



دانشکده علوم دریایی محمود آباد  
گروه مهندسی دریایی

# عملیاتی PLC500

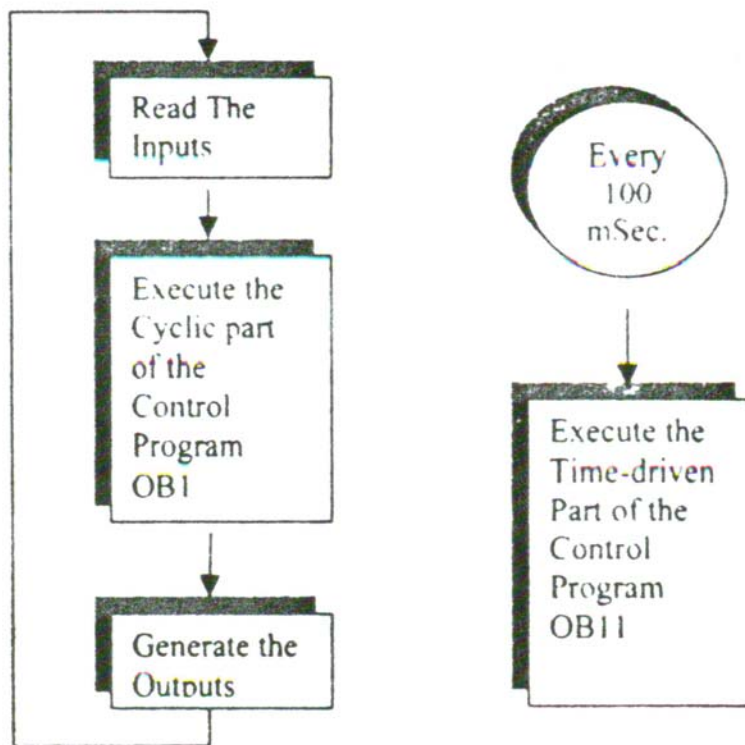
تهیه کننده : عباس خسروبیگی

اسفند ۸۴

## بسمه تعالی

### ساختار برنامه کنترلی

PLC برنامه‌های کنترلی را به دو روش جداگانه اجرا می‌نماید. این حالت عبارتند از سیکلی (Cyclic) و زمانی (Time Driven) در حالت سیکلی PLC ورودیها را می‌خواند، بخشی از برنامه را که توسط OB1 تعریف شده اجرا و خروجی مناسب را تولید می‌نماید. این عمل پی‌درپی تکرار می‌شود. در حالت زمانی، PLC هر 100ms یکبار بخش دیگری از برنامه را که توسط OB11 مشخص شده اجرا می‌کند.



A. Cyclic Mode

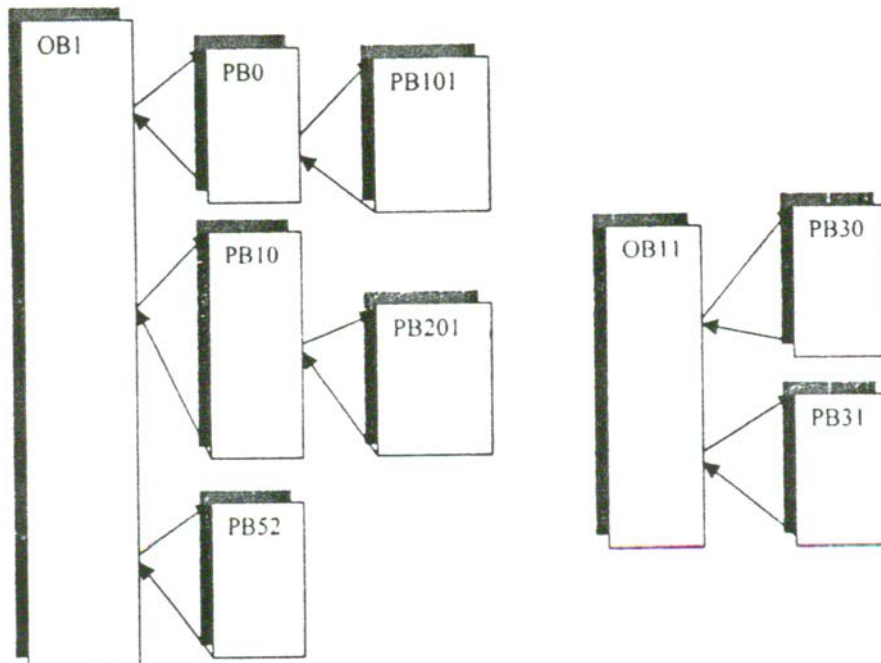
B. Time-driven Mode

برای راحتی در برنامه‌نویسی و دستیابی به حداکثر سرعت در اجرا توسط PLC، برنامه‌های کنترلی باید به بلوک‌های تقسیم‌گردند، PLC از دو نوع بلوک استفاده می‌کند، بلوک‌های سازماندهی (OB) و بلوک‌های برنامه (PB) که هر کدام از آنها توسط عددی که در دنباله آنها می‌آید مشخص می‌شوند. در حالیکه OB ها تنها به OB1 به OB11 محدود می‌شوند، PB ها را می‌توان آزادانه از PB0 تا PB255 عددگذاری کرد.

OB1: بصورت سیکنال و پی‌درپی توسط PLC فراخوانی و اجرا می‌شود.

OB11: هر 100 میلی ثانیه یکبار توسط PLC فراخوانی و اجرا می‌شود.

مطابق شکل زیر قسمتهای مختلف برنامه را می‌توان به PB های مختلفی تقسیم کرد و آنها را از OB های مناسب بصورت شرطی یا غیرشرطی فراخوانی نمود.



تذکر: برای دستیابی به حداکثر سرعت OB11 را تا اندازه ممکن کوچک نگه دارید.

### انواع حافظه در PLC

EEPROM برای Backup برنامه‌های RAM استفاده می‌شود. برنامه OS و مترجم در EPROM قرار دارد. بخشی از حافظه RAM توسط باتری پشتیبان تغذیه می‌شود.

ناحیه پایدار	%۱۰۰
ناحیه ناپایدار	%۵۰

### Retentive (پایدار) و Non – Retentive (ناپایدار)

این موضوع در مورد flagها – اطلاعات – تایمرها و کانترها صدق می‌کند.

### روشهای کنترل بیان مدارهای کنترلی

۱- روش ladder

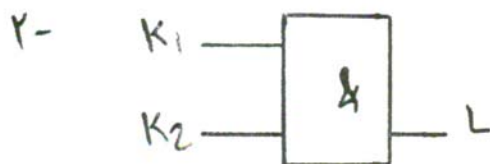
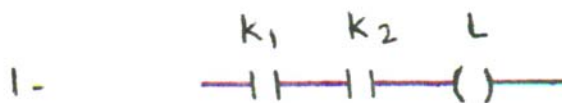
۲- روش CSF (Control System Flowchart)

۳- روش STL (Statement List)

مثال ۱: مدار مقابل را در نظر بگیرید



بیان این مدار به روشهای فوق :



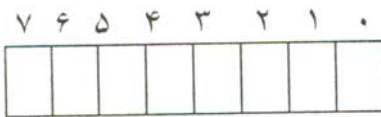
```

۲- LD I K1
    A I K2
    = Q L
    BE

```

### آدرس دهی ورودی‌ها و خروجی‌ها

I برای ورودی و Q برای خروجی  
m.n



m شماره بایت  
n شماره بیت

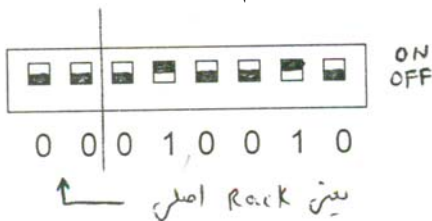
نوشتن برنامه برای مثال ۱

```

LD I 0.0
A I 0.1
= Q 2.0
BE

```

با DIP switch های روی کارت های ورودی و خروجی می‌توان آدرس بایتها را تنظیم کرد



آدرس کارت با تنظیم فوق 12H می‌شود. روی هر کارت ۴ بایت داریم. اگر کارت شماره صفر باشد آدرس بایت‌های روی آن می‌شود 0,1,2,3

کارت شماره یک → 4,5,6,7

کارت شماره دو → 8,9,10,11

پس از رسیدن به دستور BE پردازنده PLC مجدداً برنامه را از اول اجرا می‌کند. زمان یک بار اجرا شدن کامل برنامه را Cycle Time می‌گویند. اگر تغییرات سنسورها سریعتر از Cycle Time باشد موقع اجرا برنامه دچار مشکل می‌شویم. به طور تجربی اگر Cycle Time زیر ۱۰ میلی ثانیه باشد مشکل خاصی نداریم. یکی از پارامترهای انتخاب PLC همین مسئله می‌باشد که بستگی به توان CPU آن دارد.

## نوشتن برنامه در بلوکها

در PLC500 دو سری بلوک داریم. یکی OB ها و دیگری PB ها. OB ها فقط به OB1 و OB11 محدود می‌شوند لیکن PB ها می‌تواند از PB0 تا PB255 باشند. پروسه‌های تند و کند را توسط بلوک‌های مجزا جدا کرده و برنامه کنترل پروسه‌های کند را دیرتر اجرا می‌کنیم. مثلاً هر صدمبار که پروسه تند انجام می‌شود یک بار پروسه کند اجرا شود. این باعث می‌شود که Cycle Time کاهش پیدا کند. کاربرد دیگر PB ها نظم دادن به پروسه ها و ساختاریافتگی در برنامه نویسی است. ضمناً PB ها نباید بیشتر از هزار خط برنامه باشند. همچنین هر چه OB11 کوچکتر باشد سرعت بیشتر می‌شود. برای احضار یک PB از دستور ju استفاده می‌شود. برای تقسیم‌بندی داخلی PB ها جهت خوانایی بیشتر از \*\*\* استفاده می‌شود. به هر بخش از برنامه یک Segment می‌گویند. در برنامه مثال ۱ می‌توان بجای LD از دستور A نیز استفاده کرد زیرا AND اول کار Load را هم می‌کند.

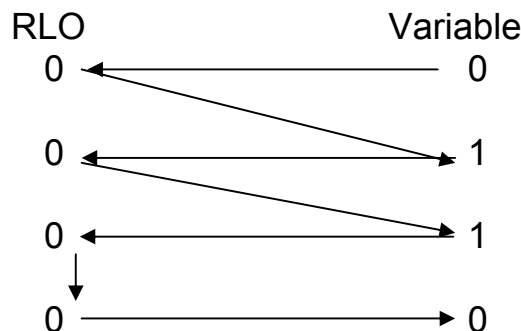
### مثال ۲:

AND سه ورودی

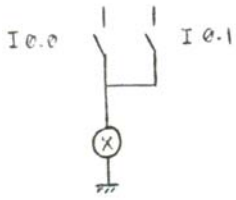
```
A I 0.0
A I 0.1
A I 0.0
= Q 0.0
BE
```

## RLO (result of logic operation)

رجیستر فوق نتایج عملیات منطقی را نگهداری می‌کند.



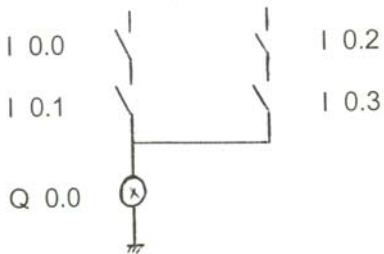
مثال ۳: OR دو ورودی



O | 0.0  
 O | 0.1  
 = Q 0.0  
 BE

### استفاده از پیرانتزها

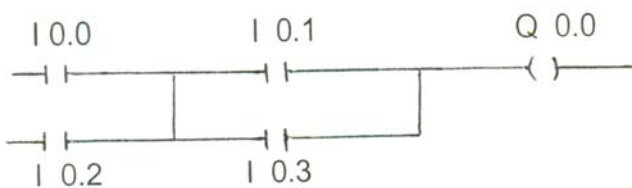
مثال ۴:



A | 0.0  
 A | 0.1  
 O (  
 A | 0.2  
 A | 0.3  
 )  
 = Q 0.0  
 BE

در هر برنامه فقط ۷ پیرانتز می‌توان باز کرد. با هر بار پیرانتز باز کردن محتویات RLO یک بیت شیفت پیدا می‌کند. برنامه فوق را روی PLC اجرا کرده و با Status نتیجه آنرا ببینید.

مثال ۵:



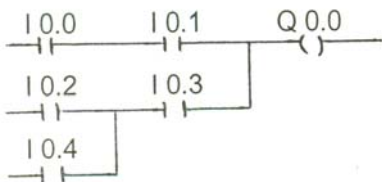
```

A (
O I 0.0
O I 0.2
)
A (
O I 0.1
O I 0.3
Q 0.0
)
BE

```

با اجرائی این برنامه روی PLC و مشاهده وضعیت در قسمت Status تاثیر بکارگیری پیرانتزها را در RLO ببینید.

مثال ۶:



```

A I 0.0
A I 0.1
O (
A (
O I 0.2
O I 0.4
)
A I 0.3
)
= Q 0.0
BE

```

### فلگ‌ها

در PLC کنترولیک نیمه پائینی حافظه RAM یعنی از صفر تا ۵۰٪ ناپایدار و نیمه بالایی پایدار است. Flagها از صفر تا ۲۵۵ ناپایدار و از ۲۵۶ تا ۵۱۱ پایدار (Retentive) می‌باشند. علت تقسیم‌بندی فضا به پایدار و ناپایدار برای قطع برق است. اطلاعاتی را که نمی‌خواهیم با قطع برق نگهداری شود (وضعیت کلید استارت از نوع Push button) در NON\_Ret. نگهداری می‌کنیم. دانستن محل فلگها در حافظه Ram برای برنامه نویسی مهم نیست چون خود PLC اینرا می‌داند و ممکن است فضایی فلگهای پایدار و ناپایدار در حافظه پیوسته نباشد.

فلگها جای مناسبی برای ذخیره نتایج میانی می‌باشد. مثال قبل را با استفاده فلگها به شکل زیر می‌نویسیم.

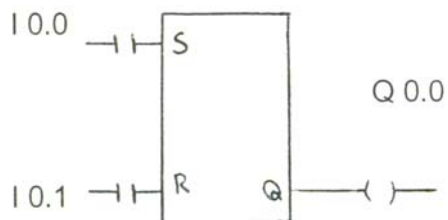


مثال ۷:

A I 0.0  
A I 0.0  
= F 5.0  
O I 0.2  
O I 0.4  
= F 5.1  
A F 5.1  
A I 0.3  
= F 5.2  
O F 5.0  
O F 5.2  
= Q 0.0  
BE

در PLC های زیمنس محل فلگهای پایدار و ناپایدار بر عکس PLC کنترنیک است.

### فلپ فلاپ SR یا RS

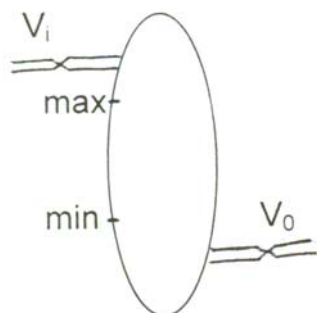


دستورات set و reset به RLO بستگی دارند و اگر  $RLO=1$  باشد اجرا می شوند.

A I 0.0  
S Q 0.0  
A I 0.1  
R Q 0.0  
BE

## مثال ۸ :

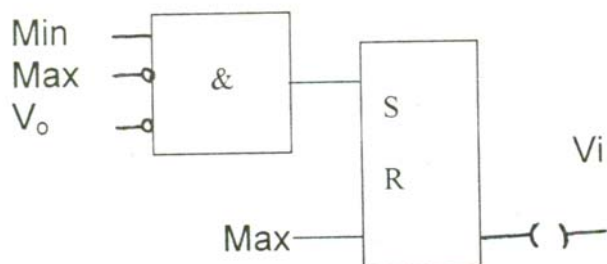
فرض کنید مخزن روبرو را با دو سنسور و دو شیر داریم



$\text{min} = 1, \text{max} = 0 \Rightarrow V_i = \text{باز}, V_o = \text{بسته}$

$\text{min} = 0, \text{max} = 1 \Rightarrow V_i = \text{بسته}, V_o = \text{باز}$

برنامه آنرا بنویسید و اجرا کنید.



## مهم: فرق SR و RS

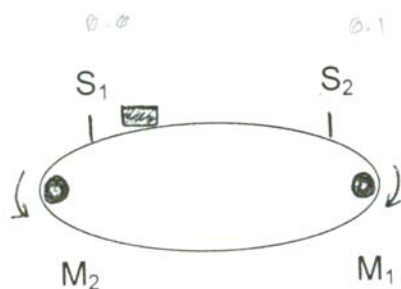
بطور کلی در PLCها هر دستوری که پایین تر باشد غالب است.

A I 0.0  
S Q 2.0  
A I 0.1  
R Q 2.0  
\*\*\*

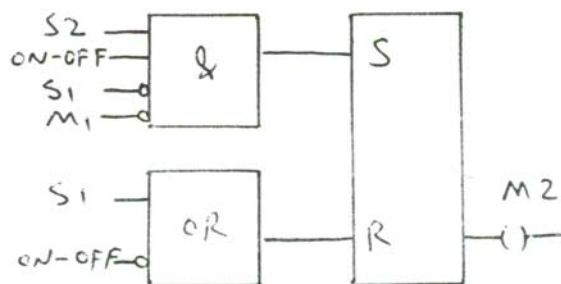
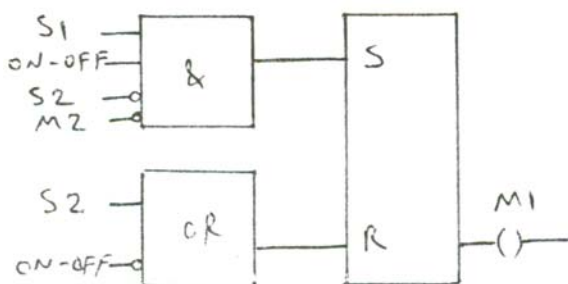
A I 0.2  
R Q 2.1  
A I 0.3  
S Q 2.1  
BE

از این موضوع برای ایمنی استفاده می‌شود. در مثال مخزن پیش اگر هر دو سنسور min و max عمل کنند و به ترتیب SR استفاده شده باشد در این صورت هر دو valve بسته می‌مانند. فرق RS با SR در شرایطی است که هر دو ورودی یک باشد. در آن صورت RS یک می‌شود ولی SR صفر می‌شود. اینکه از کدام یک استفاده کنیم بستگی به نوع پروسه دارد.

مثال ۹:



بعد از گذاشتن قطعه و روشن کردن کلید، موتور  $M_1$  روشن می‌شود و قطعه به سمت راست می‌رود تا به  $S_2$  برسد که در این صورت باید  $M_1$  خاموش و  $M_2$  روشن شود و قطعه در این مسیر تا زمانی رفت و برگشت میکند که کلید ON-OFF روشن باشد.



A I S<sub>1</sub>  
 A I ON/OFF  
 AN I S<sub>2</sub>  
 AN Q M<sub>2</sub>  
 S Q M<sub>1</sub>  
 O I S<sub>2</sub>  
 ON I ON/OFF  
 R Q M<sub>1</sub>

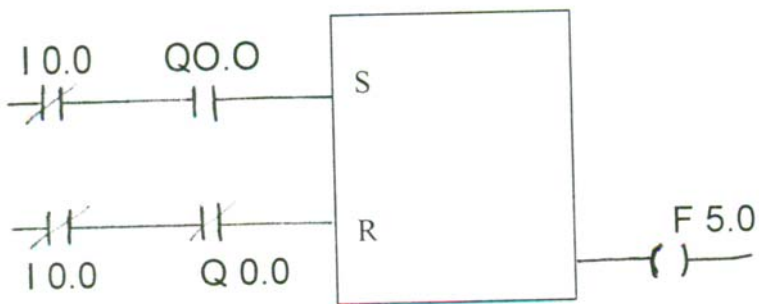
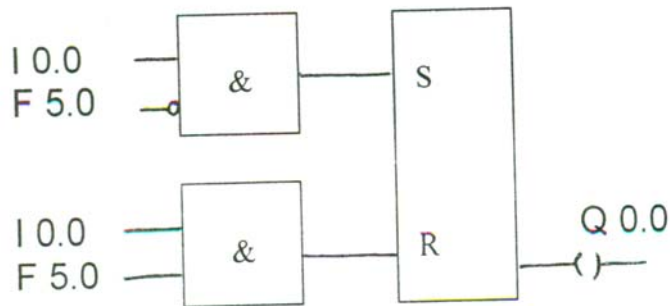
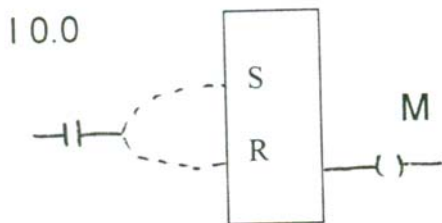
\*\*\*

A I S<sub>2</sub>  
 A I ON/OFF

AN I S<sub>1</sub>  
 AN Q M<sub>1</sub>  
 S Q M<sub>2</sub>  
 O I S<sub>1</sub>  
 ON I ON/OFF  
 R Q M<sub>2</sub>  
 BE

**مثال ۱۰:**

يك كليد فشاري داريم كه اگر يك بار فشار دهيم موتور روشن مي‌شود و بار ديگر فشار دهيم خاموش مي‌شود مثل دكمه خاموش و روشن كنترل تلويزيون.



برنامه مدار فوق را روی PLC نوشته و اجرا کنید.

**نکته:** اول PBها را به PLC بفرستید بعد OB1 را.

## Edge Detector

سوال: برنامه زیر چه کار می‌کند؟

A I 0.0  
AN F 8.0  
= F 20.0  
A I 0.0  
= F 8.0  
BE

F 20.0 پالسی به اندازه يك Cycle time ایجاد می‌کند (در لحظه‌ای که I 0.0 فعال شود - در لبه بالا رونده)

به عبارت دیگر این مدار آشکار ساز لبه است (Edge detector). اگر اپراتور کلید را زمان زیادی نگهدارد ممکن است برای اجرای Cycle time های بعدی مشکل ایجاد شود. مدار فوق (برنامه فوق) این مشکل را حل می‌کند.

## WDT (watch dog timer)

این تایمر فرجه زمانی خاصی (مثلاً 100ms) فرصت می‌دهد و اگر در این فرصت سیکل تمام نشد حتماً اشکالی وجود دارد که در این صورت PLC را Stop می‌کند. زمان شروع عملکرد تایمر فوق از لحظه ارسال خروجیهاست (یعنی اجرای دستور BE برنامه OB1).

## کار کردن با اعداد

برای کار کردن با ورودیها به صورت بایت از IB استفاده می‌شود

FB 10  
IB 0  
QB 1

اکومولاتور يك رجیستر ۴ بایتي است.



Acc4      Acc3      Acc2      Acc1

دستورات T, L برای انتقال اطلاعات به و از اکومولاتور استفاده می‌شوند.  
دو بایت اول اکومولاتور برای محاسبات اعداد صحیح استفاده می‌شود و دو بایت آخر برای محاسبات اعشاری.

L IB 0  
T QB 1  
BE

### جمع کردن دو عدد

L IB 0  
+ IB 1  
T QB 1  
BE

### کار کردن با WORDها

یک Word از دو بایت تشکیل شده است.

FW  
IW  
QW

اگر دستور L IW 0 اجرا شود نتیجه به شکل زیر است:

Acc2		Acc1	
	IB 1		IB 0

در PLC های زیمنس این دو بایت بر عکس قرار می گیرند.

### استفاده از اعداد ثابت

KH ۱۶ بیت HEX 0 FFFF

مثال

L KH 25A7  
T QW 0  
BE

نکته: برای پروژه زمانی (مثل چراغ راهنما) از ارسال عدد استفاده می کنیم.

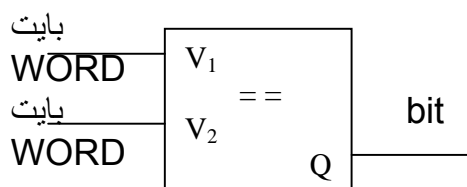
اعداد اعشاری:

KD	یک بایت دسیمال	0	255
KW	دو بایت دسیمال	0	65535

## مقایسه اعداد با هم

==	مساوي
!=	نامساوي
>	بزرگتر
>=	بزرگتر مساوي
<	کوچکتر
<=	کوچکتر مساوي

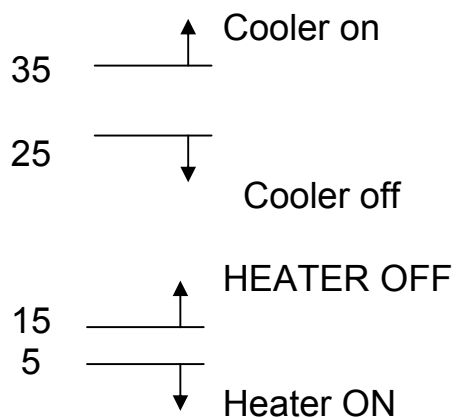
### Ladder فرم



L IB 0  
 == IB 1  
 = Q 2.0  
 BE

### مثال ۱۱:

پروژه کنترل دمای اتاق با شرایط مقابل:



```

L IB 0
>= KD 35
S Q 0.0
<= KD 25
R Q 0.0
<= KD 5
S Q 1.0
>= KD 15
R Q 1.0
BE

```

در PLC هاي زيمنس بجاي KD از KF استفاده مي شود.

### مثال ۱۲ :

برنامه اي بنويسيد كه با زدن كليد I0.0 خروجي به صورت يك در ميان روشن شود.



```

A I 0.0
LC KH 5555
TC QW 0
AN I 0.0
LC KH AAAA
TC QW 0
BE

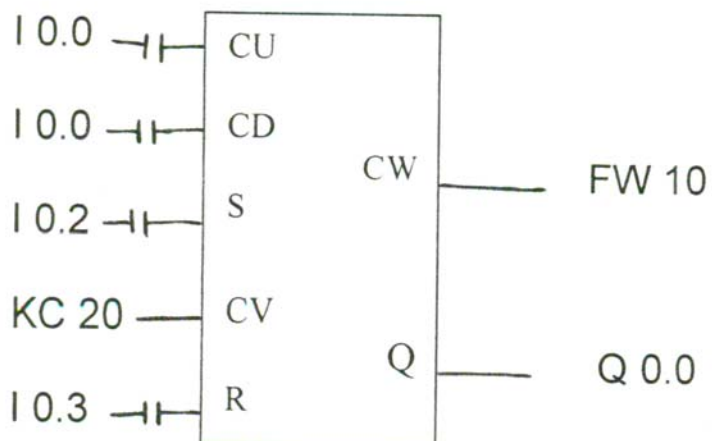
```



## شمارنده‌ها

مثال ۱۳:

زمانی که مقدار شمارنده صفر باشد Q صفر می‌شود و وقتی که غیر صفر باشد يك می‌شود.



تعداد شمارنده‌های موجود در PLC یکی دیگر از پارامترهای فنی مهم PLC برای انتخاب مدل برای خرید می‌باشد. شمارنده‌ها به دو قسمت پایدار و ناپایدار تقسیم می‌شوند.

A I 0.0  
CU C1  
A I 0.1  
CD C1  
A I 0.2  
L KC 20  
S C1  
A I 0.3  
R C 1  
L CW 1  
T FW10  
A C 1  
= Q 0.0  
BE

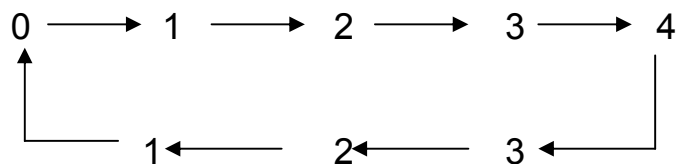
### مثال ۱۴:

برنامه‌ای بنویسید که شمارش را تا ۳۰ ادامه دهد و بعد صفر شود.

```
A I 0.1
CU C 1
L CW 1
T QW 1
== KW 30
R C 1
BE
```

### مثال ۱۵:

برنامه‌ای بنویسید که شمارش زیر را با یک کلید انجام دهد:



```
A F 20.0
AN F 1.0
CU C 1
A F 20.0
A F 1.0
CD C 1
L CW 1
T QW 1
== KW 4
S F 1.0
L CW 1
== KW 0
R F 1.0
A I 0.0
AN F 8.0
= F 20.0
A I 0.0
= F 8.0
BE
```

### مثال ۱۶:

پارکینگی با ظرفیت ۲۰ ماشین که در آن ۵ ماشین ثابت قرار دارد. اگر تعداد ماشین‌ها ۵ بود درب ورود باز درب خروج بسته چراغ E روشن و شمارش سنسور خروجی اثری ندارد.

$$5 < n \leq 10 \Rightarrow \left\{ \begin{array}{l} \text{درب ورود باز} \\ \text{درب خروج باز} \\ \text{چراغ E روشن} \end{array} \right.$$

$$10 < n < 20 \Rightarrow \left\{ \begin{array}{l} \text{درب ورود باز} \\ \text{درب خروج باز} \\ \text{چراغ M روشن} \end{array} \right.$$

$$n = 20 \Rightarrow \left\{ \begin{array}{l} \text{درب ورود بسته} \\ \text{درب خروج باز} \\ \text{چراغ F روشن} \\ \text{شمارش سنسور ورودی اثر ندارد} \end{array} \right.$$

وقتی کلید ON/OFF روشن شود مدار کار کند و وقتی خاموش شود مدار کار نکند. برنامه لازم برای کنترل پارکینگ با مشخصات فوق را نوشته و اجرا کنید.

**OB1**  
JU PB 27  
BE

## PB 27

A I 0.1	L CW 1	***
AN F 5.1	<= KD 10	***
CU C 1	= F 8.1	L CW 1
A I 0.2	A F 8.0	== KD 20
AN F 5.0	A F 8.1	R Q 0.0
CD C 1	S Q 0.0	S Q 0.1
A I 0.0	S Q 0.1	R Q 0.2
L KC 5	S Q 0.2	R Q 0.3
S C 1	R Q 0.3	S Q 0.4
AN I 0.0	R Q 0.4	= F 5.1
R C 1	L CW 1	***
***	> KD 10	L CW 1
L CW 1	= F 8.2	== KD 0
== KD 5	L CW 1	R Q 0.0
S Q 0.0	< KD 20	R Q 0.1
R Q 0.1	= F 8.3	R Q 0.2
S Q 0.2	A F 8.2	R Q 0.3
R Q 0.3	A F 8.3	R Q 0.4
R Q 0.4	S Q 0.0	BE
= F 5.0	S Q 0.1	
L CW 1	R Q 0.2	
> KD 5	S Q 0.3	
= F 8.0	R Q 0.4	

در برنامه فوق:

ON/OFF	I 0.0
S1	I 0.1
S2	I 0.2
درب ورود	Q 0.0
درب خروج	Q 0.1
چراغ E	Q 0.2
چراغ M	Q 0.3
چراغ F	Q 0.4

ضمناً وقتی کلید ON/OFF خاموش می شود باید درب ها بسته و چراغها خاموش شوند. (مهم)

## کار با شبیه ساز 7-Seg

ابتدا با مراجعه به راهنمای کار با این شبیه‌ساز و مطالعه کامل و دقیق آن و اجرای برنامه‌های نوشته شده در آن با این شبیه‌ساز آشنا شوید.

### مثال ۱۷:

برنامه‌ای بنویسید که شمارش 4 → 1 را روی خروجی 7-Seg نمایش دهد.

برای نمایش ۱ بایستی 81H یا 129 را روی خروجی بفرستیم.

ابتدا با رسم یک جدول کلید اعداد Hex یا Dec لازم برای نمایش ارقام 0 تا 9 را بدست آورید.

A I 0.1

R C 1

\*\*\*

برنامه مثال ۱۵

\*\*\*

L CW 1

== KD 0

LC KD 183

TC QB 0

\*\*\*

L CW 1

== KD 1

LC KD 129

TC QB 0

.

.

.

.

برای برنامه‌نویسی مایکروکنترلر می‌توان کار زیر را انجام داد:

.

.

L CW 1

T FB 50

JU PB 230

L FB 52

T QB 0

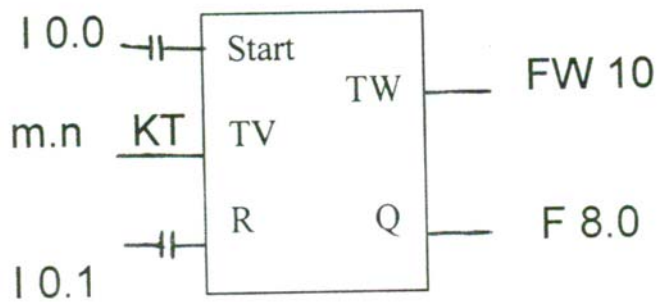
BE

حال در برنامه PB230 داریم :

L FB 50  
== KD 0  
LC KD 183  
TC FB 52  
.  
.  
.

### تایمرها

مثال ۱۸ :



$$T = mx$$

$$5 = mx1 \rightarrow m = 5 \Rightarrow KT = 5.2$$

$$5 = mx0.1 \rightarrow m = 50 \Rightarrow KT = 50.1$$

$$5 = mx0.01 \rightarrow m = 500 \Rightarrow KT = 500.0$$

دقت T

**Time base (عدد دقت)**

بهترین دقت 0.01 ثانیه می باشد.

n=0	0.01 S
n=1	0.1 S
n=2	1 S
n=3	10 S

m تعداد سرزدن هاي CPU به تایمر است.  
 در مراجع PLC اطلاعات کامل آمده است.  
 TW زماني که باقي مانده را نشان مي دهد زیرا تایمر برعکس مي شمارد.

### برنامه مثال فوق به زبان STL:

```

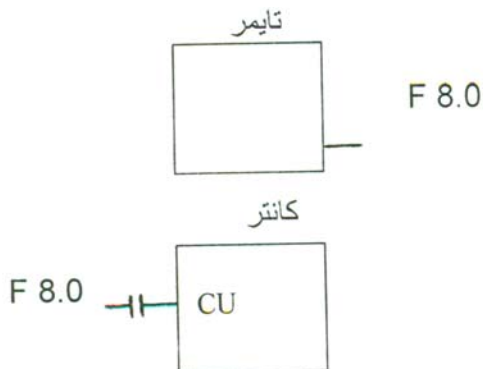
A I 0.0
AN F 8.0
L KT 5.2
SPT 1
A I 0.1
RT 1
L TW 1
TFW 10
AT 1
= F 8.0
BE
    
```

دستور SP دستور راه اندازي تایمر است که نوع تایمر را هم مشخص مي کند. تایمر SE فقط احتیاج به لبه بالارونده دارد. اگر در بين زمان تایمر مجدداً START زده شود T طولاني تر مي شود. مثل تایمر لامپ هاي راه پله ها در آپارتمانها. وقتی Reset مي آید خروجي صفر مي شود. در نوع SP هم لبه مي خواهيم و هم سطح. با دقت در نمودارهاي زماني مراجع PLC عملکرد تایمرها را شناسايي کرده و تحليل نمائيد.

کلاً تایمرها 6553 ثانیه زمان مي گیرند. يعني حدود يك ساعت و ۴۰ دقیقه. در صورتي که بخواهيم مثلاً هر ۱۰ ساعت يك بار کاري را انجام دهيم باید چند تایمر را بکار بگيريم. اگر زمان زياد باشد مي توان از يك تایمر و يك کانتر (شمارنده) استفاده کرد. همیشه در STL اول شمارنده را بنويسيد.

### مثال ۱۹:

با استفاده از يك تایمر و کانتر ۵۰ ثانیه زمان را اندازه گيري کنید.



```

ANT 1 يا AN F 8.0
CU C 1
A 10.0
ANT 1
L KT 5.2      ٥ ثانيه
SE T 1
L CW 1
== KD 10      يعني ٥٠ ثانيه
SQ 0.0
AN I 0.0
R Q 0.0
R C 1
BE

```

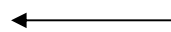
### مثال ٢٠:

برنامه‌اي بنويسيد كه روي 7-Seg اعداد 0 تا 9 را هر ثانيه يك عدد بشمارد.

```

ANT 1
CU C 1
L CW 1
>= KD 10
R C 1
A 10.0
ANT 1
L KT 1.2
SE T 1
L CW 1
T FB 50
JU PB 230
L FB 52
T QB 0
BE

```



نمایش اعداد



## شبیه‌ساز چراغ راهنما

ابتدا دفترچه راهنمای شبیه‌ساز چراغ راهنمایی (Traffic Lights simulator) را مطالعه و برنامه‌های نوشته شده در آن را اجرا کنید.

### مثال ۲۱:

با استفاده از تایمر و کانتر برنامه‌ای بنویسید که چراغ‌های راهنما را به ترتیب زیر روشن کند:

5 S R1, G<sub>2</sub>  
3 S R1, Y<sub>2</sub>  
5 S G1, R<sub>2</sub>  
3 S Y1, R<sub>2</sub>

برنامه به زبان STL:

```
AN T1
CU C1
A I 0.0
AN T 1
L KT 1.2
SE T 1
L CW 1
>= KD 0
LC KD 33
TC QB 0
> KD 5
LC KD 17
TC QB 0
> KD 8
LC KD 12
TC QB 0
> KD 13
LC KD 10
TC QB 0
== KD 16
R C 1
BE
```

## دستور پيرش شرطي

پرش شرطي به مقدار RLO وابسته است و اگر RLO يك باشد اجرا مي شود و اگر صفر باشد اجرا نمي شود.

مثال ۲۲:

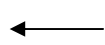
### OB1

A I 0.0

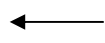
JC PB 70

A I 0.1

JC PB 80



برنامه توليد رب گوجه فرنگي



برنامه توليد پفك نمكي

مثال ۲۳:

### OB1

A I 0.0

JC PB 70

AN I 0.0

JC PB 80

مود اتوماتيك

مود دستي

### PB70

L KH 5555

T QW 0

BE

### PB 80

L KH AAAA

T QW 0

BE

## شبيه‌ساز كنترل سطح تانك

### مثال ۲۴:

ابتدا راهنمای شبيه‌ساز Tank Level Control را مطالعه و برنامه‌هاي آنرا اجرا كنيد. همانطور كه ملاحظه مي‌كنيد اين شبيه‌ساز در دو مورد دستي و اتوماتيك كار مي‌كند. در مورد اتوماتيك بعد از پرسدن تانك، ۵ ثانيه همزن كار مي‌كند و بعد خروجي باز مي‌شود. با توجه به اتصال ورودي شبيه‌ساز به IB 2 در OB1 بايد A I 2.0 را قرار دهيم.

مشكلاتي كه ممكن است در عمل وجود بيابد و بایستی در برنامه‌نویسی برای حل آنها اقدام لازم صورت گیرد به قرار زیر است:

۱- وقتی به مود دستي مي‌رويم بايد موتور خاموش شود

A I Auto/man.  
= Q چراغ Auto  
JC PB 70  
AN I Auto/man  
R Q M  
JC PB 80  
BE

۲- وقتی از وسط مود اتوماتيك به دستي مي‌رويم و بعد برمي‌گرديم سيكل اتوماتيك را شروع نمي‌كند. برای حل اين مشكل از Flag مي‌توان استفاده كرد (برای اينكه بدانيم در حال پر كردن مخزن بوده يا خالي كردن). راه ساده تر اينست كه يك حالت را شروع كنيم. مثلاً شروع كنيم به پر كردن مخزن. لذا برای پر كردن مخزن وجود شرط Lmin لازم نيست.

در پياده‌سازي سيستم‌هاي كنترل با استفاده از PLC بایستی کارشناسان ضمن تحليل دقيق سيستم، موارد اينچيني را همواره مد نظر داشته باشند تا وقوع شرايط ذکر نشده توسط مشتري، باعث ايجاد اشكال در عملکرد سيستم كنترل نشود. بدیهی است عدم در نظر گرفتن موارد اينچيني مي‌تواند آسیب‌هاي جدي به PLANT يا خود سيستم كنترل وارد نمايد.

## شبیه‌ساز رآکتور مواد شیمیایی

ابتدا راهنمای شبیه‌ساز Reaction vessel را مطالعه کنید.

### **مثال ۲۵:**

- ۱- چنانچه درجه حرارت یا فشار داخلی سیستم داخلی سیستم پایین باشد سیستم گرم کننده و همزن شروع به کار نموده LED مربوط به حالت Startup روشن شود.
- ۲- در صورتی که شرایط نرمال باشد تنها چراغ Normal روشن شود.
- ۳- در صورتی که فشار داخل سیستم بالا باشد شیر اطمینان باز شده LED مربوط به حالت ALARM چشمک بزند.
- ۴- در صورتی که درجه حرارت داخلی سیستم بالا باشد COOLING WATER باز شود LED مربوط به حالت ALARM چشمک بزند و همزن نیز شروع به کار کند.

برای چشمک زدن از FB126 استفاده کنید

7						1	0
12.8	6.4	3.2	1.6	0.8	0.4	0.2	0.1

### تحلیل:

وقتی فشار بالا باشد دما هم بالاست. بایستی به نحوی از صحت سنسورها اطمینان حاصل شود. همزن وقتی کار می‌کند که یا هیتر یا کولر کار کند. ALARM وقتی چشمک می‌زند که شیر اطمینان یا کولر روشن شوند.

### انواع پروسه‌ها:

- ۱- ترتیبی: مانند مثال قبل
- ۲- ترکیبی: مانند این مثال

در پروسه‌های ترتیبی از RESET, SET استفاده کنید.  
در پروسه‌های ترکیبی از OR, AND استفاده کنید.

O I 2.7  
O I 2.5  
= Q 0.3  
= Q 0.4  
\*\*\*  
AN I 2.7  
AN I 2.6

```

AN I 2.5
AN I 2.4
= Q 0.5
A I 2.4
AN I 2.7
= Q 0.1
***
A I 2.6
AN I 2.5
= Q 0.2
.
.
.

```

\* با مراجعه به کتابچه راهنمای کاربری PLC 500 دستورات STL را مرور کنید.  
 \*\* با مراجعه به راهنمای نرم افزار PLC500 با عملکرد بخشهای مختلف نرم افزار آشنا شوید.

```

:L      IB  2
:T      FB  0
****
:A      F   0.1
:A      F   0.3
:=      F   1.1
****
:AN     F   0.0
:A      F   0.3
:=      F   1.2
****
:AN     F   0.1
:A      F   0.2
:O(
:A      F   0.0
:AN     F   0.3
:)
:=      F   1.3
****
:O      F   1.2
:O      F   1.3
:=      F   1.0
****
:A      F   0.0
:=      F   1.4
****
:AN     F   0.0
:AN     F   0.1
:=      F   1.5
****
:A      F   0.1
:A      F  126.2
:=      F   1.6
:A      I   1.6
****
:L      FB  1
:T      QB  0
:BE

```

## چند نکته

- دستور S با دستور = فرق می‌کند. اگر Set کنیم باید یک فکری هم برای reset کنیم.
- از دستور JU زیاد استفاده نکنید. چون برنامه را کند می‌کند و ممکن است LOOP های بی‌نهایت درست کند.
- بعد از دستور BE هر دستوری باشد اجرا نمی‌شود ولی بعد از دستورات BEU یا BEC دستورات بعدی اجرا می‌شوند. این دو دستور در وسط برنامه استفاده می‌شوند.

مثال:

```
A I 0.0  
JC = MTX  
BEU
```

.

.

```
MTX: L KH AAAA  
T QW 2  
BE
```

- دستورات IN و OUT برای کارتهای آنالوگ هستند که در دوره PLC پیشرفته در مورد آنها بحث خواهد شد.

موفق باشید