



Bu proje Avrupa Birliđi ve Türkiye Cumhuriyeti tarafından finanse edilmektedir.

ROBOLAB

Robot Laboratuvarının Kurulması
ve Sanayi İşbirliđi ile Uygulaması,
Robot Eđitimlerinin Verilmesi
Cumhuriyet Üniversitesi Mühendislik Fakültesi



- Robot Programlama Dili
- Robot Benzetim
- Robotik Sistemlerde Çevre Donanım
- Robotik Sistemlerde Güvenlik
- Robotik Sistemlerde Mekanik Bakım
- Robotik Sistemlerde Elektrik Bakım

ROBOT EĐİTİMLERİ

DERS NOTLARI



İNSAN KAYNAKLARININ
GELİŞTİRİLMESİ
PROGRAM OTORİTESİ



ÇSGB
T.C. ÇALIŞMA VE
SOSYAL GÜVENLİK
BAKANLIđI



CUMHURİYET ÜNİVERSİTESİ



KAMU İ. - ÇAMŞAĞI

"Bu yayın Avrupa Birliđi ve Türkiye Cumhuriyeti'nin mali katkısıyla hazırlanmıştır. Bu yayının içeriğinden yalnızca Cumhuriyet Üniversitesi Mühendislik Fakültesi sorumludur ve bu içerik hiçbir şekilde Avrupa Birliđi veya Türkiye Cumhuriyeti'nin görüş ve tutumunu yansıtmamaktadır."

Endüstriyel Robotların Kullanımı Ve Programlanması

Hedef Grup: Orta ve Yüksek Öğrenim Öğrencileri



İçindekiler

İçindekiler	1
1 Robot Teknolojisine Giriş	7
1.1 Genel Bakış	7
1.2 Giriş	7
1.3 R.U.R. - Rossum'un Üniversal Robotu	7
1.4 Robot Kanunları	8
1.5 İlk Robot	9
2 Endüstriyel Robotların Kullanım Alanları	10
2.1 Genel Bakış	10
2.2 Endüstriyel Robot Kullanımı	10
2.3 Robot Uygulamaları Örnekleri	13
3 Bir Robot Sisteminin Bileşenlerine Genel Bakış	24
3.1 Genel Bakış	24
3.2 Bir Robot Hücresinin Bileşenleri	24
3.3 Robot Seçimi	25
3.4 Kumanda Konfigürasyonu	26
3.5 Efektör/Alet Seçimi	27
3.6 Enerji Beslemesi Seçimi	27
3.7 Çevre Birimleri Bağlantısı (Alan Bus)	28
3.8 Sensörlerin Kullanımı	29
3.9 Güvenlik Tertibatları	29
4 Endüstriyel Robot	34
4.1 Genel Bakış	34
4.2 Robotik Alanına Giriş	34
4.3 Tanım ve Yapı	35
4.4 Bir Robotun Mekanik Düzeni	36
4.5 Ana Aksların Konumu	38
4.6 Mutlak Hassasiyet ve Tekrarlama Hassasiyeti	40
5 Robot Kumandası	42
5.1 Genel Bakış	42
5.2 Robot Kontrol Ünitesi Ölçüleri	44
5.3 Robot Kontrol Ünitesi Asgari Mesafeleri	44
5.4 Robot Kontrol Sistemine Genel Bakış	45
5.5 Uygulamaya ve Veri Yolu Sistemlerine Genel Bakış	46
5.5.1 KUKA Controller Veri Yolu, KCB	48
5.5.2 KUKA Ssistem Veri Yolu, KSB	49
5.5.3 KUKA Extension Bus, KEB	50

5.5.4 KUKA Line Interface, KLI	51
5.6 Enerji Verimliliği	52
6 Robotu Hareket Ettirme	53
6.1 Genel Bakış	53
6.2 Programlama El Cihazı smartPAD	53
6.2.1 Ön Taraf	53
6.2.2 Arka Taraf	55
6.3 Robot Kumandasının Mesajlarını Okuma ve Yorumlama	56
6.4 İşletim Türünü Seçme Ayarlama	58
6.5 Robot Akslarını Tek Tek Hareket Ettirme	60
6.6 Robotla İlişkili Koordinat Sistemleri	63
6.7 Robotu Dünya Koordinat Sisteminde Hareket Ettirme	65
6.8 Robotu Tool Koordinat Sisteminde Hareket Ettirme	69
6.9 Robotu Base Koordinat Sisteminde Hareket Ettirme	73
7 İşleme Alma	77
7.1 Genel Bakış	77
7.2 Ayar Prensipleri	77
7.3 Robotu Ayarlama	80
7.4 Robotta Yükler	82
7.5 Alet Yük Verileri	82
7.6 Robotta Ek Yükler	83
7.7 Bir Aleti Ölçme	85
7.8 Bazı (Base) Ölçümleme	92
7.9 Güncel Robot Pozisyonunun Sorgulama	96
8 Robot Programları Yürütme	99
8.1 Genel Bakış	99
8.2 İlkendirme Sürüşü Yapma	99
8.3 Robot Programlarını Seçme ve Başlatma	100
9 Program Dosyalarının Kullanımı	107
9.1 Genel Bakış	107
9.2 Program Modülleri Oluşturma	107
9.3 Program Modülleri İşleme	108
10 Programlanmış Hareketleri Oluşturma ve Değiştirme	110
10.1 Genel Bakış	110
10.2 Yeni Hareket Komutları Oluşturma	110
10.3 Çevrim Süresi En Uygun Hareketler Oluşturma (Aks Hareketi)	112
10.4 Rota Hareketleri Oluşturma	118
10.5 Hareket Komutlarını Değiştirme	125

11 Robot Programında Mantıksal Fonksiyonları Kullanma	129
11.1 Genel Bakış	129
11.2 Mantık Programlamasına Giriş	129
11.3 Bekleme Fonksiyonlarını Programlama	130
11.4 Basit Anahtarlama Fonksiyonlarını Programlama	134
11.5 Yörünge Anahtarlama Fonksiyonlarını Programlama	136
12 Uzman Düzeyine Giriş	140
12.1 Genel Bakış	140
12.2 Uzman Düzlemini Kullanma	140
13 Schleifen, Bedingte Anweisungen und Fallunterscheidungen	144
13.1 Genel Bakış	144
13.2 Program Akışının Kontrolü	144
13.3 Taşlama	144
13.4 Koşullu Talimatlar ve Durum Analizleri	147
14 Alt Programlar ve Fonksiyonlar	150
14.1 Genel Bakış	150
14.2 Lokal Alt Programlarıyla Çalışma	150
14.3 Global Alt Programlarıyla Çalışma	152
14.4 Parametreleri Alt Programlara Aktarm	154
15 Değişkenler ve Kararlaştırmalar	155
15.1 Genel Bakış	155
15.2 KRL'de Veri Yönetimi	155
15.3 Basit Veri Tipleriyle Çalışma	157
15.3.1 Değişkenlerin Bildirimi	157
15.3.2 Basit Veri Tipleri Olan Değişkenleri İlkendirme	160
15.3.3 KRL İle Basit Veri Tipli Değişken Değerlerinin Manipülasyonu	161
15.4 KRL İle Arraylar / Alanlar Mit KRL	165
15.5 KRL İle Yapılar	168
16 KRL İle Hareket Programlama	172
16.1 Genel Bakış	172
16.2 KRL İle Hareketler Programlama	172
16.3 Robot Pozisyonlarını Hesaplama Veya Manipüle Etme	180
16.4 Status ve Turn Bitlerini Bilinçli Değiştirme	181
17 Bir Üst Düzey Kontrol Sistemiyle Çalışma	186
17.1 Genel Bakış	186
17.2 PLC'^den Program Başlatma İçin Hazırlık	186
17.3 PLC Bağlantısını Uyarlama (Cell.src)	187
17.4 Harici Otomatik Konfigürasyonu ve Kullanılması	189

18 WorkVisual İle Programlama	198
18.1 Genel Bakış	198
18.2 WorkVisual İle Bağlantı	198
18.3 WorkVisual İle Proje Yönetimi	205
18.3.1 WorkVisual İle Proje Açma	205
18.3.2 Projeleri WorkVisual İle Karşılaştırma	210
18.3.3 Projeyi Robot Kumandasına Aktarma (Yükleme)	214
18.3.4 Projeyi Robot Kumanda Sisteminde Etkinleştirme	217
18.4 KRL Programlarını WorkVisual İle İşleme	220
18.4.1 Dosya Kullanımı	220
18.4.2 KRLEditörü Kullanımı	226
19 Ek	232
19.1 Kısaltmalar	232
19.2 Kullanılan Kavramlar	233
19.3 Güvenlik KR C4'ten Alıntı	234
İndeks	240

1 Robot Teknolojisine Giriş

1.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Giriş
- R.U.R. - Rossum'un Üniversal Robotu
- Robot kanunları
- İlk robot

1.2 Giriş



Resim 1-1

Artan ürün ve tür çeşitliliği nedeniyle rekabet edilebilirliğin korunması ve artırılması için başka özelliklerin yanı sıra üretimde verimlilik ve esneklik artırılmalıdır. Bunun için gerekli olan otomasyonda endüstriyel robotların (IR) kullanılması uygun bir çözümdür.

“Robot” kelimesi, ağır iş anlamına gelen Slav “robota” sözcüğünden gelmektedir.

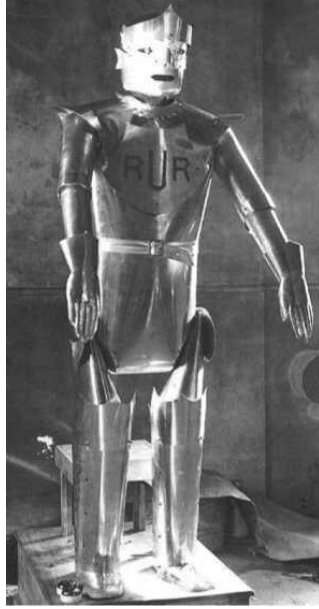
Ancak teknik açıdan endüstriyel robotlar tanımlarına göre başka otomasyon araçlarından ve iş makinelerinden ayrılmaktadır. Ancak bu konuda uluslararası çapta bir kavram kargaşası söz konusudur, çünkü manipülatörler veya yerleştirme cihazları gibi benzer sistemler de robotlar arasında sayılmakta ve istatistiklerde sık sık belirtilmektedir.

Bunun nedeni, tüm sistemlerde mekanik yapının, sabit parçalı kinematik bir zincirden ve kavrayıcı veya aletli (örn. kaynak tabancası) bir elin monte edilmiş olduğu bir koldan (veya birden fazla koldan) oluşmasıdır.

1.3 R.U.R. - Rossum'un Üniversal Robotu

R.U.R. (Çekçe: Rosumovi Umeli Roboti), Çek yazar Karel Capek'in 1921 yılında yayınlamış olan bir tiyatro oyununun adıdır.

Oyun, insanların işini kolaylaştırmak için insanlara benzeyen makineler (robotlar) üreten bir firmayı konu almaktadır. Oyunun ilerleyen bölümlerinde bu makineler insanlara karşı devrim başlatır ve insanlığı yok eder.



Resim 1-2: R.U.R. - Rossum'un Universal Robotu

Tiyatro oyununun adı oyunda bu makineleri üreten firmanın adı olan R.U.R, Rossum'un Üniversal Robotu'ndan gelmektedir. Rossum adı, yazarın ironik bir kinayesidir: Çekçe sözcük rozum ("rosumm" gibi okunur, ancak ilk hece daha kısa telaffuz edilir), akıl, zeka anlamına gelmektedir. Orijinal adın doğru tercümesi "Alemşümul Suni Adamlar Fabrikası"dır, Rossum ve üniversal kelimeleri sadece Çekçe kısaltma R.U.R.'u korumak amacıyla kullanılmıştır.

"Robot" kelimesi, bu tiyatro oyununun ardından kısa süre sonra birçok ülkenin günlük diline yerleşmiştir. [8]

1.4 Robot Kanunları

Robot kanunları, Isaac Asimov tarafından I, Robot (1950) kitabında (derleme bilimkurgu hikayeleri) açıklanmıştır ve o zamandan beri bir robotun ne ve nasıl olması gerektiğine dair algıyı önemli ölçüde etkilemektedir. Asimov tarafından açıklanan robotlar, davranışları ve kararları bakımından bu kanunlara bağlıdır.

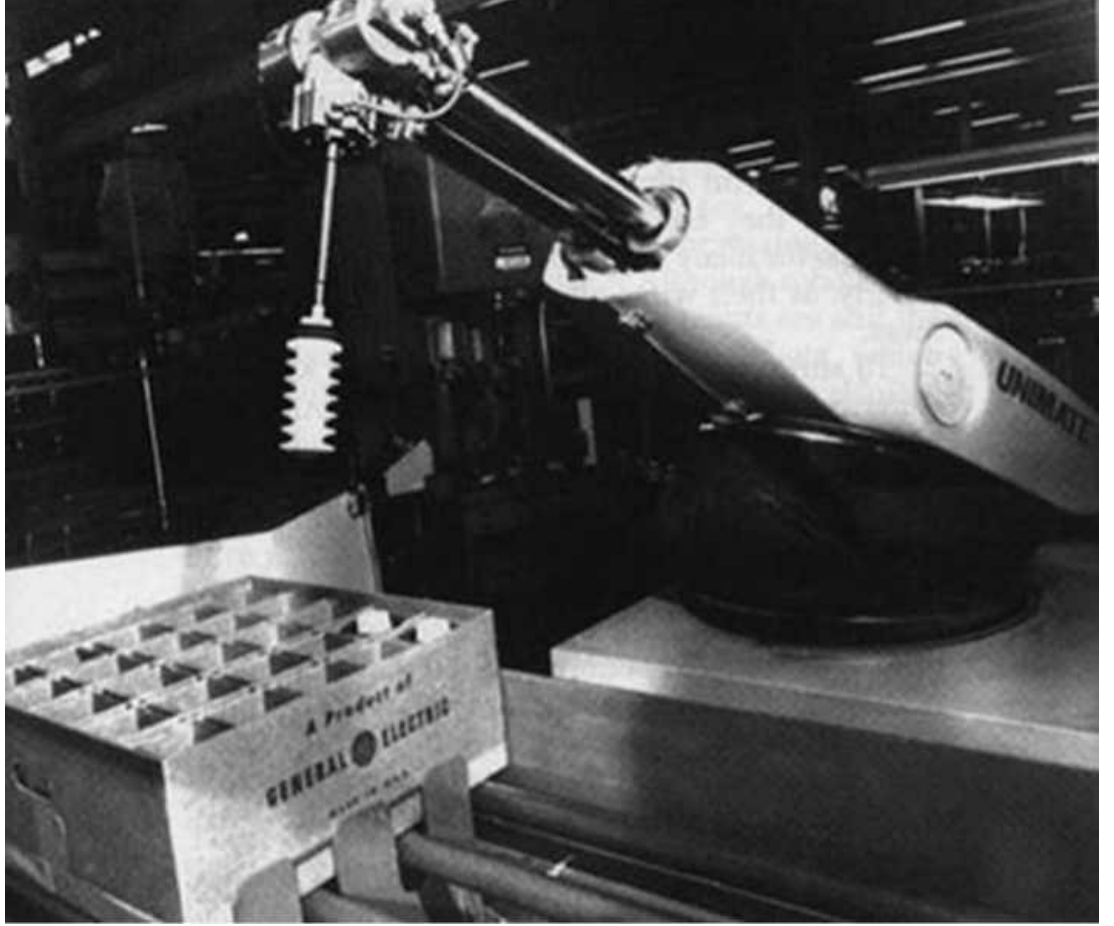
Bu kanunlar ilk etapta "literatürde" yer alan robotlar için geçerlidir, ancak günümüzün robotlarının programlamasını etkilemiştir ve türetilmiş bir biçimde olmak üzere örneğin temizleme robotlarının rekabetlerinde geçerlidir. Günümüzün robot programlayıcılarından çoğu bunu bilmesede günümüzün endüstriyel robotları da bu kanunlara göre programlanmaktadır.

Asimov kanunları aşağıdaki gibidir (orijinal metin):

1. Bir robot hiçbir şekilde insanoğluna zarar veremez veya pasif kalmak suretiyle zarar görmesine izin veremez.
2. Bir robot kendisine insanlar tarafından verilen komutlara birinci kuralla çelişmediği sürece itaat etmek zorundadır.
3. Bir robot birinci veya ikinci kuralla çelişmediği sürece kendi varlığını korumak zorundadır.

Bu kanunlar hiyerarşik bir yapıya sahiptir. Kanunlar çok açık olsa da "kusursuz değildir", özellikle insanlar tarafından insancıl bir şekilde ve böylece eksik dikkate alındıkları için. [8]

1.5 İlk Robot



Resim 1-3: İlk endüstriyel robot (Tip: Unimate, Üretici: Unimation, İşletime alma: 1961)

İlk endüstriyel robot (daha sonra Unimate olarak adlandırılmıştır), mucit olan George Devol ve Joseph Engelberger'in 1956 yılında bir buluşmada bir bilimkurgu romanı hakkında konuşmaları ile ilk defa ortaya çıkmıştır. Bu iki mucit roman nedeniyle gerçek bir robot geliştirmeye karar vermiştir.

Unimate, 1962 yılında General Motors'da (Trenton, USA) bir konveyöre entegre edilmiştir. Bu robotun görevi, bir metal pres makinesinden sıcak iş parçalarını alıp istiflemektir. Robotun programı, bir manyetik tambur üzerine kaydedilen, çok sayıda münferit adım talimatından oluşmaktaydı. Böylece çeşitli otomasyon görevlerini yerine getirme kabiliyetine sahipti.

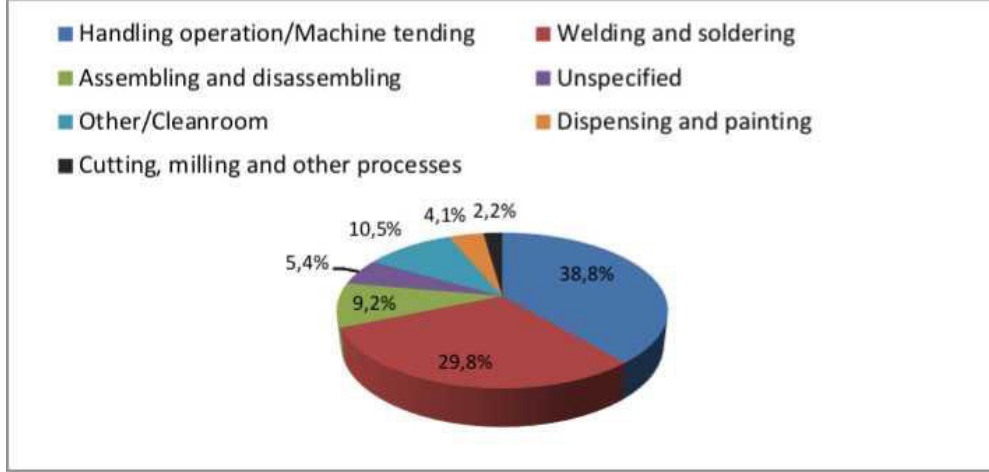
2 Endüstriyel Robotların Kullanım Alanları

2.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Robot teknolojisine yönelik istatistikler
- Uygulama örnekleri

2.2 Endüstriyel Robot Kullanımı



Resim 2-1: Robot uygulamaları

Lejant:

[Kaynak: IFR World Robotics 2011]

Alet Kullanımı Alet kullanımı aşağıdakileri kapsamaktadır:

- Nokta kaynak için robotlar
- Rotalı kaynak için robotlar
- Kaplama işlemleri için robotlar
- Çapak giderme işlemleri için robotlar

İşlenen Parça Elleçlemesi İşlenen parça elleçlemesi aşağıdakileri kapsamaktadır:

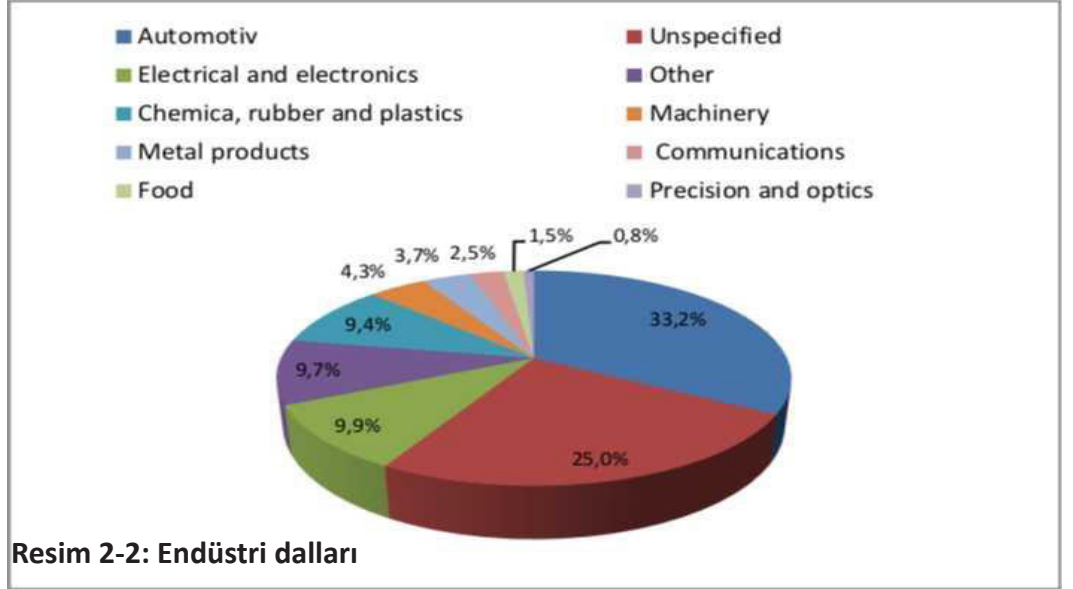
- Sabit aletlerin altında işlenen parça elleçlemesi için robotlar (sabit nokta kaynak tabancası, perçin zımbası, kaynak tabancası).
- Takım tezgahlarının ve işleme merkezlerinin beslemesi için robotlar.
- Preslere, basınç ve enjeksiyonlu kalıplama makinelerine yükleme yapmak ve bunları boşaltmak için robotlar.
- Dövme, tavlama ve sertleştirme tesislerinde ve cam ürün üretiminde işlenen parçaların elleçlemesi için robotlar.

Montaj

Montaj:

- Otomobil endüstrisinde tekerleklerin, açılır
- Ekranların montajında kullanılan robotlar.

Endüstride Kullanılan Robotlar



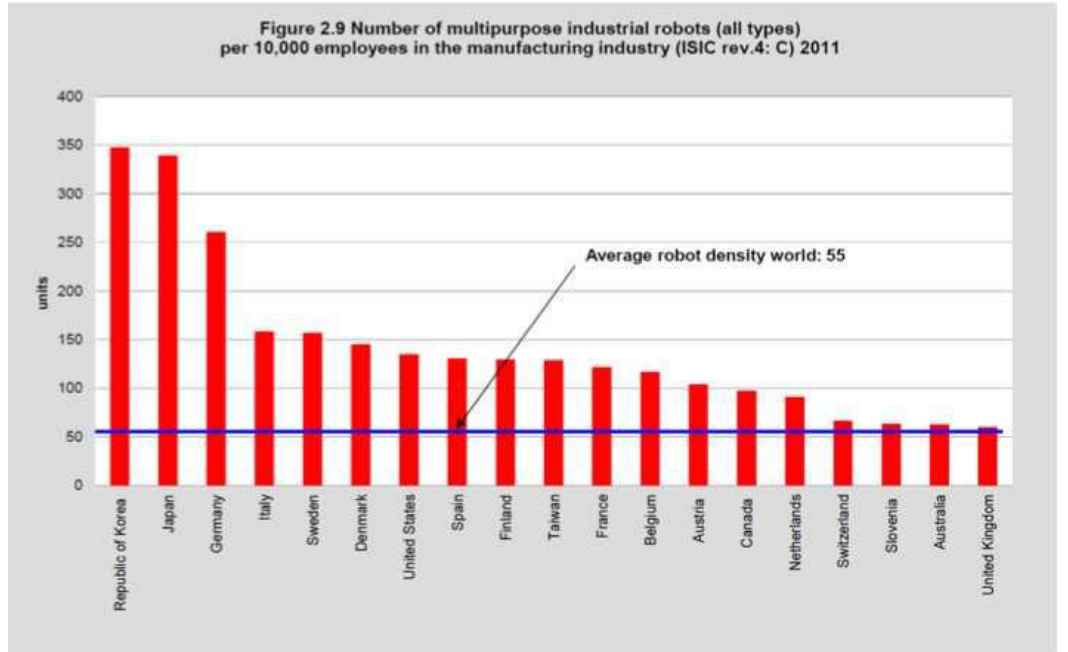
Lejant:

[Kaynak: IFR World Robotics 2011]

Robot yoğunluğu 2010

Üretimde 10.000 çalışan başına robot sayısı, [Kaynak: IFR World Robotics 2011]

Ortalamanın üzerinde bir robot yoğunluğuna sahip ülkeler (51)



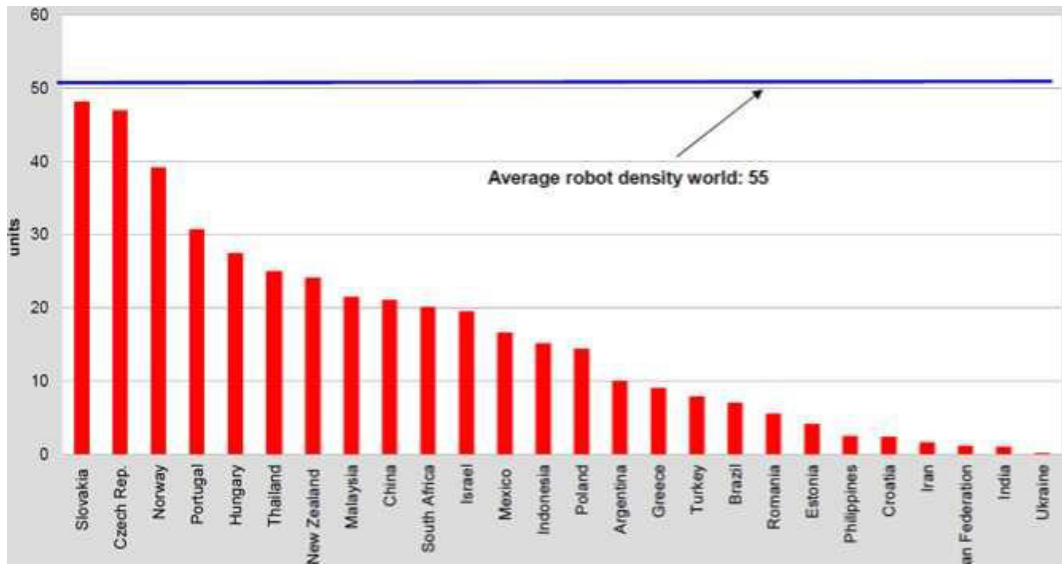
Resim 2-3: Robot yoğunluğu 1

Lejant:

Kaynak: IFR World Robotics 2012

Ortalamanın altında bir robot yoğunluğuna sahip ülkeler (51)

Figure 2.10 Number of multipurpose industrial robots (all types) per 10,000 employees in the manufacturing industry (ISIC rev.4: C) 2011



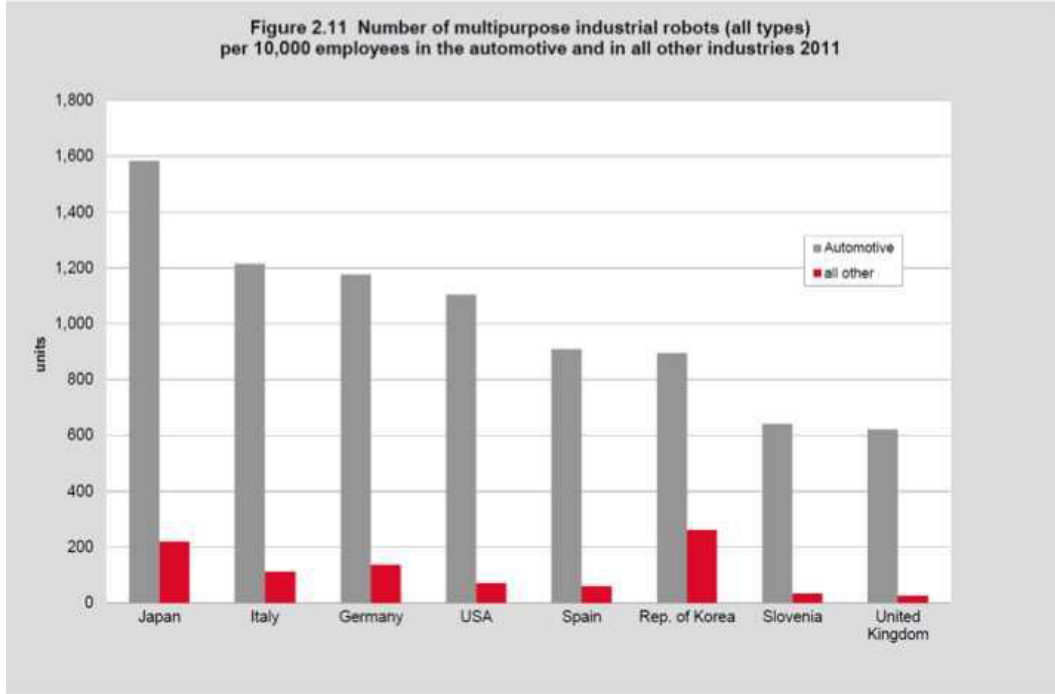
Resim 2-4: Robot yoğunluğu 2

Lejant:

Kaynak: IFR World Robotics 2012

1

Otomobil endüstrisinde ülkelere göre sıralanmış robot yoğunluğu



Resim 2-5: Otomobil endüstrisinde robot yoğunluğu

Lejant:

Kaynak: IFR World Robotics 2012

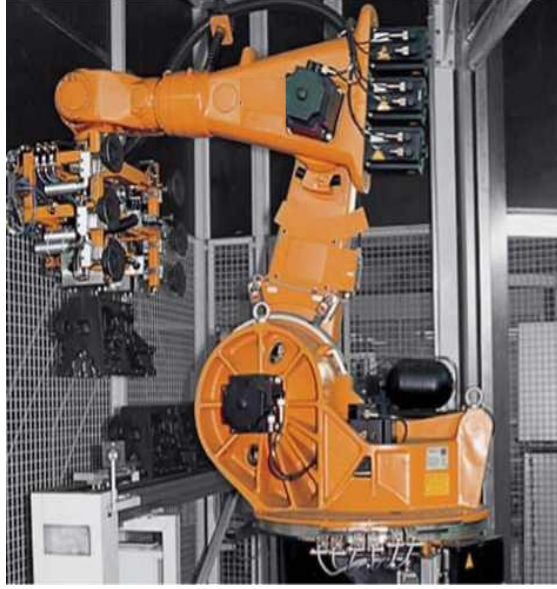
2.3 Robot Uygulamaları Örnekleri



Resim 2-6: Bir kaportanın nokta kaynak işlemi



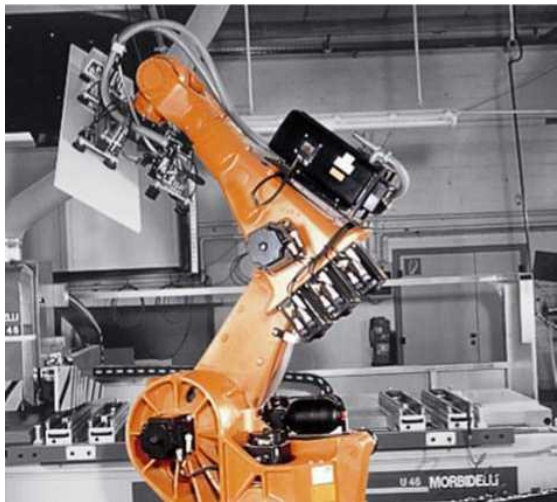
Resim 2-7: Koruma gazı ile kaynak



Resim 2-8: Motor bloklarının elleçlenmesi



Resim 2-9: Bira fıçılarının elleçlenmesi



Resim 2-10: Mobilyaların elleçlenmesi



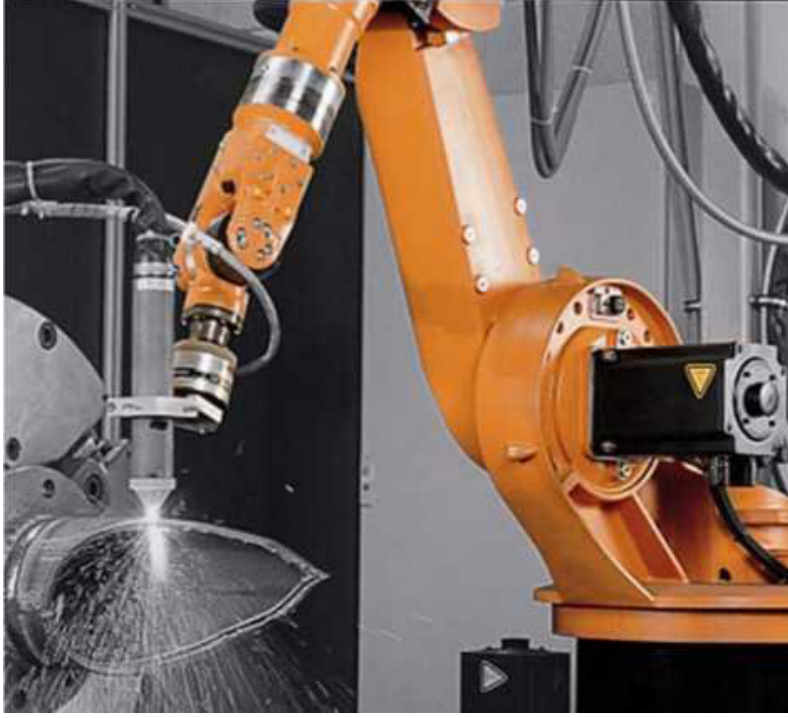
Resim 2-11: Sac parçaları elleçleme - Sacları bükme



Resim 2-12: Sac parçaları elleçleme - Pres zincirleyiciler



Resim 2-13: İşleme - Taşlama ve parlatma



Resim 2-14: İşleme - döküm parçalarının plazma kesimi



Resim 2-15: İşleme - etin kesilmesi



Resim 2-16: Gıda ürünlerinin işlenmesi



Resim 2-17: Montaj - lastik contaların birleştirilmesi



Resim 2-18: Otomobil koltuklarının montajı



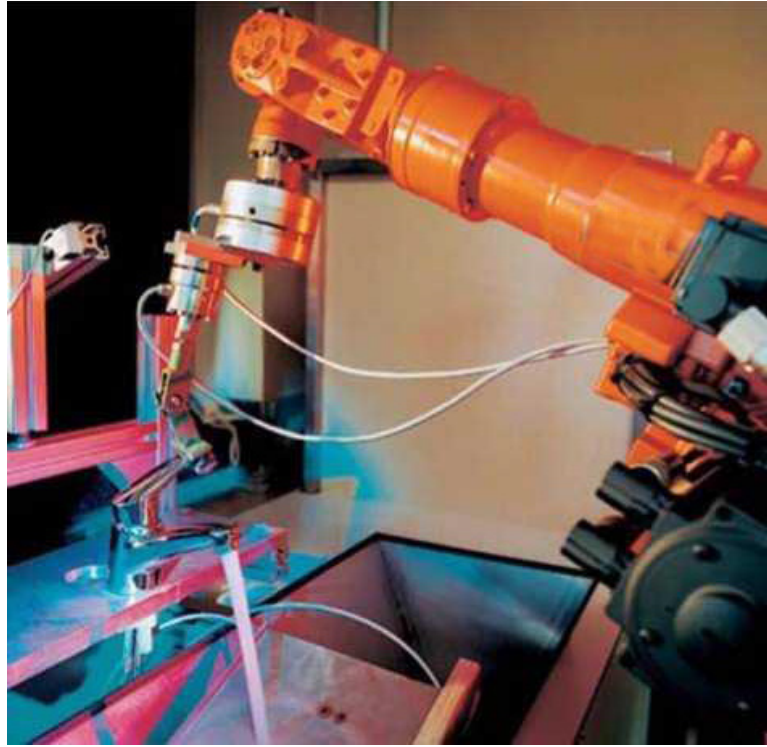
Resim 2-19: Kartonların paletlenmesi



Resim 2-20: Kimyasal madde uvallarının paletlenmesi



Resim 2-21: Meşrubat kasalarını paletleme



Resim 2-22: Su armatürlerinin kontrolü



Resim 2-23: Kontrol - Otomobil koltuklarının sürekli kontrolü



Resim 2-24: Ölçüm - Lazer inline ölçüm



Resim 2-25: Rüzgar kanalında ölçüm - Ayrıntı



Resim 2-26: Rüzgar kanalında ölçüm



Resim 2-27: Entertainment - ROBOCOASTER



Resim 2-28: Tıp teknolojileri

3 Bir Robot Sisteminin Bileşenlerine Genel Bakış

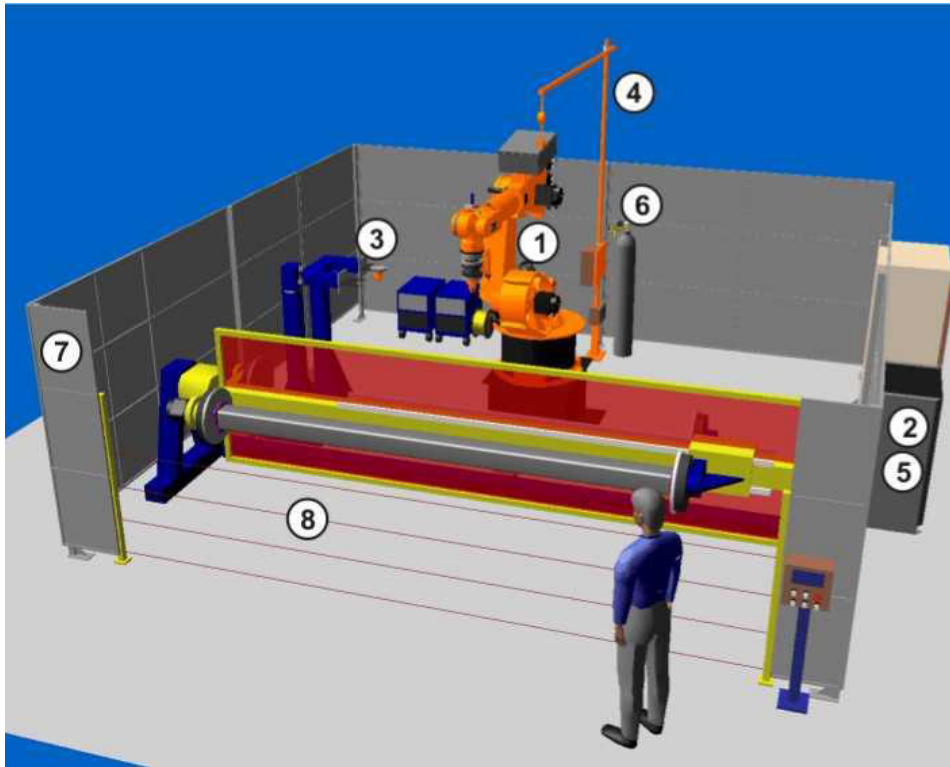
3.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Bir robot hücresinin bileşenleri
- Robot için seçim ölçütleri
- Robot akslarının ve ilave aksların kumandası
- Alet seçimi
- Enerji beslemesi seçimi
- Çevre birimlerin bağlantısı
- Sensörlerin kullanımı
- Güvenlik tertibatları

3.2 Bir Robot Hücresinin Bileşenleri

Bir robot sistemi/robot hücresi normalde aşağıdaki bileşenlerden oluşmaktadır



Resim 3-1: Koruyucu gaz kaynak hücresi

Poz.	Açıklama
1	Robot
2	Kumanda sistemleri
3	Alet/alet değiştiricisi
4	Enerji beslemesi
5	Çevre birimlerin bağlantısı
6	Sensor sistemi
7	Korkuluk
8	Işık perdesi kullanıldığında yerleştirme bölgesi

3.3 Robot Seçimi

Robot seçimine yönelik kriterler:

Yükler

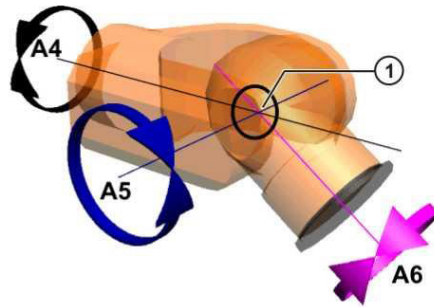
- **Yük:** Yük bir kütle, atalet momenti ve robot tarafından uygulanan statik ve dinamik kuvvetlerden oluşan bir fonksiyondur.
- **Nominal yük:** Normal koşullar altında (sıcaklık, hava nemi ...) ve herhangi bir performans özelliğinde azalma olmadan robot flanşına etki edebilen azami yük.
- **İlave yük:** Robot tarafından nominal yüke ilave olarak taşınabilen yük. Robot koluna, bağlantı koluna ve/veya döner tablaya takılır.

Kullanım koşulları

- **Uygulama:** Üretici, robot için öngörülen temel kullanım türünü/türlerini belirler.
Tipik uygulama durumları örnekleri:
 - Elleçleme
 - Montaj
 - Nokta kaynak
 - Rotalı kaynak
 - Yapıştırma maddelerinin/sızdırmazlık maddelerinin uygulanması
 - Malzemelerin işlenmesi (frezeleme)
- **Normal koşullar (ortam):** (EN ISO 9946) Üretici, belirtilen performans özelliklerinin yerine getirilebildiği ortam koşullarının sınır değerlerini belirtir. Bu sınır değerler, depolama ve işletimin hasarsız olması için mevcut olmalıdır. Ortam koşulları bunlarla sınırlı olmamakla birlikte aşağıdakileri kapsamaktadır:
 - Sıcaklık °C
 - Bağıl hava nemi (%)
 - Sınır yükseklik konumu (m)
 - Elektromanyetik parazit
- Tekrarlanabilirlik hassasiyeti ve mutlak hassasiyet

Alana yönelik bilgiler

- **Çalışma alanı:** Kullanıcı programı tarafından belirtilen tüm hareketler kapalı olduğunda gerçekten kullanılan sınırlı alan kısmı. Alanın büyüklüğü, el kök noktası ile sınırlanır.



Resim 3-2: Bilek dibi noktası

Aşağıdakilerde farklı çalışma alanları:

- Standart robotlar

- Tavan robotu
- Konsol robotları
- Paletleme robotları
- SCARA Robotları
- Lineer akslı robot
- **Dış ölçüler ve kütle:** Konstrüksiyon için ölçü değerleri (mm) ve robot ağırlığı önem taşır (örn. taşıma/yerleştirme/değiştirme).
- **Temel montaj yüzeyi:** Güvenli bir işletimin mümkün olmasını sağlamak amacıyla gerekli olan, robotun ana çerçevesini monte etmek için temel montaj yüzeyinin tanımı.
- **Mekanik arabirim:** Standarda göre ölçülü çizim (ISO 9409-1)

Robot hızı

- Takt süresine ulaşmak için azami tek aks hızı ve azami nominal hat hızı bilgileri mevcut olmalıdır.

Takt süresi tespiti

- **Tahmin:** Bir taslak veya CAD çizimi yardımıyla takt süresini tahmin etmek için kesin olmayan, ancak hızlı bir şekilde uygulanabilen seçenek. Sonuç, deneyim değerlerine de bağlıdır.
- **Simülasyon:** Robot hücresinin planlanması ve sürecin **KUKA**.Sim veya başka bir robot simülasyon yazılımı (Robcad (em-Workplace)) ile simülasyonu. Çok kesin sonuçlar elde edilebilir.
- **Deneme hazırlığı:** İlave olarak süreç parametrelerinin (örn. yapıştırma hızı) optimize edilmesini sağlayan ve bir kalite kontrolüne imkan veren yüksek maliyetli seçenek (malzeme, süre).

Güvenlik

Robot, güncel standart uyarınca olmalıdır (DIN EN ISO 10218-1). Güvenlik için hangi güvenlik elemanlarının kullanılacağına (sistem güvenliği veya kişilerin güvenliği) seçimi.

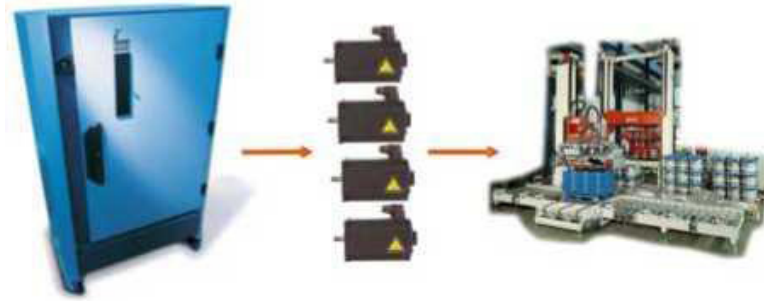
3.4 Kumanda Konfigürasyonu

KRC kumandası, her türlü taşıma kapasitesi sınıfındaki robot için kesintisiz bir kumanda ve tahrik konsepti sunmaktadır.



Resim 3-3: (V)KR C4 aks regülasyonu

Ancak KRC kumandası ve motorlar aracılığıyla yabancı kinematikler de kontrol edilebilir. Bunun için KMC - Motion Control tanımı kullanılır.



Resim 3-4: Yabancı kinematik ile KMC

3.5 Efektör/Alet Seçimi

Modern dünyada efektör kavramı, teknik alanında da kendine bir yer edinmiştir. Burada aktör kavramı ile aynı anlamda kullanılmaktadır. Robot teknolojisinde kavrayıcılar, ölçüm araçları, aletler ve programa göre çalışma alanında hareket eden ve robotun ortamını manipüle etmek için robota hizmet eden diğer işlem elemanları, uç efektörleri olarak tanımlanır.

Her alt için Tool Center Point (TCP) adı verilen bir alet merkez noktası tanımlanır. Robotun kumandası, bütün spesifikasyonlar TCP'nin hareketleri ile ilgili olacak şekilde yapılandırılabilir.

Olası robot aletleri:

- Kavrayıcı (çene kavrayıcı, vakum kavrayıcı)
- Kaynak tabancası
- Kaynak tabancası
- Boyama memesi
- Yapıştırma memesi
- Su jeti kafası
- Lazer kaynak/kesme optiği
- Delme/frezeleme kafası
- Vidalayıcı
- Kesme aleti (testere, bıçak)
- Ölçüm sensörleri

3.6 Enerji Beslemesi Seçimi

Robot elindeki efektör (alet) için enerji ve kumanda sinyalleri (örn. basınçlı hava, kavrayıcı için sinyaller vs.) gereklidir. Enerjinin türü, aletin türüne göre değişir. Enerji beslemesi için temel olarak iki seçenek vardır:

- robottan bağımsız harici enerji beslemesi
- robotta entegre enerji beslemesi

Harici Enerji Beslemesi

Harici enerji beslemesinde enerji, robottan bağımsız olarak kollar veya benzer tertibatlar ve asılı hortum veya kablo paketleri üzerinden beslenir. Ancak bu enerji beslemesi türü beraberinde robot hareketleri esnasında, özellikle işleme alma aşamasında bir yerde asılı kalması halinde hortum ve kablo paketinin hasar görmesi riskini getirir. Ayrıca normalde büyük hortum ve kablo paketleri gereklidir.

- Veri yolu sistemleri
 - PROFINET
 - PROFIBUS
 - INTERBUS
 - ETHERNET IP
 - Can-Bus/DeviceNet
 - Ethernet

3.8 Sensörlerin Kullanımı

Sensör

Sensör veya ölçüm sensörü, teknoloji alanında belirli fiziksel veya kimyasal özelliklerin yanı sıra (örn.: ısı radyasyon, sıcaklık, nem, basınç, parlaklık, manyetiklik, hızlanma, kuvvet) ortamının madde özelliklerini de kalite bakımından veya ölçüm değeri olarak miktar bakımından tespit edebilen bir parçadır.

Sensörler, teknoloji alanında otomatik süreçlerde sinyal vericileri olarak önemli bir rol oynarlar. Sensörler tarafından tespit edilen değerler veya durumlar, çoğu zaman elektriksel olarak güçlendirilmiş bir şekilde ilgili kumanda sisteminde işlenir ve kumanda sistemi de buna bağlı olarak ilave adımlar tetikler.

Robot hücreesindeki veya robottaki sensör sistemi, robotun ortamının özelliklerini tespit eder:

- Nesnelerin durumu (örn. açık veya kapalı)
- Engellerle çarpışma
- Teknik süreçte fiziksel değerler (örn. kuvvet)
- Pozisyon işaretlerinin ve nesnelerin konumu
- Nesnelerin konturu
- Ortam resimleri (piksel resimler)

3.9 Güvenlik Tertibatları

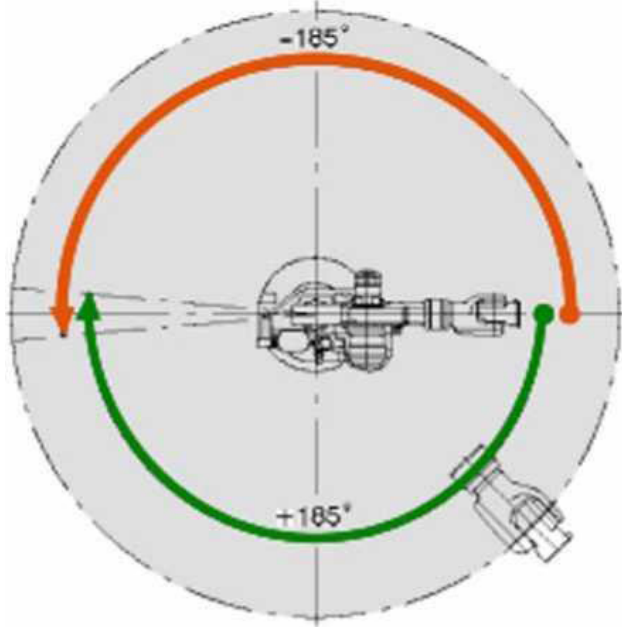
Endüstriyel robotlar kullanıldığında, insan ve robot arasında her zaman temas olur. Doğrudan temas normalde işleme alma aşaması ve ayarlama ile bakım çalışmalarıyla sınırlıdır. Üretim aşamasında insanların ve robotların çalışma alanları sıkı bir şekilde birbirinden ayrılmıştır. Ancak fiziksel bir temasın mümkün olduğu kısa süre içerisinde de insanların hatalı davranışları nedeniyle sık sık ağır kazalar meydana gelmektedir.

Robotlarda mümkün olan kişi ve/veya sistem koruma sistemleri:

- Yazılım son şalteri
- Mekanik son tahditler
- Çalışma alanlarında alan denetimi
- Safe Robot teknolojisi
- Korkuluk
- Harici güvenlik sensörleri
 - Emniyet şalteri
 - Işık perdesi
 - Emniyet matları
 - Lazer tarayıcı

$$\text{\$SOFTN_END}[1] = -185^\circ$$

Yazılım
Limit Şalteri



$$\text{\$SOFTN_END}[1] = -185^\circ$$

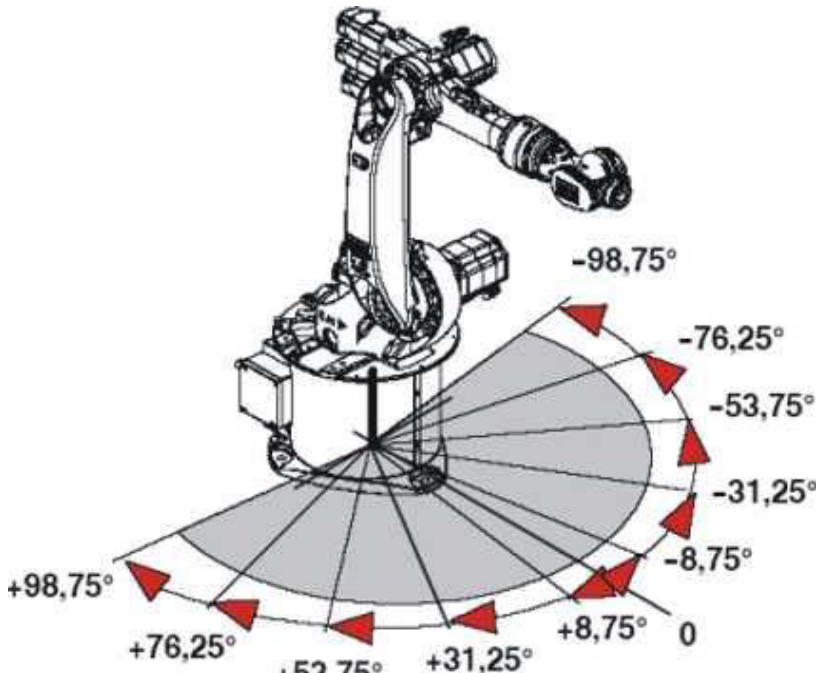
Resim 3-8: Yazılım son şalteri, pozitif/negatif bölge

Robotun çalışma alanı tüm akslarda yazılım son şalteri üzerinden sınırlanır. 1, 2, 3 ve 5 akslarının çalışma alanları mekanik olarak tampon görevi gören son tahditler üzerinden sınırlanır.



Yazılım son şalterleri sadece sistem koruması için kullanılabilir.

Mekanik
Son
Dayanaklar



Resim 3-9: Çalışma bölgesi sınırlaması, eksen 1



Resim 3-10: Uç tahdidi, eksen 1

Aksesuar olarak çalışma alanı sınırlamaları sunulmaktadır. Bu sınırlamalar, 1 ile 3 arasındaki akslar için ilgili çalışma alanını yapılan işleme bağlı olarak sınırlamak üzere mekanik tahditler olarak teslim edilir.

i Mekanik çalışma alanı sınırlamaları, kişileri korumak ve sistemi korumak için kullanılabilir

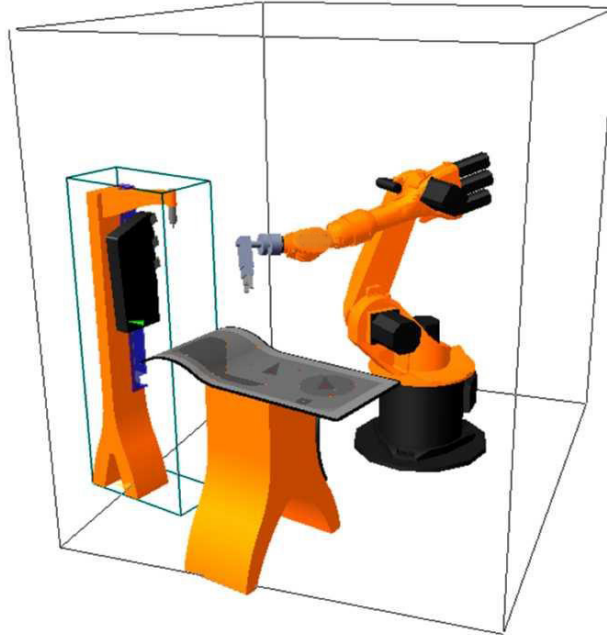
Çalışma Alanlarında Alan Denetimi

Bu çalışma alanları kullanıldığında, sistem yazılımında standart çalışma alanları ile SafeRobot teknolojisinin çalışma alanları arasında ayırım yapılmalıdır:

Robotların çalışma alanları

Standart çalışma alanları

- En fazla 8 kübik veya aksa özgü çalışma alanı otomatik olarak denetlenebilir. Bu çalışma alanları, daha karmaşık biçimler oluşturmak üzere üst üste de binebilir.



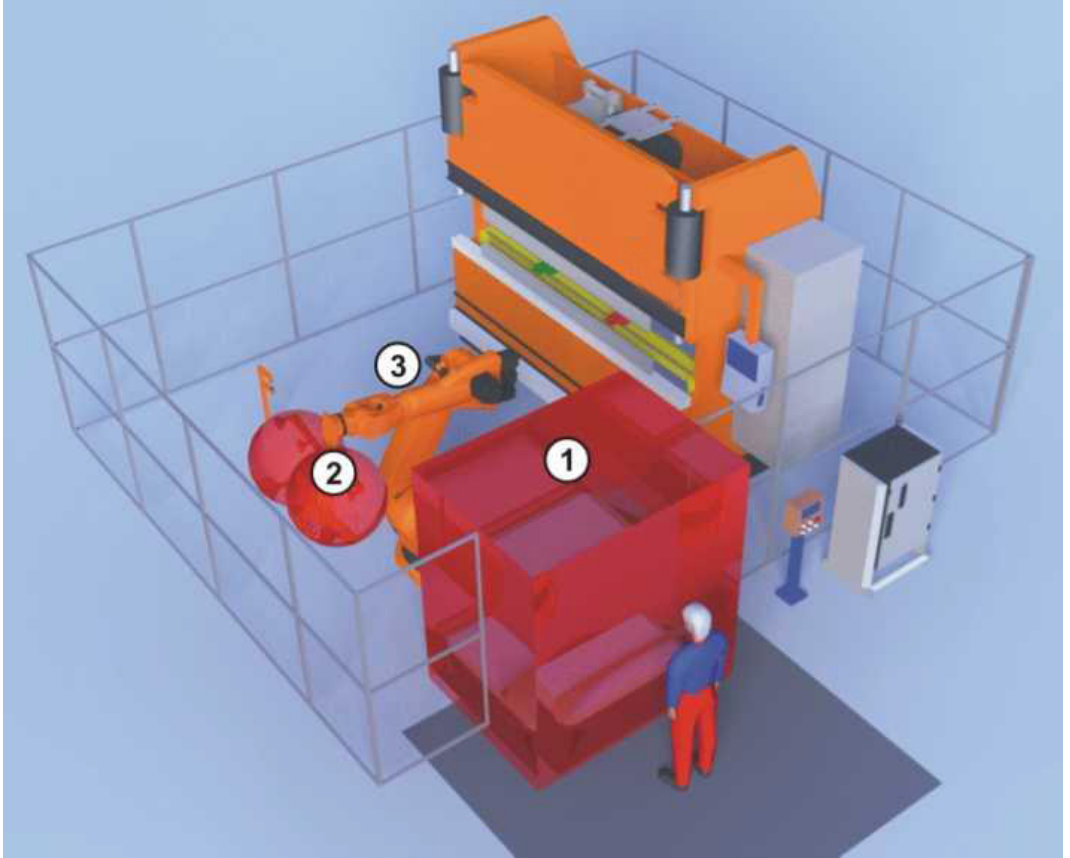
Resim 3-11: Kartezyen iş mekanları örneği

- Tanımlanmış olan bu çalışma alanlarından birisi ihlal edildiğinde, kumanda sistemi, önceden atanmış olan bir çıkış sağlar. Sağlanan çıkış sinyali daha sonra KRL programı veya harici bir ana bilgisayar tarafından işlenebilir. Burada robot durdurulabilir ve bir arıza iletisi verilebilir.

i Standart çalışma alanları (standart yazılımdan) sistem koruması için kullanılamaz.

Kişiler için güvenli çalışma alanları

- 16 adet serbest yapılandırılabilir (kartezyen veya aksa özgü) denetim alanı (çalışma alanları veya koruma alanları)



Resim 3-12: Bir kartezyen koruma alanı örneği

1 Koruma alanı

2 Alettaki küreler

3 Robot

- 16 adet serbest yapılandırılabilir alet - en fazla 6 küreye kadar olmak üzere modellenilebilir.
- SafeOperation güvenlik fonksiyonları Kategori 3 ve performans seviyesi d (EN ISO 13849- 1:2007 uyarınca) taleplerine uygundur. Bu, EN 62061 uyarınca SIL 2'ye eşittir

i SafeOperation altında planlanan ve test edilen çalışma alanları, kişilerin korunması veya sistemin korunması için kullanılabilir.

Koruyucu Korkuluklar



Resim 3-13: Korkuluk arızalı

Koruyucu tertibat gereksinimleri, ek l'deki **Makine Yönergesi**'nde madde 1.4 altında da ayrıntılı olarak açıklanmıştır.

Ayıran ve ayırmayan koruyucu tertibatlar

- sağlam bir yapıya sahip olmalıdır, yani darbelere ve vuruşlara karşı dayanıklı olmalıdır,
- güvenli bir şekilde doğru pozisyonda tutulmalıdır,
- ek tehlikelere neden olmamalıdır, yani kapanma sonucu ezilme tehlikesi, radyasyon tehlikesi,
- atlanmamalı veya etkisiz hale getirilmemelidir,
- ...

Ayrıca ayırıcı koruyucu tertibatlar, eğer mümkünse, malzemelerin ve maddelerin dışarı fırlamasına veya düşmesine ve makine tarafından üretilen emisyonlara karşı koruma sağlamalıdır.



Diğer bilgileri ilgili normlarda ve yönetmeliklerde bulabilirsiniz.

Harici Güvenlik Sensörleri

Kumanda personelinin robotun hareket alanında bulunması kaçınılmazsa (örn. yapı parçaları yüklerken), tehlike bölgesi, bir emniyet plakası, bir ışık perdesi veya bir lazer tarayıcı ile emniyete alınmalıdır. Koruyucu kapıları veya kapakları emniyete almak için genelde emniyet şalterleri kullanılır.



Diğer bilgileri ilgili normlarda ve yönetmeliklerde bulabilirsiniz.

4 Endüstriyel Robot

4.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Robot nedir?
- Robotun yapısı
- Ana aksların konumu
- Mutlak hassasiyet ve tekraralama hassasiyeti

4.2 Robotik Alanına Giriş

Robot Nedir? *Robot terimi ağır iş anlamına gelen Slav robota sözcüğünden gelmektedir.*

Bir endüstriyel robotun resmi tanımı: "Bir robot serbest programlanabilen, program kontrollü bir elleçleme cihazıdır."

Yani robot tanımına, kumanda ve kumanda cihazı ve bunların bağlantı kabloları ve yazılım da dahildir.



Resim 4-1: Endüstriyel robotlar

- 1 Kumanda ünitesi (kumanda kabini (V)KR C4)waw
- 2 Manipülatör (robot mekanizmaları)
- 3 Kumanda ve programlama için el cihazı (smartPAD)

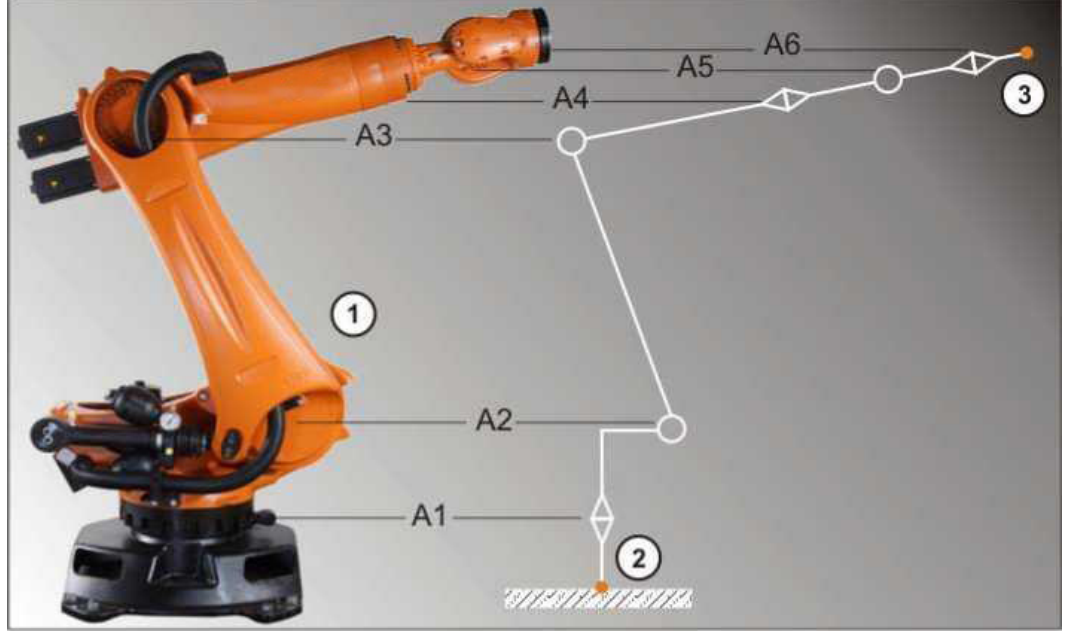
Endüstriyel robotun sistem sınırları dışındaki her şey çevre birimleri olarak adlandırılır:

- Aletler (efektör/Tool)
- Koruyucu düzenek
- Taşıma bantları

- Sensörler
- Makineler
- vs.

4.3 Tanım ve Yapı

Robotlar özellikleri nedeniyle "Esnek otomatik tertibatlar" sınıfına atanır.



Resim 4-2: Manipülator

- 1 Manipülator (robot mekanizmaları)
 - 2 Kinematik zincirin başı: Robot ayağı (ROBROOT)
 - 3 Kinematik zincirin serbest ucu: Flanş (FLANGE)
- A1 Robot aksları 1 ila 6
- ...
- A6

VDI Yönergesi 2860

VDI Yönergesi 2860 uyarınca tanım:

Robotlar, birden fazla aksı olan ve hareket sırası ve yollar veya açılar bakımın

dan hareketleri serbest bir şekilde (yani mekanik müdahale olmadan) programlanabilen ve bazı durumlarda sensör kontrollü olan, universal olarak kullanılabilen otomatik hareket tertibatlarıdır. Robotlar kavrayıcıları aletler veya başka üretim araçları ile donatılabilirler ve elleçleme ve/veya üretim işlemleri uygulayabilirler.

Avrupa standardı EN 10218-1 (eskiden EN 775) uyarınca tanım:

Robot/endüstriyel robot, üç veya daha fazla aksta programlanabilen ve otomasyon teknolojisinde kullanım için sabit bir yere takarak ya da hareketli olarak kullanılabilen, otomatik olarak kumanda edilen, serbest olarak programlanabilen çok amaçlı bir manipulatördür.

Robot aşağıdakileri içermektedir:

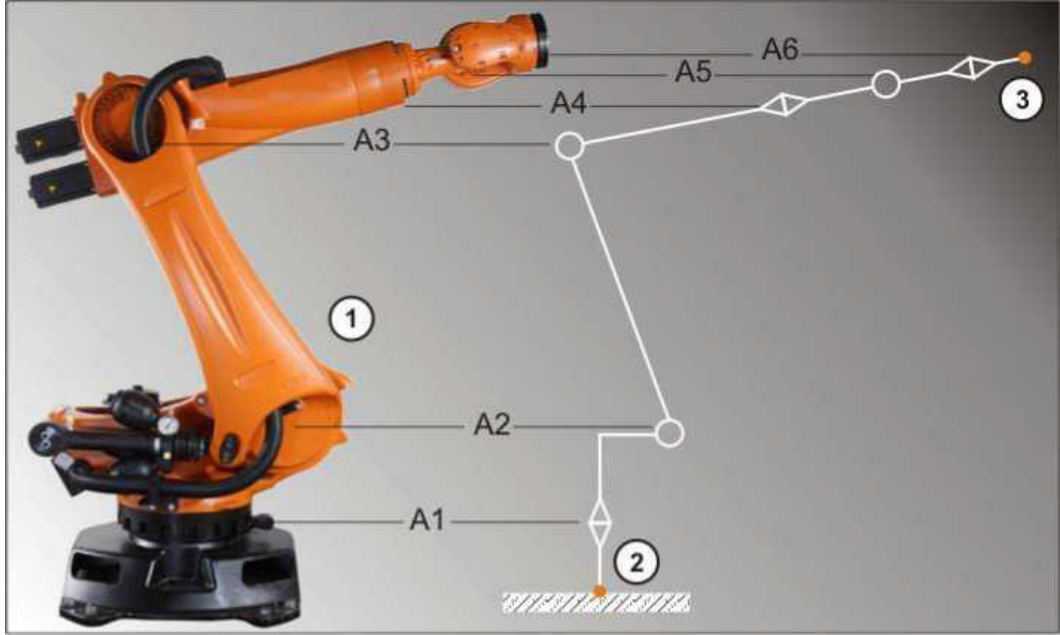
- manipülator (aktüatörler dahil)
- programlama el cihazı ve her iletişim arabirimi (donanım ve yazılım) dahil olmak üzere kontrol ünitesi

Bu, robot kumandası tarafından kontrol edilen tüm ilave aksları kapsamaktadır.

4.4 Bir Robotun Mekanik Düzeni

Manipülâtör Nedir?

Manipülâtör, robotun asıl mekanik düzenidir. Belirli sayıda hareketli, birbirine zincirlenen halkalardan (akslar) oluşur. Kinematik bir zincir olarak da bilinir.



Resim 4-3: Manipülâtör

- 1 Manipülâtör (robot mekanizmaları)
- 2 Kinematik zincirin başı: Robot ayağı (ROBROOT)
- 3 Kinematik zincirin serbest ucu: Flanş (FLANGE)
- A1 Robot aksları 1 ila 6
- ...
- A6

Her bir aksın hareketi, servo motorların hedefe yönelik regülasyonu ile sağlanır. Bunlar redüksiyon dişlileri üzerinden manipülâtörün münferit bileşenlerine bağlıdır.



Resim 4-4: Robot mekanik sistemi komponentlerine genel bakış

- 1 Ana kasa
- 4 Bağlantı kolu

2 Döner tabla

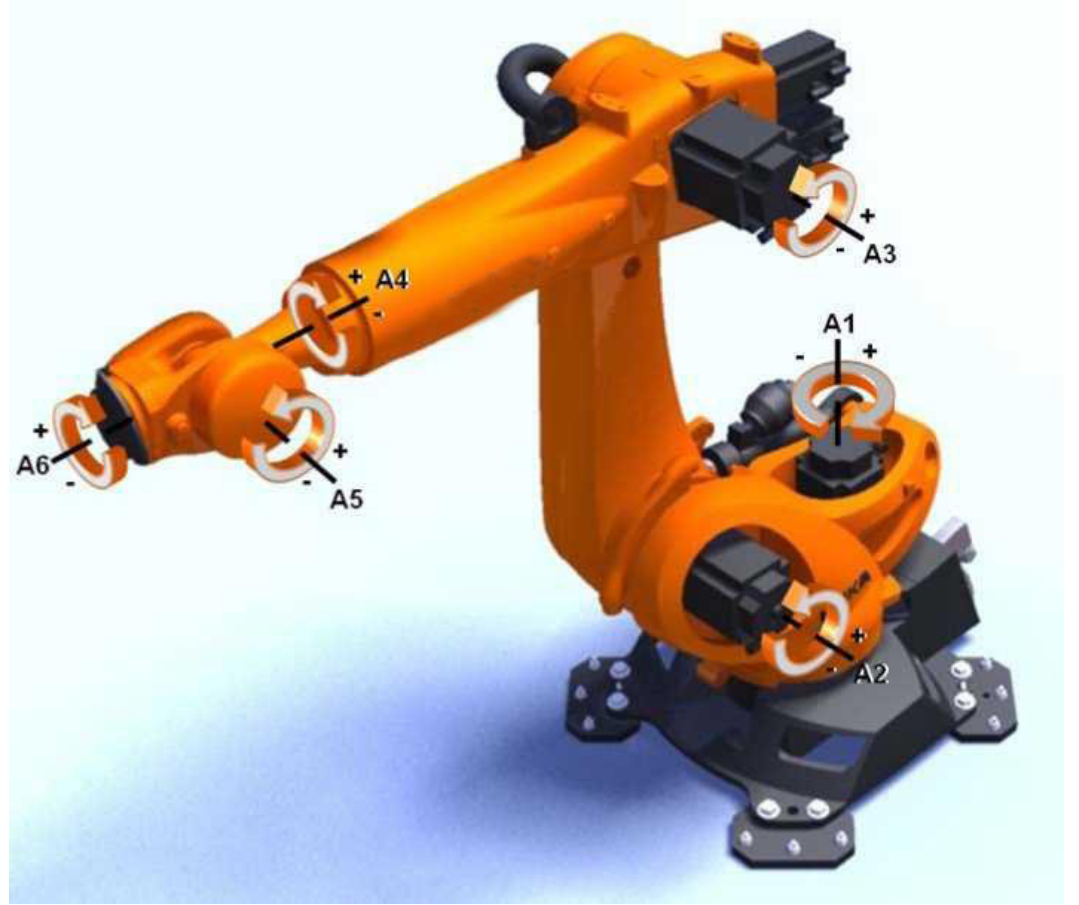
5 Kol

3 Ağırlık dengeleyici

6 El

Bir robot mekanik düzeninin bileşenleri çoğunlukla alüminyum ve çelik dökümden oluşur. Bazı durumlarda karbon elyafı bileşenler de kullanılmaktadır.

Tek tek akslar alttan (robotun ayağından) en üste kadar (robotun flanşı) numaralandırılmıştır.



Resim 4-5: Robotu serbestlik dereceleri

Ürün paletinden manipülatöre ait teknik verilerden bölümler:

- **Aks sayısı:** 4 (SCARA ve paralelkenar robotu) ile 6 (standart dikey maf-sallı kol robotu)
- **Erişim mesafesi:** 0,35 m (KR 5 scara) ile 3,9 m (KR 120 R3900 ultra K)
- **Net ağırlığı:** 20 kg ila 4700 kg.
- **Doğruluk:** 0,015 mm - 0,2 mm tekrarlama doğruluğu.

A1 ile A3 arasındaki eksen bölgeleri ile robotun el eksenini A5, tamponlu mekanik son dayanaklarla sınırlandırılmıştır.



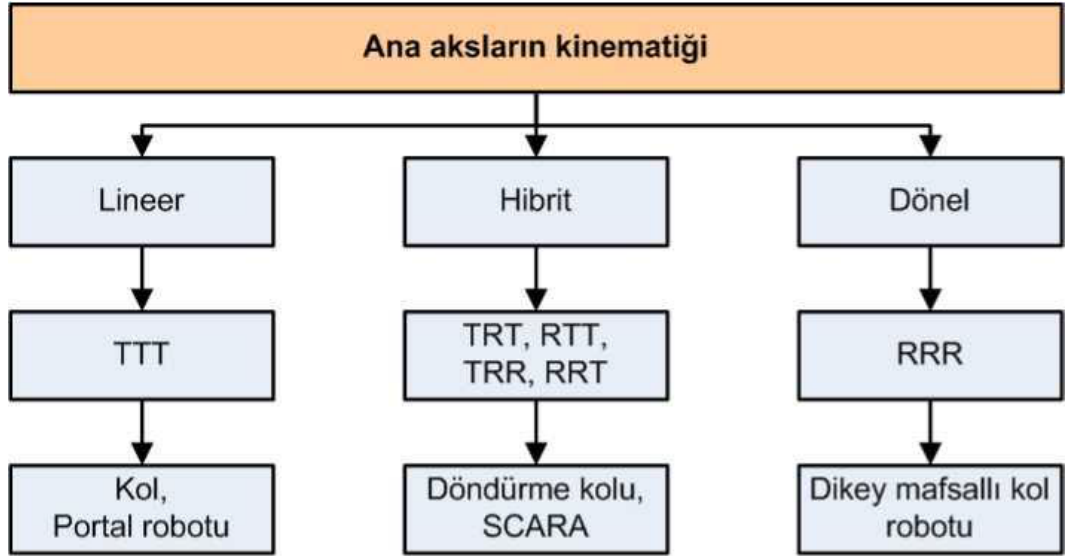
İlave akslara başka mekanik son dayanaklar monte edilmiş olabilir.

DUYURU Robot veya bir ilave aks, bir engele veya mekanik uç tahdidindeki veya eksen bölgesi sınırlamasındaki bir tampona çarptığı zaman, robot sisteminde maddi hasarlar meydana gelebilir. Endüstriyel robotun tekrar işleme alınmasından önce, üretici firma ile görüşülmesi gereklidir. Robotun işleme devam ettirilmesinden önce ilgili tampon yeni bir tampon ile değiştirilmelidir. Eğer robot (ilave aks) 250 mm/s'den yüksek bir hızla bir tampona çarparsa, robot (ilave aks) değiştirilmeli veya üretici firma tarafından bir yeniden işleme alma işlemi uygulanmalıdır.

4.5 Ana Aksların Konumu

Robotlar arasında ana aksların kinematik türüne göre ayırım yapılır:

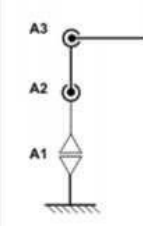
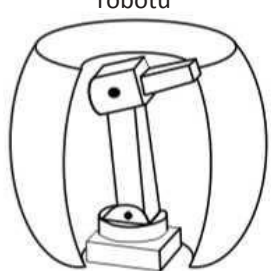
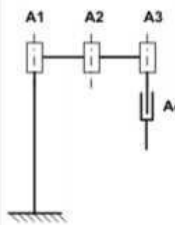
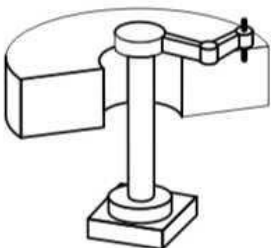
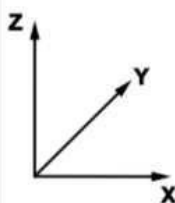
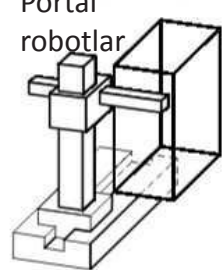
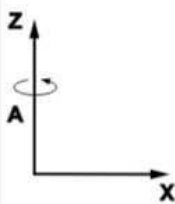
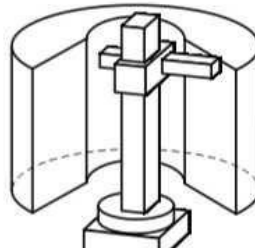
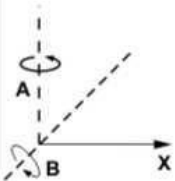

- ötelemeli hareket (T)
- rotatif hareket (R)



Resim 4-6: Ana aksların kinematığı

(>>> Resim 4-7) resminde farklı ana aks konumu seçenekleri ve bunun sonucunda elde edilen koordinatlar ayrıntılı bir şekilde gösterilmektedir:

1. Silindir koordinatları
2. Küre koordinatları
3. Dikey mafsallı koordinatları
4. Yatay mafsallı koordinatları (Scara robotları)
5. Kartezyen koordinatlar (portal robotları)

Aks kombinasyonlar	Koordinat tanımı	Çalışma alanları
3 döner aks RRR	Mafsal koordinatları (dikey) 	torus benzeri 
1 lineer 3 döner aks RRRT	Mafsal koordinatları (yatay) 	silindirik 
3-lineer aks TTT	Kartezik koordinatlar 	prizmatik 
2 lineer 1 döner aks RTT	Silindirik koordinatları 	silindirik 
1 lineer 2 döner aks RRT	Küre koordinatları 	küresel 

Resim 4-7: Hareket aksları farklı bir düzene sahip olan robot tipleri (ana akslar 1 - 3)

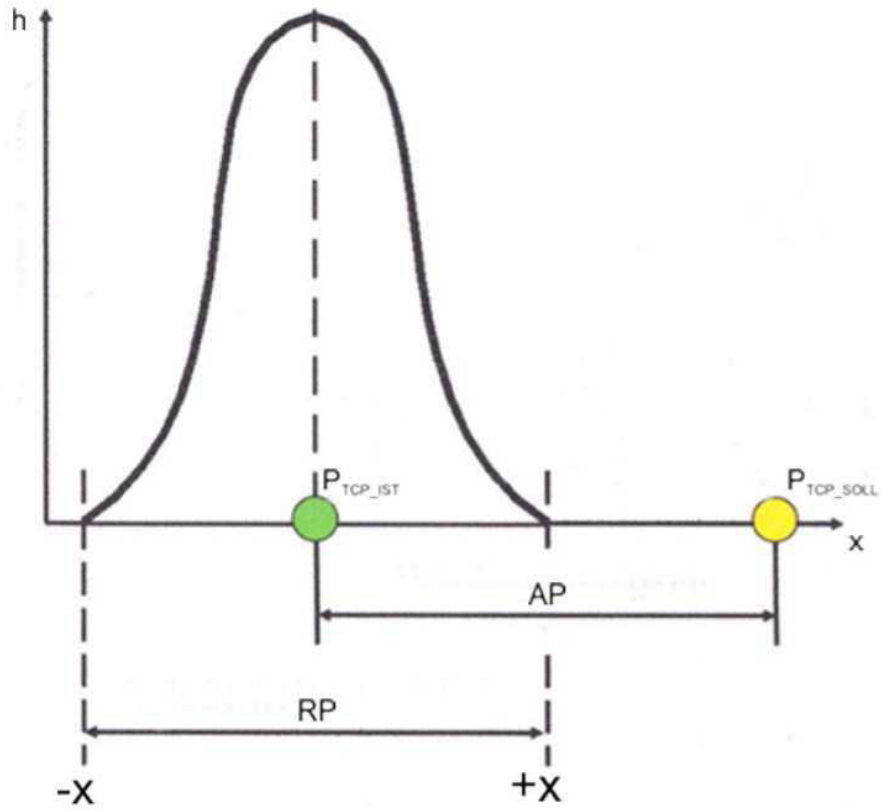
Robotun elinde aleti/kavrayıcıyı konumlandırmak ve yönlendirmek için üç ilave hareket aksı bulunur.

4.6 Mutlak Hassasiyet ve Tekrarlama Hassasiyeti

Mutlak hassasiyet, endüstriyel bir robotun Tool Center Point(TCP) noktasını kartezik koordinatlarla belirtilen bir alan noktasında belirli bir küre yarıçapı içerisinde belirtilen bir yönden tolerans alanı olarak konumlandırma kabiliyetidir. Mutlak hassasiyet, Tool Center Point noktasının ve belirtilen bir çalışma alanı noktasına ve belirtilen bir yönlendirmeye olan tüm serbestlik derecelerinde yönlendirmesinin ortalama sapmasını tanımlamaktadır (DIN EN ISO 9283).

Tekrarlama hassasiyeti bir endüstriyel robotun Tool Center Point noktasını çalışma alanında programlanmış olan bir noktada belirli bir küre yarıçapının içerisinde ve tanımlanmış olan bir yönlendirmeye belirlenmiş olan koşullar altında istenen sıklıkta yeniden konumlandırma kabiliyetini tanımlamaktadır (DIN EN ISO 9283).

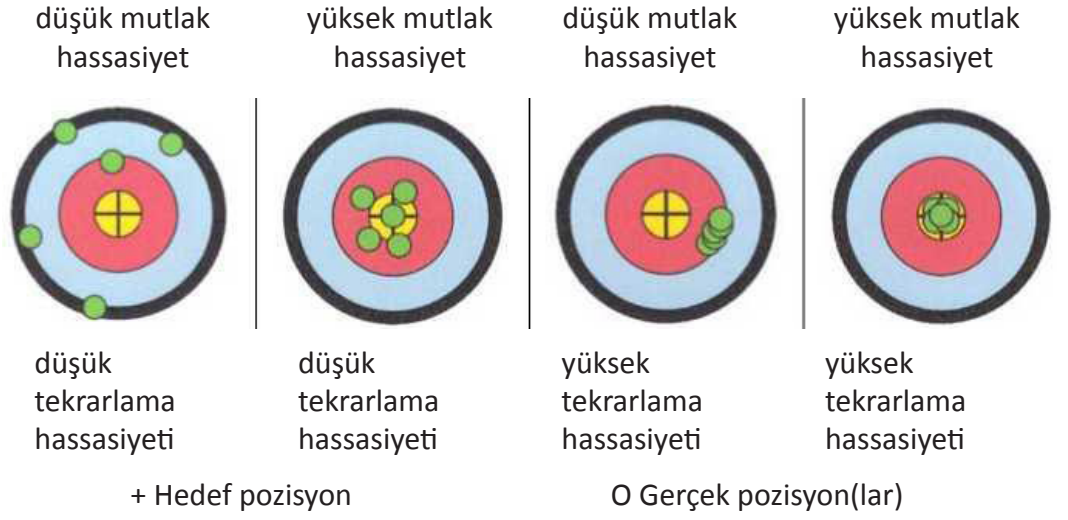
Mutlak hassasiyet, normalde tekrarlama hassasiyetinden çok daha kötüdür.



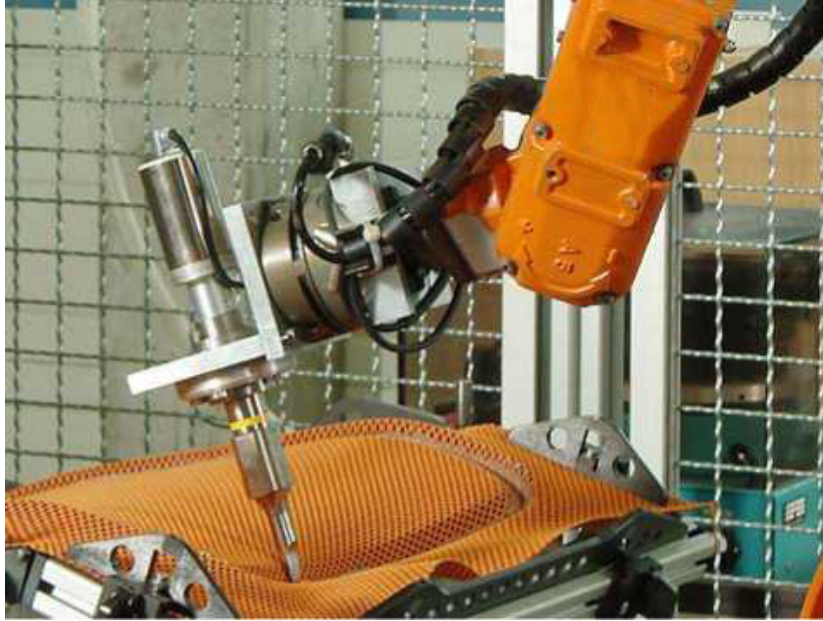
Resim 4-8: Endüstriyel robotların mutlak hassasiyeti ve tekrarlama hassasiyeti

σ	Sıklık dağılımı
AP	Mutlak hassasiyet
RP	Tekrarlama doğruluğu
P_{TCP_IST}	TCP'nin gerçek pozisyon değeri
P_{TCP_SOLL}	TCP'nin hedef pozisyon değeri

Mutlak hassasiyet, örneğin metin olarak veya çevrimdışı olarak programlanmış olan sayısal gerçek pozisyonların hedef pozisyona göre olan ortalama değer konumunu tanımlamaktadır.



Tekrarlama hassasiyeti, münferit gerçek pozisyonların, gerçek pozisyonların ortalama değerine göre olan sapmasını tanımlar (bkz. küre çapı ve küre orta noktası).



Resim 4-9: Yapıştırma uygulaması

Rota hassasiyeti, robotun, TCP'sini kartezik çalışma alanında tanımlanmış olan bir rota üzerinde tanımlanmış bir hızla hareket ettirme kabiliyetidir. TCP, belirtilen rota çevresinde belirli bir yarıçapın içerisinde kalmazsa veya belirli bir hız aralığı içerisinde (hedef hız) kalmazsa, bir sürükleme hatası söz konusudur.

5 Robot Kumandası

5.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Teknik verilere genel bakış
- Robot kumandasının bileşenleri
- Veri yolu sistemlerine genel bakış
- Enerji verimliliği

Temel Veriler

Dolap tipi	KR C4
Aks sayısı	Maks. 8
Ağırlık (trafosuz)	150 kg
Koruma sınıfı	IP 54
DIN 45835-1 uyarınca ses düzeyi	Ortalama 67 dB (A)
Soğutuculu ve soğutucusuz kurulum olanakları	Yan yana, mesafe 50 mm
Eşit yük dağılımında ünite tavanı yük kapasitesi	1.500 N

Şebeke Bağlantısı

Robot kumandası sadece topraklanmış nötr noktalı bir şebekeye bağlanabilir.

Topraklı bir yıldız noktası yoksa veya burada belirtilmeyen bir şebeke voltajı mevcutsa, bir trafo kullanılmalıdır

Nominal bağlantı gerilimi, isteğe göre:	AC 3x380 V, AC 3x400 V, AC 3x440 V veya AC 3x480 V
İzin verilen nominal bağlantı gerilim toleransı	Nominal bağlantı gerilimi \pm % 10
Şebeke frekansı	49 ... 61 Hz
Robot kumandası bağlantı noktasına kadar şebeke empedansı	$<$ 300 m Ω
Tam yük akımı	Bkz. tip levhası
Şebeke tarafındaki sigorta, trafosuz	min. 3x25 A gecikmeli
Şebeke tarafındaki sigorta, ayrıncı trafolu	13 kVA'da min. 3x32 A gecikmeli
Eş potansiyel topraklama	Eş potansiyel topraklama hatları ve tüm koruyucu iletkenler için ortak nötr nokta güç ünitesinin referans busudur.

İklimsel Koşullar

Soğutucusuz işletimde ortam sıcaklığı	+5 ... +45 °C (278 ... 318 K)
Soğutuculu işletimde ortam sıcaklığı	+20 ... +50 °C (293 ... 323 K)
Akülerle depolama ve taşımada ortam sıcaklığı	-25 ... +40 °C (248 ... 313 K)
Aküsüz depolama ve taşımada ortam sıcaklığı	-25 ... +70 °C (248 ... 343 K)
Sıcaklık değişiklikleri	maks. 1,1 K/dk

Nem sınıfı	DIN EN 60721-3-3'e göre 3k3; 1995
Kurulum yüksekliği	<ul style="list-style-type: none"> 1000 m üNN'ye kadar güç azalması yok 1000 m ... 4000 m üNN, %5/ 1000 m güç azalması ile

DUYURU Akülerin tümüyle boşalarak hasar görmesini önlemek amacıyla, akülerin depolama sıcaklığına bağlı olarak düzenli aralıklarla şarj edilmesi gerekir. Depolama sıcaklığı +20 °C veya altında ise, aküler 9 ayda bir şarj edilmelidir. +20 °C ila +30 °C arasındaki bir depolama sıcaklığında, aküler 6 ayda bir şarj edilmelidir. +30 °C ila +40 °C arasındaki bir depolama sıcaklığında, aküler 3 ayda bir şarj edilmelidir.

Sarsıntıya Dayanıklılık

Zorlama türü	Nakil sırasında	Sürekli işletim durumunda
Efektif ivme değeri (sürekli sarsıntı)	0,37 g	0,1 g
Frekans aralığı (sürekli sarsıntı)	4...120 Hz	
İvme (X/Y/Z yönünde şok)	10 g	2,5 g
Dalga biçimi/süresi (X/Y/Z yönünde şok)	Yarı sinüs/11 ms	

Daha yüksek mekanik yükler beklendiği takdirde, kontrol ünitesi amortisörlü bileşenler üzerine kurulmalıdır.

Kontrol Ünitesi

Besleme gerilimi	DC 27,1 V ± 0,1 V
------------------	-------------------

Kontrol PC'si

Ana işlemci	bkz. teslim edilen model
DIMM bellek modülleri	bkz. teslim edilen model (min. 2 GB)
Sabit disk	bkz. teslim edilen model

smartPAD

Besleme gerilimi	DC 20...27,1 V
Ölçüler (GxYxD)	yakl. 33x26x8 cm ³
Ekran	Dokunma duyarlı renkli ekran 600x800
Ekran büyüklüğü	8,4" nokta
Arabirimler	USB
Ağırlık	1,1 kg

Hat Uzunlukları

Hat tanımları, hat uzunlukları (standart) ve özel uzunluklar için manipülâtörün işletim ya da montaj kılavuzuna ve/veya KR C4 harici robot kumandaları kablolarının montaj ve işletim kılavuzuna bakılmalıdır.



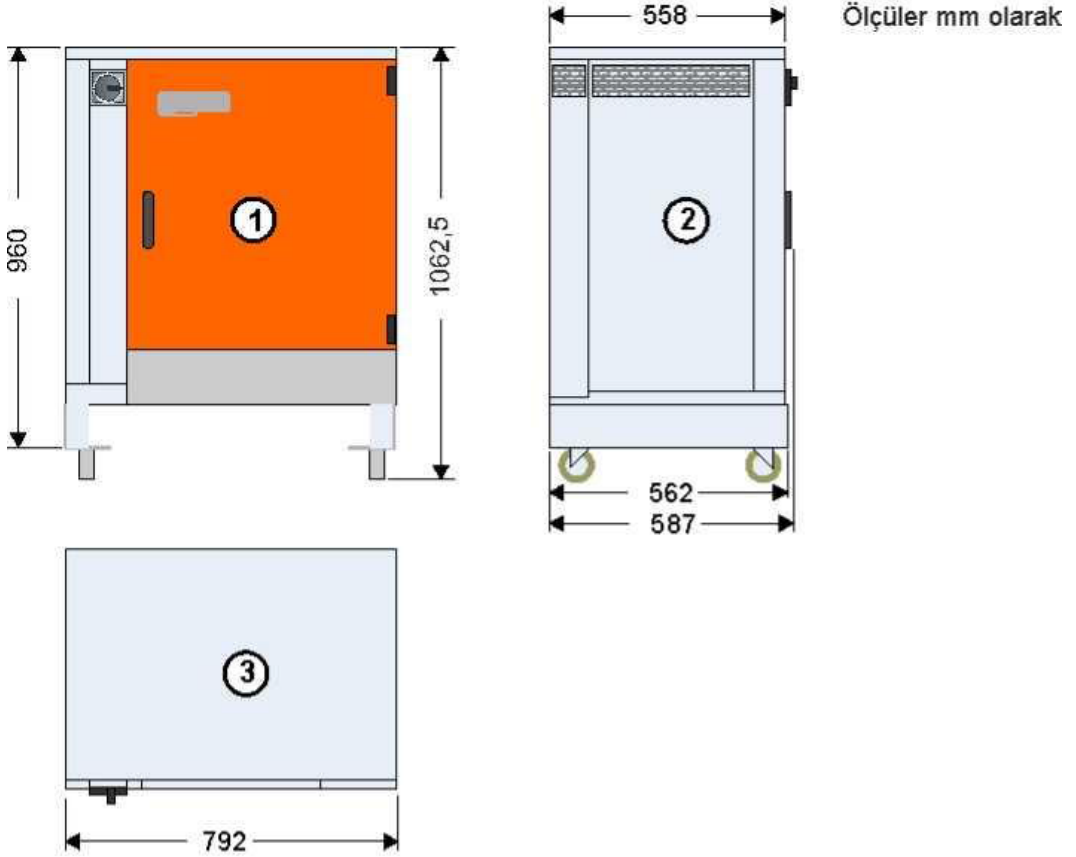
smartPAD uzatma kablolarının kullanılması halinde sadece iki uzatma kablosu kullanılmalıdır. 50 m'lik toplam kablo uzunluğu aşılmamalıdır.



RDC kutusuna ait kanallar arasındaki maksimum hat uzunluğu farklı 10 m olmalıdır.

5.2 Robot Kontrol Ünitesi Ölçüleri

Resim (>>> Resim 5-1), robot kontrol ünitesinin ölçülerini gösterir.

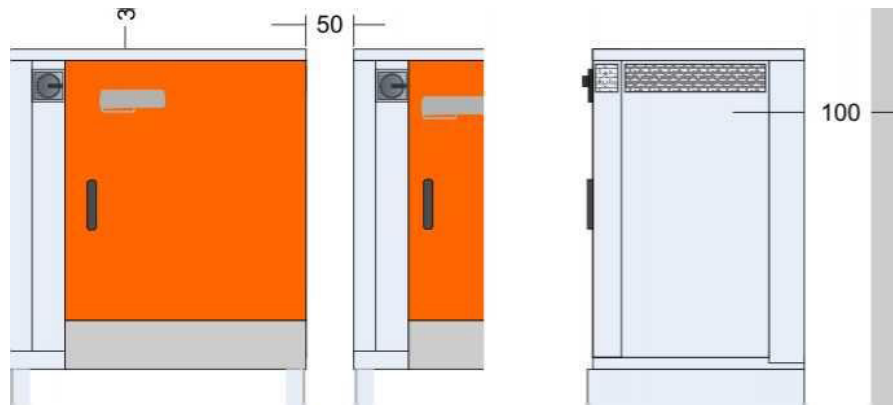


Resim 5-1: Ölçüler

- 1 Önden görünüm
- 2 Yandan görünüm
- 3 Yukarıdan görünüm

5.3 Robot Kontrol Ünitesi Asgari Mesafeleri

Resim (>>> Resim 5-2), robot kontrol ünitesinde uyulması gereken minimum mesafeleri gösterir.



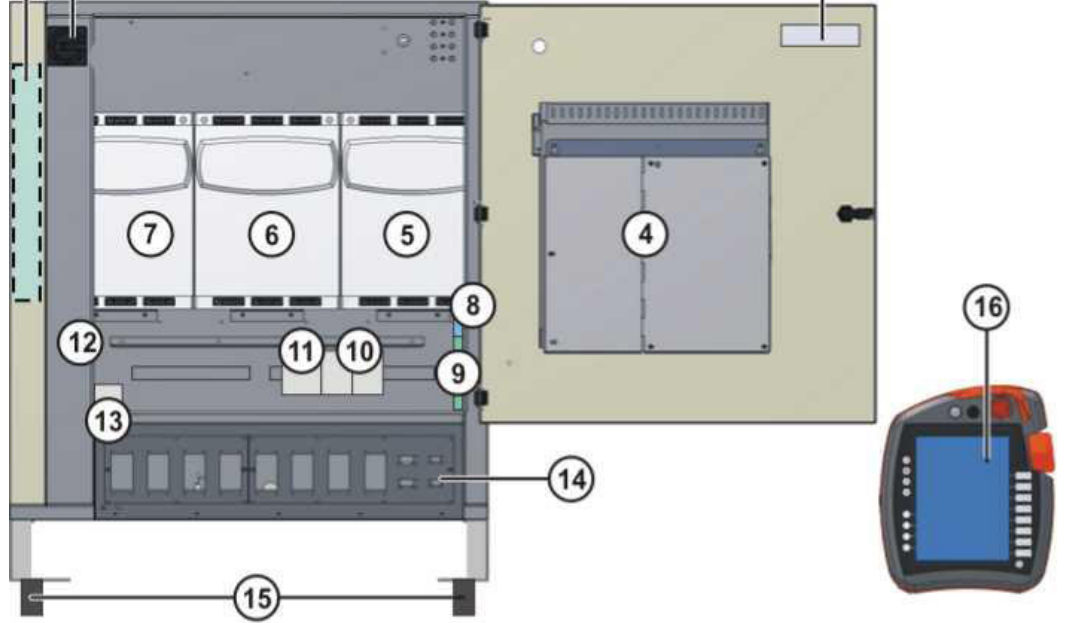
Resim 5-2: Asgari mesafeler

DUYURU Minimum mesafelere uyulmadığı takdirde robot kontrol ünitesi hasar görebilir. Belirtilen minimum mesafelere mutlaka uyulmalıdır.

i Robot kontrol ünitesinde belirli bakım ve onarım çalışmaları , yan taraftan veya arkadan yapılmalıdır. Bunun için robot kontrol ünitesi erişilebilir durumda olmalıdır. Yan veya arka duvarlara erişilemediğinde robot kontrol ünitesi, bu çalışmaların yapılabileceği bir konuma hareket ettirilebilmelidir.

5.4 Robot Kontrol Sistemine Genel Bakış

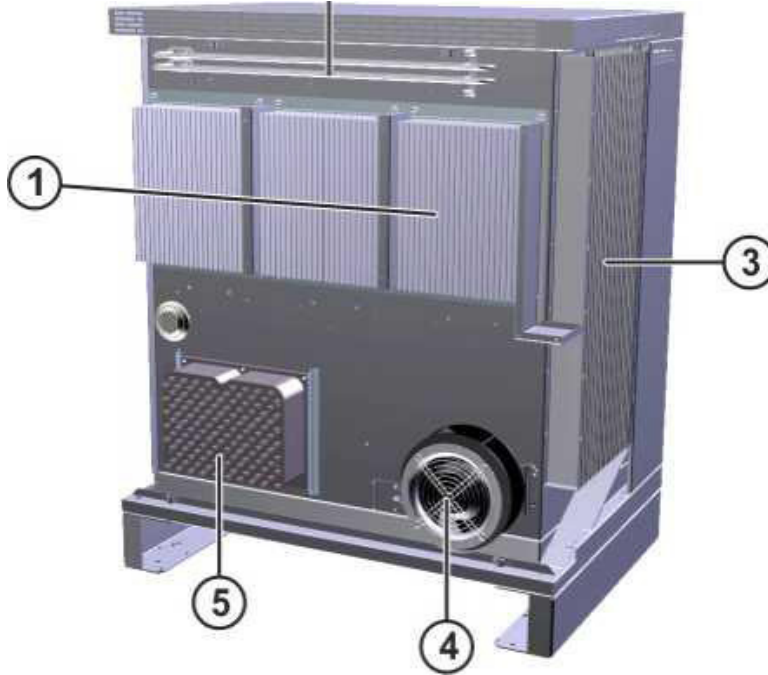
KR C4 Önden Görünümü



Resim 5-3: Robot kontrol ünitesinin önden görünümüne genel bakış

- | | |
|---|--------------------|
| 1 Ağ filtresi | 9 CCU |
| 2 Ana şalter | 10 Kontaktörler |
| 3 CSP | 11 Ağ anahtarı |
| 4 Kontrol PC'si | 12 Emniyet elemanı |
| 5 Sürücü regülatörlü sürücü güç kaynağı | 13 Aküler |
| 6 Eksen 4 ila 6 için sürücü regülatörü | 14 Bağlantı paneli |
| 7 Eksen 1 ila 3 için sürücü regülatörü | 15 Kasa |
| 8 Fren filtresi | 16 smartPAD |

KR C4 Arka Görünümü

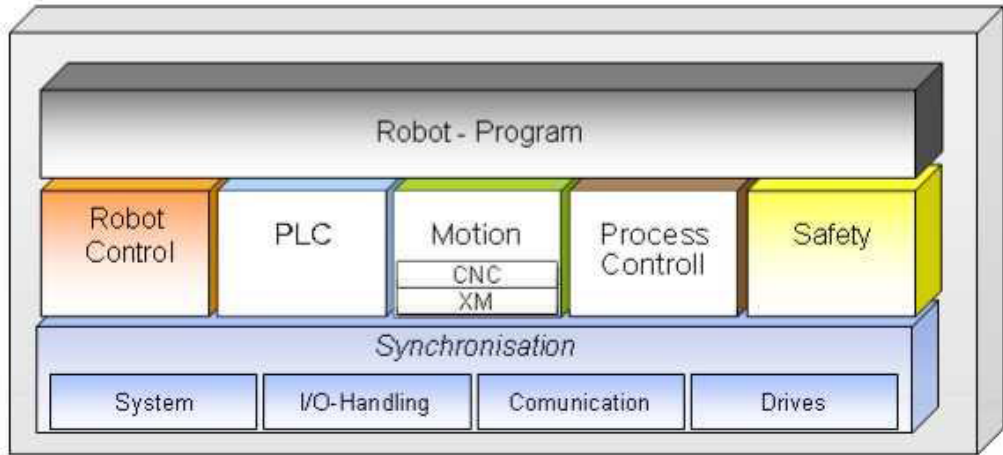


Resim 5-4: Robot kumanda ünitesinin arkadan görünümüne genel bakış

- 1 Balast dirençleri
- 2 Isı eşanjörü
- 3 Dış fan
- 4 Alçak gerilim güç kaynağı

5.5 Uygulamaya ve Veri Yolu Sistemlerine Genel Bakış

Uygulamalara Genel Bakış



Resim 5-5: Sütun modeli genel görünümü

- Robot programlarından dolaba özgü donanım arabirimlerine erişebilmek için kumanda ünitesinde uygulamalar ve ilave seçenekler yüklü olmalıdır.

- **Sistem ile ilgili uygulamalar** arasında aşağıdakiler yer almaktadır:

- **RC (Robot Control)**

Robot kumandası için çekirdek sistem

- **Safety**

Üretici firmanın kendi entegre edilmiş olduğu güvenlik kontrol sistemi
Müşteriye özel seçenekler arasında aşağıdakiler yer almaktadır:

- **PLC**

genel süreç kontrolü için entegre edilebilir Soft-PLC'dir

- **XM (eXtended Motion)**

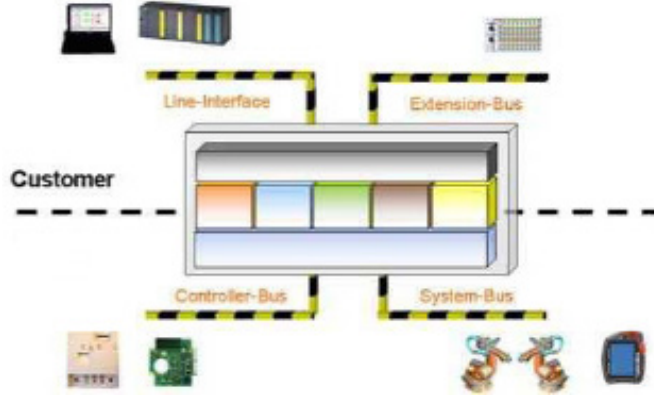
bir MotionControl kitaplığı için entegre edilebilir yürütme zamanı sistemidir

Process Control

proses kontrol sistemlerinin entegrasyonu için genel platformdur örn. Vision entegre edilebilir

Veri Yolu Sistemlerine Genel Bakış

- KR C4 kontrol sistemlerinde dört farklı EtherNet tabanlı veri yolu sistemi vardır
- Bu veri yolu sistemlerinin her biri farklı kontrol sistemi bileşenlerini bağlamaktadır.

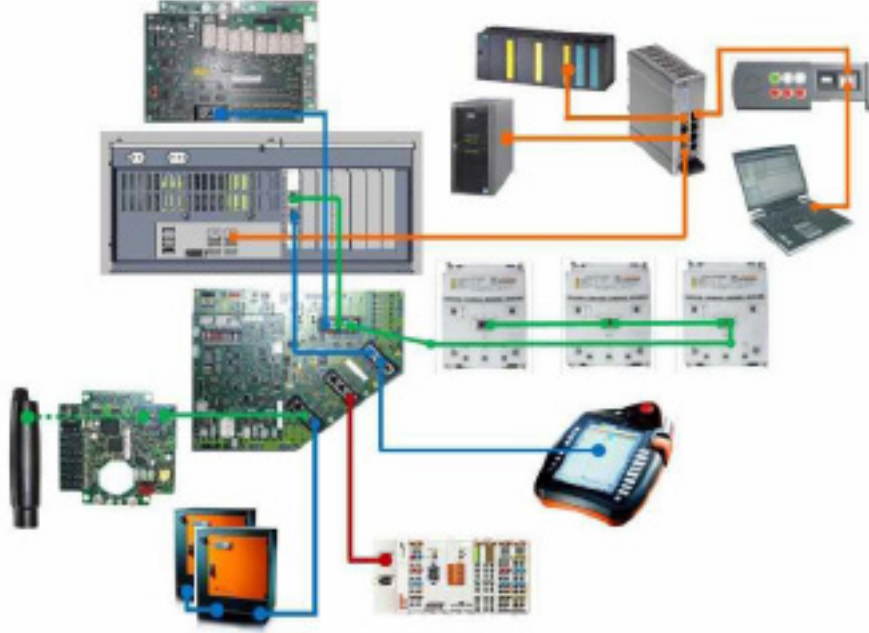


Resim 5-6: KR C4 veri yolu sistemlerine genel bakış


Bus	Açıklama
KCB (KUKA Controller Bus)	Tahrik devresinin katılımcılarının bağlanması: <ul style="list-style-type: none">■ RDC (Resolver Digital Converter)■ KPP (KUKA Power Pack)■ KSP (KUKA System Pack)■ EMD (Electronic Mastering Device - Elektronik Ayar Cihazı)
KSB (KUKA System Bus)	Aşağıdakilerin bağlanması: <ul style="list-style-type: none">■ smartPAD■ SIB (Safety Interface Board)■ SIB extended■ RoboTeam■ diğer KUKA seçenekleri
KEB (KUKA Extension Bus)	Aşağıdakilerin bağlanması: <ul style="list-style-type: none">■ EtherCat E/As■ PROFIBUS için ağ geçidi■ DeviceNet için ağ geçidi
KLI (KUKA Line Interface)	Aşağıdakilerin bağlanması: <ul style="list-style-type: none">■ PLC■ Alan bus bağlantısı■ PROFINET & PROFI-safe■ EtherNet/IP & CIP safety■ TCP/IP üzerinden ağ bağlantısı:■ Veri arşivleme■ Tanımlama■ VRP (Virtual Remote Pendant)■ vs.

Ayrıntılı veri yolu görünümü:

Anakart D3075-KBus



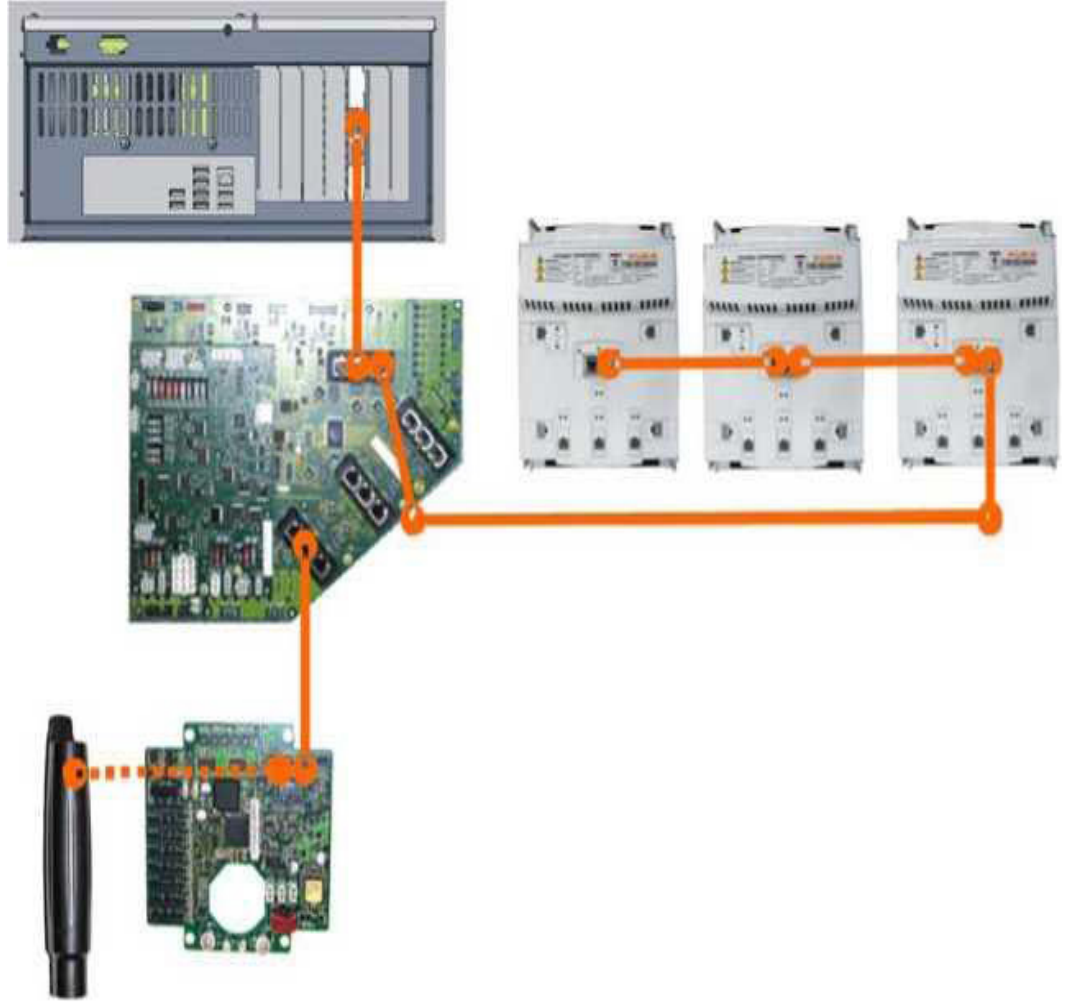
Resim 5-7: Veri yolu mimarisi

	Anakart sürümüne bağlı olarak KLI ve KSB bağlantılarının yeri değiştirilmiştir.
	Anakart D2608-K
	<ul style="list-style-type: none">• DualNIC üzerinde KLI bağlantısı• Anakart üzerinde KSB bağlantısı
	Anakart D3076-K
	<ul style="list-style-type: none">• Anakart üzerinde KLI bağlantısı• DualNIC üzerinde KSB bağlantısı

5.5.1 KUKA Controller Veri Yolu, KCB

Açıklama

KUKA Controller Bus'a genel bakış:



Resim 5-8: KCB veri yolu mimarisi

KCB ana bilgileri

- EtherCat tabanlı sürücü veri yolu
- Çevrim süresi 125 milisaniye
- FSOE (Fail Safe Over EtherCat)

Aşağıdaki cihazlar KCB'ye aittir:

- KPP - KUKA Power Pack
- KSP A1 - 3 -KUKA Servo Pack
- KSP A4 - 6 -KUKA Servo Pack
- RDC - Resolver Digital Converter
- EMD - Electronic Mastering Device (bağlanabilir katılımcı)

5.5.2 KUKA Ssistem Veri Yolu, KSB

Açıklama

KUKA System Bus'a genel bakış:



Resim 5-9: KSB veri yolu mimarisi

DUYURU	Kullanılan KPC anakart sürümüne bağlı olarak KLI ve KSB için farklı donanım arabirimleri kullanılır.
Anakart D2608-K	<ul style="list-style-type: none">• DualNIC üzerinde KLI bağlantısı• Anakart üzerinde KSB bağlantısı
Anakart D3076-K ile KPC	<ul style="list-style-type: none">• Anakart üzerinde KLI bağlantısı• DualNIC üzerinde KSB bağlantısı

KSB ana bilgileri

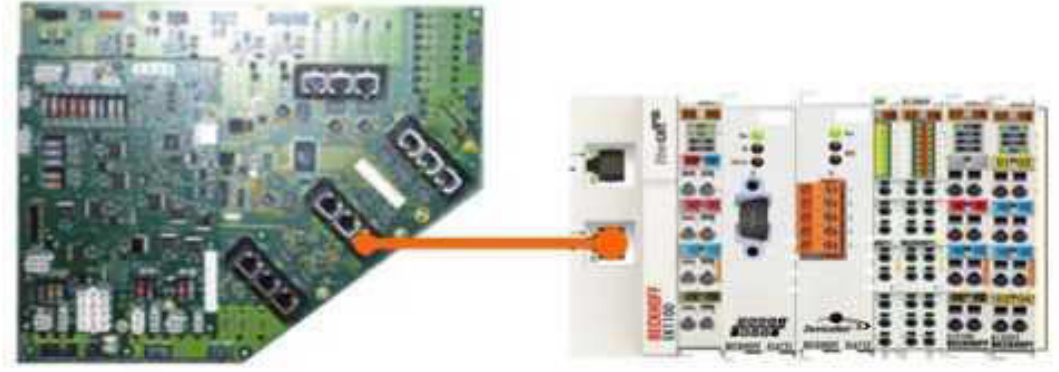
- EtherCat tabanlı veri yolu
- Çevrim süresi 1 milisaniye
- FSOE (Fail Safe Over EtherCat)
-

Aşağıda belirtilen cihazlar KSB'ye bağlanır:

- smartPAD (RDP üzerinden HMI)
- RoboTeam (bir bağlantı kablosu üzerinden)
- Safety bağlantısı; SIB (Safety Interface Board) X11, X13

5.5.3 KUKA Extension Bus, KEB

Açıklama KUKA Extension Bus'a genel bakış:



Resim 5-10: KEB veri yolu mimarisi

KEB ana bilgileri

- CCU'da EtherCAT Master
- Çevrim süresi 1 milisaniye
- ayırık müşteri G/Ç'ların basit entegrasyonu için DeviceNet yedeği olarak yüksek performanslı alan veri yolu (güncel olarak güvenli G/Ç yok)
- WorkVisual ile konfigürasyon

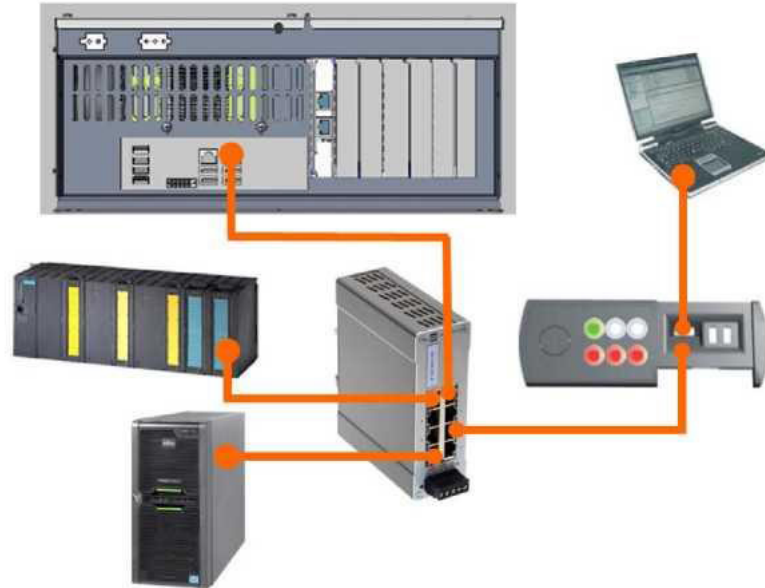
Aşağıda cihazlar KEB'ye bağlanır:

- müşteriye özel EtherCAT G/Ç modülleri
- Profibus ve DeviceNet için ağ geçidi çözümleri

5.5.4 KUKA Line Interface, KLI

Açıklama

KUKA Line Interface'e genel bakış:



Resim 5-11: KLI veri yolu mimarisi

DUYURU

KSS yazılım sürümüne bağlı olarak KLI ve KSB bağlantılarının yeri değiştirilmiştir.

Anakart D2608-K

- DualNIC üzerinde KLI bağlantısı
- Anakart üzerinde KSB bağlantısı

Anakart D3076-K

- Anakart üzerinde KLI bağlantısı
- DualNIC üzerinde KSB bağlantısı

KLI Ana Bilgileri

- Ethernet tabanlı müşteri arabirimleri X66, X67

Tesise ve üst düzlemlere bağlantı (müşteri ağı, sunucu)

- Ethernet tabanlı alan buslar (PROFINET, PROFI-safe, EtherNet/IP, CIP Safety)
- Standart Ethernet (örn. arşivleme ve veri alışverişi)
- WorkVisual ile konfigürasyon

Aşağıda belirtilen cihazlar KLI'ye bağlanır:

- Servis dizüstü bilgisayar
- Müşteri G/Ç modülü, PLC, Güvenlik PLC
- Sunucu, yönetim merkezi bilgisayar

5.6 Enerji Verimliliği

KR C4 ile enerji

KR C4 sisteminde özellikle, üretim ve bekleme konumunda enerji tüketimi ko-tasarrufu nüsuna önem verilmektedir.

Enerjiden tasarruf sağlayan ısınma konsepti

- Dıştaki soğutma devresi yüklenmeye ve çevre sıcaklığına bağlı olarak sıcaklık düzenlemeli bir 24 V fanı kullanır.
- Dolap iç fanı iptal edilmişse -> PC fanı ve bilinçli hava yönlendirmesiyle sirkülasyonu garantiye alın -> Isıyı atma için yüzeylerin daha fazla kullanılması -> 27 V güç kaynağının harici soğutma devresi üzerinden ısıyı atması.
- Daha az kurulu soğutma kapasitesi ve daha düşük dönme hızı (bekleme durumunda yakl. 220 W).

KUKA "Energy Efficiency Modes"

- Sürücü veri yolu modu
 - Sürücü veri yolu devre dışı bırakma (frenler uygulandı)
 - KUKA-PC güvenlik kontrol sistemli ve PROFINET bağlantısı henüz aktif
 - Sürüş hazırlık yakl. 10 san. sonra
 - Tüketim yakl. 158 W
- Mode Sleep
- Kontrol sisteminde tüm sistemlerin devre dışı bırakılması (Ethernet "Low Level" hariç)
- KUKA-PC güvenlik kontrol sistemli ve PROFINET "Sleep" modunda
- Sürüş hazır 30 - 60 san. sonra (uygulamaya bağlı)
- Tüketim yakl. 40 W

Etkinleştirme

Tesis düzleminde bir komut veya kontrol sisteminde bir kumanda faaliyeti sonucunda etkinleştirme/devre dışı bırakma

- PROFINET: Handling AIDA ve PNO ile standartlaştırılmıştır (PLC düzleminde ProfiEnergy profili)
- Ethernet: Magic-Packet ile etkinleştirme
- Mahaldeki kontrol sisteminde manüel
 - KUKA smartPAD üzerindeki menü noktasından Power-Down
 - Ana şalter üzerinden Wake-Up

AIDA	Alman Otomotif OEM'leri standardizasyon girişimi
PNO	Profibus kullanıcı organizasyonu

6 Robotu Hareket Ettirme

6.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Mesajları okuma ve yorumlama
- İşletim türlerini seçme ve ayarlama
- Robot akslarını tek tek hareket ettirme
- Dünya koordinat sisteminde robotu hareket ettirme
- Tool koordinat sisteminde robotu hareket ettirme
- Base koordinat sisteminde robotu hareket ettirme


6.2 Programlama El Cihazı smartPAD

6.2.1 Ön Taraf

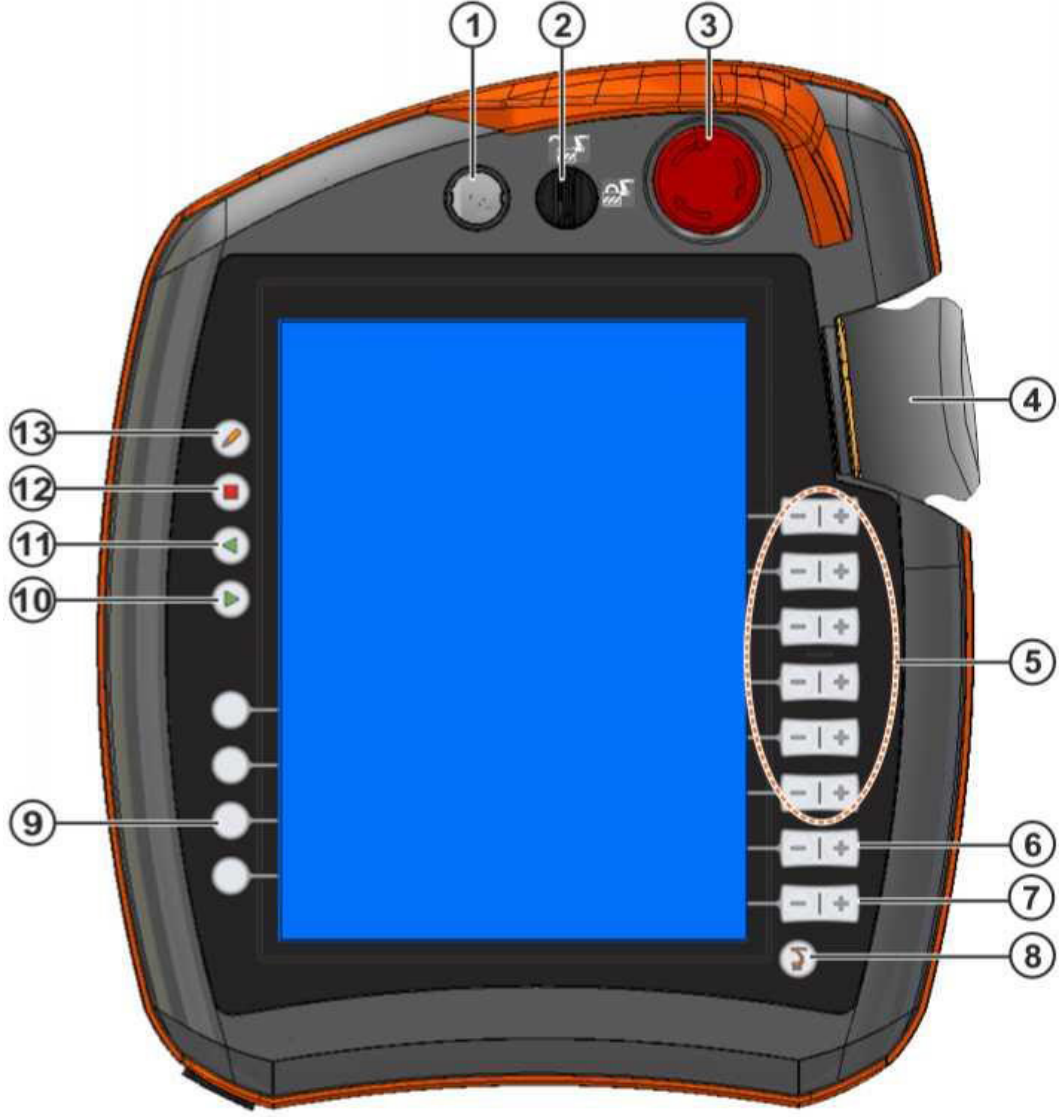
Fonksiyon

smartPAD, endüstriyel robotun programlama el cihazıdır. smartPAD, endüstriyel robotun kumandası ve programlanması için ihtiyaç duyulan tüm kullanım ve görüntüleme olanaklarına sahiptir.

smartPAD, dokunmatik bir ekrana sahiptir: smartHMI, parmakla veya bir işaretleme kalemiyle kullanılabilir. Harici fare veya harici klavye gerekli değildir.

 Bu belgede smartPAD, sık sık genel adı "KCP" (KUKAControl Panel) ile adlandırılmaktadır.

Genel Bakış



Resim 6-1: smartPAD ön tarafı

Poz.	Açıklama
1	smartPAD cihazını ayırmak için tuş
2	Bağlantı yöneticisini çağırmak için anahtarlı salter. Salter sadece anahtar içindeyse çevrilebilir. Bağlantı yöneticisi üzerinden işletim türü değiştirilebilir.
3	ACİL DURDURMA cihazı. Tehlikeli durumlarda robotu durdurmak için ACİL DURDURMA cihazı, basıldığında kilitletir.
4	Space Mouse: Robotu manuel hareket ettirmek için
5	Hareket tuşları: Robotu manuel hareket ettirmek için
6	Programm-Override'ı ayarlamak için tuş
7	Programm-Override'ı ayarlamak için tuş
8	Ana menu tuşu: smartHMI'da menu noktalarını görüntüler
9	Durum tuşları. Durum tuşları özellikle teknoloji paketlerindeki parametreleri ayarlamaya yarar. Tam fonksiyonları hangi teknoloji paketlerinin yüklendiğine bağlıdır.

Poz.	Açıklama
10	Başlat tuşu: Başlat tuşu ile bir program başlatılır
11	Başlat geri tuşu: Başlat geriye tuşu ile bir program geriden başlatılır. Program adım adım işlenir.
12	STOP tuşu: STOP tuşu ile çalışan bir program durdurulur.
13	Klavye tuşu: Klavyeyi görüntüler. Genelde klavyenin özel olarak açılması gerekmez, çünkü smartHMI klavye üzerinden giriş yapılması gerektiğini algılar ve klavyeyi otomatik olarak görüntüler.

6.2.2 Arka Taraf

Genel Bakış



Resim 6-2: smartPAD'in arka tarafı

- | | |
|-----------------------|------------------|
| 1 Onay şalteri | 4 USB bağlantısı |
| 2 Başlat tuşu (yeşil) | 5 Onay şalteri |
| 3 Onay şalteri | 6 Tip levhası |

Açıklama

Öge	Açıklama
Tip levhası	Tip levhası
Başlat tuşu	Başlat tuşu ile bir program başlatılır.

Oge	Açıklama
Onay şalteri	Onay şalterinin 3 konumu vardır: <ul style="list-style-type: none"> ■ Basılı değil ■ Orta konum ■ Basılı Manipülâtörün hareket edebilmesi için onay şalteri T1 ve T2 işletim türlerinde orta konumda tutulmalıdır. Otomatik ve otomatik harici işletim türlerinde onay şalter fonksiyonsuzdur.
USB bağlantısı	USB bağlantısı örneğin arşivleme/geri yükleme için kullanılır Sadece FAT32 formatlı USB bellekler.

6.3 Robot Kumandasının Mesajlarını Okuma ve Yorumlama

Mesajlara Genel Bakış



Resim 6-3: Mesaj penceresi ve mesaj sayacı

1. Mesaj penceresi: Güncel mesaj görüntülenir
- 2 Mesaj sayacı: Her mesaj tipi için mesaj adedi


Kumanda sistemi, mesaj penceresi üzerinden kullanıcıyla iletişim kurmaktadır. Bu bağlamda beş farklı mesaj tipine sahiptir:

Mesaj tiplerine genel bakış:


Sembol	Tip
	Onay Mesajı <ul style="list-style-type: none"> ▪ Robot programının yürütülmesine devam etmek için operatörün onayının gerekli olduğu durumları belirtmek için kullanılır. (örn. "ACİL-DURDURMA onayı") ▪ Bir onay mesajı daima robotun durmasına veya yeniden başlamamasına neden olur.
	Durum Mesajı <ul style="list-style-type: none"> ▪ Durum mesajları, kumanda sisteminin güncel durumlarını bildirir. (örneğin "ACİL DURDURMA") ▪ Durum mesajları, durum devam ettiği sürece onaylanamaz.

Sembol	Tip
	Uyarı Mesajı <ul style="list-style-type: none"> Uyarı mesajları robotun doğru kullanımına ilişkin bilgi verir. (örn. "Başlatma tuşu gerekli") Uyarı mesajları onaylanabilir türde mesajlardır. Kumandayı durdurmadıkları için bunların onaylanması zorunlu değildir.
	Durum Mesajı <ul style="list-style-type: none"> Bekleme mesajları, kumandanın hangi olayı (durum, sinyal veya süre) beklediğini belirtir. Bekleme mesajları "Simüle et" butonuna basılarak manuel olarak iptal edilebilir.

DUYURU Simüle et" komutu sadece çarpışma ve diğer tehlikeler söz konusu olmadığında kullanılabilir!

 **Diyalog Mesajı**


- İletişim mesajları doğrudan operatörle iletişim kurmak / komut almak için kullanılır.
- Çeşitli cevap seçenekleri sunulan, butonların olduğu bir mesaj penceresi görüntülenir.

 Onaylanabilir bir mesaj "OK" ile onaylanabilir. Tümü OK ile, onaylanabilir tüm mesajlar bir defada onaylanabilir.

Mesajların Etkisi

Mesajlar, robotun işlevselliğine etki eder. Bir onay mesajı daima robotun durmasına veya yeniden başlamamasına neden olur. Robotu hareket ettirebilmek için mesajın önce onaylanması gerekir.

"OK" komutu (onay), kullanıcı için mesaj içeriğinin bilincine varma konusunda bir taleptir.

 **Mesajların kullanımı konusunda ipuçları:**

- Bilinçli okuyun!
- Daha eski mesajları ilk olarak okuyun. Yeni mesajlar, eski mesajların sonucunda ortaya çıkmış olabilir.
- Hemen "Tümü OK" üzerine basmayın.
- Özellikle çalıştırdıktan sonra: Mesajlara bakın. Bu sırada her bir mesajın görüntülenmesini sağlayın. Mesaj penceresi üzerine basıldığında mesaj listesi genişler.

Mesajların Kullanımı

Mesajlar daima tarihi ve saat bilgisi ile verilerek olayın tam zamanının tespit edilmesine olanak sağlarlar.



Resim 6-4: Mesajları onaylama

Mesajların görüntülenmesi ve onaylanması için izlenecek yöntem:

- Mesaj penceresine dokunarak mesaj listesini genişletin.
- Onaylama:

- "OK" ile mesajları tek tek onaylayın.
- Alternatif: "Tümü OK" ile tüm mesajları onaylayın.

3. En üstteki mesaja yeniden dokunulduğunda veya ekranın sol kenarındaki "X" işaretine dokunulduğunda mesaj listesi tekrar kapanır.

6.4 İşletim Türünü Seçme Ayarlama

- T1 (manüel azaltılmış hız)
 - Test işletimi, programlama ve öğretme (teach) için
 - Program modundaki azami hız 250 mm/s
 - Elle modundaki azami hız 250 mm/s
- T2 (manüel yüksek hız)
 - Test işletimi için
 - Program modundaki hız, programlanmış hıza göredir!
 - Elle işletim: mümkün değildir
- AUT (Otomatik)
 - Üst kontrol ünitesi olmayan endüstriyel robotlar için
 - Program modundaki hız, programlanmış hıza göredir!
 - Elle işletim: mümkün değildir
- AUT EXT (Otomatik harici)
 - Üst düzey kontrol sistemi (PLC) olan endüstriyel robotlar için
 - Program modundaki hız, programlanmış hıza göredir!
 - Elle işletim: mümkün değildir

Manuel İşletim T1 ve T2

Manuel işletim türü, ayar çalışmaları için olan işletim türüdür. Ayarlama işlemleri, otomatik işleme geçebilmek için robot üzerinde yapılması gereken tüm çalışmaları kapsamaktadır. Bunlar:

- Öğretme / programlama
- Programı, jog modunda yürütme (test etme /doğrulama)

Yeni veya değiştirilmiş programlar önce daima Manuel Azaltılmış hız (T1) işletim türünde test edilmelidir.

Manuel Azaltılmış hız (T1) işletim türünde:

- Kullanıcı koruması (koruma kapısı) etkin değildir!
- Şayet mümkünse, hiç kimsenin koruma tertibatları ile sınırlandırılmış alan içerisinde bulunmaması gerekmektedir.

Birden çok kişi koruma tertibatları ile sınırlandırılmış alanda bulunmasının gerekli olduğu durumlarda aşağıdaki konulara dikkat edilmesi gerekmektedir:

- Tüm şahısların robot sistemini önünde engel olmaksızın net bir şekilde görülebilmemesi gerekmektedir.
- Tüm şahıslar arasında göz bağlantısı kurulabilme imkanının mevcut olması gerekmektedir.
- Operatörün, tüm tehlikeleri görebileceği ve bir tehlikeden sakınabileceği bir pozisyon alması gerekmektedir.

Manuel Yüksek hız (T2) işletim türünde:

- Kullanıcı koruması (koruma kapısı) etkin değildir!
- Bu işletim türü ancak uygulama, manuel azaltılmış hızdan daha yüksek hızda bir test gerektirdiğinde kullanılmalıdır.
- Bu işletim türünde öğretme işlemlerinin yapılmasına izin yoktur.

Operatör test başlangıcı öncesinde onay tertibatlarının hatasız olarak çalıştığından emin olmalıdır.

- Operatör, tehlike bölgesinin dışında bir pozisyon almalıdır.
- Koruma tertibatları ile sınırlandırılmış alan içerisinde hiç kimsenin bulunması gerekmektedir.

İşletim türleri Otomatik ve Harici Otomatik

- Güvenlik ve koruma tertibatları mevcut ve tam işler durumda olmalıdır.
- Tüm kişiler, koruma tertibatlarıyla sınırlandırılmış bölgenin dışında olmalıdır.

İzlenecek Yöntem

i İşletim sırasında işletim türü değiştirilirse, sürücüler derhal durdurulur. Endüstriyel robot, güvenlik durumu 2 ile durur.

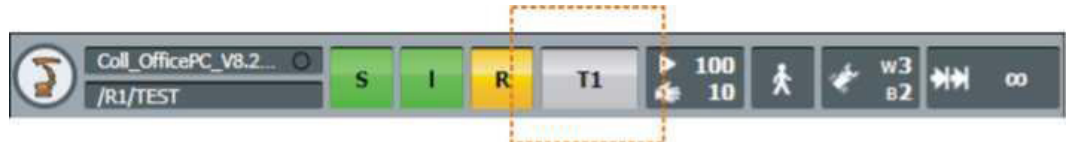
1. KCP'de bağlantı yöneticisi için olan şalteri diğer duruma getirin. Bağlantı yöneticisi görüntülenir.



2. İşletim türünü seçin.

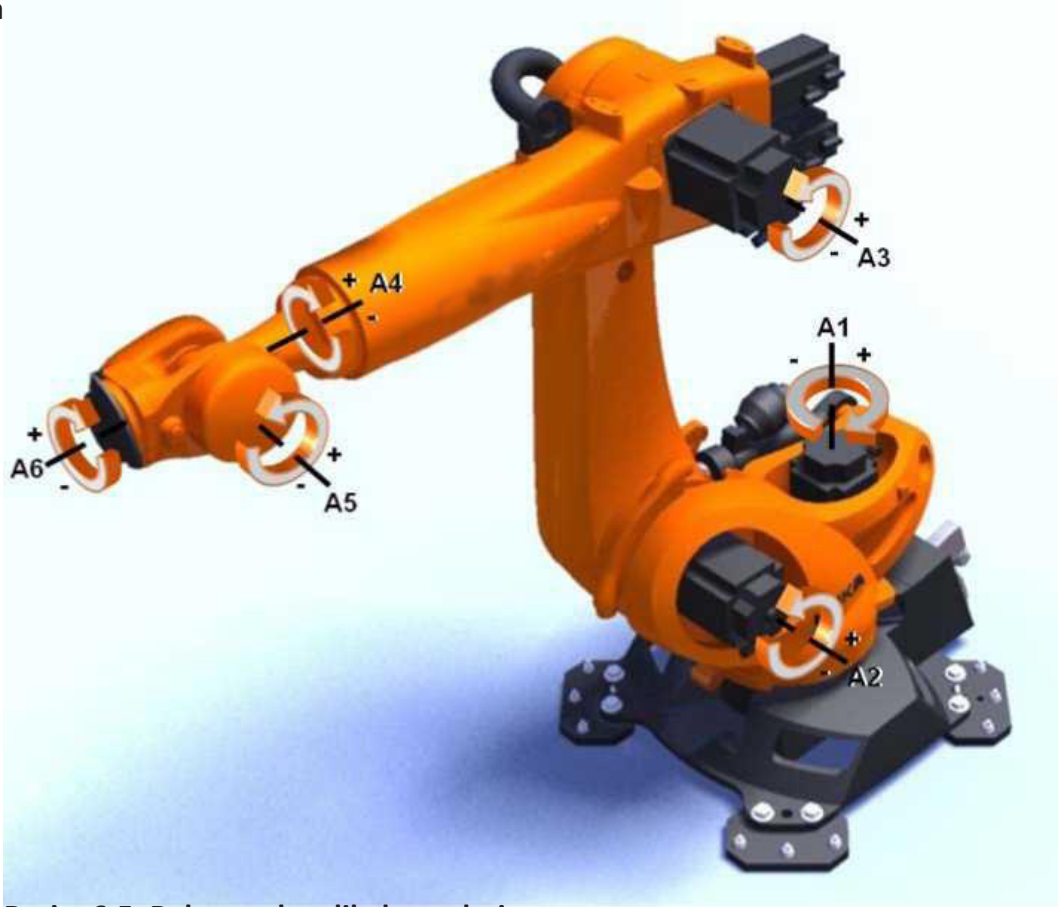


3. Bağlantı yöneticisi için olan şalteri tekrar başlangıç pozisyonuna getirin. Seçilen işletim türü, smartPAD'in durum çubuğunda görüntülenir.



6.5 Robot Akslarını Tek Tek Hareket Ettirme

Açıklama: Aksa özgü hareket



Resim 6-5: Robot serbestlik dereceleri

Robot akslarının hareketi

- Her aksı teker teker artı ve eski yöne getirin.
- Bu amaçla smartPAD hareket tuşları ya da Space Mouse faresi kullanılır.
- Hız değiştirilebilir (Hand (elle) Over-Ride: HOV).
- Elle sürüş, sadece T1 işletim türünde olanaklıdır.
- Onay tuşunun basılı olması gerekir.

Prensip

Onay tuşuna basılarak sürücüler etkinleştirilmektedir. Bir sürüş tuşu veya Space Mouse faresi kullanıldığında, robot akslarının düzenlenmesine başlanır ve istenen hareket yürütülür.

Sürekli hareket ve artırımlı hareket olanağı var. Bu amaçla durum çubuğunda artırım (inkrement) büyüklüğünü seçin.

Aşağıdaki mesaj, manuel işletimi etkilemektedir:

Mesaj	Nedeni	Çözümü
"Etkin komutlar bloke edildi"	Etkin komutların kilitlenmesine neden olan bir (DURDURMA) mesaj veya durum söz konusu (örn. ACİL-DUR basılı veya tahrikler henüz hazır değil).	ACİL-DURDURMA düğmelerinin açın ve/veya mesaj penceresindeki mesajları onaylayın. Bir onay tuşuna basıldıktan sonra sürücüler derhal kullanıma hazırdır.
"Yazılım son şalteri-A5"	Görüntülenen aksın yazılım son şalterine (örn. A5), belirtilen yönde (+ veya -) gidildi.	Görüntülenen aksı ters yönde surun.

Aksa özgü el yöntemine ilişkin güvenlik uyarıları

İşletim türü

Robotun elle işletilmesine sadece T1 (manuel azaltılmış hız) işletim türünde izin verilmiştir. Elle sürme hızı, T1 modunda azami 250 mm/s'dir. İşletim türü, bağlantı yöneticisi üzerinden ayarlanmaktadır.

Onay şalteri

Robotu sürebilmek için, bir onay şalterinin basılı olması gerekir. smartPAD cihazında 3 adet onay şalteri vardır. Onay şalterlerinin 3 konumu var:

- Basılı değil
- Orta konum
- Basılı (panik konumu)

Yazılım son şalteri

Robotun hareketi, aksa özgü el yöntemi sırasında da yazılım son şalterlerinin pozitif ve negatif azami değerleriyle sınırlandırılmaktadır.

DUYURU Mesaj penceresinde "Ayarın yapılması" mesajı görüntülenirse bu sınırların ötesine de sürüş yapılabilir. Bu durum robot sisteminde hasara yol açabilir!

İzlenecek yön-tem: Aksa özgü hareketi uygulama

1. Sürüş tuşları için seçenek olarak **Akslar** seçimi yapın



2. Elle Override ayarlayın



3. Onay şalterini orta konuma bastırın ve basılı tutun



Hareket tuşlarının yanısıra A1 ila A6 aksları görüntülenir.

4. Bir aksı pozitif veya negatif yönde hareket ettirmek için artı veya eksi hareket tuşuna basın.



Robotu acil durumlarda kumanda olmadan hareket ettirme



Resim 6-6: Serbest dönme düzeneği

Açıklama

Robot sistemi, serbest dönüş tertibatı ile bir kaza veya arıza durumundan sonra mekanik olarak hareket ettirilebilir. Serbest döndürme tertibatı, ana aksın tahrik motorları için ve robot tipine bağlı olarak el aksının tahrik motorları için de kullanılabilir. Bu tertibat **sadece** istisnai ve acil durumlarda, örn. şahısların kurtarılması amacıyla kullanılabilir. Serbest dönüş tertibatı kullanıldıktan sonra ilgili motorların değiştirilmesi gerekmektedir.



UYARI Motorlar, işletim sırasında ciltte yanmalara neden olabilecek sıcaklıklara ulaşmaktadır. Temas etmekten kaçınılmalıdır. Uygun koruyucu önlemler alınmalıdır, örn. koruyucu eldiven kullanılmalıdır.

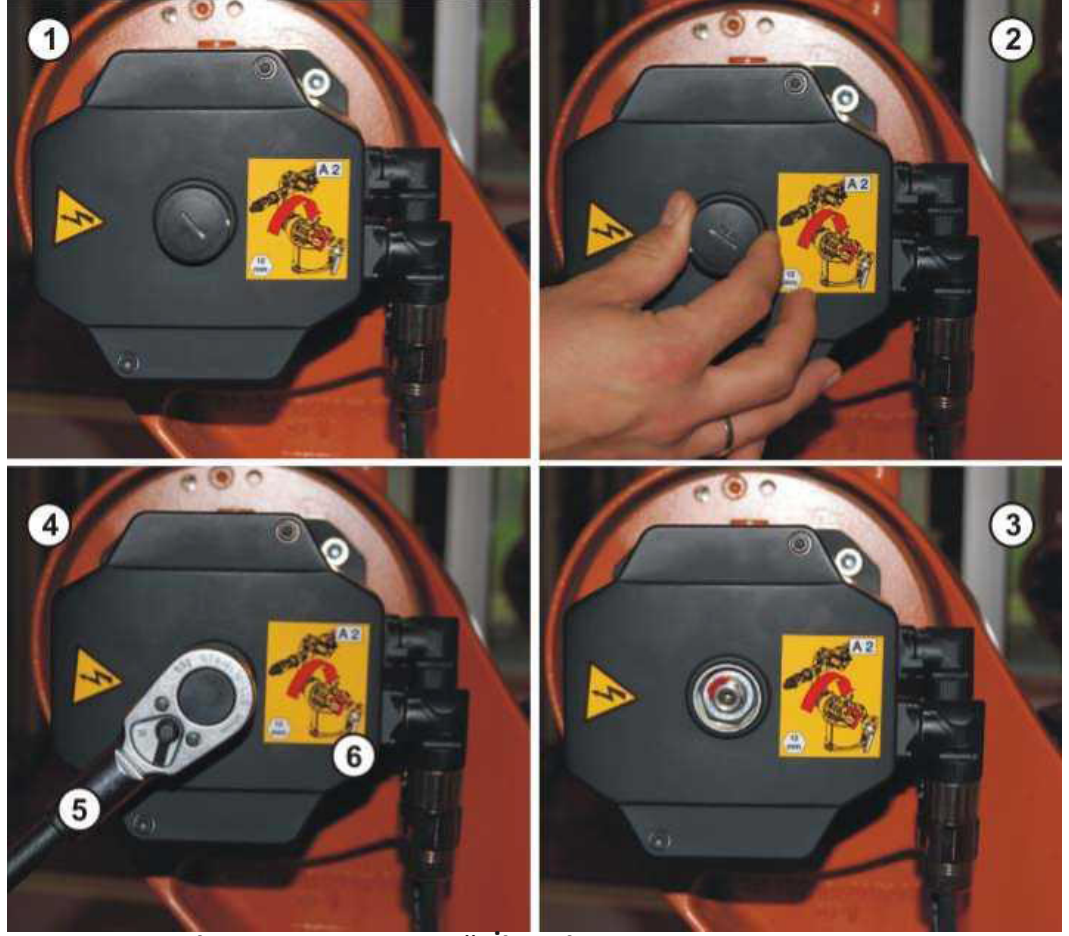
İzlenecek yöntem

1. Robot kumandasını kapatın ve izinsiz bir şekilde tekrar açılmasını önlemek için emniyete (örn. bir asma kilitle) alın.

2. Motordaki koruyucu kapağı çıkartın.

3. Serbest dönüş tertibatını ilgili motora yerleştirin ve aksı istenen yöne hareket ettirin.

Opsiyonel olarak, yönlerin motorlar üzerinde oklarla işaretlenmesi sipariş edilebilir. Mekanik motor freninin direnci ve şayet varsa, ilave aks yükleri aşılmalıdır.



Resim 6-7: Serbest Dönme Düzenegi İle Çalışma

Poz.	Açıklama
1	Kapalı koruma başlıklı motor A2
2	Motor A2'deki koruma başlığını çıkarma
3	Koruma başlığı çıkarılmış motor A2
4	Motor A2'ye serbest dönme düzenegini yerleştirme
5	Serbest döndürme tertibatı
6	Dönme yönünü gösteren levha (opsiyonel)

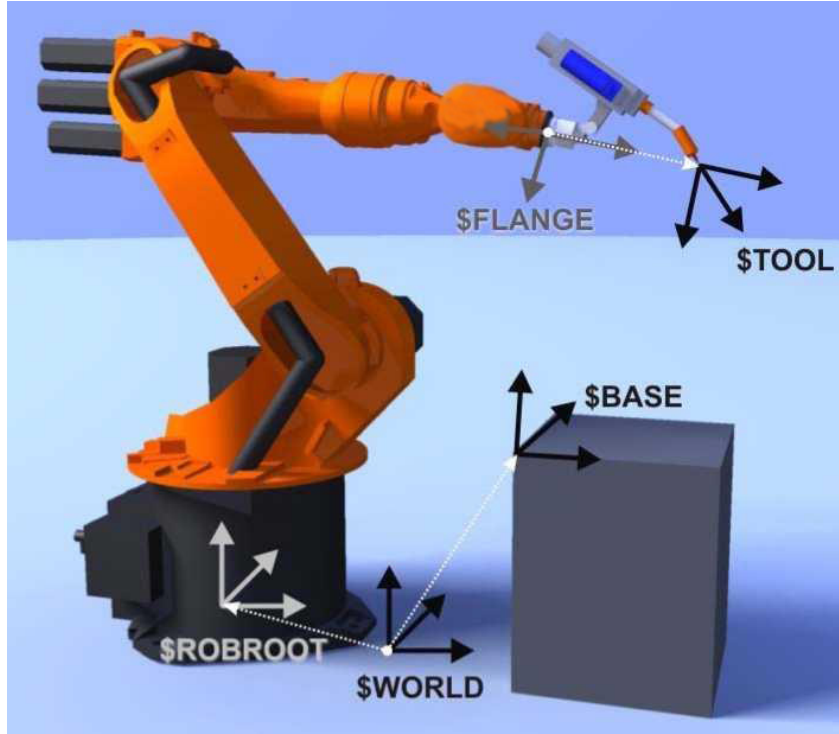
⚠ DİKKAT Bir aksın serbest dönüş düzenegi ile hareket ettirilmesi sırasında motor freni hasar görebilir. İnsanlar yaralanabilir ve maddi hasarlar meydana gelebilir. Serbest dönüş tertibatının kullanılmasından sonra ilgili motor değiştirilmek zorundadır.

i Diğer bilgiler robot için kullanım ve montaj klavuzunda bulunmaktadır.

6.6 Robotla İlişkili Koordinat Sistemleri

Endüstriyel robotların kullanımı, programlanması ve devreye alınması işlemlerinde koordinat sistemleri büyük bir öneme sahiptir. Robot kumandasında aşağıdaki koordinat sistemleri tanımlanmıştır:

- WORLD | Dünya koordinat sistemi
- ROBROOT | Robot ayağı koordinat sistemi
- BASE | Temel koordinat sistemi
- FLANGE | Flanş koordinat sistemi
- TOOL | Alet koordinat sistemi

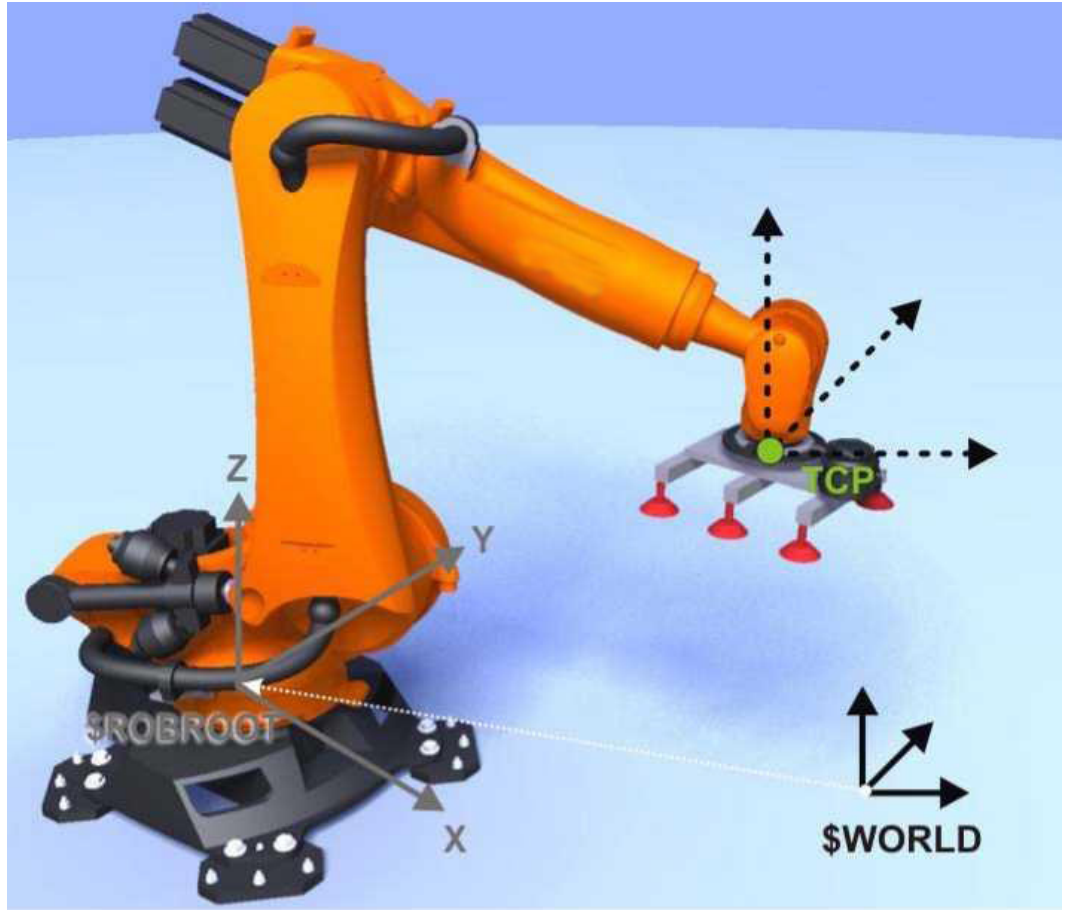


Resim 6-8: Robotun koordinat sistemleri

Adı	Yer	Kullanım	Özelliği
WORLD	serbestçe tanımlanabilir	ROBROOT ve BASE için kaynak	Genelde robot ayağında yer alır
ROBROOT	robot ayağında sabittir	Robotun başlangıç noktası	Robotun WORLD koordinat sistemine göre olan pozisyonunu tarif eder
BASE	serbestçe tanımlanabilir	İş parçaları, düzenekler	WORLD koordinat sistemine göre Base pozisyonunu tarif eder
FLANGE	robot flanşında sabittir	TOOL için başlangıç noktasıdır	Başlangıç noktası robot flanşının ortasıdır
TOOL	serbestçe tanımlanabilir	Aletler	TOOL koordinat sisteminin kaynağı " TCP" olarak adlandırılmaktadır (TCP = Tool Center Point alet merkez noktası)

6.7 Robotu Dünya Koordinat Sisteminde Hareket Ettirme

Dünya koordinat sisteminde hareket



Resim 6-9: Dünya koordinat sisteminde elle sürme prensibi

- Robotun aleti, dünya koordinat sisteminin koordinat yönlerine göre hareket ettirilebilir.

Bu esnada **tüm** robot aksları hareket eder.

- Bu amaçla smartPAD sürüş tuşları veya Space Mouse faresi kullanılır.
- Standart ayarda dünya koordinat sistemi robot ayağında yer alır (Robro- ot).
- Hız değiştirilebilir (Hand (elle) Over-Ride: HOV)
- Elle sürüş, sadece T1 işletim türünde olanaklıdır.
- Onay tuşunun basılı olması gerekir.

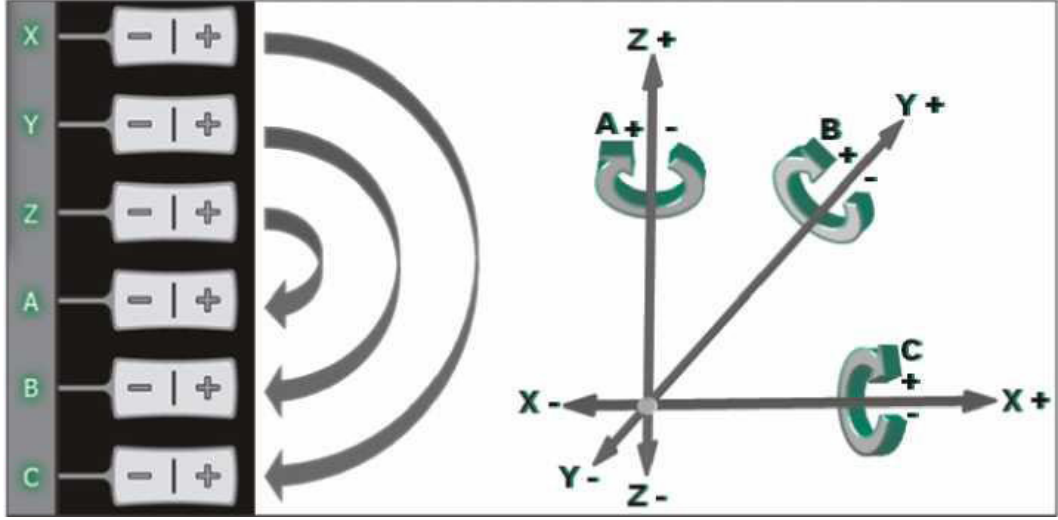
Space Mouse

- Space Mouse faresi, robotun sezgisel olarak hareket ettirilmesine olanak sağlar ve dünya koordinat sisteminde elle sürüş için ideal seçimdir.
- Fare pozisyonu ve serbestlik dereceleri değiştirilebilir durumdadır.

Dünya koordinat sisteminde elle hareket yöntemi

Bir robot bir koordinat sisteminde iki farklı şekilde hareket ettirilebilir.

- Koordinat sisteminin oryantasyon yönleri boyunca ötelemeli (doğrusal) olarak: X, Y, Z.
- Koordinat sisteminin oryantasyon yönleri etrafında dönel (döndürülerek/ çevrilerek) olarak: A, B ve C açıları



Resim 6-10: Kartezyen koordinat sistemi

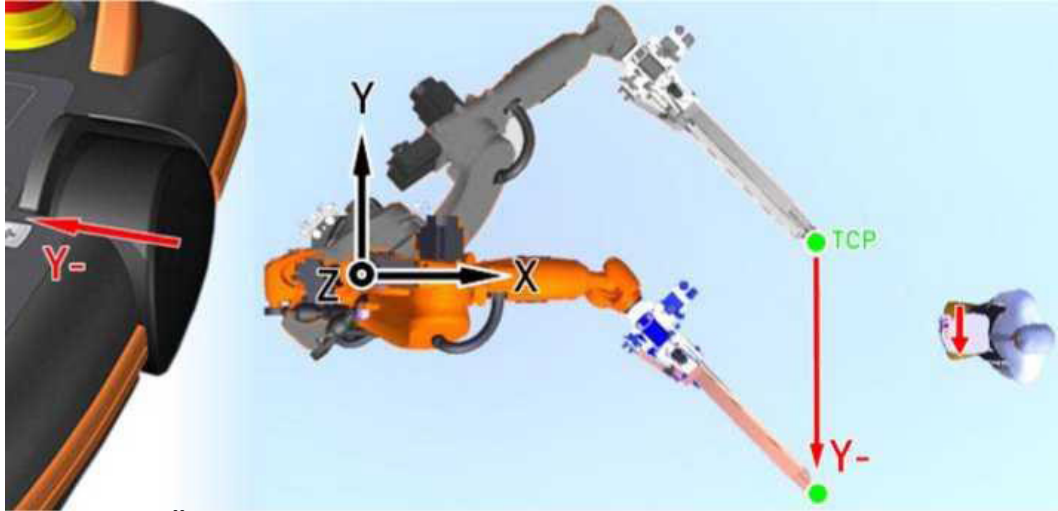
Bir sürüş komutu verildiğinde (örn. bir hareket tuşuna basıldığında) kumanda ilk olarak bir yol hesap eder. Yolun başlangıç noktası alet referans noktasıdır (TCP). Yolun yönünü dünya koordinat sistemini verir. Kumanda sistemi tüm aksları, alet bu yolda (öteleme) ilerleyecek veya dönecek (dönel hareket) şekilde regüle eder.

Dünya koordinat sistemi kullanımındaki avantajlar:

- Robotun hareketi daima öngörülebilir durumdadır.
- Hareketler daima kesindir çünkü başlangıç noktası ve koordinat yönleri daima bilinmektedir.
- Dünya koordinat sistemi ayarlı bir robotta daima kullanılabilir.
- Space Mouse ile sezgisel bir kullanım mümkündür.

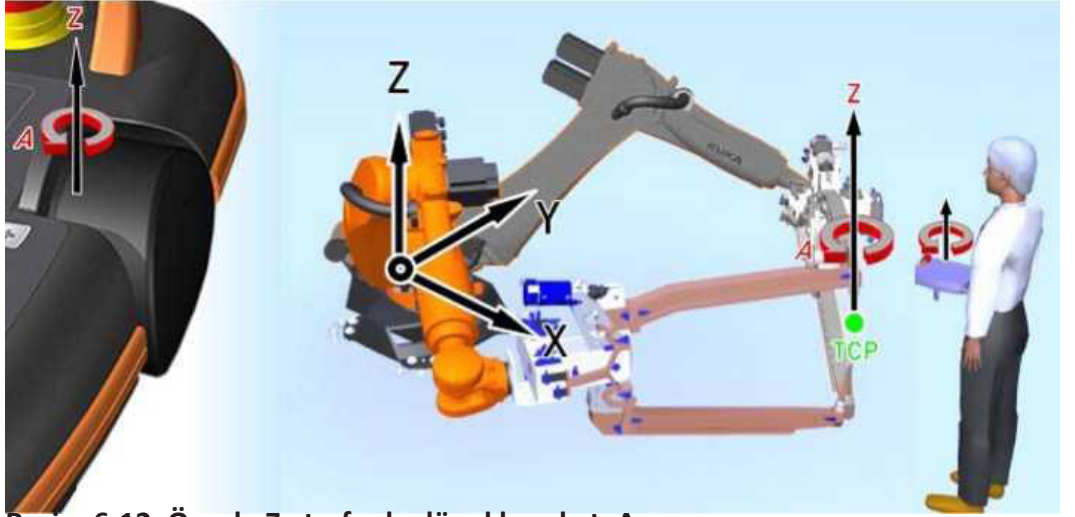
Space Mouse kullanımı

- Tüm hareket türleri Space Mouse ile mümkündür:
 - Öteleme: Space Mouse bastırılıp çekilerek



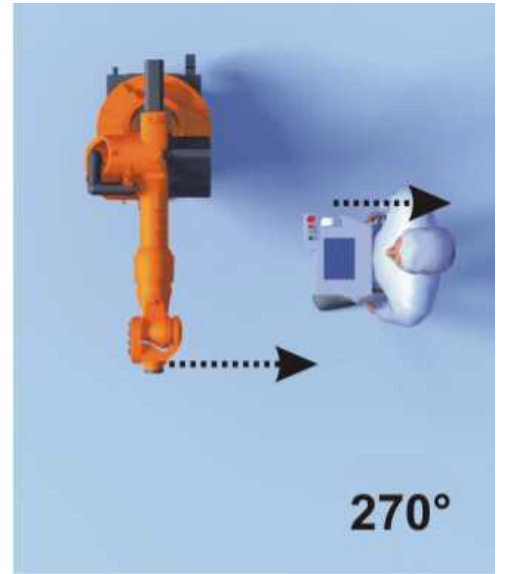
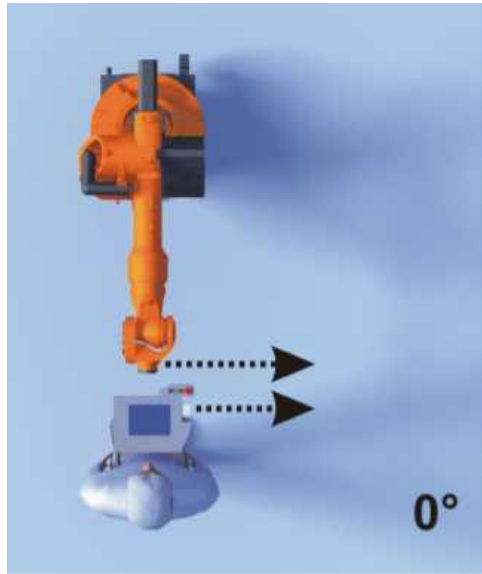
Resim 6-11: Örnek: Sola hareket

- Dönel: Space Mouse bastırılıp çevrilerek



Resim 6-12: Örnek: Z etrafında dönel hareket: A açısı

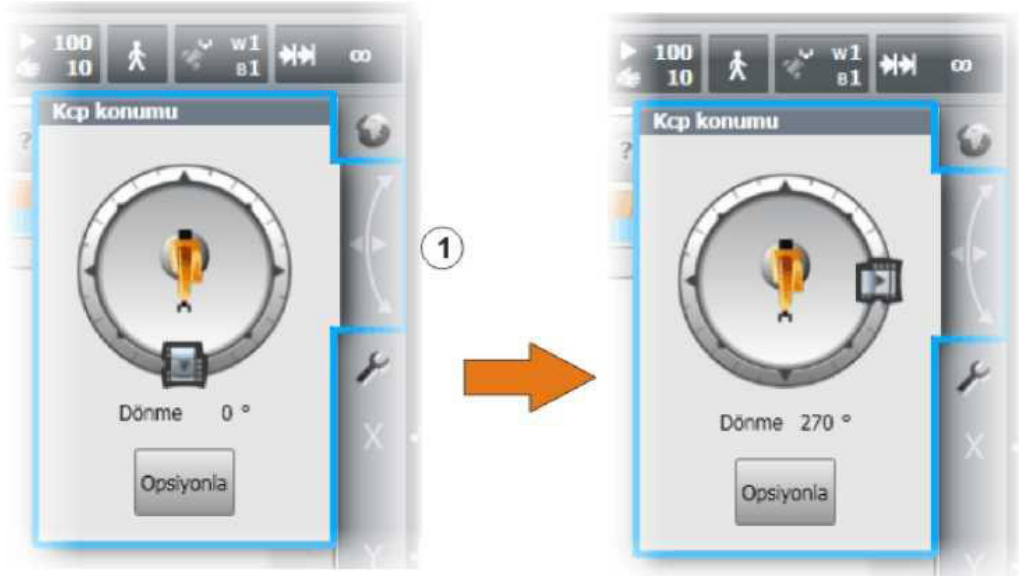
- İnsan ve robot konumuna bağlı olarak Space Mouse pozisyonu uyarlanabilir. Bilinmektedir.



Resim 6-13: Space Mouse: 0° ve 270°

Otelemeli hareket yürütme (dünya)

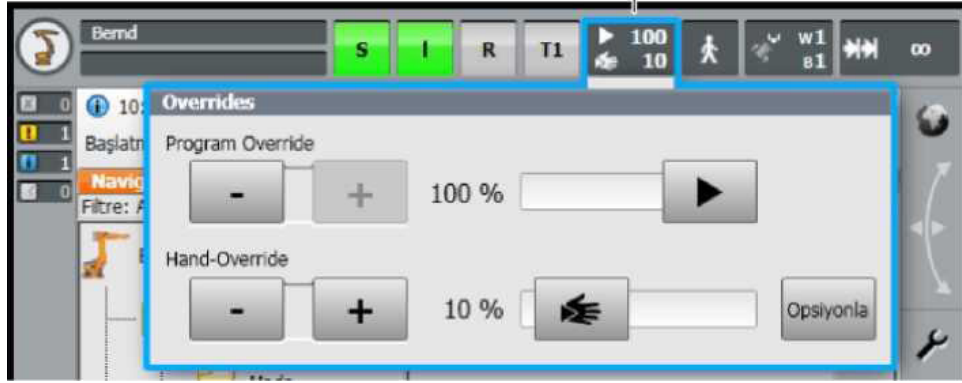
1. Kcp pozisyonunu ayar çubuğunu (1) kaydırarak ayarlayın



2. Seçenek olarak Space Mouse için Dünya seçin



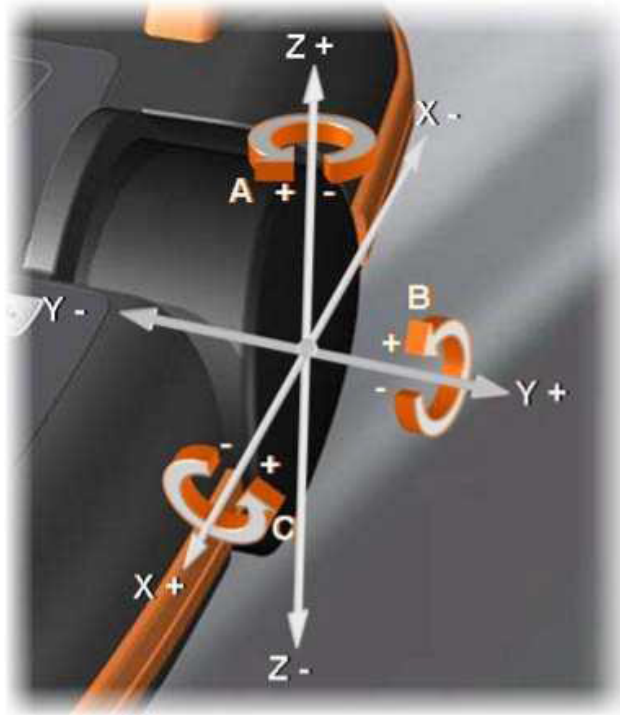
3. Elle Override ayarlayın



4. Onay şalterini orta konuma bastırın ve basılı tutun



5. Robotu Space Mouse ile ilgili yöne hareket ettirin



6. Alternatif olarak hareket tuşları da kullanılabilir



6.8 Robotu Tool Koordinat Sisteminde Hareket Ettirme

Tool koordinat sisteminde elle sürme prensibi

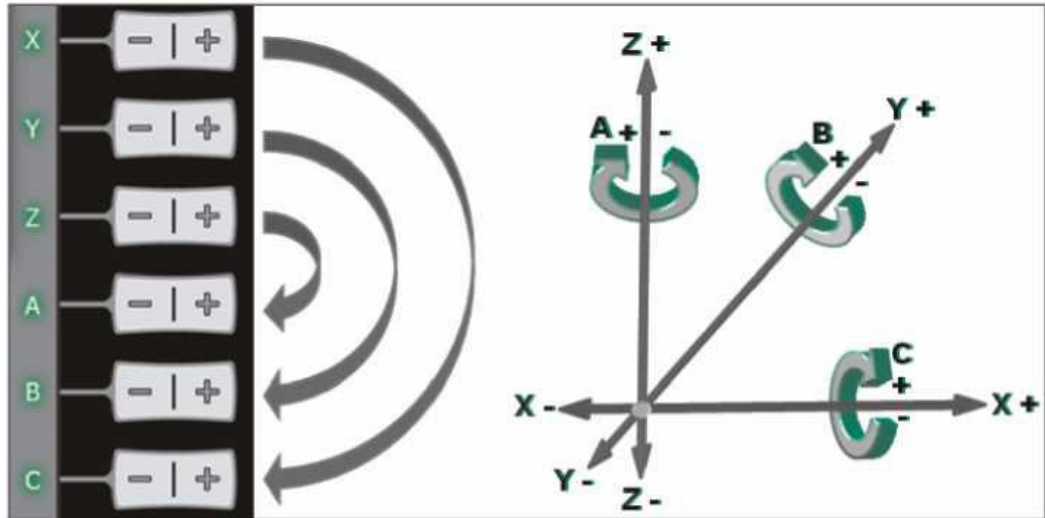


Resim 6-14: Robot Tool koordinat sistemi

- Tool koordinat sisteminde robotu, daha önce ölçülmesi yapılmış bir aletin koordinat yönlerine göre hareket ettirmek mümkün olur. Dolayısıyla koordinat sistemi yere bağlı değildir (bkz. World- / Base koordinat sistemi), robot tarafından yönlendirilmektedir. Bu esnada gerekli **tüm** robot eksenleri hareket eder. Bunun hangi eksenler olduğu sistem tarafından belirlenir ve harekete bağlıdır. Tool koordinat sisteminin başlangıç noktası **TCP** olarak adlandırılmaktadır ve aletin çalışma noktasına karşılık gelir.
- Bu amaçla smartPAD sürüş tuşları veya Space Mouse faresi kullanılır.
- 16 farklı Tool koordinat sistemi seçilebilir.
- Hız değiştirilebilir (Hand (elle) Over-Ride: HOV).
- Elle sürüş, sadece T1 işletim türünde olanaklıdır.
- Onay tuşunun basılı olması gerekir.



Tool manuel sürüş prensibi



Resim 6-15: Kartezyen koordinat sistemi

Bir robot bir koordinat sisteminde iki farklı şekilde hareket ettirilebilir.

- Koordinat sisteminin yönelme yönleri boyunca ötelemeli (doğrusal olarak) olarak: X, Y, Z
- Dönel (döndürülerek / çevrilerek) koordinat sisteminin yönelme yönleri etrafında: A, B ve C açıları

Tool koordinat sistemi kullanımındaki avantajlar:

- Tool koordinat sistemi bilindiğinde robotun hareketi daima öngörülebilir olur.
- Alet etkime yönünde sürüş yapma veya TCP etrafında yöneltme yapma olanağı vardır.
- *Alet etkime* yönü, aletin çalışma veya proses yönüdür: Bir yapıştırma nozu- lunda yapıştırıcının çıkış yönü, bir iş parçası kavrandığında kavrama yönü vs.

İzlenecek yöntem

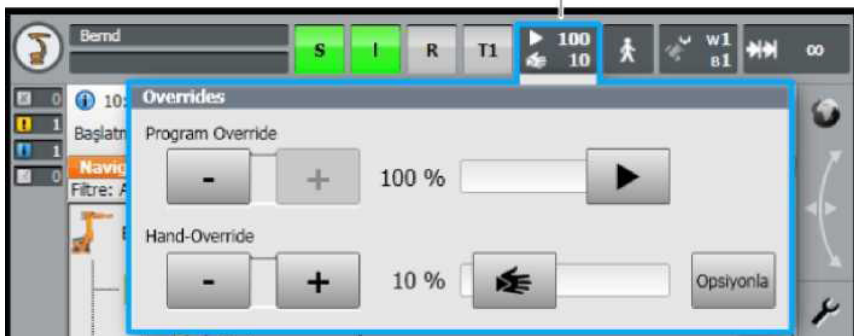
1. Kullanılacak koordinat sistemi olarak Alet seçin



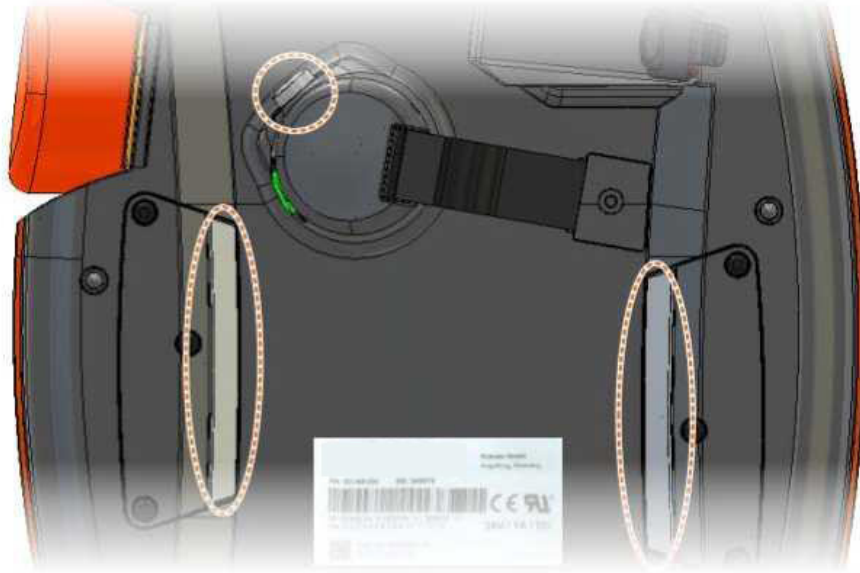
2. Alet numarasını seçin



3. Elle Override ayarlayın



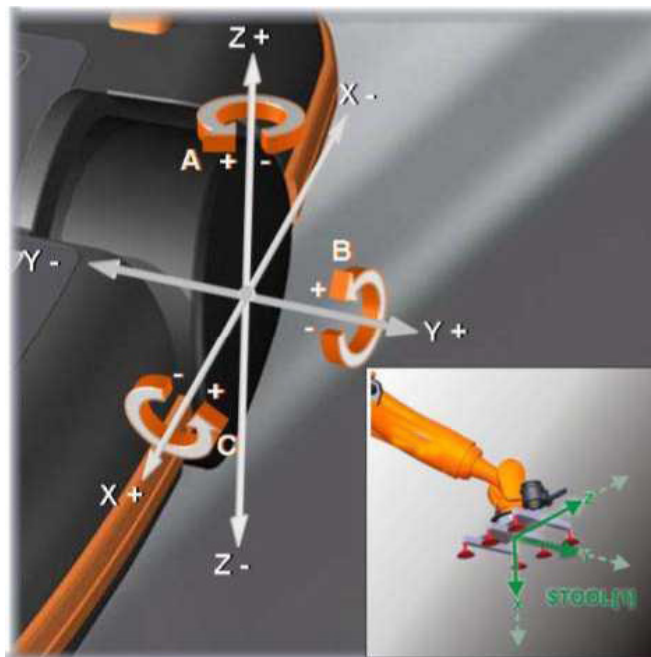
4. Onay şalterini orta konuma bastırın ve basılı tutun



5. Robotu sürüş tuşlarıyla hareket ettirin

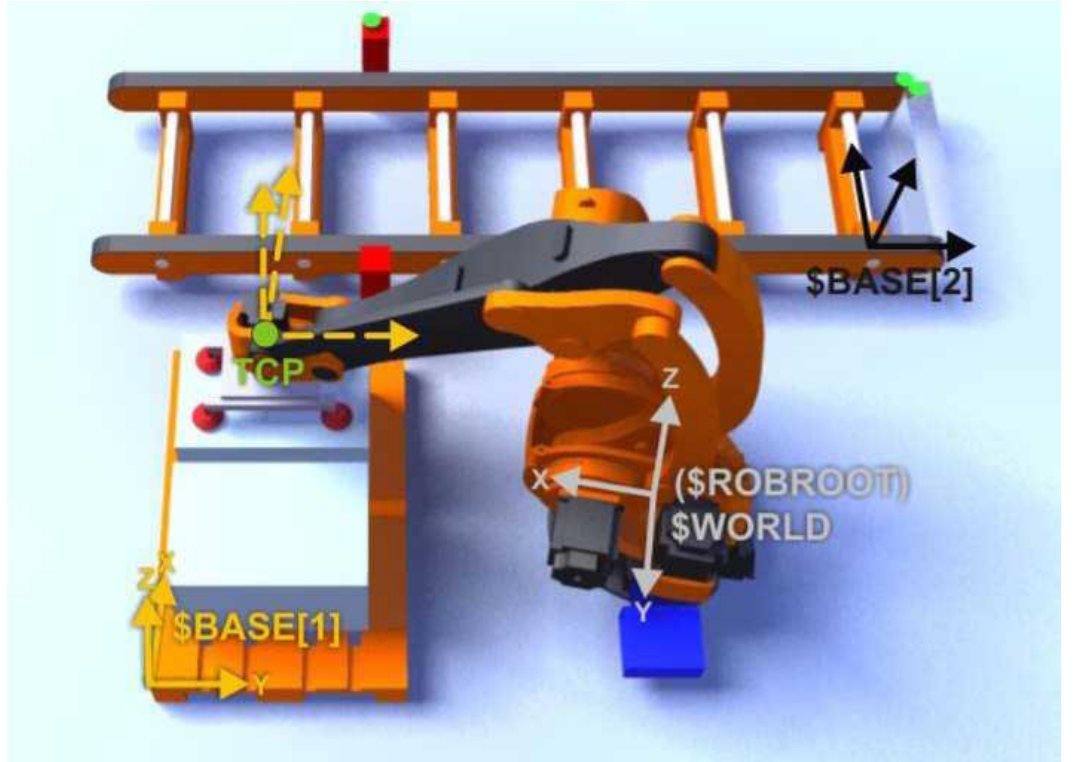


6. Alternatif: Robotu Space Mouse ile ilgili yöne hareket ettirin



6.9 Robotu Base Koordinat Sisteminde Hareket Ettirme

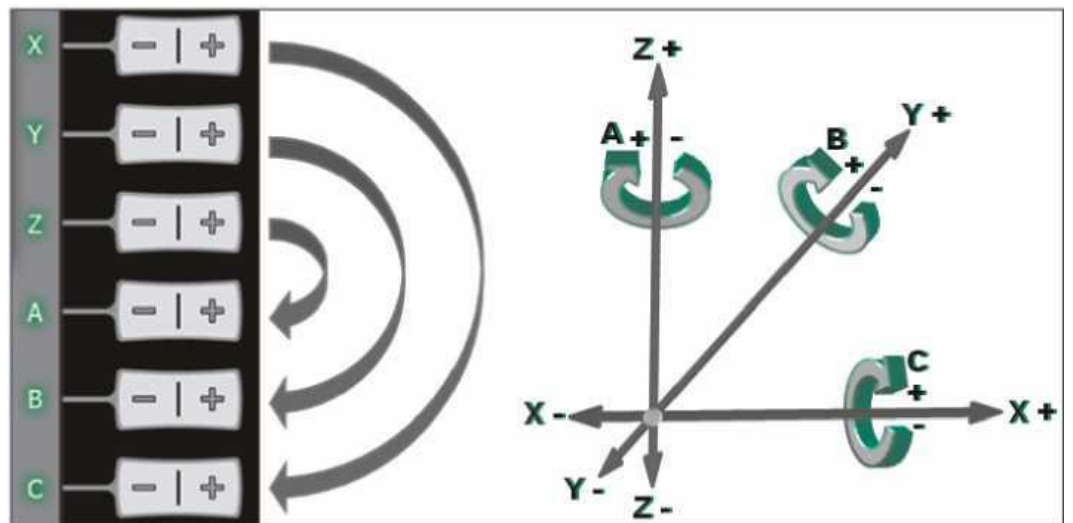
Temel koordinat sisteminde hareket



Resim 6-16: Base koordinat sisteminde manüel sürüş
Base (Temel) tarifi

- Robotun taşıdığı alet, temel koordinat sisteminin koordinat yönlerine göre hareket ettirilebilir. Temel koordinat sistemleri münferit olarak ölçümlenmiş olabilir ve genelde iş parçası kenarları, iş parçası braketleri veya paletlere göre bir yönelime sahiptirler. Bu sayede konforlu bir manüel sürüş mümkündür! Bu esnada gerekli **tüm** robot eksenleri hareket eder. Bunun hangi eksenler olduğu sistem tarafından belirlenir ve harekete bağlıdır.
- Bu amaçla smartPAD sürüş tuşları veya Space Mouse faresi kullanılır.
- 32 temel koordinat sistemi seçilebilir.
- Hız değiştirilebilir (Hand (elle) Over-Ride: HOV).
- Elle sürüş, sadece T1 işletim türünde olanaklıdır.
- Onay tuşunun basılı olması gerekir.

Temel koordinat sisteminde elle sürme prensibi



Resim 6-17: Kartezyen koordinat sistemi

Bir robot bir koordinat sisteminde iki farklı şekilde hareket ettirilebilir.

- Koordinat sisteminin yönelme yönleri boyunca ötelemeli (doğrusal olarak) olarak: X, Y, Z
- Koordinat sisteminin oryantasyon yönleri etrafında dönel (döndürülerek/ çevrilerek) olarak: A, B ve C açıları

Bir sürüş komutu verildiğinde (örn. bir hareket tuşuna basıldığında) kumanda ilk olarak bir yol hesap eder. Yolun başlangıç noktası alet referans noktasıdır (TCP). Yolun yönünü dünya koordinat sistemini verir. Kumanda sistemi tüm aksları, alet bu yolda (öteleme) ilerleyecek veya dönecek (dönel hareket) şekilde regüle eder.

Temel koordinat sistemi kullanımındaki avantajlar:

- Temel koordinat sistemi bilindiğinde robotun hareketi daima öngörülebilir olur.
- Burada da Space Mouse ile sezgisel bir kullanım mümkündür. Bunun önkoşulu, operatörün robota veya temel koordinat sistemine göre doğru durmasıdır.



Ek olarak doğru Tool koordinat sistemi ayarlanırsa temel koordinat sisteminde TCP etrafında yeniden yöneltme yapılabilir.

İzlenecek yöntem

1. Sürüş tuşları için seçenek olarak **Base** seçimi yapın



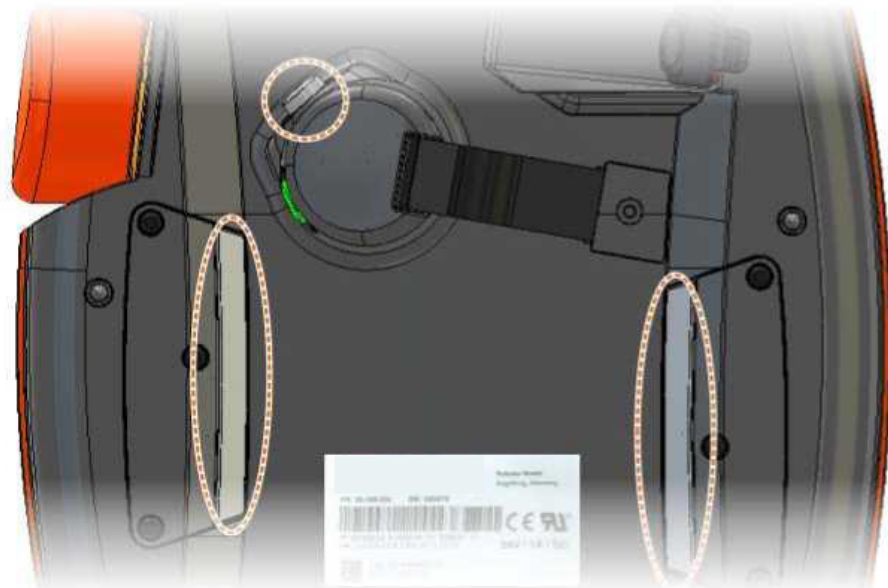
2. Tool ve Base seçin



3. Elle Override ayarlayın



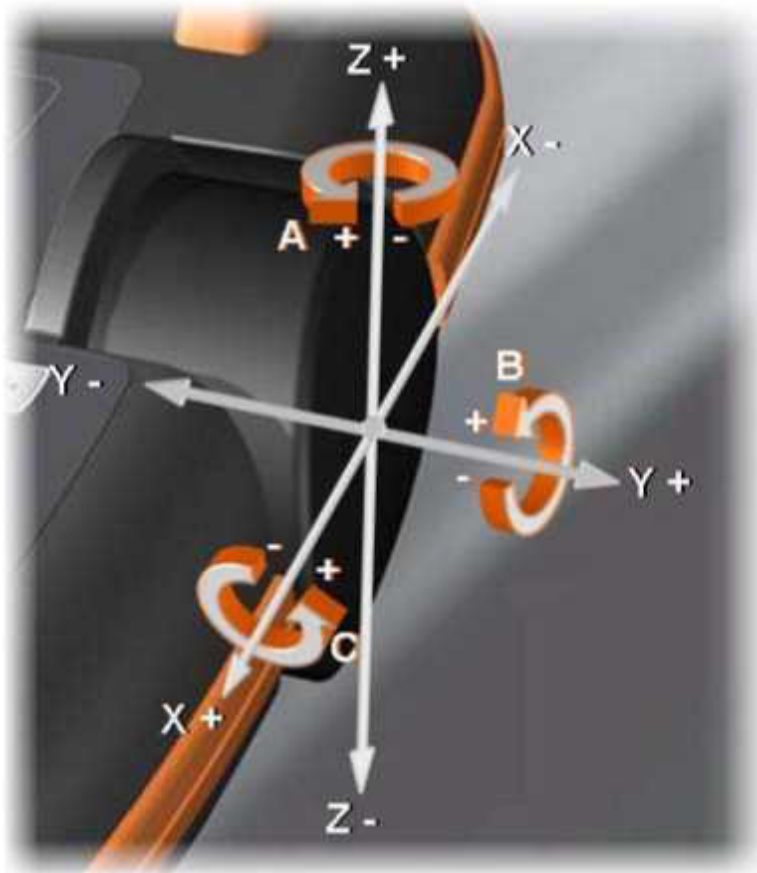
4. Onay şalterini orta konuma bastırın ve basılı tutun



5. Sürüş tuşları ile istediğiniz yöne sürüş yapın



6. Alternatif olarak Space Mouse ile sürüş yapılabilir



7 İşletime Alma

7.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Robot ayarı
- Alet ölçümü
- Yük verileri
- Base ölçümü
- Güncel robot pozisyonunu gösterilmesi

7.2 Ayar Prensipli

**Ayarlama
Neden
Gerekli?**

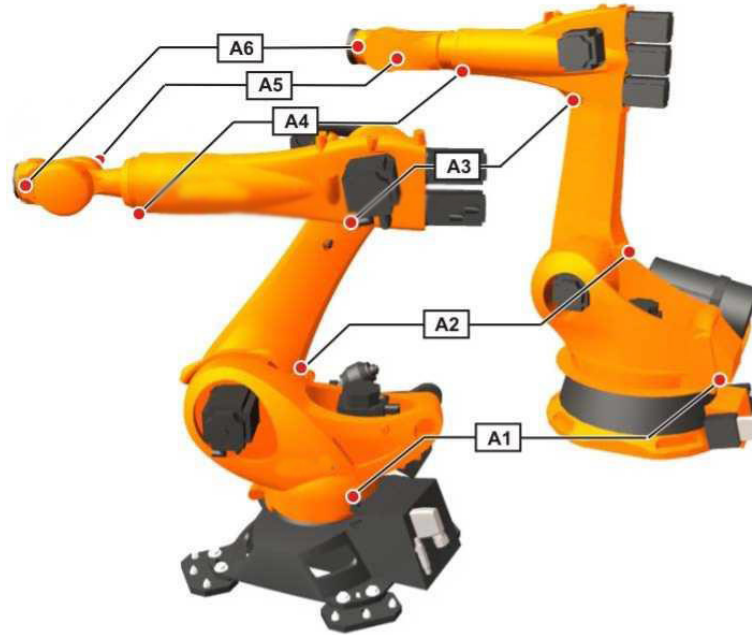
Bir endüstriyel robot sadece tümüyle ve doğru olarak ayarlıysa en iyi şekilde kullanılabilir. Çünkü sadece bu durumda tam noktasal ve yörünge hassasiyetini gösterir veya programlanmış hareketlerle hareket ettirilebilir.



Ayar sırasında her robot aksına bir referans değeri atanır.

Komple bir ayar işlemi her bir aksın tek tek ayarlanmasını içerir. Teknik bir yardımcı gereçle (EMD = Electronic Mastering Device) her aksa, kendi mekanik sıfır konumunda bir referans değeri atanır (örn. 0°). Bu sayede aksın mekanik ile elektrik konumunda uyum sağlandığı için, her aksa belli bir açı değeri verilir.

Ayar konumu her robotta benzerdir, ancak aynı değildir. Tam pozisyonlar bir robot tipinin münferit robotları arasında da farklı olabilir.



Resim 7-1: Ayar kartuşlarının pozisyonları

Mekanik sıfır konumunun açı değerleri (=referans değerleri)

Aks	"Quantec" robot nesli	diğer robot tipleri (örn. Seri 2000, KR 16 vs.)
A1	-20°	0°
A2	-120°	-90°
A3	+110°	+90°
A4	0°	0°
A5	0°	0°
A6	0°	0°

Ayarlama Neden Yapılır? Genel kural olarak bir robotun daima ayarlı olması gerekir. Aşağıdaki durumlarda ayarın yapılması zorunludur.

- Devreye alma işleminde.
- Pozisyon değerleri tespitine katkıda bulunan komponentlerde onarım tedbirleri uygulandığında (örn. enkoderli motor ya da RDC)
- Robot aksları kumanda sistemi olmadan hareket ettirildiyse, örn. serbest dönürme tertibatıyla.
- Mekanik onarımlardan / sorunlardan sonra önce robotun ayarı bozulacak ve sonra ayar yapılacaktır.
 - bir dişli kutusu değiştirildikten sonra
 - 250 mm/saniye üzerinde bir hızla son dayanağa çarptıktan sonra
 - bir çarpışmadan sonra

i Onarım tedbirlerinden önce genel olarak güncel ayar kontrol edilecektir.

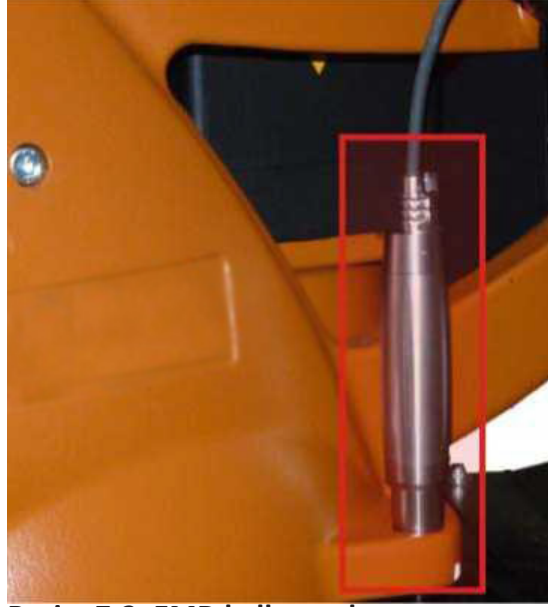
Ayara İlişkin Güvenlik Uyarıları

Ayarsız robot eksenleri olduğunda robotun işleyişi kısıtlıdır:

- Program işletimi olanaklı değil: Programlanan noktalara gidilmesi mümkün değil.
- Ötelemeli manüel sürüş yok: Koordinat sistemlerinde hareketler mümkün değil.
- Yazılım limit şalterleri devre dışıdır.

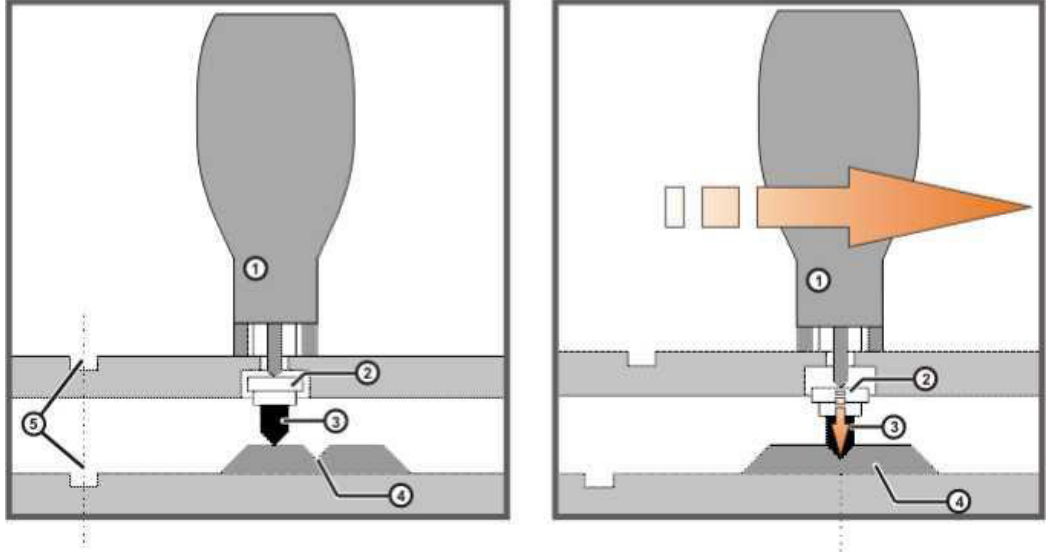
DUYURU Ayarı kaçmış bir robotta yazılım son şalterleri devre dışı bırakılmıştır. Robot, son dayanaklardaki tamponlara karşı çarpabilir, bu yüzden hasar görülebilir ve tamponlar değiştirilmelidir. Ayarı kaçmış bir robotu mümkün olduğunca hareket ettirilmemeli veya manuel Override işlemleri mümkün olduğunca azaltılmalıdır.

Ayarlamanın Yapılması



Resim 7-2: EMD kullanımda

Ayar işlemi, eksenin mekanik sıfır noktası belirlenerek yapılır. Bu işlemde mekanik sıfır noktasına varılana kadar eksen hareket eder. Bu durum, ölçüm kalemi ölçüm çentiği içerisinde en derin noktaya vardığında sağlanmış olur. Bu nedenle her eksen bir ayar kartuşu ve bir ayar işaretiyle donatılmıştır.



Resim 7-3: EMD ayarı süreci

- 1 EMD (Electronic Mastering Device - Elektronik Ayar Cihazı)
- 2 Ölçüm kartuşu
- 3 Ölçüm kalemi

- 4 Ölçüm çentiği
- 5 Ön ayar işareti

7.3 Robotu Ayarlama

Robot ayarı
olanakları



Resim 7-4: Ayar olanakları

1. Robotu ön ayar konumuna getirin.

Standart ayar-
lamada izlene-
cek yöntem



Resim 7-5: Ön ayar konumu örnekleri

2. Ana menüde İşleme alma > Ayarlama > EMD > Standart > Ayar belirleme seçin.

Bir pencere açılır. Ayarlanacak olan bütün akslar gösterilir. En küçük numaralı aks işaretlenmiştir.

3. Pencerede işaretli olan aksta, ölçme kartuşunun koruyucu kapağını çıkartın. EMD ters çevrildiğinde tornavida olarak kullanılabilir. EMD'yi ölçme kartuşuna vidalayın.



Resim 7-6: EMD'yi ölçme kartuşuna vidalama

4. Ardından ölçüm hattını EMD'ye takın ve robotun bağlantı kutusundaki X32 bağlantısına bağlayın.



Resim 7-7: EMD hattı, bağlı

⚠ DİKKAT BEMD daima, ölçme kartuşuna ölçme hattı olmadan takılmalıdır. Sonra önce ölçme hattını EMD'ye takın. Aksi takdirde ölçüm hattı hasar görebilir. EMD'yi çıkarırken de aynı şekilde daima önce EMD'den ölçme hattını çıkarın. Sonra önce ölçme kartuşundan EMD'yi çıkarın. Ölçümden sonra ölçüm hattını bağlantı X32'den çıkartın. Aksi takdirde arıza sinyalleri oluşabilir veya hasar meydana gelebilir.

5. **Ayarla** üzerine basın.

6. Onay şalterini orta konuma bastırın ve start tuşuna basılı tutun.

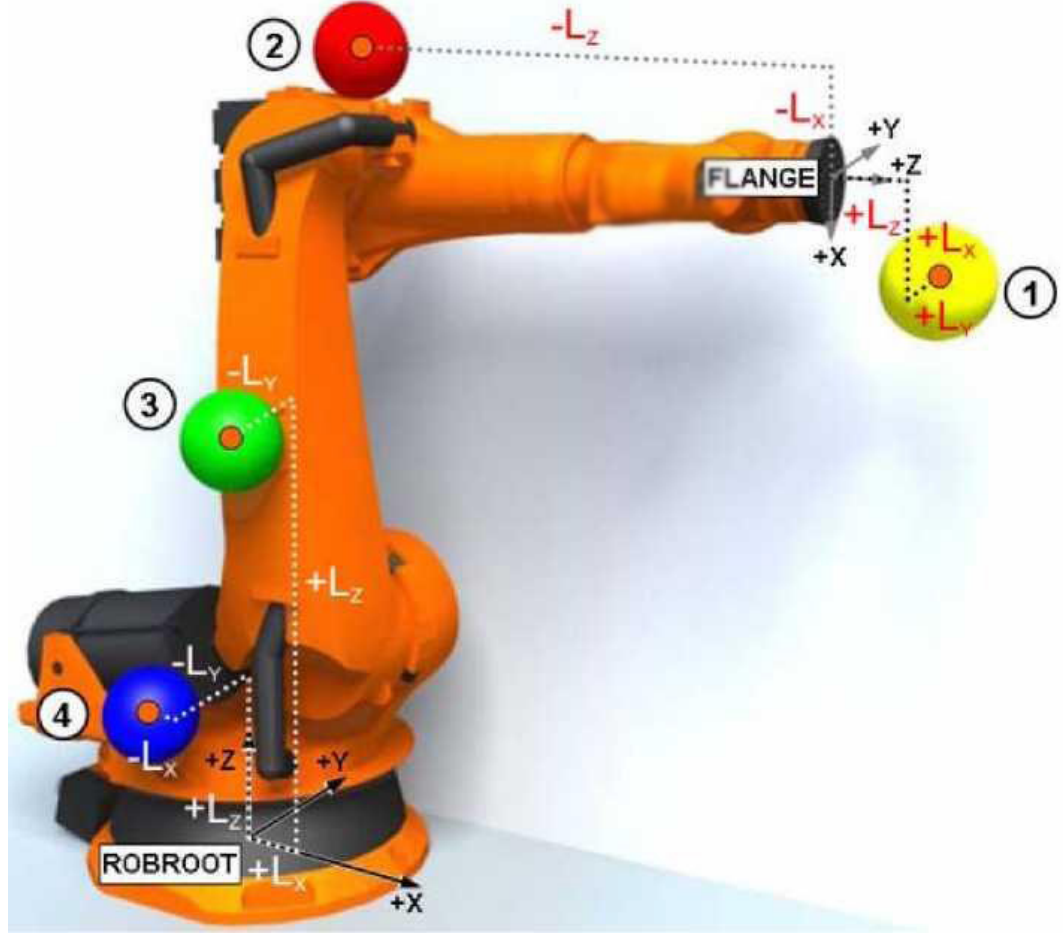


Resim 7-8: Onay ve başlatma tuşu

EMD, ölçüm çentiğinin en alt noktasını geçtikten sonra ayarlama konumuna ulaşmıştır. Robot otomatik olarak durur. Değerler kaydedilir. Pencerede aks görünmez olur.

7. Ölçme hattını EMD'den çıkartın. Sonra EMD'yi ölçme kartuşundan çıkartın ve koruyucu kapağı tekrar takın.
8. 2 - 5 arası adımları, ayarlanacak olan tüm akslar için tekrarlayın.
9. Pencereyi kapatın.
10. Ölçüm hattını X32 bağlantısından çıkartın.

7.4 Robotta Yükler



Resim 7-9: Robottaki yükler

- | | |
|---------------------|--------------------|
| 1 Taşıma kapasitesi | 3 ilave yük, aks 2 |
| 2 ilave yük, aks 3 | 4 ilave yük, aks 1 |

7.5 Alet Yük Verileri

Alet Yük Verileri Nedir?

Alet yük verileri, robot flanşına monte edilmiş olan yüklerin tümüdür. Bunlar, robota monte edilmiş olan ve birlikte hareket ettirilmesi gereken ek bir kütle oluşturur.

Kaydedilecek değerler kütle, ağırlık noktasının konumu (kütlenin etkidiği nokta), kütle atalet momentleri ve bunlara ait ana atalet akslarıdır.

Taşıma kapasitesi verileri, robot kumandasına **girilmeli** ve doğru alete **atanmalıdır**.

İstisna: Taşıma kapasitesi verileri, KUKA.LoadDataDetermination ile robot kumandasına aktarıldıktan sonra artık manuel olarak girmeye gerek yoktur.

Alet yük verileri aşağıdaki kaynaklardan alınabilir:

- Yazılım opsiyonu KUKA.LoadDetect (sadece taşıma yükleri için)
- Üretici bilgileri
- Manuel hesaplama
- CAD-Programları

Yük Verilerinin Etkileri Kayıtlı olan yük verileri çok sayıda kumanda sürecine etki etmektedir. Bunlara örn. aşağıdakiler dahildir:

- Kumanda algoritmaları (ivmenin hesaplanması)
- Hız ve ivme denetimi
- Moment denetimi
- Çarpışma denetimi
- Enerji denetimi
- ve birçok başka

Bu nedenle, yük verilerinin düzeltilmesi büyük önem taşımaktadır. Eğer robot, hareketlerini doğru girilmiş yük verileriyle yapıyorsa ...

- doğruluğun yüksek olmasından fayda sağlar,
- en iyi çevrim sürelerine sahip hareket süreçleri mümkün olur,
- robotun kullanım ömrü artar (aşınması daha az olur).

İzlenecek Yöntem

1. Ana menüde **İşletime alma > Ölçme > Alet > Alet yük verileri** seçimini yapın.

2. **Alet no.** alanında aletin numarasını girin. **Devam** ile onaylayın.

3. Taşıma kapasitesi verilerini girme:

- **M** alanı: Toprak
- **X, Y, Z** alanı: Ağırlık merkezinin yeri flanşa göre relatif
- **A, B, C** alanı: Ana atalet akslarının oryantasyonu flanşa göre relatif
- **JX, JY, JZ** alanı: Kütle atalet momentleri

(JX, koordinat sisteminin, A, B ve C ile flanşa relatif olarak bükülmüş X aksındaki ataletidir. Y ve Z aksındaki ataletler JY ve JZ analog.)

4. **Devam** ile onaylayın.

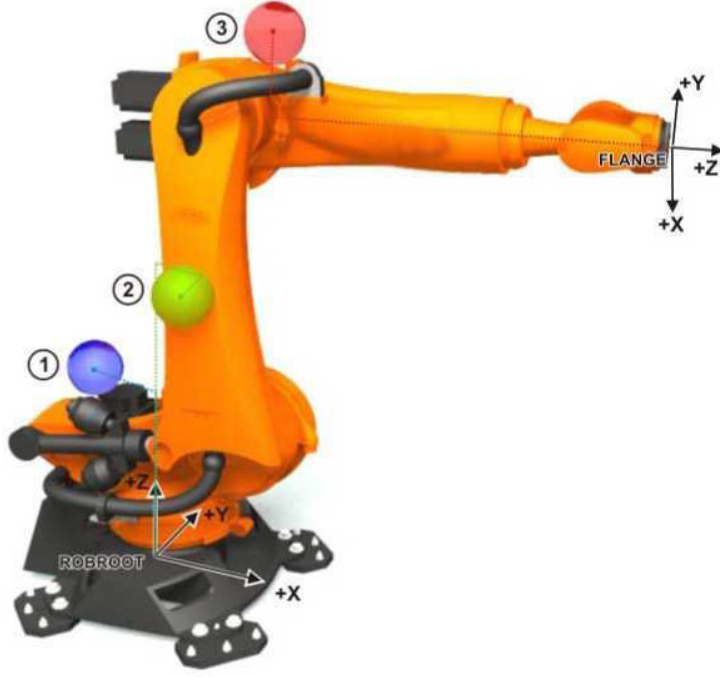
5. **Kaydet** üzerine basın.

7.6 Robotta Ek Yükler

Robotta İlave Yükler

İlave yükler, ana kasaya, bağlantı koluna veya kola ek olarak takılmış olan bileşenlerdir, örn.:

- Enerji beslemesi
- Valflar
- Malzeme beslemesi
- Malzeme stoğu



Resim 7-10: Robotta ek yükler

İlave yük verileri, robot kumandasına girilmelidir. Gerekli bilgiler arasında şunlar yer almaktadır:

- Kütle (m) kg olarak
- Kütle ağırlık noktasının, referans sistemine olan mesafesi (X, Y ve Z), mm cinsinden
- Ana atalet akslarının referans sistemine oryantasyonu (A, B ve C), derece cinsinden ($^{\circ}$)
- Kütlelerin, atalet aksları etrafındaki atalet momentleri (J_x , J_y ve J_z), kgm^2 cinsinden

İlave yük başına X, Y ve Z değerlerinin referans sistemleri:

Yük	Referans sistemi
İlave yük A1	ROBROOT koordinat sistemi $A1 = 0^{\circ}$
İlave yük A2	ROBROOT koordinat sistemi $A2 = -90^{\circ}$
İlave yük A3	FLANGE koordinat sistemi $A4 = 0^{\circ}$, $A5 = 0^{\circ}$, $A6 = 0^{\circ}$

Yük verileri aşağıdaki kaynaklardan alınabilir:

- Yazılım opsiyonu KUKA.LoadDetect (sadece taşıma yükleri için)
- Üretici bilgileri
- Manuel hesaplama
- CAD-Programları

Ek yüklerin robot hareketine olan etkisi

Yük verilerinin belirtilmesi robot hareketlerini çeşitli şekillerde etkiler:

- Rota planlama
- Hızlanmalar
- Çevrim süresi
- Aşınma

**UYARI**

Eğer bir robot yanlış yük verileriyle veya uygun olmayan yüklerle işletilirse, bedensel ve yaşamsal tehlikeler ve/veya ağır maddi hasarlar oluşabilir.

İzlenecek Yöntem

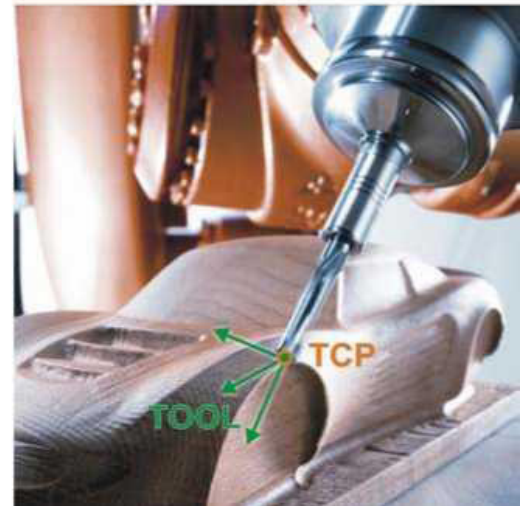
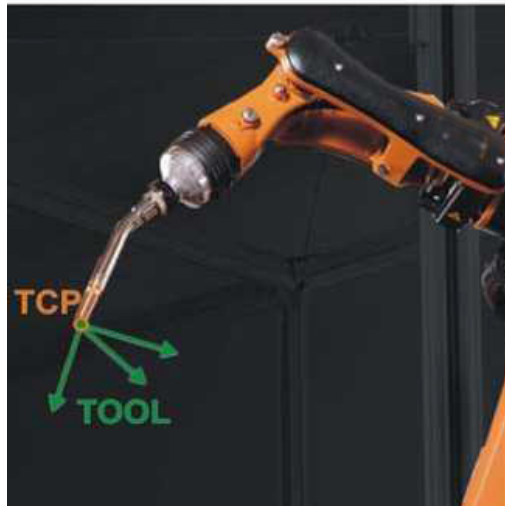
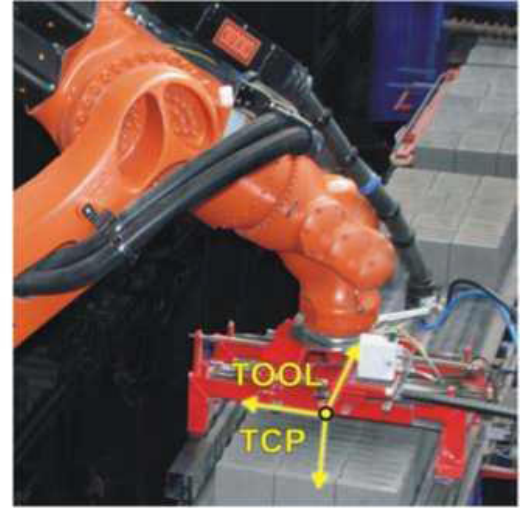
1. Ana menüde **İşletme alma > Ölçme > İlave yük verileri** seçimini yapın.
2. İlave yükün sabitlendiği aksın numarasını girin. **Devam** ile onaylayın.
3. Yük verilerini girin. **Devam** ile onaylayın.
4. **Kaydet** üzerine basın.

7.7 Bir Aleti Ölçme**Açıklama**

Bir aletin ölçümü, başlangıcı, aletin bir referans noktasında olan bir koordinat sisteminin üretilmesi demektir. Bu referans noktası TCP (Tool Center Point) olarak adlandırılır ve koordinat sistemi TOOL koordinat sistemidir.

Yani alet ölçümü şunların ölçümünü içerir ...

- TCP'nin (koordinat sistemi başlangıcı)
- Koordinat sistemi hizasının

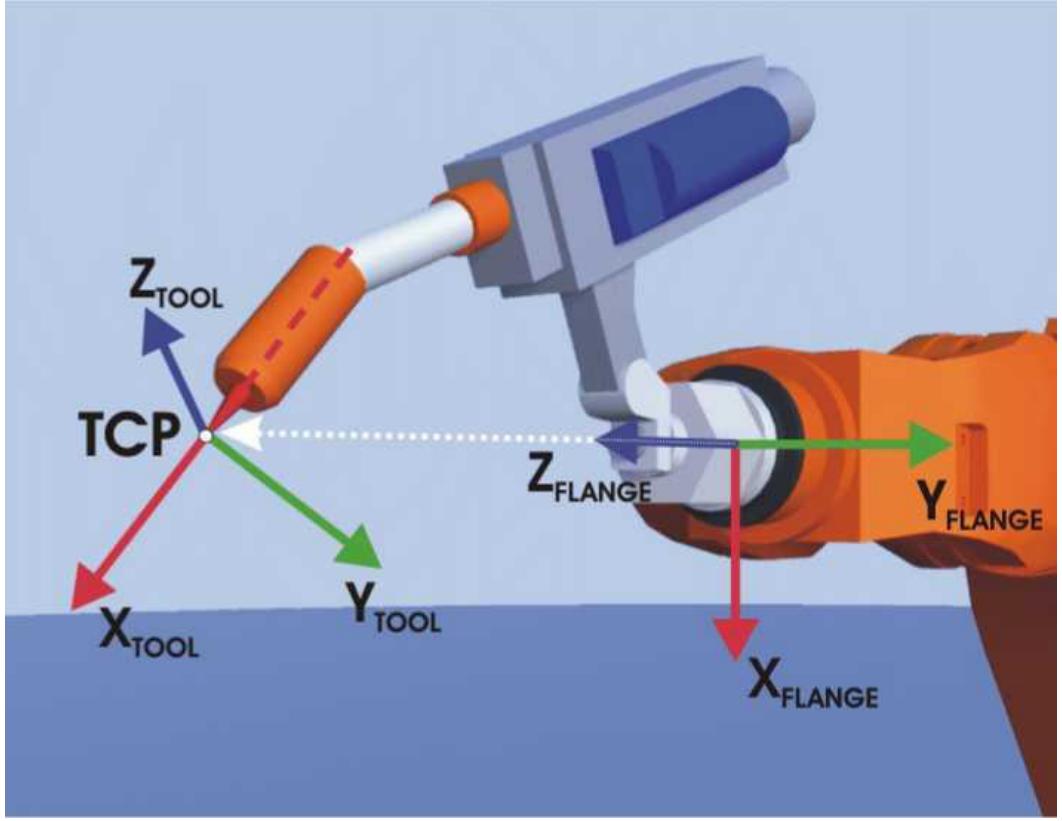


Resim 7-11: Ölçümlenmiş alet örnekleri



En çok 16 TOOL koordinat sistemi kaydedilebilir. (Değişken: TOOL_DATA[1 ... 16]).

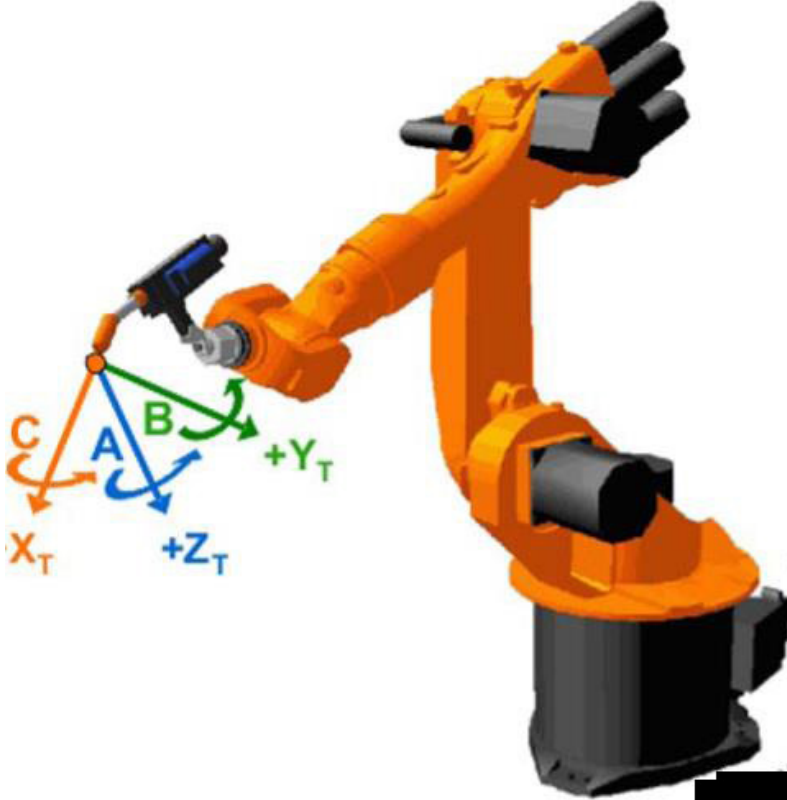
Ölçme sırasında, Tool koordinat sisteminin (X, Y ve Z olarak) flanş koordinat sistemine olan mesafesi ve de birbirlerine göre olan dönüş (A, B ve C açısı) kaydedilir.



Resim 7-12: TCP ölçümünün prensibi

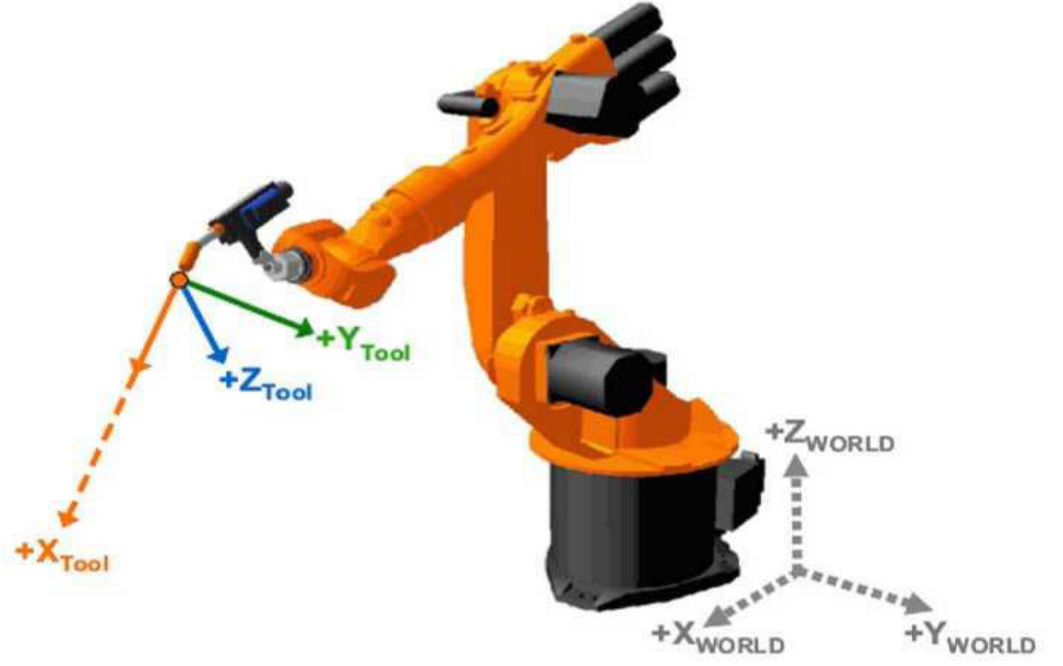
Bir alet kesin olarak ölçüldüyse, kullanıcı personel ve programlama personeli için uygulamada şu avantajlar ortaya çıkar:

- Daha iyi elle hareket yöntemi
- TCP'nin (örn. alet ucu) oryantasyonunu değiştirmek mümkündür



Resim 7-13: TCP etrafında oryantasyonu değiştirme

- Aletin çarpma yönünde hareket etme



Resim 7-14: Etkime yönü TCP

- Hareket programlamasında kullanma
- Programlanmış hareket hızı, rota boyunca TCP'de tutulur



Resim 7-15: TCP ile program işletimi

- Ayrıca rota boyunca, tanımlı bir oryantasyon mümkündür.


Alet ölçümü 2 adımdan oluşur:

**Alet ölçüm
olanakları**

Adım	Açıklama
1	TOOL koordinat sistemi başlangıcını belirleme Aşağıdaki metotlar kullanılabilir: ■ <i>XYZ 4 nokta</i> ■ <i>XYZ-Referans</i>
2	TOOL koordinat sisteminin oryantasyonunu belirleme Aşağıdaki metotlar kullanılabilir: ■ <i>ABC-World</i> ■ <i>ABC-2 nokta</i>
Alternatif	Flanşın orta noktasına olan mesafe (X, Y, Z) ve dönüş (A, B, C) için olan değerlerin doğrudan girilmesi. ■ <i>Nümerik giriş</i>

XYZ 4 nokta metoduyla TCP ölçümü

Ölçülen aletin TCP'si ile bir referans noktasına 4 farklı yönden gidilebilir. Referans noktası istenildiği şekilde seçilebilir. Robot kumanda ünitesi, farklı flanş pozisyonlarından TCP'yi hesaplar.

 Referans noktasının gidildiği 4 flanş pozisyonu birbirine göre yeterince uzak olmalıdır.

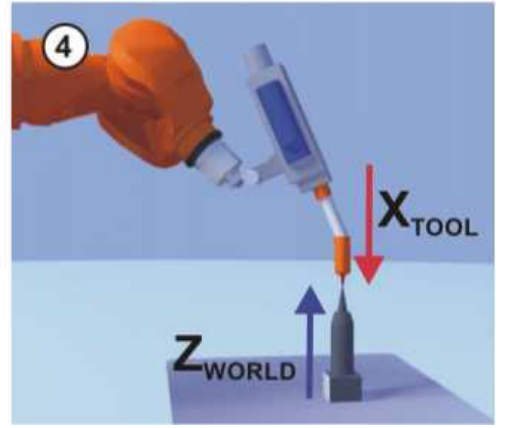
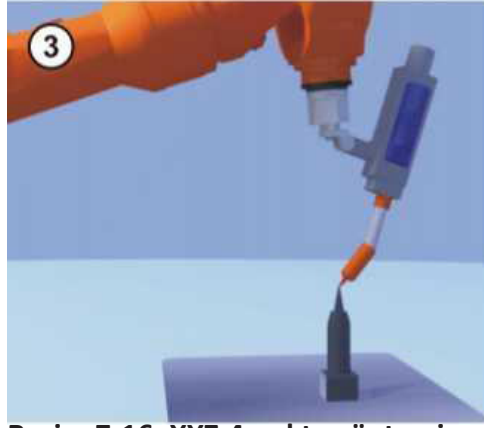
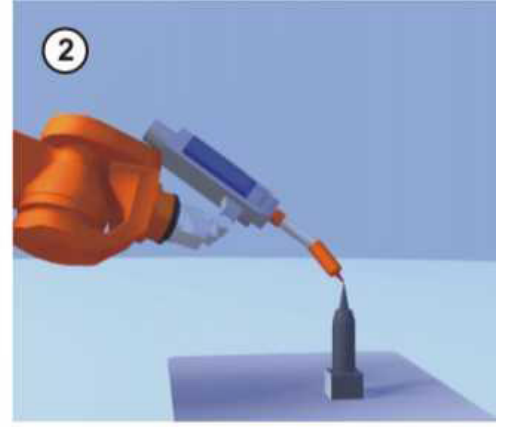
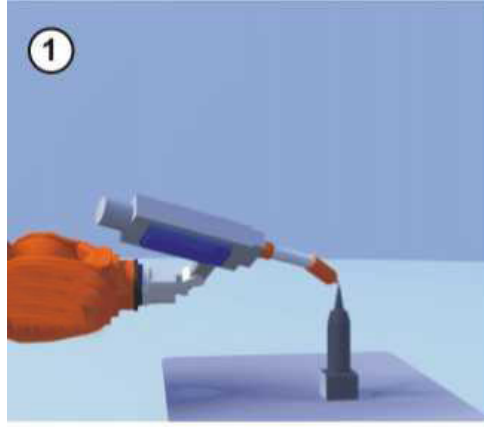
XYZ-4 nokta metodunda izlenecek yöntem:

1. **Çalıştırma > Ölçüm > Alet > XYZ 4 nokta** menü yolunu seçin. 2. Ölçülecek olan alet için bir numara ve bir isim verin. **Tamam** ile onaylayın

 1 ... 16 numaraları seçilebilir..

3. TCP ile bir referans noktasına gidin. **Tamam** ile onaylayın.

4. TCP ile referans noktasına başka bir yönden gidin. **Tamam** ile onaylayın.



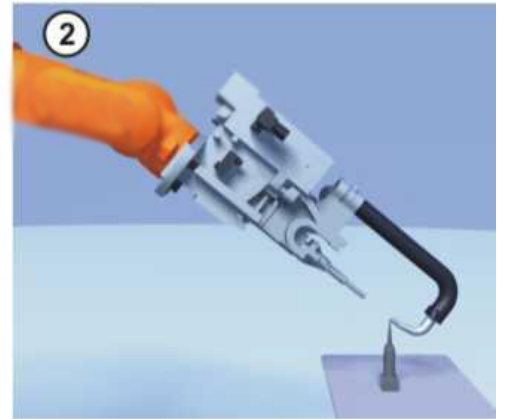
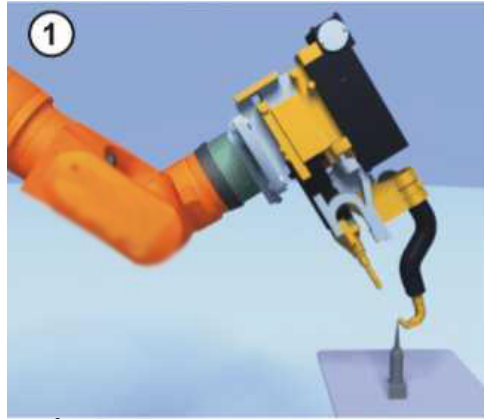
Resim 7-16: XYZ-4 nokta yöntemi

5. Adım 4'ü iki defa tekrarlayın.

6. **Kaydet**'e basın.

XYZ referans metoduyla TCP ölçümü

XYZ referans metodunda yeni bir alet, daha önce ölçülmüş olan bir aletle ölçülür. Robot kumandası, flanş pozisyonlarını karşılaştırır ve yeni aletin TCP'si- ni hesaplar.



Resim 7-17

İzlenecek yöntem

1. Bunun önkoşulu, daha önce ölçülmüş bir aletin montaj flanşına takılmış olması ve TCP verilerinin biliniyor olmasıdır.
2. Ana menüde **İşletime alma > Ölçme > Alet > XYZ referans** seçin.
3. Yeni alet için bir numara ve bir ad verin. **Devam** ile onaylayın.
4. Daha önce ölçülmüş olan aletin TCP verilerini girin. **Devam** ile onaylayın.
5. TCP ile bir referans noktasına gidin. **Ölçme** üzerine basın. **Devam** ile onaylayın.
6. Aleti serbestleştirin ve sökün. Yeni aleti monte edin.

7. Yeni aletin TCP'si ile referans noktasına gidin. **Ölçme** üzerine basın. **Devam** ile onaylayın.

8. **Kaydet** üzerine basın. Veriler kaydedilir ve pencere kapanır.

Veya **Yük verileri** üzerine basın. Veriler kaydedilir ve taşıma kapasitesi verilerinin girilebildiği bir pencere açılır.

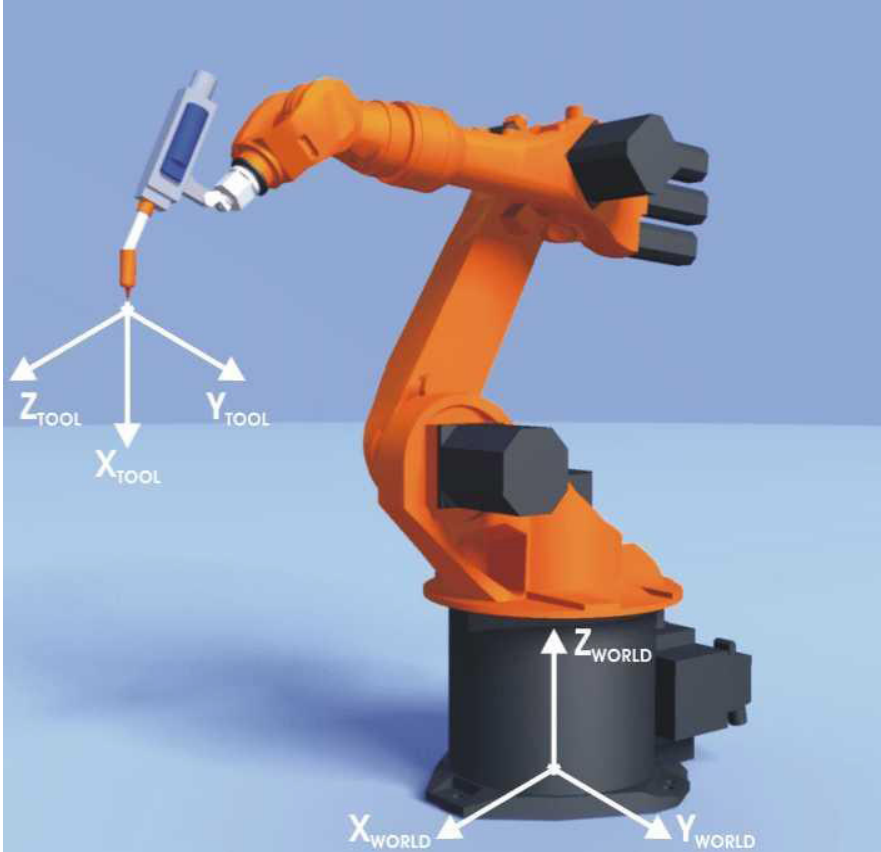
TOOL koordinat sisteminin aksları, WORLD koordinat sisteminin akslarına paralel olarak düzenlenir. Böylece TOOL koordinat sisteminin oryantasyonu robot kumanda ünitesine bildirilir.

ABC-World metoduyla oryantasyon ölçümü

Bu metotta 2 varyant vardır:

- **5D:** Robot kumandasına sadece aletin çarpma yönü bildirilir. Çarpma yönü, varsayılan olarak X eksenidir. Diğer aksların yönü, sistem tarafından belirlenir ve başka bir bilgi olmadan kullanıcı tarafından algılanamaz.
- Uygulama alanı: örn. MIG/MAG kaynağı, lazer veya su huzmesiyle kesim
- **6D:** Her 3 eksenin yönü robot kumandasına bildirilir.

Uygulama alanı: Örn. kaynak penseleri, kavrayıcılar veya yapıştırma memeleri için



Resim 7-18: ABC-World-Yöntemi

ABC-World-5D metodunda izlenecek yöntem

- Çalıştırma > Ölçüm > Alet > ABC World menü yolunu seçin.
- Aletin numarasını girin. **Tamam** ile onaylayın.
- 5D/6D** alanında o seçeneği belirleyin. **Tamam** ile onaylayın.
- 5D** seçildiyse:

+X_{TOOL} düzlemini -Z_{WORLD} düzlemine paralel olarak ayarlayın. (+X_{TOOL} = çarpma yönü)

e. **Tamam** ile onaylayın.

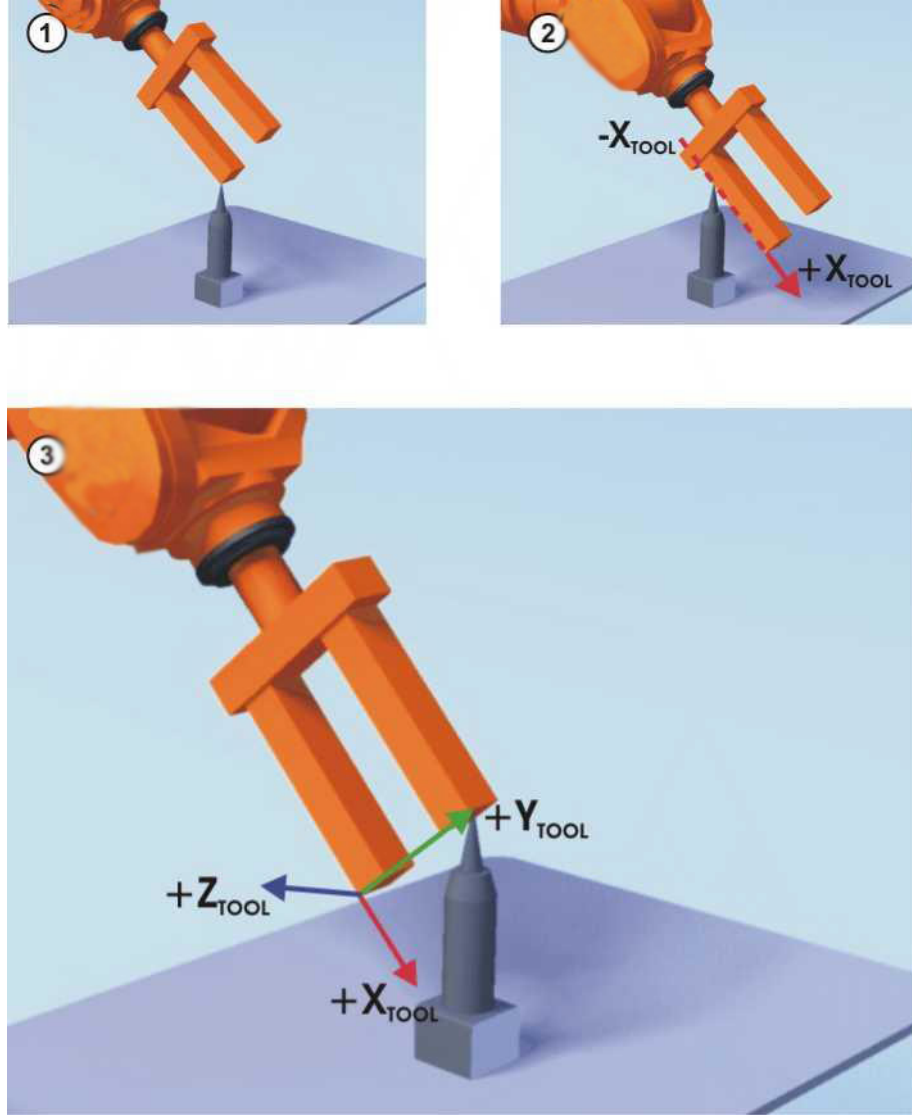
f. **Kaydet**'e basın.

ABC 2 nokta yöntemi yönelim ölçümü

X ekseninde bir noktaya ve XY ekseninde bir noktaya gidilmesi sayesinde, robot kontrol ünitesine TOOL koordinat sisteminin eksenleri bildirilir.

Bu metot, eksen yönleri özellikle kesin tespit edilmesi gerekiyorsa kullanılır.

i Aşağıdaki yöntem, alet çarpma yönü Default çarpma yönü olduğunda (= X yönü) geçerlidir. Y veya Z üzerinde çarpma yönü değiştirildiğinde izlenecek yöntem de buna uygun olarak değiştirilmelidir



Resim 7-19: ABC 2 nokta yöntemi

1. Bunun önkoşulu, TCP'nin daha önce bir XYZ yöntemiyle ölçümlenmiş olmasıdır.
2. Ana menüde **Devreye alma > Ölçümleme > Alet > ABC 2 nokta** seçin.
3. Monte edilen aletin numarasını girin. **İleri** ile onaylayın.
4. TCP ile herhangi bir referans noktasına gidin. **Ölç** düğmesine basın. **İleri** ile onaylayın.
5. X eksenindeki referans noktası, negatif X değerine (yani etkiye yönünün tersine) sahip bir nokta üzerinde duracak şekilde aleti sürün. **Ölç** düğmesine basın. **İleri** ile onaylayın.
6. Referans noktası XY düzlemi üzerinde pozitif Y değerine sahip bir nokta üzerinde durana kadar aleti hareket ettirin. **Ölç** düğmesine basın. **İleri** ile onaylayın.
7. Ya **Kaydet** düğmesine basın. Veriler kaydedilir ve pencere kapanır.

Ya da **Yük verileri** düğmesine basın. Veriler kaydedilir ve yük verilerinin girilebildiği bir pencere açılır.

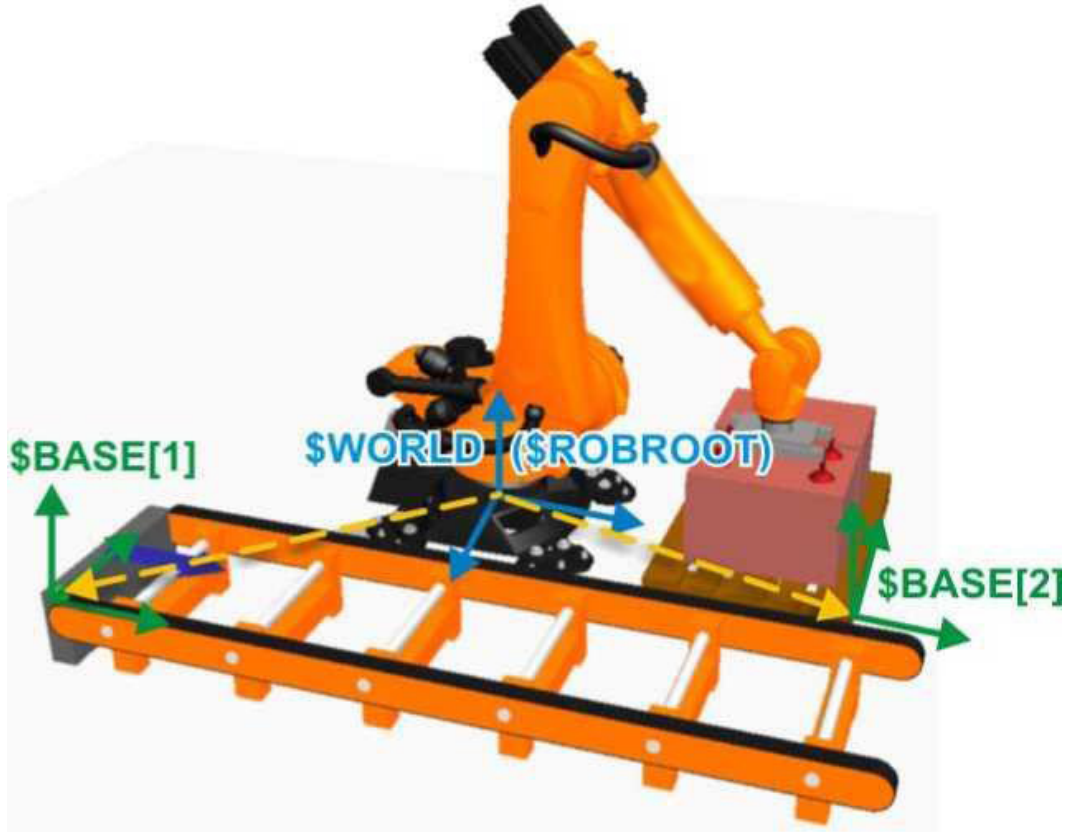
7.8 Bazı (Base) Ölçümleme

Açıklama

Bir base ölçümünün yapılması, dünya koordinat sisteminden yola çıkarak robotun çevresinde belirli bir noktada bir koordinat sistemi oluşturulması anlamına gelir. Buradaki hedef, hareketlerin ve programlanmış pozisyonların bu koordinat sistemine göre alınmasıdır. Bu nedenle bir base koordinat sistemi için, örn. işlenen parça braketlerinin, bölmelerin, paletlerin veya makinelerin tanımlı kenarları gibi referans noktaları seçilmesi mantıklıdır.

Base ölçümü iki adımda yapılmaktadır:

1. Koordinat başlangıç noktası tespiti
2. Koordinat yönlerini tanımlama

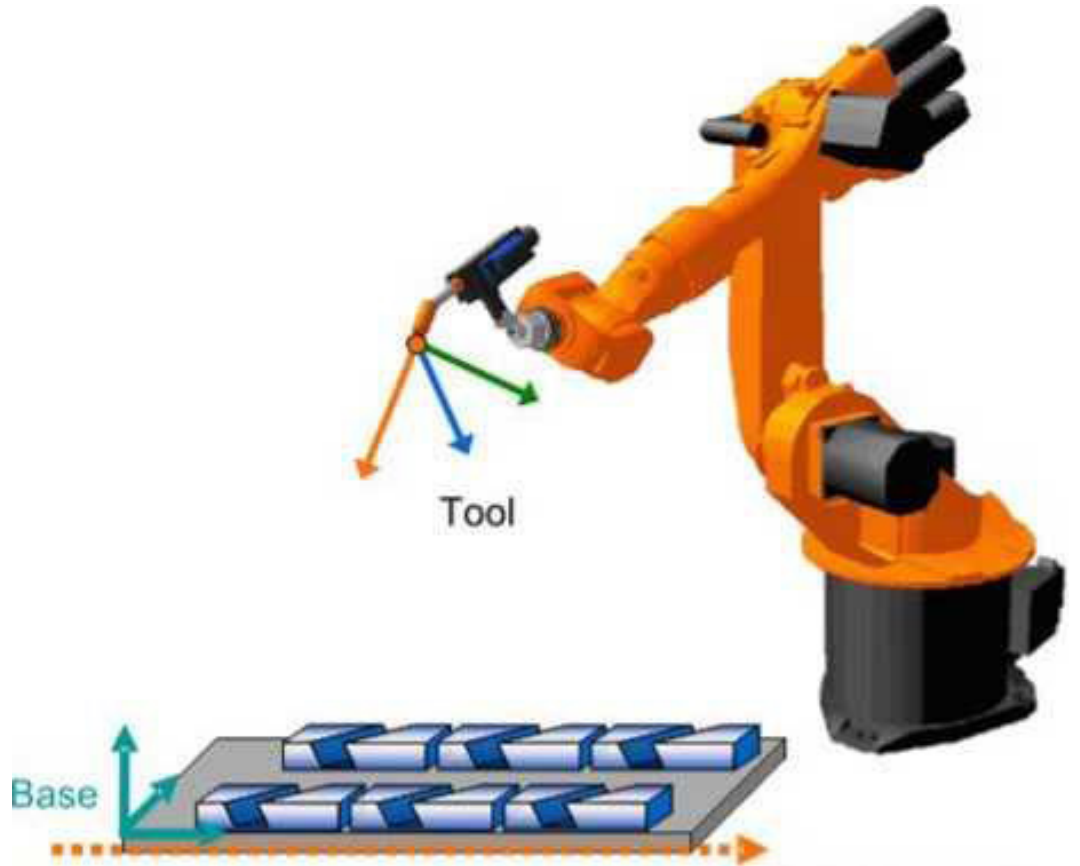


Resim 7-20: Base ölçümleme

Avantajlar

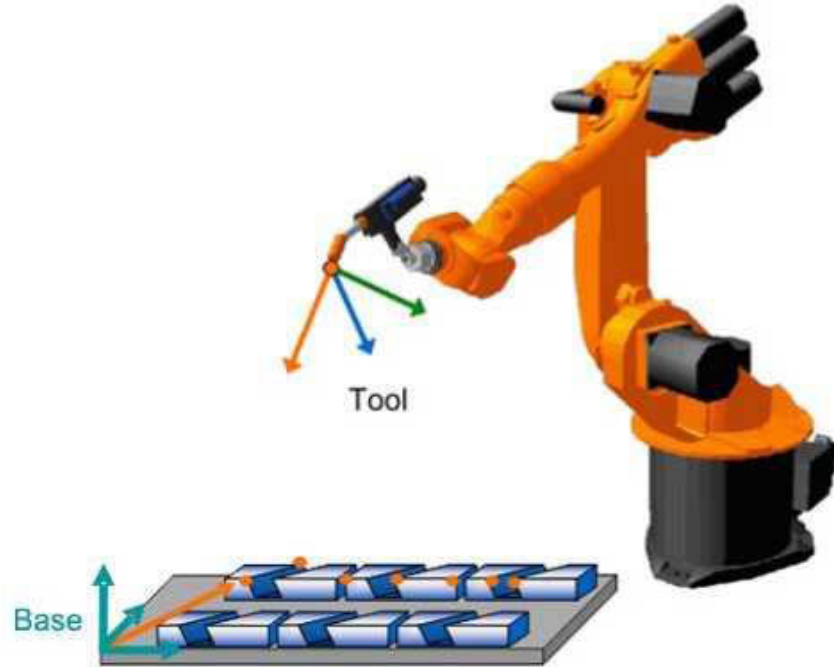
Bir base ölçümü gerçekleştirildikten sonra aşağıdaki avantajlar ortaya çıkar:

- İşlenen parçanın kenarları boyunca hareket etme:
TCP, çalışma alanının veya işlenen parçanın kenarları boyunca manuel olarak hareket ettirilebilir



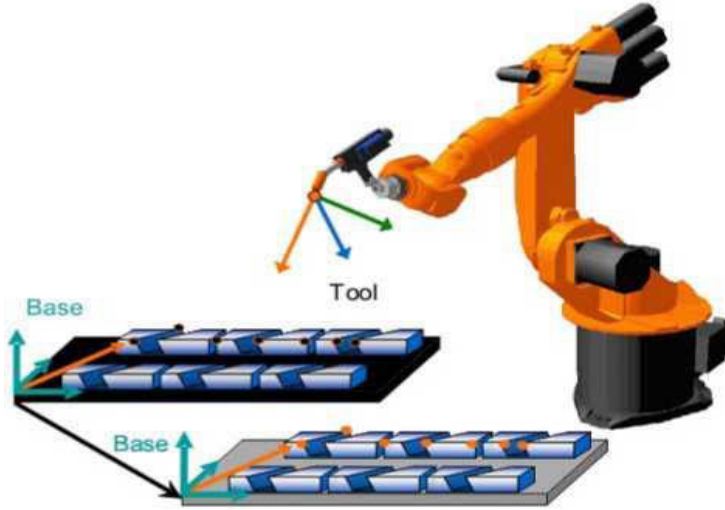
Resim 7-21: Baz ölçümünün avantajları: Hareket yönü

- Referans koordinat sistemi:
Tanıtılan (teach) noktalar, seçilen koordinat sistemine göre dir.



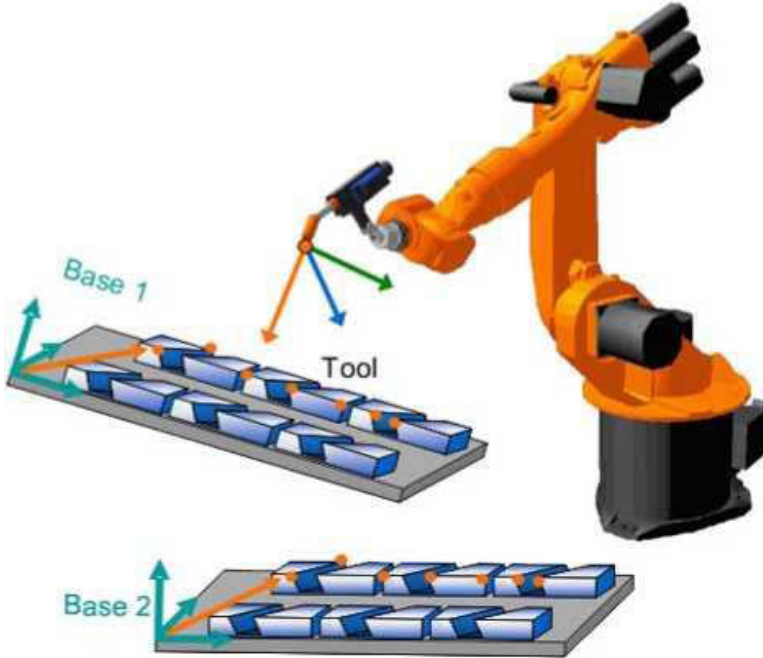
Resim 7-22: Baz ölçümünün avantajları: İstenilen koordinat sistemi referans alınabilir

- Koordinat sisteminin düzeltilmesi / kaydırılması:
Noktalar, base referans alınarak tanıtılabilir (teach). Base kaydırılması gerektiğinde, örn. çalışma alanı kaydırıldığından dolayı, noktalar da birlikte kayar ve yeniden tanıtılmaları (teach) gerekmez.



Resim 7-23: Baz ölçümünün avantajları: Baz koordinat sistemini kaydırma

- Birden fazla base koordinat sistemi kullanma:
Maksimum 32 farklı koordinat sistemi oluşturulabilir ve program adımına bağlı olarak kullanılabilir.



Resim 7-24: Baz ölçümünün avantajları: Birden fazla baz koordinat sistemi kullanma

Base ölçme olanakları

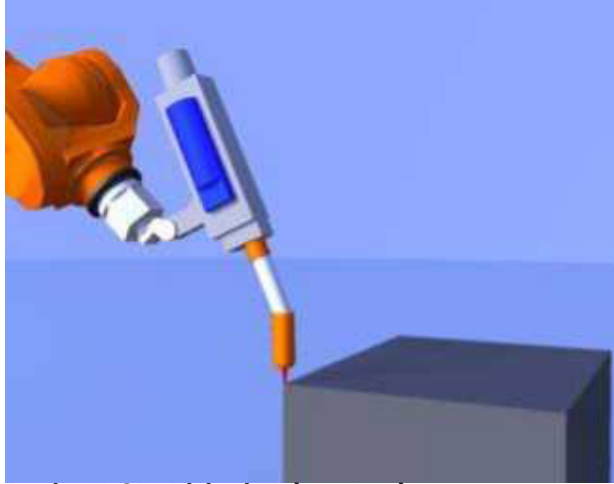
Base ölçümü için aşağıdaki metotlar kullanılabilir:

Metotlar	Açıklama
3 nokta metodu	<ol style="list-style-type: none"> 1. Başlangıç noktasını (merkezin) tanımlama 2. Pozitif X yönünü tanımlama 3. Pozitif Y yönünü (XY düzlemi) tanımlama
Endirekt metot	Endirekt metot, base'in başlangıcına gidilemediğinde kullanılır, örn. bu noktanın, işlenen parçanın iç kısmında veya robotun çalışma alanının dışında bulunması halinde. Koordinatlarının biliniyor olması gereken (CAD verisi) base'in 4 noktasına gidilir. Robot kumandası, base'i bu noktalara göre hesaplar.
Nümerik giriş	Dünya koordinat sistemine (X, Y, Z) olan mesafe ve dönüş (A, B, C) için olan değerlerin doğrudan girilmesi.

3 nokta yönteminde izlenecek yöntem

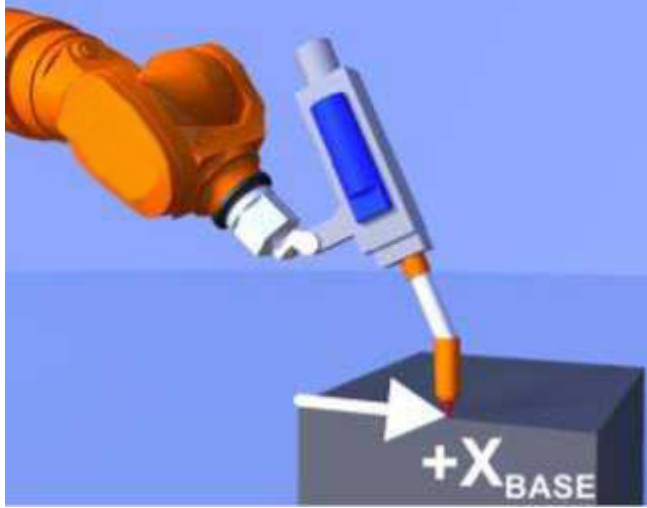
i Base ölçümü sadece daha önce ölçümü yapılmış bir aletle gerçekleştirilebilir (TCP biliniyor olmalıdır).

1. Ana menüde **İşletime alma > Ölçümleme > Base > 3 nokta** seçin.
2. Base için bir numara ve bir isim verin. **Devam** ile onaylayın.
3. Base sisteminin ölçümü için kullanılacak TCP'ye sahip aletin numarasını girin. **Devam** ile onaylayın.
4. TCP ile, yeni Base'in başlangıcına gidin. **Ölç** düğmesine basın ve pozisyonu **Evet** düğmesiyle onaylayın.



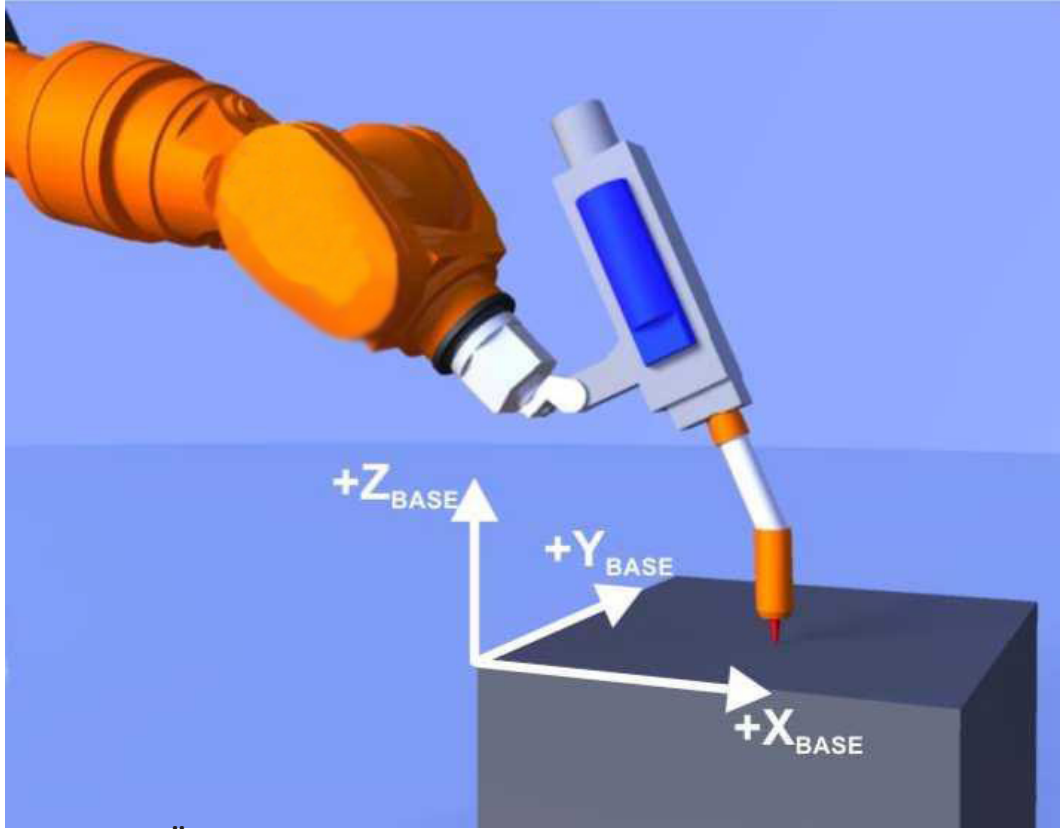
Resim 7-25: Birinci nokta: Başlangıç

5. TCP ile yeni bazın pozitif X eksenindeki bir noktaya gidin. **Ölç** üzerine basın ve pozisyonu **Evet** düğmesiyle onaylayın.



Resim 7-26: İkinci nokta: X yönü

6. TCP ile XY düzlemi üzerinde pozitif değere sahip bir noktaya gidin. **Ölç** üzerine basın ve pozisyonu **Evet** düğmesiyle onaylayın



Resim 7-27: Üçüncü nokta: XY düzlemi

7. Kaydet üzerine basın.

8. Menüü kapatın

i Üç ölçümleme noktası bir doğru üzerinde olmamalıdır. Noktalar arasında asgari bir açı olmalıdır (standart ayar 2,5°).

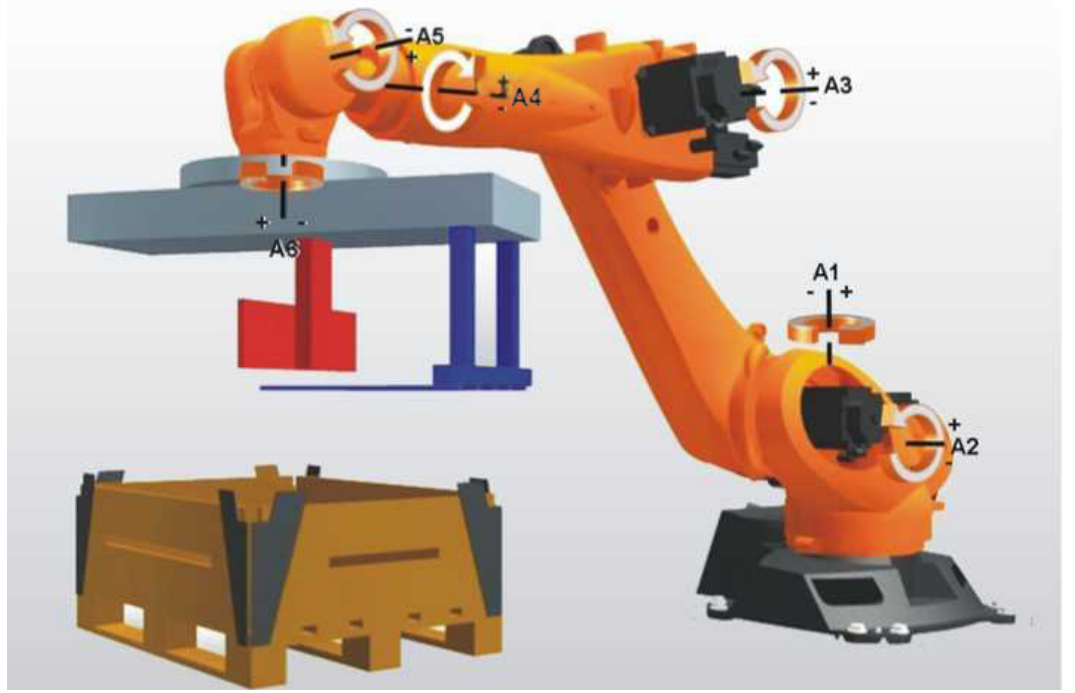
7.9 Güncel Robot Pozisyonunun Sorgulama

Robot pozisyonları görüntüleme olanakları

Güncel robot pozisyonu iki farklı şekilde görüntülenebilir:

- Eksene özel:

$\$AXIS_ACT=\{A1...,A2...,A3...,A4...,A5...,A6...,E1...,...,E6...\}$

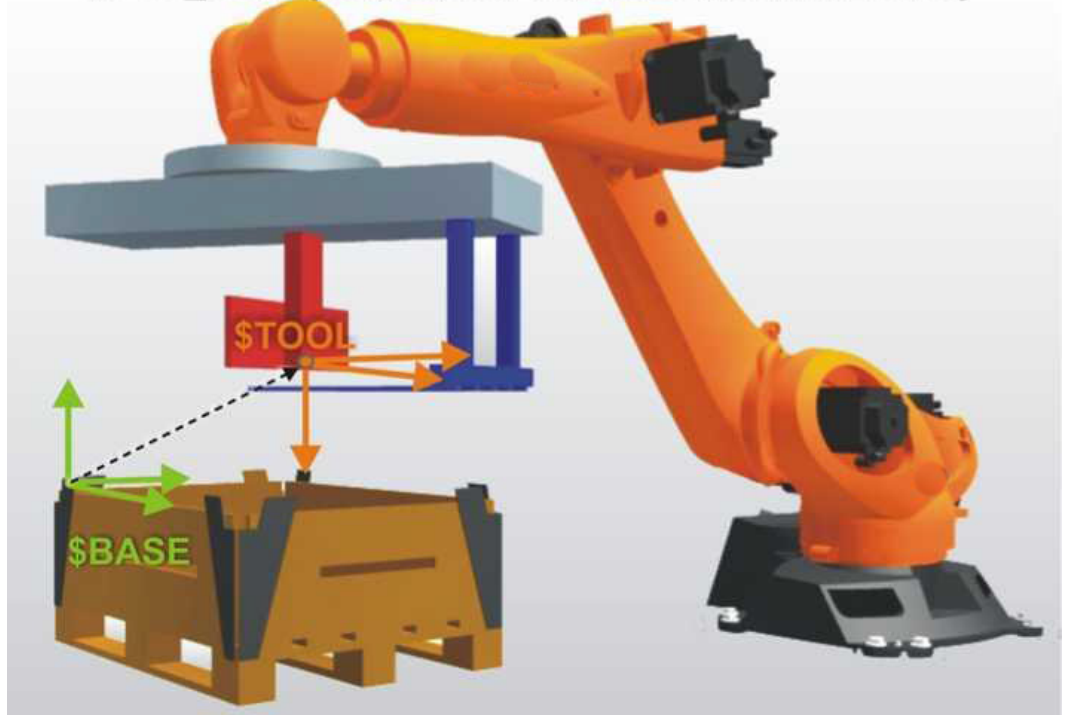


Resim 7-28: Eksene özel robot pozisyonu

Her eksen için güncel eksen açısı görüntülenir: Bu durum, ayar pozisyonundan yola çıkılarak mutlak açı değerine karşılık gelir.

- Kartezyen

$\$POS_ACT=\{X...,Y...,Z...,A...,B...,C...,S...,T...,E1...,...\}$



Resim 7-29: Kartezyen pozisyon

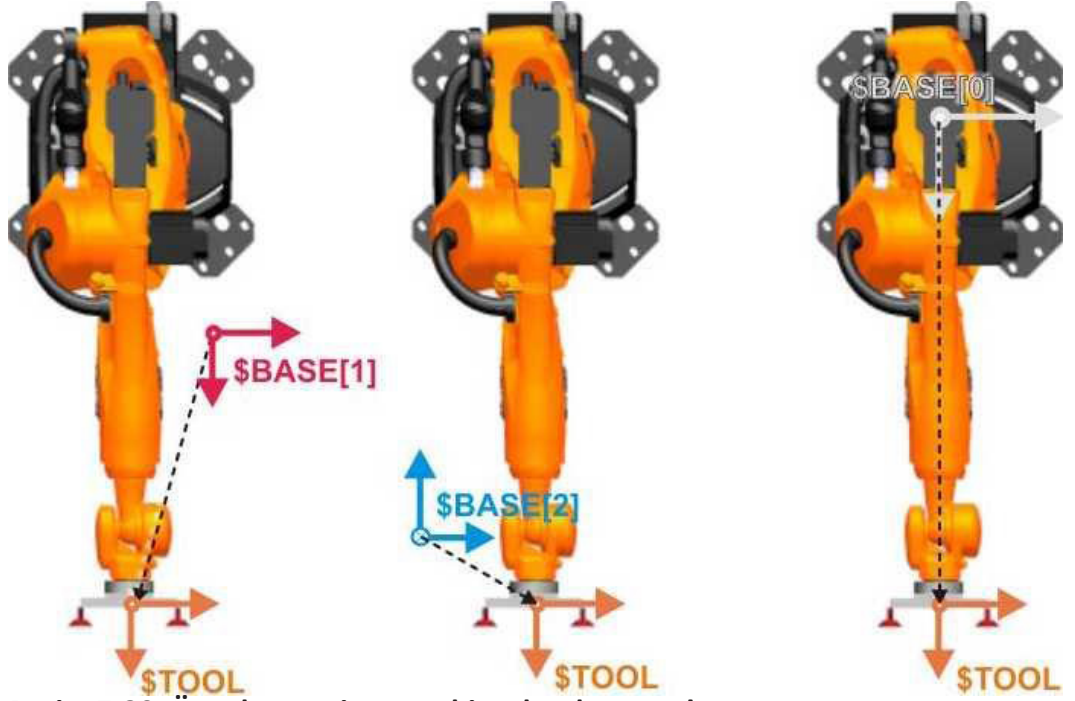
Güncel TCP'nin (Tool koordinat sistemi) güncel olarak seçilmiş temel koordinat sistemine göre olan güncel pozisyonu görüntülenir.

Tool koordinat sistemi seçilmiş değilse, flanş koordinat sistemi geçerlidir!

Temel koordinat sistemi seçilmiş değilse, dünya koordinat sistemi geçerlidir!

Farklı base koordinat sistemleriyle kartezyen konum

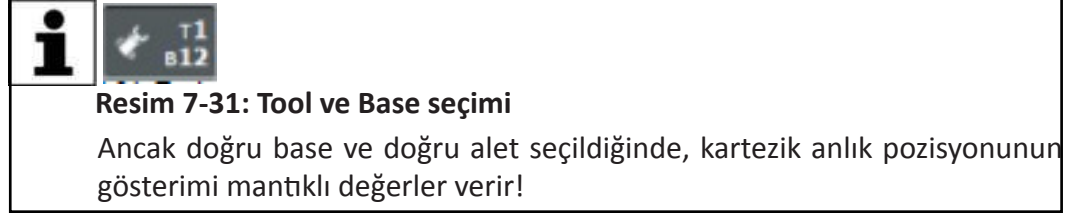
Aşağıdaki resme bakıldığında, robotun üç kez aynı konumu aldığı hemen göze çarpar. Pozisyon gösterimi, üç durumun üçünde de farklı değerler verir:



Resim 7-30: Üç robot pozisyonu - bir robot konumu!

Tool koordinat sisteminin/TCP'nin konumu ilgili base koordinat sisteminde görüntülenir:

- Base 1 için
- Base 2 için
- Base 0 için: Bu, robot ayağı koordinat sistemine karşılık gelmektedir (çoğu durumda dünya koordinat sistemine de)!



Robot pozisyonunu sorgulama

İzlenecek yöntem:

- **Gösterge > Gerçek pozisyon** menülerini takip edin. Kartezyen gerçek pozisyon görüntülenir.
- Eksene özel gerçek pozisyonu görüntülemek için, **Eksene özel** düğmesine basın
- Yeniden kartezyen gerçek pozisyonu görüntülemek için, **Kartezyen** düğmesine basın

8 Robot Programları Yürütme

8.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Başlangıç sürüşü gerçekleştirme
- Robot programlarını seçme ve başlatma

8.2 İklendirme Sürüşü Yapma

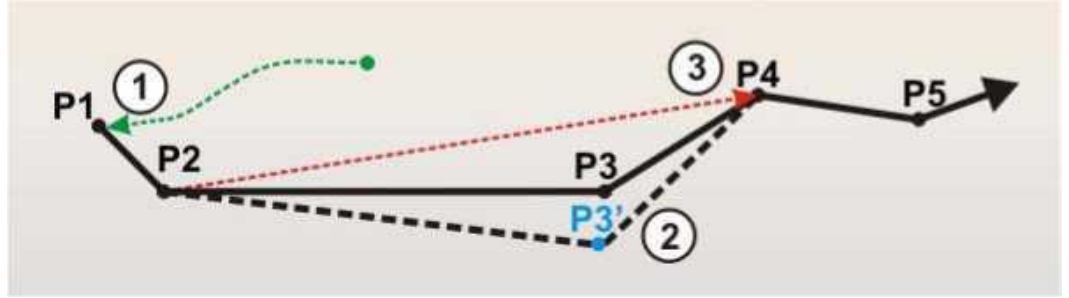
SAK Sürüşü

Bir robotunun başlangıç sürüşüne SAK sürüşü denmektedir.

i SAK, Satzkoinzidenz (cümle uyumu) anlamına gelir. Uyum burada "örtüşme" ve de "zamansal/mekansal olayların buluşması" anlamına gelir.

SAK sürüşü aşağıdaki durumlarda yürütülür:

- Program seçimi
- Program reseti (geri alma)
- Programın işletimi sırasında elle hareket yöntemi
- Programda değişiklik
- Cümle seçimi



Resim 8-1: Bir SAK sürüşü için örnek nedenler

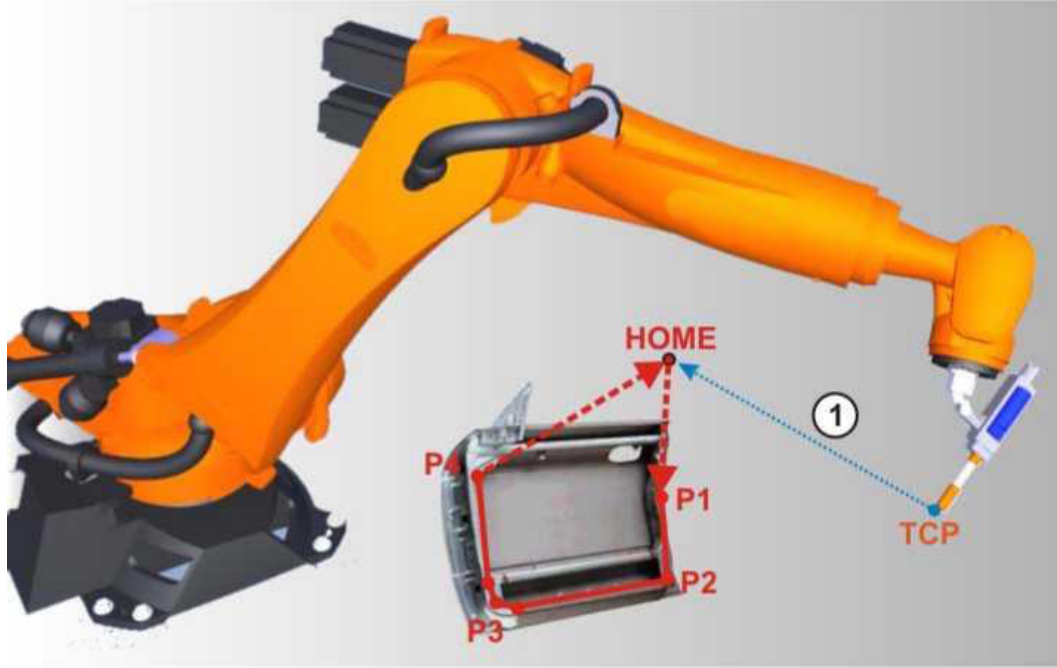
Bir SAK sürüşünün yürütülmesine ilişkin örnekler:

1. Program seçimi ya da reset (sıfırlama) sonrası Home konumuna SAK sürüşü
2. Bir hareket komutu değiştirildikten sonra SAK sürüşü: Nokta silme, tanıma (teach) vb.
3. Bir cümle seçiminden sonra SAK sürüşü

SAK sürüşü için nedenler

Bir SAK sürüşü, güncel robot konumunun robot programındaki güncel noktanın koordinatlarıyla aynı olmasını sağlamak için zorunludur.

Güncel robot pozisyonu programlanmış pozisyonla aynı olduktan sonra yörünge planlamasına geçilebilir. Dolayısıyla daima ilk olarak TCP'nin yörünge üzerine getirilmesi gerekir



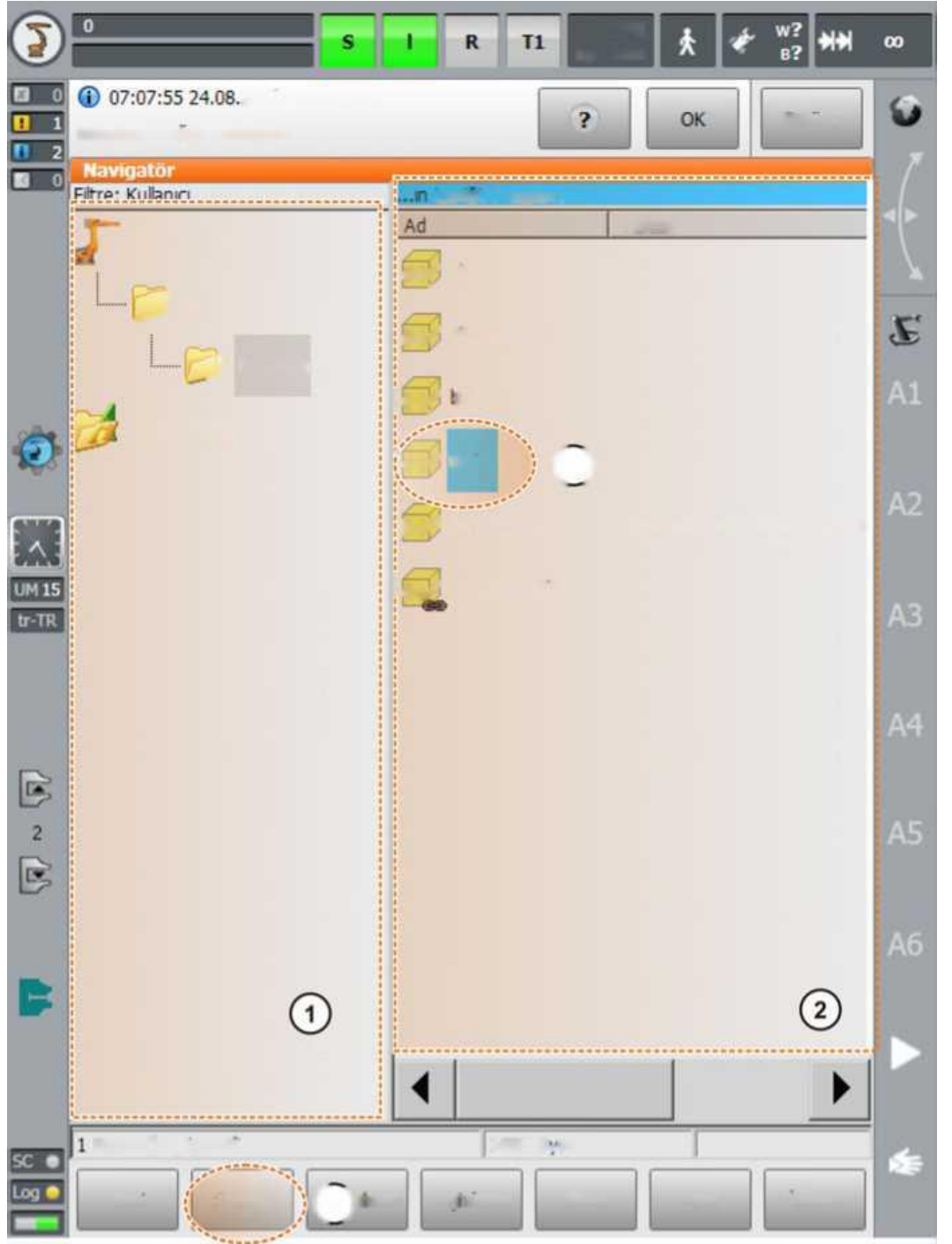
Resim 8-2: SAK sürüşü için örnek

1 Program seçiminden ya da program resetinden sonra Home pozisyonuna SAK sürüşü

8.3 Robot Programlarını Seçme ve Başlatma

Robot programlarını seçme ve başlatma

Bir robot programı yürütülecekse seçilmiş olması gerekir. Robot programları, kullanıcı arayüzünde Navigatörde bulunmaktadır. Genelde sürüş programlar klasörlerde kayıtlıdır. Cell programı (robotun, bir PLC tarafından kumanda edilmesi için yönetim programı) daima "R1" klasöründe yer alır.



Resim 8-3: Navigator

1. Navigatör: Dizin / disk sürücü yapısı
2. Navigatör: Dizin / veri listesi
3. Seçilen program
4. Bir programın seçimi için buton

Bir programın başlatılması için, Start ileri
tuşları kullanılabilir.



hem de Start geri





Resim 8-4: Program akış yönleri: İleri/geri

Bir program yürütüldüğünde, robotun program kontrollü hareketi için birden çok **program akışı türü** kullanılabilir:

	GO <ul style="list-style-type: none">■ Program durmadan program sonuna kadar çalışır.■ Test işletiminde start tuşunun basılı tutulması gerekir.
	Hareket <ul style="list-style-type: none">■ Motion-Step akış türünde, her hareket komutu tek tek yürütülür.■ Bir hareketin tamamlanmasından sonra, yeniden "Start" tuşuna basılması gerekir.
	Tekli adım Sadece "Uzman" kullanıcı grubunda kullanılabilir! <ul style="list-style-type: none">■ Incremental-Step uygulamasında satır satır yürütülür (satırın içeriğinden bağımsız olarak).■ Her satırdan sonra yeniden Start tuşuna basılması gerekir






Bir robot programının görünümü nasıldır?

```
1 DEF kuka_rocks( )
2 INI
3 PTP HOME Vel= 100 % DEFAULT
4 PTP P1 Vel=100 % PDAT1 Tool[1] Base[0]
5 PTP P2 Vel=100 % PDAT2 Tool[1] Base[0]
6 PTP P3 Vel=100 % PDAT3 Tool[1] Base[0]
7 OUT 1'' State=TRUE CONT
8 LIN P4 Vel=2 m/s CPDAT1 Tool[1] Base[0]
9 PTP HOME Vel= 100 % DEFAULT
10 END
```

Resim 8-5: Bir robot programının yapısı

1	<p>Sadece uzman kullanıcı grubunda görünür:</p> <ul style="list-style-type: none">■ "DEF Programadı()" daima programın başında yer alır■ "END" talimatı bir programın sonunu tanımlar
2	<ul style="list-style-type: none">■ "INI" satırında, programının doğru yürütülmesi için zorunlu olan standart parametrelere olan çağrılar yer alır.■ "INI" satırı daima ilk olarak yürütülmek zorundadır!
3	<ul style="list-style-type: none">■ Hareket komutları, bekleme/mantık komutlarının vs. yer aldığı asıl program metni■ "PTP Home" sürüş komutu sıklıkla bir programın başında ve sonunda kullanılır, çünkü kesin ve bilinen bir pozisyonu belirtir.

Program durumu

Simge	Renk	Tarif
	Gri	Hiçbir program seçili değil.
	Sarı	Komut imleci, seçili olan programın ilk satırında durmakta.
	Yeşil	Program seçildi ve çalışıyor.
	Kırmızı	Seçilmiş ve başlatılmış program durduruldu.
	Siyah	Komut imleci, seçilen programın sonundadır.

Program start
yürütme

Robot programlarının başlatılmasında izlenecek yöntem:

1. Programı seçme



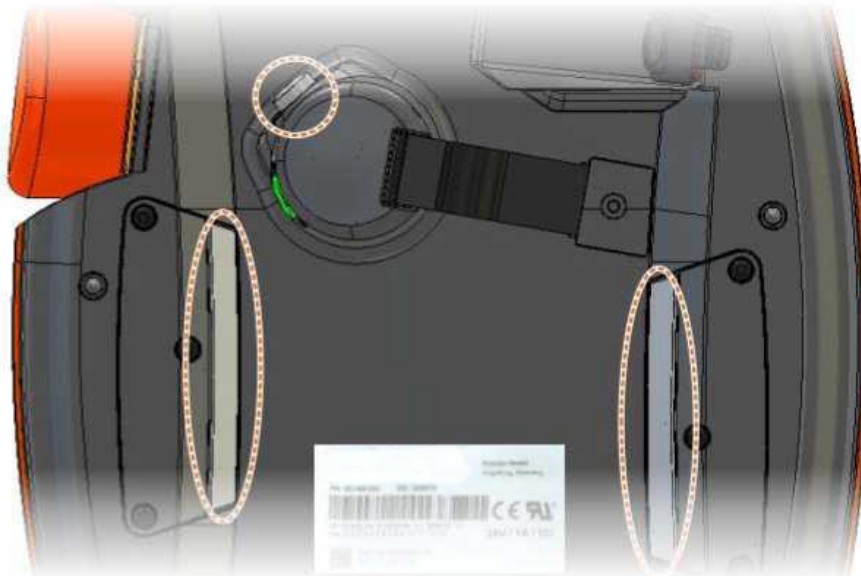
Resim 8-6: Program seçimi

2. Program hızını ayarlama (Program-Override, POV)



Resim 8-7: POV ayarı

3. Onay tuşuna basma



Resim 8-8: Onay şalteri

4. Başlat düğmesine (+) basın ve basılı tutun:

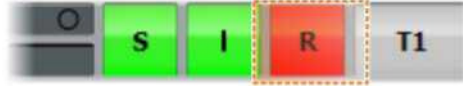
- "INI" satırı yürütülür.
- Robot, SAK sürüşünü yapar.



Resim 8-9: Program akış yönleri: İleri/Geri

UYARI Bir SAK sürüşü, seçilen hareket komutları seti PTP sürüş komutunu içerdiğinde, gerçek pozisyondan hedef pozisyona PTP hareketi olarak yürütülür. Seçilen hareket komutları seti LIN ya da CIRC komutlarını içerdiğinde SAK sürüşü LIN hareketi olarak yürütülür. Çarpışmaları önlemek için hareketi gözlemlemek gerekir. SAK sürüşünde hız otomatik olarak düşürülmüştür.

5. Hedef pozisyona varıldıktan sonra hareket durdurulur.



"SAK'ya varıldı" bilgi iletisi görüntülenir.

6. Diğer süreç adımları (ayarlıolan işletim türüne bağlı olarak):

- **T1 ve T2:** Programı başlat tuşuna basarak devam ettirin.

- **AUT:** Tahriklerin aktifleřtirilmesi.



Peřinden programı impuls ile Bařlat düęmesiyle bařlatın.

- Hücre programında iřletim türünü **EXT** olarak deęiřtirip sürüş komutunu PLC'den aktarın.

9 Program Dosyalarının Kullanımı

9.1 Genel Bakış

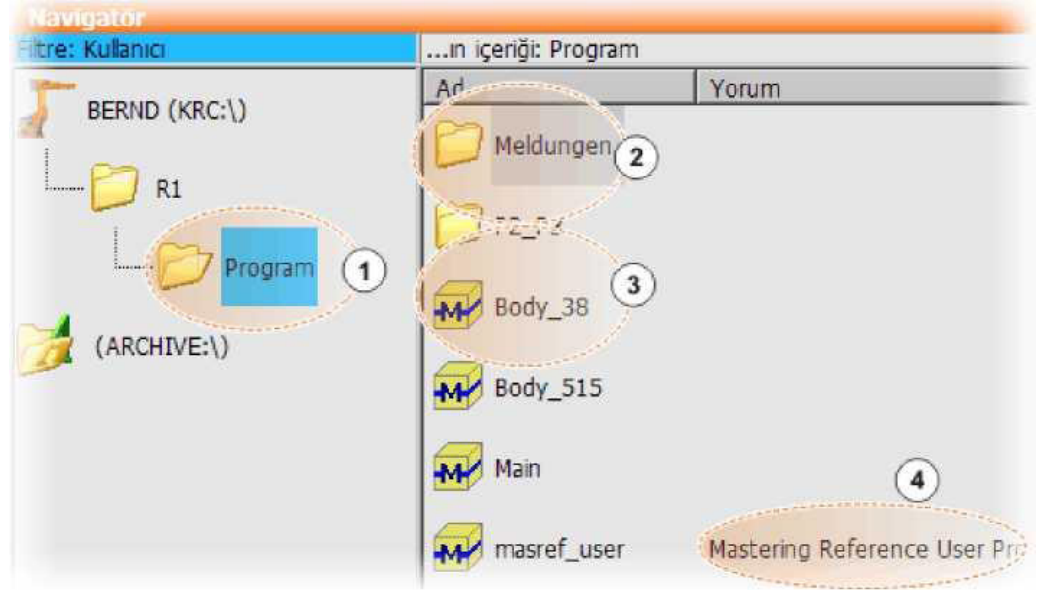
Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Program modülleri oluşturma
- Robot programlarının düzenlenmesi

9.2 Program Modülleri Oluşturma

Navigator'daki program modülleri

Program modülleri daima "Program" klasöründe kaydedilmelidir. Yeni klasörler oluşturup program modüllerini orada kaydetme olanağı da var. Modüller "M" simgesiyle işaretlidir. Bir modüle yorum eklenebilir. Bu tür bir yorum örneğinin programa ilişkin kısa bir fonksiyon açıklaması içerebilir.

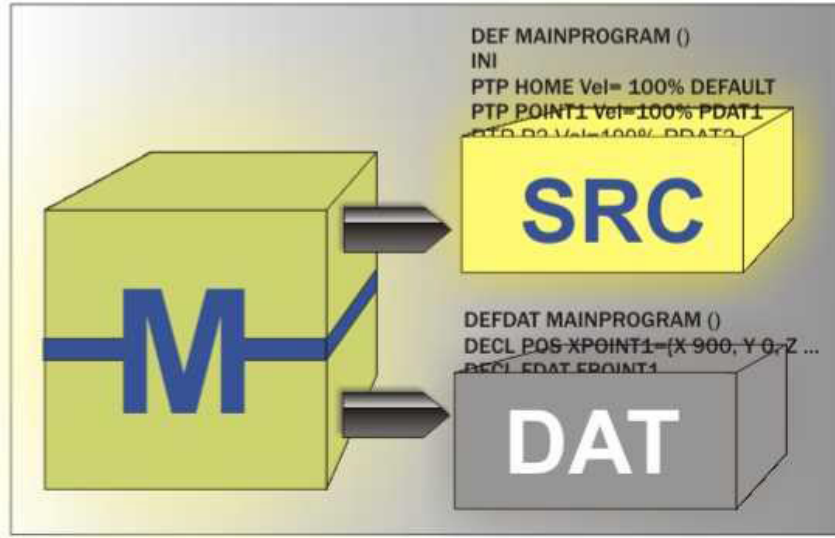


Resim 9-1: Navigator'daki modüller

- 1 Programlar için ana klasör "Program"
- 2 Diğer programlar için alt klasör
- 3 Program modülü / modül
- 4 Bir program modülünün yorumu

Program modüllerinin özellikleri

Bir modül daima iki bölümden oluşur:



Resim 9-2: Modül yapısı

- **Kaynak kodu:** SRC dosyasında program kodu yer alır.

```
DEF MAINPROGRAM ()
INI
PTP HOME Vel= 100% DEFAULT
PTP POINT1 Vel=100% PDAT1 TOOL[1] BASE[2]
PTP P2 Vel=100% PDAT2 TOOL[1] BASE[2]
...
END
```

Veri listesi: DAT dosyasında kalıcı veriler ve nokta koordinatları yer alır.

```
DEFDAT MAINPROGRAM ()
DECL E6POS XPOINT1={X 900, Y 0, Z 800, A 0, B 0, C 0, S 6, T 27, E1,
0, E2 0, E3 0, E4 0, E5 0, E6 0}
DECL FDAT FPOINT1 ...
...
ENDDAT
```

Program modülleri oluşturulmada izlenecek yöntem

1. Yeni programın, dizin yapısı içinde, altında oluşturulmasını istediğiniz klasörü işaretleyin, örn. **Program** klasörünü.
2. **Yeni** tuşuna basın.
3. Program için bir ad girin ve de gerekirse bir açıklama girin ve **OK** ile onaylayın

9.3 Program Modülleri İşleme

İşleme olanakları

Bilinen dosya sistemlerinde olduğu gibi program modülleri, smartPad navigatöründe de işlenebilir. İşlemeye şunlar dahildir:

- Çoğaltma/kopyalama
- Silme
- Yeniden adlandırma

Program sil için izlenecek yöntem

1. Dizin ağacında, dosyanın bulunduğu klasörü işaretleyin.
2. Dosya listesinde dosyayı işaretleyin.
3. **Sil** > yazılım tuşunu seçin.
4. Güvenlik sorusunu **Evet** ile onaylayın. Modül silinir.



“Uzman” kullanıcı grubunda ve “Ayrıntılar” filtre ayarında her modül için navigatörde iki dosya vardır (SRC ve DAT dosyası). Bu durum sözkonusu olduğunda her iki dosyanın silinmesi gerekir! Silinen dosyalar yeniden geri yüklenemez!

**Program adını
değiştirmede
izlenecek yöntem**

1. Dizin ağacında, dosyanın bulunduğu klasörü işaretleyin.
2. Dosya listesinde dosyayı işaretleyin.
3. **Düzenle > Yeniden adlandır** seçin.
4. Dosya adının yerine yeni dosya adını yazın ve **OK** ile onaylayın.



“Uzman” kullanıcı grubunda ve “Ayrıntılar” filtre ayarında her modül için navigatörde iki dosya vardır (SRC ve DAT dosyası). Bu durum sözkonusu olduğunda her iki dosyanın yeniden adlandırılması gerekir!

**Program çoğalt-
ma izlenecek
yöntem**

1. Dizin ağacında, dosyanın bulunduğu klasörü işaretleyin.
2. Dosya listesinde dosyayı işaretleyin.
3. **Çoğalt** tuşunu seçin.
4. Yeni modüle yeni bir dosya adı verin ve **Tamam** ile onaylayın.



“Uzman” kullanıcı grubunda ve “Ayrıntılar” filtre ayarında her modül için navigatörde iki dosya vardır (SRC ve DAT dosyası). Bu durum sözkonusu olduğunda her iki dosyanın çoğaltılması gerekir!

10 Programlanmış Hareketleri Oluşturma ve Değişirme

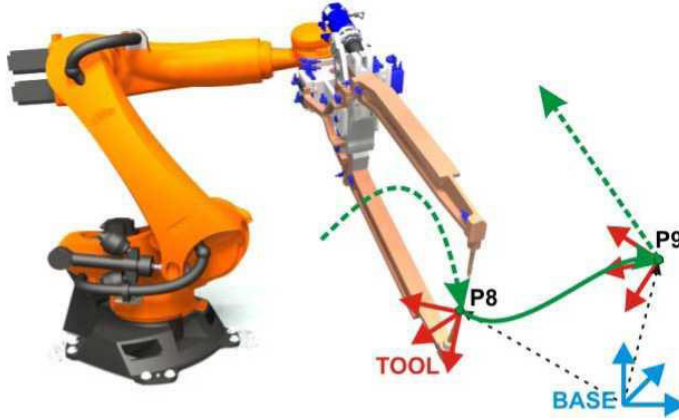
10.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Çevrim süresi en uygun hareket oluşturma
- Rota hareketi oluşturma
- Hareket komutlarını değiştirme

10.2 Yeni Hareket Komutları Oluşturma

Robot hareketlerini programlama

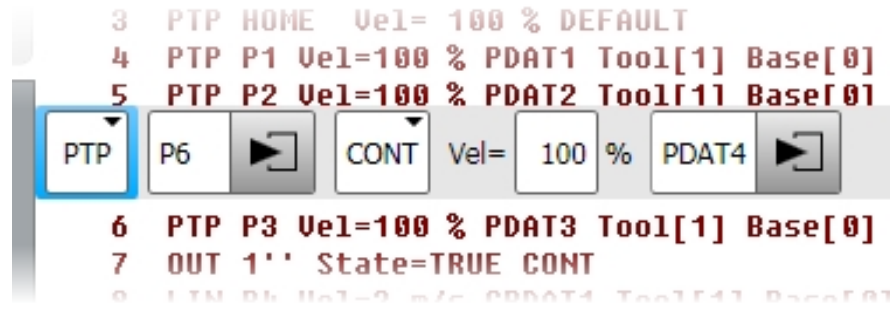


Resim 10-1: Robot hareketi

Robot hareketlerini programlamak gerektiğinde, çok sayıda soru ortaya çıkar:

Soru	Çözüm	Anahtar kelime
Robot, konumlarını ne şekilde hatırlar?	Aletin alandaki ilgili konumu kaydedilir. (Ayarlı olan tool ve base'e göre robot konumu)	POS E6POS
Robot nasıl hareket etmesi gerektiğini nereden bilir?	Hareket türü bilgisi sayesinde bilir. Noktadan noktaya, lineer ya da dairesel.	PTP LIN CIRC
Hareketleri sırasında robotun hızı nedir?	İki nokta arasındaki hız bilgisi ve ivme bilgisi programlama sırasında verilir.	Vel Acc..
Robot her noktada durmak zorunda mıdır?	Çevrim süresinden tasarruf etmek için noktaları atlamak (yaklaşarak geçmek) mümkündür. Bu durumda bir tam doğru duruş sağlanmaz.	CONT
Bir noktaya ulaşıldığında alet hangi oryantasyonu/yönlendirmeyi alır?	Yönlendirme kılavuzu, her hareket için ayrı ayrı ayarlanabilir.	ORI_TYPE
Robot bir engeli algılayabilir mi?	Hayır, robot "inatla" programlanmış olan rota sını takip eder. Çarpışmasız hareket programcının sorumluluğundadır. Fakat makineyi korumak için "Çarpışma denetimi" olanağı vardır	Çarpışma denetimi

Tanıtma (Teach-In) yönteminde robot hareketleri programlanırken bu bilgilerin aktarılması gerekir. Bu amaçla, bilgilerin rahatça girilebildiği inline formları kullanılır.



Resim 10-2: Hareket programlaması için inline formu

Hareket komutlarının programlanması için çeşitli hareket türleri kullanılabilir. Robotun çalışma sürecindeki taleplere göre hareketler programlanabilir.

- Aksa özgü hareketler (PTP: Point to Point)
- Rota hareketleri: LIN (lineer) ve CIRC (dairesel)
- SPLINE: Spline, özellikle karmaşık kavisli rotalar için uygun olan bir hareket türüdür. Bu tür rotalar genel olarak LIN ve CIRC hareketleri ile de oluşturulabilir ancak Spline'in avantajları vardır.

i Spline hareketleri bu eğitim dokümanının konusu değildir. Ayrıntılı bilgi için, KUKA System Software 8.2 Kullanım ve Programlama Kılavuzuna bakın.

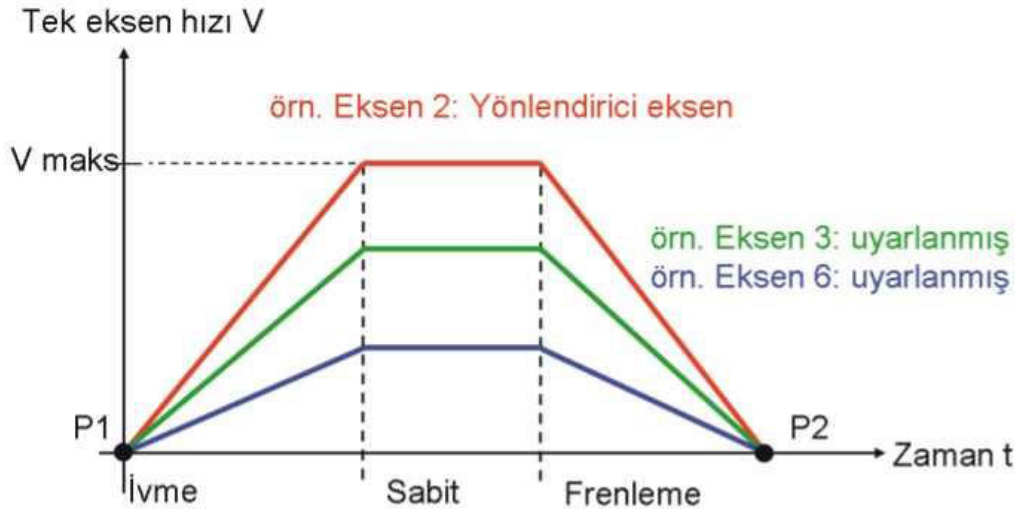
10.3 Çevrim Süresi En Uygun Hareketler Oluşturma (Aks Hareketi)

PTP hareket türü

Hareket türü	Anlamı	Kullanım örneği
	<p><i>Point-To-Point</i>: Noktadan noktaya</p> <ul style="list-style-type: none">Aksa özgü hareket: Robot, TCP'yi en hızlı rotadan hedef noktasına götürür. En hızlı rota genelde en kısa rota değildir ve dolayısıyla bir doğru şeklinde değildir. Robot aksları dönel olarak hareket ettiğinden, kavisli rotalar düz rotalara göre daha hızlı uygulanabilir.Hareketin kesin akışı öngörülemez.Hedef noktasına varmak için en uzun süreye gerek duyan aks, lider aks olarak tanımlanır.SYNCHRO PTP: Tüm akslar aynı anda başlar ve senkron olarak da dururlar.Programdaki ilk hareket bir PTP hareketi olmak zorundadır, çünkü sadece burada Durum ve Dönüş değerlendirilebilir.	<p>Nokta uygulamalarında, örn.:</p> <ul style="list-style-type: none">Punta kaynakTaşımaÖlçüm, kontrol <p>Yardımcı pozisyonlar:</p> <ul style="list-style-type: none">Ara noktalarAlanda serbest noktalar

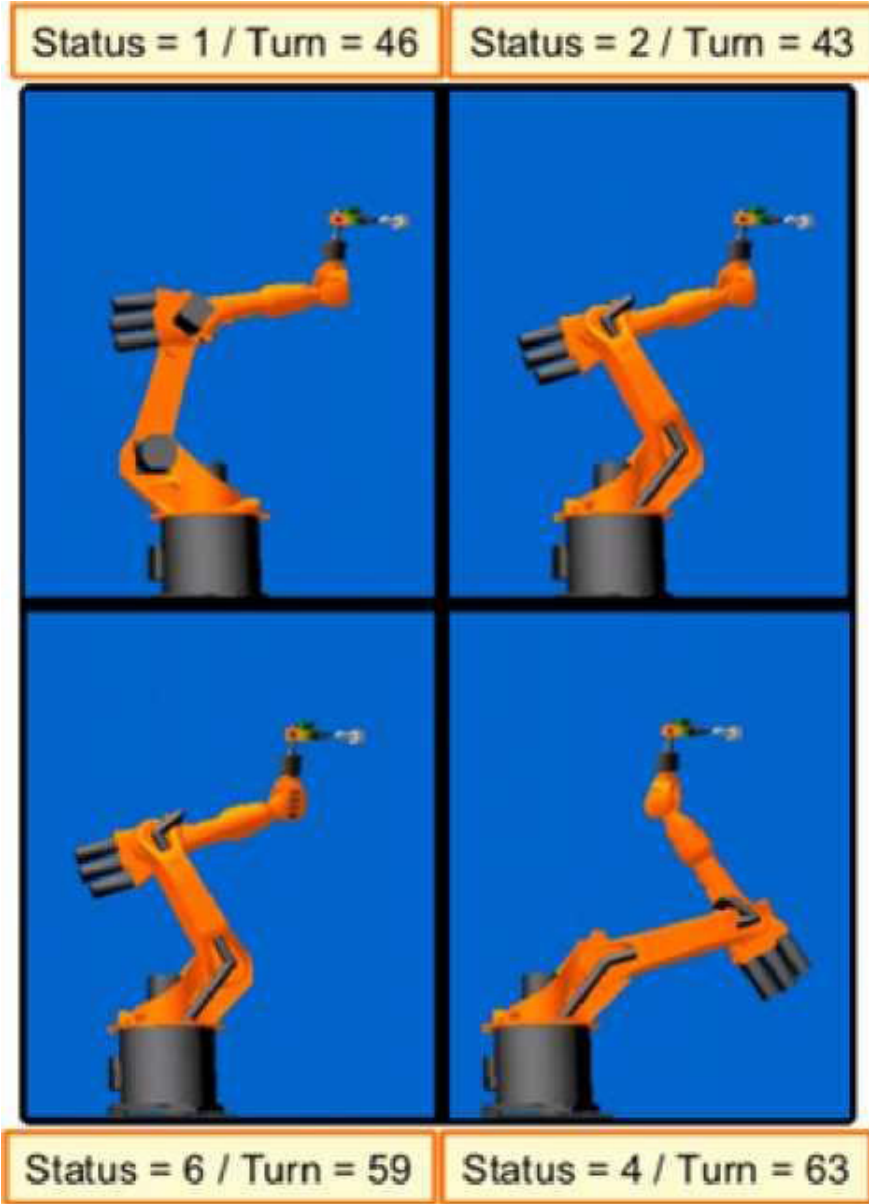
Syncho-PTP

Hedef noktasına varmak için en uzun süreye gerek duyan aks, lider aks olarak tanımlanır. Bu sırada İline formundaki hız bilgisi dikkate alınır.



Resim 10-3: Synchro-PTP

Durum & Turn "Status" ve "Turn", aynı TCP pozisyonuna sahip birçok olası aks konumları arasından belli bir aks pozisyonu belirlemeye yarar.

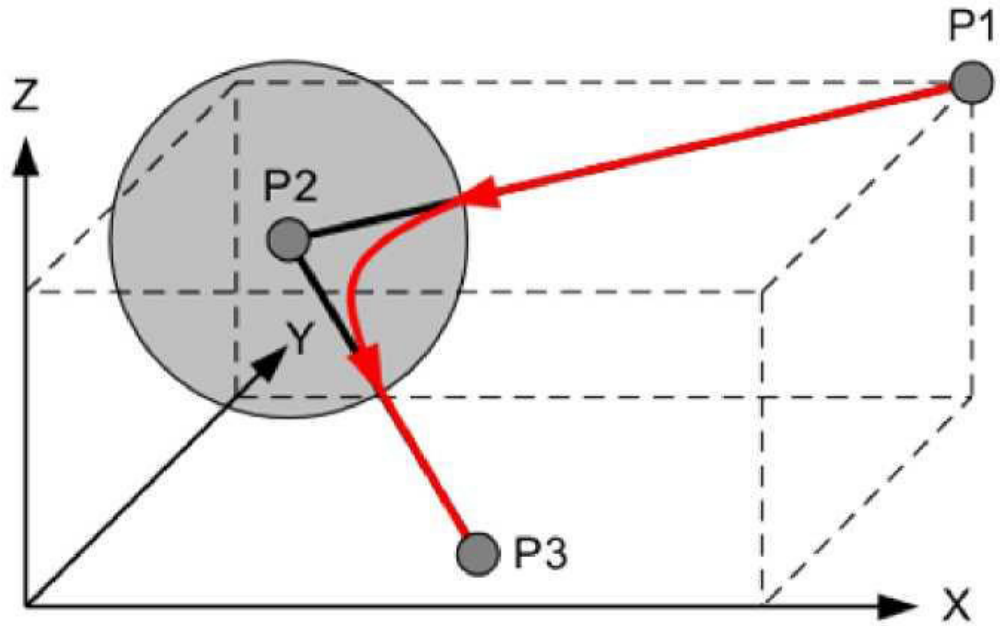


Resim 10-4: Farklı "Status" ve "Turn" değerleri sonucu farklı eksen konumları

Robot kumandası, programlanan Status ve Turn değerlerini sadece PTP hareketlerinde dikkate alır. CP hareketlerinde bu değerler yoksayılr. Bu nedenle bir KRL programında ilk hareket talimatı, POS veya E6POS tipi komple bir PTP talimatı olmalıdır. Aksi durumda başlangıç pozisyonu kesin bir biçimde tanımlanmamış olur. (AXIS veya E6AXIS tipi, komple PTP talimatı da mümkündür.)

```
DEFDAT MAINPROGRAM ()  
DECL POS XPOINT1={X 900, Y 0, Z 800, A 0, B 0, C 0, S 6, T 27}  
DECL FDAT FPOINT1 ...  
...  
ENDDDAT
```

Nokta atlama

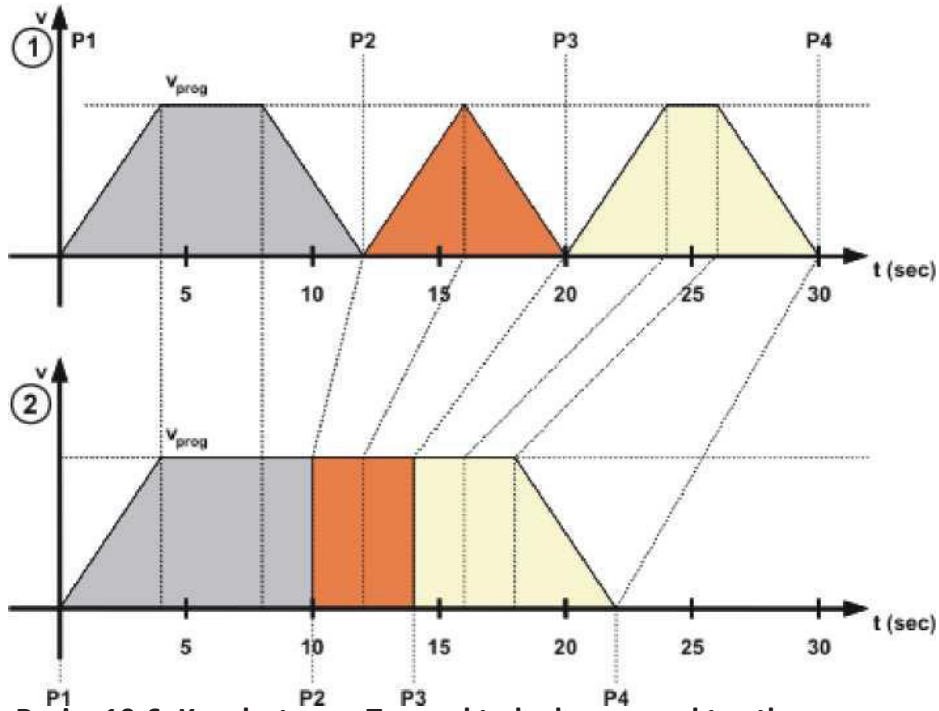


Resim 10-5: Nokta atlama

Kontrol sistemi, hareket sürecini hızlandırmak için CONT ile işaretli olan hareket komutlarını nokta atlayarak uygulama imkanına sahiptir. Nokta atlama, nokta koordinatına tam olarak gidilmediği anlamına gelir. Doğru yörünge konturunun yörüngesi daha önce terk edilir. TCP, bir nokta atlama konturu boyunca götürülür. Bu kontur, sonraki hareket komutunun doğru yörünge konturuna geçiş yapar.

Nokta atlamanın avantajları

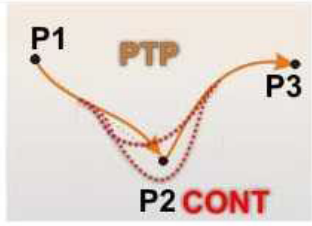
- Daha az aşınma
- Daha az takt süreleri



Resim 10-6: Karşılaştırma: Tam noktada durma - nokta atlama

Bir nokta atlama hareketi yapabilmek için, kontrol sistemi sonraki hareket komutu setlerini okuyabilecek durumda olmalıdır. Bu durum, bilgisayarda ön yürütmeye gerçekleştirilir.

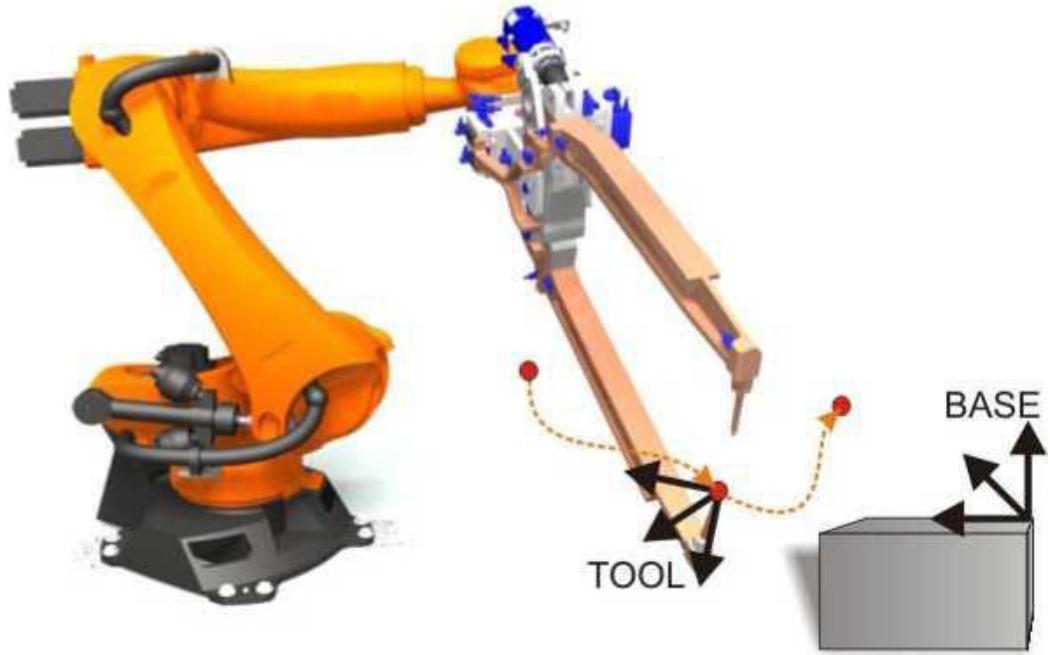
PTP hareket türünde nokta atlama

Hareket türü	Özellik	Atlama mesafesi
	<ul style="list-style-type: none"> Nokta atlama konturu öngörülemezdir 	% değeri

PTP hareketlerini oluştururken izlenecek yöntem

Ön koşullar oluşturmada

- T1 işletme türü ayarlı izlenecek yöntem
 - Robot programı seçili
- TCP'yi, hedef nokta olarak tanıtlamak (teach) olan konuma hareket ettirin



Resim 10-7: Hareket komutu

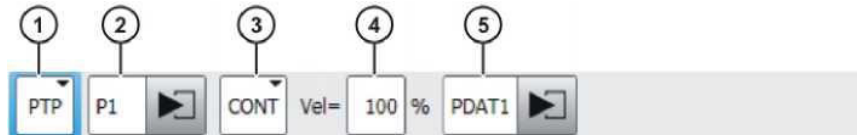
2. İmleci, hareket talimatım eklemek istediğiniz satırın sonrasına koyun.

3. Menü dizisi **Komutlar > Hareket > PTP**

Alternatif olarak ilgili satırdayken **Hareket** tuşuna da basılabilir.

Bir inline formu görüntülenir.

- PTP inline formu**



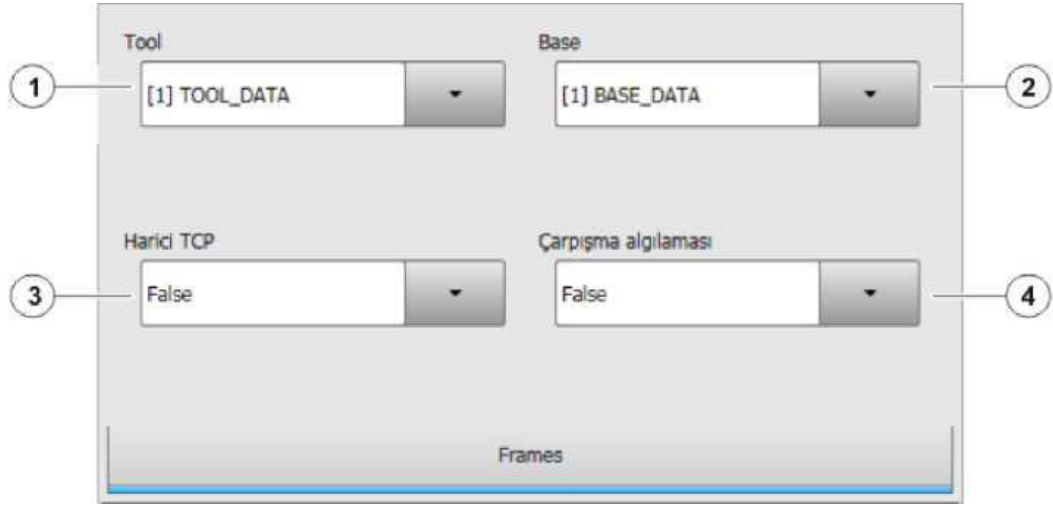
Resim 10-8: Inline-Formu PTP-Hareketi

4. İinline formuna parametre girin.

Poz.	Açıklama
1	PTP, LIN veya CIRC hareket türü
2	Hedef noktasının adı otomatik olarak verilir fakat münferit olarak üzerine yazmak mümkündür. Nokta verilerini düzenlemek için ok üzerine dokunun. Frames opsiyon penceresi açılır. CIRC türünde, hedef noktaya ek olarak bir de yardımcı nokta tanıtlmalıdır (teach). Yardımcı noktanın konumuna gidin ve Touchup HP üzerine basın

Poz.	Açıklama
3	<ul style="list-style-type: none"> ■ CONT: Hedef noktasının üzerinden atlanır. ■ [boş]: Hedef noktasının tam üzerine gidilir.
4	Hız <ul style="list-style-type: none"> ■ PTP hareketleri: 1 ... 100 % ■ Rota hareketleri: 0,001 ... 2 m/s
5	Hareket veri kaydı: <ul style="list-style-type: none"> ■ İvme ■ Üzerinden atlama mesafesi (alanda (3) CONT girildiğinde) ■ Yönlendirme kılavuzu (sadece rota hareketlerinde)

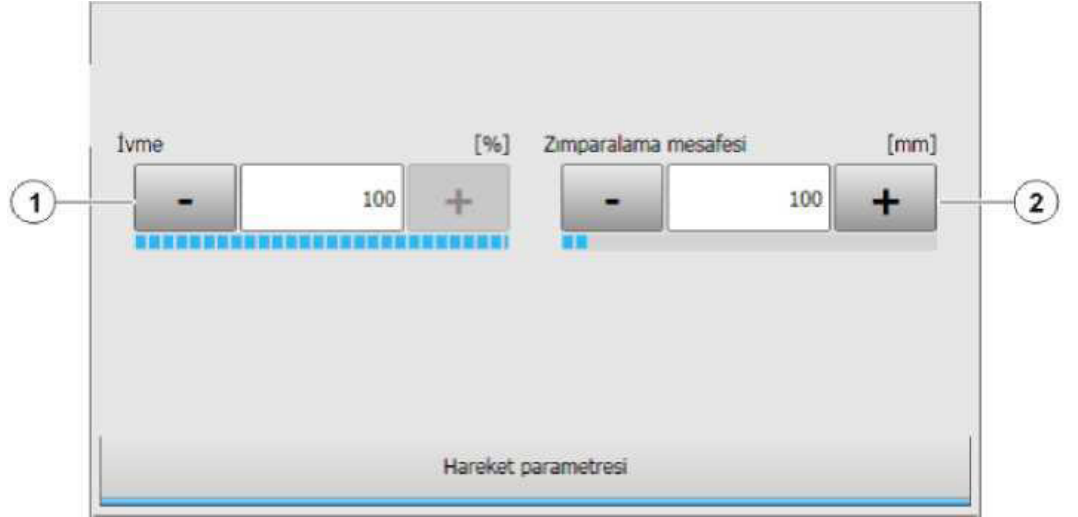
5. Frames opsiyon penceresinde tool ve base koordinat sistemi için doğru verileri girin ayrıca enterpolasyon modu (harici TCP: açık/kapalı) ve çarpışma denetimine ait bilgileri girin.



Resim 10-9: Opsiyon penceresi Frames

Poz.	Açıklama
1	Aleti seçin. External TCP alanında True olduğunda: İşlenen parçayı seçin. Değer aralığı: [1] ... [16]
2	Base seçin. External TCP alanında True olduğunda: Sabit duran aleti seçin. Değer aralığı: [1] ... [32]
3	Enterpolasyon modu <ul style="list-style-type: none"> ■ False: Alet montaj flanşına monte edilmiştir. ■ True: Alet, sabit duran bir alettir.
4	<ul style="list-style-type: none"> ■ True: Robot kumandası, bu hareket için aks momentleri belirler. Çarpışmanın algılanması için bunlara ihtiyaç duyulur. ■ False: Robot kumandası, bu hareket için aks momentleri belirlemez. Bu nedenle bu hareket için çarpışmanın algılanması mümkün değildir.

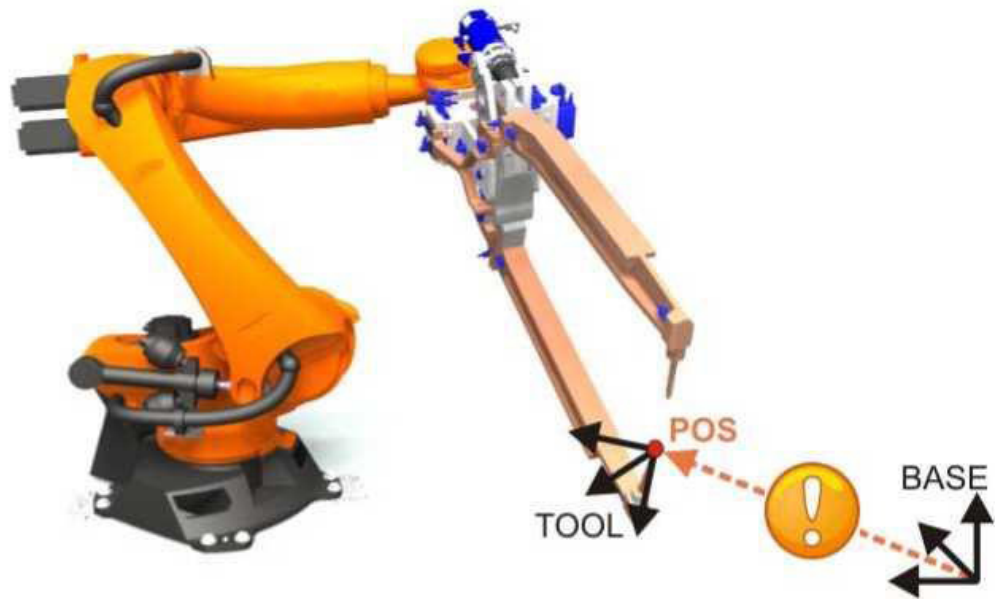
6. Hareket parametreleri opsiyon penceresinde hızlanma azami değerden düşürülebilir. Eğer üzerinden atlama etkinleştirildiyse üzerinden atlama mesafesi de değiştirilebilir. Konfigürasyona bağlı olarak mesafe mm ya da % olarak ayarlanır.



Resim 10-10: Opsiyon penceresi Hareket parametresi (PTP)

Poz.	Açıklama
1	<p>İvme</p> <p>Makine verilerinde belirtilen maksimum değer referans alınır. Maksimum değer, robot tipine ve ayarlanan işletim türüne bağlıdır. İvme, bu hareket cümlesinin lider aksı için geçerlidir.</p> <p>■ 1 ... 100 %</p>
2	<p>Bu alan sadece inline formunda CONT seçildiğinde gösterilir. Üzerinden atlamanın en erken başladığı, hedef noktasından önceki mesafe.</p> <p>Maksimum mesafe: Üzerinden atlama olmadan PTP hareketinin konturuna göre, başlangıç noktası ile hedef noktası arasındaki mesafenin en fazla yarısı olabilir.</p> <p>■ 1 ... 100 %</p> <p>■ 1 ... 1000 mm</p>


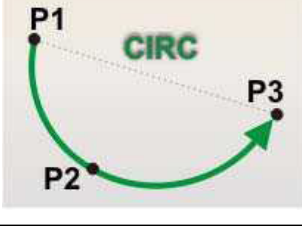
7. Komut OK ile talimatı kaydedin. TCP'nin güncel konumu, hedef noktası olarak tanıtlır (teach).



Resim 10-11: "Komut Tamam" ve "Touchup" için nokta koordinatlarının kaydetme

10.4 Rota Hareketleri Oluşturma

LIN ve CIRC hareket türleri

Hareket türü	Anlamı	Kullanım örneği
	Linear: Lineer ■ Doğrusal rota hareketi: ■ Aletin TCP'si, sabit hız ve tanımlı bir oryantasyon ile başlangıç noktasından hedef noktasına götürülür. ■ Hız ve oryantasyon TCP'ye göreler.	Rota uygulamaları, örn.: ■ Rotalı kaynak ■ Yapıştırma ■ Lazerle kaynak/kesme
	Dairesel: Dairesel ■ Dairesel rota hareketi, başlangıç noktası, yardımcı nokta ve hedef noktası ile tanımlanmıştır. ■ Aletin TCP'si, sabit hız ve tanımlı bir oryantasyon ile başlangıç noktasından hedef noktasına götürülür. ■ Hız ve oryantasyon/yönlendirme alete (alet koordinat sistemi) göreler.	LIN'de olduğu gibi rota uygulamaları: ■ Daireler, yarı çaplar, yuvarlaklıklar

Tekil konumlar

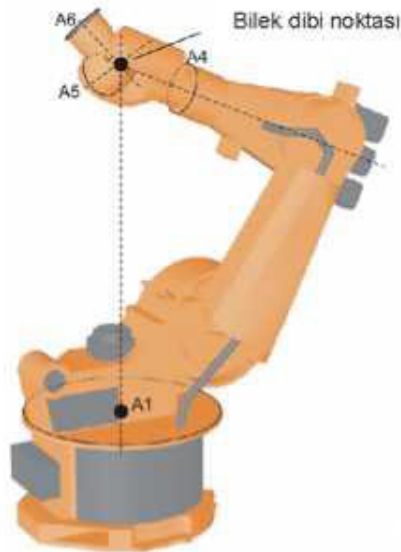
6 serbestlik derecelerine sahip robotları 3 farklı tekil konuma sahiptir.

Bir tekil konum, verilen duruma ve turn'e rağmen bir geriye transformasyonun (kartezik koordinatların aksa özgü değerlere dönüştürülmesi) tam olarak mümkün olmamasıyla belirlenir. Bu durumda veya en küçük kartezik değişiklikler çok büyük aks açısı değişikliklerine neden olduğunda, tekil konumlardan söz edilir. Bu mekanik değil matematiksel bir özelliktir ve bu nedenle sadece rota alanında mevcuttur ve aks hareketlerinde bulunmaz.

Baş aşağı tekillik α_1

Baş aşağı tekilliğinde bilek dibi noktası (= A5 ekseninin orta noktası), düşey olarak robotun A1 ekseninin üzerinde bulunur.

A1 ekseninin pozisyonu geriye transformasyon sayesinde açıkça belirlenemez ve bu nedenle istenildiği kadar değer alabilir.



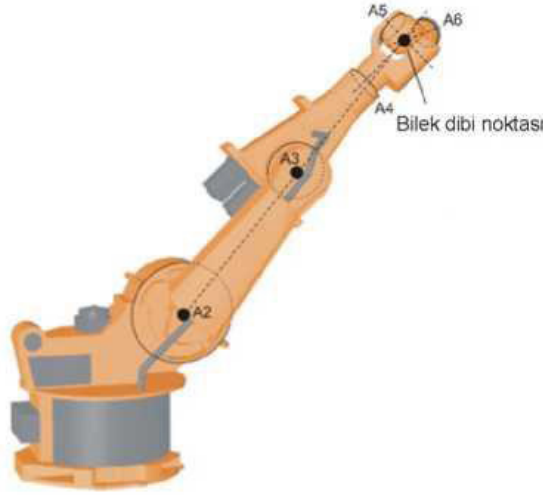
Resim 10-12: Baş aşağı tekillik (α_1 konumu)

Uzama konumları tekillik α_2

Uzama katmanı tekilliğinde bilek dibi noktası (= A5 ekseninin orta noktası), robotun A2 ve A3 ekseninin uzantısında bulunur.

Robot, kendi çalışma bölgesinin sınırında bulunmaktadır.

Geriye transformasyon açık eksen açıları sağlar, ancak küçük kartezyen hızların sonucunda A2 ve A3 eksenlerinin büyük eksen hızları vardır.

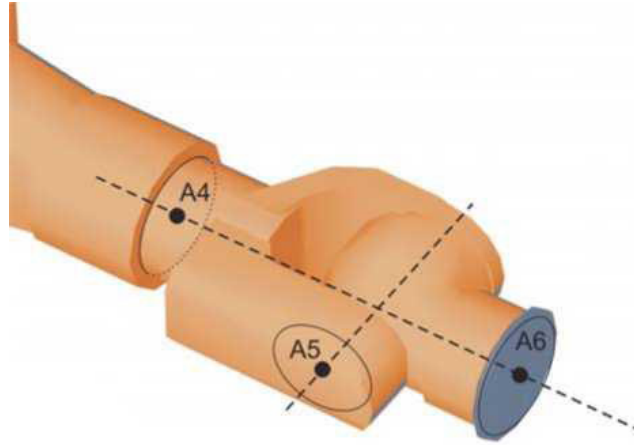


Resim 10-13: Uzama konumu (α_2 konumu)

El eksen tekilliğinde A4 ve A6 eksenleri birbirlerine paralel ve eksen A5 $\pm 0,01812^\circ$ alanı dahilinde bulunur.

Bilek eksen tekilliği α_5

Her iki eksenin konumu, bir geriye transformasyon sayesinde açıkça belirlenemez. Ancak eksen açı toplamları özdeş olan A4 ve A6 eksenleri için istenildiği kadar çok eksen konumu bulunur.



Resim 10-14: El eksen tekillik (α_5 konumu)

Yörünge hareketlerinde yönelim kılavuzu

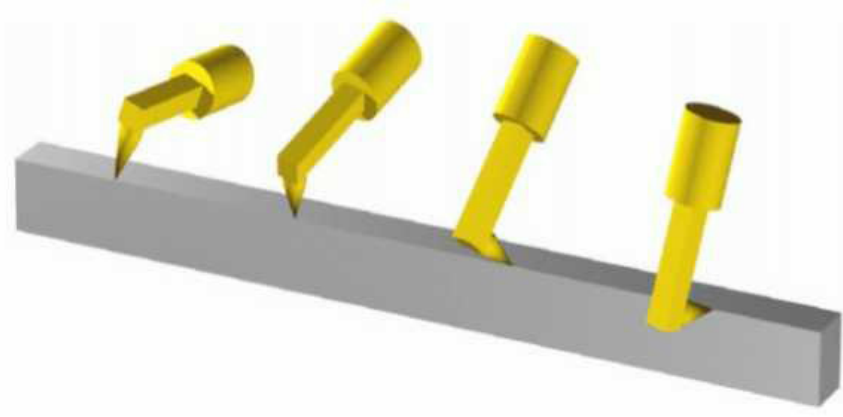
Yörüngeli hareketlerde yönelim kılavuzunu tam olarak tanımlama olanağı vardır. Alet, bir hareketin start ve varış noktasında farklı yönelimlere sahip olabilmektedir.

Yönelim kılavuzluğu, hareket türü **LIN**

- **Standart** ya da **Bilek PTP**

Aletin yönelimi, hareket sırasında sürekli değişir.

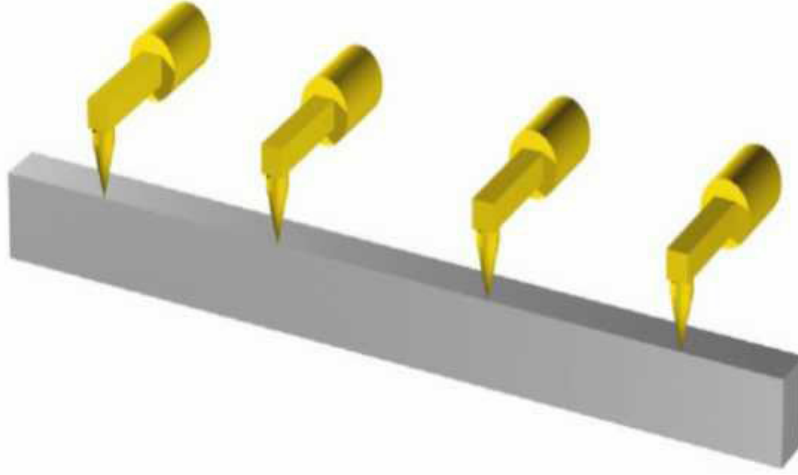
Bilek PTP olanağını robot, Standart seçeneğiyle bir bilek eksen tekilliğine girdiğinde kullanın. Çünkü yönelim bilek eksen açısının lineer aktarımı (eksene özel sürme) ile gerçekleştirilmektedir.



Resim 10-15: Standart

- **Sabit**

Aletin yönelimi, başlangıç noktasında öğretildiği gibi, hareket sırasında sabit kalır. Son noktada öğretilen yönelim yoksayılır.



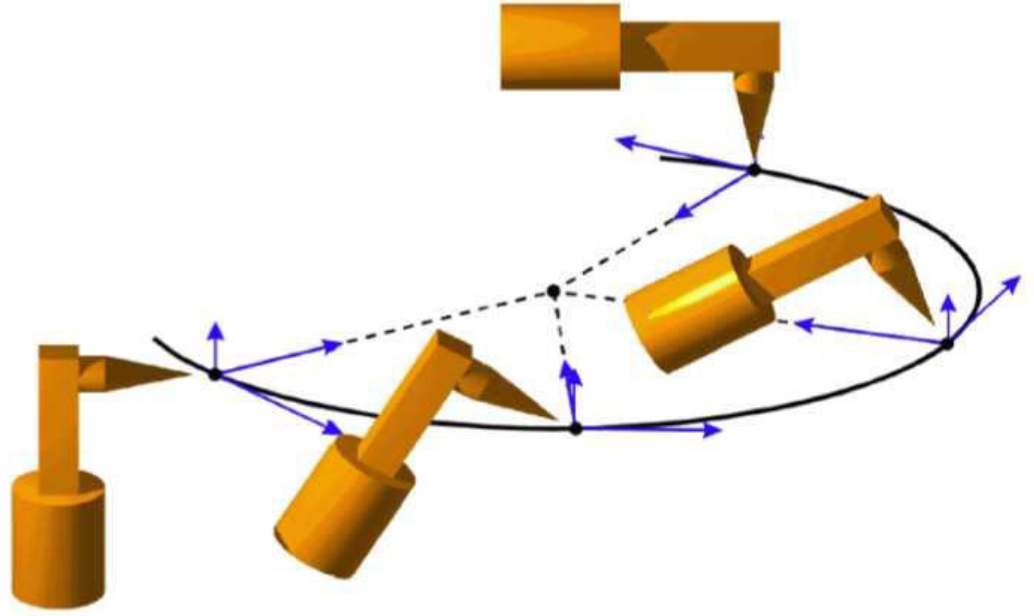
Resim 10-16: Sabit yönlendirme

Yönelim kılavuzluğu, hareket türü **CIRC**

- **Standart** ya da **Bilek PTP**

Aletin yönelimi, hareket sırasında sürekli değişir.

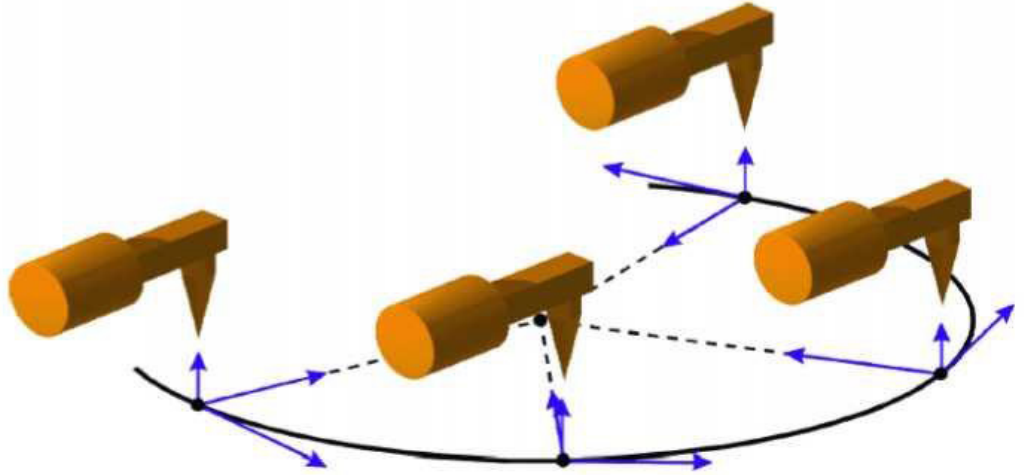
Bilek PTP olanağını robot, Standart seçeneğiyle bir bilek eksenine tekiliğine girdiğinde kullanın. Çünkü yönelim bilek ekseninin açısının lineer aktarımı (eksene özel sürme) ile gerçekleştirilmektedir.



Resim 10-17: Standart + düzleme ilişkin

- **Sabit**

Aletin yönelimi, başlangıç noktasında öğretildiği gibi, hareket sırasında sabit kalır. Son noktada öğretilen yönelim yoksayılr.

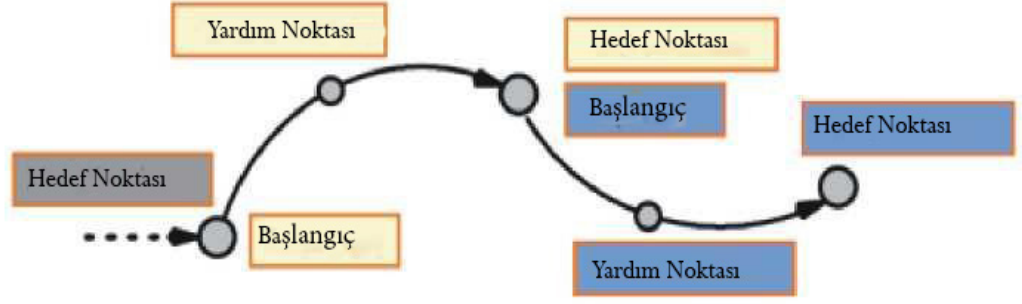


Resim 10-18: Sabit yönlendirme kılavuzu + düzleme ilişkin

CIRC'te hareket akışı

Burada aletin ya da işlenen parçanın referans noktası, dairesel bir yay üzerinde hedef noktaya hareket eder. Rota, başlangıç noktası, yardımcı nokta ve hedef noktası ile tanımlanır. Başlangıç noktası ise, bir önceki hareket komutunun hedef noktasıdır.

Aletin yardımcı noktadaki oryantasyonunun bir önemi yoktur.



Resim 10-19: CIRC ile iki daire parçası

Rota hareketlerinin üzerinden atlama

i Üzerinden atlama fonksiyonu, dairesel hareketler oluşturmak için uygun değildir. Sadece bir noktada tam olarak durma yapılmasının engellenmesi için kullanılır.

PTP, LIN ve CIRC hareket türlerinde üzerinden atlama

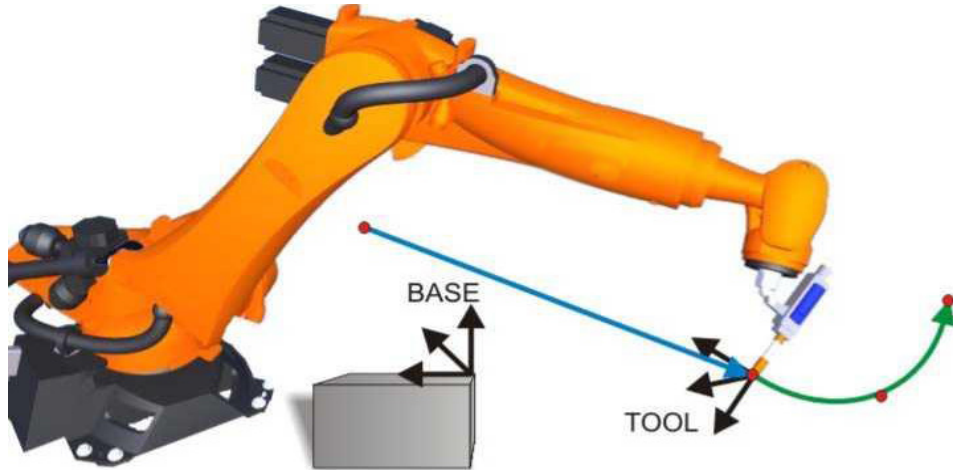
Hareket türü	Özellik	Üzerinden atlama mesafesi
	<ul style="list-style-type: none"> Yörünge seyri iki parabol dalına karşılık gelir 	mm değeri
	<ul style="list-style-type: none"> Yörünge seyri iki parabol dalına karşılık gelir 	mm değeri

LIN ve CIRC hareketlerinin oluşturulmasında izlenecek yöntem

Ön koşullar

- T1 işletme türü ayarlı oluşturulmasında izlenecek
- Robot programı seçili.

1. TCP'yi, hedef nokta olarak tanımlanacak (teach) olan konuma hareket ettirin



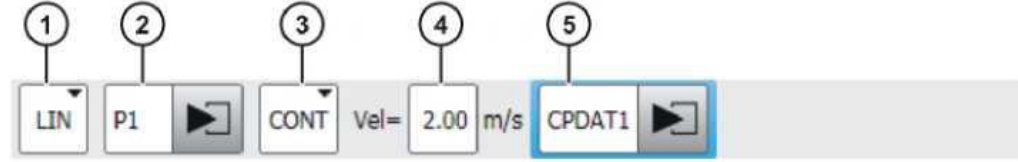
Resim 10-20: LIN ve CIRC ile hareket komutu

2. İmleci, hareket talimatını eklemek istediğiniz satırın sonrasına koyun.

3. Komutlar > Hareket > LIN veya CIRC seçin.

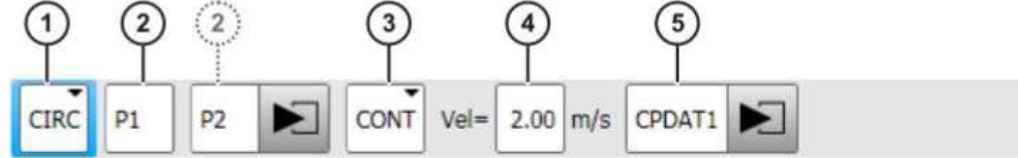
Alternatif olarak ilgili satırdayken **Hareket** tuşuna da basılabilir. Bir inline formu görüntülenir.

• LIN inline formu



Resim 10-21: LIN-Hareketi Inline-Formu

• CIRC inline formu

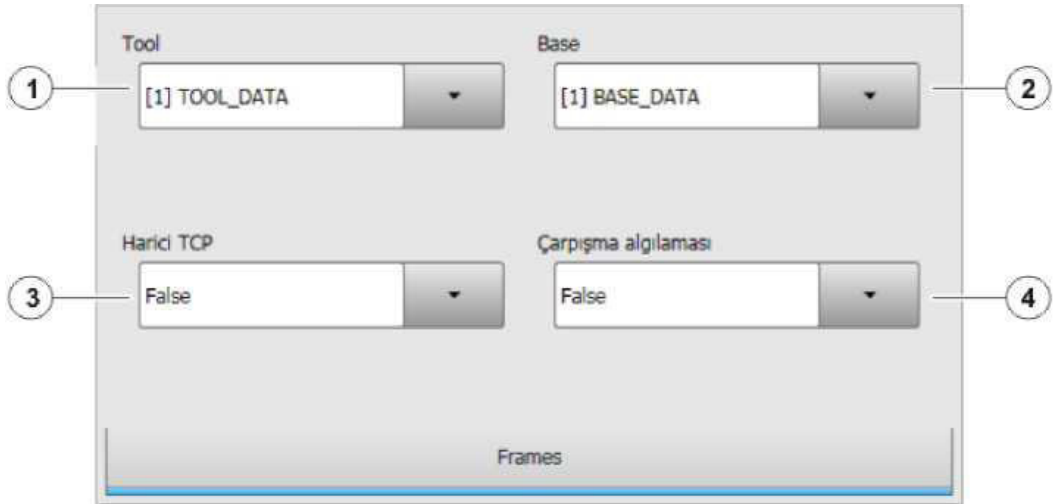


Resim 10-22: İinline formu CIRC hareketi

4. Inline formuna parametre girin.

Poz	Açıklama
1	PTP, LIN veya CIRC hareket türü
2	Hedef noktasının adı otomatik olarak verilir fakat münferit olarak üzerine yazmak mümkündür. Nokta verilerini düzenlemek için ok üzerine dokununuz. Frames opsiyon penceresi açılır. CIRC türünde, hedef noktaya ek olarak bir de yardımcı nokta tanıtılmalıdır (teach). Yardımcı noktanın konumuna gidiniz ve Touchup HP üzerine basınız. Aletin yardımcı noktadaki oryantasyonunun bir önemi yoktur.
3	■ CONT : Hedef noktasının üzerinden atlanır. ■ [boş] : Hedef noktasının tam üzerine gidilir.
4	Hız ■ PTP hareketleri: 1 ... 100 % ■ Rota hareketleri: 0,001 ... 2 m/s
5	Hareket veri kaydı: ■ İvme ■ Üzerinden atlama mesafesi (alanda (3) CONT girildiğinde) ■ Yönlendirme kılavuzu (sadece rota hareketlerinde)

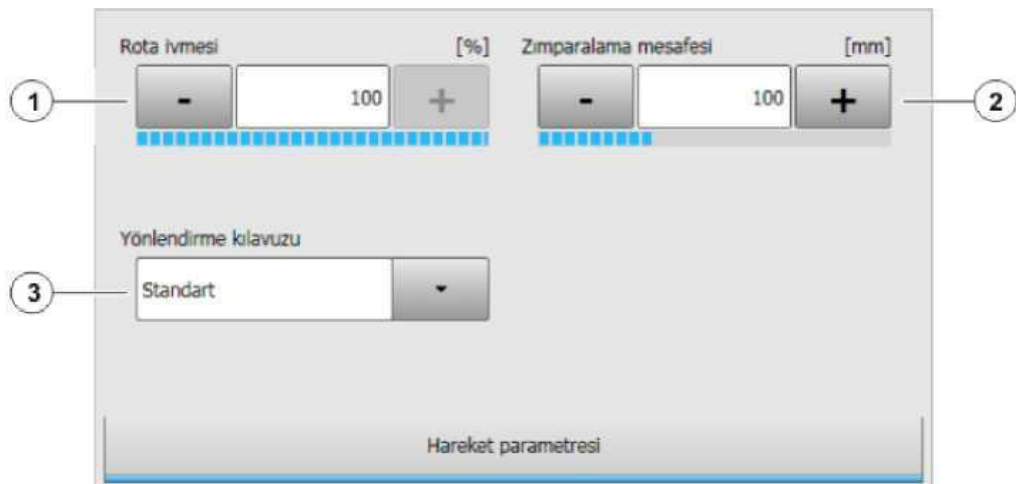
5. Frames opsiyon penceresinde tool ve base koordinat sistemi için doğru verileri giriniz ayrıca enterpolasyon modu (harici TCP: açık/kapalı) ve çarpışma denetimine ait bilgileri giriniz.



Resim 10-23: Opsiyon penceresi Frames

Poz	Açıklama
1	Aleti seçin. External TCP alanında True olduğunda: İşlenen parçayı seçin. Değer aralığı: [1] ... [16]
2	Base seçin. External TCP alanında True olduğunda: Sabit duran aleti seçin. Değer aralığı: [1] ... [32]
3	Enterpolasyon modu <ul style="list-style-type: none"> ■ False: Alet montaj flanşına monte edilmiştir. ■ True: Alet, sabit duran bir alettir.
4	<ul style="list-style-type: none"> ■ True: Robot kumandası, bu hareket için aks momentleri belirler. Çarpışmanın algılanması için bunlara ihtiyaç duyulur. ■ False: Robot kumandası, bu hareket için aks momentleri belirlemez. Bu nedenle bu hareket için çarpışmanın algılanması mümkün değildir.

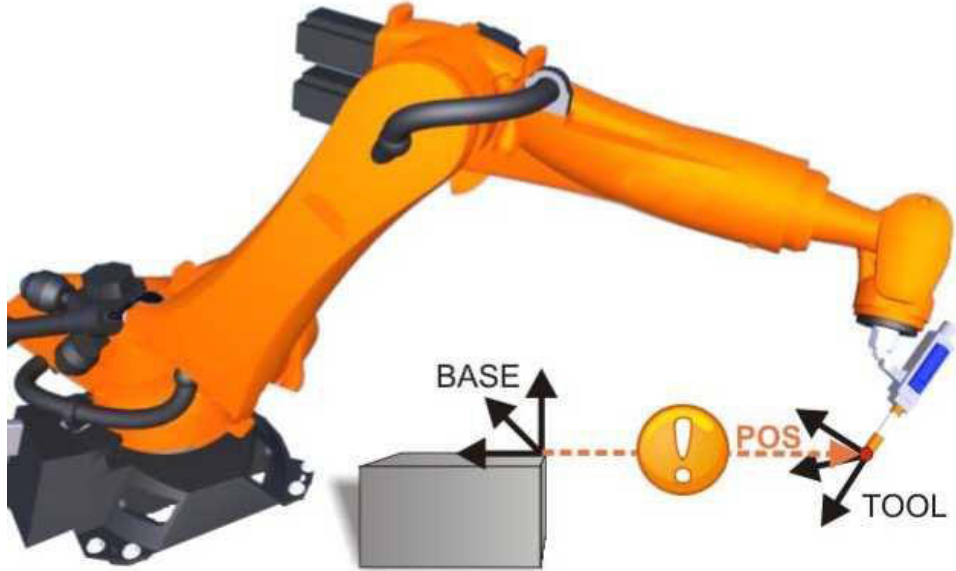
6. Hareket parametreleri opsiyon penceresinde hızlanma azami değerden düşürülebilir. Eğer üzerinden atlama etkinleştirildiyse üzerinden atlama mesafesi de değiştirilebilir. Ayrıca yönlendirme kılavuzu da değiştirilebilir.



Resim 10-24: Opsiyon pencereleri Hareket parametresi (LIN, CIRC)

Poz.	Açıklama
1	İvme Makine verilerinde belirtilen maksimum değer referans alınır. Maksimum değer, robot tipine ve ayarlanan işletim türüne bağlıdır.
2	Üzerinden atlamanın en erken başladığı, hedef noktasından önceki mesafe. Mesafe, başlangıç noktası ile hedef noktası arasındaki mesafenin en fazla yarısı olabilir. Eğer buraya daha yüksek bir değer girilirse, bu görmezden gelinir ve maksimum değer kullanılır. Bu alan sadece inline formunda CONT seçildiğinde gösterilir.
3	Yönlendirme kılavuzu seçin. <ul style="list-style-type: none"> ■ Standart ■ Elle PTP ■ Sabit yönlendirme kılavuzu (>>> "Yörünge hareketlerinde yönelim kılavuzu" Sayfa 124)

7. **Komut OK** ile talimatı kaydedin. TCP'nin güncel konumu, hedef noktası olarak tanımlanır (teach).



Resim 10-25: "Komut Tamam" ve "Touchup" için nokta koordinatlarının kaydetme

10.5 Hareket Komutlarını Değiştirme

Mevcut hareket komutlarını değiştirmenin çok farklı nedenleri olabilir:

Hareket komutlarını değiştirme

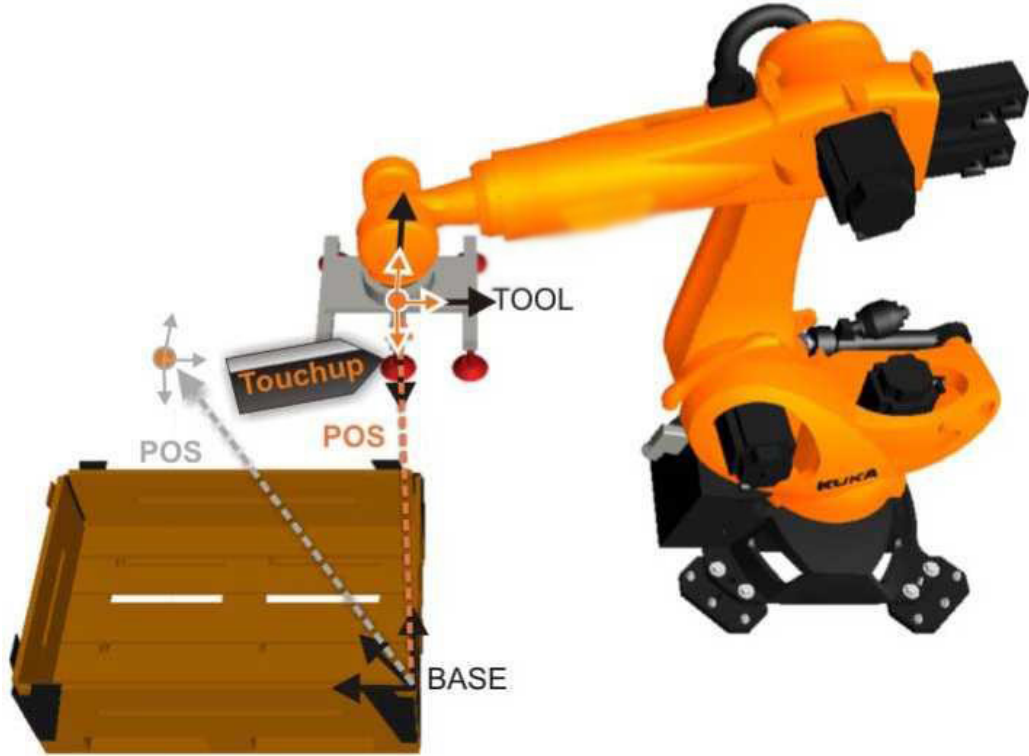
Örnek nedenler	Yapılacak değişiklik
Kavranacak parçanın konumu değişmektedir.	Pozisyon verilerini değiştirme
İşleme sırasında beş delikten birinin pozisyonu değişmiştir.	
Bir kaynak dikişinin kısaltılması gerekiyor.	Frame verilerinde değişiklik: Base ve/veya tool
Paletin konumu değişiyor.	

Örnek nedenler	Yapılacak değişiklik
Yanl ışıkla bir pozisyon yanlış base ile tanıtıldı (teach).	Frame verilerinde değişiklik: Konumun güncellenmesi ile base ve/ veya tool
İşleme çok yavaş: Çevrim süresinin iyileştirilmesi gerekir.	Hareket verilerinde değişiklik: Hız, ivme Hareket türünü değiştirme

Pozisyon verilerini değiştirme

Hareket komutları değiştirildiğinde ortaya çıkan etkiler

- Sadece noktanın veri kaydı değiştirilir: "Touchup" ile değerler güncelleştirdiğinden nokta yeni koordinatlara sahip olur.
Eski nokta koordinatlarının üzerinde yazılır ve artık kullanılamaz!

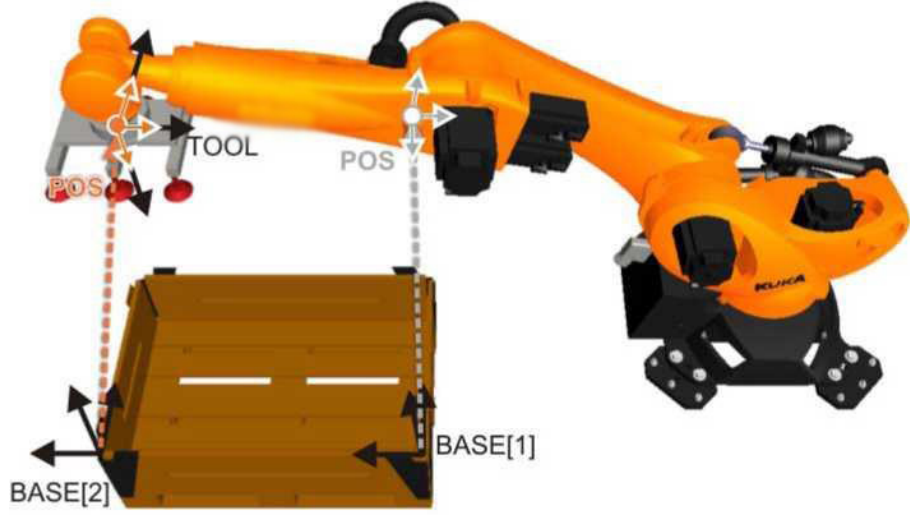


Resim 10-26: Robot pozisyonu "Touchup" ile değiştirme

Frame verilerini değiştirme

- Frame verileri (örn. tool, base) değiştirildiğinde pozisyon kaydırılır (bkz. "Vektör kaydırma")
- Robot konumu değişir!
- Noktanın eski koordinatları yine kayıtlıdır ve geçerlidir. Sadece referans (örn. base) değişir.
- Çalışma bölgesinden çıkma durumu ortaya çıkabilir! Böylece belirli robot konumları artık ulaşılmazdır.
- Robot konumunun aynı kalması isteniyor ancak Frame parametrelerinin değiştirilmesi isteniyorsa, parametreler (örn. base) değiştirildikten sonra istenen konumda koordinatlar "Touchup" komutuyla güncellenmelidir!

i Bir operatör iletisi ek bir uyarı verir: "Dikkat,nokta ilişkili frame parametrelerinin değiştirilmesinde çarpışma tehlikesi vardır!"



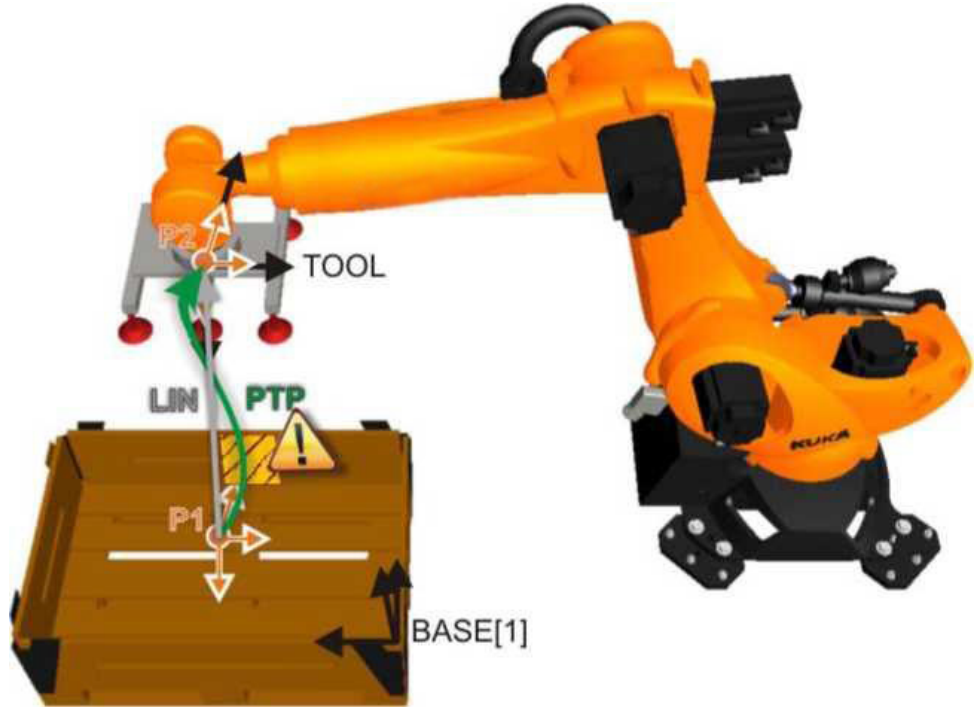
Resim 10-27: Frame verileri değişikliği (örnek olarak Base)

Hareket verilerini değiştirme

- Hız ve ivme değiştirildiğinde sürüş profili değişir. Bunun, işleme süreci üzerinde etkisi olabilir; özellikle rota uygulamalarında:
- Bir yapıştırma halkasının kalınlığı
- Bir kaynak dikişinin kalitesi

Bir hareket türünü değiştirme

- Hareket türünün değiştirilmesi, rota planında daima bir değişikliğe neden olur! Rota öngörülemez şekilde değişebileceğinden dolayı bu, olumsuz koşullarda çarpışmalara yol açabilir.



Resim 10-28: Hareket türünü değiştirme


Hareket komutlarının değiştirilmesine ilişkin güvenlik uyarıları

DUYURU Hareket komutları her değiştirildiğinde robot programı düşürülmüş hızla (işletim türü T1) test edilmek zorundadır.

Robot programının hemen yüksek hızla başlatılması robot sisteminde ya da tüm tesiste hasarlara neden olabilir çünkü öngörülemeyen hareketler beklenebilir. Tehlike bölgesinde bir kişi varsa, hayati tehlike oluşturan yaralanmalar beklenebilir.

Hareket parametrelerini deęiřtirme - Frames

1. İmleci, deęiřtirilmesi gereken talimata sahip satıra yerleřtirin.
2. **Deęiřtir** üzerine basın. Talimatla ilgili İline formu açılır.
3. Frames opsiyon penceresini açın.
4. Yeni tool veya base veya harici TCP ayarlayın.
5. "Dikkat, noktaya iliřkin frame parametrelerinin deęiřtirilmesinde çarpıřma tehlikesi vardır!" kullanıcı diyaloęunu **OK** ile onaylayın.
6. **Güncel robot konumu**, deęiřtirilmiř tool ve/veya base ayarlarıyla **korunmak istendięinde** mutlaka **Touch Up** tuřuna basılarak güncel konumun yeniden hesaplanıp kaydedilmesi gerekir.
7. Deęiřiklikleri **Tamam** komutu ile kaydedin.

 Frame parametreleri deęiřtirildięinde programlar, çarpıřma durumu açısından yeniden kontrol edilmelidir

Pozisyon deęiřtirme


Robot pozisyonunu deęiřtirmek için izlenecek yöntem

1. İřletim türü olarak T1 ayarlayın ve imleci, deęiřtirilmesi istenen deyim için yer aldıęı satıra yerleřtirin.
2. Robotu gerekli pozisyona getirin.
3. **Deęiřtir** düęmesine basın. Deyim için inline formu açılır.
4. PTP- ve LIN-hareketleri için:
 - **Touchup** düęmesine basarak, TCP'nin güncel pozisyonunu yeni hedef noktası olarak üzerlenin.
 - CIRC-hareketleri için:
 - **Touchup HP** düęmesine basarak, TCP'nin güncel pozisyonunu yeni yardımcı nokta olarak üzerlenin.
 - Veya TCP'nin güncel pozisyonunu yeni hedef nokta olarak üzerlemek için **Touchup ZP** düęmesine basın.
5. Güvenlik sorusunu **Evet** ile onaylayın.
6. Deęiřiklięi **Komut OK** ile kaydedin.

Bu yöntem ařaęıdaki deęiřiklikler için kullanılabilir:

- Hareket türü
- Hız
- İvme
- Üzerinden atlama
- Üzerinden atlama mesafesi

1. İmleci, deęiřtirilmesi gereken talimata sahip satıra yerleřtirin.
2. **Deęiřtir** üzerine basın. Talimatla ilgili İline formu açılır.
3. Parametre deęiřtirme.
4. Deęiřiklikleri **Tamam** komutu ile kaydedin.

 Hareket parametreleri deęiřtirildięinde programlar, çarpıřma ve iřlem güvenlięi açısından yeniden kontrol edilmelidir.

11 Robot Programında Mantıksal Fonksiyonları Kullanma

11.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Bekleme fonksiyonlarını programlama
- Basit anahtarlama fonksiyonlarını programlama
- Rota anahtarlama fonksiyonlarını programlama

11.2 Mantık Programlamasına Giriş

Mantık programlamasında giriş ve çıkışların kullanımı



Resim 11-1: Dijital girişler ve çıkışlar

Robot kumandasının **çevresel birimlerle** olan iletişimini gerçekleştirmek için **dijital** ve **analog girişler ve çıkışlar** kullanılabilir.

Kavram açıklamaları

Kavram	Açıklama	Örnek
İletişim	Bir arabirim üzerinden sinyal alışverişi	Bir durumun sorgulanması (kavrayıcı açık/ kapalı)
Çevre birimleri	"Ortam"	Alet (örn. kavrayıcı, kaynak kısıkaçı vb.), sensörler, malzeme sevk sistemleri vb.
dijital	Dijital teknoloji: değer ve zaman açısından ayırık sinyaller	Sensör sinyali: Parça mevcut: Değer 1 (TRUE/DOĞRU), Parça yok: Değer 0 (FALSE/YANLIŞ)
analog	Fiziksel bir büyüklüğün görüntüsü	Sıcaklık ölçümü
Girişler	Alan bus arabirimi üzerinden kumanda sistemine <i>gelen</i> sinyaller	Sensör sinyali: Kavrayıcı açık / kavrayıcı kapalı
Çıkışlar	Kumanda sistemi tarafından alan bus arabirimi üzerinden çevre birimlerine <i>gönderilen</i> sinyaller	Çeneli bir kavrayıcının kapanmasına yol açan bir valf kapatma komutu.

Robotların programlanmasında, mantık talimatları için giriş ve çıkış sinyalleri kullanılır:

- **OUT** | Programın belli bir yerinde bir çıkışın anahtarlanması
- **WAIT FOR** | Sinyale bağlı bekleme fonksiyonu: Burada kumanda sistemi bir sinyal bekler:
- Giriş **IN**
- Çıkış **OUT**
- Zaman sinyali **TIMER**
- Kumanda sistemi dahilindeki bir bellek adresi (işaret/1 bit bellek) **FLAG** veya **CYCFLAG** (çevrimsel şekilde sürekli olarak değerlendirildiğinde)
- **WAIT** | Zamana bağlı bekleme fonksiyonu: Kumanda sistemi, programda bu noktada girilmiş olan süre kadar bekler.

11.3 Bekleme Fonksiyonlarını Programlama

Bilgisayarda ön çalışma

Bilgisayarda ön çalışma, hareket cümlelerini ön çalışmada okur (kullanıcı için görünmez şekilde) ve bu sayede üzerinden atlama komutlarında kumanda sistemine rotayı planlama olanağı verir. Ancak ön çalışma ile sadece hareket verileri değil, aritmetik talimatlar ve çevre birimlerini kumanda eden talimatlar da işlenir.

```
Editor
1 DEF Depal_Box1( )
2
3 INI
4 PTP HOME Vel= 100 % DEFAULT
5 PTP P1 Vel=100 % PDAT1 Tool[5]:GRP1 Base[10]:STAT1
6 PTP P2 Vel=100 % PDAT2 Tool[5]:GRP1 Base[10]:STAT1
7 LIN P3 Vel=1 m/s CPDAT1 Tool[5]:GRP1 Base[10]:STAT1
8 LIN P4 Vel=1 m/s CPDAT2 Tool[5]:GRP1 Base[10]:STAT1
9 PTP P5 Vel=100 % PDAT3 Tool[5]:GRP1 Base[10]:STAT1
10 OUT 26'' State=TRUE
11 PTP HOME Vel=100 % PDAT4
12
13 END
```

Resim 11-2: Bilgisayarda ön çalışma

Satır	
6	Ana yürütme göstergesinin pozisyonu (gri çubuk)
9	Ön çalışma göstergesinin olası pozisyonu (görünmez)
10	Ön çalışmayı durdur işlevine yol açan komut cümlesi

Bazı talimatlar, bir ön çalışmayı durdur işlevine yol açar. Bunların arasında çevre birimleri etkileyen talimatlar bulunur, örn. OUT talimatları (kavrayıcıyı kapat, kaynak kiskacını aç). Ön çalışma göstergesi durdurulursa, üzerinden atlama yapılamaz.

Bekleme fonksiyonları

Bir hareket programındaki bekleme fonksiyonları kolaylıkla inline formları üzerinden programlanabilmektedir. Bu işlemde zamana bağlı bekleme fonksiyonu ve sinyale bağlı bekleme fonksiyonu arasında ayırım yapılmaktadır.

WAIT komutuyla, programlanan süre için robot hareketi durdurulur. WAIT daima bir ön çalışmayı durdurma işlevi tetikler.



Resim 11-3: Inline-Formu WAIT

Poz	Açıklama
1	Bekleme süresi ■ ≥ 0 s

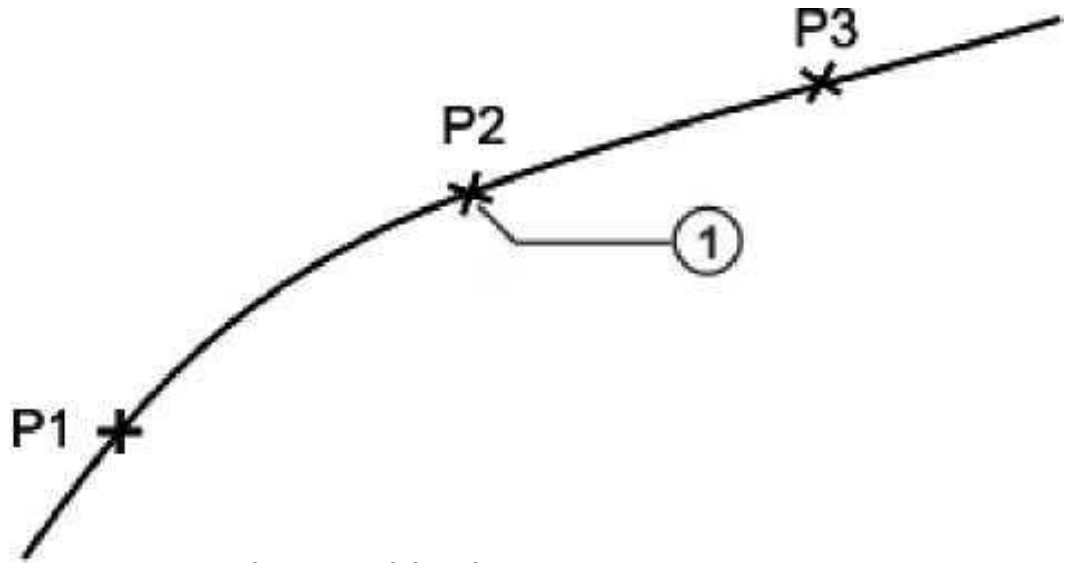
Örnek program:

PTP P1 Vel=100% PDAT1

PTP P2 Vel=100% PDAT2

WAIT Time=2 sec

PTP P3 Vel=100% PDAT3

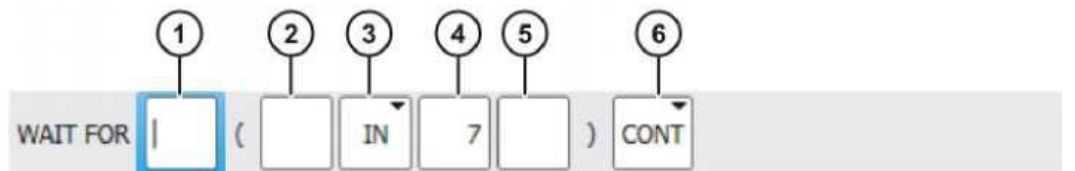


Resim 11-4: Mantık için örnek hareket


Poz	Notlar
1	Hareket 2 saniyelğine P2 noktasında kesilir.

WAIT FOR, sinyale bağlı bir bekleme fonksiyonu uygular.

Gerektiğinde birden fazla sinyal (maksimum 12) mantıksal olarak bağdaştırılabılır. Bir bağdaştırma eklendiğinde, Inline formunda ilave sinyaller ve diğer bağdaştırmalar için alanlar ekrana getirilir.



Resim 11-5: Inline-Formu WAITFOR

Poz	Açıklama
1	Dış bağdaştırma ekleyin. İşleç, parantez içindeki ifadelerin arasında yer alır. <ul style="list-style-type: none"> ■ AND ■ OR ■ EXOR NOT ekleyin. <ul style="list-style-type: none"> ■ NOT ■ [boş] İlgili butonu kullanarak istediğiniz işleci ekleyin.
2	İç bağdaştırma ekleyin. İşleç, parantez içindeki bir ifadenin içinde yer alır. <ul style="list-style-type: none"> ■ AND ■ OR ■ EXOR NOT ekleyin. <ul style="list-style-type: none"> ■ NOT ■ [boş] İlgili butonu kullanarak istediğiniz işleci ekleyin.
3	Beklenen sinyal <ul style="list-style-type: none"> ■ IN ■ OUT ■ CYCFLAG ■ TIMER ■ FLAG
4	Sinyalin numarası <ul style="list-style-type: none"> ■ 1 ... 4096
5	Sinyal için bir ad varsa burada görüntülenir. Sadece uzman kullanıcı grubu için: Uzun metin üzerine basılarak bir ad girilebilir. Ad serbestçe seçilebilir.
6	<ul style="list-style-type: none"> ■ CONT: Ön çalışmada işleme ■ [boş]: Ön çalışma stoplu işleme
	CONT girişi kullanıldığında sinyalin ön çalışma esnasında sorgulanmasına dikkat edilmelidir. Ön çalışma süresinden sonra sinyal değişiklikleri tanınmaz

Mantıksal bağlantılar

Sinyale bağlı bekleme fonksiyonları kullanıldığında mantıksal bağlantılar da kullanılmaktadır. Mantıksal ilişkilendirmelerle farklı sinyal veya durumların sorgulanması birleştirilebilir: Örneğin bağımlılıklar oluşturulabileceği gibi belirli durumların dışlanması da sağlanabilir.

Mantıksal işleci olan bir fonksiyonun sonucu daima bir doğruluk değeri verir, yani sonunda daima ya "Doğru" (değer 1) ya da "Yanlış" (değer 0) ortaya çıkar.

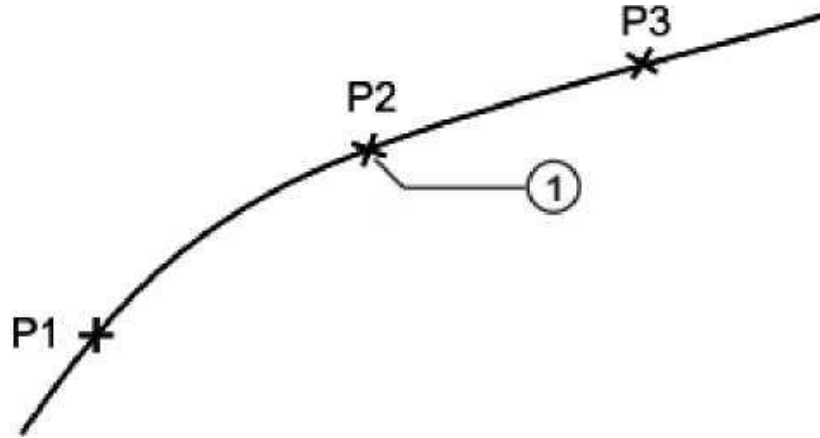


Resim 11-6: Mantıksal bağdaştırma örneği ve prensibi

- **NOT** | Bu işlenen, olumsuzlamak için kullanılır, yani değer değişir ("Doğru" olan "Yanlış" olur).
- **AND** | Bağlanan ifadelerin ikisi de doğru ise ifadenin sonucu doğrudur.
- **OR** | Bağlanan ifadelerin en az birisi doğru ise ifadenin sonucu doğrudur.
- **EXOR** | İşleç tarafından bağlanan ifadelerin ikisinin değeri farklı ise ifadenin sonucu doğrudur.

Ön çalışmalı ve ön çalışmasız işleme (CONT)

Sinyale bağlı bekleme fonksiyonları hem ön çalışmalı hem de ön çalışmasız olarak programlanabilir. **Ön çalışmasız** seçimi, hareketin her durumda noktada durdurulacağı ve sinyalin burada kontrol edileceği anlamına gelmektedir: (1) (>>> Resim 11-7): Yani noktanın üzerinden atlamak mümkün değildir.

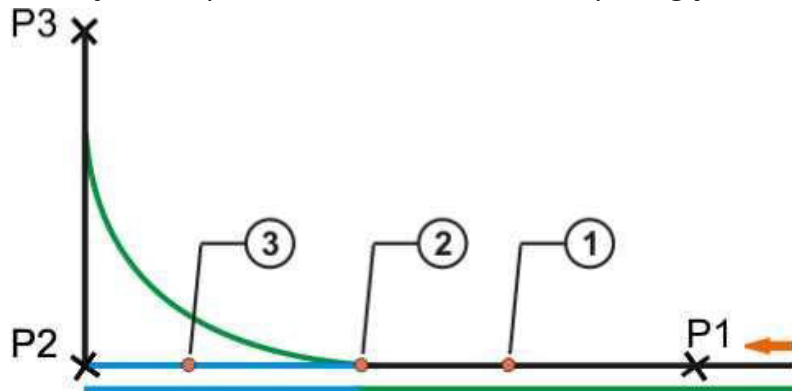


Resim 11-7: Mantık için örnek hareket

```
PTP P1 Vel=100% PDAT1
PTP P2 CONT Vel =100% PDAT2
WAIT FOR IN 10 'door signal'
PTP P3 Vel=100% PDAT3
```

i CONT olmayan bir WAIT FOR satırının işlenmesi sırasında "Üzerinden atlama mümkün değil" uyarı mesajı görüntülenir.

Ön çalışmalı olarak programlanmış sinyale bağlı bekleme işlevleri, komut satırının önüne konulmuş noktanın üzerinden atlanmasına olanak tanır. Bununla birlikte ön çalışma göstergesinin konumu kesin değildir (standart değer: üç hareket komutu), dolayısıyla sinyalin kontrolü için kesin zaman belirsizdir (1). (>>> Resim 11-8). Bunun dışında sinyalin kontrolünden sonraki sinyal değişiklikleri algılanmaz!



Resim 11-8: Ön yürütmeli mantık için örnek hareket

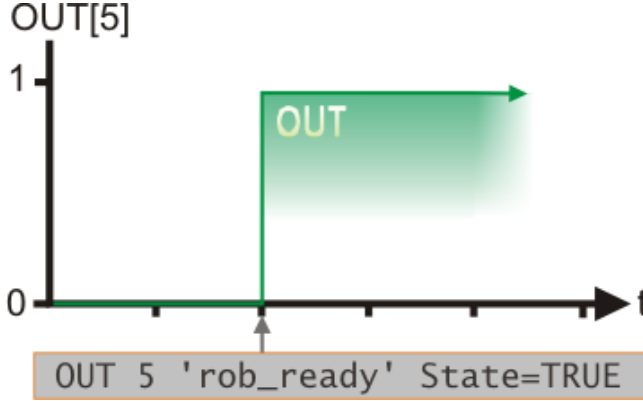
PTP P1 Vel=100% PDAT1
PTP P2 CONT Vel =100% PDAT2
WAIT FOR IN 10 'door signal' CONT
PTP P3 Vel=100% PDAT3

İzlenecek yön-
tem

1. İmleci, mantıksal deyimin arkasına eklenmesi gereken satıra getirin.
2. **Komutlar > Mantık > WAIT FOR** ya da **WAIT** menü sırasını takip edin.
3. Inline formunda parametreleri ayarlayın.
4. Deyimi **Komut Tamam** düğmesiyle kaydedin.

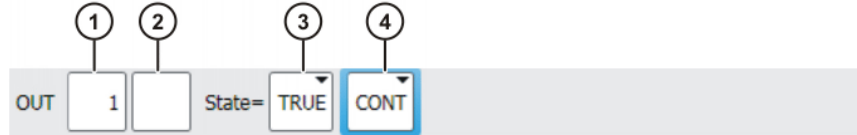
11.4 Basit Anahtarlama Fonksiyonlarını Programlama

Bir anahtarlama fonksiyonuyla, çevre birimlerine dijital bir sinyal gönderilebilir. Bu amaçla, daha önce arabirime göre tanımlanmış olan bir çıkış numarası kullanılır.



Resim 11-9: Statik anahtarlama

Sinyal statik olarak uygulanır yani çıkışa yeni bir değer uygulanana kadar eski değerini korur. Anahtarlama fonksiyonu program içerisinde bir inline formuyla gerçekleştirilir.



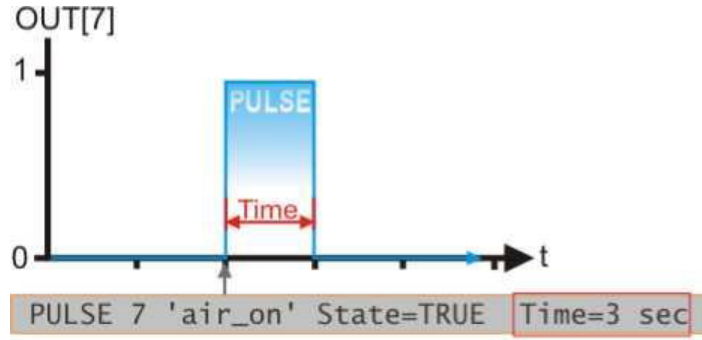
Resim 11-10: Inline-Formu OUT

Poz	Açıklama
1	Çıkışın numarası ■ 1 ... 4096
2	Çıkış için bir ad varsa burada görüntülenir. Sadece uzman kullanıcı grubu için: Uzun metin üzerine basılarak bir ad girilebilir. Ad serbestçe seçilebilir.
3	Çıkışın anahtarlandığı durum ■ TRUE ■ FALSE
4	■ CONT: Ön çalışmada işleme ■ [boş]: Ön çalışma stoplu işleme

i CONT girişi kullanıldığında sinyalin, ön çalışma esnasında verilmesine dikkat edilmelidir!

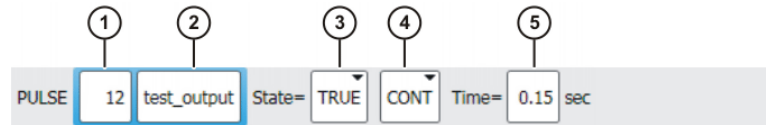
İmpulsu anahtarlama fonksiyonları

Basit anahtarlama fonksiyonunda olduğu gibi burada da bir çıkış olan değer değiştirilmektedir. Bununla birlikte impulslamada sinyal, tanımlı bir süre sonunda geri alınır.



Resim 11-11: Palslanmış seviye

Programlama yine bir inline formuyla yapılır ve tanımlı süresi olan bir impuls yazılır.



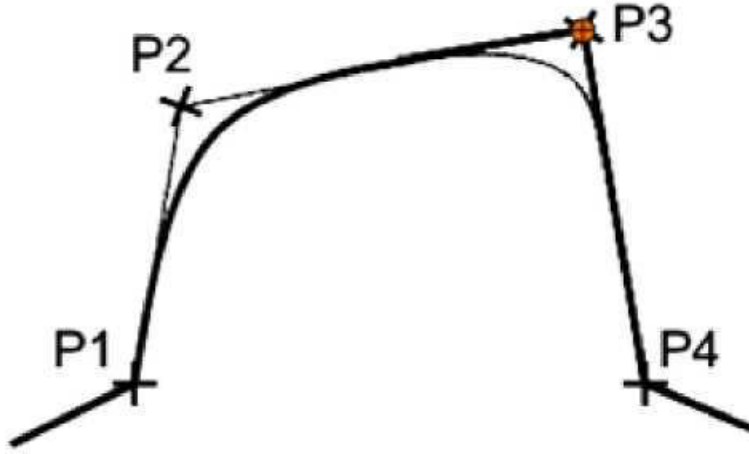
Resim 11-12: Inline-Formu PULSE

Poz	Tarif
1	Çıkışın numarası ■ 1 ... 4096
2	Çıkış için bir ad varsa burada görüntülenir. Sadece uzman kullanıcı grubu için: Uzun metin üzerine basılarak bir ad girilebilir. Ad serbestçe seçilebilir.
3	Çıkışın anahtarlandığı durum ■ TRUE: Seviye "High" ■ FALSE: Seviye "Low"
4	■ CONT: Ön yürütmeye işleme ■ [boş]: Ön yürütme stoplu işleme
5	İmpulsun uzunluğu ■ 0.10 ... 3.00 s

Anahtarlama fonksiyonlarında CONT etkileri

OUT İline formuna CONT girişi eklenmezse anahtarlama işleminde bir **ön çalışmayı durdur** zorlanır ve anahtarlama komutundan önceki noktada tam bir durma gerçekleşir. Çıkış ayarlandıktan sonra hareket sürdürülür.

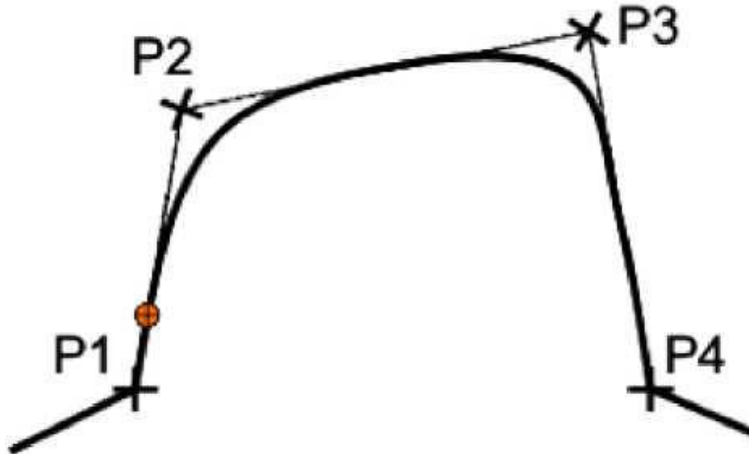
```
LIN P1 Vel=0.2 m/s CPDAT1  
LIN P2 CONT Vel=0.2 m/s CPDAT2  
LIN P3 CONT Vel=0.2 m/s CPDAT3  
OUT 5 'rob_ready' State=TRUE  
LIN P4 Vel=0.2 m/s CPDAT4
```



Resim 11-13: Ön çalışma durdurmalı anahtarlama için örnek hareket

CONT kaydının girişi, ön çalışma göstergesinin durdurulmamasına neden olur (ön çalışmayı durdur tetiklenmez). Böylece anahtarlama komutundan önce bir hareketin üzerinden atlanabilir. Sinyal, **ön çalışma** esnasında verilir.

```
LIN P1 Vel=0.2 m/s CPDAT1
LIN P2 CONT Vel=0.2 m/s CPDAT2
LIN P3 CONT Vel=0.2 m/s CPDAT3
OUT 5 'rob_ready' State=TRUE CONT
LIN P4 Vel=0.2 m/s CPDAT4
```



Resim 11-14: Ön çalışma sırasında kumanda için örnek hareket



Ön çalışma göstergesi için olan standart değer üç satırdır. Bununla birlikte ön çalışma değişebilir. Yani anahtarlama zamanının her zaman aynı olmaya bileceği hesaba katılmalıdır.

İzlenecek yöntem

1. İmleci, mantıksal deyimın arkasına eklenmesi gereken satıra getirin.
2. **Komutlar > Mantık > OUT > OUT** ya da **PULSE** menü sırasını takip edin.
3. İline formunda parametreleri ayarlayın.
4. Deyimi **Komut Tamam** düğmesiyle kaydedin.

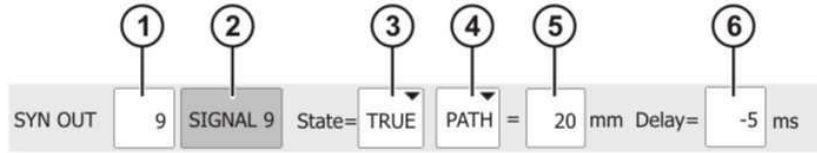
11.5 Yörünge Anahtarlama Fonksiyonlarını Programlama

Genel

Bir rota anahtarlama fonksiyonu, rota üzerinde belirli bir noktada robot hareketini durdurmadan bir çıkış ayarlayabilmek için kullanılabilir. Burada "statik" (SNY OUT) ile "dinamik" (SYN Pulse) anahtarlama arasında ayırım yapılır. Bir SYN OUT 5 anahtarlama ile bir SYN PULSE 5 anahtarlama aynı sinyale sahiptir. Sadece anahtarlama türü ve şekli değişmektedir.

Opsiyon PATH

Path opsiyonu ile, bir hareket cümlesinin hedef noktasına göre bir anahtarlama işlemi tetiklenebilir. Anahtarla işlemi yerel ve/veya zamansal olarak kaydırılabilir. Referans hareket cümlesi, bir LIN veya CIRC hareketi olabilir. PTP hareketi **olamaz**.



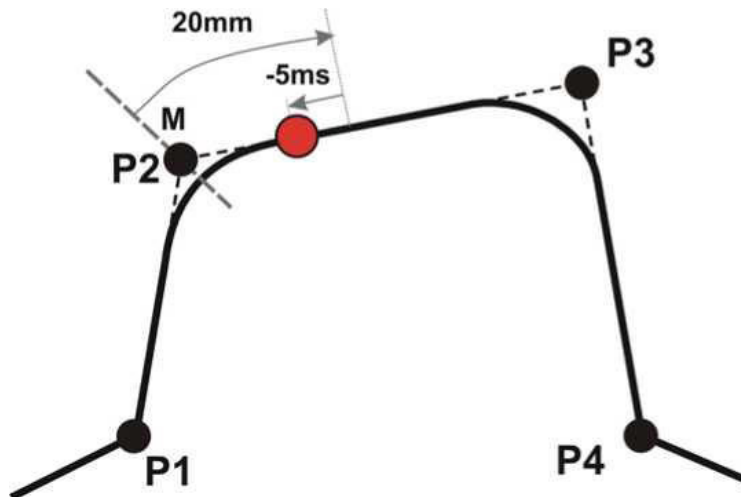
Resim 11-15: İinline formu SYN OUT, opsiyon PATH

Poz	Açıklama	Değer aralığı
1	Çıkışın numarası	1 ... 4096
2	Çıkış için bir ad varsa burada görüntülenir. Sadece uzman kullanıcı grubu için: Uzun metin tuşuna basarak bir ad girilebilir.	Serbest seçilebilir
3	Çıkışın anahtarlanacağı durumu belirtir	TRUE, FALSE
4	Anahtarlama yapılan nokta ■ PATH : Hareket cümlesinin hedef noktasına göre anahtarlama yapılır.	START, END Opsiyon PATH:
5	Anahtarlama işleminin yerel olarak kaydırılması Not : Yer bilgisi, daima hareket cümlesinin hedef noktasına göredir. Robotun hızı değiştiğinde dahi anahtarlama noktasının konumu değişmez.	-2000 ... +2000 mm
6	Anahtarlama aksiyonunu zamansal kaydırma Not :Zamansal kaydırma yerel kaydırmaya göredir.	-1000 ... +1000 ms

Path anahtarlama opsiyonu etkisi

Örnek program:

Bir freze, rotada anahtarlanmalıdır. P2'den 20 mm sonra hareket halindeyken yapı elemanının işlenmesine başlanması isteniyor. P2 noktasından 20 mm sonra (path = 20) frezenin tam devir hızına ulaşabilmesi için 5 ms önce (delay = - 5 ms) devreye alınmalıdır.



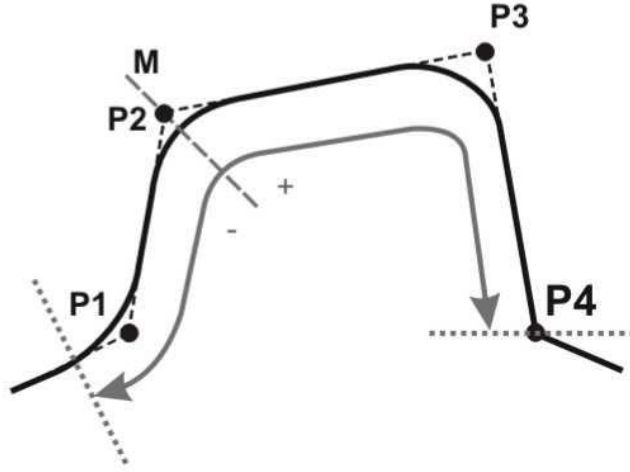
Resim 11-16: SYN OUT Path anahtarlama sınırı

```

LIN P1 VEL=0.3m/s CPDAT1
;P2'ye göre anahtarlama fonksiyonu
SYN OUT 9 'SIGNAL 9' Status= True Path=20 Delay=-5ms
LIN P2 CONT VEL=0.3m/s CPDAT2
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4

```

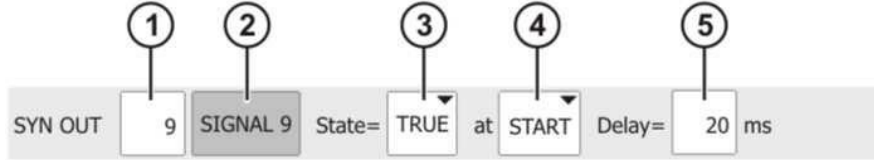
Anahtarlama sınırları



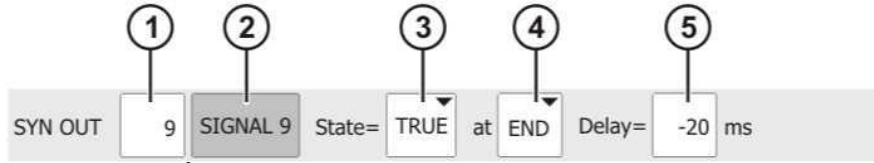
Resim 11-17: SYN OUT PATH anahtarlama sınırları

**Opsiyon START/
END**

Bir anahtarlama işlemi, bir hareket cümlesinin başlangıç veya hedef noktasına göre tetiklenebilir. Anahtarlama işlemi **zamansal** olarak kaydırılabilir. Referans hareket cümlesi, bir LIN, CIRC veya PTP hareketi olabilir.



Resim 11-18: İline formu SYN OUT, opsiyon START



Resim 11-19: İline formu SYN OUT, opsiyon END

Poz	Açıklama	Değer aralığı
1	Çıkışın numarası	1 ... 4096
2	Çıkış için bir ad varsa burada görüntülenir. Sadece uzman kullanıcı grubu için: Uzun metin tuşuna basarak bir ad girilebilir.	Serbest seçilebilir
3	Çıkışın anahtarlanaacağı durumu belirtir	TRUE, FALSE
4	Anahtarlama yapılan nokta ■ START : Hareket cümlesinin start noktasına göre anahtarlama yapılır. ■ END : Hareket cümlesinin hedef noktasına göre anahtarlama yapılır.	START, END Opsiyon PATH:
5	Anahtarlama aksiyonunu zamansal kaydırma Not : Zaman girişi mutlaklıdır. Bu nedenle anahtarlama noktasının konumu, robotun hızına göre değişir.	-1000 ... +1000 ms

**İzlenecek yön-
tem**

1. İmleci, mantıksal deyimim arkasına eklenmesi gereken satıra getirin.
2. **Komutlar > Mantık > OUT > SYN OUT** ya da **SYN PULSE** menü sırasını takip edin.
3. İline formunda parametreleri ayarlayın.
4. Deyimi **Komut Tamam** düğmesiyle kaydedin.

12 Uzman Düzeyine Giriş

12.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Uzman düzlemini kullanma

12.2 Uzman Düzlemini Kullanma

Açıklama

Robot kumanda sistemi, farklı fonksiyonları olan farklı **kullanıcı grupları** sunmaktadır. Aşağıdaki **kullanıcı grupları** seçilebilir:


- **Operatör**
Operatör için kullanıcı grubudur. Bu, standart kullanıcı grubudur.
- **Kullanıcı**
Operatör için kullanıcı grubudur. ('Operatör' ve 'Kullanıcı' kullanıcı grubu, varsayılan olarak aynı hedef grup için oluşturulmuştur.)
- **Uzmanlar**
Programcı için kullanıcı grubudur. Bu kullanıcı grubu, bir parola ile korunmaktadır.
- **Yönetici**
Fonksiyonlar, uzman kullanıcı grubundaki gibi. Ayrıca robot kontrol sisteminde eklentiler (Plug-In) entegre edilebilir. Bu kullanıcı grubu, bir parola ile korunmaktadır.
- **Güvenlik onarıcısı**
Bu kullanıcı robotun mevcut bir güvenlik konfigürasyonunu bir etkinleştirme koduyla etkinleştirebilir. KUKA.SafeOperation veya KUKA.SafeRangeMonitoring gibi bir Safe opsiyonu kullanılmadığında, güvenlikle ilgili onarıcı kullanıcı daha fazla hakka sahip olur. Örneğin standart güvenlik işlevlerinin konfigürasyonu yetkisine sahip olur.
- **Güvenlik işletme sorumlusu**
Bu kullanıcı grubu yalnızca KUKA.SafeOperation veya KUKA.SafeRangeMonitoring kullanıldığında geçerlidir. Kullanıcı grubu, bir parola ile korunmaktadır.

Uzman kullanıcı grubunun genişletilmiş fonksiyonları:

- Parola korumalı (varsayılan parola: kuka)
- KRL ile editörde programlama mümkün
- Modüller için ayrıntılı görünüm kullanılabilir
- DEF satırı görüntüleme/gizleme
- FOLD açma ve kapatma
- Program içerisinde ayrıntı görünümü görüntüleme
- Program oluşturmada hazır tanımlı template (şablonlar) arasından seçim yapılabilir
- Uzman kullanıcı grubu aşağıdaki durumlarda otomatik olarak terk edilir,
 - AUT veya AUT EXT işletim türüne geçildiğinde
 - Belirli bir süre için (300 s) kullanıcı arayüzünde bir işlem yapılmazsa



Resim 12-1: 'Düzenle' menüsü

 Uzman olarak **Düzenle** menüsünde tüm fonksiyonlar mevcuttur.

Fonksiyonlar

Şablon yardımıyla program oluşturma

- **Cell:** Mevcut Cell programı sadece yenisiyle değiştirilebilir veya Cell programı silinmişse yenisi oluşturulabilir.
- **Expert:** SRC ve DAT dosyasından oluşan ve sadece program başının ve sonunun mevcut olduğu bir modüldür.
- **Expert Submit:** Program başından ve sonundan oluşan ek Submit dosyasıdır (SUB).
- **Function:** Fonksiyon oluşturma SRC dosyası. İçerisinde sadece fonksiyon başı bir adet BOOL değişkeniyle oluşturulmaktadır. Fonksiyonun sonu var, fakat iade kısmının henüz programlanması gerekir.
- **Modül:** SRC ve DAT dosyasından oluşan ve program başının, sonunun ve temel yapının (INI ve 2x PTP HOME) yer aldığı modüldür.
- **Submit:** Program başından, sonundan ve temel yapıdan (DECLARATION, INI, LOOP/ENDLOOP) oluşan ek Submit dosyasıdır (SUB).

Filtre, programların detay listesinde nasıl görüntüleneceğini belirler. Aşağıdaki **filtreler** kullanıma sunulur:

- Program seçimini kaldırın veya kapatın.
- **Düzenle > Filtre** butonuna ve sonra da **Ayrıntı** veya **Modül** üzerine basın
- **Ayrıntı**

Programlar SRC ve DAT dosyaları olarak gösterilir. (Default ayar)

- **Modüller**

Programlar modüller olarak görüntülenir.



Resim 12-2: 'Düzenle' menüsü

FOLD açma/kapatma

- **FOLD**'lar kullanıcı için daima kapalıdır ve Uzman kullanıcı (Expert) olarak açılabilir.
- Uzman kullanıcı kendi **FOLD**'larını da programlayabilir.
- Bir Fold için için geçerli olan **sözdizimi** şu şekildedir:
;FOLD Name
Talimatlar
;ENDFOLD <Name>
ENDFOLD satırları, bu satıra foldun adı da yazılırsa daha kolay eşleştirebilir. Foldlar iç içe düzenlenebilir.

DEF satırı görüntüleme/gizleme

- **DEF satırı** varsayılan olarak gizlenmiştir. Bir programda açıklamalar yapabilmek için, **DEF satırının** görüntüleniyor olması gerekir.
- **DEF satırı** açılmış ve seçilmiş programlar için ayrı ayrı görüntülenir ve gizlenir. Ayrıntı görünümü açıksa, **DEF satırı** görünürdür ve ayrıca görüntülenmesi gerekmez.

Uzman düzlemi için yöntemi etkinleştirin ve hatayı giderin

Uzman düzlemi etkinleştirme

1. Ana menüde **Konfigürasyon > Kullanıcı grubu** seçimini yapın.
2. **Uzman** kullanıcı olarak oturum açın: **Oturum aç** düğmesine basın. **Uzman** kullanıcı grubunu işaretleyin ve **Oturum aç** ile onaylayın.
3. İstenen parolayı girin (varsayılan parola: kuka) ve **Oturum aç** ile onaylayın.

Programda hata giderme

1. Hatalı modülü navigatörde seçinEditör



Resim 12-3: Hatalı program

2. **Hata listesi** menüsünü seçin
3. Hata görüntülemesi (programadı.ERR) açılır
4. Hatayı seçin, ayrıntılı tarifi hata gösterimi penceresinde görüntülenir
5. Hata gösterimi penceresinde **Görüntüle** düğmesine basın ve hatalı programa geçin
6. Hatayı giderin
7. Editörü terk edin ve kaydedin

13 Schleifen, Bedingte Anweisungen und Fallunterscheidungen

13.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Döngüler
- Koşullu talimatlar
- Durum analizi

13.2 Program Akışının Kontrolü

Program akışının kontrolü

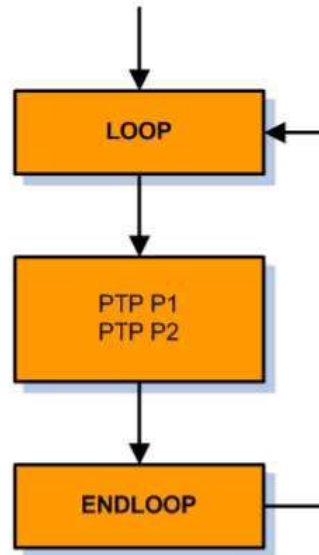
Salt hareket komutları ve iletişim komutlarının (anahtarlama ve bekleme fonksiyonları) yanında robot programlarında, program akış kontrolünü sağlayan çok sayıda yordam da yer almaktadır. Bunlara şunlar dahildir:

- **Döngüler** | Döngüler kontrol yapılarıdır. İptal koşulu sağlanana kadar bir talimat bloğunu tekrarlarlar.
 - Sonsuz döngüler
 - Sayaç döngüleri
 - Dışlayıcı ve dışlayıcı olmayan döngüler
- **Dallanmalar** | Dallanmalar kullanılarak, program bölümlerinin ancak belirli bir koşulda uygulanması sağlanabilir.
 - Koşullu dallanmalar
 - Çoklu dallanmalar

13.3 Taşlama

Sonsuz döngü

Sonsuz bir döngü, talimat bloğunu sonsuz kez yineler. Yine de bir iptal uygulanarak (EXIT fonksiyonuyla) döngüden çıkmak mümkündür.



Resim 13-1: Program akış şeması: Sonsuz döngü

Bir LOOP talimatına ilişkin örnekler:

- **EXIT** yok
P1 ve P2 için hareket komutları sürekli olarak uygulanır.

LOOP

```
PTP P1 Vel=100% PDAT1  
PTP P2 Vel=100% PDAT2
```

ENDLOOP

- **EXIT** var

P1 ve P2 için hareket komutları, giriş 30 TRUE değerine anahtarlanıncaya kadar yürütülür.

LOOP

```
PTP P1 Vel=100% PDAT1  
PTP P2 Vel=100% PDAT2  
IF $IN[30]==TRUE THEN
```

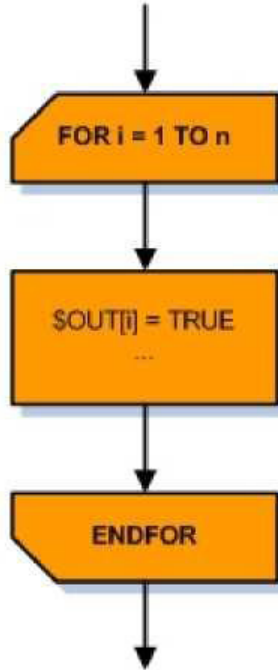
EXIT

```
ENDIF
```

ENDLOOP

Sayaç döngüsü

Sayaç döngüsüyle (FOR döngüsü), deyimler tanımlı bir sayı kadar tekrarlanabilir. Komutların yürütülme sayısı bir sayaç değişkeni yardımıyla kontrol edilir.



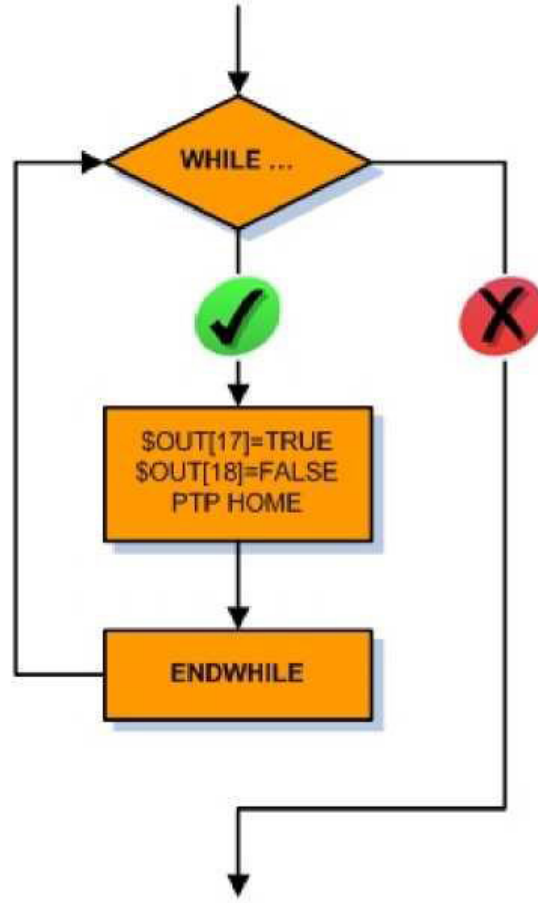
Resim 13-2: Program akışı planı: FOR döngüsü

FOR döngüsü için örnek: 1 ila 5 numaralı çıkışlar peş peşe TRUE değerine anahtarlanır. Döngü içerisinde geçişleri saymak için tam sayı (Integer) değişkeni "i" kullanılır.

```
INT i  
...  
FOR i=1 TO 5  
$OUT[i] = TRUE  
ENDFOR
```

Dışlayıcı döngü

Bir WHILE döngüsü dışlayıcı veya peşin kontrollü bir döngüdür ve döngünün iptal edilme koşulu, döngünün içindeki deyimler yürütülmeden önce kontrol edilir.



Resim 13-3: Program akış planı WHILE

WHILE döngüsü için örnek: Çıkış 17 TRUE, çıkış 18 FALSE değerine anah- tarlanır ve robot Home pozisyonuna götürülür. Fakat bunun gerçekleşmesi için döngünün başındaki koşulun (giriş 22 TRUE değerinde) sağlanmış olması gerekir.

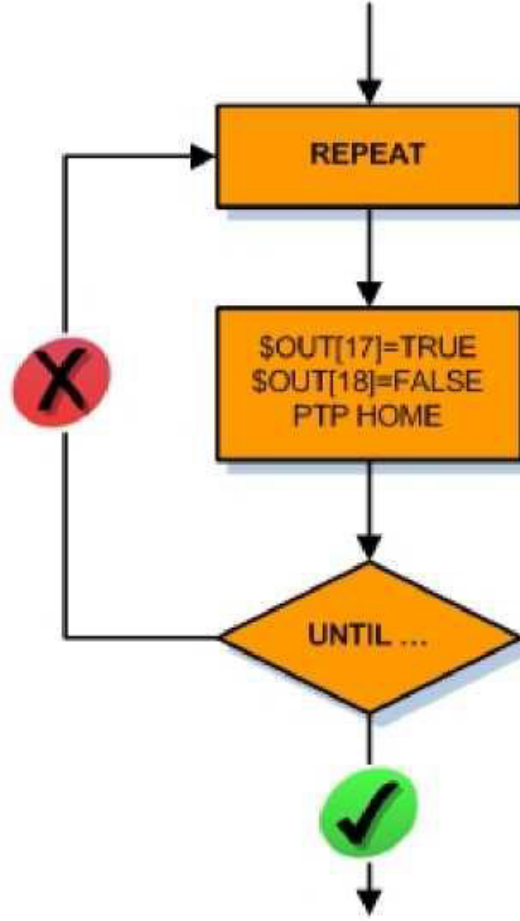
```

WHILE $IN[22]==TRUE
  $OUT[17]=TRUE
  $OUT[18]=FALSE
  PTP HOME
ENDWHILE

```

Dışlayıcı ol- mayan döngü

Bir REPEAT döngüsü, dışlayıcı olmayan veya sonradan kontrollü bir döngüdür ve döngünün iptal edilme koşulu, döngünün içindeki talimat kısmı bir kez yürütüldük- ten sonra kontrol edilir.



Resim 13-4: REPEAT program akış planı

REPEAT döngüsü için örnek: Çıkış 17 TRUE, çıkış 18 FALSE durumuna getirilir ve robot Home konumuna hareket ettirilir. Bu işlemden sonra iptal edilme koşulu kontrol edilir.

```

REPEAT
  $OUT[17]=TRUE
  $OUT[18]=FALSE
  PTP HOME
UNTIL $IN[22]==TRUE

```

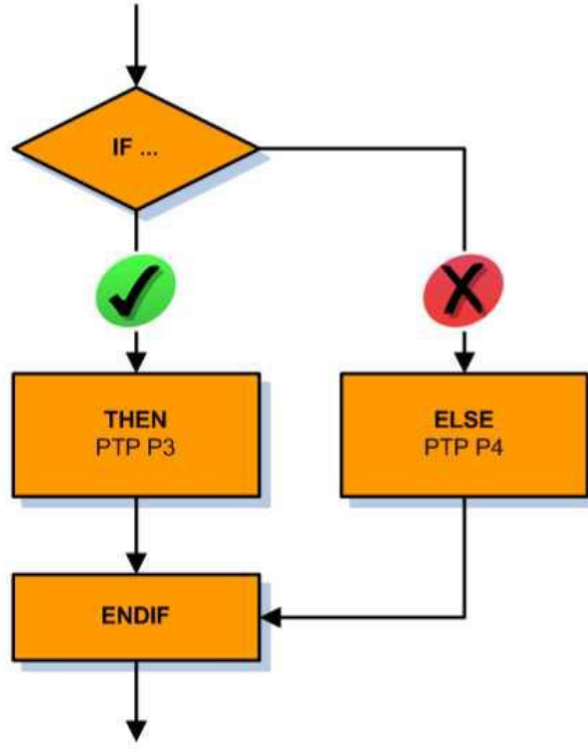
13.4 Koşullu Talimatlar ve Durum Analizleri

Koşullu dallanma

Koşullu bir dallanma (IF sorgusu) bir koşul ve iki talimat kısmından oluşur. Koşul sağlandıysa ilk talimat yürütülebilir. Koşul sağlanamadıysa ikinci talimat yürütülür.

Bir IF sorgusu için alternatifler de mevcuttur:

- İkinci talimat kısmı iptal edilebilir: ELSE olmadan IF sorgusu Bu şekilde, koşul sağlanmadığında hemen dallanma sonrası programın devamı yürütülür.
- Birden çok IF komutu iç içe düzenlenebilir (*Çoklu dallanma*): Koşullardan biri sağlanıncaya kadar sorgular sırayla işlenir.



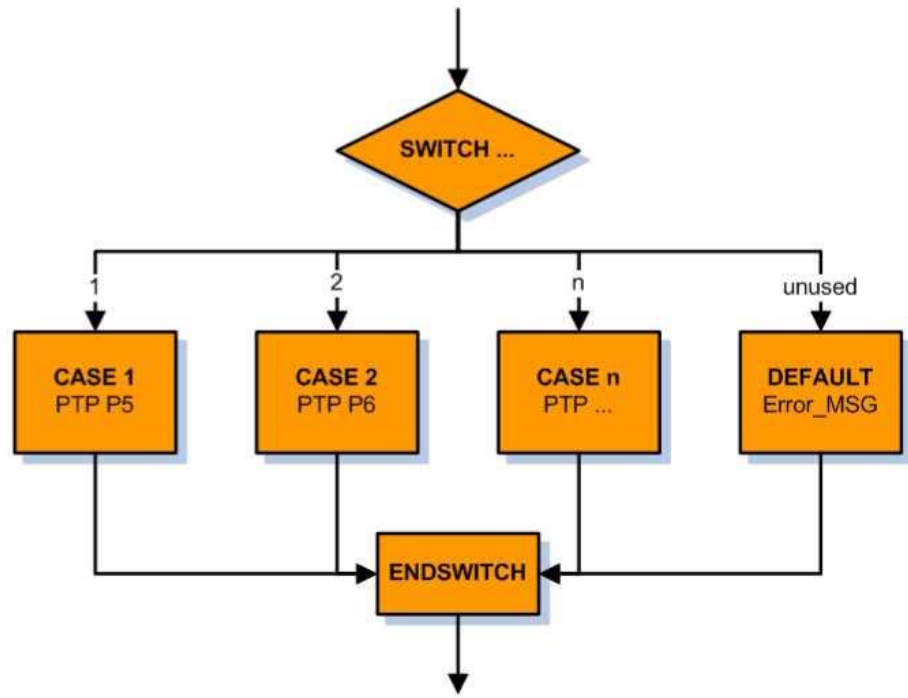
Resim 13-5: Program akış şeması: IF dallanması

IF sorgusu için örnek: Koşul sağlandığında (giriş 30 TRUE olmak zorunda) robot, P3 noktasına diğer durumda ise P4 noktasına hareket eder.

```
...  
IF $IN[30]==TRUE THEN  
    PTP P3  
ELSE  
    PTP P4  
ENDIF
```

Dağıtıcı

Bir SWITCH dallanması bir *dağıtıcı* veya *çoklu dallanmadır*. Burada önce bir ifade değerlendirilir. Sonra da ifadenin değeri durum seçeneklerinden (CASE) birinin değeriyle karşılaştırılır. Eşleme sağlanırsa ilgili durumun talimatları uygulanır.



Resim 13-6: Program akış şeması: Dağıtıcı SWITCH - CASE

"Durum" adını taşıyan tam sayı değişkeninde (Integer) önce değere bakılır. Değişkenin değeri 1 ise durum 1 (CASE 1) uygulanır: Robot P5 noktasına hareket eder. Değişkenin değeri 2 ise durum 2 (CASE 2) uygulanır: Robot P6 noktasına hareket eder. Değişkenin değeri hiçbir durumda yoksa (bu örnekte 1 ve 2 dışındaki her değer), DEFAULT dalındaki komutlar uygulanır: bir hata mesajı.

```

INT status
...
SWITCH status
  CASE 1
    PTP P5
  CASE 2
    PTP P6
  ...
  DEFAULT
    ERROR_MSG()
ENDSWITCH
  
```

14 Alt Programlar ve Fonksiyonlar

14.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Yerel alt programlar
- Genel alt programlar
- Parametreleri alt programlara aktarma

14.2 Lokal Alt Programlarıyla Çalışma

Lokal alt programları tanımlama

- Lokal alt programlar ana programdan sonra yer alır ve deyimlerle işaretlidir: DEF Ad alt program() ve END

```
DEF MY_PROG ( )  
; bu ana programdır  
...  
END
```

```
DEF LOCAL_PROG1( )  
; yerel alt program 1  
...  
END
```

```
DEF LOCAL_PROG2( )  
; yerel alt program 2  
...  
END
```

```
DEF LOCAL_PROG3( )  
; yerel alt program 3  
...  
END
```

- Bir SRC dosyası 255 lokal alt programdan oluşabilir
- Lokal alt programlar birkaç kez çağrılabilir
- Lokal program adı için parantezler gereklidir

Lokal alt programlarıyla çalışırken mevcut ilişkiler

- Lokal bir alt program yürütüldükten sonra, alt programın çağrıldığı satırdan sonra satıra bir geri atlama gerçekleşir

```
DEF MY_PROG ( )  
; bu ana programdır  
...  
LOCAL_PROG1( )  
...  
END
```

```
DEF LOCAL_PROG1( )  
LOCAL_PROG2( )  
END  
DEF LOCAL_PROG2( )  
END
```

- En çok 20 alt programı iç içe yerleştirmek mümkündür
- Nokta koordinatları kendilerine ait DAT listesinde kaydedilir ve komple doysa süresinde kullanılabilir

```
DEF MY_PROG( )
; bu ana programdır
...
PTP P1 Vel=100% PDAT1
END
```

```
DEF LOCAL_PROG1( )
...
; burası ana programla aynı pozisyondur
PTP P1 Vel=100% PDAT1
END
```

```
ENDDAT MY PROG( )
...
DECL E6POS XP1={X 100, Z 200, Z 300 . . E6 0.0}
...
ENDDAT
```

- RETURN komutuyla bir alt program sonlandırılabilir ve alt programı çağırılmış olan program parçasına dönülmesini sağlar

```
DEF MY_PROG( )
; bu ana programdır
...
LOCAL_PROG1( )
...
END
DEF LOCAL_PROG1( )
...
IF $IN[12]==FALSE THEN
RETURN ; Ana programa geri atlayış
ENDIF
...
END
```

Lokal alt programlar oluşturulurken izlenecek yöntem

1. Uzman kullanıcı grubu
2. DEF satırının görüntülenmesini sağlama
3. SCR dosyasını editörde açma

```
DEF MY_PROG( )
...
END
```

4. İmleçle END satırının **altına** atlayın
5. DEF, program adı ve parantezlerle yeni program başı verin

```
DEF MY_PROG( )
...
END
DEF PICK_PART( )
```

6. Bir END komutuyla yeni alt programı kapatın

```
DEF MY_PROG( )
...
END
DEF PICK_PART( )
END
```

7. Enter tuşuna basıldıktan sonra ana ve alt program arasına bir enine çubuk çekilir

```
DEF MY_PROG( )  
...  
END
```

```
DEF PICK_PART( )  
END
```

8. Ana ve alt program düzenlenmeye devam edilebilir

9. Programı kapatma ve kaydetme

14.3 Global Alt Programlarıyla Çalışma

Global alt programları tanımlama

- Global alt programların kendi SRC ve DAT dosyaları vardır

```
DEF GLOBAL1( )  
...  
END
```

```
DEF GLOBAL2( )  
...  
END
```

Lokal altprogramlarıyla çalışırken mevcut ilişkiler

- Global alt programlar birkaç kez çağrılabilir
- Lokal bir alt program yürütüldükten sonra, alt programın çağrıldığı satırdan sonra satıra bir geri atlama gerçekleşir

```
DEF GLOBAL1( )  
...  
GLOBAL2( )  
...  
END
```

```
DEF GLOBAL2( )  
...  
GLOBAL3( )  
...  
END
```

```
DEF GLOBAL3( )  
...  
END
```

- En çok 20 alt programı iç içe yerleştirmek mümkündür
- Nokta koordinatları kendilerine ait DAT listesinde kaydedilir ve sadece ilgili programda kullanılabilir

```
DEF GLOBAL1( )  
...  
PTP P1 Vel=100% PDAT1  
END
```

```
DEFDAT GLOBAL1( )
DECL E6POS XP1={X 100, Z 200, Z 300 . . E6 0.0}
ENDDAT
```

Global2 () içerisinde P1 için farklı koordinatlar

```
DEF GLOBAL2( )
...
PTP P1 Vel=100% PDAT1
END
```

```
DEFDAT GLOBAL2( )
DECL E6POS XP1={X 800, Z 775, Z 999 . . . E6 0.0}
ENDDAT
```

- RETURN komutuyla bir alt program sonlandırılabilir ve alt programı çağırılmış olan program parçasına dönülmesini sağlar

```
DEF GLOBAL1( )
...
GLOBAL2( )
...
END
```

```
DEF GLOBAL2( )
...
IF $IN[12]==FALSE THEN
RETURN ; GLOBAL1( ) içerisinde geri atlama
ENDIF
...
END
```

**Global alt programlarla programlama-
da izlenecek yöntem**

1. Uzman kullanıcı grubu
2. yeni program oluşturma

```
DEF MY_PROG( )
...
END
```

3. ikinci programı yeni oluşturma

```
DEF PICK_PART( )
...
END
```

4. MY_PROG programının SCR dosyasını editörde açın
5. Alt program çağırısını program adı ve parantezlerle programlayın

```
DEF MY_PROG( )
...
END
DEF PICK_PART( )
...
END
```

6. Programı kapatma ve kaydetme

14.4 Parametreleri Alt Programlara Aktarma

Parametreleri aktarma Parametrelerin bir alt programa aktarılması, parantez içerisinde program açıldığında gerçekleşir:

```
DEF MY_PROG( )
...
CALC ( K, L)
...
END
```

Parametreleri aktarmak için iki seçenek vardır:

Parametre	Açıklama
IN parametresi	Değişkenin değeri ana program içerisinde değişmeksizin aynıdır'. Bu aktarma türüne "Call by Value" de denir.
OUT parametresi	Alt program değeri okur, değiştirir ve yeni değeri ana programa geri yazar. Bu aktarma türüne "Call by Reference"

```
DEF CALC (R:IN, S:OUT)
...
END
```



Birden fazla parametre aktarıldığında, sıralamaya dikkat edilmelidir!

Örnek fonksiyon Bir fonksiyon bir alt program gibi açılır ancak asla tek başına duramaz. Değer, her zaman aynı veri tipinde bir değişkene atanmalıdır.

```
DEF MY_PROG( )
DECL REAL result, value
...
result = CALC(value)
...
END
```

Bir fonksiyon, belirli bir değeri RETURN ile ana programa geri gönderen ve bir veri tipine sahip olan bir alt programdır.

```
DEFECT REAL CALC(num:IN) ;Fonksiyonun veri tipi REAL'dir
DECL REAL return_value, num
...
RETURN(return_value) ;Değeri ana programa geri verir
ENDFCT
```

KUKA standart fonksiyonlar

Matematiksel fonksiyonlar:

Açıklama	KRL fonksiyonu
Mutlak değer	ABS(x)
Kök	SQRT(x)
Sinüs	SIN(x)
Kosinüs	COS(x)
Tanjant	TAN(x)
Arkkosinüs	ACOS(x)
Arktanjan	ATAN2(y,x)

15 Değişkenler ve Kararlıtırmalar

15.1 Genel Bakış

Bu eğitim ünitesinde Őu konular anlatılmaktadır:

- KRL'de veri yönetimi
- Basit veri tipleriyle çalıřma
- Alanlar
- Yapılar
- Numaralandırma tipi

15.2 KRL'de Veri Yönetimi

KRL ile deęiř- kenlerle çalıřma

Deęiřkenlere iliřkin genel bilgiler

- KRL ile yapılan robot programlamasında bir deęiřken en genel anlamıyla bir robot iř sürecinin seyri sırasında ortaya çıkan hesap büyüklükleri ("deęerler") için bir kaptır.
- Bir deęiřken, bilgisayarın belleęinde kendisine atanmıř belirli bir adrese sahiptir.
- Bir deęiřken, KUKA anahtar sözcüęü olmayan bir adla iřaretlenir.
- Her deęiřken belirli bir veri tipine baęlıdır.
- Veri tipinin bildirilmesi deęiřkeni kullanmadan önce gereklidir.
- KRL dilinde yerel (lokal) ve global deęiřkenler arasında ayırım yapılmaktadır.
- KRL'de deęiřkenlerin kullanım ömrü
- Kullanım ömrü, deęiřkenin bellek yerini kendisi için ayırdıęı süredir.
- Yürütüm deęiřkenleri program veya fonksiyon terk edildięinde kendilerine ayrılmıř bellek yerini iade ederler.
- Veri listesindeki deęiřkenler güncel deęerlerini kalıcı olarak bellek yerinde korurlar.

KRL içinde deęiřkenlerin geçerlilięi

- Lokal olarak bildirilmıř deęiřkenler sadece bildirildikleri program içerisinde kullanılabilir ve görünürdüler.
- Global deęiřkenler merkezi (global) bir veri listesinde tanımlanmıřtır.
- Global deęiřkenler lokal bir dosyada da oluşturulup **global** anahtar sözcüęü eklenebilir.

KRL ile veri tipleri

- Bir veri tipi, nesnelerin bir kümede toplanmasını belirtir.
- Önceden tanımlı standart veri tipleri
- Kendi tanımladıęınız standart veri tipleri
- Önceden tanımlı KUKA veri tipleri

KRL ile deęiř- ken kullanımı

Adlandırma kuralları

- KRL dilinde adlar en çok 24 karakter uzunluęa sahip olabilir.
- KRL içindeki adlar, harfler (A-Z), rakamlar (09) ve "_" ve "\$" özel karakterlerini içerebilir.
- KRL içinde adların rakamlarla başlaması yasaktır.
- KRL içindeki adlar anahtar sözcük olamaz.
- Büyük ve küçük harf ayırımı yapılmaz.

KRL ile veri tipleri

- **Önceden tanımlı standart veri tipleri**

Basit veri tipleri	Tam sayı	Kayan noktalı sayı (gerçek sayı)	Mantıksal değerler	Tek karakter
Anahtar kelime	INT	REAL	BOOL	CHAR
Değer aralığı	$-2^{31} \dots (2^{31}-1)$	$\pm 1.1 \cdot 10^{-38} \dots \pm 3.4 \cdot 10^{38}$	TRUE / FALSE	ASCII karakter seti
Örnekler	-199 ya da 56	-0.0000123 ya da 3.1415	TRUE veya FALSE	"A" veya "q" veya "7"

- **Alanlar / Arraylar**

Voltage 10] = 12.75

Voltage 11] = 15.59

- Aynı veri tipi olan birden çok değişkeni indeks yardımıyla kaydetme
- İklendirme veya değer değiştirme indeks yardımıyla gerçekleştirilir
- Azami alan büyüklüğü veri tipinin bellek ihtiyacına bağlıdır

- **Numaralandırma tipi**

color = #red

- Numaralama veri tipinin tüm değerleri oluşturma aşamasında adları- la(A-çıkMetin) tanımlanır
- Sistem bir sıra belirler
- Azami öğe sayısı bellek yerine bağlıdır

- **Birleşik veri tipi / yapı**

Date = {day 14, month 12, year 1996}

- Farklı veri tiplerinin komponentlerinden birleştirilmiş veri tipi
- komponentler basit veri tiplerinden veya yapılardan oluşabilir
- Münferit komponentlere erişmek mümkündür
- **Kullanım ömrü / Geçerlilik**
 - **SRC dosyasında oluşturulan değişkenlere yürütüm değişkenleri denir**
 - ve her zaman için görüntülenmeyebilirler
 - bunlar sadece bildirildikleri program bölümünde geçerlidirler
 - en son program satırına (END satırı) vardıklarında kendilerine ayrılmış bellek yerlerini serbest bırakırlar
- **Lokal DAT dosyasındaki değişkenler**
 - İlgili SRC dosyasının programı akışı sırasında daima görüntülenebilirler
 - Komple SRC dosyasında ve tüm lokal alt programlarda kullanılabilirler
 - global değişken olarak da oluşturulabilirler
 - Güncel değeri DAT dosyasında alırlar ve yeniden çağrıldıklarında kayıtlı olan değerle devam ederler
- **ŞCONFIG.DAT sistem dosyasındaki değişkenler**
 - tüm programlarda kullanılabilirler (global),
 - bir program etkin olmasa dahi daima görüntülenebilirler,
 - Güncel değerlerini ŞCONFIG.DAT dosyasından alırlar.

Değişkenlerin çift bildirimi

- Çift bildirim aynı karakter dizileri (adlar) kullanıldığında ortaya çıkar.
- Farklı SRC veya DAT dosyalarında aynı ad kullanıldığında çift bildirim söz konusu **değildir**.
- Aynı SRC ve DAT dosyasında çift bildirim yapılması yasaktır ve bir hata mesajı üretilmesine neden olurlar.
- SRC veya DAT ve \$CONFIG.DAT dosyasında çift bildirimlere izin verilmektedir:
- Değişkenin bildirimini yapıldığı program yordamı sonra erdiğinde sadece lokal değeri değiştirilir \$CONFIG.DAT dosyasındaki değeri değiştirilmez.
- "Yabancı" bir program yordamı yürütüldüğünde sadece \$CONFIG.DAT dosyasındaki değer alınır ve değiştirilir.

KUKA sistem verileri

- KUKA sistem veri tipleri şu şekildedir:
 - Numaralama veri tipi, örneğin işletim türü (mode_op)
 - Yapı, tarih/saat (date)
- Sistem bilgileri KUKA sistem değişkenlerinden alınmaktadır. Bunlar
 - güncel sistem bilgilerini okurlar,
 - güncel sistem yapılandırmalarını değiştirirler,
 - Önceden tanımlıdır ve "\$" karakteriyle başlarlar
 - \$DATE (güncel saat ve tarih)
 - \$POS_ACT (güncel robot pozisyonu)
 - ...

15.3 Basit Veri Tipleriyle Çalışma

Aşağıda değişkenlerin oluşturulması, ilklendirilmesi ve değiştirilmesi açıklanmaktadır. Burada sadece basit veri tipleri kullanılmaktadır.

KRL ile basit veri tipleri

- Tam sayılar (INT)
- Gerçek sayılar (REAL)
- Mantıksal değerler (BOOL)
- Tek karakter (CHAR)

15.3.1 Değişkenlerin Bildirimi

Değişken oluşturma

Değişken bildirme

- Bildirim daima kullanımda önce yapılmak zorundadır
- her değişkene bir veri tipinin eşleştirilmesi gerekir
- Ad verilirken ad kurallarına uyulmak zorundadır
- bildirim için olan anahtar sözcük **DECL**
- DECL anahtar sözcüğü dört adet basit veri tipinde kullanılmayabilir
- Değer atamaları ön yürütme göstergesiyle gerçekleşir
- Değişken bildirimi farklı farklı gerçekleştirilebilir, çünkü buradan ilgili değişkenlerin kullanım ömrü ve geçerliliği ortaya çıkar
 - SRC dosyasında bildirim
 - Lokal DAT dosyasında bildirim
 - \$CONFIG.DAT dosyasında bildirim
 - Lokal DAT dosyasında "global" anahtar sözcüğü ile bildirim

- Sabitler oluşturma
 - Sabitler **CONST** anahtar sözcüğü ile oluşturulmaktadır
 - Sabitler sadece veri listelerinde oluşturulabilir

SRC dosyasında program yapısı

Değişken bildirim prensibi

- Bildirim bölümünde değişkenler bildirilmek zorundadır.
- İklendirme bölümü ilk değer atamasıyla, fakat genelde "INI" satırıyla başlar
- Deyim bölümünde değerler atanır veya değiştirilir

```
DEF main( )
; Bildirim bölümü
...
; İklendirme bölümü
INI
...
; Deyimler bölümü
PTP HOME Vel=100% DEFAULT
...
END
```

- Standart görünümü değiştirme
- DEF satırı sadece Uzman kullanıcı olarak görüntülenebilir
- Bu işlem, modüllerde "INI" satırından önce bildirim bölümüne ulaşmak için gereklidir
- DEF ve END satırlarını görebilmek için; alt programlarda değişken aktarımında da önemlidir

Değişken bildirimini planlama

- Kullanım ömrünü tespit etme
 - **SRC-File**: Yürütüm değişkeni, program yordamının sonunda "ölür"
 - **DAT** dosyası: Değişken, sona ermeden sonra korunur
- Geçerliliği/Kullanılabilirliği belirleme
 - **SRC** dosyasında lokal: sadece bildirim yapıldığı program yordamında kullanılabilir. Dolayısıyla değişken sadece lokal DEF ve END satırları arasında kullanılabilir (ana program **veya** lokal alt program).
 - **DAT** dosyasında lokal: komple programda geçerli, yani tüm lokal alt programlarda da.
 - **\$CONFIG.DAT**: global kullanılabilir, yani tüm program yordamlarında bir okuma ve yazma erişimi mümkündür
 - Global değişken olarak **DAT** dosyasında lokal: global kullanılabilir, tüm program yordamlarında şu koşullarda okuma ve yazma erişimi mümkündür: DAT dosyasına PUBLIC anahtar sözcüğü konur ve bildirimde şu anahtar sözcük eklenirse: GLOBAL.
- Veri tipini belirleme
 - **BOOL**: klasik "EVET"/"HAYIR" sonuçları
 - **REAL**: Yuvarlama hatalarını önlemek için hesap işlemlerinin sonuçları
 - **INT**: sayaç döngüleri veya parça sayaçları için klasik sayaç değişkenleri
 - **CHAR**: sadece bir karakter
String veya metin sadece CHAR alanı şeklinde gerçekleştirilebilir
- Ad verme ve bildirim
 - programın kolay okunabilir olmasını sağlamak için DECL kullanın,
 - anlamlı, kendinden açıklayıcı değişken adları kullanın,
 - şifremsi adlar veya kısaltmalar kullanmayın,
 - anlamlı ad uzunluklarından yararlanın, yani her defasında 24 karakter kullanmayın.

Basit bir veri tipi SRC dosyasında değişkeni oluşturma

olan değişkenin tanımlanmasında izlenecek yöntem

1. Uzman kullanıcı grubu
2. DEF satırının görüntülenmesini sağlama
3. SRC dosyasını editörde açma
4. Değişkenlerin bildirimini yapma

```
DEF MY PROG ( )  
DECL INT counter  
DECL REAL price  
DECL BOOL error  
DECL CHAR symbol  
INI  
...  
END
```

5. Programı kapatma ve kaydetme

DAT dosyasında değişkeni oluşturma

1. Uzman kullanıcı grubu
2. DAT dosyasını editörde açma
3. Değişkenlerin bildirimini yapma

```
DEFDAT MY_PROG  
EXTERNAL DECLARATIONS  
DECL INT counter  
DECL REAL price  
DECL BOOL error  
DECL CHAR symbol  
...  
ENDDAT
```

4. Veri listesini kapatma ve kaydetme

Değişkeni \$CONFIG.DAT dosyasında oluşturma

1. Uzman kullanıcı grubu
2. SYSTEM klasöründe \$CONFIG.DAT dosyasını editörde açın

```
DEFDAT $CONFIG  
BASISTECH GLOBALS  
AUTOEXT GLOBALS  
USER GLOBALS  
ENDDAT
```

3. "USER GLOBALS" foldunu seçin ve "Fold aç/kpt" düğmesiyle açın

4. Değişkenlerin bildirimini yapma

```
DEFDAT $CONFIG  
...  
;=====  
;Userdefined Types  
;=====  
;=====  
;Userdefined Externals  
;=====  
;Userdefined Variables  
;=====  
DECL INT counter  
DECL REAL price  
DECL BOOL error  
DECL CHAR symbol  
...  
ENDDAT
```

5. Veri listesini kapatma ve kaydetme

DAT dosyasında global değişkeni oluşturma

1. Uzman kullanıcı grubu
2. DAT dosyasını editörde açma
3. Program başında veri listesine PUBLIC anahtar sözcüğünü ekleme

```
DEFDAT MY PROG PUBLIC
```

4. Değişkenlerin bildirimini yapma

```
DEFDAT MY PROG PUBLIC  
EXTERNAL DECLARATIONS  
DECL GLOBAL INT counter  
DECL GLOBAL REAL price  
DECL GLOBAL BOOL error  
DECL GLOBAL CHAR symbol  
...  
ENDDAT
```

5. Veri listesini kapatma ve kaydetme

15.3.2 Basit Veri Tipleri Olan Değişkenleri İklendirme

KRL ile ilk- lendirmenin tarifi

- Her bildirimden sonra bir değişken sadece bir bellek yeri ayırır. Bu aşamada içerdiği değer daima geçersiz bir değerdir.
- SRC dosyasında bildirim ve iklendirilme daima iki ayrı satırda gerçekleşir.
- DAT dosyasında bildirim ve iklendirilme daima bir satırda gerçekleşir.
- Bir sabit, bildirim sırasında hemen iklendirilmek zorundadır.
- İklendirme bölümü ilk değer atamasıyla başlar

İklendirme prensipleri

Tam sayıları iklendirme

- Ondalık sayı olarak iklendirme

```
value = 58
```

- İkili sayı olarak iklendirme

```
value = 'B111010'
```

İkili:	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
Ondalık	32	16	8	4	2	1

Hesaplama: $1*32+1*16+1*8+0*4+1*2+0*1 = 58$

- Onaltılık sayı iklendirme

```
value = 'H3A'
```

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ondalık	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Hesaplama: $3*16 + 10 = 58$

KRL ile başlatmada izlenecek yöntem

SRC dosyasında açıklama ve başlatma

1. SRC dosyasını editörde açma
2. Açıklama yapıldı
3. Başlangıç durumuna getirin


```
DEF MY PROG ( )
DECL INT counter
DECL REAL price
DECL BOOL error
DECL CHAR symbol
INI
counter = 10
price = 0 .0
error = FALSE
symbol = "X"
...
```

4. Programı kapatma ve kaydetme
DAT dosyasında açıklama ve başlatma

1. DAT dosyasını editörde açma
2. Açıklama yapıldı
3. Başlangıç durumuna getirin

```
DEFDAT MY_PROG
EXTERNAL DECLARATIONS
DECL INT counter = 10
DECL REAL price = 0.0
DECL BOOL error = FALSE
DECL CHAR symbol = "X"
...
```

4. Veri listesini kapatma ve kaydetme

DAT dosyasında açıklama ve SRC dosyasında başlatma

1. DAT dosyasını editörde açma
2. Açıklama yapın

```
DEFDAT MY_PROG
EXTERNAL DECLARATIONS
DECL INT counter
DECL REAL price
DECL BOOL error
DECL CHAR symbol
...
```

3. Veri listesini kapatma ve kaydetme
4. SRC dosyasını editörde açma
5. Başlangıç durumuna getirin

```
DEF MY_PROG ( )
...
INI
counter = 10
price = 0.0
error = FALSE
symbol = "X"
...
```

6. Programı kapatma ve kaydetme

15.3.3 KRL İle Basit Veri Tipli Değişken Değerlerinin Manipülasyonu

KRL ile değişken değerlerini değiştirme olanaklarının lislenmesi

Değişken değerleri program yordamlarında (SRC dosyası) göreve bağlı olarak farklı şekillerde değiştirilir. Aşağıda en yaygın yöntemler ele alınmaktadır. Bit işlemleri ve standart fonksiyonlarla manipülasyonlar mümkündür, fakat burada derinlemesine açıklanmamaktadır.

Veri manipülasyonu

- **Temel hesap türleri**
 - (+) Toplama
 - (-) Çıkarma
 - (*) Çarpma
 - (/) Bölme
- Karşılaştırma işlemleri
 - (==) aynı/eşit
 - (<>) eşit değil
 - (>) büyüktür
 - (<) küçüktür
 - (>=) büyük eşittir
 - (<=) küçük eşittir
- Mantıksal işlemler
 - **(NOT)** Tersleme
 - **(AND)** Mantıksal VE
 - **(OR)** Mantıksal VEYA
 - **(EXOR)** Dışlayıcı VEYA
- Bit işlemleri
 - **(B_NOT)** Bit bit tersleme
 - **(B_AND)** Bit bit VE'li bağdaştırma
 - **(B_OR)** Bit bit VEYA'lı bağdaştırma
 - **(B_EXOR)** Bit bit dışlayıcı VEYA'lı bağdaştırma

Standart fonksiyonlar

- Mutlak değer fonksiyonu
- Kök fonksiyonu
- Sinüs ve kosinüs fonksiyonu
- Tanjant fonksiyonu
- Arkkosinüs fonksiyonu
- Arktanjan fonksiyonu
- String manipülasyonu birkaç fonksiyon

Veri manipülasyonundaki ilişkiler

- REAL ve INT veri tipi kullanımında değer değişikliği
 - yukarı / aşağı yuvarlama

```
;Bildirim
DECL INT A,B, C
DECL REAL R,S,T
; İklendirme
A = 3           ; A= 3
B = 5 .5       ; B= 6 (x 5 itibaren yukarı yuvarlanır)
C = 2 .25      ; C= 2( aşağı yuvarlama)
R = 4          ; R= 4.0
S = 6 .5       ; S= 6.5
T = C          ; T= 2.0 (aşağı yuvarlanmış değer alınır)
```

- Aritmetik işlemlerin sonuçları (+;-;*)

Operatörler	INT	REAL
INT	INT	REAL
REAL	REAL	REAL

```

;Bildirim
DECL INT D,E
DECL REAL U,V
; İklendirme
D = 2
E = 5
U = 0.5
V = 10.6
; Komut bölümü (veri Manipülasyonu)
D = D*E ; D = 2 * 5 = 10
E = E+V ; E = 5 + 10.6 = 15.6 -> yukarı yuvarlama E=16
U = U*V ; U = 0.5 * 10.6 = 5.3
V = E+V ; V = 16 + 10.6 = 26.6

```

- Aritmetik işlemlerin sonuçları (/)
 - Tam sayı değerleriyle işlemlerdeki özellikler:
 - Salt tam sayı işlemlerinin ara sonuçlarında tüm virgöl sonrası değerler atılır.
 - Bir tam sayı değişkenine değer atandığında sonuç normal hesap kurallarına göre yuvarlanır.

```

; Bildirim
DECL INT F
DECL REAL W
; İklendirme
F = 10
W = 10.0
; Komut bölümü (veri manipülasyonu)
; INT / INT -> INT
F = F/2 ; F=5
F = 10/4 ; F=2 (10/4 = 2.5 -> Virgöl sonrası atılır)
; REAL / INT -> REAL
F = W/4 ; F=3 ( 10.0/4 =2.5 -> yukarı yuvarlama)
W = W/4 ; W=2.5

```

Karşılaştırma işlemleri

Karşılaştırma işlemleriyle mantıksal ifadeler oluşturulabilir. Böylesi bir karşılaştırmanın sonucu daima BOOL veri tipine sahiptir

Operatör/KRL	Açıklama	izin verilen veri tipleri
=	aynı/eşit	INT, REAL, CHAR, BOOL
<>	eşit değil	INT, REAL, CHAR, BOOL
>	büyüktür	INT, REAL, CHAR
<	küçüktür	INT, REAL, CHAR
>=	büyük eşittir	INT, REAL, CHAR
<=	küçük eşittir	INT, REAL, CHAR

```

;Bildirim
DECL BOOL G,H
; İklendirme / Komut Bölümü
G=10>10.1 ; G=FALSE
H= 10/3 == 3 ; H=TRUE
G= G<>H ; G=TRUE

```

Mantıksal işlemler

Mantıksal işlemlerle mantıksal ifadeler oluşturulabilir. Böylesi bir işleminin sonucu daima BOOL veri tipine sahiptir.

İşlemler		NOT A	A AND B	A OR B	A EXOR B
A=TRUE	B=TRUE	FALSE	TRUE	TRUE	FALSE
A=TRUE	B=FALSE	FALSE	FALSE	TRUE	TRUE
A=FALSE	B=TRUE	TRUE	FALSE	TRUE	TRUE
A=FALSE	B=FALSE	TRUE	FALSE	FALSE	FALSE

```
; Bildirim
DECL BOOL K,L,M
; İklendirme/komut bölümü
K = TRUE
L = NOT K ; L=FALSE
M = (K AND L) OR (K EXOR L) ; M=TRUE
L = NOT (NOT K) ; L=TRUE
```

İşleyiciler sahip oldukları öncelik sırasında göre yürütülürler

Öncelik	Operatör
1	NOT (B_NOT)
2	Çarma (*); bölme (/)
3	Toplama (+), çıkarma (-)
4	AND (B_AND)
5	EXOR (B_EXOR)
6	OR (B_OR)
7	her türlü karşılaştırma (==; <>; ...)

```
; Bildirim
DECL BOOL X, Y
DECL INT Z
; İklendirme/komut bölümü
X = TRUE
Z = 4
Y=(4*Z+16 <> 32) AND X ; Y= FALSE
```


Veri manipülasyonunda izlenecek yöntem

1. Değişken veya değişkenler için veri tipini belirleyin
2. Değişkenlerin geçerliliğini ve kullanım ömrünü belirleyin
3. Değişken bildirimini yapın
4. Değişkeni iklendirin
5. program yordamlarında yani daima SCR dosyasında değişkeni manipüle edin
6. SRC dosyasını kapatma ve kaydetme

Dur komutu

HALT komutu programı durdurur. Son geçen hareket talimatı eksiksiz uygulanır.

Program sadece başlat tuşuna basılarak devam ettirilebilir. Ardından HALT komutundan sonraki talimat yürütülür.

 Bir Interrupt programında, ancak ön çalışma tümüyle işlendikten sonra program durdurulur.

15.4 KRL İle Arraylar / Alanlar Mit KRL

KRL ile alanların tanımı

Array olarak da bilinen alanlar, aynı veri tipinde olan birden çok değişken için, indeksler yardımıyla erişilebilen bellek yerleri sunarlar

- Alanlar için ayrılmış olan bellek sınırlıdır. Maksimum alan büyüklüğü, veri tipinin bellek ihtiyacına bağlıdır
- Açıklama sırasında alan büyüklüğü ve veri tipinin bilinir olması gerekir
- KRL'de başlangıç indeksi daima 1 sayıdır
- Başlatma daima tek tek yapılabilir
- SRC dosyasında başlatma bir döngü yardımıyla da yapılabilir

Alan boyutları

- 1 boyutlu alan

```
dimension[4]= TRUE
```

- 2 boyutlu alan

```
dimension[2,1]= 3.25
```

- 3 boyutlu alan

```
dimension[3,4,1]= 21
```

- 4 boyutlu veya daha üst boyutlu alan KRL tarafından desteklenmemektedir

Alanların kullanımındaki ilişkiler

Alan değişkenlerinin kullanım ömrü ve geçerliliği, basit veri tipli değişkenlerinin kullanımındakiyle aynıdır

Alan açıklama

- SRC dosyasında oluşturma

```
DEF MY_PROG ( )  
DECL BOOL error[10]  
DECL REAL value[50,2]  
DECL INT parts[10,10,10]  
INI  
...  
END
```

- Veri listesinde oluşturma (\$CONFIG.DAT dahil)

```
DEFDAT MY_PROG  
EXTERNAL DECLARATIONS  
DECL BOOL error[10]  
DECL REAL value[50,2]  
DECL INT parts[10,10,10]  
...  
ENDDAT
```

SRC dosyasında alan açıklama ve başlatma

- Her alanı indeks çağrısıyla teker teker

```
DECL BOOL error[10]
INI
error[1]=FALSE
error[2]=FALSE
error[3]=FALSE
error[3]=FALSE
error[4]=FALSE
error[5]=FALSE
error[6]=FALSE
error[7]=FALSE
error[8]=FALSE
error[9]=FALSE
error[10]=FALSE
```

- Uygun döngüler aracılığıyla

```
DECL BOOL error[10]
INI
DECL INT x
FOR x = 1 TO 10
  error[x]=FALSE
ENDFOR
```

- Alanı veri listesinde başlatma



Döngü tamamlandıktan sonra x, 11 değerine sahiptir

- Her alanı indeks çağrısıyla teker teker ve peşinden veri listesinde değer gösterimiyle

```
DEFDAT MY_PROG
EXTERNAL DECLARATIONS
DECL BOOL error[10]
error[1]=FALSE
error[2]=FALSE
error[3]=FALSE
error[4]=FALSE
error[5]=FALSE
error[6]=FALSE
error[7]=FALSE
error[8]=FALSE
error[9]=FALSE
error[10]=FALSE
```

- Veri listesinde izin verilmeyen açıklama ve başlatma

```
DEFDAT MY PROG
EXTERNAL DECLARATIONS
DECL BOOL error 10]
DECL INT size = 32
error [1]= FALSE
error [2]= FALSE
error [3]= FALSE
error [4]= FALSE
error [5]= FALSE
error [6]= FALSE
error [7]= FALSE
error [8]= FALSE
error [9] = FALSE
error [10] =FALSE
```



On adet hata bildirimini üretir “İlk değer seti ilk bölümde değil”



Ana menüde **Gösterge > Değişken > Tek** seçildiğinde ve **Adı** alanında değişkenlerin adı, köşeli parantez ile ve indeks olmadan girildiğinde komple alan gösterilebilir (örn. error[])

Alanları veri listesinde bildirme ve SRC dosyasında başlatma

- Bir alan veri listesinde bu şekilde oluşturulursa güncel değerleri veri listesinde görüntülenemez; güncel değerler sadece değişken gösterimi üzerinden denetlenebilir.

```
DEFDAT MY_PROG
EXTERNAL DECLARATIONS
DECL BOOL error[10]
```

```
DEFDAT MY_PROG
INI
  error[1]=FALSE
  error[2]=FALSE
  error[3]=FALSE
  ....
error[10]=FALSE
```

veya

```
DEF MY_PROG ( )
INI
FOR x = 1 TO 10
  error[x]=FALSE
ENDFOR
```

Döngüler aracılığıyla başlatma

- 1 boyutlu alan

```
DECL INT parts[15]
DECL INT x
FOR x = 1 TO 15
  parts[x]= 4
ENDFOR
```

- 2 boyutlu alan

```
DECL INT parts_table[10,5]
DECL INT x, y FOR x = 1 TO 10
  FOR y = 1 TO 5
    parts table[x, y]= 6
  ENDFOR
ENDFOR
```

- 3 boyutlu alan

```
DECL INT parts_palette[5,4,3]
DECL INT x, y, z
FOR x = 1 TO 5
  FOR y = 1 TO 4
    FOR z = 1 TO 3
      parts_palette[x, y, z]= 12
    ENDFOR
  ENDFOR
ENDFOR
```

Array kullanımında izlenecek yöntem

1. Alan için veri tiplerini belirleme
2. Alanın geçerliliğini ve kullanım ömrünü belirleyin
3. Alan bildirimini yürütün
4. Alan öğelerini ilklendirin
5. Program yordamlarında, yani daima SCR dosyasında, alanı manipüle edin
6. SRC dosyasını kapatma ve kaydetme

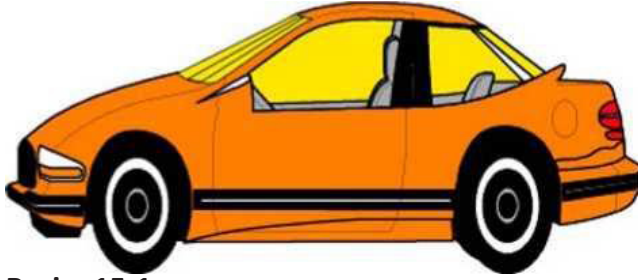
```

DEF MY_PROG ( )
DECL REAL palette_size[10]
DECL INT counter
INI
; İklendirme
FOR counter = 1 TO 10
    palette size[counter] = counter * 1.5
ENDFOR
...
; Değerleri tek tek değiştirme
palette size[8] = 13
...
; Değer karşılaştırma
IF palette_size[3] > 4.2 THEN
...

```

15.5 KRL İle Yapılar

Birkaç tekil
bilgili değiş-
kenler



Resim 15-1

- Birleşik veri vtipi: **Yapı**
- Alanlar yardımıyla aynı veri tipine sahip değişkenler birleştirilebilir. Gerçek dünyadaki değişkenler ise genelde farklı veri tiplerinden oluşur.
- Örneğin bir otomobilin motor gücü ya da kilometresi Integer (tamsayı) ti- pindedir. Fiyatı için ise Real (gerçek sayı) tipi daha uygundur. Buna karşın bir klima sisteminin mevcut olup olmadığı daha çok bir Bool veri tipidir.
- Bunların tümü bir arabayı tarif eder.
- Bir yapıyı STRUC anahtar sözcüğü ile kendiniz tanımlayabilirsiniz
- Bir yapı, farklı veri tiplerinin bir birleşimidir.

```
STRUC CAR_TYPE INT motor, REAL price, BOOL air_condition
```

- Bir yapıyı kullanabilmek için önce tanımlanmış olması gerekir.

Yapı kullanımı

Bir yapının kullanılabilirliği/tanımlaması

- Lokal olarak oluşturulan yapılar END satırına ulaşıldığında geçersizdir
- Birden çok programda kullanılan yapılar \$CONFIG.DAT dosyasında bildirilmek zorundadır
- Kendinizin tanımladığı yapılar daha iyi anlaşılabilmeleri için **TYPE** ile sona ermelidir
- Bir yapıda basit INT, REAL, BOOL ve CHAR veri tipleri kullanılabilir.

```
STRUC CAR_TYPE INT motor, REAL price, BOOL air_condition
```

- Bir yapıya CHAR alanları eklenebilir

```
STRUC CAR TYPE INT motor, REAL price, BOOL air_condition, CHAR car_model[15]
```

- Bir yapıda örneğin POS gibi bilinen yapılar da kullanılabilir.

```
STRUC CAR_TYPE INT motor, REAL price, BOOL air_condition, POS car_pos
```

- Yapı tanımlamasından sonra bir yapıya ilişkin bir iş değişkeninin bildirilmesi gerekir.

```
STRUC CAR_TYPE INT motor, REAL price, BOOL air_condition DECL CAR_TYPE my_car
```

Bir yapıyı başlatma / değiştirme

- Başlatma, parantez işaretleriyle gerçekleştirilebilir.
- Başlatma sırasında parantezlerle sadece sabitler (sabit değerler) kullanılabilir.
- Değer ataması için belirli bir sıra gözetmek gerekmez.

```
my_car = {motor 50, price 14999.95 air_condition TRUE}
```

```
my_car = {price 14999.95, motor air_condition TRUE}
```

- Bir yapıda tüm yapı öğelerinin belirtilmesi gerekmez.
- Bir yapı bir yapı ögesiyle başlatılır.
- Başlatılmayan değerler bilinmezdir veya bilinmeyene ayarlıdır.

```
my_car = {motor 75}; fiyat ve klima sistemi için değer bilinmiyor
```

- Başlatma, nokta ayırıcıyla da gerçekleştirilebilir.

```
my_car.price = 9999.0
```

- Bir yapı ögesi teker teker bir nokta ayırıcı yardımıyla daima yeniden değiştirilebilir. Bu sırada diğer yapı öğeleri silinmez ve değiştirilmez.

```
my_car = {price 14999.95, motor 50, air_condition TRUE}
```

```
my_car.price = 12000.0 ; fiyat ve klima sistemi için değer korunur
```

Ancak:

```
my car = {price 14999.95, motor 50, air_condition TRUE}
```

```
my car = {price 12000.0}; fiyat ve klima sistemi için değer silinir
```

Program sistemde kayıtlı olan çok sayıda hazır tanımlı yapı kullanır Örnekler, konumlarda ve mesaj programlamasında bulunabilir.

Konumlar konusunda önceden tanımlı program yapıları

- **AXIS:** STRUC AXIS REAL A1, A2, A3, A4, A5, A6
- **E6AXIS:** STRUC E6AXIS REAL A1, A2, A3, A4, A5, A6, E1, E2, E3, E4, E5, E6
- **FRAME:** STRUC FRAME REAL X, Y, Z, A, B, C
- **POS:** STRUC POS REAL X, Y, Z, A, B, C INT S,T
- **E6POS:** STRUC E6POS REAL X, Y, Z, A, B, C, E1, E2, E3, E4, E5, E6 INT S,T

Bir yapıyı bir konumla başlatma

- Parantezle başlatma sırasında sadece sabitler (sabit değerler) kullanılabilir

```
STRUC CAR_TYPE INT motor, REAL price, BOOL air_condition, POS car_pos
```

```
DECL CAR_TYPE my_car
```

```
my_car = {price 14999.95, motor 50, air_condition TRUE, car_pos {X 1000, Y 500, A 0}}
```

- Başlatma, nokta ayırıcıyla da gerçekleştirilebilir

```
my_car.price = 14999.95
```

```
my_car.motor = 50
```

```
my_car.air_condition = TRUE
```

```
my_car.car_pos = {X 1000, Y 500, A 0}
```

- Nokta ayırıcıyla yapılan başlatmada değişkenler de kullanılabilir

```
my_car.price = my_price*0.85
my_car.car_pos.X = x_value
my_car.car_pos.Y = 750+y_value
```

Yapı oluşturma

1. Yapı tanımlama

```
STRUC CAR_TYPE INT motor, REAL price, BOOL air_condition
```

2. Çalışma değişkenleri bildirimini

```
DECL CAR_TYPE my_car
```

3. Çalışma değişkenlerini ilklendirme

```
my_car = {motor 50, price 14999.95, air_condition TRUE}
```

4. İş değişkenlerinin değerlerini değiştirme ve/veya değerlerini karşılaştırma

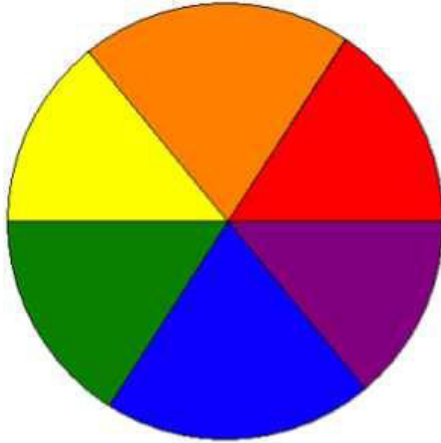
```
my_car.price = 5000.0
```

```
my_car.price = value_car
```

```
IF my_car.price >= 20000.0 THEN
...
ENDIF
```

15.6 ENUM Numaralandırma Veri Tipi

Değişken
değeri olarak
açık metin



Resim 15-2

- Numaralama veri tipi, yeşil, sarı veya mavi gibi sınırlı sayıda sabitten oluşur

```
ENUM COLOR_TYPE green, blue, red, yellow
```

- Sabitler serbest seçilebilen adlardır
- Sabitler programcı tarafından belirlenir
- Bir numaralama tipini kullanabilmek için önce tanımlanmış olması gerekir
- COLOR_TYPE veri tipinde KutuRengi gibi bir iş değişkeni, sadece bir sabitin değerini alabilir
- Bir sabitin atanması işlemi için daima şu simge kullanılır: #

Bir numaralan-
dırma veri tipini
kullanma

Kullanılabilirlik / kullanımı

- Sadece bilinen sabitler kullanılabilir
- Bir numaralandırma tipi sınırsız kez kullanılabilir
- Bir numaralandırma tipi tek başına kullanılabilir

```
ENUM COLOR_TYPE green, blue, red, yellow
```

- Bir numaralandırma tipi bir yapıya birleştirilebilir

```
ENUM COLOR_TYPE green, blue, red, yellow  
STRUC CAR_TYPE INT motor, REAL price, COLOR_TYPE car_color
```

Geçerlilik/Kullanım ömrü

- Lokal olarak oluşturulan numaralandırma veri tipleri END satırına ulaşıldığında geçersizdir
- Birden çok programda kullanılan numaralandırma veri tipleri ŞCON- FIG.DAT dosyasında bildirilmek zorundadır

Ad verme

- Numaralandırma veri tipleri ve bunların sabitlerinin adları kendi kendilerini açıklayıcı nitelikte olmalıdır
- Anahtar sözcüklerin kullanılması yasaktır
- Kendinizin tanımladığı numaralandırma veri tipleri daha iyi anlaşılabilmesi için **TYPE** ile sona ermelidir

Bir numara- landırma veri tipi oluşturma

1. Numaralandırma değişkenlerinin ve sabitlerin tanımı

```
ENUM LAND_TYPE de, be, cn, fr, es, br, us, ch
```

2. Çalışma değişkenleri bildirim

```
DECL LAND_TYPE my_land
```

3. Çalışma değişkenlerini ilklendirme

```
my_land = #be
```

4. Çalışma değişkenlerinin değer karşılaştırması

```
IF my_land == #es THEN
```

```
...
```

```
ENDIF
```

16 KRL İle Hareket Programlama

16.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- KRL ile hareket programlama
- KRL ile göreceli hareketleri programlama
- Robot pozisyonlarını hesaplama ve manipüle etme
- "Status" ve "Turn" öğelerine yönelik değişiklik

16.2 KRL İle Hareketler Programlama

Bir hareketin tanımı

Bir hareket için zorunlu bilgiler

- Hareket türü - PTP, LIN, CIRC
- Hedef pozisyonu ve duruma göre yardımcı pozisyon
- Tam durma veya nokta atlama
- Olası nokta atlama mesafesi
- Hız - PTP (%) ve yörünge hareketi (m/s)
- İvme
- Alet - TCP ve yük
- Çalışma base
- Robot yönlendirmeli veya harici alet
- Yörünge hareketlerinde yönelim kılavuzu
- Olası nokta atlama mesafesi
- CIRC dairesel harekette daire açısı

Hareket programlama prensibi

Hareket türü **PTP**

- PTP **Hedef nokta** <C_PTP <Yörünge nokta atlama>>

Öğe	Açıklama
<i>Hedef nokta</i>	<p>Tip: POS, E6POS, AXIS, E6AXIS, FRAME</p> <p>Hedef nokta kartezyen veya eksene özel olarak belirtilebilir. Kartezyen koordinatlar BASE koordinat sistemini esas alır.</p> <p>Hedef noktanın tüm komponentleri verilmezse, kontrol sistemi, eksik olan komponentler için önceki pozisyonun değerlerini alır.</p>

Öğe	Açıklama
C PTP	Hedef noktanın atlanmasını sağlar. PTP-PTP nokta atlamasında C_PTP bilgisinin verilmesi yeterlidir.
Yörünge nokta atlama	Sadece PTP-CP nokta atlama için Bu parametre, nokta atlamanın en erken ne zaman başlayacağını tanımlar. Olası bilgiler: C DIS Mesafe parametresi (varsayılan): Nokta üzerinden atlama en erken, hedef noktaya olan mesafe ŞAPO.CDIS değerinin altına düştüğünde başlar. C ORI Yönelim parametresi: Nokta üzerinden atlama en erken, baskın yönelim açısı ŞAPO.CDIS değerinin altına düştüğünde başlar. C VEL Hız parametresi: Nokta üzerinden atlama en erken, hedef noktasına doğru frenleme aşamasındaki hız ŞAPO.CVEL değerinin altına düştüğünde başlar.

- Robot, DAT dosyasından bir pozisyona gider. Pozisyon daha önce inline formu aracılığıyla öğretilmiştir ve bu P3 noktasında atlama yapar.

PTP XP3 C_PTP

- Robot, girilen bir pozisyona gider.
 - Eksene özel (AXIS ya da E6AXIS)

PTP {A1 0, A2 -80, A3 75, A4 30, A5 30, A6 110}

- Uzay pozisyonu (güncel etkin alet ve Base ile)

PTP {X 100, Y -50, Z 1500, A 0, B 0, C 90, S 3, T 35}

- Robot sadece bir ya da birden çok ünite girilerek hareket eder

PTP {A1 30} ; sadece A1, 30°'ye sürülür

PTP {X 200, A 30} ; sadece X 200mm'ye ve A 30°'ye

Hareket türü **LIN**

- LIN Hedef nokta <Yörünge nokta atlama>

Öge	Açıklama
Hedef nokta	<p>Tip: POS, E6POS, FRAME</p> <p>Hedef noktanın tüm komponentleri verilmezse, kontrol sistemi, eksik olan komponentler için önceki pozisyonun değerlerini alır.</p> <p>POS ya da E6POS tipinde bir hedef nokta içerisinde Statür ve Turn bilgisi, LIN hareketlerinde (CIRC hareketlerinde olduğu gibi) yoksayılır.</p> <p>Koordinatlar BASE koordinat sistemine göre dir.</p>
Yörünge nokta atlama	<p>Bu parametre, hedef noktada atlama yapılmasını sağlar. Aynı zamanda nokta atlamanın en erken ne zaman başlayacağını tanımlar. Olası bilgiler:</p> <p>C DIS</p> <p>Mesafe parametresi: Nokta üzerinden atlama en erken, hedef noktaya olan mesafe ŞAPO.CDIS değerinin altına düştüğünde başlar.</p> <p>C ORI</p> <p>Yönelim parametresi: Nokta üzerinden atlama en erken, baskın yönelim açısı ŞAPO.CDIS değerinin altına düştüğünde başlar.</p> <p>C VEL</p> <p>Hız parametresi: Nokta üzerinden atlama en erken, hedef noktasına doğru frenleme aşamasındaki hız ŞAPO.CVEL değerinin altına düştüğünde başlar.</p>

- Robot, hesaplanmış bir pozisyona gider ve ABLAGE[4] noktasında atlama yapar

LIN ABLAGE[4] C_DIS

Hareket türü **CIRC**

- CIRC Yardımcı nokta, hedef nokta<, CA Daire açısı> <Yörünge nokta atlama>


Öge	Açıklama
Yardımcı nokta	<p>Tip: POS, E6POS, FRAME</p> <p>Yardımcı noktanın tüm komponentleri verilmezse, kontrol sistemi, eksik olan komponentler için önceki pozisyonun değerlerini alır.</p> <p>Yönelim açısı ve yardımcı nokta içerisindeki Status ve Turn bilgileri genel kural olarak yoksayılır.</p> <p>Yardımcı noktayı atlamak mümkün değildir. Daima tam olarak noktaya gidilir.</p> <p>Koordinatlar BASE koordinat sistemine göre dir.</p>
Hedef nokta	<p>Tip: POS, E6POS, FRAME</p> <p>Hedef noktanın tüm komponentleri verilmezse, kontrol sistemi, eksik olan komponentler için önceki pozisyonun değerlerini alır.</p> <p>POS ya da E6POS tipinde bir hedef nokta içerisinde Status ve Turn bilgisi, CIRC hareketlerinde (LIN hareketlerinde olduğu gibi) yoksayılır.</p> <p>Koordinatlar BASE koordinat sistemine göre dir.</p>

Oğe	Açıklama
<i>Daire açısı</i>	<p>Daire hareketinin toplam açısını verir. Bu sayede, programlanan hedef noktanın dışında, hareketin uzatılmasını veya kısaltılmasını sağlar. Böylece asıl hedef nokta artık programlanan hedef nokta değildir.</p> <p>Birim: Derece. Sınırlama yok. Özel olarak, 360° değerinden büyük bir daire açısı programlanabilir.</p> <ul style="list-style-type: none"> ■ Daire açısı pozitif: Dairesel rota, Başlangıç noktası > Yardım noktası > Hedef nokta yönünde hareket eder. ■ Daire açısı negatif: Dairesel rota, Başlangıç noktası > Hedef nokta > Yardım noktası yönünde hareket eder.
<i>Yörünæ nokta atlama</i>	<p>Bu parametre, hedef noktada atlama yapılmasını sağlar. Aynı zamanda nokta atlamanın en erken ne zaman başlayacağını tanımlar. Olası bilgiler:</p> <p style="text-align: center;">C DIS</p> <p>Mesafe parametresi: Nokta üzerinden atlama en erken, hedef noktaya olan mesafe \$APO.CDIS değerinin altına düştüğünde başlar.</p> <p style="text-align: center;">C ORI</p> <p>Yönelim parametresi: Nokta üzerinden atlama en erken, baskın yönelim açısı \$APO.CDIS değerinin altına düştüğünde başlar.</p> <p style="text-align: center;">C VEL</p> <p>Hız parametresi: Nokta üzerinden atlama en erken, hedef noktasına doğru frenleme aşamasındaki hız \$APO.CVEL değerinin altına düştüğünde başlar.</p>

- Robot, DAT dosyasında belirlenen pozisyonlara gider. Pozisyonlar daha önce inline formu aracılığıyla öğretilmiştir ve 190°lik bir daire açısını izler.

CIRC XP3, XP4, CA 190

- Daire açısı **CA**

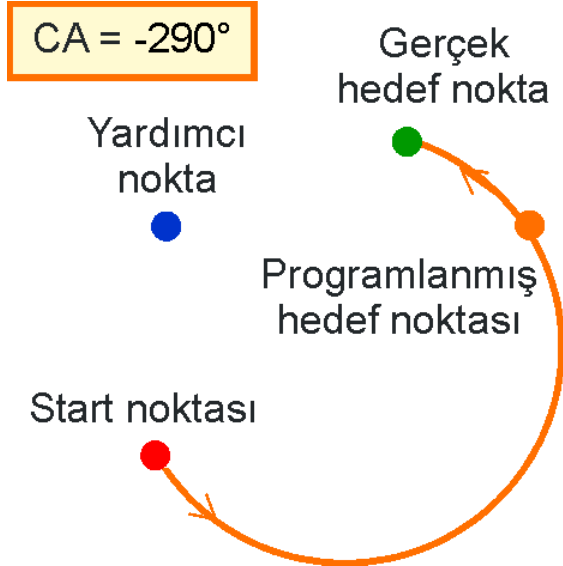
 Gerçek hedef noktada, programlanmış hedef noktada öğretilmiş olan yönelim üzerlenilir.

- **Daire açısı pozitif (CA>0):** Daire, programlanmış yönde sürülür: Başlangıç-Yardımcı-Hedef nokta



Resim 16-1: Daire açısı CA = +290°

- Daire açısı negatif (CA<0): Daire, programlanmış yönün tersi yönde sürülür: Başlangıç-Hedef-Yardımcı nokta



Resim 16-2: Daire açısı CA = -290°

Hareket programlaması için ön ayarlar

- Mevcut ayarlar üzerlenilebilir:
 - INI satırının işlenmesinden
 - En son inline formundan
 - İlgili sistem değişkenlerinin en son ayarlarından
- İlgili sistem değişkenlerini değiştirme veya ilklendirme

Hareket parametreleri sistem değişkenleri

- Alet: \$TOOL ve \$LOAD
 - Ölçülen TCP'yi etkinleştirme

\$TOOL = tool_data[x] ; x = 1...16

- İlişkili yük verilerini etkinleştirme

\$LOAD = load_data[x] ; x = 1...16

- **Referans bazı / çalışma bazı:: \$BASE**

- Ölçülen Base etkinleştirme

\$BASE = base_data[x] ; x = 1...32

- Robot yönlendirmeli veya harici alet: \$IPO_MODE

- Robotla yönlendirilen alet

\$IPO_MODE = #BASE

- Harici alet

\$IPO_MODE = #TCP

- **Hız:**

- PTP hareketinde

\$VEL_AXIS[x] ; x=1...8 her eksen için

- LIN veya CIRC hareket türlerinde

\$VEL.CP = 2.0 ; [m/s] Yörünge hızı

\$VEL.ORI1 = 150 ; [°/s] Çevirme hızı

\$VEL.ORI2 = 200 ; [°/s] Dönme hızı



Aletin etkime yönü çoğu durumda X eksenidir. Döndürme hızı, bu X eksenini etrafında C açısıyla döndürme işlemidir. Çevirme hızı, diğer iki açı (A ve B) etrafında çevrilir.

- İvme

- PTP hareketinde

\$ACC_AXIS[x] ; x=1...8 her eksen için

- LIN veya CIRC hareket türlerinde

\$ACC.CP = 2.0 ; [m/s²] Yörünge ivmesi

\$ACC.ORI1 = 150 ; [°/s²] Sallanma ivmesi

\$ACC.ORI2 = 200 ; [°/s²] Dönme ivmesi

- Üzerinden atlama mesafesi

- Sadece PTP hareketinde: **C_PTP**

\$APO_CPTP = 50 ; Nokta atlama büyüğü [%] şu öğede: C_PTP

PTP XP3 C_PTP

- LIN, CIRC yörünge hareketlerinde ve PTP'de: **C_DIS**

Hedef noktaya olan mesafe \$APO.CDIS değerinin altına düşmelidir

\$APO.CDIS = 250.0 ; [mm] Mesafe

PTP XP3 C_DIS

LIN XP4 C_DIS

- LIN, CIRC yörünge hareketlerinde: **C_ORI**

Domine eden yönelim açısı \$APO.CORI değerinin altına düşmek zorundadır

\$APO.CORI = 50.0 ; [°] Açı

LIN XP4 C_ORI

- LIN, CIRC yörünge hareketlerinde: **C_VEL**

Hedef noktaya olan frenleme evresindeki hız \$APO.CVEL değerinin altında olmak zorunda

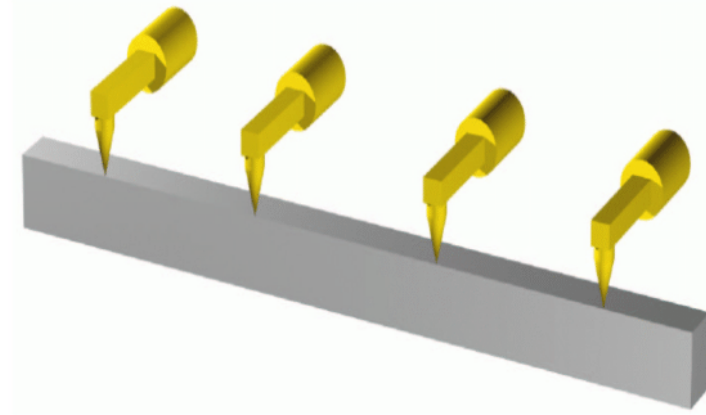
\$APO.CVEL = 75.0 ; [%] Yüzde

LIN XP4 C_VEL

- **Oryantasyon kılavuzu:** sadece LIN ve CIRC'de
 - LIN ve CIRC için **ŞORI_TPYE**

ŞORI_TYPE = #CONSTANT

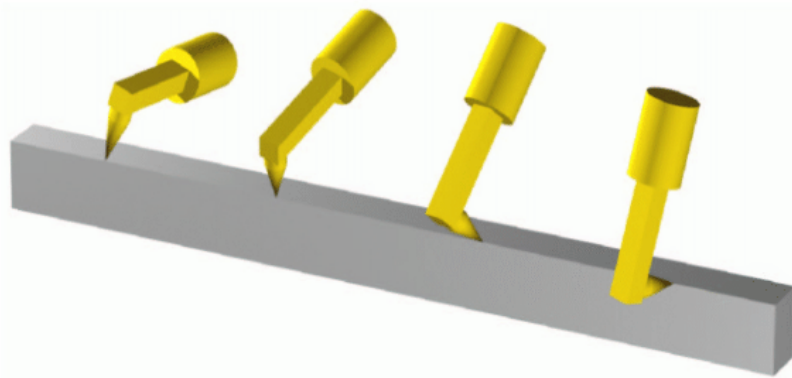
Yörünge hareketi sırasında yönelim sabit kalır Son nokta için programlı yönelim yoksayılır



Resim 16-3: Sabit yönlendirme kılavuzu

ŞORI_TYPE = #VAR

Yörünge hareketi sırasında yönelim sürekli olarak, hedef noktadaki yönelime doğru değişir



Resim 16-4: Standart veya elle PTP

ŞORI_TYPE = #JOINT

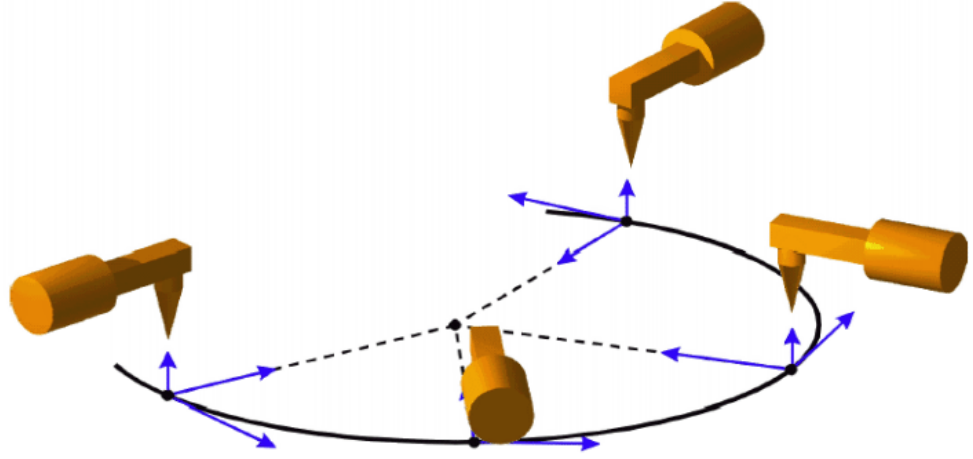
Yörünge hareketi sırasında aletin yönelimi başlangıç pozisyonundan son pozisyona değiştirilir. Bu, bilek eksen açısının doğrusal aktarımı sayesinde gerçekleşir. Bilek tekilliği sorunu bu opsiyonla önlenir, çünkü alet etkiye yönü etrafında döndürme ve çevirme yoluyla yönelim kılavuzluğu yapılmaz.

- Sadece CIRC hareketinde: **ŞCIRC_TPYE**



ŞORI_TYPE = #JOINT ile bilek eksen açısının lineer aktarımı sözkonusu olduğunda **ŞCIRC_TPYE** değişkeni önemsizdir.

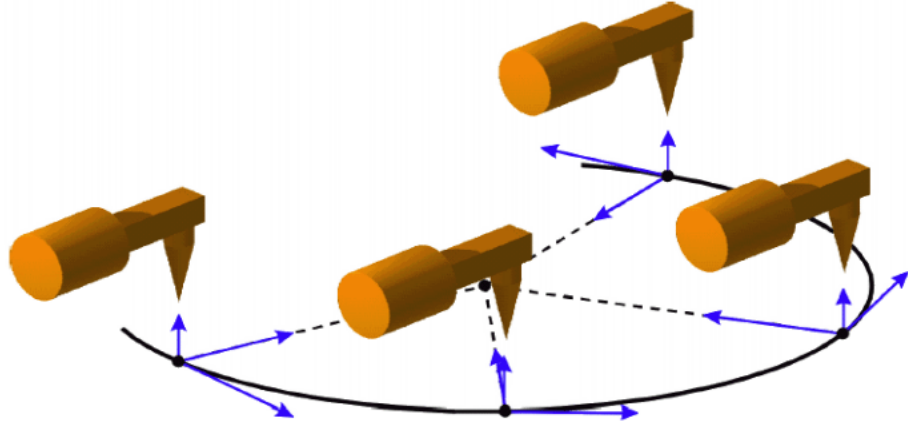
Dairesel hareket sırasında yörüngeye ilişkin yönelim kılavuzu



Resim 16-5: Sabit yönlendirme, rotayla ilgili

```
$CIRC_TYPE = #BASE
```

Dairesel hareket sırasında uzaya ilişkin yönelim kılavuzu



Resim 16-6: Sabit yönlendirme, bazla ilgili

KRL ile hareketler programlanırken izlenecek yöntem

1. Uzman kullanıcı olarak programı Aç tuşuyla editöre yükleyin
2. Hareket programlamasına ilişkin ön ayarları kontrol edin ve üzerlenin veya yeniden ilklendirin:
 - Alet (\$TOOL ve \$LOAD)
 - Base ayarları (\$BASE)
 - Robot yönlendirmeli veya harici alet (\$IPO_MODE)
 - Hız
 - İvme
 - Olası nokta atlama mesafesi
 - Olası yönelim kılavuzu
3. Hareket komutu oluşturma, şunlardan oluşur:
 - Hareket türü (PTP, LIN, CIRC)
 - Hedef noktası (CIRC için yardımcı nokta dahil)
 - CIRC için olası radyan açısı (CA)

- Nokta atlamayı etkinleştirme (C_PTP, C_DIS, C_ORI, C_VEL)
4. Yeniden hareket olduğunda nokta 3'e geri
 5. Editörü kapatın ve kaydedin

16.3 Robot Pozisyonlarını Hesaplama Veya Manipüle Etme

Açıklama

Robot hedef konumları

- Yapılara kaydedilir:
- Önceden tanımlanan aşağıdaki yapılar vardır.
 - AXIS
Aks açısı A1 ... A6
 - E6AXIS
Aks açısı A1 ... A6 ve E1 ... E6
 - POS
Konum (X, Y, Z), status ve turn (S, T) ile yönlendirme (A, B, C)
 - E6POS
Konum (X, Y, Z), status ve turn (S, T) ile yönlendirme (A, B, C) ... E6
 - FRAME -
Konum (X, Y, Z), yönlendirme (A, B, C)
- DAT dosyasında mevcut konumlar manipüle edebilir.
- Mevcut konumların tek üniteleri, nokta ayırıcısı yardımıyla seçilerek değiştirilebilir.

Prensip

DUYURU Hesaplama sırasında doğru TOOL ve BASE ayarlarına dikkat edilmesi ve peşinden hareket programlamasında etkinleştirilmesi önemlidir. Buna dikkat edilmemesi halinde istenmeyen hareketler ve çarpışmalar ortaya çıkabilir.

Önemli sistem değişkenleri

- **\$POS_ACT**: Güncel robot konumu. Değişken (E6POS), TCP'nin BASE koordinat sistemine göre olan hedef konumunu tarif eder.
- **\$AXIS_ACT**: Güncel aksa özgü robot konumu (hedef değer). Değişken (E6AXIS), güncel aks açılarını veya aks konumlarını içerir

Mutlak hedef konumunu hesaplama

- DAT dosyasındaki konumu bir defalığına değiştirme

```
XP1.x = 450 ; Yeni X değeri 450mm
XP1.z = 30.0*distance ; Yeni Z değeri hesaplanır
PTP XP1
```

- DAT dosyasındaki konumu her çalışmada değiştirme

```
; X değeri her defasında 450mm kaydırılır
XP2.x = XP2.x + 450
PTP XP2
```

- Konum devralınır ve bir değişkene kaydedilir

```
myposition = XP3
myposition.x = myposition.x + 100 ; X değerine 100mm eklenir
myposition.z = 10.0*distance ; Yeni Z değerini hesaplayın
myposition.t = 35 ; Turn değerini ayarlama
PTP XP3 ; Konum değiştirilmedi
PTP myposition ; Hesaplanan konum
```

İzlenecek yöntem

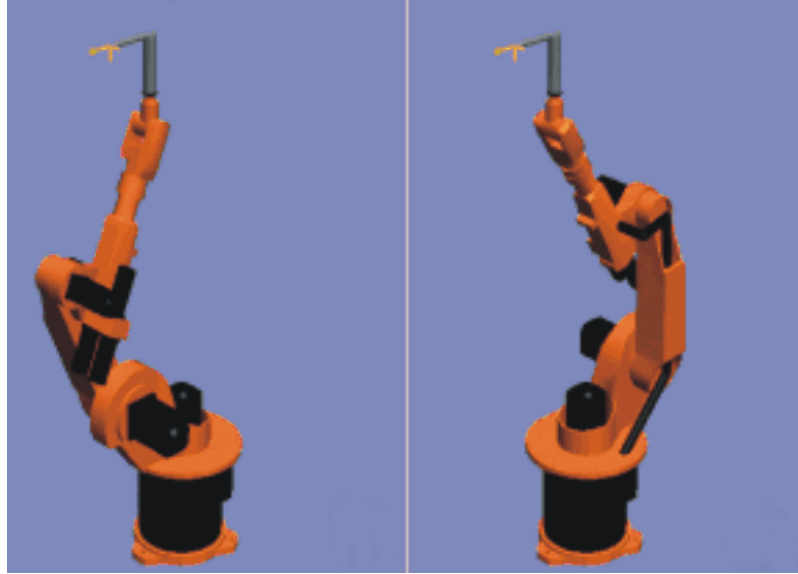
1. Uzman kullanıcı olarak programı **Aç** tuşuyla editöre yükleyin

2. Pozisyonu hesaplayın/manipüle edin. Yeni hesaplanan değerleri duruma göre yeni bir değişkende geçici olarak saklayın.
3. Hareket programlaması için olan ön ayarları kontrol edin ve üzerlenin veya yeniden ilklendirin:
 - Alet (\$TOOL ve \$LOAD)
 - Base ayarları (\$BASE)
 - Robot yönlendirmeli veya harici alet (\$IPO_MODE)
 - Hız
 - İvme
 - Olası nokta atlama mesafesi
 - Olası yönelim kılavuzu
4. Hareket komutu oluşturma, şunlardan oluşur:
 - Hareket türü (PTP, LIN, CIRC)
 - Hedef noktası (CIRC için yardımcı nokta dahil)
 - CIRC için olası radyan açısı (CA)
 - Nokta atlamayı etkinleştirme (C_PTP, C_DIS, C_ORI, C_VEL)
5. yeniden hareket olduğunda nokta 3'e geri
6. Editörü kapatın ve kaydedin

16.4 Status ve Turn Bitlerini Bilinçli Değişirme

Açıklama

TCP'nin pozisyon (X, Y, Z) ve yönelime (A, B, C) ilişkin değerleri, bir robotun pozisyonunu tam olarak belirlemek için yeterli değildir, çünkü TCP aynıken birden çok eksen konumu mümkündür. "Status" ve "Turn" bilgileri, çeşitli olası eksen konumları arasından, kesin bir eksen pozisyonu belirlemeye yarar.



Resim 16-7: Örnek: TCP aynı, eksen konumu farklı

- Status (S) ve Turn (T), POS ve E6POS veri tiplerinin öğelerindedir:

```
STRUC POS REAL X, Y, Z, A, B, C, INT S, T
```

```
STRUC E6POS REAL X, Y, Z, A, B, C, E1, E2, E3, E4, E5, E6, INT S, T
```

- Robot kontrol sistemi programlanmış Status ve Turn değerlerini **sadece PTP hareketlerinde** dikkate alır. CP hareketlerinde bu değerler yoksayılr.
- Bu nedenle, bir KRL programındaki ilk hareket deyimini, robot için kesin bir başlangıç konumunun tanımlı olabilmesi için aşağıdaki deyimlerden biri olmak zorundadır:

- POS veya E6POS tipi eksiksiz bir PTP deyimini
 - Ya da AXIS veya E6AXIS tipi, eksiksiz bir PTP deyimini
- o"**Eksiksiz**", varış noktasının bütün bileşenlerinin verilmesi gerektiği anlamına gelir. Default-HOME pozisyonu daima eksiksiz bir PTP deyimidir.

- Diğer deyimlerde Status ve Turn bilgisini yazmak gerekmez.
 - Robot kontrol sistemi, o ana kadar geçerli olan Status değerini korur.
 - Turn değeri, CP hareketlerinde yörüngeden hesaplanır.
 - PTP hareketlerinde robot kontrol sistemi, mümkün olan en kısa yolu veren (yani yazılım limit şalterlerini tetiklemeyip aynı zamanda start açısına en yakın olan) Turn değerini seçer.

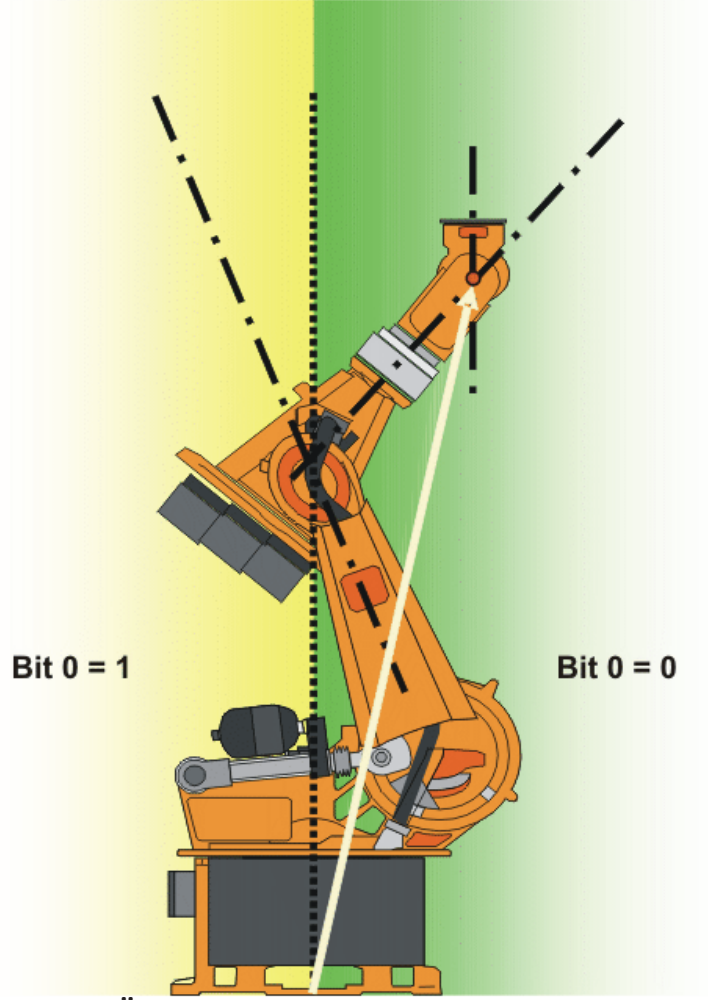
Fonksiyon

DURUM

- Durum göstergesi eksen konumunda çok anlamlılığı engeller.
- **Bit 0:** el bileklerinin (A4, A5, A6) kesişme noktasının pozisyonunu belirtir.

Poz.	Değer
Baş aşağı bölgesi	Bit 0 = 1
Bilek eksenlerinin kesişme noktasının x değeri, A1 koordinat sistemine göre negatif olursa robot baş aşağı bölgesindedir.	
Temel bölge	Bit 0 = 0
Bilek eksenlerinin kesişme noktasının x değeri, A1 koordinat sistemine göre alındığında pozitif olursa robot temel bölgesindedir.	

Eksen 1 açısı 0° olduğunda, A1 koordinat sistemi, ŞROBROOT koordinat sistemiyle aynı olur. 0° açıdan farklı değerlerde ise eksen 1 ile birlikte hareket ettirilir.



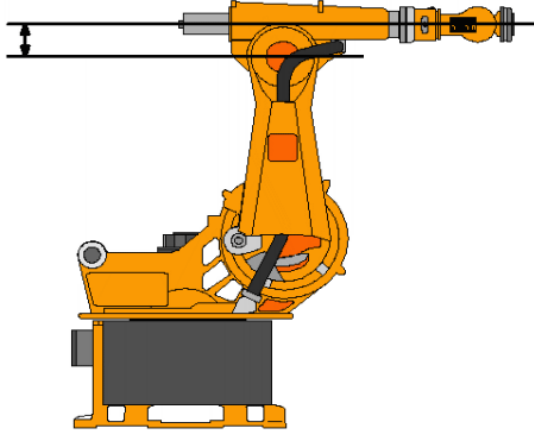
Resim 16-8: Örnek: Bilek eksenlerinin kesişme noktası (kırmızı nokta) temel bölgede yer alır.

- **Bit 1:** Eksen 3 pozisyonunu belirtir. Bit 1 değerinin değiştiği açı robot tipine bağlıdır.

3 ve 4 numaralı eksenlerin kesiştiği robotlar için:

Poz.	Değer
$A3 > 0^\circ$	Bit 1 = 1
$A3 < 0^\circ$	Bit 1 = 0

3 ve 4 numaralı eksenler arasında ofseti olan robotlar için, Bit 1 değerinin değiştiği açı bu ofsetin büyüklüğüne bağlıdır.



Resim 16-9: A3 ve A4 arası ofset: Örnek KR 30

- **Bit 2:** Eksen 5 pozisyonunu belirtir.

Poz.	Değer
$A5 > 0$	Bit 2 = 1
$A5 < 0$	Bit 2 = 0

- **Bit 3 kullanılmaz** ve değeri daima 0'dır.
- **Bit 4:** Noktanın bir mutlak doğruluklu robotla öğretilmiş olup olmadığını belirtir.

Nokta, bitin değerinden bağımsız olarak hem mutlak doğruluklu hem de mutlak doğruluklu olmayan robotlarla sürülebilir. Bit 4 sadece bilgi amaçlıdır ve robot kontrol sisteminin noktayı nasıl hesapladığına ilişkin bir etkiye sahip değildir. Bunun anlamı, bir robot çevrim dışı programlansa dahi Bit 4 yoksayılabilir.

Tarif	Değer
Nokta, mutlak doğruluklu bir robotla öğretilmedi	Bit 4 = 0
Nokta, mutlak doğruluklu bir robotla öğretildi	Bit 4 = 1

TURN

- Turn bilgisi, +180 °'den büyük veya -180 °'den küçük eksen açılarını özel bir sürme yöntemi (örn. ara noktalar) kullanmadan sürebilme olanağı verir. Tekil bitler, dönel eksenlerde eksen değerinin ön işaretini şu şekilde belirlemektedir:

Bit = 0: Aç $\geq 0^\circ$

Bit = 1: Aç $< 0^\circ$

- Tüm eksenlere genel bakış

Değer	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	$A6 \geq 0^\circ$	$A5 \geq 0^\circ$	$A4 \geq 0^\circ$	$A3 \geq 0^\circ$	$A2 \geq 0^\circ$	$A1 \geq 0^\circ$
1	$A6 < 0^\circ$	$A5 < 0^\circ$	$A4 < 0^\circ$	$A3 < 0^\circ$	$A2 < 0^\circ$	$A1 < 0^\circ$

Örnek:

DECL POS XP1 = {X 900, Y 0, Z 800, A 0, B 0, C 0, S 6, T 19}

T 19 karşılığı T 'B010011'. Bunun anlamı:

Eksen	Açı	İkili:
A 1	negatif	1
A 2	negatif	2
A 3	pozitif	4

İzlenecek yöntem

Eksen	Açı	İkili:
A 4	pozitif	8
A 5	negatif	16
A 6	pozitif	32

Uzman kullanıcı olarak programı **Aç** tuşuyla editöre yükleyin

2. **Status** ve **Turn** manipülasyonu. Yeni hesaplanan değerleri duruma göre yeni bir değışkünde geçici olarak saklayın.

3. Hareket programlamasına ilişkin ön ayarları kontrol edin ve üzerlenin veya yeniden ilklendirin:

- Alet (\$TOOL ve \$LOAD)
- Base ayarları (\$BASE)
- Robot yönlendirmeli veya harici alet (\$IPO_MODE)
- Hız
- İvme
- Olası nokta atlama mesafesi
- Olası yönelim kılavuzu

4. Hareket komutu oluşturma, şunlardan oluşur:

- Hareket türü (PTP, LIN, CIRC)
- Hedef noktası (CIRC için yardımcı nokta dahil)
- CIRC için olası radyan açısı (CA)
- Nokta atlamayı etkinleştirme (C_PTP, C_DIS, C_ORI, C_VEL)

5. Yeniden hareket olduğunda nokta 3'e geri

6. Editörü kapatın ve kaydedin

17 Bir Üst Düzey Kontrol Sistemiyle Çalışma

17.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- PLC'de program başlatmak için hazırlık
- PLC bağlantısı uyarlama
- Harici otomatik arabirimini konfigüre etme

17.2 PLC'den Program Başlatma İçin Hazırlık

Tesiste birleştirilmiş robotlar

Robot süreçleri merkezi bir noktadan kumanda edilecekse (ana bilgisayar veya PLC), bu, "otomatik harici" arabirimi üzerinden gerçekleşir.



Resim 17-1: PLC bağlantısı

Tesis yapısı prensibi

KR C4 ile bir PLC arasındaki iletişimi başarılı bir şekilde gerçekleştirebilmek için aşağıdakiler gereklidir:

- **Harici otomatik işletim türü:** Bir yönetim bilgisayarının veya bir PLC'nin, robot sisteminin kontrolünü üstlendiği işletim türüdür.
- **CELL.SRC:** Robot programlarının dışarıdan seçimi için organizasyon programıdır.
- PLC ve robot sinyal alışverişi: **Harici otomatik arabirim**, giriş ve çıkış sinyallerinin

konfigürasyonu için:

- Robota giden kumanda sinyalleri (girişler): Başlatma ve durdurma sinyali, program numarası, hata onayı
- Robot durumu (çıkışlar): Tahrikler, konum, hata vs. ile ilgili durum

Programı haricinden başlatma için güvenlik uyarıları

CELL programı seçildikten sonra bir SAK sürüşünün yürütülmesi gerekir.



UYARI

Bir SAK sürüşü, seçilen hareket komutları seti PTP sürüş komutunu içerdiğinde, gerçek pozisyonun hedef pozisyonuna PTP sürüşü olarak yürütülür. Seçilen hareket komutları seti LIN ya da CIRC komutlarını içerdiğinde SAK sürüşü LIN hareketi olarak yürütülür. Çarpışmaları önlemek için hareketi gözlemleyin. SAK sürüşünde hız otomatik olarak düşürülmüştür.

SAK sürüşü bir kez yapıldıktan sonra harici start sırasında başka SAK sürüşü yürütülmez.

UYARI Otomatik harici işletiminde SAK sürüşü yoktur. Bu, robotun başlattıktan sonraki ilk programlanmış pozisyona, programlanmış (düşürülmüş) hızla hareket etmesi ve orada durmaması anlamına gelir

Programı hariç-
ten
başlatmada
izlenecek yöntem

Ön koşullar

- T1 veya T2 işletim türü
 - Otomatik harici ve CELL.SRC programı için girişler/çıkışlar konfigüre edilmiştir.
1. Navigatörde CELL.SRC programını seçin. CELL programı daima KRC:\R1 dizininde yer alır
 2. Program-Override'ı %100'e ayarlayın. (Bu önerilen ayarlama. Gerekğinde başka bir değer ayarlanabilir).



Resim 17-2: Cell seçimi ve manuel Override ayarı

1. HOV ayarı
 2. Cell.src seçimi
1. SAK sürüşü yürütün:
Onay şalterine basın ve basılı tutun. Ardından başlat tuşuna basın ve mesaj penceresinde "CÜU ulaşıldı" görüntülenene kadar basılı tutun.
 2. 4. "Harici otomatik" işletim türünü seçin.
 3. 5. Programı, üst düzeydeki bir kumanda ünitesinden (PLC) başlatın.

17.3 PLC Bağlantısını Uyarlama (Cell.src)

Cell.src organi-
zasyon
programı

PLC'den aktarılan program numaralarının yönetimi için, Cell.src organizasyon programı kullanılır. "R1" klasöründe yer almaktadır. Her olağan programda olduğu gibi Cell programında uyarlanabilir, fakat program yapısının korunması gerekir.

Cell programının yapısı ve işlevselliği

```
1 DEF CELL ( )
6 INIT
7 BASISTECH INI
8 CHECK HOME
9 PTP HOME Vel= 100 % DEFAULT
10 AUTOEXT INI
11 LOOP
12 P00 (#EXT_PGNO,#PGNO_GET,DMV[],0)
13 SWITCH PGNO ; Select with Programmnumber
14
15 CASE 1
16 P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0)
17 ;EXAMPLE1 ( ) ; Call User-Program
18
19 CASE 2
20 P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0)
21 ;EXAMPLE2 ( ) ; Call User-Program
22
23 CASE 3
24 P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0)
25 ;EXAMPLE3 ( ) ; Call User-Program
26
27 DEFAULT
28 P00 (#EXT_PGNO,#PGNO_FAULT,DMV[],0)
29 ENDSWITCH
30 ENDLLOOP
31 END
```

Resim 17-3: Cell (Hücre) programı

1 İklendirme ve Home pozisyonu

- Base parametrelerini iklendirme
- "Home" konumundan sonra robot pozisyonunun kontrolü
- Otomatik harici arabirimini iklendirme

2 Sonsuz döngü:

- Program numarasının "P00" modülü tarafından sorgulanması
- Tespit edilen program numarasıyla seçim döngüsüne giriş.

3 Program numarası seçim döngüsü

- Program numarasına uygun olarak ("PGNO" değişkeninde yer alır) ilgili dala ("CASE") atlama gerçekleşir.
- Dal içerisinde kayıtlı robot programı yürütülür.
- Geçersiz program numaraları "Default" dalına atlanmasına neden olur.
- Yürütme tamamlandıktan sonra döngü tekrarlanır.

İzlenecek yöntem

1. "Uzman" kullanıcı grubuna geçin.
2. CELL.SRC dosyasını açın.
3. "CASE" bölümünde "EXAMPLE" adını, ilgili program numarası tarafından çağrılacak programın adıyla değiştirin. Adın başındaki noktalı virgülü silin.

```

LOOP
P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
SWITCH PGNO ; Select with Programnumber

CASE 1
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
main()

CASE 2
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
body_38()
body_515()

DEFAULT
P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
ENDSWITCH
ENDLOOP

```

Resim 17-4: Uyarlanmış bir Cell programı örneği

4. Programı kapatın ve değişiklikleri kaydedin.

17.4 Harici Otomatik Konfigürasyonu ve Kullanılması

Açıklama



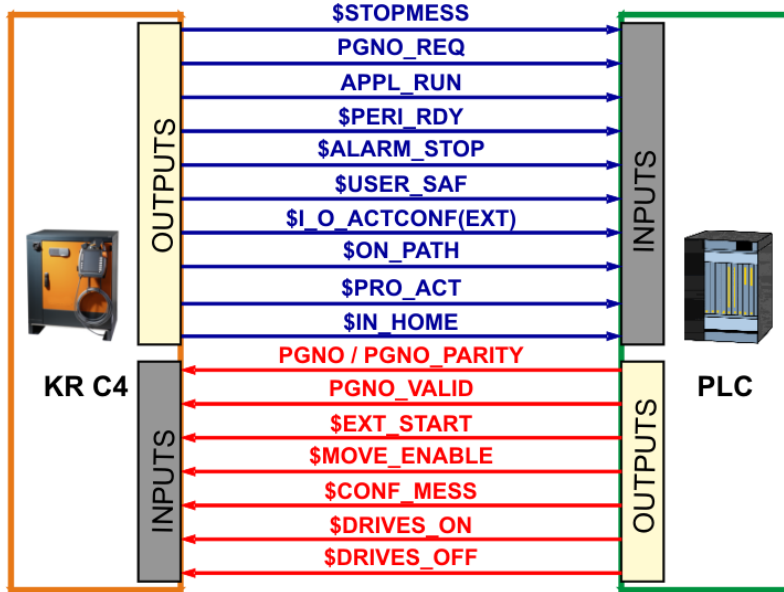
Resim 17-5: PLC bağlantısı

- Harici Otomatik arabirimi ile robot süreçleri hiyerarşide üstte olan bir kumanda (örn. SPS (PLC)) tarafından kumanda edilir.
- Harici otomatik arabirim üzerinden üst kumanda sistemi robot kontrol ünitesine, robot süreçleri için ilgili sinyalleri aktarır (örn. hareket onayı, hata onayı, program başlatma vb.). Robot kontrol ünitesi üst kumanda sistemine işletim ve arıza durumlarına ilişkin bilgi aktarır.

Harici Otomatik arabirimini kullanmak için, aşağıda belirtilen konfigürasyonlar gerçekleştirilmelidir:

1. CELL.SRC programı konfigüre edilmelidir.
2. Harici Otomatik arabirimine ait girişler/çıkışlar konfigüre edilmelidir.

Harici Otomatik arabirimi girişlerinin/ çıkışlarının kullanılması Arabiriminin en önemli sinyallerine genel bakış



Resim 17-6: En önemli sinyallere genel bakış, Harici Otomatik

Girişler (robot kumandasından bakıldığında)

- **PGNO_TYPE** - Program numarası-Tip
Bu değişken, hiyerarşide üstte olan kumanda tarafından aktarılan program numarasının hangi formatta okunacağını belirlemektedir.

Değer	Açıklama	Örnek
1	Bineer sayı olarak okunması. Hiyerarşide üstte olan kumanda tarafından program numarası bineer kodlanmış bir tam sayı değeri olarak devredilir.	0 0 1 0 0 1 1 1 => PGNO = 39
2	BCD değeri olarak okunması. Hiyerarşide üstte olan kumanda tarafından program numarası bineer kodlanmış bir ondalık değer olarak devredilir.	0 0 1 0 0 1 1 1 => PGNO = 27
3	N" içinden "1 olarak okunması. Program numarası hiyerarşide üstte olan kumanda veya çevre tarafından N" içinden "1 olarak kodlanmış değer olarak devredilir.	0 0 0 0 0 0 1 => PGNO = 1 0 0 0 0 1 0 0 0 => PGNO = 4

* Bu devretme formatında PGNO_REQ, PGNO_PARITY ve PGNO_VALID değerleri değerlendirilmez ve böylece anlamsızdır.

- **PGNO_LENGTH** - Program numarası uzunluğu
Bu değişken hiyerarşide üstte olan kumanda tarafından aktarılan program numarası için bit genişliğini belirlemektedir. Değer aralığı: 1 ... 16.
PGNO_TYPE'in değeri 2 ise, sadece 4, 8, 12 ve 16 bit genişliklerine müsaade edilir.
- **PGNO_PARITY** - Program numarası-Parite biti
Parite bit'inin hiyerarşide üstte olan kumanda tarafından aktarılan giriş.

Giriş	Fonksiyon
Negatif değer	Tek parite
0	Değerlendirme yok
Pozitif değer	Çift parite

PGNO_TYPE'in değeri 3 olduğunda, PGNO_PARITY değerlendirilmez.

- **PGNO_VALID** - Program numarası geçerli
Program numarasının okunması için hiyerarşide üstte olan kumandan gelen komutun aktarıldığı giriş.

Giriş	Fonksiyon
Negatif değer	Numara, sinyalin düşen kenarı ile birlikte devralınır.
0	Numara sinyalin artan kenarı ile EXT_START hattında devralınır.
Pozitif değer	Numara, sinyalin artan kenarı ile birlikte devralınır.

- **\$EXT_START** - Harici başlatma
Bu girişi ayarlayarak, etkin G/Ç arabiriminde bir program (norma durumda CELL.SRC) başlatılabilir veya sürdürülebilir.

i Sinyalin sadece artan kenarı değerlendirilmektedir.

⚠ UYARI Otomatik harici işletiminde SAK sürüşü yoktur. Bu, robotun başlattıktan sonraki ilk programlanmış pozisyona, programlanmış (düşürülmüş) hızla hareket etmesi ve orada durmaması anlamına gelir

- **\$MOVE_ENABLE** - Sürüşün serbest bırakılması
Giriş, robot tahriklerinin hiyerarşide üstte olan kumanda tarafından kontrol edilmesi için kullanılmaktadır.

Sinyal	Fonksiyon
TRUE	Manuel işletim ve program uygulaması mümkün olmaktadır
FALSE	Tüm tahriklerin durdurulması ve tüm etkin komutların kilitlenmesi

i Tahrikler, hiyerarşide üstte olan kumanda tarafından durdurulmuş ise, bu durumda "TÜM SÜRÜŞL. SERBEST BIRAKILMASI" mesajı gösterilir. Robotun hareket ettirilmesi ancak bu mesaj silindikten ve yeniden harici bir başlatma sinyali verildikten sonra mümkün olmaktadır.

i Devreye alma esnasında \$MOVE_ENABLE değeri sıklıkla \$IN[1025] değerinin üzerine projelendirilir. Bundan sonra başka bir girişin projelendirilmesi unutulursa, harici başlatma mümkün olmaz.

- **\$CONF_MESS** - Mesaj onayı
Bu girişin ayarlanması ile hiyerarşide üstte olan kumanda, hata mesajlarının arıza nedeni giderilir giderilmez onları kendi onaylamaktadır.

i Sinyalin sadece artan kenarı değerlendirilmektedir.

- **\$DRIVES_ON** - Tahrikler açık
Bu girişte bir High-İmpuls en az 20 ms süreyle mevcut olduğunda, hiyerarşide üstte olan kumanda robot tahriklerini açmaktadır

- **\$DRIVES_OFF** - Tahrikler kapalı
Bu girişte bir Low-İmpuls en az 20 ms süreyle mevcut olduğunda, hiyerarşide üstte olan kumanda robot tahriklerini kapatır.

Çıkışlar (robot kumandasından bakıldığında)

- **\$ALARM_STOP** - ACİL-DUR
Bu çıkış aşağıda belirtilen ACİL-DUR durumlarında reset edilir:
 - KCP'deki ACİL-DUR düğmesine basıldı. (Dâh. AcilKapat)
 - Harici ACİL-DUR



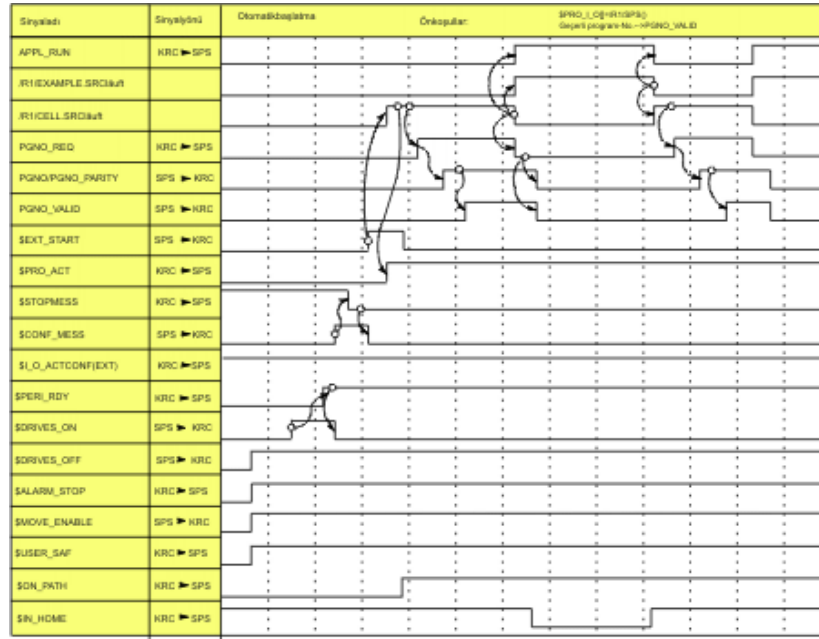
ACİL-DUR'da **\$ALARM_STOP** ve **Dâh. AcilDur** çıkışların durumları, ACİL-DUR'un türünü göstermektedir:

- Her iki çıkış FALSE durumunda: ACİL-DUR, KCP'de tetiklendi
- **\$ALARM_STOP** FALSE durumunda, **Dâh. AcilDur** TRUE durumunda: Harici ACİL-DUR

- **\$USER_SAF** - Kullanıcı koruması/Koruyucu kapı
Koruyucu ızgara sorgulama şalterinin açılmasında (İşletim türü AUT) veya bir onay şalterinin elden bırakılması esnasında (İşletim türü T1 veya T2) bu çıkış reset edilir.
- **\$PERI_RDY** - Tahrikler hazır
Bu çıkışın ayarlanması ile robot kumandası hiyerarşide üstte olan kumandaya, robot tahriklerinin açık olduğunu bildirmektedir.
- **\$STOPMESS** - Stop mesajı
Bu çıkış robot kumandası tarafından, hiyerarşide üstte olan kumandaya robotun durmasını gerektiren bir mesajın oluştuğunu göstermek için ayarlanır. (Örnekler: ACİL-DUR, kullanıcı koruması olmadan sürüş serbest bırakıldı)
- **\$I_O_ACTCONF** - Harici Otomatik etkin
Harici Otomatik işletim türü seçili olduğunda bu çıkış TRUE'dur ve giriş \$I_O_ACT TRUE'dur (normal durumda her zaman \$IN[1025] olmaktadır).
- **\$PRO_ACT** - Program etkin/çalışıyor
Bu çıkış, robot seviyesinde bir süreç etkin olduğunda daima ayarlanmış durumdadır. Program veya Interrupt çalışıldığı sürece, süreç etkindir. Program sonundaki program işleme süreci ancak, tüm İmpuls çıkışları ve Trigger'ler çalışılarak tamamlandıktan sonra etkisiz duruma geçer.
- **PGNO_REQ** - Program numarası sorgulaması
Bu çıkıştaki sinyal değişimi ile hiyerarşide üstte olan kumandadan bir program numarası aktarılması talep edilir.
PGNO_TYPE'in değeri 3 olduğunda, PGNO_REQ değerlendirilmez.
- **APPL_RUN** - Uygulama programı çalışıyor
Bu çıkışın ayarlanması ile robot kumandası hiyerarşide üstte olan kumandaya, şu anda bir programın çalıştığını bildirmektedir.
- **\$IN_HOME** - Robot HOME-pozisyonunda
Bu çıkış hiyerarşide üstte olan kumandaya, robotun HOME-pozisyonunda bulunup bulunmadığını bildirmektedir.
- **\$ON_PATH** - Robot hat üzerinde
Bu çıkış, robot programlanmış olan hattı üzerinde bulunduğu sürece ayarlanmış durumdadır. SAK-sürüşünden sonra çıkış ON_PATH olarak ayarlanır. Bu çıkış, robot hattan çıkana, program reset edilene veya bir cümle seçimi yapılanaya kadar ayarlanmış durumda kalır. ON_PATH sinyalinin bir tolerans penceresi yok; robot hattan çıkar çıkmaz, bu sinyal reset edilir.

Harici Otomatik haberleşmesi prensibi

Toplam sürece genel bakış



Resim 17-7: Tesisin otomatik başlatılması ve PGNO_VALID tarafından program numarasının onaylanması ile normal işletim

Kısmi bölümlere alt bölümlendirme

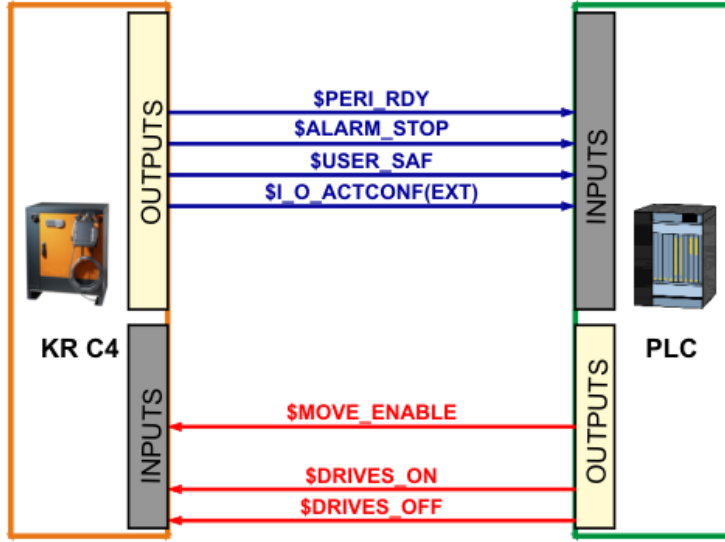
1. Tahriklerin açılması
 2. Mesajların onaylanması
 3. Cell programının başlatılması
 4. Program numarası devredilmelidir ve uygulama tamamlana kadar çalışmalıdır
- Bu bölgelerin her biri için koşullar sağlanmalıdır ve ayrıca robot durumlarının SPS'ye (PLC) bildirme olanağı mevcut olmalıdır.



Resim 17-8: Handshake

Bu ön tanımlı Handshake'lerinin kullanılması mantıklıdır.

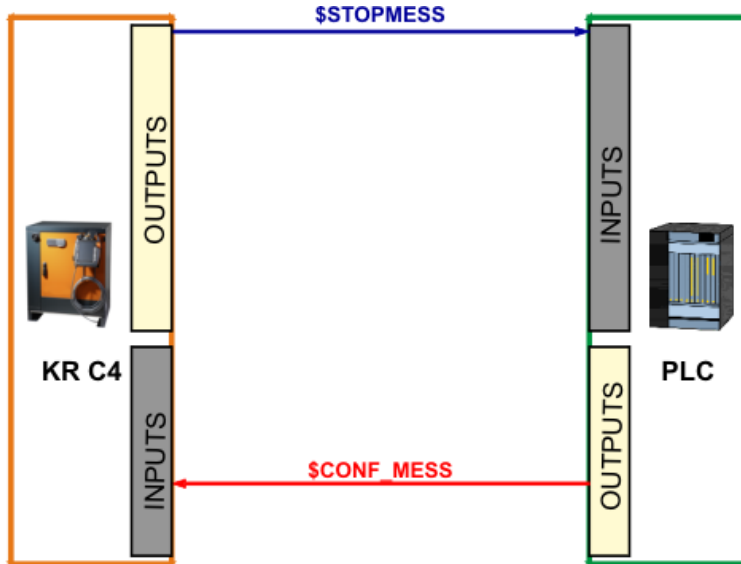
Tahriklerin açılması



Resim 17-11

- Önkoşullar
 - \$USER_SAF - Koruyucu kapı kapalı
 - \$ALARM_STOP - Hiçbir ACİL-DUR yok
 - \$I_O_ACTCONF - Harici Otomatik etkin
 - \$MOVE_ENABLE - Sürüşün serbest bırakılması mevcut
 - \$DRIVER_OFF - Tahrikler kapalı, etkinleştirilmiş değil
- Tahriklerin açılması
 - \$DRIVES_ON - Tahriklerin en az 20 ms için açılması
- Tahrikler hazır
 - \$PERI_RDY - tahrikler için geri bildirim gelir gelmez, \$DRIVES_ON sinyali geri alınır

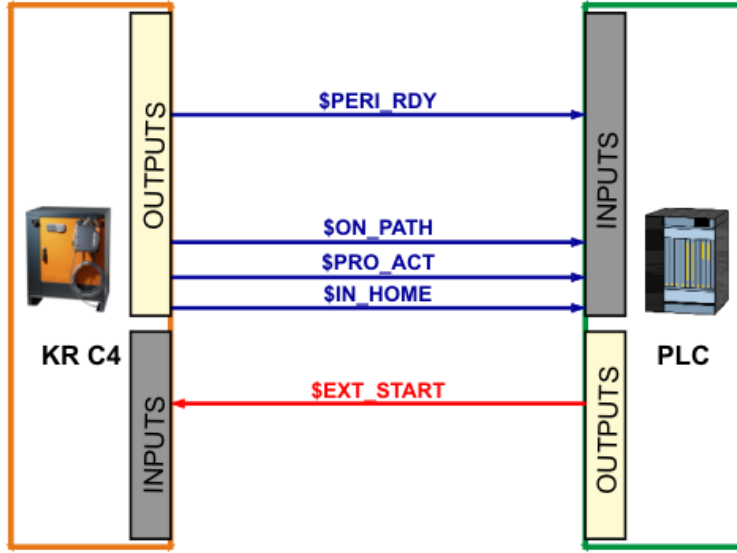
Mesajların onaylanması



Resim 17-15

- Önkoşullar
 - \$STOPMESS - Stop mesajı bekliyor
- Mesajı onaylayın
 - \$CONF_MESS - Mesajın onaylanması
- Onaylanabilir mesajlar silindi
 - \$STOPMESS - Stop mesajı artık beklemiyor \$CONF_MESS şimdi geri alınabilir

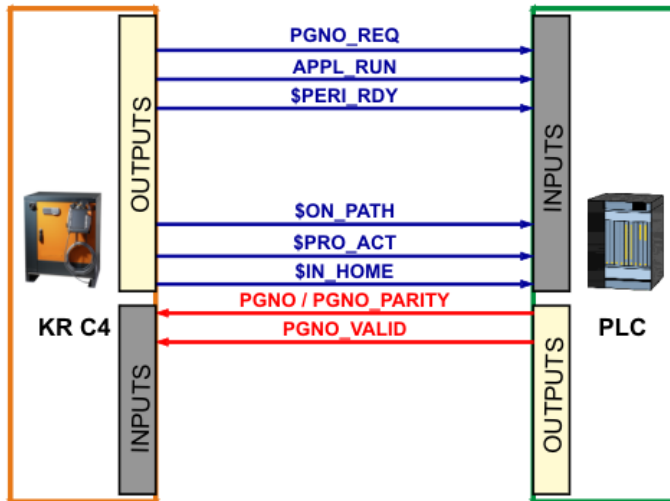
Programın (CELL.SRC) hariçten başlatılması



Resim 17-20

- Önkoşullar
 - \$PERI_RDY - Tahrikler hazır
 - \$IN_HOME - Robot HOME-pozisyonunda
 - \$STOPMESS yok - Hiçbir stop mesajı beklemiyor
 - Harici başlatma
 - \$EXT_START- Harici başlatmanın açılması (pozitif kenar)
 - CELL programı çalışıyor
 - \$PRO_ACT - CELL programı çalışıyor bildirimini yapıyor
 - \$ON_PATH - Robot programlanmış hattı üzerinde bulunur bulunmaz,
- \$EXT_START sinyali geri alınır

Program numarası devri ve uygulama programının çalıştırılması



Resim 17-27

Önkoşullar

- \$PERI_RDY - Tahrikler hazır
- \$PRO_ACT - CELL programı çalışıyor
- \$ON_PATH - Robot hat üzerinde
- \$IN_HOME - Robot HOME-pozisyonunda, yeniden çalıştırmada gerekli değil
- PGNO_REQ - Program numarası sorgulaması bekliyor
- Program numarası aktarımı ve onayı
- Program numarası devri

(Doğru dosya tipi (PGNO_TYPE), Program numarası uzunluğu (PGNO_LENGTH) ve Program numarası için ilk bit (PGNO_FBIT) ayarlı durumda)

- PGNO_VALID- Program numarası geçerli olacak şekilde kumanda (onay) edilmelidir (pozitif kenar)
- Uygulama programı çalışıyor
 - APPL_RUN - Uygulama programı çalışıyor bildirimini yapıyor
 - Robot HOME-pozisyonunu terk ediyor. Uygulama programı sonlandırıldığında robot yeniden HOME-pozisyonuna geri gelmektedir.

İzlenecek yöntem

1. Ana menüde sırasıyla **Konfigürasyon > Girişler/Çıkışlar > Harici Otomatik** öğelerini seçin.
2. **Değer** sütununda işlenecek hücreyi işaretleyin ve **Düzenle** ögesine basın.
3. İstenen değeri girin ve **Tamam** ögesine basarak kaydedin.
4. 2. ve 3. adımları tüm işlenecek değerler için tekrarlayın.
5. Pencereyi kapatın. Değişiklikler devralınır.

1	2	3	4	5
Harici otomatığın konfigürasyonu: Girişler				
	Tanım	Tip	Ad	Değer
1	Program numarası tipi	Yar	PGNO_TYPE	1
2	Program numarasını yansıtma	Yar	REFLECT_PROG_I	0
3	Bit genişliği program no.	Yar	PGNO_LENGTH	8
4	İlk bit programı no.	HO	PGNO_FBIT	33
5	Parite biti	HO	PGNO_PARITY	41
6	Program no. geçerli	HO	PGNO_VALID	42
7	Program başlatma	HO	\$EXT_START	1026
8	Sürüş izni	HO	\$MOVE_ENABLE	1025
9	Hata onayı	HO	\$CONF_MESS	1026
10	Tahrikler kapalı (invers)	HO	\$DRIVES_OFF	1025
11	Tahrikler açık	HO	\$DRIVES_ON	140
12	Arabirimi etkinleştirme	HO	\$I_O_ACT	1025

Resim 17-30: Harici otomatığın giriş konfigürasyonu

Poz.	Açıklama
1	Numara
2	Girişin/çıkışın uzun metin adı
3	Tip <ul style="list-style-type: none">■ Yeşil: Giriş/çıkış■ Sarı: Değişken veya sistem değişkeni (\$...)
4	Sinyalin veya değişkenin adı
5	Giriş/çıkış numarası veya kanal numarası
6	Çıkışlar tematik olarak sekmelere ayrılmıştır.

Harici otomatığın konfigürasyonu: Çıkışlar				
	Tanım	Tip	Ad	Değer
1	Kumanda hazır	NO	\$SRC_RDY1	
2	Acil çıkış çemberi kapalı	NO	\$ALARM_STOP	
3	Kullanıcı koruması kapalı	NO	\$USER_SAF	
4	Tahrikler hazır	NO	\$PERI_RDY	
5	Robot ayarlandı	NO	\$ROB_CAL	
6	Arabirim etkin	NO	\$I_O_ACTCONF	
7	Genel arıza	NO	\$STOPMESS	
8	Program yanıtması için ilk bit	NO	PGNO_FBIT_REFL	
9	Dahili Acil-Kapatma	NO	\$ALARM_STOP_I	

6

Başlatma koşulları Program durumu Robot konumu İşletim türü

Resim 17-31: Harici Otomatik çıkışlarının konfigürasyonu

18 WorkVisual İle Programlama

18.1 Genel Bakış

Bu eğitim ünitesinde şu konular anlatılmaktadır:

- Ürün açıklaması
- Robot kumandasına bağlantı
- Projeleri yönetme
- WorkVisual ile programlama

WorkVisual yazılım paketi, KR C4 kontrollü robot hücreleri için mühendislik ortamıdır. Aşağıdaki işlemlere sahiptir:

- Alan busları kurma ve kablolama
- Robotları çevrimdışı programlama
- RoboTeams çevrimdışı konfigüre etme
- Güvenlik konfigürasyonunu işleme
- Projeleri robot kumandasına aktarma
- Projeleri robot kumandasından WorkVisual'a aktarma
- Projeyi başka bir proje ile karşılaştırma ve gerektiğinde farkları devralma
- Uzun metinleri yönetme
- Opsiyon paketlerini yönetme
- Tanılama işlevi
- Robot kumandasına ilişkin sistem bilgilerini çevrimiçi görüntüleme
- Trace'leri konfigüre etme, kayıtları başlatma, Trace'leri değerlendirme (osiloskop ile)

18.2 WorkVisual İle Bağlantı

IP'nin manuel Koşullar

olarak verilmesi

- Kumanda dolabı, bir IT ağına bağlı olmadan çalışmaktadır
- Yönetici hakları ve boş bir ağ bağlantı noktası olan bir dizüstü bilgisayar.
- CSP'de ağ yuvası, KUKA PC'nin dahili KUKUA Line Interface arabirimine bağlıdır (dahili Switch üzerinden de mümkündür).

DUYURU	Standart teslimat durumunda KLI, 172.31.1.147 numaralı statik IP adresinde önceden ayarlıdır.
---------------	--



Resim 18-1: IP manuel

Kumanda dolabında IP adresinin manuel olarak verilmesi

- KLI arabiriminin konfigürasyonu, HMI tarafından bir asistan ile desteklenir.
- Uzman kullanıcı profili gereklidir.
- Konfigürasyonda hatalar kırmızı işaretlidir ve kaydetme işlemi reddedilir.
- Konfigürasyonun standart görünümü ve ayrıntılı görünümü sunulur.
- Standart görünümde sadece Windows arabirimi (büro ağına bağlantı) yapılandırılabilir.
- Ayrıntılı görünüm üzerinden ilave olarak sanal ağlar yapılandırılabilir.
- Menü yolu: KUKA tuşu > İşletime alma > Ağ konfigürasyonu.

Ağ konfigürasyonu

Windows arabirimi (virtual5)

Adres tipi: ▼

Gelişmiş konfigürasyonu etkinleştirme

Windows arabirimi, WorkVisual üzerinden kumanda ile iletişim kuran ağ arabirimidir.
(Gelişmiş bir konfigürasyon olmaksızın PROFINET kullanıldığında statik bir IP kullanılmalıdır.)

Resim 18-2: KLI-HMI virtual5

Adres tipi	Anlamı
Dinamik IP	Tüm ayarlar bir DHCP sunucusu tarafından yapılır
Sabit IP	IP adresi Alt ağ maskesi
IP yok	Standart ağ geçidi ayrı olarak ayarlanmak zorundadır. geçici olarak bir arabirimi gizlemek için
Gerçek zaman IP	Roboteam
Karışık IP	özel teknoloji paketleri konfigürasyonu



virtual5 sanal ağı DHCP için ayarlıysa PROFINET iletişimi mümkün değildir.

- Etkinleştir butonuna basın.
- Ayrıntılı görünümün konfigürasyon sayfası açılır.

Ağ konfigürasyonu

Konfigüre edilen arabirimler:

- virtual5 (Profinet)
 - Queue
 - Queue

Arabirim özellikleri:

Arabirim tanımı: Profinet

Adres tipi: Sabit IP adresi

IP adresi: 172 . 31 . 1 . 147

Alt ağ maskesi: 255 . 255 . 0 . 0

Standart ağ geçidi: 0 . 0 . 0 . 0

Windows arabirimi

Arabirimler | Kullanıcı tanımlı filtreler | NAT | Dahili alt ağlar

Arabirim ekle | Arabirimi kaldır | Filtre ekle | Filtreyi kaldır | Kaydetme

Resim 18-3: Ağ konfigürasyonu: Virtual 5

Arabirimler	Arabirim	Anlamı
virtual5	Tanım	Arabirim için ad, örn. PROFINET
	Adres tipi	Sabit IP Dinamik IP IP yok Gerçek zaman IP Karışık IP
	IP adresi	PROFINET-IP manüel gir
	Alt ağ maskesi	Alt ağ maskesini manüel gir
	Standart ağ geçidi	Standart ağ geçidi manüel gir
	Windows arabirimi	Bu arabirim için NAT kurallarının geçerli olup olmadığını belirler. Sadece bir arabirim yapılandırılmışsa bu durum standart olarak geçerlidir.

Arabirimler	Arabirim	Anlamı
1. Queue	Alıcı filtresi Profinet	Özel, ayarlanamayan PROFINET bağlantı noktaları
2. Queue	Alma filtresi paketler	Tümünü al - Tüm paketler kabul edilir Hedef IP adresi - sadece bu IP adresi için paketler kabul edilir

- **Arabirimler** butonu üzerinden yeni bir sanal arabirim "**virtual6**" eklenir.
- Bu arabirime, PROFINET ve büro ağı için ayrı IP bölgeleri kullanılması gerektiğinde ihtiyaç duyulur.

Ağ konfigürasyonu

Konfigüre edilen arabirimler:

- ▼ virtual5 (Profinet)
 - Queue
 - Queue
- ▼ **virtual6 (WINDOWS)**
 - Queue

Arabirim özellikleri:

Arabirim tanımı:

Adres tipi:
 ▼

Windows arabirimi

Arabirimler
Kullanıcı tanımlı filtreler
NAT
Dahili alt ağlar

Arabirim ekle
Arabirimi kaldır
Filtre ekle
Filtreyi kaldır
Kaydetme

Resim 18-4: Ağ konfigürasyonu: Virtual 6

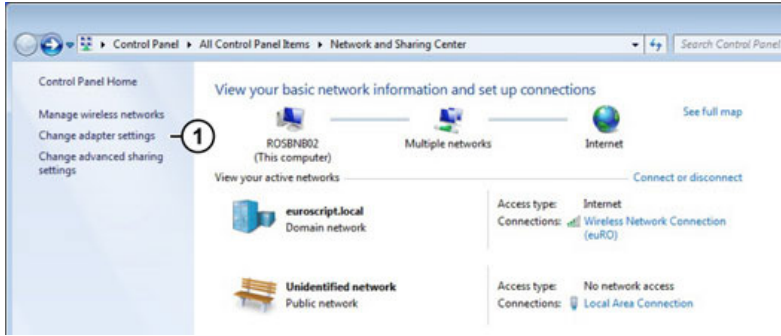
Arabirimler	Arabirim	Anlamı
virtual6	Tanım	Arabirim için ad, örn. büro ağı
	Adres tipi	Sabit IP Dinamik IP IP yok Gerçek zaman IP Karışık IP
	IP adresi	PROFINET-IP manüel gir
	Alt ağ maskesi	Alt ağ maskesini manüel gir
	Standart ağ geçidi	Standart ağ geçidi manüel gir

Arabirimler	Arabirim	Anlamı
	Windows arabirimi	Bu arabirim için NAT kurallarının geçerli olup olmadığını belirler. İki arabirimde manüel olarak seçilmek zorundadır.
Queue	Alma filtresi paketler	Tümünü al - Tüm paketler kabul edilir Hedef IP adresi - sadece bu IP adresi için paketler kabul edilir

i Bu ünite de örnek olarak Windows 7 işletim sistemi olan bir dizüstü bilgisayara IP adresi verme tarif edilmektedir.

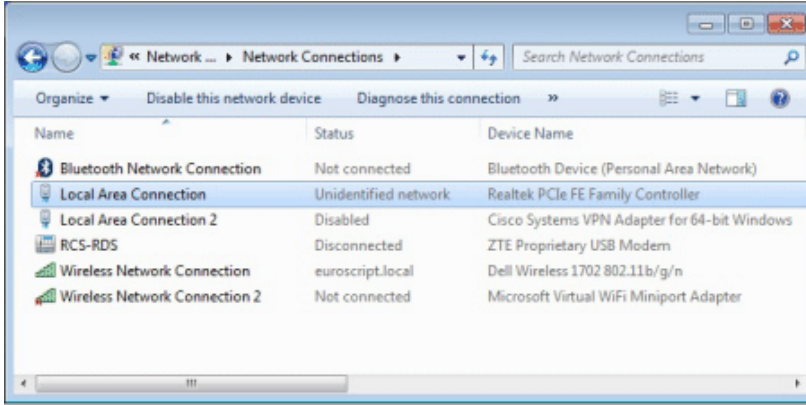
Dizüstü bilgisayarda IP adresinin manuel olarak verilmesi

1. Windows'da **Denetim Masası > Ağ ve İnternet > Ağ ve Onay Merkezi** açın.
2. Adaptör ayarlarını değiştir (1) seçeneğini açın.



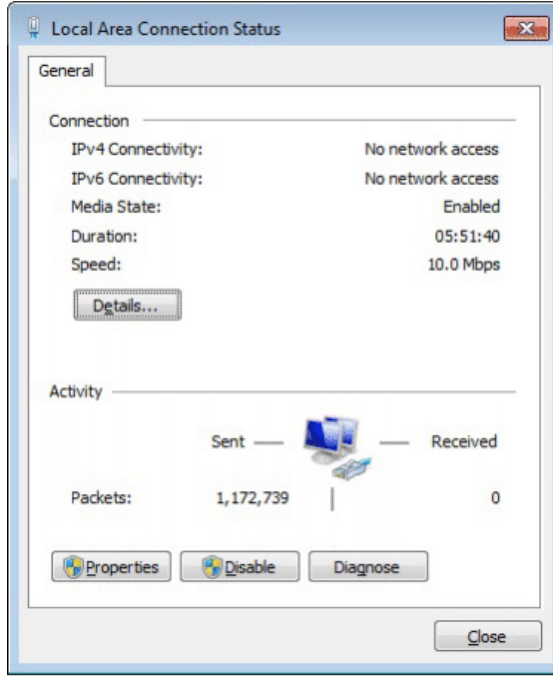
Resim 18-5: Ağ ve onay merkezi

3. Fareyle doğru LAN kartını işaretleyin ve üzerine çift tıklayarak açın



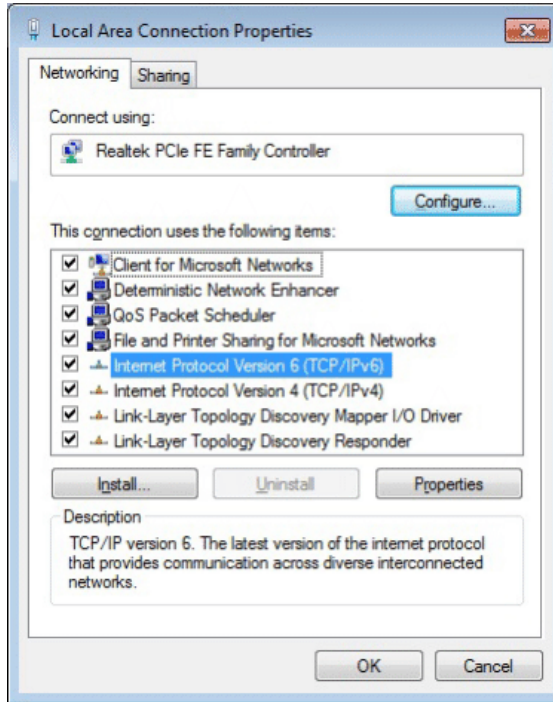
Resim 18-6: Ağ bağlantıları

4. LAN bağlantısının durumu penceresinde **Özellikler** ayarına tıklayın
Açmak için yönetici hakları gereklidir.



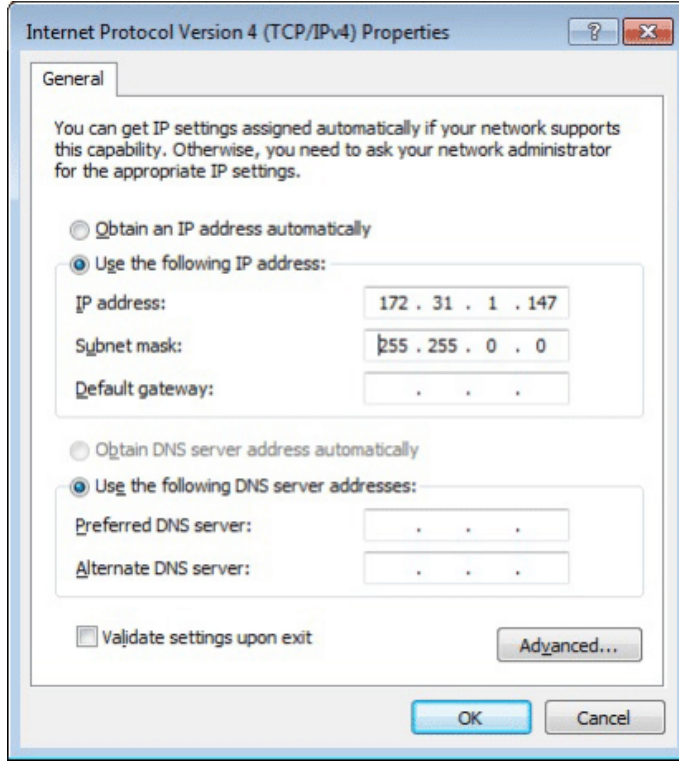
Resim 18-7: LAN bağlantısının durumu

5. Ağ sekmesinde **İnternet protokolü sürüm 4 (TCP/IPv4)** seçeneğini işaretleyin. **Özellikler** seçeneğini açın.



Resim 18-8: Ağ bağlantısı özellikleri penceresi

6. **Özellikler** butonuna tıklayın.



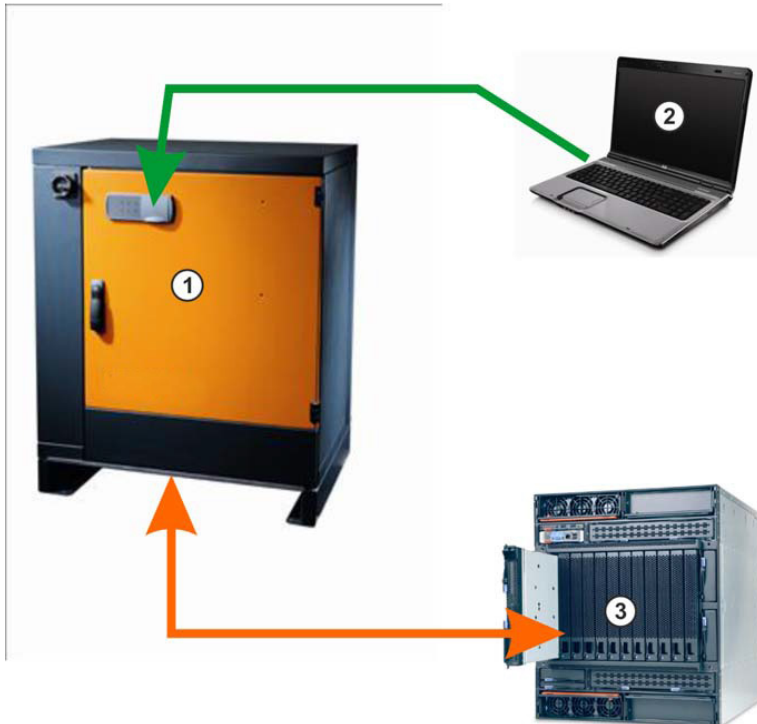
Resim 18-9: Alternatif konfigürasyon özellikleri penceresi

7. Radyo butonu **Kullanıcı tanımlı** ayarında olmalıdır.
8. IP adresini ve alt ağ maskesini girin
9. Tüm pencereleri **Tamam butonlarıyla** kapatın.

DHCP üzerinden Koşullar

IP'nin verilmesi

- Kumanda dolabı bir IT ağına bağlıdır.
- Yönetici hakları ve boş bir ağ bağlantı noktası olan bir dizüstü bilgisayar mevcut.
- CSP'de ağ yuvası, KUKA PC'nin dahili KUKUA Line Interface arabirimine bağlıdır (dahili Switch üzerinden de mümkündür).
- IT (DHCP) üzerinden otomatik olarak adres verilir.



Resim 18-10: DHCP üzerinden IP

izlenecek yöntem

1. Dizüstü bilgisayarı kumanda dolabına bağlayın (>>> Resim 18-10)
2. Yukarıda belirtilen koşullar yerine getirildiğinde, kumanda dolabına ve dizüstü bilgisayara IT sunucudan (3) otomatik olarak bir IP adresi atanır.
3. Kumanda dolabından alınan IP adresini tanılama monitöründe kontrol edin. KUKA menüsü tanılama tuşu tanılama monitörü Virtual5 veya Virtual 6

Ad	Değer	Birim
Interface-ID	virtual5	--
TX-Counter	8	--
TX-Packeterrors	0	--
RX-Counter	93	--
RX-Packeterrors	0	--
Link-status	--	--
IP-Address	172.31.1.147	--
MAC-address	00:19:99:F4:0B:31	--
Subnetmask	255.255.0.0	--
Protocols	IPSTACK IPv4,IPSTACK ARP,IPSTACK RARP,IPSTACK 802.1Q VLAN	--

Resim 18-11: Tanılama monitörü

4. Dizüstü bilgisayarda alınan IP adresini kontrol edin.
5. WorkVisual'ı açın

18.3 WorkVisual İle Proje Yönetimi

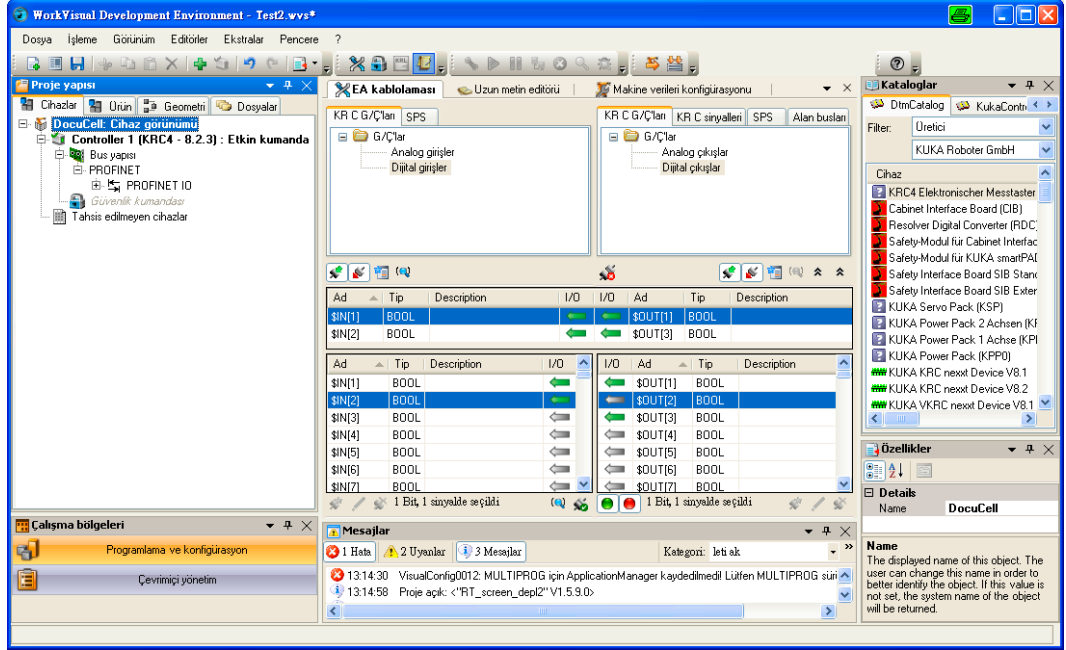
Proje seyri

1. Projeleri robot kontrol sisteminden WorkVisual'a aktarma (>>> 18.3.1 "WorkVisual ile proje açma" Sayfa 214)
2. Projeyi değiştirme, örn. KRL programları (>>> 18.4 "KRL programlarını WorkVisual ile işleme" Sayfa 229)
3. Proje karşılaştırma (mergen) (>>> 18.3.2 "Projeleri WorkVisual ile karşılaştırma" Sayfa 219)
4. Projeyi WorkVisual'dan robot kontrol sistemine yükleme (deployen) (>>> 18.3.3 "Projeyi robot kumandasına aktarma (yükleme)" Sayfa 223)
5. Proje etkinleştirme (onaylama) (>>> 18.3.4 "Projeyi robot kumanda sisteminde etkinleştirme" Sayfa 226)

18.3.1 WorkVisual İle Proje Açma

WorkVisual kısa tarifi

WorkVisual yazılım paketi, KR C4 kontrollü robot hücreleri için mühendislik ortamıdır. Aşağıdaki işlemlere sahiptir:



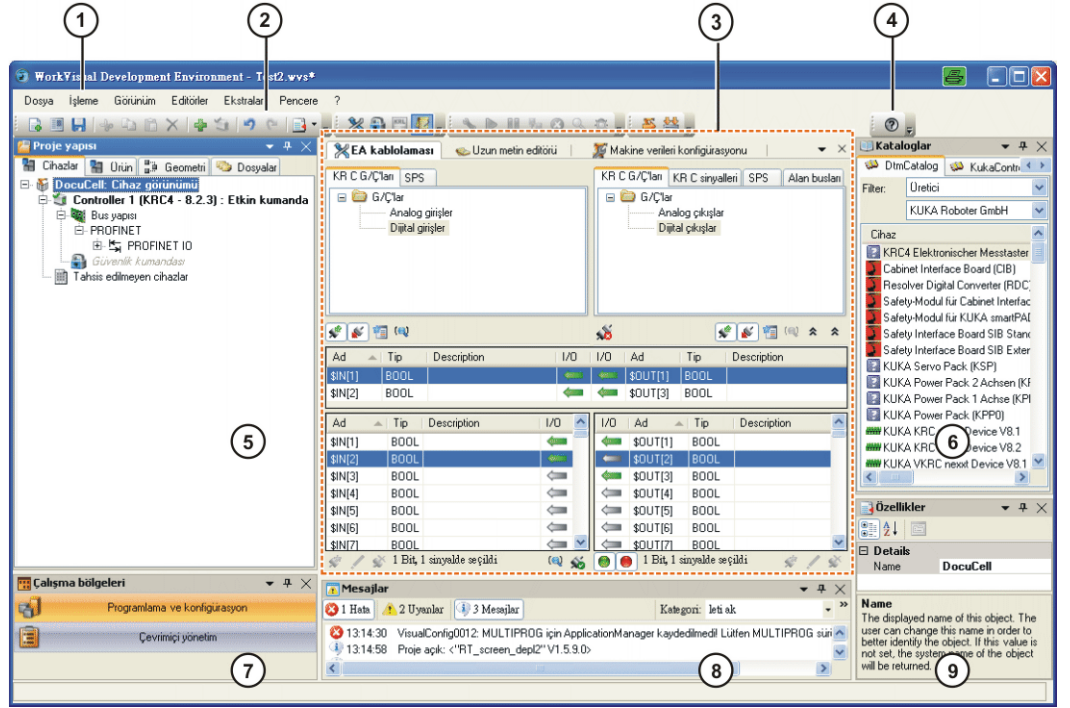
Resim 18-12: WorkVisual kullanıcı arayüzü

- Projeleri robot kumandasından WorkVisual'a aktarma:
Ağ bağlantısı bulunan her robot kumandasında istenen herhangi bir proje seçilip WorkVisual'a aktarılabilir. Bu işlem, ilgili proje ilgili PC'de henüz mevcut değilse de mümkündür.
- Projeyi başka bir proje ile karşılaştırma ve gerektiğinde farkları devralma:
Bir proje başka bir projeye karşılaştırılabilir. Bu, robot kumandasındaki bir proje veya yerel olarak kaydedilen bir proje olabilir. Kullanıcı, her fark için durumu güncel projede olduğu gibi bırakmak mı ya da diğer projedeki durumu devralmak mı istediğine karar verebilir.
- Projeleri robot kumandasına aktarma.
- Alan busları kurma ve kablolama.
- Güvenlik konfigürasyonunu işleme.
- Robotları çevrimdışı programlama.
- Uzun metinleri yönetme.
- Tanılama işlevi.
- Robot kumandasına ilişkin sistem bilgilerini çevrimiçi görüntüleme.
- Trace'leri konfigüre etme, kayıtları başlatma, Trace'leri değerlendirme (osiloskop ile).

WorkVisual kullanıcı arayüzünün yapısı ve işlevi

Kullanım arayüzünde öğelerin tümü varsayılan ayar olarak görünür değildir, ihtiyaca göre ekrana getirebilir veya ekrandan kaldırabilirsiniz.

Burada gösterilenlerin haricinde de pencereler ve editörler mevcuttur. Bunları Pencereler ve Editörler menü noktası üzerinden görüntüleyebilirsiniz.

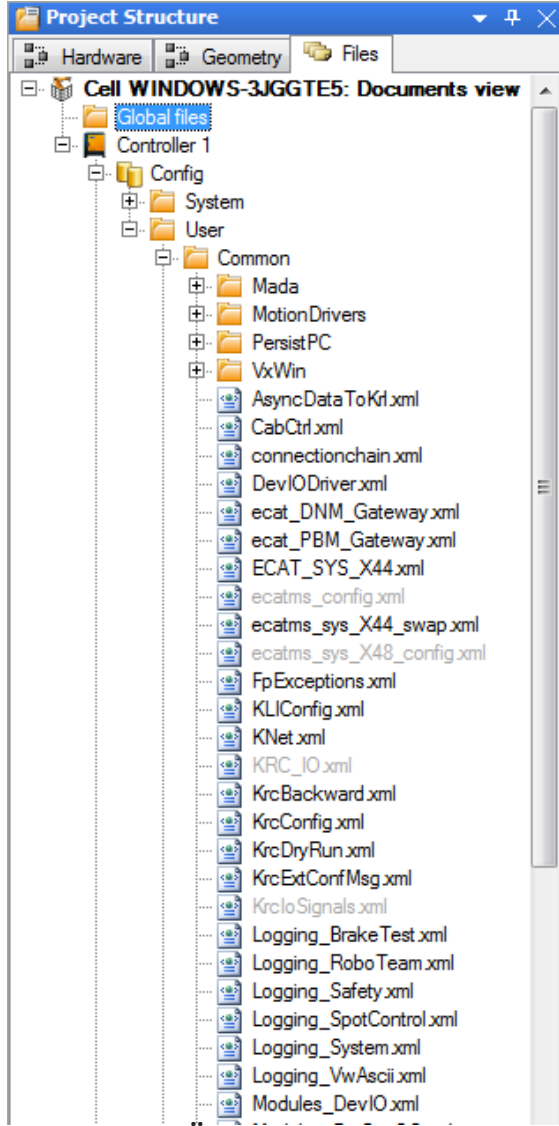


Resim 18-13: Kullanıcı arayüzüne genel bakış

Poz.	Açıklama
1	Menü çubuğu
2	Buton çubuğu
3	Editör alanı Açık olan bir editör burada görüntülenir. Buradaki ömekte olduğu gibi aynı anda birden fazla editör açık olabilir. Bu durumda üst üste yerleştirilirler ve kayıt kartları vasıtasıyla seçilebilirler.
4	Yardım butonu
5	Proje yapısı penceresi
6	Kataloglar penceresi Eklenen tüm kataloglar bu pencerede görüntülenir. Kataloglar için-deki öğeler sürük ve bırak işleyle pencerede Cihazlar veya Geometri sekmesine eklenebilir.
7	Çalışma bölgeleri penceresi
8	Mesajlar penceresi
9	Özellikler penceresi Bir nesne seçildiğinde nesnenin özellikleri bu pencerede görüntülenir. Özellikler değiştirilebilir. Gri alanlardaki münferit özellikler değiştirilemez.

Proje yapısı penceresi

- Proje yapısı sekmesi



Resim 18-14: Örnek: Otomatik üretilen dosyalar gri renkli

- **Cihazlar:**

Cihazlar sekmesinde tüm cihazlar yer alır. Burada münferit cihazlar bir robot kumandasına atanabilir.

- **Geometri:**

Geometri sekmesinde, harici kinematikler için makine verileri (örn. ilave akslar gibi) oluşturulur.

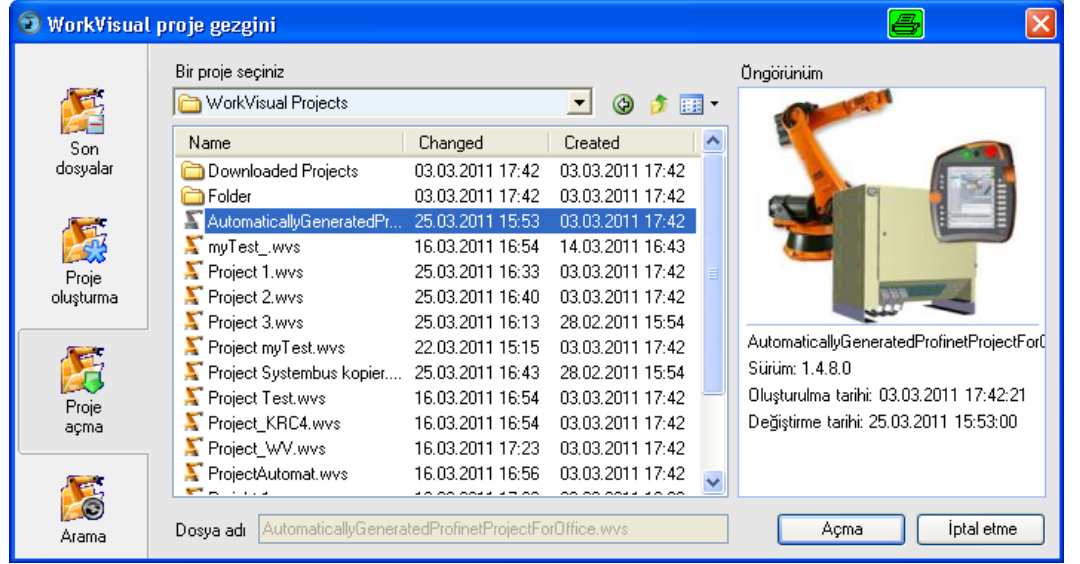
- **Dosyalar:**

Dosyalar sekmesinde projeye ait program ve konfigürasyon dosyaları yer alır.

Dosya adlarının renkli görünümü:

- Otomatik üretilen dosyalar (Kod üret fonksiyonu ile): Gri
- WorkVisual'a manuel eklenen dosyalar: mavi
- Bir robot kumandasından WorkVisual'a aktarılmış olan dosyalar: siyah

WorkVisual proje gezgini

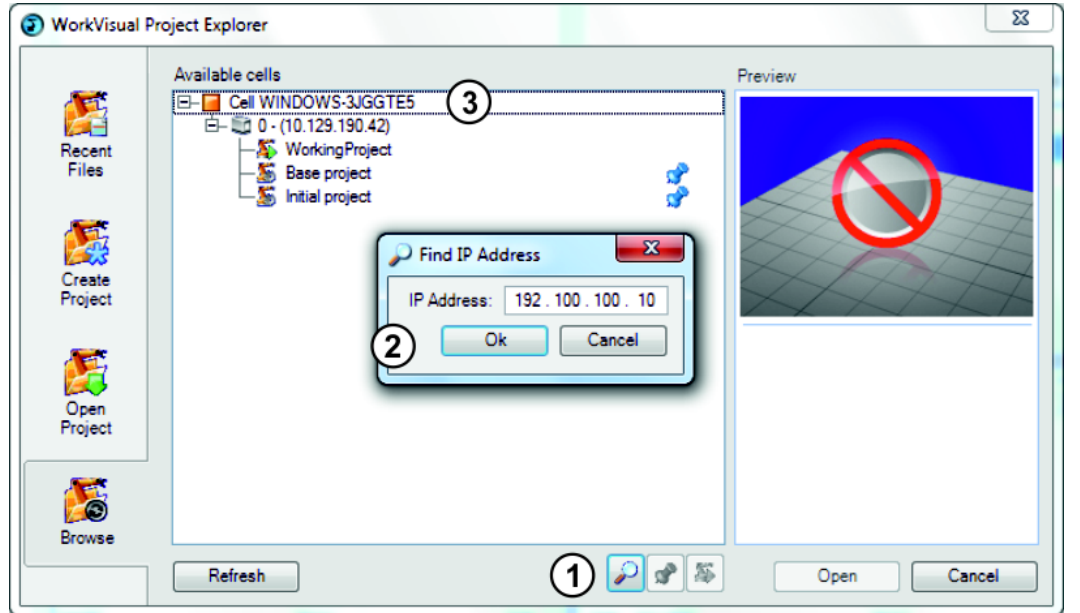


Resim 18-15: Proje gezgini

Son dosyalar, en son kullanılan dosyaları görüntüler

- **Proje oluşturma** üretildi
- Yeni, boş bir proje açar.
- Yeni bir projeyi şablon yardımıyla oluşturmanızı sağlar.
- Yeni bir projeyi mevcut bir projeyi şablon olarak kullanarak oluşturmanızı sağlar.
- **Proje aç** seçeneği mevcut projeleri açmak için kullanılır
- **Ara** seçeneği, robot kumanda ünitesinden bir projeyi yüklemek için kullanılır

Not



Resim 18-16: Find IP Address

Bazen ağ bağlantısı olmasına rağmen yine de arama penceresinde hiç kumanda görüntülenmeyebilir. Aşağıda izlenecek yöntemle çözüm sağlanır:

- WorkVisual Explorer penceresinde **Büyüteç** (1) butonuna tıklayın.
- **Find IP Address** (2) penceresi açılır.
- İstedığınız kumanda dolabının IP adresini girin.
- IP adresini **OK** (2) ile devralın.
- Kumanda artık **Mevcut hücreler** (3) penceresinde görünür.

WorkVisual ile Ağ bağlantısı bulunan her robot kumandasında bir proje seçilebilir ve **Work-proje yüklemede** Visual'a aktarılabilir.

izlenecek yöntem Bu işlem, ilgili proje ilgili PC'de henüz mevcut değilse de mümkündür.

Proje, **Dosyalarım\WorkVisual Projects\Downloaded Projects** dizinine kaydedilir.

1. **Dosya > Proje ara** menü sırasını seçin. **Proje gezgini** açılır. Solda **Ara** sekmesi seçilidir.
2. **Mevcut hücreler** alanında istenen hücrenin düğümünü açın. Bu hücreye ait tüm robot kumandaları görüntülenir.
3. İstlenen robot kumandasının düğümünü açın. Tüm projeler görüntülenir.
4. İstedığınız projeyi işaretleyin ve **Aç** üzerine tıklayın. Proje, WorkVisual'da açılır.

18.3.2 Projeleri WorkVisual İle Karşılaştırma

Açıklama

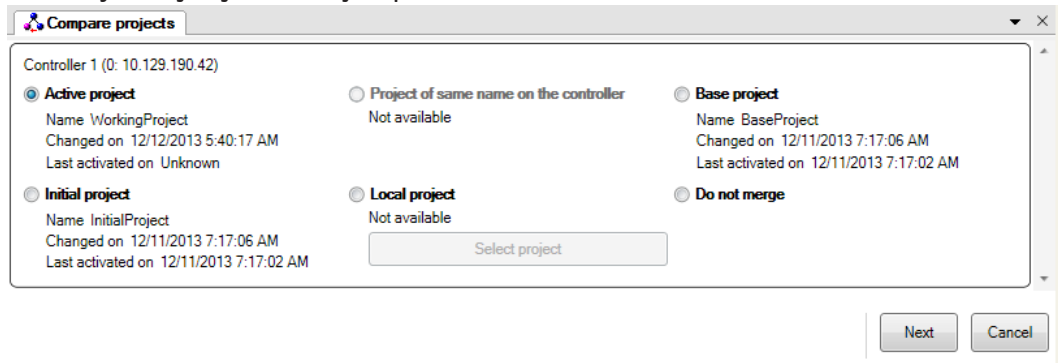
WorkVisual içindeki bir proje başka bir projeye karşılaştırılabilir.

- Bu, robot kumanda ünitesindeki bir proje veya yerel olarak kaydedilen bir proje olabilir.
- Farklar anlaşılır şekilde listelenir ve ayrıntılı bilgiler görüntülenebilir.
- Kullanıcı her fark için ayrı ayrı karar verebilir,
 - Durumu güncel projede olduğu gibi bırakıp bırakmamaya
 - Ya da durumu diğer projeden almaya

Proje karşılaştırma prensibi

Proje birleştirme

- **Proje karşılaştırma** seçim penceresi



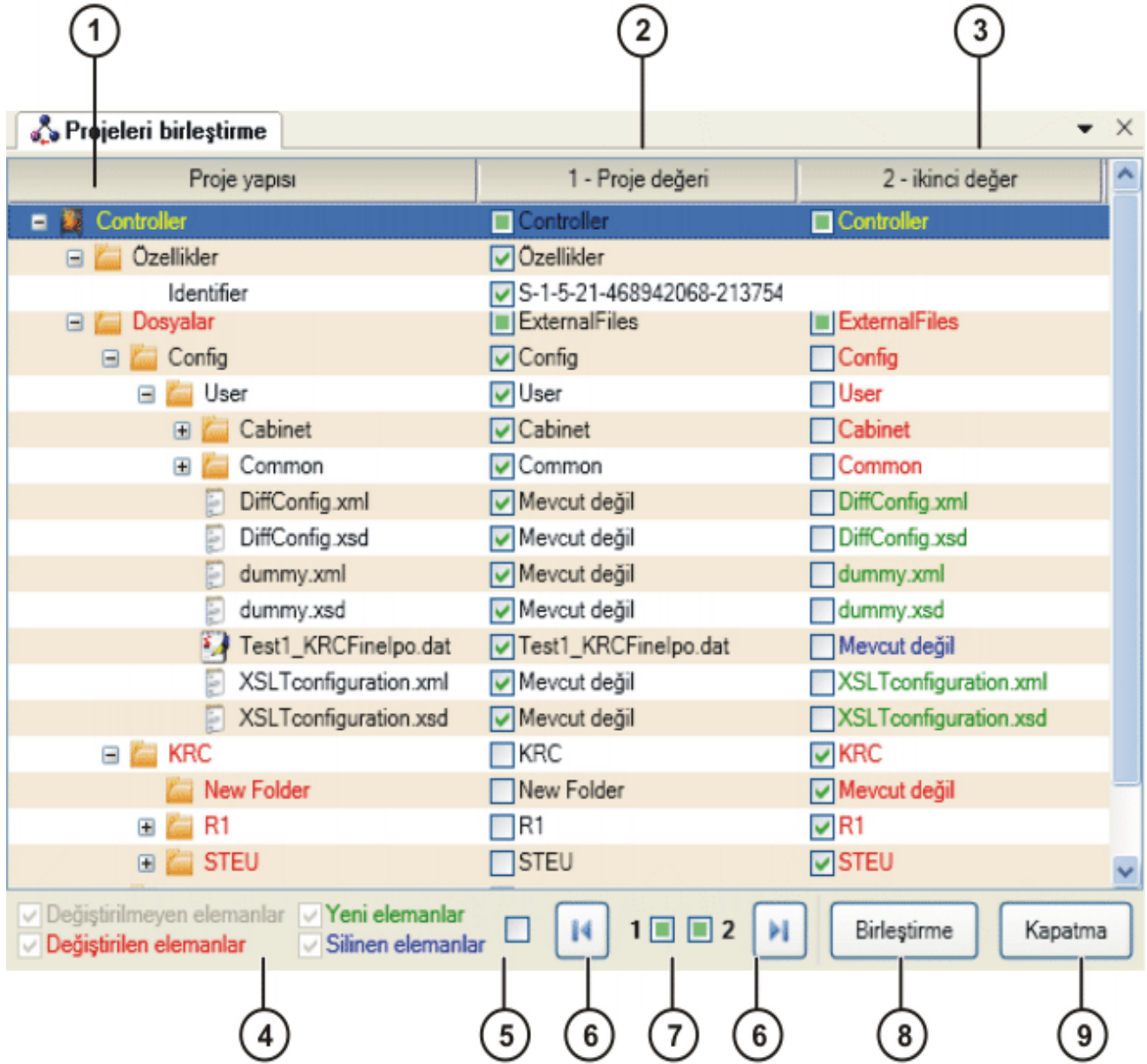
Resim 18-17: "Karşılaştırma" için proje seçme

Etkin proje

- Kumanda ünitesinde aynı adlı proje (sadece ağ bağlantısı varsa Mümkündür)
- Base proje
- Başlangıç projesi
- Yerel proje (dizüstü bilgisayardan)

Karşılaştırma:

Projeler arasındaki farkları gösteren bir genel bakış görüntülenir. Her bir fark için hangi durumun devralınacağını seçilebilir.



Resim 18-18: Örnek: Farklara genel bakış

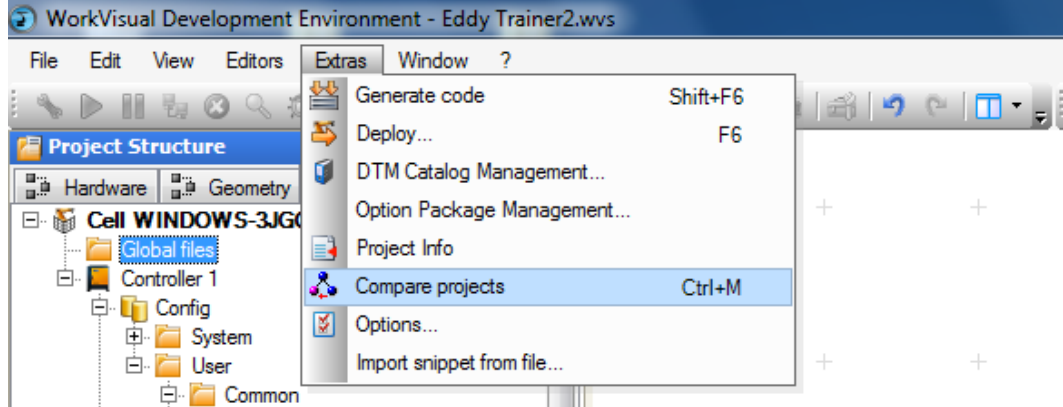
Poz.	Açıklama
1	Robot kontrol sistemi için düğüm. Çeşitli proje alanları alt düğümlerde gösterilir. Karşılaştırmaları görüntülemek için düğümleri açın. Birden fazla robot kumandası varsa bunlar alt alta listelenir. <ul style="list-style-type: none"> ■ Bir satırda işaret çentiğini devralınacak değerin yanına koyun. (Alternatif: Sayfanın altındaki onay kutularını kullanın.) ■ Kullanılamaz yanında bir işaret olması, öğenin devralınmayacağını veya eğer önceden mevcutsa, projeden silineceği anlamına gelir. ■ Bir düğümde işaret çentiği konulursa otomatik olarak tüm alt düzey elemanlara da işaret çentiği konulmuş olur. Bir düğümdeki işaret çentiği kaldırılırsa otomatik olarak tüm alt düzey elemanlardaki işaret çentikleri de kaldırılmış olur. Alt düzey elemanlar tek başına da işlenebilir. ■ Kutucuğun dolu olması şu anlama gelir: Alt düzey elemanlardan en az biri seçili, fakat tümü değil.
2	WorkVisual'da açık olan projedeki durum.
3	Karşılaştırılan projedeki durum.
4	Farkların çeşitli tiplerini ekrana getirmek ve ekrandan kaldırmak için filtre.
5	TRUE: Genel bakışta işaretlenen satıra ilişkin ayrıntılı bilgiler görüntülenir.
6	Geri oku: Görüntüdeki fokus önceki farka atlar. İleri oku: Görüntüdeki fokus sonraki farka atlar. Kapalı olan düğümler bu esnada otomatik olarak açılır.
7	Onay kutuları, fokusun o anda bulunduğu satırın durumunu görüntülerler. İşaret çentikleri doğrudan satır içinde konulmak yerine burada da konulabilir veya kaldırılabilir.
8	Seçilen değişiklikleri açılan projeye devralır.
9	Projeleri birleştir penceresini kapatır.

Renklerin açıklaması:

Sütun	Açıklama
Proje yapısı	Her eleman, seçildiği sütunda sahip olduğu renk ile gösterilir.
WorkVisual	Tüm elemanlar siyah görüntülenir.
Seçilen değer	<ul style="list-style-type: none">Yeşil: Açılan projede mevcut olmayan fakat karşılaştırılan projede mevcut olan elemanlar.Mavi: Açılan projede mevcut olan fakat karşılaştırılan projede mevcut olmayan elemanlar.Kırmızı: Tüm diğer öğeler. Bunlar arasında birden fazla renkte eleman içeren üst düzey elemanlar da yer almaktadır.

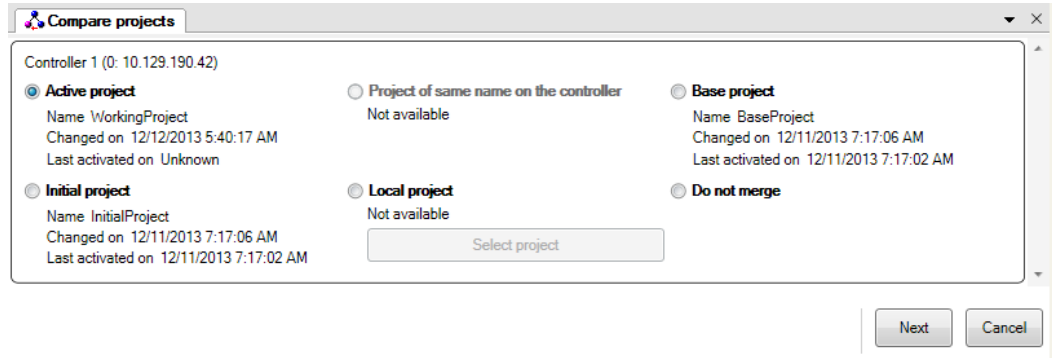
Proje karşılaştırmasında izlenecek yöntem

1. WorkVisual içerisinde **Ekstralar > Projeleri karşılaştırma** menü sırasını takip edin. **Projeleri karşılaştırma** penceresi açılır.



Resim 18-19: Projeleri karşılaştırma

2. Güncel WorkVisual projesinin karşılaştırılacağı projeyi seçin, örneğin reel robot kumandasında yer alan aynı ada sahip proje.



Resim 18-20: "Karşılaştırma" için proje seçme

3. **Devam** üzerine tıklayın. İlerleme çubuğu görüntülenir. Proje birden fazla robot kumanda ünitesi içeriyorsa her biri için ayrı bir çubuk görüntülenir.
4. İlerleme çubuğu dolduğunda ve durum **Birleştirme için hazır** görüntülediğinde: **Farkları görüntüleme** üzerine tıklayın. Projeler arasındaki farkları gösteren bir genel bakış görüntülenir.
Herhangi bir fark saptanmazsa bu durum mesaj penceresinde görüntülenir.8. adım ile devam edin. Daha sonra başka işleme gerek yoktur.
5. Her bir fark için hangi durumun devralınacağını seçin. Bunun bir geçitteki tüm farklar için yapılmasına gerek yoktur.Eğer uyuyorsa varsayılan seçim de bırakılabilir.
6. Değişiklikleri WorkVisual'a devralmak için **Birleştirme** üzerine basın.
7. 5. ve 6. adımları istediğiniz sıklıkta tekrarlayın. Bu sayede farklı alanlar daha sonra tekrar tekrar işlenebiliyor.

Başka farklar yoksa aşağıdaki mesaj görüntülenir: **Başka fark yok.**

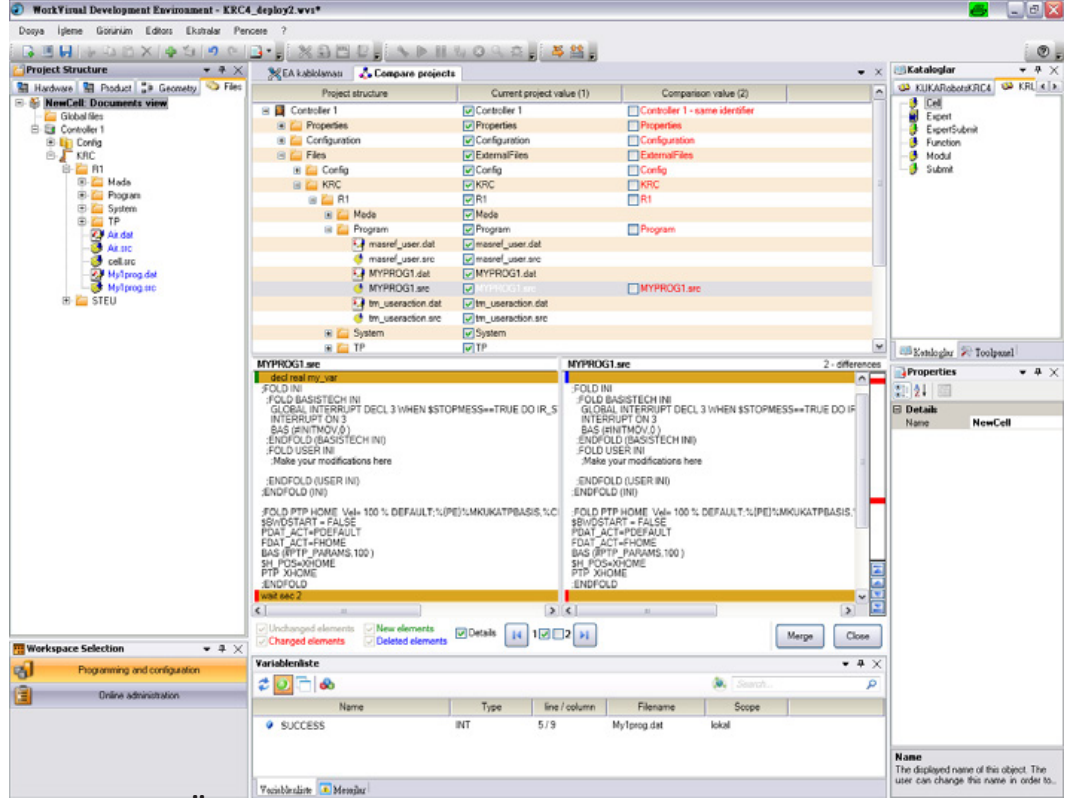
8. Projeleri karşılaştırma penceresini kapatın.

9. Robot kumandasında proje içerisinde ilave aksların parametreleri değiştirilmişse bunların şimdi WorkVisual'da güncellenmesi gerekir:

- ilave aks için **Makine verileri konfigürasyonu** penceresini açın.
- **Genel aksa özgü makine verileri** alanında makine verilerinin içe aktarılmasına ilişkin tuşa basın.
Veriler güncelleniyor.

10. Projeyi kaydedin.

Proje karşılaştırma örneği



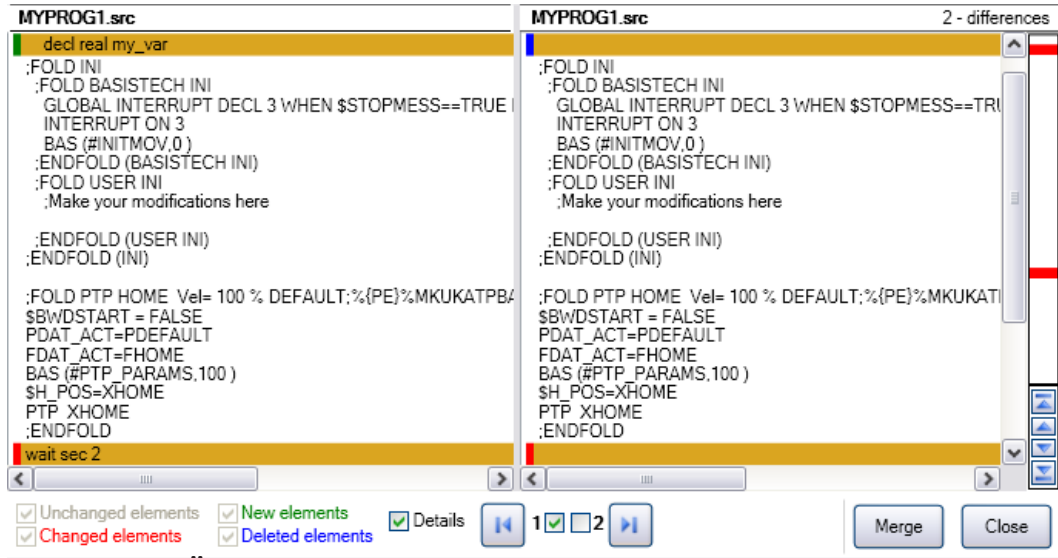
Resim 18-21: Örnek: Proje karşılaştırmaya genel bakış

Dosyanın/dosyaların hangi durumunun üstlenileceği kararı.

Project structure	Current project value (1)	Comparison value (2)
Controller 1	<input checked="" type="checkbox"/> Controller 1	<input type="checkbox"/> Controller 1 - same identifier
Properties	<input checked="" type="checkbox"/> Properties	<input type="checkbox"/> Properties
Configuration	<input checked="" type="checkbox"/> Configuration	<input type="checkbox"/> Configuration
Files	<input checked="" type="checkbox"/> ExternalFiles	<input type="checkbox"/> ExternalFiles
Config	<input checked="" type="checkbox"/> Config	<input type="checkbox"/> Config
KRC	<input checked="" type="checkbox"/> KRC	<input type="checkbox"/> KRC
R1	<input checked="" type="checkbox"/> R1	<input type="checkbox"/> R1
Mada	<input checked="" type="checkbox"/> Mada	
Program	<input checked="" type="checkbox"/> Program	<input type="checkbox"/> Program
System	<input checked="" type="checkbox"/> System	
TP	<input checked="" type="checkbox"/> TP	
Air.dat	<input checked="" type="checkbox"/> Air.dat	<input type="checkbox"/> Not available
Air.src	<input checked="" type="checkbox"/> Air.src	<input type="checkbox"/> Not available
cell.src	<input checked="" type="checkbox"/> cell.src	
My1prog.dat	<input checked="" type="checkbox"/> My1prog.dat	<input type="checkbox"/> Not available
My1prog.src	<input checked="" type="checkbox"/> My1prog.src	<input type="checkbox"/> Not available
STEU	<input checked="" type="checkbox"/> STEU	

Resim 18-22: Örnek: Proje birleştirme

Detaylar etkinleştirildiğinde dosyaların farkları görüntülenebilir.



Resim 18-23: Örnek: Ayrıntılar aktif

18.3.3 Projeyi Robot Kumandasına Aktarma (Yükleme)


Açıklama

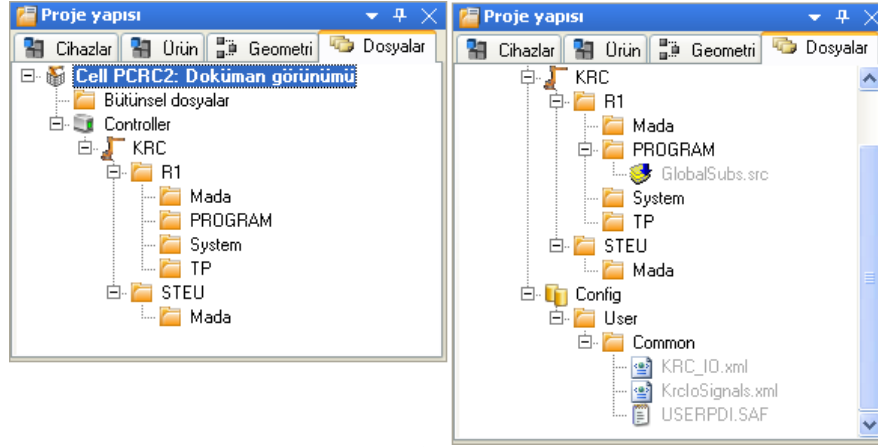
- Projede yapılan değişiklikler WorkVisual aracılığıyla kumanda ünitesine gönderilmelidir.
- Bu işlem KUKA tarafından "**Yükleme**" olarak adlandırılır.
- Bir proje, robot kumandasına aktarılırken daima önce kod üretilir.
- "**Yükleme**" için önkoşul, gerçek robot kumanda ünitesine bir ağ bağlantısının olmasıdır.

i Reel robot kumandasında, daha önce aktarılan fakat hiç etkinleştirilmeye bir proje varsa, başka bir projenin aktarılması sonucunda bu projenin üzerine yazılır.
Bir projenin aktarılması ve etkinleştirilmesi sonucunda reel robot kumandasında mevcut aynı ada sahip bir projenin üzerine yazılır (güvenlik sorunundan sonra).

Fonksiyonlar

Kod üretme

- Bu yöntemle kod ayrı olarak üretilebilir ve böylece üretme işleminin hatasız tamamlanıp tamamlanmadığı kontrol edilebilir.
- İşlem şu şekilde yapılır:
 - **Ekstralar > Kodu üret** menü dizisi
 - veya buton 
 - Kod, **Proje yapısı** penceresinde **Dosyalar** sekmesinde görüntülenir. Otomatik üretilen kod açık gri renkte görüntülenir.

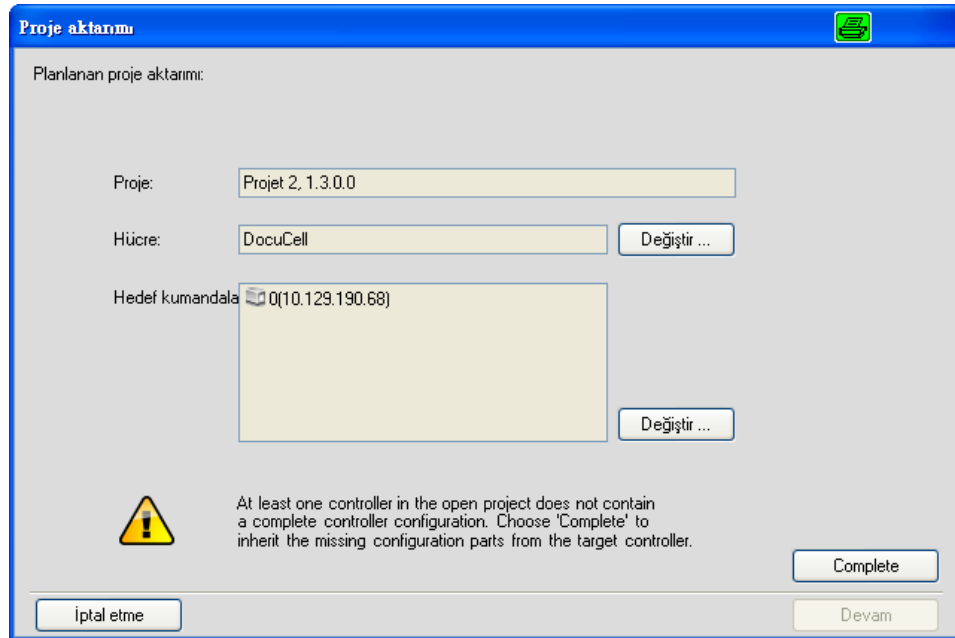


Resim 18-24: Kod üretme örneği: önce - sonra

- Kod üretilir. İşlem sona erdiğinde mesaj penceresinde aşağıdaki mesajlar görüntülenir: **Proje <"{0}" V{1}> derlendi. Sonuçlar için dosya ağacına bakınız.**

İşlem talimatı

1. Menü çubuğunda **Yükleme ...** butonuna tıklayın. **Proje aktarımı** penceresi açılır.

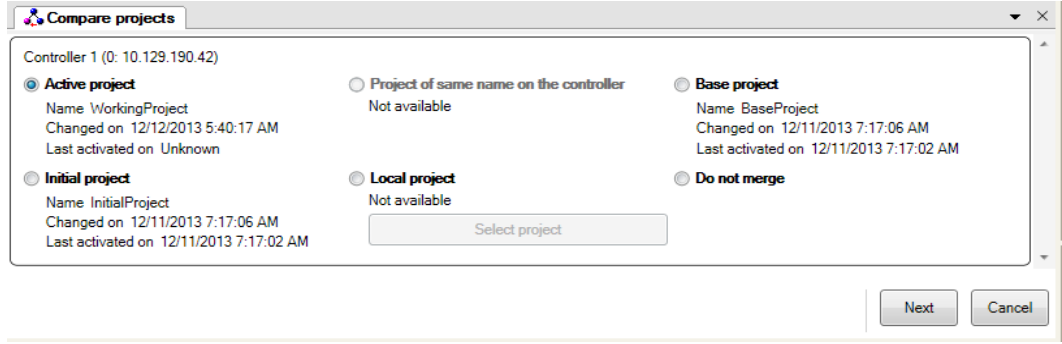


Resim 18-25: Eksik konfigürasyona ilişkin not içeren genel bakış

2. Daha önce robot kumanda ünitesinden WorkVisual'a geri aktarılmayan bir proje söz konusu ise, henüz tüm konfigürasyon dosyalarını içermiyordur (konfigürasyon dosyaları arasında makine verileri dosyaları, güvenlik konfigürasyonu dosyaları ve çok sayıda başka dosya yer almaktadır). Bu bir not ile gösterilir.

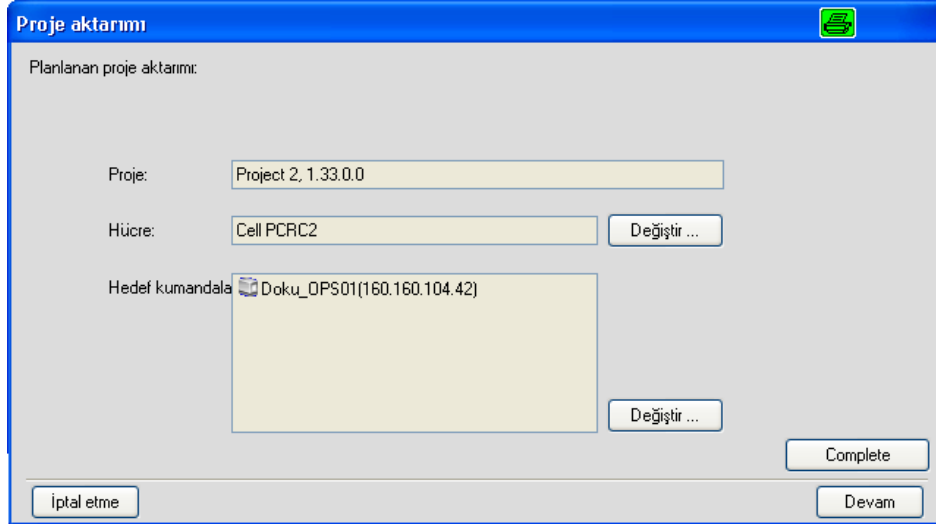
- Bu not gösterilmiyorsa: 13. adım ile devam edin.
- Bu not gösteriliyorsa: 3. adım ile devam edin.

3. **Tamamla** üzerine tıklayın. Aşağıdaki güvenlik sorusu sorulur: **Proje kaydedilmeli ve etkin kumanda geri alınır! Devam etmek istiyor musunuz?**
4. Soruya **Evet** yanıtını verin. **Projeleri birleştir** penceresi açılır.



Resim 18-26: "Karşılaştırma" için proje seçme

5. Konfigürasyon verileri devralınacak olan bir proje seçin, örneğin reel robot kumandasında etkin olan proje.
6. **Devam** üzerine tıklayın. İlerleme çubuğu görüntülenir. Proje birden fazla robot kumanda ünitesi içeriyorsa her biri için ayrı bir çubuk görüntülenir.
7. İlerleme çubuğu dolduğunda ve durum ve **Birleştirme için hazır** görüntülendiğinde: **Farkları görüntüleme** üzerine tıklayın. Projeler arasındaki farkları gösteren bir genel bakış görüntülenir.
8. Her bir fark için hangi durumun devralınacağını seçin. Bunun bir geçitteki tüm farklar için yapılmasına gerek yoktur. Eğer uyuyorsa varsayılan seçim de bırakılabilir.
9. Değişiklikleri devralmak için **Birleştirme** üzerine basın.
10. 8. ve 9. adımları istediğiniz sıklıkta tekrarlayın. Bu sayede farklı alanlar daha sonra tekrar tekrar işlenebiliyor. Başka farklar yoksa aşağıdaki mesaj görüntülenir: **Başka fark yok.**
11. **Projeleri karşılaştırma** penceresini kapatın.
12. Menü çubuğunda **Yükleme...** butonuna tıklayın. Hücre atamasına genel bakış yeniden görüntülenir. Eksik konfigürasyonu ilişkin not artık görüntülenmez.



Resim 18-27: Genel bakış

13. **Devam** üzerine tıklayın. Program üretme başlar. İlerleme göstergesi % 100'e ulaştığında program üretilmiştir ve proje aktarılmıştır.
14. **Etkinleştir** üzerine tıklayın.

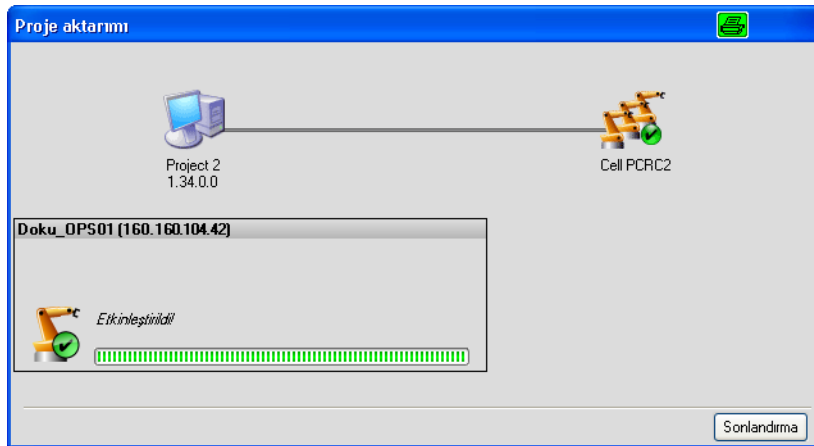
UYARI AUT ve AUT EXT işletim türlerinde proje, sadece program değişiklikleri söz konusu olduğunda güvenlik sorgulaması olmadan etkinleştirilir.

15. Sadece T1 ve T2 işletim türlerinde: KUKA smartHMI, [...] *projesinin etkinleştirilmesine izin vermek istiyor musunuz?* güvenlik sorusunu sorar. Ayrıca etkinleştirme sonucunda bir projenin üzerine yazılıp yazılmayacağı ve bunun hangi proje olduğu görüntülenir.

Önemli herhangi bir projenin üzerine yazılmıyorsa: Soruyu 30 dakika içinde **Evet** ile onaylayın.

16. Robot kumandasında etkin olan projeye karşılaştırıldığında yapılacak değişikliklere ilişkin bir genel bakış görüntülenir. **Ayrıntılar** onay kutusu aracılığıyla değişikliklere ilişkin ayrıntılar görüntülenebilir.

17. Genel bakışta *Devam etmek istiyor musunuz?* güvenlik sorusu açılır. **Evet** yanıtını verin. Proje, robot kumandasında etkinleştirilir. WorkVisual'da bir onay görüntülenir.



Resim 18-28: WorkVisual'da onaylama

18. **Proje aktarımı** penceresini **Sonlandırma** ile kapatın.

19. Robot kumanda ünitesinde soruya 30 dakika içinde cevap verilmezse proje yine de aktarılır. Fakat robot kumanda ünitesinde etkin değildir. Bu durumda proje ayrıca etkinleştirilebilir.

UYARI Robot kumandasında bir projenin etkinleştirilmesinden sonra güvenlik konfigürasyonu kontrol edilmelidir! Eğer bu kontrol yapılmazsa robot, yanlış verilerle çalıştırılabilir. Bunun sonucunda ölüm, ağır yaralanmalar ya da büyük maddi hasarlar oluşabilir.

UYARI Bir projenin etkinleştirilmesi başarısız olursa WorkVisual'da bir hata mesajı görüntülenir. Bu durumda aşağıdaki önlemlerden biri alınmalıdır:

- Ya yeniden bir proje etkinleştirin. (Aynı proje ya da başka bir proje).
- Ya da robot kumandasını soğuk başlatma ile yeniden başlatın.

18.3.4 Projeyi Robot Kumanda Sisteminde Etkinleştirme

Açıklama

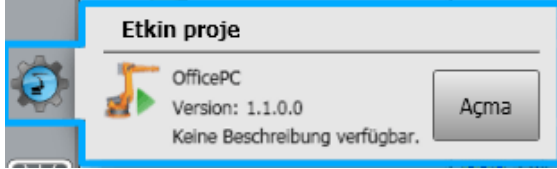
- Proje, doğrudan robot kumanda ünitesinde etkinleştirilebilir

- Proje, robot kumanda ünitesinde WorkVisual üzerinden de etkinleştirilebilir (burada daha fazla açıklanmamaktadır, bunun için WorkVisual çevrimiçi yardımını kullanın)

Proje yönetimi fonksiyonu

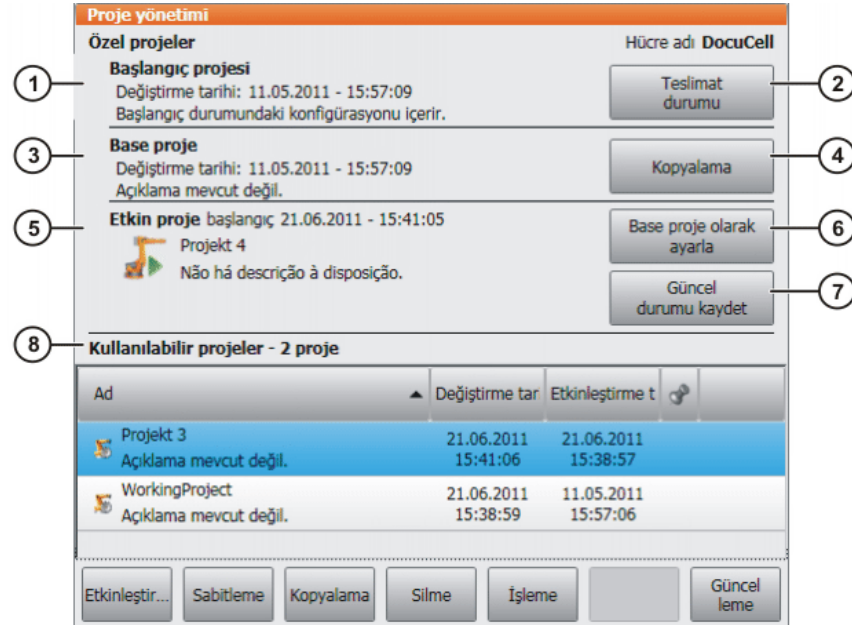
Genel Konular

- Robot kumanda sistemi, kumanda sisteminde birden çok projeyi yönetme olanağına sahiptir.
- Tüm fonksiyonlar sadece **Uzman** kullanıcı grubunda kullanılabilir
- Çağrı şu şekilde yapılır:
- **Dosya > Proje yönetimi** menü dizisi
- Kullanıcı arayüzünde **WorkVisual simgesi** tuşu ve ardından **Aç** butonu



Resim 18-29: Kullanıcı arayüzünde proje görüntüleme

Kullanım / Kumanda



Resim 18-30: Proje yönetimi penceresi

Poz.	Açıklama
1	Başlangıç projesi görüntülenir.
2	Robot kumandasının teslimat durumunu geri yükler. Uzman kullanıcı grubunda ve üzerinde mevcuttur.
3	Base proje görüntülenir.
4	Base projenin bir kopyasını oluşturur. Uzman kullanıcı grubunda ve üzerinde mevcuttur.

Poz.	Açıklama
5	Etkin proje görüntülenir.
6	Etkin projeyi Base proje olarak kaydeder. Etkin proje etkin kalmaya devam eder. Uzman kullanıcı grubunda ve üzerinde mevcuttur.
7	Etkin projenin iğnelenmiş bir kopyasını oluşturur. Uzman kullanıcı grubunda ve üzerinde mevcuttur.
8	Projelerin listesi. Etkin proje burada görüntülenmez.

- **Proje yönetimi** penceresi sıradan projelerin yanı sıra aşağıdaki özel projeleri içermektedir:

Proje	Açıklama
Başlangıç projesi	Başlangıç projesi her zaman mevcuttur. Kullanıcı tarafından değiştirilemez. Robot kumandasının teslimat sırasındaki durumunu içerir.
Base proje	Kullanıcı, etkin projeyi Base proje olarak kaydedebilir. İşlev genelde çalışan, denenmiş bir proje durumunu saklamak için kullanılır. Base proje etkinleştirilemez ama kopyalanabilir. Base proje artık kullanıcı tarafından değiştirilemez. Fakat yeni bir Base projenin kaydedilmesiyle üzerine yazılabilir (güvenlik sorusundan sonra). Tüm konfigürasyon verilerini içermeyen bir proje etkinleştirildiğinde eksik bilgiler Base projeden devralınır. Örneğin eski bir WorkVisual sürümündeki bir proje etkinleştirildiğinde bu durum söz konusu olabilir. Konfigürasyon dosyaları arasında makine verileri dosyaları, güvenlik konfigürasyonu dosyaları ve çok sayıda başka dosya yer almaktadır.

- Düğmelerin tarifi

Buton	Açıklama
Aktifleştirme	İşaretlenen projeyi etkinleştirir. İşaretli proje iğnelenmişse: İşaretli projenin bir kopyasını oluşturur. (İğnelenmiş bir projenin kendisi etkinleştirilemez, ancak kopyası etkinleştirilebilir.) Kullanıcı, kopyanın hemen etkinleştirileceğine mi yoksa şimdiye kadar kullanılan projenin mi etkin kalacağına karar verebilir. Uzman kullanıcı grubunda ve üzerinde mevcuttur.
Sabitleme	İğnelenen projeler değiştirilemez, etkinleştirilemez veya silinemez. Ancak kopyalanabilirler veya iğneleri kaldırılabilir. Yani örneğin bir projenin yanlışlıkla silinmesini önlemek için projeyi iğneleyebilirsiniz. Sadece iğnelenmemiş bir proje işaretlenmişse kullanılabilir. Uzman kullanıcı grubunda ve üzerinde mevcuttur.
Pinleri çözme	Projenin iğnelemesini kaldırır. Sadece iğnelenmiş bir proje işaretlenmişse kullanılabilir. Uzman kullanıcı grubunda ve üzerinde mevcuttur.
Kopyalama	İşaretlenen projeyi kopyalar. Uzman kullanıcı grubunda ve üzerinde mevcuttur.

Buton	Açıklama
Silme	İşaretili projeyi siler. Sadece etkin olmayan, iğnelenmemiş bir proje işaretlenmişse kullanılabilir. Uzman kullanıcı grubunda ve üzerinde mevcuttur.
İşleme	İşaretlenen projenin adının ve/veya açıklamasının değiştirilebileceği bir pencere açar. Sadece iğnelenmemiş bir proje işaretlenmişse kullanılabilir. Uzman kullanıcı grubunda ve üzerinde mevcuttur.
Güncelleştirme	Proje listesini günceller. Bu şekilde örneğin görüntü açıldığından beri robot kumandasına aktarılmış olan projeler görüntülenir.

İzlenecek yöntem

i	<p>Sınırlama: Etkinleştirme, güvenlik konfigürasyonunun İletişim parametreleri alanında değişikliklere neden oluyorsa güvenlik onarıcısı veya üstü kullanıcı grubunun seçilmiş olması gerekir.</p> <p>AUT veya AUT EXT işletim türü seçilmişse: Proje, sadece sonucunda KRL programları değişecekse etkinleştirilebilir. Proje, başka değişikliklere neden olabilecek ayarlar içeriyorsa, etkinleştirilemez.</p>
----------	---

1. **Dosya > Proje yönetimi** menü sırasını seçin. **Proje yönetimi** penceresi açılır.
2. İstedığınız projeyi işaretleyin ve **Etkinleştir** düğmesi ile etkinleştirin.
3. KUKA smartHMI, [...] *projesinin etkinleştirilmesine izin vermek istiyor musunuz?* güvenlik sorusunu sorar. Ayrıca etkinleştirme sonucunda bir projenin üzerine yazılıp yazılmayacağı ve bunun hangi proje olduğu görüntülenir.
Önemli herhangi bir projenin üzerine yazılmıyorsa: Soruyu 30 dakika içinde **Evet** ile onaylayın.
4. Robot kumandasında etkin olan projeyle karşılaştırıldığında yapılacak değişikliklere ilişkin bir genel bakış görüntülenir. **Ayrıntılar** onay kutusu aracılığıyla değişikliklere ilişkin ayrıntılar görüntülenebilir.
5. Genel bakışta *Devam etmek istiyor musunuz?* güvenlik sorusu açılır. **Evet** yanıtını verin. Proje, robot kumandasında etkinleştirilir.

⚠ UYARI	Robot kumandasında bir projenin etkinleştirilmesinden sonra güvenlik konfigürasyonu kontrol edilmelidir! Eğer bu kontrol yapılmazsa robot, yanlış verilerle çalıştırılabilir. Bunun sonucunda ölüm, ağır yaralanmalar ya da büyük maddi hasarlar oluşabilir.
----------------	--

18.4 KRL Programlarını WorkVisual İle İşleme

- Dosya kullanımı
(>>> 18.4.1 "Dosya kullanımı" Sayfa 229)
- KRL editörü kullanımı
(>>> 18.4.2 "KRL editörü kullanımı" Sayfa 235)

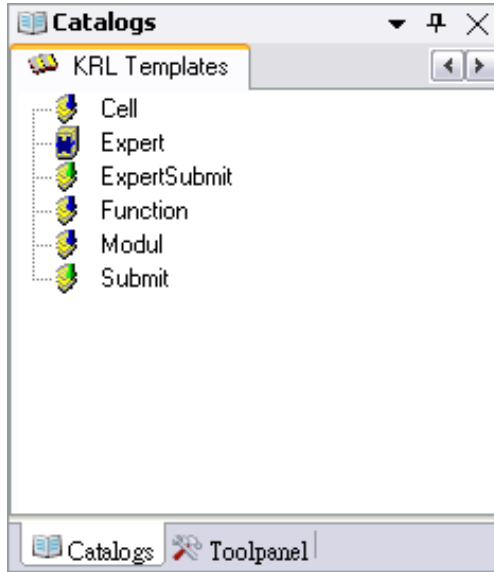
18.4.1 Dosya Kullanımı

Açıklama

- Mevcut dosyayı KRL editörüne yükleme
- Katalogdan yeni dosya ekleme
- Harici dosya ekleme

Katalogan Template prensibi

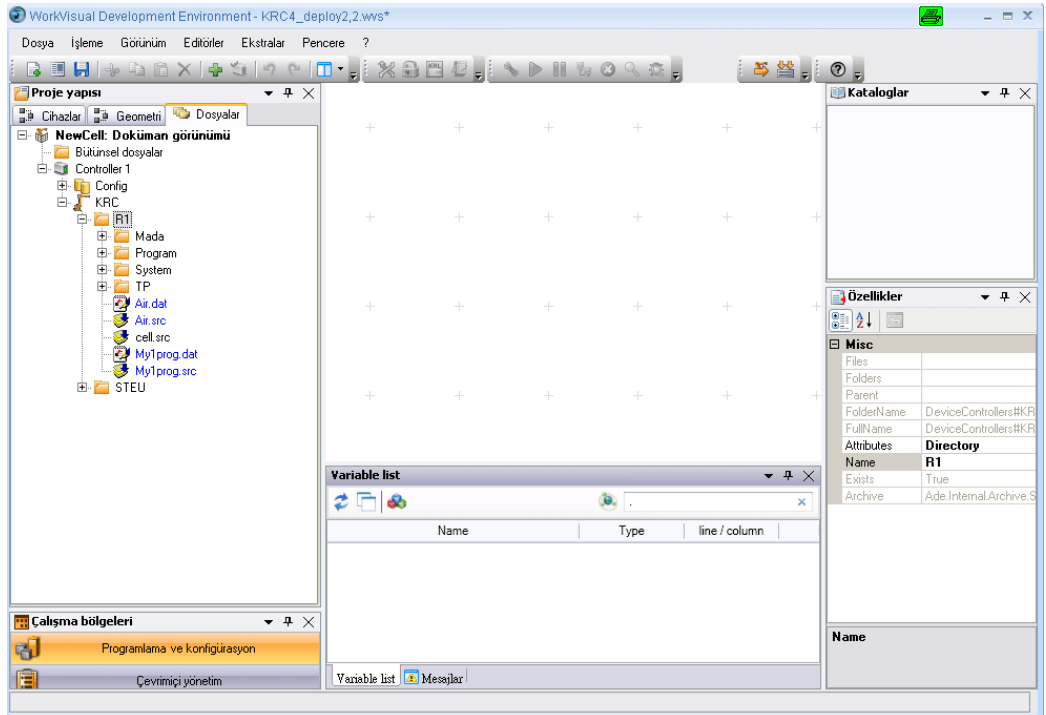
- Template'leri kullanılmadan önce ilgili katalogun yüklenmesi gerekir
- **Dosya > Katalog ekle** üzerinden ilgili Template'leri etkinleştirin
 - KRL Templates: *KRL Templates.afc*
 - VW Templates: *VW Templates.afc*
- KRL Templates katalogu



Resim 18-31: KRL Templates için katalog

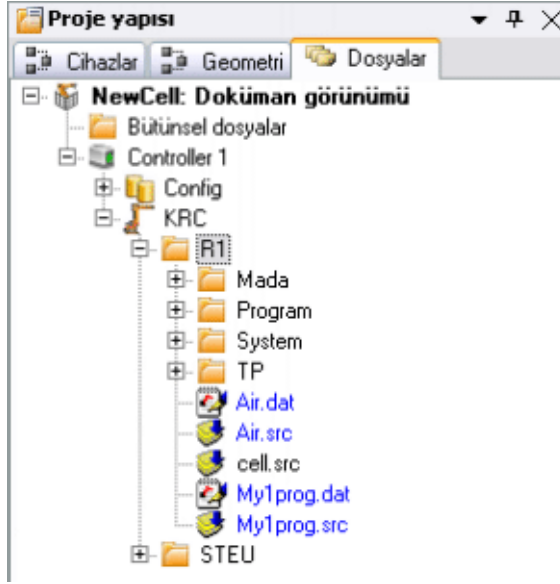
Eylem deyimleri İzlenecek yöntem Dosyayı (SRC/DAT) KRL editöründe açma

1. Proje ağacı dosya değiştirin





Resim 18-32: WorkVisual proje ağacı

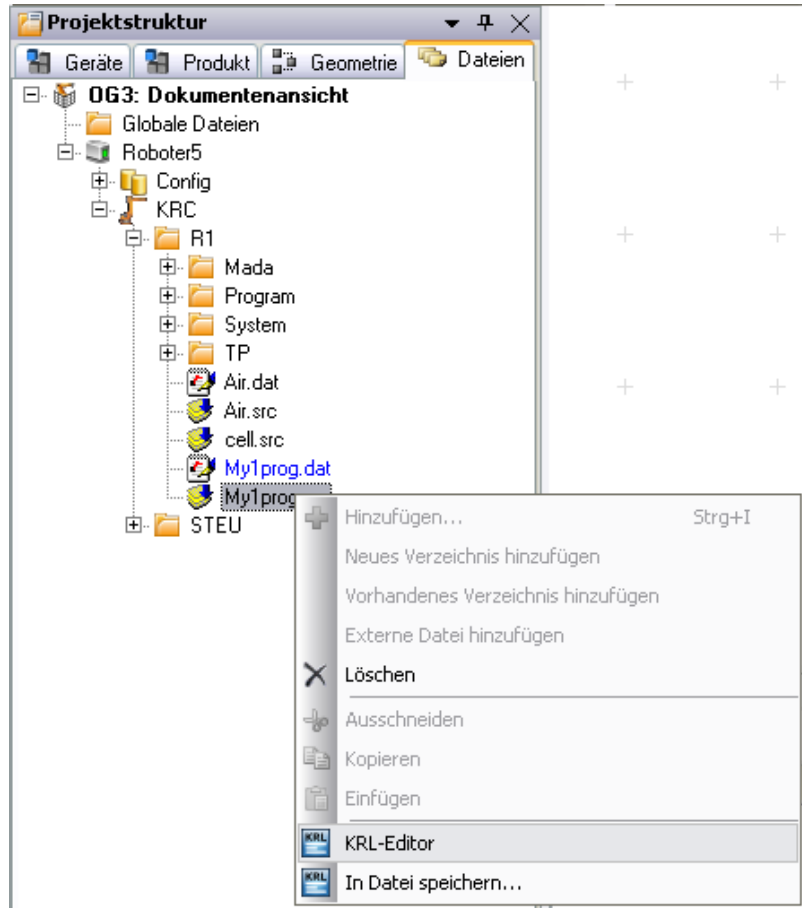
2. Dizini R1 dizinine kadar açın



Resim 18-33: WorkVisual Proje ağacı dosya yapısı (R1)


3. Dosyayı seçin ve veya

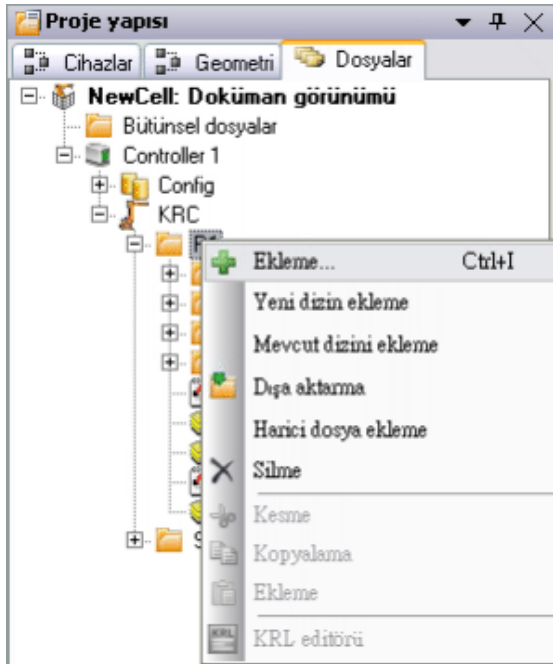
- Çift tıklayarak
- Sembol çubuğu düğmesine 
- Sağ tıklayıp bağlam menüsünde Krl editörü 



Resim 18-34: WorkVisual bağlam fare menüsü (KRL editörü)

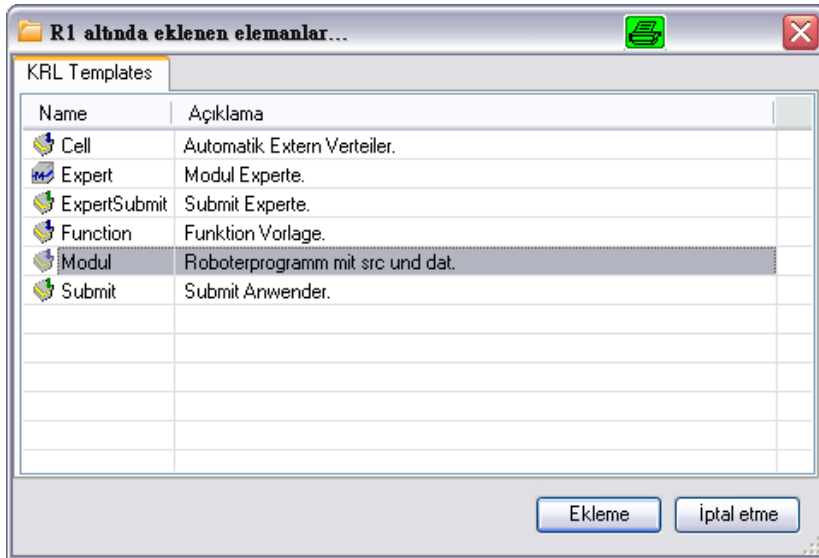
İzlenecek yöntem KRL şablonları yardımıyla dosya ekleme

1. Proje ağacı dosya değiştirin
2. Dizini R1 dizinine kadar açın
3. Yeni dosyanın oluşturulacağı klasörü seçin
4. Sağ tıklayıp bağlam menüsüne  ekleyin



Resim 18-35: WorkVisual bağlam menüsü (Ekle)

5. Şablon seçin

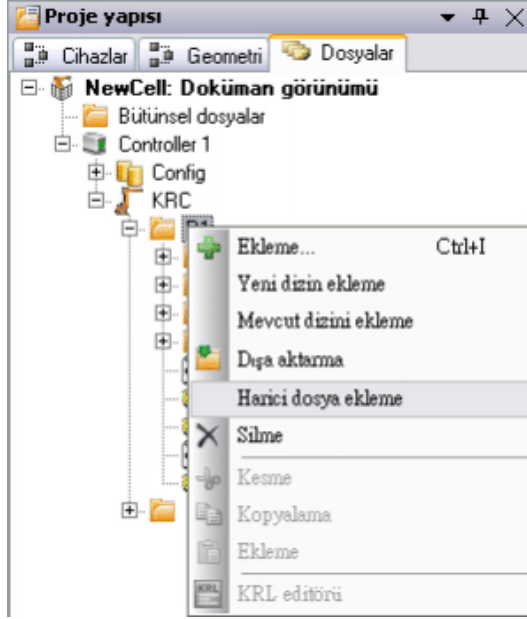


Resim 18-36: WorkVisual KRL şablonları (Templates)

6. Program adını verin

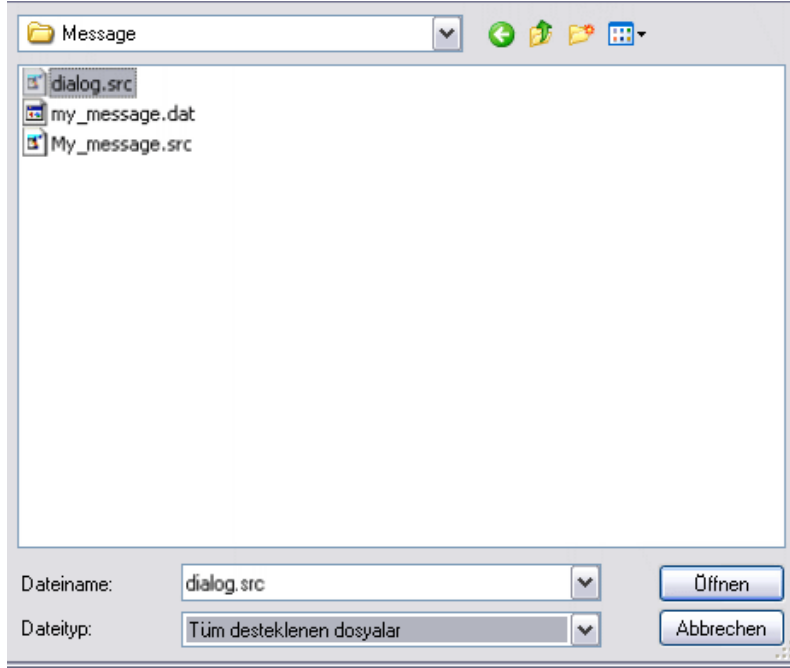
İzlenecek yöntem Harici dosya ekleme

1. Proje ağacı dosya değiştirin
2. Dizini R1 dizinine kadar açın
3. Yeni dosyanın oluşturulacağı klasörü seçin
4. Sağ tıklayıp bağlam menüsünde Harici dosya ekle



Resim 18-37: WorkVisual bağlam menüsü (Harici dosya ekle)

5. Dosyayı/dosyaları seçin ve **Aç** düğmesine basın



Resim 18-38: WorkVisual Harici dosya ekle

18.4.2 KRL Editörü Kullanımı

KRL editörü tarifi Program işleme (SRC/DAT)

- Doğrudan KRL girişiyle
- KRL deyimleri için hızlı girişle (KRL-Snippets)
- Alet kutusunda inline formlarıyla

KRL editörünün özellikleri

- Editörü yapılandırma
- KRL editöründe renk tarifi
- Hata algılama (KRL çözümleyici)
- Değişken listesi
- KRL editöründe ek işleme fonksiyonu

KRL editörünün renk tarifi

Renklerin tarifi

KRL editörü girilen kodun bileşenlerini algılar ve otomatik olarak farklı renklerde görüntüler.

Kod bileşeni	Renk
KRL anahtar kelimeleri (; FOLD ve ; ENDFOLD dışında)	orta mavi
; FOLD ve ; ENDFOLD	gri
Sayılar	koyu mavi
Strings (tırnak işaretleri "... içindeki metin)	kırmızı
Açıklamalar	yeşil
Özel karakterler	mavi yeşil
Diğer kod	Siyah

- Renk kullanımına ilişkin örnek

```

15
16 ;FOLD PTP HOME Vel= 100 % DEFAULT;%{PE}%MKUKATPBASIS
17 $BWDSTART = FALSE
18 PDAT_ACT=PDEFAULT
19 FDAT_ACT=FHOME
20 BAS (#PTP_PARAMS,100 )
21 $H_POS=XHOME
22 PTP XHOME
23 ;ENDFOLD
24

```

Resim 18-39: KRL editöründeki renkler için örnek

1	KRL anahtar sözcükleri: mavi
2	Yorum: yeşil
3	FOLD: gri
4	Diğer kod: siyah

KRL editörü hata algılama

- KRL editörü otomatik hata algılama özelliğine sahiptir
- Program kodunda tespit edilen hataların altı kırmızı çizgili olur
- Hatalar, mesaj penceresinde sadece KRL çözümleyici kategorisi seçiliyken görünür.
- KRL hataları ve kısmen yapısal hatalar algılanır (program içerisinde yanlış yerde bildirim yapılması).
- Değişkenlerdeki tapaj hataları algılanmaz.

i Hata saptama özelliği tüm hataları saptamaz. Hiç bir şeyin altının çizilmemiş olması programın hatasız olduğu anlamına gelmez.

```

22
23 ;FOLD PTP HOME Vel= 100 % DEFAULT;%{PE}%MKUKATPBASIS ,
24 $BWDSTART = FALSE
25 PDAT_ACT=PDEFAULT
26 FDAT_ACT=FHOME
27 BAS (#PTP_PARAMS,100 )
28 $H_POS=XHOME
29 PT XHOME
30 ;ENDFOLD

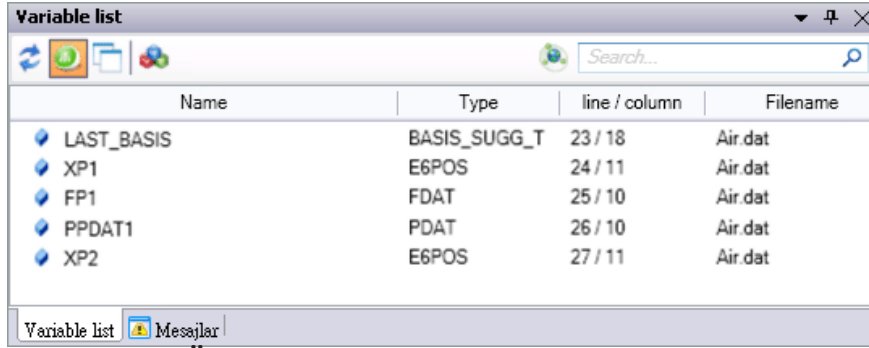
```

Resim 18-40: Hata saptama örneği

1	KRL editörü: Hatanın altı kırmızı çizilmiş
2	Mesaj penceresi: KRL çözümleyici kategorisi seçili
3	Mesaj penceresi: Satır ve sütun numarası bilgisini içeren hata açıklaması

Değişken listesi- nin fonksiyonu

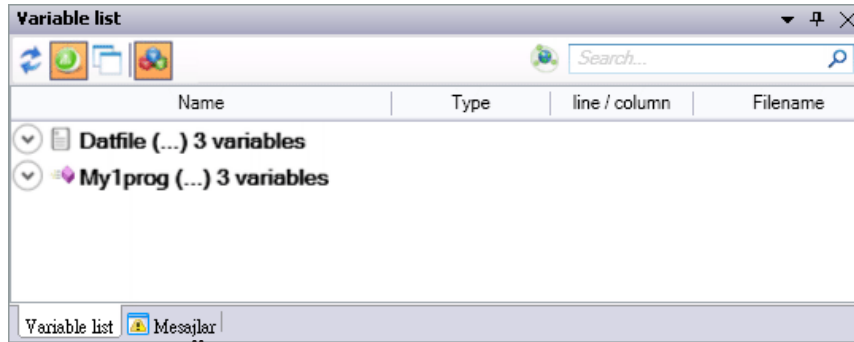
- Belirli bir dosyada açıklanmış olan tüm KRL değişkenleri bir listede bir arada görüntülenebilir.



Name	Type	line / column	Filename
LAST_BASIS	BASIS_SUGG_T	23 / 18	Air.dat
XP1	E6POS	24 / 11	Air.dat
FP1	FDAT	25 / 10	Air.dat
PPDAT1	PDAT	26 / 10	Air.dat
XP2	E6POS	27 / 11	Air.dat

Resim 18-41: Örnek değişken listesi

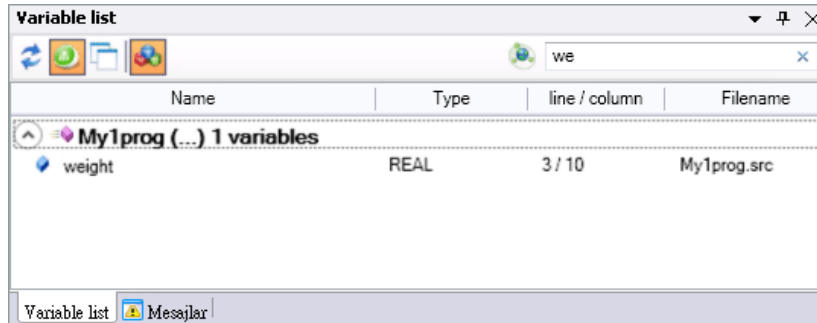
- SRC dosyalarında her zaman ilgili DAT dosyasındaki değişkenler de görüntülenir veya tersine.



Name	Type	line / column	Filename
Datfile (...) 3 variables			
My1prog (...) 3 variables			

Resim 18-42: Örnek değişken listesi (SRC ve DAT dosyalarından değişkenler)

- Değişken adını veya adın bir kısmını arama alanına girin. Arama sonucu hemen görüntülenir.



Name	Type	line / column	Filename
My1prog (...) 1 variables			
weight	REAL	3 / 10	My1prog.src

Resim 18-43: Örnek değişken listesi (arama fonksiyonu)






i Arama alanına bir şey girilmişse giriş, odak başka bir dosyaya alınsa da geçerlidir. Genel itibarıyla sadece arama alanı boş olduğunda bir dosyanın tüm değişkenleri görüntülenebilir.

- Ayar olanakları

Name	Type	line / column	Filename	Scope
SUCCESS	INT	5 / 9	Modul.dat	lokal
my_var	INT	11 / 13	Modul.dat	lokal

Resim 18-44: Değişken listesi penceresi

Bir sütuna tıklayarak listeyi ilgili sütuna göre sıralayabilirsiniz.

Buton	Açıklama
	Yeni oku
	Ağaçtaki seçilen dosya değiştiğinde daima listeyi günceller
	Güncel editör değiştiğinde listeyi günceller
	Değişkenleri, lokal alt fonksiyonlara göre gruplandırır (SRC/DAT)
	Buton basılı: Arama, tüm genel değişkenler bazında yapılır.

Değişkenleri yeniden adlandırma

Bir değişken adı tek bir işlemle kullanıldığı her noktada değiştirilebilir.

Değişken bir DAT dosyasında açıklanmış olsa da ve birden çok SRC dosyasında kullanılsa da, bunu gerçekleştirmek mümkündür.

İzlenecek yöntem

1. Herhangi bir noktada, istediğiniz değişkeni işaretleyin.
2. Sağ tıklayıp bağlam menüsünde **Yeniden adlandır** öğesini seçin.
3. Bir pencere açılır. Adı değiştirin ve **OK** ile onaylayın.

Bir değişkenin açıklamasına atlama

İzlenecek yöntem

1. İmleci, değişken adının üzerine veya ilk harfinin başına veya son harfinin sonuna konumlandırın.
2. Sağ tıklayın ve bağlam menüsünde **Açıklamaya git** öğesini seçin.

Ek işleme fonksiyonları

Sık kullanılan işleme fonksiyonları bağlam menüsünde **İşle** üzerinden çağrılabilir. Bunlar arasında:

- Kes, Ekle, Kopyala, Sil
- **Geri al, Geri yükle**
- **Ara..., Değiştir...**
- **Git...**
- **Hepsini markalama**

Mevcut diğer işleme fonksiyonları bağlam menüsünde **Gelişmiş** altında yer alır:

Gelişmiş > ...	Açıklama
Sekme kullan	İşaretili alanda boşluk karakterlerini sekmelerle değiştirir. Ön koşul: Sekme kullan konfigürasyon ayarı etkin olmalıdır.
Sekme sil	İşaretili alanda sekmeleri boş işaretlerle değiştirir.

Gelişmiş > ...	Açıklama
Girintiyi artır	İmlecin bulunduğu noktaya boş işaret veya sekme ekler.
Girintiyi azalt	İmlecin yer aldığı noktanın solundaki sekmeyi veya boş işareti siler.
Yorum durumuna getirme	İşaretili satırın başına noktalı virgül koyar.
Yorum durumundan kaldırma	İşaretili satırın başındaki noktalı virgülü siler.
Tüm foldları kapat	Güncel görüntülenen dosyanın tüm foldlarını kapatır.
Tüm foldları aç	Güncel görüntülenen dosyanın tüm foldlarını açar.

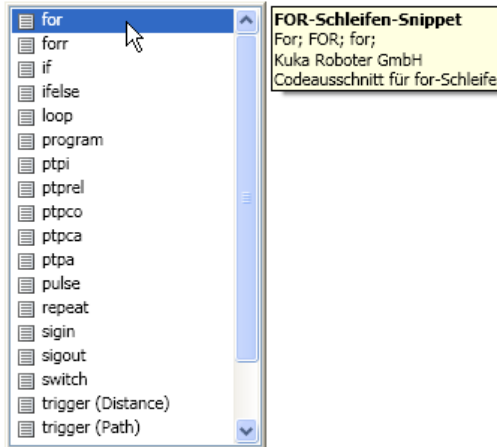
Eylem deyimleri KRL deyimleri (KRL-Snippets) için hızlı giriş

Bir FOR döngüsü programlanacak. Komple sözdizimini FOR ... = ... TO ... STEP ... girmek zorunda kalmamak için KRL-Snippet kullanılır. Sadece söz diziminin değişken kısımlarının doldurulması gerekir.

Code Snippets ile programlamada izlenecek yöntem

1. İmleci istediğiniz bir noktaya yerleştirin.

- Sağ tıklayın ve bağlam menüsünde **Kod parçası ekle** girişini seçin. Bir liste alanı ekrana getirilir. İsteddiğiniz talimata çift tıklayın.



Resim 18-45: Giriş tuşuyla devralma veya çift tıklama

- Veya: Kısaltmayı yazın ve TAB tuşuna basın.



Resim 18-46: Arama fonksiyonlu KRL kod parçacıkları (Snippets)

(Kod parçalarının yer aldığı liste alanı çağrılarak kısaltmalar saptanabilir. Fare talimatın üzerine getirildiğinde pencerenin yanında kısaltma görüntülenir. Son iki harf her zaman SD'dir ve yazılmamaları gerekir.)

2. KRL söz dizimi otomatik olarak eklenir. İlk değişken kısım kırmızı fonludur. İstediğiniz değeri girin.

```
FOR counter = start TO stop STEP 1  
ENDFOR
```

Resim 18-47: İlk deęişken kısım mavi fonludur

3. Giriş tuşu ile sonraki deęişken kısma geçin. İsteddiğiniz deęeri girin.

4. 3. adımı tüm deęişken kısımlar için tekrarlayın.

FOLDS

KRL editörünün içerięi, normal bir KRL programı gibi fold'larla yapılandırılabilir

```
32  
33 + OUTPUTS  
38  
39  
40  
41  
42  
43
```

Resim 18-48: Kapalı fold

```
32  
33 - ;fold outputs  
34   $OUT[1]=true  
35   $OUT[2]=true  
36   $OUT[3]=true  
37 -;endfold (outputs)  
38  
39
```

Resim 18-49: Açık fold

Bir Fold'u açma:

İzlenecek yöntem

- Kapalı Fold'un kutucuęunu çift tıklayın.
- Veya: Artı işaretini tıklayın.

Bir Fold'u kapatma:

İzlenecek yöntem

- Eksi işaretini tıklayın.

Tüm fold'ları açma veya kapatma:

- Bağlam menüsü: **Fold'lar > Tümünü aç** veya **Tümünü kapat**

19.1 Kısaltmalar

KR C4 kumanda sistemiyle olan çalışmalarınızı kolaylaştırmak için burada en sık kullanılan kısaltmaların bir özeti verilmektedir.

Kavram	Açıklama
CCU	Cabinet Control Unit
CCUsr	Cabinet Control Unit small robot
CIB	Cabinet Interface Board
CIBsr	Cabinet Interface Board small robot
CSP	Controler System Panel
Dual-NIC kartı	İkili ağ kartı
EDS	Electronic Data Storage (bellek kartı)
EMD	Electronic Mastering Device (eskinden EMT), robot ayarı içindir
CEM	Elektromanyetik Uyumluluk.
GBE	Gigabit EtherNet
KCB	KUKA Controller Bus
KEB	KUKA Extension Bus
KCP	KUKA Control Panel (programlama el cihazı), yeni kavram: smartPAD
KLI	KUKA Line Interface
KOI	KUKA Operator Panel Interface
KPC	KUKA kumanda sistemi PC
KPP	KUKA Power Pack
KPPsr	KUKA Power Pack small robot
KRL	KUKA Robot Language (KUKA Roboter prog- ramlama dili)
KSB	KUKA System Bus
KSP	KUKA Servo Pack
KSPsr	KUKA Servo Pack small robot
KSr	KUKA System Bus
KSI	KUKA Service Interface
LWL	Licht-Wellen-Leiter (fiber optik iletken)
OPI	Operator Panel Interface (smartPAD bağlantısı)
PMB	Power Management Board
RCD	Residual Current Device; kaçak akım rölesi (FI)
RDC	Resolver Digital Converter
SATA	Serial Advanced Technology Attachment (işlemci ve sabit disk arasında veri yolu)
SIB	Safety Interface Board
SBC	Single Brake Controll
STO	Safe Torque Off

Kavram	Açıklama
SION	Safety Input Output Node
USB	Universal Serial Bus (bilgisayarı ek cihazlara bağlamak için veri yolu sistemi)
USV	Kesintisiz güç kaynağı

19.2 Kullanılan Kavramlar

DURDURMA 0, DURDURMA 1 ve DURDURMA 2, EN 60204-1:2006 uyarınca durdurma tanımlamalarıdır.

Kavram	Açıklama
Aks aralığı	Her aksın hareket edebileceği derece veya milimetre ile ifade edilen alandır. Aks bölgesi, her bir aks için ayrıca tanımlanmalıdır.
Durma mesafesi	Durma hattı = Tepki hattı + Frenleme hattı Durma mesafesi tehlike bölgesinin bir parçasıdır.
Çalışma bölgesi	Çalışma bölgesinde manipülâtör hareket edebilir. Çalışma bölgesi münferit aks bölgelerinden oluşur.
İşletici (Kullanıcı)	Bir endüstriyel robotun işleticisi, işletme sahibi, işveren veya endüstriyel robotun kullanılmasından sorumlu olan yetkili kişi olabilir.
Tehlike bölgesi	Tehlike bölgesi, çalışma bölgesi ile durma hattını kapsamaktadır.
Kullanım süresi	Güvenlik açısından önemli bir parçanın kullanım süresi, parçanın müşteriye teslimatından itibaren başlar. Bir parçanın kullanım ömrü, parçanın, robot kumandasında veya başka bir noktada kullanılıp kullanılmadığı ile ilgili değildir, çünkü güvenlik açısından önemli yapı elemanları depolama esnasında da eskir.
KCP	Programlama el cihazı KCP (KUKA Control Panel) endüstriyel robotun kullanımı ve programlanması için ihtiyaç duyulan tüm fonksiyonlara sahiptir. KR C4 için KCP versiyonunun adı KUKA smartPAD'dir. Bu belgede genelde genel tanım "KCP" kullanılmaktadır.
KRF	Kontrollierte Roboterfahrt (kontrollü robot sürüşü) KRF, yalnızca KUKA.SafeOperation veya KUKA.SafeRangeMonitoring kullanıldığında kullanılabilen bir işletim türüdür. Robot bir denetimi ihlal edip güvenlik kumandası tarafından durdurulduğunda, KRF işletim türüyle ihlal edilen bölgeden dışarı sürülebilir.
Manipülâtör	Robot mekanikliği ve buna ait elektrik tesisatı
Koruma bölgesi	Koruma bölgesi tehlike bölgesinin dışında bulunur.
İşletimi güvenli şekilde durdurma	İşletimi güvenli şekilde durdurma, bir durma denetimidir. Robot hareketini durdurmaz, robot akslarının durup durmadığını denetler. Akslar, işletimi güvenli şekilde durdurma sırasında hareket ettirilirse güvenlik nedeniyle durma DURDURMA 0 tetiklenir. İşletimi güvenli şekilde durdurma harici olarak da tetiklenebilir. İşletimi güvenli şekilde durdurma tetiklenirse robot kumandası, alan busuna bir çıkış ayarlar. Çıkış, tetikleme sırasında tüm akslar durmuşsa ve dolayısıyla güvenlik nedeniyle durma DURDURMA 0 tetiklenirse de ayarlanır.
Güvenlik nedeniyle durma DURDURMA 0	Güvenlik kumandası tarafından tetiklenen ve gerçekleştirilen bir durdurma. Güvenlik kumandası derhal tahrikleri ve frenlerin gerilim beslemesini kapatır. Bilgi: Bu durdurma belgede, güvenlik nedeniyle durma 0 olarak tanımlanır.

Kavram	Açıklama
Güvenlik nedeniyle durma DURDURMA 1	Güvenlik kumandası tarafından tetiklenen ve denetlenen bir durdurma. Frenleme işlemi, robot kumandasının güvenlik amaçlı olmayan parçası tarafından gerçekleştirilir ve güvenlik kumandası tarafından denetlenir. Manipülâtör durmaz güvenlik kumandası, tahrikleri ve frenlerin gerilim beslemesini kapatır. Güvenlik nedeniyle durma DURDURMA 1 tetiklenirse robot kumandası, alan busuna bir çıkış ayarlar. Güvenlik nedeniyle durma DURDURMA 1 harici olarak da tetiklenebilir. Bilgi: Bu durdurma belgede, güvenlik nedeniyle durma 1 olarak tanımlanır.
Güvenlik nedeniyle durma DURDURMA 2	Güvenlik kumandası tarafından tetiklenen ve denetlenen bir durdurma. Frenleme işlemi, robot kumandasının güvenlik amaçlı olmayan parçası tarafından gerçekleştirilir ve güvenlik kumandası tarafından denetlenir. Tahrikler devrede ve frenler açık kalır. Manipülâtör durmaz işletimi güvenli şekilde durdurma tetiklenir. Güvenlik nedeniyle durma DURDURMA 2 tetiklenirse robot kumandası, alan busuna bir çıkış ayarlar. Güvenlik nedeniyle durma DURDURMA 2 harici olarak da tetiklenebilir. Bilgi: Bu durdurma belgede, güvenlik nedeniyle durma 2 olarak tanımlanır.
Güvenlik seçenekleri	Standart güvenlik fonksiyonlarına ilave güvenli denetimleri yapılandırılmayı mümkün kılan opsiyonlar için müdahale. Örnek: SafeOperation
Durdurma kategorisi 0	Tahrikler derhal kapatılır ve frenler devreye girer. Manipülâtör ve ilave akslar (opsiyonel), rotaya yakın frenler. Bilgi: Bu durdurma kategorisi belgede DURDURMA 0 olarak tanımlanır.
Durdurma kategorisi 1	Manipülâtör ve ilave akslar (opsiyonel), rotaya sadık kalarak frenler. ■ T1 işletim türü: Robot durduğu anda ve de en geç 880 ms sonra tahrikler kapatılır. ■ T2, AUT, AUT EXT işletim türleri: Tahrikler 1,5 sn sonra kapatılır. Bilgi: Bu durdurma kategorisi belgede DURDURMA 1 olarak tanımlanır.
Durdurma kategorisi 2	Tahrikler kapatılmaz ve frenler devreye girmez. Manipülâtör ve ilave akslar (opsiyonel), hatta bağlı bir frenleme rampasıyla frenler. Bilgi: Bu durdurma kategorisi belgede DURDURMA 2 olarak tanımlanır.
Sistem entegratörü (Tesis entegratörü)	Sistem entegratörleri, endüstriyel robotu emniyetli bir şekilde tesise entegre eden ve işleme alan kişilerdir.
T1	Manuel azaltılmış hız test işletim türü (<= 250 mm/sn)
T2	Manuel yüksek hız test işletim türü (> 250 mm/sn'ye izin verilmektedir)
İlave aks	Manipülâtöre ait olmayan fakat robot kumandasıyla birlikte kumanda edilen hareket aksı. Örn. KUKA lineer birim, döner tabla, Posiflex

19.3 Güvenlik KR C4'ten Alıntı

Bu bölümde KUKA Güvenlik KR C4'ten konular sadece alıntı olarak ele alınır. Burada odak noktası, planlamadır.

- Güvenlik fonksiyonlarına genel bakış**
- İşletim türü seçimi
 - Kullanıcı koruması (= ayırıcı koruyucu düzenekleri kilitlemek için bağlantı)
 - ACİL-DURDURMA tertibatı
 - Onay tertibatı


- Harici işletimi güvenli şekilde durdurma
- Harici güvenlik nedeniyle durma 1 ("KR C4 compact" kumanda sistemi çeşidinde bulunmaz)
- Harici güvenlik nedeniyle durma 2
- T1'de hız denetimi


Endüstriyel robotun güvenlik fonksiyonları aşağıdaki gereklilikleri sağlamaktadır:

- **Kategori 3** ve **Performance Level d**, EN ISO 13849-1:2008 normuna göre

Gereklilikler sadece aşağıdaki koşullarda sağlanmaktadır:

- ACİL-DURDURMA tertibatına en az 6 ayda bir basıldığında.

 **TEHLİKE** Güvenlik fonksiyonları ve koruyucu düzenekleri işler durumda olmayan endüstriyel robot, şahısların yaralanmasına veya maddi zararlara neden olabilir. Güvenlik fonksiyonları veya koruyucu düzenekler devre dışı bırakıldığında veya söküldüğünde endüstriyel robot çalıştırılmamalıdır.

 Tesis planlaması sırasında ayrıca tüm sistemin güvenlik fonksiyonları planlanmalı ve sağlanmalıdır. Endüstriyel robot, tüm sistemin güvenlik sistemine dahil edilmelidir.

Güvenlik kontrol ünitesi Güvenlik kontrol ünitesi, kontrol ünitesi PC'si içinde bir birimdir. Güvenlik ile ilgili sinyalleri ve güvenlik ile ilgili denetimleri birleştirir.


Güvenlik kontrol ünitesinin görevleri:

- Tahrikleri kapatma, frenleri etkinleştirme
- Fren rampasını denetleme
- Durmayı denetleme (durduktan sonra)
- T1'de hız denetimi
- Güvenlik ile ilgili sinyalleri değerlendirme
- Güvenlik amaçlı çıkışları ayarlama

İşletim türü seçimi

Endüstriyel robot aşağıdaki işletim türlerinde işletilebilir:

- Manuel azaltılmış hız (T1)
- Manuel yüksek hız (T2)
- Otomatik (AUT)
- Harici otomatik (AUT EXT)
- KRF

 Bir program çalışırken işletim türünü değiştirmeyin. Bir program çalışırken işletim türü değiştirilirse endüstriyel robot güvenlik durumu 2 ile durur.

İşletim türü	Kullanım	Hızlar
T1	Test işletimi, programlama ve öğretme (teach) için	<ul style="list-style-type: none"> ■ Program sorgulama: Programlanmış hız, azami 250 mm/sn ■ Elle işletim: El yöntemiyle hareket hızı, maksimum 250 mm/s
T2	Test işletimi için	<ul style="list-style-type: none"> ■ Program sorgulama: Programlanmış hız ■ Elle işletim: Mümkün değil
AUT	Üst düzey kumandası olmayan endüstriyel robotlar için	<ul style="list-style-type: none"> ■ Program işletimi: Programlanmış hız ■ Elle işletim: Mümkün değil
AUT EXT	Üst düzey bir kumanda ünitesi olan endüstriyel robotlar için, örn. PLC	<ul style="list-style-type: none"> ■ Program işletimi: Programlanmış hız ■ Elle işletim: Mümkün değil
KRF	KRF yalnızca KUKA.SafeOperation veya KUKA.SafeRange-Monitoring kullanıldığında kullanılabilir. Robot bir denetimi ihlal edip güvenlik kumandası tarafından durdurulduğunda, KRF işletim türüyle ihlal edilen bölgeden dışarı sürülebilir. Hızlar T1'de olduğu gibi	

Kullanıcı koruması

- Kullanıcı koruması sinyali, ayırıcı koruma düzeneklerinin, örn. koruyucu kapının kilitlenmesi için kullanılır.
- Bu sinyal olmadan otomatik işletim mümkün olmaz.
- Otomatik işletim sırasında sinyal kaybı olduğunda (örn. koruma kapısı açılırsa) manipülatör, güvenlik nedeniyle durma 1 ile durur.
- Manuel azaltılmış hız (T1), manuel yüksek hız (T2) ve KRF işletim türlerinde kullanıcı koruması etkin değildir.



UYARI

Sinyal kaybindan sonra otomatik işleme yalnızca koruyucu düzeneklerin kapatılması yoluyla değil, ancak ilaveten onaylama işleminin de tamamlanmasından sonra devam edilebilir. Bunun gerçekleşmesi sistem entegratörünün sorumluluğundadır. Bu önlemlerle örn. koruyucu kapının kapanması nedeniyle tehlike bölgesinde şahısların bulunması halinde otomatik işletimin yanlışlıkla devam etmemesi sağlanır.

- Onaylama, öncesinde gerçekten tehlike bölgesi kontrolü yapılabilecek şekilde tasarlanmalıdır. Buna izin vermeyen onaylamalar (örn. koruyucu düzeneğin kapanmasıyla otomatik olarak gerçekleşenler) geçersizdir.
- Buna uyulmadığı takdirde ölüm, ağır yaralanmalar veya büyük maddi hasarlar meydana gelebilir.

ACILDURDURMA Endüstriyel robotun ACİL DURDURMA tertibatı, KCP'deki ACİL DURDURMA cihazıdır.

- Tehlike yaratacak bir durum söz konusu olduğunda veya acil durumlarda bu cihaza basılmalıdır.
- ACİL-DURDURMA cihazına basıldığında endüstriyel robotun göstereceği tepkiler şunlardır:

Manipülatör ve ilave akslar (opsiyonel), güvenlik nedeniyle durma 1 ile durur.

- Çalışmaya devam edebilmek için, ACİL DURDURMA cihazının kilidi, çevrilerek açılmalıdır.
- Daima en az bir harici ACİL-DURDURMA tertibatı kurulmuş olmalıdır. Bu sayede KCP takılı değilken de ACİL-DUR tertibatının kullanılabilir olması garantilenir.

UYARI Manipülâtör ile bağlantısı olan aletler ve diğêr tertibatlar, tehlike oluşturabileceklerse, sistem ACİL DURDURMA devresine bağlanmalıdır. Bu uyarıya uyulmadığı takdirde ölüm, ağır yaralanmalar veya büyük maddi hasarlar meydana gelebilir.

Harici ACİL DURDURMA tertibatı

- Robotun hareket etmesine veya başka tehlike oluşturabilecek bir duruma yol açabilecek her kumanda istasyonunda ACİL-DURDURMA tertibatları bulunmalıdır.
- Bu tertibatların temini ve yerleştirilme yükümlülüğü sistem entegratörüne aittir.
- Daima en az bir harici ACİL-DURDURMA tertibatı kurulmuş olmalıdır. Bu sayede KCP takılı değilken de ACİL-DUR tertibatının kullanılabilir olması garantilenir.
- Harici ACİL-DURDURMA tertibatları müşteri arabirimi üzerinden bağlanır.
- Harici ACİL-DURDURMA tertibatları, endüstriyel robotun teslimat kapsamına dahil değildir.

Onay tertibatı

Endüstriyel robot onay tertibatı, KCP üzerinde bulunan onay şalterleridir. KCP üzerinde 3 adet onay şalteri vardır. Onay şalterlerinin 3 konumu vardır:

- Basılı değil
- Orta konum
- Basılı (panik konumu)

Manipülâtör, test işletim türlerinde ve KRF işletim türünde, sadece onay şalteri orta konumda tutulduğunda hareket ettirilebilir.

- Onay şalterinin bırakılması güvenlik nedeniyle durma 2'yi tetikler.
- Onay şalterine basılması güvenlik nedeniyle durma1'i tetikler.
- Kısa süreliğine 2 onay şalterini eşzamanlı orta konumda tutmak mümkündür. Bu sayede bir onay şalterinden diğerine geçilebilir. 2 onay şalteri uzun süre eşzamanlı olarak orta konumda tutulursa birkaç saniye sonra güvenlik nedeniyle durma tetiklenir.

UYARI Onay şalterleri, yapışkanlı bant ya da diğêr yardımcı cisimlerle sabitlenmemeli ve farklı şekillerde manipüle edilmemelidir. Ölüm, ağır yaralanmalar veya maddi hasarlar ortaya çıkabilir.

İşletim türleri ve koruma fonksiyonlarına genel bakış

Aşağıdaki tablo, koruma fonksiyonlarının hangi işletim türünde etkin olduğunu gösterir.

Koruma fonksiyonları	T1, KRF	T2	AUT	AUT EXT
Kullanıcı koruması	-	-	etkin	etkin
ACİL-DURDURMA tertibatı	etkin	etkin	etkin	etkin
Onay tertibatı	etkin	etkin	-	-
Program sorgulamasında azaltılmış hız	etkin	-	-	-
Jog işletimi	etkin	etkin	-	-
Yazılım limit şalteri	etkin	etkin	etkin	etkin

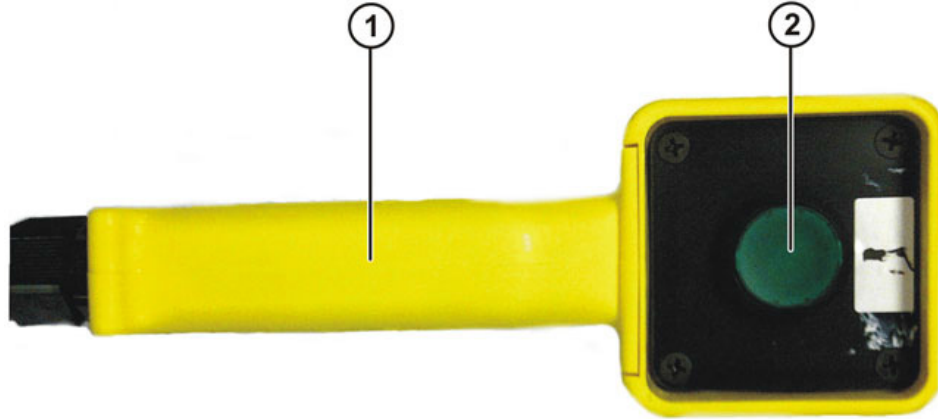
Robot kumanda ünitesi olmadan manipülatörü hareket ettirme yollar Bir kaza veya bir arıza sonrasında manipülatörü manuel olarak hareket ettirebilmek için aşağıdaki yöntemler mevcuttur:

- Serbest döndürme düzeneği (opsiyon)
Serbest döndürme düzeneği, ana aksın tahrik motorları için ve robot tipine bağlı olarak el aksının tahrik motorları için de kullanılabilir



Resim 19-1: Serbest dönme düzeneği

- Fren açma cihazı (opsiyon)
Fren açma cihazı, motorlarına erişimi serbest olmayan robot tipleri için öngörülmüştür.
Örnek AGILUS:



Resim 19-2: El kumanda cihazı

1 El kumanda cihazı

2 Kumanda butonu S1

- El akslarının doğrudan elle hareket ettirilmesi
Düşük taşıma kapasitesi versiyonlarında el aksları için serbest döndürme düzeneği yoktur. Bu el aksları direkt elle hareket ettirilebildiklerinden bu tür bir düzenek gerekli değildir.
Bu yöntemler sadece istisnai ve acil durumlarda, örn. şahısların kurtarılması amacıyla kullanılabilir.

i Hangi robot modelleri için hangi imkanların kullanılabildiği ve bunların nasıl kullanıldığı ile ilgili bilgileri için, robotlar için montaj ve işletim kılavuzuna başvurabilir ya da KUKA Roboter GmbH'ya danışabilirsiniz.

⚠ DİKKAT Motorlar, işletim sırasında ciltte yanmalara neden olabilecek sıcaklıklara ulaşmaktadır. Temas etmekten kaçınılmalıdır. Gerekli korunma önlemleri alınmalı, örneğin koruyucu eldivenler kullanılmalıdır.

İndeks

A

ACİL DURDURMA 58
ACİL DURDURMA tertibatı 246
ACİL DURDURMA, harici 246
ACİL-DURDURMA tertibatı 245, 246
Ağırlık 45
Aks aralığı 242
Aksa özgü el yöntemi 64
Aksları tek tek hareket ettirme 64
Akülerde tam deşarj 46
Alanlar 173
Alet 30
Alet kullanımı 13
Alet ölçme 89
Alet yük verileri 86
Alet yük verileri (menü noktası) 87
alma, WorkVisual proje 229
Alt program 157
Alt program, global 159
Alt program, lokal 157
Anahtar sözcük 165
Anahtarlama fonksiyonu, basit 140
Anahtarlama fonksiyonu, yörünge 142
Array 173
Ayırıcı koruma tertibatlarının kilitlemesi 245

B

Bağlantı yöneticisi 58
Base koordinat sistemi 67
Base ölçümleme 96
Başlat geri tuşu 59
Başlat tuşu 59
Bekleme fonksiyonu 136
Bekleme mesajı 61
Bildirim 165, 167
Bilek dibi noktası 123, 124
Bit işlemleri 170
Buton çubuğu 216

C

Capek Karel 7
CCU 241
CCUsr 241
CEM 241
CIB 241
CIBsr 241
CIRC hareketi 123
CSP 241

Ç

Çalışma alanlarında alan denetimi 34
Çalışma bölgeleri (pencere) 216
Çalışma bölgesi 242

D

Dağıtıcı 155
Daire açısı 184
Dallanma 151

Dallanma çoklu 155
Dallanma koşullu 154
DECL 165
Değişkenler 163, 165
Değiştirme, hareket komutları 130
Dışlayıcı döngü 152
Dışlayıcı olmayan döngü 153
Dikey mafsal koordinatları 41
Diyalog mesajı 61
Dokunmatik ekran 57
Dolap tipi 45
Döngü dışlayıcı 152
Döngü dışlayıcı olmayan 153
Döngü Sayaç döngüsü 152
Döngü Sonsuz döngü 151
Döngüler 151
Dual-NIC kartı 241
DURDURMA 0 242, 243
DURDURMA 1 242, 243
DURDURMA 2 242, 243
Durdurma kategorisi 0 243
Durdurma kategorisi 1 243
Durdurma kategorisi 2 243
Durma mesafesi 242
Durum analizleri 154
Durum mesajı 60
Durum tuşları 58
Dünya koordinat sistemi 67

E

EDS 241
Efektör 30
El yöntemi, dünya 69
EMD 241
EN 10218-1 38
EN 775 38
Endüstriyel robot 7, 9
Endüstriyel robot kullanımını 13
Enerji beslemesi 30
Enerji verimliliği 55
Enkoder kablosu uzunluk farkı 46
Entegre enerji beslemesi 31
Enterpolasyon modu 121, 129
ENUM 178
Esneklik 7
Eş potansiyel topraklama 45

F

Flange koordinat sistemi 67
Fonksiyon 161
Fonksiyonlar 157
Fonksiyonlar, matematiksel 161
FOR döngüsü 152
Fren açma cihazı 247
Frenleme hattı 242

G

GBE 241

global 163
Güvenlik fonksiyonları, genel bakış 243
Güvenlik kontrol ünitesi 244
Güvenlik KR C4 243
Güvenlik nedeniyle durma 0 242
Güvenlik nedeniyle durma 1 243
Güvenlik nedeniyle durma 2 243
Güvenlik nedeniyle durma DURDURMA 0
242
Güvenlik nedeniyle durma DURDURMA 1
243
Güvenlik nedeniyle durma DURDURMA 2
243
Güvenlik seçenekleri 243
Güvenlik tertibatları 32

H
Hareket programlama 115
Hareket tuşları 58
Harici enerji beslemesi 30
Harici güvenlik sensörleri 36
Harici Otomatik 198, 199
Hat uzunlukları 46

I
IF -ELSE 154
IF sorgusu 154
Inline formu 116

İ J
İğneleme 228
İklimsel koşullar 45
İlave akslar 243
İlave yük 28
İlave yük verileri (menü noktası) 89
İlklendirme 103, 168
İşlenen parça elleçlemesi kullanımı 13
İşletici 242
İşletim türü 62
İşletim türü seçimi 243, 244
İşletimi güvenli şekilde durdurma 242
İzin verilen nominal gerilim toleransı 45
Jog işletimi 246

K
Kararlaştrmalar 163
Karşılaştırma işlemleri 170
Kartezik koordinatlar 41
Kataloglar (pencere) 216
Kavramlar, Güvenlik 242
KCB 241
KCP 57, 241, 242
KEB 241
Kısaltmalar 241
Klavye 59
Klavye tuşu 59
KLI 241
KOI 241
Kontrol sistemi genel görünüm 48
Kontrol ünitesi 46
Koordinat sistemi 67
Koruma bölgesi 242
Koruma fonksiyonları 246
Koruma sınıfı 45

KPC 241
KPP 241
KRF 242
KRL 241
KRL editörü 235
KRL'de hareket programlama 181
KSB 241
KSI 241
KSP 241
KUKA Control Panel 57
KUKA Controller veri yolu, KCB 51
KUKA Extension Bus, KEB 53
KUKA Line Interface, KLI 54
KUKA sistem veri yolu, KSB 52
KUKA smartPAD 46, 57, 242
Kullanıcı 242
Kullanıcı ara yüzü WorkVisual 215
Kullanıcı grubu, standart 147
Kullanıcı koruması 243, 245, 246
Kullanım koşulları 28
Kullanım ömrü 163
Kullanım süresi 242
Kurulum yüksekliği 46
Küre koordinatları 41

L
LIN hareketi 123
lokal 163
LOOP 151
LWL 241

M
Manipülasyon 169, 170
Manipülatör 242
Mantık, genel 135
Mantıksal işlemler 170
Manüel sürüş, Base 77
Manüel sürüş, Tool 73
Mekanik son dayanaklar 33
Menü çubuğu 216
Mesajlar 60
Mesajlar (pencere) 216
Montaj 13
Mutlak hassasiyet 43

N
Nem sınıfı 46
Nokta atlama PTP 119
Nominal bağlantı gerilimi 45
Nominal yük 28
Numaralandırma veri tipi 178

O
Onay mesajı 60
Onay şalteri 59, 246
Onay tertibatı 246
Operatör 147
OPI 241
Ortam sıcaklığı 45
Otomatik harici 195

Ö

Ön yürütme göstergesi 165
Öncelik 172
Özellikler (pencere) 216

P

Panik konumu 65, 246
Parametreleri aktarma 161
Performance Level 244
PLC'den program başlatma 195
PMB 241
Portal robotları 41
Program akışının kontrolü 151
Program başlatma 104
Program seçimi 104
Program ad değiştirme 113
Program çoğaltma 113
Program sil 112
Programcı 147
Proje açma 214
Proje gezgini 218
Proje karşılaştırma 219
Proje yapısı (pencere) 216
Proje yükleme 219
Projeyi etkinleştirme 226
Projeyi robot kumandasına aktarma (yükle-
me)
223
PTP hareketi 117

R

R.U.R. 7
RCD 241
RDW 241
REPEAT döngüsü 153
Return 158
Robot 7, 8, 38
Robot hücresi 27
Robot kanunları 8
Robot kontrol sistemine genel bakış 48
Robot kontrol ünitesi asgari mesafeleri 47
Robot kontrol ünitesi ölçüleri 47
Robot sistemi 27
Robotta yükler 86
Robroot 67
Rossum'un Üniversal Robotu 8

S

SAK 103
Sarsıntıya dayanıklılık 46
SATA 241
Sayaç döngüsü 152
SBC 241
Scara robotları 41
Sensör 32
Sensör sistemi 32
Serbest döndürme düzeneği 247
Serbest dönüş tertibatı 66
Ses düzeyi 45
SIB 241
SION 242
Silindir koordinatları 41

Sistem entegratörü 243
smartPAD 57, 242
smartPAD uzatma kabloları 46
Sonsuz döngü 151
Space Mouse 58
Standart fonksiyonlar 170
STO 241
STOP tuşu 59
SWITCH - CASE 154

ŞŞ

ebeke bağlantısı, teknik veriler 45
Şebeke frekansı 45
Şebeke tarafındaki sigorta 45

T

T1 243
T2 243
Talimat, koşullu 154
Tam yük akımı 45
Taşlama 151
Tehlike bölgesi 242
Tekillik 123
Tekrarlama hassasiyeti 43
Temel hesap türleri 170
Temel veriler 45
Template 229
Tepki hattı 242
Tesis entegratörü 243
Tip levhası 59
Tool koordinat sistemi 67

U

Unimate 9
USB 242
USB bağlantısı 59
USV 242
Uyarı mesajı 61
Uygulamaya ve veri yolu sistemlerine genel
bakış 49
Uzman düzeyi 147

Ü

Üzerinden atlama 122, 130
Üzerinden atlama CIRC 127
Üzerinden atlama LIN 127

V

VDI Yönergesi 2860 38
Veri yolu sistemleri genel bakış 49
Veri yönetimi 163
Verimlilik 7

Y

Yapı 176
Yardım 216
Yatay mafsal koordinatları 41
Yazılım limit şalteri 33, 246
Yönelim kılavuzu 124
Yönetici 147
Yönlendirme kılavuzu 130

Yükler 28

Z

WAIT 136

WAIT FOR 137

WHILE döngüsü 152

WorkVisual 207, 214

World koordinat sistemi 67