

EGE ÜNİVERSİTESİ

EGE MYO

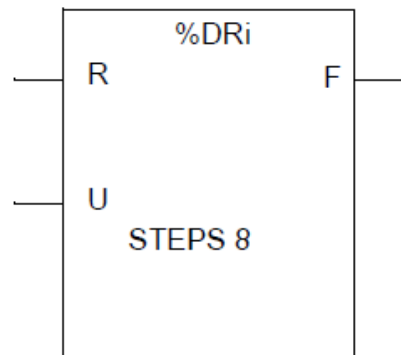
MEKATRONİK PROGRAMI

PROGRAMLANABİLİR DENETLEYİCİLER

SIRALI İŞLEMLER İÇİN DRUM CONTROLLER
FONKSİYON BLOĞU

Tambur Denetleyici fonksiyon bloğu (%DR)

- **Açıklama** Tambur denetleyici, dış olaylara göre adım değiştiren bir elektromekanik tambur denetleyiciye benzer bir prensiple çalışır. Her bir adımda kam milinin yüksek noktası, kontrol sistemi tarafından icra edilen bir buyruk verir. Tambur denetleyici için bu yüksek noktalar herbir adım için 1 durumu ile sembolize edilir ve kontrol bitleri olarak bilinen çıkış bitleri %Qi.j'ye veya dahili bitler %Mi'ye atanır.
- Aşağıdaki resimde, bir tambur denetleyici fonksiyon bloğu gösterilmiştir.



- **Parametreler** Tambur denetleyici fonksiyon bloğu, aşağıdaki parametrelere sahiptir:

| Parametre | Etiket | Değer |
|-----------------------------------|-----------|--|
| Numara | %DRi | 0-3 Kompakt Kontrolör; 0-7 Modüler Kontrolör |
| Mevcut adım numarası | %DRi.S | 0-%DRi.S-7. Okunabilen ve yazılabilen word. Yazılan değer, bir onluk (decimal) anlak değer almak zorundadır. Yazıldığında, fonksiyon bloğunun bir sonraki icrasında etkisini gösterir. |
| Adım sayısı | | 1 ila 8 (varsayılan) |
| Adım 0'a dön girişi (veya komutu) | R (Reset) | 1 durumunda, tambur denetleyiciyi adım 0'a set eder. |
| İlerle girişi (veya komutu) | U (Up) | Yükselen kenar tambur denetleyicinin bir adım ilerlemesine neden olur ve kontrol bitlerini günceller. |
| Çıkış | F (Full) | Mevcut adımın son tanımlanan adıma eşit olduğunu belirtir. İlişkilendirilmiş %DRi.F test edilebilir. (örneğin, %DRi.F=1, eğer %DRi.S= konfigüre edilen adım sayısı -1 ise) |
| Kontrol bitleri | | Adımla ilgili ve Konfigürasyon Editöründe tanımlanmış çıkışlar ve dahili bitler (16 kontrol biti) |

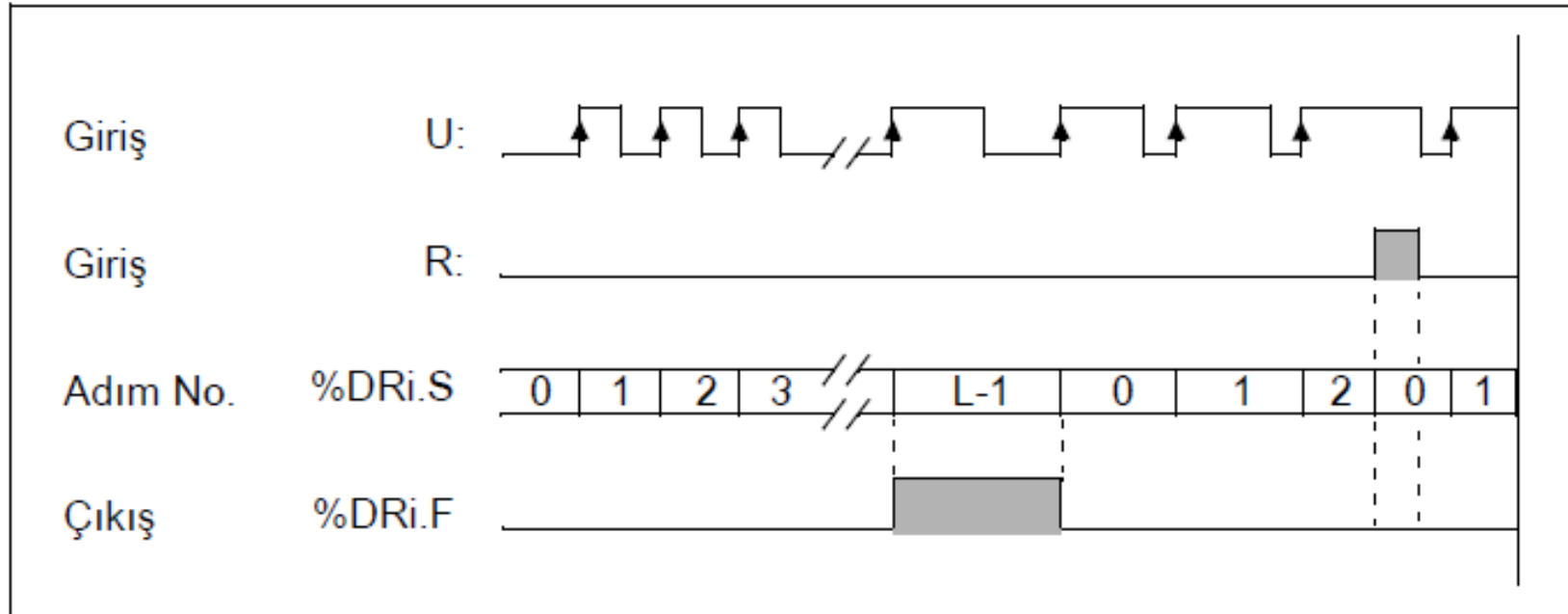
Tambur Denetleyici Fonksiyon Bloğunun Çalışması

- **Açıklama** Tambur denetleyici aşağıdakilerden oluşur:
 - Sekiz adımda (0-7) organize edilmiş bir sabit veri matrisi ve 0'dan F'ye kadar numaralanmış sütunlarda düzenlenmiş 16 veri biti (adımın durumu).
 - Ya %Q0.i veya %Q1.i çıkışlarına ya da %Mi dahili bitlerine karşılık gelen bir kontrol bitleri listesi. Mevcut adım sırasında, kontrol bitleri bu adım için tanımlanmış ikilik durumları alır.
- Aşağıdaki tablodaki örnek, tambur denetleyicinin ana karakteristiklerini özetler.

| Sütun | 0 | 1 | 2 | | D | E | F |
|-----------------|-------|-------|-------|--|-------|-------|-------|
| Kontrol Bitleri | %Q0.1 | %Q0.3 | %Q1.5 | | %Q0.6 | %Q0.5 | %Q1.0 |
| Adım 0 | 0 | 0 | 1 | | 1 | 1 | 0 |
| Adım 1 | 1 | 0 | 1 | | 1 | 0 | 0 |
| | | | | | | | |
| Adım 5 | 1 | 1 | 1 | | 0 | 0 | 0 |
| Adım 6 | 0 | 1 | 1 | | 0 | 1 | 0 |
| Adım 7 | 1 | 1 | 1 | | 1 | 0 | 0 |

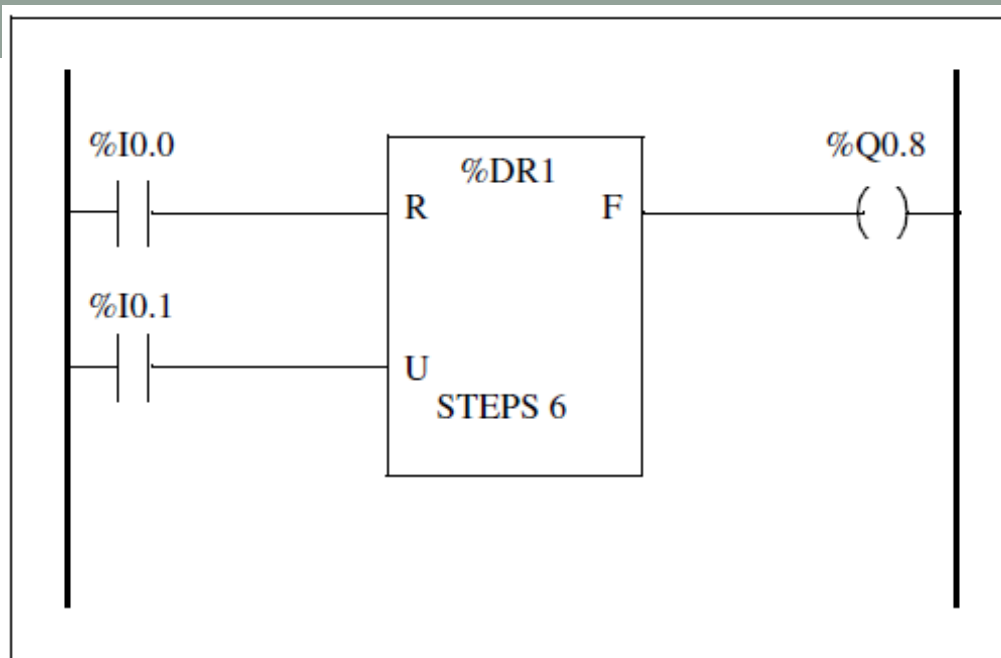
- **Çalışma** Yukarıdaki örnekte, adım 5, mevcut adımdır, kontrol bitleri %Q0.1, %Q0.3 ve %Q1.5, 1 durumuna; kontrol bitleri %Q0.6, %Q0.5 ve %Q1.0, 0 durumuna set edilmiştir.
- Mevcut adım numarası, U girişindeki her bir yükselen kenarda (veya U buyruğunun etkinleştirilmesiyle) bir arttırılır. Mevcut adım, program tarafından değiştirilebilir.

- **Zamanlama Diyagramı** Aşağıdaki diyagram, tambur denetleyici çalışmasının zamanlamasını gösterir.



Tambur Denetleyicileri Programlama ve Yapılandırma

- **Açıklama** Aşağıdaki, bir tambur denetleyiciyi programlamak ve yapılandırmak için bir örnektir.
- %I0.1 girişinin 1 yapıldığı her zaman, ilk 6 çıkış (%Q0.0 ila %Q0.5) ardarda aktive edilir. %I0.0 girişi, çıkışları 0'a reset eder.
- **Programlama Örneği**
- Aşağıdaki resim, bir tambur denetleyici fonksiyon bloğu ile birlikte dönüştürülebilir ve dönüştürülemez programlama örneklerini verir.



Ladder diyagramı

```

BLK    %DR1
LD     %I0.0
R      %DR1
LD     %I0.1
U      %DR1
OUT_BLK
LD     F
ST     %Q0.8
END_BLK

```

Dönüştürülebilir program

```

LD     %I0.0
R      %DR1
LD     %I0.1
U      %DR1
LD     %DR1.F
ST     %Q0.8

```

Dönüştürülemez program

- **Yapılandırma** Aşağıdaki bilgi, yapılandırma sırasında tanımlanır:

- Adımların sayısı: 6
- Her bir tambur denetleyici adımı için çıkış durumları (kontrol bitleri).

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Step 1 : | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Step 2 : | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Step 3 : | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Step 4 : | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Step 5 : | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Step 6 : | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- Kontrol bitlerinin atanması

| | | | |
|-----|-------|-----|-------|
| 1 : | %Q0.0 | 4 : | %Q0.1 |
| 2 : | %Q0.2 | 5 : | %Q0.3 |
| 3 : | %Q0.4 | 6 : | %Q0.5 |

Örnekler

SORU1: Sonraki sayfadaki şekilde görülen bir yıkama kazanı uygulaması yapılmak isteniyor.

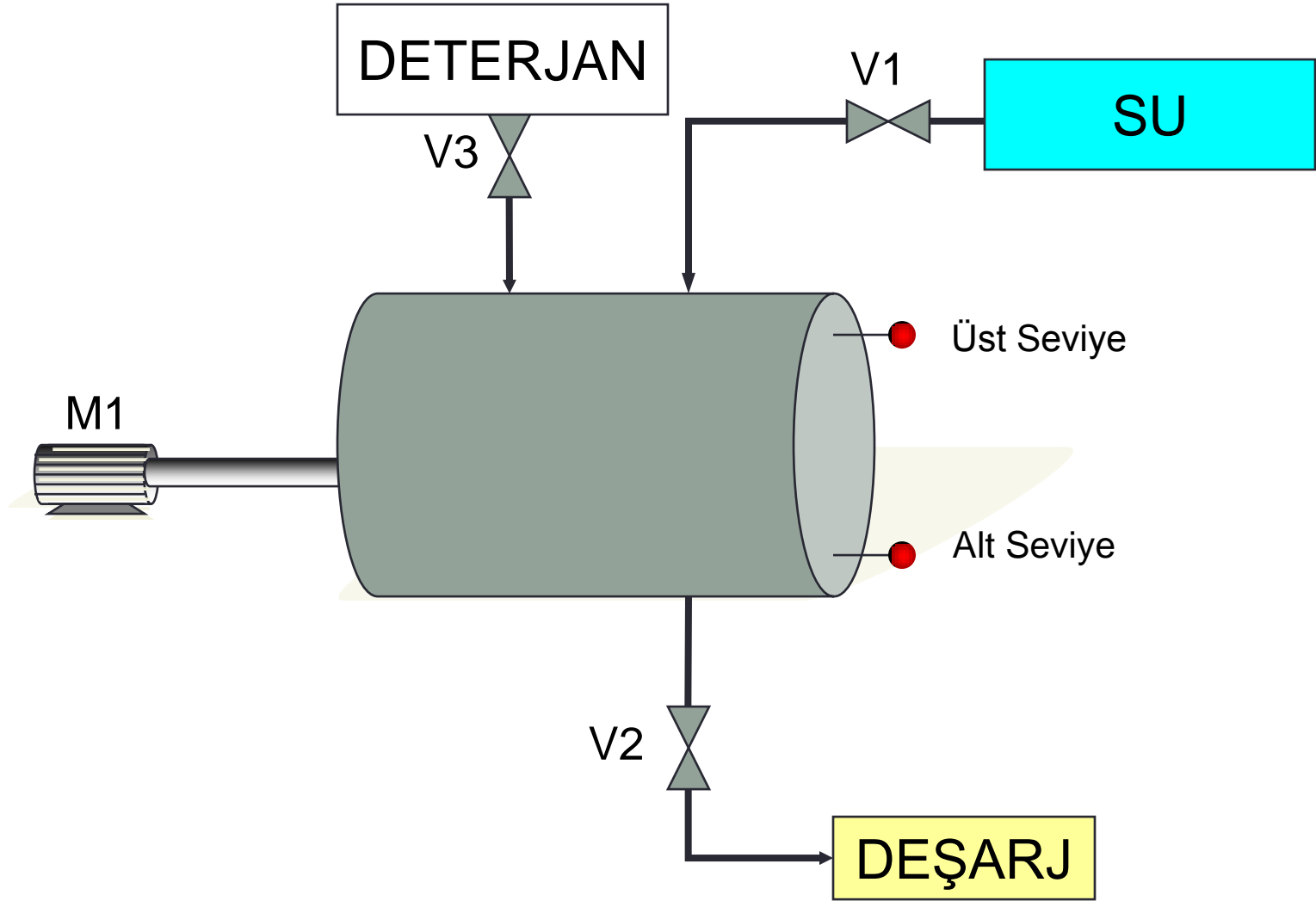
◆ Start butonuna basıldığında sırasıyla:

- 1) V1 vanası açılır.
- 2) Kazan dolunca V1 kapanır ve 5 sn.boyunca V3 açılarak kazana deterjan eklenir.
- 3) Deterjan eklendikten sonra M1 motoru 10 sn. devrede kalır.
- 4) 10sn. Sonunda M1 devreden çıkar, V2 açılarak kazanın suyu boşaltılır.
- 5) Kazan tamamen boşaldıktan sonra deterjan eklemeyen (V3 vanası açılmayacak)
aynı işlemler tekrarlanarak durulama işlemi gerçekleştirilir.
- 6)Tekrar start butonuna basılması için sistem hazır vaziyette bekler.

Aşağıdaki I/O giriş-çıkışları kullanın:

%I0.0 = Start Butonu, %I0.1 =Kazan Üst Seviye, %I0.2 =Kazan Alt Seviye
%Q0.0 = M1 otomatik çalış, %Q0.1= V1 otomatik aç, %Q0.2= V2 otomatik aç,
%Q0.3= V3 otomatik aç

Drum(tambur denetleyici) bloğunu kullanarak kontrol devresini oluşturun.



CEVAP1: Bu soruda öncelikle yapmamız gereken Drum fonksiyonunun(Tambur Denetleyicinin) konfigürasyonudur. Aşağıda %DR0 üzerinde yapılan ayarları görmekteyiz. Bu ayarlamadan sonra programın yapısını arka sayfada görebilirsiniz.

Drum Controllers

Drum: %DR

Symbol:

Number of Steps:

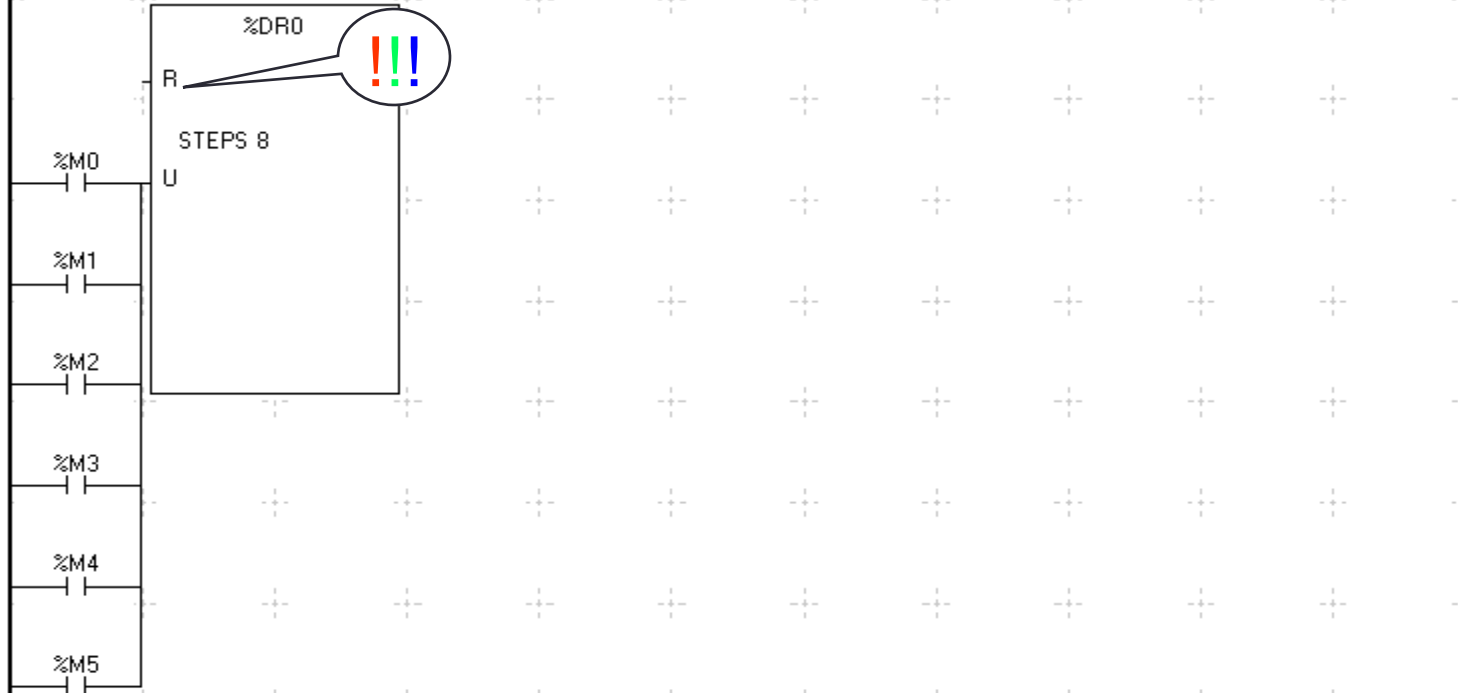
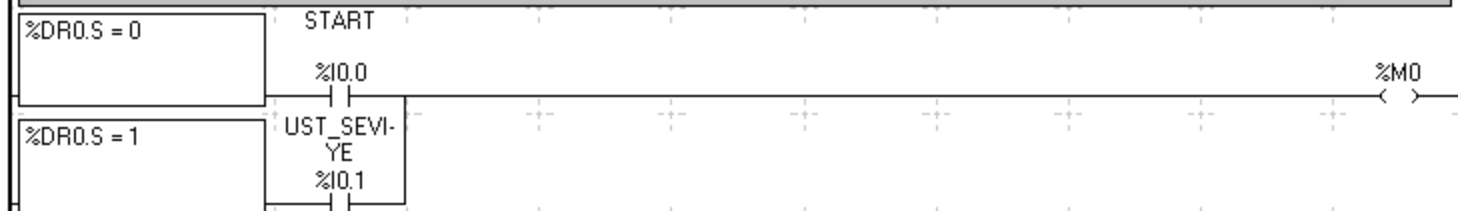
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Step 0: | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Step 1: | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Step 2: | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Step 3: | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Step 4: | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Step 5: | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Step 6: | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Step 7: | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Outputs:
%Q?.? or %M?

| | | | |
|--------|------------------------------------|---------|----------------------|
| Bit 0: | <input type="text" value="%Q0.0"/> | Bit 8: | <input type="text"/> |
| Bit 1: | <input type="text" value="%Q0.1"/> | Bit 9: | <input type="text"/> |
| Bit 2: | <input type="text" value="%Q0.2"/> | Bit 10: | <input type="text"/> |
| Bit 3: | <input type="text" value="%Q0.3"/> | Bit 11: | <input type="text"/> |
| Bit 4: | <input type="text"/> | Bit 12: | <input type="text"/> |
| Bit 5: | <input type="text"/> | Bit 13: | <input type="text"/> |
| Bit 6: | <input type="text"/> | Bit 14: | <input type="text"/> |
| Bit 7: | <input type="text"/> | Bit 15: | <input type="text"/> |

OK
Cancel
Previous
Next
Help

NOT: Drum fonksiyonu kullandığınızda R(Reset) girişini mutlaka kullanın. Bu harici şartlardan dolayı oluşabilecek kilitlemelerde gerekli olacaktır.

RUNG 0 TAMBUR DENETÇİNİN BASAMAK ATLAMA NOKTALARI

RUNG 1 START BUTONUNA BASILMASI VE TANKIN DOLMASI (V1)


RUNG 2 DETERJANIN KAZANA EKLENMESİ (V3)

%DR0.S = 2

%TM0

IN

Q

%M1

TYPE TON
TB 1 sec
ADJ Y
%TM0.P 5

RUNG 3 KARIŞTIRICININ DETERJANLI YIKAMA İÇİN ÇALIŞMASI (M1)

%DR0.S = 3

%TM1

IN

Q

%M2

TYPE TON
TB 1 sec
ADJ Y
%TM1.P 10

