optik disk ve elektronik okuyucu devresi (b)..	281
Şekil 4-9. CR1 robot denetleyici iç görüntüsü (a) ve arızalı servo sürücü kartı (b).	282

Şekil 4-10. FMS100 Görüntü İşleme hücresindeki robotun eklemleri yağlanırken çekilmiş görüntüler.	283
Şekil 4-11. Çarpma nedeniyle kırılmış havya ünitesi görüntüsü (a), havya kontrolörü içindeki PLC’nin yazılımını değiştirmek için sökülmüş halinin görüntüsü (b).	284

TABLO LİSTESİ

Tablo 1-1. YÖK veritabanındaki lisansüstü çalışmaların üniversitelere göre dağılımı. 25	25
Tablo 1-2. YÖK veritabanındaki lisansüstü çalışmaların bölümlere göre dağılımı. 25	25
Tablo 2-1. Stok hücreesindeki 48 adet rafta bulunan iş-parçaları dizilimi. 41	41
Tablo 2-2. FMS100 bünyesinde tanımlı imalat süreçleri ve açıklamaları..... 51	51
Tablo 2-3. FMS100 sisteminde kullanılan iş-parçalarına ait seri numaraları..... 83	83
Tablo 3-1. Çeşitli tarihlerde hizmete giren AS-i sürümleri arası karşılaştırma tablosu. 119	119
Tablo 3-2. Robot sayısal girişlerine yapılan elektriksel bağlantılar. 153	153
Tablo 3-3. Robot sayısal çıkışlarına yapılan elektriksel bağlantılar..... 153	153
Tablo 3-4. İmalat Veritabanı içindeki tablo ve alanlar. 198	198
Tablo 3-5. İmalat planlama aşamasında sorulması gereken sorular ve örnek cevaplar.199	199
Tablo 3-6. Örnek uygulamaya ait konumlar ve açıklamaları. 253	253
Tablo 3-7. 2016-2017 yılı bahar döneminde EİS dersini alan öğrencilerin başarı durumları. 265	265
Tablo 3-8. Değerlendirme testi sonuçları. 266	266

1. GİRİŞ

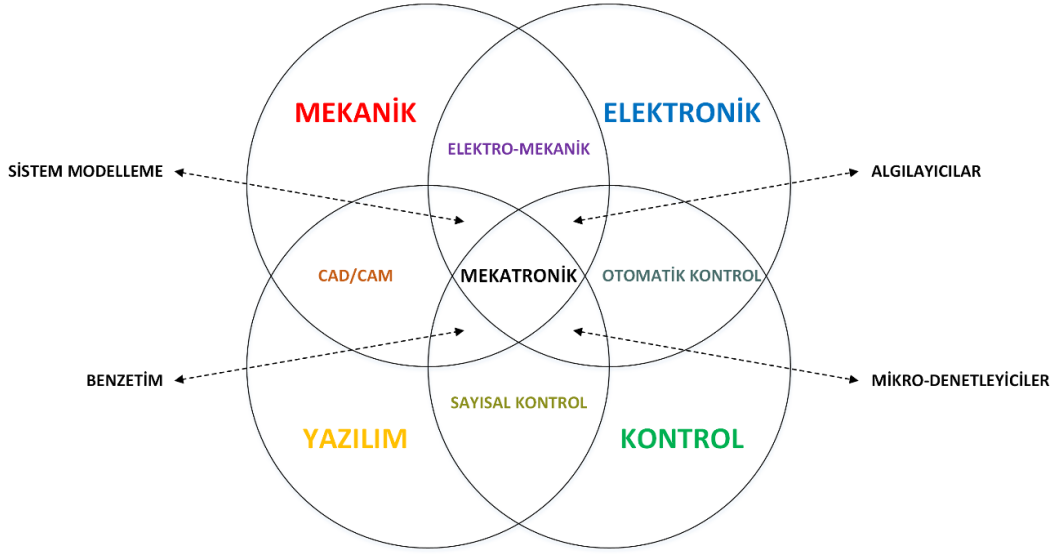
Mekatronik kelimesi, mekanizma kelimesinin “mek-” ile elektronik kelimesinin “-tronik” bölümlerinin bileşiminden türetilmiştir. Yapı itibariyle Mekatronik sistemler, mekanizmanın nerede bittiği ve elektroniğin nerede başladığının net biçimde ayırlamadığı, organik olarak iç içe geçmiş teknolojileri ifade eder [1]. Mekatronik kavramı ilk olarak 1969’da Japon Yaskawa Electric Corporation firmasında çalışan bir elektrik mühendisi olan Ko Kikuchi tarafından ortaya atılmış ve 1972 yılında firma tarafından tescil edilmiştir. Yaskawa firmasının 1982 yılında tescil haklarından vaz geçmesiyle birlikte, Mekatronik kavramının kullanımı giderek yaygınlaşmıştır [2, 3].

Mekatronik; bir ürünün geliştirilme aşamasında mekanik, elektronik, kontrol ve yazılım alanlarına ait yapılan analitik hesaplamalarda; ürünün tasarım ve üretim aşamalarının da göz önüne alınması yaklaşımıdır. Böylelikle elektromekanik yapılar içeren modern ürünleri oluşturan eyleyici, sürücü ve mekanizmalar gibi alt sistem bileşenlerinin tasarım ve üretiminde en uygun performansı yakalamak amaçlanmaktadır [4].

Literatürde Mekatronik bir sistem için verilen temel örneklerden olan fotokopi makinasının yapısı ve bileşenleri, teknik açıdan irdelendiğinde:

- Optik ve kağıt üzerine baskı ünitelerini işletebilmek için mekanik yapılara;
- Mekanik yapılara hareket veren eyleyicilere;
- Mekanik yapıların işleyişini gözleyen algılayıcılara;
- Algılayıcılardan gelen geri besleme bilgisine göre eyleyicileri süren elektronik devrelere;
- Elektronik devrelerin kapalı döngü içerisinde davranışının belirlenebilmesini sağlayan kontrol algoritmalarına;
- Kontrol algoritmalarının, elektronik devrelerde bulunan mikro-denetleyiciler tarafından yürütülebilmesini sağlayan gömülü sistem yazılımlarına;
- Kullanıcı ile makina arası etkileşimi sağlayan bilgisayar yazılımlarına sahip olduğu görülmektedir.

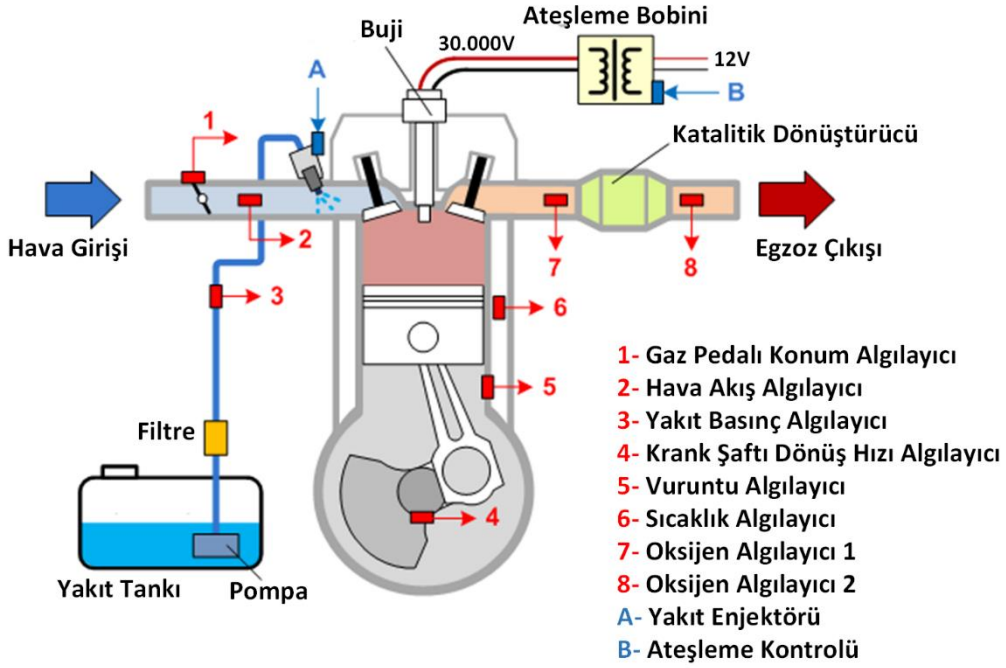
Bu yönden değerlendirildiğinde, günümüzde yaygın olarak kullanılan fotokopi makinasının Şekil 1-1’deki gibi mekanik, elektronik, kontrol ve yazılım teknolojilerinin iç içe geçmesiyle oluştuğu görülmektedir.



Şekil 1-1. Mekatronik teknolojisinin bilim alanları ile ilişkisi.

Geçmişte tamamen mekanik olarak işleyen sistemlere günümüzde gelişen elektronik ve bilgisayar teknolojisi hızla dahil olmaktadır. Artık tamamen mekanik veya tamamen elektronik bir sistemden söz etmek oldukça güçtür. Bu yaklaşımın arkasında yatan en büyük sebep, mekanik iş yapan sistemlerdeki verimliliği, çalışma kararlılığı ve hassasiyeti artırma ihtiyacıdır. Böylelikle Global dünya nedeniyle pazarda rekabet edebilecek ürünler üretebilme ve işletmelerin ayakta kalma çabasıdır.

Örnek olarak bir benzinli araba motoru ele alındığında, önceleri motorun çalışmasına yönelik tüm ayarlar, vidalar aracılığıyla ve ustalık becerisiyle yapılmaktaydı. Günümüz motorlarında ise motor devrini ölçen manyetik algılayıcılar, egzoz gazından yakıtın yanma verimini ölçen kimyasal algılayıcılar, vb. birçok algılayıcı kullanılmaktadır. Algılayıcılardan gelen sinyaller “engine control unit” (ECU) adı verilen mikro-denetleyicili elektronik devrelere iletilmektedir. Motorun ateşleme zamanı, yakıt/hava karışım oranları, vb. değerleri de yine bu devreler içerisine yazılım ile kodlanan kontrol parametreleri tarafından belirlenmektedir [5]. Böylelikle mekanik enerji üreten bir motorun çalışma ayarları, çeşitli mekanik ayar vidaları kullanmak yerine, mikro-denetleyici yazılımı içerisindeki bir takım parametreleri değiştirmek yoluyla yapılabilmektedir. Bu nedenle günümüz arabalarının içten yanmalı motorları da mekatronik sistemlere iyi bir örnek olarak verilebilir. Şekil 1-2’de [6] günümüz benzinli araç motorlarında bulunan bazı temel bileşenler görülmektedir.



Şekil 1-2. Modern bir benzinli araç motoru elemanları [6].

Robot denilince ilk olarak başı, kolları, gövdesi ve ayakları bulunan insansı mekanik yapılar akla gelmektedir. Halbuki sanayi robotları, genellikle tek bir koldan oluşan ve endüstride kaynak, montaj, kesme, boyama gibi belirli görevleri tekrar tekrar aynı başarımla yerine getirmek için tasarlanmış olan çok eklemlili mekanik yapılardır. Temel mekanik bölümlerini manipülatörler, dişli ve çarklar, tutucular oluşturur. Pnömatik/hidrolik silindirler veya elektrik motorları gibi eyleyici elemanlar, bu mekanik yapılara hareket kazandırır. Robotların yaptıkları işlerin önemi dikkate alındığında, hareketi sağlayan eyleyicilerin hassas ve eş-zamanlı kontrol edilmesi gerektiği görülecektir. Robotun sahip olduğu eklem sayısına (serbestlik derecesi) göre, eş zamanlı hareketin (enterpolasyon) sağlanması için tüm bileşenlerin hareket koordinatları tek tek hesaplanmalı ve hareket sırasında ölçülmelidir. Ayrıca robotun çalışması esnasında yürüteceği görevler kullanıcılar tarafından belirlenebilmeli ve değiştirilebilmelidir. Bu nedenle mikro-denetleyici yapıları elektronik devre olan robot kontrolörü, kullanıcının yüklemiş olduğu program işleyişi ve mekanik bileşenlerden gelen geri-besleme sinyallerini kullanarak eyleyicilerin kontrolü için gerekli hesaplamaları hızlı biçimde yapar. Bu yönüyle incelendiğinde robot da, mekatronik bir sistem için verilebilecek güzel örneklerdendir. Şekil 1-3’de CAM kodları ile ahşap üzerinde kesme (frezeleme) işlemi yapan 6 eksen bir endüstriyel robot görülmektedir.



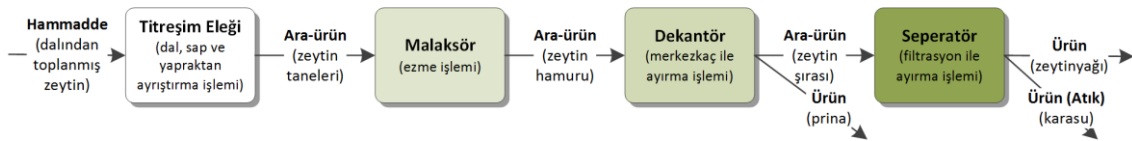
Şekil 1-3. CAM kodları ile ahşap üzerinde kesme işi (freze) yapan robot görüntüsü.

Endüstride üretim ve imalat kavramları anlam bakımından birbirleriyle sıklıkla karıştırılabilmektedir. Üretim (production) her türlü doğal kaynaktan faydalanarak ihtiyaç duyulan ara-ürün veya ürünleri elde etme sürecidir [7]. Topraktan ham petrol elde edilmesi veya zeytinden yağ çıkarılması üretime verilebilecek güzel örneklerdendir. İmalat (manufacturing) ise, önceden işlenerek elde edilmiş ara-ürün/ürünlerin işlenmesi sonucunda, daha değerli veya fazla sayıda, farklı ara-ürün/ürünler elde etme sürecidir. Ham petrolün ara ürünlerinden biri olan naftadan plastik hammaddesi poli-vinil-klorür (PVC) imal edilmesi, zeytinyağından sabun yapılması ise imalata örnek olarak verilebilir. Her ne kadar üretim ile imalat özünde farklı kavramlar olsa da, literatürde birbirleri yerine kullanılabilmektedir.

İktisatçılar, üretimi bireylerin ihtiyaçlarını karşılamak üzere yürütülen faaliyetler ile malzemeleri elde etme süreci olarak tanımlarlar. Bu süreç genel olarak ham maddelerin hazırlanması, temini, ara ürünlerin elde edilmesi ve montajı aşamalarından oluşur [8, 9]. Endüstriyel açıdan bakıldığında; üretim ve imalat süreçleri farklı ölçütlere göre tanımlanabilir. Çıktıları göz önüne alındığında, üretim ve imalat süreçleri sürekli (continuous), ayrık (discrete) ve yığın (batch) olmak üzere üç sınıfa ayrılabilir [10].

Sürekli üretimde çıktı yine sürekli bir ürün akışıdır. Hammadde süreç otomasyonunun bir ucundan girer. Üretim sisteminin çalışmasına ara verilmeden çeşitli işlemlerden geçirilerek, diğer uçtan ürün olarak çıkar. Özetle ham madde, geri dönüştürülemeyecek şekilde yapısal ve kimyasal değişikliklere uğratarak ürün elde edilir. Süreç otomasyonu açısından irdelendiğinde, malzeme ve enerji akışı sürekli olduğu için tampon alan

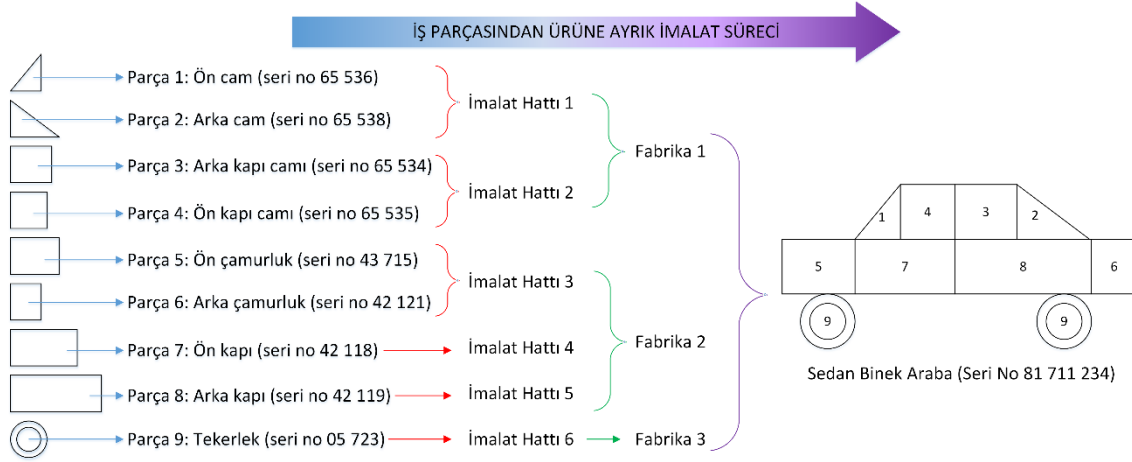
kullanımı söz konusu değildir. Bu da üretim sisteminin başlatma, normal operasyon, kapatma ve acil durum senaryolarının kontrol otomasyonunu daha da güçleştirir [11]. Sürekli üretime verilebilecek en iyi örneklerden biri, aşamaları Şekil 1-4’de şema halinde özetlenmiş olan zeytinyağı üretimidir. Halk arasında “kontinü sistem” olarak adlandırılan bu üretim sisteminde girdi (hammadde), dalından toplanmış zeytinlerdir. Sırasıyla titreşim (vibrasyon) eleğinde zeytin taneleri dal, sap ve yapraklardan ayrılarak basınçlı suyla yıkanır. Sonrasında malaksör tarafından zeytin taneleri ezilerek hamur haline getirir. Ara ürün olarak elde edilen zeytin hamuru, pompa aracılığıyla dekantöre iletilir. Merkezkaç (santrifüj) yöntemiyle yağ ile posa (prina) birbirlerinden ayrılır. Son olarak ayırıcıda (seperatör) yağ tortularından ve sudan arındırılarak zeytinyağı elde edilmiş olur [12]. Bu üretim sürecinin çıktılarından olan prina, ara-ürün olarak yeniden işlenerek sabun endüstrisinde ham madde olarak veya kömür gibi farklı yakıtlarla karıştırılarak enerji üretiminde kullanılabilir [13]. Bunun dışında sürekli üretim kimya/petro-kimya endüstrilerinde yaygın olarak tercih edilir.



Şekil 1-4. Zeytinyağı üretimi aşamaları.

Bir ürünü meydana getiren bileşenlerin rastgele zamanlarda birbirlerinden bağımsız olarak, bazı durumlarda farklı tesislerde, üretilerek sonradan bir araya getirilmesine ise ayırık (kesikli) üretim denir. Bir başka ifadeyle, üretim yapılan tesisin sürekli aynı ürün için kullanılmadığı üretim biçimidir de denebilir [8]. Ürünü meydana getiren bileşenler (iş parçaları) üretim hattı boyunca şekilsel, dayanımsal veya farklı parçalar eklenmesi gibi bir dizi fiziksel işlemde geçirilir. Bu sırada bazı kimyasal değişimler de gözlenebilse de, yine de ayırık üretimde gerçek anlamda kimyasal bir süreç söz konusu değildir. Genel olarak mekanik ve elektronik ürünlerin imalatında kullanılan bir yöntemdir. Üretim sürecinde tampon alanlar ve üniteler kullanılabildiği için süreç otomasyonu sadece üretimdeki veri akışının bütünleştirilmesine odaklanır. Bu nedenle sistemin teknik olarak kontrolü sürekli üretime göre daha kolaydır [11]. Ayırık üretim sonucunda elde edilen ürün, sürekli üretimden farklı olarak, genellikle kendisini oluşturan parçalarına ayrılabilir niteliktedir. Karışıklığı önlemek için Şekil 1-5’deki örnekte görüldüğü üzere üretilen her

bir parça (iş parçası) özel bir seri numarası ile etiketlenir [10]. Araba, uçak, cep telefonu ve mobilya üretimi örnek olarak verilebilir.



Şekil 1-5. İş parçasından ürüne ayrik imalat süreci, örnek bir araba imalatı.

Yığın üretimde ayrik üretimde olduğu gibi hammaddeler bir dizi işlemde geçirilerek nihai ürün elde edilir. Farklı olarak yığın üretimde ürünler parçalarına ayrılabilir yapıda değildir. Sürekli üretimde olduğu gibi üretimde kimyasal süreçler yer alır ve üretim çıktısı homojen yapılı, geri döndürülemeyecek şekilde işlenmiş bir yığın üründür. Bu yönleriyle irdelendiğinde yığın üretim hem sürekli hem de ayrik üretim özellikleri gösterir [10]. Yığın üretimde bazen ayrik üretimde olduğu gibi birçok farklı ürün aynı tesis içerisinde ve eş zamanlı olarak üretilebilir. Yığın üretimde elde edilecek ürün reçete ile tanımlanır. Diğer bir ifadeyle, ürünün elde edilebilmesi için kullanılacak bileşenler ve katkı oranlarının tanımlandığı dosyaya reçete denir. Yığın üretim süreci ve otomasyonu oldukça karışıktır. Aynı tesiste farklı zamanlarda üretilen ürünlerin yapısal özellikleri az da olsa farklılık gösterebilir. Örneğin yoğurttaki olduğu gibi aynı markanın aynı tip yoğurdu, farklı zamanlarda farklı tatlarda olabilir. Bu nedenle üretim çıktılarını birbirlerinden ayırmak için parti (lot) numarası kullanılır. Aynı partide üretilen ürünlerin fiziksel ve kimyasal özellikleri de aynıdır denilebilir. ISA-S88.01 standardında tanımlanmış olan yığın imalat kimya, gıda, ilaç, kozmetik ve biyoteknoloji endüstrilerinde yaygın olarak kullanılır [14, 15].

Üretim ve imalat sistemleri sürekli değişen pazar ihtiyaçlarını karşılayacak şekilde yönetilir, genişleyebilir ve güncellenebilir yapıda olmalıdır. Bunun sağlanabilmesi için birbirleri ile eş veya ayrik zamanlı işleyen makinalar, robotlar, taşıma sistemleri,

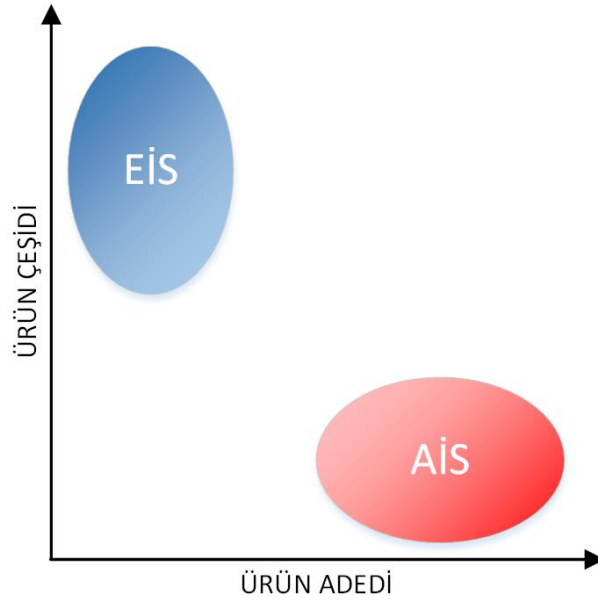
programlanabilir mantıksal denetleyiciler (PLC) ve bilgisayarlar gibi birçok bileşenin belirli bir sıra içerisinde koordineli biçimde yönetilmesi gereklidir. Bu amaçla sıralı (sequential) kontrol yöntemi geliştirilmiştir. Yöntem, üretimi makul düzeye indirgenmiş alt üretim adımlarına bölerek ve adımları belirli bir sırayla yürüterek uygulanır. Yöntemin uygulanmasında, farklı seviyelerde çalışan imalat cihazlarının ayrı ayrı programlanması gereklidir. Örneğin üretim sistemindeki robot, PLC, CNC gibi cihazlara yarı programlar yazılır. Grafcet, tüm üretim sürecinin işleyişini ve sıralamasını tanımlamada kullanılan yaygın gösterimlerden biridir. Uluslararası IEC 848 ve IEC 1131-3 standartlarında tanımlanmış olması sebebiyle endüstride kabul görmüştür. Grafcet endüstride Sequential Function Charts (SFC) olarak da adlandırılmakta olup, özellikle üretim sürecini yöneten PLC cihazlarının programlanmasında kullanılır [15].

Uluslararası rekabet, günümüzde küresel pazarda mücadele edebilecek düzeyde kaliteli ürünler geliştirilmesi ihtiyacını arttırmaktadır. Günümüzde artan rekabet koşulları nedeniyle endüstriyel gelişimin hızı da artmaktadır. Bu durum imalatçıları sadece rekabet amaçlı olmamakla birlikte ayakta kalabilmek için bile sürekli gelişen kalitede üretim yapmaya zorlamaktadır. Pazara sunulan ürünlerin öngörülen ömrü azaldıkça ve yüksek kaliteli ürün ihtiyacı arttıkça; ürün gelişimine daha çok önem verilmeye başlanmıştır. Daha verimli imalat, yönetim ve pazarlama yöntemleri kullanılarak endüstriyel işletmelerin rekabet kapasiteleri de artmıştır [11]. Bu durum mekanik ve elektronik ürünlerde geleneksel imalat yöntemlerinin bırakılarak daha esnek, düşük maliyetli, kaliteli ve hızlı imalatı mümkün kılan Bilgisayar Tümlşik İmalat (Computer Integrated Manufacturing - CIM) sistemlerinin geliştirilmesine sebep olmuştur.

CIM, ürüne ait tüm imalat aşamalarının bilgisayar aracılığıyla yönetildiği bir imalat yöntemidir. CIM sayesinde ayrık, geleneksel üretim yöntemleri bilgisayar aracılığıyla birleştirilir. Ayrık yürütülen işlemlerin birbirleriyle bilgi alışverişi yapılarak koordineli çalışmaları sağlanır. Böylelikle üretim tesisleri hızlı ve hataları asgari seviyede olacak şekilde kurulabilir, tesise ait imalat otomasyon sistemi kolaylıkla oluşturulabilir [16].

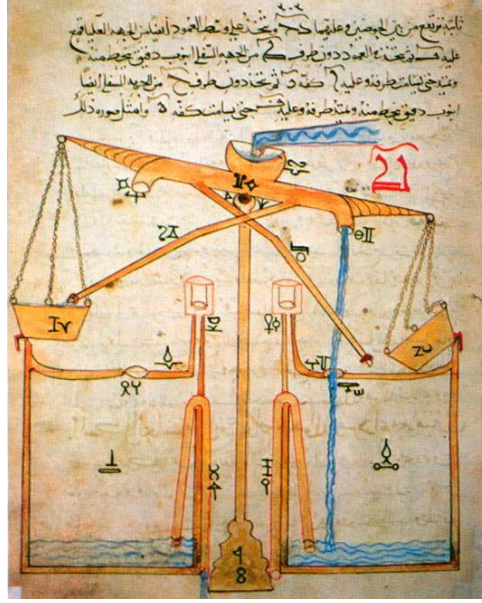
CIM kavramı, sadece belirli bir ürünü üretmeye yönelik olarak tasarlanmış olan Adanmış İmalat Sistemleri (AİS) ve birbirinden farklı ürünleri aynı sistem içerisinde üretebilecek yapıda tasarlanmış olan Esnek İmalat Sistemleri (EİS) olmak üzere ikiye ayrılır. Aynı tip ürün uzun vadede yüksek adetli üretilmek istendiğinde AİS tercih edilir. Böylelikle hem

retim sisteminin kurulum maliyeti hem de retim kendi maliyeti olduka dk seviyede tutulabilir. EİS ise bir retim sisteminde, dk maliyetli ve kk aplı deęiiklikler yapılarak birbirleri ile ortak ynleri bulunan farklı rnlerin retilmesi istendięinde tercih edilir. retim eitlilięinin bol olduęu tesislerde maliyet avantajı saęlar [17]. ekil 1-6’da rn adedi ve eidine gre AİS ile EİS arasındaki iliki grafięi gsterilmektedir.



ekil 1-6. rn eidi ve adedine gre AİS ve EİS arasındaki iliki grafięi.

Anadolu’da yaamı olan Ebu’l İZ El Cezeri dnya bilim tarihi aısından robotik ve sibernetik alanlarında alımalar yapan ilk bilim adamlarından biridir. Genel olarak hidro-mekanik basın etkili denge kurma ve hareketli sistemler kullanarak saat ve gnmz insansı robotlarının atası kabul edilebilecek “otomatik hizmeti” gibi uygulamalar gelitirmitir. Yaptıęı 50’den fazla alımayı “Mekanik Hareketlerden Mhendislikte Faydalanmayı İeren Kitap” adlı eserinde toplamıtır [18]. ekil 1-7’de El Cezeri tarafından tasarlanmı olan otomatik alıan su makinasına ait izim grlmektedir.



Şekil 1-7. Ebu'l İz El Cezeri tarafından tasarlanmış otomatik çalışan su makinasına ait çizim.

Buna rağmen modern çağda endüstriyel devrimin başlangıcı 1769 yılında James WATT'ın geliştirmiş olduğu topuzlu mekanik governör (flyball governor) olarak kabul edilir. Topuzlu mekanik governör yapı itibariyle geri beslemeli, kapalı çevrim otomatik kontrol sistemine ilk örneklerden biridir. Şekil 1-8'de kömürle çalışan topuzlu mekanik governöre sahip ilk buhar makinası örneklerinden biri görülmektedir.

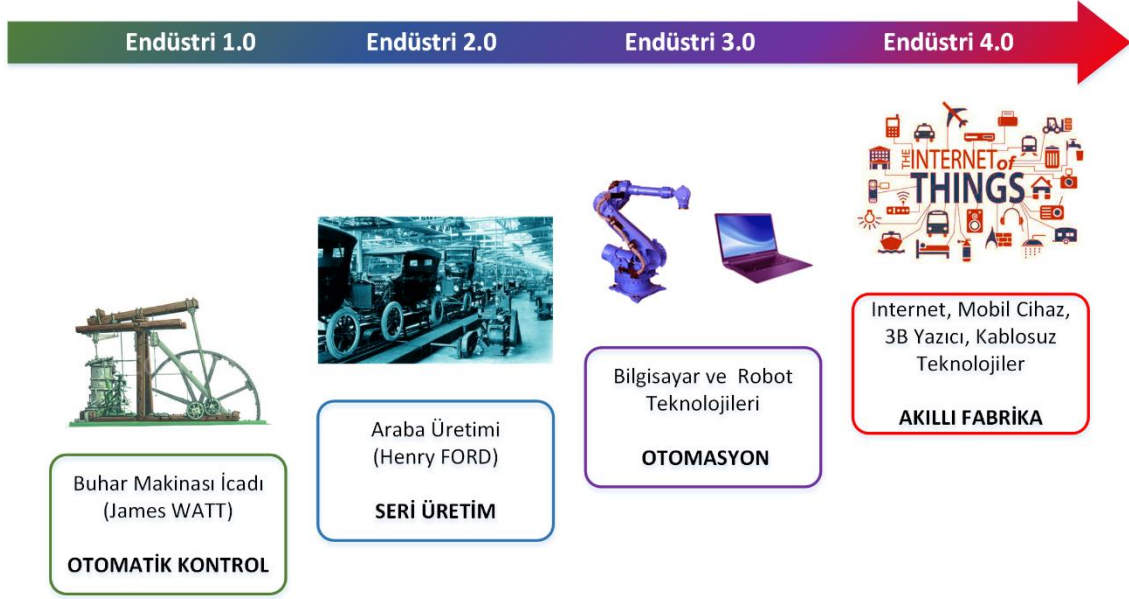


Şekil 1-8. Universidad Politecnica de Madrid'de bulunan topuzlu mekanik governöre sahip buhar makinası.

Henry Ford'un 1908 yılında duyurduğu "Model T Tourer" model arabasının 1914 yılından itibaren seri üretime geçmiş olması da ikinci endüstriyel devrim olarak kabul edilir [19]. Ford, o devirde devrim sayılan imalat hattı fikrinin temel ilham kaynağının mezbahalardaki buharlı kesim bantları olduğunu ifade etmiştir [20, 21].

İkinci dünya savaşı ve ABD ile Rusya arasında yaşanan uzay yarışının baskısıyla yarı iletken ve mantıksal (lojik) elektronik konularında hızlı gelişmeler kaydedilmiştir. Bunun sonucunda 1960 yılların ortalarında ilk bilgisayarlar bilimsel ve askeri alanlarda kullanılmaya başlanmıştır. İlerleyen yıllarda bilgisayarın beyni olan mikroişlemcilerin endüstrideki zor şartlara uyumlu hali olan mikro-denetleyicilerin geliştirilmesiyle birlikte programlanabilir mantıksal denetleyiciler (Programmable Logic Control - PLC) üretilmiştir. PLC'lerin endüstride kullanılmaya başlanması ve kesme, kaynak, boyama, vb. işlerde insan etkisini asgari düzeye indiren robotların kullanımına başlanması da üçüncü endüstri devrimi olarak kabul edilir.

Günümüzde EİS'lerin çok daha etkileşimli ve esnek sürümü olan akıllı fabrikalardan söz edilmektedir. Başını Almanya, ABD ve Japonya'nın çektiği gelişmiş ülkeler Endüstri 4.0 kavramı ortaya atmaktadır. Kavram, temelde üretim sürecinde yer alan tüm cihazlar, müşteri sipariş süreçleri ve tedarik zincirinin Internet üzerinden birbirleri ile iletişim halinde olması prensibine dayanır. Üretim ile ilgili girdi/çıkı verileri bir merkezde toplanır. Toplanan veriler, büyük veri (Big Data), veri madenciliği yöntemleriyle analiz edilerek üretimdeki maliyet hesabının keskinliği ve verimlilik arttırılmaya çalışılır. Böylelikle yüksek teknolojiye sahip ve kişiselleştirilmiş ürünlerin üretim maliyetlerinin azaltılması hedeflenir. Nesnelerin Interneti, Bulut Bilişim, Big Data, RFID, 3 boyutlu yazıcı ve endüstriyel robotlar bu "dördüncü endüstriyel devrim"in ardındaki lokomotif teknolojilerdir. Şekil 1-9'da endüstriyel devrimin tarihsel aşamaları sembolik olarak gösterilmiştir.



Şekil 1-9. Tarihsel endüstriyel devrim aşamaları.

1.1. Amaç ve Gerekçe

Marmara Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü bünyesindeki CIM laboratuvarında, eğitim amaçlı olarak imal edilmiş olan EİS (FMS100) bulunmaktadır. Sahip olduğu üretim istasyonu çeşidi ve teknolojisi bakımından FMS100, eğitim amaçlı olarak geliştirilmiş dünya genelindeki en büyük ve geniş kapsamlı esnek imalat sistemlerinden biridir. FMS100, modern bir fabrika otomasyonu kapsamında bulunabilecek robotlar, Programlanabilir Mantıksal Denetleyiciler (PLC), hidrolik/pnömatik eyleyiciler, CNC torna ve freze, değişik tipte algılayıcılar, SCADA ve MRP yazılımı, kameralı üretim kalite kontrol ünitesi ile endüstriyel iletişim protokolleri gibi birçok temel bileşeni bünyesinde bulundurur. FMS100, eğitim amaçlı geliştirilmiş bir sistemde endüstriyel alandaki temel bileşenleri içermesi bakımından oldukça önemli ve değerlidir.

Yapılan bu çalışmanın amacı, FMS100'ün Mekatronik Mühendisliği bölümü bünyesinde hem lisans hem de lisansüstü eğitimde etkin olarak kullanımının sağlanması; bu doğrultuda eğitim müfredatının hazırlanmasıdır. Hazırlanan müfredat ve eğitim materyalleri sayesinde FMS100 sistemi; Bilgisayar Destekli İmalat, Robotiğe Giriş, Robot Tekniği, Esnek İmalat Sistemleri gibi lisans ve lisansüstü derslerine katkı sağlayacaktır. Sistemin sahip olduğu MRP yazılımı ve diğer altyapıları sayesinde Endüstri Mühendisliği bölümü dersleriyle de tümleşik eğitimler verilmesi mümkündür.

Ayrıca Profibus, AS-i gibi endüstriyel haberleşme protokolleri ile Mitsubishi robotların kullanımı gibi konularda pratik Türkçe bilgi dokümanlarının da hazırlanması amaçlarımız arasındadır.

Mekatronik Mühendisliği Bölümü CIM laboratuvarında bulunan elektrik tesisatındaki aksaklıklar nedeniyle FMS100 bünyesindeki bazı bileşenlerde zamanla arızalar meydana gelmiştir. Bu nedenle de uzunca bir süre FMS100 sistemi bir bütün olarak çalıştırılamamış, üretim istasyonları kısmen ve birbirlerinden bağımsız olarak çalıştırılmak suretiyle eğitim faaliyetleri yürütülmüştür. Yapılan bu çalışma, büyük maliyetler sonucunda üniversitemize kazandırılan FMS100 sisteminin eksikliklerinin giderilmesi ve ihtiyaç duyulan Türkçe eğitim dokümanları hazırlanması sayesinde özellikle Mekatronik Mühendisliği bünyesinde olmak üzere Endüstri Mühendisliği gibi diğer mühendislik dallarında da daha etkin olarak kullanılmasını sağlaması bakımından önemlidir.

1.2. Erişilmesi Planlanan Hedefler

Mekatronik Mühendisliği bölümü lisans seviyesindeki öğrenciler, son sınıfa gelene kadar Bilgisayar Destekli Tasarım, Bilgisayar Destekli İmalat, Robot Tekniği, Programlanabilir Lojik Kontrolörler, Hidrolik ve Pnömatik Sistemler ve Algılayıcılar gibi Mekatronik Mühendisliği alanındaki temel dersleri almış olmaktadır. FMS100 CIM sistemi, bu temel derslerin içeriğinin çoğuna ait uygulamayı bünyesinde barındırmaktadır. Bu sayede öğrencilerin lisans eğitimleri boyunca almış oldukları bu temel derslerin, gerçek hayattakine benzer bir fabrika otomasyon sistemi üzerinde kullanımını görmesi; uygulama becerilerini arttırması ve bilgilerini pekiştirmesi hedeflenmiştir.

Yapılan bu çalışma ile öğrencilerin bilgi ve becerisini arttırmaya yönelik erişilmek istenen somut hedefler şu şekilde sıralanabilir:

- İmalat kavramı ve çeşitleri hakkında temel bilgi sahibi olmak,
- Esnek İmalat ve fabrika otomasyon hiyerarşisi hakkında bilgi sahibi olmak,
- FMS100 sistemi ve imalat hücrelerini tanımak,
- Endüstriyel algılayıcı tiplerini ve kullanım alanlarını öğrenmek,
- Süreç programlamada (process planning) kullanılan sıralı programlama mantığını (sequential programming) öğrenmek,
- PLC programlama becerilerini arttırmak,

- Robot kullanma ve programlama becerilerini arttırmak,
- PLC ile robot etkileşiminin nasıl kurulduğunu öğrenmek,
- İmalat hücresi bünyesinde bulunan robot ile çevresel eyleyiciler ve algılayıcılar arası bağlantıları öğrenmek,
- CNC torna ve freze kullanım becerisini arttırabilmek,
- SCADA yazılımı üzerinden bir PLC programı ile etkileşim sağlayabilmek (çalıştırmak, durdurmak, bekletmek, çıktılarını ve hata mesajlarını görüntülemek),
- SCADA yazılımı üzerinden bir robot programı ile etkileşim sağlayabilmek (çalıştırmak, durdurmak, bekletmek, çıktılarını ve hata mesajlarını görüntülemek),
- SCADA yazılımı üzerinden CNC torna ve freze programı ile etkileşim sağlayabilmek (çalıştırmak, durdurmak, bekletmek, çıktılarını ve hata mesajlarını görüntülemek),
- SCADA yazılımı ile fabrika otomasyon sistemini yönetmek, görüntülemek, temel seviye imalat kaynak planlama (MRP) hakkında bilgi sahibi olmak,
- FMS100 bünyesinde hazır olarak yazılmış süreç planlarını çalıştırma ve yeni bir süreç planı oluşturma becerisini kazanmak,
- Endüstriyel haberleşme protokolleri (RS232, Profibus DP, Profibus FMS, AS-i Bus) hakkında bilgi sahibi olmak ve uygulamalar yapmak.

1.3. Literatür Araştırması

Yapılan çalışmanın özünü oluşturan bilgisayar destekli imalat ve esnek imalat kavramları ile eğitim alanında yapılan uygulamaları hakkında literatürde yapılan araştırmalar sonucunda aşağıdaki kaynaklara ulaşılmış olup, kaynaklar çalışmanın amaç ve hedefleri doğrultusunda incelenerek aşağıda özetlenmiştir. Çalışma özetleri sunulurken öncelikle YÖK veritabanında bulunan doktora ve yüksek lisans tez çalışmaları irdelenmiştir. Ardından yurtdışı tez çalışmaları, makaleler ve konferans bildirileri sıralanmıştır.

1.3.1. YÖK veritabanında “CIM”, “bilgisayar bütünleşik imalat” ve “bilgisayar tümleşik imalat” kelimeleri tarama sonuçlarında listelenen doktora ve yüksek lisans tez çalışmalarından, sunulmakta olan çalışma ile alakalı olanlar şunlardır:

[22]'de detayları verilen ve 2011 yılında sunulmuş olan doktora tez çalışmasında özetle; değişen müşteri ihtiyaçlarına göre rekabet avantajı sağlayan gerçek zamanlı sınırlı kapasiteli üretim planlama ve kontrol yapısı önerilmiş, benzetim ortamında test edilmiş

ve sonuçlar çok değişkenli varyans analizi ile test edilmiştir; önerilen kontrol yapısının imalat performansı üzerinde etkili olduğu sonucuna varılmıştır.

[23]'de detayları verilen ve 2006 yılında sunulmuş olan doktora tez çalışmasında özetle; entegre bir imalat sistemi kurulması için hiyerarşik olarak ilişkilendirilmiş değişik imalat fonksiyonlarından oluşan REMIMS adı verilen bir referans model geliştirilmiş, model bünyesindeki fonksiyonların birbirleri ile haberleşmesini sağlayan bir bilgi protokolü oluşturulmuştur.

[24]'de detayları verilen ve 2003 yılında sunulmuş olan doktora tez çalışmasında özetle; BDX sistemlerinden meydana gelen Bilgisayar Bütünleşik İmalat (BBİ) sisteminin geliştirilmesi ve analizinin yapılmasında çok kıstasla karar verme tekniği olan Analitik Hiyerarşi Prosesi (AHP) tekniği kullanılmış, uygulanabilirliğinin görülmesi açısından örnek bir firmanın üretim organizasyonu modellenerek benzetim modeli oluşturulmuş ve sonuçlar incelenmiştir.

[25]'de detayları verilen ve 2001 yılında sunulmuş olan doktora tez çalışmasında özetle; mobilya endüstrisinde olan imalat firmalarında kullanılabilecek ana üretim çizelgeleme modeli işlemleri için yeni bir kural-tabanlı bilgisayar yazılımı ile süreçteki işlemlerin basitleşmesini ve hızlanmasını sağlayacak karar destek sistemi oluşturulması anlatılmıştır.

[26]'da detayları verilen ve 2003 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; CIM yatırımlarını değerlendirme yöntemlerinden biri olan Analitik Hiyerarşi Prosesi (AHP) yönteminin bulanık kümeler teorisi ile birlikte kullanıldığı ve Chang'ın (1996) çalışmasında önerdiği genişletilmiş analiz yöntemi kullanılarak; Bilgisayar Bütünleşik İmalat (CIM) sistemlerine yatırım analizi ve değerlendirmesi yapılmıştır.

[27]'de detayları verilen ve 2002 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; değişik yapılarıdaki atölyelerin çizelgelenmesi probleminin çözümü için Birleştirilmiş Modelleme Dili (UML) ve Microsoft Windows'un Dağıtık İnternet Uygulamaları mimarisi (Windows DNA) altında Darboğazı Kaydırma Metodu (SBP) ve imalat süresini kısaltma metodu algoritmaları kullanılmıştır.

[28]'de detayları verilen ve 2001 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; kesici takım envanterinin kullanım takibi, ihtiyaç planlaması oluşturulması ve satın alma önerilerini oluşturabilen takım yönetim sistemi geliştirilmiştir. Bu çalışmanın

oluşturulmasında destek ve tasarım için IDEFO, E-R modelleme, IDEFlx, Yourdon & Coad nesneye yönelik analiz yöntemi ve süreç diyagramları kullanılmış olup Microsoft Windows'un Dağıtık İnternet Uygulamaları mimarisi (Windows DNA) altında Visual Basic ve Visual InterDev ile programlanmış, İnternet Information Server ile web sunucusu, Microsoft Transaction Server nesne sunucusu ve Microsoft SQL Server'da veri tabanı sunucusu olarak kullanılarak sistem oluşturulmuştur.

[29]'da detayları verilen ve 2000 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; malzeme İhtiyaç Planlamasını (MİP) ve Tam Zamanlı Üretim (TZÜ) sistemlerinin ayrı ayrı ve beraber kullanılarak örnek benzetim modelleri oluşturulmuş ve sonuçların analiz ve değerlendirilmesi yapılmıştır.

[30]'da detayları verilen ve 1993 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; Bilgisayar Tümlleşik Üretim (BTÜ) sistemleri incelenmiş ve bununla ilgili sistem kurulum aşamalarını içeren rehber hazırlanmıştır. Bu rehber yardımıyla oluşturulan örnekler ile de uygulanabilirliği gösterilmiştir.

[31]'de detayları verilen ve 1992 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; Marmara Üniversitesi CIM (Bilgisayar Bütünleşik İmalat) laboratuvarında yapılan Tersine Yürütülen Prosesler Mühendisliği (Reverse Engineering) projelerinin daha verimli olmasını sağlamak için uygun işlem adımları oluşturularak, maliyetlerin azaltılması ve kalite kontrolün koordine edilebilmesini destekleyecek çözümler önermektedir.

1.3.2. YÖK veritabanında “esnek imalat” ve “FMS” kelimeleri tarama sonuçlarında listelenen doktora ve yüksek lisans tez çalışmalarından, sunulmakta olan çalışma ile alakalı olanlar şunlardır:

[32]'de detayları verilen ve 2007 yılında sunulmuş olan doktora tez çalışmasında özetle; Sayısal Denetimli (SD) makineler için belirlenen zorlayıcı koşullar ile problem oluşturularak Esnek İmalat Sistemleri (EİS)'nde kapasite paylaşırma modeli yardımıyla çözümü incelenmiştir. Deneyde Tamsayılı Doğrusal Modeli, Lagrange Gevşetme Yöntemi gibi çözüm yöntemleri kullanılarak elde edilen sonuçlar açıklanmıştır.

[33]'de detayları verilen ve 2005 yılında sunulmuş olan doktora tez çalışmasında özetle; üst düzey PN'ler kullanan FMS'lerin atölye zamanlama probleminin modellenmesi ve analizi için nesne yönelimli bir yaklaşım öneriyoruz. Grafik niteliği ve matematiksel

temel, Petri net tabanlı yöntemleri gerçek zamanlı çizelgeleme ve esnek üretim sistemleri (FMS'ler) kontrolünde çekici hale getirdi. Önerilen yaklaşımda öncelikle bir FMS'in bir nesne modelleme diyagramı oluşturduk ve kaynak çekişmesi problemini çözmek için buluşsal bir kural-taban oluşturduk ve sonra sistemin yüksek düzey PN'ler tarafından dinamik davranışını formüle edeceğiz. Modelleme metodolojisi bir FMS ile gösterilmiştir ve önerilen kural tabanlı sistem, part akış süresi ve gecikmeye bağlı performans ölçümleri açısından çeşitli sevk kurallarıyla karşılaştırılmıştır. Son olarak, farklı işletim konfigürasyonları altındaki sistem performansı, incelenen sistem için en iyi uygulama politikasını bulmak için analiz edilmiştir.

[34]'de detayları verilen ve 2004 yılında sunulmuş olan doktora tez çalışmasında özetle; Üretim Sistemlerin ve bireylerin performans ölçümlerinde, kullanılacak faktörlerin Bulanık Mantık Analitik Hiyerarşi Süreci (BAHP) ile tanımlanması ve yeni sistemlerin geliştirmesiyle ilgili bilgiler verilmiş pekiştirmek amacıyla Esnek İmalat Sistemleri (EİS) örneği verilmiştir.

[35]'de detayları verilen ve 2003 yılında sunulmuş olan doktora tez çalışmasında özetle; Esnek imalat Sistemi (EİS)'nin işleyişinde verimlik artırımı ve esneklik için Bulanık (Fuzzy) karar verme algoritmaları kullanılarak etkili çizelgeleme için geliştirilen yöntemdir.

[36]'da detayları verilen ve 2003 yılında sunulmuş olan doktora tez çalışmasında özetle; Esnek İmalat Sistemlerinde (EİS) üretim öncesi hazırlık aşamalarında ki problemlerin çözümü ve verim artırımı için kullanılan tavlama benzetimi, kısa ve uzun dönem hafızalı tabu arama algoritmaları kullanılarak karşılaştırmaları yapılmış ve sonuçlar değerlendirilmiştir.

[37]'de detayları verilen ve 2000 yılında sunulmuş olan doktora tez çalışmasında özetle; Yaygın bir tipte esnek imalat hücresi modeli Visual Basic programlama dili yardımıyla modüler yapıda parametrik atölye benzetim yazılımı oluşturulmuştur. Burada MAXQ ve MINQ yöntemleri geliştirilerek diğer çizelgeleme yöntemleriyle imalat sistemlerinde uygulanmasının karşılaştırması yapılmış ve sonuçları açıklanmıştır. Özellikle, çizelgeleme politikalarının diğer metotlara uygulandığı durumlarda, ortalama akış süresinde %30 ve gecikmeye bağlı performans kriterlerinde de ortalama %60'ları geçen iyileşmeler gözlemlenmiştir.

[38]'de detayları verilen ve 1998 yılında sunulmuş olan doktora tez çalışmasında özetle; Simülasyon programı kullanılarak, planlama, tasarım ve analiz süreçlerine bağlı olarak farklı tasarım konfigürasyonlarında oluşturulacak Esnek İmalat Sistemi (EİS)'nin ve imalatı yapılacak parçaların sayıları ve stok durumu gibi yapay sinir ağı yardımıyla optimum sistem konfigürasyonunun belirlenmesi çalıştırılmıştır. Her konfigürasyon ayrı ayrı analiz edilmiş ve sonuçları açıklanmıştır.

[39]'da detayları verilen ve 1997 yılında sunulmuş olan doktora tez çalışmasında özetle; Esnek İmalat Sistemleri (EİS) 'de planlama ve çizelgeleme problemlerinin çözümü ve ek olarak birkaç problem çözümünü de içerisine alan yeni bir yaklaşım olarak Esnek İmalat Yönetim Sistemi (EİYS) tasarlanmıştır. Bu tasarlanan yaklaşım için Esnek İmalat Sistemi Yöneticisi (EİSYÖN) adında ki yazılım özellikleri ve yapısı açıklanmıştır.

[40]'da detayları verilen ve 1997 yılında sunulmuş olan doktora tez çalışmasında özetle; CNC tezgâhıyla beraber aynı bilgisayardan kontrolü sağlanan üç eksenli robot kolu ile bir Esnek İmalat Sistemi yapılmıştır. Bu çalışmada robot kolunun tasarımdan yapılmasına kadar ki kullanılan yöntemler açıklanmıştır. Oluşturulan düzenekteki hesaplanan ve uygulama sonucundaki değerler karşılaştırılmıştır.

[41]'de detayları verilen ve 2014 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; Üretim süreçlerinde verimliliği etkileyen imalat sistemlerinde kullanılan döner ve kayar mafsallı robotların sisteme negatif etkilerinden olan titreşim ve gürültülerinin nedenleri ve önlenmesi için yöntemler denenmiştir. Micro FMS sisteminde farklı hızlarda ve farklı tür ve ağırlıklarda kullanılan malzemelerle bir CNC Besleme istasyonunda olan Mitsubishi RV2-AJ Robotu kullanılarak deneyler yapılmış ve sonuçlar analiz edilerek değerlendirilmesi yapılmıştır.

[42]'de detayları verilen ve 2010 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; Örnek bir Bilgisayar Bütünleşik İmalat (CIM) laboratuvarının benzetim modelini oluşturmada ve kuyruk ağları ile performans ölçümü ve analizleri için Ortalama Değer Analizi (ODA) algoritmaları kullanılması test edilerek değerlendirmesi yapılmıştır.

[43]'de detayları verilen ve 2009 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; Esnek İmalat Sistemleri (EİS)'nde takım atamadaki verimliliğin artırımı için önerilen matematiksel fonksiyonlar ve genetik algoritmanın kullanımı ve detaylı olarak sayısal analizlerle test edilmesidir.

[44]'de detayları verilen ve 2006 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; Geliştirilen Genetik Algoritma (GA)'nın Esnek İmalat Sistemleri (EİS)'nde kesici takımların yerleştirilmesi problemlerinin çözümünde kullanılması ve kesin çözüm algoritmalarıyla karşılaştırmasının değerlendirilmesidir.

[45]'de detayları verilen ve 2005 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; Esnek imalat sisteminin modellenmesi ve benzetimde esneklik düzeyinin yükseltilmesi için sistemde genel olarak kullanılan diğer yardımcı programlarında programa eklenmesinin sonuçları incelenmiştir. Pilot olarak kullanılan bir esnek imalat laboratuvar sistemi üzerinden ARENA® programı yardımıyla model oluşturma süreçleri değerlendirilmiştir.

[46]'da detayları verilen ve 2005 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; sınırlı miktarda işleme ucu barındıran cihazların yer aldığı esnek imalat sistemlerinde uç değişimi için gerekli manevra sayıları farklı algoritmalar kullanılarak optimize edilmeye çalışılmıştır. Bu algoritmalar içerisinde Dal – Sınır algoritması ve Işın Araştırma sezgisel yöntemleri üzerinde durulmuştur. Problemin büyüklüğüne bağlı olarak algoritmaların başarı yüzdelerinin farklılık gösterdiği görülmüştür.

[47]'de detayları verilen ve 2004 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemlerinin kısıtlı yerleşim ve takım kapasitelerinin verimi arttırmak adına kullanılabilecek yaklaşımlar üzerinde durulmuştur. SD makinelerinin imalat kapasiteleri üzerinden yapılan problem tanımı Tam sayılı Doğrusal Model olarak formüle edilmiştir ve NP zor olduğu gösterilmiştir. Orta boyutlu problemlerde bu yaklaşımın başarılı olduğu görülmüştür. Lagrange gevşetme tekniği ve sezgisel yöntemler kullanılarak daha büyük boyutlu problemlerin çözümüne yönelik başarılı sonuçlar elde edilmiştir.

[48]'de detayları verilen ve 2003 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemlerin proses yönetimi ve makine ucu değiştirme süreçleri üzerinden bir problem tanımı yapılmış ve buna çözüm aranmıştır. Makine ucu değiştirme sırasının optimizasyonu için dal-sınır algoritması ve sezgisel yöntemler kullanılmıştır. Problemin büyüklüğü arttıkça sezgisel algoritmaların daha başarılı olduğu görülmüştür.

[49]'da detayları verilen ve 2002 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; tek tip parça üreten esnek imalat tezgâhlarının üretkenliği arttırmak için hücresel

izelgeleme yaklaşıımı zerinde durulmuştur. Hcrelerin retim verimini arttırmak iin srelere iliřkin izelgeleme sırası geliřtirilen matematiksel model tarafından belirlenmektedir.

[50]'de detayları verilen ve 2000 yılında sunulmuş olan yüksek lisans tez alışmasında zetle; esnek imalat sistemlerinde takımların kullanımına ynelik yeni stratejiler geliřtirilmeye alışılmıştır. Tez alışmasında takım ynetim stratejileri arasında yer alan para ve takım bazlı stratejilerden ağırlıklı olarak takım bazlı stratejiler zerine alışması hedeflenmiştir. Takım bazlı stratejilerle ilgili olarak modelleme ve analiz alışmalarının yapılması planlanmıştır.

[51]'de detayları verilen ve 2000 yılında sunulmuş olan yüksek lisans tez alışmasında zetle; esnek imalat sistemlerinin seilmesi ve deęerlendirilmeye alınması ile ilgili kullanılan yaklaşımlar incelenmiştir. Bu yaklaşımlardan yaygın olarak kullanılanların problemlere deęerlendirme biimindeki yetersizliklerden bahsedilmiştir. Esnek imalat sistemlerinin deęerlendirilmesinde daha kapsayıcı bir yaklaşımın benimsendięi ve bulunak kmeler teorisine dayanan stratejiler zerinde durulmuştur. Bu tez alışmasında doęrusal modelleme yaklaşımları kullanılarak en iyi EİS alternatiflerini tespit eden bir zm yntemi tasarlanmıştır.

[52]'de detayları verilen ve 2000 yılında sunulmuş olan yüksek lisans tez alışmasında zetle; esnek imalat sistemlerinin tesis edilmesi ncesi kullanılan yatırım analizi teknikleri incelenmiştir. Klasik iskontolu nakit akışı analizi ve kısıtlarından bahsedilmiştir. Opsiyon temelli ıkarım ve analiz yaklaşımlarının ortaya ıkışı, geliřmesi ve maliyet analizi iin kullanımından bahsedilmiştir. Esnek imalat sistemlerinin gruplamasında kullanılan tezgah, sre, rn, para, para ynlendirme, retim hacmi, geliřtirme ve rn hacmi kavramları zerinde durulmuştur. Kantitatif zmlerin ne srldę yayınlr zerinde incelemelerde bulunulmuştur.

[53]'de detayları verilen ve 2000 yılında sunulmuş olan yüksek lisans tez alışmasında zetle; mhendislik eęitiminde kullanılmak zere laboratuvar bir esnek imalat sistemi tasarımı gerekleřtirilmiştir. Tasarımı srelerine iliřkin detaylı bilgiler tez ierisinde verilmiştir.

[54]'de detayları verilen ve 2000 yılında sunulmuş olan yüksek lisans tez alışmasında zetle; esnek imalat sistemlerindeki teknolojik geliřmelere paralel olarak ortaya ıkan

Bilgisayar Tümüleşik Üretim kavramı ve uygulamaları üzerinde durulmuştur. Tez çalışması kapsamında IDEFO ve IDEF1X standartları kullanılarak Ortadoğu Teknik Üniversitesi Makine Mühendisliği Bölümü Bilgisayar Tümüleşik Üretim Laboratuvarı'nda uygulanmasını hedeflenmiştir. Bunlara ek olarak çalışmalar tez ile sınırlı kalmadan KOBİ düzeyindeki işletmeler içinde uygulanabilirliğinin önünü açacak yazılım tasarımlarının yapılması hedeflenmiştir.

[55]'de detayları verilen ve 2000 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; otomatik imalat sistemlerinde fiziksel modelleme yaklaşımlarından bahsedilmiştir. ODTÜ-Endüstri Mühendisliği- Üretim Modelleme Laboratuvarı'nda yer alan fiziksel modelleme sistemi üzerinde çalışabilecek şekilde bir esnek imalat yönetim yazılımı hazırlanmıştır. Tez çalışması sırasında geliştirilen program üzerinden imalat sisteminin çeşitli bileşenleri ve çıktıları kontrol edilebilmektedir.

[56]'da detayları verilen ve 1999 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemi vasıtasıyla gerçekleştirilen bir üretim sırasında sistemde oluşabilecek sapma ve hataların tanınması ve minimize edilmesi için yapay sinir ağı tabanlı bir modelleme çalışması yapılmıştır. Geriye yayılım algoritması gibi yaklaşımlar kullanılarak YSA'nın eğitimi iyileştirilmeye çalışılmıştır.

[57]'de detayları verilen ve 1998 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; bir esnek imalat sistemi üzerinden altı ay boyunca toplanan veriler kullanılarak sistemin bilgisayar modeli oluşturulmuştur. Modelleme çalışması ile sistemin daha verimli çalışması için yapılması gereken düzenlemeler tespit edilmeye çalışılmıştır. Matematiksel model LINGO sürüm 3.1 programı ile oluşturulmuştur. Simülasyon programı ise SIMAN 4'te yazılmıştır.

[58]'de detayları verilen ve 1998 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat gerçekleştirilirken çeşitli arızalar sebebiyle üretimde oluşacak gecikmeleri kompanse edebilmek amacıyla METU-R-CAPP isimli bir esnek imalat planlama yazılımı geliştirilmiştir.

[59]'da detayları verilen ve 1998 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; Orta Doğu Teknik Üniversitesi Makina Mühendisliği Bölümünde bulunan METUCIM laboratuvarının esnek imalat problemlerini minimize etmek adına parça ve

takım temelli sezgisel yaklaşımlar geliştirilmiştir. Yöntemlerin karşılaştırılmalı analizleri yapılmıştır.

[60]'da detayları verilen ve 1997 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemlerinde kullanılan çizelgeleme politikalarını iyileştirmek adına karışık tam sayılı bir çizelgeleme formülasyonu geliştirilmiştir. Formülasyon farklı boyutlardaki problemler üzerine uygulanarak modelin cevap süresi ölçülmüştür.

[61]'de detayları verilen ve 1997 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat tezgâhlarında işlenen parçaların ölçülmesi, DNC oluşturma ve parça imalat toleranslarının tespiti konuları üzerinde durulmuştur.

[62]'de detayları verilen ve 1997 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemlerin işlem verimliliğinin geliştirilerek, işletme maliyetlerinin Avrupa birliği ülkeleri ile rekabet edebilir seviyeye indirilebilmesi için tam sayılı programlama ve modelleme çalışmaları üzerinde durulmuştur. Tam sayılı programlama yaklaşımı makine – parça ailelerinin oluşturulması probleminde devreye sokulmuştur.

[63]'de detayları verilen ve 1997 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat hücresi akış hatlarında kesici uç ve çizelgeleme problemlerinin daha etkin biçimde çözülmesine olanak sağlayabilecek bütünleşik bir algoritma tasarlanmıştır.

[64]'de detayları verilen ve 1996 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemleri için paket programlar vasıtasıyla oluşturulan modelleme ve simülasyon çalışmalarından bahsedilmiştir. İmalat kriterlerinin sistemin performansı üzerindeki etkilerinden bahsedilmiştir. Farklı imalat kriterlerine ilişkin verilerin elde edebilmek amacıyla bir esnek imalat sistemi ve bilgisayar simülasyonu tasarlanmıştır.

[65]'de detayları verilen ve 1996 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemlerinin çalışma performansını etkileyen takım akışı problemleri üzerinde durulmuştur. Takım akışı problemlerinin çözümünde kullanılan Full Clustering ve Differential Clustering takım atama yaklaşımlarının kullanıldığı bir yazılım geliştirilmiştir.

[66]'da detayları verilen ve 1995 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemlerinin ortaya çıkışı ve bu sistemlerin teknolojik gelişim süreçleri hakkında genel bir bilgilendirme yapılmıştır. Daha sonra esnek imalat

sistemlerinde en sık karşılaşılan çizelgeleme hatalarının minimize edilmesi amacıyla yapılan modelleme çalışmalarından bahsedilmiştir. Modelleme çalışmaları SIMAN platformu üzerinde gerçekleştirilmiştir. Esnek imalat sistemleri üzerinden deneysel metotlarla elde edilen veriler bilgisayar modeline entegre edilmiştir. Simülasyon üzerinden geri besleme alarak imalat sisteminin çalışmasını optimize edebilecek yaklaşımlar ön görülmüştür. Bunun yanında esnek imalat sistemlerinin sağlıklı olarak işletilebilmesinde insan – makine etkileşiminin önemi vurgulanmıştır.

[67]'de detayları verilen ve 1994 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemlerinin bölümleri, kurulması ve işletilmesi ile ilgili temel bilgilere yer verilmiştir. Bilgisayar destekli üretim ve otomasyon sistemleri anlatılmıştır. Esnek imalat kavramı farklı açılarıyla ele alınmıştır. Çizelgeleme kavramından bahsedilmiştir. Çalışmanın sonunda ise makine hücrelerinin oluşturulması problemlerinin çözümüne yönelik örnek çalışmalar yapılmıştır.

[68]'de detayları verilen ve 1993 yılında sunulmuş olan yüksek lisans tez çalışmasında özetle; esnek imalat sistemlerinde karşılaşılan taktik planlama problemleri için genel bir kapsam oluşturan jenerik modelleme yaklaşımıyla ilgili bir çerçeve oluşturulmuştur. Kümeleme, yükleme ve akış rotası belirleme problemlerinin çözümü için jenerik bir formülasyon önerilmiştir.

1.3.3. YÖK veritabanında “esnek üretim” kelimesi tarama sonuçlarında listelenen doktora ve yüksek lisans tez çalışmalarından, sunulmakta olan çalışma ile alakalı olanlar şunlardır:

[69]'da detayları verilen ve 1993 yılında sunulmuş olan doktora tez çalışmasında özetle; robotlar ile entegre çalışan esnek imalat sistemleri için zincir tahrikli bir konveyör tasarımı yapılmıştır. Konveyör üzerinde hareket eden cisimlerin şekil tanınması YSA tabanlı bir görüntü işleme sistemi ile gerçekleştirilmiştir. Konveyörde akan cisimlerin depolanabilmesi için dört serbest derecesine sahip bir robot kol tasarlanmıştır. Tasarlanan cihazlar bir araya getirilerek entegre çalışma testleri yapılmıştır.

[70]'de detayları verilen ve 2011 yılında sunulmuş olan doktora tez çalışmasında özetle; eğitim amaçlı üretilmiş olan bir esnek imalat sisteminin mekatronik eğitiminde kullanılabilirliğini arttırmak için ek donanım ve yazılımlar tasarlanarak sisteme entegre edilmiştir.

[71]'de detayları verilen ve 2010 yılında sunulmuş olan doktora tez çalışmasında özetle; esnek imalat hücrelerinde imalat talebinde yaşanabilecek değişiklikler sonucunda sistemin en hızlı ve etkili biçimde yeniden konfigüre edilebilmesi için bir matematik programlama modeli oluşturulmuştur. Model LINGO 8.0 programı ile oluşturulmuştur.

[72]'de detayları verilen ve 2010 yılında sunulmuş olan doktora tez çalışmasında özetle; esnek üretim sistemlerin görüntü işleme çalışmaları ile ilgili eğitimlerde kullanılması amacıyla XR3 deneysel robotu ile renk tanıma teknikleriyle konveyörde akmakta olan cisimlere müdahale edebilen bir sistem geliştirilmiştir. Robotun programlaması için Türkçe bir program geliştirilmiştir. Deneysel sırasında kullanılmak üzere gaziXR3 isimli bir arayüz programı geliştirilmiş ve kullanıma sunulmuştur.

[73]'de detayları verilen ve 2010 yılında sunulmuş olan doktora tez çalışmasında özetle; esnek imalat sistemlerinde çizelgeleme problemlerinin çözümüne yönelik çalışmalar gerçekleştirilmiştir.

[74]'de detayları verilen ve 2003 yılında sunulmuş olan doktora tez çalışmasında özetle; esnek üretim sistemlerinde iş yükleme, çizelgeleme ve kesici uç yönetimi problemleri birlikte çözülmektedir. Amaç, operasyon koşullarının belirlenmesi, parçaların birbirinden farklı paralel makinalara yüklenmesi ve çizelgelenmesi kararlarının verilmesidir. Teslim tarihi ve maliyet ile ilgili amaçlar dikkate alınmaktadır. Teslim tarihi ile ilgili amaçlar, toplam ağırlıklı gecikme süresi ve toplam ağırlıklı erken bitirme/gecikme sürelerinin enazlanmasıdır. Maliyet ile ilgili amaç, işlem, kesici uç ve işleme dahil olmayan maliyetlerin toplamından oluşan üretim maliyetinin enazlanmasıdır. Esnek üretim sistemlerinin en önemli özelliklerinden birisi üretim zamanlarının, operasyon koşullarına bağlı olarak, kontrol edilebilir olmasıdır. Kontrol edilebilir üretim zamanları, çizelgeleme problemlerini çözerken, farklı çözüm alternatiflerinin bulunmasında sağladığı esneklikten dolayı, dikkate alınmıştır.

[75]'de detayları verilen ve 2001 yılında sunulmuş olan doktora tez çalışmasında özetle; prizmatik parçaların otomatik bağlama kalıplarının tasarımında kullanılmak üzere bir C++ tabanlı bilgisayar programı geliştirilmiştir. Prizmatik parçalara üç boyutlu ait modeller BDT platformu üzerinde tasarlanmıştır. Bağlama elemanlarının robot yardımıyla montajı için robot yazılımları geliştirilmiştir.

[76]'da; ilk uygulaması Northern Michigan University de gerçekleştirilen CIM sistemi hakkında bilgi verdiği için incelenmiş olup; içeriğinde özetle: CIM sistemlerinin özellikleri ve faydaları, gerçekleştirilen CIM uygulaması, akademik gereklilikler, CIM donanımının eğitim amaçlı kullanılması ve CIM sistemi uygulamasının bileşenlerinin açıklandığı görülmektedir.

[77]'de; imalat sistemleri, sürekli tümleşik süreç sistemlerinin (CIP) gereksinimleri, yapıları, karakteristikleri, işlevleri ve olası problem tanımlamaları, CIM sistemleri ile aralarındaki farklar hakkında bilgi verdiği için incelenmiş olup; içeriğinde özetle: sürekli imalat endüstrilerinde CIP sistemlere neden gereksinim duyulduğu, sürekli imalat endüstrilerinde mevcut bulunan sorunların açıklandığı, bilgisayar tümleşik imalat sistemleri (CIM) ile CIP sistemleri arasındaki farkların irdelenerek CIP karakteristiklerinin açıklandığı, tümleşik dağıtık akıllı yazılım sistemi platformunun yapısı ve işlevlerinin neler olduğunun belirtildiği görülmektedir.

[78]'de; ülkemizde gerçek bir CIM sisteminin eğitim alanında kullanımına yönelik yapılan örnek bir çalışma olduğu için incelenmiş olup; içeriğinde özetle: eğitim amaçlı CIM laboratuvarında Sistem Mühendisliği bölümü müfredatı için geliştirilen prototip tasarım ve üretim projesinden bahsedildiği, CIM'in tarihçesi ve projede kullanılan CIM sisteminin bileşenlerinin tanıtıldığı, sistem üzerinde Endüstri Mühendisleri ve Otomasyon ve Kontrol Mühendisliği bölümü öğrencileriyle uygulanan eğitim sonuçlarının karşılaştırmalı olarak irdelendiği görülmektedir.

1.3.4. Literatür araştırma sonuçlarının yorumlanması:

Sunulmakta olan bu tez çalışması konusuyla alakalı YÖK veritabanında “CIM”, “bilgisayar bütünleşik imalat”, “bilgisayar tümleşik imalat”, “esnek imalat”, “FMS” ve “esnek üretim” anahtar kelimeleriyle yapılan araştırmalar sonucunda, 1992-2016 yılları arası yapılmış toplam 54 adet lisansüstü çalışmaya rastlanmıştır.

Çalışmaların konuları irdelendiğinde EİS çizelgeleme, imalat hücreleri ve cihazlarıyla ilgili geliştirme çalışmaları, verimlilik ve maliyet analizi, çeşitli alanlarda tasarım ve modelleme çalışmaları yapıldığı; çalışmalardan sadece 3 adetinin (%5,6) eğitime yönelik EİS tasarım ve geliştirme çalışması olduğu, hiç birinde eğitim materyali hazırlanmadığı görülmüştür.

EİS alanında eğitim amaçlı yapılmış en önemli çalışma, T.C. Milli Eğitim Bakanlığı tarafından Endüstriyel Otomasyon Teknolojileri grubuna hazırlanmış Fabrika Otomasyon Sistemleri modülleridir. Sunulmakta olan bu tez çalışması sonucu üretilen eğitim materyallerin lise seviyesi mesleki teknik eğitim alanına da uyarlanması söz konusudur.

YÖK veri tabanında yapılan literatür taraması sonuçlarına ait Tablo 1-1’de lisansüstü çalışmaların üniversitelere göre dağılımı, Tablo 1-2’de ise bölümlere göre dağılımları verilmiştir. Tablolardan anlaşılabacağı üzere sunulmakta olan bu tez çalışması konuları kapsamına yakın en çok lisansüstü çalışma Orta Doğu Teknik Üniversitesi; bu alanda en çok çalışan ise endüstri mühendisliği bölümüdür.

Tablo 1-1. YÖK veritabanındaki lisansüstü çalışmaların üniversitelere göre dağılımı.

Üniversite	Yüksek Lisans	Doktora	Toplam	Genel Toplam Yüzdesi
Orta Doğu Teknik Üniversitesi	14	3	17	31,48%
Sakarya Üniversitesi	3	5	8	14,81%
Gazi Üniversitesi	2	4	6	11,11%
İstanbul Teknik Üniversitesi	3	1	4	7,41%
Marmara Üniversitesi	3	1	4	7,41%
Bilkent Üniversitesi	1	2	3	5,56%
Boğaziçi Üniversitesi	2	-	2	3,70%
Erciyes Üniversitesi	1	1	2	3,70%
Galatasaray Üniversitesi	2	-	2	3,70%
Yıldız Teknik Üniversitesi	2	-	2	3,70%
Çukurova Üniversitesi	-	1	1	1,85%
Dokuz Eylül Üniversitesi	-	1	1	1,85%
Karabük Üniversitesi	-	1	1	1,85%
Kırıkkale Üniversitesi	1	-	1	1,85%
TOPLAM	34	20	54	100%

Tablo 1-2. YÖK veritabanındaki lisansüstü çalışmaların bölümlere göre dağılımı.

Bölüm	Yüksek Lisans	Doktora	Toplam	Genel Toplam Yüzdesi
Endüstri Mühendisliği	22	10	32	62,75%
Makina Mühendisliği	9	5	14	27,45%
İşletme Yönetimi	1	1	2	3,92%
Makina Eğitimi	-	1	1	1,96%
Mekatronik Mühendisliği	1	-	1	1,96%
Elektrik Elektronik Mühendisliği	-	1	1	1,96%
TOPLAM	34	20	54	100%

2. MATERİYAL VE YÖNTEM

Bu bölümde tez konusu olan bilgisayar destekli esnek imalat sistemi (FMS100) hakkında genel bilgiler; sistemi oluşturan istasyonlar ve bünyelerinde bulunan otomasyon donanımları hakkında teknik bilgiler ve kullanım bilgileri verilmektedir. FMS100 sistemi hakkındaki bilgiler, üretici firmanın her bir istasyon için İngilizce hazırladığı 8 adet el kitapçığından birebir çevrilmiştir. Gerektiğinde, Türkçe'ye uyumu ve cümlelerin anlaşılmasını sağlamak için çevirilerde değişiklikler, cümle, kelime ve paragraf ilavesi yapılmıştır.

Bölümde sadece “FMS100 Manual” [79] isimli el kitabının çevrisi yer almaktadır. FMS100'ü oluşturan diğer 7 bölüme ait dokümanların çevirileri ise Ekler bölümünde sunulmuştur.

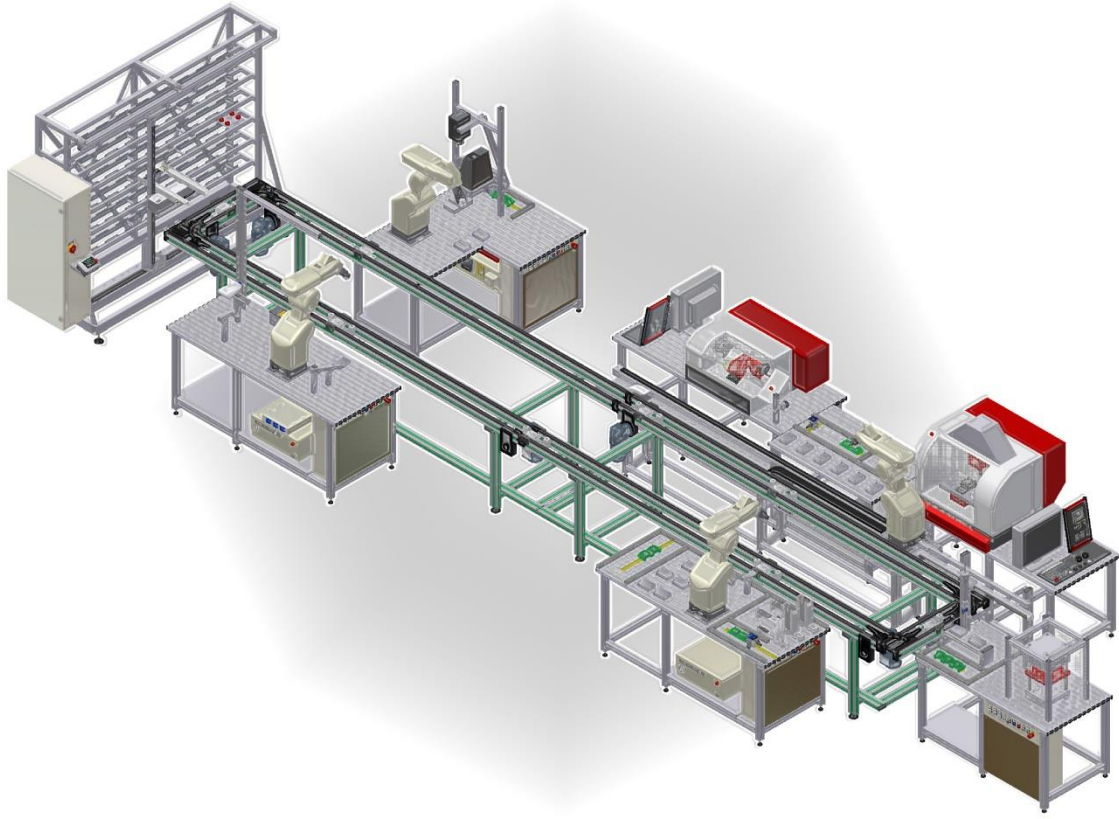
Ayrıca FMS100 sistemi üzerinde hangi yöntemlerle eğitim verildiği anlatılmaktadır.

2.1. Materyal

Bu kısımda çalışmada kullanılan eğitim sistemi olan FMS100 yapısı, imalat istasyonları ve istasyon bünyesinde yer alan donanımlar tanıtılmıştır.

2.1.1. FMS100 Bilgisayar Destekli Esnek İmalat Sistemi

FMS100 sistemi, endüstriyel otomasyon ve iletişim alanında temel ve ileri seviye eğitim verilebilmesi için tasarlanmıştır. Türünün en gelişmişlerinden biri olan ve bilgisayar ortamındaki çizimi Şekil 2.1'deki gibi olan FMS100, eğitim amaçlı tasarlanmış ve geliştirilmiş bir fabrika otomasyon sistemi olup; çeşitli imalat hücrelerinden meydana gelmektedir. Sistemi oluşturan hücreler komple bir üretim sistemi gibi koordineli çalışabildikleri gibi, her bir hücre FMS100'den bağımsız olarak tek başına da çalışabilir. Ayrıca istendiği takdirde hücrelerdeki robot, PLC gibi donanımlar da, hücreden bağımsız olarak çalıştırılabilir.

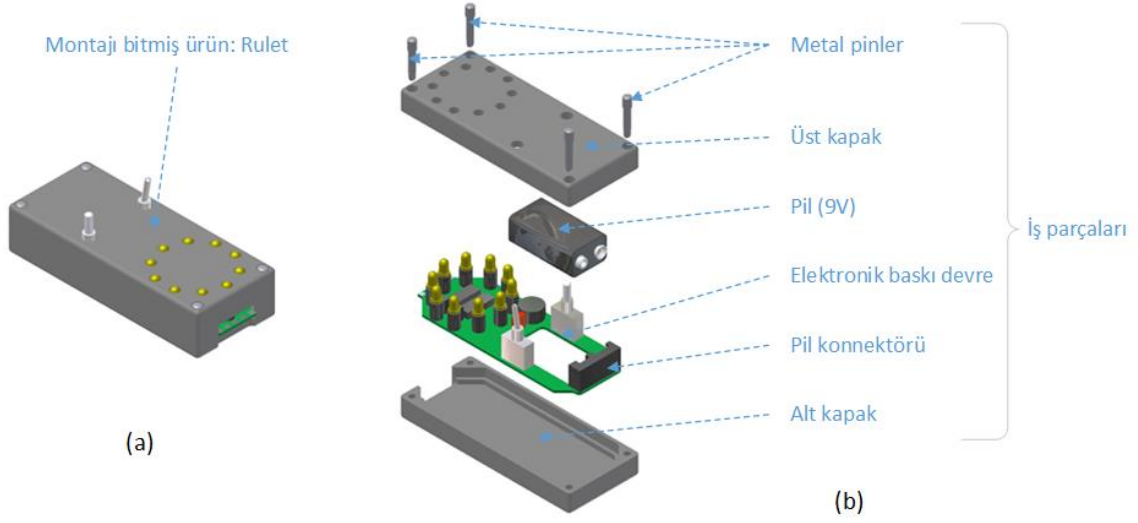


Şekil 2-1. FMS100 genel görünüşe ait bilgisayar çizimi görüntüsü [79].

FMS100 Şekil 2.2 (a)'da bilgisayar çizimi görülmekte olan elektronik ruleti üretecek şekilde tasarlanmıştır. Elektronik ruletin seçilme sebebi, üretimi için hem mekanik hem de elektronik işçilik gerektirmesi ve robotik montaja uygun olmasıdır.

FMS100'ün örnek ürünü olan elektronik rulet; Şekil 2.2 (b)'de görüleceği üzere delikli üst kapak, elektronik kart, alt kapak, 9V pil, pil konektörü ve kapakları bir arada tutan metal pin parçalarından meydana gelmektedir. Ruleti oluşturan bu parçaların hepsi FMS100 tarafından üretilmemektedir. Örneğin pil, pil konektörü ve baskı devre hazır olarak gelmektedir. Sistem bünyesinde ham olarak gelen metal pinler CNC tornada işlenmektedir. CNC freze ile plastik kapaklar üzerine delikler açılarak üst kapak, kenarlarına kanal açılarak alt kapak imal edilmektedir. Pil konektörünü robot baskı devre üzerine lehimlemektedir.

Sistem bünyesinde ürüne ait gerek işlenmiş gerekse işlenmemiş halde bulunan tüm parçalar genel olarak “iş-parçası” olarak isimlendirilmektedir.



Şekil 2-2. FMS100'ün ürettiği örnek ürün olan elektronik rulete ait bilgisayar çizimi (a), ruleti oluşturan iş-parçalarının bilgisayar çizimi (b).

Genel hatlarıyla bakıldığında FMS100 sistemi şu imalat hücreleri ve bileşenlerden oluşmaktadır:

2.1.1.1. Taşıma Sistemi (Transport System)

FMS100 bünyesinde işlenmemiş/işlenmiş iş-parçalarının özel tasarlanmış paletler üzerinde istasyonlar arası taşınması görevini yürütür. En temel bileşeni olan konveyör sisteminin bilgisayar ortamındaki çizimi Şekil 2-3'deki gibidir.

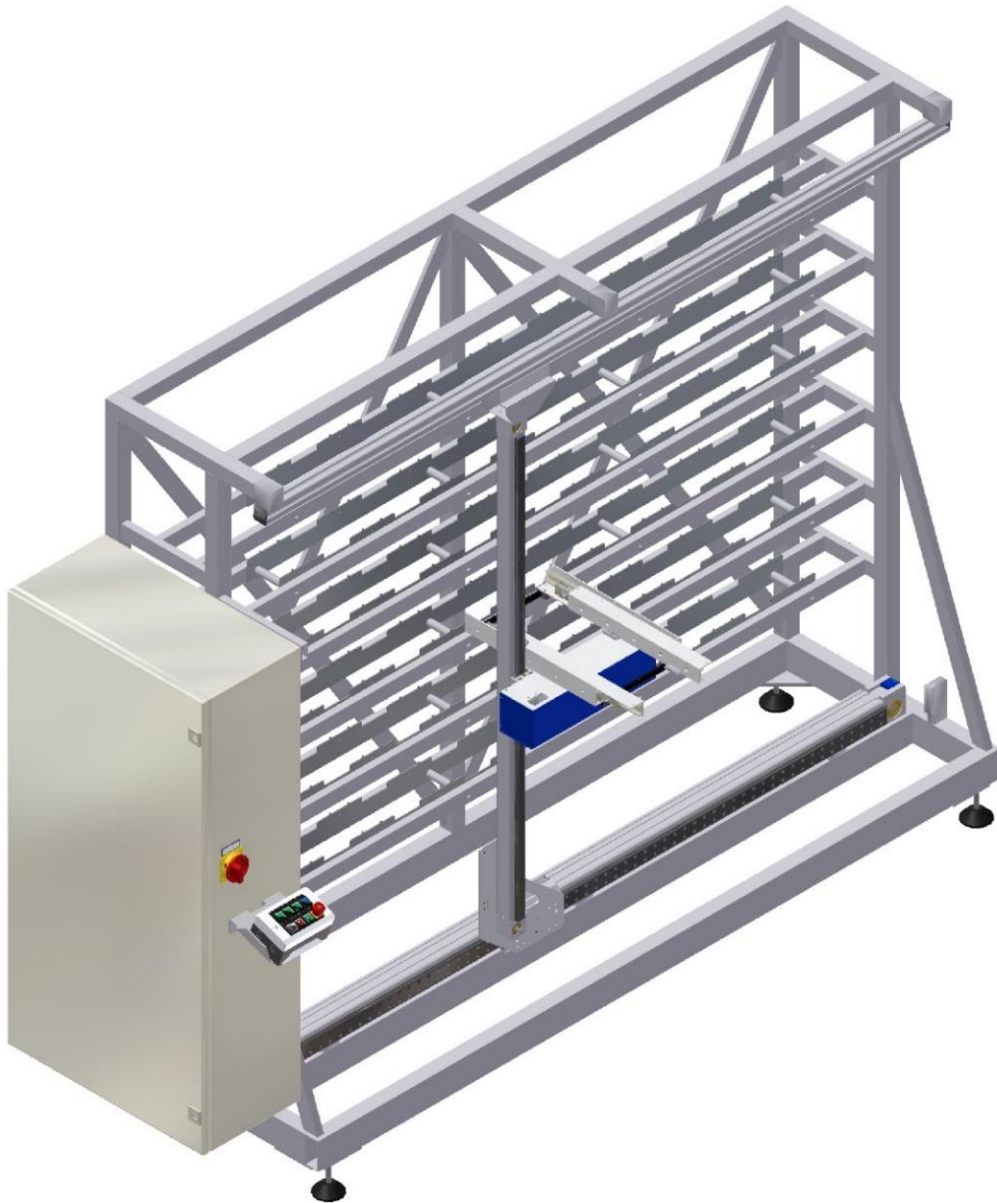


Şekil 2-3. İmalat hücreleri arası iş-parçalarının taşınmasını sağlayan konveyör hattının bilgisayar çizimi [79].

Hat üzerinde bulunan bantı hareket ettirebilmek için 0,12KW güce sahip 3 fazlı 6 adet asenkron motor kullanılmaktadır.

2.1.1.2. Stok Hücresi (AS-RS)

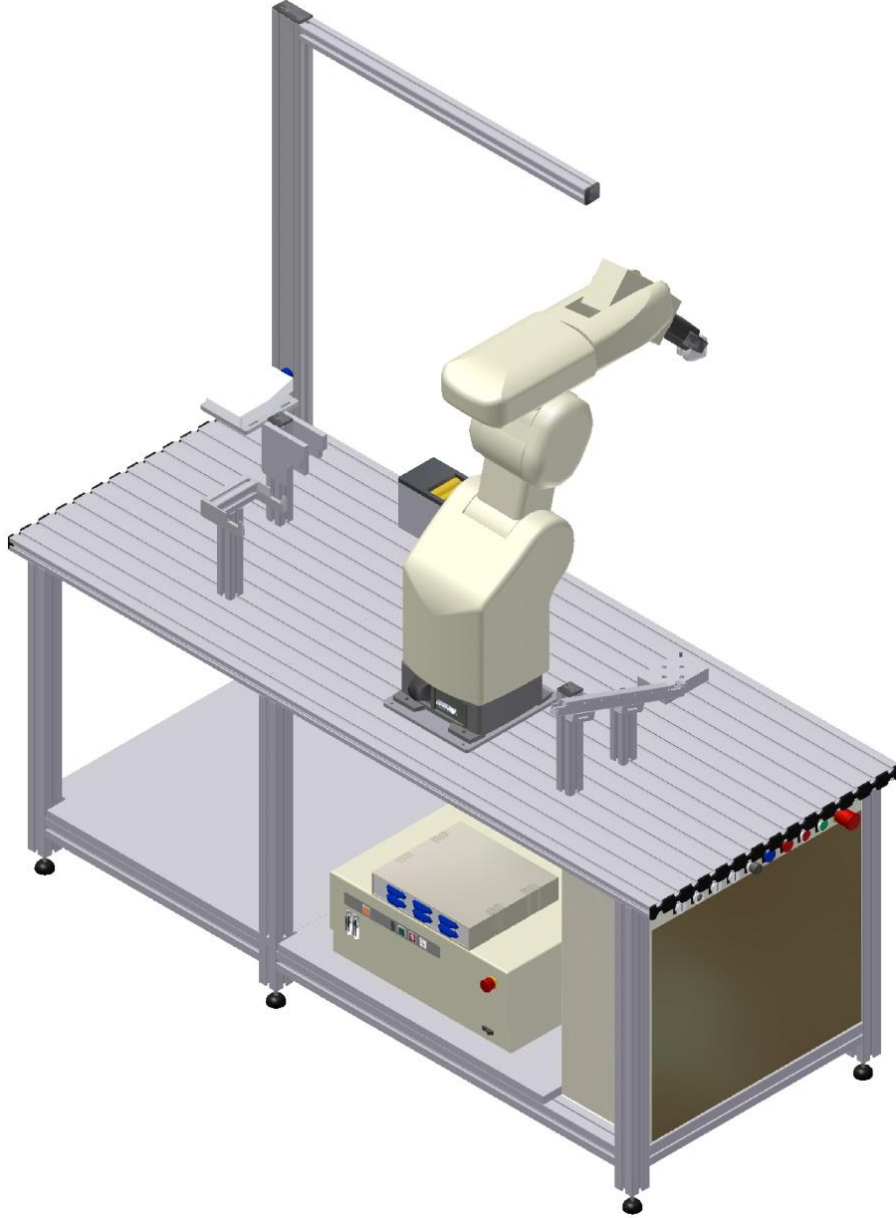
AS-RS kelimesi, İngilizce “automatic storage – retrieve system” (otomatik saklama – geri getirme sistemi) kelimelerinin baş harflerinden meydana gelmektedir. Stok hücresi, FMS100 bünyesinde işlenecek/işlenmiş iş-parçalarının sınıflandırılarak depolanması ve istenildiğinde sisteme geri gönderilmesi görevini yürütür. Bilgisayar çizimi Şekil 2.4’deki gibi olan istasyon 48 adet numaralandırılmış rafa sahiptir. İş-parçaları, Stok hücresi bünyesinde yardımcı paletler üzerinde olacak şekilde, Teleskop adı verilen çatal biçimli mavi renkli bir cihaz ile taşınır. Hücre denetleyicisi olarak IPC (endüstriyel PC) kullanılmış olup, FMS100 otomasyon sistemindeki ismi “İstasyon 1”dir.



Şekil 2-4. Yardımcı paletlerin depolandığı AS-RS sistemine ait bilgisayar çizimi [79].

2.1.1.3. Lehimleme Hücresi (Soldering)

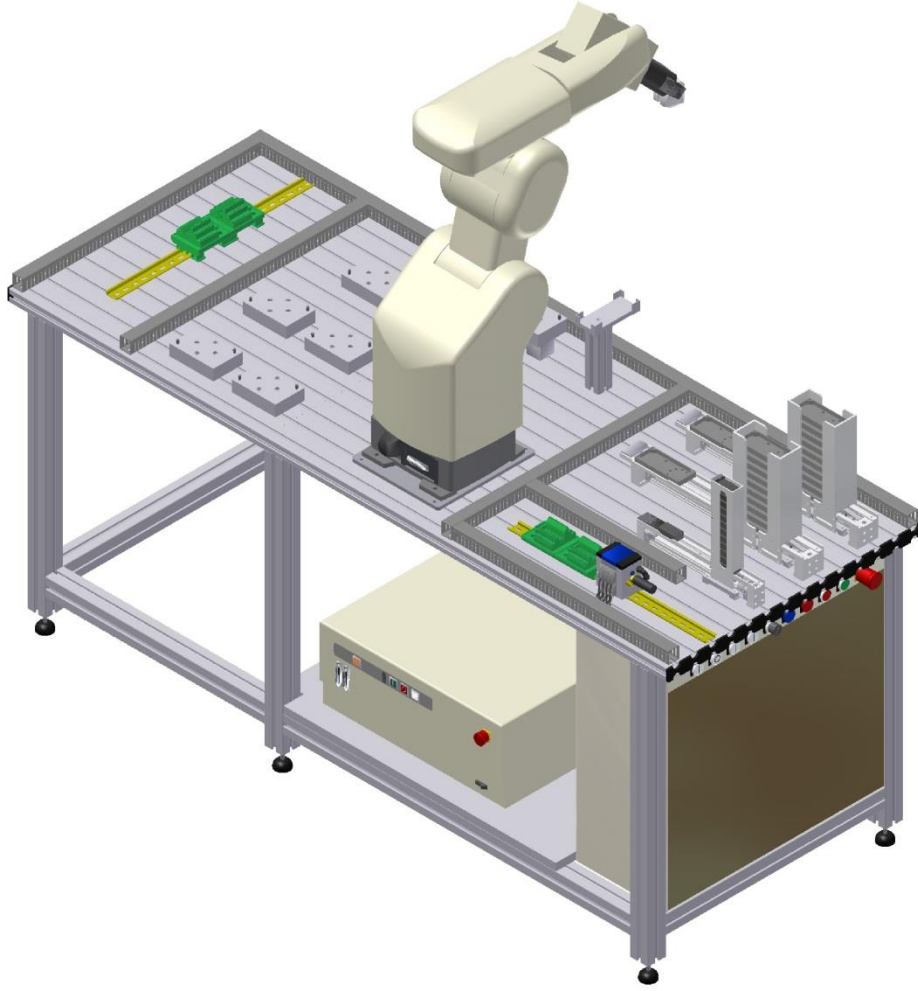
Hücrenin görevi, elektronik kart üzerine pıl bağlantısının yerleştirilmesi ve lehimlemesidir. Bu işlemler, pnömatik olarak değiştirilebilir tutucu (gripper) ve lehimleme eli bulunan bir robot tarafından gerçekleştirilir. Şekil 2.5’de Lehimleme hücresinin bilgisayar çizimi görülmektedir. Hücre denetleyicisi olarak S7-313C model PLC kullanılmış olup, FMS100 otomasyon sistemindeki ismi “İstasyon 2”dir.



Şekil 2-5. Lehimleme hücresi bilgisayar çizimi [79].

2.1.1.4. Montaj Hücresi (Assembly)

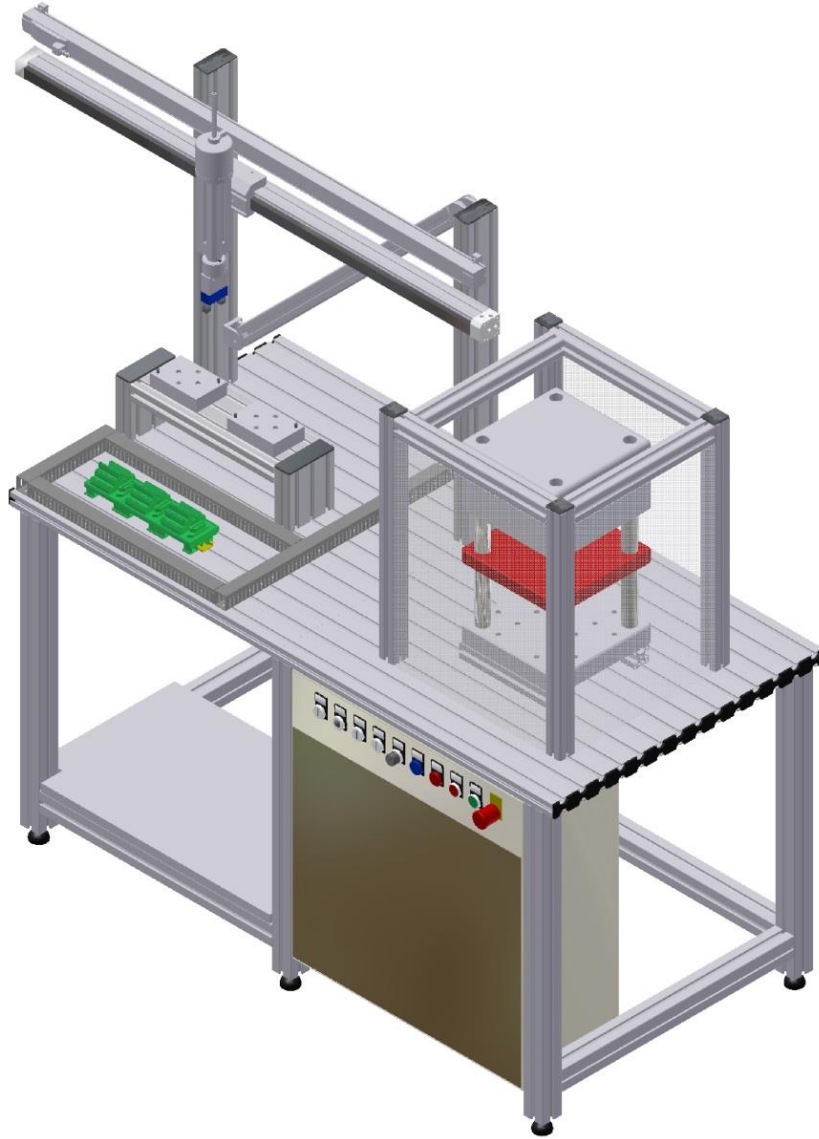
Bu hücre, elektrik tahrikli tutucu ele sahip bir robot aracılığıyla; ürüne ait işlenmiş alt/üst kapak, elektronik devre, metal pin gibi iş-parçaları ile 9V pili bir araya getirme görevini yürütür. Bilgisayar çizimi Şekil 2-6'daki gibi olan hücrede denetleyici olarak IPC kullanılmıştır. FMS100 otomasyon sisteminde “İstasyon 3” olarak isimlendirilmiştir.



Şekil 2-6. Montaj hücresi bilgisayar çizimi [79].

2.1.1.5. Hidrolik-Baskı Hücresi (Hydraulic)

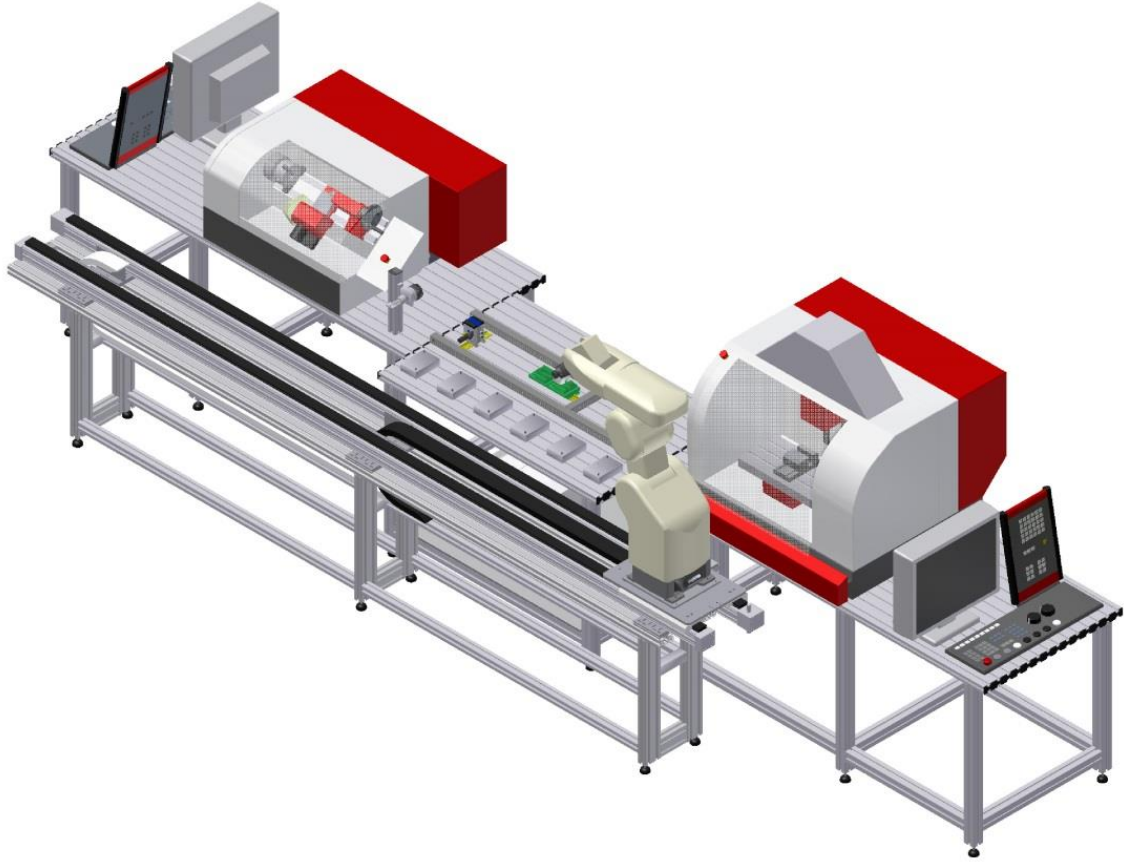
Bilgisayar çizimi Şekil 2-7’de görülmekte olan hücrenin görevi, montaj sırasında alt-üst kapaklar arasına yerleştirilmiş olan metal pinleri baskılayarak sıkıştırmaktır. Böylelikle ön-montajı yapılmış ürünün ürün montajı tamamlanmış olur. FMS100 otomasyon sisteminde “İstasyon 4” olarak isimlendirilmiştir. Hücre denetleyicisi olarak kullanılan S7-314C-2DP model PLC aynı zamanda taşıma sisteminin kontrolünü de sağlamaktadır.



Şekil 2-7. Hidrolik Baskı hücresi bilgisayar çizimi [79].

2.1.1.6. CNC-Çifti Hücresi (CNC-Coupling)

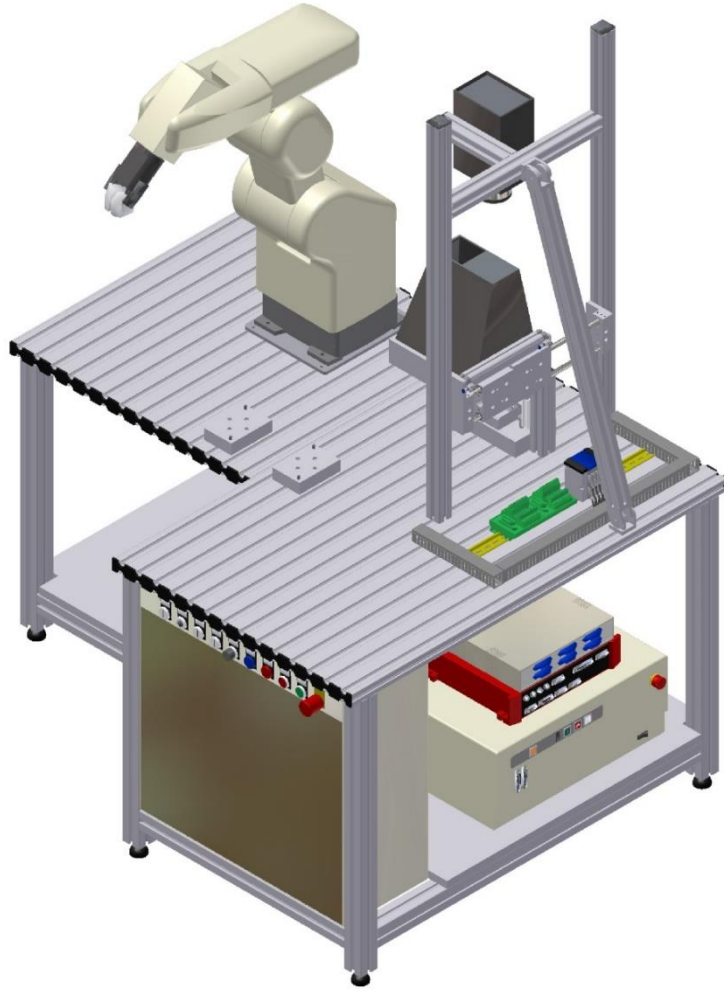
Ürüne ait üst/alt kapaklarda delik/kanal açma gibi CNC freze işleri ile kapakları birleştirmek için kullanılan metal pinlerin işlenmesi gibi CNC torna işleri bu hücrede yürütülür. Bilgisayar çizimi Şekil 2-8’deki gibi olan hücrenin bünyesinde bir doğrusal-yatak bulunmaktadır. Yatağa monte edilmiş bir robot aracılığıyla iş-parçalarını taşıma sisteminden alma, parçaları torna ve freze tezgahlarına yüklenme/boşaltma görevleri yürütülür. Hücre bünyesindeki robot, doğrusal-yatak, CNC torna ve freze gibi cihazların denetimi, hücre bünyesindeki CR1 model robot denetleyici tarafından yürütülür. Bu nedenle denetim için özel bir PLC kullanılmamış olan hücre, FMS100 otomasyon sisteminde “İstasyon 5” olarak isimlendirilmiştir.



Şekil 2-8. CNC-Çifti hücresi bilgisayar çizimi [79].

2.1.1.7. Görüntü İşleme Hücresi (Vision)

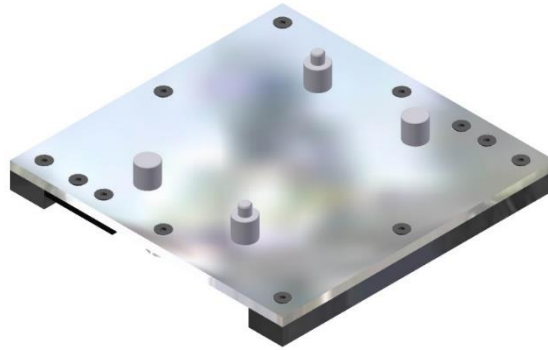
FMS sistemi bünyesinde üretilen ürünle ilgili kalite kontrol işlemleri sistemdeki bu son hücrede yürütülür. Robot tarafından taşıma sisteminden alınan ve montajı bitmiş durumdaki ürünler, öncelikle test alanına alınır ve ürüne ait elektronik kartın fonksiyon testleri yapılır. Aynı zamanda görüntü işleme yöntemleri ile ürünün kapaklarında yapılan delik açma gibi mekanik işlemlerin doğru yapılıp yapılmadığı da yine bu hücrede kontrol edilir. Bilgisayar çizimi Şekil 2-9'deki gibi olan hücrede denetleyici olarak IPC kullanılmış olup; görüntü işleme işleri için özel tasarım bir endüstriyel PC kullanılmıştır. FMS100 otomasyon sisteminde “İstasyon 6” olarak isimlendirilmiştir.



Şekil 2-9. Görüntü İşleme istasyonu bilgisayar çizimi [79].

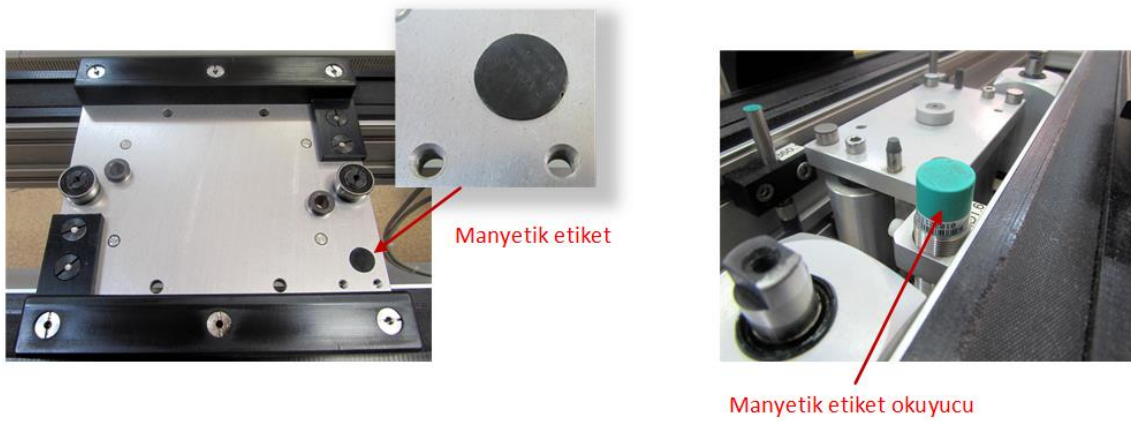
2.1.1.8. Ana-Palet

Taşıma sistemi bünyesindeki konveyör üzerine hareket ederek, çeşitli tipteki iş-parçalarını taşıyan yardımcı paletlerin istasyonlar arası taşınmasını sağlar. Şekil 2-10’da görülmekte Ana-Palet üzerinde yardımcı paletleri konumlandırmak için 4 adet ayak bulunur.



Şekil 2-10. Ana palet bilgisayar çizimi [79].

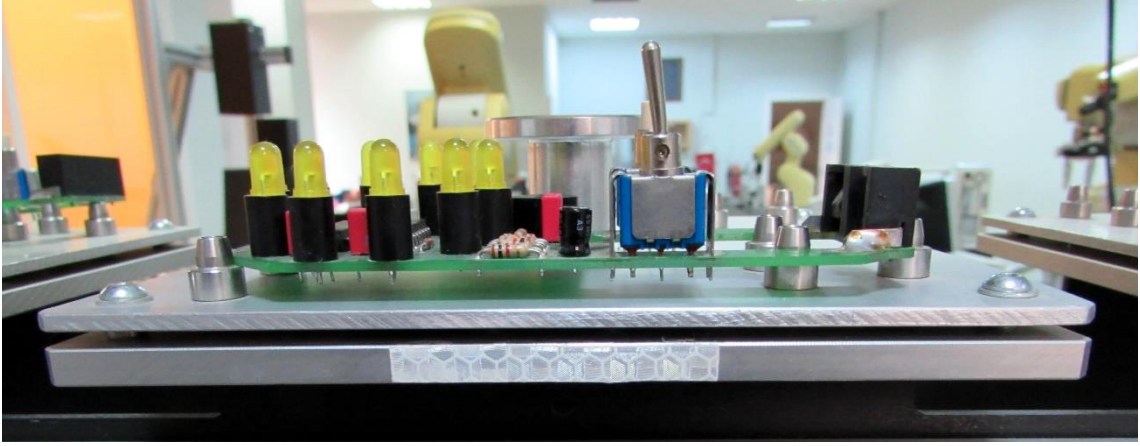
Fiziksel olarak birbirlerinin aynı olan Ana-Paletleri ayırt edebilmek için manyetik etiket sistemi kullanılmaktadır. Şekil 2-11’de görülmekte olan sistem, etiket okuyucu ve birbirinden farklı olacak şekilde kodlanmış manyetik etiketlerden oluşmaktadır. Her paletin altında kendine özel olarak kodlanmış bir manyetik etiket bulunmaktadır. İstasyonlarda bulunan manyetik etiket okuyucu sayesinde, istasyona gelen paletin altında bulunan etiket kodu okunur. Böylelikle hangi paletin hangi istasyona vardığı SCADA tarafından sürekli takip edilir. FMS100 bünyesinde toplam 10 adet Ana-Palet bulunmaktadır.



Şekil 2-11. Ana-Palet altındaki manyetik etiket ve her istasyonda bulunan etiket okuyucu görüntüleri.

2.1.1.9. Yardımcı Palet

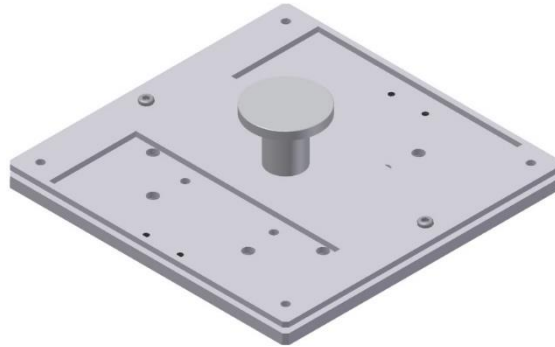
FMS100 sisteminin ürettiği örnek ürünü oluşturan işlenmiş/işlenmemiş durumlardaki üst/alt kapak, elektronik kart ve metal pin gibi iş-parçalarını taşımak için kullanılırlar. İş-parçaları, Stok hücresinde bu paletler üzerinde tutulur. Aynı zamanda iş-parçalarının istasyonlar arası taşınması da yine bu paletlerin, Ana-Palet üzerine yerleştirilmesiyle sağlanır. Her bir yardımcı paletin alt tarafında, paletin Ana-Palet üzerindeki ayaklar üzerinde sabit kalabilmesi için açılmış delikler bulunur. Paletlerin orta bölümünde yer alan dairesel kol ise robotların yardımcı paletleri tutup taşıyabilmesi içindir. Ayrıca her bir yardımcı paletin ön ve arka yüzünde Şekil 2-12’de görülmekte olan ve kedi-gözü olarak isimlendirilen ışık yansıtıcı etiketler bulunur.



Şekil 2-12. Yardımcı palet üzerindeki optik yansıtıcı görüntüsü.

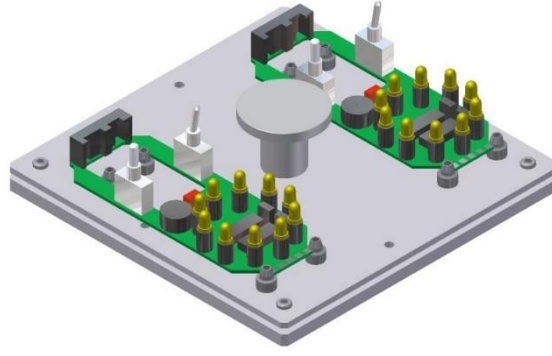
Böylelikle Stok hücresinde yer alan Kartezyen konumlandırma sistemindeki (Teleskop) optik algılayıcı, yansıtıcı etiketlerden yansıyan ışığı kullanarak paletleri algılar. Yardımcı paletlerin Stok hücresindeki raflara yerleştirilmesi veya raflardan alınması işlemleri sırasında, ilgili rafın dolu mu boş mu olduğu kontrol edilmesi sağlanmış olur. FMS bünyesinde değişik özelliklere sahip üç farklı tipte yardımcı palet bulunur:

Şekil 2-13’de görülmekte olan ve Palet-2ED olarak isimlendirilen yardımcı palet, rulete ait işlenmiş/işlenmemiş alt ve üst kapaklar ile montajı bitmiş/bitmemiş ruletleri taşır. Fiziksel boyutlar açısından işlenmiş ve işlenmemiş kapaklar arasında herhangi bir fark yoktur. Sistemde toplam 26 adet Palet-2ED bulunur.



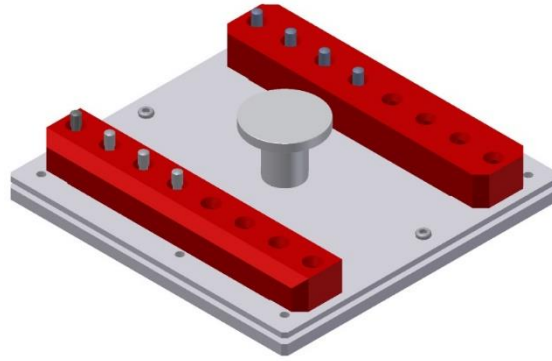
Şekil 2-13. Palet 2ED isimli yardımcı paletin bilgisayar çizimi [79].

Örnek ürüne ait işlenmiş/işlenmemiş elektronik kartlar çiftler halinde Şekil 2-14’de görülmekte olan Palet-2PCB üzerinde taşınır. Sistemde toplam 13 adet Palet-2PCB bulunur.



Şekil 2-14. Palet 2PCB isimli yardımcı paletin bilgisayar çizimi [79].

Örnek ürüne ait alt ve üst kapakları birleştirmede kullanılan metal pinler Şekil 2-15’de görülmekte olan Palet 8PIN üzerinde taşınır. Paletin üzerinde iki sıra halinde bulunan pin yuvalarından sağ taraftaki, işlenmemiş pinleri tutacak şekilde geniş; sol taraftaki ise işlenmiş pinleri tutacak şekilde dar üretilmiştir. İşlenmemiş pinler düz alüminyum silindir şeklinde olup çapları kalındır. İşlenmiş pinler ise kapaklardaki deliklere girip sıkışmak üzere özel olarak şekillendirilmiş ve daha ince çaplıdır. Bu nedenle işlenmiş ve işlenmemiş pinler birbirleri ile aynı yuvaya yerleştirilmezler. Sistemde toplam 9 adet Palet-8PIN bulunur.



Şekil 2-15. Palet 8PIN isimli yardımcı paletin bilgisayar çizimi [79].

2.1.1.10. Haberleşme Şeması

FMS100 gibi birbirinden farklı özelliklere sahip imalat hücreleri ve cihazların bulunduğu bir yapıda doğal olarak farklı endüstriyel haberleşme protokolleri kullanılmaktadır. Bu protokoller Profibus FMS, Profibus DP, AS-i Bus, Ethernet ve RS232’dir. Hücreler arası iletişim temel olarak Profibus FMS üzerinden sağlanır. Taşıma Sistemi üzerindeki istasyon duraklarında bulunan ve Ana-Palet’lerin takibini sağlayan Pepperl & Fuchs manyetik etiket sisteminin elemanları ise Profibus DP üzerinden haberleşir. Taşıma sisteminde bulunan algılayıcı ve eyleyici cihazlar ise AS-i Bus üzerinden haberleşir. CNC

çifti istasyonunda bulunan Melfa RV-2AJ model robot SCADA istasyonu ile Ethernet üzerinden haberleşir. Yine CNC Çifti istasyonunda bulunan Turn-55 torna ve Mill-55 freze makinaları SCADA istasyonu ile RS232 üzerinden haberleşir.

Şekil 2.16’da görülmekte olan haberleşme ağındaki her bir cihaza, kullandığı protokole göre sabit (statik) adresler atanmıştır. Böylelikle sistem bünyesindeki kontrol ve haberleşme yapısı sabitlenmiştir. Ağ bünyesindeki elemanlara erişim için atanan adresler protokollerine göre aşağıdaki gibidir:

Profibus FMS Adresleri:

- Stok istasyonu #8
- Lehimleme istasyonu #34
- Montaj istasyonu #6
- Hidrolik-Baskı istasyonu #36
- CNC-Çifti istasyonu #32
- Görüntü İşleme istasyonu #16
- SCADA istasyonu #20

Profibus DP Adresleri:

- Taşıma Sistemi Manyetik Etiket Sistemi (istasyon 1-3 arası) #3
- Taşıma Sistemi Manyetik Etiket Sistemi (istasyon 4-6 arası) #4
- Hidrolik-Baskı istasyonu PLC (SPC200) #5
- Hidrolik-Baskı istasyonu ve Taşıma Sistemi PLC (S7300 CP343-2) #2

Ethernet Adresleri

- SCADA istasyonu SCADA Bilgisayarı: 192.168.10.20
- SCADA istasyonu ERP Bilgisayarı: 192.168.10.1
- CNC-Çifti istasyonu Melfa RV-2AJ Robot Kontrolörü: 192.168.10.10:9002

RS232 Bağlantı Yuvaları

- SCADA Bilgisayarı COM1: Turn55 Torna
- SCADA Bilgisayarı COM2: Mill55 Freze



Şekil 2-18. Konveyör hareket yönüne göre yerleştirilmiş Ana-Palet'lerin görüntüsü.

- Sistemdeki Lehimleme, Montaj, CNC-Çifti hücrelerine 4 bar (en fazla 6 bar) basınçta hava sağlanır;
- CNC makinalarının ve robotların ana güç anahtarları “OFF” konumuna, robot el terminalleri (Teach Pendant) “Disable” konumuna getirilir;
- Tüm imalat hücrelerindeki “AUTO/MAN” anahtarları, Şekil 2-19 (a) ve (b)’de görüldüğü gibi “AUTO” konumuna alınır;



(a)



(b)

Şekil 2-19. İmalat hücreleri otomatik ve elle çalışma tipi anahtarlarının görüntüsü: Stok hücresi (a), diğer hücreler (b).

- Acil durdurma butonları kontrol edilir, eğer basılı olan (ON) varsa, çekili duruma (OFF) getirilir;
- Hücre kontrol panosundaki “Controler On” butonuna basılarak hücre denetleyici aktif hale getirilir;
- Hücre kontrol panosundaki “Reset” butonu ışığı yanıp sönmeye başladığı anda, hücredeki robot denetleyici ana güç anahtarı “ON” konumuna getirilir;
- Hücre kontrol panosundaki “Reset” butonuna birkaç saniye basılı tutulur, robot kendini yeniden başlatır ve başlangıç noktasına doğru hareket eder, hücre kontrol panosundaki “Reset” butonu ışığı sabit olarak yanmaya başlar;

- H cre kontrol panosundaki “Automatic On” butonu ışıđı yanıp s nmeye bařladıđı anda butona basılır ve h cre FMS100 ile entegre olarak  alıřmaya bařlar, buton ışıđı sabitlenir;
- CNC torna ve freze tezgahlarına ait bilgisayarlar a ılır, WinNC32 programı  alıřtırılır;
- CNC torna ve freze tezgahlarının  alıřma tipi se im anahtarı (mode switch) “FMS” konumuna alınır, cihazların ana g   anahtarları “ON” durumuna getirilir.

2.1.1.11.2. Stok H cresinin Devreye Alınması

FMS100’de herhangi bir  retim iřlemine bařlamadan  nce, Stok h cresindeki raflarda bulunan malzemeler ile SCADA bilgisayarındaki ERP yazılımı stok verilerinin (AS-RS Admin) eřleřmiř olması gereklidir.  rneđin Stok h cresindeki 8 numaralı rafta 60430 seri numaralı iřlenmemiř 8’li pin bulunurken, ERP yazılımı veritabanında 8 numaralı rafta 62400 seri numaralı boř palet varmiř gibi g r lebilir. Bu gibi uyumsuzluklar neticesinde FMS100  alıřırken, Stok h cresi, yanlıř iř-par alarını yanlıř imalat h crelerine g nderebilir. Bu durumda iř-par aları zarar g rebileceđi gibi h crede bulunan donanımlar da hasar g rebilir.

Stok h cresi FMS100 ile bir b t n olarak  alıřmaya bařlamadan  nce yapılması gereken kontrol ve d zenlemeler řu řekildedir:

- Eđer Teleskop doluysa, bořaltılır;
-  alıřma tipi se im anahtarı otomatik (AUTO) konumuna alınır;
- Reset butonu lambası yanıp s nmeye bařladıđı anda Reset butonuna basılır, Teleskop bařlangı  konumunu bulmak  zere harekete ge er;
- Automatic On butonu lambası yanıp s nmeye bařladıđı anda butona basılır ve h cre FMS100 ile eř zamanlı  alıřmaya bařlar, buton ışıđı sabit yanmaya bařlar;
- Stok h cresindeki raflar, bařlangı ta ařađıdaki Tablo 2-1’e g re d zenlenir.

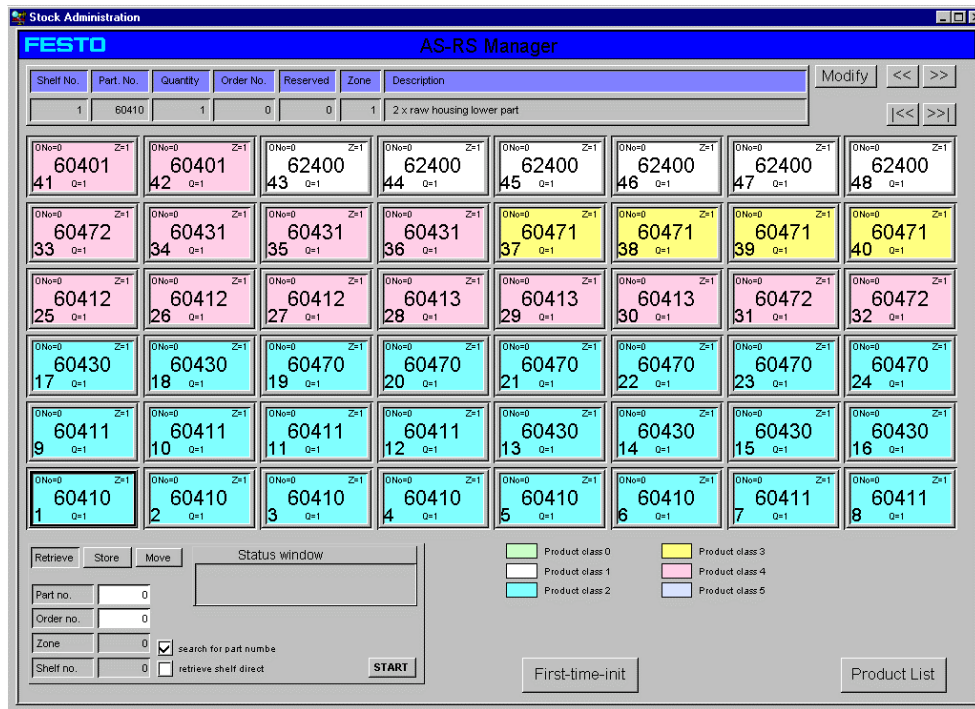
Tablo 2-1. Stok h cresindeki 48 adet rafta bulunan iř-par aları dizilimi.

Raf No	Seri No	Palet Adı	A�ıklama
1	60410	Palet-2ED	2 x iřlenmemiř alt kapak
2	60410	Palet-2ED	2 x iřlenmemiř alt kapak
3	60410	Palet-2ED	2 x iřlenmemiř alt kapak
4	60410	Palet-2ED	2 x iřlenmemiř alt kapak
5	60410	Palet-2ED	2 x iřlenmemiř alt kapak
6	60410	Palet-2ED	2 x iřlenmemiř alt kapak

7	60411	Palet-2ED	2 x işlenmemiş üst kapak
8	60411	Palet-2ED	2 x işlenmemiş üst kapak
9	60411	Palet-2ED	2 x işlenmemiş üst kapak
10	60411	Palet-2ED	2 x işlenmemiş üst kapak
11	60411	Palet-2ED	2 x işlenmemiş üst kapak
12	60411	Palet-2ED	2 x işlenmemiş üst kapak
13	60430	Palet-8PIN	8 x işlenmemiş pin
14	60430	Palet-8PIN	8 x işlenmemiş pin
15	60430	Palet-8PIN	8 x işlenmemiş pin
16	60430	Palet-8PIN	8 x işlenmemiş pin
17	60430	Palet-8PIN	8 x işlenmemiş pin
18	60430	Palet-8PIN	8 x işlenmemiş pin
19	60470	Palet-2PCB	2 x lehimlenmemiş baskı devre
20	60470	Palet-2PCB	2 x lehimlenmemiş baskı devre
21	60470	Palet-2PCB	2 x lehimlenmemiş baskı devre
22	60470	Palet-2PCB	2 x lehimlenmemiş baskı devre
23	60470	Palet-2PCB	2 x lehimlenmemiş baskı devre
24	60470	Palet-2PCB	2 x lehimlenmemiş baskı devre
25	60412	Palet-2ED	2 x işlenmiş alt kapak
26	60412	Palet-2ED	2 x işlenmiş alt kapak
27	60412	Palet-2ED	2 x işlenmiş alt kapak
28	60413	Palet-2ED	2 x işlenmiş üst kapak
29	60413	Palet-2ED	2 x işlenmiş üst kapak
30	60413	Palet-2ED	2 x işlenmiş üst kapak
31	60472	Palet-2PCB	2 x lehimlenmiş ve test edilmiş baskı devre
32	60472	Palet-2PCB	2 x lehimlenmiş ve test edilmiş baskı devre
33	60472	Palet-2PCB	2 x lehimlenmiş ve test edilmiş baskı devre
34	60431	Palet-8PIN	8 x işlenmiş pin
35	60431	Palet-8PIN	8 x işlenmiş pin
36	60431	Palet-8PIN	8 x işlenmiş pin
37	60471	Palet-2PCB	2 x lehimlenmiş fakat test edilmemiş baskı devre
38	60471	Palet-2PCB	2 x lehimlenmiş fakat test edilmemiş baskı devre
39	60471	Palet-2PCB	2 x lehimlenmiş fakat test edilmemiş baskı devre
40	60471	Palet-2PCB	2 x lehimlenmiş fakat test edilmemiş baskı devre
41	60401	Palet-2ED	2 x ön-montajı yapılmış, pin çakılmamış ürün
42	60401	Palet-2ED	2 x ön-montajı yapılmış, pin çakılmamış ürün

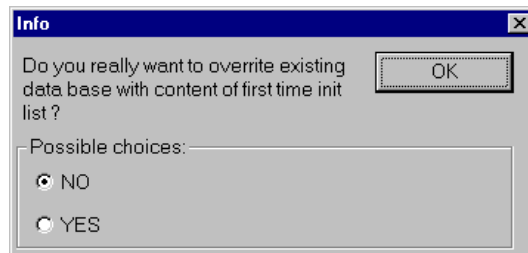
43	62400	Palet-2ED	Montajı biten ürünler ve kapaklar için boş palet
44	62400	Palet-2ED	Montajı biten ürünler ve kapaklar için boş palet
45	62400	Palet-2ED	Montajı biten ürünler ve kapaklar için boş palet
46	62400	Palet-2ED	Montajı biten ürünler ve kapaklar için boş palet
47	62400	Palet-2ED	Montajı biten ürünler ve kapaklar için boş palet
48	62400	Palet-2ED	Montajı biten ürünler ve kapaklar için boş palet

- Sonrasında COSIMIR Factory Control yazılımının Şekil 2-20’de görülmekte olan Stok Yönetimi (AR-RS Manager) penceresinden ilk kez başlatma (First-time-init) butonuna basılır.



Şekil 2-20. Stok yönetim ekranı “Stok Administration” görüntüsü.

- Ekran Şekil 2-21’deki uyarı mesajı gelir. Mesaj içeriğinde tüm stok kayıtlarının silineceği ve başlangıç durumuna döneceği bilgisi yer alır. “YES” tuşuna basıp onay verilmesiyle birlikte stok kayıtları da Tablo 2-1’de gösterilen raf dizilimine göre düzenlenir.



Şekil 2-21. “First-time-init” butonu uyarı mesajı.

2.1.1.11.3. Lehimleme Hücresinin Devreye Alınması

Lehimleme hücresi bünyesinde RV-E2 model robot ve havya denetleyicisi bulunmaktadır. Bu hücredeki robot tarafından pnömatik olarak değiştirilebilen iki farklı el kullanılmaktadır. Bunlardan biri pnömatik tutucu el (pneumatic gripper hand) diğeri ise havyalı el (soldering hand)'dir. Hücre FMS100 ile bir bütün olarak çalışmaya başlamadan önce yapılması gereken kontrol ve düzenlemeler şu şekildedir:

- İmalat hücresi içerisindeki tezgahlar ve palet kabul alanları boşaltılır, varsa önceden kalan iş-parçaları ve yardımcı paletler kaldırılır;
- Bölüm 2.1.1.11.1'de anlatılan işlem basamakları tek tek tamamlanarak hücre otomatik çalışma tipine getirilir;
- Robot ellerinin olması gerektikleri alanda olup olmadığı kontrol edilir;
- Havya uç temizleme süngerinin nemli olduğu (ıslak olmadığı) kontrol edilir, su haznesi boşsa doldurulur;
- Yeterli miktarda lehim teli bulunup bulunmadığı kontrol edilir;
- Havya denetleyici çalıştırılır ve havyanın uygun sıcaklığa ulaşabilmesi için yaklaşık 1 dk. beklenir;
- Pil konektörü magazininin dolu olduğu kontrol edilir.

2.1.1.11.4. Montaj Hücresinin Devreye Alınması

Hücre FMS100 ile bir bütün olarak çalışmaya başlamadan önce yapılması gereken kontrol ve düzenlemeler şu şekildedir:

- İmalat hücresi içerisindeki tezgahlar ve palet kabul alanları boşaltılır, varsa önceden kalan iş-parçaları ve yardımcı paletler kaldırılır;
- Bölüm 2.1.1.11.1'de anlatılan işlem basamakları tek tek tamamlanarak hücre otomatik çalışma tipine getirilir;
- Üst/alt kapak magazinleri ile pil magazininin dolu olduğu kontrol edilir.

2.1.1.11.5. Hidrolik-Baskı Hücresinin Devreye Alınması

Hücre FMS100 ile bir bütün olarak çalışmaya başlamadan önce yapılması gereken kontrol ve düzenlemeler şu şekildedir:

- İmalat hücresi içerisindeki tezgahlar ve palet kabul alanları boşaltılır, varsa önceden kalan iş-parçaları ve yardımcı paletler kaldırılır;
- Bölüm 2.1.1.11.1'de anlatılan işlem basamakları tek tek tamamlanarak hücre otomatik çalışma tipine getirilir;

- H cre masası altındaki Şekil 2-22’de g r lmekte olan yeşil butona basılarak hidrolik pompa  alıřtırılır.



Şekil 2-22. Hidrolik pompa  alıřtırma (yeşil) ve durdurma (kırmızı) butonları.

2.1.1.11.6. CNC- ifti H cresinin Devreye Alınması

H cre FMS100 ile bir b t n olarak  alıřmaya bařlamadan  nce yapılması gereken kontrol ve d zenlemeler řu şekildedir:

- CNC torna ve frezede takılı durumda kalmıř herhangi bir iř-par ası varsa  ıkartılır ve tezgahlar bořaltılır;
- İmalat h cresi i erisindeki palet kabul alanları bořaltılır, varsa  nceden kalan yardımcı paletler kaldırılır;
- CR1 robot denetleyici  zerindeki anahtar Şekil 2-23’de g r ld ę  gibi Auto (Ext.) konumuna alınır ve g   anahtarını a ık (ON) konumuna alınır.



Şekil 2-23. CR1 robot denetleyici  nden g r n ř .

- Şekil 2-24’de görülen panel üzerindeki acil durdurma butonu çekilir (OFF), ardından anahtar önce 0’dan 1’e sonra 1’den 0’a getirilerek CR1 robot denetleyici aktif hale getirilir;



Şekil 2-24. CNC-Çifti istasyonu kontrol paneli görüntüsü.

- CNC torna ve freze bilgisayarları açılır ve WinNC32 yazılımı çalıştırılır;
- Şekil 2-25’de görülmekte olan CNC makinaları çalışma tipi seçim anahtarı “FMS” konumuna alınır, ana güç anahtarı “ON” konumuna alınarak makinalar çalıştırılır;



Şekil 2-25. CNC cihazlarının yan tarafında bulunan güç ve çalışma tipi seçim anahtarları görüntüsü.

- Doğrusal kızak üzerindeki robotun başlangıç konumuna gidebilmesi için Şekil 2-24’de görülen panel üzerindeki “Reset” butonuna basılı tutulur
- Hücrenin otomatik tipte çalışması için panel üzerindeki “Start” butonuna basılır.

2.1.1.11.7. Görüntü İşleme Hücresinin Devreye Alınması

Hücre FMS100 ile bir bütün olarak çalışmaya başlamadan önce yapılması gereken kontrol ve düzenlemeler şu şekildedir:

- İmalat hücresi içerisindeki tezgahlar ve palet kabul alanları boşaltılır, varsa önceden kalan iş-parçaları ve yardımcı paletler kaldırılır;
- Bölüm 2.1.1.11.1’de anlatılan işlem basamakları tek tek tamamlanarak hücre otomatik çalışma tipine getirilir;

2.1.1.11.8. Taşıma Sisteminin Devreye Alınması

Taşıma sisteminin düzenlenmesi için herhangi özel bir işlem gerekmemektedir. Sadece sistemdeki konveyör üzerindeki Ana-Palet’lerin oklarının, konveyörün hareket yönünü gösterecek şekilde konumlandıkları ve palet sayısının en az 9 adet olduğu kontrol edilir. Bunun dışında her bir istasyonda bulunan palet tutucu ve durdurucu gibi eyleyicilerin, Şekil 2-26’de görülmekte olan AS-i çalışanları üzerinden önceden elle aktif edilip edilmediği kontrol edilmelidir. AS-i çalışanları üzerindeki anahtarların hepsi yukarı konuma alınarak eyleyiciler pasif duruma getirilir.



Şekil 2-26. Her istasyonda bulunan ve pnömatik vana kontrolünü yapan AS-i çalışanı görüntüsü.

2.1.1.11.9. COSIMIR FACTORY CONTROL Yazılımının Başlatılması

Tüm hücreler önceki bölümlerde anlatıldığı üzere başarılı bir şekilde otomatik çalışma tipinde başlatıldıktan sonra, SCADA bilgisayarında bulunan ve tüm sistemin kontrolü ile üretimin yönetilmesini sağlayan SCADA yazılımı COSIMIR Factory Control çalıştırılabilir. FMS100 sistemi için üretici firma tarafından önceden hazırlanmış olan COSIMIR projesi “Marmara.lpj” açılır. Yazılım, üretim (production) moduna alınarak sistem ile çalışmaya başlanır.

2.1.1.11.10. FMS100’ü Devreden Çıkarma

FMS100 sistemi ile çalışma sona erdiği zaman aşağıdaki işlem basamakları sırasıyla uygulanmalıdır:

- COSIMIR Factory Control yazılımında üretim durdurulur;
- Tüm istasyonlar ile iletişim kesilir;
- Yazılım ve SCADA ile ERP bilgisayarları kapatılır;
- CNC torna ve freze tezgahlarına ait bilgisayarlardaki WinNC32 yazılımı durdurulur ve bilgisayarlar kapatılır;
- CNC torna ve freze tezgahları kapatılır;
- Hidrolik-Baskı hücresindeki hidrolik pompa kapatılır;
- Robot denetleyicilerin ana güç anahtarları kapalı (OFF) duruma getirilir;
- İmalat hücre panolarındaki acil stop butonlarına basılarak hücre denetleyicileri kapatılır;
- Ana panodan FMS100’ün ana gücü kesilir.

*** Robot denetleyicilerin hafızalarındaki eklem sıfır noktalarına ait konum bilgilerinin saklanabilmesi için her bir denetleticide bulunan hafıza pillerinin belirli bir saat ömrü vardır. Denetleyicilerin enerjisi kesildiğinde, bu pillerdeki enerji kullanılmaya başlar. Piller tükendiği zaman, robot eklemleri sıfır konum bilgilerinin silinme ihtimali vardır. Bu nedenle kontrolörlerin enerjisi kesildiği zaman, ara ara el terminali üzerinden kalan pil ömürlerinin kontrol edilmesi gerekmektedir.

2.1.2. FMS100 Kullanımı

Bu bölümde, önceden devreye alınmış durumdaki FMS100 sisteminin kullanımı ile ilgili bilgiler verilmektedir. Sistemin oldukça karmaşık yapılı olması ve bünyesinde robot ile hidrolik-baskı gibi sakatlanmalara ve incinmelere yol açabilecek cihazlar bulunması sebebiyle; hem iş güvenliğini sağlamak, hem de sistemin arızalanma ihtimalini en aza

indirebilmek için, laboratuvarında çeşitli davranış ve kullanım kuralları tanımlanmıştır. FMS100 ile çalışmaya başlamadan önce bu kuralların iyice okunması, özümsemesi ve uygulanması oldukça önemlidir.

2.1.2.1. Laboratuvar Davranış ve FMS100 Kullanım Kuralları

- Laboratuvara gelirken, konveyör bantına sıkışabilecek ve yaralanmalara sebep olabilecek tipte sarkan kuşakları, püskülleri, vb. bulunan kıyafetler giyilmemelidir.
- Laboratuvarın ortam sıcaklığı genellikle yüksektir. Bu nedenle laboratuvarında çalışmaya başlamadan önce mont, kaban, vb. kıyafetler çıkarılarak uygun yerlere asılmalıdır.
- FMS100 ve imalat hücreleri ile sıvı temas ettirilmemeli, laboratuvara sıvı herhangi bir içeceklerle girilmemelidir.
- Laboratuvarında yüksek sesle konuşmamalı, diğer kişiler ile şakalar yapılmamalıdır.
- Herhangi bir imalat hücresiyle çalışmaya başlamadan önce, hücreye ait tanıtım kitapçığının iyice okunması gereklidir.
- Bünyesinde robot bulunan hücreler ile çalışırken en az 2 kişi olunmalıdır. Robotun hareketi sırasında bir kişi robot denetleyici üzerindeki acil durdurma butonuna her an basacakmış gibi hazırda beklerken, diğeri robota kumanda etmelidir.
- Robotun çalışması esnasında hareket uzayında bulunmamaya dikkat edilmelidir.
- İmalat hücrelerine ait masalara dayanmamalı, masa üzerine kitap, defter, vb. yabancı cisimler konulmamalıdır.
- Gerekmedikçe imalat hücresi içerisindeki anahtar, buton, algılayıcı, vb. elemanlar ellenmemelidir.
- Gerekmedikçe imalat hücresi içerisindeki kablolar ile oynanmamalıdır.
- FMS100'ün büyük ve birçok farklı hücreden oluştuğu, bu nedenle sistem üzerinde çalışan birçok farklı öğrenci grubunun da bulunacağı göz önüne alınmalı; yüksek sesle konuşma gibi dikkat dağıtıcı hareketlerden kaçınılmalıdır.
- Deney föyünde belirtilmeyen buton ve anahtarlara rastgele basılmamalıdır.
- Sınırlı sayıda olması sebebiyle sisteme ait iş-parçaları alı konulmamalıdır.

2.1.2.2. Genel İmalat Sürecinin Tanımlanması

FMS100'ün görevi örnek bir elektrikli el cihazı olan elektronik ruleti üretmektir. FMS100'ü üretici firma tarafından sistem üç farklı el cihazı üretmek üzere tasarlanmış olup, şu an sadece Şekil 2-2 (a)'da görülmekte olan elektronik rulet üretimi yapılabilmektedir.

Bir elektronik el cihazının imalatı birçok mekanik ve elektronik alt imalat süreçlerine ayrılabilir. Alt imalat süreçleri ise elde bulunan hammadde ve imalat cihazı gibi kaynaklara bağımlıdır. FMS100'ün örnek ürünü olan elektronik ruletin imalatı için gerekli alt imalat süreçleri şunlardır:

- 2 adet işlenmemiş alt kapak taşıyan yardımcı palet Stok hücrelerinden alınarak CNC-Çifti hücrelerine gönderilir. Burada CNC freze tezgahında sırasıyla temel işlemlerden geçirilir.
- İşlenmiş iş-parçaları ileride kullanılmak üzere Stok hücrelerine geri gönderilir.
- Üst kapakların işlenmesi süreci, alt kapaklar ile aynıdır. Metal pinlerin işlenmesindeki tek fark ise işlemenin CNC torna tezgahında yapılmasıdır.
- Elektronik baskı devreler montajdan önce lehimleme işlemine tabi tutulur. Lehimleme hücrelerinde robot tarafından önce pil konektörü yerleştirilir. Sonrasında yine robot tarafından havya kullanılarak, konektörün her iki yanı baskı devreye lehimlenir.
- Stok hücrendeki raflarda bulunan ve işlenmiş durumdaki iş-parçalarını taşıyan yardımcı paletler, Teleskop ile konveyör üzerindeki Ana-Palet üzerine aktararak Montaj hücrelerine taşınır. Montaj hücrelerinde her bir yardımcı palet robot tarafından, türüne göre önceden belirlenmiş çalışma alanlarına aktarılır. Sonrasında montaj tezgahına önce alt kapak yerleştirilir ve sonra içerisine elektronik kart yerleştirilir. Pil magazininden alınan pil kart üzerine bırakılır, yerine yerleşmesi için robot ile üzerine bastırılır. Üst kapak da kapatıldıktan sonra 4 adet metal pin sırasıyla deliklerine yerleştirilir.
- Montajı tamamlanan ruletler Hidrolik-Baskı hücrelerine gönderilir ve önceden yerleştirilmiş olan metal pinler bastırılırsa imalat tamamlanır.
- İmalatı tamamlanmış olan ruletler Stok hücrelerine geri gönderilerek ileride kullanılmak üzere saklanır.

2.1.2.3. Genel Alt İmalat Süreçlerinin Tanımlanması

FMS100'ün üretebildiği örnek ürün olan elektronik ruletin imalatını oluşturan alt imalat süreçleri genel hatlarıyla aşağıdaki Tablo 2-2'de verilmektedir. Sistemin kontrol ve yönetilmesini sağlayan COSIMIR Factory Control yazılımı ortamında oluşturulan "Marmara.lpj" projesinde, bu tabloda belirtilen tüm alt imalat süreçleri, süreç numaraları ile temsil edilmektedir.

Tablo 2-2. FMS100 bünyesinde tanımlı imalat süreçleri ve açıklamaları.

Süreç No	Alt İmalat Süreç Tanımı
401	2 adet alt kapağın işlenmesi süreci yürütülür. İşlenmemiş alt kapaklar, Stok hücrelerinden getirilerek CNC-Çifti hücrendeki freze tezgahında işlenir ve sonra tekrar Stok hücrelerine geri gönderilir.
402	2 adet üst kapağın işlenmesi süreci yürütülür. İşlenmemiş üst kapaklar Stok hücrelerinden getirilerek CNC-Çifti hücrendeki freze tezgahında işlenir ve sonra tekrar Stok hücrelerine geri gönderilir.
403	8 adet metal pinin işlenmesi süreci yürütülür. İşlenmemiş metal pinler Stok hücrelerinden getirilerek CNC-Çifti hücrendeki torna tezgahında işlenir ve sonra tekrar Stok hücrelerine geri gönderilir.
404	2 adet elektronik baskı devreye pil konektörünün lehimlenmesi süreci yürütülür. Lehimlenmemiş baskı devreler Stok hücrelerinden getirilerek Lehimleme hücrelerinde pil konektörü lehimlendikten sonra; ya Stok hücrelerine geri gönderilir ya da direk olarak 405 numaralı alt sürece yönlendirilir.
405	2 adet lehimlenmiş baskı devrenin elektronik fonksiyon testlerinin yapılma süreci yürütülür. Lehimlenmiş baskı devreler ya Stok hücrelerinden ya da direk olarak 404 numaralı alt süreçten getirilerek Görüntü İşleme hücrelerinde test edilir ve sonra Stok hücrelerine geri gönderilir.
406	2 adet elektronik ruletin montaj süreci yürütülür. Montaj hücrendeki ilgili palet kabul alanlarına önceden elle yerleştirilmiş olan lehimlenmiş baskı devreler ile işlenmiş metal pinler; hücre magazinlerinde hazır olarak bulunan alt/üst kapaklar ve piller kullanılarak montaj işlemi gerçekleştirilir. Montaj tamamlandıktan sonra ruletler Stok hücrelerine geri gönderilir.
407	2 adet elektronik ruletin montaj süreci yürütülür. Stok hücrelerinden sırayla lehimlenmiş baskı devreler, işlenmiş pinler, işlenmiş alt/üst kapaklar Montaj hücrelerine getirilerek montaj işlemi gerçekleştirilir. Montaj tamamlandıktan sonra ruletler Stok hücrelerine geri gönderilir.
408	2 adet ön-montajı yapılmış elektronik ruletin metal pinlerinin bastırılması ve montajın tamamlanması süreci yürütülür. Önceden parçaları bir araya getirilmiş olan ruletler, Stok hücrelerinden getirilerek Hidrolik-Baskı hücrelerinde metal pinleri tek tek bastırıldıktan sonra Stok hücrelerine geri gönderilir.

2.1.2.4. Alt-süreç 401 (Üst Kapak Frezeleme)

Kullanılan Birimler:

- Taşıma Sistemi
- Stok Hücresi
- CNC-Çifti Hücresi
- Robot Melfa RV-2AJ
- Mill 55 CNC Freze Tezgahı
- Palet-2ED

Açıklama:

Bu alt-süreçte, elektronik rulet cihazının üst kapakları CNC freze tezgahında işlenir. Süreç başlatıldığında, 2 adet işlenmemiş üst kapağı taşıyan yardımcı palet (Palet-2ED) Stok hücrelerinden alınarak, işlenmek üzere CNC-Çifti hücrelerine getirilir. Robot yardımcı paleti Ana-Palet üzerinden (konum 15) alarak imalat hücreesindeki konum 2'ye yerleştirir. Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır.

Konum 2'de bulunan yardımcı palet üzerindeki işlenmemiş üst kapaklardan ilkinin konum numarası 21, diğerinin ki ise 22 olarak belirlenmiştir. CNC freze tezgahının güvenlik kapısı açılır. Robotun freze tezgahının içerisine doğru olan hareketi başlamadan önce, güvenlik kapısının açık olup olmadığı tekrar kontrol edilir. İşlenmemiş üst kapak robot tarafından konum 21'den alınarak freze tezgahı içerisindeki mengeneyle yerleştirilir. Robot kolunun freze cihazından ayrılmasıyla birlikte kapı kapatılır ve seçilmiş olan NC programına göre üst kapak işlenmeye başlar.

İlk kapağın frezeleme işlemi tamamlandığında kapı tekrar açılır ve robotun freze içerisine doğru olan hareketi başlamadan önce kapının açık olup olmadığı yeniden kontrol edilir. İşlenmiş olan üst kapak yeniden konum 21'e taşınır. Sonrasında aynı işlemler konum 22'de bulunan diğer üst kapak için de tekrarlanır.

Her iki üst kapağın da freze işlemi tamamlandığında, istasyona boş bir Ana-Palet çağrılır. Boş Ana-Palet istasyona ulaştığında kenetlenir ve üzerine işlenmiş üst kapakları taşıyan yardımcı palet yüklenir. Son olarak yardımcı palet ileride kullanılmak üzere Stok hücrelerine gönderilir.

İşlem Basamakları:

- BAŞLAT (START)
- Stok hücrelerine boş bir Ana-Palet çağrılır;
- İstasyona gelen boş Ana-Palet kenetlendikten sonra, 60411 seri numaralı 2 adet işlenmemiş üst kapak taşıyan yardımcı palet, Ana-Palete yüklenir;
- Yardımcı palet CNC-Çifti hücrelerine gönderilir;
- Yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak CNC-Çifti hücreleri içerisindeki konum 2'ye taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- CNC freze tezgahının güvenlik kapısı açılır ve kapının açık olduğu kontrol edilir;

- İşlenmemiş ilk üst kapak konum 21'den alınarak freze tezgahı içerisindeki mengeneye yerleştirilir ve mengene sıkıştırılır;
- Güvenlik kapısı kapatılır, ilgili NC programı seçilir ve başlatılır;
- Frezeleme işlemi bitiminde güvenlik kapısı açılır ve işlenmiş iş-parçası konum 21'e geri konulur;
- Diğer üst kapak da aynı süreç içerisinde işlenir;
- Tüm üst kapaklar işlendikten sonra istasyona boş bir Ana-Palet çağrılır;
- Ana-palet geldikten sonra üzerine yardımcı palet yerleştirilir;
- Ana-palet Stok hücreğine geri gönderilir;
- Stok hücreğinde yardımcı palet 60413 seri numaralı olarak uygun rafta saklanır;
- SON (END)

2.1.2.5. Alt-süreç 402 (Alt Kapak Frezeleme)

Kullanılan Birimler:

- Taşıma Sistemi
- Stok Hücresi
- CNC-Çifti Hücresi
- Robot Melfa RV-2AJ
- Mill 55 CNC Freze Makinası
- Palet-2ED

Açıklama:

Bu alt-süreçte elektronik rulet cihazının alt kapakları CNC freze tezgahında işlenir. Süreç başlatıldığında, 2 adet işlenmemiş alt kapağı taşıyan yardımcı palet (Palet-2ED) Stok hücrelerinden alınarak, işlenmek üzere CNC-Çifti hücreğine getirilir. Robot yardımcı paleti Ana-Palet üzerinden (konum 15) alarak imalat hücrendeki konum 1'e yerleştirir. Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır.

Konum 1'de bulunan yardımcı palet üzerindeki işlenmemiş üst kapaklardan ilkinin konum numarası 11, diğerinin ki ise 12 olarak belirlenmiştir. CNC freze tezgahının güvenlik kapısı açılır. Robotun freze tezgahının içerisine doğru olan hareketi başlamadan önce, güvenlik kapısının açık olup olmadığı tekrar kontrol edilir. İşlenmemiş alt kapak robot tarafından konum 11'den alınarak freze tezgahı içerisindeki mengeneye yerleştirilir. Robot kolunun freze tezgahından ayrılmasıyla birlikte kapı kapatılır ve seçilmiş olan NC programına göre alt kapak işlenmeye başlar.

İlk kapağın frezeleme işlemi tamamlandığında kapı tekrar açılır ve robotun freze içerisine doğru olan hareketi başlamadan önce, kapının açık olup olmadığı yeniden kontrol edilir. İşlenmiş olan alt kapak yeniden konum 11'e taşınır. Sonrasında aynı işlemler konum 12'de bulunan diğer alt kapak için de tekrarlanır.

Her iki alt kapağın da frezeleme işlemi tamamlandığında istasyona boş bir Ana-Palet çağrılır. Boş Ana-Palet istasyona ulaştığında kenetlenir ve üzerine işlenmiş alt kapakları taşıyan yardımcı palet yüklenir. Son olarak yardımcı palet ileride kullanılmak üzere Stok hücrelerine gönderilir.

İşlem Basamakları:

- BAŞLAT (START)
- Stok hücrelerine boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60410 seri numaralı 2 adet işlenmemiş alt kapak taşıyan yardımcı palet, Ana-Palete yüklenir;
- Yardımcı palet CNC-Çifti hücrelerine gönderilir;
- Yardımcı palet, taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak CNC-Çifti hücrelerindeki konum 1'e taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- CNC freze tezgahının güvenlik kapısı açılır ve kapının açık olduğu kontrol edilir;
- İşlenmemiş ilk alt kapak konum 11'den alınarak freze tezgahı içerisindeki mengeneye yerleştirilir ve mengene sıkıştırılır;
- Güvenlik kapısı kapatılır, ilgili NC programı seçilir ve başlatılır;
- Frezeleme işlemi bitiminde güvenlik kapısı açılır ve işlenmiş iş-parçası konum 11'e geri konulur;
- Diğer alt kapak da aynı süreç içerisinde işlenir;
- Tüm alt kapaklar işlendikten sonra istasyona boş bir Ana-Palet çağrılır;
- Ana-Palet geldikten sonra üzerine yardımcı palet yerleştirilir;
- Ana-Palet Stok hücrelerine geri gönderilir;
- Stok hücrelerinde yardımcı palet 60412 seri numaralı olarak uygun rafta saklanır;
- SON (END)

2.1.2.6. Alt-süreç 403 (Metal Pin Tornalama)

Kullanılan Birimler:

- Taşıma Sistemi
- Stok Hücresi
- CNC-Çifti Hücresi

- Robot Melfa RV-2AJ
- Turn 55 CNC Torna Tezgahı
- Palet-8PIN

Açıklama:

Bu alt-süreçte elektronik rulet cihazının alt ve üst kapaklarının birleştirilmesi için kullanılan metal pinler CNC torna tezgahında işlenir. Süreç başlatıldığında, 8 adet işlenmemiş metal pin taşıyan yardımcı palet (Palet 8PIN) Stok hücrelerinden alınarak işlenmek üzere CNC-Çifti hücrelerine getirilir. Robot yardımcı paleti Ana-Palet üzerinden (konum 15) alarak imalat hücreesindeki konum 3'e yerleştirir. Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır.

Konum 3'de bulunan yardımcı paletin sağ tarafında işlenmemiş pinleri sıralı olarak tutan yuvalar bulunmakta olup, en üstteki yuvanın konum numarası 31'dir. Diğer yuvalara ait konum numaraları sırasıyla 32, 33...38 olacak şekilde belirlenmiştir. Paletin sol tarafında bulunan ve işlenmiş pinleri sıralı olarak tutan yuvalardan en üsttekinin konum numarası 41 olup, diğer işlenmiş pin yuvalarının konum numaraları sırasıyla 42, 43...48 olacak şekilde belirlenmiştir.

CNC torna tezgahının güvenlik kapısı açılır. Robotun torna tezgahının içerisine doğru olan hareketi başlamadan önce, güvenlik kapısının açık olup olmadığı tekrar kontrol edilir. İşlenmemiş ilk pin robot tarafından konum 31'den alınarak torna tezgahı içerisindeki mengeneye yerleştirilir. Robot kolunun torna tezgahından ayrılmasıyla birlikte kapı kapatılır ve seçilmiş olan NC programına göre pin işlenmeye başlar.

İlk pinin tornalama işlemi tamamlandığında kapı tekrar açılır. Robotun torna içerisine doğru olan hareketi başlamadan önce kapının açık olup olmadığı yeniden kontrol edilir. İşlenmiş olan pin, bu sefer işlenmiş pinlerin bulunduğu sol taraftaki sıra 31'e taşınır. Sonrasında aynı işlemler diğer 7 adet pin için de tekrarlanır.

8 pinin tornalama işlemi tamamlandığında istasyona boş bir Ana-Palet çağrılır. Boş Ana-Palet istasyona ulaştığında kenetlenir ve üzerine işlenmiş pinleri taşıyan yardımcı palet yüklenir. Son olarak yardımcı palet ileride kullanılmak üzere Stok hücrelerine gönderilir.

İşlem Basamakları:

- BAŞLAT (START)
- Stok hücrelerine boş bir Ana-Palet çağrılır;

- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60430 seri numaralı 8 adet işlenmemiş pin taşıyan yardımcı palet, Ana-Palete yüklenir;
- Yardımcı palet CNC-Çifti hücresine gönderilir;
- Yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak CNC-Çifti hücresindeki konum 3'e taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- CNC torna tezgahının güvenlik kapısı açılır ve kapının açık olduğu kontrol edilir;
- İşlenmemiş pin konum 31'den alınarak torna tezgahı içerisindeki mengeneye yerleştirilir ve mengene sıkıştırılır;
- Güvenlik kapısı kapatılır, ilgili NC programı seçilir ve başlatılır;
- Tornalama işlemi bitiminde güvenlik kapısı açılır ve işlenmiş iş-parçası konum 31'e geri konur.
- Diğer 7 adet pin de aynı süreç içerisinde işlenir;
- Tüm pinler işlendikten sonra istasyona boş bir Ana-Palet çağrılır;
- Ana-palet geldikten sonra üzerine yardımcı palet yerleştirilir;
- Ana-palet Stok hücresine geri gönderilir;
- Stok hücresinde yardımcı palet 60431 seri numaralı olarak uygun rafta saklanır;
- SON (END)

2.1.2.7. Alt-süreç 404 (Baskı Devre Lehimleme)

Kullanılan Birimler:

- Taşıma Sistemi
- Stok Hücresi
- Lehimleme Hücresi
- Robot Melfa RV-E2
- Palet-2PCB

Açıklama:

Bu alt-süreçte elektronik baskı devre üzerine pil konektörü robot tarafından lehimlenir. Süreç başlatıldığında, 2 adet lehimlenmemiş baskı devre taşıyan yardımcı palet (Palet-2PCB) Stok hücresinden alınarak Lehimleme hücresine getirilir. Robot yardımcı paleti Ana-Palet üzerinden (konum 15) alarak imalat hücresindeki konum 1'e yerleştirir. Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır.

Konum 1'de bulunan yardımcı palet üzerindeki lehimlenmemiş baskı devrelerden ilkinin konum numarası 11, diğerinin ki ise 12 olarak belirlenmiştir.

İlk baskı devre konum 11'den alınarak lehimleme tezgahına taşınır. Magazinden alınan pil bağlantısı baskı devre üzerine yerleştirilir. Sonrasında robot pnömatik tutucu elini (gripper) bırakır ve lehimleme elini alır. Pil konektörü, baskı devre üzerine sağ ve sol taraflarından lehimlenir. Lehimleme işlemi bittikten sonra robot lehimleme elini bırakarak tekrar tutucu elini alır. Lehimleme işlemi bitirilen kart tezgahından alınarak tekrar konum 11'e geri yerleştirilir. Diğer baskı devrenin de lehimleme işlemi benzer şekilde yapılır.

Her iki baskı devrenin lehimleme işlemi tamamlandığında istasyona boş bir Ana-Palet çağrılır. Boş Ana-Palet istasyona ulaştığında kenetlenir ve üzerine lehimlenmiş baskı devreleri taşıyan yardımcı palet yüklenir. Bu noktadan sonra sürece iki farklı şekilde devam edilebilir:

- Lehimleme işlemi tamamlanan baskı devreler direk Stok hücreğine gönderilerek 60471 seri numarasıyla saklanır.
- Lehimleme işlemi tamamlanan baskı devreler Alt-süreç 405 çalıştırılarak, test edilmek üzere, Görüntü İşleme hücreğine gönderilir. Testleri tamamlanan iş-parçaları Stok hücreğine gönderilerek 60472 seri numarasıyla saklanır.

İşlem Basamakları:

- BAŞLAT (START)
- Stok hücreğine boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60470 seri numaralı 2 adet lehimlenmemiş baskı devre taşıyan yardımcı palet, Ana-Palete yüklenir;
- Yardımcı palet Lehimleme hücreğine gönderilir;
- Yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Lehimleme hücreesindeki konum 1'e taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- Lehimlenmemiş ilk baskı devre konum 11'den alınarak montaj alanına taşınır;
- Magazinde pil konektörü bulunup bulunmadığı kontrol edilir;
- Robot magazinden pil konektörünü alır ve baskı devre üzerine yerleştirir;
- Robot pnömatik tutucu elini bırakır ve lehimleme elini alır;
- Pil konektörü baskı devre üzerine sağ ve sol taraftan lehimlenir;
- Robot lehimleme elini bırakır ve pnömatik tutucu elini yeniden alır;
- Lehimleme işlemi bitmiş baskı devre konum 11'e geri konulur;
- Diğer baskı devre konum 12'den alınarak benzer şekilde lehimlenir;
- Lehimleme işlemi bittikten sonra istasyona boş bir Ana-Palet çağrılır;

- Ana-Palet geldikten sonra üzerine yardımcı palet yerleştirilir;
- Kullanıcıya baskı devrenin testinin yapılıp yapılmayacağı sorulur;
- Test yapılmayacaksa; Ana-Palet Stok hücreğine geri gönderilir ve yardımcı palet 60471 seri numaralı olarak uygun rafta saklanır;
- Test yapılacaksa; alt-süreç 405 çalıştırılır, Ana-Palet Görüntü İşleme hücreğine gönderilir, testlerin bitiminde Ana-Palet Stok hücreğine geri gönderilir ve yardımcı palet 60472 seri numaralı olarak uygun rafta saklanır;
- SON (END)

2.1.2.8. Alt-süreç 405 (Baskı Devre Test)

Kullanılan Birimler:

- Taşıma Sistemi
- Stok Hücresi
- Görüntü İşleme Hücresi
- Görüntü İşleme Sistemi (Visi-Control)
- Elektronik Test Ünitesi
- Robot Melfa RV-E2
- Palet-2PCB

Açıklama:

Bu alt-süreçte pil konektörü lehimlenmiş elektronik baskı devrelerin eleman yerleşimi kontrolü ve fonksiyon testleri yürütülür. Süreç başlatıldığında, 2 adet lehimlenmiş baskı devre taşıyan yardımcı palet (Palet 2PCB), kullanıcının seçimine bağlı olarak direk alt-süreç 404'ten veya Stok hücrelerinden alınarak Görüntü İşleme hücreğine getirilir. Robot yardımcı paleti Ana-Palet üzerinden (konum 15) alarak imalat hücreesindeki konum 1'e yerleştirir. Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır.

Konum 1'de bulunan yardımcı palet üzerindeki lehimlenmiş baskı devrelerden ilkinin konum numarası 11, diğerinin ki ise 12 olarak belirlenmiştir.

Robot ilk baskı devreyi konum 11'den alarak elektronik test ünitesine taşınır. Test ünitesinde baskı devrenin alt tarafındaki bağlantılara metal pinler değiştirilerek elektriksel temas sağlanır. Fonksiyon testleri gerçekleştirilir. Sonrasında görüntü işleme sistemine ait kamera baskı devre üzerine taşınır ve baskı devre üzerindeki eleman yerleşimleri kontrol edilir. Diğer baskı devrenin de test işlemleri benzer şekilde yürütülür.

Her iki baskı devrenin test işlemleri tamamlandığında istasyona boş bir Ana-Palet çağrılır. Boş Ana-Palet istasyona ulaştığında kenetlenir. Üzerine lehimlenmiş ve testleri tamamlanmış baskı devreleri taşıyan yardımcı palet yüklenir. Son olarak yardımcı palet ileride kullanılmak üzere Stok hücreğine gönderilir.

İşlem Basamakları:

- BAŞLAT (START)
- Eğer alt-süreç 404'ten direk olarak yönlendirilmemişse, Stok hücrelerinden boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60470 seri numaralı 2 adet lehimlenmemiş baskı devre taşıyan yardımcı palet, Ana-Palete yüklenir;
- Eğer alt-süreç 404'ten yönlendirilmişse, süreç bu adımdan itibaren yürütülür;
- Yardımcı palet Görüntü İşleme hücreğine gönderilir;
- Yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Görüntü İşleme hücreesindeki konum 1'e taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- Lehimlenmiş ilk baskı devre konum 11'den alınarak elektronik test ünitesine taşınır;
- Baskı devrenin alt tarafındaki bağlantılara metal pinler değiştirilerek elektriksel temas sağlanır ve fonksiyon testleri gerçekleştirilir;
- Fonksiyon testi bitiminde görüntü işleme sistemine ait kamera baskı devre üzerine taşınır ve eleman yerleşimi kontrol edilir;
- Kontrol işlemi bitmiş baskı devre konum 11'e geri konulur;
- Diğer baskı devre de konum 12'den alınarak benzer şekilde test ve kontrol edilir;
- Test ve kontrol işlemleri bittikten sonra istasyona boş bir Ana-Palet çağrılır;
- Ana-palet geldikten sonra üzerine yardımcı palet yerleştirilir;
- Ana-palet Stok hücreğine geri gönderilir;
- Stok hücreğinde yardımcı palet 60472 seri numaralı olarak uygun rafta saklanır;
- SON (END)

2.1.2.9. Alt-süreç 406 (Elle Montaj)

Kullanılan Birimler:

- Taşıma Sistemi
- Stok Hücresi
- Montaj Hücresi
- Robot Melfa RV-E2
- Üst Kapak Magazini

- Alt Kapak Magazini
- Pil Magazini
- Palet 2PCB
- Palet 2ED
- Palet 8PIN

Açıklama:

Bu alt-süreçte örnek ürün olan elektronik ruletin montaj işlemleri elle yürütülür. Süreç başlatılmadan önce, işlenmiş alt/üst kapak ve pil magazinlerinin dolu olduğu kontrol edilir. Sonrasında konum 6'ya 60431 seri numaralı 8 adet işlenmiş pin taşıyan Palet 8PIN ile konum 7'ye 60472 seri numaralı 2 adet lehimlenmiş baskı devre taşıyan Palet 2PCB elle yerleştirilir.

Süreç başlatıldığında, 62400 seri numaralı boş bir yardımcı palet (Palet-2ED) Stok hücrelerinden alınarak Montaj hücrelerine getirilir. Robot yardımcı paleti Ana-Palet üzerinden (konum 15) alarak imalat hücreesindeki konum 1'e yerleştirir. Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır.

Robotlu montaj işlemi, işlenmiş alt kapağın magazinden alınarak montaj tezgahına yerleştirilmesiyle başlar. Lehimlenmiş ve test edilmiş baskı devre konum 71'den alınarak alt kapağın içerisine yerleştirilir. Sonrasında magazinden alınan pil montaj alanındaki baskı devre üzerine belirli bir mesafeden bırakılır. Pilin yuvasına yerleşmesi ve pil konektörüne teması için robot tarafından üzerine hafifçe bastırılır. Sonrasında magazinden işlenmiş bir üst kapak alınarak montaj tezgahındaki devrenin üzerine yerleştirilir. Son olarak konum 61 ile 64 arasından alınan işlenmiş 4 adet pin, üst kapak üzerindeki deliklere sırasıyla yerleştirilir.

Montajı biten ilk ürün (rulet) boş paletin üzerindeki konum 11'e yerleştirilir. İkinci ürünün montajı da ilkinde anlatıldığı şekilde tamamlandıktan sonra, ikinci ürün konum 12'ye yerleştirilir.

Her iki ürünün montaj işlemleri tamamlandığında istasyona boş bir Ana-Palet çağrılır. Boş Ana-Palet istasyona ulaştığında kenetlenerek, üzerine montajı tamamlanmış ürünleri taşıyan yardımcı palet yüklenir. Son olarak yardımcı palet ileride kullanılmak üzere Stok hücrelerine gönderilir.

İşlem Basamakları:

- BAŞLAT (START)

- İşlenmiş alt/üst kapak ve pil magazinlerinin dolu olup olmadığı kontrol edilir;
- Kullanıcı tarafından konum 6'ya 60431 seri numaralı 8 adet işlenmiş pin taşıyan Palet 8PIN ile konum 7'ye 60472 seri numaralı 2 adet lehimlenmiş baskı devre taşıyan Palet 2PCB elle yerleştirilip yerleştirilmediği kontrol edilir.
- Stok hücre sine boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 62400 seri numaralı boş bir yardımcı palet (Palet-2ED), Ana-Paleta yüklenir;
- Yardımcı palet Montaj hücre sine gönderilir;
- Yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Montaj hücre sine konum 1'e taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- İşlenmiş alt kapak magazinden alınarak montaj tezgahına yerleştirilir;
- Lehimlenmiş ve test edilmiş ilk baskı devre konum 71'den alınarak alt kapak içerisine yerleştirilir;
- Pil magazinden alınarak baskı devrenin üzerine belirli bir mesafeden bırakılır;
- Pilin baskı devredeki yuvaya oturması ve pil konektörü ile temas edebilmesi için robot tarafından üzerine hafifçe bastırılır;
- İşlenmiş üst kapak magazinden alınarak montaj alanındaki baskı devre üzerine yerleştirilir;
- İşlenmiş pinler konum 61-64 arasından sırasıyla alınarak üst kapak üzerindeki deliklere yerleştirilir;
- Montajı biten ilk ürün konum 11'e taşınır;
- Aynı montaj süreci diğer ürün için de yürütülür ve montajı bitmiş ürün konum 12'ye taşınır;
- Montaj işlemleri bittikten sonra istasyona boş bir Ana-Palet çağrılır;
- Ana-palet geldikten sonra üzerine yardımcı palet yerleştirilir;
- Ana-palet Stok hücre sine geri gönderilir;
- Stok hücre sine yardımcı palet 60501seri numaralı olarak uygun rafta saklanır;
- SON (END)

2.1.2.10. Alt-süreç 407 (Otomatik Montaj)

Kullanılan Birimler:

- Taşıma Sistemi
- Stok Hücre si
- Montaj Hücre si
- Robot Melfa RV-E2
- Pil Magazini

- Palet 2PCB
- Palet 2ED
- Palet 8PIN

Açıklama:

Bu alt-süreçte örnek ürün olan elektronik ruletin montaj işlemleri tam otomatik olarak yürütülür. Ürünün montajı için gerekli pil dışındaki tüm parçalar Stok hücresinden getirilir.

Süreç başlatıldığında, birer adet 60431 seri numaralı 8 adet işlenmiş pin taşıyan Palet-8PIN, 60412 seri numaralı 2 adet işlenmiş alt kapak taşıyan Palet 2ED, 60413 seri numaralı 2 adet işlenmiş üst kapak taşıyan Palet-2ED, 60472 seri numaralı lehimlenmiş ve test edilmiş baskı devre taşıyan Palet 2PCB ile 62400 seri numaralı boş Palet 2ED Montaj hücresine gönderilir. Paletler istasyona vardıklarında sırasıyla; 60431 seri numaralı palet konum 4'e, 60412 seri numaralı palet konum 2'ye, 60413 seri numaralı palet konum 3'e ve 60472 seri numaralı palet konum 5'e robot tarafından yerleştirilir. Sonrasında robotik montaj işlerine başlanır.

Bu sırada 62400 seri numaralı boş palet istasyona ulaştığında, program akışında bir kesme oluşur. Robot montaj işlemini yarıda bırakarak 62400 seri numaralı boş paleti konum 1'e yerleştirir. Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır. Sonrasında robot montaj işlemine kaldığı yerden devam eder.

Robotlu montaj işlemi, işlenmiş alt kapağın konum 21'den alınarak montaj alanına yerleştirilmesiyle başlar. Lehimlenmiş ve test edilmiş baskı devre konum 51'den alınarak alt kapağın içerisine yerleştirilir. Sonrasında magazinden alınan pil montaj alanındaki baskı devre üzerine belirli bir mesafeden bırakılır. Pil konektörüne temasın sağlanması için robot tarafından pilin üzerine hafifçe bastırılır. Sonrasında konum 31'den işlenmiş bir üst kapak alınarak montaj alanındaki devrenin üzerine yerleştirilir. Son olarak konum 41 ile 44 arasından alınan işlenmiş 4 adet pin, üst kapak üzerindeki deliklere sırasıyla yerleştirilir.

Montajı biten ilk ürün (rulet) boş paletin üzerindeki konum 11'e yerleştirilir. İkinci ürünün montajı da ilkinde anlatıldığı şekilde tamamlandıktan sonra, ürün konum 12'ye yerleştirilir.

Her iki ürünün montaj işlemleri tamamlandığında istasyona boş bir Ana-Palet çağrılır. Boş Ana-Palet istasyona ulaştığında kenetlenerek üzerine montajı tamamlanmış ürünleri taşıyan yardımcı palet yüklenir. Son olarak yardımcı palet ileride kullanılmak üzere Stok hücreesine gönderilir.

İşlem Basamakları:

- BAŞLAT (START)
- Stok hücreesine boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60431 seri numaralı 8 adet işlenmiş pin taşıyan yardımcı palet (Palet-8PIN), Ana-Palete yüklenir;
- Yardımcı palet Montaj hücreesine gönderilir;
- Stok hücreesine boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60412 seri numaralı 2 adet işlenmiş alt kapak taşıyan yardımcı palet (Palet-2ED), Ana-Palete yüklenir;
- Yardımcı palet Montaj hücreesine gönderilir;
- Stok hücreesine boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60413 seri numaralı 2 adet işlenmiş üst kapak taşıyan yardımcı palet (Palet-2ED), Ana-Palete yüklenir;
- Yardımcı palet Montaj hücreesine gönderilir;
- Stok hücreesine boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60472 seri numaralı 2 adet lehimlenmiş baskı devre taşıyan yardımcı palet (Palet-2PCB), Ana-Palete yüklenir;
- Yardımcı-palet Montaj hücreesine gönderilir;
- Stok hücreesine boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 62400 seri numaralı boş bir yardımcı palet (Palet-2ED), Ana-Palete yüklenir;
- Yardımcı-palet Montaj hücreesine gönderilir;
- 60431 seri numaralı yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Montaj hücreesindeki konum 4'e taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- 60412 seri numaralı yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Montaj hücreesindeki konum 2'ye taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- 60413 seri numaralı yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Montaj hücreesindeki konum 3'e taşınır;

- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- 60472 seri numaralı yardımcı palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Montaj hücresindeki konum 5'e taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- Ürüne ait tüm iş-parçaları imalat hücresine alındıktan sonra, 62400 seri numaralı boş paletin gelmesi beklenmeden montaj işlemlerine başlanır;
- 62400 seri numaralı yardımcı palet istasyona ulaştığında program akışında kesme oluşur. Palet taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Montaj hücresindeki konum 1'e taşınır. Montaj programına kaldığı yerden devam edilir;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;
- İşlenmiş ilk alt kapak konum 21'den alınarak montaj tezgahına yerleştirilir;
- Lehimlenmiş ve test edilmiş ilk baskı devre konum 51'den alınarak alt kapak içerisine yerleştirilir;
- Pil magazinden alınarak baskı devrenin üzerine belirli bir mesafeden bırakılır;
- Pilin baskı devredeki yuvaya oturması ve pil konektörü ile temas edebilmesi için robot tarafından üzerine hafifçe bastırılır;
- İşlenmiş ilk üst kapak konum 31'den alınarak montaj tezgahındaki baskı devre üzerine yerleştirilir;
- İşlenmiş pinler konum 41-44 arasından sırasıyla alınarak üst kapak üzerindeki deliklere yerleştirilir;
- Montajı biten ilk ürün konum 11'e taşınır;
- Aynı montaj süreci diğer ürün için de yürütülür, montajı biten ürün konum 12'ye taşınır;
- Montaj işlemleri bittikten sonra istasyona boş bir Ana-Palet çağrılır;
- Ana-palet geldikten sonra üzerine yardımcı palet yerleştirilir;
- Ana-palet Stok hücresine geri gönderilir;
- Stok hücresinde yardımcı palet 60501 seri numaralı olarak uygun rafta saklanır;
- SON (END)

2.1.2.11. Alt-süreç 408 (Hidrolik baskılama)

Kullanılan Birimler:

- Taşıma Sistemi
- Stok Hücresi
- Hidrolik-Baskı Hücresi
- Servo Pnömatik Sistem
- Palet-2ED

Açıklama:

Bu alt-süreçte ön-montaj işlemleri tamamlanmış durumdaki örnek ürünün (rulet), üst kapağı üzerindeki metal pinlerin, hidrolik baskılama işlemleri yürütülür.

Süreç başlatıldığında, 60401 seri numaralı 2 adet ön-montajı tamamlanmış örnek ürünü taşıyan Palet 2ED, kullanıcının seçimine bağlı olarak direk Alt-süreç 407'den veya Stok hücresinden alınarak Hidrolik-Baskı hücresine getirilir. Hücredeki Servo Pnömatik Sistem, yardımcı paleti Ana-Palet üzerinden (konum 15) alarak imalat hücresindeki konum 1'e yerleştirir. Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır.

Ön-montajı yapılmış ürünlerden ilki Servo Pnömatik Sistem tarafından yardımcı palet üzerindeki konum 11'den alınarak kızığa yerleştirilir. Hidrolik-baskı ünitesinin güvenlik kilidi ve kapısı açılır. Kızak içeri girerek ürünü hidrolik-baskının altına taşır. Güvenlik kapısı ve kilidi kapanıp kilitlendikten sonra hidrolik-baskı aktif olur. Ürünün üst kapağı üzerindeki pinler bastırılarak alt ve üst kapak birleştirir. Böylelikle ürünün montaj işlemi tamamlanır. Sonrasında güvenlik kapağı açılarak kızak dışarı çıkar. Montajı tamamlanmış olan ürün Konum 11'e geri yüklenir. Diğer ürün de konum 12'den alınarak aynı baskılama işlemlerinden geçirilir

Her iki ürünün hidrolik baskılama işlemleri tamamlandığında istasyona boş bir Ana-Palet çağrılır. Boş Ana-Palet istasyona ulaştığında kenetlenerek üzerine montajı tamamlanmış ürünleri taşıyan yardımcı palet yüklenir. Son olarak yardımcı palet ileride kullanılmak üzere Stok hücresine gönderilir.

İşlem Basamakları:

- BAŞLAT (START)
- Eğer alt-süreç 407'den direk olarak yönlendirilmemişse, Stok hücresinden boş bir Ana-Palet çağrılır;
- Hücreye gelen boş Ana-Palet kenetlendikten sonra 60401 seri numaralı 2 adet ön-montajı tamamlanmış ürün taşıyan yardımcı palet, Ana-Palete yüklenir;
- Eğer alt-süreç 407'den yönlendirilmişse, süreç bu adımdan itibaren yürütülür;
- Yardımcı palet Hidrolik-Baskı hücresine gönderilir;
- Yardımcı palet, taşıma sistemindeki Ana-Palet üzerinden (konum 15) alınarak Hidrolik-Baskı hücresindeki konum 1'e taşınır;
- Yükünü boşaltan Ana-Palet diğer işlemlerde kullanılmak üzere serbest bırakılır;

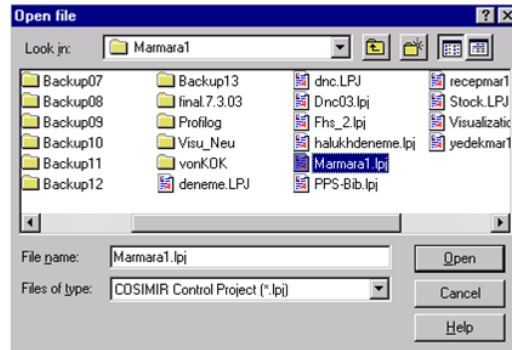
- Ön-montajı tamamlanmış ilk ürün Servo Pnömatik Sistem tarafından konum 11’den alınarak kızağa taşınır;
- Hidrolik-baskı ünitesinin güvenlik kilidi ve kapısı açılır;
- Kızak içeri doğru hareket etmeden önce kapının açık olup olmadığı kontrol edilir;
- Kızak ürünü hidrolik-baskı ünitesinin içerisine taşır;
- Güvenlik kapısı ve kilidi kapatılır;
- Ürünün kapağı üzerindeki pinler bastırılarak alt ve üst kapağın birleşmesi sağlanır;
- Güvenlik kilidi ve kapısı açılır;
- Kızak montajı bitmiş ürünü hidrolik-baskı ünitesinin dışına taşır;
- Servo Pnömatik Sistem montajı bitmiş ürünü konum 11’e geri taşır;
- Diğer ürün de konum 12’den alınarak benzer şekilde baskılanır;
- Montaj işlemleri bittikten sonra istasyona boş bir Ana-Palet çağrılır;
- İstasyona Ana-Palet geldikten sonra üzerine yardımcı palet yerleştirilir;
- Ana-Palet Stok hücresine geri gönderilir;
- Stok hücresinde yardımcı palet 60501 seri numaralı olarak uygun rafta saklanır;
- SON (END)

2.1.2.12. SCADA Yazılımı (COSIMIR Factory Control)

FMS100 sisteminin imalat yönetimi ve izleme işlemleri COSIMIR Factory Control yazılımı üzerinden gerçekleştirilir.

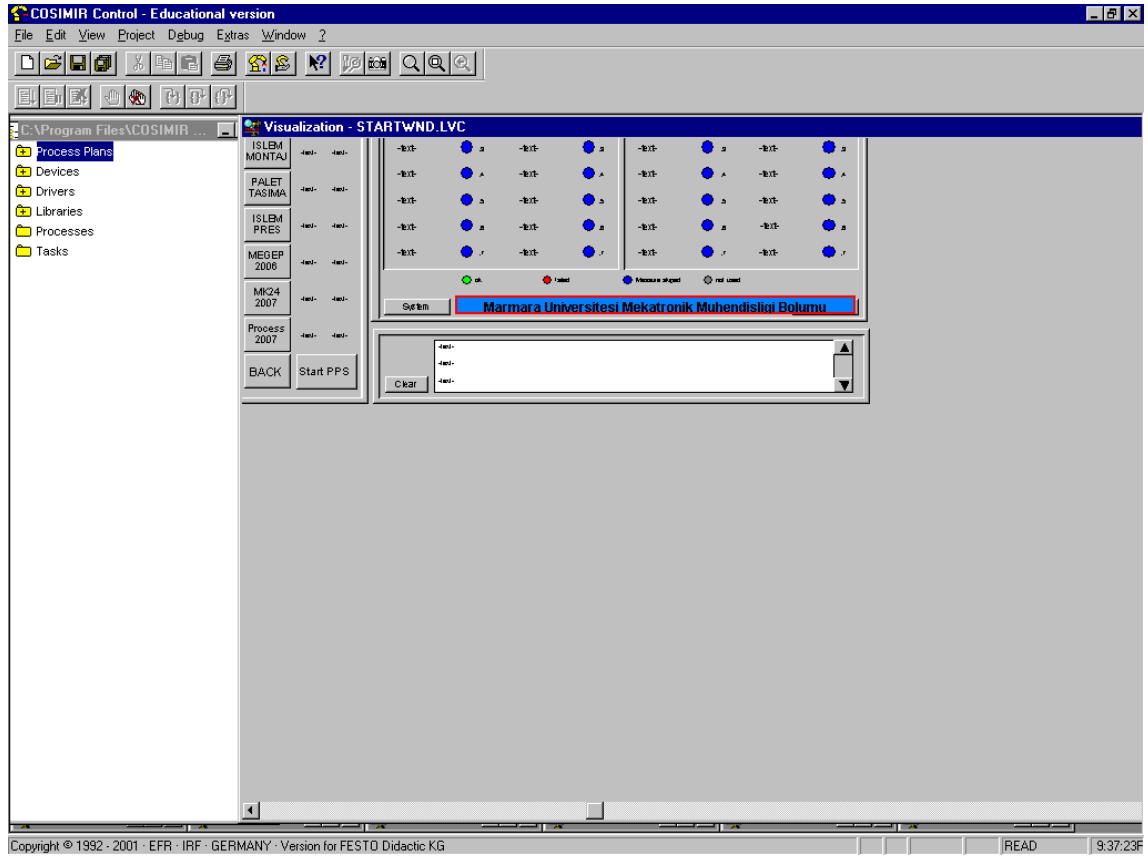
2.1.2.12.1. Yazılımın Çalıştırılması

SCADA bilgisayarı üzerinde çalışan yazılım, masaüstündeki Şekil 2-27 (a)’da görülen COSIMIR Factory Control simgesi çift tıklanarak başlatılır. Yazılım için önceden hazırlanmış olan “Marmara.lpj” projesi Şekil 2-27 (b)’deki gibi açılır.



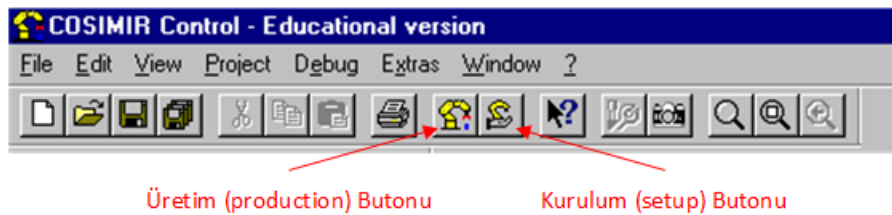
Şekil 2-27. COSIMIR Factory Control yazılımının masaüstü kısayol simgesi (a), “Marmara1.lpj” dosyası seçme penceresi (b).

Ekrana Şekil 2-28'deki gibi arayüz görüntüsü gelir.



Şekil 2-28. COSIMIR Factory Control yazılımı ekran görüntüsü.

Yazılımın temel olarak iki farklı çalışma modu vardır: kurulum (setup) ve üretim (production). Şekil 2-29'de görülmekte olan butonlar tıklanarak yazılım çalışma tipleri arasında geçiş yapılabilir.



Şekil 2-29. COSIMIR Factory Control yazılımı üretim ve kurulum çalışma tipi seçim butonları.

2.1.2.12.2. Kurulum Moduna Geçilmesi

COSIMIR Factory Control yazılımında, herhangi bir proje açıldığında ekrana gelen Şekil 2-30'deki proje penceresinde, FMS100 sistemi bünyesinde kullanılan üretim süreci

planları (process plans), cihazlar (devices), sürücüler (drivers), kütüphaneler (libraries), süreçler (processes) ve görevler (tasks) görüntülenir.

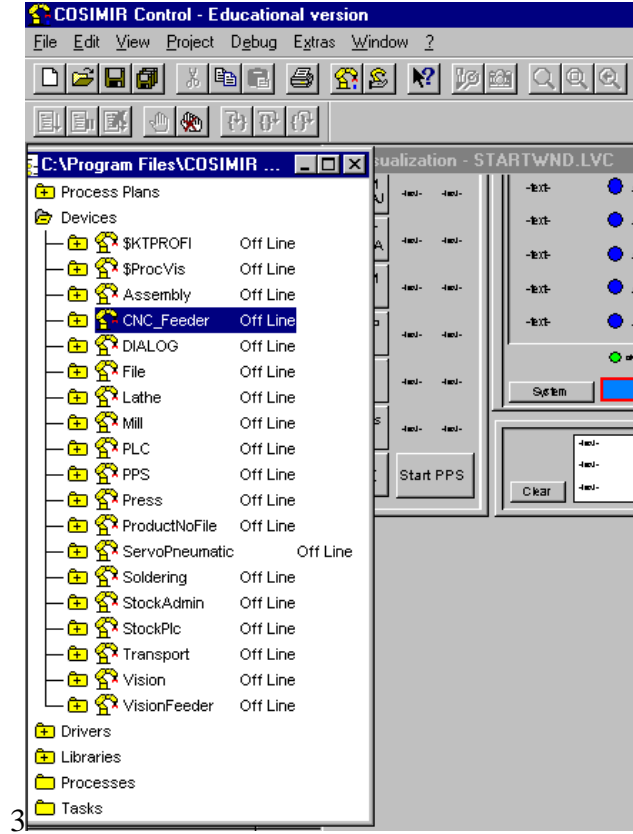


Şekil 2-30. COSIMIR Factory Control yazılımı proje menüsü görüntüsü.

Kurulum butonuna basıldığında proje menüsünde görülen öğeler üzerinde değişiklik yapılabilir. Bunun için değişiklik yapılmak istenen öğenin sol tarafındaki “+” işaretine tıklanır. Ekrana gelen alt menüden, üzerinde değişiklik yapılmak istenen asıl öğe seçilir. Projede tanımlı herhangi bir cihazın haberleşme ağından çıkarılması veya iletişim protokolünün değiştirilmesi gibi ayarlar bu pencere üzerinden yapılır. Kurulum modundan çıkıldıktan sonra, süreç planları ve sistem ayarları üzerinde herhangi bir değişiklik yapılmasına izin verilmez.

FMS100 ile üretime geçmeden önce yapılacak ilk işlem, yazılımın her bir cihaz ile hangi protokol üzerinden haberleşeceği gibi ayarların kontrol edilmesidir. Bu bölümde yapılacak yanlış bir tanımlama, SCADA yazılımı ile cihaz arasında iletişim hatalarına sebep olabilir.

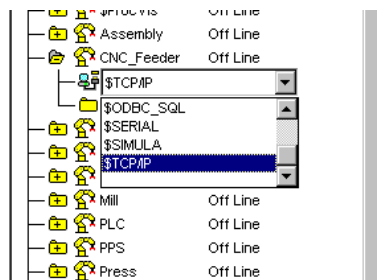
Bunun için Şekil 2-30’deki proje penceresinden “Devices” öğesine tıklanır. Böylelikle sistemde tanımlı bulunan tüm cihazlar ve durumları Şekil 2-31’de olduğu gibi, ağaç yapısı biçiminde görüntülenir.



Şekil 2-31. Proje penceresi “Devices” sekmesi içeriği görünümü.

Pencerde görüntülenen cihazların üzerine tıklandığında, cihazın hangi protokol ile haberleştiği görüntülenir. Protokol adının yan tarafındaki boş alan tıklandığında, kullanıcı tarafından seçilebilir durumdaki tüm protokoller Şekil 2-32’deki gibi listelenir. Bu listeden haberleşmede kullanılacak protokol seçilebilir.

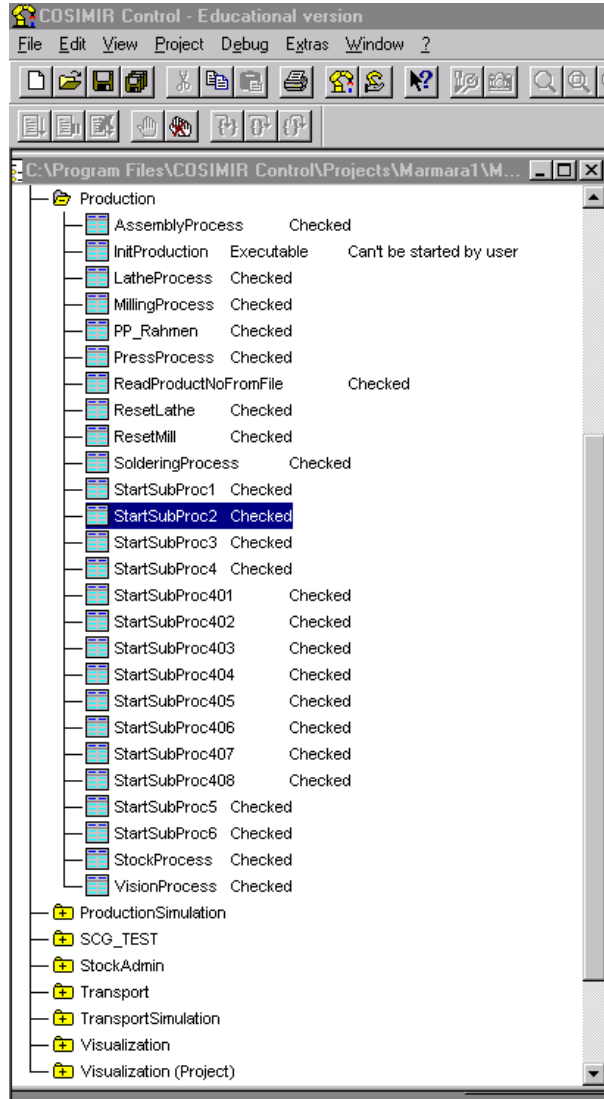
Örneğin “CNC Feeder” cihazının SCADA yazılımı ile haberleşme protokolü değiştirilme işlemi Şekil 2-32’deki pencereden yapılır. “CNC Feeder” ögesinin sağ tarafındaki boş alana tıklandığında, Şekil 2-32’de görülen alt menü açılır. Bu menüden cihazın haberleşme protokolü seçilir. Şekle göre cihaz SCADA yazılımıyla “TCP/IP” protokolü üzerinden haberleşmektedir.



Şekil 2-32. Cihaz iletişim protokolü değişim penceresi.

* Not: Eğer FMS bünyesindeki herhangi bir cihaz veya istasyonda arıza varsa, ilgili cihazın protokolü “\$SIMULA” olarak seçilmelidir. Bu durumda SCADA yazılımı ilgili cihazı ve durumunu dikkate almayarak haberleşme listesine eklemeyecektir.

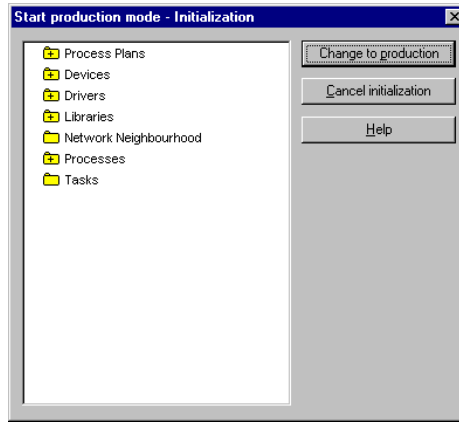
Kurulum modundaki bir diğer önemli husus ise süreç planlarıdır. FMS100 bünyesinde üretimle ilgili tüm süreçler bu planlar içerisinde tanımlanmıştır. Şekil 2-33’de görülmekte olan pencerede yine üretici firma tarafından önceden tanımlanmış süreç planları görülmektedir. “StartSubProc401” ile “StartSubProc408” arası süreç planları, daha önce Tablo 2-2’de açıklanmış olan imalat süreçlerinin tanımlandığı dosyalardır. Diğer süreç planları ise FMS100’ün çalışması için tanımlanmış alt-süreçleri bulundurulur. Yeni süreç planları bu bölümde tanımlanarak aktifleştirilebilir.



Şekil 2-33. COSIMIR Factory Control yazılımında “Marmara1.lpj” proje dosyasında tanımlanmış süreç planları.

2.1.2.12.3. Üretim Moduna Geçilmesi

Üretim moduna geçildiğinde ise öncelikle sisteme bağlı cihazlar ile önceden belirlenen protokoller üzerinden iletişim kurulmaya çalışılır. Her bir istasyonun başlangıç ayarlarının ve durumlarının kontrolü yapılır. Herhangi bir hata algılanırsa üretime geçilmeden önce hata mesajı görüntülenir. Eğer hiçbir hata ile karşılaşılmazsa, yazılım üretim modunda açılarak sistem elemanlarının ve süreç-planlarının canlı olarak görüntülendiği Şekil 2-34'deki pencere açılır.



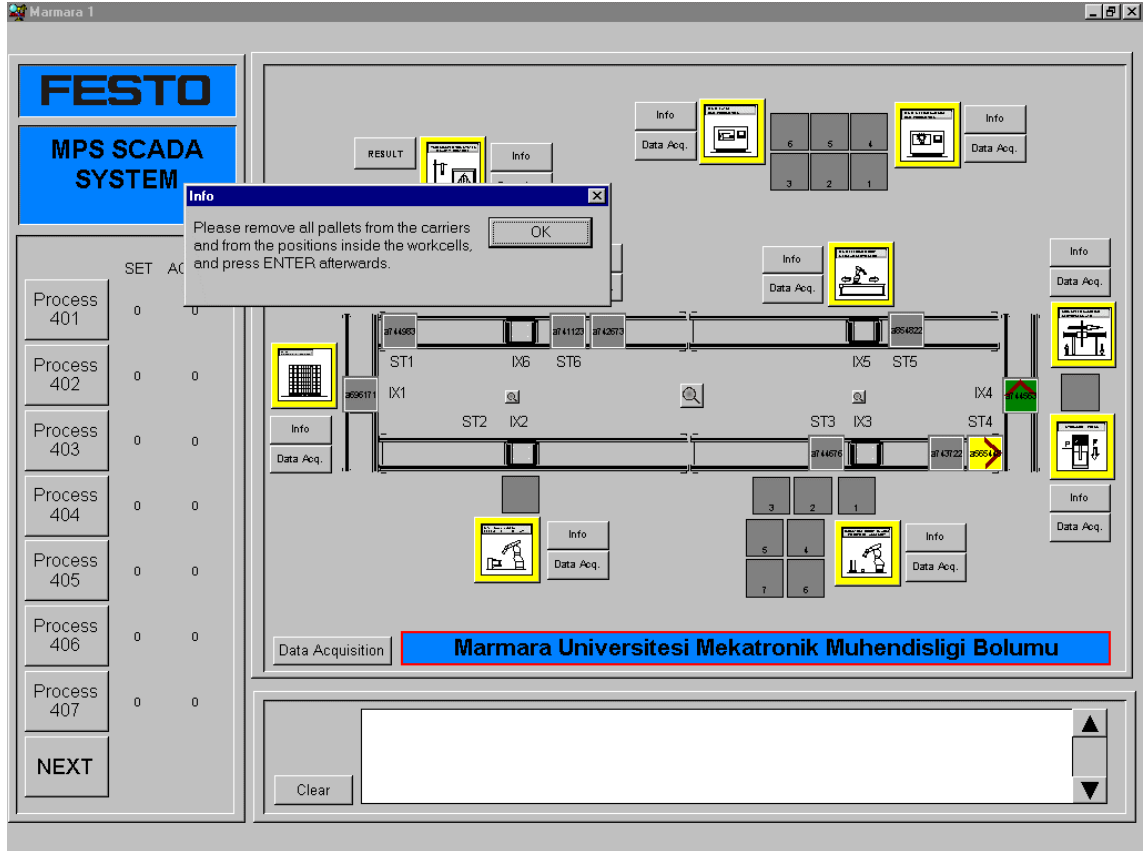
Şekil 2-34. COSIMIR Factory Control yazılımı üretim modunda başlatma ekran görüntüsü.

* Not: Üretim moduna geçmeden önce tüm yardımcı paletlerin sistem üzerinden kaldırılması gereklidir. Bu konuyla ilgili olarak Bölüm 2.1.1.11.1'de anlatılan FMS100'ü devreye almayla ilgili işlem basamakları gözden geçirilmelidir.

2.1.2.12.4. Üretim Sürecinin Başlatılması

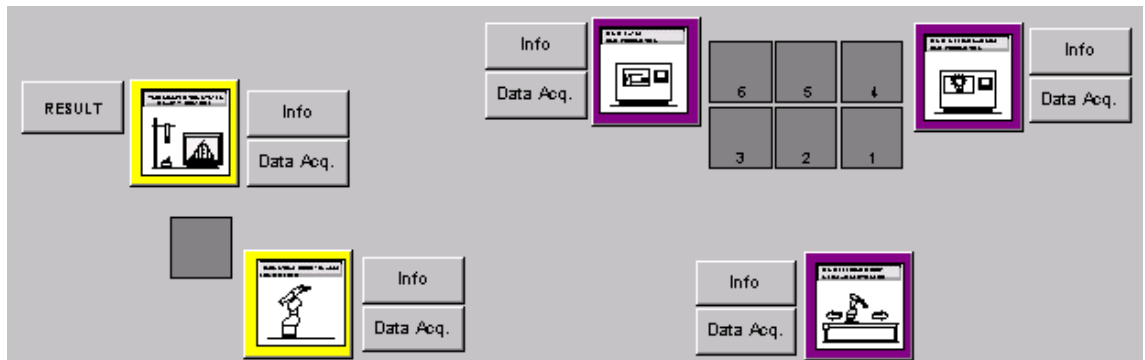
Bu noktada Bölüm 2.1.1.11.1'de anlatılan FMS100'ü başlatma protokollerine uygun olarak sistemin başlatıldığı; tüm üretim hücrelerinin otomatik moda ve çalışmaya hazır olduğu kabul edilmektedir. Şekil 2-35'de FMS100 başarılı bir şekilde üretim modunda başlatıldığında kullanıcının karşısına gelen ekran görüntüsü yer almaktadır.

Ekranda ilk olarak FMS100 bünyesindeki tüm imalat hücrelerinde bulunan palet kabul alanlarının boşaltılması gerektiğine dair bir uyarı mesajı görüntülenmektedir. Ekranın sol tarafında, daha önce Tablo 2-2'de açıklanan ve Şekil 2-33'deki pencerede görüntülenen süreç planlarını başlatan butonlar yer almaktadır. "Process401" ile "Process407" arasındaki süreçler bu pencereden başlatılabilir. Sol alt taraftaki "NEXT" butonuna basılınca, diğer üretim süreçlerini başlatan butonları bulunduran, bir sonraki ekrana geçiş yapılabilir.



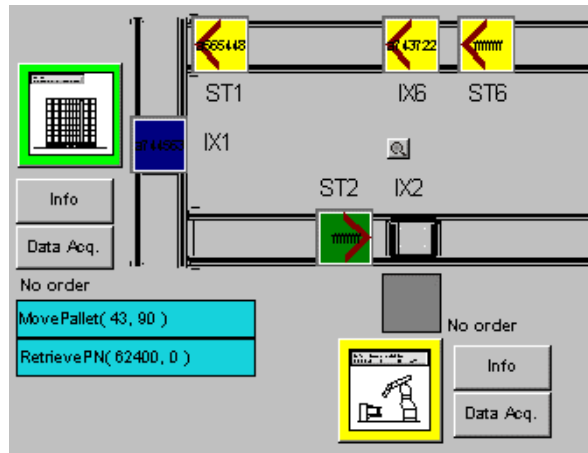
Şekil 2-35. COSIMIR Factory Control yazılımının üretim modunda açıldığında ekran görüntüsü.

Ekranın ortasındaki alanda ise FMS100 bünyesindeki imalat hücreleri ve durumları görüntülenir. Şekil 2-36’de sarı renkli arka plana sahip hücreler ile iletişim kurulmuş veya hücre “Simula” modunda anlamına gelir. Mor renkli arka plana sahip hücreler ile iletişim kurulamadığını gösterir.



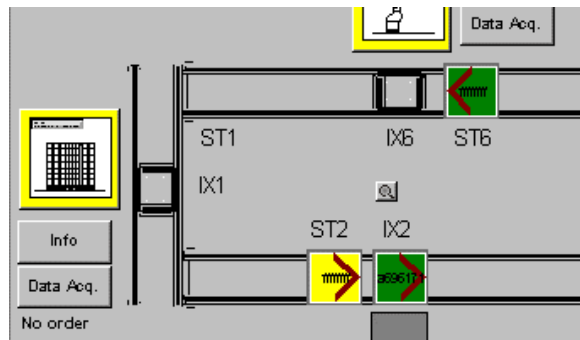
Şekil 2-36. Üretim ekranında iletişim kurulabilmiş (sarı) ve kurulamamış (mor) hücrelerin görüntülenmesi.

Herhangi bir üretim süreci veya görev başlatıldığında, ilgili hücrelerde yürütülen görevler ile ilgili açıklamalar Şekil 2-36'deki gibi görüntülenir. Şekilde Stok hücresinin arka plan renginin açık yeşil olması, hücrenin aktif olduğunu gösterir. Hücre simgesinin alt tarafındaki mavi kutularda ise yürütülmekte olan görevler gösterilir. Şekil 2-36'ya göre Stok hücresinde “Move Pallet(43, 90)” ve “Retreive PN(62400,0)” görevleri yürütülmektedir. Bu görevler hücreden 62400 seri numaralı boş Palet 2ED'nin 43 numaralı raftan alınarak, istasyonda kenetlenmiş durumdaki Ana-Palet üzerine yerleştirilmekte olduğu anlamına gelir.



Şekil 2-37. Üretim ekranında aktif olan hücre ve görevlerin görüntülenmesi.

Taşıma sistemi üzerinde hareket etmekte olan Ana-Paletlerin kod numaraları ve hangi istasyonda oldukları, Şekil 2-38'deki gibi, yine bu ekran üzerinden görüntülenebilir. FMS100 üretim modunda ilk olarak başlatıldığında, SCADA yazılımı paletleri tanımlayana kadar palet numaraları ekranda “ffffffff” olarak görüntülenir. Tanımlama işlemi bittiğinde her palet simgesi üzerine o paletle ait manyetik etiket ile kodlanmış numara görüntülenir.



Şekil 2-38. Üretim ekranı taşıma sisteminde tanımlanmış ve hareket halindeki Ana-Paletlerin görüntüsü.

2.1.2.12.5. Üretim Sürecinin Sonlandırılması

FMS100 üretim modu sonlandırılmak istendiğinde, COSIMIR Factory Control yazılımı ana ekranına dönülerek Şekil 2-29'deki üretim (production) düğmesine basılır. Bu andan itibaren yazılım öncelikle tüm üretim süreç ve alt-süreçlerini durdurmaya başlar. Bu sırada Şekil 2-39'daki pencere görüntülenir. Bu işlemler belirli bir süre alacağı ve ileride sorunlarla karşılaşılması için beklenmelidir.

* Not: Eğer ki herhangi bir üretim süreci veya alt sürecinde bir aksaklık oluşmuş ve yazılım bir türlü üretim modunu sonlandıramıyorsa, o zaman Şekil 2-39'daki "Cancel all processes" butonuna basılır.



Şekil 2-39. Üretim sonlandırma (finish production) ekranı görüntüsü.

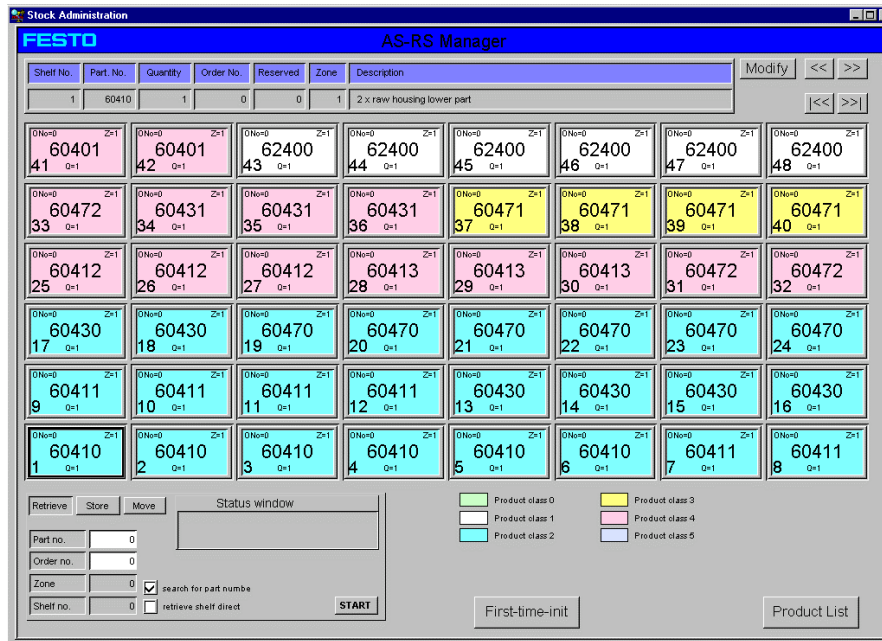
Üretim süreçleri sonlandırıldıktan sonra cihazlar ile iletişim kesilecektir. Böylelikle Şekil 2-28'deki başlangıç ekranı görüntülenecektir. İstendiği takdirde tekrar kurulum moduna geçilerek sistemde değişiklik yapılabilir veya SCADA yazılımı sonlandırılabilir.

2.1.2.13. Stok Yönetim Ekranı

Daha önce Bölüm 2.1.1.2'de anlatıldığı üzere; Stok hücresi temel olarak FMS100 bünyesindeki işlenmiş/işlenmemiş iş-parçalarının saklanması (Store), gerektiğinde üretim istasyonlarına gönderilmesi (Retreive) veya raflar arasında yer değiştirilmesi (Move) için kullanılır. Bu istasyon numaralandırılmış 48 adet raf ve raflardaki iş-parçalarını taşıyan yardımcı paletlere erişim sağlayan 3 eksen Kartezyen konumlandırma sisteminden (Teleskop) oluşmaktadır. Raflardaki iş-parçaları parça numaralarıyla birbirlerinden ayrılmaktadır. Örneğin 60410 numarası 2 adet işlenmemiş alt kapağın bulunduğu yardımcı paleti temsil ederken, 60472 numarası 2 adet lehimlenmiş ve test edilmiş baskı devreyi temsil etmektedir.

FMS100 üretim süreci aktifken, hangi rafta hangi tip iş-parçası sakladığının takip edilmesi oldukça önemlidir. Bu bilgiler SCADA hücresindeki ERP bilgisayarında bulunan veritabanında saklanır. Stok bilgilerinde oluşabilecek bir karışıklık, üretim hatalarının oluşmasına, hatta sistem bileşenlerinin arızalanmasına bile sebep olabilir. Örneğin; 28 numaralı rafta gerçekte 60413 seri numaralı işlenmemiş üst kapak olduğu halde, veri tabanındaki kayıt 60472 seri numaralı lehimlenmiş ve test edilmiş baskı devre şeklinde olsun. FMS100 alt-süreç 407 kapsamında montaj işlemi yürütürken, 60472 seri numaralı 2 adet lehimlenmiş baskı devre talep ettiğinde, Stok hücresi gerçekte 60413 seri numaralı işlenmemiş üst kapak gönderecektir. Sonrasında montaj aşamasında robot işlenmiş alt kapak üzerine baskı devre yerleştirmeye kalkışacak, fakat baskı devre olması gereken alanda üst kapak olacağı için, baskı devreyi tutayım derken büyük bir ihtimalle üst kapağı delegecek veya aşırı zorlanma (overload) alarmı vererek duracaktır. Bu nedenle; SCADA yazılımında, Stok hücresindeki raflarda bulunan yardımcı paletler ile stok bilgilerini düzenleyebilmek için özel bir yönetim ekranı bulunmaktadır.

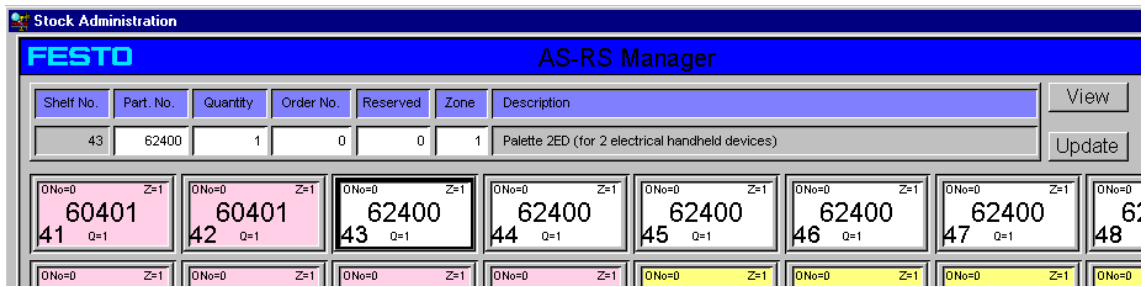
SCADA yazılımı üretim modunda başlatıldığı anda Şekil 2-40'de görülmekte olan Stok Yönetim Ekranı (AS-RS Manager) da otomatik olarak açılmaktadır. Ekran üzerinde 48 adet rafı temsil eden numaralandırılmış kutucuk ve her bir kutucuk içerisinde rafta bulunan iş-parçasına ait seri numarası gibi bilgiler görülmektedir. Hiçbir seri numarası görülmeyen raflar ise boş olarak kabul edilmektedir.



Şekil 2-40. Stok yönetim ekranı görüntüsü (stok administration).

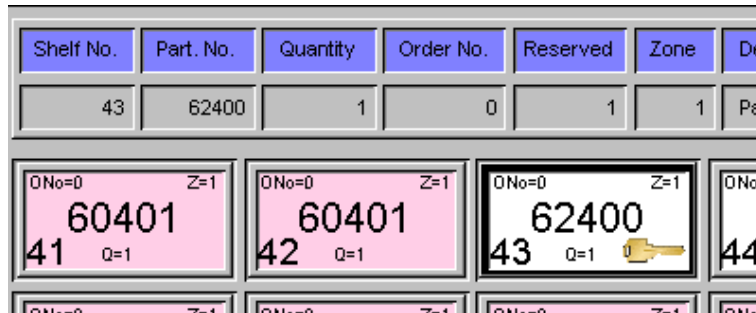
2.1.2.13.1. Raf Seçme ve Düzenleme İşlemi

Raflar, üzerlerine tıklanarak veya klavye ok tuşları kullanılarak seçilebilir. Seçili olan raftaki bilgilerde değişiklik yapılmak istendiğinde Şekil 2-40'deki stok yönetim ekranını sağ üst köşesindeki “Modify” butonuna basılır. Şekil 2-41'de görüldüğü gibi ekranın üst tarafındaki alanda seçili rafa ait bilgiler değiştirilebilir şekilde görüntülenir. Buradaki arka planı beyaz olan kutucuklardan, seçili olan rafta bulunan iş-parçasına ait parça numarası (part. No.), adet (quantity), sipariş numarası (order no.), seçili (reserved) veya bölge (zone) gibi bilgiler değiştirilebilir. Update butonuna basılmasıyla birlikte bu alandaki bilgiler ERP bilgisayarında bulunan veritabanına kaydedilir.



Şekil 2-41. Stok yönetim ekranı hücre bilgileri düzenleme işlemi ekran görüntüsü.

Herhangi bir rafta bulunan iş-parçasını getirmek veya taşımak için öncelikle rafın seçili (reserved) duruma getirilmesi gereklidir. Bunun için ilgili raf (örneğin Şekil 2-41'deki 43 numaralı raf) tıklandıktan sonra “Modify” butonuna basılarak “Reserved” kutucuğuna “1” yazılır ve “Update” butonuna basılır. Seçili durumdaki rafın sağ alt köşesinde Şekil 2-42'deki gibi anahtar resmi görüntülenir. Eğer boş bir raf seçili duruma getirilirse, seçim iptal edilene kadar bu rafa hiçbir iş-parçası saklanamaz.



Şekil 2-42. Stok yönetim ekranında seçili duruma getirilmiş rafa ait ekran görüntüsü.

2.1.2.13.2. Getir İşlevi

Stok hücreesinde herhangi bir raftan, herhangi bir iş-parçasını alarak taşıma sistemindeki boş bir Ana-Palet üzerine yerleştirme işlevidir. Bunun için farklı yöntemler kullanılabilir.

Öncelikle Şekil 2-43'deki “Retrieve” butonuna basılarak getir işlevi aktif edilir. Eğer şekildeki uygun rafları tara (search for part. Number) kutucuğu işaretliyse, parça numarası (part. no.) kutusuna getirilmek istenilen iş-parçasına ait seri numarası girilir. Sipariş numarası (order no.) her zaman 0'dır. “START” butonuna basılmasıyla birlikte sistem, ERP bilgisayarındaki veritabanı kayıtlarından tarama yapar. Seri numarası eşleşen, bulduğu ilk uygun rafı (raf numarası en küçük) seçili (reserved) duruma getirir. Sonrasında Teleskop, seçili raftaki yardımcı paleti alarak istasyonda kenetlenmiş durumdaki boş Ana-Palet üzerine getirir. Boşalan raf boş olarak işaretlenir.

Şekil 2-43. Stok yönetim ekranı üzerindeki getir (retrieve) işlevi bölümü görüntüsü.

Eğer Şekil 2-44'da görülmekte olan direk raf numarasıyla getir “retrieve shelf direct” kutusu işaretliyse, “Shelf no.” kutucuğu aktif hale gelir. Kutucuğa parametre olarak sadece raf numarası girilir. Şekildeki örnekte raf numarası 2 olarak girilmiştir. “START” butonuna basılmasıyla birlikte 2 numaralı rafta ne olduğu dikkate alınmadan, Teleskop 2 numaralı raftaki yardımcı paleti alarak boş Ana-Palet üzerine getirir. Boşalan raf yine boş olarak işaretlenir.

Şekil 2-44. Stok yönetim ekranı üzerindeki getir (retrieve) işlevi “direk raf numarası ile getir” bölümü görüntüsü.

2.1.2.13.3. Sakla İşlevi

Taşıma sistemi ile Stok hücreğine gelmiş iş-parçaları, sakla işlemi ile raflarda depolanır. Hücrenin genel özelliği gereği stokta bulunan iş-parçaları kullanıcı tarafından çeşitli özelliklere göre sınıflandırılabilir. Örneğin tüm metal parçalar bölge 1 (zone 1), plastik parçalar bölge 2 (zone 2) şeklinde ayrılabilir. Daha önce bahsedilen Şekil 2-41'deki düzenleme alanında bölge (zone) kutucuğuna bölge numarası verilebilir.

Sakla işlemi temel olarak boş raf arama (search for empty shelf) modu ve direk rafta saklama (store shelf direct) modu olmak üzere iki farklı şekilde yapılabilir. Boş raf arama modu da kendi içerisinde bölgeleme aktif veya pasif olmak üzere iki farklı şekilde yapılabilir. Eğer Şekil 2-45'deki "search for empty shelf" kutusu ile "enable zoning" seçili ise; "part. no." alanına parça numarası ve "order no." alanına "0" girildikten sonra "START" butonuna basıldığında; iş-parçası, bölge numarası uyan olan ilk boş rafa yerleştirilir. Eğer "enable zoning" seçili değilse, sistem iş-parçasını bölge numarasına bakmaksızın bulduğu ilk boş rafa yerleştirir.

UNo=U	Z=1	UNo=U	Z=1	UNo=U	Z=1	UNo=U	Z=1
1	60410	2	60410	3	60410	4	60410
Q=1		Q=1		Q=1		Q=1	

RetrieveStoreMove

Status window

Part no. 62400

Order no. 0

Zone 0

Shelf no. 0

☒ search for empty shelf

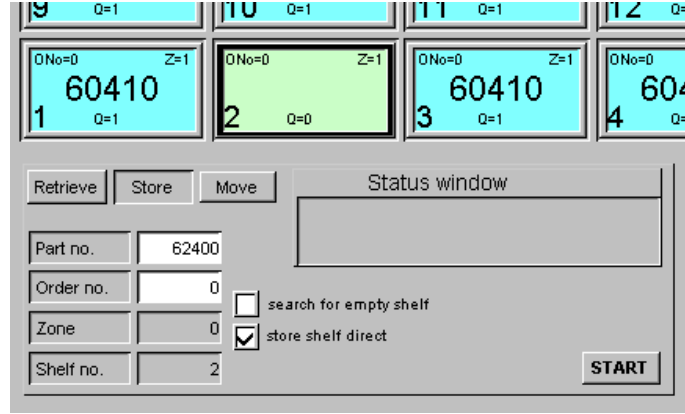
☐ store shelf direct

☐ enable zoning

START

Şekil 2-45. Stok yönetim ekranı üzerindeki sakla (store) işlevi "boş raf arama modu" bölümü görüntüsü.

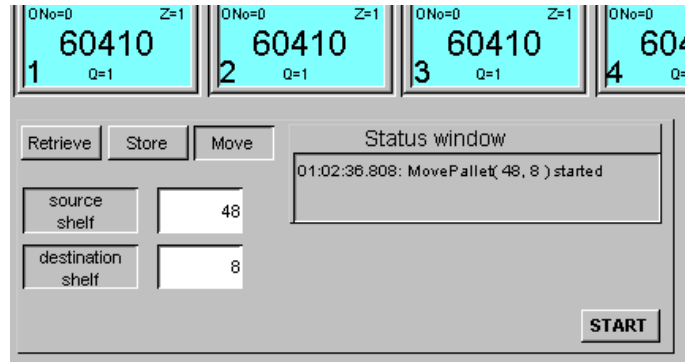
Bir diğer saklama yöntemi de parçanın direk olarak istenen rafta saklanmasıdır. Bunun için "store shelf direct" kutusu işaretliken parça numarası ve raf numarası parametreleri girildikten sonra "START" butonuna basılmasıyla birlikte, sistem parçayı istenen rafa yerleştirir. Şekil 2-46'deki örnekte boş olarak görülen 2 numaralı rafa 62400 seri numaralı iş-parçası depolanır.



Şekil 2-46. Stok yönetim ekranı üzerindeki sakla (store) işlevi “direk raf numarası ile depola” bölümü görüntüsü.

2.1.2.13.4. Taşıma İşlevi

İstendiği takdirde stoktaki raflarda bulunan iş parçaları raflar arasında taşınabilir. Bunun için “Move” butonuna basılarak taşıma moduna geçilir. Taşınmak istenen iş-parçasının bulunduğu kaynak ve taşınmak istendiği hedef raf numaraları Şekil 2-47’da görülmekte olan “source shelf” ve “destination shelf” alanlarına parametre olarak girilir. “START” butonuna basılmasıyla birlikte kaynak ve hedef raflar seçili duruma gelir. Teleskop kaynak raftaki iş-parçalarını taşıyan paleti alarak hedef rafa taşır. Şekil 2-48’deki örnekte 48 numaralı raftaki yardımcı palet 8 numaralı rafa taşınır.

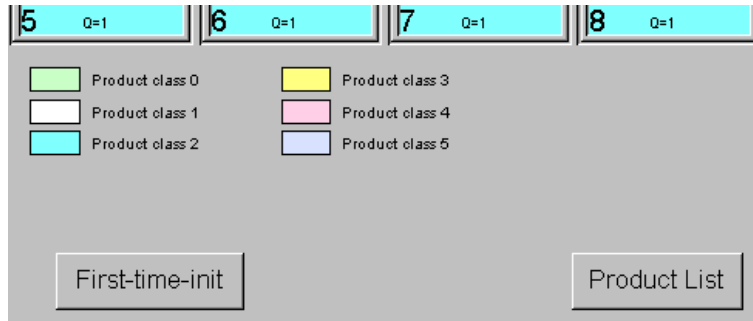


Şekil 2-47. Stok yönetim ekranı üzerindeki taşı (move) işlevi görüntüsü.

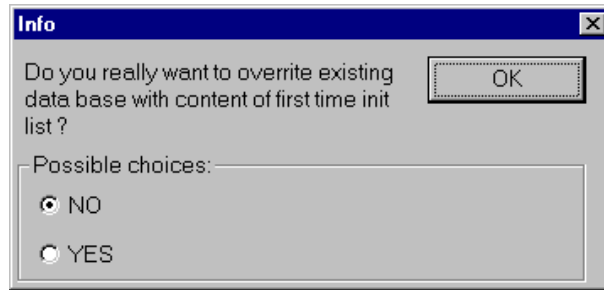
Stok hücresinde taşıma işlemi yapılırken sistem öncelikle kaynak ve hedef konumların dolu olup olmadığını kontrol eder. Bu işlemi Bölüm 2.1.1.9’da anlatıldığı üzere yardımcı palet üzerindeki optik yansıtıcı ile Teleskop üzerindeki optik algılayıcı sayesinde yürütür. Taşıma işlevi kaynak konumun boş olması veya hedef konumun dolu olması durumlarında ayrı hata mesajları döndürür.

2.1.2.13.5. İlk Kez Başlatma İşlemi

Stok hücresindeki raflarda bulunan iş-parçaları ile veritabanında bulunan kayıtların eşleşebilmesi için ilk kez başlatma (first-time-init) işlemi gerçekleştirilir. Bunun için iş-parçalarını taşıyan yardımcı paletler, parça numaralarına göre Bölüm 2.1.1.11.2’de anlatıldığı düzende dizilir. Şekil 2-48’da görülmekte olan “first-time-init” butonuna basılmasıyla birlikte, veritabanı kayıtlarının değişebileceği üzerine Şekil 2-49’deki uyarı mesajı görüntülenir. Kullanıcının onay vermesiyle birlikte veritabanı kayıtları da başlangıç düzenine getirilerek eşleşme sağlanır.



Şekil 2-48. Stok yönetim ekranı üzerindeki “First-time-init” ve “Product List” butonları görüntüsü.



Şekil 2-49. Veri tabanı kayıtlarının değişebileceğine dair uyarı mesajı görüntüsü.

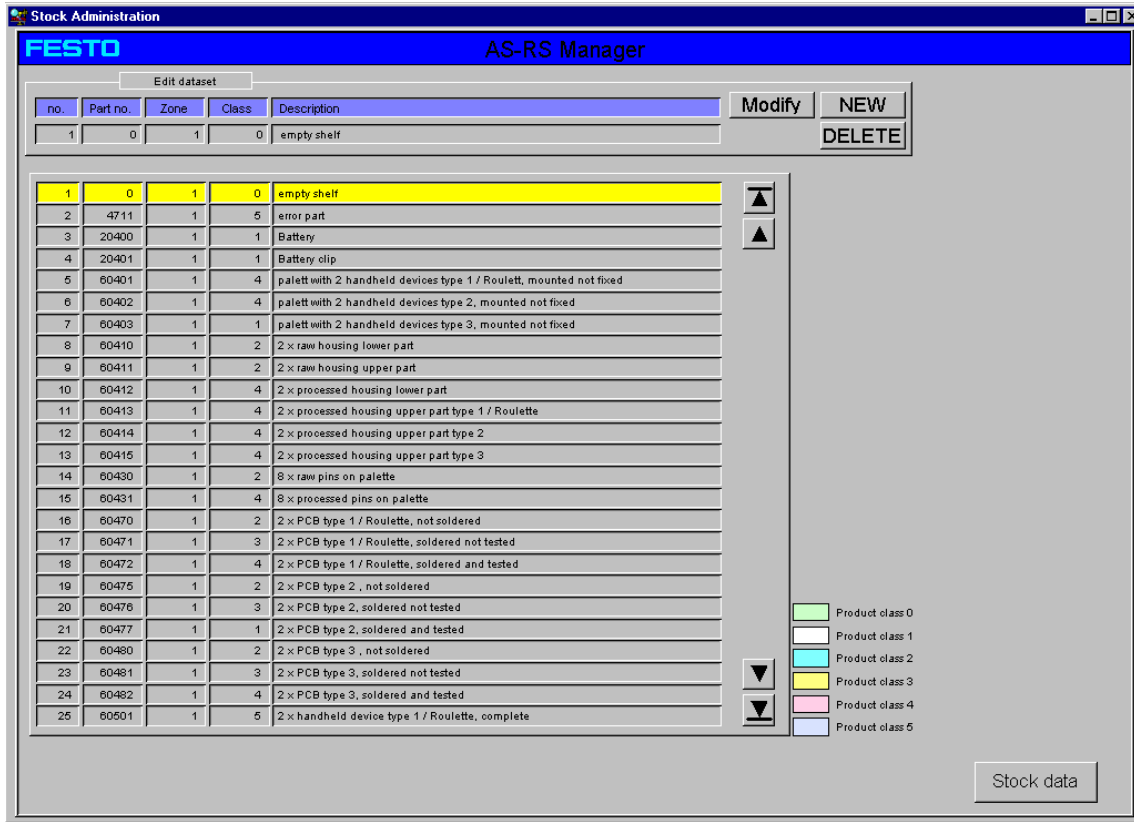
2.1.2.13.6. Ürün Listesi ve Düzenlenmesi

FMS100 bünyesinde kullanılacak, işlenecek ve üretilecek tüm iş-parçalarına ait kayıtlar “Product List” butonuna basılmasıyla gelen ekranda listelenir. Listede her iş-parçasına ait parça numarası, bölge, sınıf ve açıklama alanları yer alır. İş-parçaları raflarda farklı dolgu renkleriyle gösterilerek şu şekilde sınıflanmıştır:

- “Product class 0”: açık yeşil, hiç parça yok (boş)
- “Product class 1”: beyaz, işlenmemiş parça (ham)
- “Product class 2”: yeşil, işlenmiş kapak
- “Product class 3”: sarı, işlenmiş silindir

- “Product class 4”: pembe, silindir taşıyan palet
- “Product class 5”: mavi, kapak taşıyan palet

“UPDATE” butonuna basıldığında listenin üst tarafında düzenleme yapılabilecek alanlar ile “Modify”, “New” ve “Delete” butonları görüntülenir. Listedeki herhangi bir iş-parçası seçilerek “Modify” butonuna basılmasıyla birlikte listenin üstündeki alanlar aktif olur. Bu alanlarda yer alan bilgilerde değişiklik yapıldıktan sonra “Save” butonuna basılmasıyla birlikte değişiklikler kaydedilir. “New” butonuna basılmasıyla yeni bir kayıt açılabilir ve “Delete” butonuyla da mevcut kayıt silinebilir.



Şekil 2-50. “Product List” penceresi görünümü.

2.1.2.13.7. Yazılıma Ait Uyarı ve Hata Mesajları

AS-RS Manager ekranı üzerinden çeşitli işlevleri yerine getirirken uyarı ve hata mesajları ile karşılaşılabilir. Her mesajdan sonra kullanıcının onayı istenir. Görüntülenen mesajlar ve açıklamaları kısaca aşağıdaki gibidir:

- “no entry found in database”: Getir modundayken ve parça numarası arama ekranında veritabanında olmayan bir parça numarası girildiğinde;
- “no part available in selected source shelf”: Getir modunda boş bir raf seçildiğinde;

- “no empty shelf or no shelf with same zoning class for requested store command found”: Bölge numarası hatalı girildiğinde;
- “destination shelf already occupied”: Taşıma veya Saklama modunda hedef raf dolu olduğunda;
- “the part number you tried to store is not yet defined in product list Please enter first dataset into product list and try again !”: Saklama modunda veritabanında bulunmayan bir parça numarası girildiğinde.

2.1.2.14. İmalat Hücresi Kontrolü

Her imalat hücresinin kendi kontrolörü vardır. Bu kontrolör hücrenin imalat işlerini etkili bir şekilde yürütmesini ve gerektiğinde diğer istasyonlar ile yüksek seviyede koordineli çalışabilmesini sağlamaktadır. FMS100 sistemi ile koordinasyon SCADA yazılımı üzerinden sağlanır.

Yazılım, hücre kontrolörleriyle koordinasyonu Profibus üzerinden gönderdiği iş emirleriyle sağlamaktadır. Bütün iş emirleri aynı yöntemle işlenir:

- SCADA, imalat hücresine parametreleriyle birlikte iş emrini gönderir;
- Hücre ACK mesajıyla geri bildirimde bulunur;
- Hücre imalat işlemini yürütür;
- Hücre, iş emrinin tamamlanmasının ardından parametreleriyle birlikte sonucu yazılıma bildirir.

Her bir imalat hücresi çalışmaya hazır durumda başlatıldığında, SCADA yazılımındaki hücre kontrol birimi tarafından istasyon denetleyicilerdeki iş emri arayüzleri de aktive edilir. Hücre kontrol birimi, 5 er saniyelik periyotlarla hücre denetleyicilerin durumunu kontrol eder. Böylelikle bir önceki imalat işlemi bitmeden, yeni bir iş emrinin gönderilmesi ve oluşabilecek karışıklık önlenmiş olur. SCADA yazılımındaki hücre kontrol birimi aktif olmadığı durumlarda, imalat hücresi bağımsız çalışma (stand alone) modundadır. Bu moddayken imalat hücresi kontrol paneli üzerinden çalıştırılabilir durumdadır.

2.1.2.15. FMS100 Bünyesindeki Parça Numaraları

Sistem bünyesinde kullanılan her bir işlenmiş/işlenmemiş iş-parçası, ham madde ve CNC yazılımlarını temsilen özel parça numaraları belirlenmiştir. İş-parçaları, Tablo 2-3’de görülmekte olan parça numaraları şeklinde iş emirlerinde veya imalat süreçlerinde parametre olarak kullanılmaktadır.

Tablo 2-3. FMS100 sisteminde kullanılan iş-parçalarına ait seri numaraları.

Standart Parçalar 20XXX	
20400	9V pil
20401	Pil bağlantısı
İş-parçası Takımları 60XXX	
60401	2 x montajı yapılmış pin çakılmamış ürün
60410	2 x işlenmemiş alt kapak
60411	2 x işlenmemiş üst kapak
60412	2 x işlenmiş alt kapak
60413	2 x işlenmiş üst kapak
60430	8 x işlenmemiş pin
60431	8 x işlenmiş pin
60470	2 x lehimlenmemiş baskı devre
60471	2 x lehimlenmiş fakat test edilmemiş baskı devre
60472	2 x lehimlenmiş ve test edilmemiş baskı devre
60501	2 x ürün (rulet)
Ekipmanlar 62XXX	
62400	2 x üst/alt kapak ve ürün taşıyan yardımcı palet
62401	2 x baskı devre taşıyan yardımcı palet
62402	8 x metal pin taşıyan yardımcı palet
Programlar 63XXX	
63412	Alt kapağı işleyen CNC programı
63413	Üst kapağı işleyen CNC programı

2.2. Çalışmada Uygulanan Eğitim Yöntemi

Yapılan bu çalışmada Bölüm 2.1’de detayları verilen eğitim amaçlı fabrika otomasyon sistemi için bir eğitim modülü hazırlanmıştır. Hazırlanan eğitim modülü 2015-2016 ve 2016-2017 eğitim yıllarında Marmara Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü öğrencilerine Esnek İmalat Sistemleri (EİS) dersi kapsamında uygulanmıştır. EİS dersi, Mekatronik Mühendisliği bölümü ders programında eski sistemde 8. dönemde zorunlu ders, yeni sistemde ise 6. dönemde seçmeli ders olarak yer almaktadır.

Modülün uygulanması kapsamında genel olarak öğretme ve anlatım yöntemleri kullanılmıştır. Bu kapsamda üretim ve imalat kavramları ile süreçleri, çeşitleri, bileşenleri

hakkında kavramsal bilgi verilmiştir. EİS tanıtılarak sistem bileşenleri, süreç dinamikleri, otomasyon piramidi hakkında teorik bilgiler verilmiştir. Sonrasında EİS'e örnek bir model olan FMS100'ü oluşturan sistemler tek tek gösterilerek tanıtılmıştır. Canlandırma ve gösteri yöntemleri kullanılarak FMS100'ü oluşturan imalat istasyonları tek tek çalıştırılmış ve sistemin bir bütün olarak işleyişi gösterilmiştir.

Aktif öğretim yöntemi kapsamında genel olarak öğrencilere istasyonlarda uygulayacakları çeşitli görevler verilmiş ve kendi kendilerine öğrenme sağlanmıştır. Bu kapsamda uygulanan öğrenme yöntemleri ve aşamaları şunlardır:

1. aşama: Problem çözme (deney föyü verilerek uygulama yapmaları),
2. aşama: Soru cevap (Lab. eğitimi sırasında deney föyü uygulanırken öğrenci sorularını cevaplama)
3. aşama: Proje hazırlama (İmalat süreçlerinin sıralamasını değiştirerek kendi yapması, üretim süresi hesaplanması, konveyör hızının hesaplanması)
4. aşama: Sunum tekniklerini içermektedir. (Deney raporu, çalışma raporu hazırlıyor, ödevler)

EİS dersi, Mekatronik Mühendisliği bölümü öğrencilerinin lisans eğitimi sırasında almış oldukları birçok dersin içeriğini kapsamaktadır. Bu nedenle öğrenciler EİS dersini almadan da otomasyon ve bileşenleri hakkında bilgi sahibidirler. EİS dersinde yeni bilgiler de verilmenin yanı sıra lisans eğitimi boyunca otomasyon alanında alınmış olan bilgilerin pekiştirilmesi sağlanmaktadır. Bu nedenle dersin ve bu çalışma kapsamında hazırlanan modüllerin öğrencilerin bilgi seviyesine ne kadar etkiğini ölçmek için 50 soruluk bir test hazırlanmıştır. Hazırlanan test, hiç değiştirilmeden, öğrencilere dönem başında, ara sınav ve final sınavı öncesinde uygulanmıştır. Test sonuçları 3.3 Tartışma bölümünde yorumlanmıştır.

3. BULGULAR VE TARTIŞMA

Bu bölümde FMS100 sisteminin eğitimi için hazırlanan ders notları ve deney föyleri yer almaktadır. Ayrıca bölümün sonunda FMS100 sisteminin eğitim yönünden yeterliliği tartışılmıştır.

Bu çalışma sonucu elde edilecek eğitim materyalleri öncelikle FMS100 sistemiyle birlikte verilecek eğitimde kullanılacağı için, bu bölüm içerisinde verilecek örnek kodlar hazırlanırken, FMS100 sisteminde bulunan donanımlar göz önüne alınmıştır. Örneğin PLC ile alakalı örnek kodlar açıklanırken S7-300 model PLC'ye göre hazırlanmış kodlar kullanılmıştır.

3.1. Hazırlanan Ders Notları

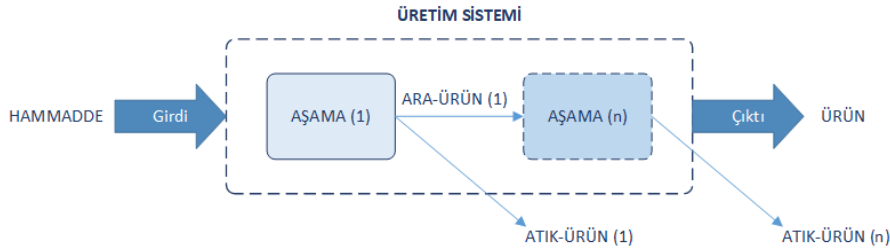
Bu kısımda eğitim amaçlı fabrika otomasyon sistemi FMS100 için hazırlanmış olan ders notları sunulmuştur.

Ders notları hazırlanırken, FMS100'ü üreten firmanın hazırladığı ve bölüm 2.MATERYAL VE YÖNTEM de Türkçeleştirilmiş olan tanıtım dokümanlarından da kısmen faydalanılmıştır. Özellikle direk FMS100 ile ilgili konularda faydalanılacak başka da kaynak bulunamadığından ve dokümanları Türkçeleştirmek için de ayrıca çaba harcandığından dolayı bu durum zaruri hale gelmiştir.

3.1.1. Ürün, Üretim ve İmalat Kavramlarına Giriş

Fabrika otomasyon sistemleri hakkında detaylara inmeden önce ürün, üretim ve süreçleri hakkındaki temel kavramlara değinmek gerekir. Kısaca bir üretim sisteminin girdisine hammadde, çıktısına ise ürün denir. Hammadde ürün olana kadar, üretim sistemi içerisinde aşama aşama çeşitli fiziksel veya kimyasal değişikliklere uğrar. Her bir aşama sonucunda ara-ürünler elde edilir. Elde edilen ara-ürünler, genellikle bir sonraki aşamanın hammaddesidir. Eğer ki elde edilen ara ürünler, üretim sistemi içerisinde değerlendirilmeden uzaklaştırılacaksa, bunlara da atık ürün denir. Atık ürünler bir başka üretim sisteminin hammaddesi olabilir veya çeşitli yöntemlerle imha edilebilir. Hammaddeden arzu edilen ürünün elde edilmesini sağlayan, bünyesinde çeşitli sayıda fiziksel veya kimyasal süreç aşamaları bulunduran yapıya da üretim sistemi denir. Büyüklüğü, girdi ve çıktıların çeşidi vb. özelliklerine göre üretim sistemi; atölye,

imalathane veya fabrika adını alabilir. Bir üretim sistemindeki hammadde, ürün, ara-ürün ve atık ürün ilişkisi Şekil 3-1'deki gibi tanımlanabilir.



Şekil 3-1. Bir üretim sistemindeki hammadde, ürün, ara-ürün ve atık ürün ilişkisi.

Üretim ve imalat kavramları anlam bakımından sıklıkla birbirlerinin yerine kullanılabilmektedir. Aslen üretim, doğada hazır olarak bulunan kaynaklardan ihtiyaç duyulan ürünleri elde etme sürecidir. Kaynaktan kasıt; petrol, meyve-sebze gibi yer altından ve üzerinden elde edilen her türlü işlenmemiş doğal malzemedir. Deniz kumundan silika eldesi, zeytinden yağ eldesi, pamuktan iplik eldesi vb. üretime örnek olarak verilebilir.

İmalat (manufacturing) ise, önceden işlenerek elde edilmiş (üretilmiş) ara-ürün veya ürünlerin yeniden işlenmesi sonucunda, daha değerli veya daha çok sayıda, farklı ara-ürün veya ürünler elde edilme sürecine denir. Örneğin silikadan titanyum çubuk eldesi, zeytinyağından sabun eldesi, iplikten kumaş dokunması vb. imalata örnek olarak verilebilir.

Üretimin, anlam olarak kapsamı imalattan daha geniştir. Fakat örneğin kimse bir uçak gövdesinin deniz kumundan üretildiğini söylemez. Bunun yerine uçak gövdesinin titanyum çubuklardan imal edildiği söylenir ki; titanyum çubuklar da silikadan, aslında deniz kumundan üretilir.

Otomasyon teknolojisi açısından da üretim ve imalat süreçlerinde farklılıklar vardır. Üretimde genellikle kazıp çıkarma, karıştırma, kesme, sıkma, eleme, ezme, öğütme, kaynatma, damıtma gibi daha temel ve kaba fiziksel teknikler kullanılır. İmalatta ise kaynak yapma, CNC torna ve freze ile işleme, lazer ile kesme, boyama, birleştirme, lehimleme gibi daha ileri seviye karmaşık teknolojiler kullanılır. Ayrıca üretim sürecinde elde edilen ürünler hala birçok endüstri için hammadde sınıfında olup, imalat sürecinde elde edilen ürünler genellikle son kullanıcıya hitap edebilecek seviyededir.

Sonuç olarak her ne kadar köken ve anlam bakımından üretim ile imalat arasında farklılıklar olsa da, literatürde birbirleri yerine sıklıkla kullanılabilmektedir.

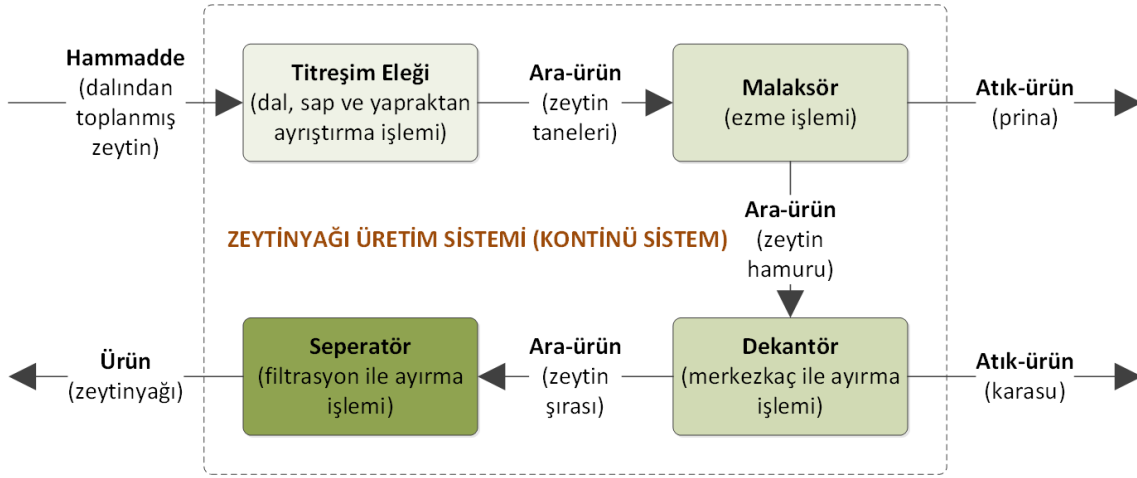
İktisadi açıdan üretim, bireylerin ihtiyaçlarını karşılamak üzere yürütülen faaliyetler ile gerekli malzemeleri elde edilme süreci olarak tanımlanır. Bu süreç genel olarak ham maddelerin hazırlanması ve temini, ara ürünlerin elde edilmesi ve montajı aşamalarından oluşur. İşletme yönetimi açısından bakıldığında, üretim ve imalat süreçleri; üretim yöntemi, ürün cinsi, ürün miktarı ve akış süreci gibi farklı ölçütlere göre sınıflandırılabilir. Endüstriyel otomasyon açısından üretim ve imalat süreçleri, çıktıları göz önüne alınarak; sürekli (continuous), ayırık (discrete) ve yığın (parti-batch) olmak üzere üç sınıfa ayrılabilir.

3.1.1.1. Sürekli Üretim

Sürekli üretimde çıktı, sürekli bir ürün akışıdır. Hammadde süreç otomasyonunun bir ucundan girer ve üretim sisteminin çalışmasına ara verilmeden çeşitli işlemlerden geçirilerek, diğer uçtan ürün olarak çıkar. Özetle ham madde, geri dönüştürülemeyecek şekilde yapısal ve kimyasal değişikliklere uğratılarak ürün elde edilir. Süreç otomasyonu açısından irdelendiğinde, üretimde aşamalar arası malzeme ve enerji akışı sürekli olduğu için, tampon depolama alan kullanıma şansı yoktur. Bu durum üretim sisteminin başlatılma, normal operasyon, kapatılma ve acil durum senaryolarına ait kontrol otomasyonunu daha da güç hale getirir.

Sürekli üretime verilebilecek en iyi uygulama örneklerden biri, aşamaları Şekil 3-2'deki şemada özetlenmiş olan zeytinyağı üretimidir. Halk arasında “kontinü sistem” olarak adlandırılan bu üretim sisteminde girdi (hammadde), dalından toplanmış zeytinlerdir. Öncelikle titreşim (vibrasyon) eleğinde zeytin taneleri dal, sap ve yapraklarından ayrılarak basınçlı suyla yıkanır. Sonrasında malaksör tarafından zeytin taneleri ezilerek hamur haline getirilir. Ara ürün olarak elde edilen zeytin hamuru, pompa aracılığıyla dekantöre iletilir. Merkezkaç (santrifüj) yöntemiyle zeytin hamurundaki yağ ile posa (prina) birbirlerinden ayrılır. Son olarak ayırıcıda (seperatör) yağ tortularından ve sudan arındırılarak zeytinyağı elde edilmiş olur. Bu üretim sürecinin atık-ürünlerinden olan prina, yeniden işlenerek sabun endüstrisinde hammadde olarak veya kömür gibi farklı yakıtlarla karıştırılarak enerji üretiminde yakıt olarak kullanılabilir. Karasu ise baş edilmesi gereken kötü bir atıktır. Örneğin günümüzde yaygınlaşmaya başlayan solucan

gübresini üretiminde karasuyu kullanımı yoluyla gübre verimini arttırmaya ve böylelikle atığı faydalı hale getirmeye yönelik araştırma çalışmaları yürütülmektedir.

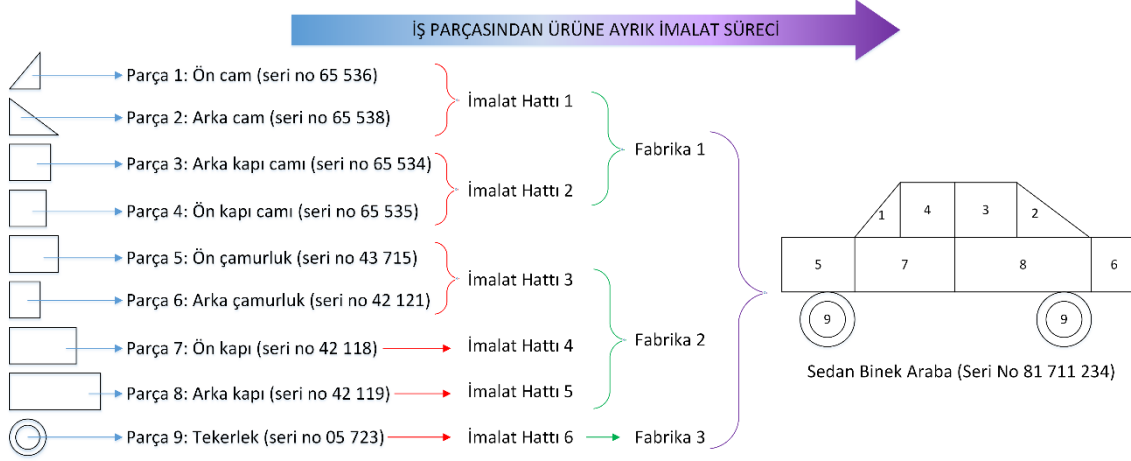


Şekil 3-2. Zeytinyağı üretim sistemi ve aşamaları.

Bu örneğin dışında, sürekli üretim, kimya ve petro-kimya endüstrilerinde yaygın olarak kullanılmaktadır.

3.1.1.2. Ayrık Üretim

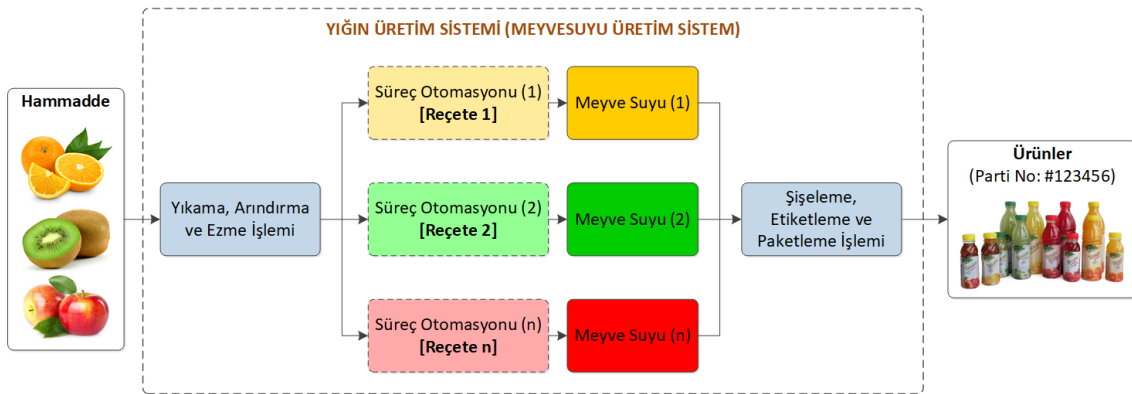
Bir ürünü meydana getiren bileşenlerin rastgele zamanlarda birbirlerinden bağımsız olarak, bazı durumlarda farklı tesislerde, üretilerek sonradan bir araya getirilmesine ise ayırık (kesikli) üretim denir. Bir başka ifadeyle, üretim yapılan tesisin sürekli aynı ürün için kullanılmadığı üretim biçimidir de denebilir. Ürünü meydana getiren bileşenler (iş parçaları) üretim hattı boyunca şekil değiştirme, malzeme olarak güçlendirme veya farklı parçalar ekleme gibi bir dizi fiziksel işlemden geçirilir. Bu sırada hammaddede bazı kimyasal değişimler gözlenebilse de, yine de ayırık üretimde gerçek anlamda kimyasal bir süreç söz konusu değildir. Genel olarak mekanik ve elektronik ürünlerin imalatında kullanılan bir yöntemdir. Üretim sürecinde tampon depo alanları ve üniteler kullanılabildiği için süreç otomasyonu sadece üretimdeki veri akışının bütünleştirilmesine odaklanır. Bu nedenle sistemin teknik olarak kontrolü sürekli üretime göre daha kolaydır. Ayrık üretim sonucunda elde edilen ürün, sürekli üretimden farklı olarak, genellikle kendisini oluşturan parçalarına ayrılabilir niteliktedir. Karışıklığı önlemek için Şekil 3-3'deki örnekte görüldüğü üzere, üretilen her bir parça (iş parçası) özel bir seri numarası ile etiketlenir. Araba, uçak, cep telefonu ve mobilya üretimi örnek olarak verilebilir.



Şekil 3-3. İş parçasından ürüne ayrik imalat süreci, örnek bir araba imalatı.

3.1.1.3. Yığın (Batch, Parti) Üretim

Yığın üretimde ayrik üretimde olduğu gibi hammaddeler bir dizi işlemde geçirilerek nihai ürün elde edilir. Farklı olarak yığın üretimde ürünler parçalarına ayrılabilir yapıda değildir. Sürekli üretimde olduğu gibi üretimde kimyasal süreçler yer alır ve üretim çıktısı homojen yapılı, geri döndürülemeyecek şekilde işlenmiş bir “yığın” üründür. Bu yönleriyle irdelendiğinde yığın üretim hem sürekli hem de ayrik üretim özellikleri gösterir. Yığın üretimde bazen ayrik üretimde olduğu gibi, birçok farklı ürün aynı tesis içerisinde ve eş zamanlı olarak üretilebilir. Şekil 3-4’de gösterilen meyve suyu üretimi buna en güzel örneklerden biridir. Her meyve suyuna ait üretim süreci temelde farklı olabilse de; yıkama, saplarından ayırma, ezme, pastörize etme ve şişeleme gibi aşamalar tüm meyve suyu çeşitleri için ortak olabilir. Sonuç olarak yığın üretim yapan tek bir tesiste, farklı tipte meyve suları farklı ambalajlarda olacak şekilde üretilebilir.



Şekil 3-4. Örnek bir yığın üretim sistemi, meyve suyu üretimi.

Yığın üretimde elde edilecek ürün reçete ile tanımlanır. Diğer bir ifadeyle, ürünün elde edilebilmesi için kullanılacak bileşenler ve katkı oranları ile süreç otomasyonunda uygulanacak ısıtma/soğutma, karıştırma vb. işlemlere ait parametrelerin tanımlandığı dosyaya reçete denir.

Yığın üretim süreci ve otomasyonu oldukça karmaşıktır. Aynı tesiste farklı zamanlarda üretilen ürünlerin yapısal özellikleri az da olsa farklılık gösterebilir. Örneğin yoğurtta olduğu gibi aynı markanın aynı tip yoğurdu, farklı zamanlarda farklı tatlarda, fiziksel ve kimyasal özellikte olabilir. Bu değişimin en büyük sebebi, farklı zamanlarda üretime giren hammadde olan sütün farklı yapıda olabilmesidir. Hammaddedeki değişim, mayalanma süreci sonunda elde edilen üründe de farklılıklara neden olur. Bu nedenle yığın imalatı farklı tarihlerdeki girdiler ile elde edilen üretim çıktılarını birbirlerinden ayırmak için parti (lot) numarası kullanılır. Böylelikle aynı partide üretilen, aynı lot numarasına sahip ürünlerin fiziksel ve kimyasal özellikleri de “benzerdir” denilebilir. Ürünün kalitesinde bir hata tespit edilirse, tüm parti bundan etkilenir. ISA-S88.01 standardında tanımlanmış olan yığın imalat kimya, gıda, ilaç, kozmetik ve biyoteknoloji endüstrilerinde yaygın olarak kullanılır.

3.1.2. Bilgisayar Destekli İmalat ve Türevleri

Uluslararası rekabet, günümüzde küresel pazarda mücadele edebilecek düzeyde kaliteli ürünler geliştirilmesi ihtiyacını arttırmaktadır. Günümüzde artan rekabet koşulları nedeniyle endüstriyel gelişimin hızı da artmaktadır. Bu durum imalatçıları sadece birbirleriyle rekabet için değil, ayakta kalabilmek için bile sürekli gelişen kalitede üretim yapmaya zorlamaktadır. Pazara sunulan ürünlerin öngörülen ömrü azaldıkça ve yüksek kaliteli ürün ihtiyacı arttıkça; ürün geliştirmeye verilen önem de artmıştır. Bununla birlikte daha verimli imalat, yönetim ve pazarlama yöntemleri kullanılarak endüstriyel işletmelerin rekabet kapasiteleri de artmıştır. Sonuç olarak mekanik ve elektronik ürünlerde geleneksel imalat yöntemlerinin bırakılarak daha esnek, düşük maliyetli, kaliteli ve hızlı imalatı mümkün kılan Bilgisayar Tümlü İmalat (Computer Integrated Manufacturing - CIM) sistemlerinin geliştirilmesine sebep olmuştur.

CIM, ürüne ait tüm imalat aşamalarının bilgisayar aracılığıyla yönetildiği bir imalat yöntemidir. CIM sayesinde geleneksel ve ayrıık üretim yöntemleri bilgisayar aracılığıyla birleştirilir. Ayrıık yürütülen işlemlerin birbirleriyle bilgi alışverişi sayesinde koordineli

alıřmaları saęlanır. Bylelikle retim tesisleri hızlı ve hataları asgari seviyede olacak řekilde kurulabilir, tesise ait imalat otomasyon sistemi kolaylıkla oluřturulabilir.

CIM kavramı, sadece belirli bir rn retmeye ynelik olarak tasarlanmış olan Adanmış İmalat Sistemleri (AİS) ve birbirinden farklı rnleri aynı sistem ierisinde retebilecek yapıda tasarlanmış olan Esnek İmalat Sistemleri (EİS) olmak zere iki blmde incelenebilir.

Aynı tip rn uzun sre boyunca ve yksek adetli olarak retilmek istendięinde, AİS tercih edilir. Bylelikle hem retim sisteminin kurulum maliyeti hem de retim kendi maliyeti olduka dřk seviyede tutulabilir. Kısacası AİS’de, tek tesis tek rn yaklařımı vardır. rn eřitlilięi yok denecek kadar azdır. Buna karřılık retim adetleri olduka fazladır. Bu tip imalat sistemleri kolay deęiřmeyen, uzun sre kullanımda kalan rnler iin uygundur. rneęin toplu ięne veya ahřap kurřun kalem gibi yıllar boyunca řekli, biimi ve iřlevi deęiřmeyecek; retim adetleri yksek olan rnlerin retimi iin AİS uygun bir imalat sistemidir.

Bir retim sisteminde kk aplı ve dřk maliyetli deęiřiklikler yapılarak, birbirleri ile ortak ynleri bulunan farklı rnler retilmek istendięinde EİS tercih edilir. EİS rn eřitlilięinin bol, retim adedinin az olduęu tesislerde maliyet avantajı saęlar. Bu bakımdan zetlenirse EİS’de, tek tesis ok eřitli rn yaklařımı vardır. rn eřidi fazla olduęu halde, her bir eřidin retim adedi AİS’e gre “azdır” denebilir. EİS’deki rn eřitlilięi kavramı, birbirleri ile ortak ynleri olan rn eřitleri anlamına gelmektedir. rneęin cep telefonu retmek iin kurulmuş bir EİS’de, farklı zelliklere sahip cep telefonları retilir. retilen telefonların hepsinde dokunmatik ekran, mikrofon-hoparlr ve tuř takımı gibi zellikler benzer olsa da; telefon eřitlerinde farklı kapasitede iřlemci, RAM, hafıza elemanı kullanılıyor olabilir. Aynı EİS’de, yine birbirleri ile benzerlikleri bulunan cep telefonu ile tablet retilir. Fakat bir cep telefonu ile neredeyse hibir benzerlięi olmayan bir buzdolabının, aynı EİS bnyesinde retilmesi zordur. Byle bir EİS tasarımı, EİS’in “ok eřitli rnleri az maliyetle tek sistemde retebilme” felsefesine aykırıdır. Cep telefonlarının ortalama iki yıl mr olduęu ve elektronik cihazların her sene yeni modellerinin piyasaya srldę de gz nne alındıęında, EİS ile retilen rnlerin mrleri AİS’le retilenlere gre “kısadır”

denilebilir. Şekil 1.6’da ürün adedi ve çeşidine göre AİS ile EİS arasındaki ilişki grafiği gösterilmektedir.

Esneklik, değişen piyasa koşullarına göre üretimin hızlı ve etkili bir şekilde uyum sağlayabilme yeteneğine denir. Üretimde esneklik oldukça geniş bir kavram olduğu için aşağıdaki gibi alt sınıflara ayrılabilir:

- Tezgah Esnekliği: Parça işleme çeşitliği, gerekli ayarların kolay ve çabuk yapılabilme ölçütüdür.
- Üretim Süreci Esnekliği: Farklı malzemeler kullanarak ve farklı şekillerde üretim yapılabilme ölçütüdür.
- Ürün Esnekliği: Yeni bir ürüne imalatına geçebilme ölçütüdür.
- İş Akışı Esnekliği: Üretim tesisinde meydana gelebilecek beklenmedik bir aksaklığın üretimi ne ölçüde etkileyeceğinin ölçütüdür.
- Hacim Esnekliği: Gelen siparişe göre üretim adetlerini değiştirebilme ölçütüdür.
- Genişleyebilme Esnekliği: Üretim tesisinin kolay bir şekilde büyütülebilmesi (kapasite artırımı) ölçütüdür.
- İşlemsel Esneklik: Ürüne ait her bir parçanın işlem sıralamasının değiştirilebilme ölçütüdür.

Günümüzde EİS’lerin çok daha etkileşimli ve esnek sürümü olan akıllı fabrikalardan söz edilmektedir. Başını Almanya, ABD ve Japonya’nın çektiği gelişmiş ülkeler Endüstri 4.0 kavramı ortaya atmaktadır. Kavram, temelde üretim sürecinde yer alan tüm cihazlar, müşteri sipariş süreçleri ve tedarik zincirinin Internet üzerinden birbirleri ile iletişim halinde olması prensibine dayanır. Üretim ile ilgili girdi/çıktı verileri bir merkezde toplanır. Toplanan veriler, büyük veri (Big Data), veri madenciliği yöntemleriyle analiz edilerek üretimdeki maliyet hesabının keskinliği ve verimlilik arttırılmaya çalışılır. Böylelikle yüksek teknolojiye sahip ve kişiselleştirilmiş ürünlerin, mevcut üretim altyapısı kullanılarak üretim maliyetlerinin azaltılması hedeflenir. Nesnelerin Interneti, Bulut Bilişim, Big Data, RFID ve optik etiketleme teknolojileri, 3 boyutlu yazıcı ve endüstriyel robotlar bu “dördüncü endüstriyel devrim”in ardındaki lokomotif teknolojilerdir. Şekil 1.9’da endüstriyel devrimin tarihsel aşamaları sembolik olarak gösterilmiştir.

3.1.3. Esnek İmalat Sistemlerinin Temel Bileşenleri

Hiyerarşik açıdan Bilgisayar Destekli İmalat (CIM) kavramının altında yer alan Esnek İmalat Sistemleri (EİS), yapısı ve işleyiş şekli itibarıyla ayrık üretim sistemi sınıfına

girmektedir. Bu nedenle ayrıık üretim sistemine ait temel teknolojiler ve bileşenler řu şekilde özetlenebilir:

3.1.3.1. Hammadde, İş-Parçası ve Ürün

Bir EİS'in girdisi tedarik zinciri yönetimi (supply chain management) tarafından sağlanan hammaddedir. Hammadde (raw material), işlenmemiş veya yarı-ürün olarak sisteme girebilir. EİS bünyesinde dolaşımda olan ve üzerinde imalat işleri yürütölmekte olan maddeye ise iş-parçası (workpiece) denir. Tüm imalat ve hata kontrol işlemleri tamamlanmış iş-parçasına ise ürün (product) denir.

3.1.3.2. İstasyon ve Hücre Kavramları

EİS kapsamında nihai ürünün eldesine yönelik kesme, delme, kaynak yapma, lehimleme, birleştirme vb. her tür imalat işleminin yapıldığı birimlerdir. Bünyelerinde (torna, freze, vb.), muayene ve test cihazları, yıkama-boyama alanları, yükleme-boşaltma alanları, robotik elemanlar bulunabilir. Her ne kadar amaçları aynı olsa da, istasyon ve hücre kavramları arasında nüans vardır. İş-parçası üzerindeki imalat işleri taşıma sistemi üzerinde yapılıyorsa, bu birim bir imalat istasyonudur. İş-parçası taşıma sistemi üzerinden alınıyor, ayrı bir birimde işleniyor ve tekrar taşıma sistemi üzerine bırakılıyorsa, bu birime imalat hücresi denir. FMS100 sistemi bünyesindekilerin hepsi imalat hücreleridir.

3.1.3.3. Taşıma Sistemi

EİS bünyesinde iş-parçalarının imalat istasyonları ve hücreleri arasında taşınmalarını sağlar. İş-parçaları genellikle palet adı verilen alt sistemler üzerinde konumlandırılır ve taşıma sistemine palet üzerinde yerleştirilir. Paletler iş-parçasının taşıma sistemine yüklenmesi ve geri alınması işlerinde kolaylık sağlar. Malzemelerin taşıma sistemindeki takibi için paletler üzerine optik veya manyetik (RFID) etiketler de eklenir.

EİS bünyesinde en sık kullanılan taşıma sistemi konveyör bantlarıdır. Bunlar istendiğı zaman durdurulabilen veya hızı değıştirilebilen motorlar tarafından makara ile sürölen elastik bantlardır. Paletler bu bantlar üzerinde hareket eder. Günümüzde gelişen elektrikli ve otonom araç teknolojisi sayesinde optik veya lazer tarayıcı (lidar) kontrollü araçlar da iş-parçalarını taşıma işlerinde yaygınlaşmaktadır.

3.1.3.4. Depolama Sistemi (AS-RS)

Bir imalat sisteminin olmazsa olmaz bileşenlerinden biridir. İmalat sırasında işlenmemiş hammaddeler, iş-parçaları ve ürünlerin depolandığı alana denir. Genellikle sıra sıra

dizilmiş çok katlı raflardan oluşur. Tasarıma göre silindir biçiminde yerleştirilmiş raflardan da oluşabilir. Karışıklığı önlemek için her raf farklı bir numara ile kodlanır. Kodlamada optik veya manyetik etiketler (RFID) de kullanılabilir.

3.1.3.5. Eyleyiciler

Otomasyon alanında parçaların, mekanizmaların hareketini sağlayan elemanlara denir. Aktif olarak iş yapan ($\dot{I}ş = Kuvvet \times Yol$) elemanlardır. Kontrol döngülerinde genellikle çıkış elemanlarıdır. Hava tahrikli (pnömatik) veya sıvı yağ tahrikli (hidrolik) silindirler, kızaklar ile her türlü motor (senkron, asenkron, adım motoru, fırçalı-fırçasız dc motor, vb.) bu grupta yer alır. Çalışmaları genellikle elektronik bir sürücü devre ile sağlanır.

3.1.3.6. Algılayıcılar

Otomasyon sistemlerinde her tür fiziksel büyüklüğü (sıcaklık, basınç, kuvvet, yoğunluk, ivme, kapasitans, indüktans, mesafe, vb.) algılamak veya ölçmek için kullanılan elemanlardır. Kontrol döngülerinde genellikle çıkış değişkenini ölçen geri besleme (feedback) elemanı olarak kullanılırlar. Endüstriyel uygulamalarda ise imalat sürecinin yürütülebilmesi için kilit eleman konumundadırlar. Örneğin bir yürüyen bant üzerinde ilerleyen kolinin bant üzerinde etiket basan cihazın altına gelip gelmediği yaklaşım algılayıcıları (proximity) tarafından algılanır. Koli gelmişse bantın durması, etiketleme işleminin başlatılması işlemleri vb. yaklaşım algılayıcısı tarafından tetiklenir.

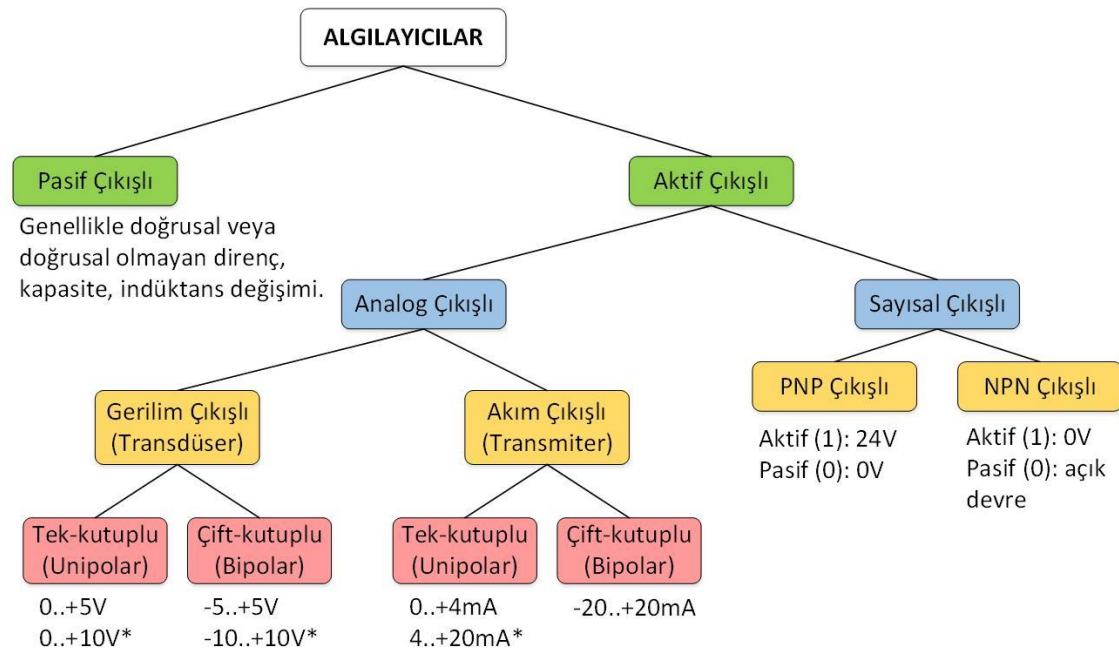
Algılayıcılar ürettikleri çıkışa göre pasif ve aktif olarak iki gruba ayrılırlar. Pasif çıkışlı algılayıcılar, algıladıkları fiziksel büyüklüğü genellikle elektriksel direnç, kapasite veya indüktans değişimi olarak ifade ederler. Algıladıkları büyüklüğe göre ürettikleri çıkış değişimleri doğrusal olmaya da bilir. Bu nedenle ölçüm işlerinde kullanılabilmeleri için çıkışlarını doğrusallaştıran özel tasarım harici elektronik devrelere ihtiyaç duyarlar.

Aktif çıkışlı algılayıcılar ise, bünyelerinde sahip oldukları elektronik devreler sayesinde algıladıkları fiziksel büyüklüğü çıkışlarında gerilim veya akım değişimi olarak ifade ederler. Ayrıca bir elektronik devreye ihtiyaç duymazlar. Aktif çıkışlı algılayıcılar kendi içinde analog ve sayısal çıkışlı olmak üzere ikiye ayrılır.

Analog çıkışlı algılayıcılar, algıladıkları fiziksel büyüklükleri teorik olarak sonsuz olasılıkla ifade ederler. Gökkuşağının renkleri gibi çıkışları tanım aralığı içerisinde herhangi bir değer olabilir. Gerilim ve akım çıkışlı olmak üzere iki tipte olabilirler. Gerilim çıkışlı olanlarına transdüser (transducer), akım çıkışlı olanlarına transmitter

(transmitter) denir. Analog çıkışlı algılayıcılar da kendi içlerinde tek kutuplu (unipolar) ve çift kutuplu (bipolar) olmak üzere ikiye ayrılırlar. Tek kutupluların çıkışları 0 ile pozitif değerler arasında değişirken, çift kutuplular çıkışlarında negatif veya pozitif değer üretebilirler.

Sayısal çıkışlı algılayıcıların çıkışları mantıksal 0 veya 1 gibi iki durumlu olabilir. Algıladıkları değişken ya vardır ya da yoktur, siyah veya beyaz gibi. Herhangi bir ara değer söz konusu değildir. Elektriksel olarak NPN veya PNP çıkışlı olmak üzere iki tipte olabilirler. PNP çıkışlı olanlar, mantıksal 1 çıkışını algılayıcının besleme gerilimi; mantıksal 0 çıkışını ise şase gerilimi ile ifade ederler. NPN çıkışlı olanlar ise mantıksal 1 çıkışını şase gerilimi; mantıksal 0 çıkışını ise akım geçişine izin vermeyen “sonsuz” direnç ile ifade ederler. Algılayıcıların çıkış tiplerine göre sınıflandırılması Şekil 3-5’de verilmiştir.



* Endüstride en sık kullanılan çıkış standartlarıdır.

Şekil 3-5. Algılayıcıların çıkış tiplerine göre sınıflandırılması.

İmalat sistemlerinde sıkça kullanılan algılayıcı tiplerinden biri olan yaklaşım algılayıcıları aktif sayısal çıkışlı grubuna girer. Yaklaşım algılayıcıları manyetik, kapasitif, indüktif veya optik değişimleri kullanarak algılama alanları içerisinde bir cisim olup olmadığını algırlarlar. Algılanmak istenen cismin tipine (metal, ametal, saydam, vb.) göre seçilirler ve “var-yok” gibi iki durumlu çıkış üretirler.

3.1.3.7. Programlanabilir Denetleyiciler

EİS bünyesindeki imalat istasyonu ve hücrelerinin istenilen şekilde çalışabilmelerini sağlayan, imalat sürecini yürüten ve denetleyen elemanlardır. Kontrol döngülerindeki karşılaştırma işlemleri ile kontrol algoritmasını yürütmek için kullanılırlar. Çeşitli sayıda sayısal ve analog giriş/çıkışa sahiptirler. Genellikle girişlerine buton, anahtar ve algılayıcı gibi elemanlar bağlanır. Çıkışlarına ise eyleyicileri çalıştıran sürücüler bağlanır. Programlanabilir Mantıksal Denetleyici (Programmable Logic Controller - PLC) en yaygın kullanılan elemandır. EİS'in büyüklüğüne ve ihtiyaçlarına göre endüstriyel bilgisayar (Industrial Personal Computer - IPC) veya dağıtık kontrol sistemleri (Distributed Control System) de denetleyici olarak kullanılabilir.

3.1.3.8. Endüstriyel Haberleşme Sistemleri

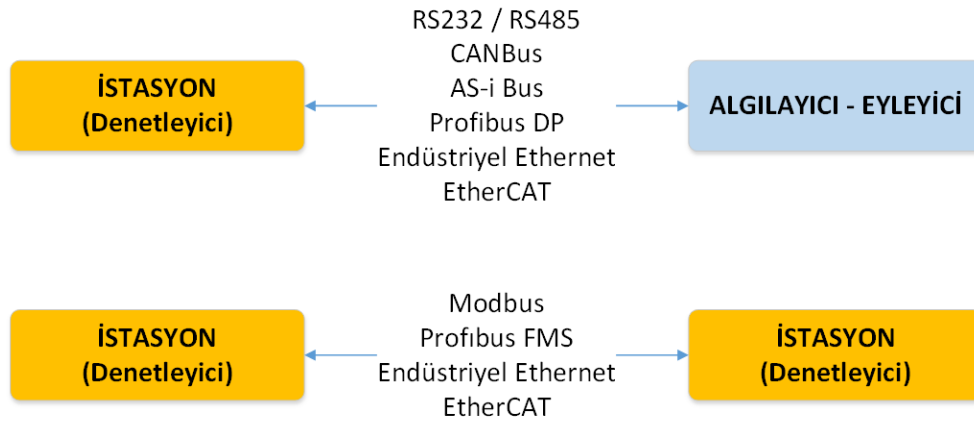
EİS bünyesinde farklı yapıdaki birçok teknoloji birlikte ahenk içerisinde çalışır. Bu nedenle elemanlar arası sağlıklı bir iletişim kurulması kilit öneme sahiptir. Kullanılan elemanların seviyelerine veya tiplerine göre birbirleri ile iletişim yöntemleri de farklılık gösterir. Burada seviyeden kasıt, iletişimin istasyonlar/hücreler arası mı kurulacağı yoksa istasyonlar/hücreler bünyesindeki algılayıcı veya eyleyiciler arasında mı kurulacağıdır.

Bir imalat sisteminde asıl işi yapan elemanlar algılayıcı ve eyleyicilerdir. Bu yüzden bu elemanları kontrol eden programlanabilir denetleyiciler ile aralarındaki iletişim hızı yüksek olmalıdır. Böylelikle imalat işleri yüksek hassasiyet ve hızla yürütülebilir. Fakat farklı işleri yürüten istasyonlar/hücreler arası iletişimin hızlı olması çok da gerekli değildir. Bunun yerine güvenilirlik ve uzak mesafelere erişim kriterleri ön plana çıkmaktadır.

Bu duruma öğretmen-öğrenci ile öğretmen-öğretmen arasındaki iletişim yoğunluğu farkı örnek olarak verilebilir. Öğretmen, öğrencisini belirli bilgi ve beceri seviyesine getirebilmek için daha yoğun bir iletişim kurmak durumundadır. İletişim daha hızlı ve yakındır. Fakat iki öğretmenin birbirleri arasındaki iletişim yoğunluğu görece olarak daha az, yavaş ve seyrek.

Endüstride farklı özelliklere sahip marka bağımlı ve bağımsız haberleşme protokolleri bulunmaktadır. Bunlardan bazıları (RS232, RS485) (1969/1998) Electronics/Telecommunication Industries Association (EIA/TIA), (AS-i Bus) (1994) AS-International Association gibi bağımsız kuruluşlar tarafından geliştirilmiş olsa da;

bazıları da (CANBus) (1991) Bosch ve (Modbus) (1979) Scheinider gibi ticari kurumlar tarafından geliştirilmiş ve sonradan genel kullanıma açılmıştır. Profibus ve türevleri gibi bazı protokoller de, sadece kendini geliştiren firmalara (Siemens) ait ürünlerde kullanılmaktadır. Diğer firmalar lisans ücretlerini ödeyerek bu protokolleri kullanabilmektedir. Şekil 3-6'da istasyon/hücre denetleyicileri arası ve algılayıcı/eyleyici gibi alt birimler arası iletişimde kullanılabilecek örnek haberleşme protokolleri verilmiştir.

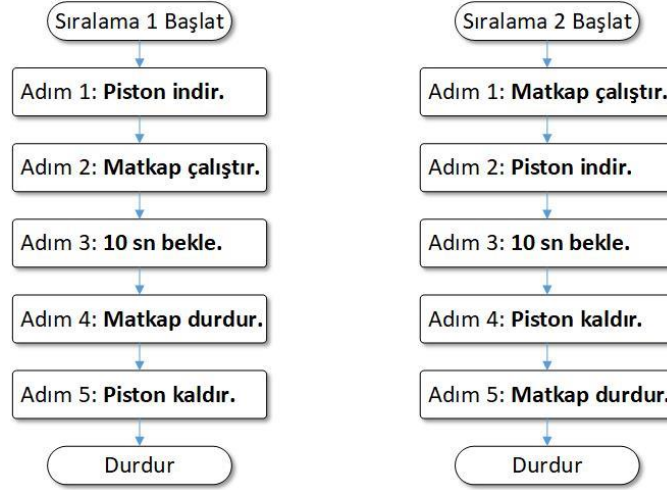


Şekil 3-6. EİS bünyesinde kullanılabilecek örnek haberleşme protokolleri.

3.1.4. Sıralı Programlama Yaklaşımı

Çalışmakta olan bir otomasyon sisteminde olaylar önceden belirlenmiş bir sırayla gerçekleşir. Örneğin Şekil 3-7'de verilen 1. sıralamada; önce piston iner, ardından pistonun ucuna bağlı matkap 10 saniye boyunca çalışır, matkap durur ve son olarak da piston kalkar. Otomasyon sürecinde yürütülecek görevlerin niteliğine göre, işlem sıralaması farklı tanımlanabilir. Böylelikle süreç sonunda farklı çıktılar elde edilebilir.

Örneğin ilkinden farklı olarak 2. sıralamada; önce matkap çalışır, ardından piston iner, 10 saniye bekler, piston kalkar ve son olarak da matkap durur. Bu sıralama ilkinden farklı olsa da neticede benzer bir görevi yerine getirir. Fakat sadece sıralamanın değiştirilmesi nihayetinde, yürütülecek görevin tamamen değişmesi anlamına da gelmeyebilir.



Şekil 3-7. Aynı görevi yapan iki farklı adım sıralama örneği.

Fabrika otomasyonu gibi büyük sistemlerde aynı anda birçok görev birlikte yürütülmektedir. Yürütülen görevlerde işlem adımlarının sıralaması çoğunlukla kritik önem taşır. Bir adım biter diğeri başlar. Otomasyonu oluşturan tüm süreç, PLC ve IPC gibi endüstriyel denetleyiciler içerisine yüklenen programlar ile denetleyici giriş ve çıkışları arasında kurulan mantıksal ilişkilere dönüştürülür. Geleneksel yapısal programlama yöntemleri, böylesine karmaşık sistemlerin süreç algoritmalarının çıkarılabilmesi ve endüstriyel denetleyicilerin programlanabilmesi için yetersiz kalır.

Fransa’da bulunan bir grup bilim adamı (AFCET- Association Francaise pour la Cybernétique Economique et Technique) tarafından karmaşık otomasyon sistemlerinin programlanmasında ve program kodunun sonradan takibinde yardımcı olacak grafiksel bir gösterim şekli tanımlanmıştır. Bu gösterime grafik içeriği nedeniyle “GR” kısaltması ve topluluğun adı olan “AFCET” kelimelerinin birleşimi olan “GRAFCET” adı verilmiştir. Teorik yönü matematiksel binary Petri Net’lere dayanan GRAFCET, 1977 yılında Fransa’da PLC’lerin programlanması işlerinde kullanılan resmi bir gösterim olarak kabul edilmiştir. 1988 yılında GRAFCET üzerinde yapılan ufak değişiklikler sonucunda Sequential Function Chart (SFC) adını almış ve International Electrotechnical Commission (IEC) tarafından da uluslararası bir standart (IEC 848) olarak kabul edilmiştir. 1995 yılında SFC ile PLC’nin direk programlanabilmesinin de önü açan IEC 1131-3 standardı kabul edilmiştir [[10, 15, 80]].

Otomasyon sistemlerinde kullanılan mantıksal denetleyiciler için geliştirilmiş farklı programlama dilleri vardır. Üreticiden üreticiye değişmekle beraber bu dillere merdiven

diyagramı (ladder diagram – LAD), fonksiyonel blok diyagramı (functional block diagram – FBD) ve komut listesi (statement list – STL) örnek olarak verilebilir. Her ne kadar bazı PLC üreticileri direk SFC yaklaşımıyla program yazmaya olanak sağlasa da, SFC aslında başlı başına bir programlama dili değildir. PLC programı yazımında hangi dil veya yöntem kullanılırsa kullanılsın; SFC, oluşturulacak yazılımın kodlarının geliştirilmesini sağlayan bir şematik algoritma gösterim biçimidir.

C gibi yapısal programlama dilleri için tanımlanmış olan algoritma gösterim biçimi PLC programlama için uygun değildir. Bu durum birazda PLC’nin çalışma mantığının farklılığından kaynaklanır. Bilgisayar dahil olmak üzere standart bir işlemci, kendisinde koşturulan programa ait kodları satır satır yürütür. Enerji verildiği sürece sonsuz bir döngü içerisinde [while(1){...}] işlemci kod satırlarında yazılı olan görevleri tek tek işler ve hemen sonucu üretir. Örneğin yukarıda Şekil 3-7 de verilen “Sıralama 1” e ait algoritma için mikro-denetleyici için hazırlanmış C kodu Şekil 3-8 deki gibi yazılabilir.

```
#include <htc.h>
#include "delay.h"

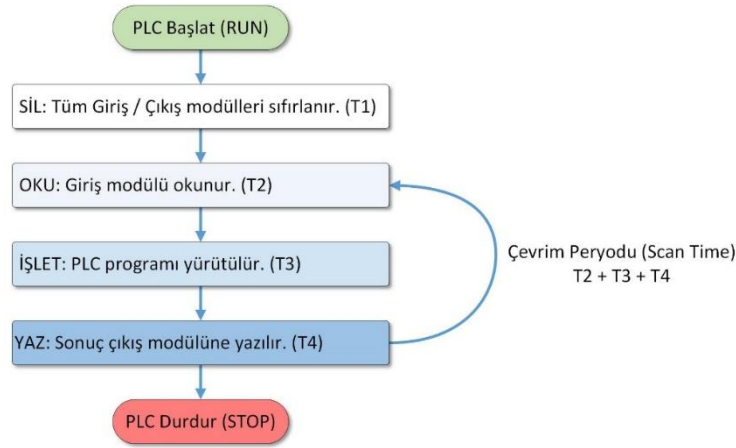
#define PROCESSOR 18F4520
#define XTAL_FREQ 8000000
#define baslatButon PORTDbits.RD0
#define pistonCikis PORTBbits.RB0
#define matkapCikis PORTBbits.RB1

void picHazirla()
{
    TRISB = 0x00;        // PORTB Çıkış olarak ayarlanır.
    PORTB = 0x00;        // PORTB Başlangıç değeri atanır.
    TRISD = 0x01;        // PORTD.0 Giriş, gerisi Çıkış olarak ayarlanır.
    PORTD = 0x00;        // PORTD Başlangıç değeri atanır.
}

void main(void)
{
    picHazirla();
    while(1)
    {
        if(!baslatButon)    // Eğer Başlat butonuna basılmışsa...
        {
            while(!baslatButon); // Başlat butonu bırakılana kadar bekler.
            pistonCikis = 1;    // Piston aşağı iner.
            matkapCikis = 1;    // Matkap çalışmaya başlar.
            delay(15);          // 15 sn bekler.
            matkapCikis = 0;    // Matkap durur.
            pistonCikis = 0;    // Piston yukarı çıkar.
        }
    }
}
```

Şekil 3-8. “Sıralama 1” için mikro-denetleyici C kodu örneği.

PLC’de ise Şekil 3-9’de görüldüğü gibi “RUN” duruma getirildiğinde çalışmaya başlayan çevrimler vardır. Her çevrimde önce girişler okunur, sonra yüklü olan koda göre çıkış bilgisi üretilir ve en son olarak çıkışlar yazılır. PLC yüklü olan kodu her çevrimde bir bütün olarak işler ve sonucu öyle üretir. PLC “STOP” duruma getirilene kadar çevrimler içerisinde tanımlanmış kodlar bu şekilde yürütölmeye devam eder.



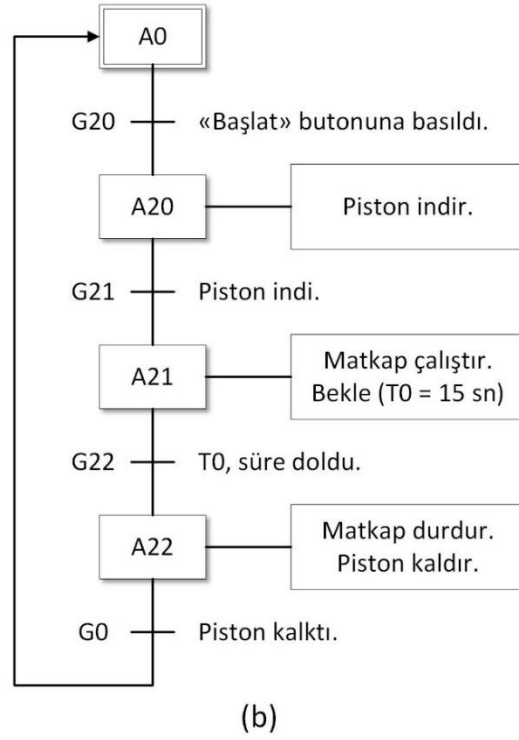
Şekil 3-9. PLC çalışması ve çevrimi yapısı.

SFC gösteriminin geliştirilmesinin temelinde de, PLC ile aynı standart işlemcide olduğu gibi satır satır işlem yapılmasının istenmesi yatmaktadır. Bu nedenle SFC yaklaşımında bir satırdaki işlem sonuçlanmadan diğer satıra geçiş engellenir. Satırda tanımlanan kodun işlenmesi bitene kadar PLC işlemcisi beklemek durumdadır. Böylesi bir çalışma şekli yapısal programlamada kullanılan algoritma mantığıyla şema haline getirilmek istenirse, her adım sonunda bir karar sembolü kullanıp, adımın sonlanıp sonlanmadığının bir döngü içerisinde sürekli sorgulanması gerekmektedir. Böylesi bir gösterim şekli algoritma şemasını anlaşılması güç olan karmaşık bir hale getirir. SFC’nin asıl geliştirilme ihtiyacı da buradan doğmuştur.

Şekil 3-10 de daha önce verilmiş olan piston ve matkap ikilisine ait örnek için hazırlanmış akış şemaları görölmektedir. Akış şemalarından biri (a) standart yapısal programlama gösterimine göre hazırlanmıştır. Bir adımdan diğer adıma geçişte karar döngüsü içerisinde adımın sonlanıp sonlanmadığı sürekli kontrol edilmektedir. Şekilden de anlaşıldığı üzere bu gösterim şekli hem uzun hem de çok fazla karar döngüsü içermektedir. Bu nedenle (a) gösteriminin sıralı programlamada kullanımı uygun değildir.

Diğer akış şeması Şekil 3-10 (b) ise SFC gösterimi ile hazırlanmıştır. Her bir adımda yürütölecek görevler belirlenmiş ve yan taraftaki kutular içerisine yazılmıştır. Bir adımın

sonlanıp bir sonraki adımın başlamasının kararını ise geçiş elemanları ve koşulları belirlemiştir. Bunlar da adımları temsil eden kutular arasına yazılmıştır. Bu nedenle (b) gösterimi sıralı programlama yönteminin ifadesi için daha uygundur.



Şekil 3-10. Aynı süreç için hazırlanmış standart programlama akış şeması (a) ile SFC gösterimi (b) karşılaştırması.

3.1.4.1. Sıralı Durum Grafiği Yapısı

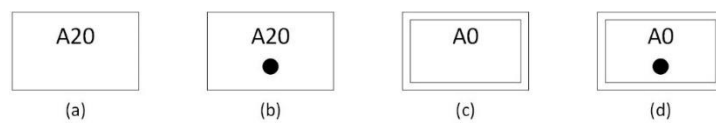
Gerek uluslararası gerekse ülkemiz kaynaklı literatürde yaptığımız araştırmalar sonucunda kağıt üzerinde SFC gösteriminin standart bir biçimi olmadığı anlaşılmıştır [MEGEP, Üsküdar inovasyon merkezi, Siemens SFC, Grafcet]. Gösterimin temelleri IEC 1131-3'te tanımlandığı gibi olsa da, farklı firmaların farklı SFC gösterim yöntemleri geliştirdikleri görülmüştür. Bu nedenle, bu bölümde anlatılan SFC gösterim biçimi, birçok kaynaktan elde edilen veriler ışığında, eğitim ve planlama amaçlı kağıt üzerinde gösterim için tarafımızdan belirlenen uygun bir yöntemdir. SFC kısaltmasının hem anlam hem de içerik bakımından en uygun Türkçe karşılığının Sıralı Durum Grafiği (SDG) olduğu kanaatine varılmıştır. Bundan sonra kısaca SDG olarak adlandırılacaktır.

SDG yaklaşımında bir otomasyon sisteminde yürütülecek görevlerin en küçük adımlara bölünerek adımların tek tek yürütülmesi mantığı vardır. Bu durum karışıklığı önlediği gibi, hazırlanan kodun daha rahat okunabilmesini, ileri de kod üzerindeki değişikliklerin daha kolay yapılabilmesini ve kodun başkalarına anlatılabilmesini sağlar.

Sıralı programlama yaklaşımı temel olarak “adımlar” (step) ve “geçişler” den (transition) oluşur. Her bir adımda önceden belirlenmiş görevler yürütülür. Bir adımın bitip diğer bir adımın başlamasının kararı ise geçişlerle tanımlanır. Sırası geldiğinde yürütülecek kontak çıkışları, sayıcı, zamanlayıcı, vb. işlemler adımlar içerisinde tanımlanır. SDG gösteriminde kullanılan elemanlar ve özellikleri aşağıdaki gibidir:

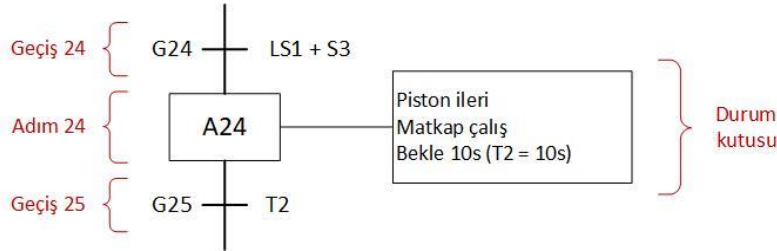
1) Adım: Adımlar Şekil 3-11 (a) ve (b) deki gibi dikdörtgen kutu ile ifade edilir. Kutu içerisine “A” harfiyle başlayan adım numarası yazılır. SDG gösterimlerinde adım numaraları tektir ve yalnızca bir kere kullanılır.

SDG en az bir başlangıç adımıyla başlar. Başlangıç adımı sıralı programın çalışması için gerekli değişken atamaları (zamanlayıcı, sayıcı, karşılaştırıcı, vb.) ve hesaplamalar gibi ön işlemler gerçekleştirilir. Başlangıç adımı Şekil 3-11 (c) ve (d) deki gibi iç içe iki dikdörtgen kutu ile ifade edilir. Gösterimde aktif olan adımlar ise kutu içerisine nokta koyularak belirtilir.



Şekil 3-11. SDG yapısında basit adım (a), aktif adım (b), başlangıç adımı (c) ve aktif başlangıç adımı (d) gösterimi.

Adımlarda yürütülecek görevler ise adımın sağ tarafına eklenen durum kutusu içinde tanımlanır. Kağıt üzerindeki gösterimde sadece adım içerisinde yürütülecek işlemleri açıklayan cümleler yazılması yeterlidir. Örneğin Şekil 3-12 da görüldüğü gibi “piston ileri”, “matkap çalış” ve “Bekle 10s (T2 = 10s)” şeklinde olabilir. Durum kutusu içinde tanımlanan görevler ait oldukları adımda ve aynı anda yürütülür.

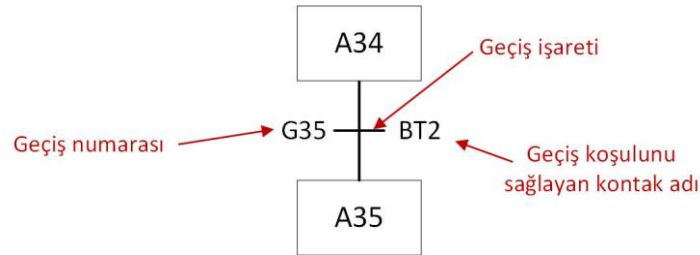


Şekil 3-12. SDG’de her adımda yürütülecek görevlerin gösterimi.

2) Geçiş: SDG’nin bir diğer önemli elemanı ise geçişlerdir. Her bir adım geçişinde, önceki adımın bitiş ve sonraki adımın başlangıcını belirleyen geçiş koşulları tanımlanır. Geçiş koşulları sınır anahtarı, sayıcı, zamanlayıcı ve karşılaştırmacı gibi bit tipi çıkış veren değişkenler ile karşılaştırma komutları çıkışlarına ait kontakların mantıksal kapılar (VE, VEYA, vb.) ile bağlanmasıyla oluşturulur. Koşul sonucu mantıksal 1 olursa, bir sonraki adıma geçiş gerçekleşir. Bir önceki adım ise sonlandırılır.

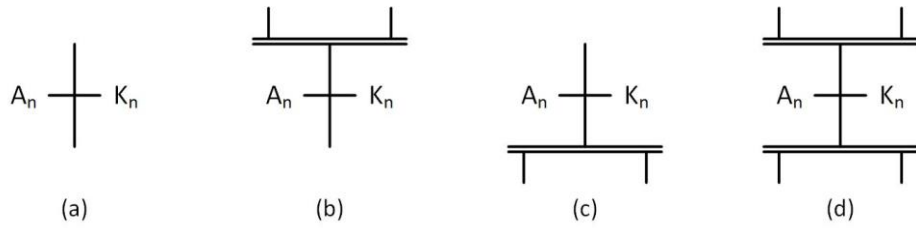
Geçişler SDG gösteriminde Şekil 3-13 deki gibi adıma bağlı dik olarak çizilen çizgiye eklenmiş yatay bir çizgi ile ifade edilir. Bu çizginin sol tarafına “G” harfiyle başlayarak geçiş numarası ve sağ tarafına ise geçişin aktif olmasını sağlayacak koşul ifadesi yazılır. Geçiş numarası kendinden sonra gelen adımın numarasıyla aynıdır.

Geçiş koşulunda yer alan ifadeler arasındaki mantıksal ilişki VE ise nokta (.), VEYA ise artı (+) işaretleri ile gösterilir. Örneğin Şekil 3-12’deki Geçiş 24 (G24)’e ait koşul ifadesi “LS1+S3” şeklindedir. Bu LS1 veya S3 anlamına gelmektedir.



Şekil 3-13. SDG’de geçiş elemanları ve koşullarının gösterimi.

Geçiş elemanı ardışık iki adımı birleştirmede kullanıldığı gibi, birden fazla dalın birleşiminde veya ayrımında da kullanılabilir. Şekil 3-14 deki gösterimde iki adım arası geçişi “basit geçiş” elemanı (a); birden fazla paralel dalın birleşerek tek bir dal üzerinden geçişlerini “birleşim geçişi” elemanı (b); tek bir dal üzerinden gelen akışın birden fazla paralel akıma ayrılmasını “ayrım geçişi” elemanı (c); birden fazla paralel daldan gelen akışın tek bir geçişte birleşip tekrar birden fazla paralel dala ayrılmasını ise “birleşim ve ayrım geçişi” elemanı sağlar.



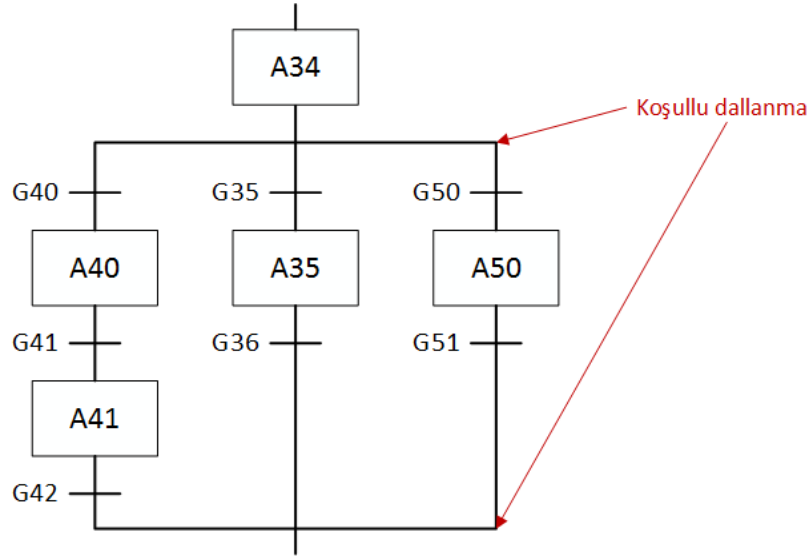
Şekil 3-14. Basit geçiş (a), birleşim geçişi (b), ayrım geçişi (c) ile birleşim ve ayrım geçişi (d)

Çoklu geçişler paralel dallar arasında gerçekleştiği için hepsinde VE kapısı mantığı vardır. (b) ve (d) de olduğu gibi paralel dallardan gelen tüm akışlara ait girişler geçişe ulaşmadıkça, geçiş koşulu tetiklenmez ve geçiş elemanı sonraki adımları tetikleyecek çıkışı üretmez. (c) ve (d) de olduğu gibi paralel dallara ayrılan geçişlerde ise, geçiş elemanı koşul sağlandığında aynı anda birden çok adımı tetikleyecek birden çok çıkışı aktif eder.

4) Koşullu Dallanma: SDG gösteriminde tanımlanan işlem sıralamasında, belirli koşullar gerçekleştiğinde farklı adımların tetiklenmesi gerekiyorsa koşullu dallanma elemanı kullanılır. Koşullu dallanma, adımdan sonra eklenen tek hatlı yatay çizgi ile ifade edilir. Koşullu dallanma elemanından sonra gelen geçiş elemanlarından hangisi tetiklenirse, sadece tetiklenen daldaki işlem adımları yürütülür.

Şekil 3-15’deki örnek gösterimde süreç otomasyonunda 34 numaralı adım gerçekleştikten sonra eğer G40 geçişi tetiklenirse A40 adımı, eğer G34 geçişi tetiklenirse A35 adımı, eğer G50 geçişi tetiklenirse A50 adımı yürütülecektir.

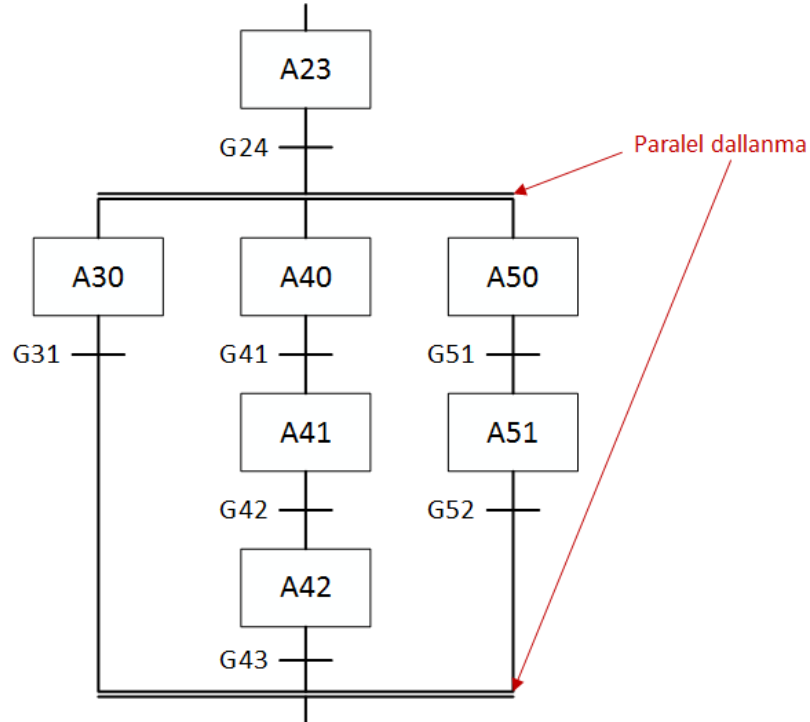
Eğer Şekil 3-15 deki SDG gösteriminde yer alan işlemi standart yapısal programlamaya ait algoritma yöntemiyle gösterilmek istenseydi, gösterimde 3 farklı karar elemanı kullanılması gerekirdi. Bu da gösterimin hazırlanması ve anlaşılması için daha fazla çaba harcanmasını gerektirirdi.



Şekil 3-15. Koşullu dallanma gösterimi.

5) Paralel Dallanma: Eğer ki birbirine paralel olarak sırayla yürütülmek istenen adımlar varsa paralel dallanma kullanılır. Geçiş elemanından sonra eklenen çift hatlı yatay çizgiler ile ifade edilir.

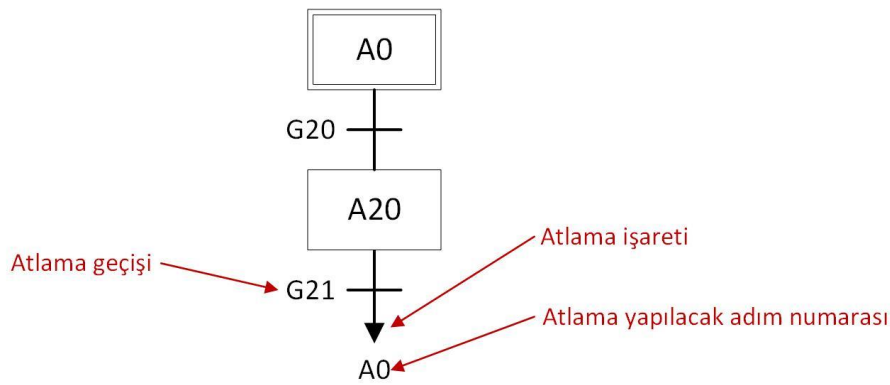
Şekil 3-16'daki örnekte 23 numaralı adım gerçekleştikten sonra G24 geçişi tetiklenirse; A30, A40 ve A50 adımları aynı anda aktif edilir. Böylelikle 3 farklı paralel akış başlatılmış olur.



Şekil 3-16. Paralel dallanma gösterimi.

6) Atlama: Kendisine ait olan geiř elemanı tetiklendiğinde iřlem akıřının istenilen adıma atlamasını saęlar. Yönlendirilmiş baęlantı çizgilerinin kullanımına karřılık SDG gösteriminin karıřıklıęının azaltılmasını saęladığı için tercih edilir. Ařaęı doęru bir ok ile gösterilir. Atlama yapılmak istenen adımın numarası okun altına yazılır. SDG gösteriminde her bir adıma ayrı numara verildięi ve mükerrer adım numarası bulunmadığı için karıřıklık olmaz. SDG gösteriminde her bir atlama sadece bir kez kullanılabilir.

řekil 3-17 deki örnek gösterimde 20 numaralı adım gerekleřtikten sonra G14 geiři tetiklenirse A0 adımı aktif edilir.



řekil 3-17. Atlama gösterimi.

3.1.4.2. SDG Gösterimi Kuralları

Her ne kadar SDG gösterimi temelleri IEC 1131-3 standardına baęlı olsa da; uygulamada farklı firmaların kullandığı farklı gösterim yöntemleri ve kuralları bulunmaktadır. Yaptığımız arařtırmalar neticesinde belirlediğimiz eęitim amalı SDG gösterim kuralları řunlardır:

- SDG gösterimi mümkün oldukça yalın ve anlaşılır olmalıdır.
- SDG gösteriminde PLC içindeki gerek kontak adresleri ve ıkıřlar yazılmamalıdır. Bunlar yerine eřdeř isimler (alias) kullanılmalıdır.
- SDG gösterimi yukarıdan ařaęı doęru olacak řekilde hazırlanır ve yorumlanır. İlk adım en tepede olacak řekilde sıralanır.
- Bir SDF’de en az bir adet bařlangı adımı olmalıdır. İçeriğinde hiçbir iřlem yapılmasa bile bařlangı adımı gösterimi zorunludur.
- SDG içerisindeki adımlara belirli bir numara verilir ve her bir adım sadece bir kez kullanılır.
- Adımlara ait isimler “A” harfi ile bařlar ve ařaęıdaki gibi isimlendirme yapılır:

- A0-A9 arası: Başlangıç adımlarına verilen isimdir. İlk başlangıç adımı “A0” isimlendirilir.
- A10-A19 arası: Özel durumlar için ayrılmıştır. Normal şartlarda kullanılmaz.
- A20-A899 arası: Program içinde yer alan ve normal işlemleri yürüten adımlara verilen isimdir.
- Bir PLC programı bütünüyle SDG yaklaşımıyla oluşturulmuş olmayabilir. PLC programı içerisinde SDG gösterimi ayrı bloklar içinde yazılabilir. Mantıksal olarak yazılan program kodları içerisinde istenilen bölümde SDG bloklarına geçiş yapılabilir.
- Bir PLC programında SDG yöntemi ile oluşturulmuş 10 adede kadar ayrı program bloğu kullanılabilir.
- Bir SDG en fazla 16 sütundan (paralel veya koşullu dallanma) oluşabilir.
- PLC programı oluşturulurken ilk önce SET/RESET elemanları kullanılarak adımlar ve geçişler tanımlanır. En son olarak adımlar içerisinde a bağlı gerçek motor, kontaktör gibi gerçek sistemleri çalıştıracak çıkışlar aktif edilir.

3.1.4.3. SDG Yaklaşımı ile Programlama

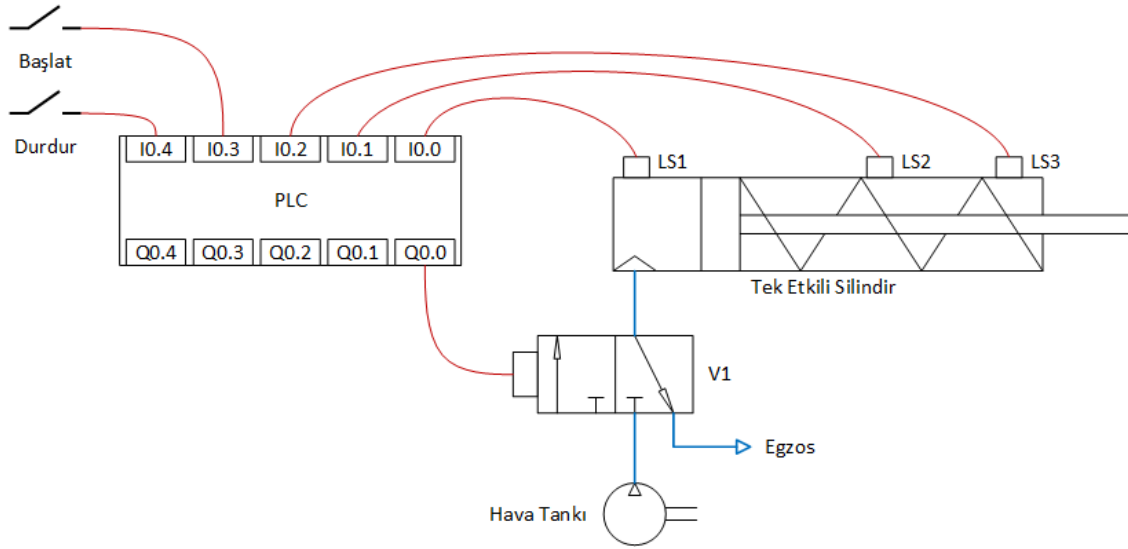
SDG yaklaşımını kullanarak iki farklı yöntemle süreç otomasyonu programlanabilir. Bu yöntemlerden biri SDG’yi sadece program algoritmasını “kağıt” üzerinde oluşturmak ve planlama için kullanmaktır. Bu yöntemde algoritma belirlendikten sonra PLC programı merdiven diyagramı (LAD), komut listesi (STL) vb. diller kullanılarak geliştirilebilir.

Bazı PLC üreticileri direk SDG ile programlamaya izin vermektedir. Örneğin S7-300 PLC’leri programlamak için kullanılan “Simatik Manager” yazılımında “Graph Editor” isimli araç vardır. Aynı üreticiye ait yeni nesil programlama aracı olan “TIA Portal” yazılımında ise “Graph 7” aracı vardır. Yine yaygın kullanılan FX serisi PLC’lerin programlanmasında kullanılan “Melsec MEDOC” yazılımında “SFC View” aracı vardır. Bu araçlar kullanılarak SDG blokları oluşturulur. Bloklar içerisine ise adımlar ve geçiş koşulları için gerekli kodlar yazılır. Bu şekilsel programlama yöntemi nispeten diğer yöntemlere göre hem daha kolay hem de daha anlaşılırdır.

3.1.4.3.1. SDG Gösterimi ile Sadece Algoritma Çıkarma Yöntemi

Bu yöntemde ilk önce PLC ile yapılmak istenen işi oluşturan adımlar detaylı bir şekilde tek tek belirlenir. Belirlenen adımlar, adımlar arası geçiş koşulları ve akış şeması SDG yaklaşımıyla gösterildikten sonra PLC kodları yazılmaya başlanır.

Örnek olarak Şekil 3-18 de bağlantı şeması gösterilmekte olan bir pnömatik silindir ve PLC ile gerçekleştirilmek istenen görevi ele alalım. Silindir üzerinde çeşitli konumlara yerleştirilmiş “LS1”, “LS2” ve “LS3” olmak üzere toplam 3 adet manyetik sınır anahtarı bulunmaktadır. “Başlat” butonuna basılmasıyla birlikte “LS1” de bulunan piston hareket etmeye başlar. Piston “LS3” e vardığı zaman tekrar “LS1” e geri döner. 15 sn. bekledikten sonra piston bu sefer “LS2” ye kadar hareket eder ve vardığında geri dönerek “LS1” e vardığında görev tamamlanır.

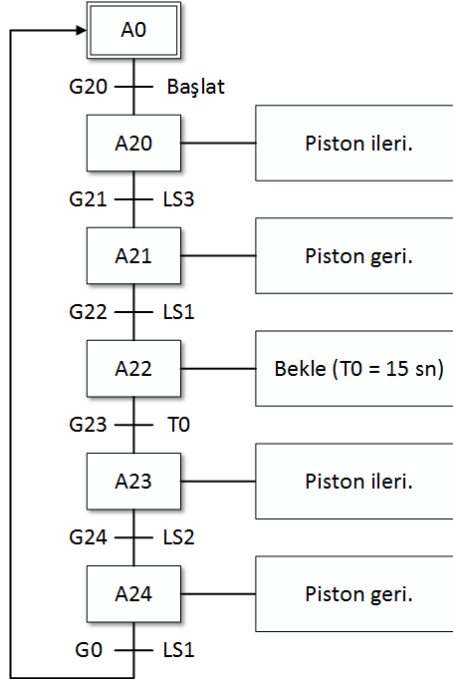


Şekil 3-18. Örnek uygulamaya ait elektro-pnömatik bağlantı şeması.

Böylesi bir göreve ait SDG şemasının hazırlanabilmesi için öncelikle göreve ait alt adımların belirlenmesi gereklidir. Bu örnekteki adımlar şu şekilde tanımlanabilir:

- Adım 1: LS1’den LS3’e ileri yönde hareket etme,
- Adım 2: LS3’den LS1’e geri yönde hareket etme,
- Adım 3: 15 sn. bekleme,
- Adım 4: LS1’den LS2’ye ileri yönde hareket etme,
- Adım 5: LS2’den LS1’e geri yönde hareket etme.

Yukarıda verilen tanımlamalara göre Şekil 3-18 deki örnek uygulama için hazırlanan SDG gösterimi Şekil 3-19 de görüldüğü gibidir:



Şekil 3-19. Örnek uygulamaya ait SDG gösterimi.

Örnek uygulamaya ait akış şeması oluşturulduktan sonra PLC programı yazılmaya başlanır. Bu kısımda PLC kodları S7-300 PLC için yazılmış olup açıklayıcı olması bakımından ilişkili bölümler halinde aktarılacaktır.

SDG’de başlangıç adımı, genellikle PLC “RUN” duruma getirildiğinde sadece ilk çevriminde çıkış veren, sonrasında pasif duruma geçen kontak ile tetiklenir. “İlk vuruş” olarak da isimlendirilen bu kontak adresi ve kullanım özellikleri, PLC üreticisi markaya göre değişiklik gösterir. Örneğin S7-200 PLC’de ilk vuruş kontak adresi “SM0.1” iken, FX-2N PLC’de “M8002”, MCT-20 PLC’de “M1924”tür. S7-300 PLC’lerde ise OB100 bloğu içerisine eklenecek bir satırlık kod ile boşta olan herhangi bir kontak adresinden “ilk vuruş” sinyali alınabilir.

Bu örnek uygulamada S7-300 PLC kullanıldığı ve tüm yazılım SDG yaklaşımıyla hazırlandığı için, başlangıç adımının tetiklenmesinde “ilk vuruş kontağı” kullanılmıştır. S7-300 PLC’lerde OB100, ilk vuruşta gerçekleştirilecek işlemlerin tanımlandığı kod bloğudur. Sadece PLC “RUN” duruma getirildiğindeki “ilk vuruşta” aktif olur ve içerisinde yazılan kodlar sadece bir kez çalıştırılır. PLC’nin enerjisi kesilip tekrar verildiğinde bile aktif olmaz.

Örnek uygulama için OB100 içerisinde hazırlanmış kod Şekil 3-20 de görüldüğü gibidir. Buna göre OB100 aktif olduğunda, diğer bir ifadeyle PLC “RUN” duruma getirildiğinde;

eğer ki “Stop” butonu basılı değilse, ilk vuruş için seçilmiş olan M20.0 kontağı aktif olur. İstenirse program içerisinde önceden kullanılmamış olan başka bir kontak adresi de “ilk vuruş” kontağı olarak seçilebilir. Burada kullanılan M20.0’ın özel bir anlamı yoktur.

OB100 : "Program Baslangic Tetikleme"
Network 1: İlk Vurus Kontagi Devresi



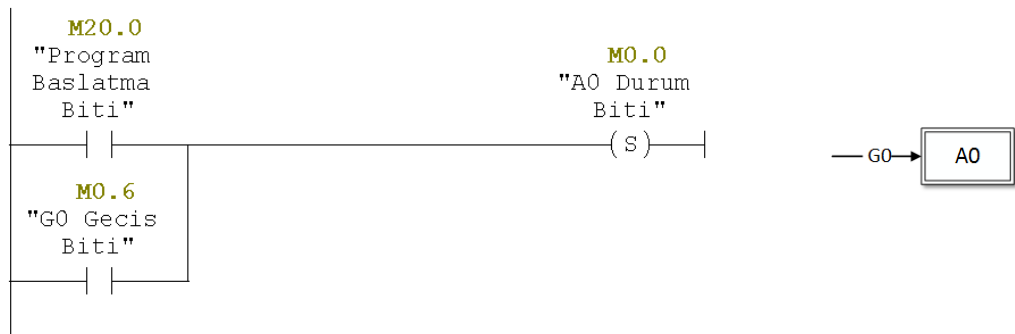
Şekil 3-20. İlk vuruş kontağı için OB100 bloğuna ait PLC kodu.

Bütünüyle SDG yaklaşımıyla oluşturulmamış PLC programlarında “Program Baslama Biti” (M20.0), normal merdiven diyagramı içerisinde tetiklenecek herhangi bir kontak da olabilir.

Örnek uygulamamız için hazırlanan PLC programını başlatacak ilk vuruş kontağı ile ilgili kod bölümü hazırlandıktan sonra, asıl sıralı otomasyonu sağlayacak bölüm ile ilgili kodların hazırlanmasına geçilebilir. Bu bölüm e ait kodlar OB1 bloğu içerisine yazılır.

Şekil 3-21 de görülmekte olan “Network 1” kod satırında, “Program Başlama Biti” ve “G0 Geçiş Biti”, 0. adımın geçiş koşullarını temsil etmektedir. Buna göre “Program Başlama Biti” veya “G0 Geçiş Biti” aktif olduğunda, “A0 Durum Biti” de aktif olacaktır. Bu andan itibaren PLC programında, 0. adımda yapılmak istenen tüm işlem adımlarına ait geçişlerde, koşul olarak “A0 Durum Biti” eklenecektir.

OB1 : "Ornek 1: Piston Kontrol Otomasyonu"
Network 1: A0 Baslangic Adimi



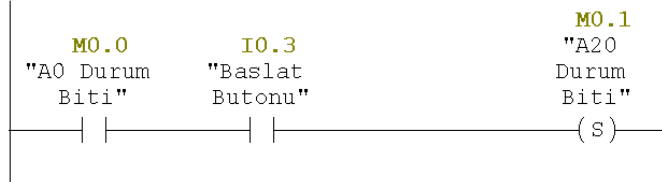
(a)

(b)

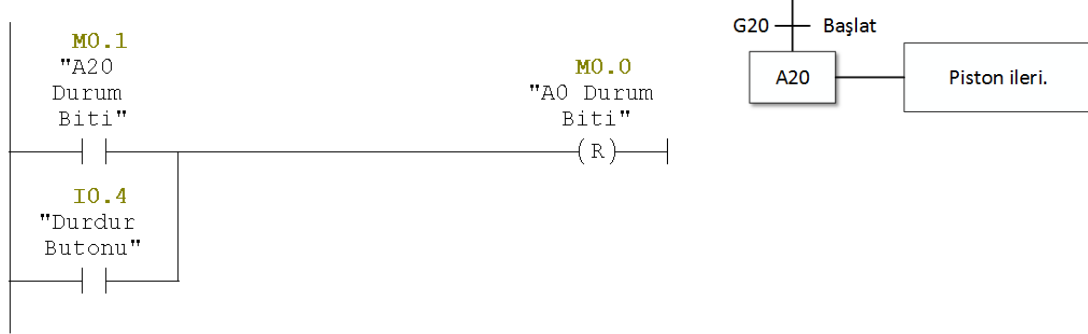
Şekil 3-21. 0 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b).

Şekil 3-22 deki “Network 2” kod satırında, “A0 Durum Biti” aktifken “Başlat” butonuna basılırsa; 20 numaralı adımın geçiş koşulları oluşur ve 20. adım durum biti aktif olur. Dikkat edilirse “A0 Durum Biti” aktif olmadığı sürece “Başlat” butonuna basılması, 20 numaralı adımın tetiklenmesi için yeterli olmayacaktır.

Network 2 : G20 Gecis Kosulu Tanimi



Network 3 : A20 Islemleri Tanimi



(a)

(b)

Şekil 3-22. 20 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b).

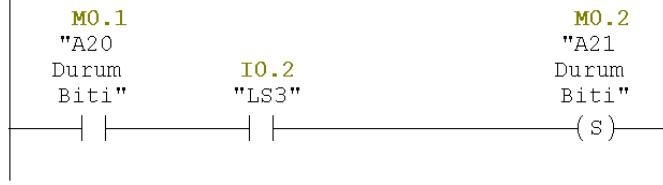
SDG'nin temel kurallarından biri, bir adım aktif edilirken bir önceki adımın pasif hale getirilmesidir. Şekil 3-22'deki PLC kod bloğunda 0. adım “Network 2” de değil de “Network 3” de pasif duruma getirilmektedir. Bunun sebebi 0. adım durum biti hem aktif olup hem de kendini reset edemeyecektir. Aksi durumda sinyal çakışması olacaktır.

“A20 Durum Biti” aktif olduğu anda örnekte istendiği üzere piston ileri doğru hareket eder. Bunun gibi çıkış işlemlerine ait kodlar, Şekil 3-28'deki gibi PLC kod bloğunun sonlarında verilmektedir. Böylelikle adım sıralaması etkilenmeden çıkışlar ile adımların eşleşmesi kolaylıkla yapılabilir.

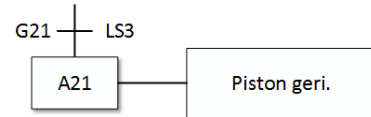
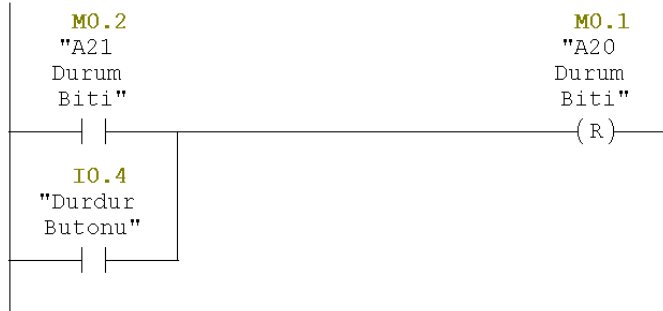
Piston “LS3” sınır anahtarına ulaştığında “LS3”ün bağlı bulunduğu PLC girişi aktif olur. Bu durum 21. adımın geçiş koşuludur. Şekil 3-23'de görüldüğü üzere “Network 4” te “A21 Durum Biti” aktif edilir. Bu durumda piston geriye doğru hareketine başlar. “Network 5”te ise “A20 Durum Biti” resetlenerek 20 adım pasif duruma getirilir. Eğer 21 numaralı adımda piston hareketi dışında sayıcı, zamanlayıcı kullanımı gibi

kullanılmak istenen ek fonksiyonlar varsa, bunlar da “Network 5” içerisinde tanımlanmalıdır.

Network 4 : G21 Gecis Kosulu Tanimi



Network 5 : A21 Islemleri Tanimi



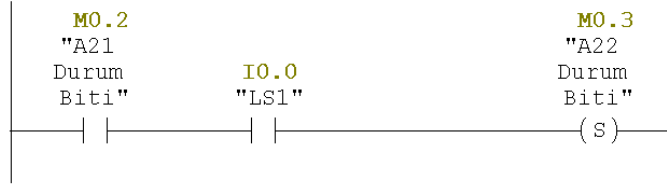
(a)

(b)

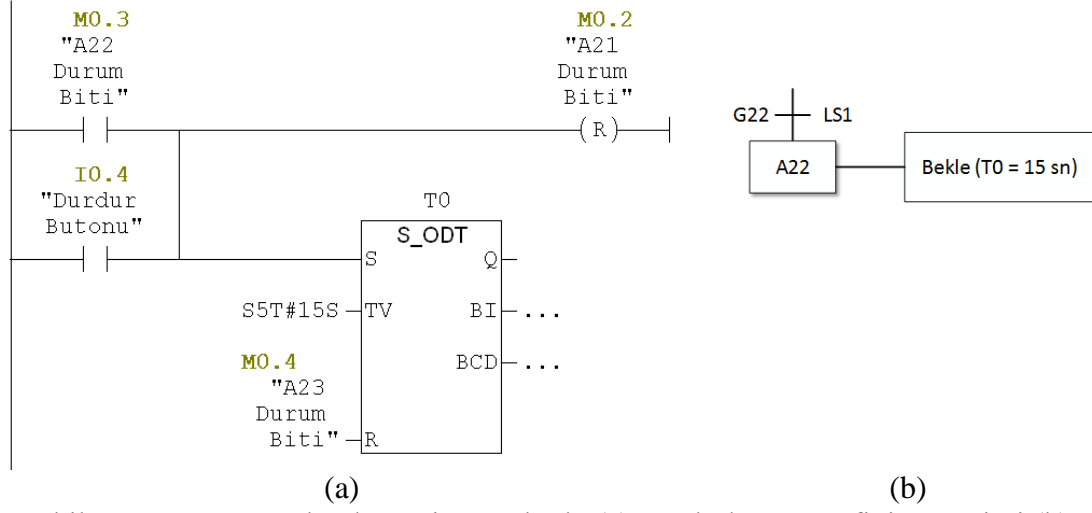
Şekil 3-23. 21 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b).

Piston “LS1”e ulaştığında geriye doğru hareketini tamamlamış demektir. Bu nedenle bir sonraki adım A22’nin geçiş koşulu gerçekleşmiştir. Şekil 3-24’de görüldüğü üzere “Network 6”da “A22 Durum Biti” aktif edilir. “Network 7”de ise “A21 Durum Biti” resetlenerek 21 adım pasif duruma getirilir. Aynı zamanda 15 sn. bekleme yapılmasını sağlayan T0 zamanlayıcısı da çalışmaya başlar.

Network 6: G22 Gecis Kosulu Tanimi



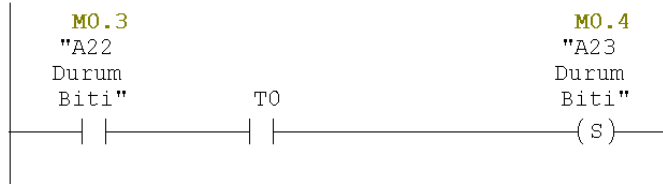
Network 7: A22 Islemleri Tanimi



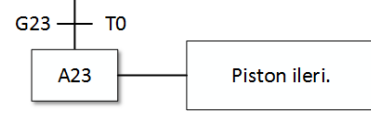
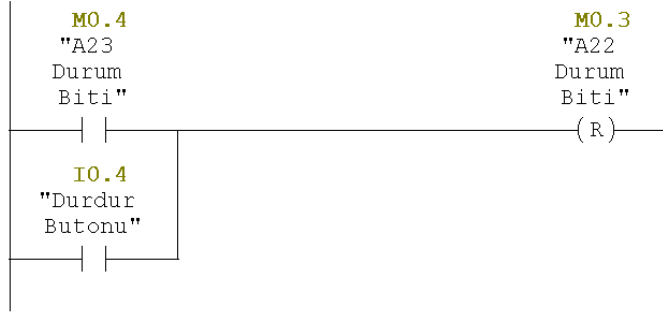
Şekil 3-24. 22 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b).

T0 zamanlayıcısının süresi dolduğunda 22. adım tamamlanmış ve 23. adım geçiş koşulu oluşmuştur. T0 kontağının aktif olmasıyla birlikte A23'ün geçiş koşulu gerçekleşmiştir. Şekil 3-25'de görüldüğü üzere "Network 8'de "A23 Durum Biti" aktif edilir. Bu durumda piston tekrar ileri doğru hareketine başlar. "Network 9'da ise "A22 Durum Biti" resetlenerek 22 adım pasif duruma getirilir.

Network 8 : G23 Gecis Kosulu Tanimi



Network 9 : A23 Islemleri Tanimi



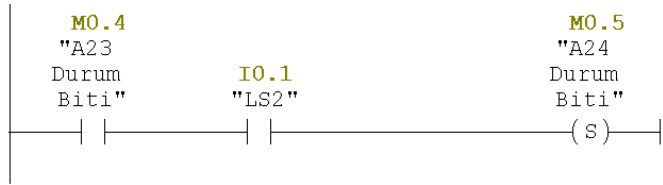
(a)

(b)

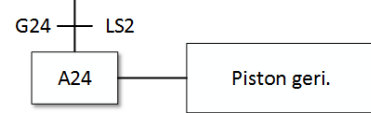
Şekil 3-25. 23 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b).

Piston “LS2”ye ulaştığında ileriye doğru hareketini tamamlamış demektir. Bu nedenle bir sonraki adım A24’ün geçiş koşulu gerçekleşmiştir. Şekil 3-26’de görüldüğü üzere “Network 10”da “A24 Durum Biti” aktif edilir. Bu durumda piston tekrar geri doğru hareketine başlar. “Network 11”de ise “A23 Durum Biti” resetlenerek 23. adım pasif duruma getirilir.

Network 10 : G24 Gecis Kosulu Tanimi



Network 11 : A24 Islemleri Tanimi



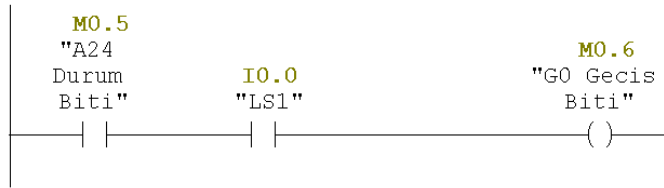
(a)

(b)

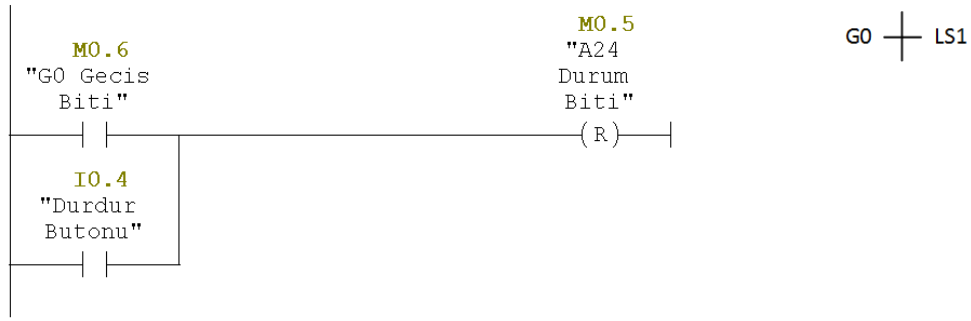
Şekil 3-26. 24 numaralı adıma ait PLC kodu (a), sıralı durum grafiği gösterimi (b).

Piston “LS1”e ulaştığında geriye doğru hareketini tamamlamış demektir. Bu nedenle G0’ın geçiş koşulu gerçekleşmiştir. Şekil 3-27’de görüldüğü üzere “Network 12”de “G0 Geçiş Biti”ne anlık çıkış verilir. Bu çıkış Şekil 3-21’teki “Network 1” de görülmekte olan “A0 Durum Biti”ni tetikler. Böylelikle program ilk adıma dönmüş olur. “Network 13”de ise “A24 Durum Biti” nin resetlenerek 23. adım pasif duruma getirilir. Bu durumda da “G0 Gecis Biti” pasif duruma geçerek sadece 1 çevrimlik çıkış üretilmiş olur.

Network 12 : G0 Gecis Kosulu Tanimi



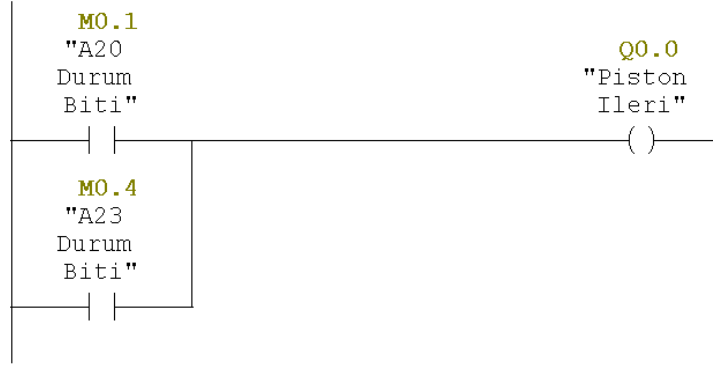
Network 13 : G0 Gecis Kosulu Tanimi Devami



Şekil 3-27. 0 numaralı geçişe ait PLC kodu (a), sıralı durum grafiği gösterimi (b).

Örnek 1’de verilmiş olan sistemde PLC’ye bağlı sadece tek bir çıkış kullanılmaktadır. O da aktif olduğunda pistonun ileri doğru hareketini sağlayan pnömatik vanayı tetikler. Örnekteki piston ve vana yay geri dönüşlü olduğu için PLC çıkışı dolayısıyla da vana pasif hale geldiğinde piston geriye doğru hareket eder. Şekil 3-28’de görülmekte olan “Network 14” de sistemde kullanılan tek çıkış olan “Piston İleri”nin hangi adımlar aktif olduğu sürece çıkış vereceği tanımlanmıştır. Buna göre 20. ve 23. adımlar aktif olduğu sürece piston ileri yönlü hareket eder. Diğer adımlarda “Piston İleri” çıkışı pasif olduğu için ya piston geriye doğru hareket ediyordur ya da geridedir.

Network 14 : Gercek Sisteme Cikis Islemleri



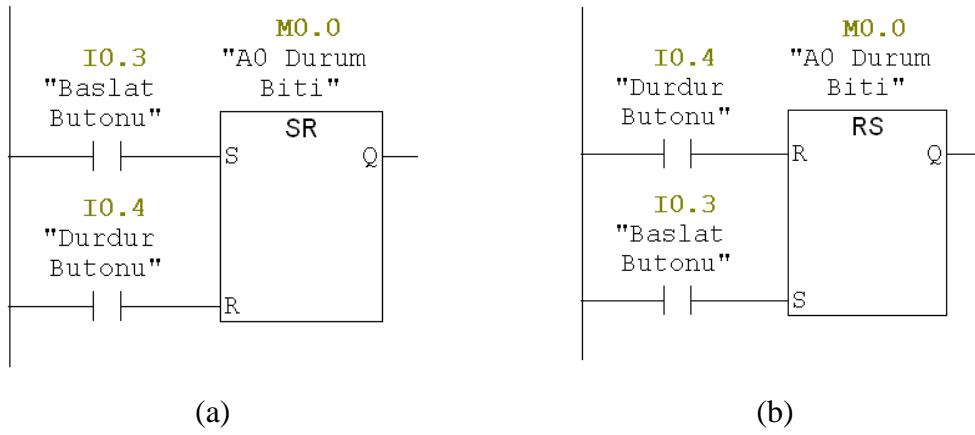
Şekil 3-28. Gerçek sisteme çıkış işlemlerine ait PLC kodu.

Eğer ki ileride aynı senaryo içerisinde farklı adımlarda farklı çıkışlar da verilmek istenirse, hangi adımda hangi çıkışın aktif edileceğine dair tanımlamalar bu bölümde yapılır. SDG adımlarının PLC programının baş tarafında, gerçek çıkış koşullarının ise en sonda tanımlanması; daha anlaşılır ve müdahale edilebilir olmasını sağlar. Böylelikle ileride oluşacak ihtiyaçlara göre kodun daha kolay değiştirilebilmesini sağlar.

Yukarıda Örnek 1 için sunulan kodlar, tüm marka PLC’lerde standart olarak kullanılan set ve reset elemanları ile hazırlanmıştır. Böylelikle örnek SDG uygulamasının istendiğinde S7-300’den bağımsız olarak da PLC koduna dönüştürülebilmesi amaçlanmıştır. Yukarıda Örnek 1 için sunulmuş olan PLC programı, PLC hazır fonksiyonlarından olan SR (set-reset) veya RS (reset-set) elemanları kullanılarak da oluşturulabilir. Bu durumda Örnek 1’e ait program kodları daha da kısılacaktır.

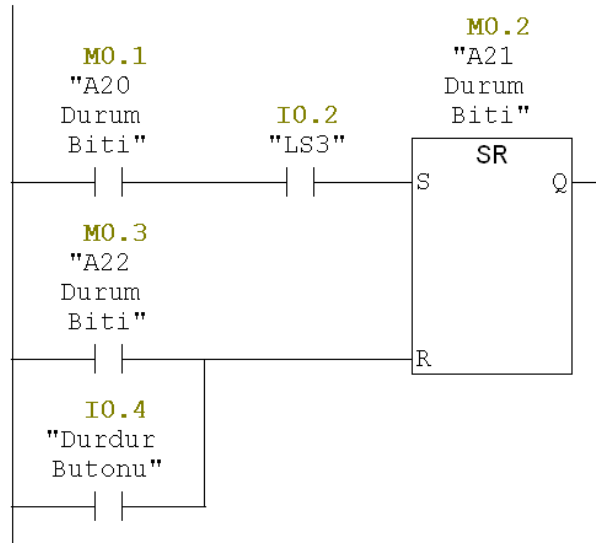
SR (set-reset) veya RS (reset-set) elemanları herhangi bir çıkışı set ve reset edecek koşulları kolaylıkla tanımlamayı sağlar. SR ve RS aynıymış gibi görünse de çalışma şekilleri farklıdır. SR elemanında reset işlemi önceliklidir. Çıkışı set veya reset edecek iki koşul (kontak girişi vb.) aynı anda aktif olduğunda, sinyal çakışmasını önlemek için reset işlemi gerçekleştirilir. Şekil 3-29 (a) da verilmiş olan örnekte görüldüğü üzere “Başlat Butonu” ve “Durdur Butonu”na aynı anda basılırsa “A0 Durum Biti” reset olacaktır.

RS elemanında ise set işlemi önceliklidir. Çıkışı set ve reset edecek iki koşul aynı anda aktif olduğunda, sinyal çakışmasını önlemek için set işlemi gerçekleştirilir. Şekil 3-29 (b) de verilmiş olan örnekte görüldüğü üzere “Başlat Butonu” ve “Durdur Butonu”na aynı anda basılırsa “A0 Durum Biti” set olacaktır.



Şekil 3-29. SR elemanı kullanım örneği (a), RS elemanı kullanım örneği (b)

Buna göre örnek olması açısından Şekil 3-23 ve Şekil 3-24’de görülmekte olan networkler SR ve RS elemanları kullanılarak Şekil 3-30’deki gibi oluşturulabilir. Buna göre “A20 Durum Biti” ve “LS3” aktif olursa “A21 Durum Biti” aktif (set); “A22 Durum Biti” veya “Durdur Butonu” aktif olursa “A21 Durum Biti” pasif (reset) olur. Dikkat edilirse bu yöntem diğerine göre daha anlaşılır ve kolaydır.



Şekil 3-30. Örnek 1’e ait 4, 5, 6 ve 7. networklerin SR ve RS elemanları ile yazılmış kodları.

3.1.5. AS-i Endüstriyel Haberleşme Protokolü ve Cihazları

AS-i ismi, Eyleyici Algılayıcı Arayüzü (Actuator Sensor Interface) kelimelerinin kısaltılmasından meydana gelmiştir. Otomasyon sistemlerindeki geleneksel paralel tip kablolamaya alternatif olarak geliştirilmiş esnek, düşük maliyetli ve akıllı bir endüstriyel sayısal ağ çözümüdür. Endüstriyel ağlarda bulunan anahtar, on-off buton, algılayıcı, enkoder, analog giriş-çıkış, röle gibi elemanların; basit iki iletkenli bir kablo üzerinden PLC, DCS ve kişisel bilgisayar gibi yönetim aygıtlarına bağlanabilmesi için geliştirilmiştir. Uygulama kolaylığı ve düşük maliyeti sayesinde AS-i; konveyör sistemleri, paketleme makinaları, süreç otomasyon sistemleri, havaalanı bagaj taşıma hatları, asansörler, şişeleme ve yiyecek endüstrisi gibi birçok alandaki otomasyon uygulamasında yer almaktadır.

Tarihçesine bakıldığında, ilk olarak 1980'lerin sonunda endüstriyel temassız algılayıcılar üretmekte olan 11 ayrı firmanın bir araya gelerek oluşturduğu birlik tarafından geliştirilmeye başlanan AS-i, sonraları açık teknoloji uluslararası bir standart (EN 50 295) halini almıştır. Protokolün tarihsel gelişimi sırasında farklı sürümler üretilmiş olup, sürüm özellikleri özetle aşağıda ve Tablo 3-1'deki gibidir:

- Sürüm 2.04 (1994): AS-i ile ilgili ilk prototip uygulama 1994 Hannover fuarında tanıtılmıştır. Standart ilk olarak tanımlandığında, bir AS-i ağı 31 adet sayısal (on/off) giriş/çıkış cihazı destekleyecek kapasitedeydi. Her bir cihazın 4 bit'lik giriş ve 4 bit'lik çıkış verisi iletebilme kapasitesinde olduğu göz önüne alındığında her bir AS-i ağında sayısal olarak çalışan en fazla 124 giriş ve 124 çıkış bulunabilmekteydi. Otomatik cihaz ekleme/çıkarma gibi önemli özellikler de sistem kapsamında desteklenmekteydi. Ağ güncelleme periyodu, ağda ekli cihaz sayısı ile her bir cihazın güncelleme zamanının (yaklaşık 150us) çarpımı ile elde edilmekle birlikte, en uzun güncelleme zamanı yaklaşık 5ms'dir. Bu zamana, bazı durumlarda gereken cihaz ile bağlantı tekrarları dahil değildi.
- Sürüm 2.11 (1998): Artan ihtiyaçlar neticesinde standart 1998 yılında güncellenmiştir. En temel iyileştirme AS-i ağına analog giriş/çıkış aygıtlarının eklenmesi ve bir ağ içerisinde kullanılabilecek giriş/çıkış cihaz sayısının 62'ye yükseltilmesi alanında olmuştur. Cihaz Hata Bit'i (Peripheral Fault Bit) oluşturularak ağ bünyesindeki cihazların arıza teşhisi sağlanmıştır. Geriye/ileriye dönük uyumluluğu sağlayabilmek için yönetici ile cihazlar arasında iletilen veri çerçeve (data frame) boyutu değiştirilmemiştir. Bunun yerine dört çıkış bitinden birisi sanal A ve B cihazlarını ayırt etmek için kullanılmıştır. Böylelikle adres

uzayı 1A'dan 31A'ya ve 1B'den 31B'ye olmak üzere genişletilmiştir. Sonuç olarak AS-i ağındaki bir cihaz sayısal 4 giriş ve 3 çıkış destekler hale gelmiş, bir ağ üzerinde toplam 248 (4x62) giriş ve 186 (3x62) çıkış desteklenebilir hale gelmiş olup tam olarak dolu bir ağın güncellenme periyodu yaklaşık 10ms'ye çıkmıştır.

- Sürüm 3.0 (2004): 2004 yılı itibariyle endüstriyel Ethernet protokolünün yaygınlaşmasıyla birlikte AS-i protokolünde de iyileştirmeler yapılma ihtiyacı doğmuştur. Bunun sonucunda yeni sürümle birlikte A/B adreslemesini destekleyen sayısal 4 giriş 4 çıkışa sahip cihaz desteği; sayısal 8 giriş 8 çıkışa sahip cihaz desteği; ayarlanabilir 8, 12 veya 16 bit hızlı analog kanal desteği; full-duplex seri iletişim kanal desteği gibi yeni özellikler eklenmiştir. Böylelikle en kötü senaryoda bile sayısal 4 giriş 4 çıkışa sahip toplam 62 cihaz için güncelleme zamanı girişler için 10ms, çıkışlar içinse 20ms olmaktadır.

Tablo 3-1. Çeşitli tarihlerde hizmete giren AS-i sürümleri arası karşılaştırma tablosu.

Sürüm	2.04	2.11	3.0
Yıl	1994	1998	2004
Desteklenen Toplam Çalışan Adedi	31	62	62
Desteklenen Toplam Giriş/Çıkış Adedi	124 Giriş / 124 Çıkış	248 Giriş / 186 Çıkış	248-496 Giriş / 248-496 Çıkış
Ağ Güncelleme Periyodu (desteklenebilecek tüm cihazlar bağlıken)	5ms	10ms	10-30ms
Yönetici Profilleri	M0, M1, M2	M3	M4
Çalışan Profilleri	Diğer tüm AS-i çalışan profilleri	S-X.A (genişletilmiş adresleme modunda çalışma) S-7.3 (1/2/4 analog giriş veya 1/2/4 sayısal çıkış) S-7.4 (1/2/4 analog giriş veya 1/2/4 sayısal çıkış)	S-7.5.5 (100 bps çift yönlü; 2SG/2SÇ) S-7.A.5 (50 bps çift yönlü; 2SG/2SÇ) S-B.A.5 (50 bps çift yönlü; 20 chars/s) S-7.A.7 (4SG/4SÇ) S-7.A.A (8SG/8SÇ) S-7.A.8 (1 analog kanal; 1SÇ) S-7.A.9 (2 analog kanal) S-6.0 (yüksek hızlı analog kanal)

Anahtar Özellikler	Ağa yeni çalışan ekleme/çıkarma	Desteklenen çalışan ve G/Ç adedinde artış Genişletilmiş profil kimlik kodları Çalışan hata durumunu belirleme Basit analog G/Ç desteği	Desteklenen G/Ç adedinde artış İki yönlü senkronize veri iletimi Yeni iletişim tipleri Gelişmiş analog G/Ç desteği
Standart	EN 50295:1998 IEC 62026-2:2000		IEC 62026-2:200X

3.1.5.1. Donanım Yapısı ve Bileşenleri

AS-i protokolü, fabrika otomasyon yönetim hiyerarşisi içerisindeki en alt katman olan Katman 1’de yer alan bir bağlantı sistemidir. Sistem içerisindeki normal veri, güvenlik-öncelikli veri ile sisteme bağlı bileşenlerin besleme gerilimi aynı kablo üzerinden iletilir. Böylelikle basit on-off komut cihazları olan motor kontrol üniteleri, ışık bariyerleri, optik/indüktif/kapasitif yaklaşım algılayıcıları ve daha karmaşık yapıları olan oransal valfler gibi elemanların yanında; acil-durdurma butonları gibi yüksek öncelikli güvenlik elemanları da aynı ağa bağlı olarak güvenle kullanılabilir.

Genel amaçlı bir AS-i ağı dört temel bileşenden meydana gelir: kablo, güç kaynağı, yönetici (master), çalışan (slave). Bunların yanında ağın çalışma mesafesini genişletmek için tekrarlayıcı (repeater), ağ topolojisine ilaveler yapmak için dağıtıcı (distributor) ve hat sonlandırıcıları (terminator) gibi elemanlar da kullanılabilir. AS-i ağı bileşenleri ve özellikleri özetle aşağıdaki gibidir:

3.1.5.1.1. Kablo:

AS-i endüstriyel haberleşme teknolojisinin temeli, ağ içerisinde iletişim halinde olacak cihazlar arası besleme gerilimi ve kontrol verilerin aynı fiziksel kablo üzerinden iletilmesine dayanır. Bu amaçla geliştirilmiş olan IEC 62026-2, CENELEC ve DIN VDE 0281 standartlarına uygun IP-67 sınıfı korumalı olan kablo, içyapısında iki bağımsız kablonun burulmamış (un-twisted) ve manyetik-koruma-kalkansız (un-shielded) bir şekilde birleştirilmesiyle oluşturulmuştur.

AS-i ağlarında, dış tasarımı silindirik şeklinde olanının yanında çoğunlukla özel geometride esnek malzemeden üretilmiş (Şekil) kablo çeşidi kullanılır. Bu özel geometrik tasarımın amacı, hem algılayıcı/eyleyici gibi cihazların ağ kablosuna kolay bağlanabilmesini

sağlamak hem de elektriksel ters bağlantıdan kaynaklanabilecek arızaları önlemektir. Kabloların farklı renk ve özellikte olan çeşitleri vardır. Sarı renk kablo 30VDC ile çalışır ve hem veri iletimi hem de besleme amaçlı kullanılır. Siyah renk kablo 24VDC ile çalışır ve bazı tip eyleyicileri sadece haricen beslemek için kullanılır. Bunun yanında henüz deneysel amaçlı kullanılan kırmızı renkli kablo ise 230VAC'e kadar olan gerilimi taşımak için kullanılmaktadır [SMAR].

3.1.6. Bilgisayar Destekli Talaşlı İmalat

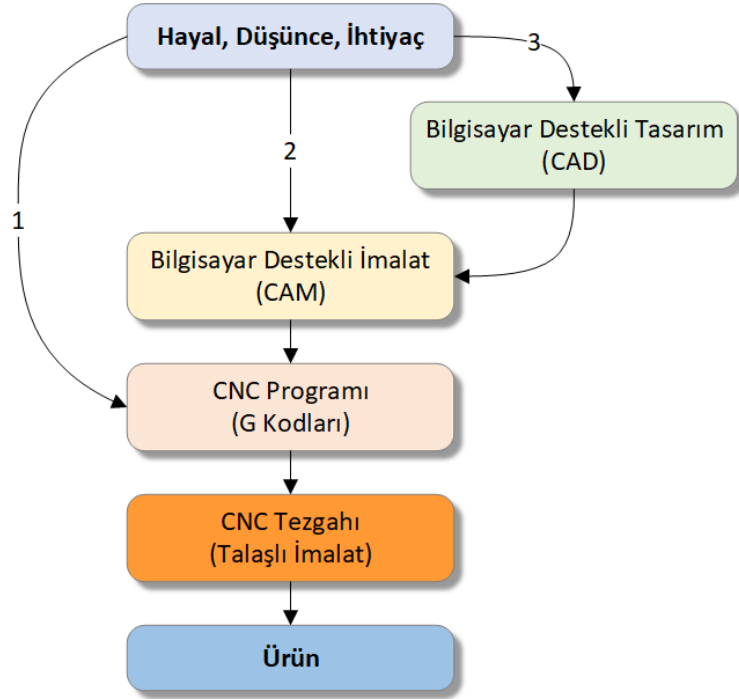
Bir sistemi oluşturan mekanik parçaların imalatı tasarım aşamasından başlar. Tasarım süreci, çoğunlukla bir ihtiyaç ile başlar ve düşünce, hayal gücü ile devam eder. Tasarımın fikri boyutu olgunlaştıktan sonra imalata giden süreç Şekil 3-31'de görüldüğü gibi çeşitli dallara ayrılır. Tasarımcının yetenekleri, tecrübesi ve bilgisi yeterliyse, fikrini direk CNC tezgahın anladığı dil olan G kodlara dökülebilir.

Diğer bir seçenek ise mekanik tasarım destekli bir CAM yazılımı kullanarak, tasarıma ait G kodlarını CAM yazılımına ürettirmektir. CAM işlemi, tasarımın CNC tezgahının anlayacağı dile dönüştürme işlemidir. Tasarımın imalatı için gerekli takım yollarının çıkartılma işlemidir. Birçok durumda CAM işlemi gerçekleştirilirken programcının da bazı kararlar vermesi gerekir. Örneğin parça saat yönünde mi kesilecek, içten mi dıştan mı kesilecek, kesim açısı ne olacak gibi.

En sık kullanılan yöntem ise fikrin önce bir CAD programında çizim haline dönüştürülmesi, sonrasında CAM programı aracılığıyla G kodlarının ürettirilmesidir.

Bu yöntemlerden hangisinin tercih edileceği, fikri tasarımın karmaşıklığı ile de ilgilidir. Tasarım karmaşıklaştıkça geniş kapsamlı bir CAD programı kullanılması ve sonrasında deneyimli bir programcı tarafından CAM işlemi yapılarak G kodlarının üretilmesi daha sağlıklı bir yöntemdir. Fakat bir dikdörtgen kesme, delik açma vb. basit işlemler ise direk G kodlar yazılarak yapılabilir.

Tasarımla alakalı üretilen G kodlar CNC tezgahına yüklenerek operatör tarafından imalat gerçekleştirilir. Tezgahtan bazen nihai ürün elde edilirken bazen de nihai ürünü imal etmek için kullanılacak kalıplar elde edilir.



Şekil 3-31. Bilgisayar destekli talaşlı imalat aşamaları.

Mekanik bir parçanın imalatı oldukça detaylı ve geniş bir konudur. Bu alanda kullanılan birçok yöntem bulunmakta olup, en yaygın olarak kullanılan üç yöntem: kalıp çıkarma ile döküm, talaşlı imalat ve eklemeli imalatdır.

Kalıp çıkarma ile döküm yönteminde imal edilmek istenen hedef parçanın önce kalıbı çıkartılır. Daha sonra kalıbın içerisine sıvı hale getirilmiş (fiziksel olarak eriyik haline getirilmiş katı malzeme veya reaksiyonla katılaşabilen özel kimyasal sıvılar) malzeme dökülerek, dökülen malzemenin zamanla katılaşması sonucu hedef parça elde edilir.

Eklemeli imalat ise 3B yazıcı teknolojisiyle birlikte yaygınlaşmış bir imalat yöntemidir. Prensipte ısıtılarak sıvı hale getirilebilen ve soğuduğunda katılaşan bir hammaddenin çok ince katmanlar halinde üst üste eklenmesi şeklinde çalışır. Tuğladan duvar örülmesi bu yönetime örnek olarak verilebilir. Bazı uygulamalarda toz veya sıvı haldeki hammaddenin belirli dalga boyundaki lazer ışını ile etkileşime girerek katılaştırılması ve birbirlerine eklenmesi yöntemi (lazer sinterleme) de kullanılmaktadır.

Diğer bir imalat türü olan talaşlı imalat, hedef parça imal edilirken, hammadde üzerinden parça (talaş) kopartılması yöntemi kullanılır. Talaş koparmak için geleneksel olarak çelik veya elmas ile güçlendirilmiş kesici takımlar kullanılabildiği gibi, günümüzde yüksek güçlü lazer ışınlarının kullanımı da yaygınlaşmaktadır.

Geleneksel talaşlı imalatda kullanılan iki temel tezgah türü vardır: freze ve torna. Her iki tezgahda da hammaddeyi işleyen bir kesici takım ve kesici takım/hammaddeyi döndüren bir elektrik motoru vardır. Program içerisinde hızı istenilen değere ayarlanabilen bu motora genellikle “spindle” adı verilir. Motor iş mili denilen döner bir şafta bağlıdır.

Freze tezgahında kesici takımın iş miline bağlanmasına yarayan aparata “takım tutucu”, “malafa” veya “mandren” denir. Freze tezgahı, kesici takım kendi eksenini etrafında dönerken tezgaha genellikle “mengene” (clamp) ile sabitlenmiş hammadde üzerinden talaş kaldırması yöntemiyle çalışır. Döner vaziyetteki kesici takımın hammadde üzerinde gideceği konumlar ise NC programında tanımlanır. Her freze tezgahının sahip olduğu eksen sayısı değişebilmekle birlikte, tezgah XYZ olmak üzere en az üç eksene sahiptir.

Torna tezgahında ise hammadde dönen bir iş miline bağlanır. Parçanın iş miline bağlanmasına yarayan aparata “ayna” (chuck) denir. Hammadde kendi eksenini etrafında dönerken kesici kafanın parça üzerinde hareket etmesiyle, parça imalatı gerçekleşir. Genellikle torna tezgahları iki eksenli olup, NC programındaki kodlara göre kesici kafa çap ekseninde (X) ve/veya boy ekseninde (Z) hareket eder.

3.1.6.1. CNC Tezgahı Keserek İlerleme ve İş Mili Devir Hesabı

Bir parçanın CNC tezgahında sağlıklı bir şekilde işlenebilmesi için keserek ilerleme ve iş mili devrinin doğru belirlenmesi gereklidir. İdeal kesme hızları belirlenirken işlenecek malzemenin cinsi (ahşap, alüminyum, çelik, vb.), kesici takımın çapı, kesici takımın diş sayısı, kesici takımın kesme özellikleri vb. parametreler göz önüne alınır. Bu iki hızın değeri doğru olarak belirlenmezse; imalat sonucu elde edilen yüzey kalitesi, imalatın süresi, kesici takımın ve tezgahın ömrü gibi parametreler olumsuz açıdan etkilenir.

NC programında keserek ilerleme ve iş mili devri olması gerekenden düşük tanımlanmış olabilir. Özellikle tezgah çalışırken çıkan seslerden ve talaşın yapısından, kesici takım dönüş hızının doğru ayarlanıp ayarlanmadığı anlaşılabılır. Eğer tiz bir ses duyuluyor veya talaş un gibi ufalanmış olarak çıkıyorsa, dönüş hızı olması gerekenden yüksektir denebilir. Eğer tezgah zorlanıyormuş gibi titreşerek bas bir ses duyuluyorsa veya çok iri parçalı talaşlar çıkıyorsa, takım dönüş hızı olması gerekenden düşüktür denebilir.

Freze tezgahı için keserek ilerleme ve iş mili devir hesaplarına geçmeden önce bazı temel kavramlar ve sembollerin açıklanması faydalı olacaktır:

- fz: Diş başına düşen ilerleme hızı (mm/dev)

- Vc: Kesici takım kesme hızı (m/dk)
- D: Kesici takım çapı (mm)
- Z: Kesici takım diş sayısı
- S: İş mili devir sayısı (dev/dk)
- F: Keserek ilerleme hızı (mm/dk)

Benzer şekilde kesici takımların işleyebileceği malzeme özellikleri tanımlanırken aşağıdaki gibi bazı kısaltma ve semboller kullanılır. Bu semboller özellikle kesici takım katalog değerlerinin okunması aşamasında önemlidir.


- P: çelik türleri
- M: paslanmaz çelikler
- K: döküm
- N: demir olmayan malzemeler
- S: titanyum
- H: ısıl dirençli alaşımlar

Torna ve freze tezgahlarında parça işlenirken yüzey kalitesinin homojen olabilmesi için, tüm yüzeylerde yapılan kesme işlemi sırasında kesici ucun çizgisel hızının aynı olması gereklidir. Freze tezgahında parça işlenirken kesici takım (D) çapı ve diş sayısı (Z) sabittir. Bu nedenle kesici ucun çizgisel hızı da sabittir. İş mili devir sayısı (S) bir kere hesaplanır ve işleme boyunca genellikle değeri sabit tutulur. Torna tezgahında ise parça işlenirken çap sürekli değiştiğinden dolayı kesici takım çapı (D) de sürekli değişir. Bu nedenle kesici ucun çizgisel hızını sabit tutabilmek için, iş mili devir sayısı (S) çap ile birlikte belirli bir oranla sürekli değişir. Bu nedenlerle torna ve freze tezgahlarında keserek ilerleme ve iş mili devir hesabı farklı yapılır.

CNC freze tezgahlarında:

- İş mili devir hesabı $S = \frac{V_c * 1000}{D * \pi}$ formülüyle yapılır.
- Keserek ilerleme hesabı $F = S * Z * fz$ formülüyle yapılır.

Örneğin alaşımsız çelik (St 37) bir malzeme üzerinde, 8 mm çapında ve 4 adet dişe sahip bir kesici takım ile frezeleme yapılmak istensin. Kesici takımın katalogu incelendiğinde, Şekil 3-32'deki gibi alaşımsız çelik malzeme için $V_c = 130-160$ m/dk arası, $fz = 0,04$ mm/dev olarak verildiği görülmektedir.



Malzeme Material	Gerilme Dayanımı Tensile Strength Hardness N/mm²	DIN	Malzeme No Material No	Kesme Değeri Cutting Speed Vc-m/dk	Diş Baş İlerleme / Fz mm/dev Feed Per Tooth									
					ø4	ø6	ø8	ø10	ø12	ø14	ø16	ø18	ø20	
Çelik Steel	Alaşımsız Çelik Unalloyed Steel	<450	St.37	1010	130-160	0.02	0.03	0.04	0.04	0.05	0.06	0.07	0.08	0.09
	İmalat Çeliği Free Cutting Steel	400-550	CK40	1040	120-140	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.07	0.07
	İslah Çeliği Heat Treatment Steel	550-900	42CrMo4	4140	100-130	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.06
	Soğuk İş Takım Çeliği Tool Steel	900-1200	2379 2080	D2-D3	90-120	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.06
	Sıcak İş Takım Çeliği Heat Treatment Tool Steel	900-1200	2344	H13	70-90	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.06	0.06
Döküm Cast Iron	Grü Dökme Demir Grey Cast Iron	≤550	GG25	GG25	120-140	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.07	0.07
	Alaşımlı Dökme Demir Alloyed Cast Iron	300-700	GGL-NiCr35.2	GGL-NiCr35.2	120-140	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.07	0.07
	Hassas Döküm Malleable Cast Iron	400-750	GT55S	GT55S	100-120	0.02	0.03	0.04	0.04	0.05	0.05	0.06	0.07	0.07

Şekil 3-32. Bir freze kesici takımına (freze çakısı) ait katalog bilgileri.

Buna göre $V_c = 150$ m/dk için;

- İş mili devir sayısı $S = \frac{V_c \cdot 1000}{D \cdot \pi} = \frac{150 \cdot 1000}{8 \cdot 3,14} \cong 5971 \text{ dev/dk}$ olarak hesaplanır.
- Keserek ilerleme hızı ise $F = S \cdot Z \cdot f_z = 5971 \cdot 4 \cdot 0,04 \cong 955 \text{ mm/dk}$ olarak hesaplanır.

Bu durumda freze tezgahı için yazılacak NC programının başlangıç satırlarından birine “N10 S5971 F955 M3” kodu girilir. Böylelikle kesici takım dönüş ve ilerleme hızları tanımlanmış ve takım saat yönünde dönecek şekilde iş mili döndürülmeye başlanır.

CNC torna tezgahında keserek ilerleme ve iş mili devir hesapları frezeye göre biraz daha farklıdır. Bunun başlıca sebebi torna tezgahında hammadde dönmekte olduğu için, hammaddenin çapı aynı zamanda kesici takım çapı olmuş olur. Frezede sabit olan takım çapı değeri tornada değişkendir. Ayrıca torna tezgahlarında tek kesici takım vardır ($Z=1$). Bu nedenle takım kataloğunda verilen diş başına ilerleme hızı (f_z) değeri, NC programda F parametresi olarak girilir. Torna tezgahı, girilen F parametresi değerini sürekli iş mili devri ile çarparak frezedeki keserek ilerleme hızını (F) hesaplar.

Hammadde işlenirken iş mili dönüş hızı sabit tutulduğunda, işlenen çap değiştiğinde kesici takıma etki edecek kesici takım kesme hızı da değişecektir. Örneğin sabit iş mili devrinde işlenecek çap arttıkça, kesici takıma etkiyen çizgisel hız da artacaktır. Bu da parçanın yüzey pürüzlülüğü ve takım ömrü parametrelerini olumsuz yönde etkileyecektir. Bu durumun önüne geçmek için torna tezgahlarında kesici takım kesme hızını (V_c) sabit tutan seçimlik otomatik değişken iş mili devir ayarı özelliği eklenmiştir. NC programda G96 komutuyla aktif edilen bu özellikte S parametresiyle kesici takım kesme hızı (V_c) tanımlanır. Tornada işlenecek çap değiştiğinde kesici takıma etkiyen çizgisel hız (keserek

ilerleme hızı F mm/dk) sabit kalacak şekilde iş mili devri sürekli değiştirilir. Böylelikle işlenen parçanın tüm yüzeylerinde elde edilen pürüzlülük homojen olur.

Örneğin kataloğunda $V_c = 200$ m/dk ve $f_z = 0,095$ mm/dev yazılı olan bir kesici takım kullanılarak torna tezgahında bir parça işlenecekse, hazırlanan NC programının başlangıç satırlarından birine “N20 G96 S200 M3” girilir. Sonrasında kesimin yapılacağı kod satırına ise “N30 G01 X-2 Z-10 F0.095” yazıldığı düşünülürse, bu kod satırında kesici takım X ekseninde -2 mm, Z ekseninde -10 mm hareket ederken dış başına ilerleme hızı 0,095 mm/dev olur.

Vida dişi açma gibi bazı uygulamalarda istenilen kaliteyi elde edebilmek için iş mili devir sayısının sabit tutulması önemlidir. Torna tezgahına sabit iş mili devri girebilmek için G97 kodu kullanılır ve S parametresiyle girilen değer, direk iş milinin devir sayısıdır. Örneğin NC programda “N10 G97 S1200 M4” kodu girildiğinde, iş mili saat yönünün tersi yönde ve 1200 dev/dk ile dönmeye başlar.

3.1.6.2. Temel G Kodlar

Bilgisayar destekli imalatla CNC tezgahlarının programlanması için kullanılan kodlardır. “G” harfi ile başladıkları için genel olarak G kodları adı verilmiştir. Tezgahtaki kesici takım veya parçanın yörünge hareket fonksiyonları, ölçü birimi dönüşümü, CNC programını başlatma/durdurma/bekletme, kesici takım çapı telafisi verme gibi işlevler bu kodlar ile tanımlanır. G kodlar bir CAM programı üzerinden üretilebileceği gibi bir metin belgesine deneyimli bir kullanıcı tarafından da satır satır yazılabilir. Aşağıda sıkça kullanılan temel G kodlar listelenmiştir:

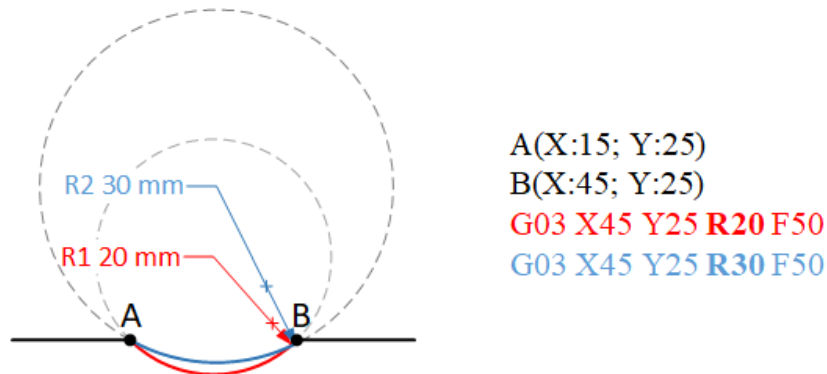
- G00: Takımın kesmeden doğrusal bir yörüngede ilerlemesini sağlar. Bu nedenle keserek ilerleme hızına göre daha yüksek hızda hareket eder. Hareket hızı CNC makinasının üreticisi tarafından belirlenir ve değeri makina parametrelerinden değiştirilebilir. CNC makinasının sahip olduğu eksen sayısını dikkate alarak, komuta parametre olarak koordinat değerleri girilir. Örneğin “G00 X55” kodu tezgahı kesmeden X ekseninde +55 mm konumuna hareket ettirir. “G00 Y-34 Z3” kodu ise tezgahı Y ekseninde -34 mm konumuna hareket ettirirken aynı anda Z eksenini de +3 mm konumuna hareket ettirir. Tezgah bileşik hareket (enterpolasyon) yapar.
- G01: Takımın keserek doğrusal yörüngede ilerlemesini sağlar. Hareket komutları G00 da olduğu gibidir. G00’dan farklı olarak G01’de keserek ilerleme olacağı için kullanıcı F parametresiyle hesapladığı ilerleme hızını girebilir. Örneğin “G01 X20

Y10 F100” kodu tezgahın X ekseninde +20 mm ile Y eksenin +10 mm konumuna aynı anda ve 100 mm/dk hızla hareket etmesini sağlar.

- G02/G03: Takımın keserek dairesel yörüngede (circular enterpolasyon) ilerlemesini sağlayan komutlardır. G02 saat yönünde, G03 ise saat yönü tersine hareketi sağlar. İki farklı şekilde kullanılabilirler: yarıçap tanımlayarak, merkez tanımlayarak.

Yarıçap tanımlama yönteminde dairesel hareket komutundan (G02/G03) sonra kesici takımın ulaşması istenen hedef nokta koordinatları (örneğin X, Y) ile çizilecek hayali dairenin yarıçapı (R) tanımlanır. Kesici takımın bulunduğu nokta ile ulaşılmak istenen hedef nokta arasındaki mesafe hayali dairenin çapından (2R) küçük olmalıdır. G02/G03 dairesel yörüngede keserek ilerleme komutları olduğu için, satır sonunda kesme hızı F ile tanımlanır.

Şekil 3-33’deki örnekte kesici takımın A noktasından B noktasına hareketine yönelik iki farklı dairesel yörünge gösterilmiştir. A noktasının koordinatları (X:15; Y:25) ve B noktasının koordinatları (X:45; Y:25) olsun. Takım ucunun A noktasında olduğu durumda “G03 X45 Y25 R20 F50” kodu işletilirse, kesici takım r1 (20 mm) yarıçaplı hayali dairenin kırmızı-çizgili yayı üzerinden 50 mm/dk hızla geçer. Eğer aynı kod “R30” olacak şekilde değiştirilirse, bu sefer kesici takım r2 (30 mm) yarıçaplı hayali dairenin mavi-çizgili yayı üzerinden 50 mm/dk hızla geçer. Her iki takım yoluna dikkat edilirse daire çapı küçüldükçe, iki nokta arası çizilebilen yay uzunluğu büyümekte ve kesik derinleşmektedir. Tam tersi de geçerlidir.

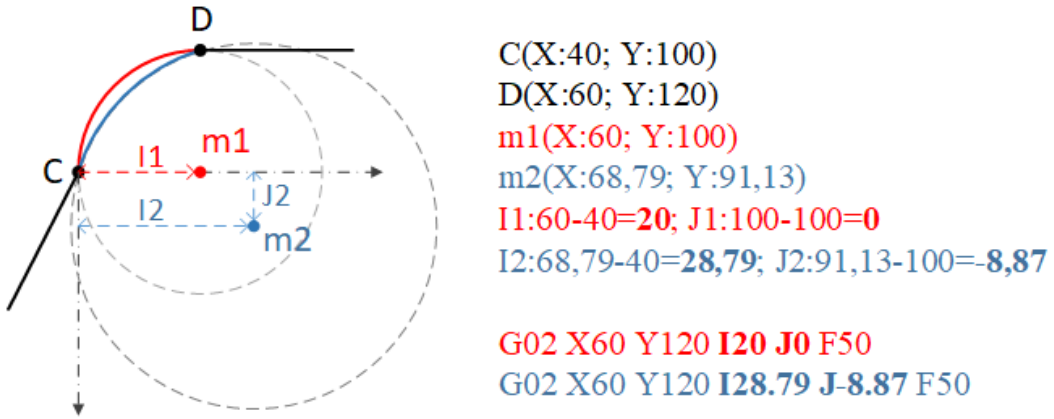


Şekil 3-33. G03 komutuyla R1 ve R2 yarıçaplı dairesel takım yörüngeleri oluşturulması.

Merkez tanımlama yöntemi ilk yöntem ile benzerlikler içerse de biraz daha karmaşıktır. Bu yöntemde dairesel hareket komutundan (G02/G03) sonra yine kesici takımın ulaşması istenen hedef nokta koordinatları (örneğin X, Y) girilir. Sonrasında ilk yöntemden farklı olarak hayali dairenin yarıçapı yerine merkez koordinatları girilir. Hayali dairenin merkez koordinatları dairesel hareketin

başlangıç noktasına göre tanımlanır. Hayali çemberin merkezinin X değerinin, dairesel hareketin başlangıç noktasının X değerine olan mesafesi (mm) I harfiyle; hayali çemberin merkezinin Y değerinin, dairesel hareketin noktasının Y değerine olan mesafesi (mm) J harfiyle tanımlanır. G02/G03 dairesel yörüngede keserek ilerleme komutları olduğu için, satır sonunda kesme hızı F ile tanımlanır.

Şekil 3-34'deki örnekte kesici takım C noktasından D noktasına hareketine yönelik iki farklı dairesel yörünge gösterilmiştir. C noktasının koordinatları (X:40; Y:100) ve D noktasının koordinatları (X:60; Y:120) olsun. Takım ucunun C noktasında olduğu durumda "G02 X60 Y120 I20 J0 F50" kodu işletilirse, kesici takım m1 merkezli hayali dairenin kırmızı-çizgili yayı üzerinden 50 mm/dk hızla geçer. Eğer aynı kod "I28.79" ve "J-8,87" olacak şekilde işletilirse, kesici takım m2 merkezli hayali dairenin mavi-çizgili yayı üzerinden 50 mm/dk hızla geçer. Bu örnek kodlardaki I ve J değerleri, hayali çemberlerin m1/m2 merkezleri ile C noktası koordinatlarının farkından hesaplanmıştır. $I1=m1(X)-C(X)$ formülüne göre $I1=60-40=20$; $J1=m1(Y)-C(Y)$ formülüne göre $J1=100-100=0$ şeklinde hesaplanır. Benzer hesaplama m2 için de yapılır. Dikkat edilirse her iki takım yolunda daire çapı küçüldükçe, iki nokta arası çizilebilen yay uzunluğu büyümekte ve bombeleşme artmaktadır.



Şekil 3-34. G02 komutuyla m1 ve m2 merkezli dairesel takım yörüngeleri oluşturulması.

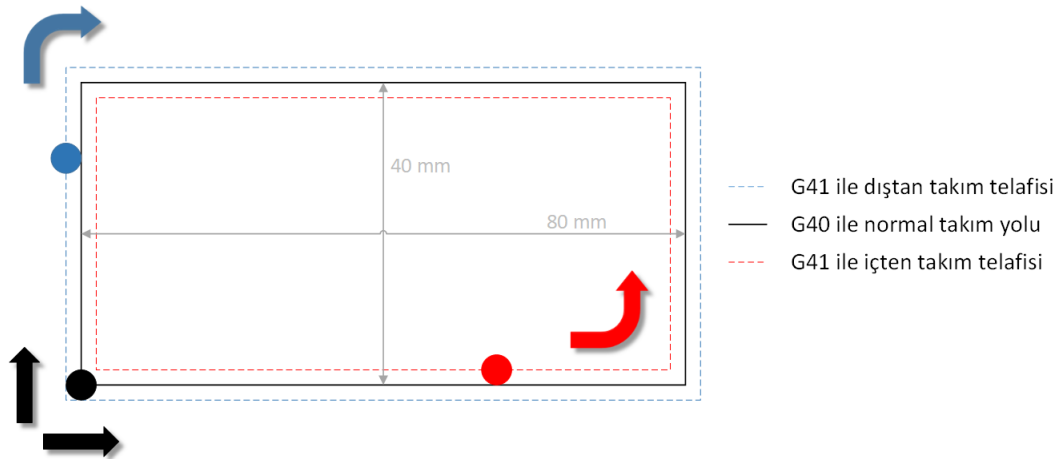
- G04: Bekleme komutudur. Parametre "X" ile tanımlanırsa birimi saniye, "P" ile tanımlanırsa birimi milisaniyedir. Örneğin "G04 X1.5" komutu programda 1,5 saniyelik gecikme verirken "G04 P200" komutu 200 ms gecikme verir.
- G20/G21: Sırasıyla inç ve metrik ölçü sistemlerine geçişi sağlayan komutlardır.
- G40/G41/G42: Elle yazılan G kodlar kullanılarak oluşturulan takım yolları, iş milinin merkezi dikkate alınarak hesaplanır. Bu durumda aynı program çalıştırılmış olsa da, iş mili çapının değişmesine göre elde edilecek parça

ölçülerinde değişiklik gözlenir. Bu nedenle özellikle elle yazılan kodlarda takım çapı telafisi yapılması gereklidir. CAM programları takım yollarını, imalatla kullanılacak takım çaplarını dikkate alıp telafilerini hesaplayarak çıkartır.

Takım çapı telafisi yapılmadan oluşturulmuş bir NC programında, örneğin bir 100 x 100 mm ölçülerindeki alüminyum bir levha içinden 40 x 80 mm ölçülerinde bir dikdörtgen parça çıkarılmak istensin. Kullanılacak takım çapı 2 mm seçilirse, elde edilecek parçanın ölçüleri 38 x 78 mm olur. Eğer aynı program 4 mm çaplı takımla işletilirse, bu durumda elde edilecek parçanın ölçüleri 36 x 76 mm olur.

Bir önceki örnekte olduğu gibi takım çapında meydana gelecek değişikliklerden imalatın etkilenmemesi için, CNC programında G41 ve G42 takım çapı telafi kodları kullanılır. G41 ile saat yönünde hareketle işlenecek konumun solundan, saat yönü tersi hareketle sağından telafi verilir. G42 ile ise saat yönünde hareketle işlenecek konumun sağından, saat yönü tersi hareketle solundan telafi verilir. G40 ile ise önceden verilmiş tüm takım telafileri iptal edilir.

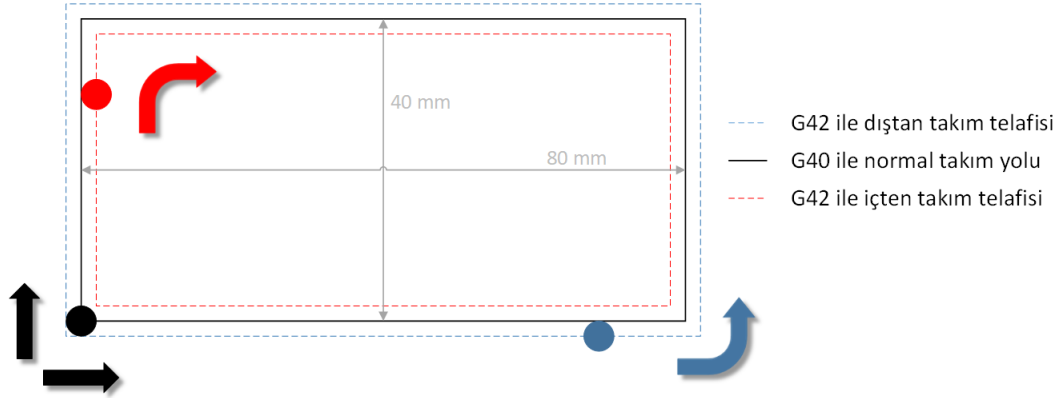
Şekil 3-35'deki siyah-çizgili sınırlar imal edilmek istenen asıl parçayı (40 x 80 mm) temsil eder. Bu aynı zamanda G40 kullanılıp takım telafisi iptal edildiğindeki takım yolunu temsil eder. G41 kullanıldığında, eğer takım saat yönünde dönecek şekilde hareket ederse, takım mavi-çizgili yolu takip eder. Diğer bir ifadeyle takım parça sınırlarının dış tarafından hareket eder ve imal edilmek istenen parça iç tarafta kalır. Eğer takım saat yönü tersi dönecek şekilde hareket ederse, takım kırmızı-çizgili yolu takip eder. Diğer bir ifadeyle takım parça sınırlarının iç tarafından hareket eder ve imal edilmek istenen parça dış tarafta kalır.



Şekil 3-35. G41 koduyla takım yolu telafisi.

Şekil 3-36'deki siyah-çizgili sınırlar da imal edilmek istenen asıl parçayı (40 x 80 mm) temsil eder. G41'in tam tersi olarak G42 kullanıldığında, eğer takım saat yönünde dönecek şekilde hareket ederse, takım kırmızı-çizgili yolu takip eder.

Diğer bir ifadeyle takım parça sınırlarının iç tarafından hareket eder ve imal edilmek istenen parça dış tarafta kalır. Eğer takım saat yönü tersi dönecek şekilde hareket ederse, takım mavi-çizgili yolu takip eder. Diğer bir ifadeyle takım parça sınırlarının dış tarafından hareket eder ve imal edilmek istenen parça iç tarafta kalır. G40 kullanıldığında ise takım yolu telafisi iptal olur ve takım siyah-çizgili yolu takip eder. G42 kullanıldığında ise takım yolu telafisi iptal olur ve takım siyah-çizgili yolu takip eder.



Şekil 3-36. G42 koduyla takım yolu telafisi.

- G54-59: CNC tezgahlarında bulunan her bir eksenin fabrikada üretim aşamasında belirlenmiş başlangıç noktaları vardır. Kesici takımın koordinatlarını da tezgah bu başlangıç noktalarına göre ölçülendirir ve makina (machine) koordinatları denir. Tezgahın eksenlerinin sıfır noktalarını belirleyebilmek için her bir eksene üretici tarafından elektronik sınır anahtarı yerleştirilmiştir. Tezgah açıldığında kullanıcıdan ilk olarak her eksen bu sınır anahtarlarına kadar giderek mutlak başlangıç konumunun belirlenmesini ister. Bu andan itibaren NC programındaki konumları tanımlamada mutlak başlangıç noktası kullanılır.

CNC tezgahında parça imalatına başlamadan önce NC programında tanımlanan başlangıç noktasıyla parça üzerindeki başlangıç noktasının örtüştürülmesi gereklidir. Bu işlemi yapabilmek için markadan markaya değişebilmekle birlikte G54-59 arası komutlar kullanılır.

Bu aradaki komutlar ile tanımlanmış adreslere 6 farklı ürün için 6 farklı başlangıç noktasına ait makina koordinatları yazılabilir.

- G96/G97: İş mili devir sayısı ayarının yapıldığı kodlardır. Özellikle CNC torna tezgahları için hazırlanan programlarda kullanılır. G96 ile yapılan tanımlamada kesme hızı sabit kalacak şekilde iş mili devir sayısı tezgah tarafından otomatik olarak ayarlanır. G97 ile yapılan tanımlamada iş mili dönüş hızı sabit kalır. Torna tezgahında kesme hızının sabit kalması, değişen çaplara sahip yüzeylerin işlenme kalitesinin (yüzey pürüzlülüğü) aynı olmasını sağlar.

3.1.6.3. Temel M Kodlar

M fonksiyonları CNC makinasının motor dönüş yönü ayarı, soğutma sıvısı/havasını açma/kapama, takım değiştirme gibi ek işlevleri yerine getiren yardımcı fonksiyon kodlarıdır. Fonksiyonların işlevleri CNC makinasının üreticisine göre değişiklik gösterebilir. Aşağıda genel olarak kullanılan M fonksiyonları ve görevleri açıklanmıştır.

- M03/M04: Takım ucunu döndüren motoru (spindle) çalıştıran, genellikle programın en başında kullanılan komuttur. M03 ucun saat yönünde, M04 ise ucun saat yönü tersi dönmesini sağlar.
- M05: Takım ucunu döndüren motoru durduran komuttur.
- M02/M30: Programı sonlandıran komutlardır. Programın bitişini ifade ederler. M02’de program sonlandıktan sonra program imleci en son satırda kalır. Başlat butonuna tekrar basıldığında program yürütülmez. Kullanıcının programı baştan başlatması gereklidir. Özellikle programın bir kez çalıştırılacağı durumlarda güvenlik sağladığı için tercih edilir. M30’da ise program sonlandıktan sonra program imleci en başa döner. Başlat butonuna basıldığında otomatik olarak program en başından itibaren yürütülmeye başlanır. Özellikle aynı program artarda çalıştırılacağı durumlarda pratiklik sağladığı için tercih edilir.
- M08/M09: Sırasıyla soğutma sıvısını açan ve kapatan kodlardır.
- M71/M72: Sırasıyla soğutma havasını açan ve kapatan kodlardır.

3.1.6.4. NC Programlama Mantığı

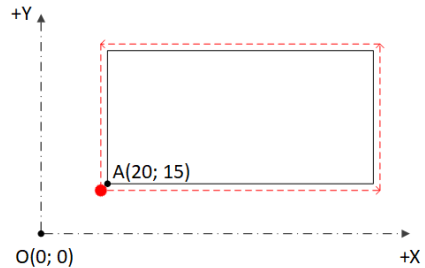
NC programları, CNC tezgahlarının çalışmasını sağlayan, G ve M kodları ile oluşturulmuş imalat programlarıdır. Program satırlardan ve satırlar içinde yazılı kodlardan oluşur. CNC tezgahının markasına göre değişiklik göstermekle birlikte, genellikle kod harfinden (G veya M) sonra kullanılan tek veya çift haneli rakam, komutu tanımlar. Bazı tezgahlar kodu kısaca “G0” şeklinde kabul edebildiği gibi, bazı tezgahlarda uzun uzun “G00” şeklinde yazılması gereklidir. Benzer şekilde bazı tezgahlarda her satıra N harfi ile başlayan ve alt satırlara doğru artan bir numara verilmesi gereklidir (N10, N20, N25, ... gibi). Bazılarında ise programlarda satır numarası vermeye gerek yoktur. Ayrıca kimi tezgahlarda her satır sonu özel bir karakter “;” ile sonlandırılırken kimisinde böyle bir zorunluluk yoktur.

Program kullanıcının istediği satırdan itibaren başlatılabilir. Tezgah her seferinde tek satırda tanımlı görevleri yürütür. Aktif olan satırdaki görevler tamamlanmadan bir sonraki satıra geçmez. Birden fazla satırın aynı anda yürütülmesi söz konusu değildir.

NC programlama mantığında bir satırda aktif edilen G kod, bir sonraki G koda gelene kadar aktif kalır. Bu özellik gereksiz yere aynı kodu tekrar tekrar yazma gereksinimini kaldırdığı için faydalıdır. Özellikle hareket komutları olan G00-G03 ile sıklıkla kullanılır. Örneğin “N50 G00 X100 Y200;” şeklinde tanımlanmış bir kod satırı aktif olduğunda, tezgah aynı anda hem X ekseninde 100 mm’ye hem de Y ekseninde 200 mm’ye doğru hareket edecektir. Sonuç olarak tezgah her iki eksenin ortasında olacak şekilde çapraz bir hareket sergileyecektir. Eğer kod “N50 G00 X100; N55 Y200;” şeklinde olursa, tezgah önce X ekseninde 100 mm’ye sonra Y ekseninde 200 mm’ye doğru hareket edecektir. Benzer şekilde bir satırda F parametresiyle ilerleme hızı tanımlaması yapılmışsa, bir sonraki tanımlamaya kadar bir önce yapılan tanımlama geçerlidir.

Aşağıda Şekil 3-37’de örnek olarak baskı devre bakır plakette 40 x 80 mm ölçülerinde dikdörtgen bir parçanın kesimi için oluşturulmuş bir NC programı görülmektedir. Bakır plakette, yüzeyi 0,2 mm kalınlığında bakırla kaplı olan ve toplam 1,4 mm kalınlığındaki FR4 plakadan oluşmaktadır. Kesici takım olarak 4 mm çaplı iki dişli freze çakısı kullanılmakta olup, takım ölçüleri tezgahta T0101 olarak kayıtlıdır. Takım ömrünü uzatmak ve mevcut iş milinin gücü kapsamında çalışabilmek için kesim işlemi 2 aşamada gerçekleştirilecektir. 1. turda bakır yüzey kesileceği için ilerleme hızı düşük, sonraki turda ise ilerleme hızı daha yüksek tutulacaktır.

```
N10 G40 G00 Z5; // Takım telafisi iptal eder ve takımı 5 mm yukarı çeker.
N15 X0 Y0; // Kesmeden orijin noktasına gider.
N20 T0101 // 1 numaralı takımı seçer. (4 mm çaplı freze çakısı)
N25 G42 S2000 M3; // Takım telafisi verir, spindle hızını 2000 dev/dk ayarlayarak motoru saat yönünde çalıştırır.
N30 G00 X20 Y15; // Kesimin başlayacağı A noktasına gider.
N35 G01 Z-0.4 F200; // Z ekseninde -0.4 mm derinliğe iner, ilerleme hızı 200 mm/dk.
N40 X100; // Keserek X100'e gider. (1. tur)
N45 Y55; // Keserek Y55'e gider. (1. tur)
N50 X20; // Keserek X20'ye gider. (1. tur)
N55 Y15; // Keserek Y15'e gider. (1. tur)
N60 Z-1.5 F300; // Z ekseninde -1.5 mm derinliğe iner, ilerleme hızı 300 mm/dk.
N65 X100; // Keserek X100'e gider. (2. tur)
N70 Y55; // Keserek Y55'e gider. (2. tur)
N75 X20; // Keserek X20'ye gider. (2. tur)
N80 Y15; // Keserek Y15'e gider. (2. tur)
N85 Z5; // Takımı yüzeyin 5 mm yukarısına çeker.
N90 M5; // Spindle durdurur.
N95 M2; // Programı sonlandırır.
```

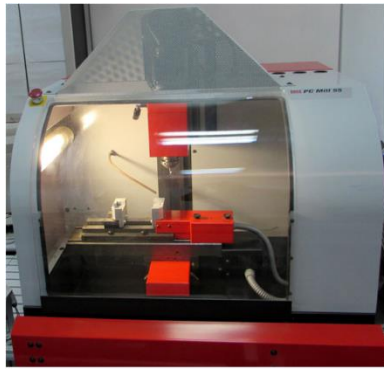


Şekil 3-37. Bakır plaketten 40 x 80 mm ebatlarında parça kesimi örneği.

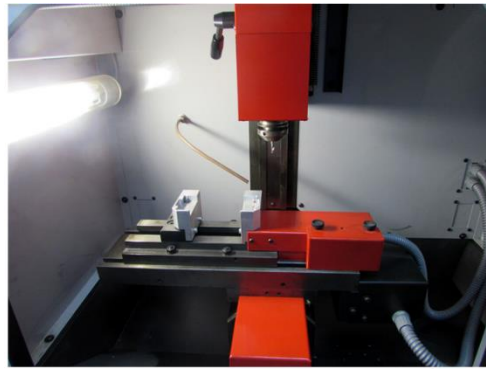
3.1.6.5. Turn55 ve Mil55 CNC Tezgahları Tanıtımı

FMS100 bünyesinde genellikle eğitim amaçlı kullanıma uygun tasarlanmış olan Turn55 ve Mill55 tezgahlarından birer adet bulunmaktadır.

Mill55 tamamen bilgisayar kontrollü 3 eksen (XYZ) bir freze tezgahıdır. Şekil 3-38’de dıştan ve içten görülmekte olan tezgahın elektrik kumandalı mengenesi, elektro-pnömatik kumandalı açılıp/kapanan tezgah kapısı ve soğutma havası çıkışı gibi ek özellikleri bulunmaktadır. Tezgah iş miline bağlı mandrene 16 mm çapa kadar freze çakısını elle bağlamak mümkündür. Otomatik takım değiştirme sistemi yoktur.



(a)



(b)

Şekil 3-38. FMS100 bünyesindeki Mill55 tezgahı: dıştan görünüş (a), içten görünüş (b).

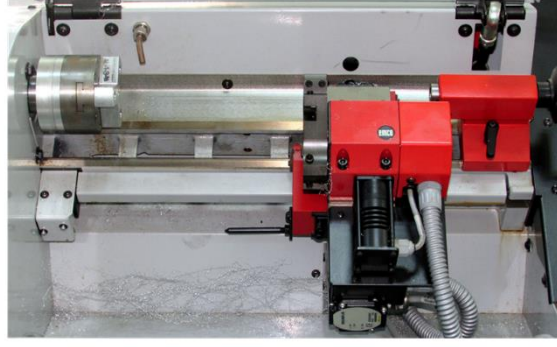
Bunun dışında Mill55 tezgahın sahip olduğu diğer teknik özellikleri aşağıdaki gibidir.

- Eksen hareket sınırları (X/Y/Z): 190 / 125 / 190 mm
- Parça bağlanabilen taban alanı: 420 x 125 mm
- Tabana yüklenebilecek en fazla yük: 10 kg
- İş biline bağlanabilecek en geniş takım: 16 mm
- İş milini süren motor (spindle) gücü: 0,75 kW
- İş mili devri aralığı: 100-3500 dev/dk
- İş milinin üretebildiği en fazla tork değeri: 8 Nm
- Kesici takım değiştirme: elle

Turn 55 de tamamen bilgisayar kontrollü 2 eksen (XZ) bir torna tezgahıdır. Şekil 3-39’da dışı ve içi görülmekte olan tezgahın elektro-pnömatik kumandalı açılıp/kapanan aynası ve tezgah kapısı, soğutma havası çıkışı ve otomatik takım değiştirme sistemi gibi ek özellikleri bulunmaktadır.



(a)



(b)

Şekil 3-39. FMS100 bünyesindeki Turn55 tezgahı: dıştan görünüş (a), içten görünüş (b).

Bunun dışında Turn55 tezgahın sahip olduğu diğer teknik özellikleri aşağıdaki gibidir.

- Eksen hareket sınırları (X / Z): 48 / 236 mm
- Tezgaha bağlanabilen en fazla parça boyu / çapı: 215 / 52 mm
- İş milini süren motor (spindle) gücü: 0,75 kW
- İş mili devri aralığı: 120-4000 dev/dk
- İş milinin üretebildiği en fazla tork değeri: 14 Nm
- Kesici takım değiştirme: 8 seçenekli otomatik

Şekil 3-40'da görülmekte olan otomatik takım değiştirme sistemi kapsamında, tezgahın sahip olduğu elektrik kontrollü döner kafa sayesinde, tezgaha aynı anda 8 farklı kesici takım bağlanabilir. NC programı içerisinde verilecek komutlar sayesinde imalat sırasında kesici takım değiştirilebilir.



Şekil 3-40. Turn55 tezgahının sahip olduğu otomatik takım değiştirme sistemi.

Her iki tezgahın da kendilerine ait birer kişisel bilgisayarı vardır. Tezgahları kontrol eden WinNC32 yazılımı bu bilgisayarlarda çalışır. WinNC32 yazılımı ile tezgahlara NC programı yüklenebilir, görüntülenebilir, değiştirilebilir, çalıştırılabilir ve durdurulabilir.

Ayrıca tezgaha ait tüm ayarlar, tanımlamalar yine bu yazılım üzerinden yapılır. Yazılım bir anlamda tezgahın kullanıcı arayüzüdür. Şekil 3-41’de tezgah bilgisayarları ve özel tasarım kontrol klavyesi görülmektedir.

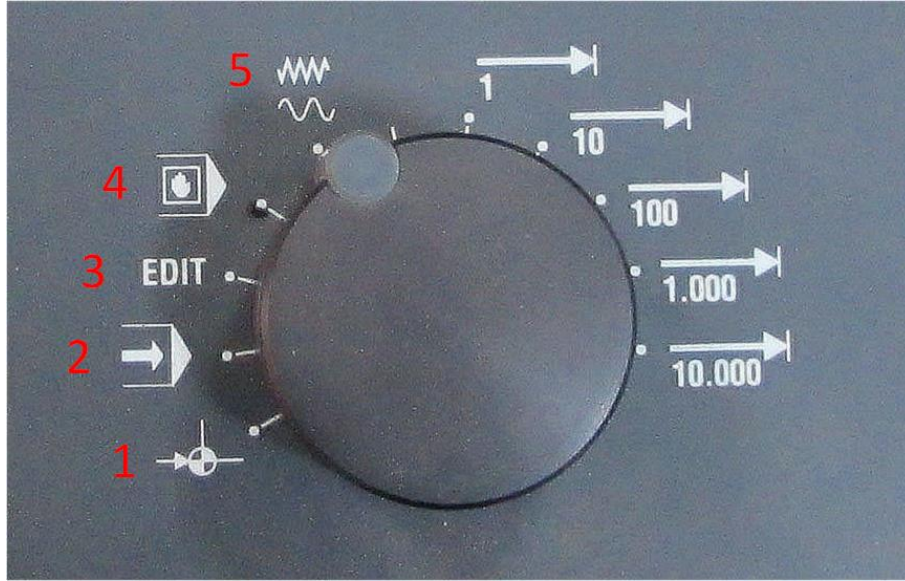


Şekil 3-41. FMS100 bünyesindeki CNC tezgahlarına ait bilgisayar ve kontrol klavyesi görünümü.

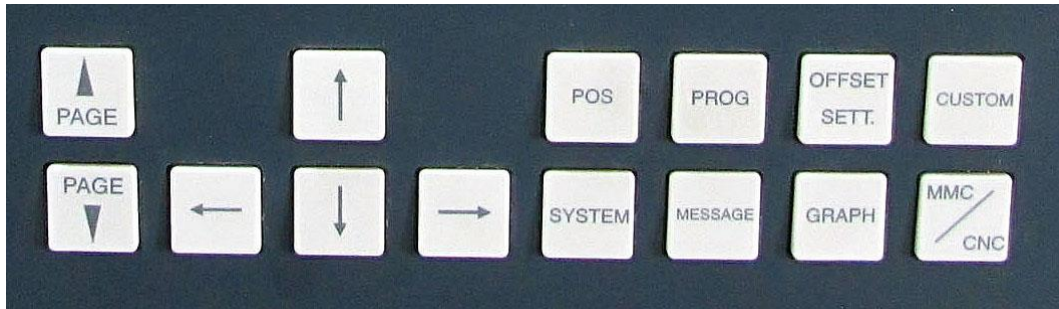
“DNC”, “Direct Numerical Control” veya “Distributed Numerical Control” kelimelerinin kısaltılmışı olup, 1990’lı yılların başında CNC tezgahlarına ağ üzerinden uzaktan erişebilmek için geliştirilmiş bir arayüzdür [81]. Her iki tezgah da sahip oldukları “DNC” arayüzü sayesinde RS232 bağlantısı üzerinden uzaktan erişime imkan vermektedir. Özel bir yazılım kullanarak veya FMS100’ün SCADA yazılımı üzerinden tezgahların tüm özelliklerini kullanabilmek; tezgaha program yüklemek, düzenlemek, çalıştırmak ve durdurmak mümkündür. Bu nedenle her iki tezgah da FMS100 sistemi ile tam uyumlu çalışabilir. DNC arayüzünü aktif/pasif hale getirmek için tezgahlarda “FMS mode” seçim anahtarı vardır. Şekil 3-42’de tezgahları ve FMS modunu açmak/kapamak için kullanılan anahtarlar görülmektedir. FMS modu kapalıyken tezgah tamamen kendi bilgisayarı üzerinden kullanılır. FMS modu açıldığında ise DNC arayüzü aktif olur ve tezgah ister FMS100 SCADA yazılımından ister özel DNC yazılımından komut alabilir.

3.1.6.6. Turn55 ve Mill55 CNC Tezgahlarının Kullanımı

CNC tezgahlarının kullanımı ve önceden hazırlanmış bir NC programının yürütülmesi süreci bu bölümde aşamalar halinde anlatılmıştır. Birçok aşamada kullanılacak ve referans verilecek olan kontrol klavyesinin çalışma tipi seçim anahtarı Şekil 3-44’de görüldüğü gibi numaralandırılmıştır. Ayrıca programlama klavyesinin önemli tuşları da Şekil 3-45’de görüldüğü gibidir.



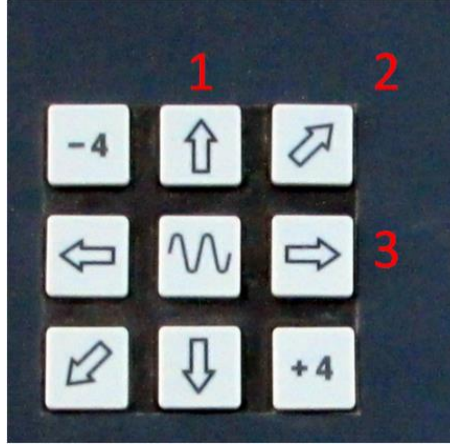
Şekil 3-44. CNC kontrol klavyesi çalışma tipi seçim anahtarı görünümü.



Şekil 3-45. CNC programlama klavyesinin sık kullanılan tuşları görünümü.

3.1.6.6.1. Tezgah Eksenlerini Sıfırlamak:

Tezgah tüm eksenlerde başlangıç (referans) noktasına gönderilir. Bunun için çalışma tipi seçim anahtarı Şekil 3-44 (1)’deki gibi başlangıç moduna alınır. Eğer takım başlangıçta parçanın yüzey seviyesinin altındaysa ve başlangıç hareketi yaparsa, parçaya çarpabilir. Bu durumda ya takım ya da parça zarar görebilir. Bunu önlemek için, önce Şekil 3-46’deki 1 numaralı butona basılarak Z eksen, sonra diğer eksenler (2, 3) sıfırlanır.



Şekil 3-46. Kontrol klavyesi üzerindeki eksen tuşları: Z (1), Y (2), X (3).

3.1.6.6.2. Kesici Takım Boy/Çap Telafi Ayarlarını Yapmak:

CNC tezgahında farklı boy ve çapta kesici takım kullanılabilir. Bu nedenle her kesici takım için tezgahın programda vermesi gereken pay (offset) değerleri tezgah parametrelerinde tanımlanmalıdır.

3.1.6.6.3. Kesici Takımı Eksenlerde Serbest Hareket Ettirmek (JOG):

Kullanıcı iş milini CNC tezgahının sahip olduğu eksenlerde istediği yöne hareket ettirebilir. Bunun için çalışma tipi seçim anahtarı Şekil 3-44 (5)'deki serbest hareket (JOG) moduna alınır. Sonrasında Şekil 3-46'deki ok butonları kullanılarak iş mili istenilen eksenle hareket ettirilebilir.

3.1.6.6.4. Tezgaha NC Programı Yükleme:

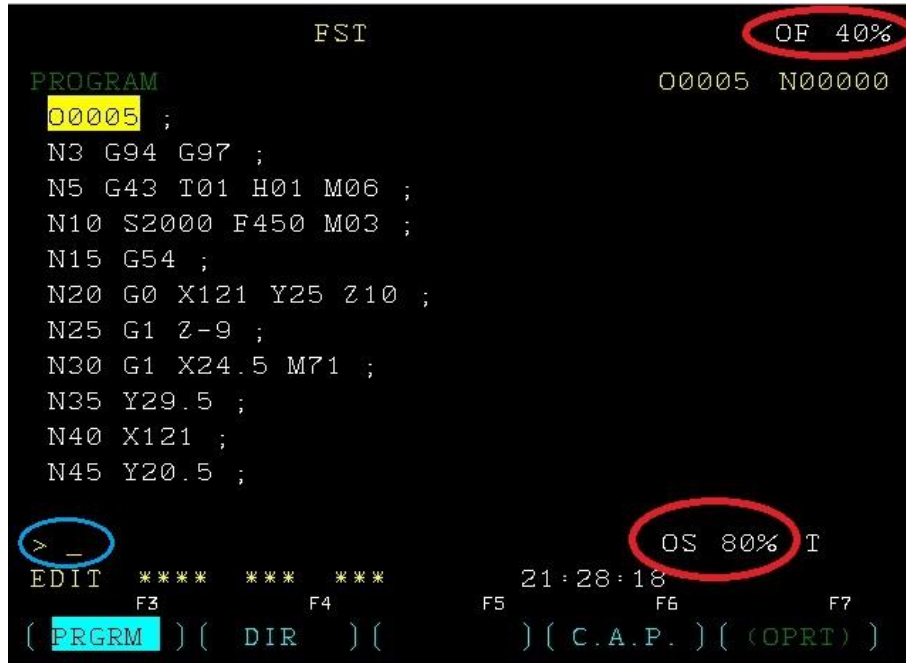
NC programına ait G kodlar tezgaha yüklenir. FMS100 bünyesindeki Tun55 ve Mill55 tezgahlarında yüklü olan programlar, tezgaha ait bilgisayardaki "C:\Winnc32\Fanuc21.m\Prg\" klasöründe saklanır. Önceden hazırlanmış programlar bu klasöre kopyalanır. Program isimleri mutlaka büyük "O" harfi ile başlar ve sonrasında sayısal değer yazılır. Örneğin "O30", "O42" vb. Başka şekilde isim verilemez.

3.1.6.6.5. Yeni Bir NC Programı Oluşturmak

Yeni bir NC programı oluşturabilmek için öncelikle çalışma tipi seçim anahtarı Şekil 3-44 (3)'deki gibi düzenleme (EDIT) moduna alınır. Sonrasında programlama klavyesinde Şekil 3-45'de görülmekte olan "PROG" tuşuna basılır. Bilgisayar klavyesinin "F4" tuşuna basılarak tezgahta yüklü olan programlar görüntülenir. Şekil 3-47'de mavi çizgili daire içine alınmış bölüme program ismi yazılır ve bilgisayar klavyesinin "Enter" tuşuna basılır. Ekranda boş bir programlama sayfası görüntülenir

3.1.6.6. NC Programı Görüntülemek ve Değiştirmek (EDIT):

Tezgahta yüklü bulunan NC programları ve program içeriklerini görüntüleyebilmek için, önce çalışma tipi seçim anahtarı Şekil 3-44 (3)'deki gibi düzenleme (EDIT) moduna alınır. Sonrasında programlama klavyesindeki Şekil 3-45'de görülmekte olan aşağı ok butona basıldığında aktif olan program içeriği görüntülenir. Örneğin tezgahta aktif olan "O5" isimli program içeriği Şekil 3-47'deki gibi görüntülenir. Ok tuşlarına basılarak kod satırları arasında gezinti yapılabilir.



Şekil 3-47. Mill55 tezgahında yüklü "O5" isimli NC programı ekran görüntüsü.

NC program klasöründe yüklü başka bir programı seçebilmek için Şekil 3-45'deki PRG tuşuna yeniden basılır. Ok tuşları kullanılarak bilgisayarda yüklü programlar arasında gezinti yapılabilir. Program seçmek için bir başka yöntem ise imleç Şekil 3-47'de mavi renk çember içine alındığı gibi ekranın sol alt köşesindeyken "O5" yazıp, programlama klavyesinde aşağı ok tuşuna basmaktır.

3.1.6.6.7. NC Programın Başlangıç Satırını Seçmek:

NC programı ilk satırdan başlamalıdır gibi bir kural yoktur. Örneğin önceden başlanmış ve bir şekilde yarıda kalmış bir program, kaldığı yerden itibaren yürütülebilir. Bunun için önce tezgah çalışma tipi seçim anahtarı Şekil 3-44 (3)'deki gibi düzenleme (EDIT) moduna alınır. Sonrasında Şekil 3-45'de görülmekte olan yukarı/aşağı ok tuşları kullanılarak, program içerisinde başlanmak istenilen satır aktif edilir. Eğer programı en

başından itibaren başlatmak isteniyorsa, ok tuşlarıyla ilk satıra gidilir veya Şekil 3-48 (1)'deki sıfırlama (reset) butonuna basılabilir.



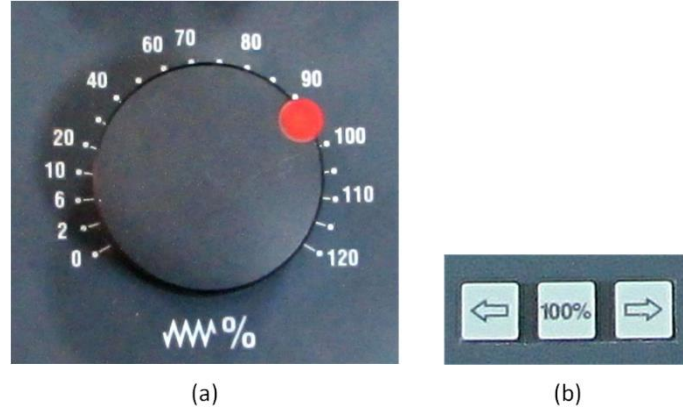
Şekil 3-48. Kontrol klavyesi program kontrol tuşları görünümü.

3.1.6.6.8. Kesici Takım İlerleme/Dönme Hız Oranını Değiştirmek:

CNC tezgahının keserek ilerleme ve iş mili devir sayıları, kullanılacak malzemenin cinsi ve takımın özelliklerine göre önceden hesaplanır. Hesaplanan hız değerleri NC programı içerisinde kullanılan kodlarla (F, S) tanımlanır. Bazı durumlarda NC program içerisinde tanımlanmış olan hız değerlerini değiştirmeden, tezgahın keserek ilerleme ve iş mili dönüş hızlarını değiştirmek gerekebilir. Özellikle yazılan NC programı ilk kez çalıştırılacaksa, programı normalden yavaş çalıştırmak, programdaki olası hataların tespiti açısından fayda sağlayacaktır. Eğer programın işleyişinde bir hata yoksa hız oranları %100'e getirilir. Böylece NC programı içinde tanımlanmış olan hızlarda tezgah hareket, iş mili döner eder.

CNC tezgahlarda genellikle iki tip hız oranı (override) ayarı vardır. Biri keserek ilerleme hızı oranı (override feedrate - OF) diğeri ise iş mili devri baskılama oranıdır (override spindle - OS). Bu oranlar NC programda tanımlanmış olan hız değerlerine uygulanarak anlık tezgah hızları hesaplanır. Örneğin NC programda keserek ilerleme hızı F100 (100 mm/dk) ve iş mili devir sayısı 500 devir/dk "S500" olarak tanımlanmış olsun. OF%50, OS%30 şeklinde ayarlandığında, tezgahın anlık keserek ilerleme hızı $100 \times \%50 = 50$ mm/dk; anlık iş mili dönüş hızı ise $500 \times \%30 = 150$ devir/dk olacaktır.

FMS100 bünyesindeki CNC tezgahlarda keserek ilerleme hızı oranı Şekil 3-49 (a)'daki döner anahtardan, iş mili hızı oranı ise Şekil 3-49 (b)'deki butonlardan ayarlanabilir.

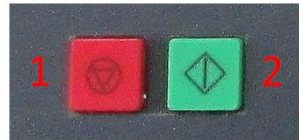


Şekil 3-49. Kesici takım hız oranı ayarları görünümü: ilerleme (a), dönme (b).

Şekil 3-47’de kırmızı renk çemberler içinde görüldüğü üzere ekranın sağ üst köşesinde “OF” ve sağ alt tarafta “OS” ile anlık hız yüzdelik oranları görüntülenir.

3.1.6.6.9. NC Programı Satır Satır Çalıştırmak:

Program istenirse satır satır çalıştırılabilir. Özellikle yazılan NC programı ilk kez çalıştırılacaksa, programı adım adım çalıştırmak olası hataların tespitini kolaylaştırır. Bunun için önce çalışma tipi seçim anahtarı Şekil 3-44 (2)’deki gibi otomatik (Auto) moduna getirilir. Şekil 3-48 (6)’deki tek satır çalıştırma (single block operation - SBL) tuşuna basılır. Ekranda SBL ifadesi çıkar. Bu durumdayken Şekil 3-50 (2)’deki Başlat butonuna her basıldığında sadece bir satır yürütülür.



Şekil 3-50. CNC kontrol klavyesindeki program başlat/durdur tuşları.

3.1.6.6.10. NC Programı “Kuru” Çalıştırmak:

Tezgaha parça bağlamadan yapılan çalıştırma yöntemine denir. İlk kez çalıştırılacak bir NC programda kesici takımın hareketlerinin gözlenmesinde ve olası hataların tespitinde faydalı yöntemlerden biridir. Bu moddayken tezgah parça bağlanmadan boşta çalışacağı için, iş milini çalıştırmaz ve eğer NC programı içerisinde soğutma havası/suyu açma/kapama komutları varsa; bu komutları çalıştırmaz. Kuru çalışma yapabilmek için önce çalışma tipi seçim anahtarı Şekil 3-44 (2)’deki gibi otomatik (Auto) moduna getirilir. Şekil 3-48 (4)’deki “Dry Run” tuşuna basıldığında ekranın sol üst köşesinde “DRY” uyarısı görüntülenir. Şekil 3-50 (2)’deki Başlat butonuna basıldığı anda kuru çalışma başlar.

3.1.6.6.11. NC Programı Çalıştırmak/Durdurmak:

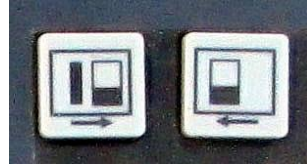
Programı çalıştırmak için çalışma tipi seçim anahtarı Şekil 3-44 (2)'deki gibi otomatik (Auto) moduna getirilir. Sonrasında Şekil 3-50 (2)'deki Başlat butonuna basılır. Şekil 3-50 (1)'deki Durdur butonuna basılarak program herhangi bir anda durdurulabilir. Tekrar Başlat butonuna basılırsa program en son kaldığı satırdan çalışmaya devam eder.

3.1.6.6.12. Serbest Kod Çalıştırmak:

Kullanıcı, herhangi bir program oluşturmadan istediği G kodları yazarak tezgahta çalıştırabilir. Bunun için çalışma tipi seçim anahtarının Şekil 3-44 (4)'deki gibi serbest kod çalıştırma (MDI) moduna alınması gerekir. Sonrasında programlama klavyesi kullanılarak istenilen G kodlar yazılır. Şekil 3-50 (2)'deki Başlat butonuna basılmasıyla birlikte G kodlar tezgah tarafından çalıştırılır.

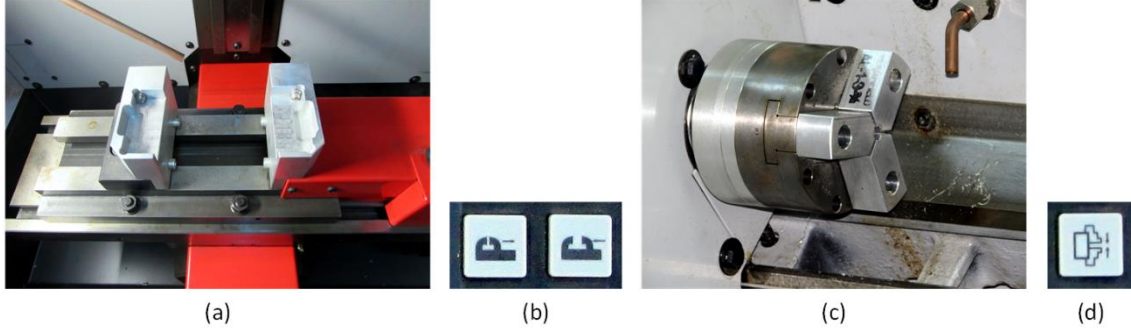
3.1.6.6.13. Tezgaha Parça Bağlamak:

NC programı tezgaha yüklenip, düzenlenip denendikten sonra sıra asıl imalata gelmiştir. Bunun için tezgaha işlenecek parça bağlanmalıdır. Talaşlı imalat sırasında etrafı kirletmemek ve iş güvenliğini arttırmak için tezgah kapalı bir kabin içerisinde. Bu nedenle parçayı yerleştirebilmek için önce kabin kapısının açılması, parçanın bağlanması ve sonrasında kapının kapatılması gereklidir. FMS100 bünyesindeki CNC tezgahlarında kapılar (door) Şekil 3-51'deki butonlar kullanılarak açılır ve kapanır.



Şekil 3-51. CNC kontrol klavyesindeki tezgah kapısı açma/kapama tuşları.

Freze tezgahlarında parçanın bağlanarak sabitlendiği yere mengene (clamp), Şekil 3-52 (a); torna tezgahlarında ise ayna (chuck), Şekil 3-52 (c), adı verilir. Mill55 tezgahında mengene Şekil 3-52 (b)'deki butonlara basılarak açılıp kapatılabilir. Turn55 tezgahında ise ayna Şekil 3-52 (d)'deki butona basılarak açılıp kapatılabilir.



Şekil 3-52. Tezgahlardaki mengene ve ayna ile kontrol tuşları görüntüsü: mengene (a), mengene açma/kapama tuşu (b), ayna (c), ayna açma/kapama tuşu (d).

Torna aynasına parça bağladıktan sonra aynanın el ile bir miktar döndürülmesinde ve parçanın yalpa yapıp yapmadığının gözlenmelidir. Gerekirse parça sökölüp yalpa yapmayacak şekilde yeniden bağlanmalıdır.

3.1.6.6.14. Parça Üzerinde Program Başlangıç Konumunu Belirlemek:

Tezgaha parça bağlandıktan sonra, parçanın da sıfır (başlangıç, orijin) noktasının belirlenmesi gerekir. NC programı kullanıcı tarafından sıfırlanan konumu başlangıç olarak kabul edecek ve ona göre çalışacaktır. Parça sıfırlama yapmak için önce çalışma tipi seçim anahtarı Şekil 3-44 (5)'deki serbest hareket (JOG) moduna alınır. Sonrasında Şekil 3-46'daki ok tuşları kullanılarak kesici takım eksenlerde parçaya değene kadar hareket ettirilir.

Freze tezgahına bağlanan parça için önce Z ekseninde sıfırlama yapılır. İş mili belirli bir devirde çalıştırılarak kesici takım parçanın yüzeyine hassas biçimde değdirilir ve Z eksenini sıfırlanır. Burada iş milinin döndürölme sebebi, yüzeye değme denemeleri sırasında kesici takımın zarar görmesini önlemektir. Sonrasında parçanın sol alt tarafında uygun görölün bir noktaya gidilir ve XY eksenleri de sıfırlanır.

Torna tezgahında da benzer bir yöntem geçerlidir. İş mili belirli bir hızda döndürölme başlanır. İlk önce Z eksenini sonra X ekseninde parçaya değerek eksen sıfırları belirlenir.

3.1.6.6.15. Soğutma Havasını Açmak/Kapamak:

CNC tezgahlarında kesici takım ömrünün uzaması ve takımın sağlıklı çalışabilmesi için soğutulması önemlidir. Bunun için tezgahın marka modeline göre soğutma havası ve/veya sıvısı kullanılır. FMS100 bünyesindeki tezgahlarda soğutma sıvısı kullanma özelliği yoktur. Soğutma için sadece hava kullanılabilir. Soğutma havasının açılıp

kapanması NC programa yazılan M kodlar veya Şekil 3-53'deki buton kullanılarak gerçekleştirilebilir.

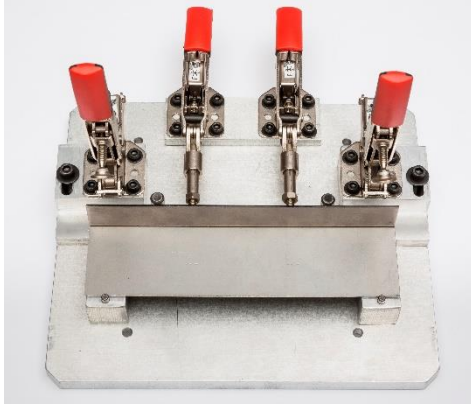


Şekil 3-53. CNC kontrol klavyesindeki soğutma havası açma/kapama tuşu.

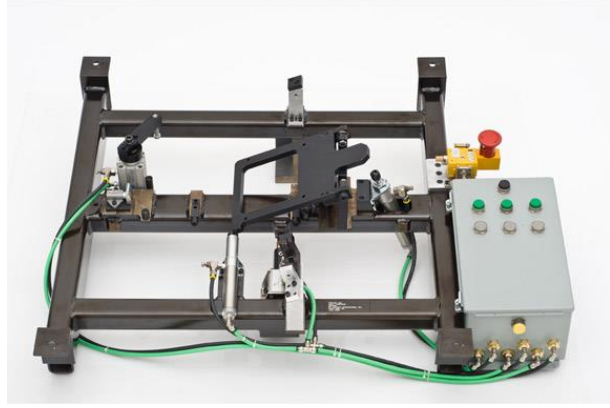
3.1.7. Endüstriyel Robot Kullanımı ve Programlanması

Robotlar tekrarlanabilirlik konusundaki başarıları ve hassasiyetleri nedeniyle endüstride geniş kullanım alanına sahiptir. Özellikle imalata yönelik fabrika otomasyon sistemlerinde kaynak, montaj, boyama (sıvı, akışkan malzeme püskürtme) ve CNC tezgahlarının besleme/boşaltma işlerinde sıklıkla kullanılırlar. Günümüzde endüstriyel robotlar çoğunlukla yurtdışından hazır ürün olarak ithal edilmektedir. Bu nedenle Mekatronik Mühendisliği bölümü mezunlarının robot alanında yaptıkları temel işler; genel anlamda robotların imalat sistemine dahil edilmesi (entegrasyon) odaklı olan fikstür tasarımı, tutucu el (gripper) tasarımı, robot programlanması, imalat hücresi tasarımı ve otomasyonu olarak özetlenebilir.

Fikstür tasarımı, robotun imalat sırasında işleyeceği parçayı sabitlemek için kullanılacak mekanizmaların tasarlanması işidir. Fikstür çoğunlukla robotlu kaynak işleri için, kaynak edilecek parçanın sabitlenmesi amacıyla tasarlanır. Bu nedenle kaynak yapılacak veya işlenecek parça değiştikçe, fikstürün tasarımı da değişmektedir. En basit fikstür örneği Şekil 3-54 (a)'da görüldüğü gibi, bir parçayı sıkıştırarak tutmaya yarayan mingeneler olabileceği gibi; bünyesinde elektromekanik, pnömatik, vb. sistemler bulunan karmaşık yapıları fikstürler de olabilir. Örneğin Şekil 3-54 (b)'de, bir egzersiz aletine ait parçanın imalatına yönelik kaynak işleri için özel olarak tasarlanmış, pnömatik tutuculara sahip gelişmiş bir fikstür görülmektedir.



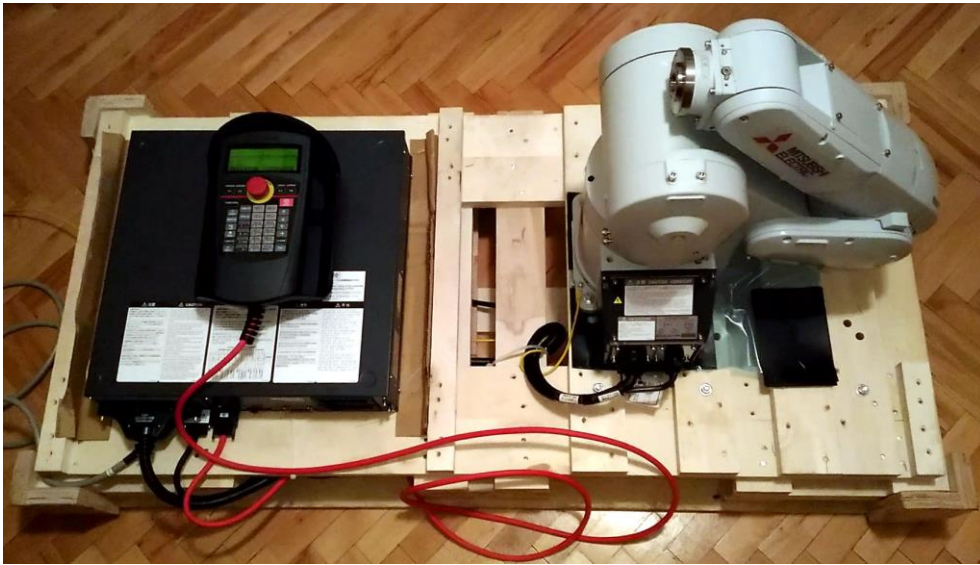
(a)



(b)

Şekil 3-54. Parçayı sıkıştırarak tutmak için tasarlanmış basit bir kaynak fikstürü [82] (a); pnömatik tutuculara sahip gelişmiş bir kaynak fikstürü görüntüleri [83].

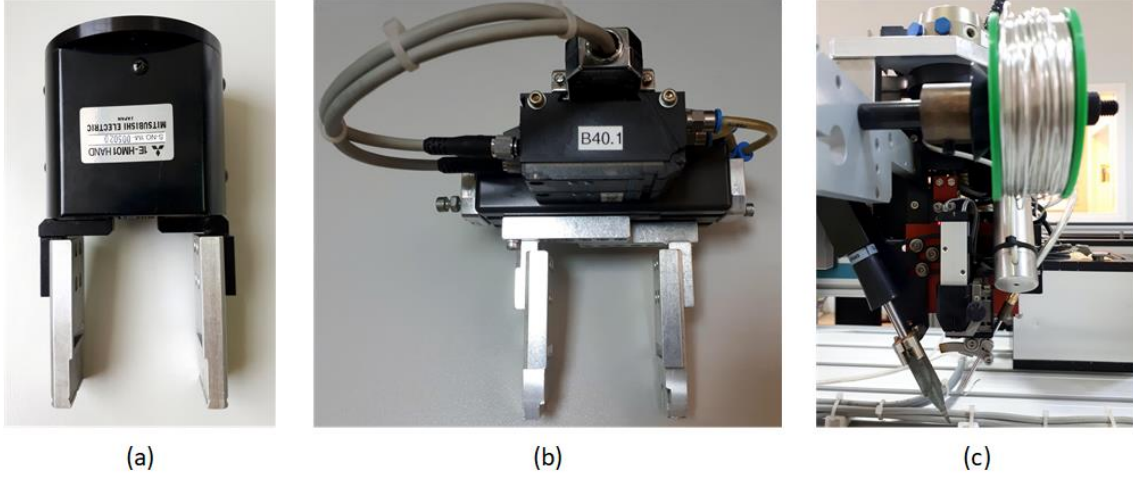
Endüstriyel robotların fabrika çıkışında herhangi bir tutucu eli yoktur ve bileğe kadar üretilirler. Şekil 3-55’de yeni kutu açılışı yapılmış ve fabrika çıkışında ahşap bir taşıma paleti üzerine sabitlenmiş bir robot ile sürücüsü ve el terminali (teach pendant) görülmektedir.



Şekil 3-55. Yeni kutu açılışı yapılmış olan RV4-FLM model robot görüntüsü.

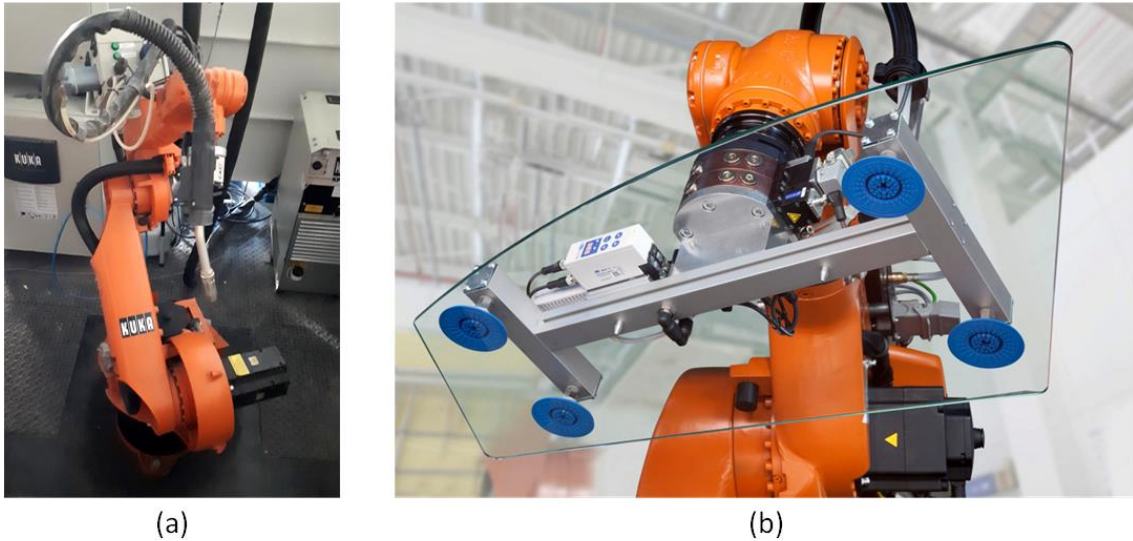
Robotun kullanılacağı uygulamaya özel el tasarımı yapılması gerekir. Tasarlanan el ile robot bir parçayı tutabilir (gripper); kaynak, lehimleme, boyama işleri yapabilir; parça kesip işleyebilir. Şekil 3-56’da FMS100 bünyesindeki robotlarda kullanılan üç farklı el örneği görülmektedir. (a) ve (b)’de bulunan eller parça tutmak için, (c)’de bulunan el ise

lehim yapmak için tasarlanmıştır. Ayrıca (a)'da kullanılan elde tutma kuvveti fırçalı DC motor ile sağlanırken, (b)'deki elde pnömatik piston kullanılmıştır.



Şekil 3-56. FMS100 bünyesindeki robotlarda kullanılan üç farklı el örneği görüntüleri; elektrik tahrikli tutucu el (a), pnömatik tahrikli tutucu el (b) ve lehimleme eli (c).

Bunlar dışında yine Marmara Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği bölümü CNC laboratuvarında bulunan ait kaynak robotu Şekil 3-57 (a)'da görülmektedir. Robotta gaz altı kaynak yapmaya yarayan el (takım) bulunmaktadır. Şekil 3-57 (b)'de ise araba camı gibi hassas bir parçayı taşımak için tasarlanmış vakumlu vantuzlara sahip bir tutucu el görülmektedir.



Şekil 3-57. Kaynak robotu ve gaz altı kaynak eli (takımı) görüntüsü (a); araba camı taşımak için tasarlanmış vakumlu vantuzlu tutucu el görüntüsü (b).

Robotun kullanılacağı yere sabitlenmesi de önemli bir konudur. Bunun için robotun çalışırken tabanına etkiyen momentlere dayanacak şekilde montajı yapılmalıdır. Robotun mimarisi ve imalat hücrelerinde yürüteceği görevler göz önüne alınarak robotu sabitleme mekanizmaları tasarlanmalıdır. Saplamalar kullanılarak robot direk zemine, masa gibi ayaklı bir düzeneğe veya doğrusal kızak gibi hareketli bir zemin üzerine monte edilebilir.

Örneğin FMS100 Montaj hücrelerinde bulunan robot Şekil 3-58 (a)'da görüldüğü gibi masa üzerine sabitlenmiştir. Burada kullanılan masa ile ilgili önemli bir ayrıntı vardır. Dikkat edilirse masanın alt tarafında dört bir tarafındaki ayakları arasına kirişler eklenmiştir. Bir anlamda alt tarafta ayaklar arasında bir çerçeve oluşturulmuştur. Robot masanın üzerinde hareket ederken oluşan momentler nedeniyle masanın salınımını azaltmak için özellikle yapılmıştır.

Masanın yüzeyinde bulunan oluklar arası mesafe ile robot tabanındaki bağlantı delikleri arasındaki mesafeler tam olarak örtüşmemektedir. Bu nedenle de Şekil 3-58 (b)'de görüldüğü üzere robot tabanıyla masa arasına bir parça tasarlanmıştır. Bu parça üzerinde hem robot tabanındaki delik mesafelerine uygun, hem de oluklar arası mesafelere uygun delikler bulunmaktadır. Böylelikle ara parça önce robot tabanına alt taraftan sabitlenir. Sonrasında üstteki bağlantı vidaları ile ara parça, üzerinde robot bulunur şekilde, masa üzerine sabitlenir. FMS100 bünyesindeki Lehimleme ve Görüntü İşleme hücrelerinde de robotlar aynı yöntemle masalara sabitlenmiştir.



(a)



(b)

Şekil 3-58. FMS100 Montaj hücrelerine robotun montaj görüntüleri: robotun üzerine sabitlenen masa (a), robotun masa üzerine sabitleniş biçimi (b).

Diğer bir örnekte ise FMS100 CNC-Çifti hücresinde bulunan RV-2AJ robot, Şekil 3-59’da görüldüğü gibi doğrusal bir kızak üzerine sabitlenmiştir. Bu sayede robot, imalat hücresindeki kızak üzerinde yaklaşık 3.000 mm strokta hareket edebilmektedir. Her iki CNC cihazına da parça yükleme ve boşaltma işlerini tek bir robot yürütebilmektedir.



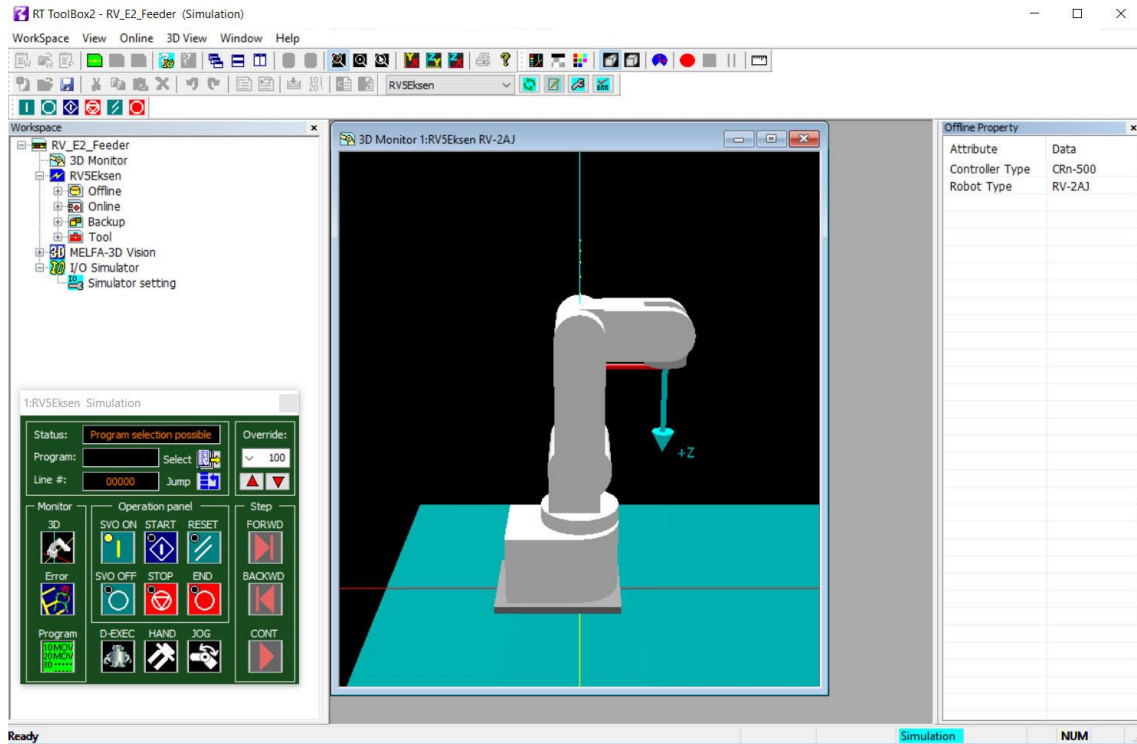
Şekil 3-59. FMS100 sistemindeki CNC-Çifti istasyonunda bulunan ve doğrusal bir kızak üzerinde 3.000 mm strok ile hareket edebilen robot görüntüsü.

Mekanik montajın bitmesinden sonra robotlar, bünyesinde yer alacakları imalat istasyonunda bulunan otomasyon sistemine dahil (entegre) edilmelidir. Robota ait elektriksel bağlantılar temelde güç hatları ve otomasyon bağlantıları olmak üzere iki bölüme ayrılır. Robotun modeline göre eklemlerinde yüksek torklara sahip motorlar kullanıldığı için, robotlara tek faz 220VAC veya üç faz 380VAC gerilim ile güç sağlanır. Otomasyon tarafında ise robotun imalat hücresinde yer alan çeşitli algılayıcılar ve eğer varsa, istasyon denetleyicisiyle bağlantı kurulması gereklidir. İmalat istasyonlarında denetleyici olarak genellikle PLC kullanılır. Robotun denetleyicisi ile PLC arasında haberleşmenin sağlanması için, haberleşme protokolüne uygun elektriksel bağlantıların kurulması gerekir. Güncel robot denetleyicileri klasik paralel ve seri iletişimin yanı sıra

Ethernet, Profibus gibi ileri seviye yüksek hızlı iletişim protokollerini de desteklemektedir. Özetle robotun imalat sistemine elektriksel olarak dahil olabilmesi için, güç hatları ile otomasyon bağlantılarını düzenleyen pano tasarımı yapılmalıdır.

Robotlar imalat sistemine mekanik ve elektriksel olarak eklendikten sonra, sıra robotun yapacağı işe göre “eğitilmesi” programlanması sürecine gelir. Robotun eğitime öncelikle program içinde uğraması istenilen noktaların tek tek öğretilmesiyle başlanır. Gerek el terminali gerekse bilgisayar yazılımı kullanılarak robot istenilen noktalara tek tek götürülür ve noktalar kaydedilir.

Sonrasında öğretilen noktalar arası ilişki, yazılacak program kodları ile tanımlanır. Uygulamada kullanılan robotun marka ve modeline göre program kodlama dili değişiklik gösterir. Örneğin Mitsubishi marka robotlar Q-Basic dili tabanlı Movemaster Command, MelfaBasic IV veya MelfaBasic V gibi diller ile programlanabilir. Her üreticinin robot programlanması için geliştirdiği kendi bilgisayar arayüz programı (Integrated Development Environment - IDE) vardır. Örneğin Şekil 3-60’da Mitsubishi marka robotların programlanabilmesi için kullanılan RT Toolbox2 yazılımına ait ekran görüntüsü yer almaktadır.



Şekil 3-60. Mitsubishi marka robotları programlamak için kullanılan RT Toolbox2 isimli yazılıma ait ekran görüntüsü.

RT Toolbox2 yazılımında robota ait tüm parametrelere ulaşılabilirdiği gibi, robotun programlanabilir, ekrandaki animasyon üzerinden programın benzetim işleri yürütülebilir. Ayrıca hazırlanan robot programı robota yüklenebilir, robotta yüklü bulunan programlar yazılıma çekilebilir, programlar üzerinde değişiklik yapıp robota yeniden yüklenebilir. Yazılım üzerindeki kullanıcı penceresinden (operatör panel) istenilen robot programı başlatılabilir, durdurulabilir, yeniden başlatılabilir, robot hareket hızı değiştirilebilir

Sonuç olarak herhangi bir robotun, kutusundan çıkartılıp, herhangi bir imalat hücresine mekanik montajından elinin ve fiktürün tasarımına, otomasyon tasarımı yapılarak elektriksel olarak güç ile sinyal bağlantılarının yapılmasına ve programlanarak hazır hale getirilmesi işlerinin tümüne robot entegrasyonu denir. Gelecekte hazır servo sistemler, eklem dişli grupları (harmonic drive) ve gömülü robot denetleyicileri kullanarak kişisel robot tasarımı yapılması da önemli bir iş alanı olacaktır.

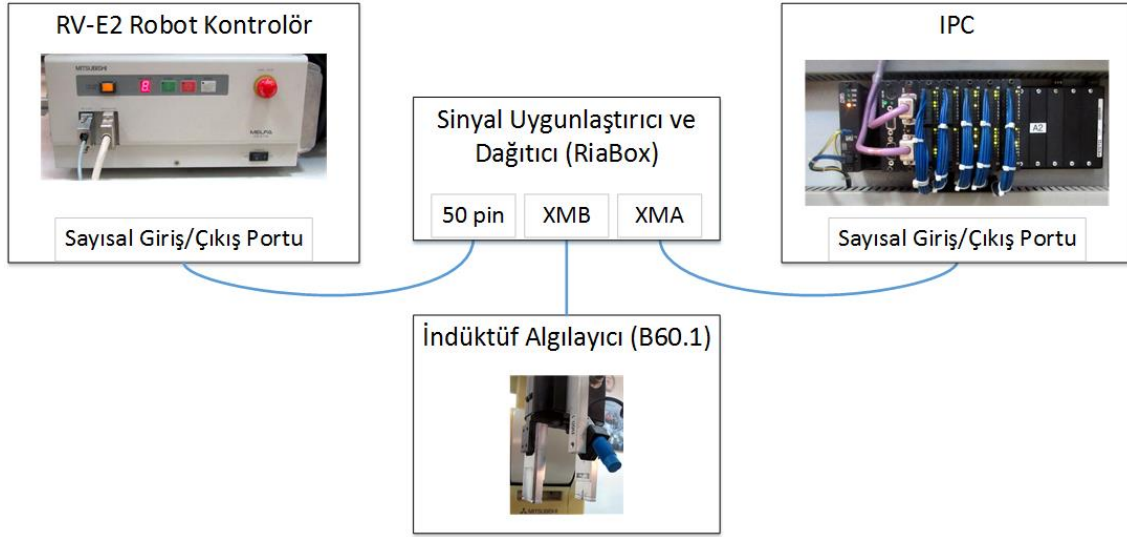
Sunulmakta olan bu çalışmanın üzerinde yürütüldüğü FMS100 sistemi bünyesinde toplam 4 adet endüstriyel robot bulunmaktadır. Bunlardan biri 5 eksenli RV-2AJ ve diğerleri ise 6 eksenli RV-E2 modeldir. RV-E2 model robot diğerine göre daha eski olduğu için programlanmasında Movemaster Command dili kullanılmaktadır. RV-2AJ robot ise daha yeni bir model olup MelfaBasic IV dili ile programlanmaktadır. Bu nedenle bu bölümde her iki dil ile programlama ayrı ayrı anlatılacaktır.

3.1.7.1. PLC ile Robot Kontrol Arayüzü Bağlantısı

PLC'ler ile robotlar arası iletişimi sağlamak için çeşitli yöntemler vardır. Bunlardan en temel ve eski olanı paralel giriş/çıkış birimlerini kullanmaktır. Oldukça fazla kablo bağlantısı gerektiren bu yöntemde haberleşme hızı çok yüksek olup neredeyse her iki sistemde bulunan işlemcilerin çalışma hızı kadardır denebilir. Bir nevi eş zamanlı çalışmadan söz edilebilir. Yeni nesil robot denetleyiciler üzerinde dahili olarak Ethernet, Profibus vb. gelişmiş endüstriyel haberleşme bağlantı noktaları vardır. Temelleri seri iletişim teknolojisine dayanan bu yöntemler kullanılarak PLC ile robot denetleyici arasındaki iletişim tanımlı protokoller üzerinden kolaylıkla sağlanabilir.

FMS100 sisteminde toplam dört robotik imalat hücresi bulunmaktadır. Hücrelerin üçünde RV-E2 model robot bulunmakta olup hücrenin otomasyonu PLC/IPC denetiminde yürütülmektedir. PLC/IPC'ler ile RV-E2 model robotlar arasındaki iletişim yapısına

örnek olarak Montaj hücresindeki yapı verilebilir. Şekil 3-61’de görüldüğü gibi PLC/IPC tarafında sayısal giriş çıkışlar, RV-E2 robot tarafında ise kontrol arayüzü üzerinden bağlantı gerçekleştirilir. Birbirinden ayrı olarak farklı firmalar tarafından farklı teknolojiler kullanılarak üretilmiş her iki sisteme ait sayısal G/Ç arası elektriksel uyumu sağlamak için RiaBox adı verilmiş olan özel cihazlar kullanılmıştır.



Şekil 3-61. Montaj istasyonundaki robot ve çevre birimleri arası bağlantı şeması.

Montaj istasyonu bağlantı şemasına göre robot 50 pin kablo ile RiaBox’a bağlanmıştır. Robot tutucu ucunda takılı olan ve montaj sırasında, program içinde kullanılmak istendiğinde metal pinleri algılamada kullanılan indüktif algılayıcı (B60.1) RiaBox’ın XMB isimli portu üzerinden robot kontrolörün 4 numaralı sayısal girişine (IN4) bağlanmıştır. IPC ise RiaBox’ın XMA isimli portuna bağlanmıştır. Böylelikle hem cihazlar arası ilgili pinler arası elektriksel bağlantılar kurulmuş, hem de sinyal uyumu sağlanmıştır.

RV-E2 robot kontrolör, endüstriyel algılayıcı ve PLC gibi harici cihazlar arası iletişim kontrol arayüzü kartı ile sağlanır. RV-E2 robot kontrolörde fabrika çıkışlı olarak 1 adet kontrol arayüz kartı vardır ve kart sayısı 3’e kadar arttırılabilir. Her bir arayüz kartı sayısal 20 adet giriş ve 16 adet çıkışa sahiptir. Kart üzerindeki giriş ve çıkışların sahip olması istenen özellikler de robot parametre ayarları üzerinden belirlenir.

FMS100 imalat hücrelerindeki robot denetleyici ile PLC/IPC arası iletişim de kontrol arayüzü üzerinden sağlanır. Genel anlamda kontrol arayüzü üzerinden yürütülen işlevler şunlardır:

- Robotun başlangıç (reference) konumuna hareketinin başlatılması,
- Robot kontrolörde yüklü bulunan 16 adede kadar farklı programdan herhangi birinin seçilmesi,
- Robota program başlat / durdur / başa-dön komutlarının verilmesi,
- Robotun durumunun sorgulanması (program yürütülüyor, bekliyor, durdu, tamamlandı, alarm var, vb.).

Kontrol arayüzünün kullanılabilmesi için robot kontrolör el-kitabında [[84]] detayları belirtilen şekilde yapılandırma ayarlarının yapılması gereklidir. Tablo 3-2’de görüldüğü üzere kontrol arayüzündeki dört adet sayısal giriş biti, program seçiminde kullanılmak üzere ayarlanmıştır. Olasılık teorisi gereği dört bit (IN0 – IN3) ile $2^4 = 16$ farklı program seçilebilir. Bu nedenle PLC/IPC üzerinden çalıştırılmak istenen programlar, robot denetleyici içerisinde 0-15 arası sayılar ile isimlendirilmelidir.

Robotta yürütülmek istenen program numarası IN0-IN3 bitleri ile belirlendikten sonra PLC’den IN16’ya gönderilen sinyal (strobe) ile program seçim işlemi tamamlanır. Bu andan itibaren sıra yine PLC’den gönderilecek Program Başlat, Durdur, Başa-dön komutlarına gelmiştir. Bu komutlar da sırasıyla yine PLC’nin ilgili sayısal çıkışlarından robot kontrolör IN17, IN18 ve IN19 girişlerine gönderilecek pozitif sinyaller ile aktif edilir.

Sayısal girişlere uygulanacak sinyalin çeşidi yürütülmek istenen işleve göre değişiklik gösterir. Bazı işlevlerin yürütülebilmesi için ilgili sayısal girişin sürekli aktif (mantıksal 1) olması gereklidir. Giriş pasif (mantıksal 0) duruma getirildiğinde işlevin yürütülmesi de durdurulur. Bazı işlevler ise ilgili sayısal giriş pasif durumdan aktif duruma geçişlerde yürütülmeye başlar. Sonrasında giriş sinyalinin aktif veya pasif olmasının bir önemi yoktur. Bu tip giriş sinyaline yükselen kenar tetikleme adı verilir. RV-E2 model robotta yükselen kenar tetikleme işlevinin yürütülebilmesi için tetik sinyali periyodunun en az 100 ms olması gereklidir [[84]].

Örneğin Tablo 3-2’de açıklanan IN16 girişi yükselen kenar ile tetiklenir. Bu nedenle program başlat işlevi için girişin pasiften aktif duruma getirilmesi gerekir. IN16 pasif duruma getirildiğinde program durdurulur.

Tablo 3-2. Robot sayısal girişlerine yapılan elektriksel bağlantılar.

İşlev	Pin	Robot Adresi	RiaBox Pin	PLC Adresi
Program adının BCD kodu (Bit 0) **	15	IN0	13	A0.0
Program adının BCD kodu (Bit 1) **	16	IN1	14	A0.1
Program adının BCD kodu (Bit 2) **	17	IN2	15	A0.2
Program adının BCD kodu (Bit 3) **	18	IN3	16	A0.3
Program Yükle (yükselen kenar)	24	IN16 (PGN) *	17	A0.4
Program Başlat (aktif sinyal)	25	IN17	18	A0.5
Program Durdur (aktif sinyal)	49	IN18	19	A0.6
Program Başa-dön (aktif sinyal)	50	IN19	20	A0.7

* Kullanılmadan önce RV-E2 kontrolör el kitabına göre yapılandırılmalıdır.

** Kontrolör yapılandırma ayarları değiştirilerek bu 4 bit, robot program akışı içerisinde PLC gibi harici sistemlerden sinyal girişi için kullanılabilir.

Robot kontrolörden harici cihazlara gönderilen çıkış sinyalleri ve açıklamaları Tablo 3-3’de verilmiştir. Robot kontrolör, PLC’den komut aldıktan sonra başlatılan programın durumunu dört bit (OT0-OT3) üzerinden PLC’ye bildirir. Örneğin robot kontrolör OT0-OT3 çıkış bitleri arasına binary “1110” yazıldığında, “robot kaynak konum bilgisini bekliyor”; binary “1111” yazıldığında “robot hedef konum bilgisini bekliyor” durum bilgisi PLC’ye gönderilir.

Robotun iletişime hazır ve komut beklediği bilgisi OT12 çıkışı üzerinden gönderilir. Benzer şekilde programın başladığı ve yürütmekte olduğu OT13, bekleme durumunda olduğu OT14, hata olduğu OT15 sayısal çıkış bitleri üzerinden gönderilir.

Tablo 3-3. Robot sayısal çıkışlarına yapılan elektriksel bağlantılar.

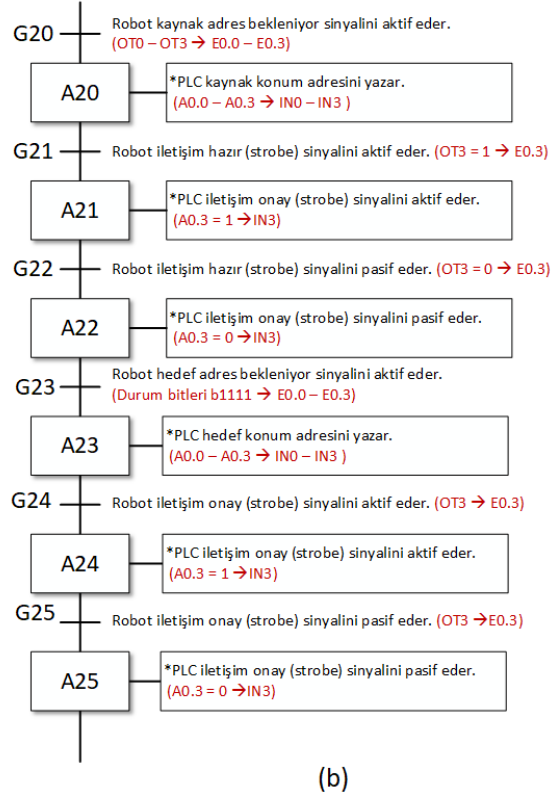
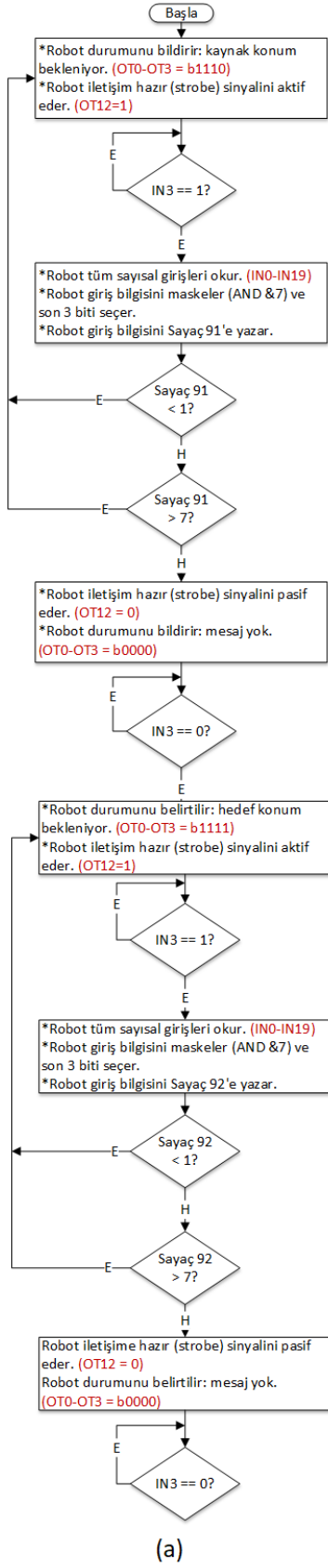
İşlev	Pin	Robot Adresi	RiaBox Pin	PLC Adresi
Program Durum (Bit 0) *	4	OT0	1	E0.0
Program Durum (Bit 1) *	5	OT1	2	E0.1
Program Durum (Bit 2) *	6	OT2	3	E0.2
Program Durum (Bit 3) *	7	OT3	4	E0.3
İletişime Hazır (Strobe) *	35	OT12	5	E0.4
Robot BAŞLADI (pozitif sinyal)	36	OT13	6	E0.5
Robot BEKLEMEDE - başlamaya hazır - (pozitif sinyal)	37	OT14	7	E0.6
Robot HATA (pozitif sinyal)	38	OT15	8	E0.7

* Kontrolör yapılandırma ayarları değiştirilerek bu 5 bit, robot program akışı içerisinde PLC gibi harici sistemlere sinyal çıkış vermek için kullanılabilir.

Harici bir sistem ile robot arası iletişime örnek olması açısından, Montaj hücresindeki robotta yüklü olan 6 numaralı programın işleyişi aşağıda açıklanmıştır. 6 numaralı program, FMS100 sisteminin bir ürünü olan elektronik rulete ait alt/üst kapakların ve yarı montajı tamamlanmış ürünlerin, hücre içerisinde taşınması için kullanılır. Örnek uygulamaya ait robot programı SDG yöntemiyle gösterime uygun olmadığı için, program algoritması Şekil 3-62 (a)'daki gibi standart algoritma yöntemiyle hazırlanmıştır. Hücresin tüm otomasyonunu yöneten IPC'de yüklü programın, 6 numaralı robot programıyla etkileşimli olduğu bölüm ise SDG gösterimiyle açıklanmıştır.

Şekil 3-62'den de anlaşılacağı üzere robotun durum bilgisini güncellemesiyle birlikte IPC ile etkileşim aşağıda görülen sıralamayla başlar:

- Robot durum bilgisini (binary 1110, hexadecimal E) değiştirerek kaynak adresi beklediğini bildirir.
- IPC kaynak konum bilgisini yazar.
- Robot iletişime hazır olduğu sinyali (strobe) aktif eder. IPC den cevap bekler.
- IPC iletişim onay sinyali (strobe) aktif eder.
- Robot kaynak adresi okur, 91 numaralı hafıza alanına kaydeder, kontrol eder. Eğer uygunsa iletişim onay sinyali pasif eder. Durum bilgisini (binary 0000, hexadecimal 0) değiştirerek mesajı olmadığını bildirir. IPC'den cevap bekler.
- IPC iletişim onay sinyali (strobe) pasif eder.
- Robot durum bilgisini (binary 1111, hexadecimal F) değiştirerek hedef adresi beklediğini bildirir.
- IPC hedef konum bilgisini yazar.
- Robot iletişime hazır olduğu sinyali (strobe) aktif eder. IPC den cevap gelmesini bekler.
- IPC iletişim onay sinyali (strobe) aktif eder.
- Robot kaynak adresi okur, 92 numaralı hafıza alanına kaydeder, kontrol eder, eğer uygunsa iletişim onay sinyali pasif eder. Durum bilgisini (binary 0000, hexadecimal 0) değiştirerek mesajı olmadığını bildirir. IPC'den cevap gelmesini bekler.
- IPC iletişim onay sinyali (strobe) pasif eder.
- Robot 6 numaralı programa kaldığı yerden devam eder
- IPC programına kaldığı yerden devam eder.



Şekil 3-62. Montaj istasyonu 6 numaralı robot programı algoritması gösterini (a), IPC programının 6 numaralı robot programı ile etkileşimli olan bölümü SDG gösterimi (b).

3.1.7.2. Movemaster Komut Kümesi

Bu bölümde Lehimleme, Montaj ve Görüntü İşleme hücrelerinde bulunan RV-E2 model robotların programlanmasında kullanılan Movemaster Command komut kümesi anlatılacaktır. Komut kümesi bünyesinde birçok komut bulunmakta olup detaylara [referans robot] kaynağından ulaşılabilir. Bu bölümde FMS100 robotik uygulamalarında sıklıkla kullanılabilecek komutlar açıklanacaktır.

Movemaster Command dili Q-Basic tabanlı olduğu için her satıra 10'ar artımlı numara verilir. Böylelikle sonradan araya satır eklenirse ara numara verilebilir. Noktalı virgül “;” işareti ile herhangi bir satıra açıklama yazılabilir.

3.1.7.2.1. Program Akış Denetim Komutları

SC (Set Counter – Sayaca Değer Gir)

Sayaç, istenilen değeri saklayan hafıza alanına denir. SC komutu herhangi bir değerin (sayısal veya karakter dizisi olabilir) sayaca yüklenmesini sağlar. Robot denetleyici sayısal değerleri 1-99 arası toplam 99 adet farklı sayaçta saklayabilir. Benzer şekilde karakterleri ve karakter dizilerini “\$1” ile “\$99” arası toplam 99 adet farklı sayaçta saklayabilir.

Komut şu şekilde kullanılır: SC parametre1, parametre2

Parametre1: Eğer sayaçta sayısal değer saklanacaksa, parametre değeri “1” ile “99” arasındaki herhangi bir sayı olabilir. Eğer ki sayaçta karakter dizisi saklanacaksa, parametre değeri baş tarafına “\$” işareti gelerek “\$1” ile “\$99” arası herhangi bir değer olabilir.

Parametre2: Eğer sayaçta sayısal bir değer saklanacaksa; 16 bit uzunlukta, -32768 ile 32767 arası herhangi bir değer girilebilir. Eğer sayaçta karakter dizisi saklanacaksa, dizi uzunluğu sayaç numarasının karakterlerinin uzunluğu da dahil olmak üzere en fazla 118 karakter olabilir. Girilebilecek karakterler 0-9 arası nümerik, A-Z arası alfabetik ve tırnak işareti dışındaki noktalama işaretleri olabilir.

Örneğin:

10 SC 35,40 ;35 numaralı sayaca 40 değerini yükler.

20 SC \$2,“TAMAM” ;\$2 numaralı karakter sayacına “TAMAM” değerini yükler.

IC (Increment Counter – Sayaç Değerini Arttır) / DC (Decrement Counter – Sayaç Değerini Azalt)

IC komutu istenilen herhangi bir sayısal sayaç değerini 1 arttır. DC ise sayaç değerini 1 azaltır.

Komut şu şekilde kullanılır: IC parametre1 veya DC parametre1

Parametre1: Sayısal değer saklanan “1” ile “99” arasındaki sayaç numarasıdır.

Örneğin:

10 SC 35,40 ;35 numaralı sayaca 40 değerini yükler.

20 IC 35 ;35 numaralı sayacın değerini 1 arttırır.

30 DC 35 ;35 numaralı sayacın değerini 1 azaltır.

CP (Compare Counter – Sayacı Karşılaştır)

İstenilen herhangi bir sayaç değerini, karşılaştırma işlerinde kullanabilmek için hafızaya yükler.

Komut şu şekilde kullanılır: CP parametre1

Parametre1: Eğer sayaçta sayısal değer saklanıyorsa, “1” ile “99” arasındaki herhangi bir sayı olabilir. Eğer ki sayaçta karakter dizisi saklanıyorsa, baş tarafına “\$” işareti gelerek “\$1” ile “\$99” arası herhangi bir değer yazılır.

Örneğin:

10 SC 35,40 ;35 numaralı sayaca 40 değerini yükler.

20 CP 35 ;35 numaralı sayacın içindeki değeri karşılaştırma yapılabilmesi için hafızaya yükler.

EQ (Equal – Eşit) / NE (Not Equal – Eşit Değil)

Karşılaştırma için hafızaya yüklenmiş değer, istenilen herhangi bir değer ile eşit (EQ) veya eşit olmamasına (NE) bakar. Her iki durumda da koşul sağlanıyorsa, istenilen bir program satırına dallanır.

Komut şu şekilde kullanılır: EQ parametre1, parametre2 veya NE parametre1, parametre2

Parametre1: Eğer sayısal bir değer karşılaştırılacaksa; 16 bit uzunlukta, -32768 ile 32767 arası herhangi bir değer girilebilir. Eğer ki karakter dizisi karşılaştırılacaksa,

karşılaştırılmak istenen karakter dizisinin saklandığı karakter sayaç numarası baş tarafına “\$” işareti konularak girilir.

Parametre2: Koşul doğruysa dallanılacak program satır numarası girilir. Satır numarası 1 ile 9999 arası olabilir.

Örneğin:

10 SC \$35,“TAMAM” ;35 numaralı karakter sayacına “TAMAM” değerini yükler.

20 CP \$35 ;35 numaralı karakter sayacı değerini karşılaştırma yapılabilmesi için hafızaya yükler.

30 EQ \$3,150 ;Hafızadaki değer (“TAMAM”) ile \$3 numaralı karakter sayacı içindeki değeri karşılaştırır, eşitse 150 numaralı program satırına dallanır. Eşit değilse bir alt satıra geçer.

40 NE \$5,100 ;Hafızadaki değer (“TAMAM”) ile \$5 numaralı karakter sayacı içindeki değeri karşılaştırır, eşit değilse 100 numaralı program satırına dallanır. Eşitse bir alt satıra geçer.

SM (If Smaller – Küçükse) / LG (If Larger - Büyükse)

Karşılaştırma için hafızaya yüklenmiş değer, istenilen herhangi bir değerden küçük (SM) veya büyük olmasına (LG) bakar. Her iki durumda da koşul sağlanıyorsa, istenilen bir program satırına dallanır.

Komut şu şekilde kullanılır: SM parametre1, parametre2 veya LG parametre1, parametre2

Parametre1: Eğer sayısal bir değer karşılaştırılacaksa; 16 bit uzunlukta, -32768 ile 32767 arası herhangi bir değer girilebilir. Eğer ki karakter dizisi karşılaştırılacaksa, karşılaştırılmak istenen karakter dizisinin saklandığı karakter sayaç numarası baş tarafına “\$” işareti konularak girilir.

Parametre2: Koşul doğruysa dallanılacak program satır numarası girilir. Satır numarası 1 ile 9999 arası olabilir.

Örneğin:

10 SC 21,44 ;21 numaralı sayaca 44 değerini yükler.

20 CP 21 ;21 numaralı sayaç değerini karşılaştırma yapılabilmesi için hafızaya yükler.

30 SM 0,100 ;Hafızdaki değ r (44) 0'dan k       100 numaralı satıra dallanır, deęilse bir alt satıra ge er.

40 LG 80,150 ;Hafızdaki değ r (44) 80'den b       150 numaralı satıra dallanır, deęilse bir alt satıra ge er.

50 EQ 20,200 ;Hafızadaki değ r (44) 20'ye eęitse 200 numaralı satıra dallanır, deęilse bir alt satıra ge er.

EA (Enable Act – Harici Kesme Aktiv) / DA (Disable Act – Harici Kesme İptal)

EA komutu Harici giriř sinyali ile robot programına kesme sinyali g nderilebilmesini saęlar. Komut ile tanımlanan giriř biti istenilen durumda olduęunda (aktif/pasif, lojik1/0) robot programı durur ve belirlenen program satırına dallanır. Harici kesme giriři eski konumuna d nene kadar veya dallanılan program satırında DA komutu olmadıęı s rece, program durur vaziyette kalır. DA komutu ise harici kesmeyi iptal eder.

Komut řu řekilde kullanılır: EA [parametre0] parametre1, parametre2 [,parametre3] veya DA parametre1

Parametre0: İsteęe baęlı kullanılacak bir parametredir. Pozitif “+” veya negatif “-” iřareti olabilir. Eęer hi bir se enek girilmezse fabrika ayarı olarak pozitif kabul edilir. İřaret pozitif ise tanımlanan giriř biti aktif (lojik 1) olduęunda veya iřaret negatif ise tanımlanan giriř biti pasif (lojik 0) olduęunda kesme ger ekleřir ve belirlenen program satırına dallanır.

Parametre1: Harici kesme olarak kullanılacak giriř biti tanımlanır. Giriř biti 0 ile 32767 arasında olabilir. Aynı program i erisinde 8 adede kadar farklı giriř kesme sinyali i in tanımlanabilir. 0 ile 8999 harici kesme giriři, 0 ile 299 arası genel giriř, 900 ile 903 arası robot eli ile ilgili giriřlerdir. 9000 alarm giriřini temsil eder. Parametre2 9000 olarak ayarlandıęında ve herhangi bir alarm ger ekleřtięinde, belirlenen program satırına dallanır. 9001 acil durdurma (emergency stop) giriřini temsil eder. Parametre2 9001 olarak ayarlandıęında ve acil durdurma butonuna basıldıęında, belirlenen program satırına dallanır. 9002 haberleřme kesme sinyalini temsil eder. Parametre2 9002 olarak ayarlandıęında ve RS232C portundan herhangi bir veri geldięinde, belirlenen program satırına dallanır.

Parametre2: Kesme sinyali geldiğinde dallanılacak program satır numarası girilir. Satır numarası 1 ile 9999 arası olabilir.

Parametre3: İsteğe bağlı kullanılacak bir parametredir. Kesme sinyali geldiğinde gerçekleştirilecek dallanma biçimini tanımlar. 0 veya 1 olabilir. Eğer hiçbir seçenek girilmezse fabrika ayarı olarak 0 kabul edilir. Kesme sinyali geldiğinde, 0 ise Parametre3 ile tanımlanan program satırına dallanır; 1 ise alt program çağırma şeklinde çalışır.

Örneğin:

10 EA +4,70 ;4 numaralı harici kesme giriş biti aktif (lojik1) olduğunda 70 numaralı program satırına dallanır.

20 MO 1 ;1 numaralı konuma eksenel enterpolasyon ile hareket eder.

30 ED ;Programı sonlandırır.

70 DA 4 ;4 numaralı harici girişin kesme özelliğini iptal eder.

80 MO 2 ;2 numaralı konuma eksenel enterpolasyon ile hareket eder.

90 GT 20 ;20 numaralı program satırına dallanır.

GT (Go To – Program Satırına Git)

Herhangi bir koşula bakmaksızın robot programın istenilen bir satırına atlar.

Komut şu şekilde kullanılır: GT parametre1

Parametre1: Atlanacak program satır numarası girilir. Satır numarası 1 ile 9999 arası olabilir.

Örneğin:

10 SC 5,8 ;5 numaralı sayaca 8 değerini yükler.

20 GT 210 ;210 numaralı program satırına atlar.

GS (Go Sub – Alt Programa Git) / RT (Return – Geri Dön)

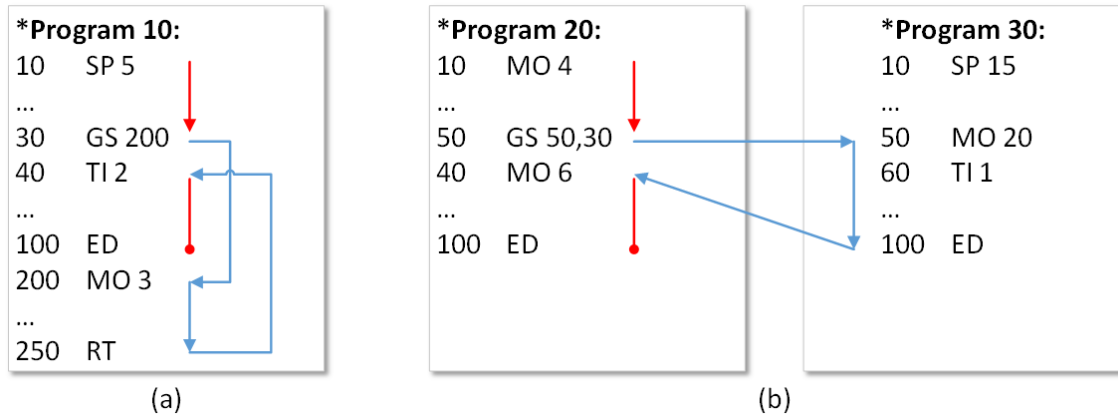
GS komutu, herhangi bir koşula bakmaksızın GT komutu gibi robot programın istenilen bir satırına veya farklı bir programın istenilen satırına atlar. RT alt programdan dönüş komutudur ve alt program bloğunun sonuna eklenir. RT komutuyla birlikte hiçbir parametre tanımlanmamışsa, ana programa dönerek GS komutunun verildiği satırın bir alt satırından itibaren çalışmaya devam eder. Alt program bloğunun sonunda ED komutu

da olabilir. Alt program ED komutuyla bitirildiğinde de RT komutuyla aynı davranışı gösterir. RT komutunda istenirse parametre olarak geri dönüş satırı girilebilir. Böylelikle alt program sonunda ana program bloğunun istenilen satırına dönülebilir. GS, RT ve ED komutlarıyla alt programa dallanma işlemine yönelik örnek Şekil 3-63’de verilmiştir.

Komut şu şekilde kullanılır: GS [parametre1][,parametre2] veya RT [parametre1]

Parametre1: İsteğe bağlı kullanılacak bir parametredir. GS komutu için alt programda atlanacak satır numarası girilir. RT komutu için ana programda dönülecek satır numarası girilir. Satır numarası 1 ile 9999 arası olabilir.

Parametre2: İsteğe bağlı kullanılacak bir parametredir. Atlanacak alt program adı girilerek farklı bir programa atlama sağlanabilir. Alt program adı 0 ile 65536 arası tam sayı olabileceği gibi, 1 ile 8 karakter arası uzunluğa sahip karakter dizisi olabilir. Karakter dizisinde 0-9 arası rakamlar, A-Z arası karakterler ve *+,./:;[]”” hariç, diğer noktalama işaretleri olabilir.



Şekil 3-63. GS komutu ile alt programa dallanma uygulamalarına örnek: aynı program içinde dallanma (a), farklı programlar arası dallanma (b).

Örneğin:

```
10 GS 100    ;1Alt program bloğunun 100 numaralı satırına atlar.
20 SC 1,12   ;1 numaralı sayaca 12 değerini girer.
...
90 ED        ;Programı sonlandırır.
100 MO 35    ;35 numaralı konuma eksenel enterpolasyon ile hareket eder.
```


101 MO 36 ;36 numaralı konuma eksenel enterpolasyon ile hareket eder.

102 MO 37 ;37 numaralı konuma eksenel enterpolasyon ile hareket eder.

103 RT ;Alt program bloğundan ana program bloğuna geri döner (bu örnekte 20 numaralı satıra döner).

GO (Gripper Open – Tutucu El Açık) / GC (Tutucu El Kapalı)

GO komutu tutucu elin açılmasını, GC komutu elin kapanmasını sağlar. Robot parçayı tutma konumuna gittiğinde, parçayı tutmadan/bırakmadan önce belirli bir zaman aralığına ihtiyaç vardır. Bu nedenle GO/GC komutları öncesi ve sonrası TI komutu ile zaman gecikmesi verilmesi daha sağlıklıdır. İstenirse GP komutu kullanılarak tutma kuvveti ve tutma/bırakma gecikme periyotları ayarlanabilir. Tutucu el açma/kapama işlevleri 900-903 arası çıkış bitleri üzerinden kontrol edilir. Robot için iki farklı el tanımlaması yapılabilir. 900 ve 901 numaralı çıkış bitleri 1 numaralı eli, 902 ve 903 numaralı çıkış bitleri 2 numaralı eli kontrol eder. Eğer elektrik tahrikli bir el kullanılmışsa bu çıkışlar doğrudan elin kontrol uçlarına bağlanır. Eğer pnömatik tahrikli bir el kullanılmışsa bu çıkışlar doğrudan pnömatik selenoid valfine bağlanır. OB komutu ile çıkışların durumu değiştirilerek de eli açma/kapama işlemi gerçekleştirilebilir.

Komut şu şekilde kullanılır: GO [parametre1] veya GC [parametre1]

Parametre1: İsteğe bağlı kullanılacak bir parametredir. Açılacak/kapanacak el numarası girilir. 0 veya 1 değeri girilebilir. Fabrika ayarı olan 0 el 1'in, 1 ise el 2'nin kontrol edilmesini sağlar.

Örneğin:

10 MO 5,C ;5 numaralı konuma, el kapalı şekilde, eksenel enterpolasyon ile hareket eder.

20 TI 2 ;0,2 saniye gecikme verir.

30 GO ;Eli açar.

40 TI 2 ;0,2 saniye gecikme verir.

50 MO 3,O ;5 numaralı konuma, el açık şekilde, eksenel enterpolasyon ile hareket eder.

TI (Timer – Zaman Gecikmesi)

Belirlenen sürede zaman gecikmesi verir. En düşük değeri 100 ms'dir. Özellikle tutma/bırakma gibi hassas hareketlerin öncesinde ve sonrasında kullanılır.

Komut şu şekilde kullanılır: TI parametre1

Parametre1: Beklenecek gecikme süresi belirlenir. Süre 100 ms'nin katları olmalıdır. Süre sayacı 0 ile 32767 arası olabilir.

Örneğin:

10 TI 5 ;500 ms gecikme verir.

20 TI 200 ;20 s gecikme verir.

ED (End)

Programı sonlandırır.

Komut şu şekilde kullanılır: ED

Örneğin:

10 MO 1 ;1 numaralı konuma eksenel enterpolasyon ile hareket eder.

20 ED ;Programı sonlandırır.

3.1.7.2.2. Hareket Komutları

MJ (Move Joint – Eksenel Hareket)

İstenilen eklemün bulunduđu konumdan artımsal olarak derece cinsinden hareket etmesini sağlar. Girilebilecek en küçük açı değeri 0,01 derecedir. Açı değeriñin negatif olması ters yönlü hareketi ifade eder.

Komut şu şekilde kullanılır: MJ parametre1, parametre2, parametre3, parametre4, parametre5, parametre6

Parametre1-6: Her bir eklem için girilecek artımsal açı değeri belirlenir. Eklem parametreleri sırasıyla; 1: bel (waist), 2: omuz (shoulder), 3: dirsek (elbow), 4: bilek döndürme (twist), 5: bilek kaldırma (pitch), 6: bilek yuvarlama (roll) şeklindedir.

Örneğin:

10 MJ 80,0,0,0,0,0 ;1 numaralı bel eksenini pozitif yönde 100 derece hareket ettirir.

20 MJ 0,30,-40,0,0,0 ; Aynı anda 2 numaralı omuz eksenini pozitif yönde 30 derece, 3 numaralı dirsek eksenini negatif yönde 40 derece hareket ettirir.

MO (Move - Hareket)

Robot uç noktasını (örneğin elini) XYZ modunda eksenel enterpolasyon ile hareket ettirmeyi sağlar. İstenirse hareket sırasında elin açık/kapalı olma durumu parametre ile tanımlanabilir ve robot elin durumunu ayarladıktan sonra hareketine başlar.

Komut şu şekilde kullanılır: MO parametre1 [,parametre2]

Parametre1: Robotun hareket etmesi istenen konuma ait XYZABC koordinat değerleridir. Bu koordinat değerleri konum olarak önceden “.POS” dosyasına kaydedilmiş olmalıdır. 1 ile 999 arası değer girilebilir.

Parametre2: İsteğe bağlı kullanılacak bir parametredir. Robotun hareketi sırasında tutucu elin açık/kapalı olması durumunun önemli olduğu koşullarda bu parametre kullanılır. Parametre değeri “O” olursa hareket sırasında el açık, “C” olursa el kapalıdır.

Örneğin:

10 MO 3,O ;3 numaralı konuma, el açık şekilde, eksenel enterpolasyon ile hareket eder.

20 MO 2,C ;2 numaralı konuma, el kapalı şekilde, eksenel enterpolasyon ile hareket eder.

30 MO 5 ;5 numaralı konuma eksenel enterpolasyon ile hareket eder.

MS (Move Straight – Doğrusal Hareket)

Robot uç noktasını (örneğin elini) XYZ modunda doğrusal enterpolasyon ile hareket ettirmeyi sağlar. İstenirse hareket sırasında elin açık/kapalı olma durumu parametre ile tanımlanabilir ve robot elin durumunu ayarladıktan sonra hareketine başlar.

Komut şu şekilde kullanılır: MS parametre1 [,parametre2]

Parametre1: Robotun hareket etmesi istenen konuma ait XYZABC koordinat değerleridir. Bu koordinat değerleri konum olarak önceden “.POS” dosyasına kaydedilmiş olmalıdır. 1 ile 999 arası değer girilebilir.

Parametre2: İsteğe bağlı kullanılacak bir parametredir. Robotun hareketi sırasında tutucu elin açık/kapalı olması durumunun önemli olduğu koşullarda bu parametre kullanılır. Parametre değeri “O” olursa hareket sırasında el açık, “C” olursa el kapalıdır.

Örneğin:

10 MS 2,O ;2 numaralı konuma, el açık şekilde, doğrusal enterpolasyon ile hareket eder.

20 MS 3,C ;3 numaralı konuma, el kapalı şekilde, doğrusal enterpolasyon ile hareket eder.

30 MS 1 ;1 numaralı konuma doğrusal enterpolasyon ile hareket eder.

MT (Move Tool – Takım Düzleminde Hareket)

Robot uç noktasını (örneğin elini) XYZ modunda eksenel enterpolasyon ile istenilen konumun takım ekseninde uzağına hareket ettirmeyi sağlar. İstenirse hareket sırasında elin açık/kapalı olma durumu parametre ile tanımlanabilir ve robot elin durumunu ayarladıktan sonra hareketine başlar.

Komut şu şekilde kullanılır: MT parametre1, parametre2 [,parametre3]

Parametre1: Robotun hareket etmesi istenen konuma ait XYZABC koordinat değerleridir. Bu koordinat değerleri konum olarak önceden “.POS” dosyasına kaydedilmiş olmalıdır. 1 ile 999 arası değer girilebilir.

Parametre2: Robot uç noktasının hedef konumdan takım ekseninde ne kadar uzağa hareket edeceğini belirler. Negatif değer takım ekseninde hedef konumdan uzağa, pozitif değer ise hedef konumun yakınına hareket eder. En küçük artım değeri 0,01 mm olabilir. -3276,80 ile 3276,70 arası değer girilebilir.

Parametre3: İsteğe bağlı kullanılacak bir parametredir. Robotun hareketi sırasında tutucu elin açık/kapalı olması durumunun önemli olduğu koşullarda bu parametre kullanılır. Parametre değeri “O” olursa hareket sırasında el açık, “C” olursa el kapalıdır.

Örneğin:

10 SP 20 ;Hareket hızı seviyesini 20 değerine ayarlar.

20 MT 2,-100,C ;2 numaralı konumun takım ekseninde 100 mm uzağına, el kapalı şekilde, eksenel enterpolasyon ile hareket eder.

30 SP 5 ;Hareket hızı seviyesini 5 değerine ayarlar.
40 MS 2 ;2 numaralı konuma doğrusal enterpolasyon ile hareket eder.
50 TI 1 ;100 ms gecikme verir.
60 GO ;Tutucu eli açar.
70 TI 1 ;100 ms gecikme verir.
80 MTS 2,-100,O ;2 numaralı konumun takım ekseninde 100 mm uzağına, el kapalı şekilde, doğrusal enterpolasyon ile hareket eder.

MTS (Move Tool Straight - Takım Düzleminde Doğrusal Hareket)

Robot uç noktasını (örneğin elini) XYZ modunda doğrusal enterpolasyon ile istenilen konumun takım ekseninde uzağına hareket ettirmeyi sağlar. İstenirse hareket sırasında elin açık/kapalı olma durumu parametre ile tanımlanabilir ve robot elin durumunu ayarladıktan sonra hareketine başlar.

Komut şu şekilde kullanılır: MTS parametre1, parametre2 [,parametre3]

Parametre1: Robotun hareket etmesi istenen konuma ait XYZABC koordinat değerleridir. Bu koordinat değerleri konum olarak önceden “.POS” dosyasına kaydedilmiş olmalıdır. 1 ile 999 arası değer girilebilir.

Parametre2: Robot uç noktasının hedef konumdan takım ekseninde ne kadar uzağa hareket edeceğini belirler. Negatif değer takım ekseninde hedef konumdan uzağa, pozitif değer ise hedef konumun yakınına hareket eder. En küçük artım değeri 0,01 mm olabilir. -3276,80 ile 3276,70 arası değer girilebilir.

Parametre3: İsteğe bağlı kullanılacak bir parametredir. Robotun hareketi sırasında tutucu elin açık/kapalı olması durumunun önemli olduğu koşullarda bu parametre kullanılır. Parametre değeri “O” olursa hareket sırasında el açık, “C” olursa el kapalıdır.

Örneğin:

100 MTS 5,-50 ;5 numaralı konumun takım ekseninde 50 mm uzağına doğrusal enterpolasyon ile hareket eder.

110 MTS 4,-100,O ;4 numaralı konumun takım ekseninde 100 mm uzağına, el açık şekilde, doğrusal enterpolasyon ile hareket eder.

SP (Speed – Hareket Hızı Ayarı)

Robot hareketinin hız seviyesi, hızlanma/yavaşlama süresi, sürekli rotada hareket gibi özelliklerinin ayarlanmasını sağlar.

Komut şu şekilde kullanılır: SP parametre1 [,parametre2] [,parametre3]

Parametre1: Hareket hızı seviye ayarıdır. 0 ile 30 arası toplam 31 farklı değer alabilir. Hareket hızı seviyeleri, eksenel enterpolasyonda her bir eksen motorunun en yüksek hız değerinin (rpm) oranı olacak şekilde, doğrusal enterpolasyonda robot uç noktasının (elinin) en yüksek doğrusal hareket hızının (650 mm/s) oranı olacak şekilde önceden tanımlanmıştır.

Parametre2: İsteğe bağlı kullanılacak bir parametredir. Robot hareketinin hızlanma/yavaşlama seviye ayarıdır. Robotun en yüksek hıza çıkma süresidir. Robot en yüksek hızdan daha düşük bir hızda hareket edecekse, gerçek hızlanma/yavaşlama süresi daha kısa olacaktır. Parametre değeri “H” olduğu zaman yüksek ivmelenme (en fazla 0,2 s), “L” olduğu zaman düşük ivmelenme (en fazla 0,4 s) gerçekleşir.

Parametre3: İsteğe bağlı kullanılacak bir parametredir. Eğer robot belirli bir rotada ardışık konumlara uğrayarak sürekli hareket edecekse, rota hareketinin hızını arttırmak ve harekette süreklilik sağlamak için kullanılır. 0 değeri girilirse sürekli hareket özelliği iptal, 1 değeri girilirse özellik aktif demektir.

Örneğin:

10 SP 20,1 ;Hareket hızı seviyesini 20 değerine ayarlar ve sürekli hareketi aktif eder.

20 MO 2,C ;2 numaralı konuma, el kapalı şekilde, eksenel enterpolasyon ile hareket eder.

30 MO 3,C ;3 numaralı konuma, el kapalı şekilde, eksenel enterpolasyon ile hareket eder.

40 MO 7,C ;7 numaralı konuma, el kapalı şekilde, eksenel enterpolasyon ile hareket eder.

50 MT 1,-50,C ;1 numaralı konumun takım ekseninde 50 mm uzağına, el kapalı şekilde, eksenel enterpolasyon ile hareket eder.

60 SP 5,0 ;Hareket hızı seviyesini 5 değerine ayarlar, sürekli hareketi iptal eder.

70 MS 1 ;1 numaralı konuma doğrusal enterpolasyon ile hareket eder.

80 TI 1 ;100 ms gecikme verir.

90 GO ;Tutcu eli açar.

100 TI 1 ;100 ms gecikme verir.

110 MTS 1,-100,O ;2 numaralı konumun takım ekseninde 100 mm uzağına, el açık şekilde, doğrusal enterpolasyon ile hareket eder.

3.1.7.2.3. Sayısal Giriş/Çıkış Komutları

ID (Input Direct – Doğrudan Giriş)

Herhangi bir koşul aramadan robot sayısal girişlerindeki veriyi okur. Sayısal girişler arasında robot elini tanımlayan ve kontrol eden girişler de vardır. Okunan veri direk hafızaya alınır ve karşılaştırma komutları tarafından kullanıma uygundur.

Komut şu şekilde kullanılır: ID [parametre1]

Parametre1: Okunmak istenen robot girişine ait bit adresi tanımlanır. 0 ile 32767 arası değer girilebilir. Hiçbir değer girilmezse fabrika ayarı olarak değer 0'dır. Parametre olarak tanımlanan bit adresinden başlamak kaydıyla her seferinde 16 bit sıralı girişi okur. Robot elinin durumunu okumak için 900'lü adreslere bakmak gerekir.

Örneğin:

100 ID ;Robot sayısal girişlerinin, 0'ıncı bitten başlayarak, 16 adedini okur.

110 EQ 55,130 ;Okunan değer 55'e eşitse 130 numaralı satıra dallanır.

120 ED ;Programı sonlandırır.

130 MO 3 ;3 numaralı konuma eksenel enterpolasyon ile hareket eder.

140 ID 120 ;Robot sayısal girişlerinin, 120'inci bitten başlayarak, 16 adedini okur.

150 TB +0,180 ;Eğer 120 numaralı giriş biti aktif (lojik 1) ise, 180 numaralı satıra dallanır.

160 TB +5,200 ; Eğer 125 numaralı giriş biti aktif (lojik 1) ise, 200 numaralı satıra dallanır.

170 ED ;Programı sonlandırır.

180 MO 5 ;5 numaralı konuma eksenel enterpolasyon ile hareket eder.

190 ED ;Programı sonlandırır.

200 MO 6 ;6 numaralı konuma eksenel enterpolasyon ile hareket eder.

210 ED ;Programı sonlandırır.

CL (Counter Load – Sayaç Yükle)

Hafızada yüklü olan değerin (sayısal veya karakter dizisi olabilir) sayaca yüklenmesini sağlar. Robot denetleyici sayısal değerleri 1-99 arası toplam 99 adet farklı sayaçta saklayabilir. Benzer şekilde karakterleri ve karakter dizilerini “\$1” ile “\$99” arası toplam 99 adet farklı sayaçta saklayabilir.

Eğer sayaçta sayısal bir değer saklanacaksa; 16 bit uzunlukta, -32768 ile 32767 arası herhangi bir değer girilebilir. Eğer sayaçta karakter dizisi saklanacaksa, dizi uzunluğu sayaç numarasının karakterlerinin uzunluğu da dahil olmak üzere en fazla 118 karakter olabilir. Girilebilecek karakterler 0-9 arası nümerik, A-Z arası alfabetik ve tırnak işareti dışındaki noktalama işaretleri olabilir.

Komut şu şekilde kullanılır: CL parametre1

Parametre1: Eğer sayaçta sayısal değer saklanacaksa, parametre değeri “1” ile “99” arasındaki herhangi bir sayı olabilir. Eğer ki sayaçta karakter dizisi saklanacaksa, parametre değeri baş tarafına “\$” işareti gelerek “\$1” ile “\$99” arası herhangi bir değer olabilir.

Örneğin:

10 ID ;Robot sayısal girişlerinin, 0’ıncı bitten başlayarak, 16 adedini okur.

20 CL 35 ;Hafızadaki değeri 35 numaralı sayaca yükler.

30 CP 34 ;34 numaralı sayaçtaki değeri hafızaya yükler.

40 CL 16 ;Hafızadaki değeri 16 numaralı sayaca yükler.

50 SC \$4, “BCD” ;\$4 numaralı karakter sayacına “BCD” değerini yükler.

60 CP \$4 ;\$4 numaralı karakter sayaçtaki değeri hafızaya yükler.

70 CL \$10 ;Hafızadaki değeri \$10 numaralı karakter sayacına yükler.

TB (Test Bit – Bit Değerini Kontrol Et)

Hafızada yüklü olan değer her hangi bir bitine bakarak istenilen program satırına dallanır.

Komut şu şekilde kullanılır: TB [parametre0] parametre1, parametre2

Parametre0: İsteğe bağlı kullanılacak bir parametredir. Pozitif “+” veya negatif “-” işareti olabilir. Eğer hiçbir seçenek girilmezse fabrika ayarı olarak pozitif kabul edilir. İşaret pozitif ise tanımlanan giriş biti aktif (lojik 1) olduğunda veya işaret negatif ise tanımlanan giriş biti pasif (lojik 0) olduğunda belirlenen program satırına dallanır.

Parametre1: Hafızadan okunmak istenen değer hangi bit olduğu tanımlanır. 0 ile 15 arası değer girilebilir.

Parametre2: Dallanılacak satır numarası girilir. Satır numarası 1 ile 9999 arası olabilir.

Örneğin:

10 ID ;Robot sayısal girişlerinin, 0’ıncı bitten başlayarak, 16 adedini okur.

20 TB +1,90 ;Eğer 1 numaralı giriş biti aktif (lojik 1) ise, 90 numaralı satıra dallanır.

30 MS 1 ;34 numaralı sayaçtaki değeri hafızaya yükler.

40 ED ;Hafızadaki değeri 16 numaralı sayaca yükler.

...

90 MO 10 ;10 numaralı konuma eksenele enterpolasyon ile hareket eder.

OB (Output Bit – Çıkış Biti)

İstenilen bir çıkış bitinin değerini değiştirir.

Komut şu şekilde kullanılır: OB [parametre0] parametre1

Parametre0: İsteğe bağlı kullanılacak bir parametredir. Pozitif “+” veya negatif “-” işareti olabilir. Eğer hiçbir seçenek girilmezse fabrika ayarı olarak pozitif kabul edilir. İşaret pozitif ise tanımlanan çıkış biti aktif (lojik 1) olur. Benzer şekilde işaret negatif ise tanımlanan çıkış biti pasif (lojik 0) olur.

Parametre1: Hafızadan okunmak istenen değerin hangi bit olduğu tanımlanır. 0 ile 15 arası değer girilebilir.

Parametre2: Durumu değiştirilmek istenen çıkış biti girilir. 0 ile 32767 arası değer girilebilir.

Örneğin:

10 OB -8 ;8 numaralı robot sayısal çıkışının durumunu pasif (lojik 0) yapar.

20 OB +4 ;4 numaralı robot sayısal çıkışının durumunu aktif (lojik 1) yapar.

OD (Output Direct – Doğrudan Çıkış)

Herhangi bir koşula bakmaksızın istenilen değeri robot sayısal çıkışlarına yazar.

Komut şu şekilde kullanılır: OD parametre1 [, parametre2] [, parametre3]

Parametre1: Robot sayısal çıkışına yazılmak istenen sayısal değerdir. 16 bit uzunlukta olabilir ve -32768 ile 32767 arası herhangi bir değer girilebilir.

Parametre2: İsteğe bağlı kullanılacak bir parametredir. Değeri değiştirilmek istenen robot çıkışının başlangıç bit adresi tanımlanır. 0 ile 32767 arası değer girilebilir. Hiçbir değer girilmezse fabrika ayarı olarak değer 0'dır.

Parametre3: İsteğe bağlı kullanılacak bir parametredir. Değeri değiştirilmek istenen robot çıkışının bit sayısı tanımlanır. 1 ile 16 arası değer girilebilir. Hiçbir değer girilmezse fabrika ayarı olarak değer 16'dır.

Örneğin:

10 OD &FFFF ;0. bitten itibaren 16 adet robot çıkışını aktif (lojik 1) yapar.

20 OD &FFFF,8 ;8. bitten itibaren 16 adet robot çıkışını aktif (lojik 1) yapar.

30 OD &FFFF,8,4 ;8. bitten itibaren 4 adet robot çıkışını aktif (lojik 1) yapar.

3.1.7.3. COSIMIR Industrial Yazılımı ve Kullanımı

COSIMIR Industrial yazılımı, robotik imalat hücresi tasarımı için üretilmiş bir yazılımdır. Tasarıma yazılım içerisinde önceden tanımlanmış olan robotlar, tutucular (gripper), yaklaşım algılayıcıları (proximity sensor), pnömatik silindirler, konveyör hatları gibi imalat bileşenleri de eklenebilir. Aynı zamanda imalat hücresindeki bileşenleri denetleyen kontrol yazılımının da oluşturulması için bir geliştirme ortamı (IDE) sunar. COSIMIR Industrial yazılımında imalat hücresi denetim işini robot denetleyici üstlendiği

temel alınmıştır. Diğer bir ifadeyle, robot denetleyici imalat hücresindeki imalat bileşenlerini denetleyen merkezi birim olarak kabul edilir. Bu nedenle COSIMIR Industrial yazılımı ile imalat hücresine ait kavramsal tasarım yapıldıktan sonra, robotun programlanmasına yönelik çalışmalar da yürütülebilir. Robota konumlar öğretilir, program geliştirilebilir, geliştirilen programın benzetimi yapılabilir, program robot denetleyiciye yüklenebilir, denetleyicide yüklü bir program bilgisayara aktarılabilir, aktarılan program üzerinde değişiklik yapılabilir. Yazılımın FMS100 bünyesinde yüklü sürümü Mitsubishi marka robot programlanması için kullanılan Movemaster Command ve Melfa Basic dillerinin çeşitli sürümlerini desteklemektedir.

3.1.7.3.1. Yeni Bir Proje Oluşturma

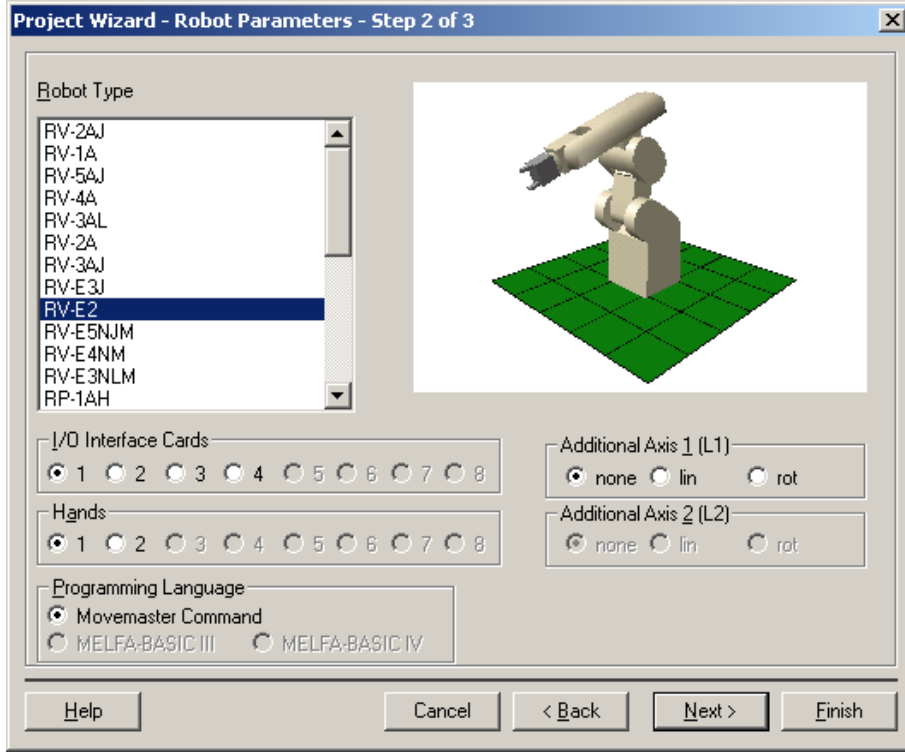
COSIMIR Industrial içerisinde yapılacak işler birer proje olarak tanımlanmaktadır. Oluşturulan proje kapsamında bir imalat hücresinin tüm imalat bileşenleri tanımlanabilir ve imalat hücre denetleyicisi olarak kabul edilen robot programlanabilir.

Yeni bir proje aşağıdaki adımlar sırasıyla takip edilerek oluşturulabilir:

- 1) COSIMIR Industrial yazılımında “File>Project Wizard” seçilir.
- 2) Ekranı gelen “Step 1” isimli pencerede;
 - “Project Name” kutucuğuna bir proje adı yazılır.
 - “Program Name” kutucuğuna 40’den büyük olacak bir program adı yazılır. 40’den küçük rakamlarda isime sahip programlar FMS100 sistemi tarafından kullanılmaktadır. İsimlerde çalışma olması, FMS100’ün çalışmasında hatalara sebep olabilir.
 - “Created by” kutucuğuna programın kimler tarafından oluşturulduğu yazılır.
 - “Initials” kutucuğuna programı oluşturanların varsa unvanları yazılır.
 - “Description” kutucuğuna oluşturulan programda neler yapıldığı, neyin amaçlandığı ve varsa program ile ilgili notlar yazılır. Bu bölüm ne kadar açıklayıcı yazılırsa, sonradan program üzerinde değişiklik yapılmak istendiğinde bu bilgiler oldukça faydalı olacaktır.
- 3) Pencerenin örnek olarak doldurulmuş hali Şekil 3-64’de görüldüğü gibidir. Tüm kutucuklar doldurulduktan sonra “Next” butonuna basılır.

Şekil 3-64. Örnek olarak doldurulmuş proje sihirbazı penceresi görüntüsü.

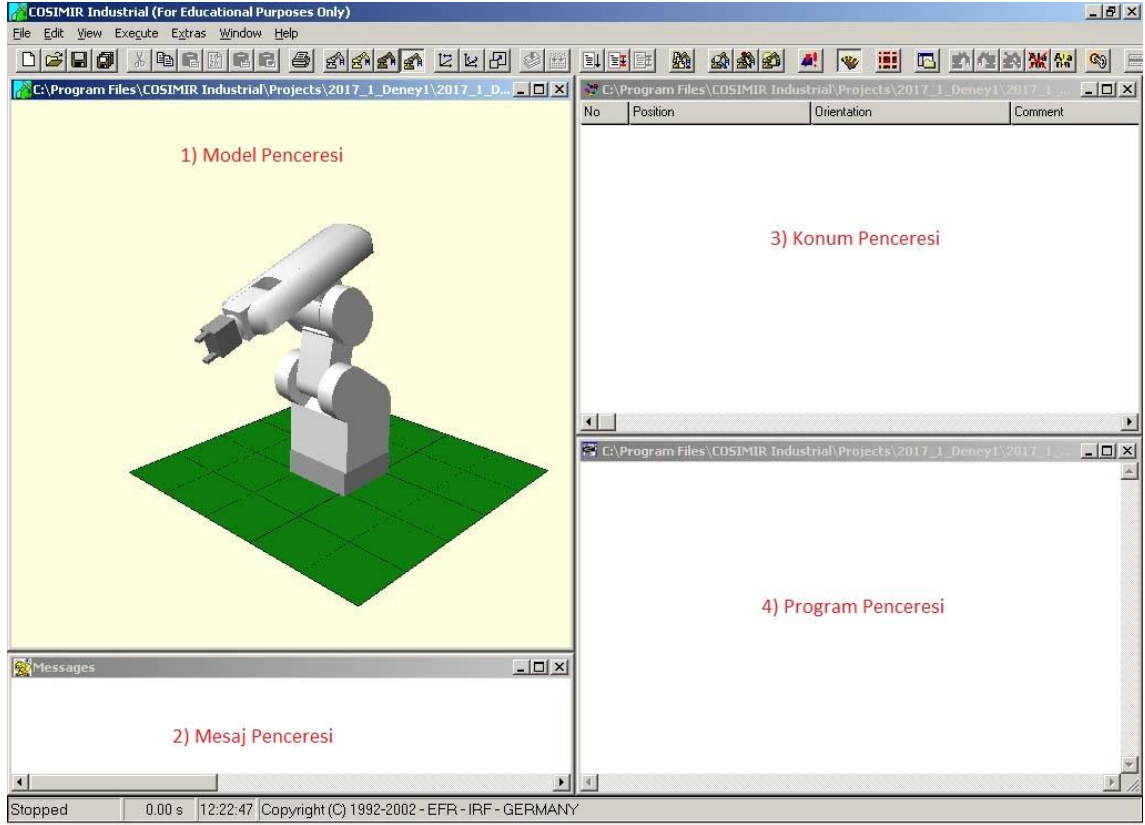
- 4) Ekranı gelen “Step 2” isimli pencereadaki parametreler;
 - Robot modeli “Robot Type”: RV-E2,
 - Robotun kullandığı arayüz kartı numarası “I/O Interface Cards”: 1,
 - Kullanılacak programlama dili “Programming Language”: Movemaster Command,
 - İlave eksen ve tipi (doğrusal, döner) “Additional Axis 1 (L1)”: none olacak şekilde belirlenir.
- 5) FMS100 Montaj istasyonu robotu için belirlenmiş örnek proje sihirbazı penceresi Şekil 3-65’de görüldüğü gibidir. Son olarak “Finish” butonu tıklanarak proje oluşturulur.



Şekil 3-65. Montaj istasyonu parametrelerini içeren proje sihirbazı penceresi görüntüsü.

6) Ekranı Şekil 3-66'deki gibi boş bir proje penceresi gelir. Proje penceresi dört alt pencereden oluşur:

- Model penceresi: Sol üst tarafta görüntülenen pencerede robot benzetimi görüntülenir. Benzetim sayesinde yazılan robot programı ve pozisyonların erişilebilirliği test edilebilir. Projede kullanılan modelin dosya adı "model_adı.MOD" şeklinde verilir.
- Mesaj Penceresi: Sol alt tarafta görüntülenen pencerede, COSIMIR Industrial tarafında gönderilen komutlara robotun gönderdiği cevaplar görüntülenir.
- Konum penceresi: Sağ üst tarafta görüntülenen pencerede robot programında kullanılacak konumlar yer alır. Projede oluşturulan konum dosyası adı "pencere_adı.POS" şeklinde verilir.
- Program penceresi: Sağ alt tarafta görüntülenen pencerede robot programına ait kodlar yer alır. Projede oluşturulan program dosyası adı "program_adı.MRL" şeklinde verilir.



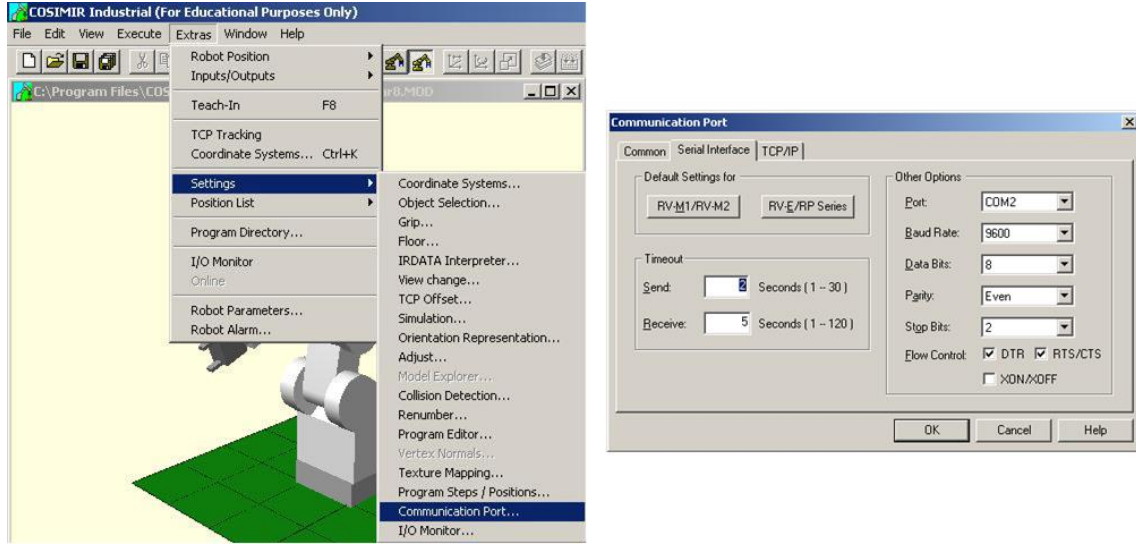
Şekil 3-66. Boş bir COSIMIR Industrial yazılımı proje penceresi.

3.1.7.3.2. Bilgisayar ile Robot Arası Bağlantı Kurma

COSIMIR Industrial yazılımıyla robot denetleyiciyi haberleştirmek için, öncelikle iletişim ayarlarının yapılması gereklidir. Robot kontrolör ile bilgisayar RS232C protokolü üzerinden haberleşmektedir.

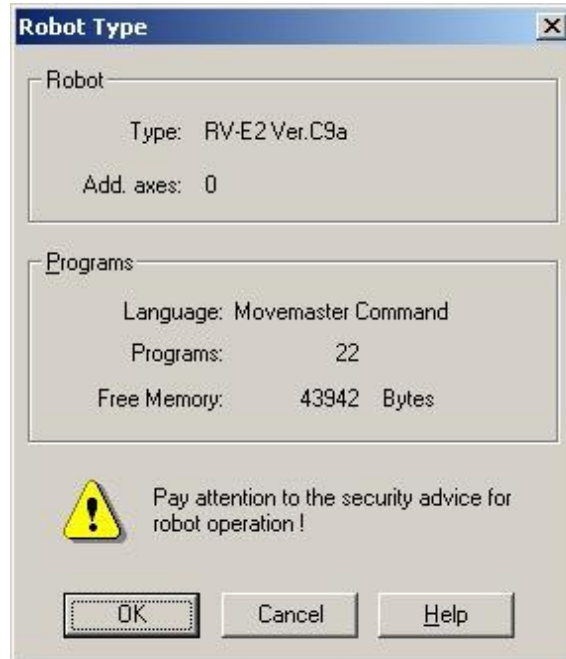
- 1) COSIMIR Industrial yazılımında Şekil 3-67'deki gibi “Extras > Settings > Communication Port” seçilir.
- 2) Şekil 3-67'deki “Communication Port” penceresinden “Serial Interface” sekmesine girilerek ayarla şu şekilde yapılır:
 - “Port: COM2”,
 - “Baud Rate: 9600”,
 - “Data Bits: 8”,
 - “Parity: Even”,
 - “Stop Bits: 2”,
 - “Flow Control: DTR, RTS/CTS”.

Not: FMS100 robot programlama bilgisayarının “COM1” seri portu arızalı olduğu için “COM2” kullanılmaktadır.



Şekil 3-67. COSIMIR Industrial iletişim ayarları penceresi.

- 3) Sonrasında “Execute > Init Connection” yapılarak robot ile bağlantı kurulur. Bağlantı başarıyla kurulduğunda ekranda Şekil 3-68’deki mesaj görüntülenir. “OK” butonuna basıldıktan sonra robot iletişime hazır demektir. Robota proje yüklenebilir (download) veya robottaki proje bilgisayara geri yüklenebilir (upload).



Şekil 3-68. Robot ile bağlantı başarıyla kurulduğunda ekrana gelen bilgi mesajı.

3.1.7.3.3. JOG İşlemi İle Robotu Hareket Ettirme

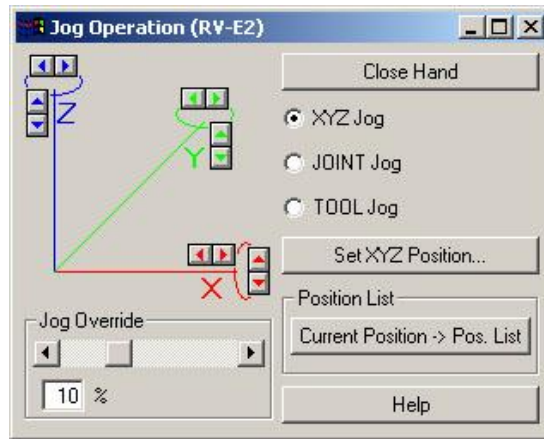
COSIMIR Industrial yazılımıyla robot denetleyici arası bağlantı kurulduktan sonra JOG işlemi ile robot hareket ettirilebilir. Böylelikle robotun eli istenilen konumlara götürülebilir, robot eğitilebilir.

- 1) COSIMIR Industrial yazılımında “Execute > JOG Operation” seçilir veya “F8” tuşuna basılır.
- 2) Ekranı gelen “JOG Operation” penceresinden öncelikle robotun hareket enterpolasyon tipi seçilir:
- 3) “JOINT Jog”: Eksenel enterpolasyon tipidir. Kullanıcı her bir eksenin ulaşmasını istediği açıyı elle kendisi belirler. Bu tip seçildiğinde Şekil 3-69’deki pencerede her bir eklemi (Waist, Shoulder, Elbow, Twist, Pitch, Roll) tek tek ileri/geri hareket ettirebilmek için butonlar görüntülenir.



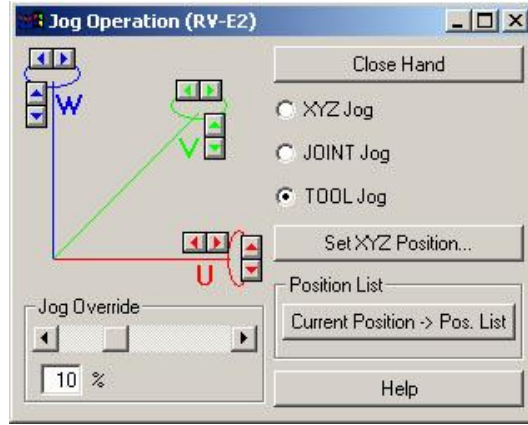
Şekil 3-69. “JOINT Jog” tipi hareket penceresi.

- 4) “XYZ Jog”: Robot tabanının tam orta noktasını merkez (orijin, sıfır noktası) kabul eden enterpolasyon tipidir. Kullanıcı robotun uç noktasının (elinin) ulaşmasını istediği konumu tabandaki merkeze göre belirler. Her bir eksenin açısını robot kontrolör kendisi hesaplayarak hareket eder. Bu tip seçildiğinde Şekil 3-70’deki pencerede Z eksenı yukarı, X eksenı sağa ve Y eksenı pencereden içeri olacak şekilde koordinat düzlemi ve robot ucunu XYZABC eksenlerinde ileri/geri hareket ettirebilmek için butonlar görüntülenir.



Şekil 3-70. “XYZ Jog” tipi hareket penceresi.

- 5) “TOOL Jog”: Mantık olarak XYZ Jog’a benzemekle birlikte robot takımının uç noktasını (elinin) merkez (orijin, sıfır noktası) kabul eden enterpolasyon tipidir. Kullanıcı robotun uç noktasının (elinin) ulaşmasını istediği konumu yen robotun ucuna göre belirler. Her bir eksenin açısını robot kontrolör kendisi hesaplayarak hareket eder. Bu tip seçildiğinde Şekil 3-71’deki pencerede W yukarı, U sağa ve V pencereden içeri olacak şekilde koordinat düzlemi ve robot ucunu kendi uç noktasına göre WUVTSR eksenlerinde ileri/geri hareket ettirebilmek için butonlar görüntülenir.



Şekil 3-71. “TOOL Jog” tipi hareket penceresi.

- 6) “Close Hand” / “Open Hand” butonlarına basılarak robot eli kapatılıp/açılabilir.
- 7) “Set Joint Coordinates” butonuna basılarak robotun uç noktasının ulaşması istenilen eksen açıları elle girilebilir. Yine “Set XYZ Position” butonuna basılarak robotun uç noktasının ulaşması istenilen eksen koordinatları elle girilebilir.
- 8) “Current Position -> Position List” butonuna basılarak robot uç noktasının o için bulunduğu koordinatlar robot programının konum listesine eklenir.
- 9) “Jog Override” bölümü kullanılarak robotun hareket hızı değiştirilebilir.

3.1.7.3.4. Robot Programlama

Bir robot projesi konum listesi ve program kodları olmak üzere iki temel bölümden oluşur. Konum listesi robotun hareketi sırasında uğrayacağı tüm noktalara ait koordinatları içerir. Robotun sahip olduğu serbestlik derecesine (eksen sayısı) göre noktalara ait konum bilgileri de değişiklik gösterir. Örneğin 5 eksen bir robot için tanımlanacak bir konum bilgisi XYZAC eksen koordinatlarından oluşabilirken (robotun mimarisine göre XYZAB, XYZBC, vb. olabilir); 6 eksen bir robot için konum bilgisi XYZABC eksen koordinatlarından oluşur.

Program kodları ise konum listesinde kayıtlı konumlar arası ilişkiyi tanımlamak için kullanılır. Robot uç noktasının hareketi sırasında listedeki hangi konumlara, hangi sırada

gideceği, robot elinin açılıp kapanması, harici giriş/çıkışların kontrolü, koşullar ve döngülerin oluşturulması gibi işlevler program kodları ile tanımlanır. Kodlamada kullanılacak programlama dili robotun marka ve modeline göre değişiklik gösterir. FMS100 bünyesindeki robotlar Movemaster Command ve MelfaBasic IV dillerini desteklemektedir.

Konumları Öğretme (Eğitme)

Robotun uç noktasının hareketi sırasında uğrayacağı tüm noktalara ait konum bilgilerinin kaydedilmesi işlemidir. Örneğin bir imalat işlemi sırasında, A konumunda bulunan X parçası alınıp B konumuna taşınabilir. Bir başka örnekte Y parçası üzerindeki C konumuna delik açılabilir. Bu işlemlerin robot ile gerçekleştirilmesi robot programının yazılabilmesi için öncelikle A, B ve C noktalarının koordinatlarının belirlenmesi gerekir. Sonrasında her bir nokta için en az 1 adet güvenli erişim konumu (A', B', C') belirlenir.

Güvenli erişim konumları; robot kolunun hareketi sırasında A, B, C konumları arasında geçiş yaparken oluşabilecek çarpmaları-kazaları önlemek için kullanılır. Uygulamaya göre değişiklik göstermekle birlikte, güvenli erişim konumları genellikle hedef konumun takım ekseninde 50-100 mm üstünde bir konum olarak belirlenir.

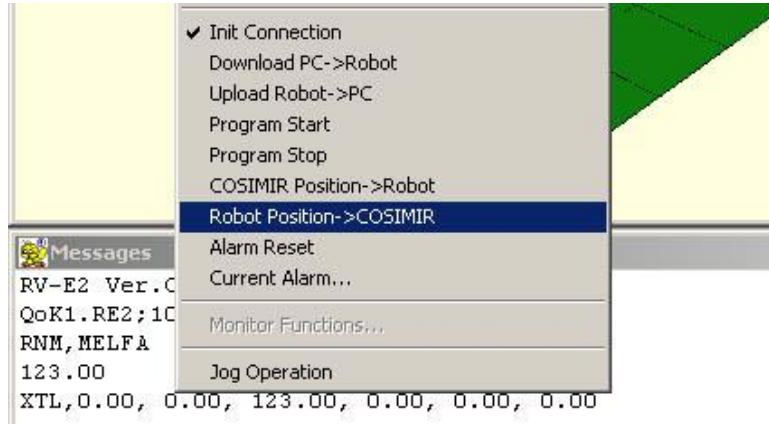
Örneğin program içeriği gereği robot uç noktasının A noktasına erişmesi gerekiyorsa, uç nokta öncelikle güvenli olarak kabul edilen A' noktasına yönlendirilir. Sonrasında doğrusal enterpolasyon ile ve yavaş hareketlerle A noktasına ulaşır. Güvenli erişim konumu ile hedef konum arası hareket her zaman doğrusal enterpolasyon ve düşük hızlarda gerçekleştirilir. Eğer robotun A noktasından B noktasına doğru hareketi gerekiyorsa; yine robot öncelikle güvenli olan A' noktasına ve sonrasında güvenli olan B' noktasına eriştikten sonra B noktasına yönlendirilir. Robot uç noktası denildiğinde, çeşitli amaçlarla üretilmiş olan tutucu (gripper) el veya matkap gibi takımların en uç noktası akla gelmelidir.



Şekil 3-72. Robot uç noktasının A noktasından B noktasına hareketi.

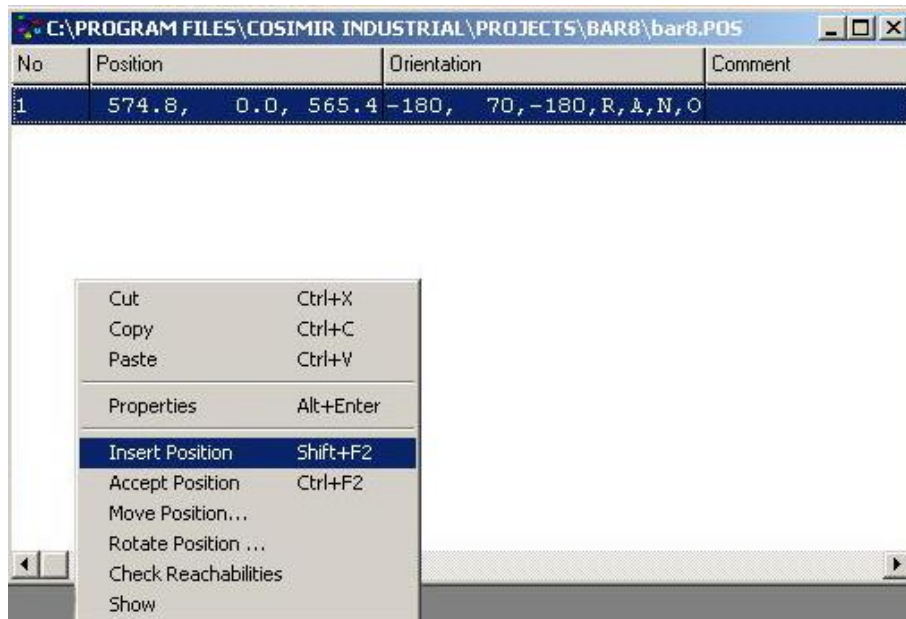
COSIMIR Industrial yazılımı ile robota konumları öğretebilmek için, robot uç noktasının program kapsamında izlenmesi istenen yörüngedeki tüm hedef ve ara noktalara tek tek

götürülerek, noktalara ait konum bilgilerinin kaydedilmesi gerekir. Robot uç noktası; gerek el terminali (Teach Pendant - TP) gerekse Bölüm 3.1.7.3.3’da anlatıldığı gibi COSIMIR yazılımı üzerinden JOG işlemiyle hareket ettirilebilir. Robot uç noktası hedef konuma ulaştığında, ulaşılan konum bilgisi öncelikle COSIMIR’e aktarılmalıdır. Bunun için Şekil 3-73’de görüldüğü gibi “Execute > Robot Position -> COSIMIR” tıklanır. Benzetim penceresindeki sanal robot, gerçek robotun duruş biçimini kopyalar.



Şekil 3-73. Robot eklem koordinat bilgilerinin COSIMIR yazılıma aktarılması.

Sonrasında hedef konuma ait aktarılan koordinat bilgileri konum penceresine kaydedilir. Bunun için Şekil 3-74’de görülmekte olan konum penceresi üzerinde sağ tuş tıklanarak ekrana gelen seçenek penceresinde “Insert Position” tıklanır veya klavyeden “Shift+F2” tuşlarına basılır. Böylelikle konum kaydedilir.



Şekil 3-74. COSIMIR Industrial projesindeki konum penceresi.

COSIMIR Industrial yazılımında oluşturulmuş örnek bir konum bilgisi penceresi Şekil 3-75’de görülmektedir. Pencerede görülen kutucuklar ve açıklamaları şu şekildedir:

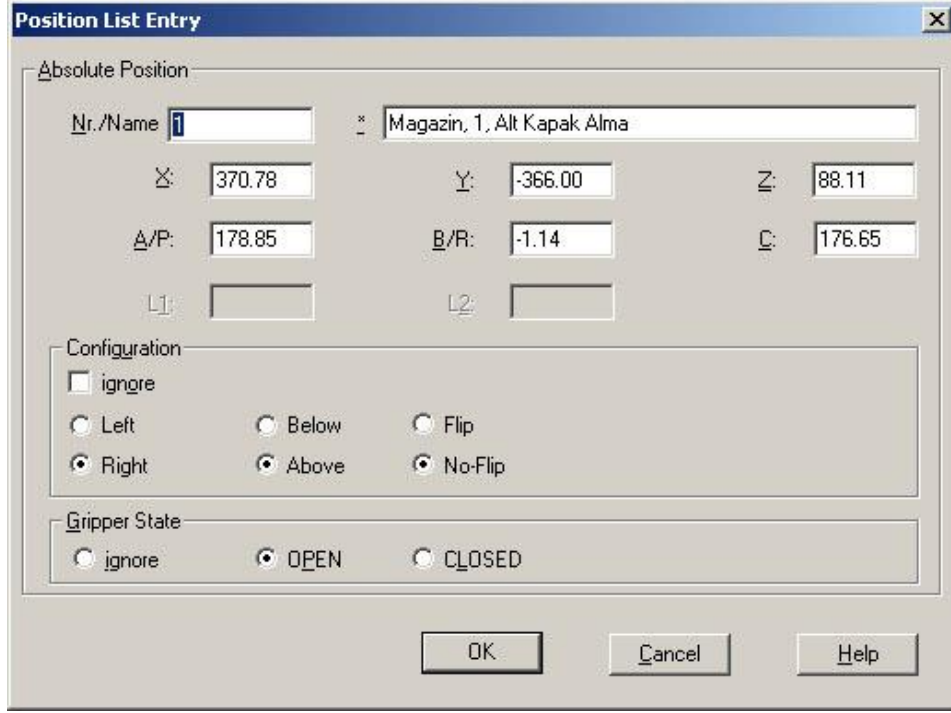
- “Nr./Name” : Konuma verilen isimdir. İsim harflerden oluşabileceği gibi tamsayı numaralar şeklinde de olabilir. Robot için program kodlarında hareket edilecek konumlar bu isimler ile belirtilecektir. Bu nedenle bir proje içerisindeki her bir konuma birbirinden farklı isim verilmesi gereklidir.
- “*”: Konum ile ilgili açıklamanın yazıldığı kutucuktur. Bu kısım program yazımı sırasında hangi konumun ne için olduğunun hatırlanması ve anlaşılması için oldukça önemlidir. Konum listeye kaydedilirken mümkün olduğu kadar açık ve yalın şekilde açıklama yazılmalıdır.
- “XYZ”: Listeye kaydedilmiş olan konumun, merkezi koordinat sistemine göre XYZ değerlerinin gösterildiği kutucuklardır. Merkezi koordinat sistemine göre robotun merkezi ($X=0$, $Y=0$, $Z=0$ noktası), robot tabanının tam orta noktasıdır. Bu kutucukta yazan değerler milimetre cinsinden merkeze olan uzaklığı tanımlar.
- “ABC”: Listeye kaydedilmiş olan konumun, merkezi koordinat sistemine göre ABC değerlerinin gösterildiği kutucuklardır. ABC değerleri, merkezi koordinat sistemine göre robotun merkezinde bulunduğu varsayılan düzlem ile yapılan açıyı temsil eder. Bu kutucukta yazan değerler derece cinsindendir.

“Configuration” bölümünde görüntülenen parametreler robotun istenilen noktaya erişimi sırasında hangi yönden yaklaşacağını belirler. Bazı durumlarda, erişilmek istenen nokta robotun erişim uzayı içerisinde olduğu halde, robot ters yönden gitmeye çalışarak istenilen noktaya erişemeyip hata verebilir. Bir nevi, dilimizde bir deyim olan “kulağı tersten tutmaya çalışmak” gibidir. Buna robot programcılığında “ters hareket” denir. Bu bölüme eklenmiş parametreler sayesinde robot programcısına ters hareketin önüne geçebilecek tanımlamalar yapma imkanı sağlanır.

“Ignore” seçilirse robot denetleyicisi kendisi en uygun yolu seçerek hedefe yaklaşmaya çalışacaktır. Tabi bu durumda ters hareket ihtimali de vardır.

- “Left/Right”: Hedef noktaya soldan mı yoksa sağdan mı yaklaşılacağını belirler.
- “Below/Above”: Hedef noktaya üstten mı yoksa alttan mı yaklaşılacağını belirler.
- “Ters/Düz”: Robotun mevcut yaklaşım yönünü tüm açılardan tersine çevrilip çevrilmeyeceğini belirler.
- “Gripper State”: Hedef konumda robot tutucu elinin açık (OPEN) mı, kapalı (CLOSED) mı yoksa bir önceki durumunu korur şekilde (ignore) olacağını belirler. “Gripper State” parametresinde ne olursa olsun, robot programında

yazılacak tutucu el açma/kapama komutları hiyerarşik olarak bu parametrelerin üzerindedir. Kodda ne yazarsa o gerçekleştirilir.



The image shows a software window titled "Position List Entry". It contains several input fields and a configuration section. The "Absolute Position" section has fields for "Nr./Name" (containing "1"), "X" (370.78), "Y" (-366.00), "Z" (88.11), "A/P" (178.85), "B/R" (-1.14), and "C" (176.65). There are also empty fields for "L1" and "L2". The "Configuration" section has a checkbox for "ignore" (unchecked) and three radio button groups: "Left" (selected), "Below", and "Flip"; "Right", "Above", and "No-Flip". The "Gripper State" section has three radio buttons: "ignore" (selected), "OPEN", and "CLOSED". At the bottom are "OK", "Cancel", and "Help" buttons.

Şekil 3-75. Örnek bir konum bilgisi penceresi.

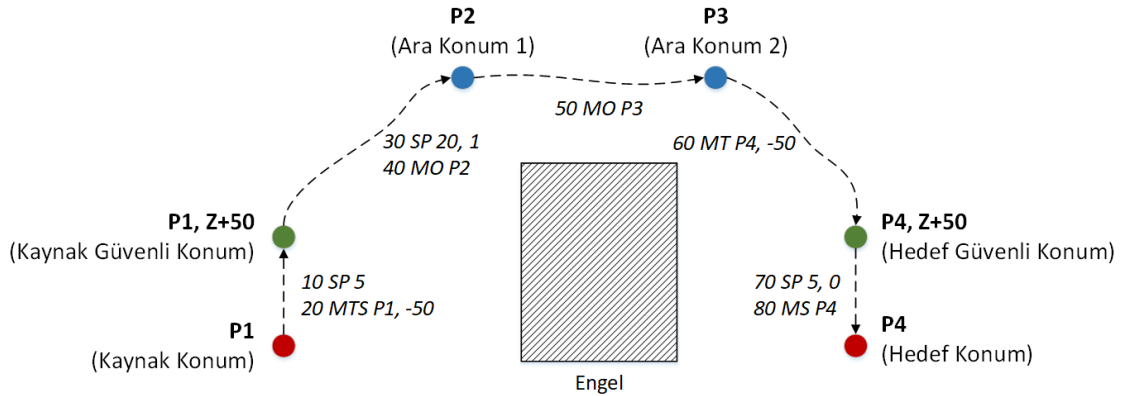
Robot Programı Kodlarını Yazma

Pozisyon penceresinde tanımlanmış olan konumlara robotun nasıl ve hangi sırayla erişeceğini belirlemek üzere program yazılır. FMS100 bünyesinde bulunan robot denetleyici Movemaster Command veya MelfaBasic IV programlama dillerini desteklemektedir. Bu diller Q-Basic programlama dili tabanlı oldukları için, her program satırına 10'dan başlayarak 10'ar artacak şekilde numara verilir. Satır numaralarının 10'ar artımlı verilmesinin sebebi, programı geliştirirken gereken durumlarda kod satırı aralarına sonradan kod satırları eklenebilmesini sağlamaktır.

Robot programı aracılığıyla, önceden tanımlanmış konumlara hangi sıra ve hızla gidileceği, hareketin eksenel mi yoksa doğrusal enterpolasyon ile mi yapılacağı, hareket sırasında tutucu elin açık/kapalı olacağı gibi parametreler belirlenir. Robotun yörüngesi planlanırken karar verilmesi gereken en önemli konulardan biri; nerede eksenel, nerede doğrusal enterpolasyon yapılacağıdır. Eksenel enterpolasyonda bir noktadan diğer bir noktaya gidilirken, robot eklemlerinin en az hareket edeceği yolu tercih eder. Bu nedenle de eksenel enterpolasyonda hareket hızı yüksektir. Fakat robot uç noktası düz bir çizgi üzerinde hareket etmez ve 3 boyutlu düzlemde eğimli bir yörünge takip eder. Doğrusal

enterpolasyonda ise bir noktadan diğer bir noktaya gidilirken, robot uç noktasının en kısa yoldan hedefe ulaşması sağlanır. İki nokta arasındaki en kısa yol ise bir doğrudur. Bu nedenle doğrusal enterpolasyonda robot uç noktası düz bir çizgi üzerinde hareket eder. Fakat bu düz hareketi sağlamak için de robot eksenlerinin, eksenel enterpolasyona göre daha çok hareket etmesi, robot denetleyicinin daha fazla hesaplama yapması gerekir. Bu durum da doğrusal enterpolasyondaki hareket hızlarının daha düşük olmasına neden olur. Her iki enterpolasyonun birbirine göre olan avantajları ve dezavantajları göz önüne alındığında; çarpma riskinden dolayı düşük hızlı ve hassas hareket gerektiren yerlerde doğrusal enterpolasyon, yüksek hızlı ve çarpma riski olmayan yerlerde ise eksenel enterpolasyon tercih edilir.

Aşağıdaki Şekil 3-76'de P1 konumundan P4 konumuna yapılan hareket ile ilgili örnek uygulama görülmektedir. Dikkat edilirse kaynak ile hedef arasında bir engel vardır. Robotun hareketi sırasında engele çarpmaması için P2 ve P3 ara konumları tanımlanmıştır.



Şekil 3-76. Bir engeli aşarak iki nokta arası robot hareketine örnek uygulama.

Robot hareketleri tanımlanırken asıl işin yapıldığı, asıl ulaşılacak istenen noktaların (örnekte P1 ve P4) üzerlerinde güvenli noktalar (örnekte P1, Z+50; P4, Z+50) belirlenir. Bu noktalar, robot uç noktası buralara ulaşırken, herhangi bir nesneye çarpmayacağı yükseklikte olmalıdır. Güvenli noktadan asıl noktalara gidilirken doğrusal enterpolasyon kullanılarak, çarpma riskine karşı düşük hızla ve hassas hareket edilir. Güvenli noktalardan sonra diğer ara noktalara gidilirken ise eksenel enterpolasyon kullanılarak hareket hızının yüksek olması tercih edilir.

Örnek uygulama için geliştirilmiş robot programı Şekil 3-77'deki gibidir. Program geliştirilirken kullanılan Movemaster Command kodları ve detaylarına Bölüm 3.1.7.2'den ulaşılabilir.

10 SP 5	;Hız ayarı seviye 5
20 MTS P1, -50	;P1'in 50 mm üzerine (Z+50) doğrusal enterpolasyonla hareket eder.
30 SP 20, 1	;Hız ayarı seviye 20, sürekli hareket aktif.
40 MO P2	;P2'ye eksenel enterpolasyon ile hareket eder.
50 MO P3	;P3'e eksenel enterpolasyon ile hareket eder.
60 MT P4, -50	;P4'ün 50 mm üzerine (Z+50) doğrusal enterpolasyonla hareket eder.
70 SP 5, 0	;Hız ayarı seviye 5, sürekli hareket pasif.
80 MS P4	;P4'e doğrusal enterpolasyon ile hareket eder.

Şekil 3-77. Örnek uygulama için Movemaster Command dili ile geliştirilmiş robot programı.

COSIMIR Industrial yazılımı ile robot programına ait hazırlanan konum (*.POS) ve program (*.MRL) dosyaları, benzetim yapmadan veya robot kontrolöre yüklemiden önce, kaydedilmelidir (File > Save All).

3.1.7.3.5. Program Benzetimi Yapma

İmalat için robotun uğrayacağı konumlar tek tek öğretilip program kodları da yazıldıktan sonra, projenin doğruluğunun sınanması gereklidir. Çünkü yazılan program kodları veya konum tanımlamalarında yapılmış olabilecek hatalar; hem robota, hem de imalat hücresine ve bileşenlerine zarar verebilir. Robot istenen görevleri yerine getiremeyebilir. Bu nedenle belirlenen robot pozisyonları ve hazırlanan robot programı, robot kontrolöre yüklenmeden önce benzetim yapılarak sınanmalıdır.

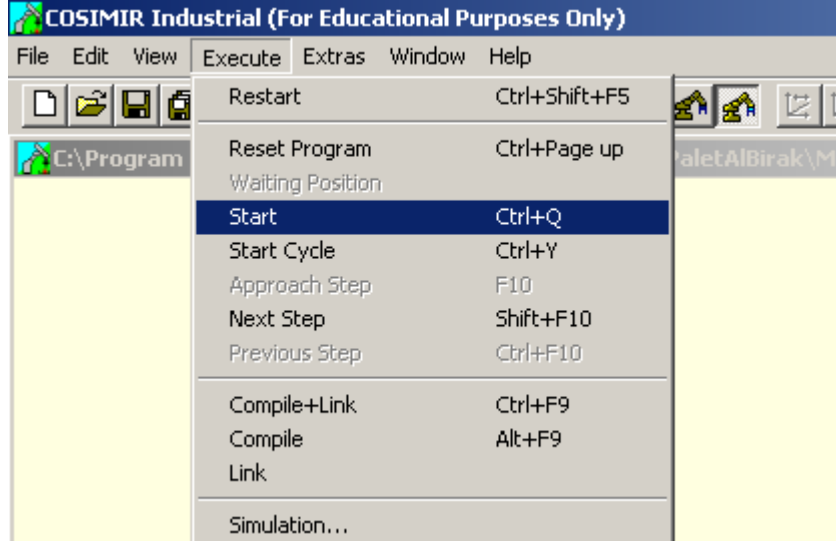
COSIMIR Industrial programındaki model penceresi içerisinde robotun durumunu ve hareketlerini gösteren 3 boyutlu sanal bir robot modeli vardır. Robotun eksen konum bilgileri bu robot modeline aktarılabilir (Robot Position -> COSIMIR) veya robot modelindeki konum bilgileri gerçek robota aktarılabilir (COSIMIR Position -> Robot). Böylelikle gerçek robot duruşuyla robot benzetim modeli arasında eşleşme sağlanabilir. Hazırlanan program derlendikten (Compile Alt+F9) sonra benzetim işlemi başlatılabilir (Start Alt+Q).

Benzetim işlemi adım adım yapılabilirdiği gibi (Next Step Shift+F10) bir döngü içerisinde sürekli çalışması da (Start Cycle Ctrl+Y) sağlanabilir. İstendiği zaman robot programı yeniden başa alınabilir (Reset Program Ctrl+Page Up). Benzetim yapmanın güzel tarafı,

COSIMIR model penceresindeki robot modeli hareket ederken aynı zamanda program penceresi içerisinde, yürütülmekte olan program satırı aktif edilir. Böylelikle, robotun hangi satırdaki hangi kod ile hangi hareketi yaptığı görülebilir. Eğer varsa, program satırlarında veya belirlenmiş olan pozisyonlarda olabilecek hatalar kolaylıkla tespit edilebilir.

Hedeflenen robot programına ait konumlar öğretilip program kodları yazıldıktan sonra benzetim için aşağıdaki işlem basamakları sırasıyla yürütülmelidir:

- 1) Robotun eksen konumları benzetim modeline aktarılmalıdır. Bunun için “Execute > Robot Position -> COSIMIR” tıklanır.
- 2) Robot programı derlenir. Bunun için Şekil 3-78’deki menüden de görüldüğü gibi “Execute > Compile” tıklanır veya klavyeden “Alt+F9” tuşlarına basılır.
- 3) Her ihtimale karşı robot programı başa alınır. Bunun için Şekil 3-78’deki menüden de görüldüğü gibi “Execute > Reset Program” tıklanır veya klavyeden “Ctrl+Page Up” tuşlarına basılır.
- 4) Benzetim işlemi başlatılır. Bunun için Şekil 3-78’deki menüden de görüldüğü gibi “Execute > Start” tıklanır veya klavyeden “Alt+Q” tuşlarına basılır. Robot programı satır satır işletilirken sonuçlar robot benzetim modeli üzerinde görülür.
- 5) İstenirse benzetim işlemi durdurulana kadar sürekli çalışacak şekilde de başlatılabilir. Bunun için Şekil 3-78’deki menüden de görüldüğü gibi “Execute > Start Cycle” tıklanır veya klavyeden “Ctrl+Y” tuşlarına basılır.
- 6) İstenirse robot programı her seferinde tek bir satır yürütülecek şekilde de başlatılabilir. Bunun için Şekil 3-78’deki menüden de görüldüğü gibi “Execute > Next Step” tıklanır veya klavyeden “Shift+F10” tuşlarına basılır. Bu seçenek her seçildiğinde sadece bir satır yürütülür ve sonucu robot benzetim modelinde görülür.



Şekil 3-78. COSIMIR Industrial yazılımı ile robot program benzetimi menüsü.

3.1.7.3.6. Projeyi Bilgisayardan Robota Yükleme

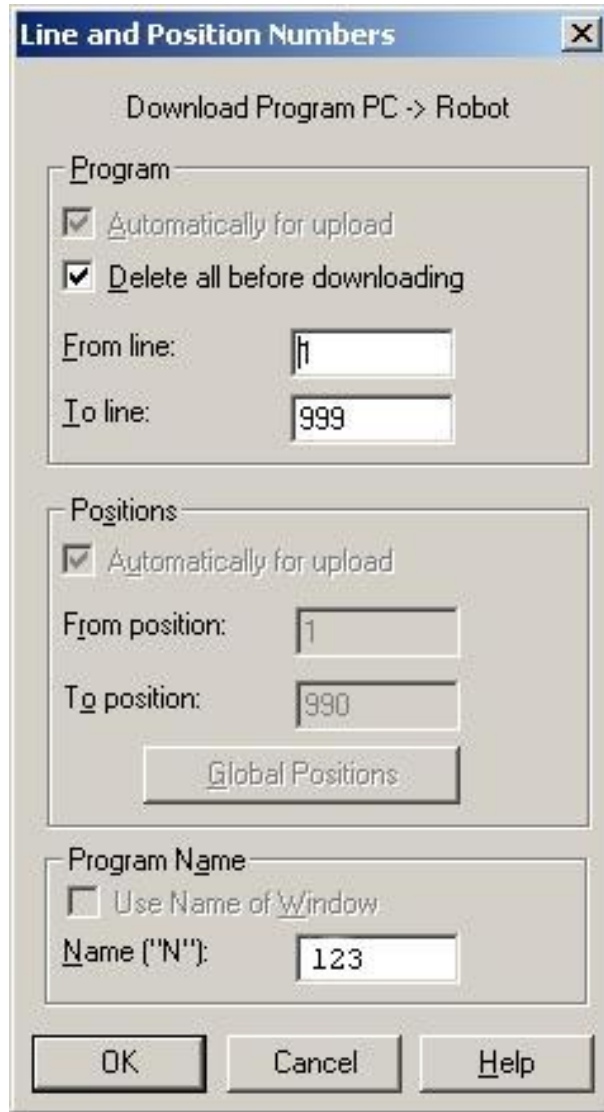
Robot programı oluşturulup benzetim işlemi de tamamlanıp, eğer varsa programda gerekli düzeltmeler de yapıldıktan sonra, program robota yüklenebilir. COSIMIR Industrial yazılımı çok dosyalı arayüz (MDI - Multiple Document Interface) yapısına sahiptir. Bu nedenle menülerden veya araç çubuklarından verilen kaydet, yeni, vb. komutlar sadece o an için aktif olan dosya için geçerlidir. Örneğin konum penceresi aktifken “Execute > Download PC -> Robot” seçeneği tıklanırsa konum dosyası (*.POS), program penceresi aktifken aynı seçenek tıklanırsa program dosyası (*.MRL) robota yüklenir.

Hata denetimi ve benzetim işlemi yapılmış programı robota yükleyebilmek için aşağıdaki işlem basamakları sırasıyla yürütülmelidir:

- 1) Bölüm 3.1.7.3.2’de anlatıldığı şekilde bilgisayar işle robot arası bağlantı kurulur.
- 2) Robota yüklenmek istenen program penceresi tıklanarak aktif hale getirilir.
- 3) “Execute > Download PC -> Robot” seçeneği tıklanır. Ekran Şekil 3-79’deki pencere gelir.
- 4) “Program Name” kısmına yüklenmek istenen programın adı yazılır. Program adına sayısal veya karakter dizisi girilebilir (örneğin “ABC123”). Fakat geliştirilen robot programının ileride FMS100 bünyesindeki PLC üzerinden de çalıştırılması isteniyorsa, sadece sayılardan oluşan bir isim vermek daha yerinde olacaktır. Program adı verilirken 100’den büyük sayısal ifadeler yazılmalıdır. Örneğin program adı Şekil 3-79’deki gibi “123” şeklinde verilebilir. Program adı 100’den aşağıda olan programlar FMS100 tarafından kullanılmaktadır. Bu nedenle FMS100’ün kullandığı

sistem programlarına zarar vermemek için, seçilecek program adları 100'den büyük sayılar şeklinde olmalıdır.

- 5) Pencerede "Delete all before downloading" seçeneği tıklanırsa, robot programı yüklenmeden önce, eğer daha önceden aynı isimde başka bir program yüklüyse, önce o program satırları silinir. Şekil 3-79'daki pencerede 1'den 999 numaralı satıra kadar silinecektir. Sonrasında yeni program satırları kontrolöre yüklenir.
- 6) "OK" tuşuna basıldığı anda yükleme başlar. Programın yüklenme durumu Mesaj Penceresi'nden izlenebilir.



Şekil 3-79. Robot projesinin program dosyasını robota yükleme penceresi.

- 7) Program dosyası robota yüklendikten sonra sıra konum dosyasını yüklemeye gelmiştir. Bunun için konum penceresi tıklanıp aktif hale getirilir.
- 8) "Execute > Download PC -> Robot" seçeneği tıklanır. Ekran Şekil 3-80'deki pencere gelir.

Line and Position Numbers

Download Position PC -> Robot

Program

☒ Automatically for upload

☒ Delete all before downloading

From line: 1

To line: 999

Positions

☒ Automatically for upload

From position: 1

To position: 990

Global Positions

Program Name

☐ Use Name of Window

Name ("N"): 123

OK Cancel Help

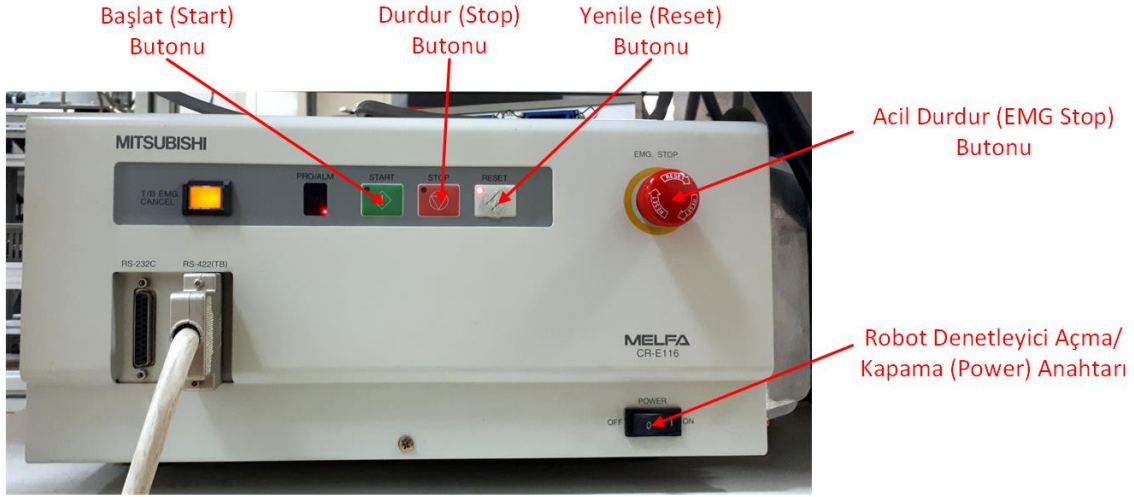
Şekil 3-80. Robot projesinin konum dosyasını robota yükleme penceresi.

3.1.7.3.7. Programı Başlatma/Yeniden Başlatma/Durdurma

Robotta yüklü olan programı başlatmadan önce, İŞ GÜVENLİĞİ açısından herkesin robotun çalışma alanı dışına çıkması; robotun beklenmeyen bir hareketine karşı önlem olarak bir kişinin Şekil 3-81’de görülmekte olan acil durdurma butonuna basabilecek şekilde hazırda beklemesi gereklidir.

COSIMIR Industrial yazılımı ile proje robota yüklendikten ve gerekli İŞ GÜVENLİĞİ ÖNLEMLERİNİ ALDIKTAN SONRA, COSIMIR Industrial yazılımı veya direk robot denetleyici üzerindeki butonlar aracılığıyla robot programı başlatılabilir / yeniden başlatılabilir / durdurulabilir.

- 1) Robot programını başlatmak için “Execute > Program Start” seçeneği tıklanır veya Şekil 3-81’deki yeşil renkli “Start” butonuna basılır.
- 2) Robot programını durdurmak için “Execute > Program Stop” seçeneği tıklanır veya Şekil 3-81’deki kırmızı renkli “Stop” butonuna basılır.
- 3) Robot programını başa almak için “Execute > Program Reset” seçeneği tıklanır veya Şekil 3-81’deki beyaz renkli “Reset” butonuna basılır.

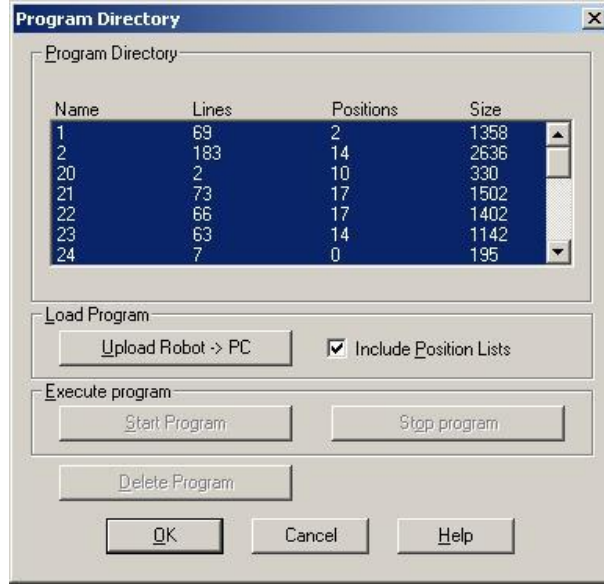


Şekil 3-81. RV-E2 model robot denetleyicinin ön taraftan görüntüsü.

3.1.7.3.8. Tüm Programları Görüntüleme ve Bilgisayara Yükleme

COSIMIR Industrial yazılımıyla robot denetleyici arası bağlantı kurulduktan sonra, robot denetleyicide yüklü bulunan tüm programlar görüntülenebilir. İstenirse seçilen programlar konum bilgileri ile birlikte bilgisayara aktarılabilir. Bunun için aşağıdaki işlem adımları izlenmelidir:

- 1) COSIMIR Industrial yazılımında “Extras > Program Directory” seçilir.
- 2) Ekranı gelen Şekil 3-82’de görülmekte olan pencerede robot kontrolörde yüklü bulunan programlar; isim, program satır sayısı, konum sayısı ve kapladıkları alan bilgileri görülecek şekilde görüntülenir.



Şekil 3-82. Program klasörü (Program Directory) penceresi görünümü.

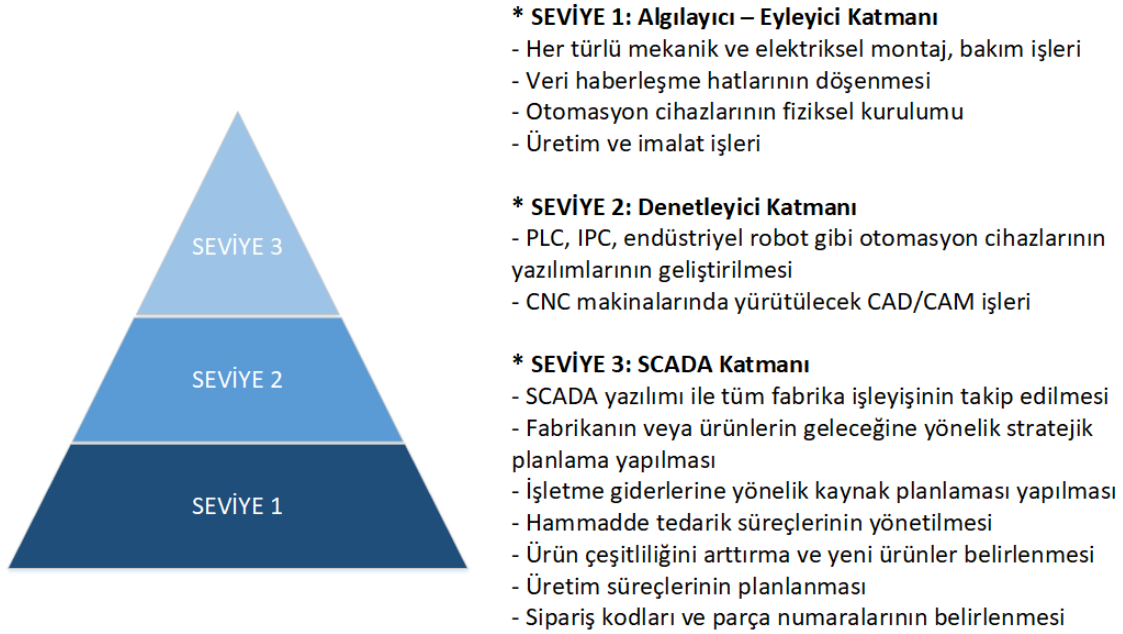
- 3) Şekil 3-82'deki pencere üzerinden, bilgisayara yüklenmek istenen program veya programlar seçilip "Upload Robot --> PC" tıklanır.
- 4) Eğer programlar yüklenmeden önce şekildeki pencerede "Include Position Lists" seçili olursa, kontrolörde yüklü bulunan pozisyon listeleri de programlarla birlikte bilgisayara aktarılır.
- 5) Bilgisayara yüklenen tüm program ve pozisyonlara "Program Files > COSIMIR Industrial > Projects" klasörü içerisinde ulaşılabilir. İstenilen program klasör içinden açılarak üzerinde gerekli düzeltmeler yapıp, kontrolöre geri yüklenebilir.

3.1.8. Fabrika Otomasyonu Hiyerarşik Kontrol Yapısı

İmalat hatları bir veya birden fazla imalat hücresinden oluşur. İmalat hücreleri birçok otomasyon cihazından meydana geldiği için otomasyon istasyonları olarak da adlandırılır. İmalat hücrelerinde bulunan otomasyon cihazları genellikle PLC, endüstriyel PC ve robot ile CNC makineleridir. Ayrıca yaklaşım algılayıcıları ve eyleyiciler gibi çevresel cihazlar da kullanılır. İmalat hücreleri arası malzemeler genellikle konveyör sistemleri ile taşınarak imalat hatları oluşturulur. Birbirinden farklı özellikte imalat hücreleri arasındaki sıralı çalışma düzenini imalat kontrol sistemi (İKS) denetler ve sağlar.

Fabrika otomasyonu birçok öğeden oluşan, genel anlamda oldukça karmaşık bir sistemdir. Bu karmaşık yapıyı düzene sokup etkin bir şekilde yönetebilmek için hiyerarşik bir yaklaşıma ihtiyaç vardır. Bu nedenle fabrika otomasyonu yapısal olarak SCADA (Supervisory Control and Data Acquisition), Denetleyici ve Algılayıcı - Eyleyici olmak

üzere üç katmanda değerlendirilir. Bazı kaynaklarda İşletme Kaynak Yönetimi (İKY) (Enterprise Resource Planning, ERP) ile Tedarik Zinciri Yönetimi (TZY) (Supply Chain Management, SCM), ayrı bir dördüncü katman olarak, SCADA katmanı üzerinde kabul edilir. Buna rağmen bu çalışmada fabrika otomasyonu yapısal piramidi Şekil 3-83’de görüldüğü gibi üç katmanlı olarak incelenmektedir.



Şekil 3-83. Fabrika otomasyonu yapısal piramidi ve katmanları.

Birinci katman (Seviye 1) olan Algılayıcı – Eyleyici katmanı, bir açıdan fabrikanın fiziksel katmanıdır. Fabrikayı oluşturan tüm birimler ve bu birimlere yönelik yürütülen görevler bu katmanda değerlendirilir. Örneğin bir imalat istasyonuna algılayıcı/eyleyici montajı, elektrik ve haberleşme hatlarının çekilmesi, CNC cihazının kullanılması, vinçlerin çalıştırılması; kısacası her türlü montaj, üretim ve imalat ile tamir ve bakım işleri bu katmanda yürütülür.

İkinci katman (Seviye 2) olan Denetleyici katmanı ise, imalat hücrelerinde bulunan otomasyon cihazlarının programlarının yazılması, test edilmesi ve iyileştirilmesi gibi işlerin yürütüldüğü katmandır. Ayrıca her türlü tasarım işi ile ARGE çalışmaları da bu katmanda yürütülür. Bu katmanda yürütülen tüm görevler Seviye 1’deki işleyişi doğrudan etkiler. Örneğin bir imalat hattındaki PLC, endüstriyel robot ve hatta talaşlı imalatta kullanılan CNC cihazlarının işletilmesine yönelik programlar da bu katmanda hazırlanır.

Üçüncü katman (Seviye 3) olan SCADA katmanı ise fabrikanın tüm denetim ve yönetim işlerinin yapıldığı katmandır. Bu katmanda iki temel yönetim yaklaşımı uygulanır. SCADA yazılımları ile fabrikada yürütülen imalat süreçlerine ait otomasyonun denetimi sağlanır. SCADA altyapısı sayesinde fabrikanın anlık olarak tükettiği enerji, hammadde gibi kaynaklar ile üretimin miktarı gibi veriler ölçülerek tek bir noktada toplanır. Toplanan veriler işletmenin yönetilmesi ve gelecek planlaması açısından stratejik öneme sahiptir. Bu nedenle temelde SCADA altyapısından elde edilen verileri kullanan İKY ve TZY'nin de SCADA katmanında yer aldığı kabul edilmektedir. Ayrıca fabrikada üretilecek ürünlere ait sipariş ve parça numaralarının belirlenmesi, imalat süreç planlarının oluşturulması gibi yönetimsel işler de yine bu katmanda yürütülür.

FMS100, sahip olduğu özellikler sayesinde fabrika otomasyonu yapısal piramidinin Seviye 1 ve Seviye 2 katmanlarının tamamını kapsamakla birlikte, Seviye 3 katmanını kısmen içermektedir. Seviye 1 kapsamında fiziken 6 adet imalat hücresinden oluşan örnek bir fabrika otomasyon sistemi sunar. Seviye 2 kapsamında imalat hücrelerinde bulunan PLC ve IPC gibi otomasyon denetleyicilerinin yanısıra, endüstriyel robot ve CNC torna/freze gibi imalat cihazlarının programlanabilmesine olanak sağlar. Seviye 3 kapsamında ise SCADA arayüzü üzerinden fabrika ve üretim süreçlerinin denetimi, yeni süreç planları ve görevler oluşturma, imalat veri tabanı kullanarak sipariş ve ürün kodları belirleme, imalat planlama gibi görevlerin yürütülebilmesini sağlar. TZY FMS100 kapsamında değildir.

3.1.8.1. COSIMIR Factory Yazılımı

FMS100'ün fabrika otomasyonuna yönelik özelliklere sahip olmasını sağlayan, arka plandaki yazılımsal güç ise COSIMIR Factory yazılımıdır. Yazılımın sahip olduğu alt yazılımlar ve birbirleriyle ilişkileri Şekil 3-84'deki gibi olup; alt yazılımların özellikleri kısaca aşağıda verilmiştir:

- COSIMIR Factory Simulation: Plant Editor yazılımı aracılığıyla, önceden hazırlanarak kütüphanesine kaydedilmiş hazır imalat hücreleri ve otomasyon cihazları kullanılarak, yeni imalat hatlarının tasarımı ve modellenmesi görevlerini yürütür. COSIMIR Factory Control yazılımı ile eş güdümlü çalışarak imalat hattında gerçekleşen olayların gerçekçi bir şekilde benzetimini yapar. Yazılımın kütüphanesinde bulunan otomasyon istasyonları ve cihazları özetle şunlardır:
 - Çeşitli marka ve model endüstriyel robotlar

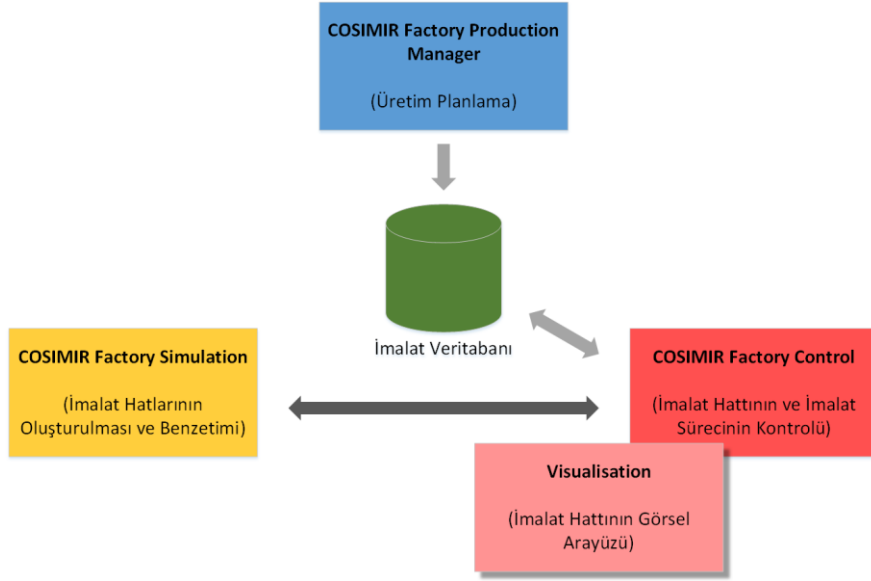
- Çeşitli marka ve model CNC cihazları
- Test ve kalite kontrol ekipmanları
- Tam otomatik depolama alanları
- COSIMIR Factory Control: Yazılım temel olarak bir imalat hattının denetimi için gerekli imalat kontrol yazılımının oluşturulmasını sağlar. Bir bakımdan kontrol yazılımı, tüm fabrikayı yöneten SCADA yazılımıdır.

Genel anlamda bir imalat hattı veya imalat hücresine ait yürütülebilen görevler şunlardır:

- İş parçaları depo alanlarından alınarak imalat istasyonlarına taşınır,
- Robotlar iş parçalarını konveyör hattındaki istasyondan alarak hücre içerisindeki önceden belirlenmiş alanlara (palet kabul alanı) taşır,
- CNC makinaları parçaları işler,
- Montaj makinaları son ürünü oluşturmak üzere parçaları bir araya getirir,
- Özel istasyonlar üretilen iş parçalarını test ve kalite kontrol işlerini yürütür,
- İmalatı bitmiş son ürünler veya ara ürünler depo alanlarında saklanır.

COSIMIR Factory Control yazılımı ile imalat süreçlerinin kontrolüne yönelik süreç planları oluşturulabilir. Stok yönetimi özelliği sayesinde imalat sürecinde kullanılan hammaddeler ile üretilen yarı-ürünler ve ürünlerin takibi yapılabilir. İmalat hattının anlık durumuyla ilgili veriler eş zamanlı olarak COSIMIR Factory Simulation yazılımı ile paylaşılır. İmalatın durumu görsel arayüz üzerinden takip edilebilir ve benzetim yapılabilir.

- COSIMIR Factory Production Manager: Bir imalat planlama sistemine ait basit görevlerin yürütülmesini sağlar. Bu görevler aşağıdaki gibi özetlenebilir:
 - İmalat veri tabanında sipariş ve montaj numaraları oluşturarak yeni iş parçalarının tanımlanması,
 - Yeni iş parçaları için imalat süreci oluşturulması ve tanımlanması,
 - Önceden tanımlı imalat veritabanını temel alarak bir görev tablosunun derlenmesi ve yürütülmesi,
 - Görev tablosunun yürütülmesi sırasında hangi adımda kalındığının gösterilmesi,
 - Stok istasyonunun yönetilmesi.

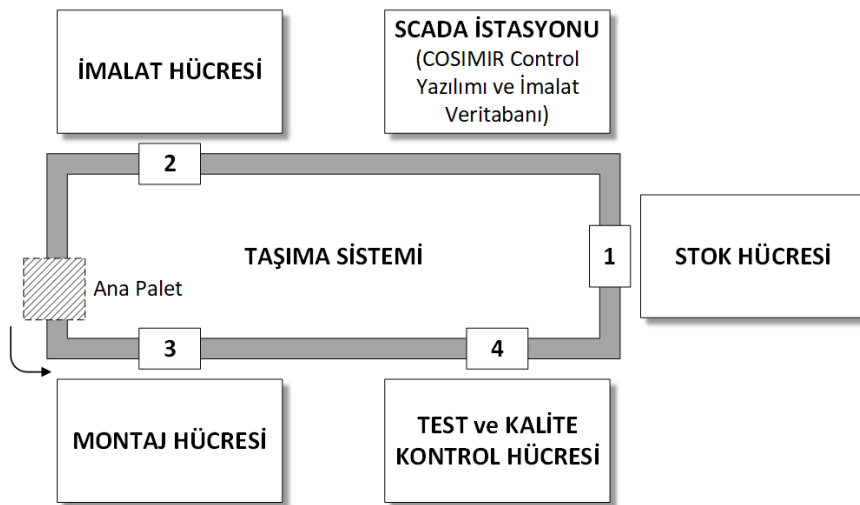


Şekil 3-84. COSIMIR Factory yazılımının alt yazılımları ve birbirleriyle ilişkileri.

3.1.8.1.1. COSIMIR Factory Yazılımında İmalat Hattı

İmalat hatları, imal edilmesi hedeflenen ürünün yapısına ve imalat süreçlerinin çeşitliliğine göre birçok bileşenden oluşabilir. Buna rağmen bir imalat hattı Şekil 3-85’da sunulduğu gibi temel olarak şu bileşenlerden oluşur:

- Taşıma sistemi (genellikle konveyör bandı)
- Otomatik stok hücresi
- İmalat ve montaj hücresi
- Görsel imalat denetim istasyonu (SCADA)
- İmalat veritabanı
- Test ve kalite kontrol hücresi



Şekil 3-85. COSIMIR Factory yazılımında oluşturulabilecek örnek bir imalat hattı bileşenleri.

3.1.8.1.2. İmalat Hattındaki Malzeme Taşıma Sistemi

Bir imalat hattındaki malzeme taşıma sistemi iki bölümde incelenir:

- İş parçalarının imalat istasyonları arasında taşınması
- İş parçalarının imalat istasyonu içinde taşınması

İş parçalarının taşınmasında genellikle konveyör bantları kullanılır. İş parçaları öncelikle paletler üzerine yerleştirilir. Sonrasında üzerleri dolu paletler konveyör bantı ile hedef istasyonlara taşınırlar. Günümüzün modern fabrikalarında çizgi izleyen veya lazer, radyo frekans, ultrasonik teknolojilerini kullanan insansız akıllı arabalar ile imalat hücreleri arası taşıma işlemleri gerçekleştirilmektedir. Her durumda iş parçaları önce palet üzerine yerleştirilir, sonrasında palet taşıma sistemi üzerine yerleştirilir ve hedefe gönderilir. Gelişen RFID ve optik etiketleme teknolojileri sayesinde iş parçalarını taşıyan her bir palet ayrı ayrı etiketlenmektedir. Böylelikle paletlerin doğru hedef istasyona ulaşmaları sağlanmaktadır. Şekil 3-86’de BMW isimli araba fabrikasında kullanılan ve 2B lazer tarama teknolojisiyle yolunu bulan, otonom araba biçimindeki akıllı taşıma sistemi görülmektedir.

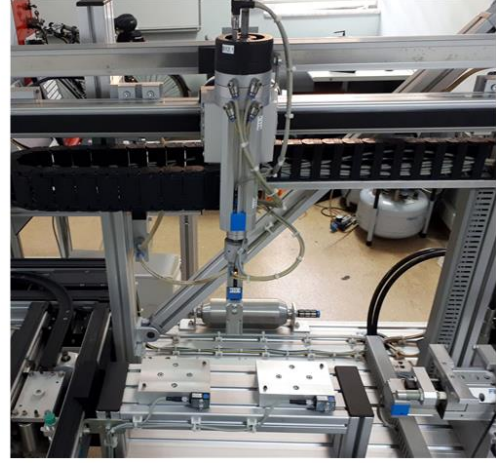


Şekil 3-86. BMW araba fabrikasında çalışan otonom taşıma sistemi [85].

Endüstriyel robotlar, genellikle imalat istasyonuna getirilen paleti ve iş parçalarını imalat hücresi içerisinde taşımak için kullanılır. İmalatı bitip işlenmiş iş parçalarını ise yeniden palet üzerine yerleştirerek paleti yeniden taşıma sistemine gönderirler. Bu tip işlerde kullanılan endüstriyel robotlar Şekil 3-87’de görüldüğü gibi 5/6 eksenli olabildiği gibi (a), sadece 2 eksenli ve 8 serbestlik dereceli Kartezyen tut/bırak (pick & place) yapıları (b) da olabilir.



(a)



(b)

Şekil 3-87. İmalat hücrelerinde iş parçaları ve paletleri taşımak için kullanılan robotik sistemler: 6 eksen endüstriyel robot (a), 2 eksenli ve 8 serbestlik dereceli Kartezyen tut/bırak sistemi (b).

FMS100 bünyesinde ana taşıma sistemi olarak dikdörtgen şekilli konveyör bant kullanılmaktadır. Bant üzerinde 6 adet istasyon bulunmaktadır. Her istasyonda Ana-Paletleri durdurmak için pnömatik silindirler ve palet kenetleme sistemi (IX) bulunmaktadır. Taşıma bantı üzerinde RFID etiketlere sahip 10 adet bağımsız Ana-Palet hareket etmektedir. Her istasyon kenetleme sisteminde, Ana-Paletlerde bulunan RFID etiketlerini okumak için özel bir algılayıcı bulunur. Böylelikle istasyonlara gelen Ana-Paletlerin etiketleri okunarak, paletlerin konveyör hattında takibi sağlanır. Ana-paletler üzerinde ise farklı tipteki iş parçalarını taşımak üzere tasarlanmış 3 farklı yardımcı palet taşınır: Palet 2ED, Palet 2PCB ve Palet 8PIN. FMS100 bünyesindeki 4 imalat hücresinde iş parçaları ve yardımcı paletleri taşımak için endüstriyel robot kullanılır. 1 imalat hücresinde 2 eksen 8 serbestlik dereceli servo pnömatik yapı kullanılırken depolama hücresinde “Teleskop” adı verilen çatallı 3 eksen Kartezyen taşıma sistemi kullanılmaktadır.

3.1.8.1.3. İmalat Hattındaki Stok Yönetimi

Bir üretim hattındaki temel birim otomatik stok hücresidir. Stok hücresi, her biri bir adet yardımcı palet saklayabilen raflardan oluşur. İhtiyaç halinde raflardaki paletler hedeflenen istasyona gönderilir. Stok hücresindeki raflarda saklanan ürünlerin bilgisi imalat veritabanında saklanır. Bu sayede İKS hangi rafta hangi iş parçasının tutulduğunu,

hangi rafın boş olduğunu bilir. Raflarda saklanan iş-parçaları değiştiğinde, değişiklik anında veritabanında da yapılır. Böylelikle stok bilgisi sürekli güncel tutulur.

Stok hücresi dışında, her imalat hücresinde değişen miktarda palet depo alanları bulunur. Böylelikle palet ve üzerindeki iş parçaları hücreye ulaştığında stok hücresinden bağımsız olarak işlem yapılabilir. Ayrıca bazı istasyonlarda iş parçaları magazinler içerisinde de saklanabilir. İhtiyaç halinde magazinlerdeki parçalar imalatı veya montajda kullanılır.

FMS100 sisteminde AS-RS (Automated Storage-Retrieve System) isimli stok hücresi kullanılır. Hücre içinde üst üste 6 sıra ve her sırada 8 adet olmak üzere toplam 48 adet numaralandırılmış raf bulunur. Her bir rafta bir adet yardımcı palet (dolu/boş) saklanır. “Teleskop” isimli çatallı 3 eksen Kartezyen sistem ile yardımcı paletlerin, raflar arası veya raf ile taşıma sistemi arası taşınması sağlanır. Numaralandırılmış raflarda hangi stok numaralı iş-parçasının tutulduğu, hangisinin boş olduğu bilgileri SQL Server tabanlı tasarlanmış imalat veritabanında saklanır.

Stok hücresindeki raflarda RFID veya optik tabanlı etiketleme sistemi bulunmamaktadır. Bu nedenle ilk başta raflarda bulunan iş-parçaları ile veritabanı kayıtları arası eşleştirme işlemi yapılması zorunludur. Önceden belirlenen bir sıralamaya göre (stok hücresi otomasyon panosu arkasındaki etikette de yazıdır) iş-parçaları raflara dizilir. Sonrasında SCADA yazılımı Stok Kontrol ekranından “First Time Init” işlemi yapılır. İmalat veritabanı önceden belirlenmiş bu dizilime göre güncellenir ve eşleşme sağlanmış olur. Bu andan itibaren raflarda yapılacak tüm değişiklikler veritabanında da gerçekleştirileceği için, güncel stok bilgisi tutulmuş olur. Her ne kadar günümüzde RFID ve optik etiketleme sistemleri gelişmiş olsa da, maliyet sorunları yüzünden, hala birçok işletmede stok takibinde eşleştirme yöntemi kullanılmaktadır.

3.1.8.1.4. Otomasyon ve İmalat Cihazları Programları

İmalat hatlarında birbirlerinden farklı imalat hücrelerinden oluşur. Her hücre içerisinde de belirli görevleri yürütmek üzere programlanmış PLC, IPC, endüstriyel robot ve CNC gibi otomasyon ve imalat cihazları bulunur. Her bir cihaz, imalatı yürüteceği tüm görevleri içerecek şekilde bağımsız olarak programlanır. Sonrasında seri/paralel bağlantılar ve haberleşme protokolleri kullanılarak merkezi kontrol yazılımına bağlanırlar.

FMS100 sisteminde Lehimleme ve Hidrolik-Baskı hücreleri S7-300 PLC ve Stok, Montaj ve Görüntü İşleme hücreleri IPC tarafından denetlenir. CNC-Çifti istasyonu ise tümüyle robot kontrolör tarafından yönetilir. Lehimleme, Montaj, CNC-Çifti ve Görüntü İşleme hücrelerinde de endüstriyel robot bulunmaktadır. CNC-Çifti hücresinde 1 er adet CNC torna ve freze cihazı bulunmaktadır.

EİS yaklaşımına göre tüm bu cihazlar, hücrenin yürüteceği görevlere göre, birbirleriyle uyumlu çalışacak şekilde ayrı ayrı programlanır. Örneğin PLC programlamak için “Simatic Manager” yazılımı kullanılırken endüstriyel robotlar “COSIMIR Industrial” yazılımı ile programlanır. Her bir cihazın yazılımı kendi içerisine yüklenir. Fakat hedef ürünü imal edebilmek için gerekli süreç planları ana kontrol bilgisayarındaki SCADA yazılımı tarafından kontrol edilir.

3.1.8.1.5. İmalat Veri Tabanı

Bilgisayar tümleşik imalat sürecinde tüm veriyi imalat veritabanı yönetir. İmalat süreci devam ederken veritabanı tüm imalat cihazları ve personelin kullanımına açıktır. Veritabanında şu konular ile alakalı veriler saklanır:

- Stok takibi
- Son ürünler, ara-ürünler ve boş ürünlere ait parça numaraları
- İmalat süreci kaynakları (taşıma sistemi cihazları, tezgah takımları, cihazlar, vb.)
- İmal edilmiş ürünlerin kalitesi
- Müşteriler ve sipariş bilgileri

Veritabanında Tablo 3-4’de görüldüğü gibi yapılandırılmış bir şekilde 3 tablo bulunur.

Tablo 3-4. İmalat Veritabanı içindeki tablo ve alanlar.

	İş-Parçaları Tablosu (Workpieces)	İmalat Süreci Tablosu (Production Process)	Müşteri Tablosu (Customer)
Alan 1	parça numarası (part no)	parça numarası (part no)	müşteri numarası (customer no)
Alan 2	parça ismi (designation)	cihaz (machine)	adres (address)
Alan 3	tip (type)	cihaz program numarası (machine program number)	sipariş numarası (order number)
Alan 4	Tasarım (design)	imalat sürecinin türü (type of production process)	*

İmalat veritabanında iş-parçaları parça numaraları ile tanımlanır. Bu nedenle son ürünler, ara-ürünler ve boş ürünlerin tanımlanabilmesi için belirli numara aralıkları ayrılmıştır. Ara-ürünler ile son ürünlerin parça numaraları önceden tanımlanmış olup boş ürün numaraları FMS100 bünyesinde üretilecek yeni ürünlerin tanımlanmasında kullanılabilir.

İmalat veritabanında bir iş-parçası şu verilere göre sınıflandırılabilir:

- Parça numarası
- Ürün ismi
- İş-parçası tipi: EİS'te mi üretilecek yoksa dışarıdan mı tedarik edilecek
- İş-parçasının fiziki yapısı
- Palet
- Maliyet
- Satış fiyatı

Bir iş parçasına ait imalat sürecinin tanımlanabilmesi için şu verilere ihtiyaç vardır:

- İş parçasının imalatında kullanılacak cihaz
- Cihazda yürütülecek görev
- Cihazda kullanılacak program
- İş-parçasının işlenerek mi yoksa montaj ile mi elde edileceği bilgisi
- İş-parçasının test ve kalite kontrolünün yapılıp yapılmayacağı bilgisi
- Hatalı imal edilmiş iş-parçalarına ne yapılacağı

3.1.8.2. COSIMIR Factory Yazılımında İmalat Kontrol Sistemi (İKS)

COSIMIR Factory yazılımında İKS hiyerarşik ve kolay uyarlanabilir bir yapıya sahiptir. Kontrol altyapısı kurulumuna geçmeden önce EİS bünyesinde üretilmek istenen ürün ve imalat süreciyle ilgili bazı temel sorulara cevap verilmesi gereklidir. Bu sorular ve verilen cevaplar, imalat planlamanın ilk aşamasını oluşturur. Örnek olarak FMS100 bünyesinde, üzerinde termometre bulunan alüminyum kalemlik üretilmek istensin. Bu örnek ürünün imalat sürecine yönelik temel sorular ve olası cevapları Tablo 3-5'de verilmiştir.

Tablo 3-5. İmalat planlama aşamasında sorulması gereken sorular ve örnek cevaplar.

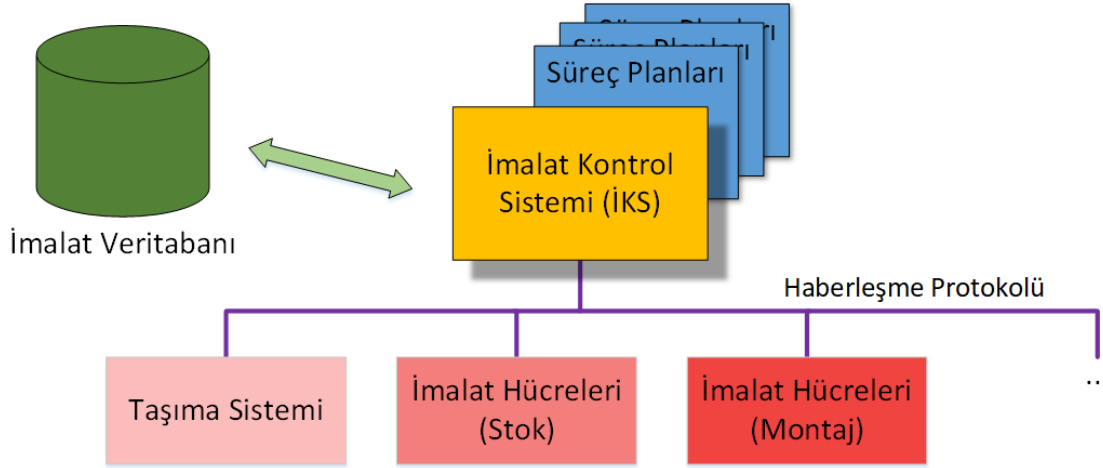
	Sorular	Cevaplar
1	İmal edilmesi hedeflenen ürün nedir?	> Üzerinde termometre bulunan alüminyum kalemlik.
2	İmal edilecek hedef ürününün parça numarası nedir?	Ürünü oluşturan tüm iş parçalarının kendine özgü parça numarası olması gereklidir. İmalat veritabanında parça numaraları kullanılarak iş parçalarının tanımlaması yapılır.

		> İmalat veritabanındaki kayıtlara bakıldığında hedef ürünün parça numarası 52368'dir.
3	İmalat hattına girilecek üretim emri nedir?	Ürünü oluşturmak için imalat hattına gönderilecek iş emri imalat tablosuna girilmelidir. >Parça numarası 52368 üret.
4	Hedef ürünü imal edebilmek için hangi iş parçaları gereklidir? Gerekli iş parçalarının parça numaraları nedir? Nereden temin edilmektedir?	İmalat veritabanında, her ürünün imalatı için gerekli iş parçalarının neler olduğu, nereden temin edileceği gibi bilgiler saklanmaktadır. > Stok istasyonunda bulunan alüminyum tabanlık (parça no: 42140) ve kalemlik (parça no: 42102). Ayrıca montaj istasyonundaki magazinde hazır bulunan termometre (parça no: 30000) ve kalem (parça no: 30100)...
5	Hedef ürünü elde etme sürecinde imal edilen yan ürünlerin (hatalı/eksik) parça numaraları nedir?	Hedef ürünü imal ederken test edilmemiş, testi geçememiş ürünler olabilir. Bu ürünlerin eksiklik ve hataları ayrı ayrı tanımlanarak ayrı parça numaraları imalat veritabanında belirlenmelidir. > Delikleri yanlış açılmış alüminyum tabanlık (parça no: 42501) > Montajı bitmiş fakat test edilmemiş ürün (parça no: 42508) > Montajı bitmiş, test edilmiş, hatalı ürün (parça no: 42510) > Montajı bitmiş fakat termometresi bulunmayan ürün (parça no: 42502)...
6	Hedef ürünü imal etmek için hangi kaynaklara (cihaz, istasyon, hücre vb.) ihtiyaç vardır?	İmalat veritabanında her ürünün imalatı için gerekli kaynakların neler olduğu saklanmaktadır. > Stok istasyonu, konveyör bantı, CNC torna ve freze, endüstriyel robot.
7	İmalat hattı ihtiyaç duyulan kaynaklara sahip mi?	Proje dosyası incelenerek ihtiyaç duyulan kaynaklara imalat hattı sahip mi? > Proje dosyasında ihtiyaç duyulan kaynakların yer aldığı tespit edilmiştir.
8	Hedef ürünün imalatı için hangi adımlar yürütülmelidir?	İmalat sürecinde yürütülecek adımların sıralaması SDG kullanılarak şematize edilebilir. Sonrasında COSIMIR Factory Control yazılımında oluşturulan "süreç planları" ile sürecin programı oluşturulur. > Taşıma sistemi komutu: Montaj istasyonuna boş bir palet gönder. > Stok istasyonu komutu: CNC istasyonuna işlenmemiş alüminyum blok gönder. > Taşıma sistemi komutu: İşlenmemiş kapakları stok istasyonundan alarak CNC istasyonuna gönder...
9	İmalat sürecini sonlandırma kriteri nedir?	Ürünün tamamlanmış olduğunu ifade eden kriterler belirlenir.

		> Termometrenin çalışıyor olması > Kalemlik üzerindeki işlemlerin doğru yapılmış olması. > Alüminyum tabanlık üzerindeki deliklerin doğru açılmış olması...
--	--	---

Hedef ürünün imalatına yönelik süreç yorumlandığında Şekil 3-88’de görülen yapı elde edilir ve aşağıdaki çıkarımlara varılabilir:

- İmalat hattı ve İKS’nin hiyerarşik bir yapısı vardır.
- İmalat hattındaki her bir cihaza erişebilmek için İKS’nin, sistemin genel yapısı ile sistemde bulunan cihazlar hakkında bilgi sahibi olması gereklidir.
- İKS’nde başlatılacak süreçler kapsamında yürütülecek görevlerin sıralaması süreç planı isimli programlama dili ile oluşturulur.
- Tam otomatik bir fabrika otomasyonu için, iş parçaları ve imalat süreçleriyle ilgili sistemden toplanan tüm veriler bir veritabanında saklanır. Böylelikle İKS gerektiğinde veritabanında kayıtlı bilgilere erişerek işlem yapabilir.



Şekil 3-88. İmalat kontrol sistemi bileşenleri.

EİS’in de temelini oluşturan imalat hatlarının hiyerarşik kontrol sistemi daha detaylı incelendiğinde Şekil 3-89’daki yapı ortaya çıkmaktadır.

Hiyerarşinin en tepesinde SCADA bilgisayarında çalışmakta olan İKS vardır ve tüm yapıyı yönetir. Kontrol sistemi bünyesinde önceden tanımlanmış süreç planları bulunur. Bir açıdan COSIMIR Factory Control yazılımındaki imalat programları süreç planlarıdır. Süreç planları da kendi içerisinde üç katmanda incelenebilir.

Hedef ürünün eldesi için yapılması gereken hesaplamalar, yürütülmesi gereken fonksiyonlar ve görevler ile tüm bunların yürütülme sıralamaları (algoritma), imalat süreç

planları içerisinde tanımlanır. Her bir imalat süreç planına özel bir isim verilir. Örneğin FMS100 bünyesinde “Process 403” metal pinleri tornalama işlemini yürütür. Süreç çalıştırıldığında özetle; stok hücrelerinden ham metal pinleri taşıyan Palet 8PIN çağrılır, palet robot aracılığıyla hücre içine taşınır, ham metal pinler robot ile tek tek CNC torna tezgahına bağlanır, işlenmiş metal pinler tekrar palete yüklenir, palet stok hücrelerine geri gönderilir ve saklanır. Örnekte de görüldüğü gibi bir imalat süreç planı kapsamında belirli bir sıralama içerisinde birçok farklı imalat hücresi ve cihaz, birçok farklı görevle yer almaktadır.

Görev süreç planları içerisinde imalat hücrelerinde yürütülebilecek tüm görevler tek tek tanımlanır. Her bir imalat hücresinde genellikle farklı cihazlar bulunur ve farklı görevler yürütülür. Bu nedenle görev süreç planları imalat hücrelerine göre farklılık gösterir. Örneğin FMS100 bünyesindeki CNC-Çifti hücresi için oluşturulmuş olan “Kapakİşle()” isimli görevi, Montaj hücresi yürütemez. Çünkü Montaj hücresinde kapakları işleyecek gerekli donanım yoktur. Bünyesinde endüstriyel robot bulunan tüm hücrelerde “PaletTaşı()” isimli görev bulunur. Fakat içeriği ve robotlarda yüklü bulunan programlar farklı olabilir. Bu nedenle isim aynı olsa da her bir imalat hücresi için görev tanımlamaları ayrı yapılır.

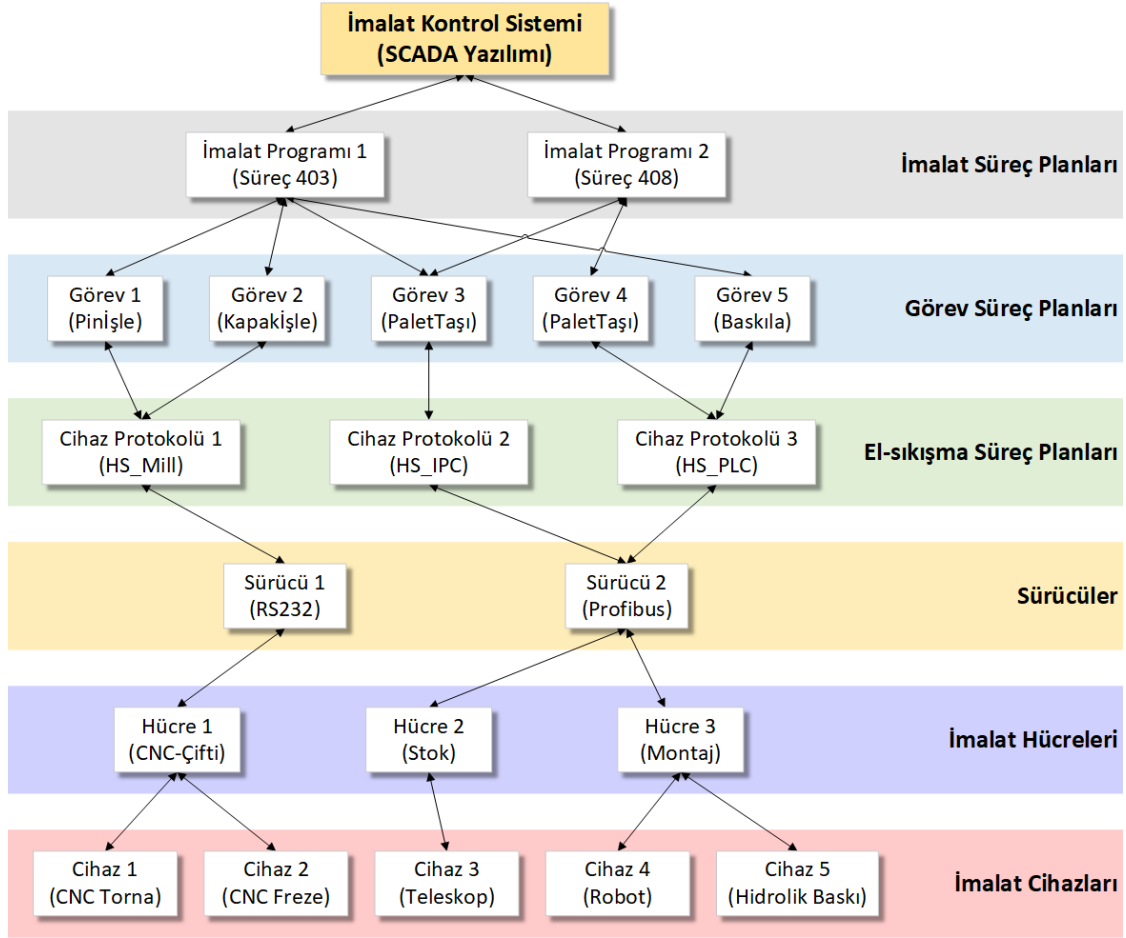
Her cihazın üreticisi tarafından belirlenmiş kendine özgü bir haberleşme protokolü vardır. Bu protokoller kullanılarak cihazın özelliklerine ve yürütebileceği görevlere uzaktan erişilebilir, komut verilebilir. COSIMIR Factory Control yazılımı bünyesinde cihaz haberleşme protokolleri el-sıkışma (handshake) süreç planları içerisinde yazılır. El-sıkışma süreç planları isimlerinden ayırt edilebilsin diye “HS_” ön ekiyle başlar. Örneğin “HS_CR1” isimli süreç planı CR1 model robot kontrolörün uzaktan kontrol edilebilmesi için oluşturulmuştur.

COSIMIR Factory Control yazılımı farklı tipte endüstriyel haberleşme teknolojilerini destekler. Bunlar arasında seri haberleşme (RS232), Profibus türevleri, AS-i Bus, TCP/IP yer alır. Bu teknolojilerin hepsi seri haberleşme için geliştirilmiş olsa da; iletişim kablosu yapıları, kablo üzerindeki elektriksel gerilim seviyeleri, kablounun karşı tarafa veya taraflara bağlanış şekli (topoloji) vb. fiziksel özellikleri değişiklik gösterir. Bu da doğal olarak her teknolojinin erişebileceği en uzak mesafe, çıkabileceği en yüksek veri iletim hızı, elektriksel gürültü dayanımı ve bağlantı topolojisi gibi özelliklerinde farklar yaratır.

Örneğin RS232 en az standart 3 damarlı bir kablo ile noktadan noktaya 15 m'ye kadar ve 115.200 Bps hızlarda güvenli iletişim sağlarken; Profibus DP 2 damarlı ekranlı kablo ile 1.200 m'ye kadar ve 12 MBps (12.000.000 Bps) hızlara çıkabilir. Sonuç olarak tüm bu iletişim teknolojilerinin yazılımlar tarafından kullanılmasını sağlayan ise sürücülerdir. Sürücüler haberleşme teknolojisi ile el-sıkışma süreç planları arasında köprü görevi görür. Bir sürücü temelde, veri okumak ve veri yazmak gibi iki farklı görev yürütür. Kendisine el-sıkışma süreç planından gelen verileri karşı tarafa yazar ve karşı taraftan gelen verileri okuyarak bir üst katmandaki el-sıkışma süreç planına iletir.

İmalat hücrelerinde farklı görevleri yürütmek üzere farklı cihazlar bulunur. Genellikle her hücre PLC veya IPC gibi programlanabilir bir denetleyici tarafından yönetilir. Hücre içerisindeki tüm buton/anahtar, algılayıcı, eyleyici ve imalat cihazları hücre denetleyicisiyle ilişkilidir. EİS yaklaşımında hücre denetleyicilerin yetkileri sınırlıdır. Çoğunlukla hücre içerisinde topladıkları verileri İKS'ne iletmek, İKS'nden gelen komutları ise ilgili imalat cihazlarına iletmek veya eyleyicileri sürmek gibi görevleri vardır. Bir anlamda şeffaf bir arayüz görevi görürler. EİS yaklaşımında imalat ile ilgili süreç programları bu denetleyicilerde yer almaz. FMS100 bünyesinde hücre denetleyicisi olarak 2 PLC (Lehimleme, Hidrolik-Baskı), 3 IPC (Stok, Montaj, Görüntü İşleme) kullanılmıştır. CNC-Çifti hücresinde ise denetim CR1 robot kontrolöre bırakılmıştır.

İmalat cihazları da imalat hücreleri bünyesinde bulunan ve sistemde asıl işi yapan elemanlardır. İmalat süreç planlarında tanımlanan görevleri tanımlanan sırada gerçekleştirirler. Genellikle imalat hücre denetleyicisine bağlı olarak çalışırlar. Bazı durumlarda direk İKS'ne de bağlanabilirler. Örneğin FMS100 bünyesindeki CNC cihazları RS232 üzerinden direk SCADA bilgisayarına bağlıdır.



Şekil 3-89. COSIMIR Factory Control yazılımındaki kontrol yapısı.

Buna göre örneğin kullanıcı tarafından “Süreç 403” isimli imalat programı çalıştırıldığında, ilgili süreç planındaki sıralamaya göre “PaletTaşı(8,1)” isimli görev çalıştırılır. Bu görev içerisinde oluşturulan komutlar “HS_IPC” el-sıkışma süreç planı ve “Profibus” sürücüsü üzerinden Montaj hücresi kontrolörü olan IPC’ye iletilir. IPC içerisinde yüklü olan program endüstriyel robotta yüklü “MovePalet(8,1)” isimli programı çalıştırır. Robot program gereği paleti taşıma sistemi üzerinden alarak hücre içerisindeki palet kabul alanına taşır. Robotun bu görevi yürütebilmesi için öncelikle istasyonda hedef paletin bulunması ve hücre içindeki palet kabul alanının boş olması gerekir. Görev başarıyla tamamlansa da, hata nedeniyle tamamlanamasa da, sonucu IPC’ye iletilir. IPC robotun yürüttüğü görev sonucunu haberleşme teknolojisini kullanarak “Profibus” sürücüsüne iletir. Sürücü ise okuduğu bilgi mesajını “HS_IPC” üzerinden “PaletTaşı” görev süreç planına iletir. Böylelikle “PaletTaşı” görevinin sonucu en üst katmana kadar çıkarak “Süreç 403” isimli imalat süreç planının ilgili kod satırına

ulaşır. Bu satırda yazan koda göre sonuç değerlendirilir ve ilgili satıra dallanma gerçekleşir.

Sonuç olarak hiyerarşik yapının kontrol sistemine kattığı esneklik ve faydalar şu şekilde özetlenebilir:

- Tüm imalat hattının yönetimi ve izlenmesi işleri tek bir merkezden (İKS) yürütülür.
- İmalat sistemindeki her hiyerarşik katmanın alt ve üst katmanlarla ilişki kurabilmek için tanımlı görevleri ve arayüzleri vardır.
- İmalat sistemine yeni bir cihaz kolaylıkla eklenebilir. İKS’nde sadece yeni bir sürücü ve bununla ilişkili el-sıkışma süreç planı oluşturulması yeterlidir. İmalat süreçleri gibi diğer bileşenler bu değişiklikten etkilenmez.
- PLC ve endüstriyel robot gibi cihazlarda yüklü programlarda yapılan değişiklikler İKS’ni etkilemez.
- İmalat süreç planlarında değişiklik yapmak oldukça kolaydır.
- İmalat hattına kolaylıkla yeni bir imalat hücresi eklenerek genişletilebilir. Bunun için İKS’ne uygun sürücüler ve el-sıkışma süreç planları eklemek yeterlidir.
- İmalat hattından elde edilen tüm bilgiler bir merkezde toplanarak veritabanına kaydedilebilir. Toplanan veriler işletme kaynak planlaması, stratejik planlama ve tedarik zinciri yönetimi uygulamalarında kullanılabilir.

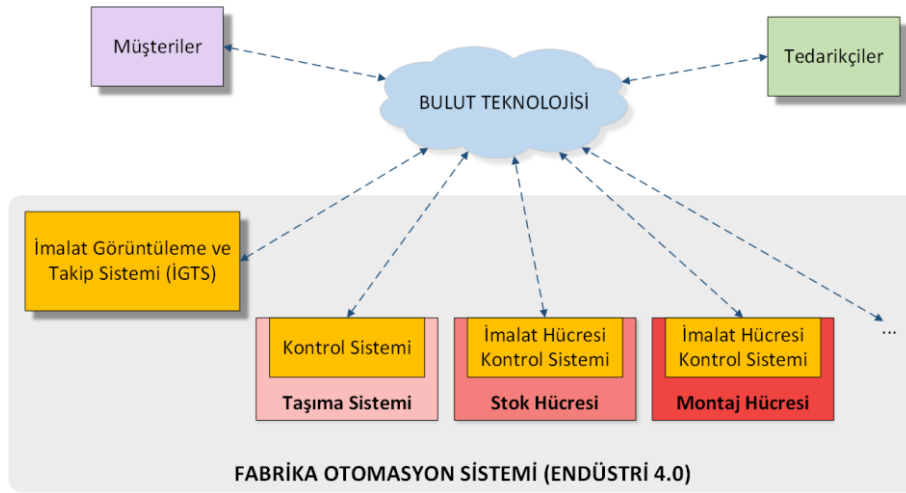
3.1.8.3. EİS ile Endüstri 4.0 Yaklaşımı Karşılaştırması

Günümüzde Endüstri 4.0 yaklaşımı bir devrim olarak kabul edilip oldukça popüler bir konudur. Bu devrimi besleyen teknolojilerin 3B yazıcı, optik ve manyetik etiketleme (RFID) ile nesnelerin Internet’i ve bulut teknolojisi olduğu söylenmektedir. Endüstri 4.0, mevcut EİS yaklaşımının farklı bir şekilde yeniden yorumlanmış halidir denebilir. Endüstri 4.0 hakkında birçok makale bulunmakla birlikte aralarındaki farkların ne olduğunu anlamak oldukça güçtür. Bu çalışmadaki EİS ve özellikle Bölüm 3.1.8.2’de açıklanan İKS incelendikten sonra Endüstri 4.0 ile getirilmek istenen yenilik daha kolay anlaşılacaktır.

EİS yaklaşımında merkezi bir İKS vardır. Bu sistem temelde bir bilgisayar, SCADA yazılımı ve veritabanından oluşur. Hedef ürünün eldesine yönelik tüm imalat süreç planları bu merkezi yapı içerisinde tanımlanır. Bu merkezi yapının tüm imalat hücreleri ve hücreler içerisindeki cihazların yürütebileceği görevleri bilmesi; cihazların her biri ile

kesintisiz iletişim kurabilmesi, bunun için el-sıkışma süreç planları ve sürücülere sahip olması gereklidir.

Endüstri 4.0 yaklaşımında ise, merkezi bir İKS yoktur. Bunun yerine sadece gelen talebe uygun üretim emrini ilgili hücreye gönderip, imalat sürecini başlatan ve hücrelerden gelen verileri merkezi olarak görüntüleyen bir imalat görüntüleme ve takip sistemi (İGTS) vardır. İmalat kontrolünde herhangi bir hiyerarşi yoktur. Şekil 3-90'de Endüstri 4.0 yaklaşımında temsili kontrol yapısı görülmektedir.



Şekil 3-90. Endüstri 4.0 yaklaşımında kontrol yapısı.

Bu yapıda her imalat hücresinin kendi kontrol sistemi vardır. Bu kontrol sistemi sadece kendi bünyesinde bulunan cihazları bilir ve haberleşir. Hücre kendi imalat sürecini kendisi denetler ve yürütür. Bir anlamda kendi imalat süreç planlarını kendi hücre denetleyicisinde yürütür. İmalat süreci sonunda elde edilen iş-parçasını taşıma sistemine geri gönderir. İmalat sürecinin durumu ve sonucu hakkındaki bilgileri buluta gönderir. Diğer imalat hücreleri de bulut üzerinden bu bilgilere erişerek, sıradaki imalat hücresi hangisiyse taşıma sistemindeki iş-parçasını alarak kendi görevini yürütür.

İGTS de bulut teknolojisi üzerinden tüm imalat hücrelerinin durumunu görüntüler. Yönetici tarafından verilen komutları ve iş emirlerini de bulut sistemine yazar. İmalat hücreleri de komut ve iş emirlerini buluttan okur. Böylelikle tüm sistem, bulut teknolojisi üzerinden eşit seviyede iletişim kurar, hiyerarşi yoktur. Bu yaklaşımın bir avantajı da, hedef ürün için siparişi veren müşteri de siparişini ne zaman teslim alabileceği, siparişi hangi aşamada; durumunu bulut üzerinden canlı olarak takip edebilir. Benzer şekilde tedarikçiler de imalat sistemi için gerekli hammaddelerin durumunu görüntüleyebilir ve

ona göre kendi tedarik süreçlerini planlayabilir. Bu da stok tutma zorunluluğunu en alt düzeye indirerek büyük maliyet avantajları sağlar.

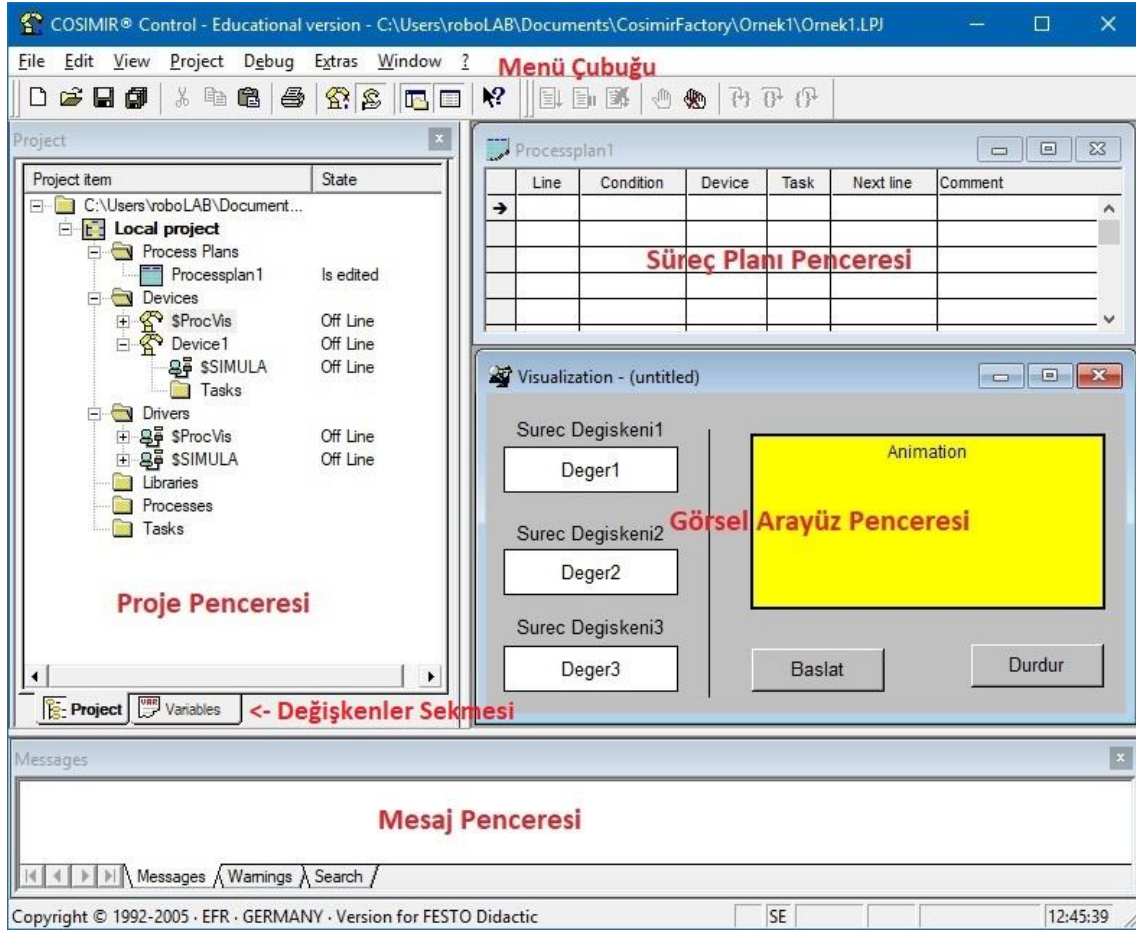
3.1.8.4. COSIMIR Factory Control Yazılımı ve Kullanımı

COSIMIR Factory Control imalat hücrelerini/istasyonlarını kontrol etmek için kullanılan bir bilgisayar yazılımıdır. Farklı üreticilere ait imalat cihazlarını iletişim arayüzleri ve sürücüler kullanarak bir araya getirerek bir EİS e dönüştürmesini sağlar. Sahip olduğu kontrol mimarisi sayesinde farklı ürünlerin, farklı miktarlarda eş zamanlı olarak üretilmesine izin verir. İmalat süreci için gerekli değişikliklere kolayca adapte olabilmeyi sağlayan ayar menülerine sahiptir.

3.1.8.4.1. Kullanıcı Arayüzü

COSIMIR Factory Control yazılımı çalıştırıldığında Şekil 3-91'deki geliştirme ekranı (Integrated Development Environment - IDE) görüntülenir. Ekranda temel olarak görüntülenen pencereler ve işlevleri aşağıdaki gibidir:

- **Proje Penceresi (Project Window):** COSIMIR Factory Control yazılımının ana işlem arayüzüdür. Tüm kontrol sisteminin projelendirilmesi ve ayarlarının yapılabilmesine olanak verir. Kontrol sistemini oluşturan alt bileşenler RESİM de görüldüğü gibi ağaç menüsünde görüntülenir.
- **Süreç Planı Penceresi (Process Plan Window):** COSIMIR Factory Control yazılımının kontrol programları süreç planlarıdır. Bu pencere üzerinden herhangi bir süreç planı açılabilir ve düzenlenebilir. COSIMIR Factory Control yazılımı üretim (production) modunda çalışırken, o an için yürütülen süreçleri ve süreç adımlarını gösterir. Böylelikle istenilen program satırına müdahale edilerek kod bazında takip ve düzenleme yapılabilir.
- **Görsel Arayüz Penceresi (Visualization Window):** Uygulamalara özel görsel arayüz (SCADA ekranı) oluşturmak için kullanılır. Arayüzde uygulama içerisinde kullanılan birimler temsili olarak görüntülenir. Global değişkenler ve süreç planları içerisindeki değişiklikler arayüzdeki elemanların da aktif olarak değişmesini sağlar.
- **Değişkenler Penceresi (Variable Window):** Süreç planları ve sürücüler bünyesinde kullanılan global değişkenleri görüntüler. Proje Penceresi'nin altındaki "Variables" sekmesi tıklandığında görüntülenir. Pencerede ağaç görünümünde listelenen değişkenlerin üzerine tıklanarak içeriklerindeki değerler değiştirilebilir.



Şekil 3-91. COSIMIR Factory Control yazılımı geliştirme ortamı (IDE) ve bileşenleri.

3.1.8.4.2. Yeni Bir İmalat Projesi Oluşturma

Yeni bir proje oluşturma veya mevcut projede değişiklik yapabilmek için COSIMIR Factory Control yazılımının ayar (setup) moduna getirilmesi gereklidir. Bu işlemin nasıl yapılacağı ile ilgili detaylar Bölüm 2.1.2.12.1’de verilmiştir. Ayar modunda iken imalat denetleyicisi pasif durumdadır ve hiçbir imalat işlemi gerçekleştirilmez.

Yeni projenin oluşturulabilmesi için öncelikle projede yürütülmek istenen imalat işlerine yönelik görev sıralamasının, görevlerin işlem basamaklarının, görevlerde kullanılacak imalat hücreleri/istasyonları ve cihazlarının belirlenmesi gereklidir. Sonrasında oluşturulan yeni projede sırasıyla sürücüler, cihazlar, süreç planları ve değişkenlerin tanımlanması gereklidir.

Bu konuya örnek bir uygulama olarak FMS100 bünyesindeki CNC freze makinasına (Mill55) robot ile parça yüklenilmek istensin. Başlangıçta tüm iş parçaları (2 adet alt kapak) CNC-Çifti istasyonunda kenetli bulunan Ana-Palet üzerindeki Palet-2ED

üzerinde olsun. İlk önce robot paleti CNC-Çifti hücresinde bulunan1 numaralı palet alanına taşısın. Sonra Palet-2ED üzerindeki ilk alt kapak alınarak freze makinasına (Mill55) yerleştirilsin. Parçalar frezede işlendikten sonra robot işlenmiş parçayı frezeden alarak tekrar paletteki yerine yerleştirsın. Tüm kapaklar işlendikten sonra işlenmiş alt kapakları taşıyan Palet-2ED CNC-Çifti istasyonuna geri taşısın. Böylelikle denetleyici ayarlarının bir imalat hücresine nasıl uyarlanacağı ve imalat sürecinin süreç planları kullanılarak nasıl kontrol edilebileceği anlaşılmış olur.

Örnek uygulamaya ait işlem basamakları şu şekildedir:

- COSIMIR Factory Control yazılımı çalıştırılır.
- “File/New” seçilerek yeni bir proje oluşturulur.
- “Project/Setup” seçilerek ayar moduna geçilir.
- Proje içerisinde kullanılacak otomasyon cihazları arası iletişimi sağlayacak sürücüler yüklenir. FMS100 bünyesinde RS232, Ethernet, Profibus FMS, Profibus DP ve AS-i Bus gibi endüstriyel haberleşme protokolleri desteklenmektedir. Ayrıca dosyaya kayıt ve veri tabanına kayıt işlemleri de yine sürücüler üzerinden gerçekleştirilir. FMS100 bünyesindeki otomasyon cihazlarının görevlerini temsil eden benzetme işlemleri için “SIMULA” sürücüsü kullanılır. FMS100 bünyesinde kullanılan tüm haberleşme protokollerine ait sürücüler “.../DRIVERS” klasörü içerisinde bulunmaktadır.

Projede kullanılacak cihazlar için Proje Penceresi üzerinden “Load Driver” tıklanır. Gelen pencere üzerinden “.../DRIVERS/SIMULA/SIMULA.LDV” sürücüsü seçilerek “Add” butonuna basılır. Proje Penceresi üzerindeki “Drivers” seçeneğinin altında “\$SIMULA” sürücüsü görüntülenir. Bu örnekte “SIMULA” sürücüsü, eklenecek robot ve freze cihazlarının işlevlerini benzetmek (temsil etmek) için kullanılacaktır.

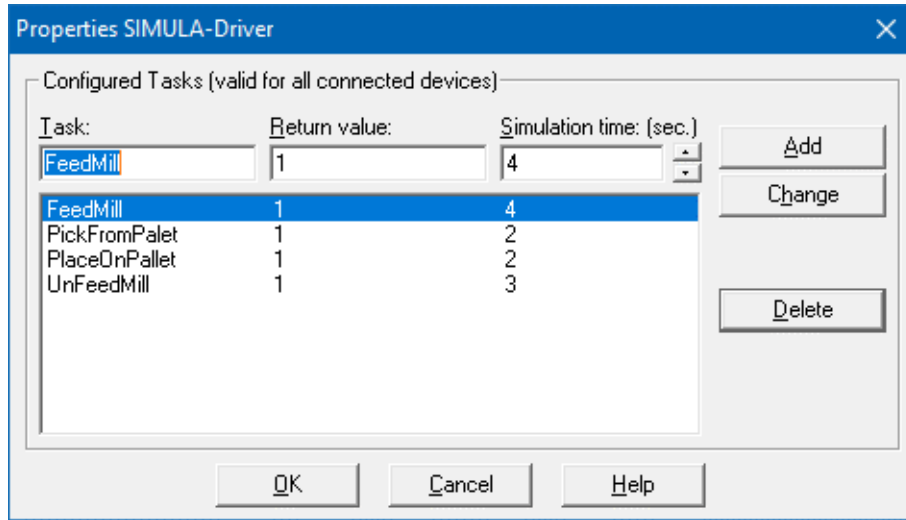
- Proje içerisinde kullanılacak otomasyon cihazları seçilir. İmalat hücresindeki her cihaz tek tek tanımlanmalıdır. “File/New/Device” seçilerek yeni cihaz projeye eklenir. Proje Penceresi üzerinden cihaza istenilen isim verilebilir. Bu örnek uygulamada projeye eklenen ilk cihaza “F2” tuşuna basılarak “Robot” adı verilir.
- Projeye eklenen cihazın haberleşmesinde kullanılacak sürücü tanımlanır. Bunun için “Robot” cihazının alt menüsündeki “Driver” simgesinin en sağındaki boş alana tıklanır ve “F2” tuşuna basılır. Ekrana gelen açılır liste penceresinden “SIMULA” seçilir.
- Örnek uygulamada robot dört farklı görev yürütmelidir: iş parçalarının palet üzerinden alınması, alınan iş parçasının freze cihazına yerleştirilmesi, işlenmiş iş

parçasının freze cihazından alınması ve palet üzerindeki eski yerine konulması.

Gerçek bir robot bu görevleri yürütmek için dört farklı program kullanabilir.

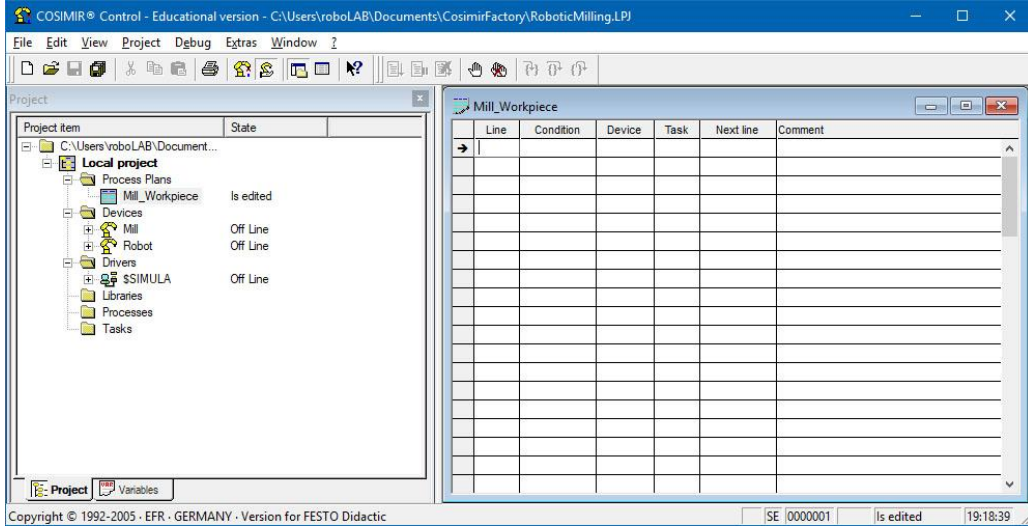
Uygulamadaki görevi yerine getirebilmek için İKS'inin farklı robot programlarını doğru sırada başlatması gereklidir. "SIMULA" sürücüsü bu durumu benzetmede kullanılır. Önceden ayarlanmış olan süre kadar bekler. Bu süre gerçek robot programının çalışma süresine eşit olmalıdır. Bekleme süresinin sonunda denetleyiciye görevin başarıyla tamamlandığı geri bildiriminde bulunur. Süreç planları gelen bu bilgiyi anında işler.

- Robotun benzetim görevlerini tanımlamak için "Robot" cihazının altındaki "SIMULA" çift tıklanır. Ekrana gelen Şekil 3-92'deki pencereden sürücünün benzetmesi istenilen görevler eklenir (add), değiştirilir (change) veya silinir. Eklenen her görev için geri bildirim değeri (return value) ve görevin benzetilme süreleri (simulation time) parametreleri tanımlanır.



Şekil 3-92. SIMULA sürücüsü ayar penceresi.

- Örnek uygulamada kullanılacak ikinci cihaz oluşturulur ve adı "Mill" olarak değiştirilir. Bu cihaza da "SIMULA" sürücüsü eklenir ve sürücü üzerine görevler eklenerek parametreleri tanımlanır.
- Projede kullanılacak sürücüler ve cihazlar tanımlandıktan sonra sıra süreç planlarını oluşturmaya gelir. COSIMIR Factory Control yazılımının kontrol programları süreç planlarıdır. Yeni bir süreç planı oluşturmak için "File/New/Process Plan" seçilir. Proje Penceresi'nde "Process Plans" bölümünde "Processplan1" şeklinde görüntülenir. "F2" tuşuna basılarak plan ismi "Mill_Workpiece" olarak değiştirilir. Süreç planının üzerine çift tıklandığında ekrana Şekil 3-93'daki gibi yeni ve boş bir Süreç Planı Penceresi gelir.



Şekil 3-93. COSIMIR Factory Control yazılımı boş süreç planı penceresi görünümü.

Oluşturulan bu süreç planı iş parçalarını frezeleme işlemlerini kontrol edecektir.

Süreç planındaki satırlar cihazlara görev atamada kullanılır. Her görevin de kendine özgü tanımlanabilir parametreleri vardır. Süreç planı, örnek uygulamadaki imalat sırasında “Robot” ve “Mill” cihazlarının yürütülecekleri görevlerin sıralamasını kontrol eder.

Süreç planı dosyası satır ve sütunlardan oluşan bir tablo yapısındadır. Sütunlar belirli tanımlamaların yapılması için farklı işlevlere sahiptir.

“Line” sütununa, ilgili olduğu satırı tanımlamak ve diğer satırlardan ayırt edebilmek için kullanıcı tarafından belirlenen bir etiket girilir. Etiket satırı tanımlayıcı bir metin veya numara olabilir. “Line” sütununa “;” karakteri le başlanırsa, ilgili satır açıklama satırına dönüşür. Normal şartlarda bu sütun boş bırakılamaz.

“Next Line” sütununa ise ilgili satır yürütüldükten sonra dallanmak istenen satır etiketi girilir. Eğer boş bırakılırsa, ilgili satırdaki görev yürütüldükten sonra bir sonraki satıra geçilir.

“Device” sütununa projede tanımlanmış cihazların isimleri veya sistem fonksiyonları girilebilir. Örneğin “.CALC” matematiksel hesaplamalar, değişken atama işlemleri vb. için kullanılır.

Bir satırdaki görev ile koşul ilişkisinde bulunan diğer satırlar “Line” hücresi boş bırakılarak tanımlanır. COSIMIR Factory Control yazılımı birbirleri ile koşul ilişkisinde bulunan satırların “Line” hücrelerini Şekil 3-94’de kırmızı daire içine alındığı gibi birleşik ve tek bir etiket biçimde gösterilir.

Line	Condition	Device	Task	Next line	Comment
; Send command					
SEND1		ABB	Write(%command)		
; Send command copy					
110		ABB	Read()		
ACK		.CALC	\$RES==CHR(6)		
130	1	ABB	DiscardReadBuffer()		
		ABB	DiscardReadBuffer()	SEND	
SEND2		ABB	Write(%command)		
; Retry if not accepted					
150		ABB	Read()		
NACK		.CALC	\$RES==CHR(1)		
170	1	ABB	DiscardReadBuffer()	SEND	
		.CALC	#ABB.state_ABB="ACCEPTED"		

Şekil 3-94. Birden fazla başlangıç koşuluna sahip satırların görünümü.

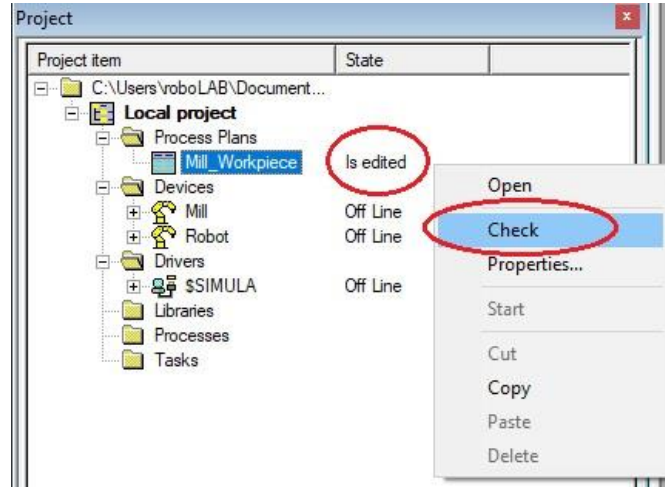
Bir önceki satırda yürütülmüş olan göreve ait geri dönüş değeri; koşullu olarak yazılmış satırlardaki “Condition” hücrelerindeki değerlerden hangisine uyuyorsa, denetleyici ilgili satırdaki görevi yürütür. Örneğin Şekil 3-94’deki planda 170 numaralı satıra dönen değerin “1” olduğu durumda, “.CALC” fonksiyonu “#ABB.state_ABB = ”ACCEPTED”” görevini yürütecektir. Sonuç olarak farklı koşul ve sonraki satır tanımlamaları yaparak kontrol sıralamasında dallanmalar oluşturulabilir.

Örnek uygulamada verilen görevleri yürütebilmek için Şekil 3-95’deki pencerede gösterilen süreç planını girilir.

Line	Condition	Device	Task	Next line	Comment
; Initialize row and column variable					
10		.CALC	%PosX=0		
20		.CALC	%PosY=0		
; Take workpiece from pallet and feed mill					
START		Robot	PickFromPallet(%PosX, %PosY)		
110		Robot	FeedMill		
; Start milling process					
120		Mill	Milling		
; Unfeed Mill and place workpiece back on pallet					
OK		Robot	UnFeedMill		
150		Robot	PlaceOnPallet(%PosX, %PosY)		
; Increment position variables (16 x 16)					
200		.CALC	%PosX=%PosX+1		
210		.NOOP		START	
	16	.CALC	%PosX=0		
220		.CALC	%PosY=%PosY+1		
230		.NOOP		START	
	16	.NOOP		END	

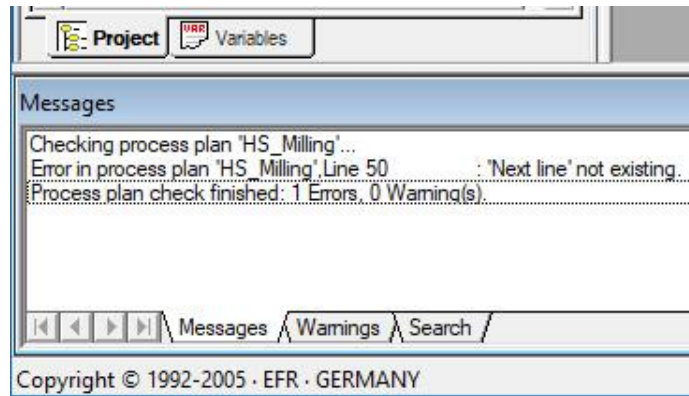
Şekil 3-95. Örnek uygulamadaki “Mill_Workpiece” isimli süreç planı.

Şekildeki planda paletlerin mevcut konumlarını belirten değişkenler vardır. Bu değişkenler robot görevlerinde parametre olarak kullanılmaktadır. Süreç planı tüm paletlerdeki iş parçaları işlenene kadar konum değişkenlerini arttırır. Hazırlanan plan “Mill_Workpiece” ismiyle kaydedilir. “Proje Penceresi”nde bulunan süreç planının durumu Şekil 3-96’deki gibi isminin sağ tarafında görüntülenir.



Şekil 3-96. Süreç planlarının yapısal durumu ve kontrol edilmesi.

COSIMIR Factory Control yazılımı kurulum (setup) modundaysa; yeni oluşturulan veya düzenlenip kaydedilen planların kenarında “Is edited” yazar. Planın isminin üzerine gelip sağ tuş yapıldığında Şekil 3-97’deki gibi gelen menüden “Check” seçilerek plan yapısal olarak kontrol edilir. Eğer planda bir yazım hatası varsa veya plan kontrol edilmemişse, yürütülemez “Not Executable” yazar. “Mesaj Penceresi”nde süreç planında belirlenen hata satır numarasıyla birlikte Şekil 3-97’deki gibi görüntülenir. Hata mesajı çift tıklandığında süreç planı açılır ve ilgili satır seçili olarak ekrana gelir. Hata varsa düzeltilip kaydedildikten sonra tekrar kontrol edilir.



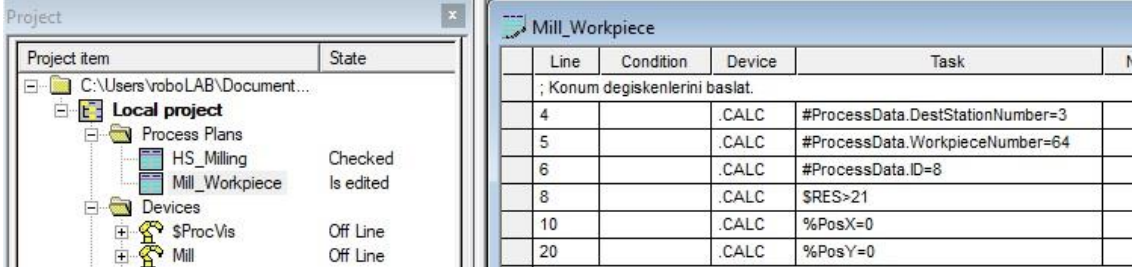
Şekil 3-97. Süreç planında yer alan hataların “Mesaj Penceresi”nde gösterilmesi.

Eğer ki süreç planında hata yoksa COSIMIR Factory Control yazılımı üretim (production) moduna alındığında çalıştırılabilir planların kenarında “Executable” yazar.

Not: COSIMIR Factory Control yazılımı çok dosyalı arayüz (MDI - Multiple Document Interface) yapısında olduğu için, menülerden veya araç çubuklarından verilen kaydet, yeni, vb. komutlar sadece o an için aktif olan dosya için geçerlidir. Bu nedenle örneğin süreç planı penceresi aktifse kaydet fonksiyonu kapalıdır. Çünkü süreç planları bir dosya içine kaydedilemez. Onun yerine Proje Penceresi seçilerek aktif hale getirilirse plan kaydedilebilir.

Projeler içerisinde değişkenlerin tanımlanmasına gerek yoktur. Çünkü kullanıldıkları anda otomatik olarak tanımlanmış da olurlar. Ayrıca değişkenlerin tipi de yoktur; sayısal (tam sayı, kesirli sayı, vb.) veya karakter dizisi şeklinde olabilirler. Değişkenler kapsama alanlarına yerel (local), global ve sistem olmak üzere üçe ayrılır. Değişken isimlerinin önüne eklenen karakter ile ait oldukları grup belirlenir. Ön karakterlerden “%” yerel, “#” global ve “\$” sistem değişkenlerini temsil eder.

Değişkenler ağaç yapısı şeklinde hiyerarşik yapıda da düzenlenebilirler. Bu durum değişkenlerin yönetimini kolaylaştırırken büyük miktardaki verinin yapısal hale getirilmesine (structure) de yardımcı olur. Farklı ağaç seviyeleri, özellikler veya klasörler; değişken isminden sonra konulan “nokta” ile ayrılırlar. Örneğin Şekil 3-98’de görüldüğü gibi “#ProcessData.ID=8” görevinde “ProcessData” nesnesinin “ID” değişkeni “8” olarak atanmıştır. Aynı nesnenin “DestStationNumber” değişkenine “3” değeri, “WorkpieceNumber” değişkenine “64” değeri atanmıştır. Burada yapısal bir tanımlama vardır.



The screenshot shows the COSIMIR Factory Control software interface. On the left, there is a 'Project' tree view with a 'Local project' folder containing 'Process Plans' (with sub-items 'HS_Milling' and 'Mill_Workpiece') and 'Devices' (with sub-items '\$ProcVis' and 'Mill'). The 'Mill_Workpiece' item is selected and its state is 'Is edited'. On the right, there is a table titled 'Mill_Workpiece' with columns 'Line', 'Condition', 'Device', 'Task', and 'N'. The table contains the following data:

Line	Condition	Device	Task	N
; Konum degiskenlerini baslat.				
4		.CALC	#ProcessData.DestStationNumber=3	
5		.CALC	#ProcessData.WorkpieceNumber=64	
6		.CALC	#ProcessData.ID=8	
8		.CALC	\$RES>21	
10		.CALC	%PosX=0	
20		.CALC	%PosY=0	

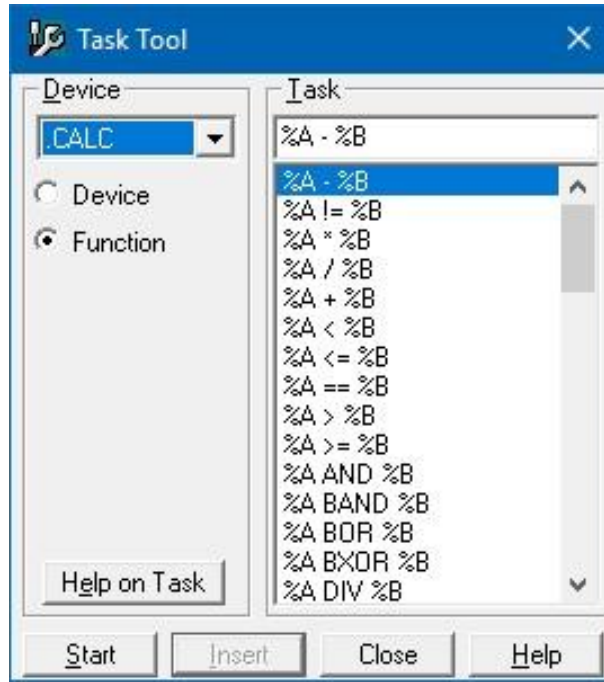
Şekil 3-98. Yerel ve yapısal değişken tanımlama görüntüsü.

Sistem değişkenlerinden “\$RES” her zaman en son yürütülen görev veya sistem fonksiyonunun geri dönüş değerini tutar. Şekil 3-98’de “\$RES” sistem değişkeninin

21'den büyük olma durumu sorgulanmaktadır. Koşul doğruysa "1", değilse "0" değeri dönecektir. Ayrıca "%PosX" ve "%PosY" yerel değişkenlerdir ve değerleri "0"dır. Dikkat edilirse "Değişkenler Penceresi"nde sadece "#" karakteri ile başlayan değişkenler görüntülenmektedir.

Değişken isminden sonra eklenen köşeli parantezler içerisine eklenen indis sayesinde diziler de kolaylıkla tanımlanabilir. Böylece dinamik değişken isimleri kullanılabilir. Örneğin #Position[%n] gibi.

Değişkenler matematiksel fonksiyonlarda hesaplamada kullanılabilir. "View/Task-Tool" (veya "F3" fonksiyon tuşu) penceresi açıldığında kullanılabilecek tüm cihazlar, görevler ve sistem fonksiyonları görüntülenebilir. Örneğin sistem fonksiyonlarından ".CALC" seçildiğinde, süreç planları içerisinde kullanılabilecek tüm temel matematiksel fonksiyonların listesi Şekil 3-99'deki gibi görüntülenir.



Şekil 3-99. "View/Task Tool" ("F3" fonksiyon tuşu) penceresi görünümü.

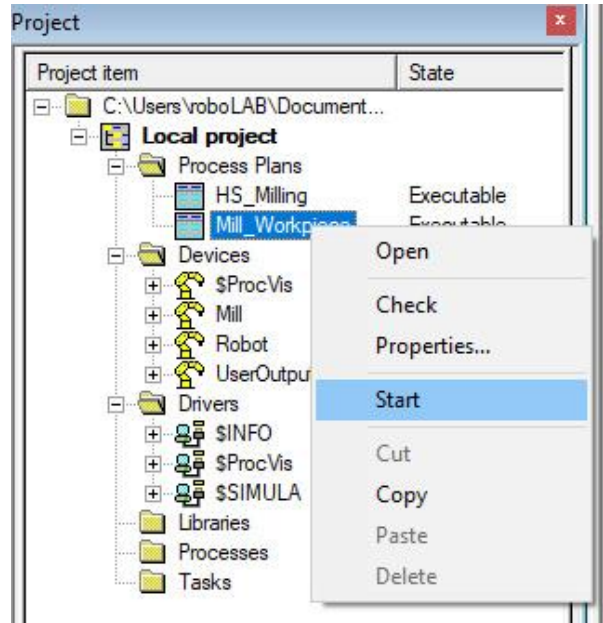
Fonksiyonlar birbirleri ile birleştirilerek daha karmaşık ifadeler elde edilebilir. Pencerede bulunan "Help on Task" butonu tıklanarak görev ile ilgili daha detaylı yardım penceresi görüntülenir. "Insert" butonu tıklanarak seçili olan cihaz, görev veya sistem fonksiyonu süreç planına eklenir. Plan üzerinde eklenen değerlerde değişiklik yapılabilir.

3.1.8.4.3. İmalatı Başlatma

COSIMIR Factory Control yazılımında proje oluşturulduktan veya mevcut bir proje açıldıktan sonra imalata başlayabilmek için yazılımın üretim (production) moduna getirilmesi gereklidir. Üretim modunda iken imalat denetleyicisi aktif durumdadır. Tüm haberleşme arayüzleri başlatılır ve çalışır duruma getirilir. Kullanıcı tarafından süreç planları başlatılabilir ve cihazlar görevleri yürütebilir.

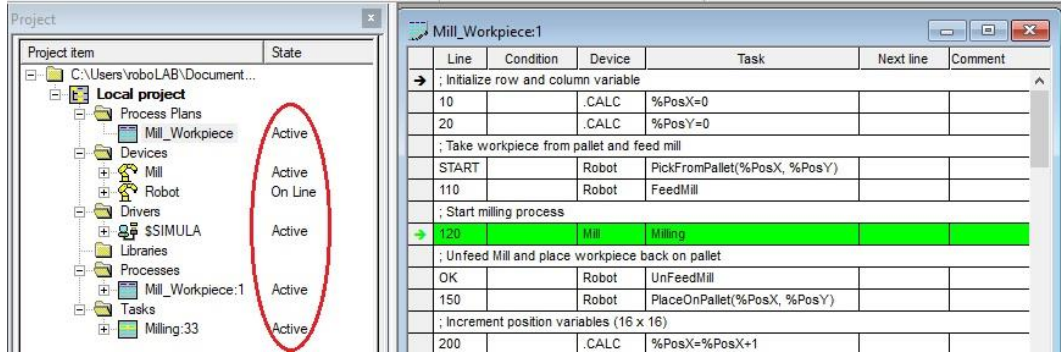
Bölüm 3.1.8.4.2’de verilen örnek projede üretimin başlatılması ile ilgili yürütülecek işlem basamakları aşağıdaki gibidir:

- COSIMIR Factory Control yazılımı üretim moduna alınır. Bu işlemin nasıl yapılacağı ile ilgili detaylar Bölüm 2.1.2.12.1’de verilmiştir.
- “Proje Penceresi”nden örnek uygulama için hazırlanmış olan süreç planı “Mill_Workpiece” seçilir. Şekil 3-100’deki gibi farenin sağ tuş menüsünden “Start” veya menü çubuğu üzerinden “Debug/Start” tıklanarak süreç başlatılır.



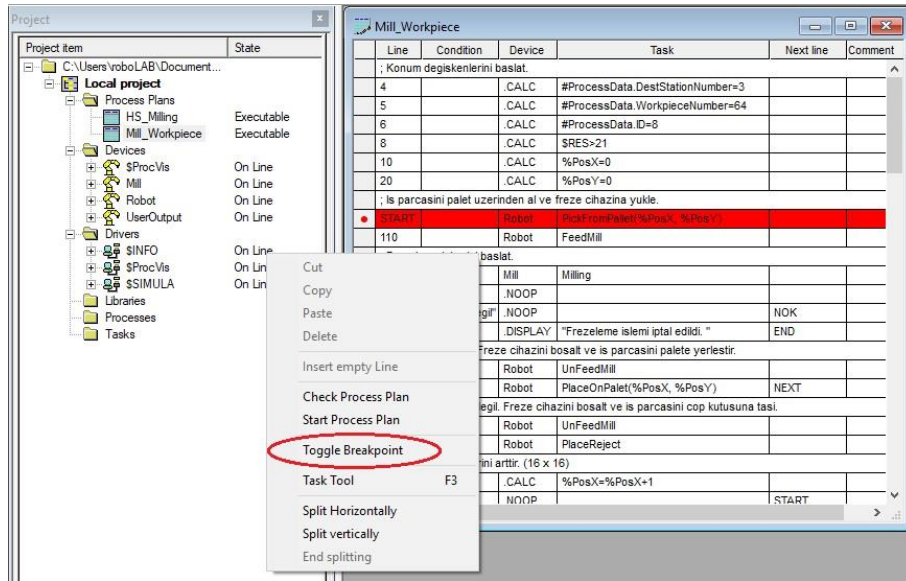
Şekil 3-100. Süreç planının Proje Penceresi üzerinden başlatılması.

- Süreç başlar başlamaz ekranda Şekil 3-101’deki gibi süreç planı penceresi görüntülenir. Bu pencere üzerinden süreç içerisinde yürütülen görevler takip edilebilir. Plan içindeki aktif olan satır yeşil olarak işaretlenir. İstenirse pencere kapatılabilir ve pencerenin kapatılması süreci sonlandırmaz. Aktif ve yürütülmekte olan süreçler ve görevler “Proje Penceresi”ndeki ağaç yapısında görüntülenir. Detayı görüntülenmek istenen süreç planı çift tıklandığında süreç planı penceresi açılır.



Şekil 3-101. Önceden başlatılmış süreç planı ve aktif olan süreçlerin takibi.

- COSIMIR Factory Control yazılımının sahip olduğu hata ayıklama (debug) özelliği sayesinde karmaşık kontrol programları denenerek adım adım çalıştırılabilir. Süreç planındaki herhangi bir satıra hata ayıklama işareti (breakpoint) ekleyebilmek için ilgili satır üzerine gelinerek sağ tuş tıklanarak gelen menüden “Toggle Breakpoint” veya “F9”a basılabilir. İlgili satır Şekil 3-102’deki gibi kırmızı renk ile işaretlenir.



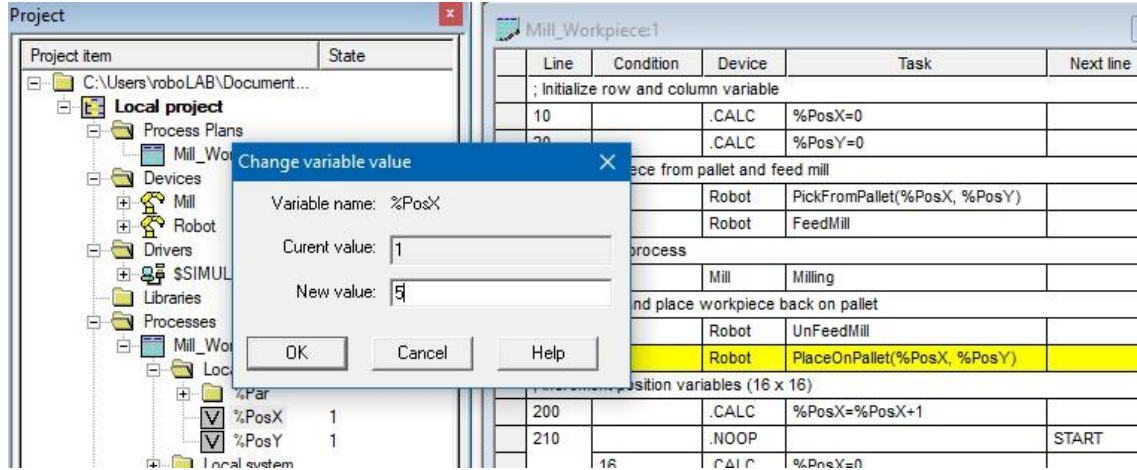
Şekil 3-102. Süreç planına hata ayıklama noktası (durak noktası - breakpoint) ekleme.

Örnek uygulamanın “Mill_Workpiece” isimli süreç planının “START” etiketli satırına Şekil 3-102’de görüldüğü gibi hata ayıklama işareti eklensin. Bu durumda süreç bu satırda duracaktır. “Debug/Next Step” veya “F8” tıklanarak plan satır satır işlenebilir.

Herhangi bir anda süreci durdurmak için “Debug/Stop Process” veya süreç planı üzerinde fare sağ tuş menüsünden “Stop” tıklanır. Planda üzerinde durulan satır sarı renk ile

işaretlenir. “Debug/Execute” tıklandığında hata ayıklama modu iptal olup imalat süreci kaldığı yerden yürütülmeye devam eder.

- “Proje Penceresi”nde herhangi bir sürece ait yerel değişkenler görüntülenebilir. İstenilen değişken üzerine çift tıklanarak veya sağ tuş menüsünden “Change” tıklanarak Şekil 3-103’deki gibi ilgili değişken içeriği değiştirilebilir.



Şekil 3-103. Sürece ait yerel değişken değerlerinin değiştirilmesi.

- Üretimi durdurmak için araç çubuğu üzerindeki “Production” düğmesine tıklanır.

3.1.8.4.4. İleri Seviye Programlama ve Kullanım

Bir önceki bölümde verilen örnekte FMS100 üzerinde COSIMIR Factory Control yazılımının temel fonksiyonları gösterilmiştir. Bu ve bundan sonraki bölümlerde yazılımın ileri seviye sistem fonksiyonları açıklanmaktadır. Böylelikle daha ileri seviye programlar ve karmaşık kontrol yapıları kurulabilecektir.

Haberleşme Protokolü (Handshake) Oluşturma

Bir önceki bölümde verilen örnek uygulamada imalat hücresine ait görevleri temsilen “SIMULA” sürücüsü kullanılmıştı. Sürücü kurulurken yürütülecek görevlere ait benzetim zamanları ve geri dönüş değerleri ayarlanmıştı. Gerçek otomasyon cihazları arası iletişimde haberleşme sürücüler ve onlar içerisinde tanımlı özel görevler kullanılır.

Genellikle bir otomasyon cihazı ile kontrol sistemi arası iletişim için bir protokol (handshake) tanımlanır. Şekil 3-104’de bir ABB robot ile COSIMIR Factory Control kontrol yazılımı arasında, seri arayüz (RS232) üzerinden kurulan iletişime ait protokolün bir bölümü görülmektedir.

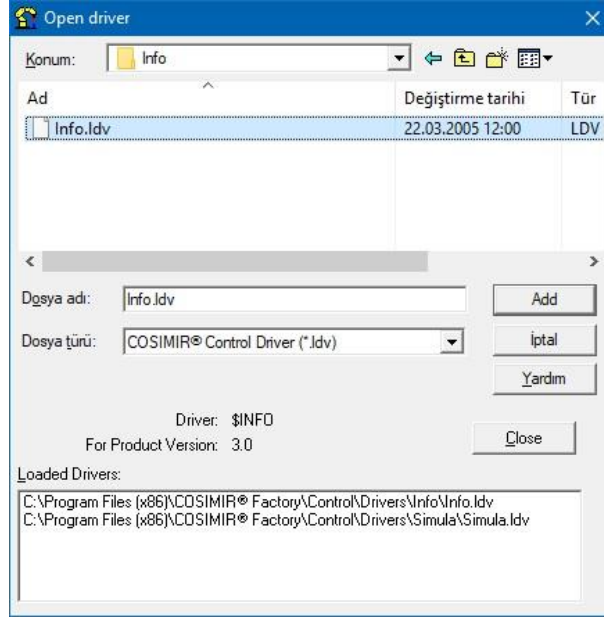
HS_ABB								
	Line	Condition	Device	Task	Next line	Comment		
	; Send command							
	SEND1		ABB	Write(%command)				
	; Send command copy							
	110		ABB	Read()				
	ACK		.CALC	\$RES==CHR(6)				
	130	1	ABB	DiscardReadBuffer()				
			ABB	DiscardReadBuffer()	SEND			
	SEND2		ABB	Write(%command)				
	; Retry, if not accepted							
	150		ABB	Read()				
	NACK		.CALC	\$RES==CHR(1)				
	170		ABB	DiscardReadBuffer()				
➔		1	.CALC	#ABB.state_ABB="ACCEPTED"				

Şekil 3-104. ABB marka robot ile RS232 haberleşme süreç planının bir bölümü.

Denetleyicinin temel görevlerinden biri de kendine bağlı cihazlar ile belirli protokoller üzerinden haberleşme kurabilmesidir. Birçok cihaz değişmez (statik) protokolleri destekler. Bu nedenle istenilen bir cihazı kontrol sistemiyle ilişkilendirebilmek için özel süreç planları oluşturulur. COSIMIR Factory Control yazılımı haberleşme protokollerinin tanımlı olduğu süreç planları sayesinde çevresindeki ve bünyesindeki sistemlerle/cihazlarla iletişim kurar. Bunu yaparken de haberleşme protokollerindeki görevleri içeren süreç planlarını kullanır.

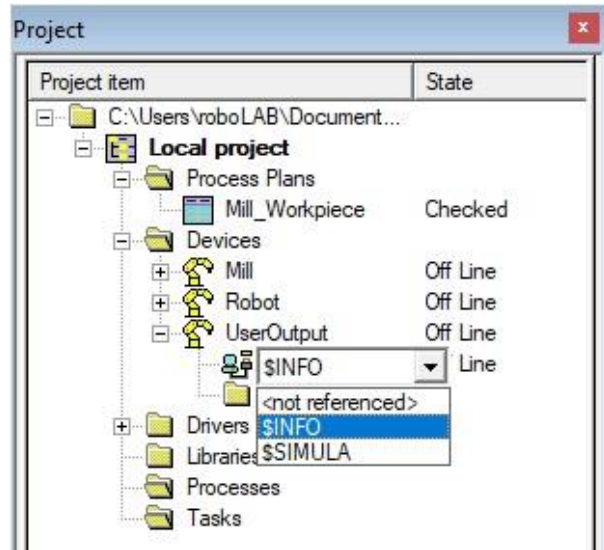
“Info.ldv” dosyası, haberleşme protokolü süreç planlarına bir örnektir. Basit kullanıcı girişlerinin nasıl uygulandığını da gösterir. Bunun için yürütülecek işlem basamakları aşağıdaki gibidir:

- Menü çubuğu üzerindeki “File/Load Driver” tıklanarak Şekil 3-105’deki pencereden “...Drivers/Info/Info.ldv” seçilir ve “Add” tıklanır.



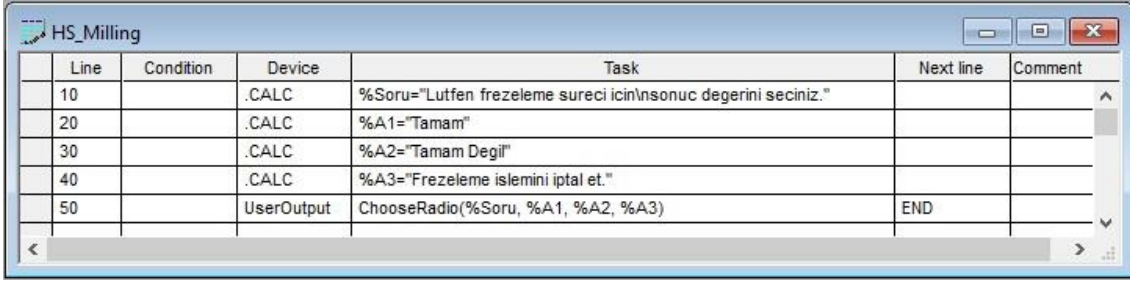
Şekil 3-105. COSIMIR Factory Control yazılımı projesine “Info” sürücüsü ekleme.

- “UserOutput” isimli yeni bir cihaz oluşturulur ve Şekil 3-106’deki gibi “Info” sürücüsü ile ilişkilendirilir.



Şekil 3-106. “UserOutput” isimli cihaz ile “Info” sürücüsünün ilişkilendirilmesi.

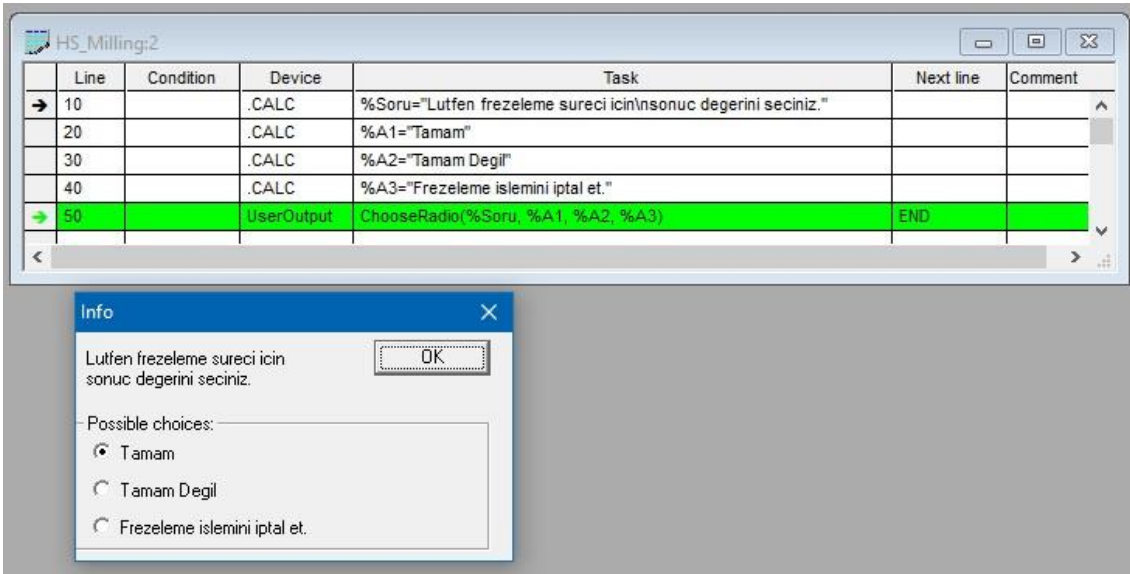
- “HS_Milling” isimli yeni bir süreç planı oluşturulur. “HS” ön eki “hand shake” kelimesinin baş harflerinden oluşturulmuştur. Bu nedenle “HS” ön ekli süreç planları genellikle cihazlar ile haberleşme için kullanılmaktadır. Bu plan frezeleme sürecinin muhtemel sonuçlarını, giriş penceresi (InputBox) üzerinden kullanıcıdan öğrenerek, freze cihazının görevlerini benzetmede kullanılır. Süreç planına çift tıklanarak boş olan içeriği görüntülenir. Süreç planı içeriği Şekil 3-107’deki gibi doldurulur.



Line	Condition	Device	Task	Next line	Comment
10		.CALC	%Soru="Lutfen frezeleme sureci icin\nsonuc degerini seciniz."		
20		.CALC	%A1="Tamam"		
30		.CALC	%A2="Tamam Degil"		
40		.CALC	%A3="Frezeleme islemini iptal et."		
50		UserOutput	ChooseRadio(%Soru, %A1, %A2, %A3)	END	

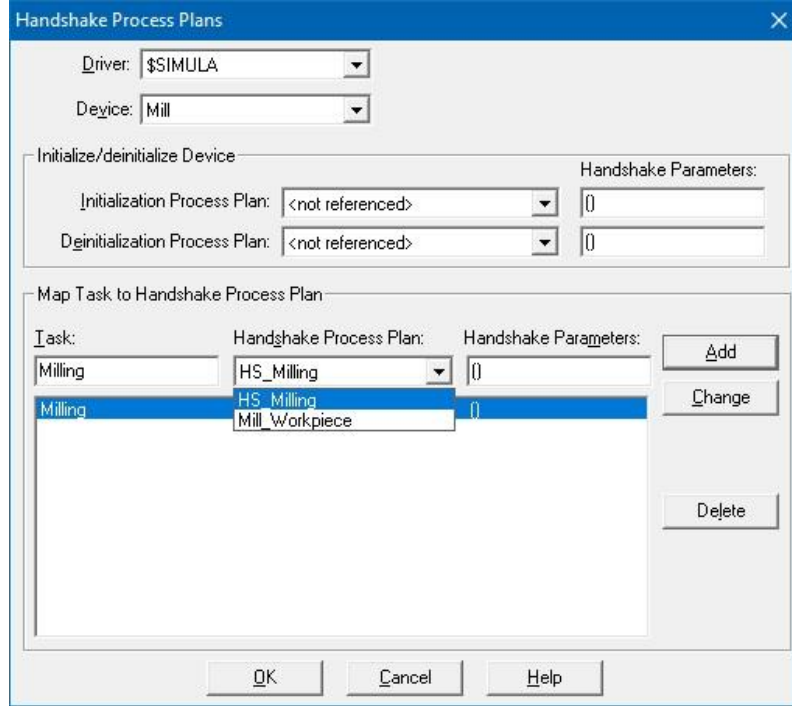
Şekil 3-107. “HS_Milling” isimli süreç planı içeriği.

- Süreç planı içerisindeki değişkenler (%A1, %A2, vb.), kullanıcıya gösterilecek ve kullanıcının bildirimlerini sisteme aktaracak giriş penceresinde yer alacak bilgilerin ve seçeneklerin dinamik olarak tanımlanabilmesi için kullanılır. Plan içerisindeki “50” numaralı satırda yer alan “ChooseRadio” komutu, değişkenlere girilen değerlere göre Şekil 3-108’de görülmekte olan giriş penceresinin görüntülenmesini sağlar. Aynı zamanda süreç planının son satırı olduğunu belirtmek için “Next Line” sütununa “END” yazılır. Komutun geri dönüş değeri aynı zamanda süreç planının da nihai geri dönüş değeridir.



Şekil 3-108. “ChooseRadio” isimli kullanıcı arayüzünün görüntüsü.

- “HS_Milling” isimli süreç planı, “Mill” cihazının “Milling” isimli görevinin benzetimi için hazırlanmıştır. O nedenle plan haberleşme protokolü olarak kullanılabilir. “Project” menüsünden “Handshake Process Plans” tıklanır ve Şekil 3-109’deki pencere görüntülenir.



Şekil 3-109. “Simula” sürücüsünde, “Mill” cihazının “Milling” isimli görevine “HS_Milling” süreç planının atanarak haberleşme altyapısının oluşturulması.

- Pencerde sürücü (driver) kutusundan “\$SIMULA” ve cihaz (device) kutusundan “Mill” seçilir. Task kutusuna “Milling” yazılır ve Handshake Process Plan kutusundan “HS_Milling” seçilir. “Add” butonuna basılmasıyla birlikte iletişim protokolü projeye eklenir.
- Kullanıcının seçimlerinin daha önceden Bölüm 3.1.8.4.2’de oluşturulmuş olan Şekil 3-95’deki “Mill_Workspace” isimli süreç planı içerisinde değiştirilmesi gereklidir. Bunun için süreç planının 130 ile 210 numara ile etiketlenmiş satırları arası Şekil 3-110’deki gibi değiştirilir.

120		Mill	Milling	
130	"Tamam"	.NOOP		
	"Tamam Degil"	.NOOP		NOK
		.DISPLAY	"Frezeleme islemi iptal edildi."	END
; Is parcası tamam. Freze cihazını boşalt ve is parcasını palete yerleştir.				
OK		Robot	UnFeedMill	
150		Robot	PlaceOnPalet(%PosX, %PosY)	NEXT
; Is parcası tamam degil. Freze cihazını boşalt ve is parcasını cop kutusuna taşı.				
NOK		Robot	UnFeedMill	
160		Robot	PlaceReject	
; Konum degiskenlerini artır. (16 x 16)				
NEXT		.CALC	%PosX=%PosX+1	
210		.NOOP		START

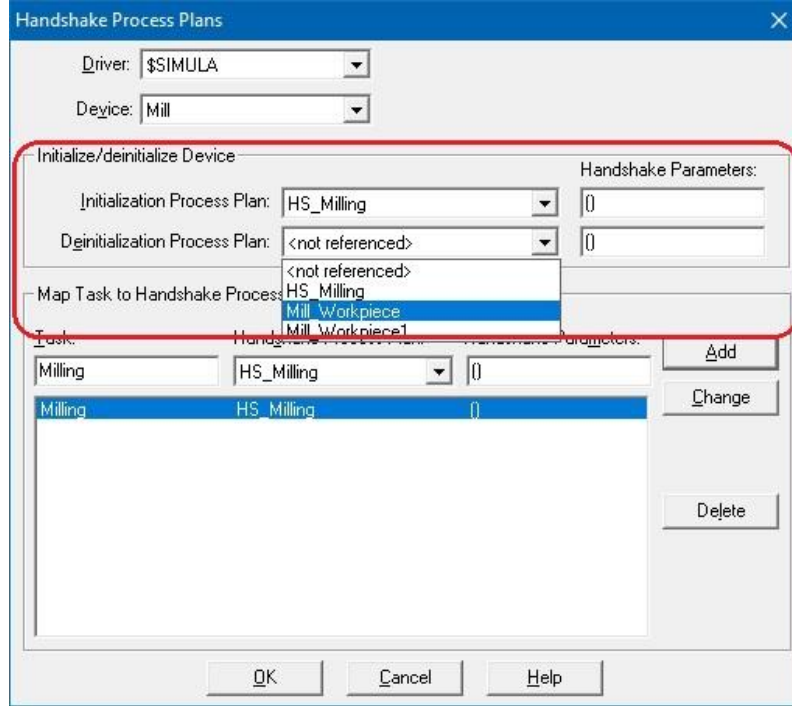
Şekil 3-110. “Mill_Workspace” süreç planında yapılacak değişiklikler.

- “Milling” isimli görevin geri dönüş değeri 130 numaralı satırdaki koşullu dallanmada değerlendirilir. “.NOOP” (no operation) sistem fonksiyonu daha iyi kuma sağlamak ve temiz bir süreç planı yapısı oluşturmak için kullanılır. Burada oluşturulan koşullu dallanma yapısına göre program; eğer parça düzgün işlenmişse “OK” etiketli satıra dallanacaktır. Sonrasında sırasıyla parçanın freze cihazından alınması (task: UnFeedMill) ve palet üzerine yerleştirme (task: PlaceOnPallet) görevleri yürütülecektir. Eğer parça düzgün işlenmemişse “NOK” etiketli satıra dallanılacak; sonrasında sırasıyla yine parçanın freze cihazından alınması (task: UnFeedMill) görevi yürütüldükten sonra atık kutusuna taşınması (task: PlaceReject) görevleri yürütülecektir. Eğer kullanıcı “Cancel pallet processing” seçmişse ekranda hata mesajı görüntülenecek (“Pallet processing cancelled”) ve süreç planı sonlandırılacaktır.

Üretim Sürecini Başlatma ve Durdurma

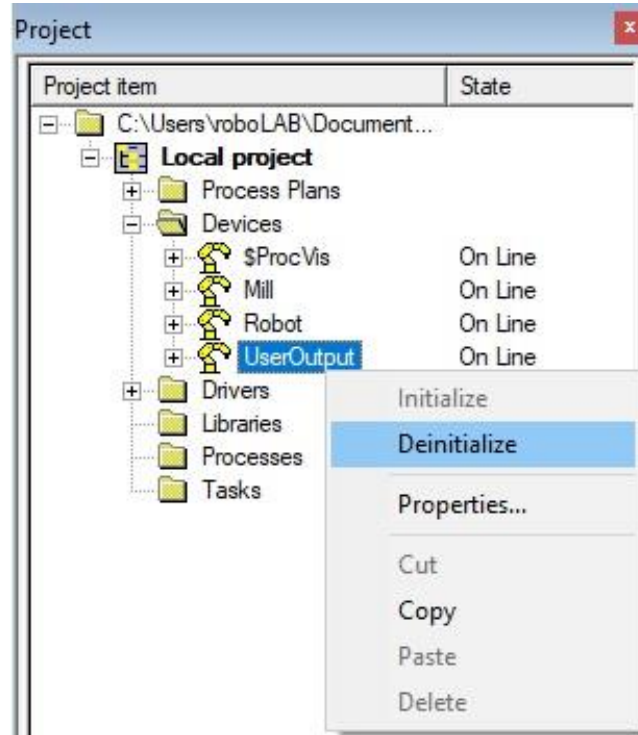
COSIMIR Factory Control projesi üretim (production) moduna alındığında ilk önce sürücüler üzerinden otomasyon cihazları ile iletişim kurulur. Örneğin RS232 portu açılır veya Profibus sürücüsü Profibus iletişimini başlatır. Bu nedenle üretim moduna geçildiğinde ilk önce sürücüler başlatılır. Sonrasında her cihaz kendi özel ayarlarına uygun olarak başlatılır. Sürücülerin başlatılması, kullanılan arayüze ve ana bilgisayarın performansına göre birkaç saniyelik zaman alabilir. Sürücülerin ve cihazların başlatılma durumu önceden Şekil 3-101’deki pencerede verildiği gibi görüntülenir.

- İletişime başlamış olan sürücü ve cihazların kenarında “On Line”, iletişim başlatma işlemleri devam edenlerin yanında ise “Initializing” yazılı olur. Aynı pencere üretim modu sonlandırılırken de ekranda görüntülenir.
- Menü çubuğundan “Project/Handshake Process Plans” seçilir. Ekrana gelen Şekil 3-111’ deki pencerede kırmızı ile işaretlenmiş bölümden “Initialization Process Plan” ve “Deinitialization Process Plan” seçilir. Denetleyici başlangıç sürecinden sonra ve sonlandırma sürecinden önce burada tanımlanan planları yürütür. İstenirse parametre tanımlaması da yapılabilir. Böylelikle özellikle başlangıç ve sonlanma zamanlarında ihtiyaç duyan uygulamalarda bu özellik kullanılabilir.



Şekil 3-111. “Handshake Process Plan” penceresinden cihazı başlatma ve sonlandırma süreçler ile ilgili ayarların yapılması.

- Üretim modundayken herhangi bir cihaz Şekil 3-112’deki gibi başlatılıp sonlandırılabilir. Böylelikle her bir cihaz bağımsız olarak test edilebilir, aksaklıklar tespit edilebilir.



Şekil 3-112. Proje Penceresi üzerinden herhangi bir cihazın sonlandırılıp başlatılması.

İfadeler, Değişkenler ve Operatörler

COSIMIR Factory Control yazılımında oluşturulan bir üretim süreci bünyesindeki görevlerin tanımlanmasında veya görev parametrelerinin tanımlanmasında ifadeler (expression) kullanılır. İfadeler aritmetik hesaplamalar, mantıksal ifadeler veya metin fonksiyonları olabilir. İfadeler tanımlanırken doğal olarak içlerinde değişkenler ve operatörler kullanılabilir.

COSIMIR Factory Control yazılımında değişkenlerinin tipinin (boolean, tamsayı, ondalıklı sayı, karakter, metin vb.) tanımlanmasına gerek yoktur. Kullanıldıkları ilk yerde otomatik olarak oluşturulurlar. Tamsayı, ondalıklı sayı gibi tüm aritmetik değişkenler hafızada (RAM) 64bit yer kaplar. İkili (binary) sistemdeki tüm değişkenler ise hafızada 32bit yer kaplar. Metin tipi değişkenler ise 10.000.000 karakter ile sınırlandırılmıştır. Karşılaştırma gibi boolean değişkenler doğru (1) ve yanlış (0) olarak ifade edilirler.

İfadelerin içerisinde kullanılan temel operatörler aşağıdaki gibidir. Tüm operatör listesi ve açıklamalarına yazılımın yardım sayfasından erişilebilir:

- Matematiksel operatörler: toplama (+), çıkarma (-), çarpma (*), bölme (/), mod alma (MOD), bölme (DIV)
- Karşılaştırma operatörleri: küçük (<), küçük eşit (<=), eşit (==), büyük eşit(>=), büyük(>), eşit değil(!=)
- Mantıksal operatörler: VE kapısı (AND), VEYA kapısı (OR), Özel VEYA (XOR), Değil kapısı veya eşlenik alma işlemi (NOT)
- Binary mantıksal işlem operatörleri: binary AND (BAND), binary OR (BOR), binary Özel VEYA (BXOR), binary eşlenik alma işlemi (BNOT)
- Binary öteleme ve döndürme operatörleri: sola kaydırma (SHL), sağa kaydırma (SHR), sola döndürme (ROL), sağa döndürme (ROR)
- Metin birleştirme operatörü (&)
- Açıklama satırı operatörü (;)

İşlemlerde operatör öncelikleri soldan sağa ve yukarıdan aşağı şu şekilde belirlenmiştir:

- *, /, MOD, DIV
- SHL, SHR, ROL, ROR
- +, -, <, <=, ==, >=, >, !=, AND, OR

Temel operatörler dışında ileri seviye bazı aritmetik operatörler şunlardır:

- TRUNC(metin ifade): metni tamsayı değere dönüştürür.
- ABS(sayısal ifade): sayının mutlak değerini alır.

- SQRT(sayısal ifade): sayının karekökünü alır.
- EXPT(sayısal ifade1, sayısal ifade2): üst alma işlemi, sayısal ifade1 üzeri sayısal ifade2 işlemini yapar.
- PI(): Pi sayısının değerini tutar.
- {SIN, COS, TAN, ASIN, ACOS, ATAN}(sayısal ifade): Trigonometrik işlemleri yaparlar.
- {MIN, MAX}(ifade1, ifade2): iki ifadeyi karşılaştırıp, en küçük veya en büyük değeri geri döndürürler.

COSIMIR Factory Control yazılımının desteklediği bazı ileri seviye metin fonksiyonları şunlardır:

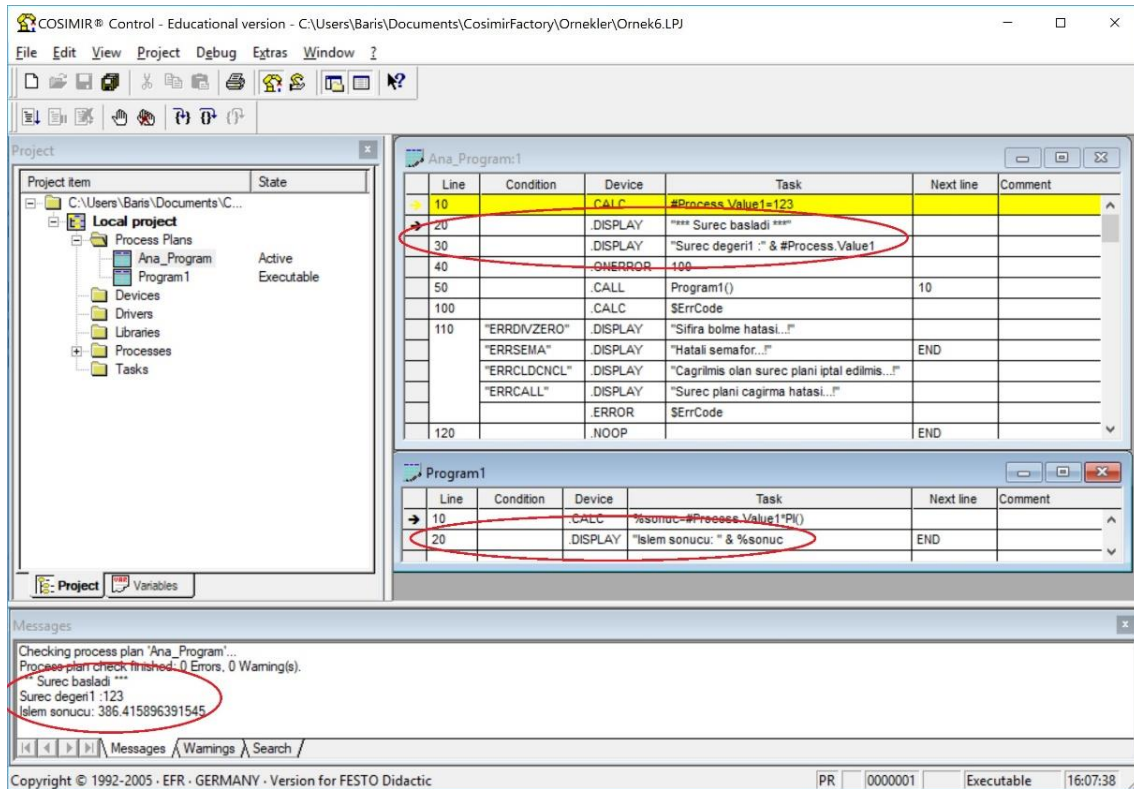
- ISNUM(ifade): eğer ifadenin içeriği sayısal bir değerse 1, değilse 0 değerini döndürür.
- LEN(ifade): ifadenin içerisindeki karakter sayısını döndürür.
- FIND(ifade1, ifade2): ifade1 içerisinde ifade2'nin ilk geçtiği yerin sıra numarasını verir. Sıra numaraları 0'dan başlar. Eğer ifade2 bulunamazsa -1 değerini döndürür.
- CHR(ifade): ifadede belirtilen ASCII kodunun karşılığı olan karakteri döndürür.
- {UPPER, LOWER}(ifade): ifadedeki karakterleri büyük/küçük karaktere dönüştürür.
- NOW(): şimdiki tarih ve saati tamsayı olarak döndürür.
- DATE(ifade): ifadeyi YYYY-AA-GG biçimine döndürür. (Y:yıl, A: ay, G: gün)
- TIME(ifade): ifadeyi SS:DD:SanSans:SalSalSal biçiminde döndürür. (S: saat, D: dakika, San: saniye, Sal: salise)
- DATETIME(ifade): ifadeyi YYYY-AA-GG-SS:DD:SanSan:SalSalSal biçiminde döndürür.

Sistem Fonksiyonları

Süreç planları içerisinde önceden tanımlanmış sistem fonksiyonları kullanılır. Bu fonksiyonlar sayesinde süreç planları üretim kontrol işlemlerini yürütür. COSIMIR Factory Control yazılımında önceden tanımlanmış sistem fonksiyonları veya cihazlarına “View/Task Tool” menüsünden veya “F3” tuşuna basarak görüntülenen pencereden ulaşılabilir. İlgili fonksiyon seçilip “Help on Task” tıklandığında İngilizce yardım menüsü açılır. Bu bölümde süreç planları içerisinde sıklıkla kullanılan sistem fonksiyonları ve görevleri örneklerle açıklanmıştır. Bölümde açıklanmayan fonksiyonlar hakkında detaylı bilgiye yardım menüsünden ulaşılabilir. Süreç planı içerisinde, sistem fonksiyonu “Device” hüccresine; yürüteceği görev ise “Task” hüccresine yazılır.

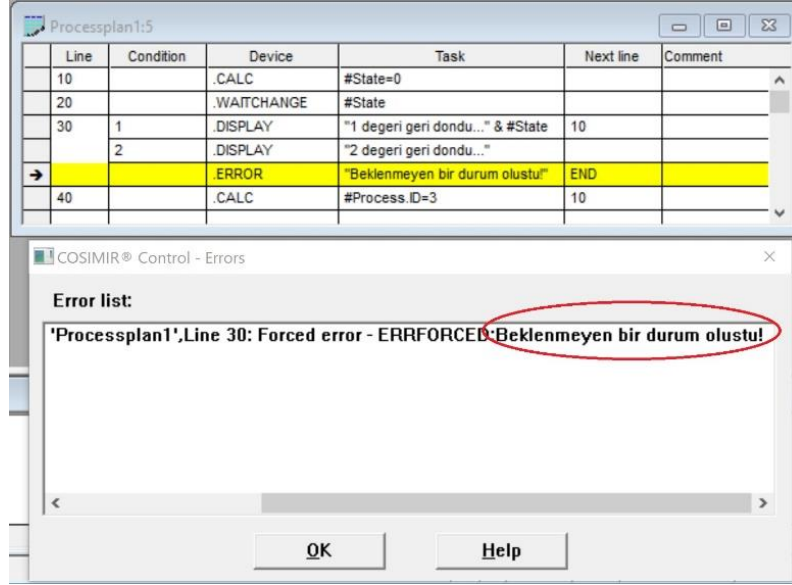
Süreç planları bünyesindeki “Device” sütununda kullanılan temel sistem fonksiyonları ve görevleri kısaca aşağıda açıklanmıştır.

- “.DISPLAY”: “Mesaj Penceresi”nde istenilen metni görüntüler. Şekil 3-113’deki örnekte “Ana Program1” isimli süreç planındaki 20 numaralı satırda Task hücrelerinde tırnak işareti içerisinde yazılı olan metin aynen Mesaj Penceresi’nde görüntülenir. İki farklı metin veya bir metin ile değişken içeriği birleştirilip de gösterilmek istenirse, 30 numaralı satırdaki gibi araya “&” işareti konulur.



Şekil 3-113. “.DISPLAY” fonksiyonu kullanımına örnek bir uygulama.

- “.NOOP”: Herhangi bir işlemin yapılmadığı fonksiyondur. Süreç planı içerisinde işlem yapılmak istenmeyen satırda “Device” hücreleri boş kalmaması için kullanılır. Böylelikle planın kullanıcı tarafından okunması ve anlaşılması kolaylaşır.
- “.ERROR”: Ekranda yeni bir pencere içerisinde hata mesajı görüntülenmesini sağlar. Şekil 3-114’de görüldüğü gibi “Task” hücrelerinde yazılı olan metni veya değişkenin içeriğini, hata mesajına ekler.



Şekil 3-114. Hata mesajı penceresi.

- “.ONERROR”: Süreç içerisinde oluşan hataların yönetilmesi için kullanılır. Hata durumunda Şekil 3-115 “Task” hücresi içerisindeki etiketli satıra dallanmasını sağlar.

30		.ONERROR	100	
40		.CALL	Program1()	10
100		.CALC	\$ErrCode	

Şekil 3-115. “.CALC” fonksiyonu kullanımına örnek kod satırları.

- “.CALC”: Aritmetik hesaplama, karşılaştırma, değişken atama, metin işlemleri vb. görevleri yürütür. Şekil 3-116’de verilen örnek süreç planı, 100’e kadar olan asal sayıları bulmakta ve sonucu “Mesaj Penceresi”nde görüntülemektedir. Örnek içerisinde “.CALC” fonksiyonu ile yürütülmüş birçok görev ve çıktıları yer almaktadır.

Line	Condition	Device	Task	Next line	Comment
; *** Asal sayı bulma programı ***					
BASLA		.CALC	%n = 3		Sayacı başlatır.
; *** ANA_DONGU başlangıcı ***					
ANA_DONGU		.CALC	%n <= 100		Sayac değerinin 100'den küçük ve eşit olma durumunu sorgular.
200	0	.CALC	%n	500	Sayac değeri 100'den büyükse 500 nolu satıra dallanır.
		.CALC	%i = 2		Sayac değeri 100'den küçük veya eşitse bölenleri test eder
; 2'den n/2'ye kadar tüm bölenleri test eder.					
BOLUM_DONGUSU		.CALC	%i <= %n / 2		n/2'ye kadar olan bölenleri test eder.
400	0	.DISPLAY	%n & " bir asal sayıdır."		n bir asal sayıdır.
		.CALC	(%n MOD %i) == 0		Bölen bulundu, asal değil.
BOLUM_TESTI		.CALC	%n = %n + 1	ANA_DONGU	Bir sonraki sayıyı kontrol eder.
	0	.CALC	%i = %i + 1	BOLUM_DONGUSU	Bir sonraki bölene kontrol eder.
; *** ANA_DONGU sonu ***					
500		.DISPLAY	"Surec planinin sonu."	END	

Şekil 3-116. “.CALC” fonksiyonu kullanımına örnek bir uygulama: asal sayı bulmaca.

- “.START”: İstenilen herhangi bir süreç planını çalıştırır. Fonksiyonunun kullanıldığı süreç planı, çağrılan alt süreç planındaki görevlerin tamamlanmasını beklemeden kaldığı yerden çalışmaya devam eder. Böylelikle her iki plan da paralel olarak çalışmış olur.
- “.CALL”: İstenilen herhangi bir süreç planını çağırmak için kullanılır. “.START” fonksiyonundan tek farkı, “.CALL” fonksiyonu çağrılan alt süreç planındaki görevlerin tamamlanmasını bekler. Alt süreç tamamlandıktan sonra ana süreç kaldığı yerden çalışmaya devam eder. Her iki fonksiyonun da kullanımına yönelik örnek Şekil 3-117’de verilmiştir.

Line	Condition	Device	Task	Next line	Comment
10		CALC	#Process.ID=3		
20		START	HS_Robot()		"HS_Robot" isimli surec plani paralel olarak başlatılarak robot ile iletişime başlanır.
30		CALL	Arayuz_Secim()		"Arayuz_Secim" isimli surec plani başlatılır ve sonucu beklenir.
40	1	NOOP		10	Eğer 1 değeri geri donmusse 10 nolu satıra dallanır.
	2	CALC	#Process.Value1=851	120	Eğer 2 değeri geri donmusse #Process.Value1 degiskenine 851 degerini atar ve 120 nolu satıra dallanır.
	3	CALC	#Process.Value2=7	200	Eğer 3 değeri geri donmusse #Process.Value2 degiskenine 7 degerini atar ve 200 nolu satıra dallanır.
		CALC	#Process.Value=0		Farkli bir deger geri donmusse bir sonraki satıra dallanır.
50		DISPLAY	"Deger yanliş"	10	
120		NOOP		END	

Şekil 3-117. “.START” ve “.CALL” fonksiyonları kullanımına örnek süreç planı.

- Örnekte 20 numaralı satırda "HS_Robot" isimli alt süreç planı, ana süreç planına paralel olarak başlatılarak robot ile iletişime başlanır. 30 numaralı satırda

"Arayuz_Secim" isimli süreç planı başlatılır ve sonucu gelene kadar beklenir. 40 numaralı satırda, eğer 1 değeri geri dönmüşse 10 numaralı satıra dallanır. Eğer 2 değeri geri dönmüşse, "#Process.Value1" değişkenine 851 değeri atanır ve 120 numaralı satıra dallanır. Eğer 3 değeri geri dönmüşse "#Process.Value2" değişkenine 7 değeri atanır ve 200 numaralı satıra dallanır. Eğer farklı bir değer geri dönmüşse bir sonraki satıra dallanır. 50 numaralı satırda Mesaj Penceresinde "Yanlış süreç değeri!" mesajı görüntülenerek tekrar 10 numaralı satıra dallanır.

- ".KILL": İstenilen bir süreç planını sonlandırır. Task hücreğine süreç planının ismi girilmişse, ilgili süreç planını sonlandırır. Eğer Task hücreğine bir değişken ismi girilmişse, en son başlatılmış olan süreç planını sonlandırır ve sonuç değerini değişken içeriğine atar. Bu fonksiyon ile ilgili örnek Şekil 3-118'de verilmiştir.

Line	Condition	Device	Task	Next line	Comment
10		.START	Processplan1()		Processplan1 başlatılır.
20		.CALC	%ProcPID=\$RES		Baslatılan surec planına ait ID numarası okunur.
30		.WAITUNTIL	\$Shutdown==1		Cosimir CONTROL programi uretim modundan cikana kadar beklenir.
40		.KILL	%ProcPID	END	Baslatilmis olan Processplan1() sureci sonlandirilir.

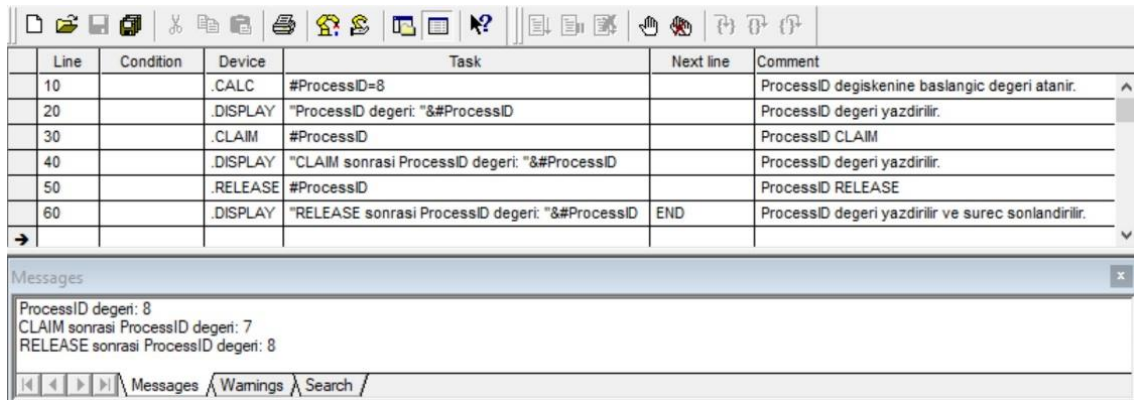
Şekil 3-118. ".KILL" fonksiyonu kullanımına örnek süreç planı.

- ".WAITCHANGE": Herhangi bir değişkenin değeri değişene kadar bekler. Task hücreğinde yazılı olan değişken değeri değiştiğinde, bir sonraki satıra veya Next Line hücreğinde belirtilen satıra dallanır.
- ".WAITUNTIL": Herhangi bir değişkenin değeri istenilen koşula uyana kadar bekler. Task hücreğinde yazılı olan koşul sağlandığında, bir sonraki satıra veya Next Line hücreğinde belirtilen satıra dallanır.
- ".SPLIT" ve ".MERGE": Süreç planında paralel çalışan kollar oluşturur. Bazı özel uygulamalarda aynı anda birden çok görevin yürütülmesi gerekebilir. ".SPLIT" fonksiyonu ile ayrılan dallar mutlaka ".MERGE" fonksiyonu ile birleştirilir.
- Device hücreğine ".SPLIT" yazıldıktan sonra Task hücreğine paralel yürütülmek istenen satırların Satır etiketleri, virgül ile ayrılarak yazılır. Next Line hücreğine ise paralel dallanmanın sonlandırılacağı satır numarası yazılır ve bu satırın Device hücreğinde ".MERGE" yazılı olur. Böylelikle ayrılmış olan paralel dallar birleşerek süreç planına kaldığı yerden devam edilir.
- ".CLAIM" ve ".RELEASE": ".CLAIM" fonksiyonu bir semafor ayırırken (rezerve) ".RELEASE" fonksiyonu semaforu serbest bırakır.

Semaforlar paralel süreçleri senkronize etmek için kullanılır. Örneğin bir imalat hücreğinde, önceden başlatılmış bir imalat süreci kapsamında çalışmakta olan bir cihaza veya değişkene, başka bir imalat süreci içerisinden erişimi engeller. Böylelikle çakışma

olasılığını önlemeyi amaçlar. Semaforlar negatif olmayan tamsayılardan (semafor ≥ 0) oluşan değişkenlerdir.

- “.CLAIM” ve “.RELEASE” fonksiyonları bir semaforu birer birer arttırıp azaltabilir. Semafor değişkenler pozitif tam sayılar olması gerektiğinden dolayı semafora o değeri ile “.CLAIM” eden herhangi bir imalat süreci durdurulacaktır. Semafora 1 değeri ile “.RELEASE” eden herhangi bir süreç ise çalışmaya devam edecektir.
- Başlangıçta semafor değişken 1 değeri başlatılmalıdır. “.CLAIM” fonksiyonu daha önce başlatılmamış bir değişkene uygulanır. Böylelikle her “.CLAIM” fonksiyonu yürütüldüğünde sadece bir imalat süreci değişkene erişebilir. Süreç tamamlandığında semafor değişken “.RELEASE” fonksiyonu ile serbest bırakılmalıdır. Benzer şekilde semafor değişkenler 2 ve 3 numaralarıyla başlatılabilir ve böylelikle birden fazla “.CLAIM” fonksiyonu yürütülebilir.



Line	Condition	Device	Task	Next line	Comment
10		.CALC	#ProcessID=8		ProcessID degiskenine baslangic degeri atanir.
20		.DISPLAY	"ProcessID degeri: "&#ProcessID		ProcessID degeri yazdirilir.
30		.CLAIM	#ProcessID		ProcessID CLAIM
40		.DISPLAY	"CLAIM sonrasi ProcessID degeri: "&#ProcessID		ProcessID degeri yazdirilir.
50		.RELEASE	#ProcessID		ProcessID RELEASE
60		.DISPLAY	"RELEASE sonrasi ProcessID degeri: "&#ProcessID	END	ProcessID degeri yazdirilir ve surec sonlandirilir.

Messages

ProcessID degeri: 8
CLAIM sonrasi ProcessID degeri: 7
RELEASE sonrasi ProcessID degeri: 8

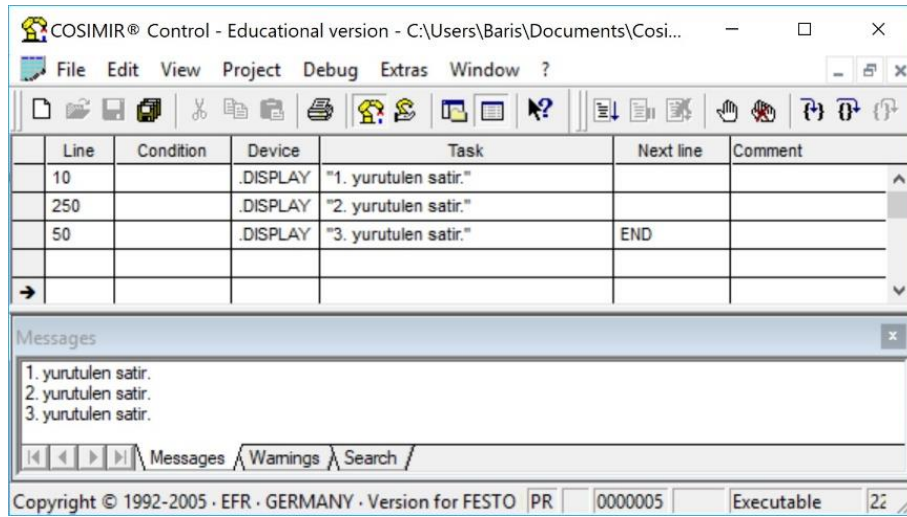
Şekil 3-119. “.CLAIM” ve “.RELEASE” fonksiyonları kullanımına örnek süreç planı.

Süreç Planları Çalışma Mantığı ve Süreç Planı Oluşturma

Süreç planlarında satırlar en üstten başlayarak ardı sıra işlenir. Her satır başına özel bir numara veya etiket metni (metin ve sayı birlikte de olabilir, örneğin “işlem1”) konur. Bu aynı satır numarasına veya etiketine sahip başka bir satır olamayacağı anlamına gelir. Satırlara numara etiketleri 10’ar 10’ar veya 100’er 100’er artacak şekilde verilir. Örneğin 120 ile 130 numaralı satırlar arasına bir satır eklenecekse, buna 125 numarası verilebilir. Böylelikle sonradan bir satırın üstüne veya altına yeni bir satır eklendiğinde süreç planındaki tüm satırlara yeniden numara verilmesine gerek kalmaz. Ara bir numara verilerek süreç düzenlenir.

Satır sonlarındaki “Next Line” hücrelerine bir satır numarası veya etiketi girilmişse, program ilgili satırdaki görevi işlendikten sonra, etiketin olduğu satıra dallanır. Genel

anlamda satırlara verilen numaranın artan bir şekilde olması, sürecin çalıştırılması için zorunluluk değildir. Süreç planı yürütülürken, eğer “Next Line” hücreğine bir satır numarası girilmemişse, satır numarasına bakılmaksızın bir alttaki satıra geçilir. Şekil 3-120’deki örnek incelendiğinde, 10 numaralı satırdan sonra 250 numaralı satır gelmiş, onun da altında 50 numaralı satır eklenmiştir. Süreç planı başlatıldığında, “Mesaj Penceresi”nde de görüldüğü gibi satır numaralarına bakılmaksızın; sırasıyla 10, 250 ve 50 numaralı satırlar yürütülmüştür. Özetle “Line” hücreesindeki numaraların artan bir şekilde olması, bizim süreç planını anlayabilmemiz açısından önemlidir. COSIMIR Factory Control yazılımı süreç planını satır satır işletirken, sadece satırların alt alta olması ve eğer varsa “Next Line” hücreсі içerisindeki satır etiketi göz önüne alır.



Şekil 3-120. Süreç planı içerisindeki satır etiketlerinin sürecin sıralamasına etkisi.

Süreç planlarında her satırda yürütülen görevin (task) sonucu, bir sonraki satırın koşul şartıdır. Böylelikle bir sonraki satırda yazılmış olan koşullara göre ilgili satıra dallanma gerçekleşir. Eğer bir sonraki satıra koşul yazılmamışsa, bir önceki satırdan geri dönen değerin önemi yoktur. Bu durumda geri dönüş değeri ne olursa olsun, tek koşullu satır işletilir.

Şekil 3-121’deki örnekte; 100 numaralı satırda “.CALC” fonksiyonu kullanılarak “\$ErrCode” isimli sistem değişkeninin içeriği çağırılmıştır. Bu değişken içeriği 110 numaralı satırın koşuludur. Değişken içeriği “ERRDIVZERO”, “ERRSEMA”, “ERRCLDCNCL”, “ERRCALL” koşullarından hangisine uyuyorsa, o satır yürütülür. Hiçbirine uymuyorsa yine 110 numaralı satır ile ilişkili “.ERROR” fonksiyonu yürütülerek kullanıcıya hata mesajı görüntülenir.

Line	Condition	Device	Task	Next line	Comment
100		.CALC	\$ErrCode		
110	"ERRDIVZERO"	.DISPLAY	"Sifira bolme hatasi...!"	120	
	"ERRSEMA"	.DISPLAY	"Hatali semafor...!"	END	
	"ERRCLDCNCL"	.DISPLAY	"Cagirlmis olan surec plani iptal ed	130	
	"ERRCALL"	.DISPLAY	"Surec plani cagirma hatasi...!"	140	
		.ERROR		END	

Şekil 3-121. Satırlarda yürütülen görevlerin geri dönüş değerlerine göre dallanma örneği.

Bu örnekten de anlaşılacağı üzere bir satırda yürütülen görevlerin sonucu; ya boolean tipi (0 veya 1), ya bir metin ya da sayısal (tam sayı, ondalıklı) değer olabilir. Eğer ki Şekil 3-121'deki örnekteki 110 numaralı satırda olduğu gibi, ardışık satırda koşul ilişkisi tanımlanmışsa, program bir önceki satırdan dönen değer ile ilgili satıra dallanır.

COSIMIR Factory Control yazılımında oluşturulan süreç planları, paralel olarak çalışabilen ve olay tetiklemeli yapıda oluşturulabilir. Olay tetiklemeliden kasıt, süreç planları içerisindeki görevler ve sistem fonksiyonları; başlatılmış bir olayın durumunu takip için, sürekli bir döngü içinde sorgulama (pooling) yapmazlar. Başlamış olan olayın sonuçlanmasını beklerler. Olay sonuçlandığında üretilen mesajı (geri dönüş değeri) alırlar. Tüm dahili sistem fonksiyonları bu şekilde çalışır.

Parametrelili Bir Süreç Planını Çağırma ve Parametre Değerlerini Okuma

Tüm programlama dillerinde bulunan fonksiyon / prosedür mantığında olduğu gibi, süreç planları başlatılırken, parametre değişkenleri üzerinden çeşitli ön değerler gönderilebilir. Bu ön değerler kullanılarak süreç planı içerisindeki görevler yürütülür ve sonuçlar üretilir. İstenirse üretilen sonuçlar da süreç planının çağrıldığı yere geri dönüş değeri olarak gönderilebilir. Bu durum programlama dillerinde kullanılan fonksiyon mantığındaki geri dönüş değeri ile aynıdır.

Bir süreç planı parametre kullanılarak çağrıldığında, parametrelerin değerlerine "%Par" yerel değişkeni üzerinden erişilebilir. "%Par" değişkeni yapısal bir değişkendir ve alt dallarına "." işareti konulduktan sonra verilen indis numaraları üzerinden erişilebilir. Örneğin bir süreç planı 2 farklı parametre ile çağırılmış olsun. Süreç planının içerisinde parametre değerlerine aşağıdaki şekilde erişilebilir:

- "%Par.0": süreç planı çağrılırken gönderilen parametre sayısını saklar.
- "%Par.1": süreç planı çağrılırken gönderilen birinci parametre değerini saklar.

- “%Par.2”: süreç planı çağrılırken gönderilen ikinci parametre değerini saklar.

Şekil 3-122.’de “AnaSurec” içerisinde “CALL” fonksiyonu ile çağrılan “AltSurec1” isimli süreç planına ait örnek görülmektedir. Örnekte, “AltSurec1” isimli süreç planında 50 numaralı satırda geri dönen değer, “AnaSurec” deki 15 numaralı satırdaki “%GeriDeger” isimli yerel değişken ile okunmaktadır. 25 numaralı satırda da okunan değer “Mesaj Penceresi”nde görüntülenmekte ve plan sonlanmaktadır.

The screenshot shows the AnaSurec software interface. The top window displays the process plan for 'AltSurec1' with the following data:

Line	Condition	Device	Task	Next line	Comment
5		.CALC	%Deger=7		AltSurec1 e gonderilecek deger ayarlanir.
15		.CALL	%GeriDeger=AltSurec1(%Deger, "Marmara", "Universitesi")		AltSurec1 3 adet parametre ile cagilir.
25		.DISPLAY	"Geri donen deger: "&%GeriDeger	END	AltSurec1 den geri donen deger okunur.

The bottom window, titled 'Messages', displays the following text:

```

Checking process plan 'AltSurec1'...
Process plan check finished: 0 Errors, 0 Warning(s).
Parametre sayisi: 3
Parametre 1 degeri: 7
Parametre 2 degeri: Marmara
Parametre 3 degeri: Universitesi
Geri donen deger: 8
  
```

Şekil 3-122. “.CALL” fonksiyonu ile çağrılan bir süreç planından dönen değer okunması.

“AltSurec1” den geri dönen değeri okumak için Şekil 3-123’deki gibi \$RES sistem değişkeni de kullanılabilir.

The screenshot shows the AnaSurec software interface. The top window displays the process plan for 'AltSurec1' with the following data:

Line	Condition	Device	Task	Next line
5		.CALC	%Deger=7	
15		.CALL	AltSurec1(%Deger, "Marmara", "Universitesi")	
25		.DISPLAY	"Geri donen deger: "&\$RES	END

The text "Geri donen deger: "&\$RES" in the task column of line 25 is circled in red.

Şekil 3-123. “.CALL” fonksiyonu ile çağrılan bir süreç planından dönen değerin \$RES sistem değişkeni ile okunması.

COSIMIR Factory Control programında süreç planı oluştururken bazı temel kalıpların, planın oluşturulmasında büyük kolaylık sağlar. Parametre kullanılarak çağrılacak bir süreç planına ait örnek Şekil 3-124’de verilmiştir. Şekildeki örnekte, sürecin nasıl yürütüleceği ile ilgili açıklamalar verilmiştir.

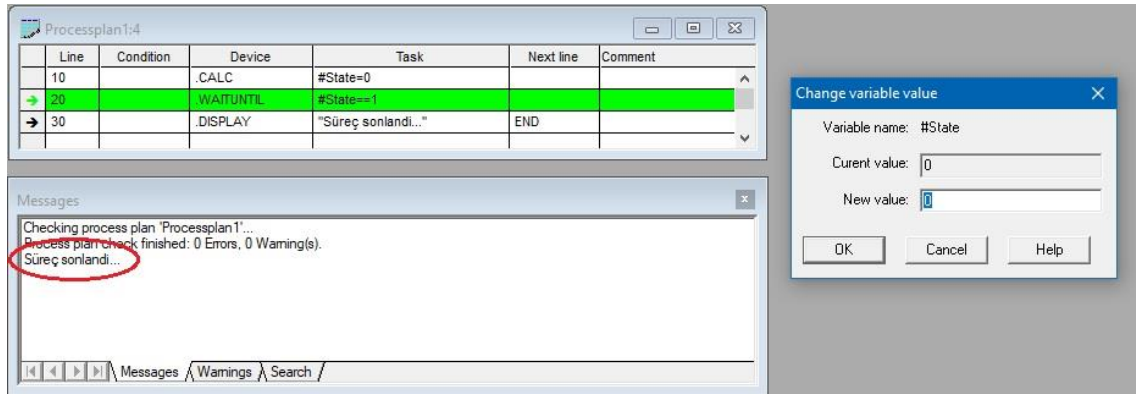
; AltSurec1					
; Bu örnek uygulamada bir süreç planına yollanan parametreler nasıl kontrol ediliş kullanilacağı gösterilmiştir					
; Bu surec planinin çalışabilmesi için, plan çağırılırken 2 adet parametrenin gönderilmesi gerekmektedir.					
; Gönderilecek parametreler: süreç adı ve süreç tipidir.					
;					
; Parametrelerin kontrol edildiği ve okundugu bölüm.					
BASLA		.CALC	%Par.0		Kac adet parametre gönderildiği bilgisi okunur.
10		DIALOG	Show("Bu surec planina en az 2 paramtre girilmelidir!")	HATA1	2'den farklı sayıda parametre gönderilirse, mesaj görüntülenir ve HATA1'e daller.
	2	.CALC	%SurecAdi = %Par.1		2 adet parametre gönderilirse, parametreler okunarak degiskenlere aktarılır.
20		.CALC	%SurecTipi = %Par.2		
;					
; Surec planina ait programin yazıldığı bölümü...					
;					
; Surec planinin sonlandırıldığı bölüm.					
HATA1		.CALC	%GerDonusDegeri = -1	SON	Geri donus degeri atanır.
HATA2		.CALC	%GerDonusDegeri = 1		
SON		.CALC	%GerDonusDegeri	END	

Şekil 3-124. Bir süreç planına 2 adet parametre gönderilmesine yönelik kalıp örneği.

Olay Tetiklemeli Süreç Planı Oluşturma

Yazılan süreç planlarında olay tetiklemeli bir yapı oluşturabilmek için “.WAITCHANGE” ve “.WAITUNTIL” gibi sistem fonksiyonları kullanılmalıdır. Bu sistem fonksiyonları Bölüm 3.1.4’de anlatılan Sıralı Programlama Yaklaşımı’nda adımlar arası geçiş elemanı olarak kullanılır. Geçiş elemanı (G0, G20 vb.) tetiklenene kadar, program bir önceki adımda bekler. Benzer şekilde bu fonksiyonlar paralel dallanmış süreçlerin birleşiminin kontrolünde de kullanılır. Sürecini erken tamamlayan dal, paralel olan daldaki sürecin tamamlanmasını bekler. Böylelikle daha temiz ve anlaşılır bir program yapısı elde edilirken, işlemlerin yürütülmesi için daha az zaman harcanır.

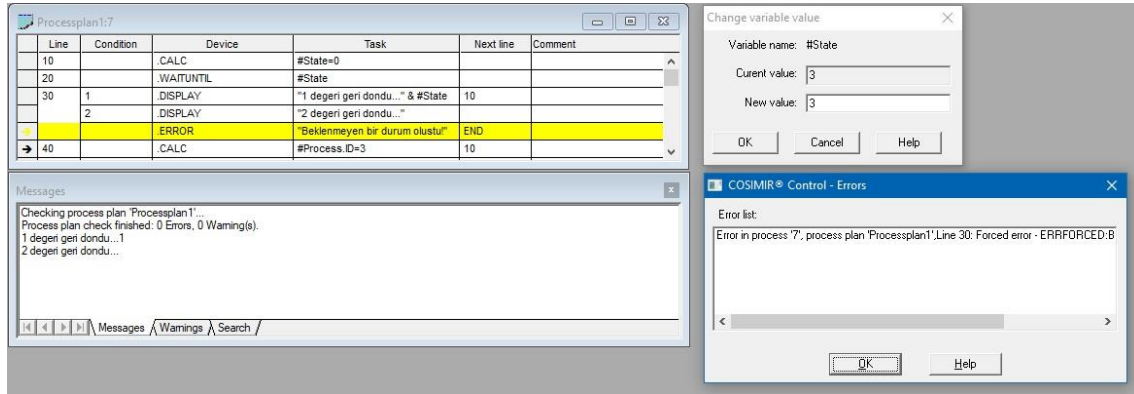
“.WAITUNTIL” fonksiyonuna örnek olarak Şekil 3-125’deki süreç planı oluşturulur ve başlatılır. “Variables” sekmesinden “#State” değişkeni içeriği 1 olarak değiştirilene kadar süreç plan 20 numaralı satırda bekleyecektir. Değişken değeri değiştiğinde ise “Mesaj Penceresi”nde “Süreç sonlandı...” yazacak ve süreç sonlanacaktır.



Şekil 3-125. “.WAITUNTIL” sistem fonksiyonu kullanımına örnek süreç planı.

“WAITCHANGE” fonksiyonuna örnek olarak Şekil 3-126’deki süreç planı oluşturulur ve başlatılır. Plana göre 10 etiketli satırda “#State” değişkeni değeri sıfırlandıktan sonra

20 etiketli satıra “#State” değişkeni değerinde değişim olması beklenir. “Variables” penceresinden “#State” değişkeni değeri sırasıyla 1 ve 2 olarak değiştirildiğinde, “Mesaj Penceresi”ndeki mesajlar görüntülenerek başa dönülür. Değişken değeri 8 olarak değiştirildiğinde ise süreç planının 30 etiketli satırı gereği hata mesajı penceresi görüntülenir.

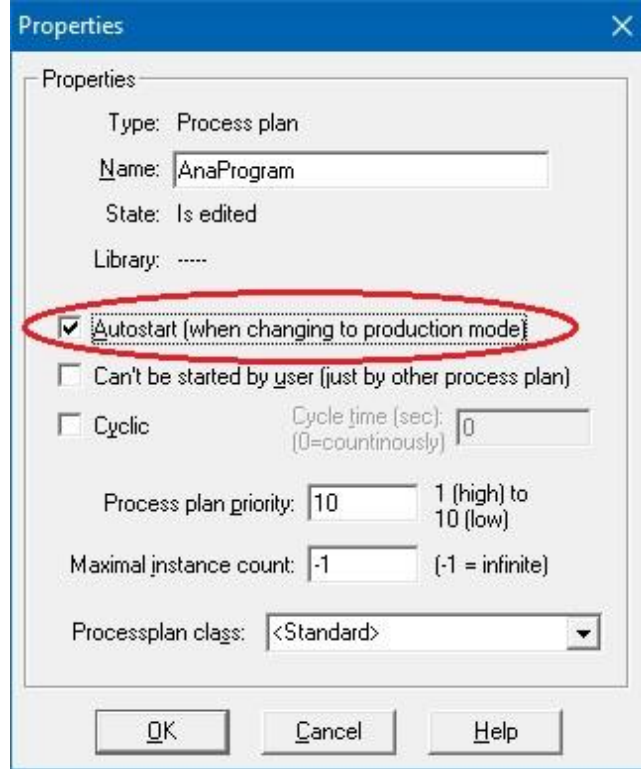


Şekil 3-126. “WAITCHANGE” sistem fonksiyonu kullanımına örnek süreç planı.

Paralel Çalışan Süreç Planı Oluşturma

Paralel işlem yapabilme özelliğinden kasıt, COSIMIR Factory Control yazılımının birçok süreci aynı anda ve paralel olarak yürütebilme özelliğidir. Bu sayede ürün odaklı programlama gerçekleştirilebilir. Ürüne ait farklı imalat süreçleri aynı anda yürütülebilir ve kontrol edilebilir.

Yazılımın paralel çalışabilme özelliği iki farklı şekilde kullanılabilir. Farklı süreç planları oluşturularak aynı anda çalışmaları sağlanabilir. Süreç planları birbirleri içerisinde “START” fonksiyonuyla çağrılabilir. Bunun dışında Proje Penceresi üzerinden süreç planı seçilerek sağ tuş özellikler (properties) tıklanıldığında ekran gelen Şekil 3-127’deki pencereden “Autostart (when changing to production mode)” seçilebilir. Bu durumda üretim moduna geçildiğinde süreç planı otomatik olarak başlatılır. Bir diğer seçenek ise Proje Penceresi veya Bölüm 3.1.8.4.5’de anlatılan görsel arayüz üzerinden, kullanıcının farklı süreç planlarını aynı zamanda başlatmasıdır.



Şekil 3-127. Süreç planının otomatik çalıştırılması ayarları.

Bir süreç planı bünyesinde ise “.SPLIT” ve “.MERGE” fonksiyonları kullanılarak paralel dallar oluşturulup birleştirilebilir. Şekil 3-128.’deki örnekte bu iki fonksiyonun kullanımı görülmekte olup, örnek uygulamadaki her satırın açıklaması “Comment” sütununda verilmiştir. Genel olarak bu örnekte paletler istasyona çağrıldıktan sonra süreç akışı iki ayrı dala ayrılmakta, her daldaki görevler birbirlerinden bağımsız şekilde yürütülmekte ve en sonunda dallar birleştirilmektedir.

Line	Condition	Device	Task	Next line	Comment
10		Transport	Require(1)		Istasyona 1 numaralı palet çağrılır.
20		Transport	Require(2)		Istasyona 2 numaralı palet çağrılır.
30		.SPLIT	100,200	300	Süreç planında paralel dallar oluşturulur.
; *** Birinci süreç dali başlangıcı ***					
100		Transport	Pos_Fanuc(1)		Palet üzerindeki parça Pos_Fanuc(1) konumuna taşınır.
110		Fanuc	Cranfield		Fanuc cihazı parça üzerinde Cranfield görevini yürütür.
120		Transport	Exit(1)	300	İşlenmiş parça Exit(1) konumuna taşınır. 300 numaralı satıra dallanır.
; *** İkinci süreç dali başlangıcı ***					
200		Transport	Pos_Adept(2)		Palet üzerindeki parça Pos_Adept(2) konumuna taşınır.
210		Adept	Assemble(#Product_ID)		Adept cihazı Assemble görevini yürütür.
220		Transport	Exit(2)		İşlenmiş parça Exit(1) konumuna taşınır. Bir alt satıra (300) dallanır.
; *** Ayrılan dalların birleşmesi ***					
300		.MERGE			Paralel dallar birleştirilir.
310		Transport	Release(1)		Palet 1 serbest bırakılır.
320		Transport	Release(2)	END	Palet 2 serbest bırakılır ve süreç sonlanır.

Şekil 3-128. “.SPLIT” ve “.MERGE” sistem fonksiyonları kullanımına örnek süreç planı.

3.1.8.4.5. İmalat Süreci Projeleri İçin Görsel Arayüz Oluşturulması

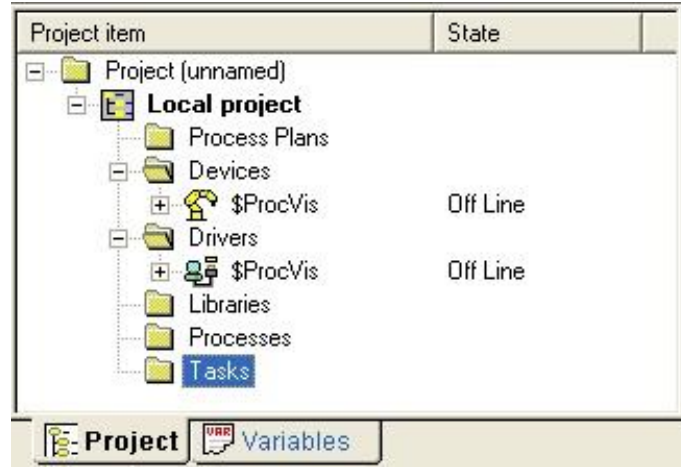
COSIMIR Factory Control yazılımında imalat süreçlerini görselleştirmek için “PRCVIS32.LDV” sürücüsü kullanılır. Böylelikle her proje uygulamasına özel kullanıcı arayüzleri oluşturulabilir.

Süreç planı için oluşturulan görsel arayüz ile kontrol sistemi arasındaki iletişim, değişkenler üzerinden sağlanır. Tanımlanmış değişkenlerdeki değişimler görsel arayüzü etkilediği gibi, görsel arayüzden verilen komutlar da değişkenlerin içeriğini değiştirir. Bu mekanizma sayesinde denetleyici, sürücüler, üretim süreçleri ve görsel arayüz arasında veri alışverişi ile etkileşim sağlar.

Yeni Bir Görsel Arayüz Oluşturma

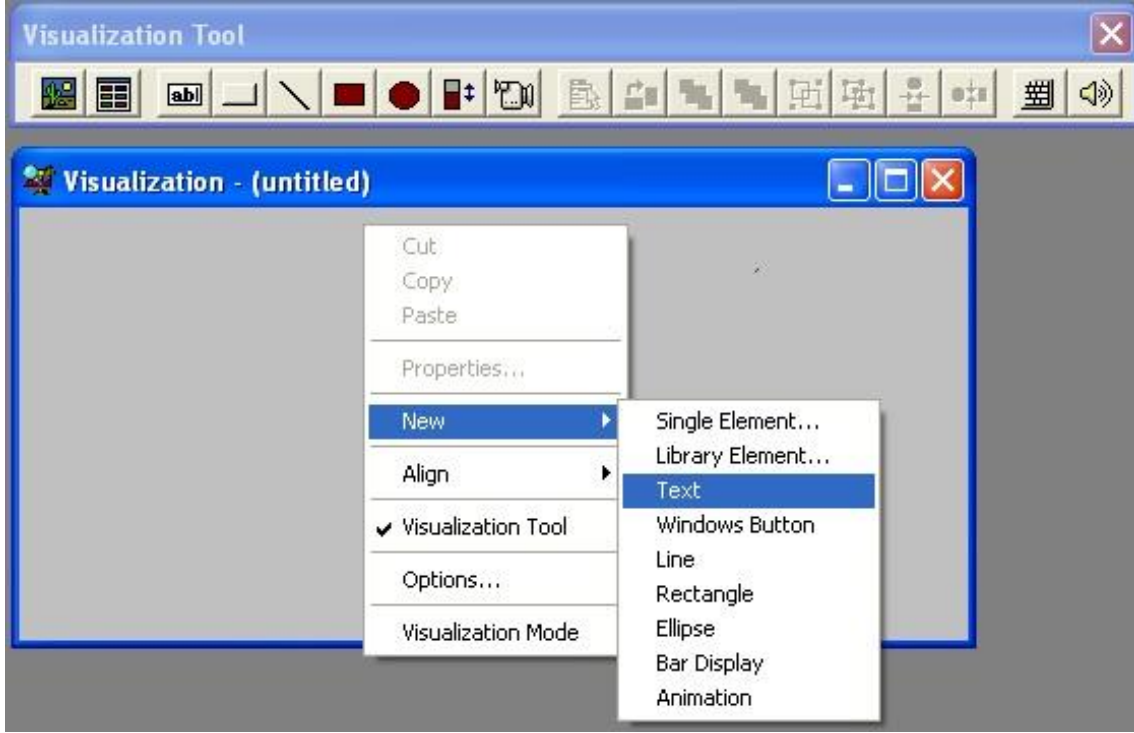
COSIMIR Factory Control yazılımında herhangi bir süreç planı için örnek bir görsel arayüz oluşturmak istenildiğinde aşağıdaki adımlar takip edilir:

- Yeni bir proje oluşturulur ve “DRIVERS/PROCVIS/PrcVis32.ldv” sürücüsü eklenir. Şekil 3-129’de görüldüğü gibi yazılım otomatik olarak “\$ProcVis” isimli bir sistem cihazı oluşturur. Bu cihaz görsel arayüzdeki elemanların kontrolü için görevlere sahiptir.



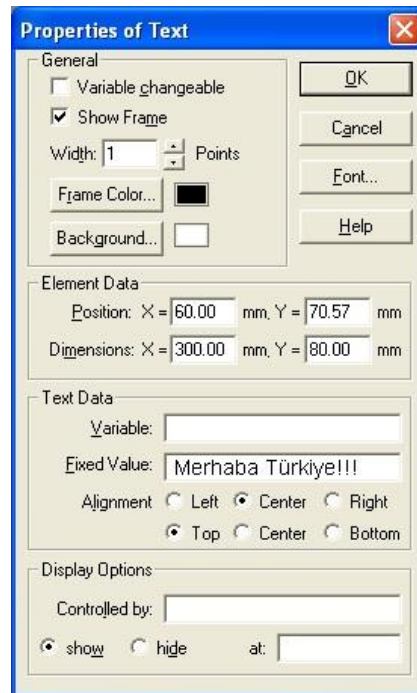
Şekil 3-129. Süreç planına görsel arayüz için “\$ProcVis” sürücüsünün eklenmesi.

- Yeni bir görsel arayüz penceresi oluşturabilmek için menü çubuğu üzerindeki “File” menüsünden “New/Visualization” seçilir. Ekranı gelen boş pencere üzerinde farenin sağ tuşuna basıldığında ŞEKİL deki gibi görüntülenen menüden görsel elemanlar seçilip pencereye eklenebilir. Menüden “New/Text” seçilerek bir metin kutusu pencereye eklenir.



Şekil 3-130. Boş görsel arayüze metin kutusu eklenmesi.

- Fare kullanılarak kutunun boyutları ve konumu değiştirilebilir. Metin kutusu seçiliyken fare sağ tuş yapılarak “Properties” tıklanır. Ekranda metin kutusuna ait değiştirilebilir tüm özellikleri bulunduran pencere Şekil 3-131.’deki gibi görüntülenir.



Şekil 3-131. Metin kutusuna ait özellikler penceresi görünümü.

- Metin kutusuna ait özellikler penceresindeki “Fixed Value” kutusuna “Merhaba Türkiye!!!” yazılır ve “OK” tıklanır. Bu durumda metin kutusunun içinde Şekil 3-132’de görüldüğü gibi “Merhaba Türkiye!!!” yazılı olacaktır. Bu yazı ne kullanıcı ne de bir süreç planı tarafından değiştirilemez durumdadır.



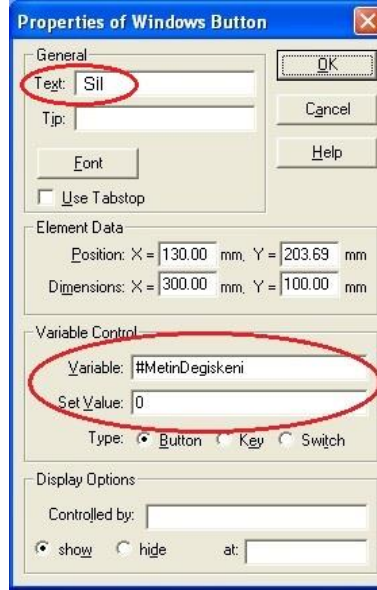
Şekil 3-132. Görsel arayüz üzerinde metin kutusu görüntüsü.

- Özellikler penceresi yeniden açılır ve “General” bölümündeki “Variable changeable” seçilir. Pencerenin “Text Data” bölümündeki “Variable” bölümüne “#MetinDegiskeni” yazılır. Böylelikle metin kutusu her zaman “#MetinDegiskeni” isimli değişkenin içeriğini görüntüleyecektir. Kullanıcı metin kutusunun değerini değiştirdiği zaman değişkenin içeriği de değişecektir.
- Menü çubuğu üzerindeki “Extras/Process Visualization/Visualization Mode” veya fotoğraf makinası şeklindeki butona basıldığında, pencerenin aktif hali görüntülenir. Bu durumda metin kutusunun içeriği değiştirildiğinde “Variables” sekmesinde “Global process plan variables” bölümünde Şekil 3-133’deki gibi görüntülenen “#MetinDegiskeni” isimli değişkenin içeriği de değişir.



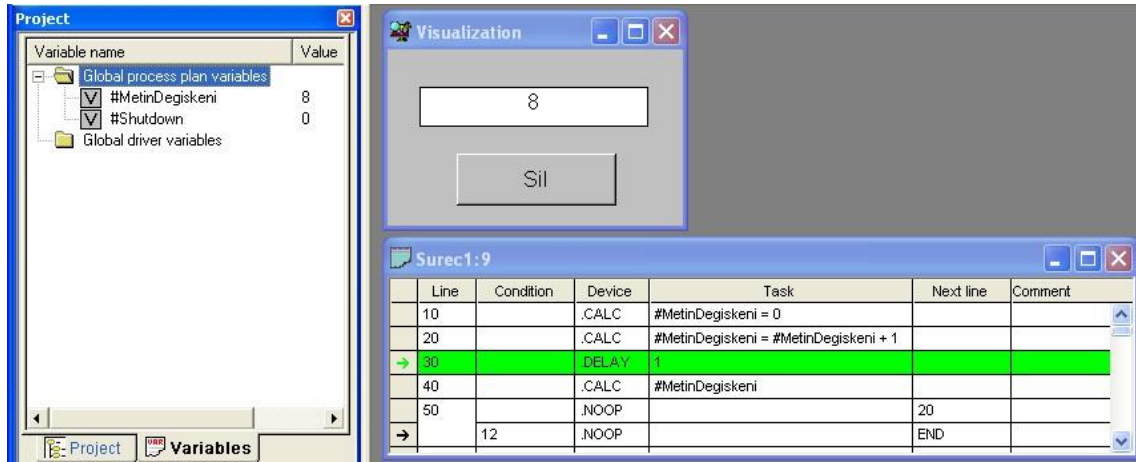
Şekil 3-133. Görsel arayüz üzerindeki metin kutusu içeriği ve değişken içeriği görüntüsü.

- “Extras/Process Visualization/Configuration Mode” seçilerek tekrar düzenleme moduna geçilir. “New/Windows Button” seçilerek mevcut pencereye buton eklenir. Bunun üzerine gelinerek farenin sağ tuşuna basılır ve “Properties” penceresi açılır. Pencere içeriği Şekil 3-134’daki gibi doldurulur. Buradaki ayarlara göre butona basıldığında “#MetinDegiskeni” isimli değişkenin içeriğine “0” değeri atanır.



Şekil 3-134. Butona ait özellikler penceresi görünümü.

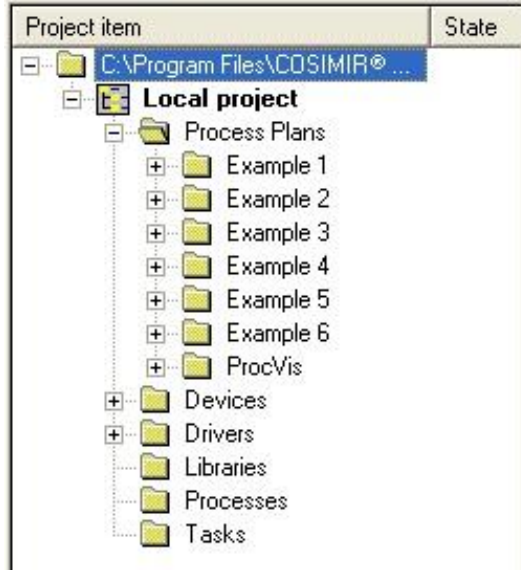
- Değişkenin içeriği bir süreç planı içerisinde de değiştirilebilir. Bunun için Şekil 3-135’de görülen süreç planı oluşturulur. Proje üretim moduna alınır ve süreç planı başlatılır. Metin kutusu içindeki değerin her saniye 1 arttığı, “Sil” butonuna basıldığında kutu içerisindeki değerin sıfırlandığı, #MetinDegiskeni isimli değişken değeri 12’ye ulaştığında sürecin sonlandığı görülecektir.



Şekil 3-135. Süreç planı içerisinde değişken değeri değiştirmeye yönelik örnek.

Görsel Arayüz Örnekleri

COSIMIR Factory Control yazılımında süreç planları için önceden hazırlanmış örnek görsel arayüz projelerine “File/Open/...Program Files/Demos/Procvis/Example/Example.lpj” dosyası açılarak ulaşılabilir. Örnek proje içerisinde süreç planları ve değişkenler için kullanılacak görsel arayüzler ve birçok arayüz bileşenlerini içeren altı farklı uygulama bulunmaktadır.



Şekil 3-136. “Example.lpj” isimli örnek proje açıldığında görüntülenen uygulamalar.

COSIMIR üretim moduna alındığında örnek uygulamaların seçim ekranının otomatik olarak görüntülenebilmesi için, Proje Penceresi içinden “Process Plans/ProcVis/ProcVis Examples” isimli süreç planı açılarak planda Şekil 3-137’deki gibi bir değişiklik yapılır.

ProcVis Examples					
Line	Condition	Device	Task	Next line	Comment
; Example Control					
START		.NOOP			
INIT		.CALC	#ProcVis.Auto = 1		
SHOW		\$ProcVis	OpenVisWnd("EXP00-01.LVC")	WAITCHANGE	
;					
SAVE		.CALC	#ProcVis.ExampleAlt = #ProcVis.Example		
; Wait for button pressed:					
WAITCHANGE		.WAITCHANGE	#ProcVis.Button		
10	"Stop"	.KILL	#ProcVis.AllPID		
		.NOOP			
20		.CALC	#ProcVis.Example = #ProcVis.Button		
30		.CALC	#ProcVis.ExampleAlt == #ProcVis.Example		
40	1	.NOOP		WAITCHANGE	
		.CLAIM	#ProcVis.LoadVisiSema		
; Exit actual example:					
50		.KILL	#ProcVis.BeginPID		
60		.CALC	#ProcVis.ExampleAlt		
END_OLD		.NOOP			
	"Example 1"	.CALL	EndExample_1()		
	"Example 2"	.CALL	EndExample_2()		
	"Example 3"	.CALL	EndExample_3()		
	"Example 4"	.CALL	EndExample_4()		
	"Example 5"	.CALL	EndExample_5()		
	"Example 6"	.CALL	EndExample_6()		

Şekil 3-137. “ProcVis Examples” isimli süreç planı ve yapılan değişikliklerin görüntüsü.

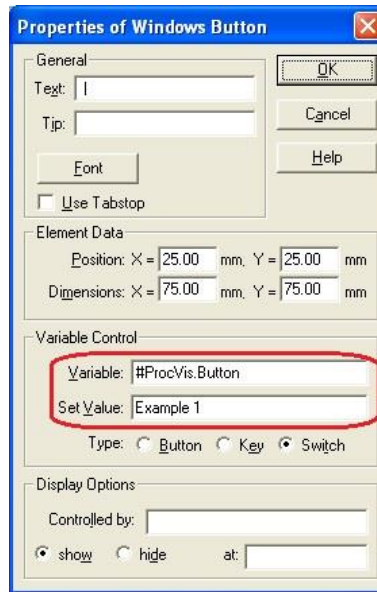
Değişiklik kaydedilir ve süreç planı kontrol (check) edilir. COSIMIR üretim moduna alındığında ekranda otomatik olarak Şekil 3-138’deki “Control” penceresi görüntülenir.

Pencere üzerinden istenilen örnek çalıştırılarak görsel arayüzlerde kullanılan elemanların nasıl çalıştıkları görüntülenir.



Şekil 3-138. Örnek uygulamaların çalıştırılabilmesini sağlayan kontrol penceresi.

“ProcVis Examples” isimli süreç planı otomatik başlama (autostart) özelliğine sahiptir. Şekil 3-137’de kısmen görüntülenmekte olan süreç planı, buton gibi kullanıcı girişlerinin çalışmasını göstermeye yöneliktir. “WAITCHANGE” etiketli satırda süreç plan “#Procvis.Button” değişkenin değeri değişene kadar bekler. “Control” penceresindeki “I” numaralı butonun özellikleri Şekil 3-139’de görülmektedir. Buna göre butona basılırsa “#Procvis.Button” değişkenine “Example 1” değeri atanır. Bu durumda “WAITCHANGE” durumu gerçekleştiği için süreç kaldığı yerden işlemeye devam eder. Bu satıra kesme noktası (breakpoint) eklenerek bu süreç planının nasıl işlediği daha iyi öğrenilebilir.



Şekil 3-139. “Control” penceresi üzerindeki “I” butonuna ait özellikler.

“I” butonuna basıldıktan sonra “ProcVis Examples” isimli süreç planı adım adım yürütüldüğünde, Şekil 3-140’daki gibi 80 numaralı satıra gelir. Bu satırda “Example_1()” isimli süreç planı başlatılır.

ProcVis Examples:326					
Line	Condition	Device	Task	Next line	Comment
70		.RELEASE	#ProcVis.LoadVisiSema		
; Start new example:					
BEGIN_NEW		.CALC	#ProcVis.Example		
80		.NOOP		SAVE	
	"All"	.START	All()	100	
	"Example 1"	.START	Example_1()		
	"Example 2"	.START	Example_2()		
	"Example 3"	.START	Example_3()		
	"Example 4"	.START	Example_4()		
	"Example 5"	.START	Example_5()		
	"Example 6"	.START	Example_6()		
90		.CALC	#ProcVis.BeginPID = \$RES	SAVE	

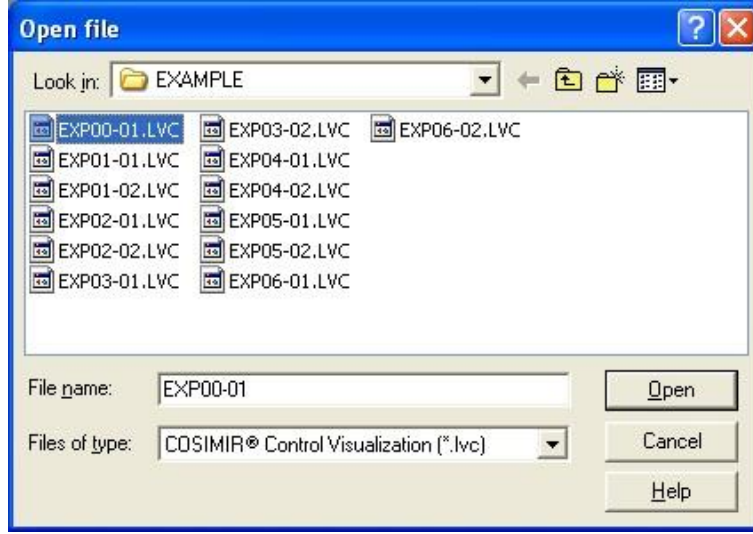
Şekil 3-140. “ProcVis Examples” isimli süreç planı seçim bölümü.

Şekil 3-141’de görülmekte olan “Example_1” isimli süreç planı, “\$ProcVis” cihazına ait “OpenVisWnd” isimli görevin nasıl kullanılacağına yönelik bir örnektir.

Example_1					
Line	Condition	Device	Task	Next line	Comment
; Call Example 1.					
START		.NOOP			
;					
INIT		.CALC	#Text.EXP01.1 = ""		
10		.CALC	#EXP01.Picture.Logo = 1		
20		.CALC	#EXP01.Picture.Angle = 0		
30		.CALC	#EXP01.Picture.XPos = 125		
40		.CALC	#EXP01.Picture.YPos = 275		
; Open visualization window of Example 1:					
LOAD_VISI		.CLAIM	#ProcVis.LoadVisiSema		
100		\$ProcVis	#EXP01.VisiWnd01 = OpenVisWnd("EXP01-01.LVC")		
110		\$ProcVis	ViewVisWnd(#EXP01.VisiWnd01, -1, -1, -1, -1, -1, 0)		
120		\$ProcVis	#EXP01.VisiWnd02 = OpenVisWnd("EXP01-02.LVC")		
130		\$ProcVis	ViewVisWnd(#EXP01.VisiWnd02, -1, -1, -1, -1, -1, 0)		
140		.START	Variable_1()		
150		.CALC	#EXP01.VarPID = \$RES		
160		.RELEASE	#ProcVis.LoadVisiSema		
170		.CALL	TextOut("You can control pictures in visibility, position and orientation through variable values.", "EXP01.1", 0)		
;					
AUTO		.NOOP			END

Şekil 3-141. “Example_1” isimli süreç planı içeriği.

Her örneğe ait görsel arayüzde bulunan nesnelerin nasıl kullanılacağını anlamak için, arayüz tasarım dosyalarının açılması gerekmektedir. Şekil 3-142’de görülen “File/Open” penceresinde dosya tipi (files of type) “COSIMIR Factory Control Visualization (*.lvc)” seçildiğinde, “...Program Files/Demos/Procvis/Example” klasörü içerisinde, örneklere ait arayüz tasarım dosyaları görüntülenir. Buradan istenilen tasarım seçilerek, tasarım penceresi üzerindeki kontrol nesnelerinin nasıl kullanıldığı öğrenilebilir. Daha detaylı bilgiye Yardım menüsünden ulaşılabilir.



Şekil 3-142. Örnek uygulamalara ait görsel arayüz tasarım dosyalarının açılması.

3.2. Hazırlanan Deney Föyleri

FMS100, bir fabrika otomasyon sistemine ait birçok bileşeni içeren oldukça kapsamlı bir sistemdir. Bu nedenle FMS100 bünyesindeki imalat ve otomasyon teknolojilerine yönelik birçok deney yapılabilir. FMS100 bünyesinde yapılabileceğini öngördüğümüz deneylerin kategorilere ayrılmış listesi aşağıdaki gibidir:

- CNC Tezgahı Kullanımı ve Programlama
 - CNC Freze Tezgahı Kullanımı ve Program Yazma/Çalıştırma
 - CNC Torna Tezgahı Kullanımı ve Program Yazma/Çalıştırma
- Endüstriyel Robot Kullanımı ve Programlama Uygulamaları
 - RV-E2 Model Endüstriyel Robot Kullanımı (Lehim, Montaj, Görüntü İşleme)
 - RV-E2 Model Endüstriyel Robot Programlama Uygulaması: COSIMIR Industrial Kullanımı ve XYZABC Eksenlerinde Hareket (Lehim, Montaj, Görüntü İşleme)
 - RV-E2 Model Endüstriyel Robot Programlama Uygulaması: Bir Konumdan Bir Konuma Parça Taşıma (Lehim, Montaj, Görüntü İşleme)
 - RV-E2 Model Endüstriyel Robot Programlama Uygulaması: Sıralı Konumlardan Çok Konuma Parça Taşıma (Montaj)
 - RV-E2 Model Endüstriyel Robot Programlama Uygulaması: Konumlar Arası Ardışık Parça Taşıma (Lehim, Montaj, Görüntü İşleme)
 - RV-E2 Model Endüstriyel Robot Programlama Uygulaması: Sayısal Giriş Durumunun Okunması (Montaj)

- RV-2AJ Model Endüstriyel Robot Kullanımı (CNC-Çifti)
- RV-2AJ Model Endüstriyel Robot Programlama Uygulaması: COSIMIR Industrial Kullanımı ve XYZABC Eksenleri ile Doğrusal Kızak Üzerinde Hareket (CNC-Çifti)
- RV-2AJ Model Endüstriyel Robot Programlama Uygulaması: Bir Konumdan Bir Konuma Parça Taşıma (CNC-Çifti)
- RV-2AJ Model Endüstriyel Robot Programlama Uygulaması: Sıralı Konumlardan Çok Konuma Parça Taşıma (CNC-Çifti)
- RV-2AJ Model Endüstriyel Robot Programlama Uygulaması: CNC Freze Tezgahına Parça Yerleştirme (CNC Çifti)
- RV-2AJ Model Endüstriyel Robot Programlama Uygulaması: CNC Torna Tezgahına Parça Yükleme/Boşaltma (CNC Çifti)
- RV-2AJ Model Endüstriyel Robot Programlama Uygulaması: CNC Freze Tezgahına Parça Yükleme/Boşaltma (CNC Çifti)
- RV-2AJ Model Endüstriyel Robot Programlama Uygulaması: Sayısal Giriş Durumunun Okunması (CNC-Çifti)
- SCADA Programlama
 - COSIMIR Factory Yazılımı Kullanımı ve Geliştirme Ortamının Tanıtımı
 - COSIMIR Factory Yazılımında Yeni Bir Süreç Planı Oluşturma – Temel İşlevler
 - COSIMIR Factory Yazılımında Yeni Bir Süreç Planı Oluşturma – Paralel Programlama
 - COSIMIR Factory Yazılımında Görsel Arayüz Oluşturma
 - COSIMIR Factory Yazılımında Görsel Arayüz Üzerinden MovePalet Görevini Çalıştırma (Sırayla Tüm istasyonlar)

Tüm bu deneylere ait hazırlanacak föyler oldukça yer kaplayacak dosyalardır. Bu tez çalışmasından sonra ileride tüm bu deney uygulamalarını da içeren bir kitap çalışması yapılması da planlandığı için, yukarıda listelenen deney föylerinin hepsi bu dokümanda bulunmamaktadır. Ayrıca bu deney konularını kapsayan örnek uygulamaların birçoğu Bölüm 2.1 (Materyal) ve Bölüm 3 (BULGULAR VE TARTIŞMA)’da yer alan konu anlatımları sırasında örnek olarak verilmiştir. Bu nedenlerle, bu tez çalışmasında deney föylerinin biçimi ve içeriği hakkında bilgi vermesi bakımından birkaç örnek deney föyü sunulmuştur.

3.2.1. Deney Adı: CNC Freze Tezgahı Kullanımı ve Program Yazma/Çalıştırma (CNC-Çifti Hücresi)

Bu uygulamada G kodlarını kullanarak doğrudan bir CNC freze programı yazma ve Mill55 model CNC tezgahının kullanımı konularında bilgi ve deney sahibi olunması amaçlanmaktadır.

Deneyde FMS100 CNC-Çifti Hücresi ve hücrede bulunan Mill55 freze tezgahı kullanılmaktadır. Deneyde özetle, verilen bir teknik çizime göre NC kodları oluşturularak freze tezgahına yüklenecek ve ham parça freze tezgahına bağlanarak işlenecektir.

Bu uygulamada temel olarak;

- Teknik bir çizimden, temel G kodları kullanarak, NC program oluşturulması,
- Mill55 CNC freze tezgahının ve kontrol klavyesinin kullanımı,
- Tezgah eksenlerinin sıfırlanması,
- Kesici takımın eksenlerde serbestçe hareket ettirilmesi (JOG),
- Tezgaha NC programı yüklenmesi,
- NC programın seçimi ve içeriğinin görüntülenmesi/değiştirilmesi (EDIT),
- NC programın başlangıç satırının seçilmesi,
- Kesici takım ilerleme/dönme hız oranlarının (feedrate) değiştirilmesi,
- NC programının satır satır çalıştırılması,
- NC programının “kuru” çalıştırılması,
- NC programının çalıştırılması/durdurulması,
- Serbest kod çalıştırılması,
- Tezgaha parça bağlanması,
- NC programın başlangıç konumunun parça üzerinde belirlenmesi,
- Soğutma havasının açılması/kapanması

konularında bilgi ve deney sahibi olunması hedeflenmiştir.

3.2.1.1. Deney Baş Kontrol Listesi

Lütfen deneye başlamadan önce aşağıdaki listede belirtilen kontrolleri tek tek yapınız.

- 1) FMS100 sistemini besleyen ana kompresör aktif mi?

Evet: Bir sonraki adıma geçiniz.

Hayır: Bölümdeki laboratuvarlara basınçlı hava sağlayan ana kompresörü çalıştırınız.

- 2) Mill55 tezgahı açık mı? (Tezgah içindeki aydınlatma lambası yanıyor mu?)

Evet: Bir sonraki adıma geçiniz.

Hayır: Şekil 3-42’de görülmekte olan tezgah açma/kapama anahtarını 1 konumuna getiriniz.

- 3) Mill55 tezgahı FMS modu kapalı mı?

Evet: Bir sonraki adıma geçiniz.

Hayır: Şekil 3-42’de görülmekte olan FMS modu anahtarını 0 konumuna getiriniz.

- 4) Mill55 tezgahı bilgisayarı ve ekranı açık mı?

Evet: Bir sonraki adıma geçiniz.

Hayır: Bilgisayar ve ekranı çalıştırınız. Arka planı siyah renk olan “Winnc32” yazılımı ekranda görüntülenene kadar bekleyiniz.

- 5) “WinNC32” yazılımı çalıştı mı? (Bilgisayar açılınca otomatik çalışması gerekir.)

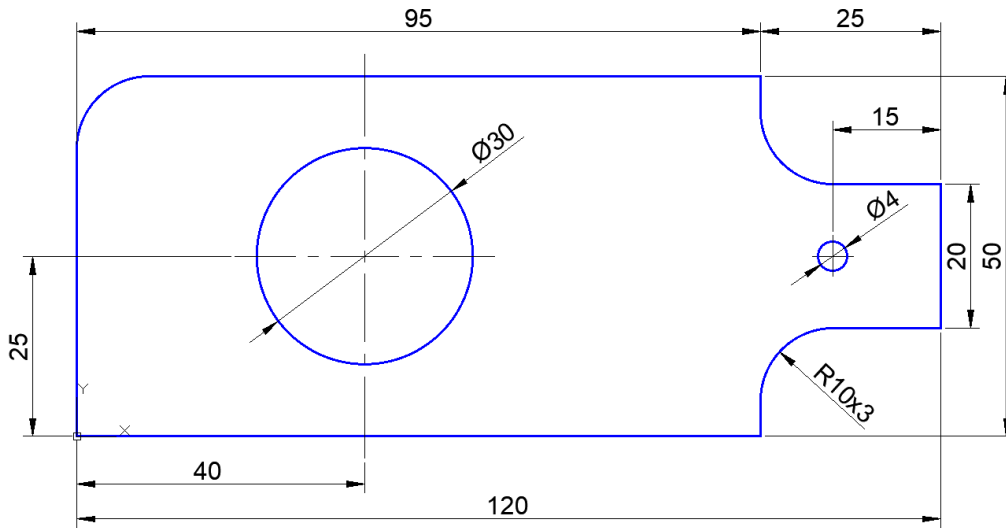
Evet: Bir sonraki adıma geçiniz.

Hayır: Bilgisayar masaüstündeki “WinNC32 Machine” simgesini tıklayarak yazılımı çalıştırınız.

3.2.1.2. Deney İşlem Adımları

Bu deney temel olarak ders notlarının “Bilgisayar Destekli Talaşlı İmalat” bölümünde anlatılan konuları kapsamaktadır. Lütfen öncelikle ders notlarının ilgili bölümünü okuyunuz.

- 1) Aşağıdaki Şekil 3-143’de yer alan teknik çizimi inceleyiniz. Ø6 mm çapında parmak freze ucu kullanılacak ve çizime göre dış taraf G42 ile telafi verilecek işlenecektir. Ø30 mm çapta görülen kısım ise G41 ile telafi verilecek içten boşaltılacaktır. Tüm işlemlerde derinlik (Z) -3mm alınacaktır. Bu şekli üretebilmek için gerekli NC programını kağıt üzerinde oluşturunuz. Oluşturduğunuz NC programını dersin öğretim elemanına gösteriniz.



Şekil 3-143. Mill55 freze tezgahında işlenecek parçaya ait teknik çizim.

- 2) Kontrol klavyesini kullanarak tezgah eksenlerini sıfırlayınız. Önce +Z eksenini sonra diğer eksenlerde sıfırlama işlemi yapınız.
- 3) Programlama klavyesini kullanarak “O88” isimli yeni bir program oluşturunuz.
- 4) Kağıt üzerinde hazırlamış olduğunuz NC programını klavyeyi kullanarak tezgaha yükleyiniz.
- 5) Mengeneye temas etmeyecek yükseklikte güvenli bir noktayı, program başlangıç noktası olarak belirleyiniz. Başlangıç noktasına ait makina koordinatlarını G56’ya kaydediniz.
- 6) Kontrol klavyesinden “Dry Run” tuşuna basarak tezgahı kuru çalışma tipine alınız. Programı başlatınız ve çalışmasını gözleyiniz. Programın çalışmasında bir aksaklık tespit ederseniz, programı düzeltip yeniden başlatınız.
- 7) Kuru çalışma sonuçları olumluysa tezgahın kapağını ve mengenesini açınız.
- 8) Ham parçayı mengeneye yerleştiriniz ve mengeneyi kapatarak parçayı sabitleyiniz.
- 9) Parça üzerinde NC programın başlangıç noktasını belirleyiniz ve makina koordinatlarını G56’ya kaydediniz.
- 10) İş mili devir hız oranını (override spindle) %100; iş mili ilerleme hız oranını (override feedrate) %50 yaparak programı çalıştırınız.
- 11) Programın işleme durumuna göre, gerekirse iş mili devir hızı oranı ile ilerleme hız oranlarını değiştiriniz.
- 12) Program bitince önce tezgahın kapısını sonra da mengeneyi açarak işlenmiş parçayı çıkartınız.
- 13) İşlediğiniz parçanın doğruluğunu kumpas ile ölçerek inceleyiniz. Sonrasında sorumlu öğretim elemanına işlediğiniz parçayı gösteriniz.

3.2.1.3. Deney Sonu Kontrol Listesi

Lütfen deney bitiminde aşağıdaki listede belirtilen kontrolleri tek tek yapınız.

- 1) Kesici uç sağlam mı? (Kırılmış veya zarar görmüş mü?)
Evet: Dersin öğretim elemanına mutlaka haber veriniz.
Hayır: Bir sonraki adıma geçiniz.
- 2) Kesici uç mengeneye çarpmayacak şekilde yukarıda mı duruyor?
Evet: Bir sonraki adıma geçiniz.
Hayır: Tezgahı JOG moduna alıp kesici takımı +Z yönünde hareket ettirerek mendeneden uzaklaştırınız.
- 3) Tezgahın içi çok kirli mi?
Evet: Tezgah içerisindeki talaşları temizleyiniz.
Hayır: Bir sonraki adıma geçiniz.
- 4) Tezgah kapısı kapalı mı?

Evet: Bir sonraki adıma geçiniz.

Hayır: Kontrol klavyesi üzerinden kapıyı kapatınız.

5) Tezgah bilgisayarı çalışıyor mu?

Evet: Önce kontrol klavyesi üzerinden “Reset” (Şekil 3-48/1) ve “Skip” (Şekil 3-48/3) tuşlarına aynı anda basarak “WinNC32” yazılımını kapatınız. Sonrasında bilgisayarı kapatınız.

Hayır: Bir sonraki adıma geçiniz.

6) Tezgah çalışıyor mu?

Evet: Tezgahın yan tarafındaki tezgah açma/kapama anahtarını 0 konumuna getirerek tezgahı kapatınız.

Hayır: Deney bitmiştir. Deney sonuçlarını içeren raporu oluşturarak dersin öğretim elemanına iletiniz.

3.2.1.4. Deney Sonuçları

Bu deney kapsamında öğrendiklerinizi aşağıda ayrılan bölüme sığacak şekilde yazınız.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3.2.2. Deney Adı: RV-E2 Model Endüstriyel Robot Programlama Uygulaması (Lehimleme, Montaj ve Görüntü İşleme Hücreleri)

Bu uygulamada COSIMIR Industrial yazılımının kullanımı, Movemaster komut kümesi kullanarak robot programlama, merkezi koordinat sistemi (XYZ modu) ve robotun hareket eksenleri konularında bilgi ve deneyim sahibi olunması amaçlanmaktadır.

Deney; FMS100 bünyesindeki Lehimleme, Montaj ve Görüntü İşleme hücreleri ve hücrelerde bulunan robotlar kullanılarak yürütülmektedir. Deneyde özetle; imalat hücrelerindeki “Pallet 1” alanına yerleştirilecek boş bir Palet2ED (62400) robot tarafından alınarak, sırasıyla XYZ eksenlerinde hareket ettirilip ABC açılarında döndürülecek ve palet tekrar yerine bırakılacaktır.

Bu uygulamada temel olarak;

- Robot ve çevre birimlerinin tanıtılması,
- El terminalinin kullanımı,
- Robotun eksenel, merkezi ve takım koordinat düzlemli düzlemli hareketleri,
- COSIMIR Industrial yazılımının kullanılması,
- Movemaster komut kümesi kullanarak robot programlama,
- Robot programının benzetimi

konularında bilgi ve deneyim sahibi olunması hedeflenmiştir.

3.2.2.1. Deney Baş Kontrol Listesi

Lütfen deneye başlamadan önce aşağıdaki listede belirtilen kontrolleri tek tek yapınız.

- 1) Hücre denetleyicisi aktif mi? (controller on)
Evet: Bir sonraki adıma geçiniz.
Hayır: Ders notlarından “İmalat Hücresini Devreye Alma ve Devreden Çıkartma” bölümüne bakınız.
- 2) Robot çalışıyor mu?
Evet: Bir sonraki adıma geçiniz.
Hayır: Robot kontrolör üzerindeki açma/kapama anahtarını “ON” duruma getiriniz.
- 3) El terminali çalışıyor mu (ekranında yazılar var mı)?
Evet: Bir sonraki adıma geçiniz.
Hayır: Laboratuvardan sorumlu öğretim elemanına haber veriniz.
- 4) El terminali aktif (enable) mi?
Evet: Bir sonraki adıma geçiniz.
Hayır: En üstteki anahtarı “Enable” konumuna getirerek el terminalini aktif hale getiriniz. Ekran arkasındaki yeşil renk aydınlatma görülecektir.

5) Robot alarm veriyor mu?

Evet: Robot kontrolör üzerindeki “RESET” tuşuna basınız.

Hayır: Bir sonraki adıma geçiniz.

6) Robot servoları aktif mi?

Evet: Bir sonraki adıma geçiniz.

Hayır: El Terminali menüsünden “2-RUN > 1-SERVO > SERVO ON (1) > INP/EXE” seçerek servoları aktif duruma getiriniz.

7) Tutucu el (gripper) güç kablosu takılı mı?

Evet: Deneye başlayabilirsiniz.

Hayır: Tutucu el üzerindeki güç kablosunu yerine takınız. Sonrasında deneye başlayabilirsiniz.

3.2.2.2. Deney İşlem Adımları

Bu deney temel olarak ders notlarının “Endüstriyel Robot Kullanımı ve Programlama” bölümünde anlatılan konuları kapsamaktadır. Lütfen öncelikle ders notlarının ilgili bölümünü okuyunuz.

- 1) COSIMIR Industrial yazılımında “File>Project Wizard” seçiniz. Ekrana gelen penceredeki “Project Name”, “Program Name”, “Created by”, “Initials” ve “Description” kutucukları doldurarak “Next” butonuna basınız. Bu adım ile ilgili detaylar için ders notları “Yeni Bir Proje Oluşturma” bölümünü okuyunuz.
- 2) Ekrana gelen pencereden “Robot Type”: RV-E2, “I/O Interface Cards”: 1, “Programming Language”: Movemaster Command, “Additional Axis 1 (L1)”: none olacak şekilde seçerek “Finish” butonuna basınız.
- 3) Konum penceresi üzerindeki boş bir alanda “Sağ Tuş > Insert Position” ile yeni bir konum ekleyiniz. Böylelikle benzetim penceresinde bulunan robot benzetim modelinin konum bilgileri pencereye eklenecektir. Bu işlemi 7 kez daha tekrarlayarak, pencereye toplamda 8 farklı konum ekleyiniz. (Eklediğiniz konumların içeriği şu aşamada önemli değildir.)
- 4) Konum penceresinde listelenen konumlar üzerinde tek tek “Sağ Tuş > Properties” seçiniz. Ekrana gelen “Position List Entry” penceresine Tablo 3-6’da verilen XYZABC değerlerini ve açıklamaları giriniz. “Configuration” ve “Gripper State” bölümündeki parametreleri “ignore” olarak bırakınız. Aynı işlemi tüm konumlar için tekrarlayınız.

Tablo 3-6. Örnek uygulamaya ait konumlar ve açıklamaları.

Adı	X	Y	Z	A	B	C	Açıklama
1	300	100	350	180	0	-90	Baslangic
2	400	100	350	180	0	-90	Baslangic X+100
3	300	200	350	180	0	-90	Baslangic Y+100
4	300	100	450	180	0	-90	Baslangic Z+100
5	300	100	350	210	0	-90	Baslangic A+30
6	300	100	350	180	30	-90	Baslangic B+30
7	300	100	350	180	0	-60	Baslangic C+30
8	369.40	157.08	128.65	179.43	-0.97	-93.75	Palet Alma/Birakma

- 5) Deney yaptığınız imalat hücresindeki “Pallet 1” alanına boş bir Palet2ED yerleştiriniz. El terminalini XYZ modunda kullanarak robota ait tutucu eli paleti tutacak konuma getiriniz.
- 6) Robot ile bağlantı kurunuz. Bunun için “Execute > Init Connection” tıklayınız.
- 7) Robot elini paleti tutmak için getirdiğiniz konumu COSIMIR yazılımındaki benzetim modeline aktarınız. Bunun için “Execute > Robot Position -> COSIMIR” tıklayınız.
- 8) Robot benzetim modelinin konumunu, konum penceresindeki 8 numaralı konum üzerine kaydederek 8 numaralı konum bilgisini güncelleyiniz.
- 9) Program penceresine aşağıdaki kodları, satır numaraları ile birlikte giriniz.

10 SP 10	130 MS 2,C	250 MS 5,C	370 SP 10
20 MTS 8,-60.00,O	140 TI 10	260 TI 10	380 MTS 8,-20.00,C
30 SP 5	150 MS 1,C	270 MS 1,C	390 SP 5
40 MS 8,O	160 TI 10	280 TI 10	400 MS 8,C
50 TI 3	170 MS 3,C	290 MS 6,C	410 TI 3
60 GC	180 TI 10	300 TI 10	420 GO
70 TI 3	190 MS 1,C	310 MS 1,C	430 TI 3
80 MTS 8,-20.00,C	200 TI 10	320 TI 10	440 MTS 8,-60.00,O
90 SP 8	210 MS 4,C	330 MS 7,C	450 SP 10
100 MS 1,C	220 TI 10	340 TI 10	460 MO 1,C
110 SP 10	230 MS 1,C	350 MS 1,C	470 ED
120 TI 10	240 TI 10	360 TI 10	

Şekil 3-144. RV-E2 model robotun merkezi koordinat sistemindeki (XYZ modu) hareketine yönelik örnek robot programı.

- 10) Oluşturduğunuz programı robota yükleyip çalıştırmadan önce benzetimini yapınız. Bunun için önce “Execute > Compile” tıklayarak yazdığınız programı derleyiniz. Eğer program kodlarında veya konum bilgilerinde bir hata varsa derleme sonucunda ortaya çıkacaktır.

- 11) Derlediğiniz programa ait benzetimin ilk satırdan itibaren çalışmaya başlaması için önce “Execute > Reset Program” tıklayınız. Sonrasında program benzetimini çalıştırmak için “Execute > Start” tıklayınız.
- 12) Projeyi robota yükleyiniz. Bunun için önce program penceresini tıklayarak aktif hale getiriniz. Sonra “Execute > Download PC->Robot” seçiniz. Ekrana gelen pencereadaki parametreleri aşağıdaki gibi belirleyerek “OK” butonuna basınız.
 - “Delete All Before Downloading”: seçili,
 - “From Line”: 0,
 - “To Line”: 9000,
 - “Name”: 110
- 13) Programı robota yükledikten sonra konum listesini de yükleyiniz. Bunun için önce konum penceresini tıklayarak aktif hale getiriniz. Sonra “Execute > Download PC->Robot” seçiniz. Ekrana gelen pencereadaki parametreleri aşağıdaki gibi belirleyerek “OK” butonuna basınız.
 - “Delete All Before Downloading”: seçili,
 - From Position: 0,
 - “To Position”: 9000,
 - “Name”: 110 (program adıyla aynı olmak zorundadır)
- 14) Deney grubunda yer alan elemanlardan birini seçerek robot denetleyici üzerindeki acil durdurma butonuna basabilecek şekilde durmasını sağlayınız. Robot hareketlerinde herhangi bir terslik görüldüğü anda acil durdurma butonuna hemen basılmalıdır.
- 15) Robot denetleyici üzerindeki yeşil renkli “Start” butonuna basarak programı çalıştırınız ve robotun hareketlerini gözlemleyiniz.
- 16) Robotun hareketleri ile ilgili gözlemlerinizi “Deney Sonuçları” bölümüne yazınız.

3.2.2.3. Deney Sonu Kontrol Listesi

Lütfen deney bitiminde aşağıdaki listede belirtilen kontrolleri tek tek yapınız.

- 1) Robot kolu gövdeden uzakta mı duruyor?
Evet: El terminalini kullanarak XYZ moduna geçiniz ve robot kolunu gövdeye yaklaştıracak şekilde –X yönünde hareket ettiriniz. Kol ile robot gövdesi arası mesafe göz kararı 10 cm olana kadar devam ediniz.
Hayır: Bir sonraki adıma geçiniz.
- 2) Tutcu el (gripper) güç kablosu takılı mı?
Evet: Tutucu el üzerindeki güç kablosunu yavaşça çıkartınız.
Hayır: Bir sonraki adıma geçiniz.
- 3) Robot servoları aktif mi?

Hayır: Bir sonraki adıma geçiniz.

Hayır: Deney bitmiştir. Deney sonuçlarını içeren raporu oluşturarak dersin öğretim elemanına iletiniz.

Bu deney kapsamında öğrendiklerinizi aşağıda ayrılan bölüme sığacak şekilde yazınız.

[illegible]

3.2.3. Deney Adı: COSIMIR Factory Control Yazılımında Yeni Bir Süreç Planı Oluşturma – Temel İşlevler

Bu uygulamada COSIMIR Factory Control yazılımı bünyesinde oluşturulabilen süreç planlarının yapısı ve yeni bir süreç planı oluşturma konularında bilgi ve deney sahibi olunması amaçlanmaktadır.

Deney; FMS100 bünyesindeki SCADA Bilgisayarı ile Stok (AS-RS) ve Montaj hücreleri kullanılarak yürütülmektedir. Deneyde özetle, Stok istasyonuna boş bir Ana-Palet çağrılır ve istasyona kenetlenir. Stok hücresinden bir adet boş Palet 2ED (62400) çağrılır ve Montaj istasyonuna taşınır. İstasyona gelen palet kenetlendikten sonra robot tarafından alınarak Montaj hücresi içerisindeki Palet 1 alanına taşınır. Sonrasında magazinden sırayla 2 adet üst kapak alınarak Palet 1 alanındaki boş palet üzerine yerleştirilir ve palet doldurulur. Montaj istasyonuna boş bir Ana-Palet çağrılır ve dolu durumdaki Palet 2ED, boş olan Ana-Palet üzerine yerleştirilir. Dolu durumdaki Ana-Palet depolama hücresine kadar taşınır ve üzerindeki dolu durumdaki Palet 2ED (60413) Teleskop tarafından boş bir rafa yerleştirilir. Deneyde, bu işlemleri gerçekleştiren robot programlarının önceden hazırlanmış durumda oldukları kabul edilmektedir.

Bu uygulamada temel olarak;

- SCADA yazılımı olan COSIMIR Factory Control yazılımının tanıtılması;
- FMS100 imalat hücrelerinin tam otomatik modda başlatılması;
- FMS100 bünyesinde tanımlı görevler (task) kullanılarak yeni bir süreç planı oluşturulması;
- Süreç planları ile robot programları arasındaki ilişkinin tanımlanması;
- Stok kodu kavramının anlaşılması

konularında bilgi ve deneyim sahibi olunması hedeflenmiştir.

3.2.3.1. Deney Baş Kontrol Listesi

Lütfen deneye başlamadan önce aşağıdaki listede belirtilen kontrolleri tek tek yapınız.

1) Montaj hücresi denetleyicisi aktif mi? (controller on)

Evet: Bir sonraki adıma geçiniz.

Hayır: Ders notlarından “İmalat Hücresini Devreye Alma ve Devreden Çıkartma” bölümüne bakınız.

2) Montaj hücresindeki robot çalışıyor mu?

Evet: Bir sonraki adıma geçiniz.

Hayır: Robot kontrolör üzerindeki açma/kapama anahtarını “ON” duruma getiriniz.

- 3) Montaj hücresindeki robot el terminali çalışıyor mu (ekranında yazılar var mı)?
Evet: Bir sonraki adıma geçiniz.
Hayır: Laboratuvarından sorumlu öğretim elemanına haber veriniz.
- 4) Montaj hücresindeki robot el terminali aktif (enable) mi?
Evet: En üstteki anahtarı “Disable” konumuna getirerek el terminalini pasif hale getiriniz. Ekran arkasındaki yeşil renk aydınlatma sönecektir.
Hayır: Bir sonraki adıma geçiniz.
- 5) Montaj hücresindeki robot alarm veriyor mu?
Evet: Robot kontrolör üzerindeki “RESET” tuşuna basınız.
Hayır: Bir sonraki adıma geçiniz.
- 6) Montaj hücresindeki robot servoları aktif mi?
Evet: Bir sonraki adıma geçiniz.
Hayır: El Terminali menüsünden “2-RUN > 1-SERVO > SERVO ON (1) > INP/EXE” yapınız.
- 7) Montaj hücresindeki robota ait tutucu el (gripper) güç kablosu takılı mı?
Evet: Bir sonraki adıma geçiniz.
Hayır: Tutucu el üzerindeki güç kablosunu yerine takınız.
- 8) Stok hücresi panosundaki ana şalter açık (On) durumda mı?
Evet: Bir sonraki adıma geçiniz.
Hayır: Hücre panosunun yanındaki ana şalteri “On” konumuna getiriniz. Yaklaşık 15 saniye hücre denetleyicisinin açılmasını bekleyiniz.
- 9) Stok hücresi kontrol panosundaki “Acil Durdur” anahtarı basılı durumda mı?
Evet: “Acil Durdur” anahtarını kendinize doğru çekerek kaldırınız.
Hayır: Bir sonraki adıma geçiniz.
- 10) Stok hücresi kontrol panosundaki “Reset” ışığı yanıyor mu?
Evet: Bir sonraki adıma geçiniz.
Hayır: Stok hücresi kontrol panosundaki “Quit” butonuna basınız.
- 11) Stok hücresi kontrol panosundaki “Reset” ışığı yanıp sönüyor mu?
Evet: “Reset” butonuna basınız ve bir sonraki adıma geçiniz.
Hayır: Stok hücresi kontrol panosundaki “Quit” butonuna basınız. “Reset” ışığı yanıp sönmeye başlayana kadar bekleyiniz.
- 12) Stok hücresi kontrol panosundaki “Start” ışığı yanıp sönüyor mu?
Evet: Bir sonraki adıma geçiniz.
Hayır: Stok hücresi kontrol panosundaki “Reset” butonuna basınız. “Reset” ışığı sabit yanmaya, “Start” ışığı yanıp sönmeye başlayana kadar bekleyiniz.
- 13) SCADA ve Veri Tabanı bilgisayarları açık mı?
Evet: Bir sonraki adıma geçiniz.

Line	Condition	Device	Task	Next Line	Comment
; ALT SUREC 4					
; Bos bir Palet 2ED (62400) Stok hucresinden cagirlarak Montaj hucresine tasinir.					
; Hucre icindeki magazinlerden 2 adet ust kapak alinarak robot ile palete yuklenir.					
; Dolu durumdaki Palet 2ED (62413) Stok hucresine tasinarak saklanir.					
;					
; Baslangic parametreleri okunur.					
BASLAT			%SurecAdi = %Par.1		
10		.CALC	%Par.0		
20		DIALOG	Show("Surec plani " & \$AS.Processes[\$PID].ProcPlan.Name & ", " & %Par.0 & " yerine 2 parametre ile baslatilmalidir!\nSurec plani sonlandirilacaktır.")	SONLANDIR	
	2	.CALC	%SurecAdi = %Par.1		
30		.CALC	%UrunTipi = %Par.2		

- 5) Süreç planı çalışırken elde edilen verilerin kullanılabilmesi ve sürecin başlangıç zamanının hesaplanabilmesi için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Surec verilerinin kullanılabilmesi için baslangic zamani hesaplanir. (PDA)					
40		.CALC	%BaslangicZamani = NOW()		
50		.CALC	%Hata = "OK"		

- 6) Süreç planında kullanılacak ham iş-parçası ve süreç sonunda elde edilecek işlenmiş iş-parçasına ait seri numaralarının belirlenebilmesi için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Ham is-parcasi ve islenmis is-parcasi seri numaralari belirlenir.					
60		.CALC	%HamSeriNo = #Product.Roulette.PartNo.EmptyPallet		
70		.CALC	%IslenmisSeriNo = #Product.Roulette.PartNo.ProcessedHousingUpperPart.Type[%UrunTipi]		

- 7) Stok istasyonuna boş bir Ana-Palet çağırmak, gelen Ana-Paleti istasyona kenetlemek ve Ana-Paleta boş bir Palet 2ED (62400) yüklemek için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; *****					
; Surec plani bunyesinde yurutulecek islemler:					
; *****					
;					
; Stok istasyonuna 1 adet bos Palet 2ED (62400) cagrilir.					
CAGIR		Transport	REQUIRE(#Production.RequirePositions.Stock,#Production.CarrierID)		
100		PLC	ClampIX1		
110		.CALL	%Sonuc = StockProcess("Retrieve",%HamSeriNo,0)		
120		Transport	SetProductId(%HamSeriNo, #Production.CarrierID)	MONTAJA	
	"STOCK"	DIALOG	Show("ALT SUREC 4:\nStok yetersizligi nedeniyle sonlandirilmistir.")		
130		.CALC	%Hata = "NOK"	SERBEST	

- 8) Stok istasyonunda bulunan ve üzerinde 1 adet boş Palet 2ED taşıyan Ana-Paleti Montaj istasyonuna gönderebilmek için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Bos Palet 2ED (62400) Montaj istasyonuna tasinir.					
MONTAJA		Transport	to_Assembly(#Production.CarrierID)		
200		.CALC	#Dev.Assembly.BufPos.1		
210		.CALC	%PaletKonum = 1		
		.NOOP		MONTAJA	

- 9) Montaj istasyonundaki boş Palet 2ED'yi alıp, Montaj hücresi içerisindeki Palet 1 alanına taşıyan robot programını çalıştırmak ve boşalan Ana-Paleti serbest bırakmak için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Bos Palet 2ED (62400) konveyor sisteminden alinarak Montaj hucresi icindeki Palet 1 alanina tasinir.					
MONTAJ		.CLAIM	#Dev.Assembly.Ready		
220		.CALC	#Dev.Assembly.Task = "MovePallet(15," & %PaletKonum & ")"		
230		Assembly	%Sonuc = MovePallet(15 , %PaletKonum)		
240		.CALC	#Dev.Assembly.Task = ""		
250		.RELEASE	#Dev.Assembly.Ready		
260		.CALC	%Sonuc		
270		0 Transport	SetProductId("",#Production.CarrierID)	YAZDIR	
		DIALOG	ChooseRadio("Montaj:\nHata <MovePallet(15, " & %PaletKonum & ")>\n" & #ErrCodes.Assembly.MovePallet[%Sonuc], "Retry","Abort","Ignore")		
280	"Retry"	.NOOP	%Sonuc	MONTAJ	
	"Abort"	.CALC	%Hata = "STORE"	SERBEST	
		.NOOP			
290		.CALC	#Dev.Assembly.BufPos.1 = %HamSeriNo		
295		Transport	Release(#Production.CarrierID)		

10) Montaj hücresindeki üst kapak magazininden (2) 2 adet işlenmiş üst kapağı sırayla alıp, Palet 1 alanında bulunan boş Palet 2ED üzerindeki alanlara yerleştiren robot programını çalıştırmak için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Hücresindeki magazinden 2 adet üst kapak sırayla alınarak boş Palet 2ED üzerine tasnir.					
300		.CALC	%Konum = 11		
PARCATASI		.CLAIM	#Dev.AssemblyReady		
310		.CALC	#Dev.Assembly.Task = "MoveChas(1," & %Konum & ")"		
320		Assembly	%Sonuc = MoveChas(1,%Konum)		
330		.RELEASE	#Dev.AssemblyReady		
340		.CALC	#Dev.Assembly.Task = ""		
350		.CALC	%Sonuc		
360	0	.CALC	%Konum		
		DIALOG	ChooseRadio("Montaj:\nHata <MoveChas(1," & %Konum & ")>\n" & #ErrCodes.Assembly.MoveChas(%Sonuc), "Retry","Abort","Ignore")		Kontrol
370	"Retry"	.NOOP	%Sonuc		
	"Abort"	.CALC	%Hata = "STORE"		PARCATASI
		.NOOP			SERBEST
Kontrol	11	.CALC	%Konum = 12		PARCATASI
		.NOOP			

11) Montaj istasyonuna boş bir Ana-Palet çağırmak ve gelen Ana-Paleti istasyona kenetleyebilmek için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Montaj istasyonuna 1 adet boş ana palet çağırılır ve gelen ana palet kenetlenir.					
380		Transport	REQUIRE(#Production.RequirePositions.Assembly,#Production.CarrierID)		
390		PLC	ClampIX3		

12) Montaj hücresindeki Palet 1 alanında bulunan ve 2 adet işlenmiş üst kapak taşıyan durumdaki Palet 2ED'yi (62413) alarak, istasyonda kenetlenmiş durumdaki Ana-Palet üzerine taşıyan robot programını çalıştırmak için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Hücresindeki Palet 1 alanında bulunan dolu durumdaki Palet 2ED (62413) konveyör sistemine tasnir.					
KONVEYOR		.CLAIM	#Dev.AssemblyReady		
400		.CALC	#Dev.Assembly.Task = "MovePallet(" & %PaletKonum & ", 15)"		
410		Assembly	%Sonuc = MovePallet(%PaletKonum,15)		
420		.CALC	#Dev.Assembly.Task = ""		
430		.RELEASE	#Dev.AssemblyReady		
440		.CALC	%Sonuc		
450	0	Transport	SetProductID("",#Production.CarrierID)		Temizle
		DIALOG	ChooseRadio("Montaj:\nHata <MoveChas(1," & %Konum & ")>\n" & #ErrCodes.Assembly.MoveChas(%Sonuc), "Retry","Abort","Ignore")		
460	"Retry"	.NOOP	%Sonuc		KONVEYOR
	"Abort"	.CALC	%Hata = "STORE"		SERBEST
		.NOOP			
Temizle		Transport	#Dev.AssemblyBufPos.1 = ""		
480		.CALC	SetProductID(%IslenmisSeriNo, #Production.CarrierID)		

13) Montaj istasyonunda bulunan ve üzerinde 1 adet dolu Palet 2ED (62413) taşıyan Ana-Paleti Stok istasyonuna göndermek ve boş bir rafta saklamak için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Montaj istasyonunda dolu durumdaki Palet 2ED (62413) Stok istasyonuna tasnir ve hücre içindeki uygun rafta saklanır.					
SAKLA		Transport	to_Stock(#Production.CarrierID)		
500		.CALL	%Sonuc = StockProcess("Store",%IslenmisSeriNo,0)		
510	"STOCK"	DIALOG	Show("ALT SUREC 4:\nStok hatasından dolayı bir süre sonra tekrar saklamayı deneyeceğiz.")		SAKLA
		.NOOP			

14) İstasyonda kenetlenmiş durumdaki Ana-Palet serbest bırakmak için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
;					
; Stok istasyonunda kenetlenmiş durumdaki ana palet serbest bırakılır.					
SERBEST		Transport	SetProductID("",#Production.CarrierID)		
600		Transport	RELEASE(#Production.CarrierID)		

15) İmalat süreciyle ilgili verilerin işlenebilmesi için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
:					
:			İmalat süreci ile ilgili veriler işlenir.		
VERİSİLE		.CALL	DetermineProductionPDA("Type4",%BaslangicZamani,%Hata)		

16) İmalat sürecini sonlandırmak için aşağıdaki satırlar eklenir:

Line	Condition	Device	Task	Next Line	Comment
:					
:			İmalat süreci sonlandırılır.		
SONLADIR		.CALL	TerminateProcessMonitoring(%SurecAdi)		

- 17) Oluşturulan süreç planı, dosya menüsü üzerinden (File>Save) seçilerek kaydedilir.
- 18) Kaydedilen süreç planı derlenir. Derleme sonucunda, eğer varsa, süreç planı içerisindeki hatalar, satır numaraları ile birlikte görüntülenir.
- 19) Montaj hücreesindeki magazinlerin dolu olduğu ve Palet 1 alanının boş olduğu kontrol edilir.
- 20) Stok istasyonunda boş Palet 2ED (62400) bulunup bulunmadığı görerek kontrol edilir. Paletin bulunduğu en küçük raf numarası bir yere not edilir.
- 21) COSIMIR Factory Control yazılımı araç çubuğu üzerinden üretim (production) moduna alınır.
- 22) SCADA ekranındaki stok yönetimi (stock administration) penceresinden boş Palet 2ED'nin (62400) bulunduğu raf numarasının uyuşup uyuşmadığı kontrol edilir. Uyuşmazlık varsa düzeltilir.
- 23) SCADA ekranındaki proje penceresinden "SubProc4" çalıştırılır.
- 24) FMS100 bünyesindeki elemanların hareketleri ile ilgili gözlemlerinizi "Deney Sonuçları" bölümüne yazınız.

3.2.3.3. Deney Sonu Kontrol Listesi

Lütfen deney bitiminde aşağıdaki listede belirtilen kontrolleri tek tek yapınız.

- 1) COSIMIR Factory Control yazılımı üretim (production) modunda ve çalışıyor mu?
Evet: Üretim modundan çıkınız.
Hayır: Bir sonraki adıma geçiniz.
- 2) SCADA bilgisayar çalışıyor mu?
Evet: Bilgisayarı kapatınız.
Hayır: Bir sonraki adıma geçiniz.
- 3) ERP bilgisayar çalışıyor mu?
Evet: Bilgisayarı kapatınız.
Hayır: Bir sonraki adıma geçiniz.
- 4) Montaj hücreesindeki robot kolu gövdeden uzakta mı duruyor?
Evet: El terminalini kullanarak XYZ moduna geçiniz ve robot kolunu gövdeye yaklaştıracak şekilde -X yönünde hareket ettiriniz. Kol ile robot gövdesi arası mesafe göz kararı 10 cm olana kadar devam ediniz.
Hayır: Bir sonraki adıma geçiniz.

- 5) Montaj hücreesindeki robota ait tutucu el (gripper) güç kablosu takılı mı?
Evet: Tutucu el üzerindeki güç kablosunu yavaşça çıkartınız.
Hayır: Bir sonraki adıma geçiniz.
- 6) Montaj hücreesindeki robot servoları aktif mi?
Evet: El Terminalini “Enable” yaparak gelen menüden “2-RUN > 1-SERVO > SERVO ON (0) > INP/EXE” seçerek servoları pasif hale getiriniz.
Hayır: Bir sonraki adıma geçiniz.
- 7) Montaj hücreesindeki robot el terminali aktif (enable) mi?
Evet: En üstteki anahtarı “Disable” konumuna getirerek el terminalini pasif hale getiriniz. Ekran arkasındaki yeşil renk aydınlatma sönecektir.
Hayır: Bir sonraki adıma geçiniz.
- 8) Montaj hücreesindeki robot çalışıyor mu?
Evet: Bir sonraki adıma geçiniz.
Hayır: Robot kontrolör üzerindeki açma/kapama anahtarını “ON” duruma getiriniz.
- 9) Stok hücresi kontrol panosundaki “Acil Durdur” anahtarı basılı durumda mı?
Evet: Bir sonraki adıma geçiniz.
Hayır: “Acil Durdur” anahtarına basınız.
- 10) Stok hücresi panosundaki ana şalter açık (On) durumda mı?
Evet: Hücre panosunun yanındaki ana şalteri “Off” konumuna getiriniz.
Hayır: Deney bitmiştir, deney sonuçlarını yazınız.

3.2.3.4. Deney Sonuçları

Bu deney kapsamında öğrendiklerinizi aşağıda ayrılan bölüme sığacak şekilde yazınız.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3.3. Tartışma

Sunulmakta olan bu çalışmanın üzerinde yürütüldüğü FMS100, eğitim amaçlı geliştirilmiş bir fabrika otomasyon sistemidir. Her ne kadar üretici firma tarafından eğitim amaçlı olarak nitelendirilse de, sistem ile birlikte yeterli seviyede eğitim materyali ve Türkçe kaynak sunulmamıştır. Sunulan kaynaklar çoğunlukla aynı ifadelerin tekrarlandığı, İngilizce kullanım kitapçığı şeklinde olup; çoğunlukla sistemin bileşenleri hakkında kısmi teknik bilgiler içermektedir. Öğrencilerin sistemi nasıl kullanacağı ve deney yapacağı ile ilgili herhangi bir ders materyali veya deney föyü yoktur.

FMS100 teknik açıdan irdelendiğinde, tasarlandığı yıllar olan 2000'lerin başının ve hatta günümüzün popüler imalat teknolojilerini bünyesinde barındırmaktadır. Bu bakımdan başarılı olarak kabul edilebilir. Fakat sistem teknik eğitim açısından irdelendiğinde; sistemin dışarı kapalı olan mimarisi, imalat istasyonlarında kullanılan denetleyiciler ve robotlar için yüklenmiş yazılım kodlarının karmaşıklığı, imalat hücrelerinin modüler olmayışı gibi etmenler FMS100'ün eğitim amaçlı kullanımını zorlaştırmaktadır.

FMS100 bünyesindeki imalat hücrelerindeki otomasyon denetleyicileri S7-300 ve üretici firmanın kendi ürünü olan IPC model denetleyicilerdir. Örneğin Lehimleme ve Montaj hücrelerinde S7-300 bulunurken, Hidrolik-Baskı ve en yoğun kullanılan Montaj hücresinde IPC bulunmaktadır. Bu durum, imalat hücrelerinde yürütülen ve otomasyonun eğitimi açısından büyük öneme sahip olan imalat hücresi denetim programının çözümlenmesi ve yeniden programlamasının öğretilmesi açısından zorluklara sebep olmaktadır. Öğrenciler EİS dersini alana kadar PLC dersi almakta ve S7-300 model PLC programlamayı öğrenmektedir. Fakat IPC'nin programlanması ve programlama yazılımı (Integrated Development Environment - IDE) S7-300'den daha farklıdır. Öğrencilere kısıtlı olan zaman içerisinde IPC'nin programlanmasının da öğretilmesi gerekmektedir. Halbuki tüm imalat hücrelerinde, endüstride yaygın kullanılan ve birçok kurumda eğitimi verilmekte olan S7-300 gibi tek tip bir denetleyici kullanılması, imalat hücresi denetim programının eğitimi açısından daha kolaylık olurdu.

Ayrıca teknik eğitimde öğrencinin bilgisini en çok pekiştiren uygulamalardan biri arıza takibidir. Eğitici sistem üzerinde çeşitli arızalar üretebilmeli, arıza kaynağının öğrenci tarafından bulunabilmesi için öğrenciye çeşitli sorular sorabilmelidir. Örneğin sistem bünyesinde, üzerinde kilit olan kapalı bir arıza kutusu olsa, eğitmen kutu içinde bulunacak

anahtarları kullanarak sistem üzerinde çeşitli arızalar üretebilse, sonra da öğrenciye bu konuda sorular sorsa; eğitim açısından çok faydalı bir uygulama olurdu. FMS100 sisteminde böyle bir arıza kutusu veya benzeri bulunmamaktadır.

Tüm bu yönleriyle irdelendiğinde, FMS100 sistemi teknik açıdan hala güncel bir fabrika otomasyon sistemi olup; üretici firma tarafından teslim edildiği haliyle, eğitim açısından eksikleri bulunmaktadır. Bu tez çalışması ile FMS100'ün teknik eğitim açısından eksikleri giderilmeye çalışılmıştır.

FMS100'ün büyüklüğü ve kapsadığı konuların genişliği göz önüne alındığında, sistemin tamamı için eğitim materyali hazırlamak oldukça güç bir iştir. Bu nedenle sunulmakta olan bu tez çalışmasının tamamlanması uzun zaman almıştır. Böylesi bir çalışmanın, endüstriyel robot ve otomasyon, makina ve eğitim gibi farklı alanlarda uzmanlaşmış bir ekip tarafından yapılması daha doğrudur. Bu çalışma tek kişi tarafından yürütülmüştür.

FMS100, Mekatronik Mühendisliği bölümü ders programında yer alan EİS dersinin temelini oluşturmaktadır. EİS dersi, eski ders programında 8. dönemde işlenmektedir. Öğrenci 8. döneme kadar; EİS'i oluşturan algılayıcılar, eyleyiciler, PLC, robotik, bilgisayar destekli tasarım ve imalat gibi temel teknolojiler hakkında bilgi sahibi olmaktadır. EİS dersinde ise, mezun olmadan önce, son kez, tüm bu almış olduğu bilgilerin örnek bir fabrika otomasyon sistemi üzerindeki kullanımı ve işleyişini öğrenmektedir. Bu bakımdan EİS dersi, mezun olana kadar edinilen bilgilerin tazelenmesini ve pekişmesini sağlamaktadır.

Bölümümüzde EİS dersi ve uygulaması, 2015-2016 ve 2016-2017 bahar dönemleri olmak üzere iki defa tarafımdan verilmiştir. Bu uygulamalar sırasında, sunulmakta olan bu tez çalışması kapsamında geliştirilen materyaller de kullanılmıştır. 2016-2017 yılından itibaren yeni ders programı devreye girmiş ve yeni programda EİS dersi 6. döneme taşınmıştır.

2016-2017 yılında EİS dersini toplam 85 öğrenci almıştır ve ders notları Tablo 3-7'da irdelenmiştir. Tabloda yer alan 17 öğrenci 6. döneminde, 64 öğrenci ise 8. dönemindedir. Mühendislik tamamlama programı kapsamında dersi alan 1 öğrenci ile derse ait sınavlara girmemiş 3 öğrenci değerlendirmeye katılmamıştır. Tabloya göre dersi 6. dönemde alan öğrencilerden bütünlemeye kalanların yüzdesi, 8. dönemde alanlara göre daha yüksektir. Dersi 6. dönemde alanların daha büyük bir kısmı bütünlemeye kalmıştır. Dersi geçenler

açısından değerlendirildiğinde; dersi 8. dönemde alıp da başarılı olanların yüzdesi, 6. dönemde alanlara göre yüksektir. Her iki grubun dönem sonu not ortalamaları değerlendirildiğinde ise; dersi 6. dönemde alanların not ortalamalarının, 8. dönemde alanlardan biraz daha fazla olduğu görülmektedir.

Tablo 3-7. 2016-2017 yılı bahar döneminde EİS dersini alan öğrencilerin başarı durumları.

	Toplam	Bütünlemeye Kalan		Dersi Geçen		Başarı Notu Ortalaması
		Sayı	Yüzde	Sayı	Yüzde	
6. Dönem	17	4	%23,53	16	%94,12	51,06
8. Dönem	64	9	%14,06	63	%98,44	50,86

Her ne kadar dersi 6. dönemde alanların başarı notu ortalaması çok az bir farkla $(((51,06 - 50,86) / 50,86) \times 100 = \% 0,39)$ 8. dönem öğrencilerinden fazla olsa da, 6. dönem öğrencilerinin bütünlemeye kalma yüzdesi 8. dönem öğrencilerinden $(((23,53 - 14,06) / 14,06) \times 100 = \%67,35)$ fazladır. EİS dersi kapsamı, Mekatronik Mühendisliği bölümü müfredatının büyük bir çoğunluğunu kapsamaktadır. Böylelikle EİS dersinde tüm bölüm müfredatı özetlenmekte ve konuların pekişmesi sağlanmaktadır. Bu nedenlerle, EİS dersinin 8. dönemde okutulması hususu, öğrencilerin EİS dersinden önce daha fazla bölüm dersi almış olması ve bilgilerinin daha fazla olgunlaşmasını; EİS dersinin daha verimli ve etkili olacağı şeklinde yorumlanabilir.

Sunulmakta olan bu tez çalışması kapsamında hazırlanan eğitim modülünün başarısını ölçmek için, 2016-2017 bahar döneminde bir çalışma yürütülmüştür. Çalışma kapsamında 50 sorudan oluşan ve cevapları 5'er şıktan oluşan çoktan seçmeli bir test hazırlanmıştır. EKLER bölümünde sunulmakta olan testin içeriğinde EİS dersi konularını kapsayacak sorular vardır. Test dönem başında, ara sınav ve final sınavı sırasında olmak üzere öğrencilere 3 kere uygulanmıştır. Böylelikle eğitim modülü uygulanmadan önce, uygulamanın ortasında ve sonunda EİS konusundaki gelişimlerini ölçmek amaçlanmıştır. Uygulamaya her üç sınavda da eksiksiz katılan 53 öğrencinin verdikleri cevaplar çalışma kapsamına alınmış ve 100 üzerinden değerlendirilmiştir. Elde edilen ortalama değerleri Tablo 3-8'de verilmiştir.

Tablo 3-8. Değerlendirme testi sonuçları.

	Ön Test Ortalaması	Ara Sınav Testi Ortalaması	Final Testi Ortalaması
2016-2017 bahar dönemi	39,35	60,49	54,87

Tabloya göre dönem ortasında, dönem başına göre ortalama değeri yaklaşık %53,72 artmış, dönem sonunda ise dönem ortasına göre not ortalaması %9,29 azalmıştır. Dönem ortasında gözlenen artış, öğrencilerin dönem boyunca almış oldukları eğitimden dolayı beklenen bir durumdur. Fakat bilginin daha da artmış olması gereken dönem sonunda gözlenen düşüş beklenti dışıdır. Bunun sebebi düşünüldüğünde ilk akla gelen, öğretilen bilgilerin kalıcılığının düşük olduğunun yanı sıra; hem final sınavı stresi, hem de öğrencilerin herhangi bir not alamayacakları bu değerlendirme testine karşı isteksizlikleri öne sürülebilir. Yine de dönem sonunda, dönem başına göre %39,44 başarı artışı gözlenmiştir.

4. SONUÇLAR

FMS100 sisteminin kapsadığı teknik konuların büyüklüğü göz önüne alındığında, birçok konuda hazırlanan ders notlarının derinliği arttırılabilir, içeriği zenginleştirilebilir. Bu da daha fazla emek ve zaman gerektirir.

Hazırlanacak eğitim modüllerinin başarısının testi de uzun zaman çalışmalar gerektirir. Yapılacak akademik çalışmalarda aynı girdiyi birçok guruba tekrar tekrar uygulamak ve çıktılarını da ölçmek gerekir.

Bu yönleriyle bakıldığında, sunulmakta olan bu tez çalışması oldukça geniş kapsamlı bir konudur. İleride, bu tez çalışması sırasında hazırlanan ders notu içeriği ve konu çeşitliliği zenginleştirilip, lise ve üniversite seviyesindeki teknik okulların geneline hitap eden bir kitap çalışması yapılması hedeflenmektedir. Bunun yanında hazırlanan eğitim modüllerinin başarımlarını ölçme çalışmaları yapılması da planlanmaktadır.

Sonuç olarak bu tez çalışmasında, EİS dersi için ders notları ve deney föyleri hazırlanmıştır. FMS100 sisteminin İngilizce olan tüm teknik dokümanları içeriği, zenginleştirilerek Türkçe'ye çevrilmiştir. Hazırlanan eğitim materyalleri içerisinde, elektronik ortamda rahatça gezinti yapılabilmesini sağlamak için çapraz bağlantılar kullanılmıştır.

Yürütülen bu tez çalışması sırasında yapılan kişisel girişimler sayesinde, 2 kez Yapı İşleri Daire Başkanlığı ihalesi gerçekleştirilmiş ve bir BAP E tipi altyapı projesi tamamlanmıştır. Bu kapsamda gerçekleşen hizmet alımları ve kişisel olarak gerçekleştirilen onarım çalışmaları sonucunda, çoğunluğu CIM laboratuvarının hatalı elektrik tesisatından kaynaklanan, FMS100 bünyesinde zaman içinde çeşitli arızalara sebep olan ve giderilen teknik aksaklıklar, genel hatlarıyla aşağıdaki gibidir:

- CIM laboratuvarındaki dağıtım panosuna gelen elektrik tesisatı, bina ana panosundan itibaren yenilenmiş;
- FMS100'ü besleyen elektrik hattına 3 faz 110KW jeneratör ve 3 faz 50KW'lık kesintisiz güç kaynağı bağlanmış, özetle elektrik tesisatındaki tüm aksaklıklar giderilmiş;
- Lehimleme hücresinde bulunan S7-300 PLC'de takılı durumdaki Profibus FMS haberleşme kartı yenisiyle değiştirilmiş, gerekli tüm kurulum ve haberleşme ayarları yapılmış;

- Lehimleme ve Hidrolik-Baskı imalat hücrelerinde bulunan arızalı durumdaki güç kaynakları yenileriyle değiştirilmiştir;
- Depolama hücresi denetleyicisinde bulunan ve Kartezyen konumlama sisteminin ara sıra kendi kendine anlamsız hareketlerde bulunarak arızaya geçmesine neden olan hata kaynağı tespit edilmiş ve hata giderilmiştir;
- Montaj hücresindeki robotta takılı olan ve arızalı tutucu el değiştirilmiştir, Görüntü-İşleme hücresindeki robotta takılı olan ve ara sıra tutukluk yapan tutucu elin bakımı yapılmıştır;
- CNC-Çifti hücresindeki robotun 3. eksen motoru, pulley dişlisi ve kayışı yenisiyle değiştirilmiştir, aynı robotun denetleyicisinde olan ve 3. eksen çıkışı arızalı olan eksen sürücü kartı yenisiyle değiştirilmiştir;
- Lehimleme hücresindeki robot havya eli tamir edilmiş, bakımı yapılmıştır, havya sürücüsünde bulunan S7-200 model PLC yazılımına müdahale edilerek Almanca olan ekran yazılımı Türkçeleştirilmiştir;
- CNC-Çifti hücresindeki CNC tezgahları temizlenmiş ve yağlanarak eksen yatak bakımları yapılmıştır;
- Tüm imalat hücrelerindeki robotların hafıza pilleri 2 kere değiştirilmiştir, robot eksenleri gres ile yağlanmış, hafıza pillerinin zamansız bitmesi sebebiyle robot eksenlerinin hafızadan silinen sıfır noktaları mekanik durdurucu yöntemiyle yeniden öğretilmiştir, tüm robotlarda bulunan ve FMS100 tarafından kullanılan programlara ait konumlar yeniden öğretilmiştir, robot programlarında güncellemeler yapılmıştır.

Sonuç olarak Marmara Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği bölümündeki FMS100 sisteminde bulunan tüm büyük teknik arızalar giderilmiştir ve tüm imalat hücreleri çalışır durumdadır. Bu haliyle FMS100 bir bütün olarak eğitime hazırdır.

KAYNAKLAR

1. Mori, T., *Mecha-tronics*. 1969.
2. Kyura, N. and H. Oho, *Mechatronics-an industrial perspective*. Mechatronics, IEEE/ASME Transactions on, 1996. **1**(1): p. 10-15.
3. Comerford, R., *Mecha... what?[mechatronics]*. IEEE Spectrum, 1994. **31**(8): p. 46-49.
4. Comerford, R., *Mecha... what?[mechatronics]*. Spectrum, IEEE, 1994. **31**(8): p. 46-49.
5. Company, T.M. *Service Training - Common Rail Diesel Engine*. [cited 2013 08 Şubat]; Available from: <http://www.youtube.com/watch?v=jWv5gYWvXaY&list=LLPRT9V8ul2XYpBT1RtEb59Q>.
6. Hubing, T., H., et al. *Engine Control, Basic Description*. [cited 2016 08.07]; Available from: http://www.cvel.clemson.edu/auto/systems/engine_control.html.
7. *Bilim ve Sanat Terimleri Ana Sözlüğü*. 2013 [cited 2013 07 Jan]; Available from: <http://www.tdkterim.gov.tr/>.
8. Özkaya, G., *Üretim Planlama Ve Kontrol*. 1987, İstanbul: İstanbul Teknik Üniversitesi Makina Fakültesi Ofset Atölyesi.
9. Koçer, M., *İşletme Yönetiminde Temel Bilgiler Ve İlkeler*. 1968, Ankara: Ankara İTİA Ders Notları.
10. Hawkins, W.M. and T.G. Fisher, *Batch control systems : design, application, and implementation*. 2nd Edition ed. 1990, USA: Instrument Society of America - ISA.
11. Rao, M., et al. *Framework of computer integrated process systems*. in *Second IEEE Conference on Control Applications*,. 1993.
12. Şatır, S. and S. Tolun. *Bölgesel Kültür Kapsamında Üreticiden Tüketicilere Zeytinyağı Ve Donanımları*. in *Tarıma Dayalı Sanayilerde Birinci Ürün ve Hizmet Tasarımı Sempozyumu ve Sergisi*. 2005. Kuşadası, İzmir.
13. Varol, M., *Prina, Kömür Ve Prina-Kömür Karışımlarının Akışkan Yataktaki Yakılması*, in *Çevre Mühendisliği*. 2006, Ortadoğu Teknik Üniversitesi: Ankara.
14. Hunt, V.D., *Introduction to Mechatronics*, in *Mechatronics:Japan's Newest Threat*. 1988, Springer US. p. 3-25.
15. Johnsson, C., *A graphical language for batch control*. 1999.
16. Kosanke, K., *The European approach for an open system architecture for CIM (CIM-OSA)-ESPRIT project 5288 AMICE*. Computing & Control Engineering Journal, 1991. **2**(3): p. 103-108.
17. ElMaraghy, H., A. , *Flexible and reconfigurable manufacturing systems paradigms*. International Journal of Flexible Manufacturing Systems, 2005. **17**(4): p. 261-276.
18. Tanış, R., *Otomatik Kontrolün Tarihi*, in *Fen Bilimleri Enstitüsü*. 2006, Marmara Üniversitesi.
19. Batchelor, R., *Henry Ford, mass production, modernism, and design*. Vol. 1. 1994: Manchester University Press.
20. Wilson, J., M. , *Henry Ford's just-in-time system*. International Journal of Operations & Production Management, 1995. **15**(12): p. 59-75.
21. Ford, H. and S. Crowther, *My Life and Work*. 1924, London: Heinemann.

22. Hakan, S., İhsan, *Bütünleşik imalat ortamlarında ürün ağacı yapısının imalat performansı üzerine etkisi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2011, Sakarya Üniversitesi: YÖK Veri Tabanı. p. 198.
23. Tekez, E., *Zeki bütünleşik imalat sistemlerinin oluşturulması için bir referans model geliştirilmesi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2006, Sakarya Üniversitesi: YÖK Veri Tabanı. p. 249.
24. Ayağ, Z., *Aşamalı bir bilgisayar bütünleşik imalat (BBİ) sistemi modeli ve benzetim tekniği ile analizi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2003, İstanbul Teknik Üniversitesi: YÖK Veritabanı. p. 158.
25. Göktaş, O., *Mobilya endüstrisinde bütünleşik imalat yönetim sistemi ve bir kural tabanlı ana üretim çizelgeleme modeli*, in *Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı İşletme Yönetimi Bölümü*. 2001, Sakarya Üniversitesi: YÖK Veri Tabanı. p. 172.
26. Yıldız, G., *CIM yatırımlarının bulanık AHP yöntemi ile değerlendirilmesi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2003, İstanbul Teknik Üniversitesi: YÖK. p. 184.
27. Akalp, M., Kaan, *Bilgisayar tümleşik üretim ortamında ağ-tabanlı atölye çizelgeleme sistemi geliştirilmesi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 2002, Orta Doğu Teknik Üniversitesi: YÖK Veritabanı. p. 152.
28. Durak, U., *Bilgisayar tümleşik imalat takımı yönetimi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 2001, Orta Doğu Teknik Üniversitesi: YÖK. p. 179.
29. Bolatoğlu, B., *Bütünleşik imalat kaynakları planlaması (MRP 2) ve tam zamanında üretim (JIT) MRP 2 and JIT*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2000, Sakarya Üniversitesi: YÖK Veritabanı. p. 179.
30. Kılınç, İ., *Orta veya öiçekli fabrikalar için bir bilgisayar tümleşik üretim modelinin geliştirilmesi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 1993, Orta Doğu Teknik Üniversitesi: YÖK. p. 139.
31. Çalkıvık, B., Sefa, *Marmara Üniversitesi bilgisayar bütünleşik imalat laboratuvarında tersine yürütülen prosesler mühendisliği projelerinde verim arttırımı*, in *Fen Bilimleri Enstitüsü*. 1992, Marmara Üniversitesi: YÖK Veri Tabanı. p. 71.
32. Özpeynirci, S., *Esnek imalat sistemlerinde paylaşırma ve makine ucu kararları*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2007, Orta Doğu Teknik Üniversitesi: YÖK. p. 124.
33. Tunçel, G., *Esnek imalat sistemlerinin gerçek zamanlı çizelgeleme ve kontrol problemi için yüksek seviye petri ağlarına dayalı bir karar destek sistemi: Nesneye yönelik bir yaklaşım*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2005, Dokuz Eylül Üniversitesi: YÖK Veritabanı. p. 145.
34. Eraslan, E., *Esnek imalat sistemlerinde performans ölçümü için bilişsel bir yaklaşım*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2004, Gazi Üniversitesi: YÖK. p. 105.
35. Bilkay, O., *Esnek imalat yönetim sistemleri için fuzzy mantığı yaklaşımı kullanılarak bir çizelgeleme yöntemi geliştirilmesi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 2003, Orta Doğu Teknik Üniversitesi: YÖK Veritabanı. p. 365.

36. Arıkan, M., *Esnek imalat sistemlerinde parça seçimi, operasyon atama ve alet deposu yerleşimi için sezgisel algoritmalar*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2003, Gazi Üniversitesi: YÖK Veritabanı. p. 198.
37. Budak, S., *Esnek imalat hücrelerinde atölye çizelgeleme kuralları için benzetim çalışması*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2000, Marmara Üniversitesi: YÖK Veritabanı. p. 114.
38. Türker, A., Kürşad, *İş parçaları ve kesici takımların esnek imalat sistemi içerisinde eş zamanlı akışının tasarımı ve analizi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1998, Sakarya Üniversitesi: YÖK. p. 214.
39. Saygın, C., *Esnek imalat yönetim sistemi için yapısal bir yöntem geliştirilmesi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 1997, Orta Doğu Teknik Üniversitesi: YÖK. p. 331.
40. Akmermer, S., *Esnek imalat sistemlerinde üç serbestlikli manipülatör tasarımı ve kontrolü*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 1997, Erciyes Üniversitesi: YÖK Veritabanı. p. 76.
41. Yücel, O., *Esnek imalat sistemlerinde döner ve kayar mafsallı robotların verimliliğini artırmada ysa ile titreşimli ivme analizi*, in *Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Bölümü*. 2014, Erciyes Üniversitesi: YÖK Veri Tabanı. p. 125.
42. Pınarbaşı, M., *Esnek imalat sistemlerinin tasarım süreci ve performans analizinde kuyruk ağı modelleri*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2010, Kırıkkale Üniversitesi: YÖK Veri Tabanı. p. 160.
43. Kuşcuoğlu, D., *Esnek imalat sistemlerinde takım atama problemine malzeme taşıma odaklı bir yaklaşım*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2009, Boğaziçi Üniversitesi: YÖK. p. 119.
44. Güçlü, İ., *Esnek imalat sistemlerinde kesici takımların yerleştirilmesi problemine genetik algoritma yaklaşımı*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2006, Boğaziçi Üniversitesi: YÖK. p. 140.
45. Yücel, N., Deniz, *Esnek imalat sistemlerinin simülasyonu: Pilot uygulama*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 2005, Orta Doğu Teknik Üniversitesi: YÖK. p. 131.
46. Kılınç, F., *Esnek imalat sistemlerinde makine ucu taşıyıcısı problemi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2005, Orta Doğu Teknik Üniversitesi: YÖK. p. 139.
47. Bilgin, S., *Esnek imalat sistemlerinde kapasite paylaşırma problemi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 2004, Orta Doğu Teknik Üniversitesi: YÖK Veritabanı. p. 131.
48. Karakayali, İ., *Esnek imalat sistemlerinde makine ucu değiştirme problemi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2003, Orta Doğu Teknik Üniversitesi: YÖK. p. 163.
49. Çorbacıoğlu, U., *Bir esnek imalat hücresine parçaların döngüsel yüklenmesi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2002, Orta Doğu Teknik Üniversitesi: YÖK Veri Tabanı. p. 87.
50. Gemici, D., Gökçe, *Esnek imalat sistemlerinde takım yönetimi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2000, Marmara Üniversitesi: YÖK Veri Tabanı. p. 64.

51. Kuzgunkaya, O., *Çok amaçlı bulanık 0-1 programlama ile esnek imalat sistemi (EİS) seçimi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2000, Galatasaray Üniversitesi: YÖK Veri Tabanı. p. 49.
52. Özoğul, C., Okan, *Esnek imalat sistemlerinin genişleme esnekliğinin reel opsiyon yaklaşımıyla değerlendirilmesi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2000, Galatasaray Üniversitesi: YÖK Veri Tabanı. p. 125.
53. Öztürk, H., Çiçek, *Laboratuvar ortamında esnek imalat sistemi ve PLC tasarımı*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2000, Gazi Üniversitesi: YÖK Veri Tabanı. p. 104.
54. Cangar, T., *Dağıtık ağ uygulamaları mimarisi kullanılan ajan temelli esnek imalat hücresi geliştirilmesi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 2000, Orta Doğu Teknik Üniversitesi: YÖK Veri Tabanı. p. 149.
55. Üstün, V., *Fiziksel esnek imalat hücresi kontrolü için nesne yönelimli yazılım geliştirilmesi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2000, Orta Doğu Teknik Üniversitesi: YÖK Veri Tabanı. p. 156.
56. İncetürkmen, B., *Bir esnek imalat sisteminde yapay sinir ağları kullanılarak üretim hatlarının belirlenmesi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1999, Sakarya Üniversitesi: YÖK Veri Tabanı. p. 90.
57. Özlü, S., *Parçaların bir döngü çizelgesiyle bir esnek imalat sistemine yüklenmesi üzerine bir araştırma*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 1998, Orta Doğu Teknik Üniversitesi: YÖK Veri Tabanı. p. 176.
58. Boran, H., *Parça elemanları yöntemiyle esnek imalat planlarının oluşturulması*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 1998, Orta Doğu Teknik Üniversitesi: YÖK Veritabanı. p. 114.
59. Öztürk, O., *Bir deneysel esnek imalat sisteminin bazı işletim problemlerinin incelenmesi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1997, Orta Doğu Teknik Üniversitesi: YÖK Veri Tabanı. p. 125.
60. Arıkan, M., *Esnek imalat sistemlerinde makina yükleme ve çizelgeleme için bir karışık tamsayılı doğrusal programlama modeli*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1997, Gazi Üniversitesi: YÖK Veritabanı. p. 91.
61. Buluş, C., *Esnek imalat sistemlerinde tek parça yapımına yönelik DNC oluşturma ve kalite denetimi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 1997, Marmara Üniversitesi: YÖK Veri Tabanı. p. 219.
62. Tansöker, S., Ertan, *Esnek imalat istemlerinde hücre oluşturma problemine 0-1 tamsayılı programlama ile modelleme yaklaşımı*, in *Fen Bilimleri Enstitüsü İşletme Yönetimi Mühendisliği Bölümü*. 1997, İstanbul Teknik Üniversitesi: YÖK Veri Tabanı. p. 82.
63. Özkan, S., *Esnek imalat hücrelerinde bütünleşik çizelgeleme ve kesici uç işletim sistemi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1997, Bilkent Üniversitesi: YÖK Veri Tabanı. p. 144.
64. Yiğit, V., *Esnek imalat sistemlerive simülasyonla bir uygulaması*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1996, Yıldız Teknik Üniversitesi: YÖK Veri Tabanı. p. 142.
65. Çoban, A., *Esnek imalat sistemlerinde takım yönetimi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1996, Sakarya Üniversitesi: YÖK Veri Tabanı. p. 77.

66. Ekrem, G., Okudan, *Esnek imalat sistemlerinin çizelgelenmesinde simülasyon yaklaşımı*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1995, Yıldız Teknik Üniversitesi: YÖK Veri Tabanı. p. 103.
67. Çelik, G., İlker, *Esnek imalat sistemleri ve alternatif rotaları göz önünde bulundurarak esnek ortamlar için üretim hücrelerinin dizaynı*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1994, İstanbul Teknik Üniversitesi: YÖK Veri Tabanı. p. 152.
68. Atlıhan, M., Kubilayhan, *Esnek imalat sistemlerinde taktik planlama problemlerini çözmek için jenerik bir modelin geliştirilmesi*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 1993, Orta Doğu Teknik Üniversitesi: YÖK Veritabanı. p. 193.
69. Ayyıldız, M., *Esnek üretim sisteminde zeki yükleme ve depolama otomasyonu*, in *Fen Bilimleri Enstitüsü Makina Eğitimi Bölümü*. 2016, Karabük Üniversitesi: YÖK Veritabanı. p. 115.
70. Yabanoba, İ., *Esnek üretim sisteminin gerçek zamanlı uzaktan erişimli kontrolü ve mekatronik eğitime uygulanması*, in *Fen Bilimleri Enstitüsü Elektrik ve Elektronik Mühendisliği Bölümü*. 2011, Sakarya Üniversitesi: YÖK Veri Tabanı. p. 134.
71. Yılmaz, E., *Esnek üretim sistemlerinde hücrelerin yeniden konfigürasyonu*, in *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2010, Çukurova Üniversitesi: YÖK Veritabanı. p. 215.
72. Tekerek, M., *Esnek üretim sisteminde görüntü işleme tekniği ile robotik eğitim modeli geliştirilmesi*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 2010, Gazi Üniversitesi: YÖK Veri Tabanı. p. 188.
73. Gültekin, H., *Robotlu esnek üretim hücrelerinde çizelgeleme*, in *Mühendislik ve Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2006, Bilkent Üniversitesi: YÖK Veri Tabanı. p. 240.
74. Türkcan, A., *Esnek üretim sistemlerinde kontrol edilebilir üretim zamanları ile çizelgeleme üzerine makaleler*, in *Mühendislik ve Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü*. 2003, Bilkent Üniversitesi: YÖK Veri Tabanı. p. 226.
75. Kayır, Y., *Esnek üretim sistemleri için unsur tabanlı otomatik bağlama kalıbı tasarımı ve robotla montajı*, in *Fen Bilimleri Enstitüsü Makina Mühendisliği Bölümü*. 2001, Gazi Üniversitesi: YÖK Veri Tabanı. p. 341.
76. Williams, D.P. *A computer integrated manufacturing cell for use in education and training*. in *Computer Integrated Manufacturing, 1988., International Conference on*. 1988. IEEE.
77. Rao, M., et al. *Framework of computer integrated process systems*. in *Control Applications, 1993., Second IEEE Conference on*. 1993.
78. Sisbot, S., et al., *Yeditepe University computer integrated manufacturing studies*. *Computer Applications in Engineering Education*, 2009. **17**(1): p. 1-12.
79. Schober, *FMS100 Manual A015*. 2003: Festo Didactic GmbH & Co. KG.
80. David, R., *Grafset: A Powerful Tool for Specification of Logic Controllers*. *IEEE Transactions on Control Systems Technology*, 1995. **3**(3): p. 253-268.
81. Wikipedia. *Direct numerical control*. 2013 [cited 2018 07.01]; DNC Interface]. Available from: https://en.wikipedia.org/wiki/Direct_numerical_control.

82. *Basit Kaynak Fikstürü*. [cited 2017 01 Mayıs]; Available from: http://newsroom.lincolnelectric.com/images/9026/media_gallery/Welding_Fixture_resized.jpg.
83. *Gelişmiş Pnömatik Kaynak Fikstürü*. [cited 2017 01 Mayıs]; A pneumatically clamped robot weld fixture for an exercise machine component.]. Available from: <http://www.zamoraengineering.com/robot.php>.
84. *Movemaster Super Reference Manual*. Vol. MEE BFP-A5654-A. 1995: Mitsubishi Electric Corporation.
85. Nica, G. *BMW fabrikasında çalışan akıllı otonom taşıma sistemi*. 2016 18 Kasım [cited 2018 03 Ocak]; Available from: <http://cdn.bmwblog.com/wp-content/uploads/2016/11/BMW-industry-4.0-6.jpg>.

EKLER

FMS100 Bünyesinde Yürütülen Bakım ve Onarım İşleri

Bu bölüm FMS100 bünyesinde şu ana kadar tespit edilip giderilen arıza ve aksaklıkların ileride de tekrar etmesi durumunda, sorunların kolaylıkla çözülebilmesi; FMS100 benzeri sistemlerde ortaya çıkabilecek sorunlar ve giderilme yolları açısından tecrübe paylaşımının sağlanması amacıyla eklenmiştir. Sonuçta bu tez çalışmasının asıl amacı, çalışmadan faydalanmak isteyenlere teknik tecrübe ve bilgi aktararak eğitime katkı sağlamaktır. Bu yönüyle eklenen bu bölüm yapılan bu çalışmaya katkı sağlayacaktır.

CIM Laboratuvarı elektrik tesisatında giderilen sorunlar:

CIM laboratuvarında bulunan bilgisayarların güç kaynakları ve anakartları sık sık arızalanıyordu. Kendi kişisel bilgisayarımın da sabit sürücüsü arızalanınca ve bir keresinde bilgisayardan USB flash hafızayı çıkartırken kıvılcım atlamasının olduğunu da gördükten sonra, laboratuvar elektrik altyapısında ciddi sorunlar olduğu kanaati oluştu. Bunun sonucunda laboratuvar bünyesindeki çeşitli topraklı prizlerde gerilim ölçümü yapıldı. Bunun sonucunda yaklaşık olarak Faz-Nötr arası 210VAC, Faz-Toprak arası 110VAC ve Toprak-Nötr arası 83VAC olarak ölçülmüştür. Sağlıklı durumda bu ölçümlerde Faz-Nötr ve Faz-Toprak arası gerilimlerin birbirine yakın (ideali 220VAC) olması; Toprak-Nötr arası gerilimin ise 3VAC'değerini geçmemesi gerekirdi (ideali 0V olup pirsizde ölçülen değer 83VAC).

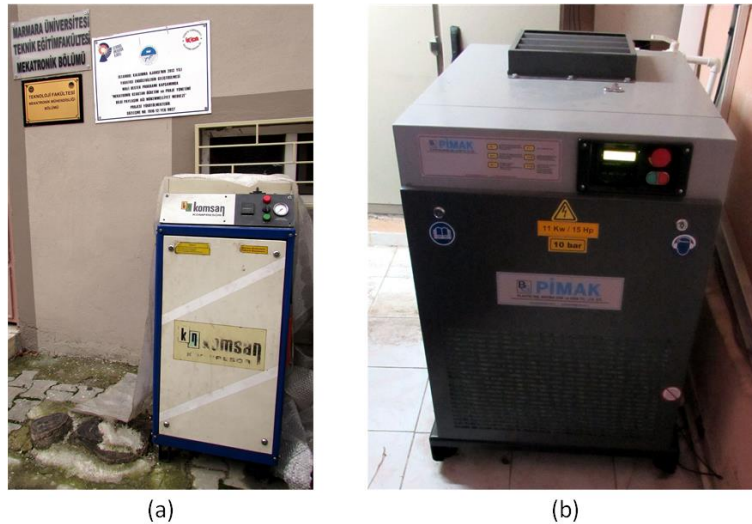
Gerek konunun “uzmanı” diğer öğretim üyelerine, gerekse ihale kapsamında onarım için gelen teknik ekiplere sorunun nedeni sorulduğunda; çoğunlukla binanın topraklamasında sorun olduğu cevabı alınmıştır. İhale kapsamında onarıma gelen teknik ekipler her seferinde binanın ana panosuna topraklama çubukları ekleyerek sorunu gidermeye çalışmışlar fakat sorun bir türlü çözülememiştir.

En sonunda bina ana panosunda yer alan faz, nötr ve toprak baraları arasında tarafımızdan yapılan ölçümlerde, değerlerin sağlıklı olduğu; Toprak-Nötr arasındaki ölçüm değerinin 0.3VAC olduğu tespit edilmiştir. Bu konuda deneyimli bir usta tarafından yapılan tespitte, laboratuvar panosundaki nötr hattının kopuk olduğu, bu durumun sıklıkla görülebildiği; bu nedenle faz geriliminin cihazdan geçtikten sonra nötr hattı yerine toprağa aktığı,

panodaki kaçak akım rölesinin arızalı olduğu için atmadığı bilgisine ulaşılmıştır. Bunun sonucunda sunulan BAP E tipi alt yapı projesi kapsamında, bina ana panosundan laboratuvar panosuna elektrik hattı yeniden çekilmiş, bina dışındaki tek faz 15KVA jeneratör üç faz 110KVA olan ile değiştirilmiş, jeneratör çıkışına üç faz 50KVA kesintisiz güç kaynağı bağlanmıştır. Böylelikle üç faz ile beslenen FMS100 elektrik altyapısı sağlıklı duruma getirilmiştir.

CIM Laboratuvarı basınçlı hava tesisatında giderilen sorunlar:

CIM laboratuvarına basınçlı hava sağlayan kompresör eskimişti. Hava tankı dolduğunda otomatik olarak durması gerekirken durmayıp aşırı ısınıyor ve yangın tehlikesi yaratıyordu. Servis veren firma çağrıldığında kompresörün ömrünü tamamladığı, kumanda devresinin artık düzeltilemeyecek kadar karmaşıklaştığı belirtilmişti. Ayrıca zeminden hava taşıyan borularda kaçaklar vardı. Bunun üzerine BAP E tipi proje kapsamına kompresörün ve hava tesisatının değişimi de eklenmiş ve proje kapsamında değişimler gerçekleşmiştir. Şekil 4-1’de değiştirilen eski kompresör ile yerine gelen yeni kompresör görülmektedir.



Şekil 4-1. Bölüme basınçlı hava sağlayan eski kompresör görüntüsü (a), yerine getirilen yeni kompresör görüntüsü (b).

Profibus FMS kartında giderilen sorunlar:

Lehimleme hücresindeki S7-300 model PLC bir türlü SCADA bilgisayarları ile haberleşemiyordu. Hidrolik-Baskı hücresindeki PLC’de bulunan Profibus FMS kartı ile değiştirilip, haberleşme ayarları yapıldığında, Lehimleme hücresindeki PLC’nin SCADA

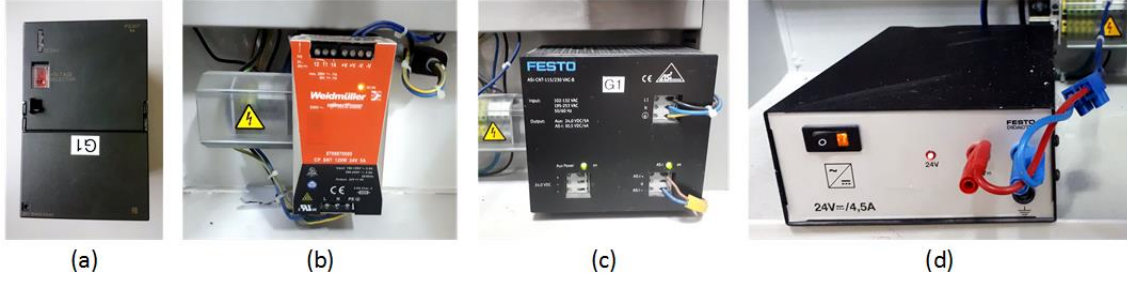
yazılımıyla haberleşebildiği tespit edilmiştir. Arızalı kart detaylı incelendiğinde, elektrik tesisatındaki nötr hattı sorunundan dolayı kartın elektronik devresinde arıza meydana geldiği görülmüştür. Şekil 4-2’de görülmekte olan arızalı kart yenisiyle değiştirilip haberleşme ayarları yapıldıktan sonra Lehimleme hücresinin haberleşme sorunu tamamen giderilmiştir.



Şekil 4-2. Lehimleme hücresinde arızalandığı için yenisiyle değiştirilen Profibus FMS modülü görüntüsü.

İmalat hücrelerindeki güç kaynaklarında giderilen sorunlar:

Elektrik tesisatındaki aksaklıklara karşı en savunmasız ve kolay sorun çıkartan eleman güç kaynaklarıdır. Özellikle günümüzde kullanılan anahtarlama tip güç kaynakları, klasik kaynaklardaki gibi asıl gerilim düşümü işini güç trafosu üzerinden yapmadıkları ve çok fazla elektronik eleman bulundurdıkları için kolaylıkla arızalanabilmektedirler. Lehimleme ve Hidrolik-Baskı imalat hücrelerinde bulunan güç kaynakları yenileriyle değiştirilerek sorunlar giderilmiştir.



Şekil 4-3. Lehimleme hücresinde arızalanan güç kaynağı (a), yerine takılan (b); Hidrolik-Baskı hücresinde arızalanan güç kaynağı (c), yerine takılan (d).

Depolama hücresi Kartezyen konumlama sistemi hatası:

Depolama hücresinde bulunan Kartezyen konumlama sistemi, normal çalışırken ara sıra kendi kendine anlamsız hareketlerde bulunuyor ve arızaya geçiyordu. FMS100'ü üreten firmanın teknik ekipleri farklı zamanlarda gelerek hücre panosunu incelemiş, pano içerisindeki IPC cihazının önce yazılımını yeniden yüklemiş, sonra da kendisini tamamen değiştirmiş, tüm kablo ve sigortaları kontrol etmiş fakat sorunu giderememiştir. Tarafımızdan da kablolar, bağlantı soketleri, sınır algılayıcıları tek tek incelenmiş fakat bir soruna rastlanılmamıştır. Birgün eğitim sırasında, sorunun dersin başladığı ve serin olan sabah saatlerinde görülmeyip, görece daha sıcak olan öğleden sonraları görüldüğü dikkatimizi çekti. Sonrasında panoyu açıp IPC'ye baktığımızda, ana işlemci modülünün aşırı derecede sıcak olduğu tespit edilmiştir. Bu durumun imalat hücresi panosu içinin aşırı sıcak olması ve pano üzerinde herhangi bir havalandırma fanının olmamasından kaynaklandığı kanaatine varılmıştır. IPC ana işlemci modülünün alt tarafına tarafımızdan iki adet bilgisayar güç kaynağı fanı yerleştirildikten sonra sorunun bir daha ortaya çıkmadığı görülmüştür.



Şekil 4-4. Stok hücresi denetleyicisi olan IPC altına yerleştirilmiş soğutma fanları.

Robot tutucu ellerinde görülen hatalar:

FMS100 bünyesindeki Montaj, CNC-Çifti ve Görüntü İşleme hücrelerindeki robotlarda elektrik tahrikli tutucu eller kullanılmaktadır. Montaj ve Görüntü İşleme hücrelerindeki eller Şekil 4-5'deki gibi aynı model olup, CNC-Çifti hücresindeki ele göre daha eski teknolojidir.



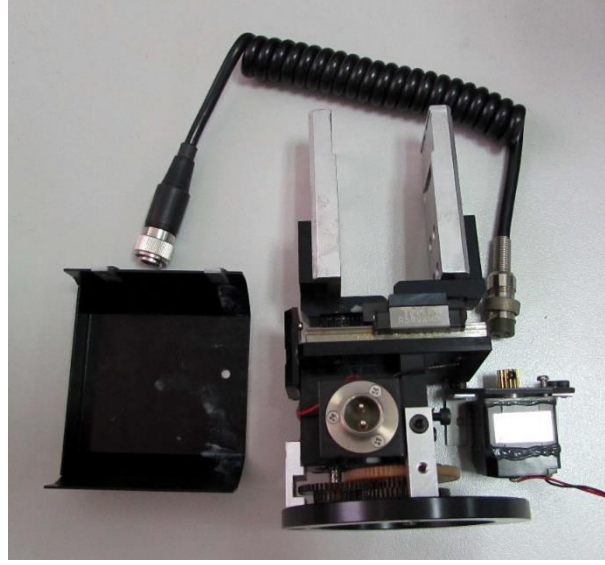
(a)



(b)

Şekil 4-5. Montaj hücresindeki robotta takılı durumdaki tutucu el görüntüsü (a), yerinden sökülmüş durumdaki tutucu el görüntüsü (b).

Robotların hafıza pillerinin bitmesi sonucunda sıfır noktalarını unutmalarını önlemek amacıyla, robotlar 7/24 çalışır durumda tutuldukları için; bu tutucu eller de sürekli enerjili ve çalışır durumdaydı. Montaj hücresindeki elin çalışmayı durdurması, Görüntü İşleme hücresindeki elin ise arada tutukluk yapması nedeniyle, mekanik arıza şüphesiyle, tutucu eller tarafımızdan sökülerek Şekil 4-6'de görüldüğü gibi içleri açılmıştır.

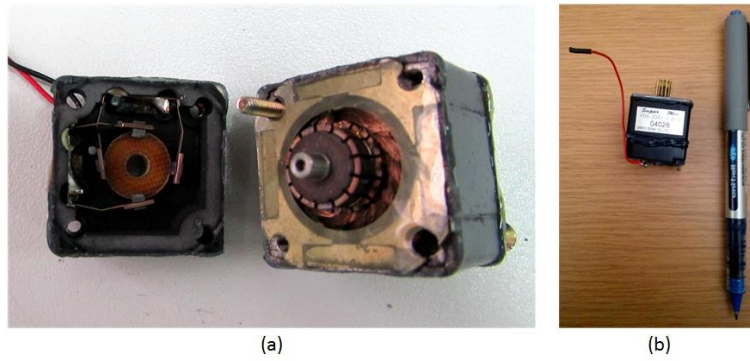


Şekil 4-6. 1E-HM01 model robotik tutucu elin sökülüş haldeki görüntüsü.

Bunun sonucunda ellerin tutma hareketini sağlayan motor ile doğrusal dişliler arasındaki gres yağının tamamen kuruduğu, yağın yoğun bir kıvama gelerek motorun hareketini daha da zorlaştırdığı görülmüştür. Her iki el de yeniden yağlandığında, tutukluk yapan eldeki sorunun giderildiği; fakat diğer elin yine de çalışmadığı tespit edilmiştir.

Tutucu ellerin açık ve kapalı durumda kalabilmeleri için, içlerindeki tahrik elemanlarının (bu örnekte sabit kutuplu fırçalı DC motor) sürekli enerjili ve çalışır durumda olmaları gerekmektedir. Bu durum tahrik elemanlarının ömrünün kısılmasına neden olmaktadır.

Arızalı eldeki sorunun DC motordan kaynaklandığı, motor içini açınca da Şekil 4-7’de görüldüğü gibi kömürlerinin bitmiş olduğu tespit edilmiştir.



Şekil 4-7. 1E-HM01 model robotik tutucu ele güç veren DC motorun iç görüntüsü (a); tutucu elin boyutlarını ifade eden dış görüntüsü (b).

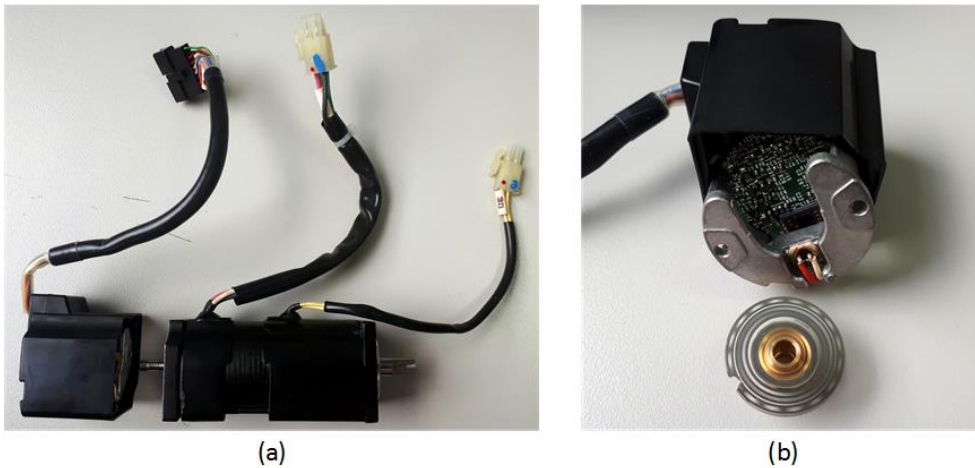
Tüm aramalarımıza rağmen arızalı motorun yenisi bulunamadığı için, arızalı el komple yenisiyle değiştirilmiştir. Ayrıca hatanın yeninden ortaya çıkmaması için, robotlar uzun

süre açık kalacaksa, tutucu ellerin enerji fişleri çekili durumda bırakılmaları, böylelikle ellerin içinde bulunan motorların ömürlerinin uzatılması sağlanmaktadır.

CNC-Çifti hücreesindeki robotun üçüncü ekseninde görülen sorunlar:

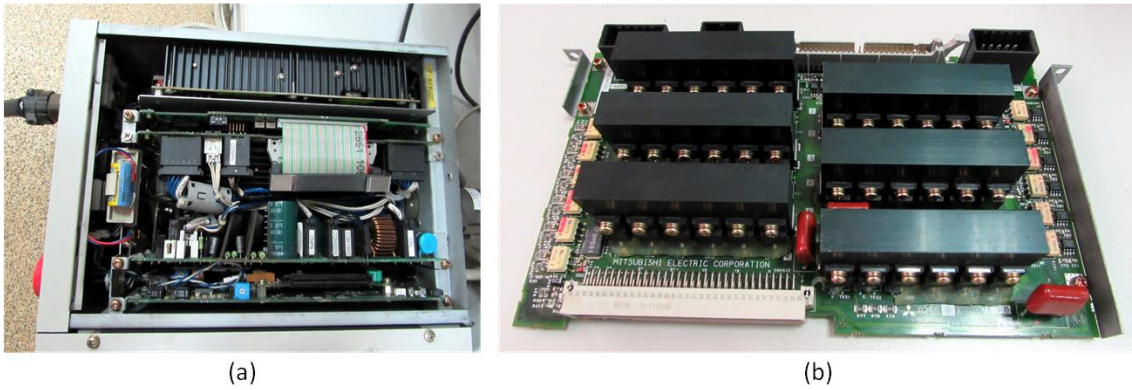
CNC-Çifti hücreesindeki robot hareket ettirilmek istendiğinde, robotun tuhaf mekanik sesler çıkararak H0963 numaralı 3. eksen aşırı yük (overload position error) arızası verdiği ve çalışmadığı görülmüştür. Arızanın kaynağını bulmak için yaptığımız incelemede, aşırı yük arızasının 3. eksen motorunun aşırı akım çekmek istemesinden kaynaklandığı tespit edilmiştir. Eksen hareket ettirilmek istendiğinde çıkan tuhaf mekanik sesler nedeniyle önce arızanın 3. eksen pulley dişlisi veya gergi rulmanları ile kayıştan kaynaklandığı düşünülmüştür. Robot 3. eksen koruma kapağı açıldığında, kayışlarda ve gergi rulmanlarında bir sorun olmadığı görülmüştür. Yine de robotun 3. eksen motoruna ait pulley dişlisi ve kayışı yenisiyle değiştirilmiştir.

3. eksen motorunu test etmek için, benzer elektriksel özelliklerdeki ve çalışır durumdaki 4. eksen motoru sökülerek arızalı motorun yerine takılmış, aynı tuhaf mekanik seslerin yine geldiği görülmüştür. Arızanın kaynağını bulmak için 3. eksen servo motoru Şekil 4-8 (a)'daki gibi sökülüştür. Konuyla ilgili daha detaylı inceleme yapıldığında, servo motorun arka tarafındaki optik enkoder diski üzerinde, Şekil 4-8 (b)'de görüldüğü gibi, bir kırık meydana geldiği ve parça koptuğu tespit edilmiştir. Bunun üzerine motorun yenisi sipariş edilmiştir.



Şekil 4-8. CNC-Çifti hücreesindeki robotun 3. eksen servo motoru ve enkoder bölümü (a), enkoder bölümü içerisindeki kırık optik disk ve elektronik okuyucu devresi (b).

Fakat gelen yeni motor da CR1 model robot denetleyiciye bağlanıp çalıştırılmak istendiğinde aynı tuhaf mekanik seslerin yine geldiği görülmüştür. Arızayı tespit için robot denetleyici Şekil 4-9 (a)'da görüldüğü gibi açılmış ve içerisindeki servo sürücü kartı çıkartılmıştır. Şekil 4-9 (b)'de görülmekte olan kart, CNC laboratuvarındaki başka bir robota ait aynı model CR1 denetleyicideki kart ile değiştirildiğinde, hatanın ortadan kalktığı görülmüştür. Bunun üzerine sürücü kartın 3. eksen ile ilgili bölümünün arızalandığı tespit edilmiştir. Üniversite tarafından desteklenen E tipi BAP projesi sayesinde arızalı sürücü kartının yenisi alınmış, kart değiştirilmiş ve sorun giderilmiştir.



Şekil 4-9. CR1 robot denetleyici iç görüntüsü (a) ve arızalı servo sürücü kartı (b).

Tüm bu çalışmanın sonucunda robotun 3. eksen motorunun çarpma, sonucu oluşan mekanik zorlanma sırasında hasar gördüğü, enkoder diskinde parça kopmalı kırık oluştuğu; bu durumun da motorun aşırı akım çekmesine ve sürücü kartındaki IGBT elemanlara zarar verdiği kanaatine varılmıştır.

Robotlar ve CNC tezgahları için yapılan bakım çalışmaları:

CIM ve CNC laboratuvarlarındaki robotlar ve CNC tezgahlarının bakımları ile arızalarının giderilmesi için E tipi BAP proje başvurusu yapılmış ve üniversite tarafından onaylanmıştır. Proje desteği sayesinde dışarıdan yapılan hizmet alımı kapsamında CIM laboratuvarındaki tüm robotların eklemleri yağlanmış, kontrolörlerinin içerisi temizlenmiştir. FMS100 Görüntü İşleme hücresindeki robotun eklemleri yağlanırken çekilmiş bazı görüntüler Şekil 4-10'deki gibidir.



Şekil 4-10. FMS100 Görüntü İşleme hücresindeki robotun eklemleri yağlanırken çekilmiş görüntüler.

Tüm imalat hücrelerindeki robotların hafıza pilleri 2 kere değiştirilmiş, robot eksenleri gres ile yağlanmış, hafıza pillerinin zamansız bitmesi sebebiyle robot eksenlerinin hafızadan silinen sıfır noktaları mekanik durdurucu yöntemiyle yeniden öğretilmiş, tüm robotlarda bulunan ve FMS100 tarafından kullanılan programlara ait konumlar yeniden öğretilmiş, robot programlarında güncellemeler yapılmıştır.

Benzer şekilde CNC-Çifti hücresindeki tezgahların içindeki talaşlar temizlenmiş ve eksen yatakları yağlanarak bakımları yapılmıştır.

Robotik havya için yapılan bakım ve güncelleme çalışmaları:

FMS100 Lehimleme hücresinde robotik havya ünitesi ve havya ucu temizleme ünitesi bulunmaktadır. Havya ünitesinin üzerinde motorlu lehim besleme ünitesi, havya kafasını X ve Z eksenlerinde hareket ettiren selenoidler bulunmaktadır. Tüm bu sistem özel olarak tasarlanmış, bünyesinde S7-200 PLC ve TD200 panel bulunan bir kontrolör tarafından işletilmektedir. Tez çalışması kapsamında havya ucu temizleme ünitesi, lehim besleme ünitesi, selenoidler gibi parçaların bakımları yapılmış, işler duruma getirilmiş; Şekil 4-11’de görüldüğü gibi çarpma nedeniyle kırılmış olan havya ünitesi değiştirilmiş; ayrıca S7-200 PLC yazılımına müdahale edilerek Almanca olan ekran yazılımı Türkçeleştirilmiştir.



(a)



(b)

Şekil 4-11. Çarpma nedeniyle kırılmış havya ünitesi görüntüsü (a), havya kontrolörü içindeki PLC'nin yazılımını değiştirmek için sökülmüş halinin görüntüsü (b).

ÖZGEÇMİŞ

Barış DOĞAN 1978 yılında İzmir’de doğdu. Lise eğitimini İzmir Çınarlı Anadolu Meslek Lisesi’nde tamamlayarak 1996 yılında mezun oldu. 1997 yılında girdiği Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Anabilimdalı Bilgisayar ve Kontrol Eğitimi Programı’ndan 2002 yılında mezun oldu. Mezuniyetten sonra Milli Eğitim Bakanlığına bağlı Kadıköy Anadolu Ticaret ve Ticaret Meslek Lisesi’nde Bilgisayar Teknik Öğretmeni olarak çalışmaya başladı. Aynı yıl Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Bilgisayar Anabilimdalı, Bilgisayar ve Kontrol Eğitimi Programı’nda Yüksek Lisans eğitime başladı. Teknik öğretmen olarak çalışırken, yüksek lisans eğitimi sırasında, 2003 ile 2006 yılları arasında toplam 3 dönem Marmara Üniversitesi Teknik Bilimler Meslek Yüksek Okulu bünyesindeki Bilgisayar Teknolojisi ve Programlama bölümünde ücretli öğretim görevlisi olarak çalıştı. 2004 yılında işlerine ve eğitimine 1 yıl ara vererek askerlik görevini Yedek Subay olarak tamamladı. Askerlik dönüşü tekrar eski görev yerinde göreve başladı. 2006 yılı, Haziran ayında Marmara Üniversitesi Teknik Eğitim Fakültesi Mekatronik Eğitimi Bölümü’ne Araştırma Görevlisi olarak atandı. 2012 yılında Bilim Sanayi ve Teknoloji Bakanlığı’nın Teknogirişim Sermaye Desteği programına kabul edildi, hibe aldı ve ARGE şirketi kurdu. 2015 yılında TÜBİTAK 2239 programı kapsamında A.B.D. Silikon Vadisi’nde girişimcilik, uluslararası ticaret ve ürün geliştirme konularında 1 ay eğitim aldı. Aynı yıl Marmara Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği bölümüne dolu kadro aktarımı ile geçti. 15 Temmuz 2016’dan sonra Marmara Üniversitesi Senatosu kararıyla öğretim görevlisi olarak görevlendirildi. 2016-2017 yıllarında 2 dönem boyunca ders verdi. Hala aynı görev yerinde çalışmalarına devam etmektedir.