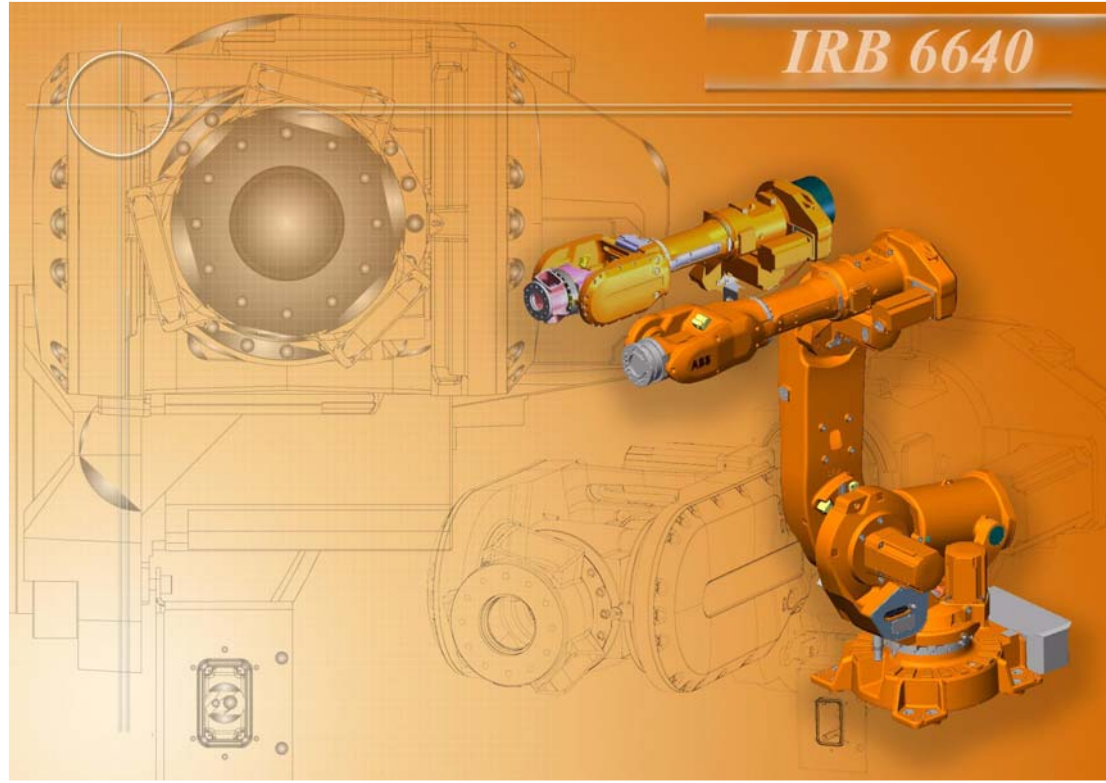


# İstanbul Teknik Üniversitesi Mekatronik Eğitim ve Araştırma Merkezi

## ABB Robotics IRB 6640 Genel Kullanım Bilgileri



Serhan Aydın AYA (serhanaya@yahoo.com)

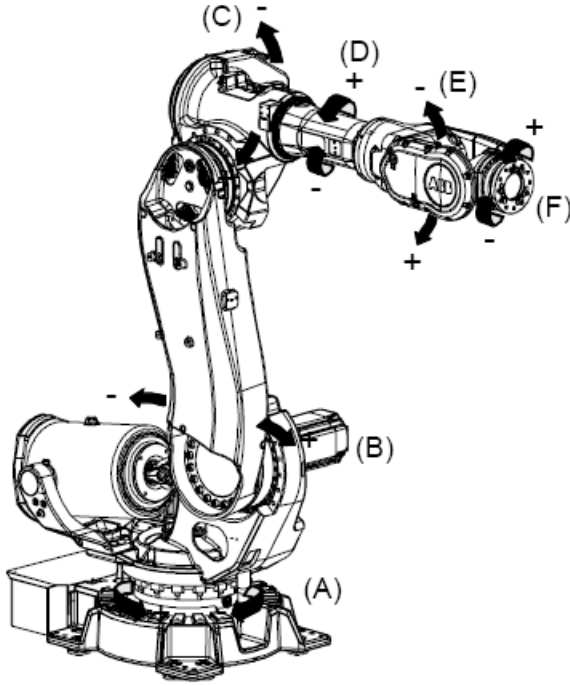
Eray ÇAKIRAY (eraycakiray@yahoo.com)

İ.T.Ü. M.E.A.M.

28/10/2008

## IRB 6640

ABB Robotics firmasının IRB 6640 serisi, yüksek performanslı endüstriyel robotlardır. IRB 6640, endüstri sınırlaması olmaksızın proses uygulamaları için idealdir (Örnek: spot welding, material handling, machine tending).



**Şekil 1** IRB 6640 manipülatörünün 6 eksenli bulunmaktadır.

A	1. eksen	B	2. eksen
C	3. eksen	D	4. eksen
E	5. eksen	F	6. eksen

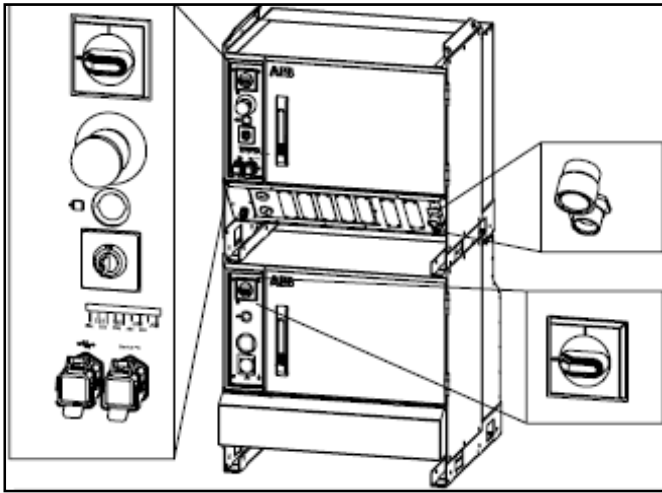
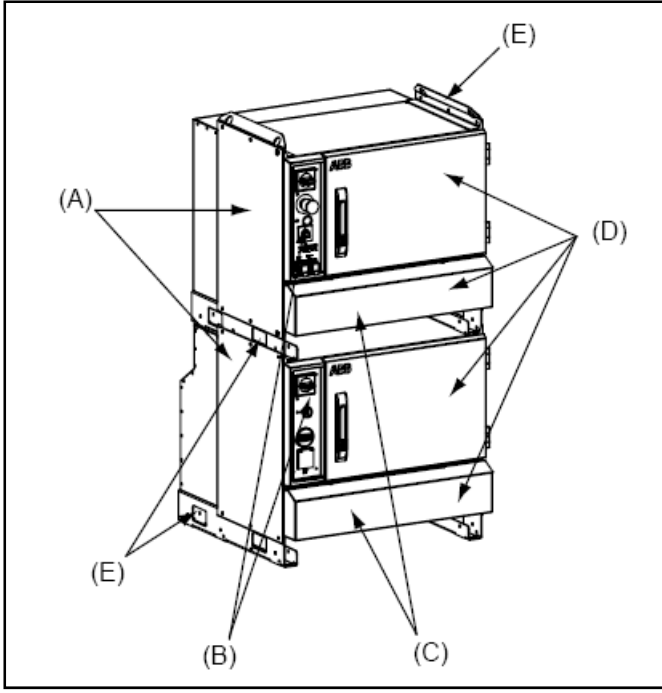
## IRC5

Kontrol ünitesi. Manipulatörü, eksenleri ve çevresel teçhizatı kontrol etmek için gerekli olan elektronik aksamı içerir.

Çift kabinli denetleyici aşağıda yazılı olan modüllerden yapılmıştır:

- Sürücü sistemini içeren sürücü modülü.
- Bilgisayarı, operatör'ün kullanımına sunulan paneli, ana elektrik devre anahtarlarını, iletişim arayüzlerini, FlexPendant bağlantısını, servis portlarını ve kullanıcının teçhizatları için (örneğin, ABB I/O kartları) ayrılmış boşluğu içeren kontrol modülü. Kontrol ünitesi, ayrıca faaliyet ve programlama süreci için gerekli olan tüm temel fonksiyonları içeren sistem yazılımını (RobotWare-OS) bünyesinde bulundurmaktadır.

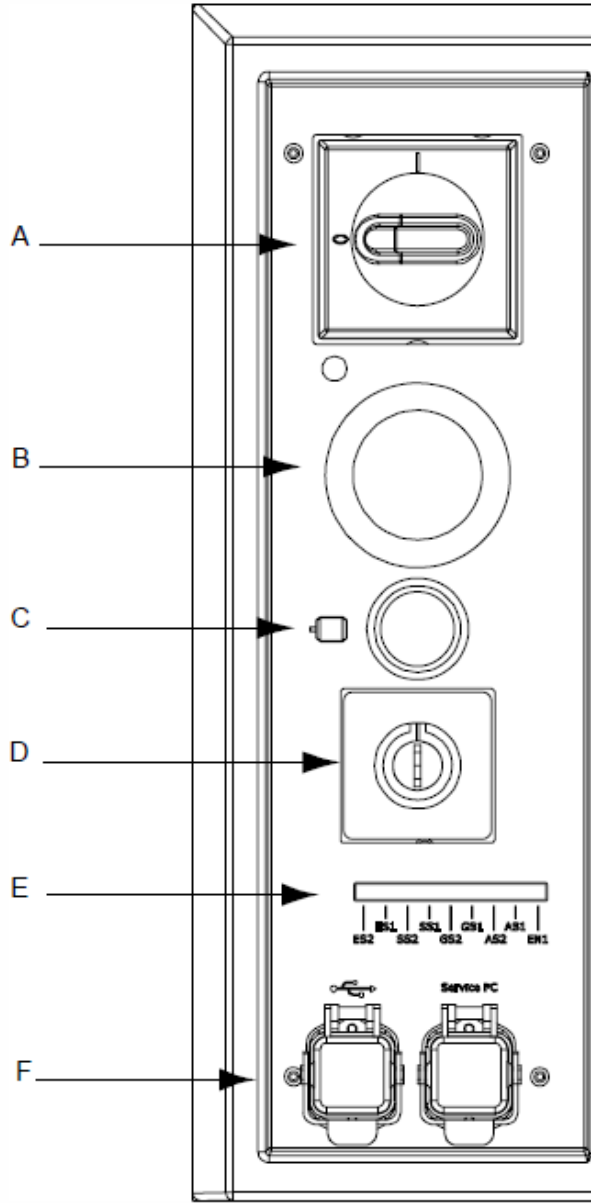
Bunların yanında bilgisayar genişleme kartları için 4 adet PCI bölmesi bulundurmaktadır.



**Şekil 2** Kontrol ünitesi (dual cabinet)

A	Kontrol Ünitesi
B	Operatör paneli
C	Konektör Koruması
D	Kapı ve konektör koruması
E	Ayaklar

## Operatör Paneli



Şekil 3 Kontrol Modülü




Pozisyonu	İsim
A	Sürücü Modülü'ne giden gücün kumandası ve şebekenin ana anahtarı
B	Acil durdurma - eğer basıldıysa, açmak için çevirilmelidir.
C	MOTORLAR AÇIK
D	Faaliyet modu seçicisi
E	Güvenlik LEDleri
F	USB bağlantısı, PC servis bağlantısı

### Motorlar Çalışır Halde:

MOTORLAR AÇIK	Faaliyet	Not
Sürekli ışık	Program çalıştırılmak üzere hazır	
Hızlı yanıp sönen ışık (4 Hz)	Robot ayarlanmamış ya da devir sayaçları güncellenmemiş.	Motorlar önceden açılmış.
Yavaş yanıp sönen ışık (1 Hz)	Koruyucu boşluk durduruculardan biri aktif halde.	Motorlar önceden kapatılmış.

### Faaliyet Modu Seçicisi

Bir kontak anahtarı kullanarak, robot iki ya da üç faaliyet moduna kilitlenebilir (tercih edilen mod seçicisine bağlı olarak).

Faaliyet Modu	Tanım	İşaretler
Otomatik mod	Ürün çalışıyor	
Manuel mod (düşürülmüş hız)	Programlama ve ayarlama maks. hız 250 mm/s	
Manuel mod (tam hız)	Tam program hızından test ediliyor. Bu mod ile düzenlendiğinde robot ANSI/UL standartlarına göre uygun olmaz.	

### Uzaktan kumanda

Operatör paneli ve FlexPendant'ın her ikisi de kabinden ayrılacak şekilde harici olarak bağlanabilir, ve robot sonra oradan kontrol edilebilir.

Uzaktan kumandanın içeriği (opsiyonel) şöyledir:

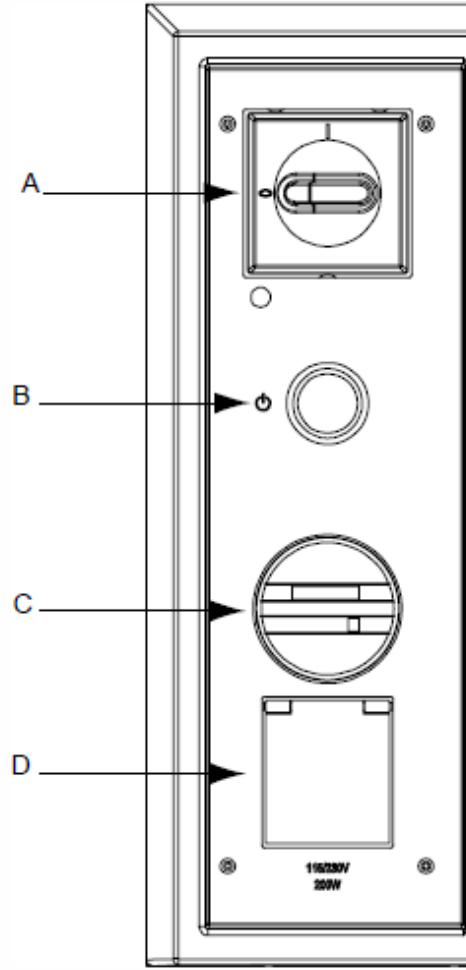
- Acil durdurma
- MOTORLAR AÇIK
- Faaliyet modu seçicisi (operating mode selector)
- Opsiyonel olarak Hot soketini kapsayan FlexPendant bağlantısı

Kontrol kabininde mevcut diğer içerik:

- Şebeke kesicisi (mains switch – ana elektrik devre anahtarı)
- Opsiyonel güvenlik LED'leri.
- Servis PC bağlantısı
- USB bağlantısı

Robot bir bilgisayar, PLC ya da kullanıcı paneli vasıtasıyla seri iletişim (serial communication) ya da sistem sinyallerini (system signal) kullanılarak uzaktan da kontrol edilebilir.

## Sürücü Modülü



Şekil 4 Sürücü Modülü

Yeri	Tanım
A	Mains isolator switch.
B	Stand by lamp indicates that electronic supply is switched on by the Control
C	Duty Time Counter (option) accumulates the hours (up to 99999.99 h) when the motors are in operation and the brakes are released
D	Service outlet 115/230 V, 200 W (option)

## FlexPendant

Taşınabilir FlexPendant (Şekil 3), kullanıcı paneli (Şekil 2) ve RobotStudio Online ile tüm faaliyet ve programlamalar yapılabilir.



**Şekil 5** FlexPendant. Büyük dokunma duyarlı renkli ekranı ile değişik sistem bilgilerini, basit dil mesajlarını, resim/grafikleri ve kullanıcıyla dokunma yolu ile etkileşimde olan alanı görüntülemektedir.

Yeri	Tanım
A	Ekran
B	Kullanıcı tanımlı tuşlar
C	Acil durdurma düğmesi
D	Joystick
E	Program çalıştırma tuşları

**Not:** FlexPendant'ın nasıl kullanıldığını öğrenebilmek için programlama bilgisi ya da bilgisayar deneyimi gerekmemektedir.

**Dokunmatik ekran:** 7.5" renkli ekran text bazlı bilginin yanında grafiksel bilgileri görüntülemektedir. Kullanıcı girişleri, menü komutlarına basarak ya da düğmeleri kullanarak vs. girilmektedir. Aynı anda birçok pencerenin açılmasına izin verilmektedir. FlexPendant ile Microsoft.NET teknolojisini baz alan kullanıcı uygulamaları kullanılabilir.



Şekil 6 Dokunma duyarlı ekran

A	<b>ABB menu</b> - ABB menüsü
B	<b>Operator window</b> - Operatör ekranı
C	<b>Status bar</b> - Durum çubuğu
D	<b>Close button</b> - Kapatma düğmesi
E	<b>Task bar</b> - Görev çubuğu
F	<b>Quickset menu</b> - Hızlı ayar menüsü

### Program çalıştırma tuşları:

Program başlangıç/bitiş ve adım adım ilerleme/gerileme icrasında kullanılan tuşlar.

### Hold-to-run

Manuel modda tam hızda program çalıştırmak için program çalıştırma tuşlarından birisi daimi olarak basılmalıdır.

### Kullanıcı-tanımlı tuşlar

Çıkışı ayarlamak için (örneğin açma/kapama mandalı) ya da sistem girdisini aktive etmek için yapılandırılmış 4 adet kullanıcı tanımlı tuş.

### Cihazı etkinleştirmek

Manuel modda, düğmeye yarım basıldığında, sistem MOTORLAR AÇIK durumuna geçer.

Cihazı etkinleştirme düğmesi serbest bırakıldığında ya da tamamiyle basıldığında, robot MOTORLAR KAPALI vaziyetini alır.



## Joystick

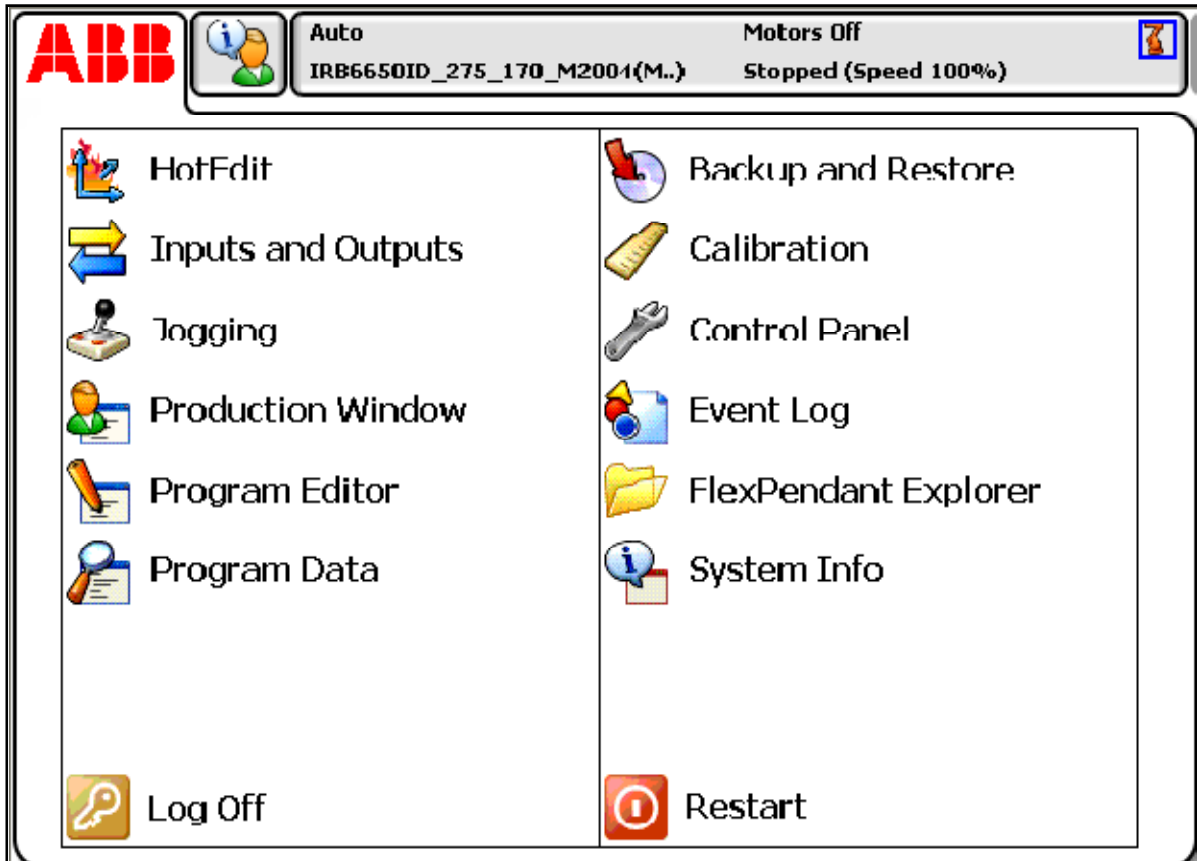
3B joystick robotu manuel olarak hareket ettirmek (jog) için kullanılır (örneğin robotu programlarken). Kullanıcı bu hareketin hızını belirler, joystickteki büyük ölçekli sapmalar robotun hızlı olarak hareket etmesine, küçük sapmalar ise robotun daha yavaş hareket etmesine sebep olacaktır.

## Acil durdurma düğmesi

Bu tuşa basıldığında robot anında duracaktır.

## ABB menüsü

Bu bölümde FlexPendant ile daha verimli çalışılmasını sağlayacak bilgiler bulunmaktadır. ABB menüsünün tüm görünümüne ait kısa bilgiler verilmiştir. Buradaki görüntülerin temel *RobotWare* sistemine ait olduğu unutulmamalıdır.



Şekil 7 ABB Menüsü

## HotEdit

HotEdit önceden programlanmış konumların düzenlenmesinde kullanılmaktadır.

## FlexPendant Explorer

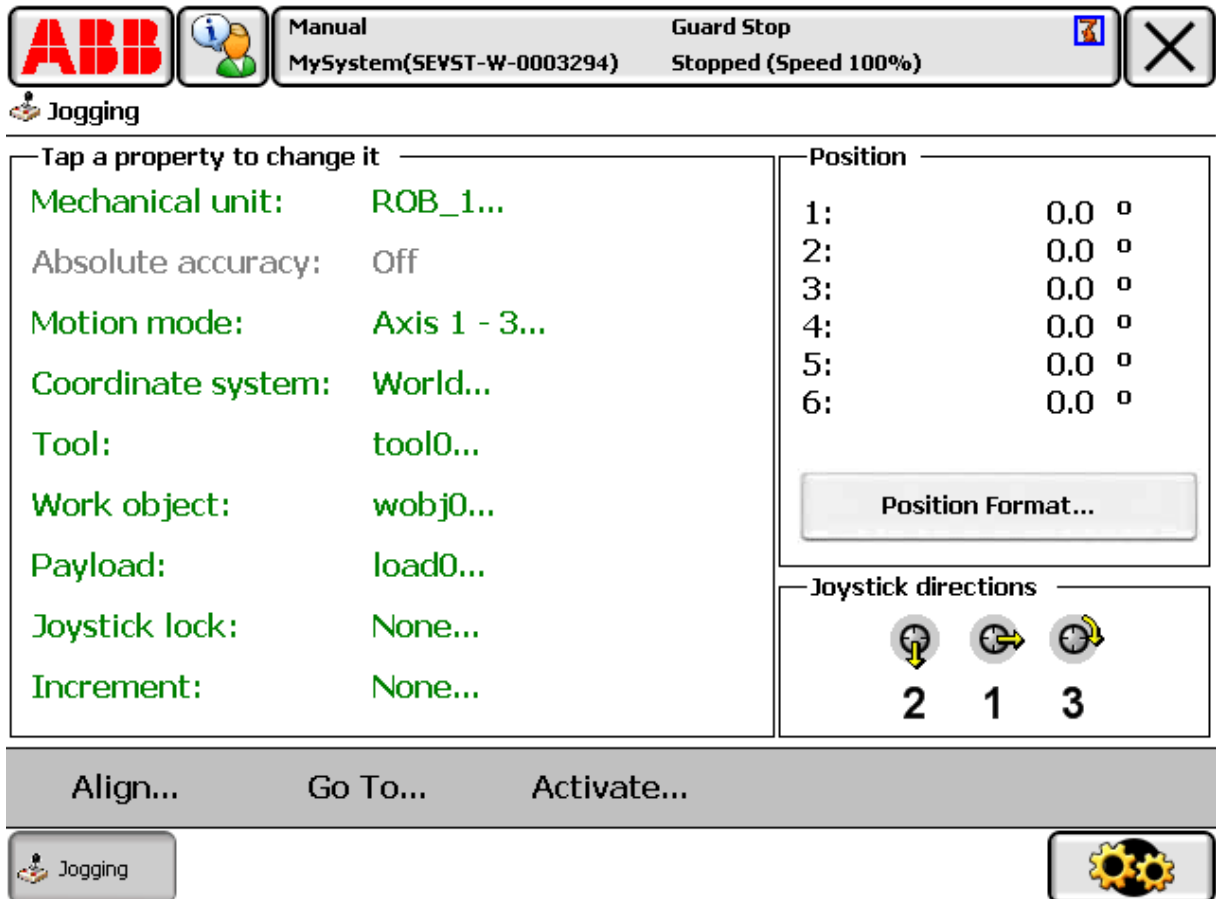
FlexPendant Explorer tıpkı Windows Explorer gibi kontrol ünitesindeki dosya sisteminin görüntülediği bir dosya yöneticisidir. Ayrıca, dosyaları veya klasörleri yeniden adlandırma, silme, ya da taşıma işlemleri gerçekleştirilebilir.

## Inputs and Outputs, I/O – Giriş ve Çıkışlar

Giriş ve çıkışlar robot sisteminde kullanılan sinyallerdir. Sinyaller sistem parametreleri ile yapılandırılırlar (ayrıntılı bilgi verilecektir.)

## Jogging

Sporda ağır tempoda ve sürekli koşu anlamında kullanılan jog terimi kullanılmaktadır. Burada, yakın anlamdaki jogging, ABB Menüsünde yer almakta ve robotun hareketi için gereken bilgileri bulundurmaktadır. <sup>1</sup>



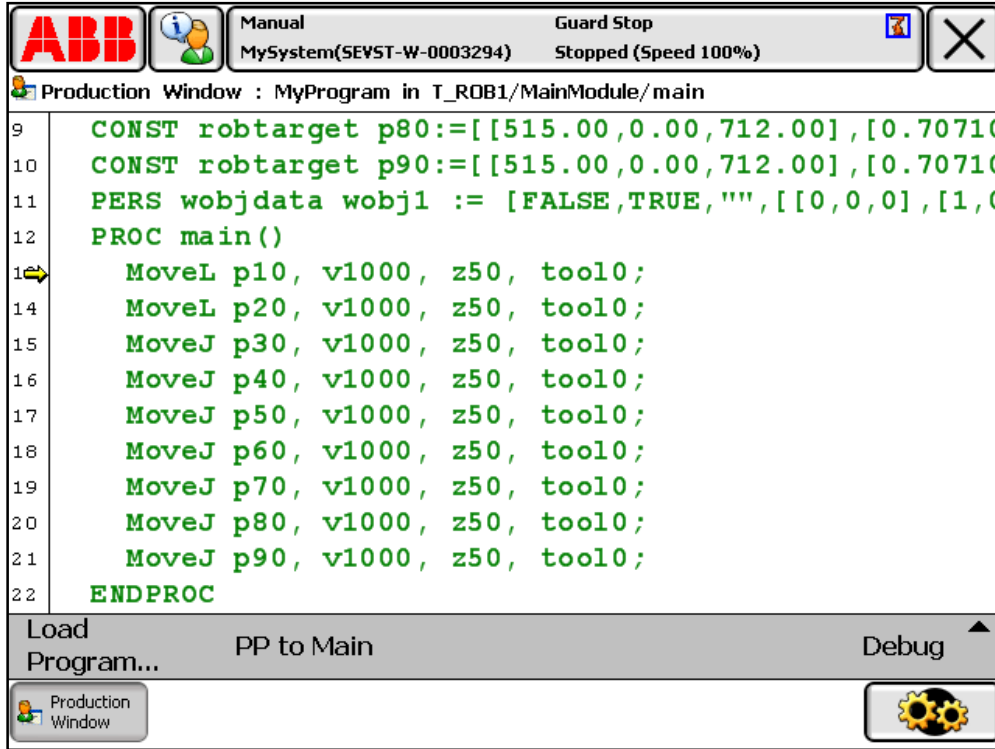
Şekil 8 Jogging menüsünde bulunan fonksiyonlar

<sup>1</sup> FlexPendant üzerinde bulunan joystick ile yapılan robot manipülasyonu da jog kelimesi ile ifade edilecektir

Özellik / Buton	Fonksiyon, Görev
<b>Mechanical unit</b>	Aktif mekanik ünitenin seçimi (Detaylı bilgi için: <a href="#">s.20-Mekanik birimin seçilmesi</a> )
<b>Absolute accuracy</b>	<b>Absolute Accuracy: Off</b> varsayılan seçimdir. Eğer robotta <i>absolute accuracy</i> seçeneği varsa, <b>Absolute Accuracy: On</b> görüntülenir.
<b>Motion mode</b>	Hareket modu seçimi (Detaylı bilgi için: <a href="#">s.21- Hareket modunun seçimi</a> )
<b>Coordinate system</b>	Koordinat sistemi seçimi (Detaylı bilgi için: <a href="#">s.24- Base (temel) koordinat sisteminde jog</a> )
<b>Tool</b>	Uç elemanın seçimi (Detaylı bilgi için: <a href="#">s.21-Uç elemanı, iş parçası ve yükün seçimi</a> )
<b>Work object</b>	İş parçasının seçimi (Detaylı bilgi için: <a href="#">s.21-Uç elemanı, iş parçası ve yükün seçimi</a> )
<b>Payload</b>	Yükün seçimi (Detaylı bilgi için: <a href="#">s.21-Uç elemanı, iş parçası ve yükün seçimi</a> )
<b>Joystick lock</b>	Joystick'in kilitleneceği doğrultuların seçimi
<b>Increment</b>	Harekette artırımının belirlenmesi
<b>Position</b>	Seçilen koordinat sistemine göre her bir eksenin konumunu gösterir. (Detaylı bilgi için: <a href="#">s.28-Incremental movement for precise positioning - Hassas konumlandırma için artırımlı hareket</a> )
<b>Position format</b>	Konum bilgilerinin formatı (Detaylı bilgi için: <a href="#">s.29-Konum, tam olarak nasıl belirlenir?</a> )
<b>Joystick directions</b>	Mevcut joystick doğrultularını <b>Motion mode</b> 'daki ayarlara göre görüntüler. (Detaylı bilgi için: <a href="#">s.21- Hareket modunun seçimi</a> )
<b>Align</b>	Mevcut uç elemanını bir koordinat sistemine göre hizalar (align).
<b>Go To</b>	Robotu seçilen konuma/hedefe hareket ettirir.
<b>Activate</b>	Mekanik ünitenin aktifleştirilmesi.

## Production Window

**Production Window**, program çalışırken program kodlarının görüntülenmesinde kullanılır.



Şekil 9 Production Window

<b>Load Program</b>	Yeni bir programın yüklenmesi
<b>PP to Main</b>	Program İşaretleyicisinin (program pointer) Ana programa (Main routine) hareket ettirilmesi.
<b>Debug</b>	Debug menüsü sade manual modda bulunmaktadır. <b>Modify Position Show Motion Pointer Edit Program</b>

## Program Data

Program Data görünümü, veri tipleri ve örnekler ile çalışma ve görüntülemeyi sağlayan fonksiyonları içermektedir.

## Program Editor

Program editor, programların oluşturulduğu ya da düzenlendiği kısımdır. Birden fazla Program Editor penceresi açılabilir, bu da, *Multitasking* seçeneği kurulmuşsa kullanışlı olabilir. Görev çubuğundaki (Task bar) Program editor butonu görevin ismini görüntüler.

ABB Manual MySystem(SEVST-W-0003294) Guard Stop Stopped (Speed 100%)

MyProgram in T\_ROB1/MainModule/main

Tasks and Programs Modules Routines

```

12 PROC main()
13 MoveL p10, v1000, z50, tool0;
14 MoveL p20, v1000, z50, tool0;
15 MoveJ p30, v1000, z50, tool0;
16 MoveJ p40, v1000, z50, tool0;
17 MoveJ p50, v1000, z50, tool0;
18 MoveJ p60, v1000, z50, tool0;
19 MoveJ p70, v1000, z50, tool0;
20 MoveJ p80, v1000, z50, tool0;
21 MoveJ p90, v1000, z50, tool0;

```

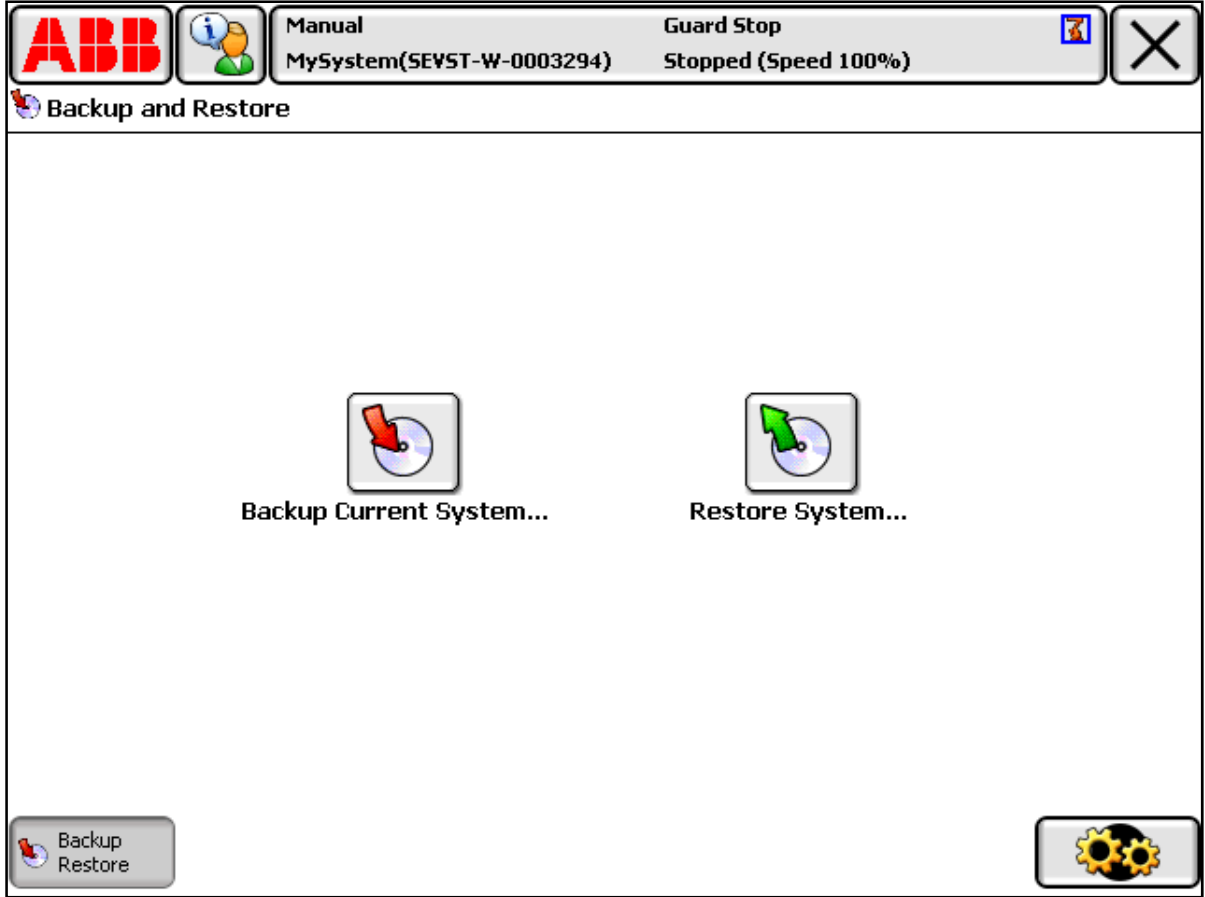
Add Instruction Edit Debug Modify Position Hide Declarations

T\_ROB1 : MainModule

<b>Tasks and Programs</b>	Program işlemleri için kullanılan menü
<b>Modules</b>	Tüm modulleri listeler
<b>Routines</b>	Tüm alt programları (routine) listeler
<b>Add Instruction</b>	Komut menüsünü açar
<b>Edit</b>	Düzenleme menüsünü açar
<b>Debug</b>	Program işaretleyicisinin hareketi, servis alt programları vs. için fonksiyonlar
<b>Modify Position</b>	
<b>Hide Declaration</b>	

## Backup and Restore

Backup and Restore menüsü sistemin yedekleme ve geri yükleme işlemlerini gerçekleştirmek amacıyla kullanılır.



**Şekil 10** Backup and Restore (yedekleme ve geri yükleme) menüsü

<b>Backup Current System</b>	
<b>Restore System</b>	

### **Calibration**

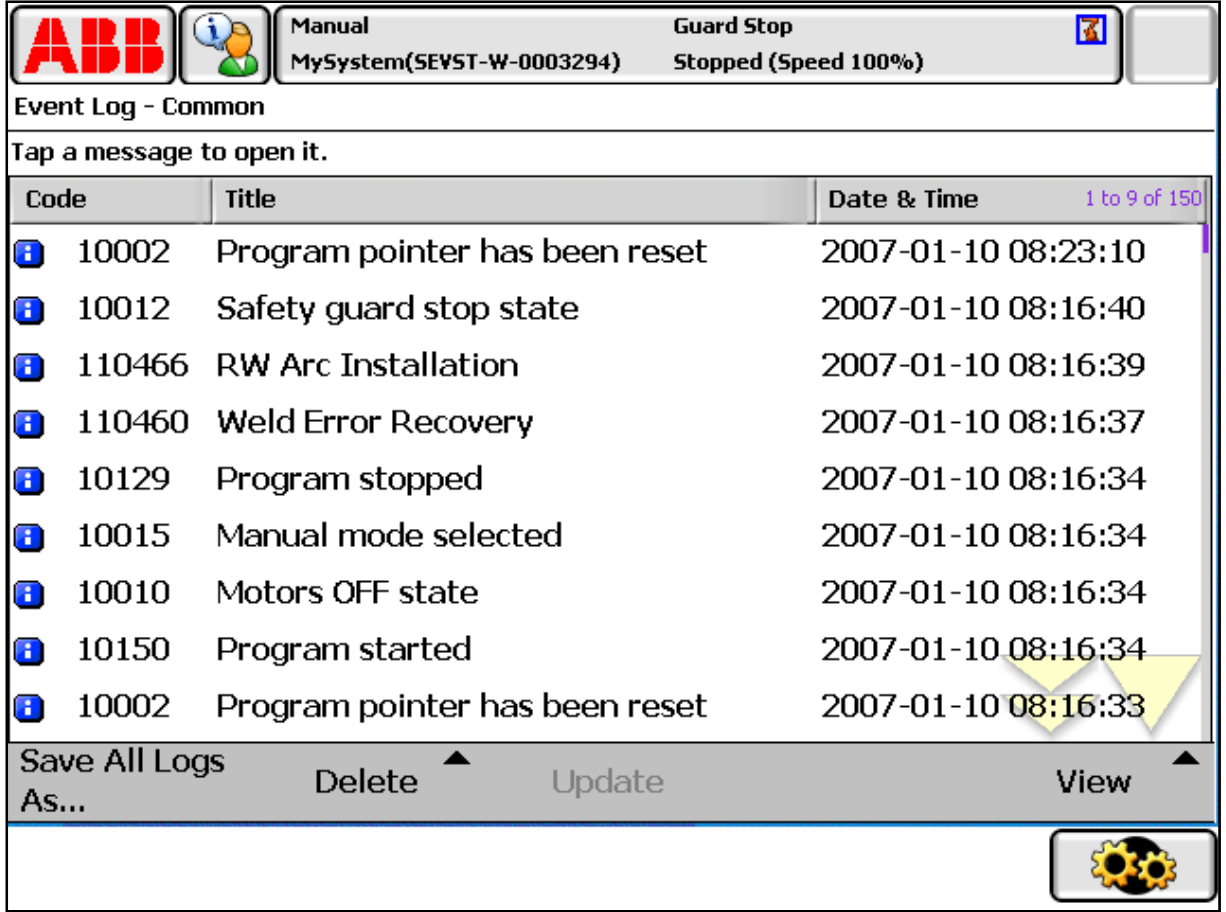
Calibration Menüsü, robot sistemindeki mekanik ünitelerin kalibrasyonunda kullanılmaktadır.

### **Control Panel**

Kontrol Paneli, robot sistemi ve FlexPendant'ı özelleştiren fonksiyonları içermektedir.

### **Event Log**

Aşağıdaki tabloda event log (olay kaydı) ile gerçekleşebilecek tüm faaliyetlerin özeti verilmiştir:

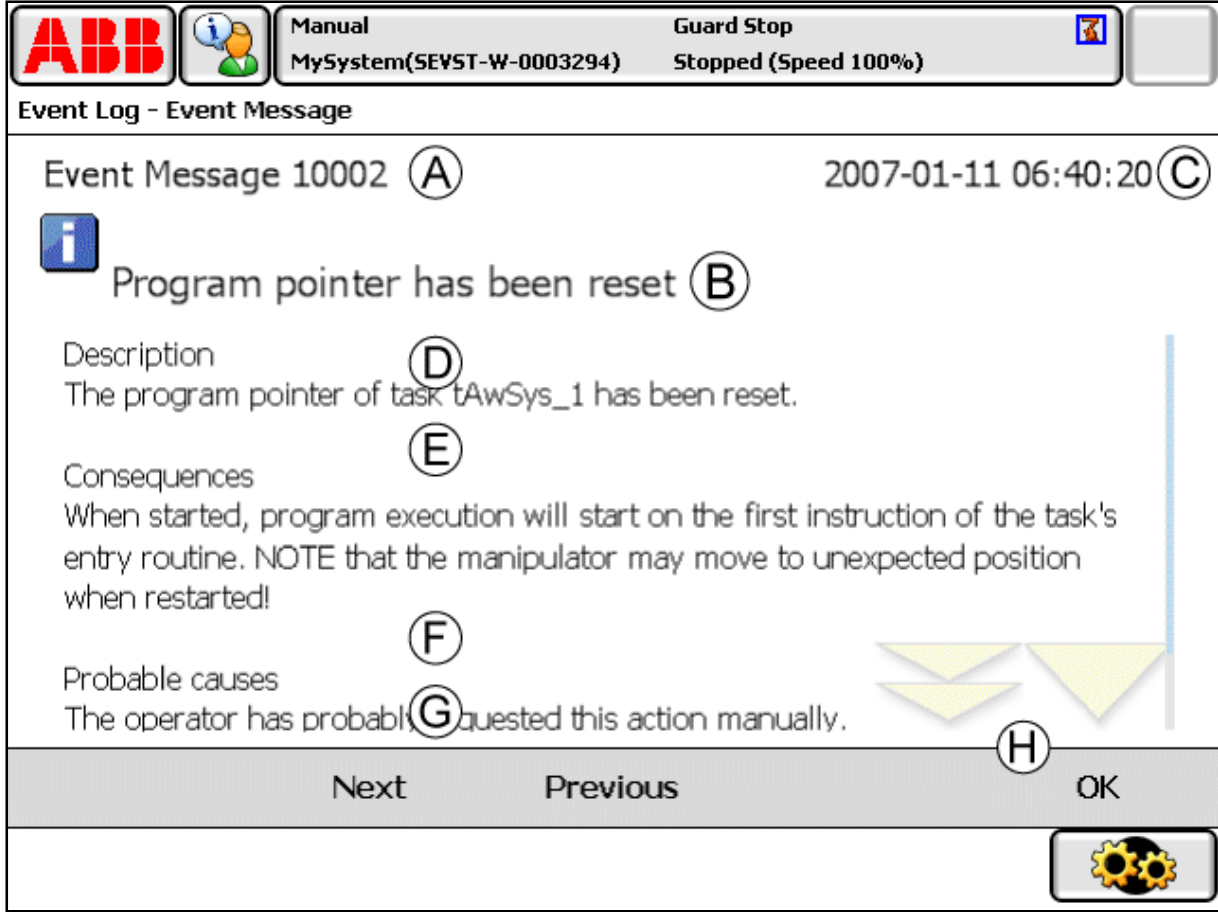


Şekil 11 Event Log

Faaliyetler	Açıklamalar
Log açılabilir	
Özel bir mesaj görüntüleniyor olabilir	Bir mesajı açmak için ona dokunun
Eğer log içeriği tek bir ekrana sığmıyorsa, bu içerik kaydırılabilir ve/veya yakınlaştırılabilir.	
Log silinebilir	
Log kaydedilebilir	
Log kapatılabilir	

### Event log mesajı

Her kaydın girdisi olayı detaylıca açıklayan bir mesajdan oluşmaktadır. Bunun yanında, genellikle sorunun nasıl çözüleceğine dair bir tavsiyeyi de içermektedir. Herhangi bir sorunla karşılaşıldığında ilk olarak bakılması gereken yerdir.



Şekil 12 Event Log - Event Message görüntüsü

A	<b>Event number</b> - Olayın kayıt numarası. Tüm hatalar numaralarına göre listelenir.
B	<b>Event title</b> - Olayın başlığı. Neler olduğunu kısaca açıklar.
C	<b>Event time marker</b> - Olayın zaman imi. Olayın tam olarak ne zaman meydana geldiğini belirtir.
D	<b>Description</b> - Açıklama. Olayın kısa açıklaması. Olayın sebep ve çıkarımlarını anlamaya yardımcı olacağı düşünülmüştür.
E	<b>Consequences</b> - Sonuçlar. Belirli bir olay nedeniyle meydana gelen, sisteme zararlı her türlü sonucun, (diğer operasyon moduna geçiş, acil durdurma) kısa açıklaması mevcuttur. Amaç, olayın sebep ve çıkarımlarının anlaşılmasına yardımcı olmaktır. (Eng: A brief description of any consequences inflicted on the system, transition to other operation mode, emergency stop, caused by the particular event. Intended to assist in understanding the causes and implications of the event.)
F	<b>Probable causes</b> - Olası nedenler. Olasılıklarına göre listelenmiş halde bulunurlar.
G	<b>Recommended actions</b> - Tavsiye edilen eylemler. Yukarıda bahsedilen “Olası nedenler” baz alınarak düzeltmeye yönelik tavsiye edilen eylemlerin listesidir. Bunlar, “Replace the xx.. (xx..’i değiştir)”, “Run the test program xx..(xx.. test programını çalıştır)” gibi çeşitlilik arz eden, örneğin problemi çözenin yanında onu izole etmeye yönelik eylemler olabilir.
H	<b>Acknowledge/OK</b> - Tanıma veya OK butonu.

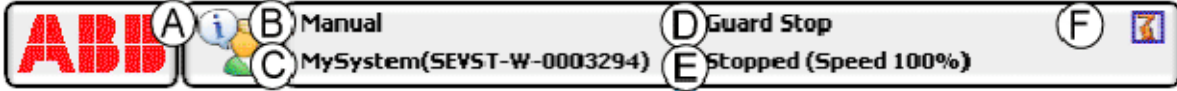


## Operator window - Operatör ekranı

Operatör ekranı programdan gelen mesajları görüntüler. Operatör ekranı, durum çubuğunda (status bar) ABB logosunun sağındaki ikona dokunularak açılır.

## Status bar - Durum çubuğu

Durum çubuğu mevcut durum (örn: operational mode, system, mechanical unit) hakkında bilgi verir.

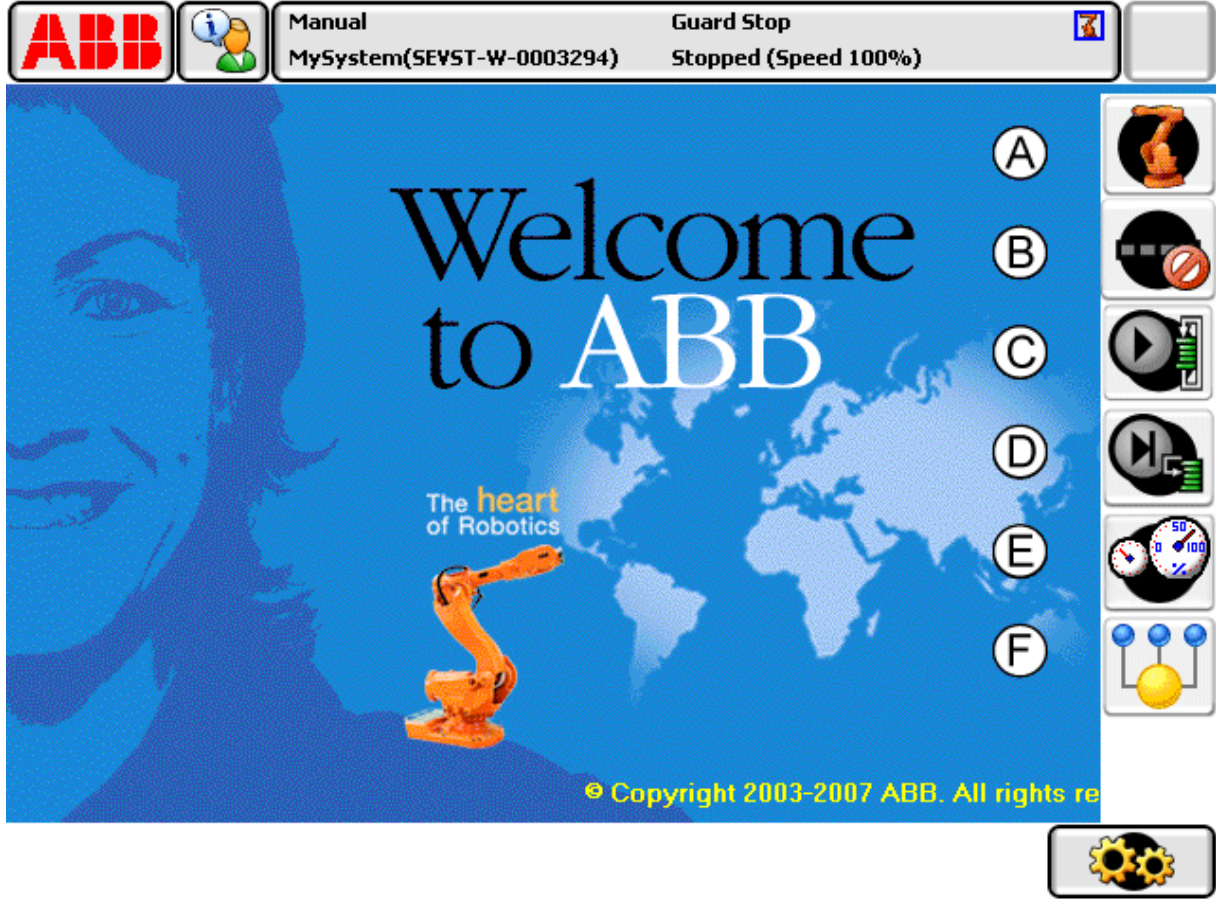


	İsim
A	Operator window - Operatör ekranı
B	Operating mode - İşletim modu
C	System name (ve Controller name) - Sistemin ve kontrol ünitesinin ismi
D	Controller state - Kontrol ünitesinin durumu
E	Program state - Programın durumu
F	Mechanical units - Mekanik üniteler. Seçilen ünite (ve seçilenle koordineli olan her ünite) işaretlenir. Aktif üniteler renkli olarak, aktif olmayan üniteler ise gri olarak görüntülenir.

## Quickset menüsü

QuickSet menüsünü kullanmak jog özelliklerini değiştirirken Jogging görünümünü kullanmaya göre hız kazandıracaktır.

QuickSet menüsünün her elemanı seçili olan nitelik değerini (property value) ya da ayarı göstermek için bir sembol kullanır. Quickset butonuna dokunarak mevcut nitelik değerleri görüntülenebilir.



Şekil 13 Quickset menüsündeki butonlar görülmektedir.

A	Mechanical unit - Mekanik birim
B	Increment - Artırım
C	Run mode - Programın devamlı yürütüleceğine veya önce çalışıp sonra duracağına karar verilen kısımdır.
D	Step mode - Programın adım adım (step-by-step) yürütmesinin nasıl olacağına karar verilecek kısımdır.
E	Speed - Hız
F	Tasks - Görevler

Menü hakkında detaylı bilgi s.31-36 arasında yer alan *QuickSet Menüsü* konusunda bulunmaktadır.

### Jogging

FlexPendant'ta bulunan joystick yardımıyla robotu hareket ettirme işlemine jogging denmektedir.

Jog, manuel modda yapılır. Program yürütülürken jog yapılamaz. Bunun dışında FlexPendant'ta ne görüntüleniyor olursa olsun jogging mümkündür.

Seçili olan hareket modu ve/veya koordinat sistemi robotun nasıl hareket edeceğini belirler. Doğrusal hareket (Linear motion) modunda TCP (Tool Center Point - Uç elemanın merkez noktası) uzayda doğrusal çizgilerde hareket eder, A noktasından B noktasına hareket gibi. TCP seçilmiş olan koordinat sisteminin eksenlerinde hareket eder.

Axis-by-axis modunda iken, belirli sürede tek robot eksenini hareket eder. Burada TCP hareketinin nasıl olduğunu tahmin etmek güçleşir.

Bir tutucu eleman vasıtasıyla bir pini, bir deliğe yerleştirme işlemi tool koordinat sisteminde - eğer sistemdeki bir koordinat (eksen), deliğe paralelse - kolayca yapılabilir. Aynı görevi temel (base) koordinat sisteminde icra etmek x, y ve z koordinatlarının her birinde jogging yapmayı gerektirdiğinden, hassaslığın sağlanmasını zorlaştıracaktır.

Uygun koordinat sisteminin seçimi jog işlemini kolaylaştıracaktır. Fakat hangi koordinat sisteminin seçileceğine dair basit veya tek bir cevap yoktur.

Doğru koordinat sistemi TCP'nin hedef konuma daha az joystick hareketiyle ulaşmasını mümkün kılacaktır.

Yer kısıtlamaları, engeller, iş parçası veya uç elemanının boyutları vs. de muhakeme sürecinde kıstas olarak alınacaktır.

### **Jogging Kavramı**

*Jog while using world zones*

*World Zones* seçeneğinin kurulu olması durumunda tanımlanmış yerler jog işlemini kısıtlayacaktır (World zones, robotun hareketini sınırlayan bir kafes gibi düşünülebilir).

*Jog with axis loads not set - Ayarlanmamış eksen yükleri ile hareket*

Eğer robotun herhangi bir eksenine bir ekipman monte edilmişse axes loads (eksen yükleri) mutlaka ayarlanmalıdır. Aksi durumda jog sürecinde aşırı yük hataları meydana gelebilir.

*Jog with tool or payload weights not set - Ayarlanmamış uç elemanı veya yük ağırlıkları ile hareket*

Eğer uç elemanlarının veya yüklerin ayarı yapılmamışsa jog sürecinde aşırı yük hatalarıyla karşılaşılabilir.

## Jogging için temel ayarlar

Mekanik birimin seçilmesi

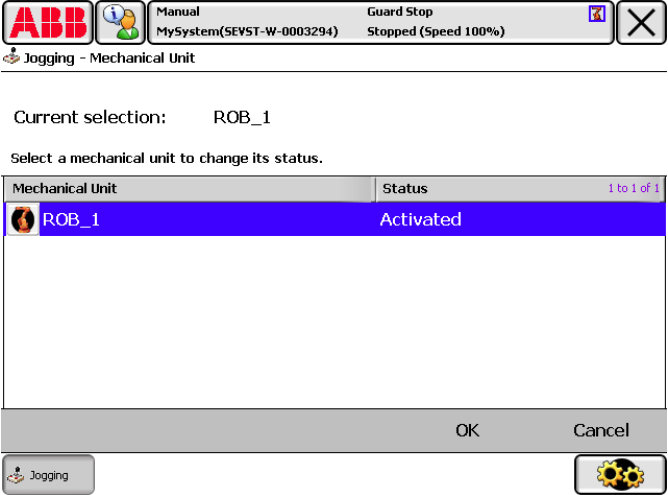
### *Kullanım örnekleri*

Robot sistemi birden fazla robottan oluşabilir. İş parçası tutucuları (workpiece handlers) veya robota montajlanmış ek eksenler gibi jog edilebilen mekanik birimler olabilir. (Ek eksene örnek: Taşıma bantı)

Eğer sistemde ek ekseni olmayan tek robottan oluşsaydı, mekanik birimin seçilmesine gerek olmayacaktı.

### *Mekanik birimlerin tanımlanması*

Jog edilebilecek her mekanik ünite, mekanik birimler listesinde gösterilir. Birimin ismi, sistem yapılandırmasında (system configuration) tanımlanır. Her birim, durum çubuğunda kullanılan bir sembole sahiptir. Aşağıdaki tabloda jog için mekanik birimlerin nasıl seçileceği açıklanmıştır:

1.	<b>ABB</b> menüsünde, <b>Jogging</b> 'i seçin.
2.	<b>Mechanical Unit</b> 'i seçin 
3.	Jog edilecek mekanik birimi seçin, ardından <b>OK</b> 'e dokununuz. Seçilen mekanik birim <b>Jogging</b> penceresini kapatsanız dahi siz diğer birimi seçene kadar kullanılacaktır.

**İPUCU:** Mekanik birimler arası geçişleri hızlandırmak için **Quickset** - Hızlı ayar menüsünü kullanın.

**NOT:** Jogging özellikleri ile ilgili yapılan herhangi bir değişiklik sadece seçili mekanik birimi etkiler.

## Hareket modunun seçimi

**Joystick Directions** bölümü joystick eksenlerinin seçili koordinat sisteminin eksenleri ile olan ilişkisini gösterir.

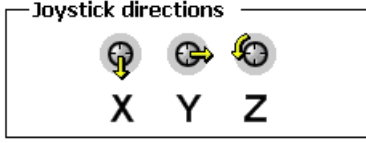
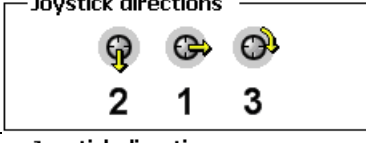
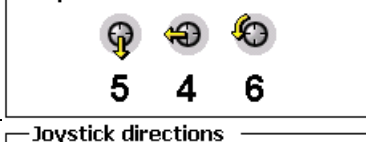
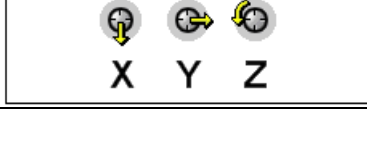
UYARI!

Doğrultu özellikleri, mekanik birimin hareket edeceği doğrultuyu göstermemektedir. Her zaman küçük joystick hareketleri ile hareket ettirmeyi deneyin böylece mekanik birimin gerçek doğrultularını öğrenmiş olursunuz.

Aşağıdaki tabloda hareket modunun seçimi açıklanmıştır:

	Eylem	Bilgi
1.	Önce <b>ABB</b> 'ye ardından <b>Jogging</b> 'e basın.	
2.	<b>Motion mode</b> seçeneğine basın.	
3.	Arzu ettiğiniz modu seçin ardından <b>OK</b> 'e basın.	Joystick doğrultularının önemi, seçimin ardından <b>Joystick direction</b> içinde gösterilir.

Joystick doğrultularının önemi hangi hareket modunun seçilmiş olduğuna bağlıdır:

Hareket modu	Örnek gösterim	Açıklama
Linear - Doğrusal		Doğrusal mod sayfa 22'de "tool oryantasyonunun ayarlanması" konusunda açıklanmıştır.
Axis 1-3 (robotlar için varsayılan seçimdir)		Axis 1-3 modu sayfa 22'de "jog axis-by-axis" konusunda açıklanmıştır.
Axis 4-6		Axis 4-6 modu sayfa 22'de "jog axis by axis" bölümünde açıklanmıştır.
Reorient		Reorient modu sayfa 22'de "jog axis-by-axis" bölümünde açıklanmıştır.

## Uç elemanı, iş parçası ve yükün seçimi

Uygun uç elemanının, iş parçasının veya yükün (payload) seçimi her zaman önem arz eder. Özellikle hedef konumlarına jog ederek hazırlanan programlarda çok önemlidir. Buradaki başarısızlık, büyük ihtimalle jog ya da program yürütülmesi esnasında aşırı yük hatası ve/veya hatalı konumlanmaya sebep olacaktır.

	<b>Eylem</b>
1.	<b>ABB</b> menüsünde, <b>Jogging</b> seçilerek jogging özellikleri görüntülenir.
2.	<b>Tool, Work object</b> veya <b>Payload</b> seçilerek mevcut uç elemanlarının, iş parçalarının veya yüklerin listesi görüntülenir.
3.	Tool, work object veya payload seçimini müteakip <b>OK</b> 'e basılır.

### **Tool oryantasyonunun ayarlanması**

#### *Kullanım örnekleri*

Ark kaynağında, taşlamada vs. kullanılan elemanlar en iyi sonucu elde etmek için iş parçasına uygun açıda yerleştirilmelidir. Delme, frezeleme ve biçme için de açının ayarlanması gerekmektedir.

Uç elemanının oryantasyonu çoğu zaman TCP'nin belirli bir noktaya hareketinde (elemanın kullanılacağı operasyonunun başlangıç noktası gibi) ayarlanır. Tool oryantasyonunun ayarlanmasını takiben belirlenen yolu tamamlamak için doğrusal modda harekete devam edilir.

#### *Tool oryantasyonunun tanımı*

Tool oryantasyonu seçili olan koordinat sistemine bağlıdır. Kullanıcı açısından bakıldığında ise bu durum farkedilmeyebilir.

#### *Tool oryantasyonunun ayarlanması*

	<b>Eylem</b>
1.	<b>ABB</b> menüsünde <b>Jogging</b> 'i seçiniz.
2.	Öncelikle <b>Motion Mode</b> 'a, ardından <b>Reorient</b> 'e ve sonrasında <b>Ok</b> 'e dokununuz.
3.	Eğer seçili değilse, <i>Uç Elemanı, iş parçası ve yükün</i> seçimi konusunda yazılan prosedürlere göre uygun elemanı seçin.
4.	Mekanik birimin motorlarının çalışması için enabling device'a (FlexPendant'ın sağ altında bulunmaktadır) basılı tutunuz. Joystick oynatıldığında uç elemanının oryantasyonu değişecektir.

İPUCU: Jogging modunu daha hızlı seçmek için **Quickset** menüsünü kullanınız.





### **Jog axis by axis**

#### *Kullanım örnekleri*

Eksenden eksene jogging şu durumlarda kullanılır:

- Mekanik üniteyi tehlikeli bir konumdan uzaklaştırırken,
- Robot eksenlerini tekilliklerden uzaklaştırırken,
- Hassas kalibrasyon için eksenleri konumlandırırken.

Hangi eksenin hareket edeceği nasıl belirlenir:

Hareket edilmesi istenen eksenler	Dokunulacak ikon	Joystick doğrultuları
1, 2 veya 3. eksen		
4, 5 veya 6. eksen		

Aşağıdaki tabloda ilgili prosedür verilmiştir:

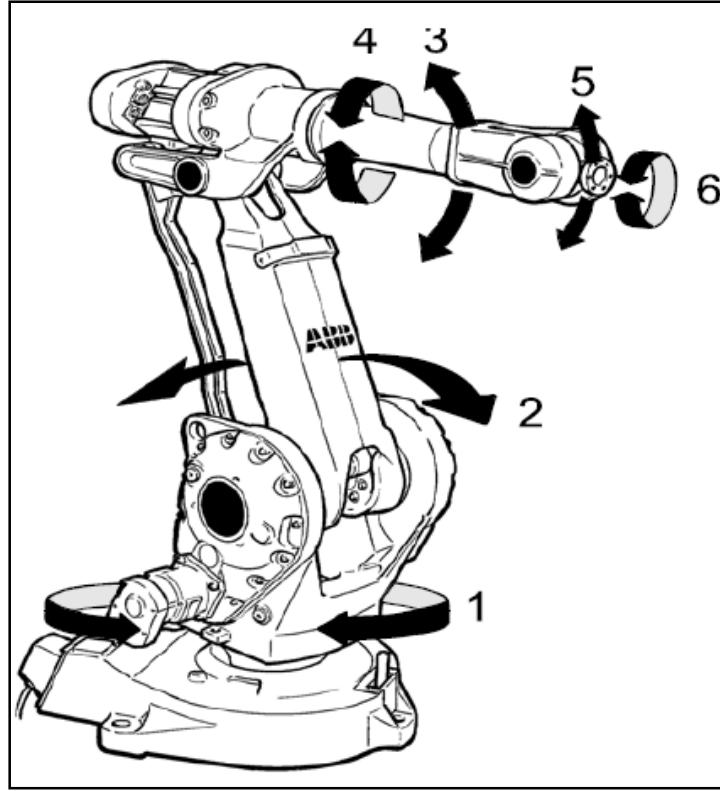
	Eylem
1.	<b>ABB</b> menüsünde <b>Jogging</b> seçilerek jogging özellikleri görüntülenir.
2.	<b>Motion mode</b> seçilir ardından axes 1-3 veya 4-6 seçilir.
3.	<b>OK</b> 'e dokunulduğunda işlem tamamlanır.
4.	Joystick'in eksen den eksene jog yaparken nasıl kullanılacağı <b>Joystick directions</b> alanında gösterilmektedir. Bunun yanında sayfa 23'teki <i>eksen ve joystick doğrultularının gösterimi</i> konusu da incelenmelidir.
5.	Enabling device basılı tutulur ve jogging'e başlanır.

## DİKKAT!

Montajlanmış her eleman burada açıklanan prosedürden etkilenecektir. Eğer sonuç oryantasyonu önemliyse bittiğinde *tool oryantasyonunun ayarlanması (sayfa 22)* konusunda verilen prosedür takip edilmelidir.

### *Eksen ve joystick doğrultularının gösterimi*

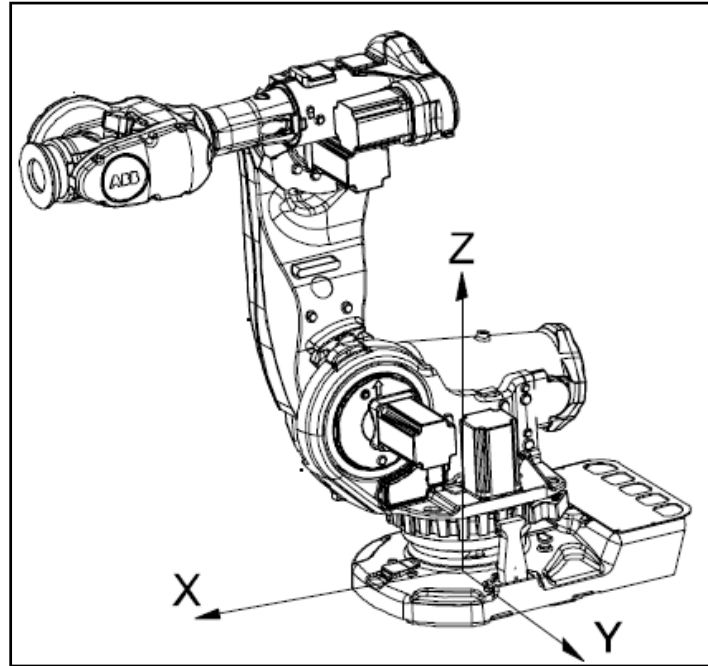
Manipülâtörün altı eksen, joystick'in üç boyutu kullanılarak manuel olarak jog edilebilir. Aşağıdaki şekilde manipülâtörün her eksenin hareket modeli gösterilmektedir:



Eksen doğrultularının gösterimi.

### Base (Temel) koordinat sisteminde jog

Temel koordinat sisteminin tanımı



	Eylem
1.	ABB menüsünde <b>Jogging</b> seçilerek jogging özellikleri görüntülenir
2.	<b>Motion mode</b> seçildikten sonra <b>Linear</b> seçeneğine ve son olarak <b>OK</b> 'e dokunulur. Daha



	önceden doğrusal hareket seçilmişse bu adım atlanabilir.
3.	<b>Coordinate System</b> seçilir, sonra <b>Base</b> 'e ardından <b>OK</b> 'e dokunulur.
4.	Enabling device basılı tutulur böylece mekanik birimin motorları aktive olur.
5.	Joystick hareket ettirildiğinde mekanik birim hareket edecektir.

İPUCU: **Quickset** menüsü kullanılarak jogging modu daha hızlı seçilebilir.

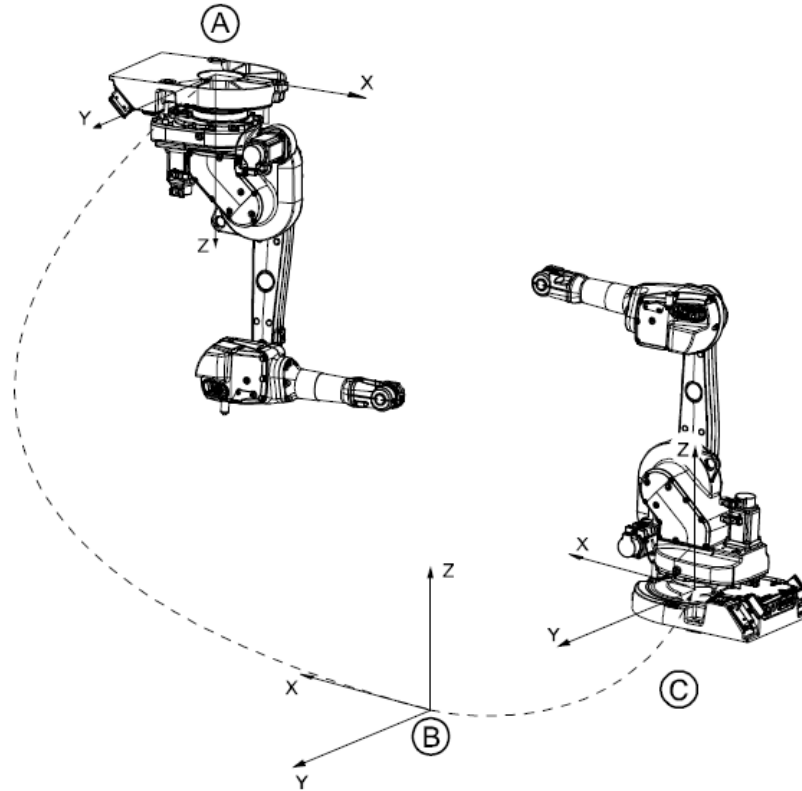
### World (dünya) koordinat sisteminde jog

#### *Kullanım örnekleri*

İki robotunuzun olduğu, birisinin yere montajlanmış halde, diğeri ise ters çevrili halde olduğu varsayalım. Ters çevrili robotun base koordinat sistemi baş aşağı olacaktır.

Ters çevrilmiş robotu base koordinat sisteminde jog etmek istediğinizde, hareket adımları tahmin edilmesi zor bir hal alacaktır. Bu durumda paylaşımlı world koordinat sisteminin seçilmesi uygun olacaktır.

#### *World koordinat sisteminin tanımı*



A	Base (Temel) koordinat sistemi
B	World (Dünya) koordinat sistemi
C	Base koordinat sistemi

Aşağıdaki tabloda world koordinatlarında jog işlemi prosedürü verilmiştir:

<b>Action</b>
---------------

1.	<b>ABB</b> menüsünde <b>Jogging</b> seçilerek jogging özellikleri görüntülenir
2.	<b>Motion mode</b> seçildikten sonra <b>Linear</b> seçeneğine ve son olarak <b>OK</b> 'e dokunulur. Daha önceden doğrusal hareket seçilmişse bu adım atlanabilir.
3.	<b>Coordinate System</b> seçilir, sonra <b>World</b> 'e ardından <b>OK</b> 'e dokunulur.
4.	Enabling device basılı tutulur böylece mekanik birimin motorları aktive olur.
5.	Joystick hareket ettirildiğinde mekanik birim hareket edecektir.

İPUCU: **Quickset** menüsü kullanılarak jogging modu daha hızlı seçilebilir.

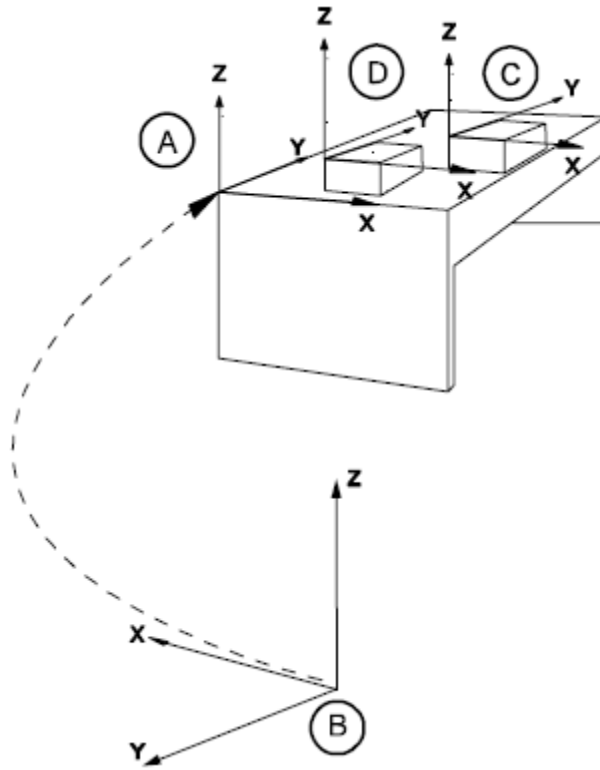
### Work object (iş parçası) koordinatlarında jog

#### Kullanım örnekleri

- Bir iş parçasının kenarları boyunca delinecek deliklerin konumlarını belirlerken,
- İki duvar arasındaki bir kutuda kaynak yaparken.

Verilen durumlar iş parçası koordinatlarının kullanımına örnek teşkil edebilir.

#### Work object koordinatlarının tanımı



A	User (kullanıcı) koordinat sistemi
B	World frame
C	Work object koordinat sistemi
D	Work object koordinat sistemi

Work object koordinatlarında jog prosedürü aşağıda verilmiştir:

	<b>Eylem</b>
1.	<b>ABB</b> menüsünde <b>Jogging</b> seçilerek jogging özellikleri görüntülenir
2.	<b>Motion mode</b> seçildikten sonra <b>Linear</b> seçeneğine ve son olarak <b>OK</b> 'e dokunulur. Daha önceden doğrusal hareket seçilmişse bu adım atlanabilir.
3.	<b>Work object</b> 'e dokunarak iş parçası seçilir
4.	<b>Coordinate System</b> seçilir, sonra <b>Work Object</b> 'e ardından <b>OK</b> 'e dokunulur.
5.	Enabling device basılı tutulur böylece mekanik birimin motorları aktive olur.
6.	Joystick hareket ettirilmediğinde mekanik birim hareket edecektir.

İPUCU: **Quickset** menüsü kullanılarak jogging modu daha hızlı seçilebilir.

### **Tool (uç elemanı) koordinatlarında jog**

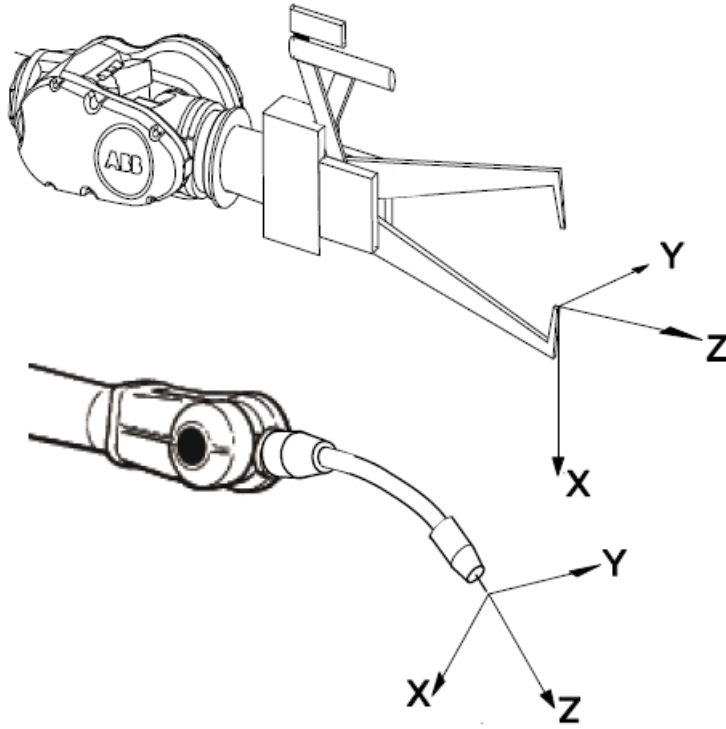
#### *Kullanım örnekleri*

Tool koordinat sistemi vida dişi açma, delik delme gibi operasyonları düzenlerken veya programlarken kullanılır.

#### *Tool koordinatlarının tanımı*

Tool (uç eleman) koordinat sistemi elemanın merkez noktasında sıfır konumuna sahiptir. Buradan hareketle elemanın konum ve oryantasyonu tanımlanır. Tool koordinat sistemi genellikle TCPF (Tool Center Point Frame) olarak kısaltılır. Tool koordinat sisteminin merkezi ise TCP (Tool Center Point) olarak kısaltılır.

Hareket esnasında uç elemanın oryantasyonunun değiştirilmek istenmediği jog işleminde (örneğin, testere bıçağını bükmeden hareket ettirmek) tool koordinat sistemi kullanışlı olacaktır.



	<b>Eylem</b>
1.	<b>ABB</b> menüsünde <b>Jogging</b> seçilerek jogging özellikleri görüntülenir.
2.	<b>Motion mode</b> seçildikten sonra <b>Linear</b> seçeneğine ve son olarak <b>OK</b> 'e dokunulur. Daha önceden doğrusal hareket seçilmişse bu adım atlanabilir.
3.	Uygun uç eleman seçilir. Eğer daha önceden tool ve/veya work object seçildiyse bu adım atlanabilir.
4.	<b>Coordinate System</b> seçilir, sonra <b>Tool</b> 'a ardından <b>OK</b> 'e dokunulur.
5.	Enabling device basılı tutulur böylece mekanik birimin motorları aktive olur.
6.	Joystick hareket ettirildiğinde mekanik birim hareket edecektir.

İPUCU: **Quickset** menüsü kullanılarak jogging modu daha hızlı seçilebilir.

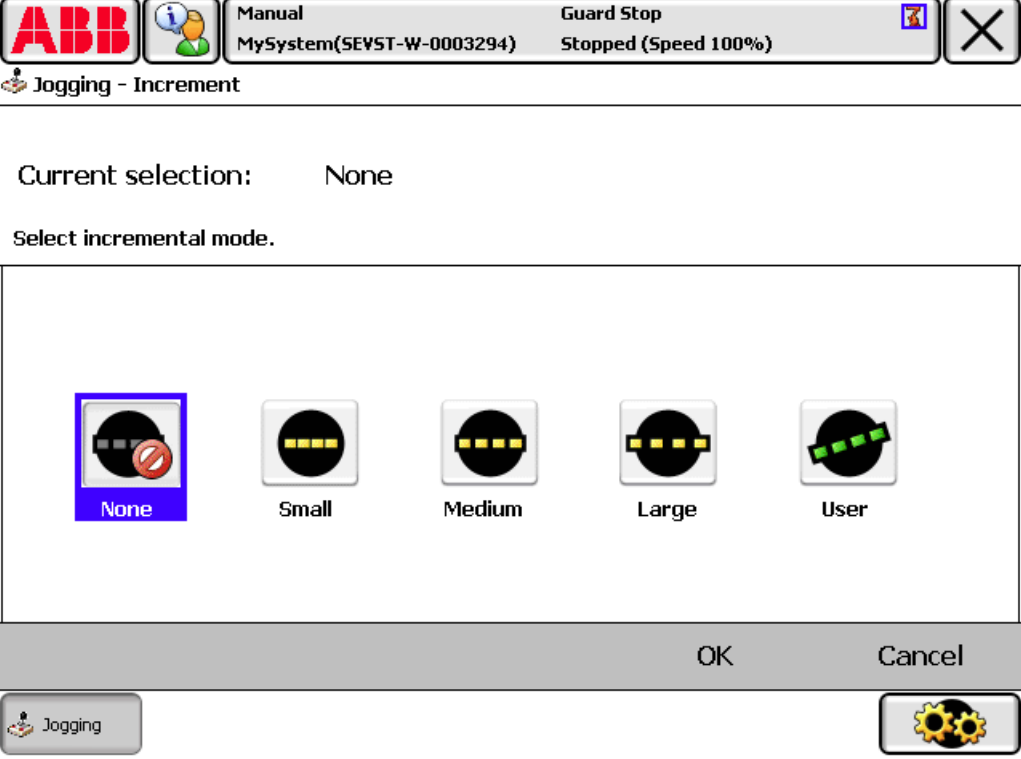
### **Incremental movement for precise positioning - Hassas konumlandırma için artırılmış hareket**

Robotu düşük genlikli yerdeştirmelerle ilerletmek için oldukça hassas konumlamayı olanaklı kılan incremental (artırılmış) hareket kullanılır.

Joystick'in her sapmasında robot bir adım (increment) ilerleyecektir. Eğer joystick'in sapma süresi bir ya da daha fazla saniye sürerse, bu adımların oluşturduğu dizi (ortalama 10 adım/saniye) joystick esas konumundan saptığı sürece gerçekleştirilecektir.

Varsayılan modda artırım yoktur, böylece joystick saptırıldığında robot daimi olarak hareket eder.

*Artırılmış hareket boyutunun belirlenmesi*

Eylem	
1.	ABB menüsünde <b>Jogging</b> seçilir.
2.	<p><b>Increment</b> seçilir.</p> 
3.	İstenilen artırım modu seçilir.
4.	<b>OK</b> 'e dokunulursa işlem bitmiş olacaktır.

Artırımlı hareket boyutu olarak small (küçük), medium (orta) veya large (büyük) artışlar arasından seçim yapılır. Kullanıcı, istediği artış miktarını tanımlayabilmektedir.

Increment (Artış)	Distance (Uzaklık)	Angular (Açısal)
Small - Küçük	0.05 mm	0.005°
Medium - Orta	1 mm	0.02°
Large - Büyük	5 mm	0.2°
User - Kullanıcı		

### Konum, tam olarak nasıl belirlenir?

*Robot konumu nasıl görüntülenir?*

Konumları her zaman;

- Uç eleman merkez noktası x, y ve z koordinatlarındaki uzaydaki nokta olarak,
- Uç eleman merkez noktasının (TCP) açısal dönüşü Euler açısı veya quaternion olarak gösterilir.

Aşağıdaki tabloda tam olarak konumun nasıl okunacağına dair prosedür verilmiştir:

Eylem	
1.	ABB menüsünde <b>Jogging</b> seçilir.
2.	Pencerenin sağında <b>Position</b> alanında konum bilgileri görüntülenecektir.

Aşağıdaki şekilde position format (konum bilgilerinin düzeni) konusunda yapılabilecek tüm ayarlamalar gösterilmektedir:

Available selections for position format:

Position shown in için seçenekler:

- World - Dünya
- Base - Temel
- Work object – İş parçası

Orientation format için seçenekler:

- Quaternion
- Euler angles

Position angle (Konum açısı) format için seçenekler:

- Angles – Açılar

Presentation angle unit için seçenekler:

- Degrees - Derece
- Radians – Radyan

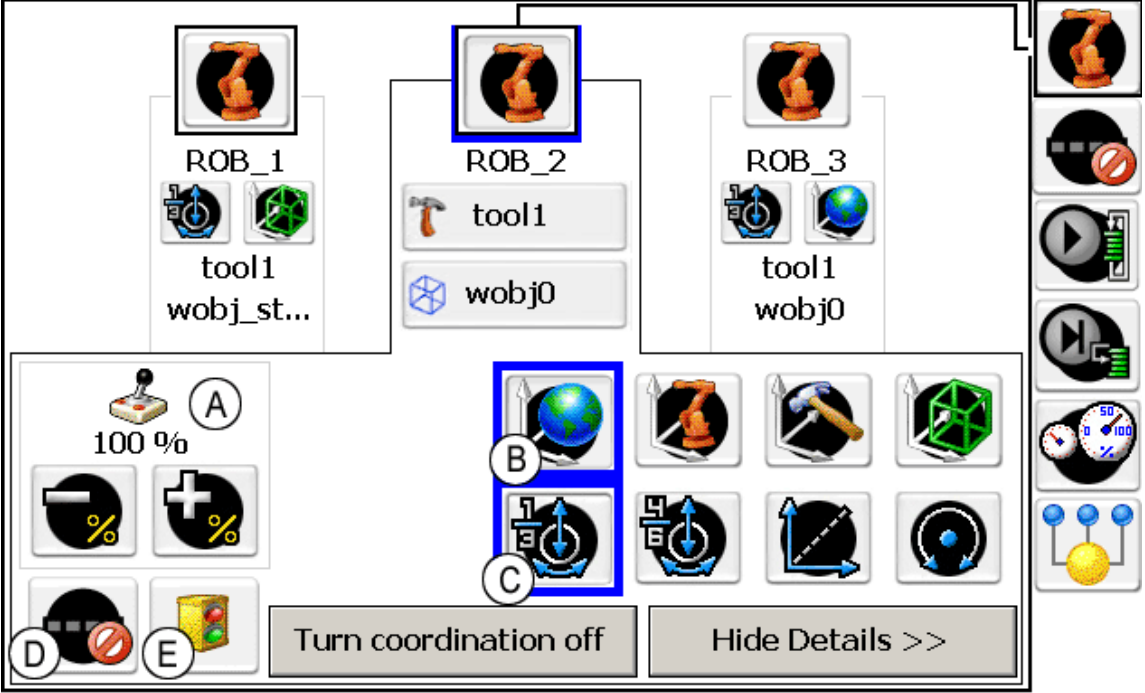
### Quickset menüsü, Mekanik birim

*Mekanik birim butonu:*

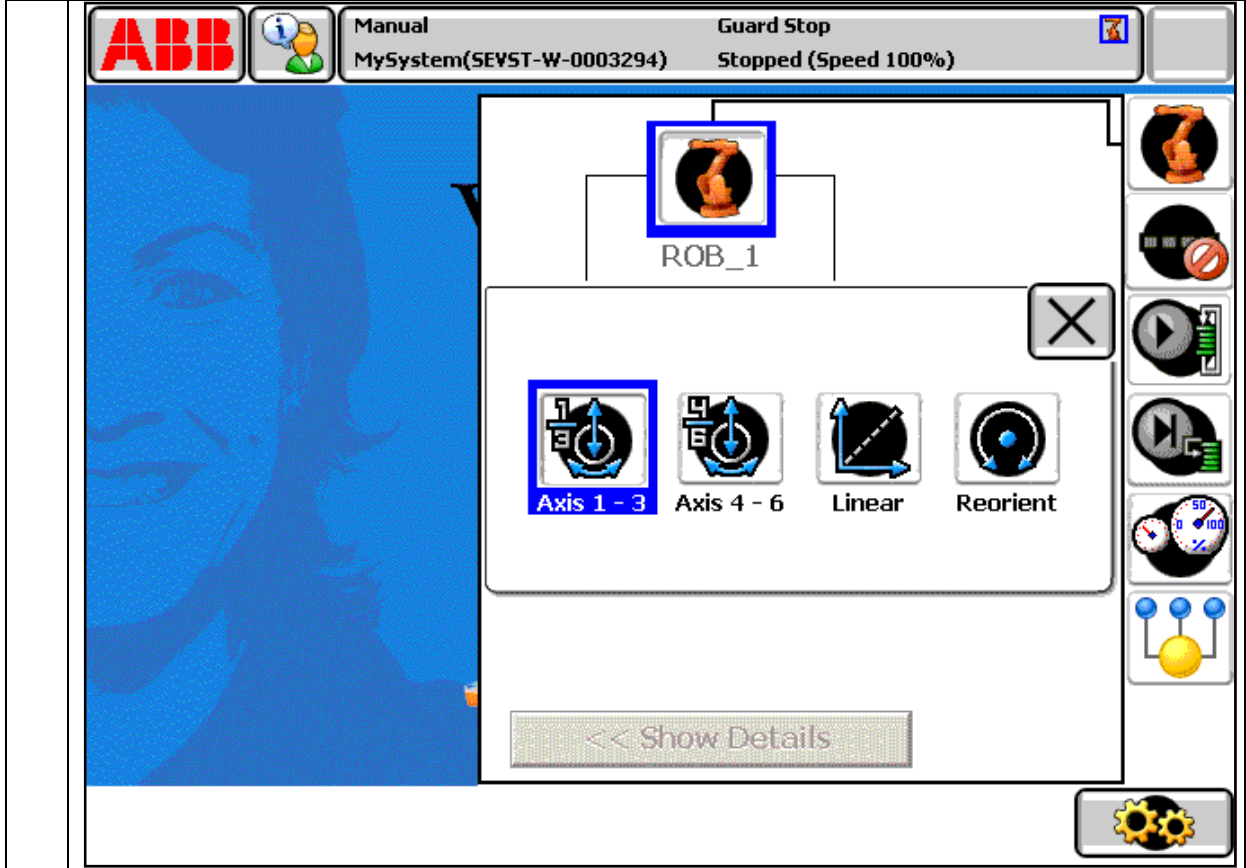
Eylem
1. Quickset menüsünde, <b>Mechanic unit</b> 'e basın; ardından mekanik birimi seçin.

Aşağıda ismi verilen düğmeler bulunmaktadır:

- A: Mekanik birim menüsü

	<ul style="list-style-type: none"> <li>• B: Mekanik birim, seçilen birim vurgulanacaktır.</li> <li>• C: Hareket modu ayarları (şekilde 1-3 eksen hareket modu seçildir)</li> <li>• D: Tool (uç eleman) ayarları (seçili tool: tool 0)</li> <li>• E: İş parçasına ait ayarlar (seçili iş parçası: work object 0)</li> <li>• F: Koordinat sistemi ayarları (seçili olan: world koordinat sistemi)</li> <li>• G: Detayları göster</li> <li>• H: Koordinasyonu kapat</li> </ul> <p>Aşağıdaki adımlarda her bir düğmenin açıklaması bulunmaktadır.</p>
2.	<p><b>Show details</b>'e dokunulursa, mekanik birim için mevcut ayarlar görüntülenecektir. Mekanik birim için yapılan tüm seçimlerin özetini gösteren bir ekran aşağıda görüntülenmektedir:</p>  <p>Şekilde harflendirilen kısımların açıklamaları ise şöyledir:</p> <ul style="list-style-type: none"> <li>• A: Jog hız ayarlarını yeniden düzenle (Mevcut seçim: %100)</li> <li>• B: Koordiant sistemi ayarları (Mevcut seçim: World koordinat sistemi)</li> <li>• C: Hareket modu ayarları (Mevcut seçim: Eksen 1-3 hareket modu)</li> <li>• D: User increment (kullanıcı tanımlı artırım)'i açma/kapama düğmesi</li> <li>• E: Jog denetimini açma/kapama düğmesi</li> </ul>
3.	<p>Hareket modu ve koordinasyon ayarları, gereken düğmeye basılarak değiştirilebilir. Yapılan herhangi bir seçimin ardından basic görünüm'e dönmek için <b>Hide details</b>'e basılır.</p>
4.	<p>Uç eleman, iş parçası, koordinasyon ve hareket modu ile ilgili değişikliklerin ayarları aşağıda verilmiştir:</p>
5.	<p>Eğer bir hareket modunun işlevselliği görüntülenmek/değiştirilmek istenirse <b>Motion mode setting</b> butonuna dokunulmalıdır. Eğer değişiklik yapılmak istenmiyorsa <b>Close</b>'a basılır.</p>

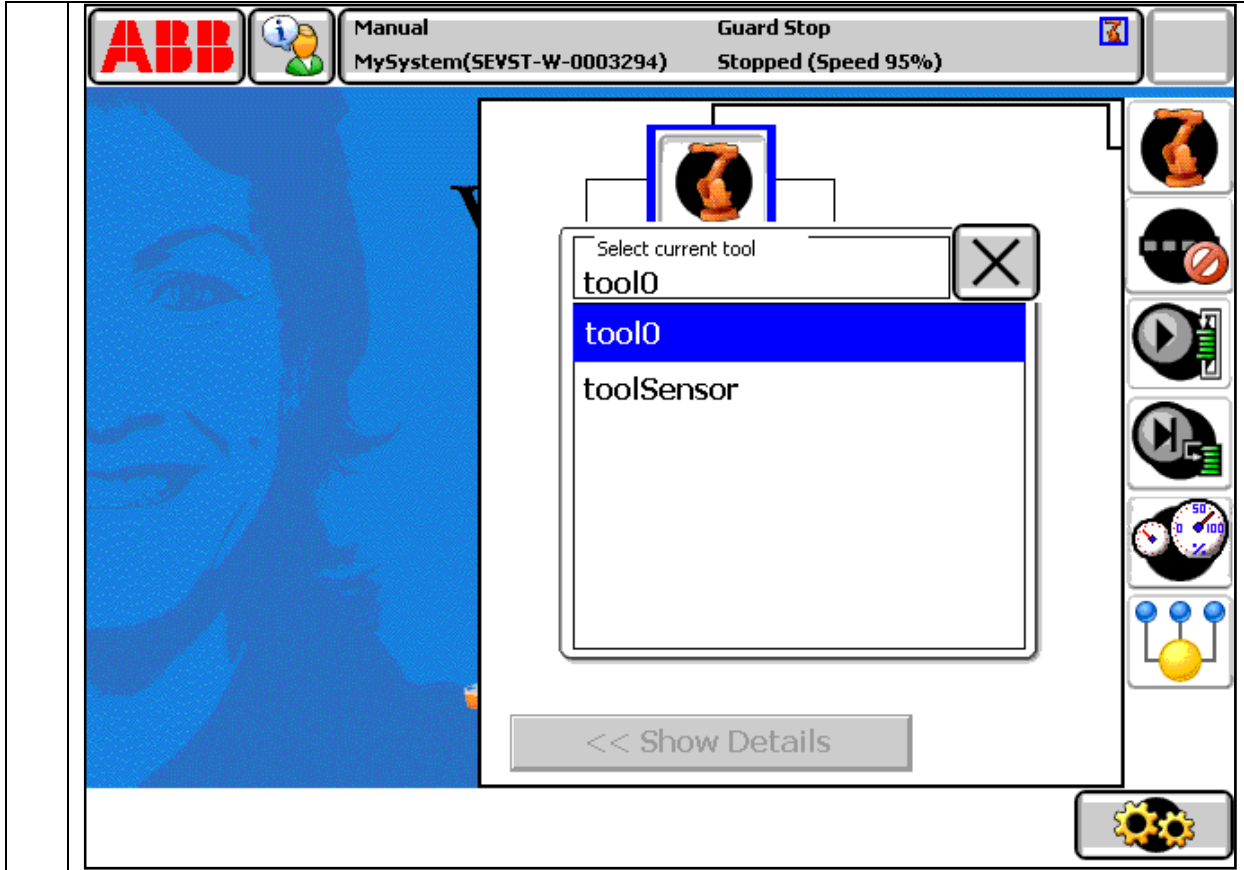




Şekilde harflendirilen kısımların açıklamaları ise şöyledir:

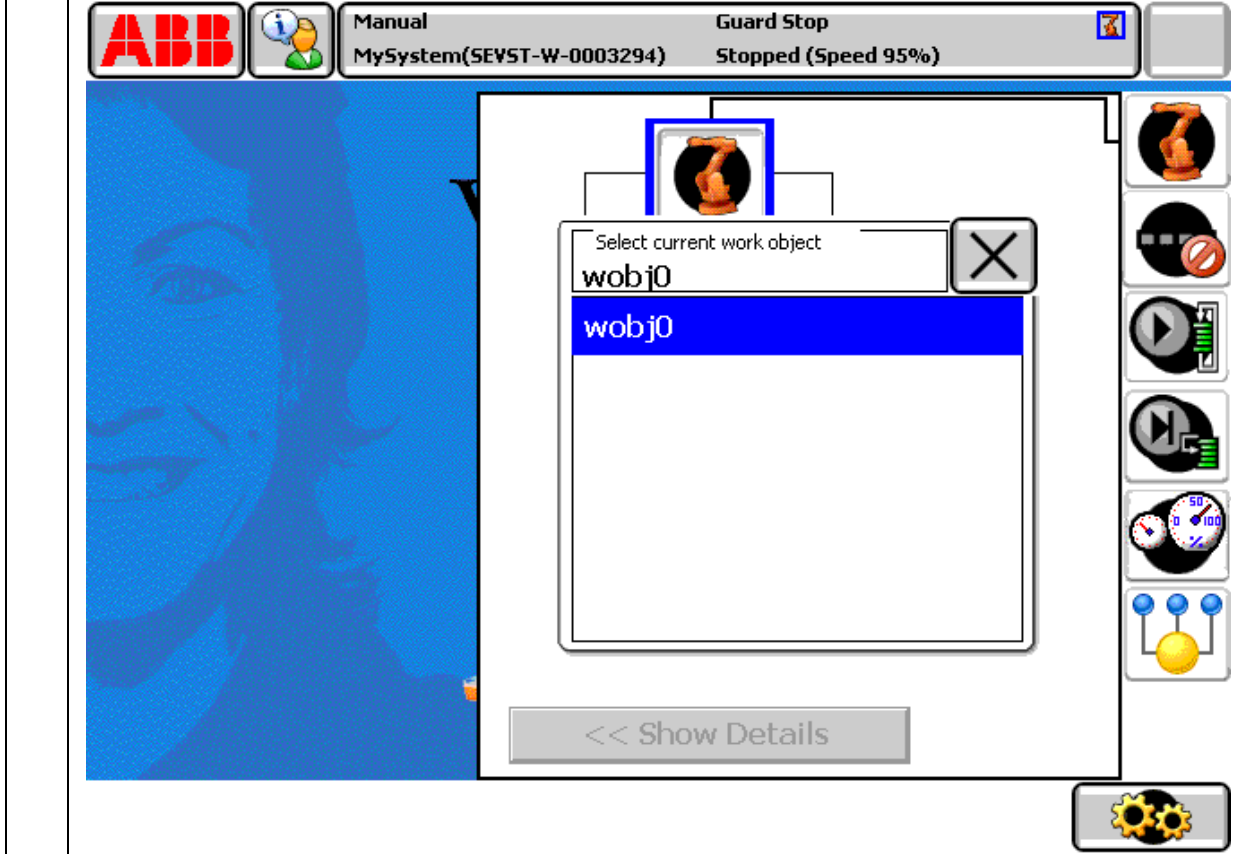
- A: Eksen 1-3 hareket modu
- B: Eksen 4-6 hareket modu
- C: Doğrusal hareket modu
- Reorient hareket modu
- Hareket modu ayarlarını kapatma butonu

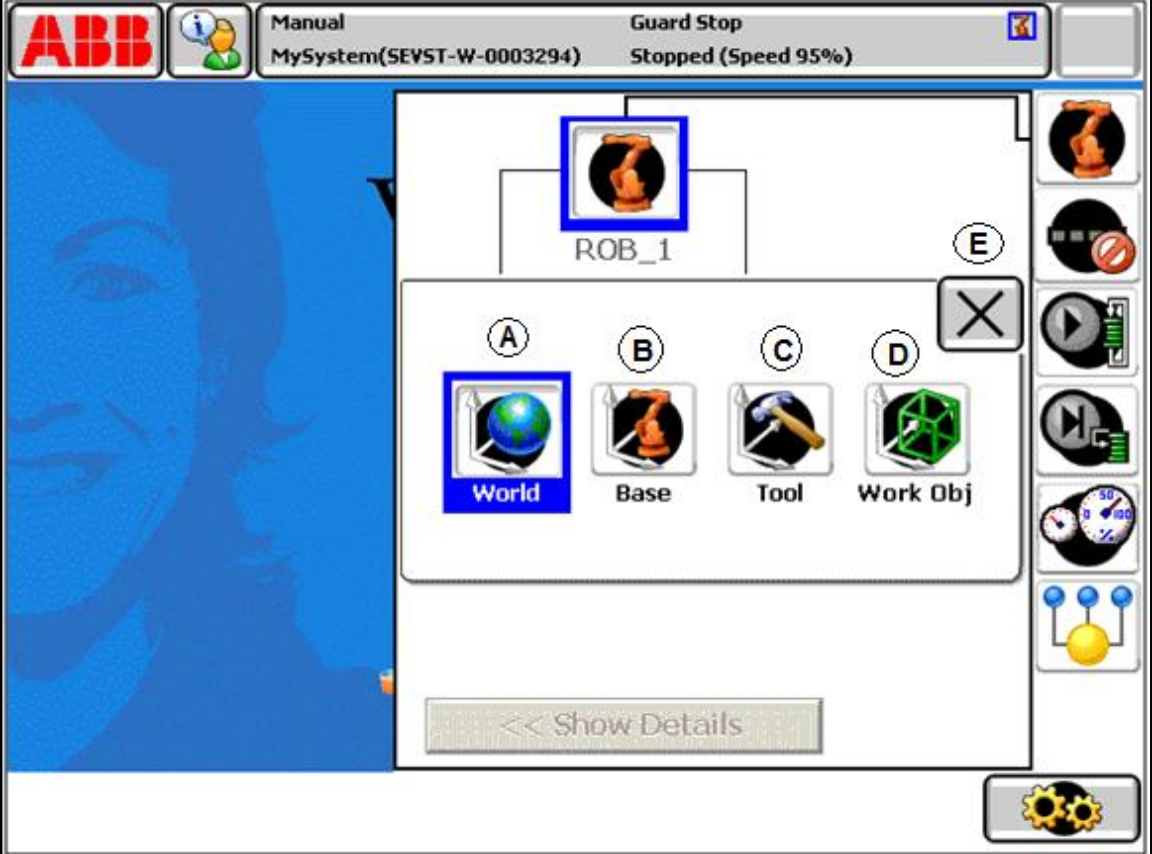
6. Mevcut uç elemanlar görüntülenmek/değiştirilmek istendiği takdirde, **Tool settings** düğmesine dokunulur. Eğer değişiklik yapılmak istenmiyorsa **Close**'a basılır.



Tüm mevcut, tanımlı elemanlar listelenir. Biri seçilerek kullanılabilir.

7. Mevcut iş parçaları görüntülenmek/değiştirilmek istendiği takdirde, **Work object settings** düğmesine dokunulur.



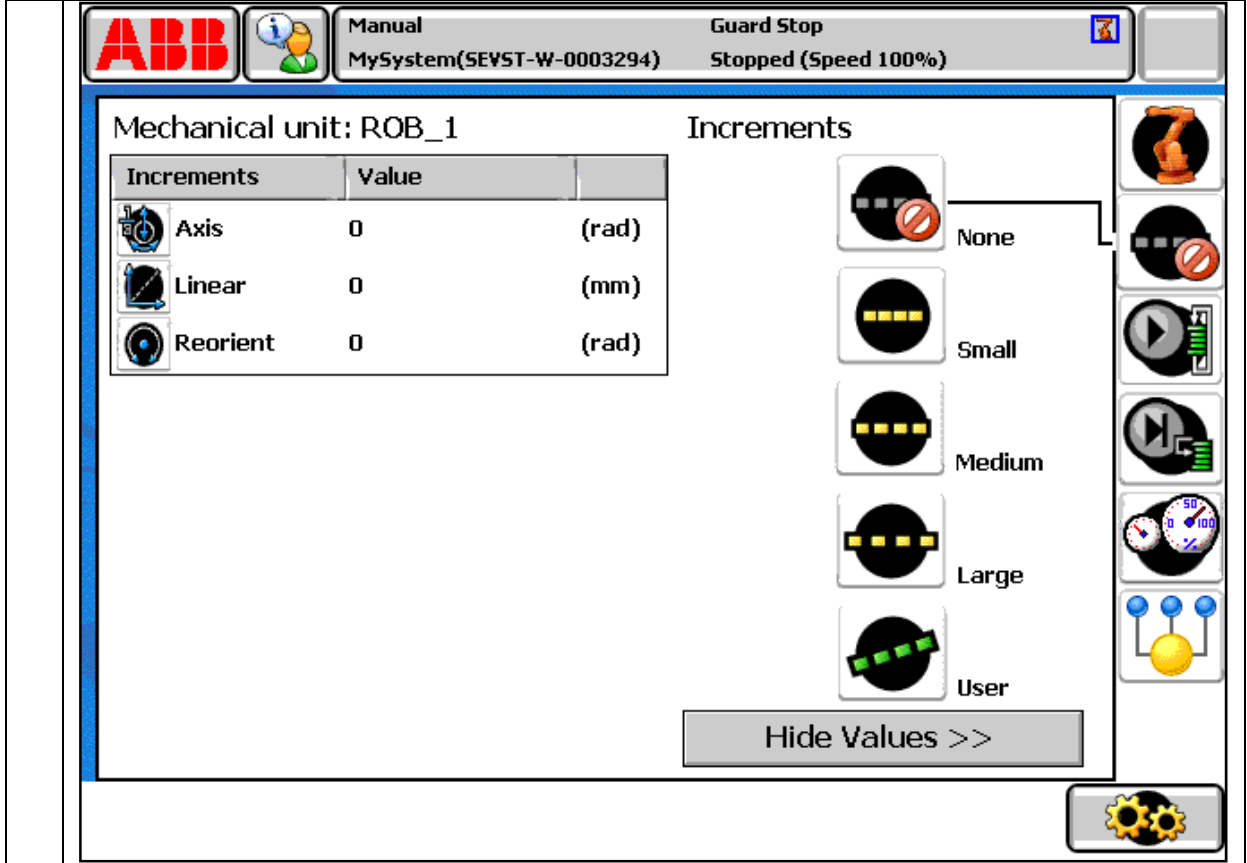
	Tüm mevcut, tanımlı iş parçaları listelenir. Biri seçilerek kullanılabilir.
8.	<p>Koordinat sisteminin işlevselliği görüntülenmek/değiştirilmek istendiği takdirde, <b>Coordinate system settings</b> düğmesine dokunulur.</p>  <p>Aşağıda belirtilen düğmeler bulunmaktadır:</p> <ul style="list-style-type: none"> <li>• World (Dünya) koordinat sistemi</li> <li>• Base (Temel) koordinat sistemi</li> <li>• Tool (Uç elemanı) koordinat sistemi</li> <li>• Work object (İş parçası) koordinat sistemi</li> </ul>

### Quickset menüsü, Increment (Artırım, Artış)

#### Increment

Bu düğmenin altındaki tüm fonksiyonlar **Jogging menüsü**'nden de ulaşılabilir.

	Eylem
1.	Artırım ile ilgili herhangi bir işlevsellik değişiminin görüntülenmesi/değiştirilmesi için <b>increment</b> düğmesine basılır.



Artış miktarı için bilgiler:

- None: Artış yok
- Small: Küçük hareket artışı
- Medium: Orta büyüklükte hareket artışı
- Large: Büyük hareket artışı
- User: Kullanıcı tanımlı hareket artışı

**Show values**'e dokunulduğunda artış değerleri görüntülenecektir.

## Programlama

### Genel:

Programlama işlemi, FlexPendant ya da RobotStudio Online vasıtasıyla gerçekleştirilebilir. Komutlar ve değişkenler uygun alternatifler arasından seçilebilir. RobotStudio Online'da, programlar serbest text formatında yazılmakta olup, "Apply Changes - Değişikleri Onayla" seçildiğinde hataları denetlenir (eğer hata yoksa, değişiklikler robot hafızasında anında etkisini gösterir).

### Programlama Ortamı:

Programlama ortamı kolaylıkla özelleştirilebilir:

- Programların, sinyallerin, sayaçların vs. adlandırılmasında atölye dili kullanılabilir.
- Uygun isimlerle yeni komutlar oluşturulabilir.
- Sık kullanılan komutlar kolayca kullanılabilecek listelerde toplanabilir.

- Konumlar, kayıtlar, uç eleman verileri yahut diğer veriler oluşturulabilir.

Programlar, program parçaları ve herhangi bir düzenleme; programı çevirme (translate- compile) ihtiyacı olmaksızın anında test edilebilir.

## RAPID

RAPID, robot sisteminin kullandığı programlama dilidir.

Aşağıda basit bir Rapid programı örneği bulunmaktadır:

```
MODULE MainModule
    VAR num uzunluk;
    VAR num genislik;
        VAR num alan;

    PROC main( )
        uzunluk := 10;
        genislik := 5;
        alan := uzunluk * genislik;
        TPWrite "Dikdörtgenin alanı: " \Num:=alan;
    END PROC
ENDMODULE
```

Bu program dikdörtgenin alanını hesaplar ve FlexPendant üzerine yazar:

```
Dikdörtgenin alanı: 50
```

## RAPID verileri

### Değişkenler

**num:** Sayısal veri. Tam sayı ya da ondalık sayı olabilir.

**string:** Dizgi, harf öbeği. En fazla 80 karakter içerebilir. Örnek: "This is a string"

**bool:** Boolean (mantıksal) değişken. Sadece DOĞRU ya da YANLIŞ değerlerini alabilir.

Tüm diğer veri tipleri yukarıda bahsedilen üçlüden temellenir.

### Değişken bildirim:

```
VAR veritipi belirteç;
```

**Örnek:**

```
VAR num uzunluk;  
VAR string isim;  
VAR bool bitti;
```

### Değer Atamak:

`:=` komutunu kullanarak değişkene değer ataması gerçekleştirilir.

```
uzunluk := 10  
isim := "Serhan"  
bitti := TRUE
```

`=` ifadesinin eşitlik anlamı taşımadığını belirtelim. Anlamı sağdaki açıklamanın soldaki değişkene ait olduğudur. İfadenin solunda sadece bir değişken bulunabilir.

```
reg1 := 2;  
reg1 := reg1 + 1;
```

Atama, değişkenin tanımlanması esnasında da yapılabilir.

```
VAR num uzunluk := 10;  
VAR string isim := "Serhan";  
VAR bool bitti := TRUE;
```

### Persistent Değişken

Persistent değişken, önemli bir farklılığının dışında temelde sıradan değişken ile aynıdır. Persistent değişken, program kapatılıp en baştan başlatılsa dahi en son atanan değerini hatırlar.

#### Persistent değişkenin tanımlanması:

PERS ifadesi persistent değişkeni tanımlamakta kullanılır. Tanımlama esnasında mutlaka bir ilk değer atanmalıdır.

```
PERS num sayi := 1;  
PERS string dizgil := "Merhaba";
```

**Örnek:** Aşağıdaki kod örneğini inceleyin.

```
PERS num nbr :=1;  
PROC main( )  
    nbr :=2;  
ENDPROC
```

## Sabitler

Bir sabit (constant), tıpkı bir deęişken gibi bir deęer ierir, fakat bu deęer tanımlama esnasında atanır ve bundan sonra bu deęer asla deęiřtirilemez. Sabit, program iinde yeni bir deęer atanamaması dıřında deęiřken ile aynı řekilde kullanılabilir.

Sabitler, CONST ifadesi ve mteakip veri tipi, belirte ve deęerin atanması ile tanımlanır.

```
CONST num yerekimi := 9.81;
```

```
CONST string karsilama := "Merhaba"
```

*Deęiřken yerine sabit* kullanıldıęında program iinde herhangi bir yerde verilen deęerin deęiřmeyeceęi kesinleřmiř olur.

## Hareket komutları

### MoveL komutu

Basit bir hareket komutu řu řekilde olabilmektedir:

```
MoveL p10, v1000, fine, tool0
```

Burada,

- MoveL robotu mevcut konumundan, belirlenen konuma doęrusal hareket etmesini saęlayan komuttur.
- p10 robotun hareket edeceęi konumu belirler.
- v1000 robotun hızının 1000 mm/s olacaęını belirtir.
- fine robotun tam olarak belirlenen konuma gideceęini ve sıradaki konuma giderken hibir kşeyi kesmeyeceęini belirtir. (*MoveL szdizimi* (s. 39) konusunu inceleyiniz)
- Tool0 belirlenen konuma hareket edecek baęlantı flanřını (robotun u kısmında bulunan) belirler.

### MoveL szdizimi

```
MoveL ToPoint Speed Zone Tool;
```

#### ToPoint

robtarget (**EK2- Robot Functions** konusunu inceleyiniz) data tipindeki bir sabit ile tanımlanan varıř noktasıdır. FlexPendant ile programlarken arzu edilen konum robot ile iřaretilenerek robtarget deęeri atanabilir. evrimdıřı programlamada, bir konumun koordinatlarını hesaplama iřlemi karmařık bir hal alabilir.

rnek: x = 600, y = -100, z = 800 konumunun tanımlanması:

```
CONST robtarget p10 := [ [600, -100, 800], [1, 0, 0, 0], [0, 0, 0, 0], [0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9] ];
```

## Speed

Hareketin hızı speeddata data tipindeki bir sabit ile tanımlanır. v5, v100, v1000 gibi önceden tanımlanmış birkaç değer bulunmaktadır. İlgili rakamlar mm/s cinsinden hızı belirtmektedir. Örneğin v5, 5 mm/s hızı ifade ederken v1000, 1000 mm/s hıza tekabül etmektedir. Bunun yanında vmax, robot için mümkün olan en yüksek hızı ifade etmektedir.

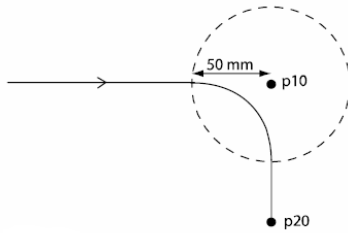
## Zone

Köşeleri, zonedata veri tipindeki bir sabit ile belirtir. Birçok önceden tanımlanmış değeri bulunmaktadır.

Önceden belirlenmiş zonedata	Değer
fine	Robot tam olarak belirlenen noktaya gider
z10	Robot güzergahı, köşeleri, ToPoint'e (varış noktası) 10 mm'den yakın olduğunda kesebilir.
z50	Robot güzergahı, köşeleri, ToPoint'e 50 mm'den yakın olduğunda kesebilir.

Aşağıdaki RAPID komutları robot güzergahının şekildeki gibi olmasını sağlayacaktır.

```
MoveL p10, v1000, z50, tool0;  
MoveL p20, v1000, fine, tool0;
```



## Tool

Robotun kullandığı uç elemanı belirtmekte olup, tooldata (**EK2- Robot Functions** konusunu inceleyiniz) veri tipindeki bir persistent değişken ile tanımlanır. Eğer bir kaynak tabancası, yapııştırıcı tabanca ya da bir kalem robota tutturulursa, bu tip uç elemanın ucu için ToPoint'i (varış noktası) tanımlamak isteriz. Bu, eğer bir tooldata daha evvelden MoveL komutunda bildirilmiş, atanmış ve kullanılmış ise kendiliğinden olacaktır.

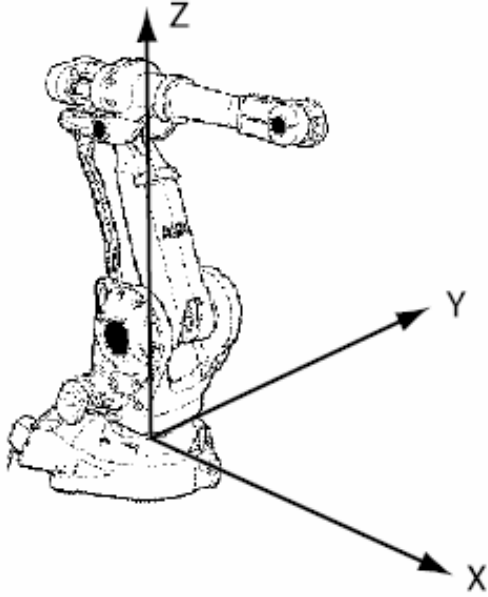
tool0 öntanımlı (predefined) bir uç eleman olup, robotun hiçbir eleman bağlanmamış halini temsil eder. tool0'ın atanma ve bildirimi yapılmamalıdır, fakat bunun dışındaki tüm uç elemanlar kullanımdan önce bildirilmeli ve tanımlanmalıdır.



## Koordinat sistemleri

### Temel koordinat sistemi – Base coordinate system

Hareket komutunun hareket ettireceği konum, koordinat sistemindeki koordinatlarıyla belirlenir. Eğer hiçbir koordinat sistemi belirlenmemişse, konum robotun temel koordinat sistemine göre verilir. (base coordinate system, ayrıca: base frame). Temel koordinat sisteminin orijini robotun tabanında yer alır. (Aşağıdaki şekli inceleyiniz),



### Özelleştirilmiş koordinat sistemleri

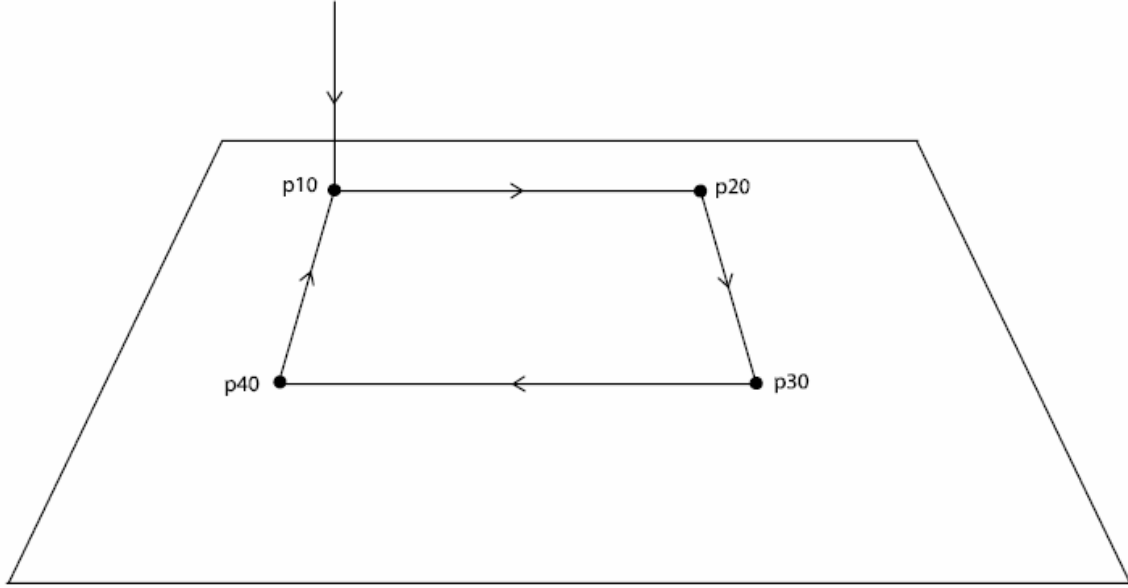
Bir diğer koordinat sistemi, hareket komutları kullanılarak tanımlanıp kullanılabilir. Hareket komutunun hangi koordinat sisteminin kullanacağı `\WObj` bağımsız değişkeni ile belirlenir. Konu ile ilgili ayrıntılı bilgi ilerleyen sayfalarda verilecektir.

```
MoveL p10, v1000, z50, tool0 \WObj:=wobj1;
```

## MoveL ile ilgili örnekler

### Kare Çizimi

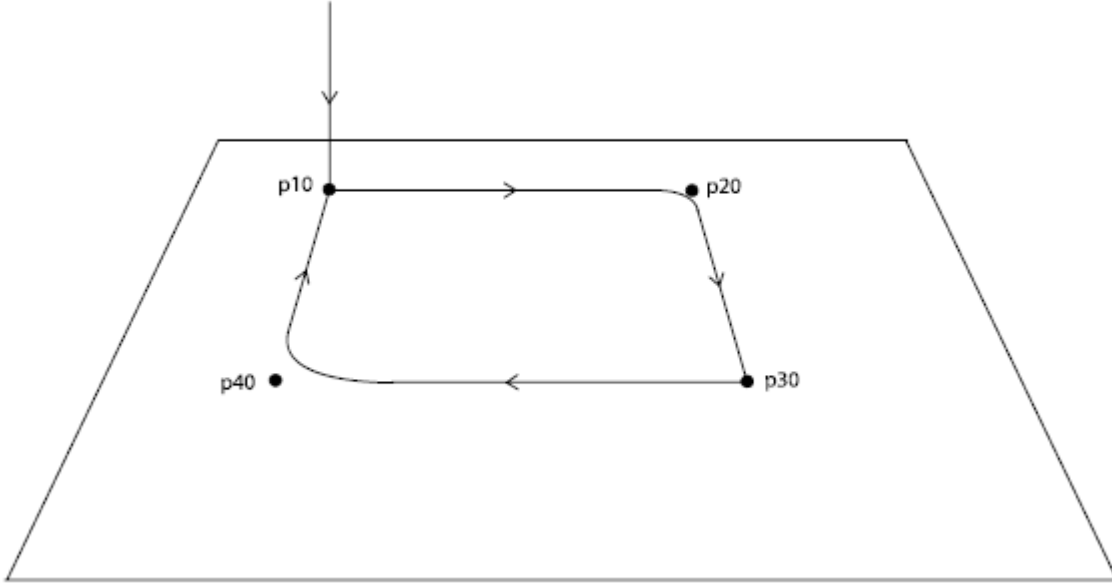
Robot bir düzlem bir yüzeyde sabit halde tutulan bir kağıt parçasının üzerinde (temassız, yukarısında) bulunmaktadır. Robottan, kalemin ucunu kağıda değene değin aşağı indirmesini daha sonra kalem ucuna kare çizdirmesini istiyoruz:



```
PERS tooldata tPen := [ TRUE, [[200, 0, 30], [1, 0, 0, 0]], [0.8,
    [62, 0, 17], [1, 0, 0, 0], 0, 0, 0]];
CONST robtarget p10 := [ [600, -100, 800], [0.707170, 0, 0.707170,
    0], [0, 0, 0, 0], [ 9E9, 9E9, 9E9, 9E9, 9E9, 9E9] ];
CONST robtarget p20 := [ [600, 100, 800], [0.707170, 0, 0.707170,
    0], [0, 0, 0, 0], [ 9E9, 9E9, 9E9, 9E9, 9E9, 9E9] ];
CONST robtarget p30 := [ [800, 100, 800], [0.707170, 0, 0.707170,
    0], [0, 0, 0, 0], [ 9E9, 9E9, 9E9, 9E9, 9E9, 9E9] ];
CONST robtarget p40 := [ [800, -100, 800], [0.707170, 0, 0.707170,
    0], [0, 0, 0, 0], [ 9E9, 9E9, 9E9, 9E9, 9E9, 9E9] ];
PROC main()
    MoveL p10, v200, fine, tPen;
    MoveL p20, v200, fine, tPen;
    MoveL p30, v200, fine, tPen;
    MoveL p40, v200, fine, tPen;
    MoveL p10,
ENDPROC
```

### Köşe bölgeleriyle beraber çizim

Önceki şekile benzer bir çizim yapılacaktır, fakat bu sefer p20 köşesinde 20 mm'lik ve p40 köşesinde 50 mm'lik bölgeler oluşturacak şekilde çizim yapılacaktır.



```

VAR tooldata tPen := ...
...
VAR robtarget p40 := ...
PROC main()
    MoveL p10, v200, fine, tPen;
    MoveL p20, v200, z20, tPen;
    MoveL p30, v200, fine, tPen;
    MoveL p40, v200, z50, tPen;
    MoveL p10, v200, fine, tPen;
ENDPROC

```

## Diğer hareket komutları

En çok kullanılan hareket komutları: MoveL, MoveJ ve MoveC'dir.

### MoveJ

MoveJ, robotu hareketin doğrusal olmasının zorunlu olmadığı zamanlarda hızlı bir şekilde bir noktadan diğerine hareket ettirmek amacıyla kullanılır.

MoveJ robotu, çalışacağı noktanın yakınındaki bir noktaya hareket ettirir. Bir MoveL komutu, örneğin, robot gövdesi mevcut nokta ile programlanan nokta arasında ise çalışmaz,. MoveJ bu tür durumlarda çalıştırılabilir.

MoveJ sözdizimi MoveL'ye benzer şekildedir:

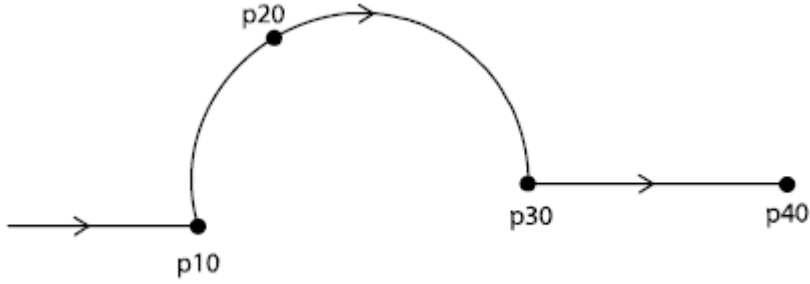
```
MoveJ p10, v1000, fine, tPen;
```

### MoveC

MoveC robotun yay parçaları çizmesinde kullanılır.

```
MoveL p10, v500, fine, tPen;
```

```
MoveC p20, p30, v500, fine, tPen;  
MoveL p40, v500, fine, tPen;
```



### Neden köşelerde özel yürütme uygulanır?

Programın yürütülmesi genellikle ifadelerin yazılış sırasıyla olur. Aşağıdaki örnekte, robot önce p10 noktasına hareket eder, daha sonra reg1 değerini hesaplar, daha sonra ise p20'ye hareket eder.

```
MoveL p10, v1000, fine, tool0;  
reg1 := reg2 + reg3;  
MoveL p20, v1000, fine, tool0;
```

Şimdi de aşağıdaki örneği inceleyiniz:

```
MoveL p10, v1000, z50, tool0;  
reg1 := reg2 + reg3;  
MoveL p20, v1000, fine, tool0;
```

Eğer reg1'in hesaplanması robot p10'da oluncaya değin başlamasaydı, robot orada durmak zorunda kalır ve bir sonraki hareket komutunu beklerdi. Aslında olan, kodun, robotun hareketinin sonrasında yürütülmesidir. reg1 hesaplanır ve robotun köşe bölgelerindeki güzergahı robot p10 noktasına ulaşmadan hesaplanır.

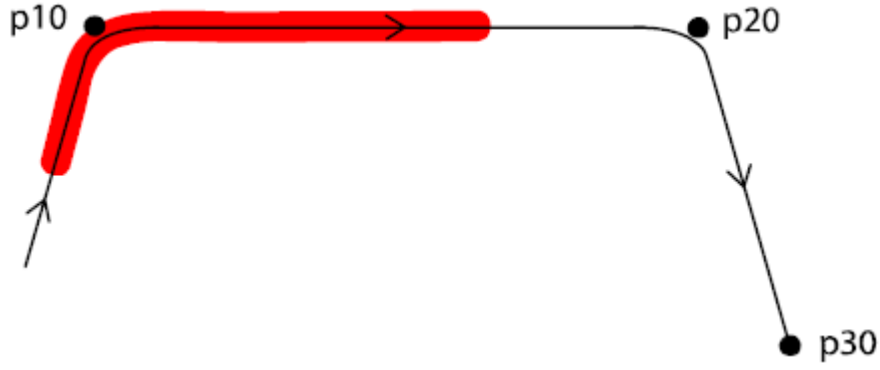
### Bu, programı nasıl etkiler?

Birçok durumda, yürütmenin zamanı programı etkilemez. Bunun yanında ne zaman etkileyeceğine dair bazı örnekler mevcuttur.

Eğer bir sprey tabancası ile p10 ve p20 arasında çizgi çizmek isterseniz, programınız şuna benzer olacaktır:

```
MoveL p10, v300, z10, tspray;  
! Start spraying  
SetDO do1, 1;  
MoveL p20, v300, z10, tspray;  
! Stop spraying  
SetDO do1, 0;  
MoveL p30, v300, fine, tspray;
```

Bu da aşağıdaki şekle benzer şekilde sonuçlanacaktır:



**Çözüm:** Eğer, köşeler için sinyalleri ayarlamak istiyorsanız, ve `fine`'i kullanmak istemiyorsanız, bir takım özel komutlar ile bu çözülebilir, örneğin `MoveLDO`, `TriggL` ve `DsipL` gibi. Bu komutlar daha sonra ayrıntılı olarak açıklanacaktır.

Robot kontrol ünitesinin köşe bölgelerini (corner zone) içeren hareket komutları arasında ne kadar sayıda (ve karmaşıklıkta) hesaplama yapacağı konusunda da kısıtlama vardır. Bu, esas olarak köşe bölgesi içeren hareket komutunun sonrasında çağrılan işlemler için problem teşkil etmektedir.

## Program Yapısı

### Rapid Prosedürü

Buraya kadar incelenen tüm kod örnekleri `main` prosedürü altında yürütülen kodlardı. Yürütme işlemi `main` prosedüründe otomatik olarak başlamaktadır ancak birbirinden farklı prosedürler de olabilir.

Prosedür; isminin, bağımsız değişkenlerinin ve yürüteceği program kodunun takip edildiği `PROC` anahtar kelimesiyle bildirilir. Bir prosedür, diğer bir prosedürden çağrılır. (`main` dışında; `main` program başlatıldığında otomatik olarak çağrılır)

### Örnek

Değişik büyüklükte dört kare çizmek istensin. Bu durumda aynı program kodunu dört defa yazmamız gerekmektedir. Fakat bu tarz düşündüğümüzde, çok fazla kod ve anlaşılması güç bir program yazmak gerekecektir.

Kısa ve doğru olarak, önce kare çizen bir prosedür yazılacak, ardından ana (`main`) prosedürü ile bu prosedür dört defa çağrılacaktır.

```
PERS tooldata tPen:= ...

CONST robtarget p10:= ...

PROC main()
  ! Call the procedure draw_square
  draw_square 100;
  draw_square 200;
  draw_square 300;
```

```

        draw_square 400;
ENDPROC

PROC draw_square(num side_size)
    VAR robtarget p20;
    VAR robtarget p30;
    VAR robtarget p40;

    ! p20 is set to p10 with an offset on the y value
    p20 := Offs(p10, 0, side_size, 0);
    p30 := Offs(p10, side_size, side_size, 0);
    p40 := Offs(p10, side_size, 0, 0);

    MoveL p10, v200, fine, tPen;
    MoveL p20, v200, fine, tPen;
    MoveL p30, v200, fine, tPen;
    MoveL p40, v200, fine, tPen;
    MoveL p10, v200, fine, tPen;
ENDPROC

```

### Prosedür Bağımsız Değişkenleri

Bir prosedürü bildirirken, bütün bağımsız değişkenler prosedür isminden sonra parantez içinde bildirilir. Bu bildirim her bağımsız değişken için veri tipini ve bağımsız değişken ismini içerir. Bağımsız değişken, değerini prosedür çağırımından alır ve prosedürün içindeki bir değişken gibi davranır.

**Not:** Bağımsız değişken prosedür dışında kullanılamaz.

```

PROC main()
    my_procedure 14, "Hello", TRUE;
ENDPROC

PROC my_procedure(num nbr_times, string text, bool flag)
    ...
ENDPROC

```

Yukarıdaki `my_procedure` prosedürünün içinde, `nbr_times` 14 değerini, `text` "Hello" değerini ve `flag` TRUE değerini alır.

Prosedürü çağırırken, bağımsız değişkenlerin sırası doğru değişkene doğru değer verilmesi açısından önemlidir. Prosedür çağırımında parantez kullanılmamaktadır.

### Prosedür İçinde Bildirilen Değişkenler

Bir prosedür içinde bildirilen değişkenler yalnızca o prosedürün içinde bulunurlar. Demek ki yukarıdaki örnekte, p10 bu modülde bütün prosedürlerde kullanılabilir, fakat p20, p30 ve p40 sadece draw\_square prosedüründe kullanılabilir.

## Modüller

Bir RAPID programı bir veya birkaç modül içerebilir. Her modül bir veya birkaç prosedür içerebilir.

Bu el kitabında gösterilen küçük ve basit programlar sadece bir modül kullanmaktadırlar. Daha karmaşık programlama çevrelerinde çoğu değişik program tarafından kullanılan bazı standart prosedürler ayrı bir modülde yer alabilir.

## Örnek

MainModule modülü bu programa özgü kodları içerir ve robotun programda ne yapması gerektiğini belirtir. figures\_module modülü kare, üçgen veya daire çizmek isteyen herhangi bir program tarafından kullanılacak standart kodları içerir.

```
MODULE MainModule
  ...
  draw_square;
  ...
ENDMODULE

MODULE figures_module
  PROC draw_square()
    ...
  ENDPROC
  PROC draw_triangle()
    ...
  ENDPROC
  PROC draw_circle()
    ...
  ENDPROC
ENDMODULE
```

## Örnek Program

Çalışmanın başlangıcında robotun bağlantı plakasına kalem takılmıştır ve yazılan program ile kalem ucunun iş parçası üzerinde tanımlanan noktalardan geçerek kare şekli çizmesi amaçlanmıştır.

- Robota takılan kalemin *tool* (*uç eleman*) olarak tanımlanması gerekecektir.
- Çizim yapılacak yüzeyin *iş parçası* (*work object*) olarak tanımlanması gerekecektir.

```
CONST robtarget p10:= [[59.57, -28.48, -562.64], [0.162839, 0.521882, -0.498266, 0.672944], [-1, -1, 0, 1], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
```

```

CONST robtarget p20:=[[50,100,0],[0.162838,0.521883,-0.498265,0.672944],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p30:=[[100,50,0],[0.16284,0.521884,-0.498264,0.672944],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p40:=[[150,100,0],[0.162839,0.521881,-0.498265,0.672946],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p50:=[[100,150,0],[0.16284,0.521883,-0.498263,0.672945],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PROC main()
  MoveL p10, v50, z0, kalem\WObj:=wobj_kagit;
  MoveL p20, v50, z0, kalem\WObj:=wobj_kagit;
  MoveL p30, v50, z50, kalem\WObj:=wobj_kagit;
  MoveL p40, v50, z50, kalem\WObj:=wobj_kagit;
  MoveL p50, v50, z50, kalem\WObj:=wobj_kagit;
  MoveL p20, v50, z50, kalem\WObj:=wobj_kagit;
  MoveL p10, v50, z0, kalem\WObj:=wobj_kagit;
ENDPROC

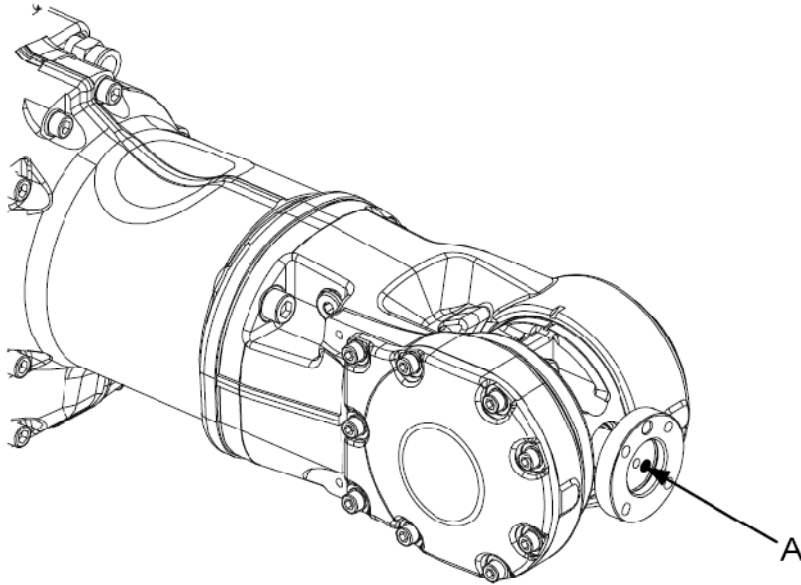
```

## Uç eleman (Tool) Yaratma

Yeni bir tool yarattığınızda veri tipi `tooldata` olan bir değişken yaratırsınız. Değişkenin ismi de elemanın ismi olur. Uç elemanın kullanılması için tanımlanması gerekli olan ağırlık, eksen takımı, oryantasyon (açısal konum) gibi başlangıç değerleri vardır.

### Tool nasıl yaratılır?

Önceden tanımlanmış uç elemanın (`tool0`) merkez noktası (tool center point - TCP) robotun bağlantı başlığındadır (mounting flange) ve robotun tabanıyla aynı oryantasyondadır. Yeni bir tool yaratılınca başka bir TCP tanımlanır.



en0400000779

A Tool center point, TCP, for tool0

### Eylem

1. **ABB** menüsünde, **Jogging**'e dokununuz.
2. Uygun elemanların listesini görmek için **Tool**'a dokununuz.



3. Yeni bir tool yaratmak için **New...**'e dokununuz.

4. **OK**'e dokununuz.

The screenshot shows the 'New Data Declaration' dialog box in the ABB software. The dialog has a title bar with the ABB logo, 'Manual MySystem(SEVST-W-0003294)', and 'Guard Stop Stopped (Speed 100%)'. The main area is titled 'New Data Declaration' and contains the following fields:

- Data type: tooldata
- Current Task: T\_ROB1
- Name: tool1 (with a dropdown arrow)
- Scope: Global (dropdown)
- Storage type: Persistent (dropdown)
- Task: T\_ROB1 (dropdown)
- Module: MainModule (dropdown)
- Routine: <None> (dropdown)
- Dimension: <None> (dropdown)

At the bottom, there are buttons for 'Initial Value', 'OK', and 'Cancel'. There is also a 'Jogging' button and a gear icon. The version number 'en0300000544' is visible at the bottom left.

Her bölüme uygun değerleri giriniz (Aşağıdaki tabloyu inceleyiniz).

### Uç elemanın tanımlama ayarları

...değiştirmek için	yapmanız gereken...	tavsiye
elemanın ismi	ismin yanındaki ... düğmesine dokununuz	Uç elemanları otomatik olarak tool10 ya da tool21 örneklerinde görüldüğü gibi bir numaranın eşlik ettiği tool ismiyle tanımlanırlar. Bunu püskürtücü, pompa ya da kaynak gibi daha açıklayıcı kelimelerle değiştirmeniz tavsiye edilir. <b>Not!</b> Eğer elemanın ismini herhangi bir programda kullandıktan sonra değiştirirseniz kullanılan programda da isim değiştirmeniz gerekmektedir.
kapsamı	menüden arzu edilen kapsamı seçiniz	Eğer uç elemanların bütün program modüllerinde kullanılabilir olması isteniyorsa, <i>global</i> olmalıdır.
bellekleme tipi	-	Uç elemanlara ait bütün değişkenler <i>persistent değişken</i> olmalıdır.
modül	eleman hangi modülde kullanılacaksa menüden bu modül seçilmelidir	

### Tool frame'i (Uç elemanın eksen takımı) tanımlama

Yeni bir tool frame tanımlamak için yapmanız gereken ilk şey dünya koordinat sisteminde (world coordinate system) bir referans noktası belirlemektir. Tanımlamada genel olarak kullanılan üç yöntem vardır. Üçü de TCP'nin kartezyen koordinatlarda tanımlanmasını gerektirir. Değişen tek şey ise açılal konumun nasıl tanımlandığıdır.

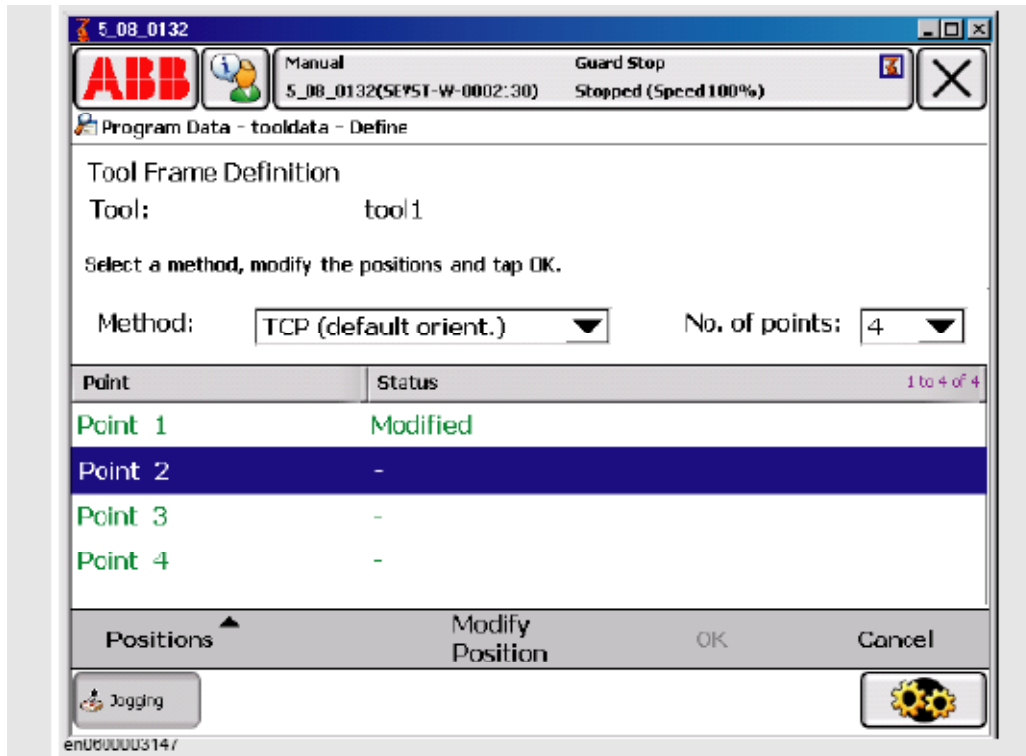
...istiyorsanız	...seçiniz
Açılal konumu (oryantasyon) robotun bağlantı plakasıyla aynı seçmek	<b>TCP (önceden tanımlanmış oryantasyon)</b>
Açılal konumu Z ekseninde yapmak	<b>TCP &amp; Z</b>
Açılal konumu X ve Z ekseninde yapmak	<b>TCP &amp; Z,X</b>

### Yöntem Seçmek

Bu prosedür, uç eleman tanımlanırken izlenecek yolu göstermektedir:

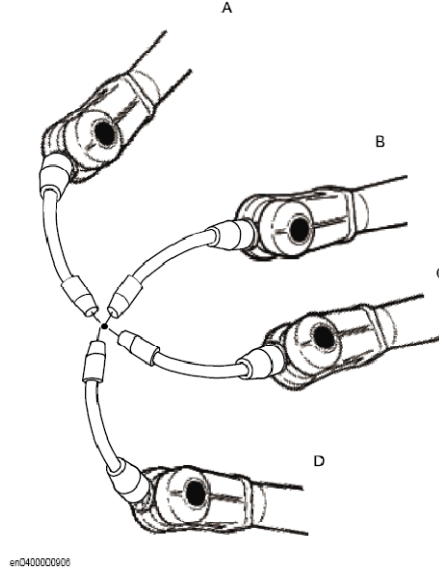
### Eylem

1. **ABB** menüsünde, **Jogging**'e tıklayınız.
2. Uygun uç elemanların listesini görmek için **Tool**'a tıklayınız.
3. Tanımlamak istediğiniz elemanı seçiniz.
4. **Edit** menüsünde, **Define**'a dokununuz.
5. Çıkan iletişim kutusunda istediğiniz metodu seçiniz.
6. Ne kadar *yaklaşım noktası* istiyorsanız seçiniz. Genelde 4 nokta yeterli olmaktadır. Ancak daha iyi sonuç almak için daha fazla nokta seçmek istiyorsanız eşit hassasiyette tanımlamaya özen gösteriniz.



### Tool frame tanımlamasıyla ilerlemek

Bu prosedür TCP'nin kartezyen koordinatlarda nasıl tanımlandığını anlatmaktadır.



	Eylem	Bilgi
1.	İlk yaklaşım noktası için robotu kontrol çubuğuyla uygun bir pozisyona, (örneğin) A'ya, hareket ettiriniz.	Uç elemanın daha iyi konum alması için küçük artırımlarla (small increments) çalışın ve referans noktasına mümkün olduğunca çok yaklaşın.
2.	Noktayı tanımlamak için <b>Modify Position</b> 'a dokununuz.	
3.	Adım 1 ve 2'yi her nokta tanımlaması için (B, C ve D gibi) tekrar edin	Kontrol çubuğu ile sabit dünya noktasını (fixed world point) değiştiriniz. Sadece uç eleman oryantasyonunu değiştirmek uygun bir sonuç vermeyecektir.
4.	Eğer TCP & Z ya da TCP & Z, X metodundan birini kullanıyorsanız oryantasyon da mutlaka tanımlanmalıdır.	
5.	Eğer herhangi bir sebepten dolayı yeniden 1 – 4 adımlarındaki kalibrasyonu yapmanız gerekirse <b>Positions</b> menüsündeki <b>Reset All</b> 'a dokununuz.	
6.	Bütün noktalar tanımlandığında bunu tekrar kullanmanızı sağlayacak biçimde bir dosyaya kaydedebilirsiniz. <b>Positions</b> menüsündeki <b>Save</b> 'e dokununuz.	
7.	<b>OK</b> 'e dokununuz. <b>Calculation Result</b> iletişim kutusu size sonucun kontrol ünitesine yazılmadan önce iptal edileceğini veya onaylanacağını soracaktır.	

**Uç eleman (tool), iş parçası (work object) ve yükün (payload) belirlenmesi**

Uygun bir uç elemanın, iş parçası'nın ve yükün belirlenmesi her zaman önemlidir. Özellikle, hedef konumlara kontrol çubuğu ile gidilen bir program yazılıyorsa önemi artacaktır. Bunun yapılmadığı durumlar genellikle aşırı yük hatalarına ve/veya kontrol çubuğuyla veya üretimde yürütülen program ile kontrol yapılırken yanlış konumlamaya sebep olmaktadır.

### **Uç eleman, iş parçası ve yükün belirlenmesi**

#### **Eylem**

1. **ABB** menüsünde, kontrol çubuğu özelliklerini görmek için **Jogging** seçeneğini seçiniz.
2. Uygun uç eleman, iş parçası ya da yük listelerini görmek için **Tool**, **Work Object** ya da **Payload**'dan birine dokununuz.
3. Seçilen uç eleman, iş parçası ya da yükten sonra **OK**'ye dokununuz.

### **Uç elemana ait verilerinin düzenlenmesi**

#### **Tool Data**

Ağırlık ve ağırlık merkezi gibi fiziksel özellikleri ve TCP'yi ayarlamak için değer ayarlarını (value settings) kullanınız.

### **Uç elemanın verilerini görüntülemek**

#### **Eylem**

1. **ABB** menüsünde **Jogging**'e dokununuz.
2. Uygun elemanların listesini görmek için **Tool**'a dokununuz.
3. Düzenlemek istediğiniz elemanı seçiniz ve **Edit**'e dokununuz.

Bir menü karşınıza çıkacaktır.

- Tanımı değiştirme (Change Declaration)
- Değeri değiştirme (Change Value)
- Silme (Delete)
- Tanımlama (Define)

4. Menüde, **Change Value**'a dokununuz.

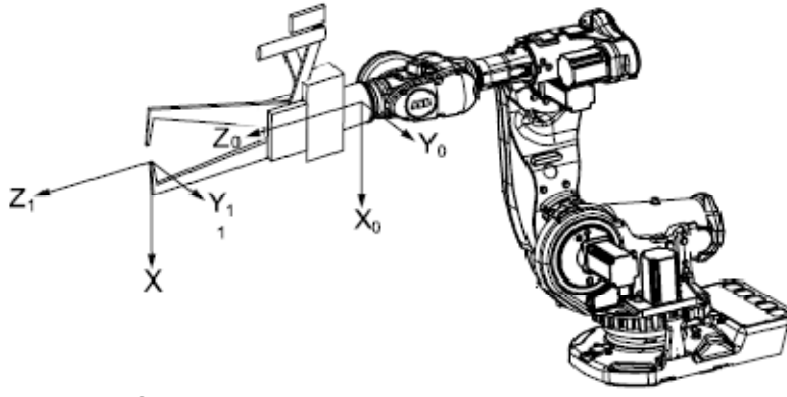
Elemanı tanımlayan veriler karşınıza çıkacaktır. Yeşil metin değerini değiştirebileceğini belirtmektedir.

5. Aşağıda anlatıldığı gibi verileri değiştirmeyi sürdürebilirsiniz.

### **Uç eleman merkez noktasının (TCP) ölçülmesi**

TCP'yi tanımlamanın en kolay yolu *Tool frame'i (Uç elemanın eksen takımı) tanımlama* bölümünde anlatılan yöntemi kullanmaktır. Eğer bu yöntemi kullanırsanız düzlem için başka değerler yazmak zorunda kalmazsınız.

Eğer elemana ait ölçümler varsa veya bir sebepten dolayı elle ölçmek istiyorsanız, değerler uç eleman verilerine yazılabilir.



en0400000881

$X_0$	X axis for tool0
$Y_0$	Y axis for tool0
$Z_0$	Z axis for tool0
$X_1$	X axis for the tool you want to define
$Y_1$	Y axis for the tool you want to define
$Z_1$	Z axis for the tool you want to define

### Eylem

1. Robotun bağlantı başlığı ile TCP arası mesafeyi tool 0'ın X eksenini boyunca ölçün.
2. Robotun bağlantı başlığı ile TCP arası mesafeyi tool 0'ın Y eksenini boyunca ölçün.
3. Robotun bağlantı başlığı ile TCP arası mesafeyi tool 0'ın Z eksenini boyunca ölçün.

### Uç elemanın tanımını düzenleme

	Eylem	Örnek	Birim
1.	TCP'nin kartezyen koordinatlarını giriniz.	tframe.trans.x tframe.trans.y tframe.trans.z	[mm]
2.	Eğer gerekiyorsa, eksen takımı oryantasyonunu giriniz.	tframe.rot.q1 tframe.rot.q2 tframe.rot.q3	Yok
3.	Uç Elemanın ağırlığını giriniz.	tload.mass	[kg]
4.	Eğer gerekiyorsa, uç elemanın ağırlık merkezini giriniz.	tload.cog.x tload.cog.y tload.cog.z	[mm]
5.	Eğer gerekiyorsa, uç elemanın moment eksenini yönelimini giriniz.	tframe.aom.q1 tframe.aom.q2 tframe.aom.q3 tframe.aom.q4	Yok
6.	Eğer gerekiyorsa, uç elemanın eylemsizlik momentini giriniz.	tload.ix tload.iy tload.iz	[kgm <sup>2</sup> ]

7.	Yeni değerleri girmek için <b>OK</b> 'e dokununuz, değişiklik yapmadan çıkmak için ise <b>Cancel</b> 'a dokununuz.		
----	--	--	--

### Bir uç elemanını silme

#### Eylem

1. **ABB** menüsünde **Jogging**'e dokununuz.
2. Uygun uç elemanların listesini görmek için **Tool**'a dokununuz.
3. Silmek istediğiniz uç elemanını seçiniz ve **Edit**'e dokununuz.
4. Seçilen elemanı silmek için **Delete**'e dokununuz.  
Bir onay ileti kutusu açılacaktır.
5. Silmek istiyorsanız ileti kutusunda **Yes**'e, istemiyorsanız **No**'ya basınız.

#### UYARI!



Silinen uç eleman, iş parçası ve yük tekrar kurtarılamaz, ve bağlantılı tüm bilgi kaybedilir. Eğer silinen uç eleman, iş parçası ya da yük herhangi bir program tarafından kullanılıyorsa, bu programlar değişiklik yapılmadan çalışmaz.  
Eğer bir uç eleman silinirse program kaldığı yerden yürütülemez.

### İş parçası (work object) yaratma

Yeni bir iş parçası yarattığınızda veri tipi `wobjdata` olan bir değişken yaratırsınız. İş Parçası yaratıldığında iş parçasının koordinat sistemi dünya koordinat sistemiyle aynı olur.

#### Eylem

1. **ABB** menüsünde, **Jogging**'e dokununuz.
2. Uygun iş parçalarının listesini görmek için **Work object**'a dokununuz.
3. Yeni bir iş parçası yaratmak için **New...**'e dokununuz.
4. **OK**'ye dokununuz.

#### İş parçası tanımlama ayarları

...değiştirmek için	Yapmanız gereken...	Tavsiye
iş parçasının ismi	ismin yanındaki ... düğmesine dokununuz	İş parçaları otomatik olarak <code>wobj10</code> ya da <code>wobj33</code> gibi bir numaranın eşlik ettiği <code>wobj</code> ismiyle tanımlanırlar. Bunu daha açıklayıcı kelimelerle değiştirmeniz tavsiye edilir. <b>Not!</b> Eğer iş parçasının ismini herhangi bir programda kullandıktan sonra değiştirirseniz kullanılan programda da isim değiştirmeniz gerekmektedir.
kapsamı	menüden arzu edilen kapsamı seçiniz	Eğer iş parçalarının bütün program modüllerinde kullanılabilir olması isteniyorsa, evrensel (global) olmalıdır.

bellekleme tipi	-	Bütün iş parçası değişkenleri kararlı (persistent) olmalıdır.
modül	iş parçası hangi modülde kullanılacaksa menüden bu modül seçilmelidir	

## İş Parçası Koordinat Sistemini Tanımlamak

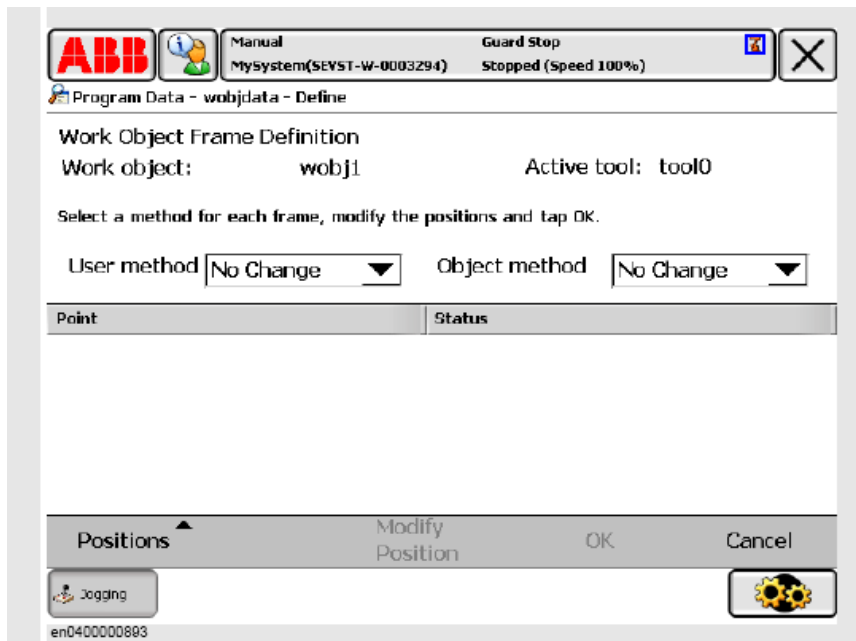
Bir iş parçasını tanımlamak, konumun gösterilmesi için robotun kullanılması anlamına gelir. Bu, X ekseninde iki nokta ve Y ekseninde bir nokta olmak üzere toplam üç noktanın tanımlanmasıyla olur. İş parçasını tanımlarken kullanıcı düzlemini (user frame), parça düzlemini (object frame) ya da her iki düzlemi birden kullanabilirsiniz. Genelde kullanıcı düzlemi ve parça düzlemi çakışır. Aksi durumda, parça düzlemi kullanıcı düzleminde ayrılır.

### Yöntem Seçmek

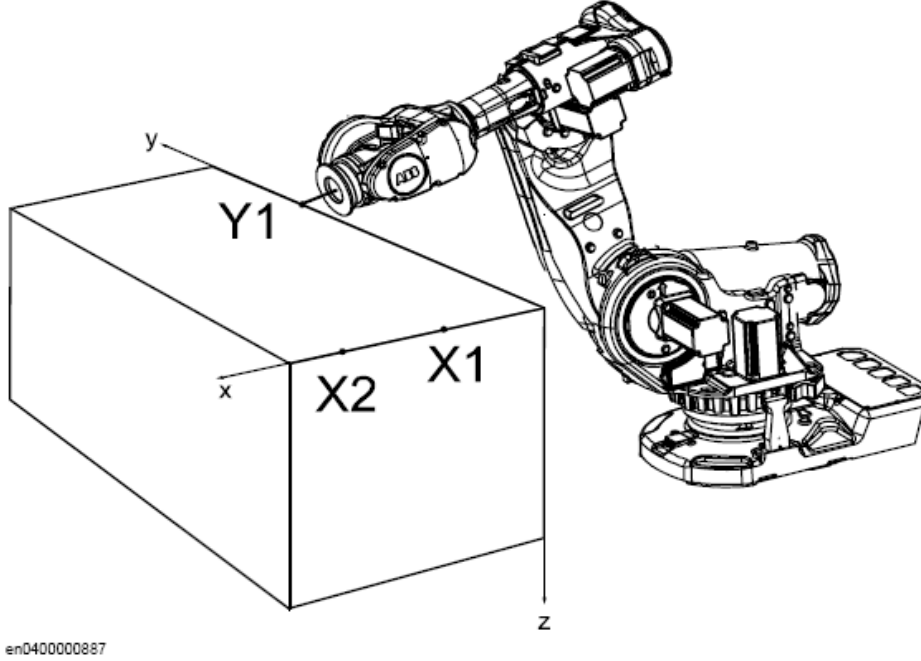
Bu yordam, kullanıcı düzlemi ya da parça düzlemi yöntemlerinden birinin ya da her ikisinin birden nasıl seçileceğini anlatmaktadır. Ancak bu yordamın önceden tanımlı iş parçası wobj0 için değil, yalnızca kullanıcı tarafından oluşturulmuş iş parçaları için kullanılabilir. İş parçası tanımlama aynı zamanda Program Verileri (Program Data) penceresinden de yapılabilir.

### Eylem

1. **ABB** menüsünde, **Jogging**'e dokununuz.
2. Uygun iş parçalarının listesini görmek için **Work object**'e dokununuz.
3. Tanımlamak istediğiniz iş parçasını seçip **Edit**'e dokununuz.
4. **User method** (Kullanıcı yöntemi) ve/veya **Object method** (Parça yöntemi) açılır menüsünden istediğiniz yöntemi seçiniz.



## Kullanıcı Düzlemini Tanımlamak

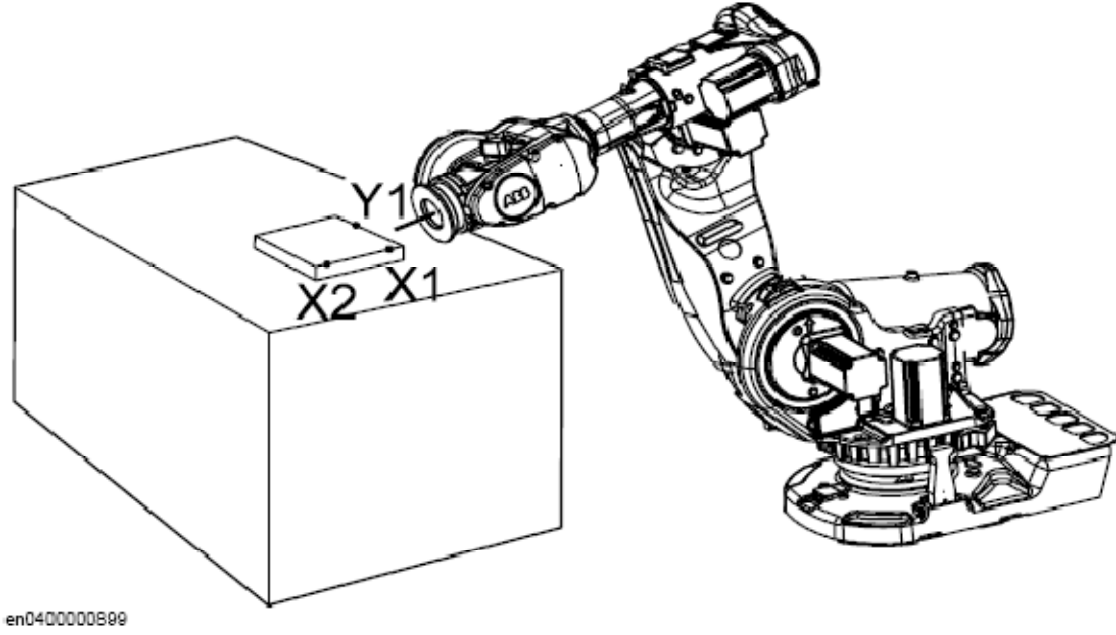


X eksenini X1-X2 noktalarından geçmekte, Y eksenini de Y1 noktasından geçmektedir.

	Eylem	Bilgi
1.	<b>User method</b> açılır menüsünde, <b>3 points</b> 'e dokununuz.	
2.	Etkinleştiriciye basınız ve robotu kontrol çubuğuyla il noktaya (X1, X2 veya Y1) hareket ettiriniz.	X1 ve X2 arasında mümkün olduğunca büyük mesafe seçilmesi daha uygun bir tanımlama yapmak için gereklidir.
3.	Listedeki noktaya dokununuz.	
4.	Noktayı tanımlamak için <b>Modify Position</b> 'a basınız.	
5.	Kalan noktalar için 2-4 adımlarını yeniden uygulayınız.	



## Parça Düzlemini Belirlemek



X eksenini X1-X2 noktalarından geçmekte, Y eksenini de Y1 noktasından geçmektedir.

### Eylem

1. **Parça yöntemi** açılır menüsünde, **3 points**'e dokununuz.
2. [Kullanıcı Düzlemini Tanımlamak](#) bölümünde 2-4 adımlarına bakın.

## İş Parçası Verilerini Düzenlemek

### Genel Açıklama

Kullanıcı ve parça düzlemi konumunu ve yönelimini ayarlamak için iş parçası verileri tanımını kullanın.

### İş Parçası Verilerini Görüntülemek

#### Eylem

1. **ABB** menüsünde **Jogging**'e dokununuz.
2. Uygun iş parçalarının listesini görmek için **Work object**'e dokununuz.
3. Düzenlemek istediğiniz iş parçasını seçiniz ve **Edit**'e dokununuz.
4. **Change Value**'ya dokununuz.

Düzenlemek istediğiniz iş parçası verileri karşınıza çıkacaktır.

### Kullanıcı ve Parça Düzlemi Değerlerini Elle Girmek

İş parçası ve kullanıcı koordinat sistemleri pozisyonlarını ayarlamanın en kolay yolu [İş Parçası Koordinat Sistemini Tanımlamak](#) bölümündeki yöntemleri kullanmaktır. Ne de olsa aşağıdaki rehberi kullanarak elle değerleri düzenleyebilirsiniz.

Değerler	Örnek	Birim
Parça düzlemi konumu kartezyen koordinatları	oframe.trans.x oframe.trans.y oframe.trans.z	[mm]
Parça düzlemi yönelimini	oframe.rot.q1 oframe.rot.q2 oframe.rot.q3 oframe.rot.q4	Yok
Kullanıcı düzlemi konumu kartezyen koordinatları	uframe.trans.x uframe.trans.y uframe.trans.z	[mm]
Kullanıcı düzlemi yönelimi	uframe.rot.q1 uframe.rot.q2 uframe.rot.q3 uframe.rot.q4	Yok

### NOT!



İş Parçası verilerini düzenlemek Program Verileri (Program Data) penceresinden de yapılabilir.

## İş parçası bildirimini (declaration) düzenlenmesi

### İş parçası açıklamasını görüntüleme

#### Eylem

1. **ABB** menüsünde **Jogging**'e dokununuz.
2. Uygun iş parçalarının listesini görmek için **Work object**'e dokununuz.
3. Düzenlemek istediğiniz iş parçasını seçiniz ve **Edit**'e dokununuz.
4. **Change Declaration**'a dokununuz.
5. İş parçası bildirimi görülecektir.
6. Uç elemanı bildirimini *İş Parçası (Work object) Yaratma* bölümünde anlatıldığı gibi düzenleyin.

### NOT!



Eğer bir iş parçasının ismini değiştirirseniz ve ismini değiştirdiğiniz iş parçası bir programda kullanılıyorsa o programda da isim değişikliği yapmanız gerekmektedir.

### Bir İş Parçasını Silme

#### Eylem

1. **ABB** menüsünde **Jogging**'e dokununuz.
2. Uygun iş parçalarının listesini görmek için **Work object**'e dokununuz.

3. Silmek istediğiniz iş parçasını seçiniz ve **Edit**'e dokununuz.
4. Seçilen uç elemanını silmek için **Delete**'e dokununuz.  
Bir onay ileti kutusu açılacaktır.
5. Silmek istiyorsanız ileti kutusunda **Yes**'e, istemiyorsanız **No**'ya basınız.

## Yük (Payload)

### Yük Yaratılınca Ne olur?

`loaddata` veri tipinde bir değişken yaratırsınız. Değişken adı da yükün adı olur.

### Yeni Bir Yük Ekleme ve Veri Bildirimini Ayarlama

Yük koordinat sistemi, yönelim de dahil olmak üzere dünya koordinat sisteminin konumuna ayarlanır.

### Eylem

1. **ABB** menüsünde, **Jogging**'e dokununuz.
2. Uygun yüklerin listesini görmek için **Payload**'a dokununuz.
3. Yeni bir uç eleman yaratmak için **New...**'e dokununuz.
4. **OK**'e dokununuz.

Konu hakkındaki detaylı bilgiler ileleyen aşamalarda verilecektir.

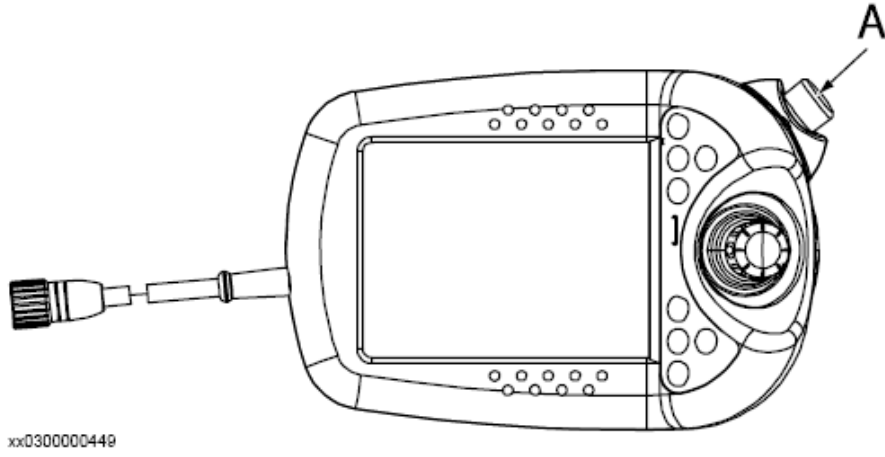
## EK1 - ACİL GÜVENLİK BİLGİSİ

### 1.1. Sistemi Durdurmak

Eğer ;

- Robot çalışırken robot çalışma bölgesinde personel varsa
- Robot bir personele veya bir mekanik teçizata zarar verecek durumdaysa hemen acil durdurma düğmesine basılmalıdır.

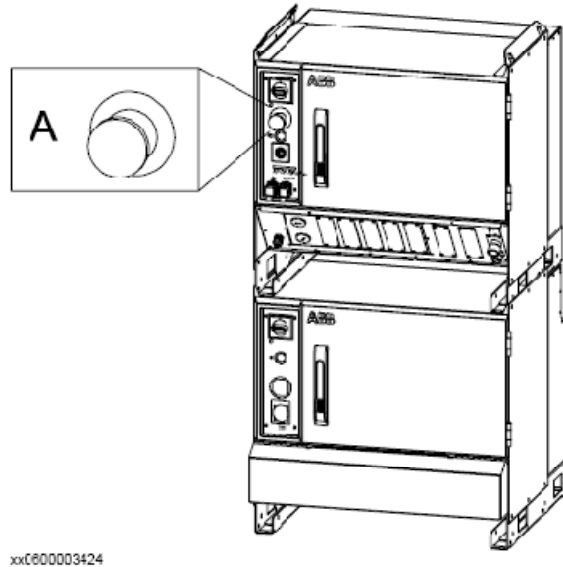
### FlexPendant Acil Durdurma Düğmesi



Şekil 14

A – FlexPendant Acil Durdurma Düğmesi

### Kontrol Ünitesi Acil Durdurma Düğmesi



Şekil 15

A - Kontrol Ünitesi Acil Durdurma Düğmesi

## **Diğer Acil Durdurma Birimleri**

Sistemi kuranlar değişik bölgelere acil durdurma düğmeleri yerleştirmiş olabilirler. Bunların nerelere yerleştirildiğini öğrenmek için sistem dokümanına bakın.

### **1.2. Yangın Söndürme**

#### **NOT!**



Sistemde (FlexPendant'ta ya da kontrol ünitesinde) oluşan herhangi bir yangın durumunda yangını söndürmek için KARBONDİOKSİT (CO<sub>2</sub>) söndürücü kullanınız!

### **1.3 Kontrol Ünitesine Giden Tüm Gücü Kapatma**

Kontrol ünitesinin her modülünde birer ana güç anahtarı bulunmaktadır. Kontrol ünitesine HİÇBİR gücün bağlı olmadığından emin olmak için tüm modüllerin ana anahtarının kapalı olması gerekmektedir.

#### **NOT!**

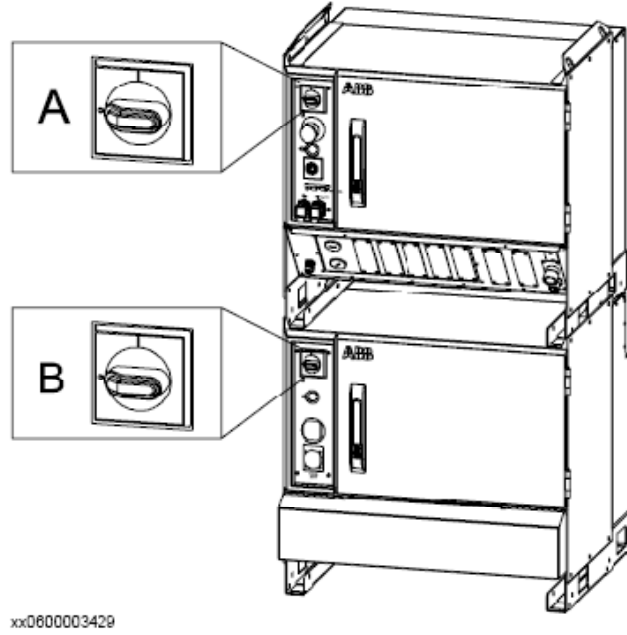


Sisteminizin ek teçizatının da gücünün kesilmesi gerekebilir. Bu anahtarların nerede bulunduğunu bulmak için sistem dokümanına bakınız.

### **Kontrol Ünitesine Giden Gücü Kapatma**

#### **Eylem**

1. Kontrol Modülündeki ana güç anahtarını kapatın
2. Sisteme bağlı bütün Sürücü Modüllerindeki ve diğer modüllerdeki (nokta kaynağı kabini gibi) ana güç anahtarını kapatın.



Şekil 3

A - Ana Güç Anahtarı, Kontrol Modülü  
B - Ana Güç Anahtarı, Sürücü Modülü

#### 1.4 Robot Kolu Tarafından Sıkıştırılmış Bir İnsanı Kurtarma

Eğer birisi robot kolu tarafından sıkıştırılmışsa, bu kişiyi kurtarmaya yönelik her türlü girişimin daha kötü sonuçlar doğurmayacağından emin olmanız gerekmektedir.

Robot frenlerini salmak robotu el ile hareket ettirmeyi mümkün kılmaktadır. Ancak sadece küçük robotların kolları insan gücüyle kaldırılabilir ağırlıkta olmaktadır. Daha büyük robotlar için vinç ya da benzeri aletler gerekmektedir. Frenleri salmadan önce elinizde gerekli aletlerin elinizde olduğundan emin olunuz!

#### UYARI!



Frenleri salmadan önce robot kolu yükünün, sıkıştırılmış kişinin üstündeki yükü arttırmayacağından emin olunuz, daha ciddi yaralanmalara yol açabilirsiniz!

#### Sıkıştırılmış Bir Kişiyi Kurtarma

Aşağıdaki prosedür, robot kolu tarafından sıkıştırılmış bir kişinin nasıl kurtarılacağını açıklar.

	Eylem	Bilgi
1.	Herhangi bir acil durdurma düğmesine basınız.	
2.	Sıkıştırılan kişinin tasarladığınız hareketle daha ciddi yaralanmayacağından emin olunuz.	

3.	Kişiyi kurtarmak için robotu hareket ettiriniz.	Robot kitleme frenlerini salmak için <i>Robot Kolunun Acil Salınması</i> bölümüne bakınız.
4.	Kurtarılan kişiye ilk yardımda bulunun.	
5.	Başkalarının yaralanma riskiyle karşılaşmaması için robot çalışma alanının temizlendiğinden emin olunuz.	

### 1.5 Robot Kolunun Acil Salınması

Acil bir durumda, robot eksenlerinden herhangi biri fren salma düğmelerine (brake release button) basılarak salınabilir.

Küçük robotlarda robot kolu elle hareket ettirilebilir ancak büyük robotlarda vinç veya benzeri bir alet gerekmektedir.

#### Daha Ciddi Yaralanma

Sıkıştırılan kişinin tasarlanan hareketle daha ciddi bir şekilde yaralanmayacağından emin olunuz. Daha ciddi yaralanmalara yol açabilirsiniz!

#### Robot Kolunu Salma

Aşağıdaki prosedür robot kolunun nasıl salınacağını açıklar.


Fren boşaltma ünitesi aşağıda resmedilmiştir. Ancak robot modeline bağlı olarak değişik bir yere yerleştirilmiş olabilir.

#### NOT!



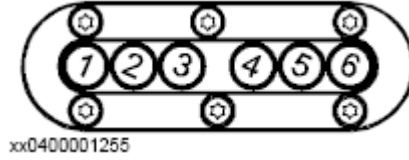
Eğer robot kontrol ünitesine bağlı değilse ya da kontrol ünitesinin gücü kapatılmışsa, robot tabanındaki bağlayıcıda (connector) harici bir güç kaynağı kullanmanız gerekir. *Fren Boşaltma Ünitesi İçin Harici Güç Kaynağı* bölümüne bakınız.

	Eylem	Bilgi
1.	Dahili fren boşaltma ünitesi eksen frenlerini kontrol etmek üzere 6 düğmeden oluşur. Düğmeler ilgili eksen takımına göre numaralandırılmıştır.	
2.		

	 <p><b>TEHLİKE!</b> Tutucu frenleri salarken, robot eksenleri çok hızlı hareket edebilir ve bazen de beklenmedik şekillerde hareket edebilir! Hiçbir personelin robot kolunun yanında olmadığından emin olunuz!</p>	<p>Frenlerin salınmasından önce robot kolunun bir vinçle ya da benzeri bir aletle kontrol altında tutulduğundan emin olunuz!</p>
<p>3.</p>	<p>Dahili fren boşaltma panelinde seçilen robot eksenine karşılık düşen düğmeyi basılı tutarak freni salınız. Elinizi düğmeden çeker çekmez frenler yeniden devreye girecektir.</p>	

### Fren Boşaltma Ünitesi

Aşağıdaki şekil robot gövdesinde ya da tabanında bulunan fren boşaltma ünitesi resmedilmiştir. Fren boşaltma ünitesi bir tabaka veya kenetleyiciyle korunmaktadır.



Şekil 4 Fren Boşaltma Ünitesi

### 1.6 Acil Durumun Telafi Edilmesi

Acil durumun telafi edilmesi basit fakat önemli bir prosedürdür. Bu prosedür, tehlikeli bir durum sürdürülürken durdurulan robot sistemi üretiminin aynı tehlikeli duruma dönmemesini garantiler.

#### Acil Durum Anahtarlarını Resetleme

Tüm basmalı-düğme stilli acil durum aygıtlarında, aygıtın acil durumdan çıkması için gerekli bir çeşit anahtarlama mekanizması bulunmaktadır. Lütfen bu özelliğin nasıl ayarlandığını görmek için fabrika ya da hücre dokümanlarını inceleyin.

#### Acil Durumun Telafi Edilmesi

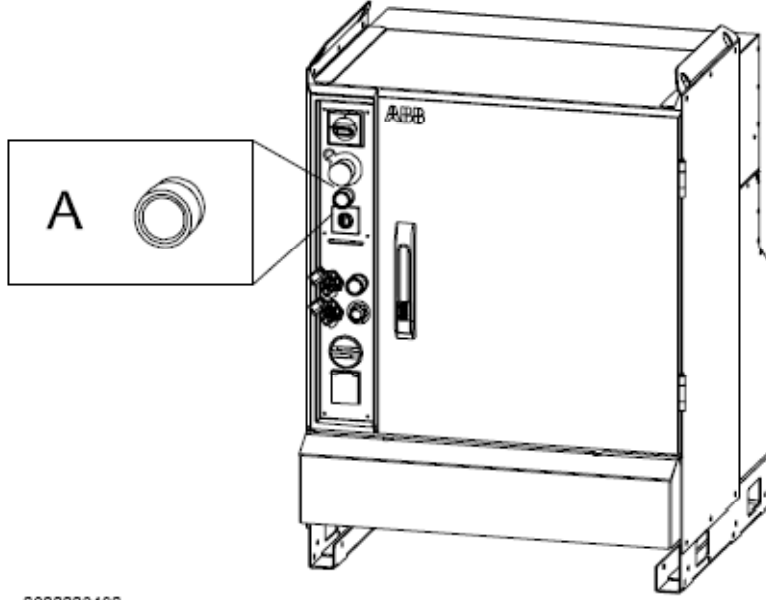
##### Eylem

1. Acil durum anında tespit edilen tehlikeli durumun giderildiğinden emin olun.
2. Acil Durum sinyali veren aygıtı ya da aygıtları tespit edin ve resetleyin.
3. Acil durumun telafi edilmesi için **Motors On** düğmesine basın.

#### Motors On Düğmesi



Motors On düğmesi kontrol ünitesinin üstündedir. Çift kabinli sistemde Motors On düğmesi kontrol ünitesinin üstündedir. Eğer robot sisteminiz farklı bir kontrol kabini tipini kullanıyorsa Motors On düğmesi aşağıdaki şekilden farklı gözükebilir.



xx0600003430

Şekil 116 Motors On Düğmesi

## EK2 - Robot Functions

### robtargert - Konum verisi

robtargert (robot target - robotun hedefi) robotun konumunu ve harici eksenleri tanımlamak amacıyla kullanılır.

Konum verileri, robot ve harici eksenleri hareket ettirecek komutlarda, konumu tanımlamak için kullanılır. Robot aynı konuma birkaç farklı yolla ulaşabilirken, eksen konfigürasyonu da belirlenir. Bu, eksen değerlerini -eğer herhangi bir şekilde belirsiz ise- tanımlar. Örnek olarak aşağıdaki durumlar verilebilir:

- Robot ileriye veya geriye doğru bir konumda ise,
- 4. eksen aşağıyı ya da yukarıyı gösteriyorsa,
- 6. eksen negatif ya da pozitif dönüş gösteriyorsa.

### UYARI!

Konum, iş parçasının koordinat sistemi baz alınarak, tüm program yerdeğiřtirmelerini kapsayacak şekilde tanımlanmıştır. Eğer konum, komut içindeki iş parçasından farklı bir iş parçası ile programlanırsa, robot beklenen istikamette hareket etmeyebilir. Hareket komutlarını programlarken kullanılan iş parçası ile aynı iş parçasının kullanıldığından emin olunmalıdır. Hatalı kullanım bir kişinin yaralanmasına sebep olabileceği gibi, robotun ya da başka bir aletin hasar almasına da sebep olabilir.

### Bileşenler

#### trans

*translation - çeviri, aktarım.*

Veri tipi: pos

Konum, mevcut parçanın koordinat sistemine göre program yerdeğiřtirmelerini de kapsayacak şekilde belirlenmiştir. Eğer, hiçbir iş parçası tanımlanmamışsa, kullanılan dünya koordinat sistemidir.

#### rot

*rotation - dönme, dönüş.*

Veri tipi: orient

Uç elemanın quaternion (q1, q2, q3, q4) formunda verilen yönelimidir (orientation). Yönelim, mevcut parçanın koordinat sistemine göre program yerdeğiřtirmelerini de kapsayacak şekilde belirlenmiştir. Eğer, hiçbir iş parçası tanımlanmamışsa, kullanılan dünya koordinat sistemidir.

#### robconf

*robot configuration - robot yapılandırması*

Veri tipi: confdata

Robotun eksen yapılandırmasını (*cf1*, *cf4*, *cf6* ve *cfx*) ifade etmektedir. Bu, 1. eksen, 4. eksen ve 6. eksen çeyrek dönüşü ile tanımlanmaktadır. İlk pozitif çeyrek dönüş (0°dan 90°a) 0 olarak tanımlanır. *cfx* bileşeninin anlamı robot tipine bağlı olarak değişmektedir. Ayrıntılı bilgi *confdata* veri tipini inceleyiniz.

### **extax**

*external axes – harici eksenler*

Veri tipi: *extjoint*

Harici eksenlerin konumudur.

Konum, her bir eksen için (*eax\_a*, *eax\_b* ... *eax\_f*) aşağıdaki gibi tanımlanır:

- Dönen eksenler için konum, kalibrasyon konumundan olan dönüşün derece biriminden ifadesi olarak tanımlanmıştır.
- Doğrusal eksenler için konum, kalibrasyon konumundan olan uzaklığın mm biriminden ifadesi olarak tanımlanmıştır.

Harici eksenler (*eax\_a* ...) mantıksal eksenlerdir. Mantıksal eksenler ile fiziksel eksen numaralarının nasıl bir ilişkide olduğu daha sonra açıklanacaktır.

9E9 değeri bağlantısı olmayan eksenler için tanımlanmıştır. Eğer konum verilerinde tanımlanmış eksenler program yürütüldüğünde aslında bağlı olan eksenlerden farklılık gösterirse aşağıdakiler uygulanır:

- Konum, konum verisinde tanımlanmamışsa (9E9 değeri) ve eksen bağlı olup (*connected*) etkinleştirilmemişse (*not activated*), değer yoksayılacaktır. Eğer eksen etkinleştirilmişse, hata ile sonuçlanacaktır.
- Konum, konum verilerinde tanımlanmasına rağmen eksen bağlı değilse değer yoksayılacaktır.

### **Örnek 1**

```
CONST robtarget p15 := [ [600, 500, 225.3], [1, 0, 0, 0], [1, 1, 0, 0], [ 11, 12.3, 9E9, 9E9, 9E9, 9E9] ];
```

p15 konumu şu şekilde tanımlanmıştır:

- Robotun, parça (*object*) koordinat sistemindeki konumu: x=600, y=500 ve z=225.3 mm'dir.
- Uç elemanın yönelimi, parça koordinat sistemiyle aynı doğrultudadır.
- Robotun eksen yapılandırması: 1. ve 4. eksen 90-180° konumunda, 6. eksen 0-90° konumundadır.
- Harici mantıksal eksenlerin konumları, a ve b, eksen tipine bağlı olarak derece ya da mm birimiyle ifade edilir. c den f'ye olan eksenler tanımsızdır.

### **Örnek 2:**

```
VAR robtarget p20;  
...
```

```
p20 := CRobT(\Tool:=tool\wobj:=wobj0);  
p20 := Offs(p20,10,0,0);
```

p20 konumu, CRobT fonksiyonu çağrılarak robotun şu anki konumu ile aynı konuma ayarlanır. Konum, daha sonra x-doğrultusunda 10 mm hareket ettirilir.

#### Yapı:

```
< dataobject of robtarget >  
  < trans of pos >  
    < x of num >  
    < y of num >  
    < z of num >  
  < rot of orient >  
    < q1 of num >  
    < q2 of num >  
    < q3 of num >  
    < q4 of num >  
  < robconf of confdata >  
    < cf1 of num >  
    < cf4 of num >  
    < cf6 of num >  
    < cfx of num >  
  < extax of extjoint >  
    < eax_a of num >  
    < eax_b of num >  
    < eax_c of num >  
    < eax_d of num >  
    < eax_e of num >  
    < eax_f of num >
```

## tooldata - Tool data

tooldata uç elemanın (kaynak tabancası ya da tutucu -gripper- gibi) karakteristiğinin tanımlanmasında kullanılır.

Uç eleman verileri robot hareketlerini aşağıda belirtilen şekilde etkilemektedir:

- Uç elemanın merkez noktası (TCP – Tool Center Point) belirlenen güzergahta, belirlenen hıza uygun noktadır. Eğer uç elemanın açısız konumu (oryantasyonu) değiştirilirse ya da harici eksen koordinatları kullanılırsa, sadece bu nokta programlanan hızda arzu edilen güzergahı takip edecektir.
- Programlanan konumlar o anki TCP'yle ve tool koordinat sistemine göre olan oryantasyon ile ilişkilidir. Bu da eğer, örneğin, uç eleman hasar gördüğü için değiştirildiğinde, tool koordinat sisteminin tekrar tanımlanması kaydıyla eski programın hala kullanılabilir olacağı anlamına gelmektedir.

Tool verileri, robotu aşağıda sıralanan amaçlar doğrultusunda kontrol çubuğuyla hareket ettiren de kullanılmaktadır:

- Uç eleman yeniden yönlendirilirken TCP'nin hareket etmeyecek şekilde tanımlanması
- Uç eleman doğrultusunda hareket ettirilirken ya da döndürülürken, yardımcı olmak amacıyla tool koordinat sisteminin tanımlanması.

### Dikkat!

Uç eleman yükünün ve -kullanıldığında- robot yükünün her zaman gerçek değeriyle tanımlanması büyük önem arz eder. Yük verilerinin hatalı tanımlanması robotun mekanik yapısında aşırı yüklenmelere sebep olabilir.

Yük verisinin hatalı olarak belirlenmesi genellikle aşağıdaki sonuçlara yol açar:

- Eğer yük için belirlenen değer gerçek değerinden büyükse:
  - Robot maksimum kapasitesinde kullanılamayacaktır.
  - Hedeflenen yoldan sapma riski oluşur.
- Eğer yük için belirlenen değer gerçek değerinden küçükse:
  - Mekanik yapıda aşırı yüklenme riski oluşur.
  - Hedeflenen yoldan sapma riski oluşur.

### Bileşenler

robhold

Robot tutuyor

Veri Tipi: bool

Robotun, uç elemanı tutup tutmadığını tanımlamaktadır.

- TRUE: Robot, uç elemanı tutmaktadır.
- FALSE: Robot, uç elemanı tutmamaktadır. Örneğin, sabit uç eleman - stationary tool.

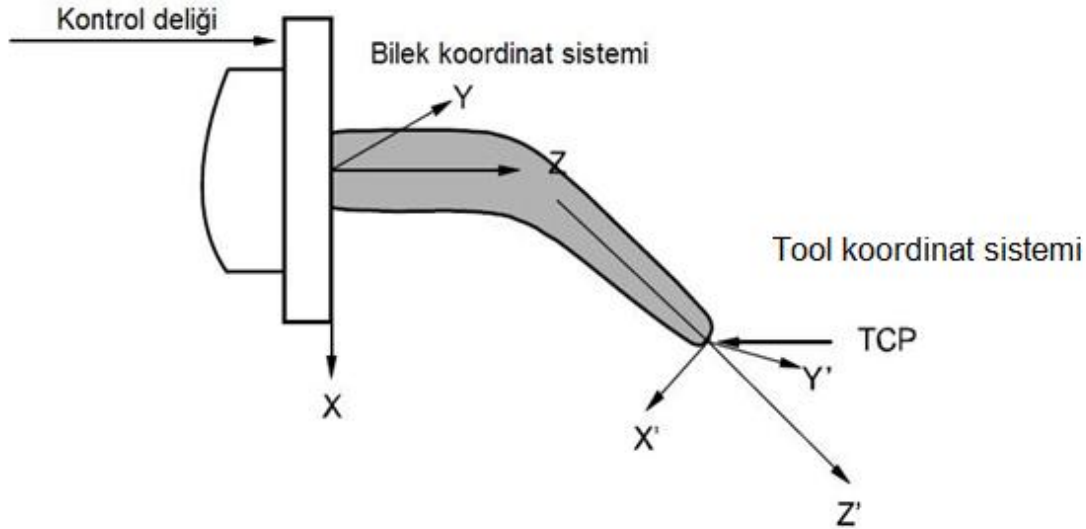
tframe

Veri tipi: pose

Tool koordinat sistemi, örneğin:

- TCP'nin milimetrik olarak verilen konumu (x,y ve z) bilek koordinat sisteminde tarif edilmiştir (aşağıdaki şekli inceleyiniz).
- Tool koordinat sisteminin yönelimi, bilek koordinat sisteminde quaternion (q1,q2, q3 ve q4) ile belirtilmiştir (aşağıdaki şekli inceleyiniz).

Eğer sabit uç eleman kullanılmışsa, tanımlama dünya koordinat sistemine (world coordinate system) göre yapılır. Uç eleman doğrultusu belirtilmemişse, tool koordinat sistemi ve bilek koordinat sistemi çakışacaktır.



tload

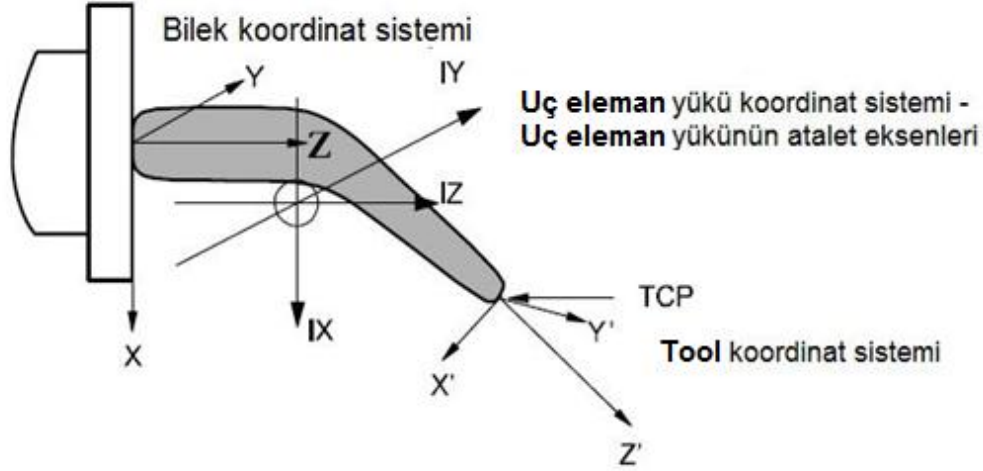
Uç elemanın yükü

Veri tipi: loaddata

Örnek:

- Uç elemanın kg biriminden ağırlığı.
- Uç eleman yükünün ağırlık merkezi (x, y ve z) milimetre birimiyle bilek koordinat sisteminde tarif edilmiştir.
- Uç eleman yükü, koordinat sisteminin yönelimi bilek koordinat sisteminde, atalet eksenlerini tanımlayarak tarif edilmiştir. Uç eleman yükü koordinat sisteminin yönelimi mutlak suretle bilek koordinat sisteminin yönelimiyle çakışmalıdır. Bu, daima 1,0,0,0 olarak ayarlanmalıdır.

- Uç elemanın, Uç eleman yükü koordinat eksenlerinin etrafındaki kütle merkezine bağlı atalet momentleri ( $\text{kgm}^2$  biriminden). Eğer bütün atalet bileşenleri 0  $\text{kgm}^2$  olarak tanımlandıysa, uç eleman noktasal kütle olarak muamele görür.



**NOT:** Sadece uç elemanın yükü belirlenecektir. Tutucu tarafından karşılanan yük GripLoad komutu dahilinde bağlanır ya da bağlantısı kesilir.

#### Örnek 1:

```
PERS tooldata gripper := [ TRUE, [[97.4, 0, 223.1], [0.924, 0,
0.383 ,0]], [5, [23, 0, 75], [1, 0, 0, 0], 0, 0, 0]];
```

Uç eleman, aşağıdaki değerler kullanılarak tanımlanmıştır:

- Robot, uç elemanı tutuyor.
- TCP, 6. eksenin 223.1 mm ve bilek koordinat sisteminin X-ekseninden 97.4 mm uzaktadır.
- Uç elemanın X ve Z doğrultuları, bilek koordinat sistemine göre 45° döndürülmüştür.
- Uç elemanın kütlesi 5 kg'dır.
- Ağırlık merkezi, 6. eksenin doğrusal olarak 75 mm uzaklıkta, bilek koordinat sisteminin X-ekseninden 23 mm uzaklıkta bir noktadadır.
- Yük, noktasal kütle olarak kabul edilebilir.

```
gripper.tframe.trans.z := 225.2;
```

Uç elemanın, gripper'in TCP'si z-doğrultusunda 225.2 mm'ye ayarlanmıştır.

#### Sınırlamalar

Uç eleman verileri persistent değişken (PERS) olarak tanımlanmalı ve routine içinde tanımlanmalıdır. Mevcut değerler program kaydedildiğinde kaydedilir, ve program yüklendiğinde tekrar erişilebilir.

## Ön-tanımlı veri

tool10 uç elemanı orijini bağlantı flanşında olan bilek koordinat sistemini tanımlar. Tool10 her zaman programdan ulaşılabilir, fakat hiç bir zaman değiştirilemez.

## Yapı

```
< dataobject of tooldata >
  < robhold of bool >
  < tframe of pose >
    < trans of pos >
      < x of num >
      < y of num >
      < z of num >
    < rot of orient >
      < q1 of num >
      < q2 of num >
      < q3 of num >
      < q4 of num >
  < tload of loaddata >
    < mass of num >
    < cog of pos >
      < x of num >
      < y of num >
      < z of num >
  < aom of orient >
    < q1 of num >
    < q2 of num >
    < q3 of num >
    < q4 of num >
  < ix of num >
  < iy of num >
  < iz of num >
```



## orient - Orientation (açısal konum, yönelim)

`orient`, yönlendirme (uçtaki elemanın yönlendirilmesi gibi) ve döndürme (koordinat sisteminin döndürülmesi gibi) amaçlı kullanılmaktadır. Yönlendirme;  $q_1$ ,  $q_2$ ,  $q_3$  ve  $q_4$  elemanlarından oluşan quaternion formunda tanımlanmaktadır. Bunların nasıl hesaplanacağı ile ilgili bilgiler aşağıda yer almaktadır:

### Bileşenler

`orient` data tipi aşağıdaki bileşenlerden oluşmaktadır:

$q_1$  Veri tipi: num  
Quaternion 1.

$q_2$  Veri tipi: num  
Quaternion 2.

$q_3$  Veri tipi: num  
Quaternion 3.

$q_4$  Veri tipi: num  
Quaternion 4.

### Örnek 1

`orient` data tipi ile ilgili temel örnekler aşağıda gösterilmiştir.

```
VAR orient orient1;  
.   
orient1 := [1, 0, 0, 0];
```

`orient1` yönlendirmesi  $q_1=1$ ,  $q_2=q_3=q_4=0$  olarak atanmıştır; bu durumda döndürme yoktur.

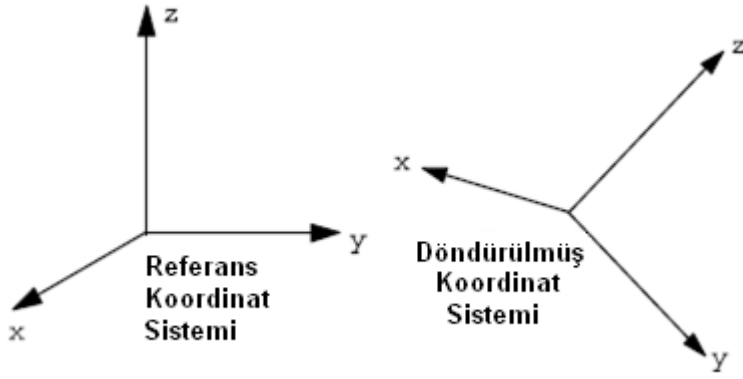
### Kısıtlamalar

Yönelim normalize edilebilmelidir : örneğin, karelerinin toplamı 1'e eşit olmalıdır.

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$$

### Quaternion nedir?

Koordinat sisteminin yönelimi (aynen uç elemanının gibi), koordinat sisteminin eksenlerinin doğrultusunu referans bir sisteme göre tanımlayan bir döndürülmüş matris (rotational matrix) ile tanımlanabilir (aşağıdaki şekli inceleyiniz).



Döndürülmüş koordinat sistemi eksenleri (x, y, z) birer vektör olup referans koordinat sisteminde aşağıdaki gibi ifade edilebilir:

$$\mathbf{x} = (x_1, x_2, x_3)$$

$$\mathbf{y} = (y_1, y_2, y_3)$$

$$\mathbf{z} = (z_1, z_2, z_3)$$

Bu, referans koordinat sisteminde x-vektörünün x-bileşeninin  $x_1$ , y-bileşeninin  $x_2$ , vb. olacağı anlamına gelir. Bu üç vektör birlikte her bir vektörün bir kolunu oluşturduğu bir matrise yerleştirilebilir (döndürülmüş matris):

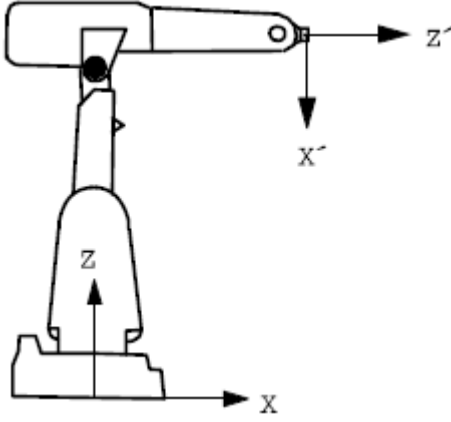
$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}$$

Quaternion, döndürülmüş matrisi tanımlamanın kısa yoludur; quaternionlar döndürülmüş matrisin elemanları baz alınarak hesaplanmaktadır:

$q_1 = \sqrt{\frac{x_1 + y_2 + z_3 + 1}{2}}$	
$q_2 = \sqrt{\frac{x_1 - y_2 - z_3 + 1}{2}}$	sign $q_2 = \text{sign}(y_3 - z_2)$
$q_3 = \sqrt{\frac{y_2 - x_1 - z_3 + 1}{2}}$	sign $q_3 = \text{sign}(z_1 - x_3)$
$q_4 = \sqrt{\frac{z_3 - x_1 - y_2 + 1}{2}}$	sign $q_4 = \text{sign}(x_2 - y_1)$

**Örnek 1** Bir uç elemanı, Z' eksenini tam karşıya gösterecek şekilde (temel koordinat sisteminin X eksenini ile aynı doğrultuda) yönlendirilmiştir. Uç elemanının Y' eksenini temel koordinat sisteminin Y-ekseniyle çakışmaktadır (aşağıdaki şekli inceleyiniz). Uç elemanın yönelimi konum verilerinde (robtarjet) nasıl tanımlanır?

Robotun ucundaki elemanın programlanan konumdaki yönelimi normalde kullanılan iş objesinin koordinat sistemiyle alakalıdır. Bu örnekte, hiçbir iş objesi kullanılmamıştır ve temel koordinat sistemi dünya koordinat sistemine denktir. Böylece, yönelim temel koordinat sistemi ile alakalı olacaktır.



Eksenler daha sonra aşağıdaki şekilde birbiriyle ilişkilenecektir:

$$x' = -z = (0, 0, -1)$$

$$y' = y = (0, 1, 0)$$

$$z' = x = (1, 0, 0)$$

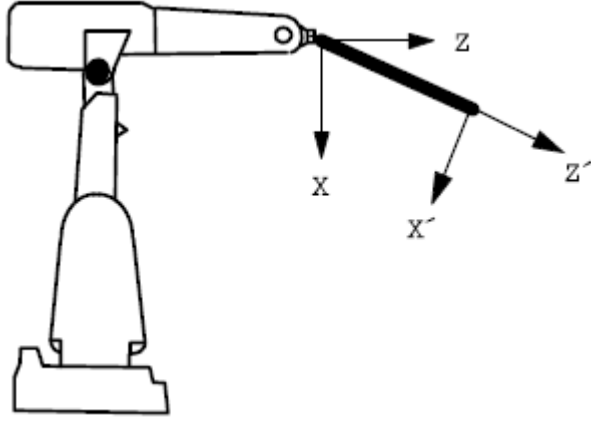
Bu da aşağıdaki döndürülmüş matrise denk gelmektedir:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Döndürülmüş matris uygun bir quaternion sağlar:

$q_1 = \sqrt{\frac{0+1+0+1}{2}} = \frac{\sqrt{2}}{2} = 0.707$	
$q_2 = \sqrt{\frac{0-1-0+1}{2}} = 0$	
$q_3 = \sqrt{\frac{1-0-0+1}{2}} = \frac{\sqrt{2}}{2} = 0.707$	sign q3 = sign (1+1) = +
$q_4 = \sqrt{\frac{0-0-1+1}{2}} = 0$	

**Örnek 2** Uç elemanın doğrultusu, X' ve Z' ekseninden bilek koordinat sistemine bağlı olarak 30° döndürülmüştür (aşağıdaki şekli inceleyiniz). Uç elemanın (tool) yönelimi, tool verilerinde nasıl tanımlanır?



Eksenler daha sonra aşağıdaki şekilde birbiriyle ilişkilenecektir:

$$x' = (\cos 30^\circ, 0, -\sin 30^\circ)$$

$$y' = (0, 1, 0)$$

$$z' = (\sin 30^\circ, 0, \cos 30^\circ)$$

Bu da aşağıdaki döndürülmüş matrise denk gelmektedir:

$$\begin{bmatrix} \cos 30^\circ & 0 & \sin 30^\circ \\ 0 & 1 & 0 \\ \sin 30^\circ & 0 & \cos 30^\circ \end{bmatrix}$$

Döndürülmüş matris uygun quaternion'u sağlar:

$q_1 = \sqrt{\frac{\cos 30^\circ + 1 + \cos 30^\circ + 1}{2}} = 0.965926$	
$q_2 = \sqrt{\frac{\cos 30^\circ - 1 - \cos 30^\circ + 1}{2}} = 0$	
$q_3 = \sqrt{\frac{1 - \cos 30^\circ - \cos 30^\circ + 1}{2}} = 0.258819$	sign q3 = sign (sin30°+sin30°) = +
$q_4 = \sqrt{\frac{\cos 30^\circ - \cos 30^\circ - 1 + 1}{2}} = 0$	

### Bileşenler

< dataobject of orient >

< q1 of num >

< q2 of num >

< q3 of num >

< q4 of num >