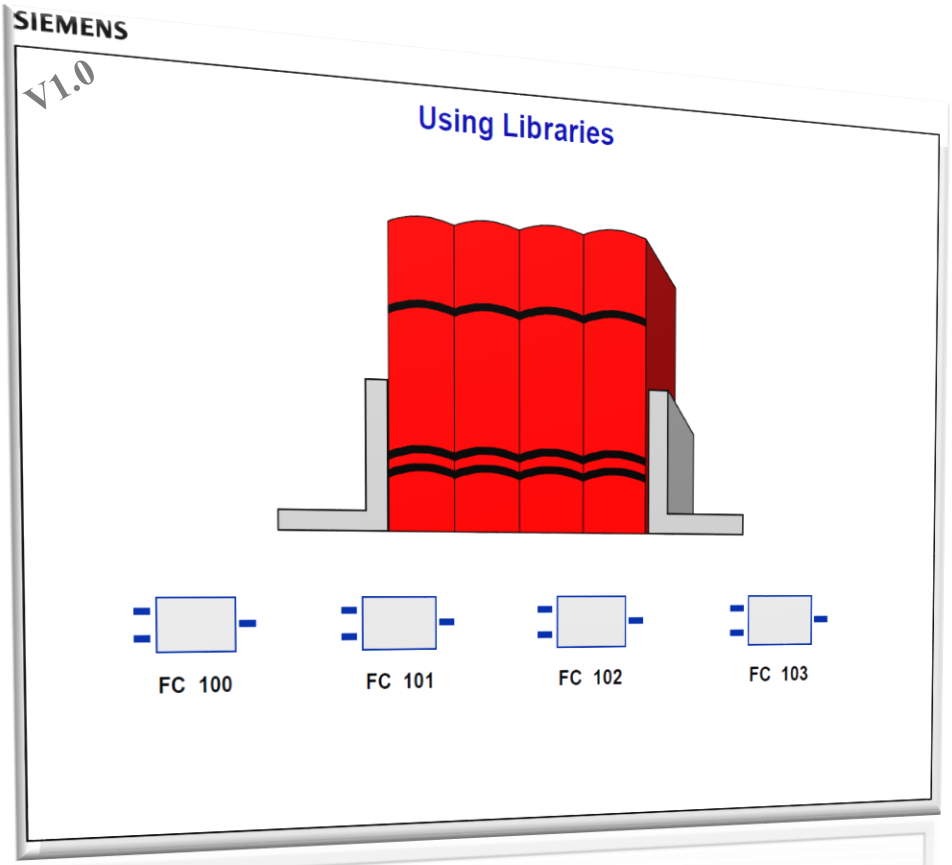
SIEMENS
Ingenuity for life

آشنایی با فانکشن های پرکاربرد در کتابخانه نرم افزار

STEP 7

PART -1



مهندس اکبر اویسی فر

إنَّ لِكُلِّ شَیءٍ زَكاةً ، وزَكاةُ العِلمِ أن يُعَلِّمَهُ أهلَهُ

هرچیزی زکاتی دارد و زکات دانش، آموختن آن است به اهلش

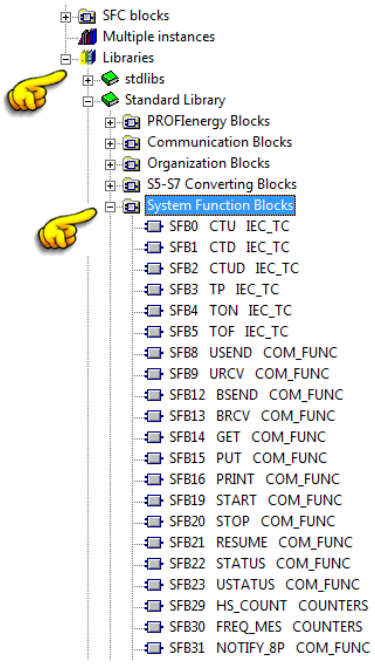
آشنایی با فانکشن های سیستمی در STEP 7

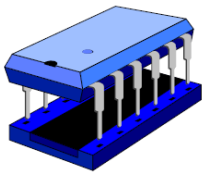
در اکثر نرم افزارهای اتوماسیون، معمولا یکسری فانکشن آماده طراحی و در قالب یک کتابخانه استاندارد، در نرم افزار در اختیار برنامه نویس قرار می گیرد. این فانکشن ها در واقع توابعی آماده می باشند که برای کاربردهای مختلف استفاده می شوند. به عنوان مثال جهت تبادل دیتا در شبکه های مختلف، برنامه نویسان نیاز به یکسری فانکشن جهت ارسال و دریافت دیتا در شبکه دارند. به همین منظور گروهی از این فانکشن ها برای انجام عملیات ارسال و دریافت دیتا در شبکه به کار می روند. همچنین در بسیاری از کاربردها برنامه نویس نیازمند طراحی یک الگوریتم PID به صورت نرم افزاری می باشد. شرکت های سازنده برای طراحی کنترلرهای PID تعدادی فانکشن طراحی و در نرم افزار در اختیار کاربران قرار می دهند. تعداد این فانکشن ها در نرم افزار مورد بحث بسیار زیاد بوده و پرداختن به تمامی آنها در این مجموعه امکان پذیر نمی باشد. در این قسمت سعی شده که فانکشن های پرکاربرد در کتابخانه نرم افزار مورد بررسی قرار گیرند.

Navin Tanavaran Avin Co

نکات:

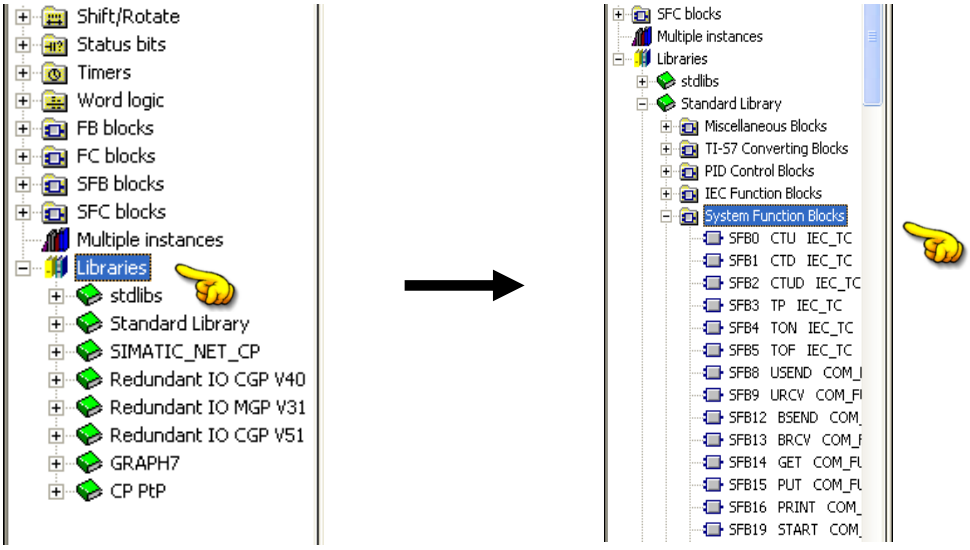
- ✓ فانکشن های سیستمی SFB-SFC در حافظه CPU ذخیره شده اند و قابل پاک شدن نمی باشند.
- ✓ تعداد فانکشن سیستمی که هر CPU می تواند ساپورت کند، با یکدیگر متفاوت می باشند. تعداد آنها بسته به مدل و ورژن CPU می تواند متفاوت باشد.
- ✓ با Online شدن به CPU در محیط نرم افزار، لیست فانکشن های سیستمی نمایش داده می شوند.
- ✓ فانکشن های سیستمی(SFB-SFC) نیاز به دانلود به CPU ندارند.





محل قرارگیری فانکشن های سیستمی

تمامی فانکشن ها در کتابخانه Libraries نرم افزار در نوار سمت چپ محیط برنامه نویسی در دسترس می باشند. فانکشن های سیستمی نیز زیرمجموعه گزینه System Function Block می باشند.



در ادامه بحث با فانکشن های سیستمی پرکاربرد آشنا می شویم.

فانکشن های سیستمی مربوط به کانترها و تایمرهای IEC

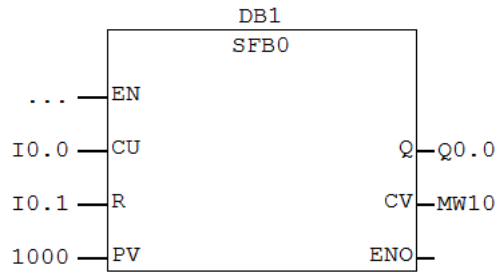
(SFB0) CTU-IEC

این فانکشن یک شمارنده بالا شمار مطابق با استاندارد IEC می باشد. در شمارنده های IEC نیازی به استفاده از دستور مقایسه کننده به ازای CV≥PV برای تولید فرمان خروجی، همانند شمارنده های S7 نمی باشد. دلیل این مطلب این است که در این شمارنده ها زمانی که مقدار جاری شمارنده، برابر مقدار ست شده در ورودی PV می شود، خروجی بیتی شمارنده فعال می گردد. حداکثر مقدار قابل شمارش در این شمارنده ها 32767 می باشد. در ضمن لازم به ذکر است که این شمارنده ها، شمارنده های نرم افزاری می باشند و تعداد آنها در حافظه سیستمی CPU همانند شمارنده های سخت افزاری تعریف نشده است.

Novin Fanavaran Asia co

نکته:

فانکشن های SFB نیاز به بلوک های DB اختصاصی دارند. تمامی پارامترهای فانکشن، آدرسی در دیتابلاک اختصاص داده شده دارند. این آدرس ها در بخش های مختلف برنامه و همچنین در نرم افزارهای مانیتورینگ در دسترس می باشند. در این مجموعه هدف از آدرس دهی مجدد پارامترها توسط حافظه M، آشنایی هر چه بیشتر خوانندگان محترم با تایپ پایه های فانکشن ها می باشد. در شکل زیر یک فانکشن SFB0 را ملاحظه می کنید.



EN : ورودی فعال ساز فانکشن

CU : ورودی بالا شمار

R : ورودی ریست

PV : ورودی Set Point

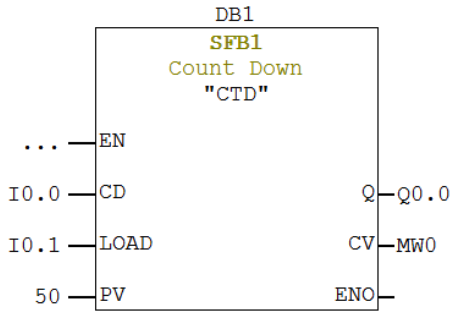
Q : خروجی بیتی

CV : مقدار جاری شمارنده

ENO : خروجی بیتی فعال ساز

(SFB1) CTD-IEC

این فانکشن یک شمارنده پایین شمار می باشد. در این شمارنده با هر لبه بالارونده ورودی CD، از عدد تنظیم شده اولیه PV یک واحدکم می‌شود. با رسیدن مقدار شمارنده به صفر، خروجی بیتی شمارنده 1 می شود. هنگامی که ورودی LD یک شود، عدد شمارنده برابر مقدار PV شده و بیت شمارنده نیز صفر می‌شود. در واقع با این ورودی، شمارنده با مقدار اولیه ری ست می‌شود. بازه شمارش در این شمارنده 32768- تا 32767+ می باشد.



EN : ورودی فعال ساز فانکشن

CD : ورودی پایین شمار

LOAD : ورودی جهت بار کردن مقدار PV

PV : مقدار دلخواه جهت بارگذاری در شمارنده

Q : خروجی بیتی

Novin Fanavaran Arian Co.

CV : مقدار جاری شمارنده

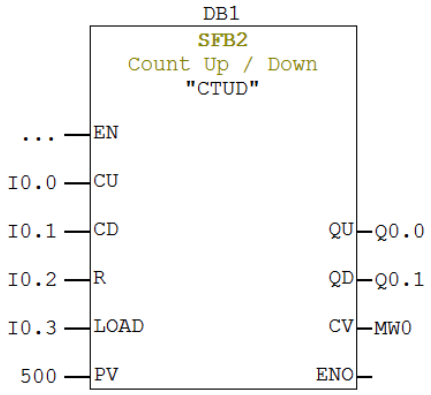
ENO : خروجی بیتی فعال ساز

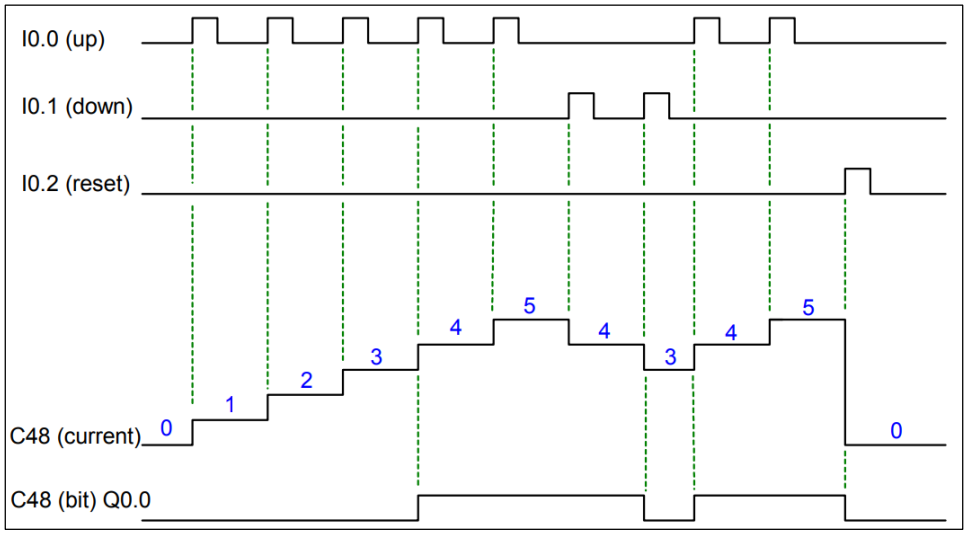
(SFB2) CTUD-IEC

این فانکشن یک شمارنده بالا / پایین شمار بوده که قابلیت شمارش در هر دو جهت را دارا می باشد. در این شمارنده اگر در یک سیکل برنامه همزمان به هر دو ورودی CU و CD پالسی ارسال شود کانتر مقدار قبلی را حفظ می کند.

بازه شمارش در این شمارنده نیز 32768- تا 32767+ می باشد. در خروجی این شمارنده جهت اطلاع از وضعیت شمارش به سمت بالا و یا پایین از دو خروجی بیتی با نام های QU و QD استفاده شده است. به عنوان مثال زمانی که خروجی CV کوچکتر مساوی عدد صفر باشد، خروجی QD فعال می شود. در غیر این صورت این خروجی در وضعیت صفر می باشد.

Akbar Oveisifar

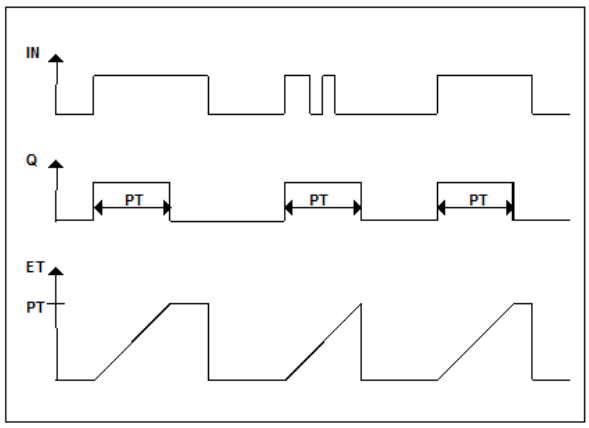


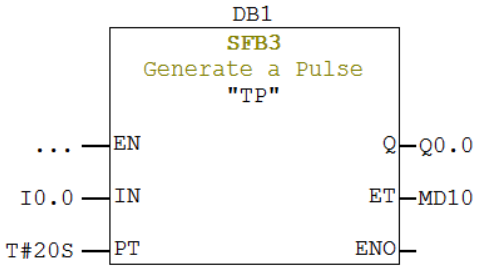


(SFB3) TP-IEC

این فانکشن یک تایمر تاخیر در قطع حساس به لبه بالا رونده ورودی می باشد. در واقع این تایمر با دریافت پالس در ورودی، خروجی(Q) را یک و شروع به شمارش کرده و بعد از سپری شدن زمان PT، خروجی را قطع می کند. اگر در حین شمارش زمان، ورودی به وضعیت 0 تغییر کند، در عملکرد تایمر تاثیری ایجاد نخواهد شد.

Akbar Oveisifar





EN : ورودی فعال ساز فانکشن

IN : ورودی راه انداز تایمر

PT : ورودی تعیین زمان با فرمت TIME

Q : خروجی بیتی

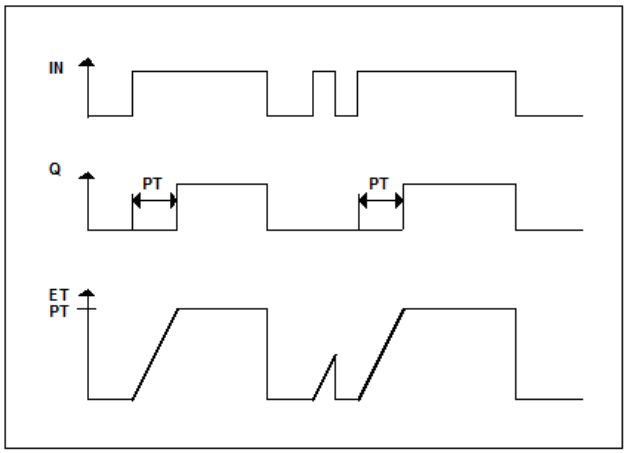
ET : خروجی مقدار زمان تایمر

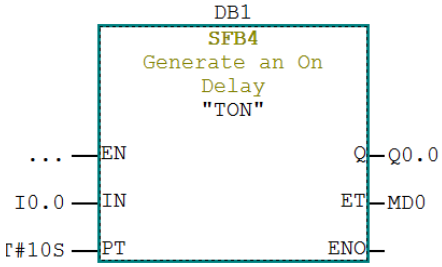
ENO : خروجی بیتی فعال ساز

نکته : در این تایمر پس از سپری شدن زمان PT ، با غیر فعال شدن ورودی IN مقدار زمان تایمر صفر می شود. در حالتی که ورودی IN در حین شمارش قطع شود، مقدار تایمر باز هم بعد از سپری شدن زمان PT صفر می شود.

(SFB4) TON-IEC

این فانکشن یک تایمر تاخیر در وصل حساس به لبه بالا و پایین رونده ورودی می باشد. این فانکشن به تایمر TON معروف می باشد. این تایمر عملکردی شبیه تایمر سخت افزاری S-ODT در تایمرهای S7 دارد.



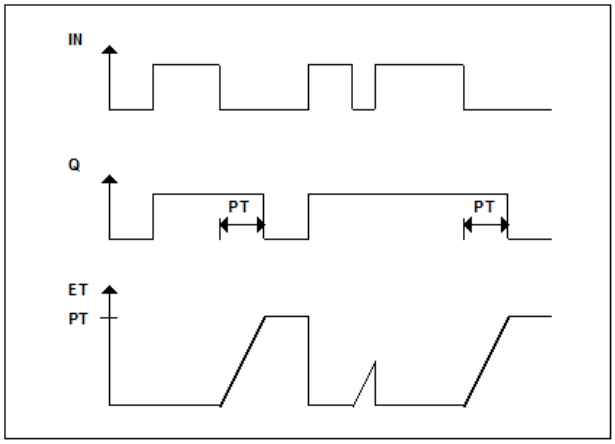


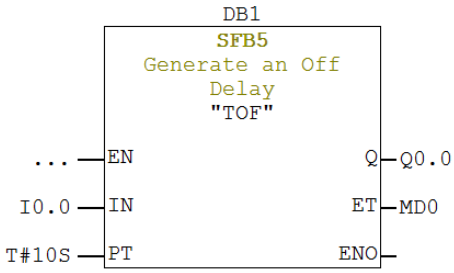
در این تایمر زمانی که ورودی IN غیرفعال شود، هم خروجی تایمر و هم زمان تایمر صفر می شود.

(SFB5) TOF-IEC

در این تایمر با یک شدن ورودی IN، بیت خروجی تایمر(Q) نیز یک می‌شود و زمان جاری صفر می‌شود. حال اگر ورودی تایمر به حالت صفر تغییر وضعیت دهد، زمان تایمر شروع و وقتی به مقدار PT رسید، بیت خروجی تایمر صفر و تایمر متوقف می‌شود. در صورتی که قبل از زمان PT ورودی IN خاموش شود، بیت خروجی تایمر در حالت 1 باقی می‌ماند. این تایمر عملکردی شبیه تایمر سخت افزاری S-OFFDT در تایمرهای S7 دارد.

Akbar Oveisifar





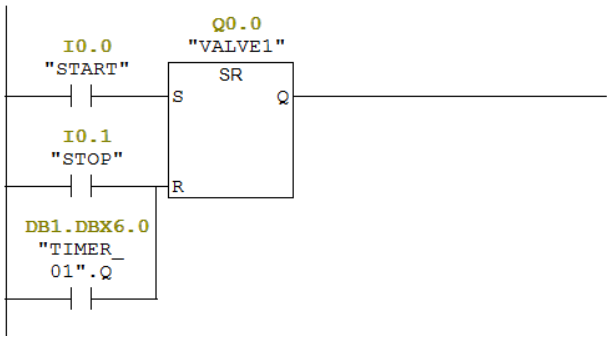
مثال ۱

در یک پروسه صنعتی فرض کنید که از دو ولو جهت باز و بسته کردن مسیر ورودی دو مایع به داخل یک مخزن استفاده می شود. نحوه ورود مایع به داخل مخزن به این صورت می باشد که با زدن شستی استارت، ولو اول باز شده و به مدت 1 دقیقه مایع A وارد مخزن می شود. پس از سپری شدن این زمان، ولو اول بسته و ولو دوم باز شود. مدت زمان باز بودن این ولو نیز 30 ثانیه می باشد.

Novin Fanavaran Aria Co.

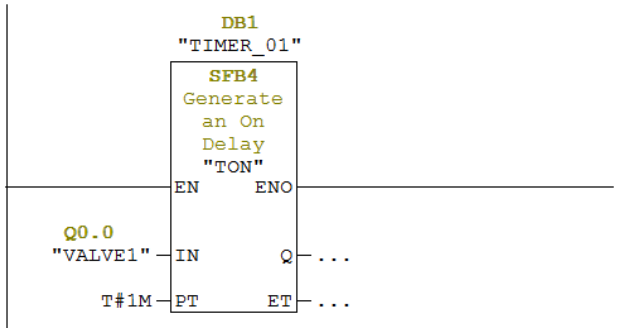
Network 1 : Title:

Comment:



Network 2 : Title:

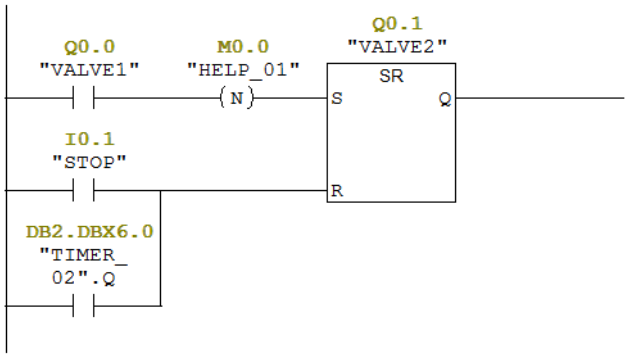
Comment:



ادامه

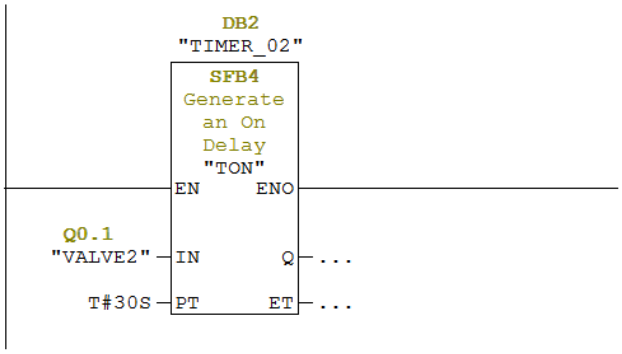
Network 3 : Title:

Comment:



Network 4 : Title:

Comment:



در این برنامه از آدرس دهی سمبلیک برای دیتابلاک ها استفاده شده است.

نکته ۱ : جهت دانلود این برنامه، حتما به این نکته توجه داشته باشید که بلوک های DB اختصاص داده شده به SFB4 نیز می بایست به CPU دانلود شوند.

نکته ۲ : جهت مانیتور کردن زمان تایمرها می توان از بلوک DB مربوط به هر SFB استفاده نمود. همچنین در جدول VAT نیز امکان مانیتور نمودن زمان وجود دارد.

نکته ۳ : زمانی که بلوک های سیستمی(SFB) به طور مستقیم از کتابخانه نرم افزار فراخوانی شوند، در هر بار فراخوانی نیاز به یک DB دارند.

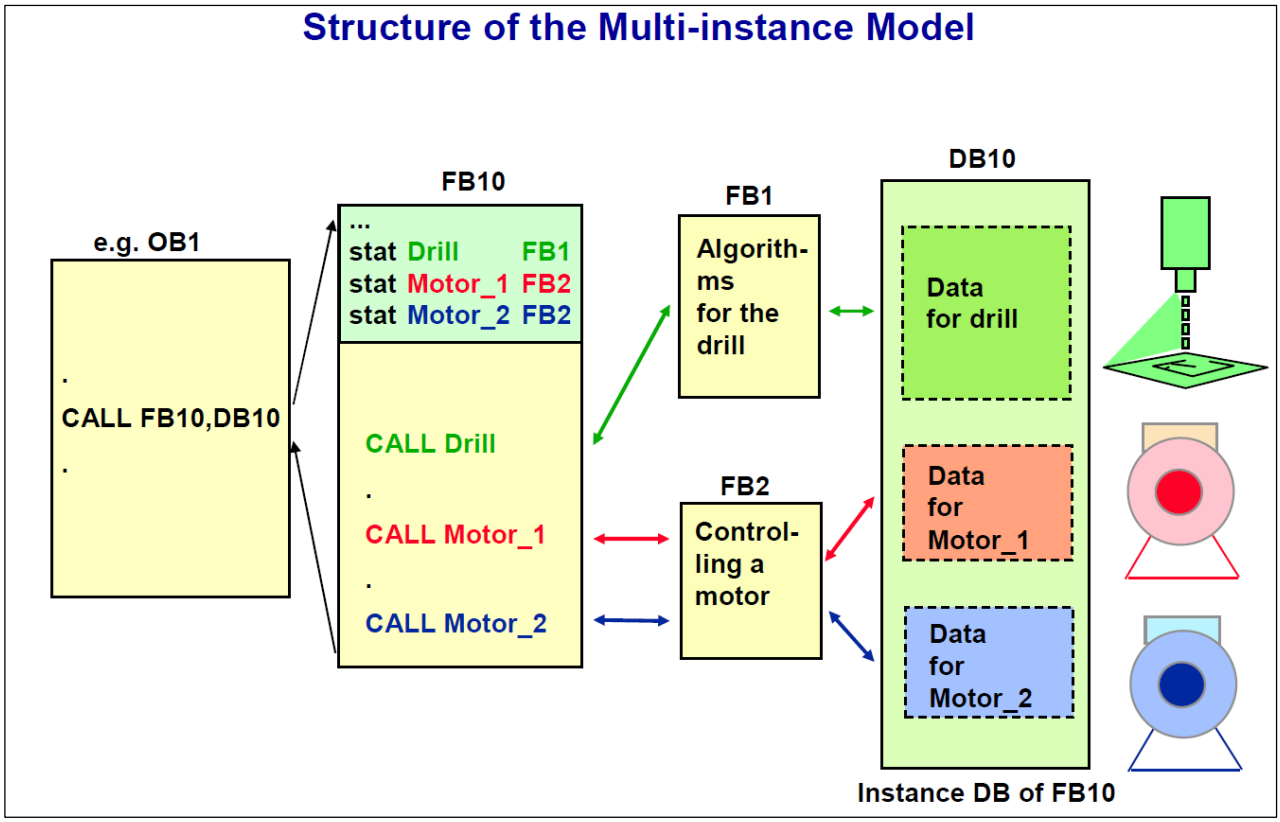
Novin Fanavaran Aria Co

استفاده از بلوک های سیستمی به صورت Multi-Instance

با استفاده از روش Multi-Instance، زمانی که یک FB یا SFB فراخوانی می شود، نیازی به اختصاص DB در هر بار فراخوانی ندارد. برای پیاده سازی این روش لازم است که توابع SFB یا FB در حافظه یک FB دیگر(STAT) تعریف یا ایجاد شوند. در ادامه، فراخوانی از بخش Multiple Instances صورت می گیرد. در این حالت تابع فراخوانی شده نیاز به DB ندارد. اختصاص DB در زمان فراخوانی FB بالادستی در OB1 یا بلوک دیگر انجام می شود.

همانطور که در شکل زیر ملاحظه می کنید، به 3 تابع Drill، Motor_1 و Motor_2 تنها یک دیتابلاک(DB10) اختصاص داده شده است. FB10 در این مثال به عنوان FB بالادستی یا مدیریتی می باشد که نهایتا در OB1 جهت اختصاص دیتابلاک(DB10) فراخوانی شده است.

Novin Farnavaran Aria Co.



مثال۲

در این مثال قصد داریم توسط روش Multi-Instances یک تابع تولید پالس طراحی کنیم به گونه ای که این تابع قابل استفاده برای زمان ها و آدرس های مختلف باشد. در این مثال از SFB4 که یک تایمر تاخیر در وصل(TON) می باشد، جهت ساختن تابع تولید پالس استفاده می کنیم.

برای اینکه بتوان از SFB4 در چندین تابع با زمانهای مختلف استفاده نمود به گونه ای که بین زمان ها تداخلی پیش نیاید، حتما می بایست تایمرها SFB4 را در قسمت STAT بلوک حافظه دار FB وارد و جهت فراخوانی آن نیز از قسمت Multiple Instances استفاده نمود. همانطور که بیان شد، زمانی که از روش Multi- Instances جهت فراخوانی یک SFB استفاده شود، SFB فراخوانی شده نیازی به DB اختصاصی نداشته و DB در زمان فراخوانی FB اختصاص داده می شود.

Nowin Farayaran Arin Co

پارامترهای IN

Contents Of: 'Environment\Interface\IN'

Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
ENABLE	Bool	0.0	FALSE			
TV_ON	Time	2.0	T#0MS			
TV_OFF	Time	6.0	T#0MS			

پارامترهای OUT

Contents Of: 'Environment\Interface\OUT'

Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
OUT	Bool	10.0	FALSE			

پارامترهای STAT

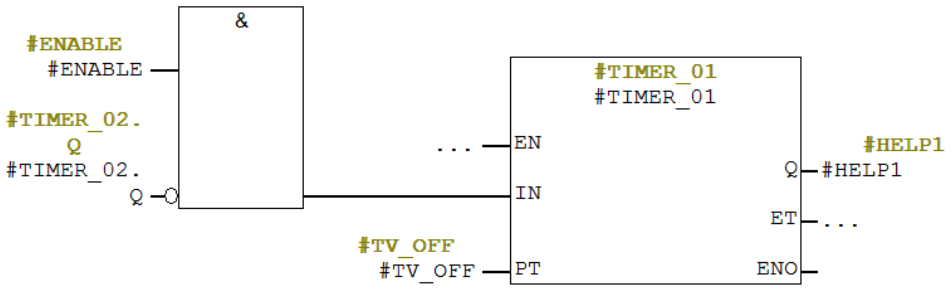
Contents Of: 'Environment\Interface\STAT'

Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
TIMER_01	TON	12.0				
TIMER_02	TON	34.0				
HELP1	Bool	56.0	FALSE			

برنامه FB1

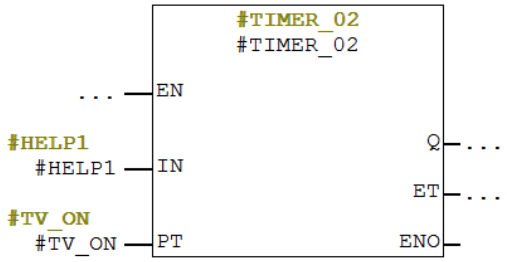
Network 1 : Title:

Comment:



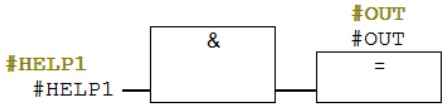
Network 2 : Title:

Comment:



Network 3 : Title:

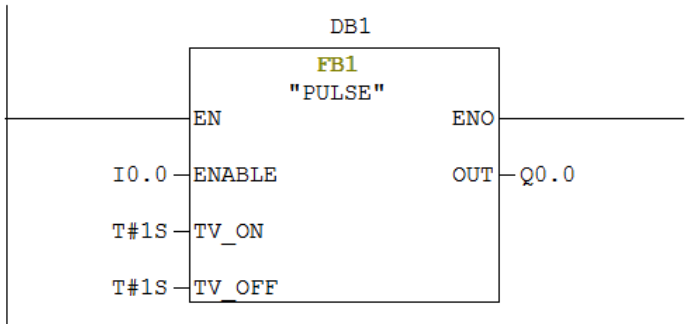
Comment:



فراخوانی FB1 در OB1

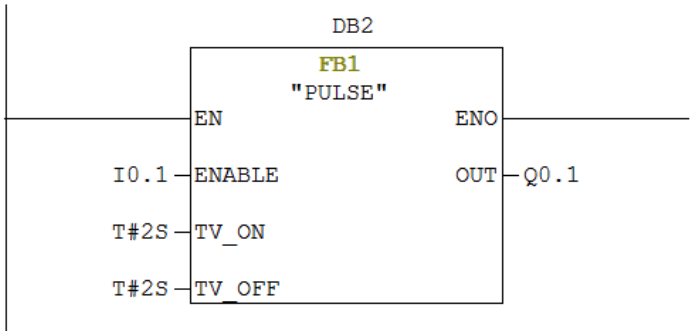
Network 1: Title:

Comment:



Network 2: Title:

Comment:



در این مثال، فانکشن SFB4 را با استفاده از روش Multi- Instances فراخوانی کردیم. نکته مهم، بحث فراخوانی FB1 در OB1 می باشد. همانطور که ملاحظه می کنید، با هر بار فراخوانی FB1 برای پالس های مختلف، یک دیتابلاک می بایست اختصاص داده شود. جهت جلوگیری از اختصاص دیتابلاک های متعدد در زمان فراخوانی، می توان از یک FB بالادستی (FB2) و روش Multi- Instances استفاده نمود. پس در ادامه، برنامه مثال ۲ را به صورت بهینه تر بازنویسی می کنیم.

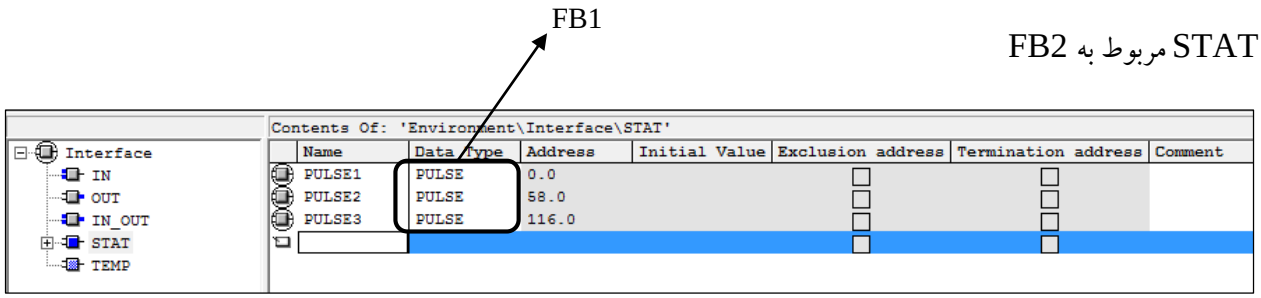
Akbar Oveisifar

مثال ۳

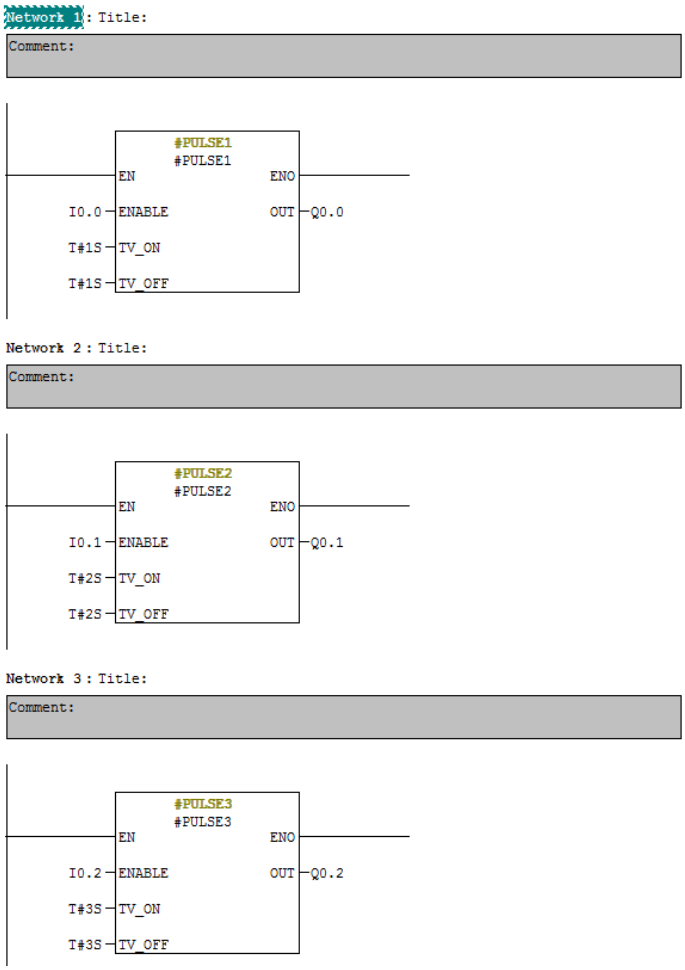
بازنویسی مثال ۲

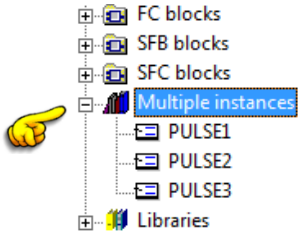
بلوک FB1 را مطابق مثال ۲ تکمیل می کنیم. در ادامه FB2 را ایجاد و FB1 را با نام های متفاوت در قسمت STAT ایجاد می کنیم.

STAT مربوط به FB2



در ادامه از قسمت Multiple Instances، سه تابع پالس را فراخوانی می کنیم.





در ادامه بلوک FB2 را در OB1 فراخوانی می کنیم.

```
Network 1 : Title:
Comment:

                    DB1
                    FB2
              "Management block"
... —EN                      ENO—
```

همانطور که مشاهده می کنید، تنها بلوک DB1 در این مثال مورد استفاده قرار گرفت. در DB1، پارامترهای مربوط به هر 3 تابع پالس در دسترس می باشند.

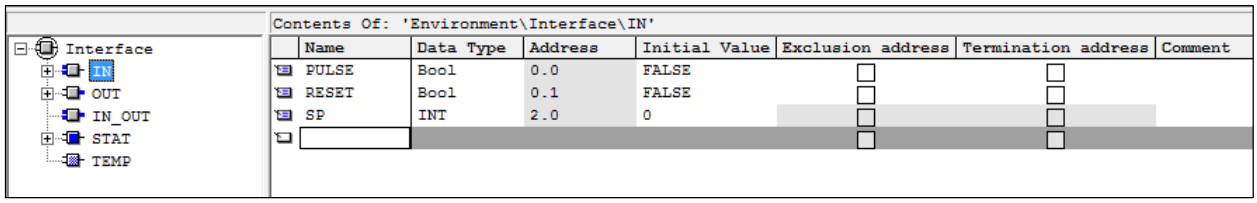
Data block Edit PLC Debug View Window Help

	Address	Declaration	Name	Type	Initial value	Actual valu
1	0.0	stat:in	PULSE1.ENABLE	BOOL	FALSE	FALSE
2	2.0	stat:in	PULSE1.TV_ON	TIME	T#0MS	T#0MS
3	6.0	stat:in	PULSE1.TV_OFF	TIME	T#0MS	T#0MS
4	10.0	stat:out	PULSE1.OUT	BOOL	FALSE	FALSE
5	12.0	stat:in	PULSE1.TIMER_01.IN	BOOL	FALSE	FALSE
6	14.0	stat:in	PULSE1.TIMER_01.PT	TIME	T#0MS	T#0MS
7	18.0	stat:out	PULSE1.TIMER_01.Q	BOOL	FALSE	FALSE
8	20.0	stat:out	PULSE1.TIMER_01.ET	TIME	T#0MS	T#0MS
9	24.0	stat	PULSE1.TIMER_01.STATE	BYTE	B#16#0	B#16#0
10	26.0	stat	PULSE1.TIMER_01.STIME	TIME	T#0MS	T#0MS
11	30.0	stat	PULSE1.TIMER_01.ATIME	TIME	T#0MS	T#0MS
12	34.0	stat:in	PULSE1.TIMER_02.IN	BOOL	FALSE	FALSE
13	36.0	stat:in	PULSE1.TIMER_02.PT	TIME	T#0MS	T#0MS
14	40.0	stat:out	PULSE1.TIMER_02.Q	BOOL	FALSE	FALSE
15	42.0	stat:out	PULSE1.TIMER_02.ET	TIME	T#0MS	T#0MS
16	46.0	stat	PULSE1.TIMER_02.STATE	BYTE	B#16#0	B#16#0
17	48.0	stat	PULSE1.TIMER_02.STIME	TIME	T#0MS	T#0MS
18	52.0	stat	PULSE1.TIMER_02.ATIME	TIME	T#0MS	T#0MS
19	56.0	stat	PULSE1.HELP1	BOOL	FALSE	FALSE
20	58.0	stat:in	PULSE2.ENABLE	BOOL	FALSE	FALSE
21	60.0	stat:in	PULSE2.TV_ON	TIME	T#0MS	T#0MS
22	64.0	stat:in	PULSE2.TV_OFF	TIME	T#0MS	T#0MS
23	68.0	stat:out	PULSE2.OUT	BOOL	FALSE	FALSE
24	70.0	stat:in	PULSE2.TIMER_01.IN	BOOL	FALSE	FALSE
25	72.0	stat:in	PULSE2.TIMER_01.PT	TIME	T#0MS	T#0MS
26	76.0	stat:out	PULSE2.TIMER_01.Q	BOOL	FALSE	FALSE
27	78.0	stat:out	PULSE2.TIMER_01.ET	TIME	T#0MS	T#0MS
28	82.0	stat	PULSE2.TIMER_01.STATE	BYTE	B#16#0	B#16#0
29	84.0	stat	PULSE2.TIMER_01.STIME	TIME	T#0MS	T#0MS
30	88.0	stat	PULSE2.TIMER_01.ATIME	TIME	T#0MS	T#0MS
31	92.0	stat:in	PULSE2.TIMER_02.IN	BOOL	FALSE	FALSE
32	94.0	stat:in	PULSE2.TIMER_02.PT	TIME	T#0MS	T#0MS
33	98.0	stat:out	PULSE2.TIMER_02.Q	BOOL	FALSE	FALSE
34	100.0	stat:out	PULSE2.TIMER_02.ET	TIME	T#0MS	T#0MS
35	104.0	stat	PULSE2.TIMER_02.STATE	BYTE	B#16#0	B#16#0
36	106.0	stat	PULSE2.TIMER_02.STIME	TIME	T#0MS	T#0MS
37	110.0	stat	PULSE2.TIMER_02.ATIME	TIME	T#0MS	T#0MS
38	114.0	stat	PULSE2.HELP1	BOOL	FALSE	FALSE
39	116.0	stat:in	PULSE3.ENABLE	BOOL	FALSE	FALSE
40	118.0	stat:in	PULSE3.TV_ON	TIME	T#0MS	T#0MS
41	122.0	stat:in	PULSE3.TV_OFF	TIME	T#0MS	T#0MS
42	126.0	stat:out	PULSE3.OUT	BOOL	FALSE	FALSE
43	128.0	stat:in	PULSE3.TIMER_01.IN	BOOL	FALSE	FALSE
44	130.0	stat:in	PULSE3.TIMER_01.PT	TIME	T#0MS	T#0MS
45	134.0	stat:out	PULSE3.TIMER_01.Q	BOOL	FALSE	FALSE
46	136.0	stat:out	PULSE3.TIMER_01.ET	TIME	T#0MS	T#0MS
47	140.0	stat	PULSE3.TIMER_01.STATE	BYTE	B#16#0	B#16#0
48	142.0	stat	PULSE3.TIMER_01.STIME	TIME	T#0MS	T#0MS
49	146.0	stat	PULSE3.TIMER_01.ATIME	TIME	T#0MS	T#0MS

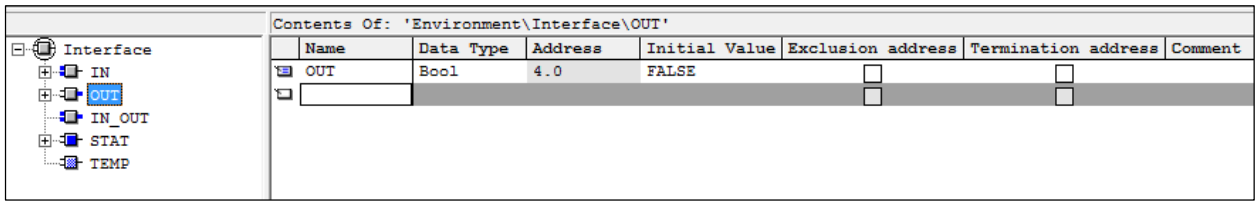
مثال ٤

در این مثال می خواهیم از شمارنده SFB0 جهت شمارش سه پالس با PV های مختلف استفاده کنیم.

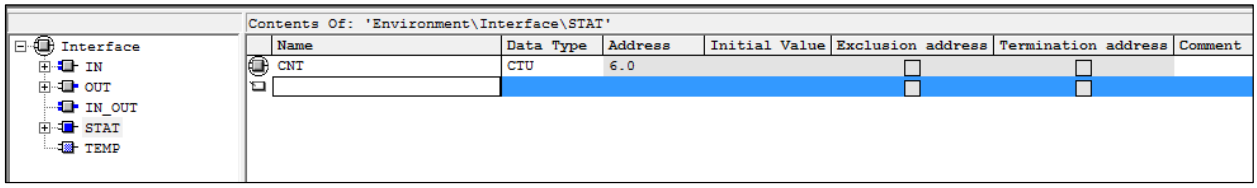
پارامترهای IN



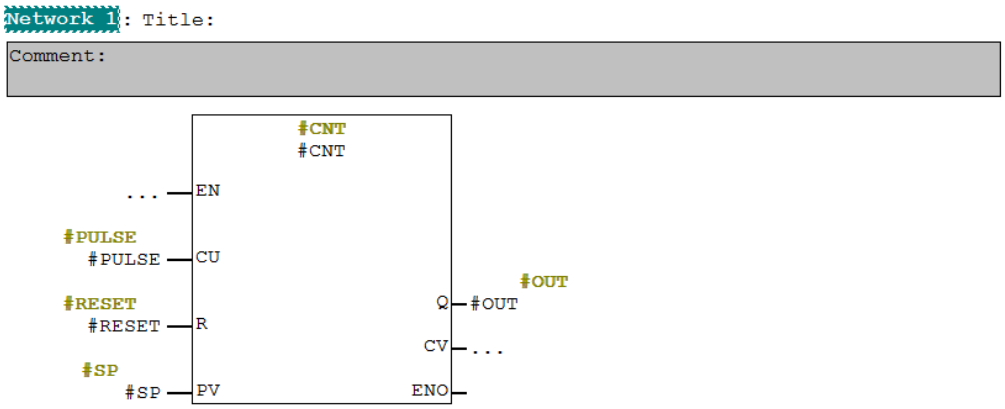
پارامترهای OUT



پارامترهای STAT



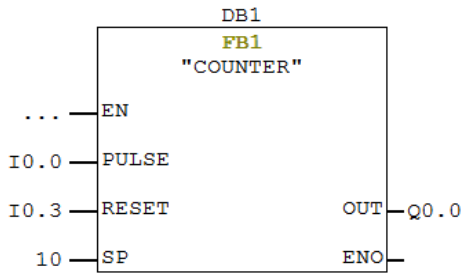
برنامه FB1



فراخوانی بلوک FB1 در OB1

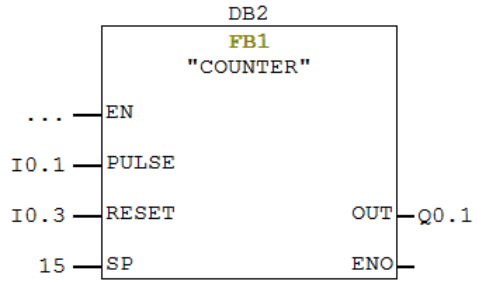
Network 1 : Title:

Comment:



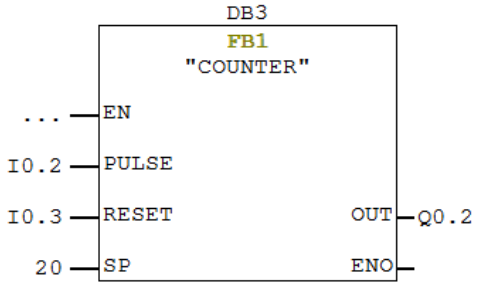
Network 2 : Title:

Comment:



Network 3 : Title:

Comment:



این مثال را با استفاده از یک FB مدیریتی، جهت کاهش تعداد DB ها بازنویسی کنید.

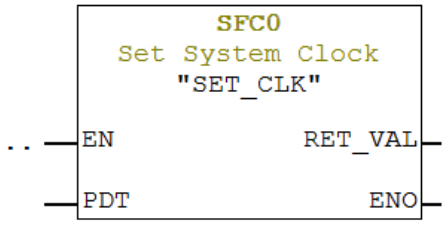
فانکشن های مربوط به تاریخ و زمان CPU

توسط این فانکشن ها می توان ساعت و تاریخ CPU را خواند و یا در صورت نیاز تنظیم نمود. در بسیاری از پروژه ها لازم است که فرمان هایی در زمان های مشخص توسط سیستم کنترل صادر شود. در این صورت برنامه نویس می تواند از زمان و تاریخ CPU در برنامه استفاده کند.

Akbar Oveisifar

SFC0 (SET-CLK)

توسط این فانکشن می توان ساعت و تاریخ CPU را تنظیم نمود.



ورودی PDT این فانکشن همان ورودی است که توسط آن می توان ساعت و تاریخ را برای CPU تعریف نمود. اما نکته قابل توجه این است که به ورودی PDT نمی توان مستقیما زمان و تاریخ را اعمال نمود. این ورودی دارای فرمتی بصورت Data –and-Time می باشد.

The Data Type: DATE_AND_TIME

Structure:

Byte n[1]	Year (90 ... 89)	Month (01 ... 12)	Byte n+1
Byte n+2	Day (01 ... 31)	Hour (00 ... 23)	Byte n+3
Byte n+4	Minute (00 ... 59)	Second (00 ... 59)	Byte n+5
Byte n+6	Milliseconds (000 ... 999)	Week-day (1..7)	Byte n+7

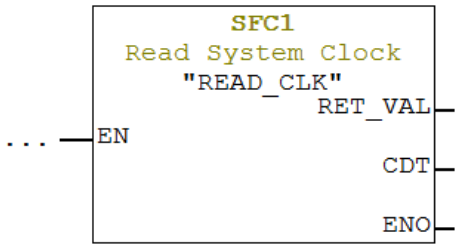
1=Sunday
2=Monday
3=Tuesday
4=Wednesday
5=Thursday
6=Friday
7=Saturday

در شکل زیر تایپ داده های پایه مختلف را در STEP7 ملاحظه می کنید.

Keyword	Length (in bits)	Example of a constant of this type
BOOL	1	1 or 0
BYTE	8	B#16#A9
WORD	16	W#16#12AF
DWORD	32	DW#16#ADAC1EF5
CHAR	8	' w '
S5TIME	16	S5T#5s_200ms
INT	16	123
DINT	32	65539
REAL	32	1.2 or 34.5E-12
TIME	32	T#2D_1H_3M_45S_12MS
DATE	16	D#1993-01-20
TIME_OF_DAY	32	TOD#12:23:45.12

SFC1 (Read-CLK)

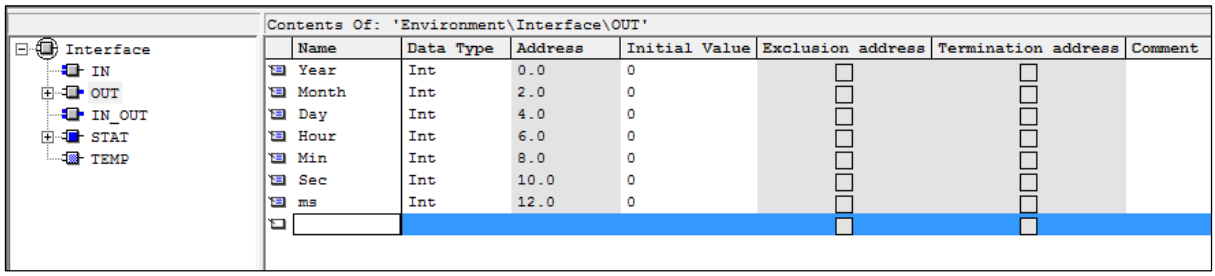
توسط این فانکشن این امکان وجود دارد که ساعت CPU را خواند و در برنامه کنترلی از آن استفاده نمود.



مثال:

در این مثال می خواهیم تاریخ و زمان CPU را خوانده و به صورت Integer نمایش دهیم.

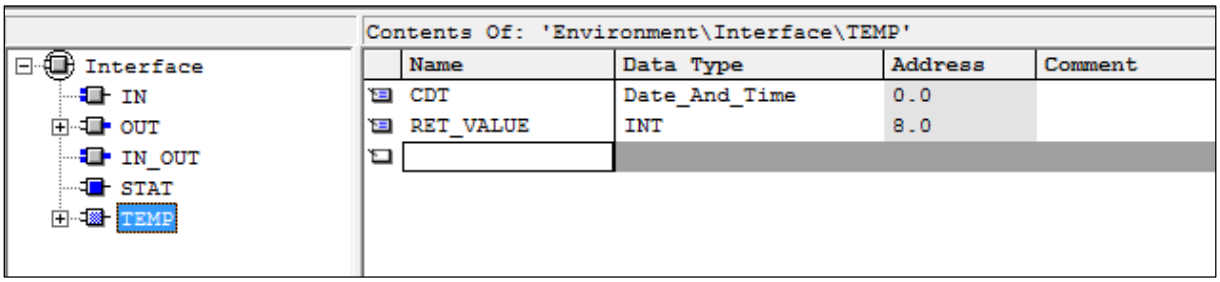
پارامترهای OUT



Contents Of: 'Environment\Interface\OUT'

Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
Year	Int	0.0	0			
Month	Int	2.0	0			
Day	Int	4.0	0			
Hour	Int	6.0	0			
Min	Int	8.0	0			
Sec	Int	10.0	0			
ms	Int	12.0	0			

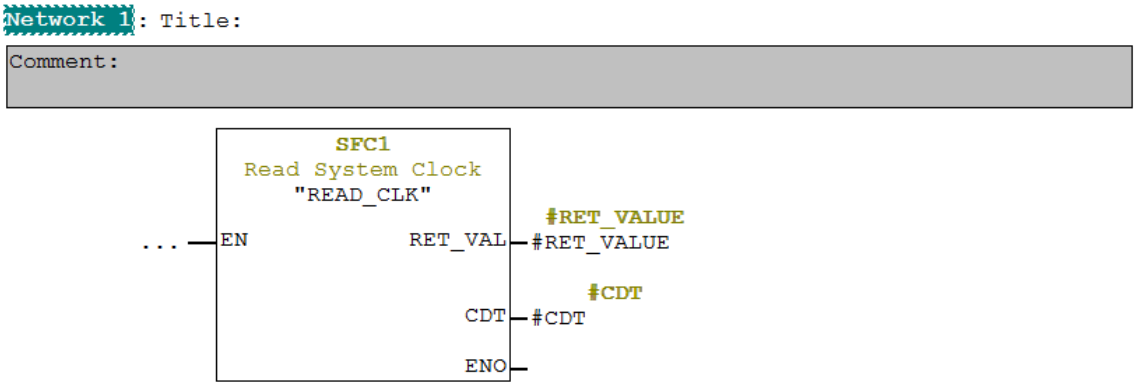
پارامترهای TEMP



Contents Of: 'Environment\Interface\TEMP'

Name	Data Type	Address	Comment
CDT	Date_And_Time	0.0	
RET_VALUE	INT	8.0	

برنامه FB1



ادامه

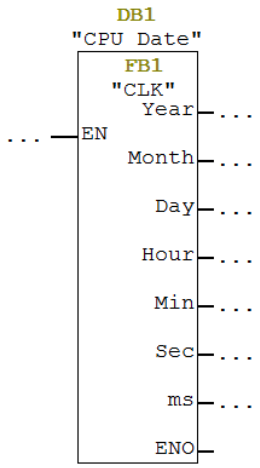
```
Network 2: Title:
Comment:

      L     LB     0
      BTI
      T     #Year                    #Year
      L     LB     1
      BTI
      T     #Month                   #Month
      L     LB     2
      BTI
      T     #Day                     #Day
      L     LB     3
      BTI
      T     #Hour                    #Hour
      L     LB     4
      BTI
      T     #Min                     #Min
      L     LB     5
      BTI
      T     #Sec                     #Sec
      L     LB     6
      BTI
      T     #ms                      #ms
```

جهت سهولت استفاده از خروجی های زمان و تاریخ در بخش های مختلف برنامه، توسط برنامه Network 2 و با استفاده از دستور تبدیل(BTI)، مقادیر Hex به Integer تبدیل شده است.

فراخوانی FB1 در OB1

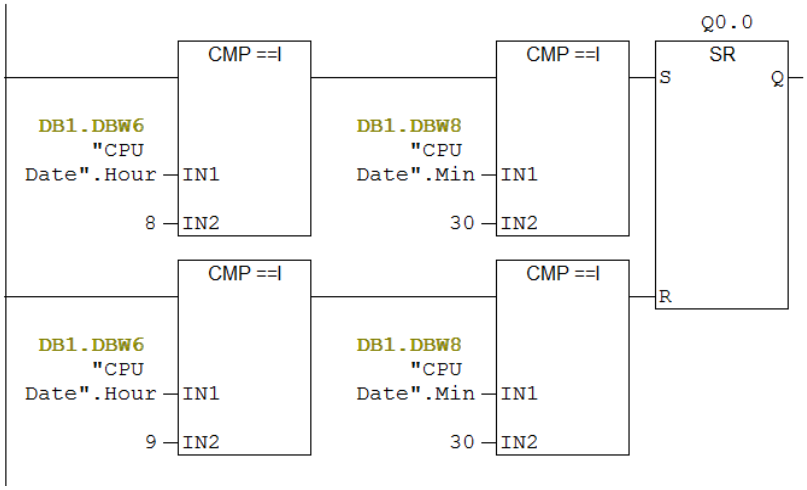
```
Network 1: Title:
Comment:
```



اگر در برنامه ای نیاز به روشن یا خاموش کردن یک مصرف کننده در تاریخ و زمان خاصی باشد، به راحتی می توان با استفاده از دستورات مقایسه، فرامین مختلف را به خروجی ارسال نمود.

Network 2: Title:

Comment:



در مثال فوق، هر روز از ساعت 8.30 الی 9.30 ، مصرف کننده متصل شده به Q0.0 روشن می باشد.

فانکشن های مربوط به شمارشگرهای ساعت در CPU(RTM)

توسط RTM ها می توان بر حسب ساعت، مدت زمان کارکرد یک مصرف کننده را ثبت نمود. از ویژگی های RTM ها می توان به حفظ مقدار قبلی در زمان Cold Restart اشاره نمود. تعداد RTM یکی از مشخصات تعریف شده در هر CPU می باشد.

انواع RTM:

RTM شانزده بیتی: ماکزیمم زمان 32767 ساعت

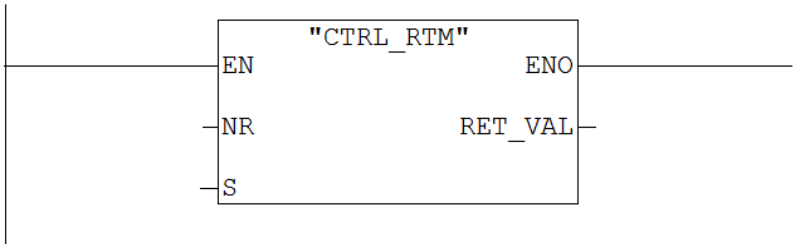
RTM سی و دو بیتی: ماکزیمم زمان 2147483647 ساعت

برخی از CPU ها تنها RTM شانزده بیتی را ساپورت می کنند.

Novin Fanavaran Arin Co

SFC3 (CTRL-RTM)

توسط این فانکشن می توان مدت زمان کارکرد یک دستگاه را بر حسب ساعت اندازه گیری نمود. فرض کنید در یک پروسه صنعتی، مدت زمان کارکرد یک مصرف کننده برای انجام عملیات تعمیرات و نگهداری بسیار مهم می باشد. همچنین این زمان سنج می بایست در مدتی که مصرف کننده خاموش می باشد، زمان را محاسبه نکند. از SFC3 جهت راه اندازی و توقف RTM استفاده می شود.



برنامه نویس حداکثر می تواند از 8 زمان سنج داخلی (RTM) در برنامه استفاده کند. در ضمن SFC3 برای کنترل RTM های ۱۶ بیتی استفاده می شود.

ورودی NR : ورودی مربوط به شماره RTM بوده که به فرم B#16#X قابل آدرس دهی می باشد. حرف X شماره RTM است که می تواند یک عدد بین 0 تا 7 اختیار کند.

ورودیS : این ورودی در واقع آدرسی از خروجی که می خواهیم مدت زمان کارکردش را ثبت کنیم می باشد. به عنوان مثال فرض کنید که در این ورودی آدرس Q5.0 وارد شده است. در این حالت مدت زمان کارکرد مصرف کننده ای که به خروجی Q5.0 متصل شده است ثبت می گردد.

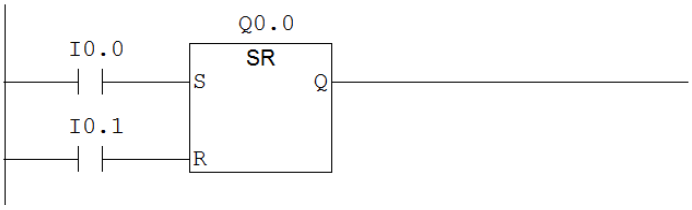
خروجی RET-VAL : خروجی مربوط به خطاهای برگشتی بوده که در محدوده WORD می باشد.

مثال

فرض کنید در یک دستگاه صنعتی یک موتور با توان بالا استفاده شده است. مدت زمان کارکرد این موتور همواره برای انجام عملیات تعمیرات و نگهداری مهم می باشد. این موتور توسط شستی I0.0 روشن و توسط شستی I0.1 خاموش می گردد. می خواهیم برنامه ای طراحی کنیم که مدت زمان روشن بودن این موتور بر حسب ساعت اندازه گیری شود.

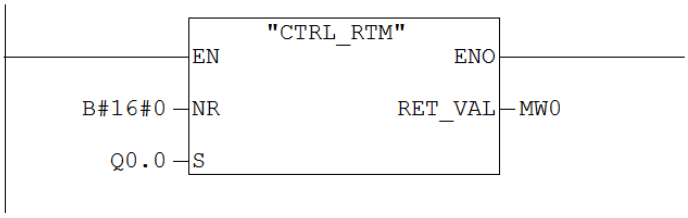
Network 1 : Title:

Comment:



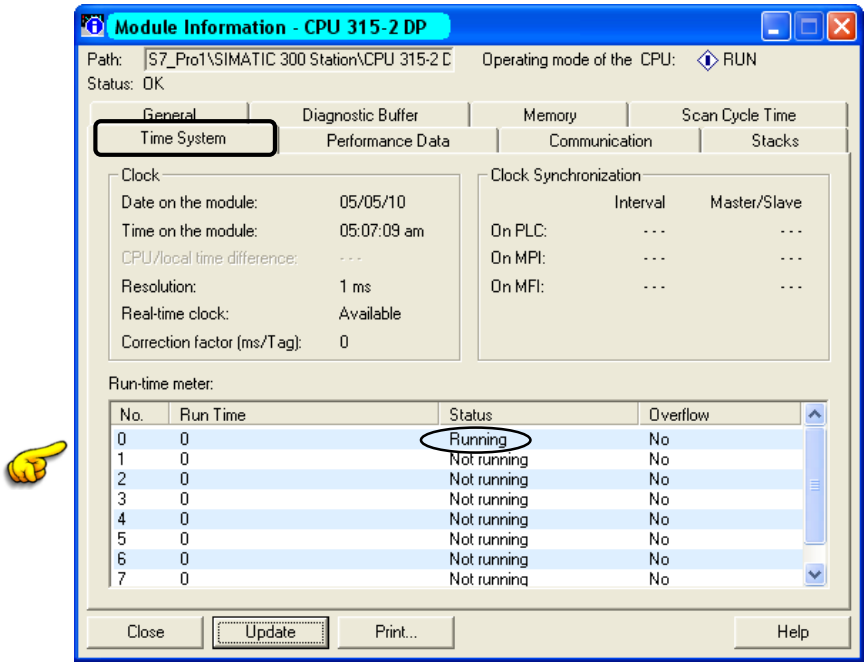
Network 2 : Title:

Comment:



Novin Farayavaran Asia co.

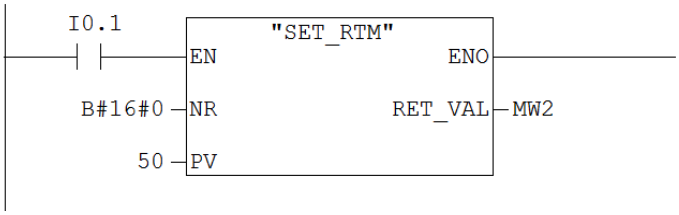
در نرم افزار جهت مشاهده وضعیت RTM ها و مشاهده ثبت کارکرد، در محیط نرم افزار به منوی PLC و گزینه Module Information مراجعه کنید. در پنجره نمایان شده، زبانه Time System را فعال نمایید.



در ادامه فرض کنید می خواهیم برای SFC3 یک مقدار اولیه در نظر بگیریم. در این صورت از فانکشن SFC2 استفاده می شود.SFC2 دارای یک ورودی با نام PV می باشد که توسط این ورودی می توان مدت زمانی که بر حسب ساعت می باشد را به عنوان مقدار اولیه برای فانکشن SFC3 تعیین نمود.

Network 3: Title:

Comment:



در برنامه شکل فوق، با فعال شدن ورودی I0.1 ، این فانکشن مدت زمان50 ساعت را برای RTM شماره 0 لود می کند.

در بسیاری از کاربردها نیاز می باشد که مدت زمان کارکرد دستگاه بر روی HMI قابل نمایش باشد. در این صورت حتما می بایست به مقادیر RTM ها دسترسی پیدا نمود. توسط فانکشن SFC4 می توان از فعال بودن و همچنین مقدار RTM مطلع شد.

ورودی NR : آدرس RTM که قصد خواندن مقدار آن را داریم می باشد.

خروجی CQ : خروجی مربوط به فعال بودن RTM بوده که به صورت بیتی آدرس دهی می شود.

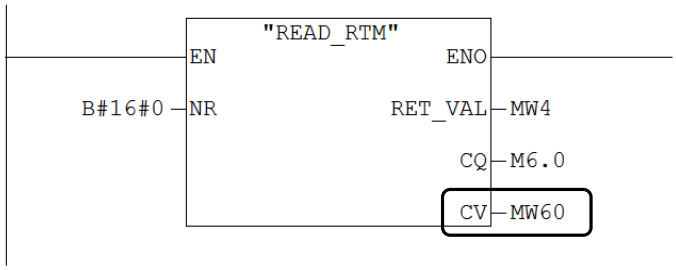
خروجی CV : مقدار جاری RTM بوده که به فرم Integer می باشد.

فرض کنید در صفحه HMI آدرس MW60 به ابزار I/O Field جهت نمایش، متصل شده است. در این حالت برنامه نویس می بایست مقدار RTM 0 را به MW60 منتقل کند.

Novin Fanavaran Aria Co.

Network 4 : Title:

Comment:



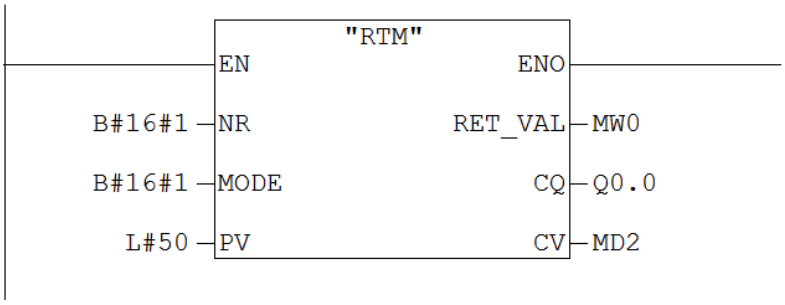
زمانی که RTM 0 فعال باشد، بیت M6.0 نیز به عنوان یک آلارم که نشانه فعال بودن RTM است فعال می شود.

SFC101(RTM)

RTM های بررسی شده در قسمت قبل، 16 بیتی بودند و حداکثر می توانستند تا 32767 ساعت را شمارش کنند. همچنین فانکشن های مربوط به فعال و یا غیر فعال نمودن RTM ها و همچنین خواندن مقدار RTM بصورت جداگانه در دسترس بودند.

SFC101 یک زمان سنج داخلی 32 بیتی بوده که در کاربردهایی که نیاز به ثبت ساعت بیشتری برای یک دستگاه می باشد به کار برده می شود. این RTM می تواند دارای شماره ای بین 0 تا 15 باشد. در ضمن تمامی فانکشن هایی که در قسمت های قبلی برای کاربردهای مختلف بررسی گردید، در این فانکشن تعبیه شده است. این RTM ها توسط تمامی CPU ها ساپورت نمی شوند.

Novin Fanavaran Aria Co.



ورودی NR : شماره RTM

ورودی Mode : این ورودی می تواند یک عدد بین 0 تا 6 باشد.

Mode=0 : در این حالت وضعیت RTM به خروجی های CQ و CV منتقل می گردد و زمانی که مقدار CV به مقدار ماکزیمم رسید متوقف می شود. مقدار ماکزیمم CV عدد 1-(2E31) ساعت می باشد.

MODE=1 : استارت RTM

MODE=2 : توقف RTM

MODE=4 : با مقدار اولیه PV ست می شود.

MODE=5 : با مقدار اولیه PV ست شده و سپس شروع به کار می کند.

MODE=6 : با مقدار اولیه PV ست شده و سپس متوقف می شود.

ورودی PV : مقدار اولیه RTM بوده که بصورت ۳۲ بیتی صحیح می باشد.

خروجی CQ : خروجی مربوط به فعال بودن RTM بوده که به صورت بیتی آدرس دهی می شود.

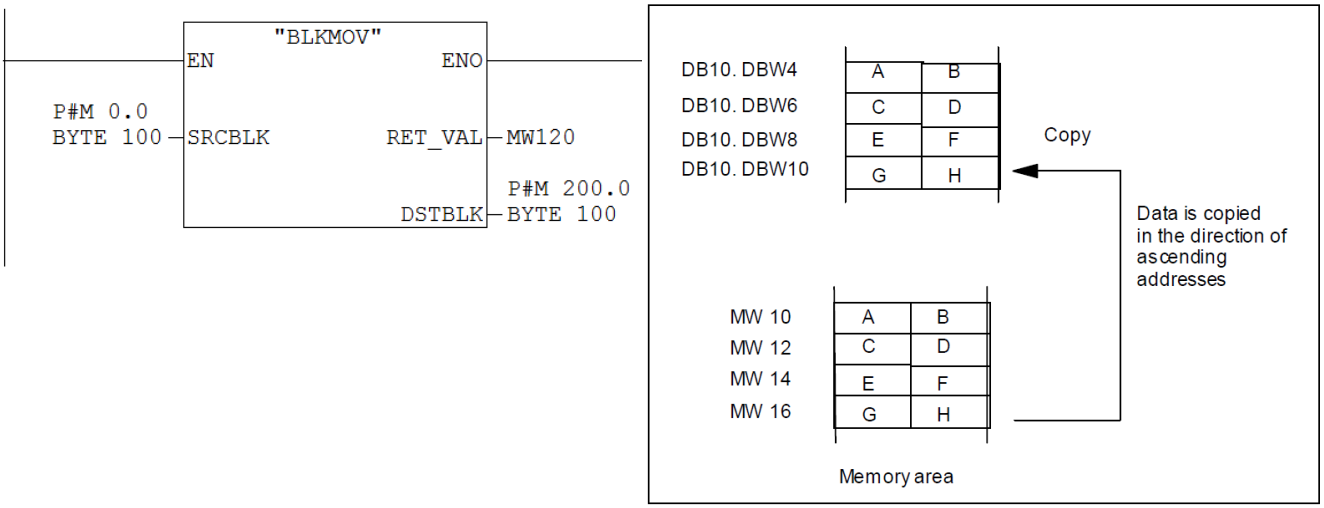
خروجی CV : مقدار جاری RTM بوده که به فرم DINT (۳۲ بیتی صحیح) می باشد.

Akbar Oveisifar

فانکشن های مربوط به انتقال گروهی

SFC20 (BLKMOVE)

توسط این فانکشن می توان یک بخش از حافظه را به بخش دیگر منتقل نمود. توسط دستور MOVE، تنها امکان انتقال ماکزیمم ۴ بایت وجود دارد. (به عنوان مثال انتقال MD0 به MD10). اما توسط این دستور می توان چندین بابت از حافظه را به بخش دیگر منتقل که در این انتقال آدرس مبدا و آدرس مقصد و تعداد آنها نیز در اختیار کاربر می باشد.



SRCBLK : آدرس شروع حافظه به همراه تعداد آن در مبدا

DSTBLK : آدرس شروع حافظه به همراه تعداد آن در مقصد

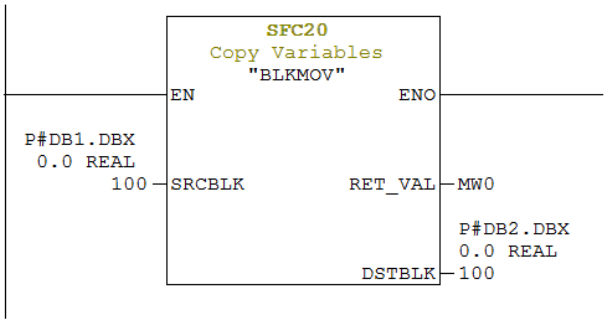
آدرس های مبدا و مقصد به صورت Pointer می باشند. این آدرس می تواند برای I ، Q و DB نیز مورد استفاده قرار گیرد. در مثال شکل فوق، محتوای MB0 تا MB99 به MB200 تا MB299 منتقل می شود. بهتر حجم فضای اختصاص داده شده به مبدا و مقصد یکسان باشند.

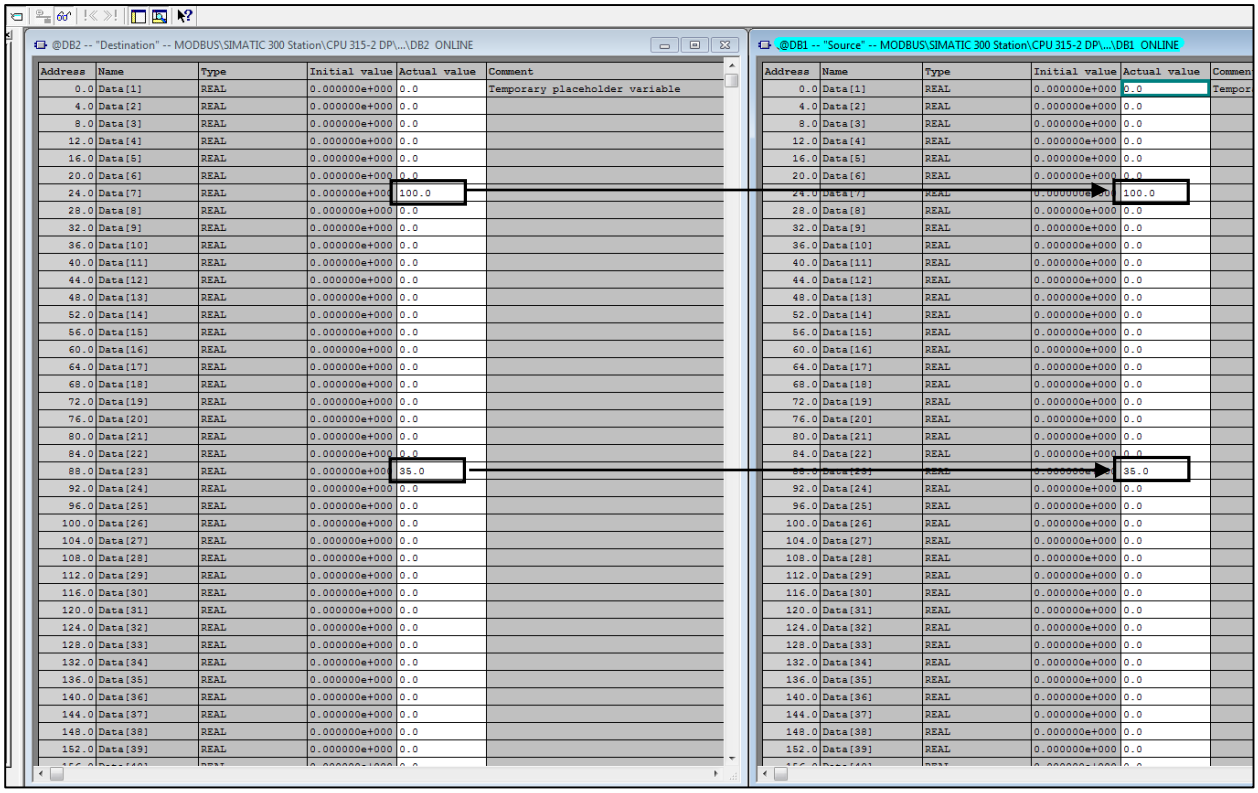
مثال

انتقال 100 سطر از DB1 به DB2

Network 1: Title:

Comment:

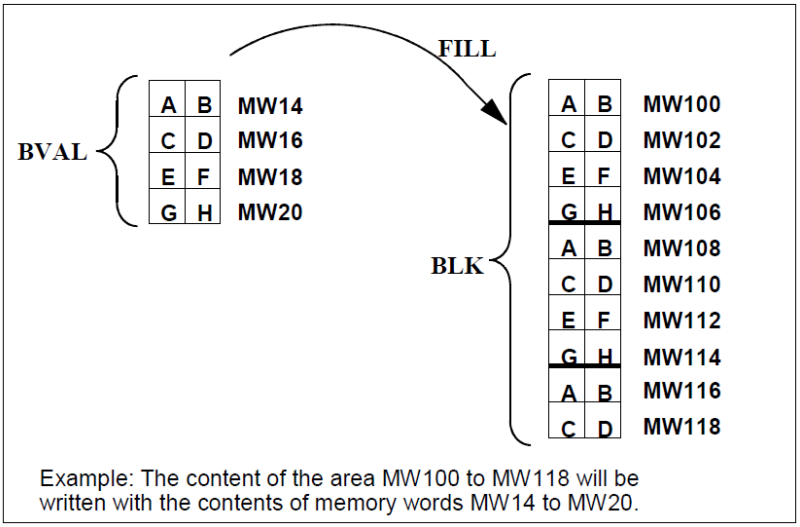




SFC21 (FILL)

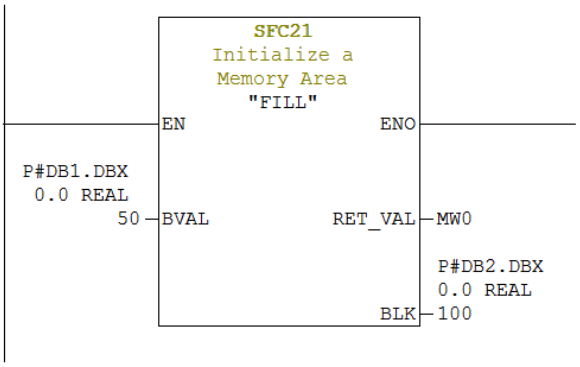
توسط این فانکشن می توان یک بخش از حافظه را با مقدار دلخواه پر و یا یک بخش از حافظه را به بخش دیگر منتقل نمود. نکته ای که در عملکرد این فانکشن می بایست دقت شود این است که اگر مقصد از مبدا بزرگتر باشد، عملیات کپی کردن تا مادامی که کل حافظه مقصد پر شود، تکرار می شود. در SFC20 عملیات انتقال فقط به اندازه حافظه مبدا در مقصد کپی می شود.

Akbar Oveisifar

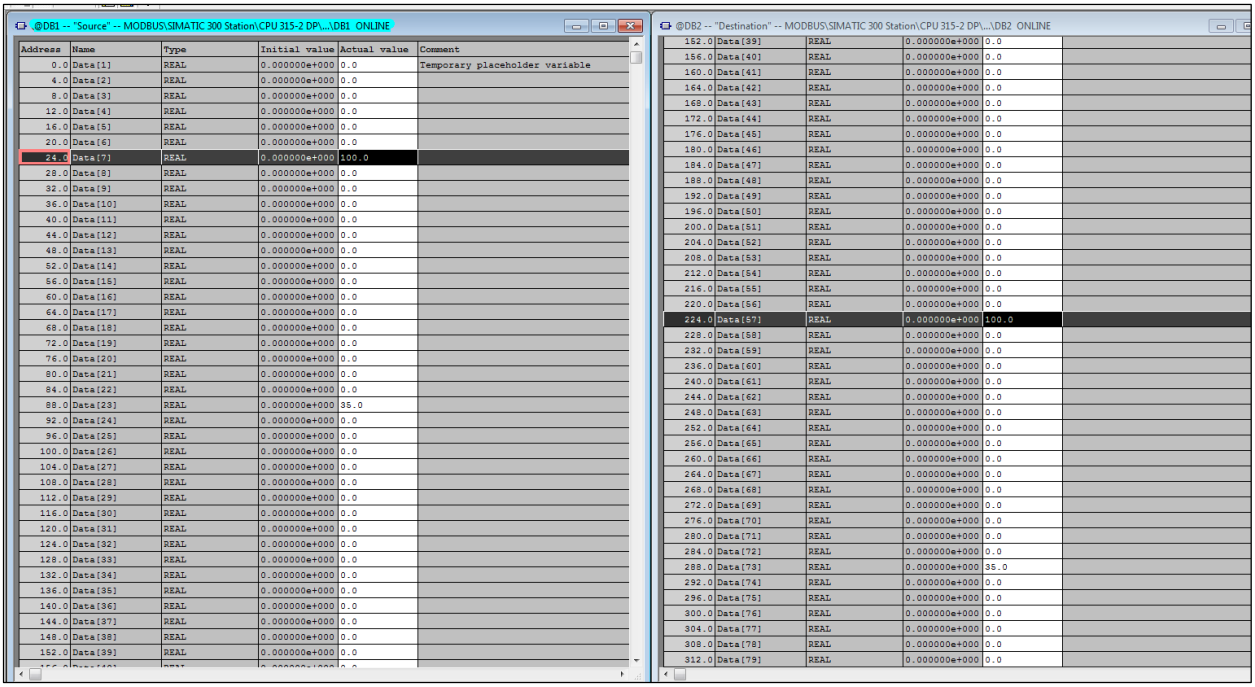


Network 1: Title:

Comment:

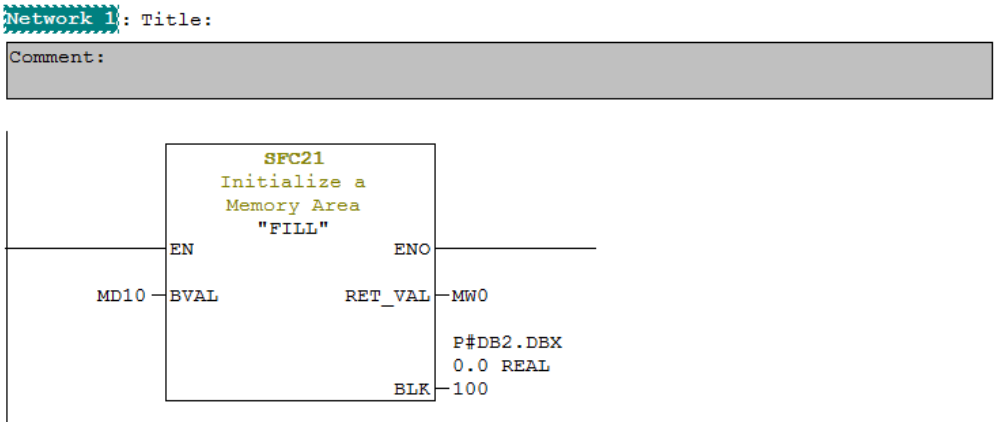


وضعیت دیتابلاک ها در وضعیت Online



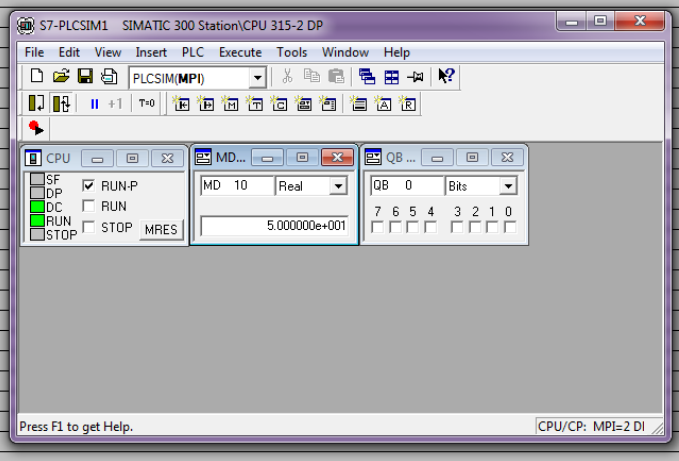
همانطور که در شکل فوق ملاحظه می کنید، مقدار سطر 7 از آرایه DB1 ، مجدد در سطر 57 از آرایه DB2 کپی شده است.

اگر نیاز باشد یک بخش از حافظه با مقدار ثابت پر شود، مقدار را می توان در یک حافظه ۴ بایتی منتقل و سپس توسط فانکشن FILL به بازه مدنظر از حافظه منتقل نمود.



وضعیت DB2 در حالت Online

Address	Name	Type	Initial value	Actual value	Comment
0.0	Data[1]	REAL	0.000000e+000	50.0	Temporary placeholder variable
4.0	Data[2]	REAL	0.000000e+000	50.0	
8.0	Data[3]	REAL	0.000000e+000	50.0	
12.0	Data[4]	REAL	0.000000e+000	50.0	
16.0	Data[5]	REAL	0.000000e+000	50.0	
20.0	Data[6]	REAL	0.000000e+000	50.0	
24.0	Data[7]	REAL	0.000000e+000	50.0	
28.0	Data[8]	REAL	0.000000e+000	50.0	
32.0	Data[9]	REAL	0.000000e+000	50.0	
36.0	Data[10]	REAL	0.000000e+000	50.0	
40.0	Data[11]	REAL	0.000000e+000	50.0	
44.0	Data[12]	REAL	0.000000e+000	50.0	
48.0	Data[13]	REAL	0.000000e+000	50.0	
52.0	Data[14]	REAL	0.000000e+000	50.0	
56.0	Data[15]	REAL	0.000000e+000	50.0	
60.0	Data[16]	REAL	0.000000e+000	50.0	
64.0	Data[17]	REAL	0.000000e+000	50.0	
68.0	Data[18]	REAL	0.000000e+000	50.0	
72.0	Data[19]	REAL	0.000000e+000	50.0	
76.0	Data[20]	REAL	0.000000e+000	50.0	
80.0	Data[21]	REAL	0.000000e+000	50.0	
84.0	Data[22]	REAL	0.000000e+000	50.0	
88.0	Data[23]	REAL	0.000000e+000	50.0	
92.0	Data[24]	REAL	0.000000e+000	50.0	
96.0	Data[25]	REAL	0.000000e+000	50.0	
100.0	Data[26]	REAL	0.000000e+000	50.0	
104.0	Data[27]	REAL	0.000000e+000	50.0	
108.0	Data[28]	REAL	0.000000e+000	50.0	
112.0	Data[29]	REAL	0.000000e+000	50.0	
116.0	Data[30]	REAL	0.000000e+000	50.0	
120.0	Data[31]	REAL	0.000000e+000	50.0	
124.0	Data[32]	REAL	0.000000e+000	50.0	
128.0	Data[33]	REAL	0.000000e+000	50.0	
132.0	Data[34]	REAL	0.000000e+000	50.0	
136.0	Data[35]	REAL	0.000000e+000	50.0	
140.0	Data[36]	REAL	0.000000e+000	50.0	
144.0	Data[37]	REAL	0.000000e+000	50.0	

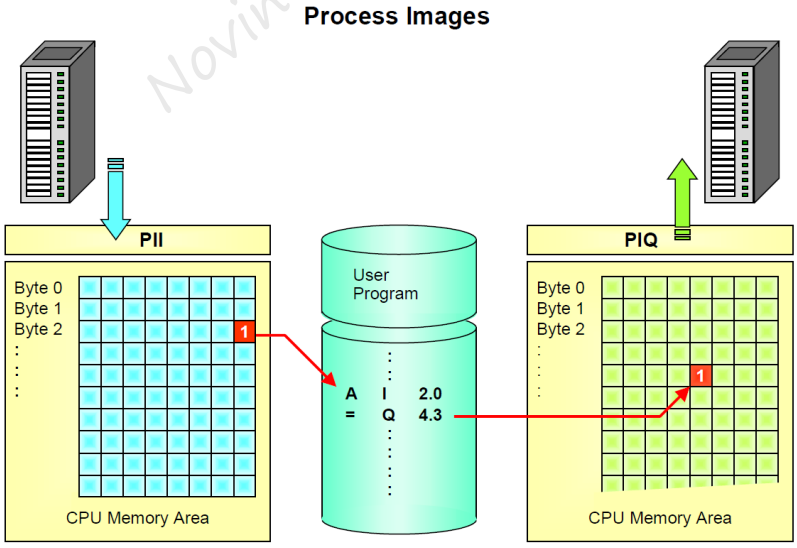


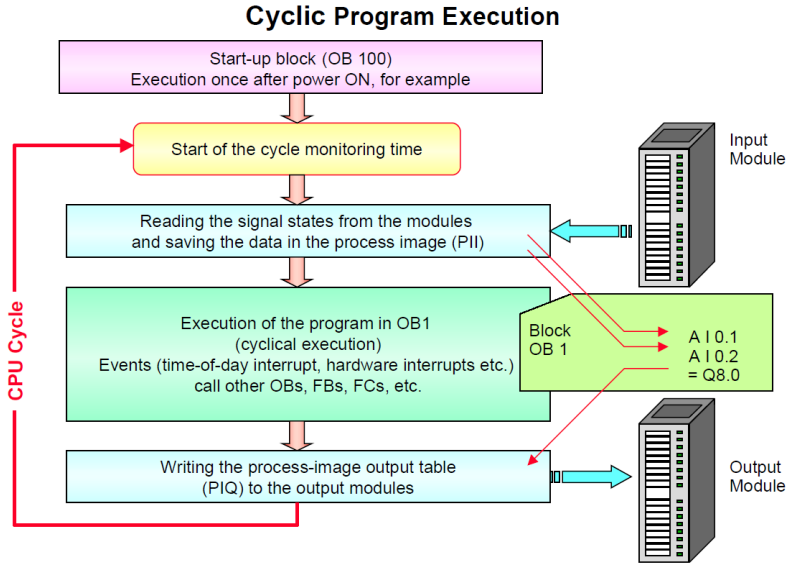
فانکشن های مربوط به ارسال فرمان مستقیم به یک یا چند بیت از خروجی

SFC79 (SET-BIT)

توسط این فانکشن می توان یک یا چند بیت از خروجی Peripheral در S7-400 را ست نمود. در واقع در حالت عادی توسط دستورات فیلیپ فلاپ و ... امکان فرمان دادن به خروجی در محدوده Peripheral به صورت بیتی وجود ندارد و حداقل می بایست یک بایت به صورت PQB استفاده شود. با استفاده از این فانکشن این مشکل برای دسترسی به صورت بیتی حل شده است. اگر به خروجی به صورت Peripheral دسترسی داشته باشیم، فرمان به صورت مستقیم به کارت سخت افزاری منتقل می شود. در حالت غیرمستقیم، فرمان ابتدا به حافظه PIQ منتقل شده و سپس در انتهای سیکل مقادیر PIQ به کارت سخت افزاری منتقل می شوند.

Novin Fanavaran Aria Co.





(RSET-BIT) SFC80

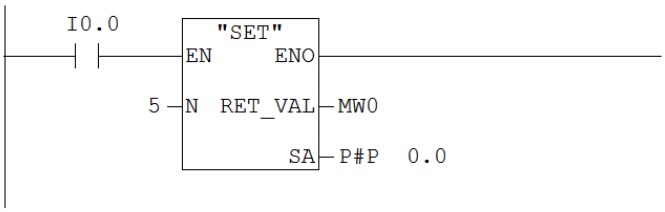
توسط این فانکشن می توان یک یا چند بیت از خروجی Peripheral در S7-400 را ریست نمود.

مثال:

در این مثال می خواهیم با فعال شدن ورودی I0.0، 5 بیت از خروجی از آدرس شروع Q0.0، ست و همچنین با فعال شدن ورودی I0.1، هر 5 بیت ری ست شوند.

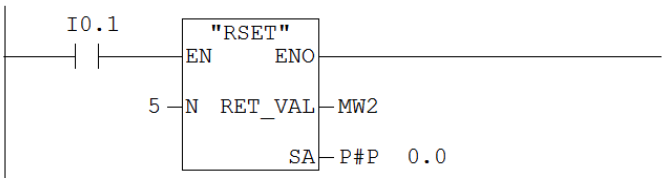
Network 1: Title:

Comment:



Network 2: Title:

Comment:



(STP)SFC46

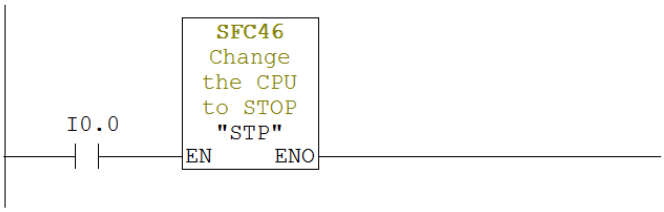
توسط این فانکشن می توان CPU را توسط برنامه به حالت STOP سوئیچ نمود. اگر در برنامه ای تحت شرایطی این فانکشن صدا زده شود، CPU متوقف خواهد شد.

مثال

فرض کنید که ورودی I0.0 به یک سنسوری متصل شده است که شرایط خطرناکی از یک پروسه صنعتی را اعلام می کند. می خواهیم برنامه ای طراحی کنیم که زمانی I0.0 فعال شد، CPU به حالت STOP سوئیچ شود.

Network 1: Title:

Comment:



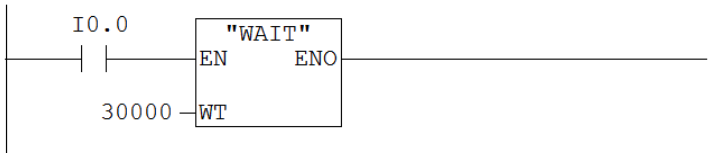
(WAIT)SFC47

توسط این فانکشن می توان در اجرای برنامه تاخیری را بر حسب میلی ثانیه ایجاد نمود. این نوع تاخیر با تاخیر ایجاد شده توسط یک تایمر متفاوت می باشد. در این تاخیر در روند اجرای برنامه تاخیر ایجاد می شود. این فانکشن دارای یک ورودی با نام WT می باشد که مدت زمان تاخیر مورد نظر که بر حسب میکروثانیه می باشد به این ورودی اعمال می شود. ماکزیمم مقدار تاخیر می تواند 32767 میکروثانیه باشد. دقت کنید که استفاده از این فانکشن باعث طولانی تر شدن زمان سیکل اسکن CPU می شود.

Akbar Oveisifar

Network 1: Title:

Comment:

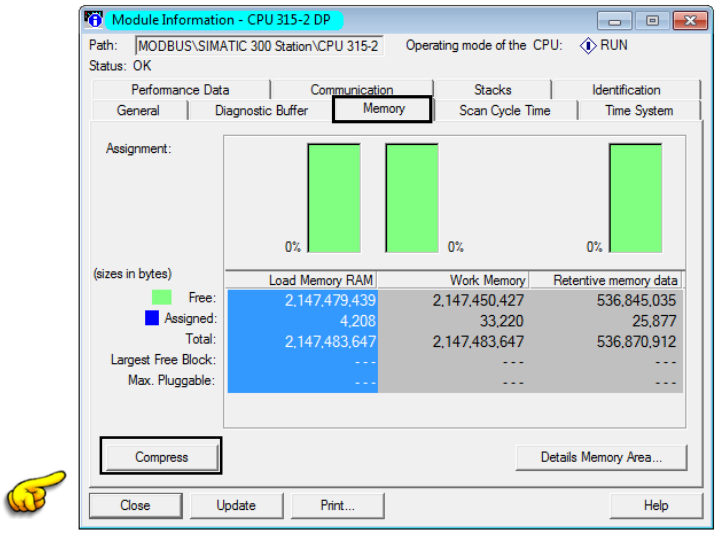


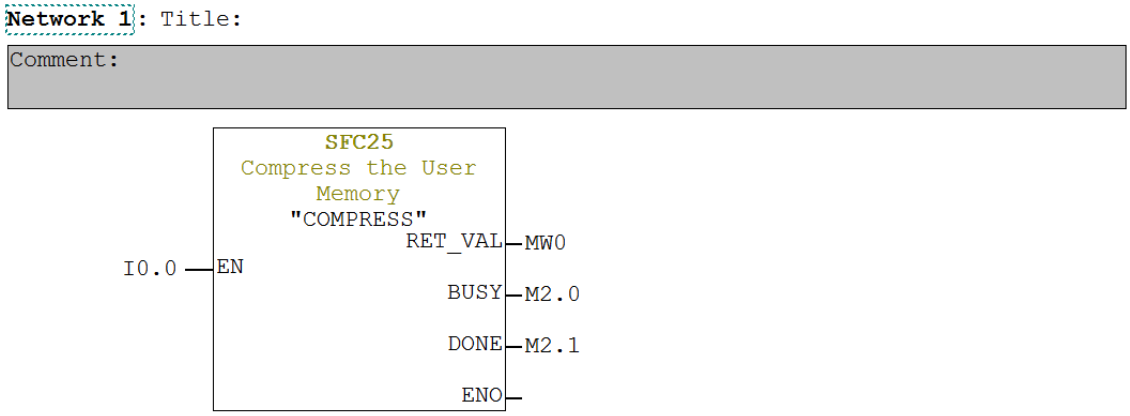
(Compress)SFC25

توسط این فانکشن می توان حافظه CPU را Compress نمود. این موضوع برای S7-400 امکان پذیر می باشد. توسط قابلیت Compress فضاهای خالی ایجاد شده در حافظه Load Memory یکپارچه می شوند. این فضاهای خالی در اثر ایجاد و پاک کردن دیتابلاک ها ایجاد می شود که باعث پرشدن کاذب فضای Load Memory می شود.

عملیات Compress به صورت دستی نیز در نرم افزار امکان پذیر می باشد.

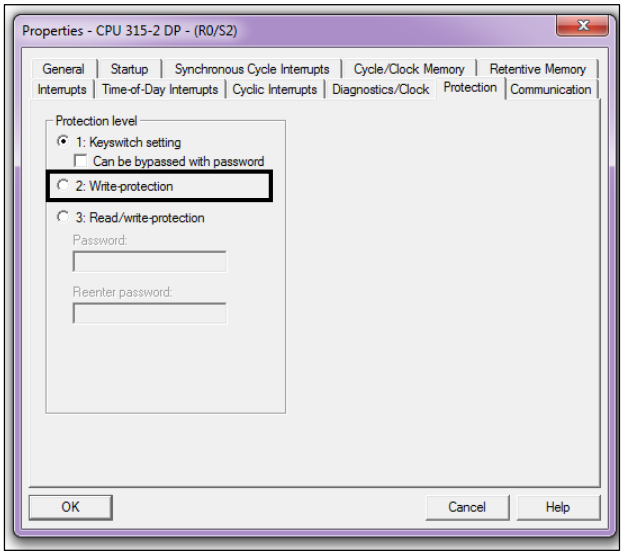
Akbar Oveisifar





(PROTECT)SFC109

توسط این فانکشن می توان توسط برنامه، CPU را در مقابل نوشتن محافظت نمود. در واقع با فعال شدن این فانکشن، امکان دانلود به CPU وجود ندارد. این فانکشن مد حفاظت را به سطح 2 سوئیچ می کند. همانطور که می دانید در سطح 2، عملیات نوشتن به CPU امکان پذیر نمی باشد.



برای استفاده از این فانکشن حتما باید در محیط HW، سطح حفاظتی برای CPU در نظر گرفته نشده باشد.(Level 1)

این فانکشن دارای ورودی MODE می باشد که می تواند دارای دو مقدار 0 یا 1 باشد.

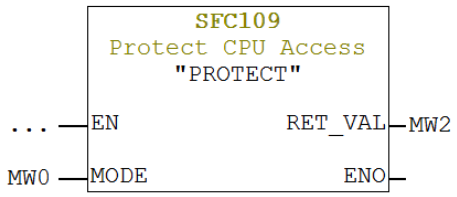
MODE=0 : سطح حفاظت 1 فعال می شود.

MODE=1 : سطح حفاظت 2 فعال شود.

Akbar Oveisifar

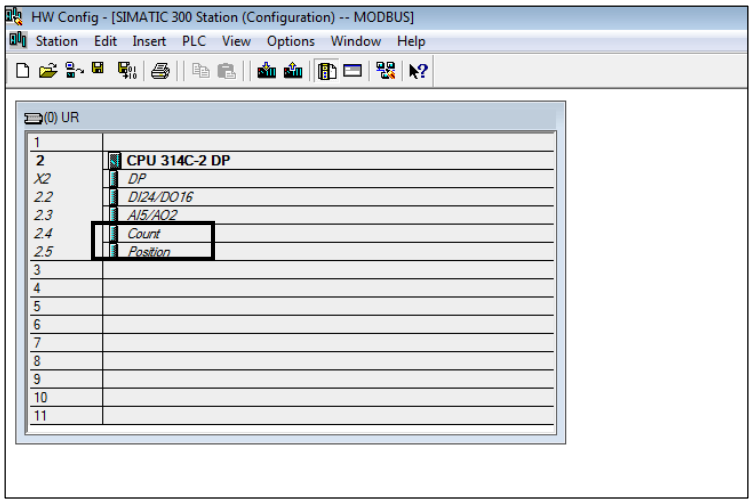
Network 1: Title:

Comment:



فانکشن های مربوط به CPU های کامپکت

در S7-300 برخی از CPU ها دارای پسوند C (کامپکت) می باشند. در این CPU ها علاوه بر I/O های Onboard، یکسری فانکشن های خاص نیز تعبیه شده است.

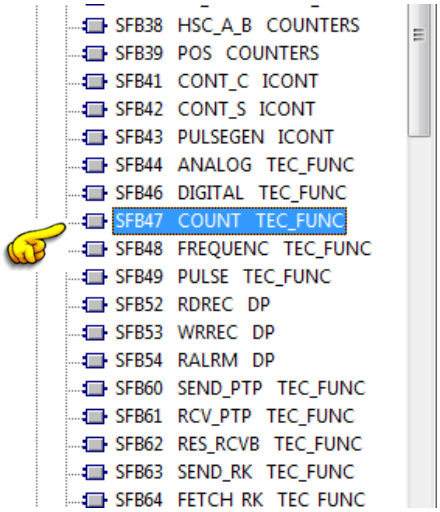


- ✓ شمارش پالس های فرکانس بالا در مد HSC
- ✓ اندازه گیری فرکانس
- ✓ تولید پالس PWM
- ✓ کنترل موقعیت دیجیتال و آنالوگ

برای استفاده از قابلیت های فوق، فانکشن های سیستمی جهت پیکربندی و برنامه نویسی در کتابخانه نرم افزار تعبیه شده است.

(COUNT) SFB47

از این فانکشن جهت کنترل شمارنده های سرعت بالا CPU استفاده می شود. این فانکشن ها دارای پارامترهای زیادی می باشند که تشریح تمامی این پایه ها خارج از بحث این مجموعه می باشد. پایه های اصلی در ادامه بررسی خواهد شد.



- **Counting frequency**
  - CPU 312C: max. 10 kHz
  - CPU 313C: max. 30 kHz
  - CPU 313C-2 DP, PtP: max. 30 kHz
  - CPU 314C-2 DP, PN/DP, PtP: max. 60 kHz
- **Signals the CPU counts**

  24 V incremental encoder with two tracks (rotary encoders) with a 90° phase shift.[1]
  - 24 V pulse encoder with direction level
  - 24 V initiator (e.g. BERO or light barrier)

ترمینال های مربوط به CPU312C

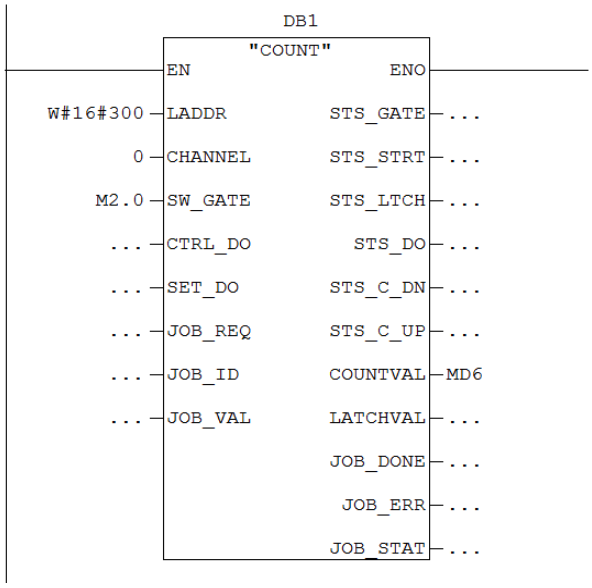
Pin Assignment CPU 312C Connector X1

Connection	Name/ address	Counting	Frequency Measurement	Pulse Width Modulation
1	-	Not connected.		
2	DI + 0.0	Channel 0: Track A/Pulse	Channel 0: Track A/pulse	-
3	DI + 0.1	Channel 0: Track B/Direction	Channel 0: Track B/direction	0/do not use
4	DI + 0.2	Channel 0: Hardware Gate	Channel 0: Hardware gate	Channel 0: Hardware gate
5	DI + 0.3	Channel 1: Track A/pulse	Channel 1: Track A/pulse	-
6	DI + 0.4	Channel 1: Track B/direction	Channel 1: Track B/direction	0/do not use
7	DI + 0.5	Channel 1: Hardware gate	Channel 1: Hardware gate	Channel 1: Hardware gate
8	DI + 0.6	Channel 0: Latch	-	-
9	DI + 0.7	Channel 1: Latch	-	-
10	DI + 1.0	-		
11	DI + 1.1	-		
12	2 M	Chassis ground		
13	1 L+	24 V power supply for the outputs		
14	DO + 0.0	Channel 0: Output	Channel 0: Output	Channel 0: Output
15	DO + 0.1	Channel 1: Output	Channel 1: Output	Channel 1: Output
16	DO + 0.2	-		
17	DO + 0.3	-		
18	DO + 0.4	-		
19	DO + 0.5	-		
20	1 M	Ground		

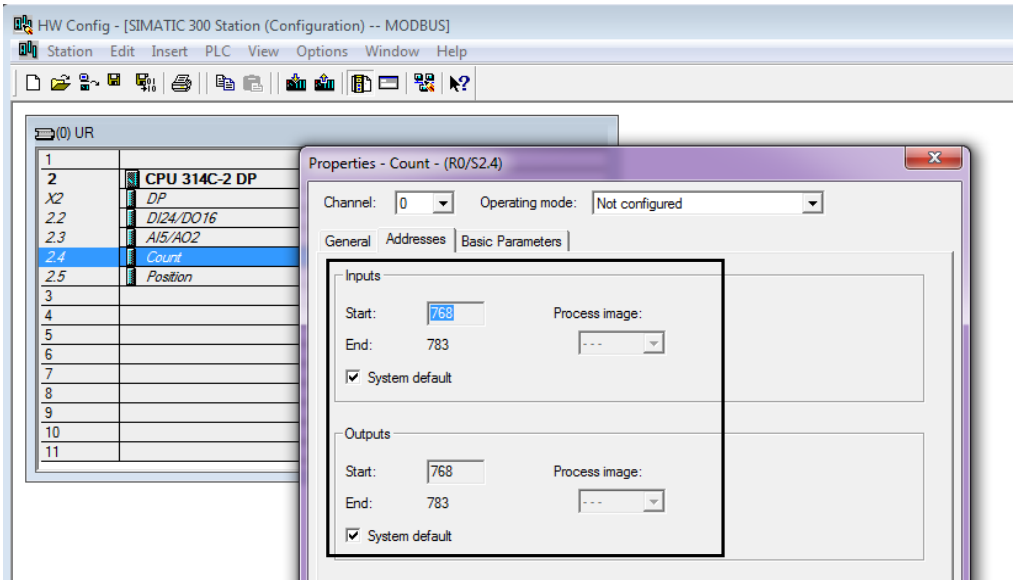
فانکشن SFB47

Network 1: Title:

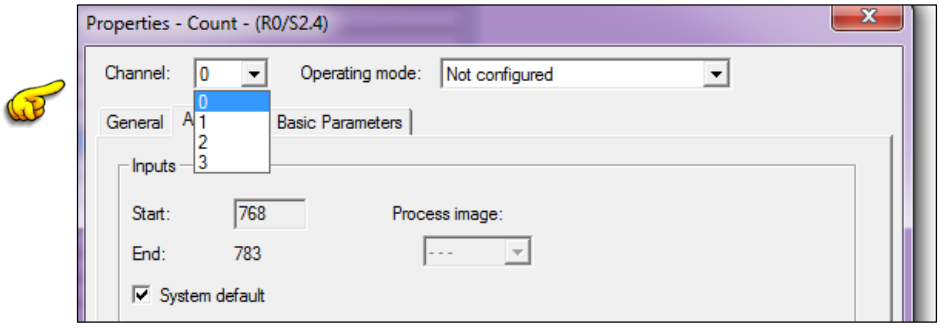
Comment:



ورودی LADDR مربوط به آدرس اولین کانال HSC می باشد که در محیط HW مشخص می شود. به صورت پیش فرض این آدرس برای کانال اول، 768 می باشد که به صورت Hex در ورودی LADDR قرار می گیرد.



ورودی CHANEL فانکشن مربوط به شماره کانال استفاده شده می باشد.

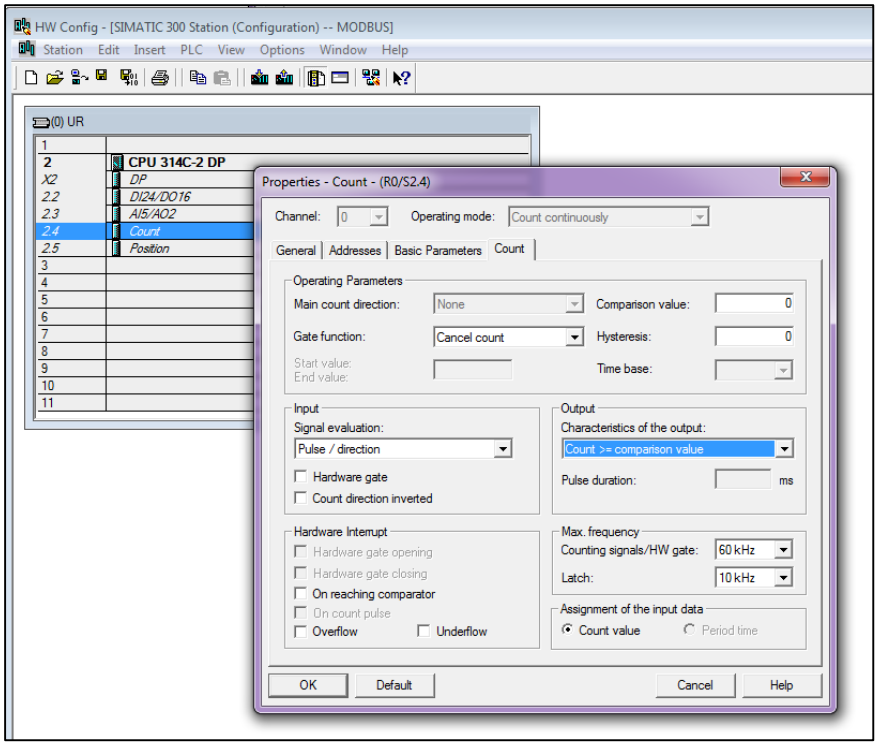


ورودی SW-GATE نیز جهت استارت و استپ شمارنده استفاده می شود.

خروجی COUNTVAL نیز جهت دریافت مقدار شمارنده می باشد. این خروجی یک مقدار 32 بیتی صحیح را بر می گرداند.

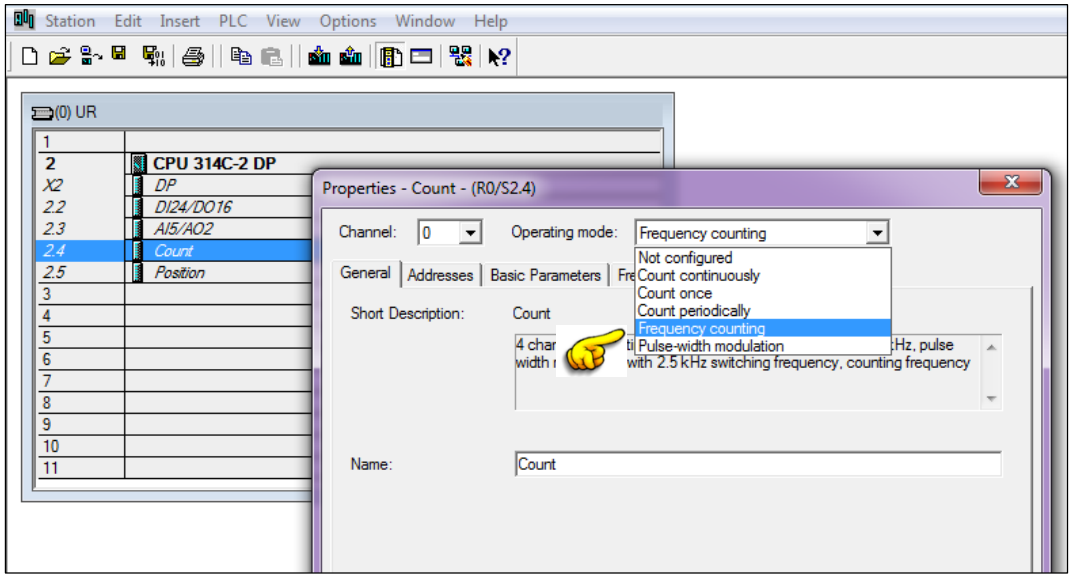
Akbar Oveisifar

سایر تنظیمات مربوط به HSC در محیط HW انجام می شود.

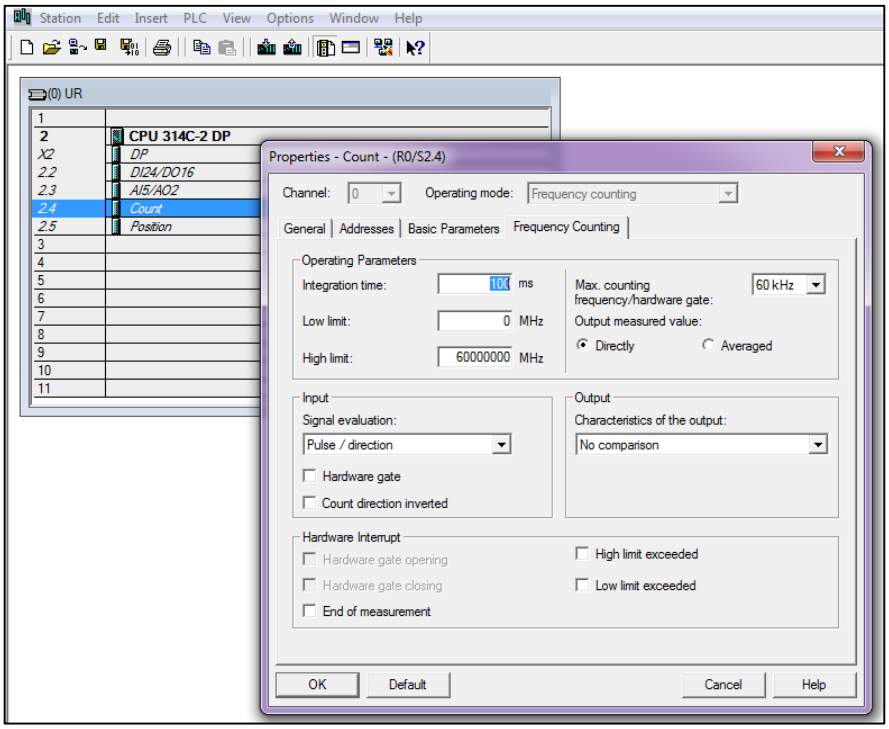


(FREQUENC) SFB48

یکی دیگر از قابلیت های CPU های کامپکت، اندازه گیری فرکانس یک پالس می باشد. این پالس می تواند از یک سنسور نصب شده بر روی شفت یک موتور یا یک انکودر دریافت شود.



تنظیمات مربوطه در محیط HW انجام می شود.



جهت خواندن مقدار فرکانس، از فانکشن SFB48 استفاده می شود.

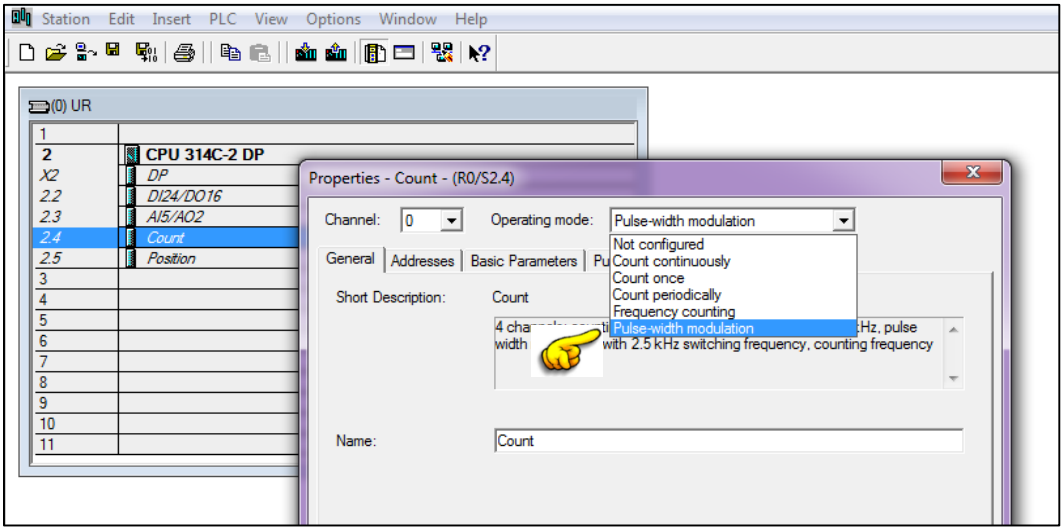
```
Network 1: Title:
Comment:
```

```
                    DB2
                "FREQUENC"
          EN                    ENO
W#16#300 — LADDR        STS_GATE — ...
       0 — CHANNEL      STS_STRT — ...
    M0.0 — SW_GATE        STS_DO — ...
     ... — MAN_DO       STS_C_DN — ...
     ... — SET_DO       STS_C_UP — ...
     ... — JOB_REQ      MEAS_VAL — MD2
     ... — JOB_ID       COUNTVAL — ...
     ... — JOB_VAL      JOB_DONE — ...
                         JOB_ERR — ...
                        JOB_STAT — ...
```

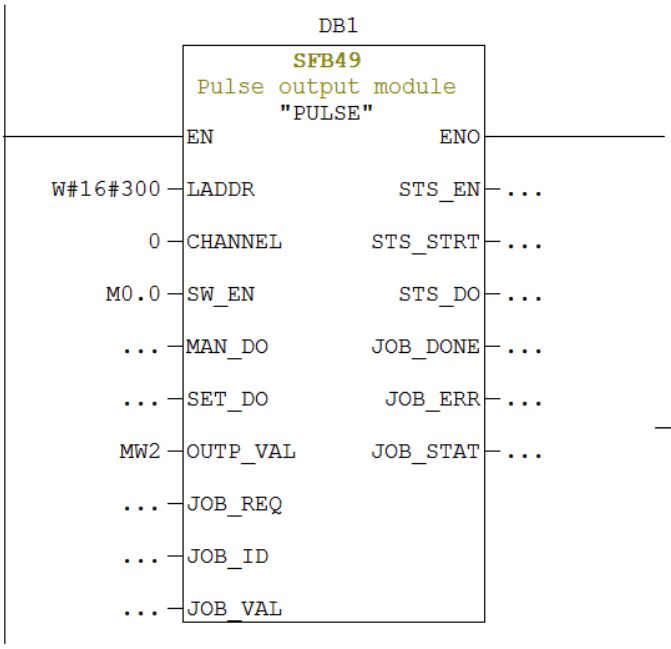
مقدار فرکانس در MD2 ذخیره می شود. با انجام یک تبدیل ساده، امکان تبدیل فرکانس به RPM وجود دارد.

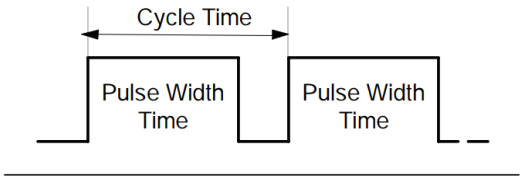
(PULSE) SFB49

از دیگر قابلیت های CPU های کامپکت، تولید پالس PWM می باشد. از کاربردهای این پالس می توان به کنترل یک المنت(هیتر) یا کنترل سرعت موتور DC توسط درایو اشاره نمود. توسط این فانکشن امکان تولید پالس فراهم می شود.



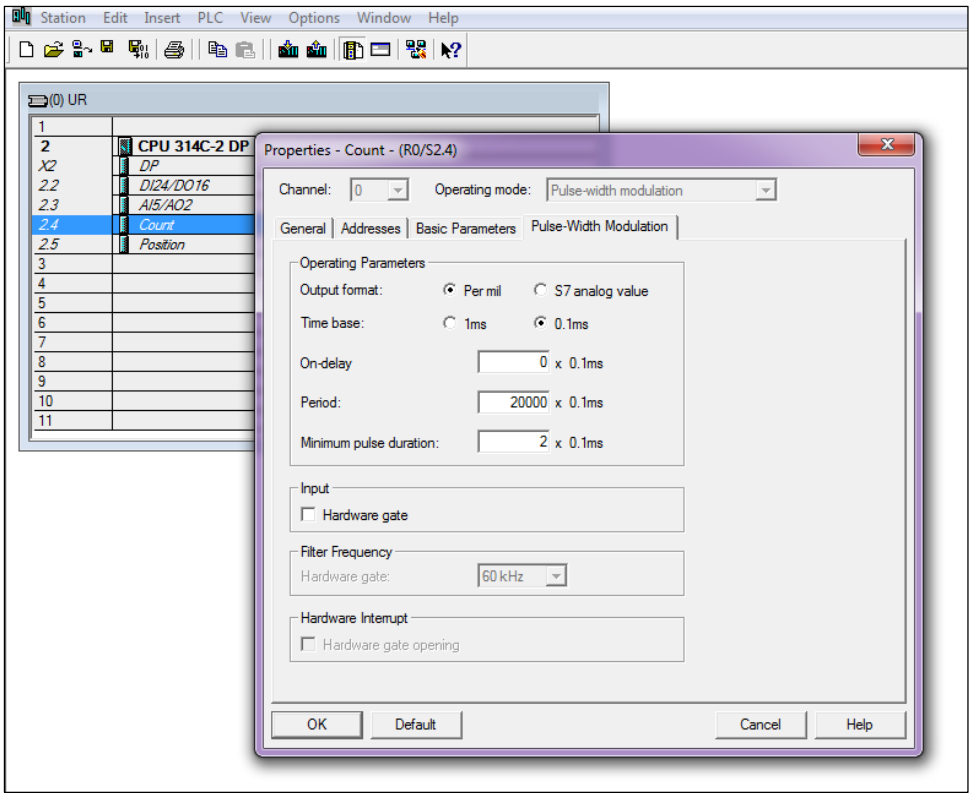

```
Network 1: Title:
Comment:
```



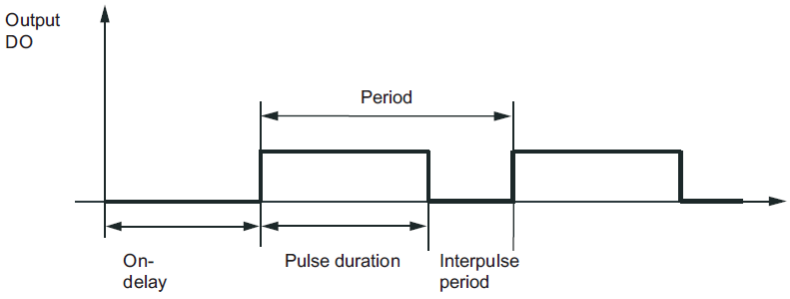


با مقدار دادن به MW2، عرض پالس را می توان تغییر داد.

تنظیمات مربوط به عرض پالس و حداقل زمان یک بودن پالس، در محیط HW انجام می شود.



Specifications of the Pulse Train	
Output frequency	0 to 2.5 kHz
Minimum pulse width	200 µs
Pulse pause accuracy Accuracy of the ON delay	±(Pulse duration x 100 ppm) ±100 µs ppm = Parts per million 0 to 250 µs The accuracy of the pulse pause can only be maintained if a maximum of one other parameter is changed in addition to the modify value during the same pulse width/pause. If several parameters are modified, the pulse width/pause may have a one-time longer or shorter length than the stated accuracy.

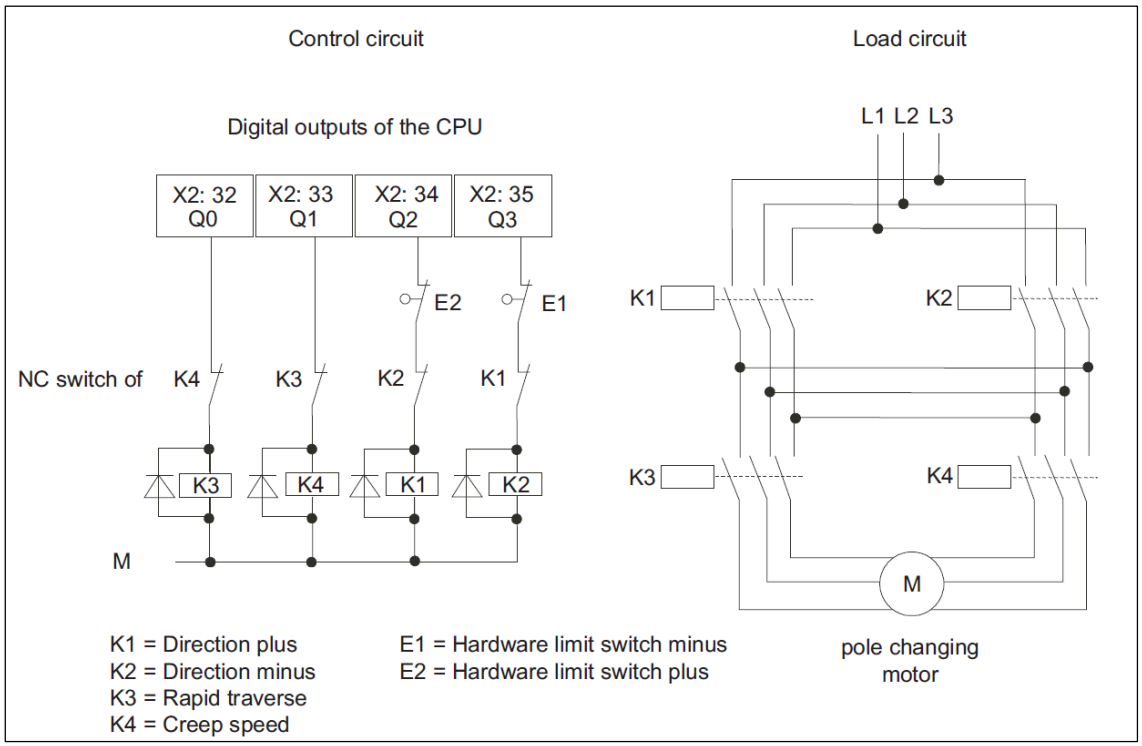


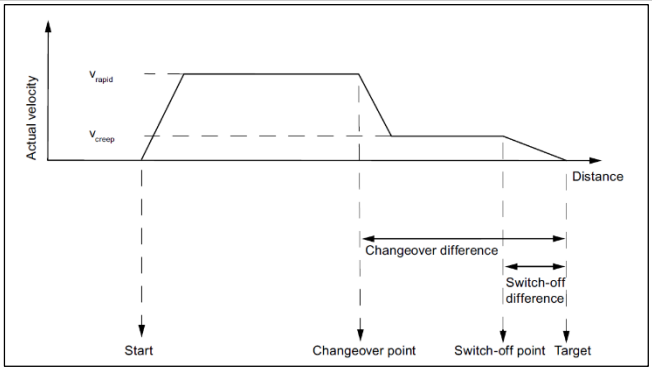
کنترل موقعیت(Positioning)

توسط CPU های کامپکت، امکان کنترل موقعیت به صورت دیجیتال و آنالوگ وجود دارد. در حالت دیجیتال، توسط مدار کنتاکتوری می توان عملیات کنترل موقعیت یک محور را انجام داد. در این حالت فیدبک توسط انکودر به CPU گزارش و سیستم کنترل، فرامین چپگرد، راستگرد، دور کند و تند را متناسب با موقعیت های تعیین شده، صادر می کند.

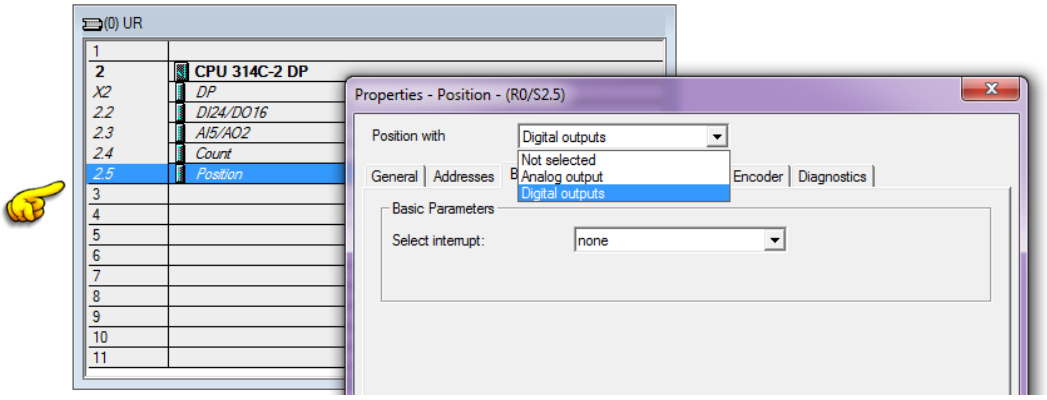
Akbar Oveisifar

Output	Control mode			
	1	2	3	4
Q0	Rapid speed	Rapid/ Creep speed	Rapid speed	Rapid speed plus
Q1	Creep speed	Position reached	Creep speed	Creep speed plus
Q2	Run plus	Run plus	Run plus	Rapid speed minus
Q3	Run minus	Run minus	Run minus	Creep speed minus

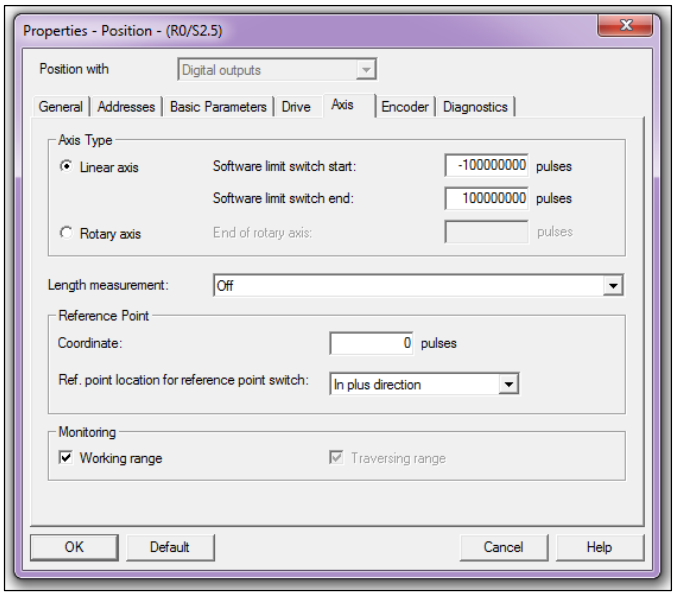




در حالت کنترل موقعیت به صورت آنالوگ، سیگنال 0-10V به یک درایو ارسال می شود.2 سیگنال دیجیتال نیز برای کنترل جهت چرخش در دسترس می باشد.



تنظیمات مربوط به درایو، انکودر، محور و نقطه رفرنس در محیط HW انجام می شود.



(ANALOG) SFB44

از این فانکشن جهت کنترل موقعیت به صورت آنالوگ در برنامه استفاده می شود. آدرس I/O مربوطه، شماره کانال، مقدار ست پوینت مربوط به موقعیت و سرعت، انتخاب مد کاری و سایر پارامترها در این فانکشن مشخص می شوند.


```
Network 1 : Title:
Comment:

                    DB1
              SFB44
          Analog positioning
               module
             "ANALOG"
          EN                ENO
W#16#310 — LADDR        WORKING — ...
       0 — CHANNEL      POS_RCD — ...
    M0.0 — DRV_EN      MSR_DONE — ...
    M0.1 — START           SYNC — ...
    M0.2 — DIR_P        ACT_POS — MD12
    M0.3 — DIR_M      MODE_OUT — ...
    M0.4 — STOP             ERR — ...
     ... — ERR_A       ST_ENBLD — ...
     MW2 — MODE_IN        ERROR — ...
     MD4 — TARGET        STATUS — ...
     MD8 — SPEED
```

(DIGITAL) SFB46

از این فانکشن جهت کنترل موقعیت به صورت دیجیتال در برنامه استفاده می شود. آدرس I/O مربوطه، شماره کانال، مقدار ست پوینت مربوط به موقعیت، انتخاب مد کاری و سایر پارامترها در این فانکشن مشخص می شوند.

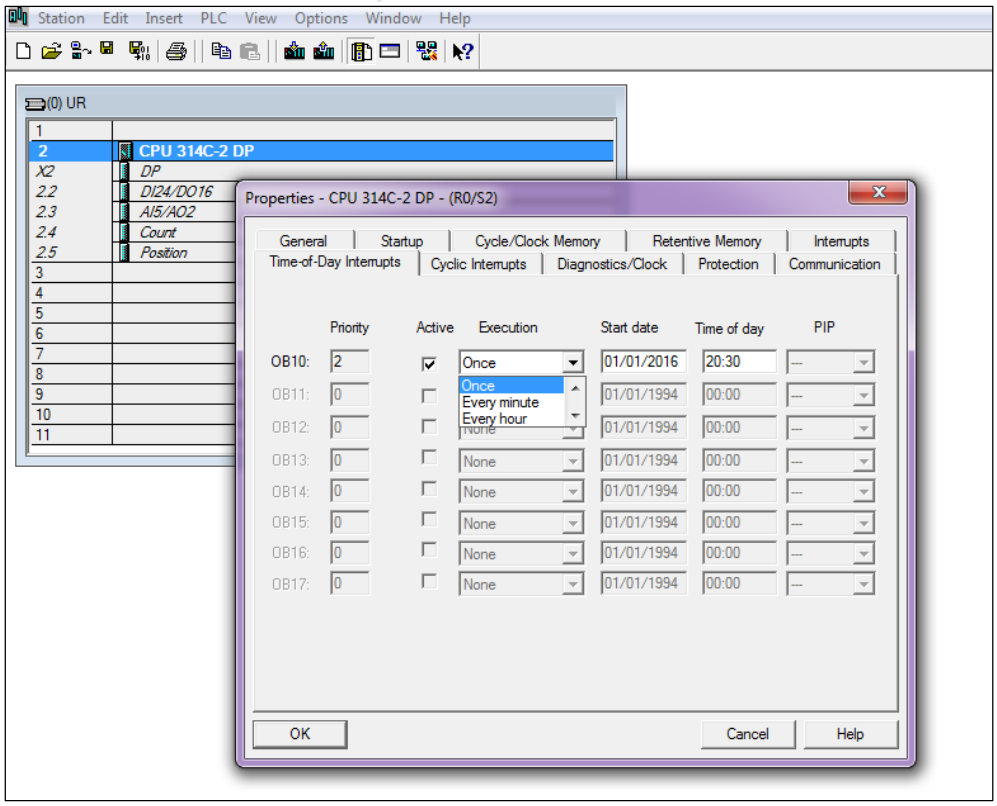

```
Network 1: Title:
Comment:

                    DB1
              SFB46
              Digital
          positioning module
              "DIGITAL"
            EN                ENO
W#16#310 — LADDR          WORKING — ...
       0 — CHANNEL        POS_RCD — ...
    M0.0 — DRV_EN        MSR_DONE — ...
    M0.1 — START             SYNC — ...
    M0.2 — DIR_P          ACT_POS — MD8
    M0.3 — DIR_M         MODE_OUT — ...
    M0.4 — STOP               ERR — ...
     ... — ERR_A         ST_ENBLD — ...
     MW2 — MODE_IN          ERROR — ...
     MD4 — TARGET          STATUS — ...
    M0.5 — SPEED
```

فانکشن های مربوط به کنترل وقفه ها

همانطور که می دانید جهت تنظیم وقفه های Time of Day، از محیط HW استفاده می شود. در این محیط تاریخ، شماره OB و پریود وقفه تنظیم می شود. در این صورت نیاز به استفاده از فانکشن خاصی در برنامه نمی باشد.

Akbar Oveisifar



یکی دیگر از روش های مدیریت این وقفه ها، استفاده از فانکشن های سیستمی می باشد. در این صورت نیازی به انجام تنظیمات در محیط HW نمی باشد.

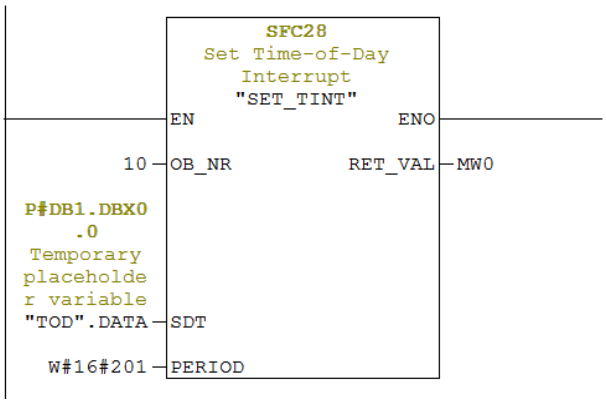
- To set time-of-day interrupts (SFC 28 "SET_TINT")
- To cancel time-of-day interrupts (SFC 29 "CAN_TINT")
- To activate time-of-day interrupts (SFC 30 "ACT_TINT")
- To query time-of-day interrupts (SFC 31 "QRY_TINT")

(SET-TINT) SFC28

توسط این فانکشن، شماره بلوک OB، تاریخ و زمان و پریود اجرایی برای وقفه مورد نظر تنظیم می شود. ورودی PERIOD با توجه به دوره اجرای مورد نظر می تواند یکی از مقادیر مشخص شده در جدول شکل زیر باشد.

Network 1 : Title:

Comment:



Parameter	Declaration	Data Type	Memory Area	Description
OB_NR	INPUT	INT	I, Q, M, D, L, constant	Number of the OB started at the time SDT + multiple of PERIOD (OB 10 to OB 17).
SDT	INPUT	DT	D, L, constant	Start date and time: The seconds and milliseconds of the specified start time are ignored and set to 0. If you want to set a monthly start of a time-of-day interrupt OB, you can only use the days 1, 2, ... 28 as a start date.
PERIOD	INPUT	WORD	I, Q, M, D, L, constant	Periods from start point SDT onwards: W#16#0000 = once W#16#0201 = every minute W#16#0401 = hourly W#16#1001 = daily W#16#1202 = weekly W#16#1401 = monthly W#16#1801 = yearly W#16#2001 = at month's end
RET_VAL	OUTPUT	INT	I, Q, M, D, L	If an error occurs while the function is active, the actual parameter of RET_VAL contains an error code.

ورودی SDT فانکشن نیز از جنس DATE-TIME می باشد. برای اختصاص زمان و تاریخ، یک سطر در DB با تایپ DT ایجاد می کنیم. در ادامه به این متغیر می توان تاریخ و زمان وقفه را منتقل نمود.

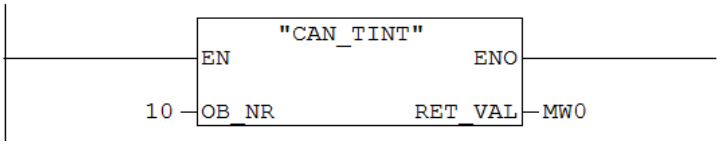
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DATA	DATE_AND_TIME	DT#90-1-1-0:0:0.000	Temporary placeholder variable
=8.0		END_STRUCT		

(CAN-TINT) SFC29

توسط این فانکشن می توان وقفه TOD فعال شده را غیر فعال نمود.

Network 2: Title:

Comment:

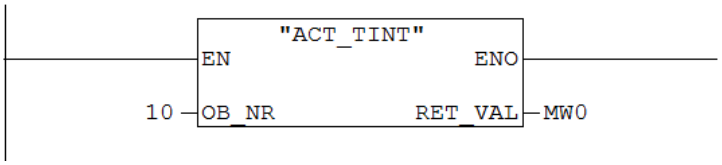


(ACT-TINT) SFC30

توسط این فانکشن می توان وقفه TOD مورد نظر را فعال نمود.

Network 2: Title:

Comment:

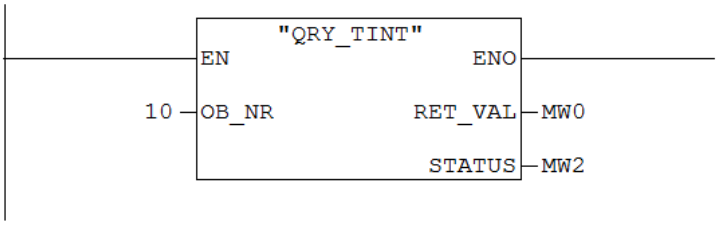


(QRY-TINT) SFC31

توسط این فانکشن می توان گزارشی از وضعیت وقفه TOD فعال شده دریافت نمود.

Network 2: Title:

Comment:



بیت های مربوط به خروجی STATUS فانکشن SFC31

Bit	Value	Meaning
0	0	Time-of-day interrupt is enabled by operating system.
1	0	New time-of-day interrupts are accepted.
2	0	Time-of-day interrupt is not activated or has elapsed.
3	-	-
4	0	Time-of-day interrupt OB is not loaded.
5	0	The execution of the time-of-day interrupt OB is not disabled by an active test function.
6	0	Base for the time-of-day interrupt is the basic time
	1	Base for the time-of-day interrupt is the local time

Akbar Oveisifar

فانکشن های مربوط به وقفه های Time Delay

جهت کنترل وقفه های Time Delay، حتما می بایست از فانکشن های سیستمی استفاده شود. توسط این وقفه ها می توان تاخیر خیلی کوتاه(حداقل ۱ میلی ثانیه) در برنامه ایجاد کرد.

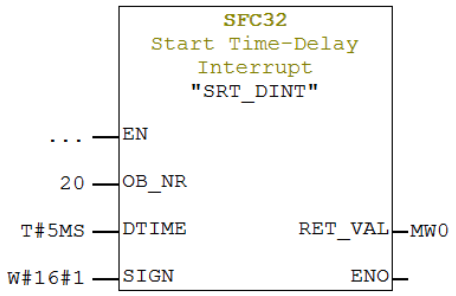
- To start time-delay interrupts (SFC 32 "SRT_DINT")
- To cancel time-delay interrupts (SFC 33 "CAN_DINT")
- To query time-delay interrupts (SFC 34 "QRY_DINT").

SFC32 (SRT-DINT)

توسط این فانکشن، وقفه TD مورد نظر فعال و زمان لازم برای ایجاد تاخیر نیز تنظیم می شود.

Network 1: Title:

Comment:



شماره OB مورد نظر، در پایه OB_NR و زمان تاخیر نیز در ورودی DTIME قرار می گیرد. ورودی SIGN نیز ورودی Identifier یا نشانه می باشد. مقدار قرار داده شده در این ورودی، در متغیر TEMP مربوط به OB فراخوانی شده(OB20) پس از تاخیر مورد نظر، دریافت می شود.

Contents Of: 'Environment\Interface\TEMP'

Interface
TEMP

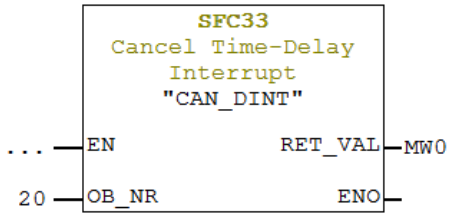
Name	Data Type	Address	Comment
OB20_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB20_STRT_INF	Byte	1.0	16#21 (OB 20 has started)
OB20_PRIORITY	Byte	2.0	Priority of OB Execution
OB20_OB_NUMBR	Byte	3.0	20 (Organization block 20, OB20)
OB20_RESERVED_1	Byte	4.0	Reserved for system
OB20_RESERVED_2	Byte	5.0	Reserved for system
OB20_SIGN	Word	6.0	Identifier input (SIGN) attached to SRT_DINT
OB20_DTIME	Time	8.0	Delay time (DTIME) input to SRT_DINT instruction
OB20_DATE_TIME	Date_And_Time	12.0	Date and time OB20 started

پایه های ورودی/خروجی فانکشن SFC32 را مشاهده می کنید.

Parameter	Declaration	Data Type	Memory Area	Description
OB_NR	INPUT	INT	I, Q, M, D, L, constant	Number of the OB, to be started after a time delay (OB20 to OB23).
DTIME	INPUT	TIME	I, Q, M, D, L, constant	Time delay value (1 to 60000 ms) You can realize longer times, for example, by using a counter in a time-delay interrupt OB.
SIGN	INPUT	WORD	I, Q, M, D, L, constant	Identifier which appears in the start event information of the OB when the time-delay interrupt OB is called.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	If an error occurs while the system function is active, the actual parameter of RET_VAL contains an error code.

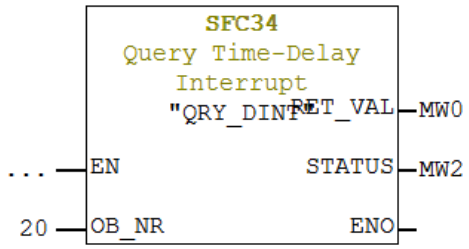
(CAN-DINT) SFC33

توسط این فانکشن می توان وقفهTD فعال شده را غیر فعال نمود.



(QRY-DINT) SFC34

توسط این فانکشن می توان گزارشی از وضعیت وقفه TD فعال شده دریافت نمود.

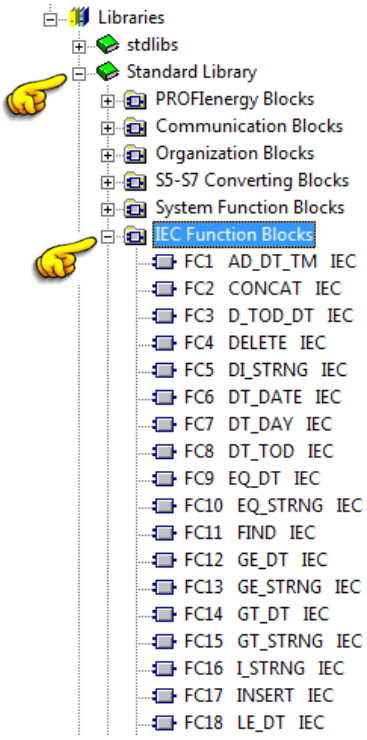


بیت های مربوط به خروجی STATUS فانکشن SFC34

Bit	Value	Meaning
0	0	Time-delay interrupt is enabled by the operating system.
1	0	New time-delay interrupts are not rejected.
2	0	Time-delay interrupt is not activated or has elapsed.
3	-	-
4	0	Time-delay interrupt-OB is not loaded.
5	0	The execution of the time-delay interrupt OB is not disabled by an active test function.

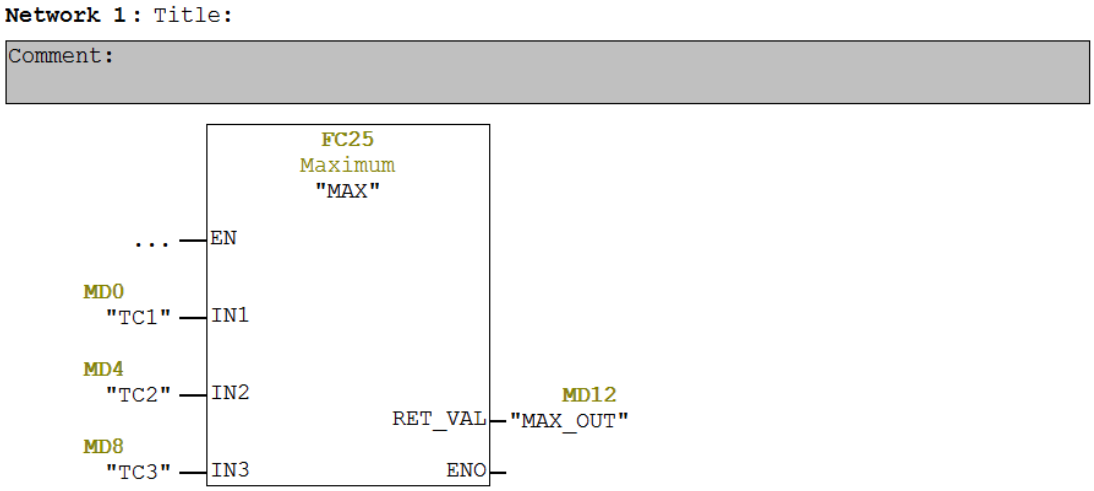
آشنایی با فانکشن های پرکاربرد کتابخانه IEC

در کتابخانه نرم افزار STEP 7، فانکشن های IEC جهت تسهیل کار برنامه نویسی در دسترس برنامه نویس می باشند. این فانکشن ها، فانکشن های سیستمی نمی باشند. در نتیجه عدم دانلود آنها، منجر به توقف CPU خواهد شد.



FC25 (MAX)

توسط این فانکشن می توان بیشترین مقدار بین سه ورودی را انتخاب و به خروجی منتقل نمود.



این فانکشن توسط سیمولاتور قابل تست می باشد.

Var - [VAT_1 -- @MODBUS\SIMATIC 300 Station\CPU 314C-2 DP\S7 Program(6) ONLINE]

Table Edit Insert PLC Variable View Options Window Help

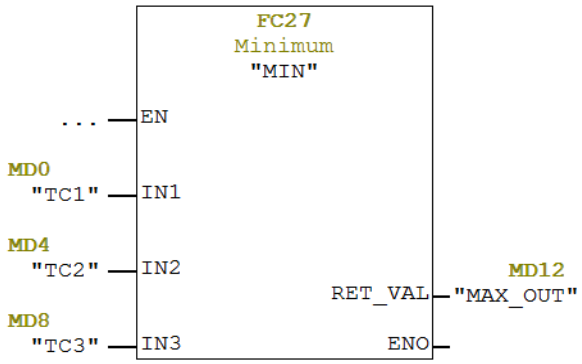
	Address	Symbol	Display format	Status value	Modify value
1	MD 0	"TC1"	FLOATING_POINT	100.0	100.0
2	MD 4	"TC2"	FLOATING_POINT	200.0	200.0
3	MD 8	"TC3"	FLOATING_POINT	250.0	250.0
4	MD 12	"MAX_	FLOATING_POINT	250.0	
5					

(MIN) FC27

توسط این فانکشن می توان کمترین مقدار بین سه ورودی را انتخاب و به خروجی منتقل نمود.

Network 1: Title:

Comment:



Var - [VAT_1 -- @MODBUS\SIMATIC 300 Station\CPU 314C-2 DP\S7 Program(6) ONLINE]

Table Edit Insert PLC Variable View Options Window Help

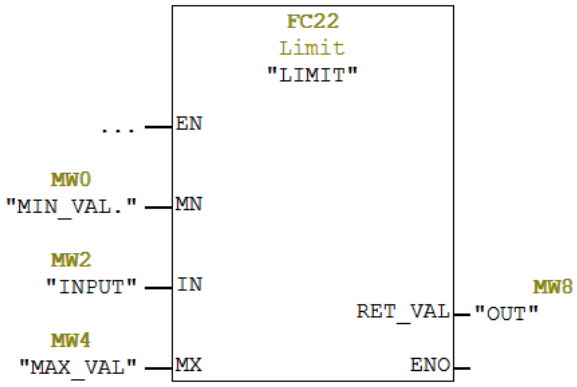
	Address	Symbol	Display format	Status value	Modify value
1	MD 0	"TC1"	FLOATING_POINT	100.0	100.0
2	MD 4	"TC2"	FLOATING_POINT	200.0	200.0
3	MD 8	"TC3"	FLOATING_POINT	250.0	250.0
4	MD 12	"MAX_	FLOATING_POINT	100.0	
5					

(LIMIT) FC22

توسط این فانکشن می توان مقدار ورودی IN را بین دو حد بالا و پایین محدود نمود.

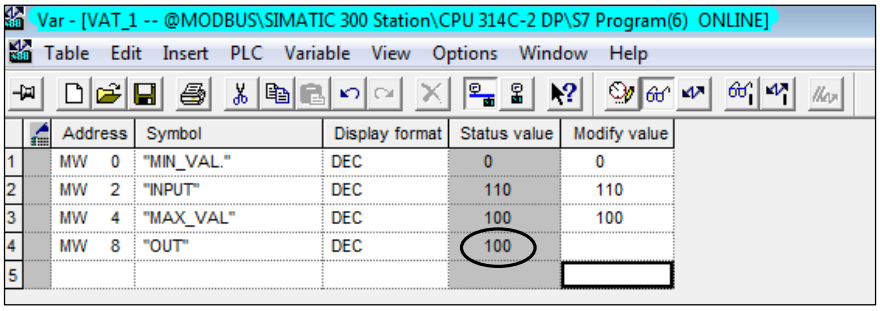
Network 1: Title:

Comment:



Akbar Oveisifar

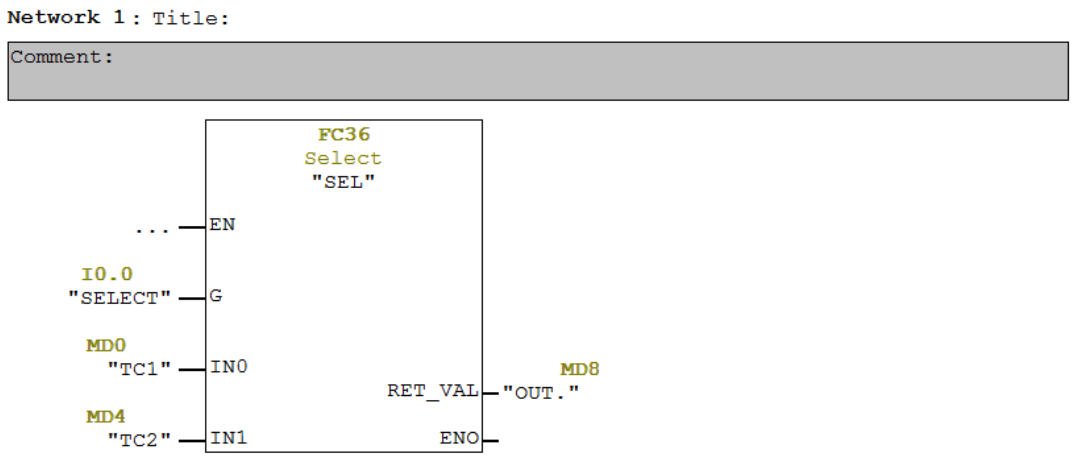
همانطور که در شکل زیر مشاهده می کنید، مقدار ورودیIN تابع برابر 110 می باشد، ولی در خروجی OUT، مقدار به حد تعیین شده در ورودی MAX که در این مثال مقدار 100 می باشد، محدود شده است.



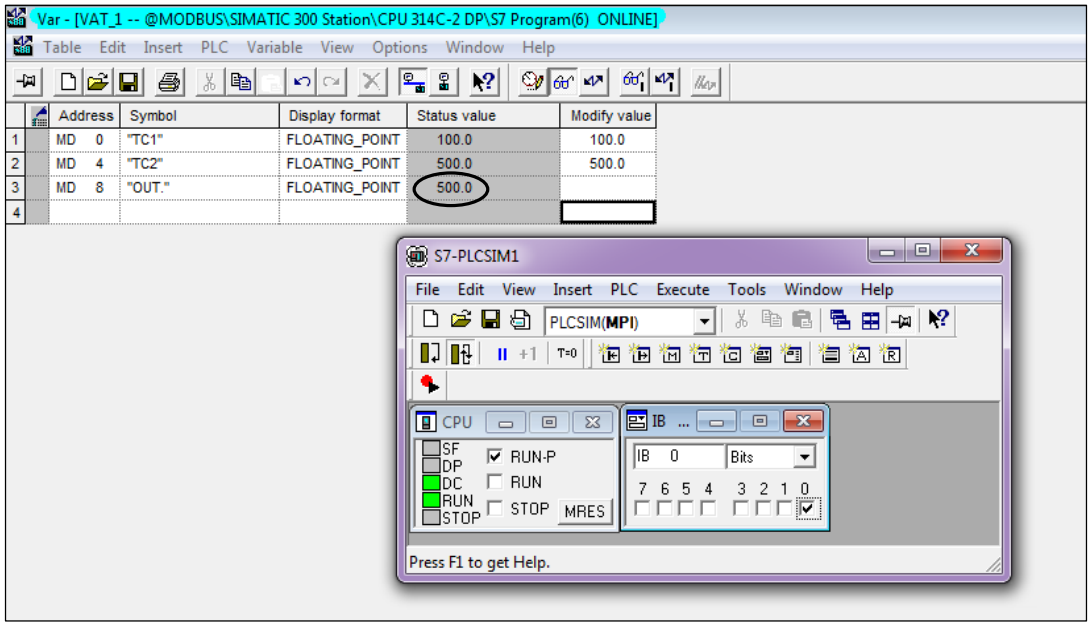
	Address	Symbol	Display format	Status value	Modify value
1	MW 0	"MIN_VAL."	DEC	0	0
2	MW 2	"INPUT"	DEC	110	110
3	MW 4	"MAX_VAL"	DEC	100	100
4	MW 8	"OUT"	DEC	100	
5					

FC36 (SEL)

این فانکشن یک مالتی پلکسر می باشد. توسط ورودی G می توان مشخص نمود که کدام ورودی به خروجی منتقل شود. ورودی G یک ورودی BOOL می باشد.



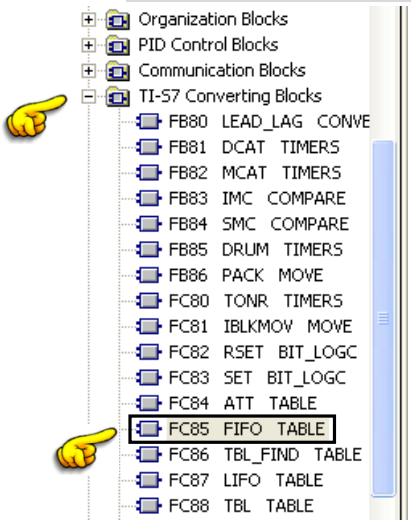
همانطور که در شکل زیر ملاحظه می کنید، با یک شدن ورودی I0.0، مقدار 500 که مربوط به IN1 می باشد، به خروجی منتقل شده است.



با 0 شدن ورودی I0.0، مقدار ورودی IN0 به خروجی منتقل خواهد شد.

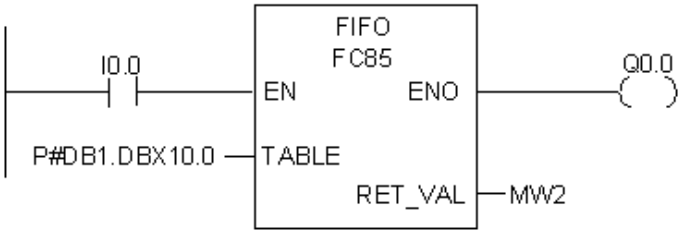
در ادامه بحث با تعدادی از فانکشن های پرکاربرد در کتابخانه نرم افزار آشنا خواهیم شد.

First Input First Output-(FIFO) FC85

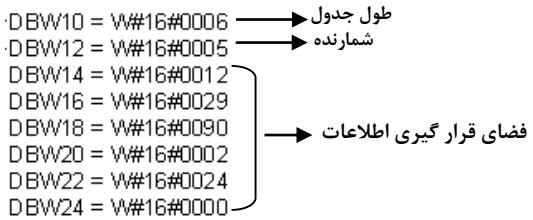


توسط این فانکشن می توان اطلاعاتی را به صورت یک جدول ثبت نمود. همچنین امکان تشخیص ترتیب وارد شدن اطلاعات نیز وجود دارد. یعنی در این فانکشن با هر بار فعال شدن ورودی EN، اطلاعات به ترتیب اولین ورود در TABLE، در خروجی RET-VAL قرارمی گیرند.

Akbar Oveisifar



نحوه قرار گیری اطلاعات در TABLE



در برنامه فوق اولین دیتا عدد ۱۲ می باشد. پس با فعال نمودن I0.0 ، این عدد به عنوان اولین ورود در MW2 قرار می گیرد. از کانتر جدول نیز یک واحد کم می شود.

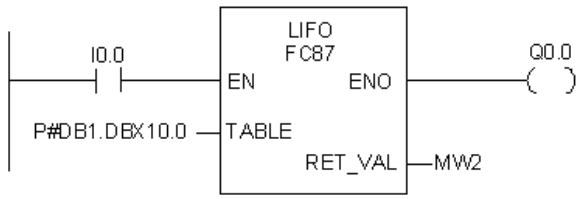
DBW10 = W#16#0006
DBW12 = W#16#0004
DBW14 = W#16#0029
DBW16 = W#16#0090
DBW18 = W#16#0002
DBW20 = W#16#0024
DBW22 = W#16#0024
DBW24 = W#16#0000

RET_VAL → MW2 = W#16#0012

اگر I0.0 برای بار دوم فعال شود، عدد ۲۹ در MW0 قرارگرفته و مجددا از کانتر جدول نیز یک واحد کم می گردد. این روند تا دیتای ۶ ادامه پیدا می کند.

Last Input First Output–(LIFO) FC87

تفاوت این فانکشن با فانکشن FIFO در این است که در FC87 ، آخرین دیتای وارد شده به جدول به عنوان اولین دیتا در خروجی قرار می گیرد.



قبل از اجرای فانکشن

DBW10 = W#16#0006
DBW12 = W#16#0005
DBW14 = W#16#0012
DBW16 = W#16#0029
DBW18 = W#16#0090
DBW20 = W#16#0002
DBW22 = W#16#0024
DBW24 = W#16#0000

MW2 = W#16#0000

بعد از اجرای فانکشن

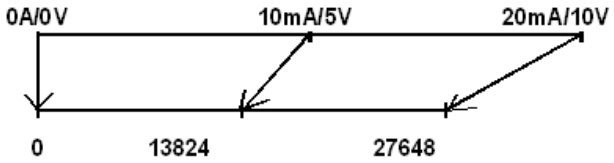
DBW10 = W#16#0006
DBW12 = W#16#0004
DBW14 = W#16#0012
DBW16 = W#16#0029
DBW18 = W#16#0090
DBW20 = W#16#0002
DBW22 = W#16#0024
DBW24 = W#16#0000

MW2 = W#16#0024

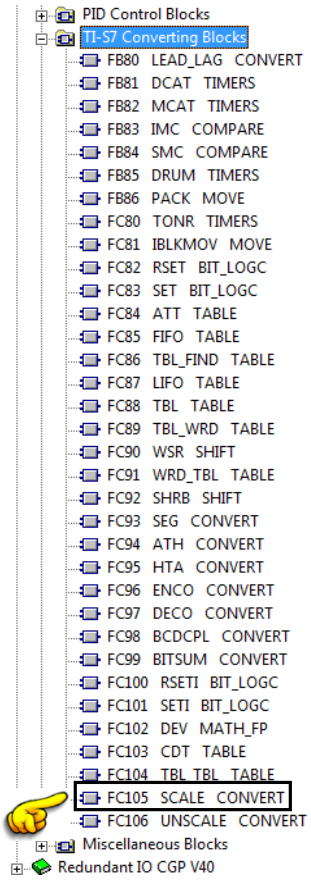
FC105 (SCALE)

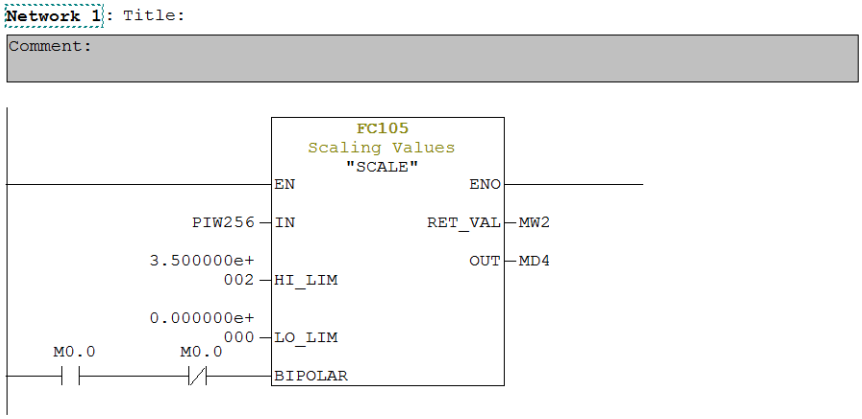
زمانی که یک ورودی آنالوگ در PLC خوانده می شود، حاصل یک عدد در بازه مشخص به صورت خام می باشد. این عدد را در اکثر مواقع بصورت مستقیم در برنامه استفاده نمی کنند. دلیل این مطلب این است که این عدد یک عدد خام بوده و هنوز معادل به واحد نشده است. در نتیجه قبل از شروع به برنامه نویسی می بایست عدد دریافت شده از مبدل ADCکارت را معادل به واحد نمود که این عمل توسط برنامه نویس انجام می شود.

SCALE یک رابطه و یا یک معادله خط می باشد. در اکثر مواقع این فرمول بصورت فانکشن آماده در اختیار برنامه نویس قرار می گیرد.



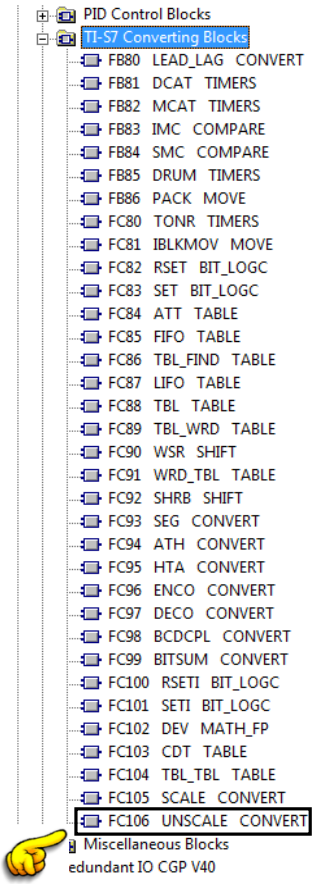
فانکشن آماده در نرم افزار برای استفاده در سیگنالهای ورودی آنالوگ، فانکشن FC105 می باشد. در واقع توسط این فانکشن می توان ورودی آنالوگ را به مقدار کمیت در بازه دلخواه تبدیل نمود.





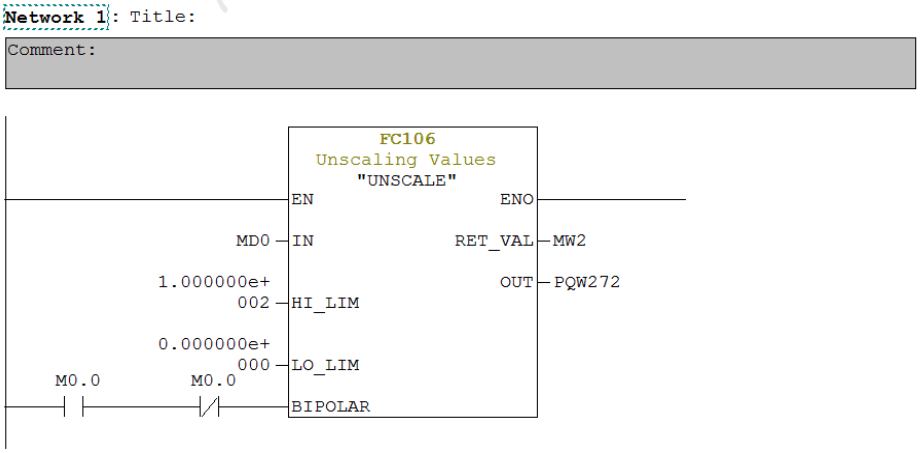
آدرس کانال مربوط به کارت AI در ورودی IN و مقادیر حد بالا و پایین کمیت نیز در ورودی های HI_LIM و LO_LIM به صورت Real وارد می شود. برای محاسبه تابع به صورت تک قطبی، ورودی BIPOLAR را در وضعیت همیشه 0 قرار می دهیم. خروجی OUT نیز خروجی اسکیل شده بین حدود بالا و پایین تعیین شده می باشد.

(UNSCALE) FC106

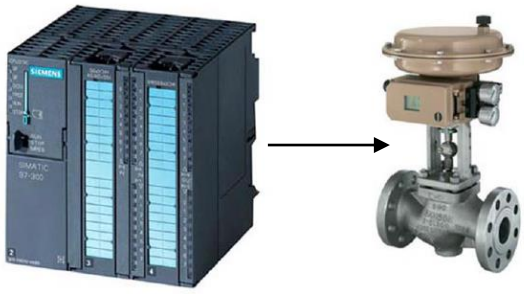


زمانی که از کارتهای خروجی آنالوگ جهت کنترل شیرهای تدریجی و یا کنترل سرعت موتور توسط درایو استفاده می شود، جهت برگرداندن مقدار Real به مقدار صحیح برای ارسال به کارت AO از تابع Unscale استفاده شود. عملکرد در کارتهای AO بدین صورت می باشد که کارت با دریافت عدد صحیح ۱۶ بیتی خام (0 تا 27648 در حالت تک قطبی) آن را به توجه به حد بالا و حد پایین مشخص شده، تبدیل به سیگنال آنالوگ می کند. عدد ارسالی به کارت همانند آنچه که در مورد AI بیان شد می تواند بازه منفی را نیز شامل شود.

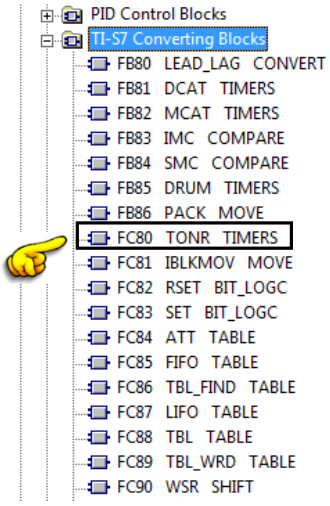
Novin Faravaran Aria Co.



در برنامه فوق، مقدار 0 تا 100 درصد که مربوط به باز شدن یک کنترل ولو می باشد، در MD0 به صورت Real قرار می گیرد. حدود بالا و پایین نیز 100.0 و 0.0 تعیین شده است. این فانکشن به ازای این مقدار در ورودیIN، مقدار خام 0 تا 27648 را به PQW272 منتقل می کند. در نتیجه سیگنال 4 تا 20 میلی آمپر به ازای این بازه عددی خام، در کانال کارت AO تولید می شود.



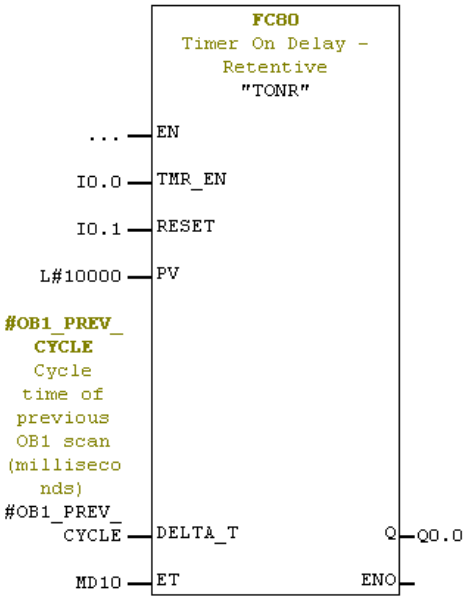
FC80 (TONR)

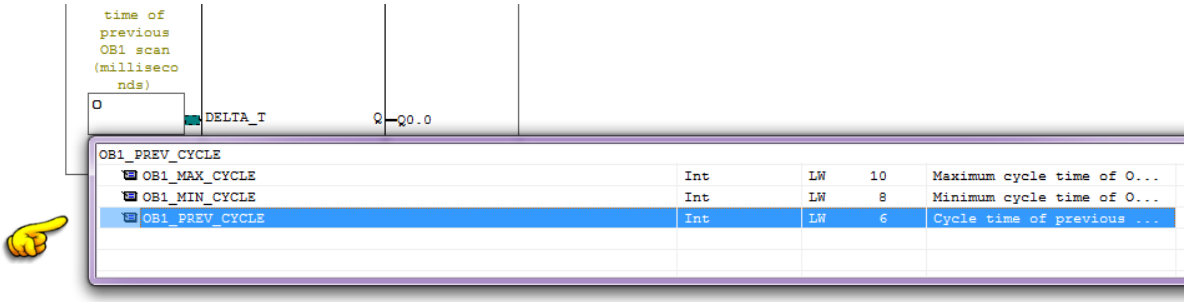


این فانکشن یک تایمر نرم افزاری تاخیر در وصل ماندگار می باشد. عملکرد این تایمر همانند تایمر TON که قبلا معرفی شد می‌باشد. با این تفاوت که اگر ورودی TMR_EN صفر شود، مقدار زمان صفر نمی‌شود، بلکه مقدار جاری تایمر حفظ شده تا مجددا ورودی یک شود. در ادامه، تایمر از مقدار قبلی حفظ شده شروع به شمارش می کند. وقتی زمان تایمر به مقدار PV رسید، بیت خروجی تایمر یک می‌شود .

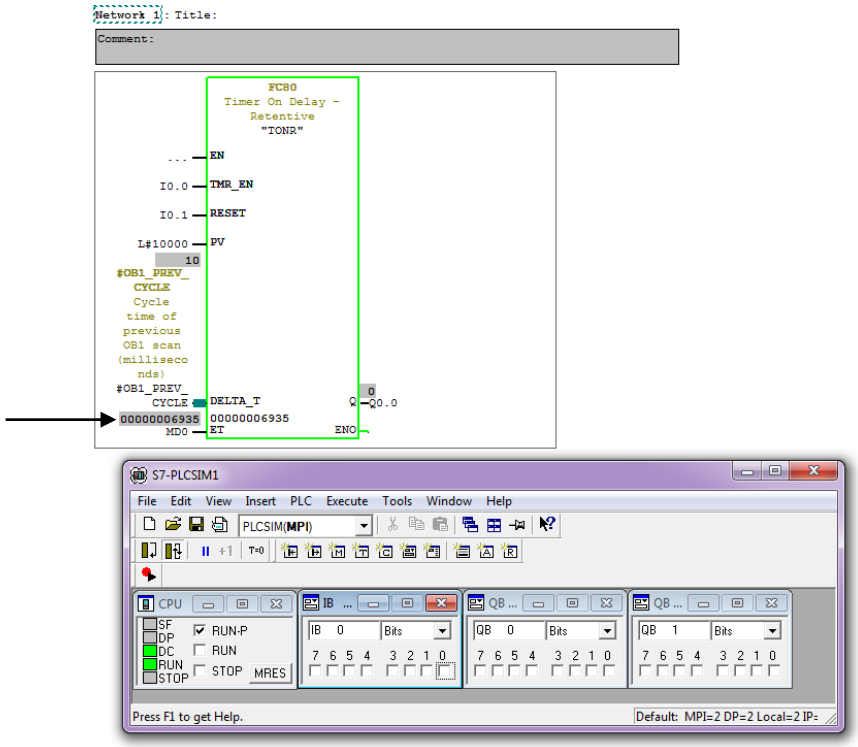
Novin Fanavaran Aria Co.

در مثال شکل زیر، مدت زمان تاخیر 10S می باشد. به ورودی DELTA-T باید زمان اسکن برنامه داده شود که در این مثال از متغیر TEMP موجود در OB1 استفاده شده است. پایه ET نیز زمان سپری شده را برمی گرداند.

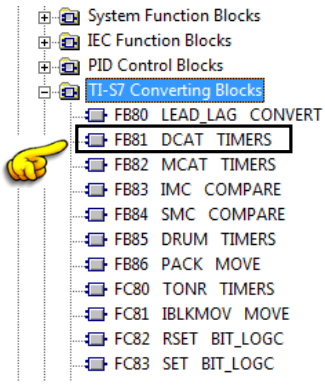




همانطور که در شکل زیر ملاحظه می کنید، با قطع ورودی I0.0 در زمان نزدیک به 7 ثانیه، مقدار زمان سپری شده ذخیره شده است. با وصل مجدد ورودی، ادامه زمان سپری و با رسیدن زمان به 10 ثانیه، خروجی Q0.0 فعال می شود.



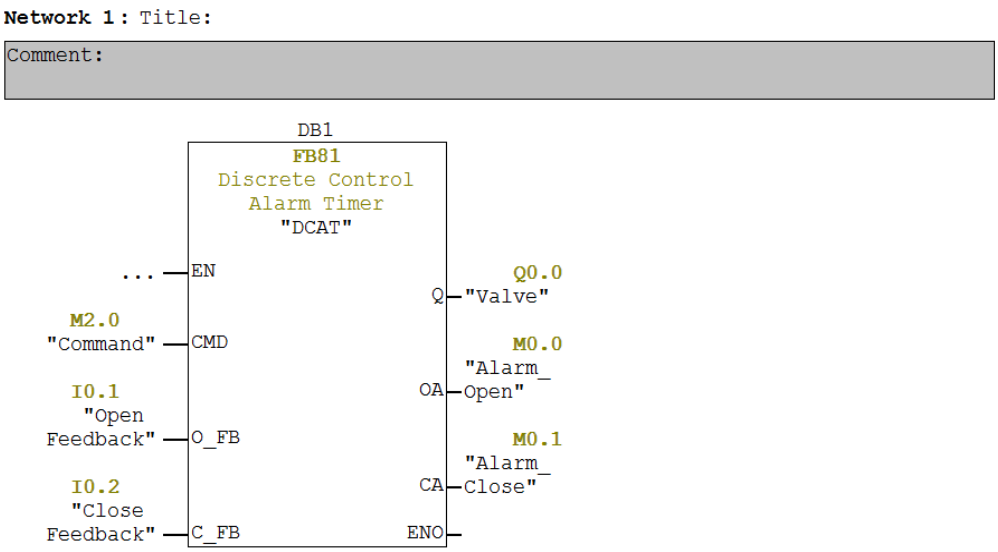
FB81 (DCAT)



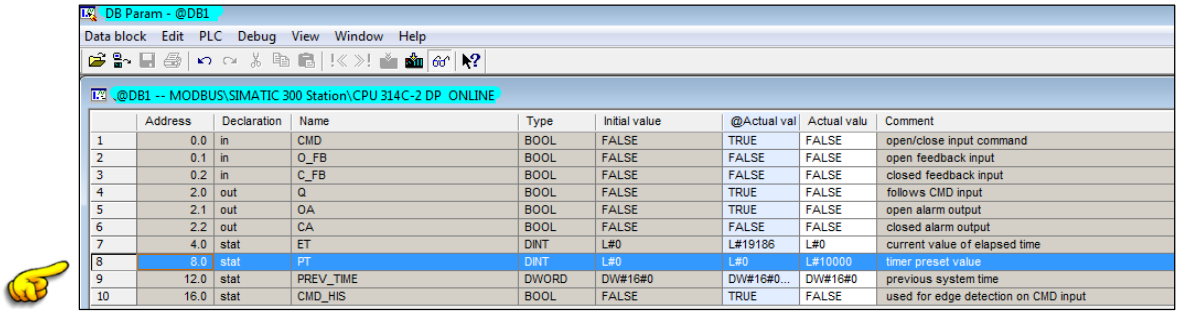
از این فانکشن یعنی Discrete Control Alarm Timer برای دریافت فیدبک Open/Close از یک عملگر دو وضعیتی استفاده می شود. اگر در طول زمان تعیین شده، فیدبک وضعیت باز یا بسته بودن ولو دریافت نشود، در خروجی آلارم تولید می شود. با فعال شدن ورودی CMD، خروجی Q نیز به صورت لحظه ای فعال می شود. زمان مورد نظر برای دریافت فیدبک، بر حسب ms در دیتا بلاک اختصاص داده شده به فانکشن مشخص می شود.

Novin Fanavaran Aria Co

در شکل زیر فانکشن FB81 را ملاحظه می کنید.



دیتابلاک در وضعیت آنلاین

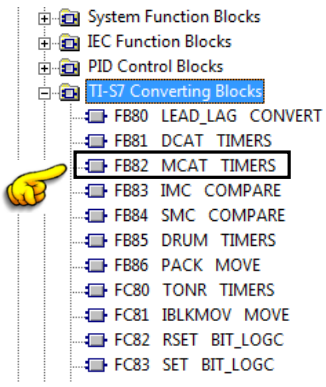


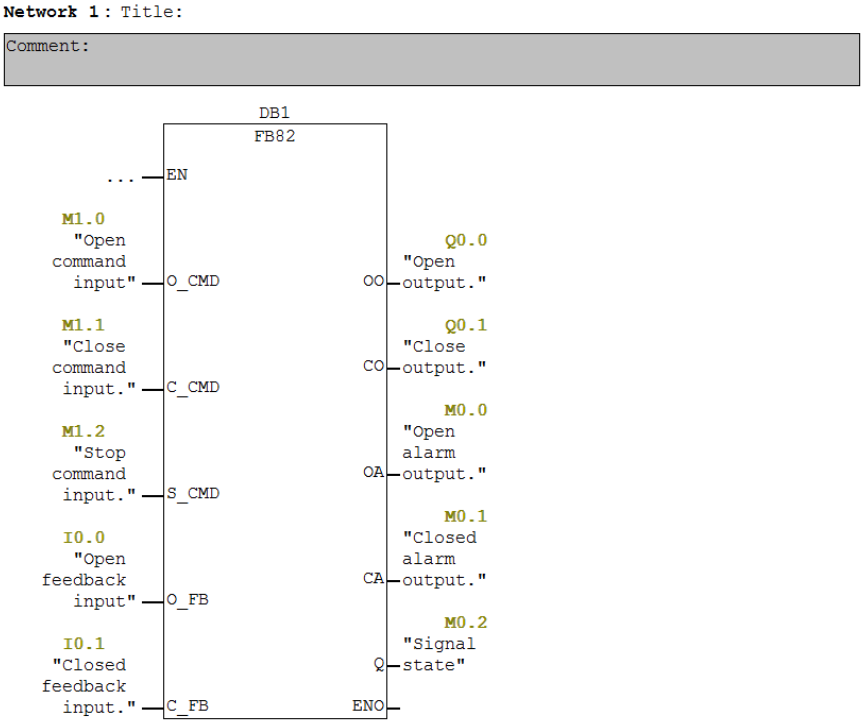
	Address	Declaration	Name	Type	Initial value	@Actual val	Actual valu	Comment
1	0.0	in	CMD	BOOL	FALSE	TRUE	FALSE	open/close input command
2	0.1	in	O_FB	BOOL	FALSE	FALSE	FALSE	open feedback input
3	0.2	in	C_FB	BOOL	FALSE	FALSE	FALSE	closed feedback input
4	2.0	out	Q	BOOL	FALSE	TRUE	FALSE	follows CMD input
5	2.1	out	OA	BOOL	FALSE	TRUE	FALSE	open alarm output
6	2.2	out	CA	BOOL	FALSE	FALSE	FALSE	closed alarm output
7	4.0	stat	ET	DINT	L#0	L#19186	L#0	current value of elapsed time
8	8.0	stat	PT	DINT	L#0	L#0	L#10000	timer preset value
9	12.0	stat	PREV_TIME	DWORD	DW#16#0	DW#16#0...	DW#16#0	previous system time
10	16.0	stat	CMD_HIS	BOOL	FALSE	TRUE	FALSE	used for edge detection on CMD input

در شکل فوق، مدت زمان دریافت فیدبک، 10 ثانیه در نظر گرفته شده است.

(MCAT) FB82

از این فانکشن برای عملگرهای سه حالته استفاده می شود(Motor Control Alarm Timer).به عنوان مثال در یک عملگر موتوری سه حالت چپگرد، راستگرد و توقف وجود دارد. در این فانکشن برای باز، بسته و متوقف شدن عملگر، سه فرمان مجزا وجود دارد که با عبارت CMD مشخص شده اند. ورودی های FB نیز مربوط به فیدبک های عملگر می باشند.



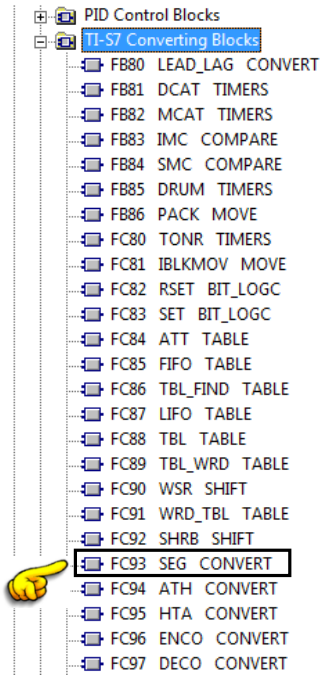


MCAT Truth Table

INPUTS								OUTPUTS								
ET	O_HIS	C_HIS	O_CMD	C_CMD	S_CMD	O_FB	C_FB	OO	CO	OA	CA	ET	O_HIS	C_HIS	Q	STATE
X	1	1	X	X	X	X	X	0	0	1	1	PT	0	0	0	Alarm
X	X	X	X	X	X	1	1	0	0	1	1	PT	0	0	0	Alarm
X	X	X	X	X	1	X	X	0	0	0	0	X	0	0	1	Stop
X	X	X	1	1	X	X	X	0	0	0	0	X	0	0	1	Stop
X	0	X	1	0	0	X	X	1	0	0	0	0	1	0	1	Begin open
<PT	1	0	X	0	0	0	X	1	0	0	0	INC	1	0	1	Opening
X	1	0	X	0	0	1	0	0	0	0	0	PT	1	0	1	Opened
>=PT	1	0	X	0	0	0	X	0	0	1	0	PT	1	0	0	Open alarm
X	X	0	0	1	0	X	X	0	1	0	0	0	0	1	1	Begin close
<PT	0	1	0	X	0	X	0	0	1	0	0	INC	0	1	1	Closing
X	0	1	0	X	0	0	1	0	0	0	0	PT	0	1	1	Closed
>=PT	0	1	0	X	0	X	0	0	0	0	1	PT	0	1	0	Close alarm
X	0	0	0	0	0	X	X	0	0	0	0	X	0	0	1	Stopped

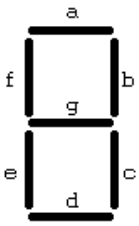
(SEG) FC93

این فانکشن جهت تولید کدهای سون سگمنت استفاده می شود. در واقع این فانکشن اعداد صحیح را در ورودی دریافت و کدهای سون سگمنت را در خروجی تولید می کند.



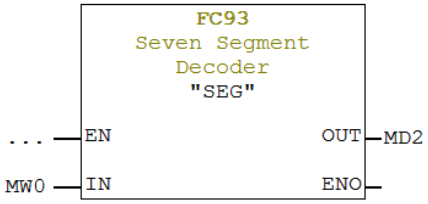
Digit	–gfedcba	Display
0000	00111111	0
0001	00000110	1
0010	01011011	2
0011	01001111	3
0100	01100110	4
0101	01101101	5
0110	01111101	6
0111	00000111	7
1000	01111111	8
1001	01100111	9
1010	01110111	A
1011	01111100	b
1100	00111001	C
1101	01011110	d
1110	01111001	E
1111	01110001	F

Seven-segment display



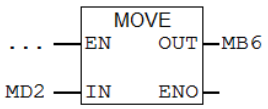
Network 1: Title:

Comment:

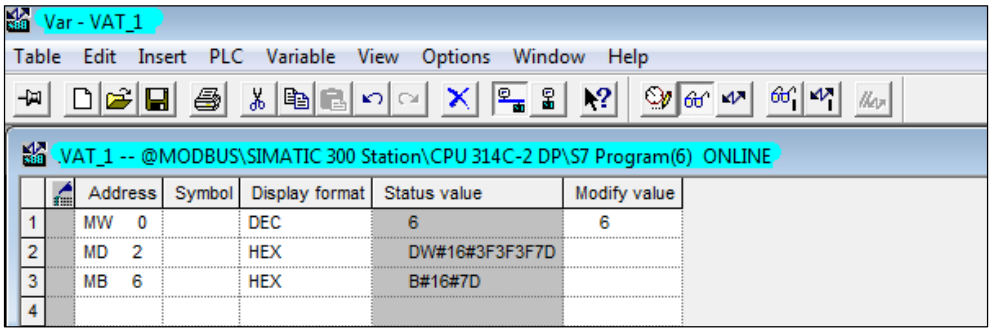


Network 2: Title:

Comment:

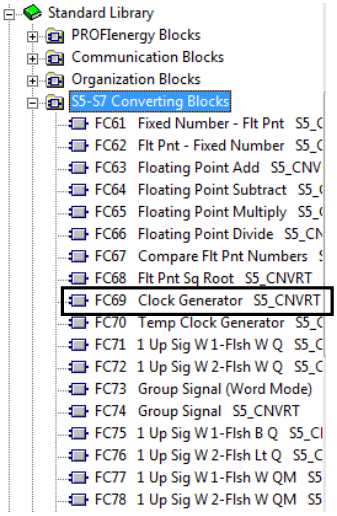


همانطور که در شکل زیر ملاحظه می کنید، مقدار 6 به صورت دسیمال در ورودی فانکشن داده شده است. خروجی فانکشن، کد هگز یا باینری مورد نیاز برای نمایش عدد 6 (7D) بر روی سون سگمنت می باشد.



	Address	Symbol	Display format	Status value	Modify value
1	MW 0		DEC	6	6
2	MD 2		HEX	DW#16#3F3F3F7D	
3	MB 6		HEX	B#16#7D	
4					

(Clock Generator) FC69



توسط این فانکشن امکان تولید پالس با فرکانس های ثابت وجود دارد. بدین منظور لازم است یک بایت به پایه FR تابع اختصاص داده شود. در این صورت بر روی 4 بیت شماره 4، 5، 6 و 7 از این بایت، چهار پالس با فرکانس های زیر تولید می شود.

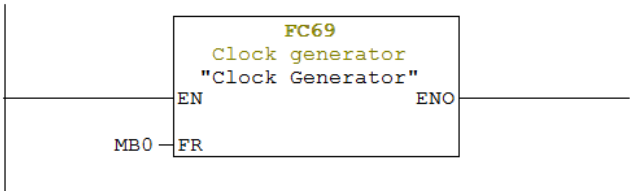
(0.63HZ , 1.25HZ , 2.5HZ , 5HZ)

نکته: این فانکشن حتما می بایست در فواصل زمانی 100 میلی ثانیه اجرا شود. یک راه حل، فراخوانی این فانکشن در بلوک OB35 می باشد.

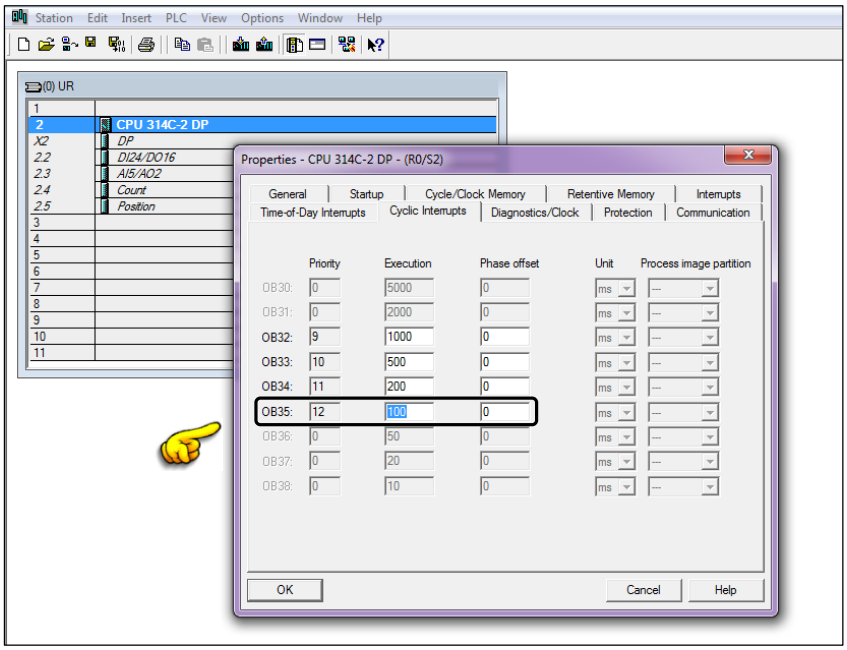
Novin Fanavaran Avia Co.

Network 1: Title:

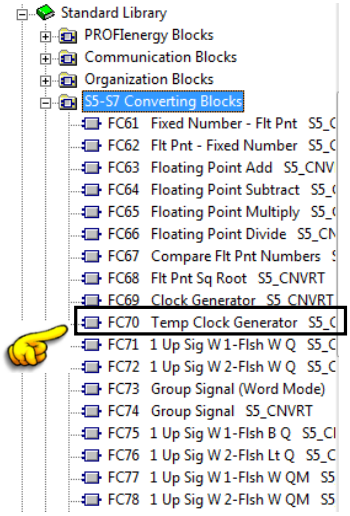
Comment:



تنظیم زمان OB35 بر روی 100ms در محیط HW



(Temp Clock Generator) FC70



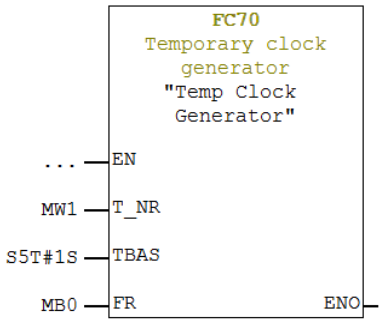
توسط این فانکشن امکان تولید پالس با فرکانس های دلخواه وجود دارد. همچنین این فانکشن می تواند در بلوک OB1 فراخوانی شود.

در برنامه شکل زیر، با اختصاص زمان 1 ثانیه به فانکشن، در بیت های 4 ، 5 ، 6 و 7 به ترتیب، با زمان های 1 ، 2 ، 3 و 4 ثانیه، پالس تولید می شود.

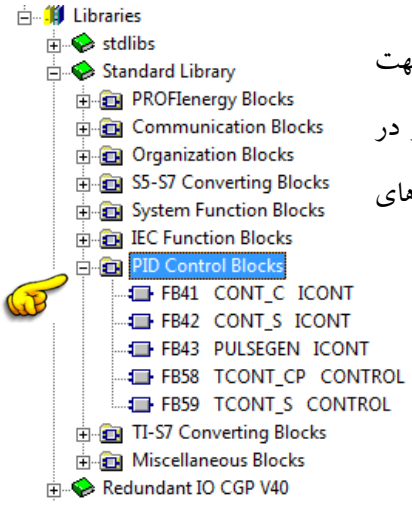
Novin Fanavaran Aria Co

Network 1 : Title:

Comment:



آشنایی مختصر با فانکشن های PID

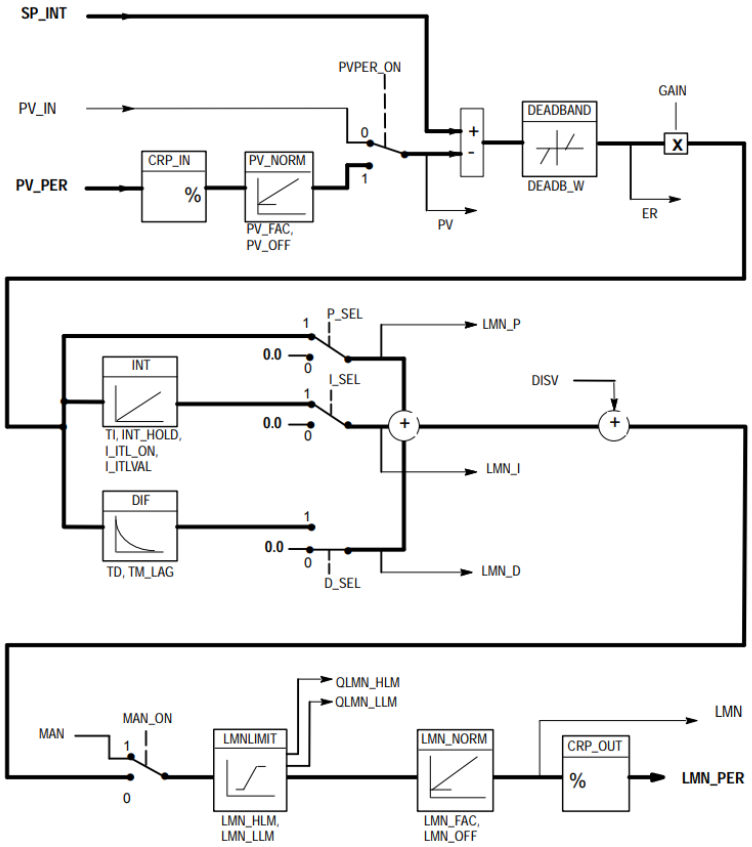


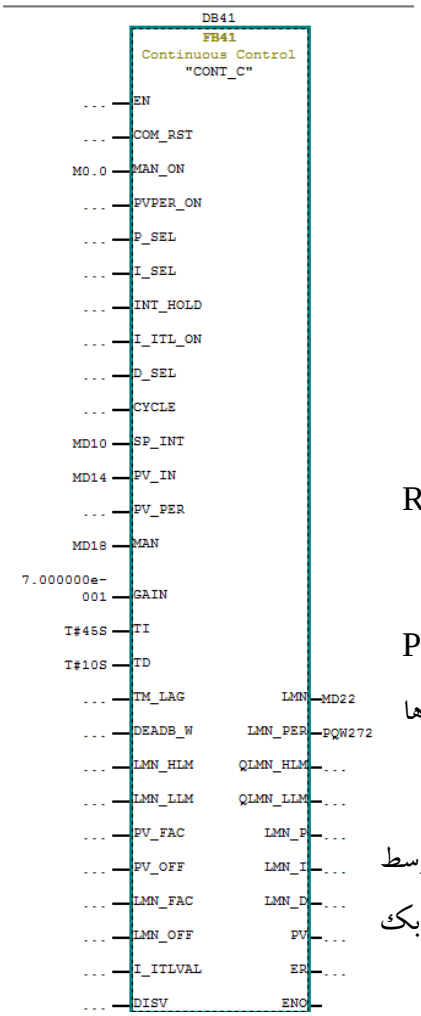
از دیگر قابلیت های نرم افزار STEP 7، در دسترس بودن فانکشن های PID متنوع، جهت طراحی انواع لوپ های کنترلی می باشد. این فانکشن ها نیز در کتابخانه نرم افزار در دسترس برنامه نویس می باشند. در این قسمت هدف آشنایی کلی با ساختار فانکشن های PID و معرفی کاربرد آنها می باشد.

Novin Fanavaran Aria co.

FB41 (CONT_C)

یکی از پرکاربردترین فانکشن های PID، فانکشن FB41 می باشد. توسط این فانکشن امکان طراحی و اجرای لوپ های کنترلی پیوسته وجود دارد. از فانکشن FB41 می توان برای انواع لوپ های پیوسته همچون دما، فشار ، فلو ، سطح و ... استفاده نمود. ورودی و خروجی این فانکشن به صورت پیوسته(آنالوگ) می باشد. در شکل زیر بخش های مختلف این فانکشن را مشاهده می کنید.





در حالت کلی، این فانکشن دو ورودی SP و PV را دریافت کرده و سپس آنها را با هم مقایسه و سیگنال خطا را تولید می کند. در ادامه پس از عبور از فیلتر بلوک DEADBAND، مقدار Error در خروجی تابع در دسترس می باشد. سپس ضرایب PID را بسته به نیاز در آن اعمال کرده و نتیجه آنها را با سیگنال اغتشاش DISV جمع می کند. در نهایت سیگنال تولید شده را بین دو حد بالا و پایین محدود و به خروجی ارسال می کند.

مقدار Set Point به پایه SP-INT داده می شود. این ورودی یک ورودی Real می باشد که می تواند از HMI دریافت شود.

مقدار فیدبک یا همان PV نیز می بایست به ورودی های PV-IN یا PV-PER اعمال شود. لازم به ذکر است که مقدار PV می بایست به یکی از این ورودی ها متصل گردد.

PV-IN : این ورودی یک ورودی Real بوده و مقدار سنسور می بایست ابتدا توسط بلوک FC105 اسکیل و خروجی آن به این ورودی وارد شود. در واقع مقدار فیدبک به صورت Unit (کمیت) به این ورودی داده می شود.

PV-PER : توسط این ورودی می توان مقدار خام دریافتی از کانال PIW را وارد بلاک نمود. در این ورودی آدرس کانالی که سنسور به آن متصل شده است، قرار می گیرد. مقدار خام اعمال شده به FB41، توسط بلوک داخلی CRP-IN بین مقدار 100+% الی 100-% توسط رابطه زیر اسکیل می شود.

$$\text{Output of CRP_IN} = \text{PV_PER} * \frac{100}{27648}$$

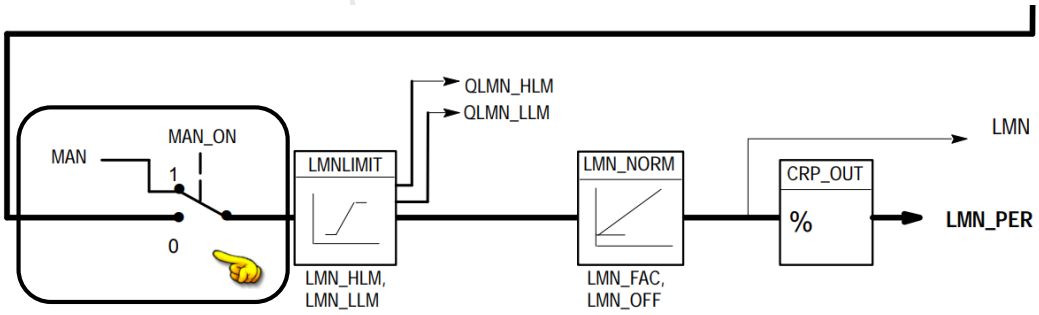
توسط ورودی های Gain ، TI و TD نیز ضرایب PID فانکشن اعمال می شوند.

Novin Fanavaran Aria Co

توسط ورودی MAN_ON می توان لوپ کنترلی را در وضعیت اتوماتیک یا دستی قرار داد. این ورودی یک ورودی بیتی بوده که در حالت پیش فرض دارای مقدار 1 می باشد.

با بودن لوپ در وضعیت Manual، مقدار دستی مورد نظر جهت ارسال به عملگر، توسط ورودی MAN داده می شود. این ورودی یک ورودی Real بوده که درصد فرمان خروجی را در حالت دستی تعیین می کند.

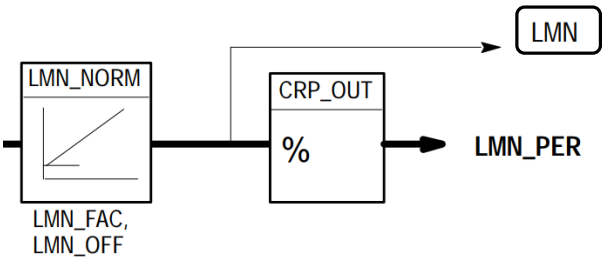
Akbar Oveisifar



LMN_PER: این خروجی به صورت مستقیم به کانال ماژول AO متصل می شود. در واقع سیگنال مربوط به عملگر از این خروجی دریافت می شود. آدرس کانال مورد نظر به صورت PQW در این خروجی قرار می گیرد. این خروجی از یک بلوک Unscale عبور می کند. بلوک CRP_OUT مقدار درصد فرمان را دریافت و خروجی را به صورت مقدار خام (0 تا 27648) به کانال AO برمی گرداند.

$$LMN_PER = LMN * \frac{27648}{100}$$

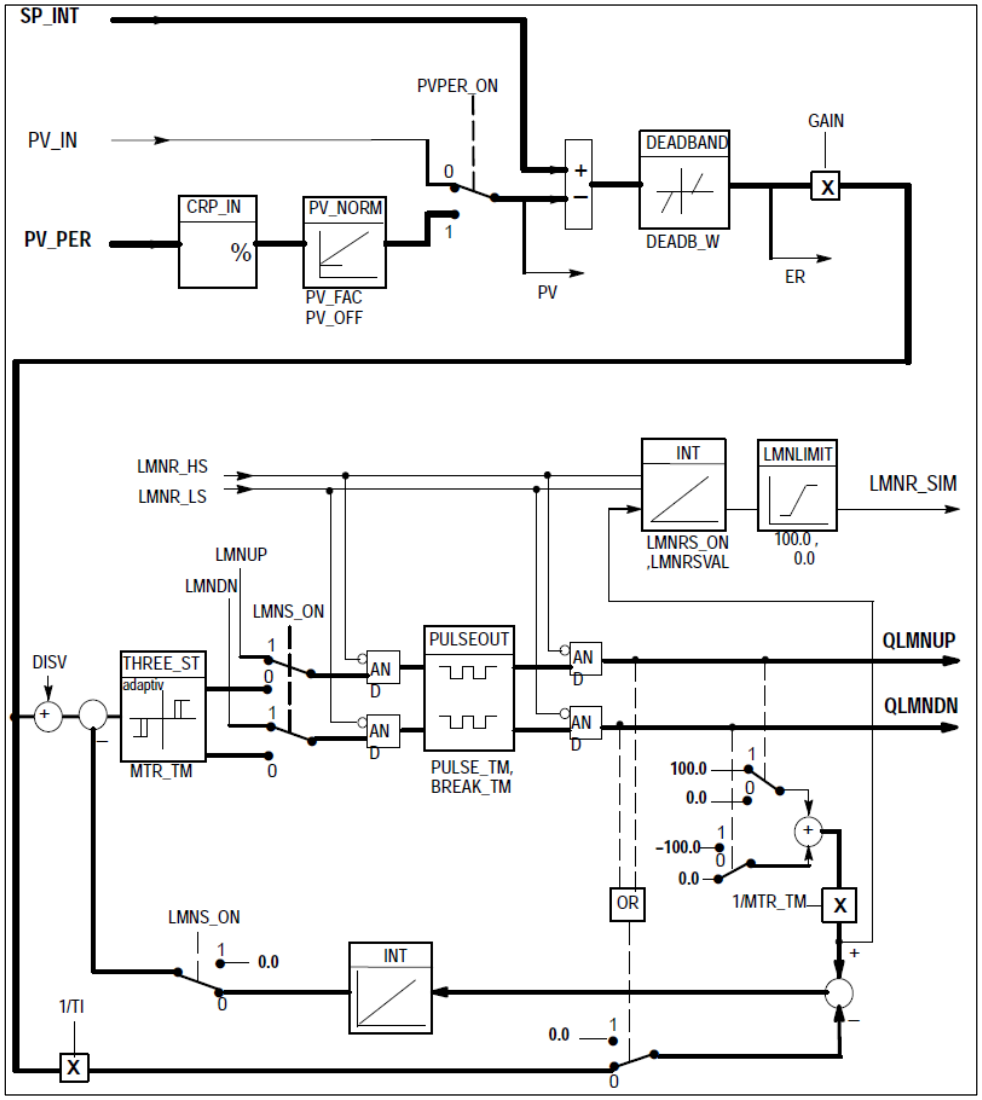
LMN: یک خروجی Real می باشد که درصد فرمان PID را نشان می دهد. این خروجی معمولا جهت نمایش به سیستم مانیتورینگ منتقل می شود.



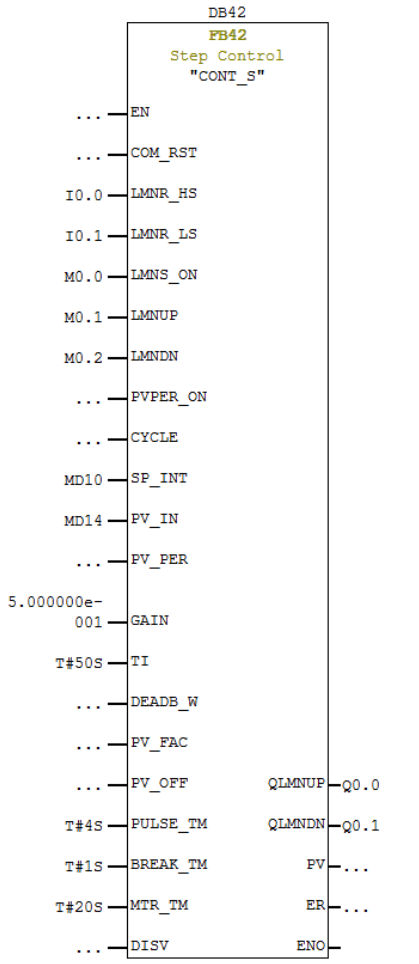
(CONT_S) FB42

از این فانکشن برای کنترل مرحله ای عملگرهای موتوری سه حالته استفاده می شود. این ولوها دارای فرامین Open، Close و Off می باشند. خروجی این تابع به صورت پالس می باشد که در دو خروجی QLMNUP و QLMNDN تولید می شود. در ضمن این فانکشن به صورت PI عملیات کنترل لوپ را انجام می دهد. فانکشن بر اساس ضرایب PI ، فاصله بین پالس ها و پهنای پالس را تغییر می دهد. زمانی که مقدار خطا بیشتر از 0 باشد، پالس در خروجی QLMNUP (فرمان باز شدن ولو)و زمانی که مقدار خطا کمتر از 0 باشد، پالس در خروجی QLMNDN (فرمان بسته شدن ولو)تولید می شود.

Akbar Oveisifar



فانکشن FB42



PULSE_TM : حداقل مدت زمان پالس

BREAK_TM : حداقل فاصله زمانی بین دو پالس

MTR_TM : زمان مورد نیاز برای تغییر کامل موقعیت عملگر

LMNR_HS : میکروسوئیچ حالت کاملا باز عملگر

LMNR_LS : میکروسوئیچ حالت کاملا بسته عملگر

LMNS_ON: تغییر مدکاری لوپ(Automatic/Manual)

LMNUP : فرمان باز شدن عملگر در مد Manual

LMNDN : فرمان بسته شدن عملگر در مد Manual

SP_INT : مقدار Set Point

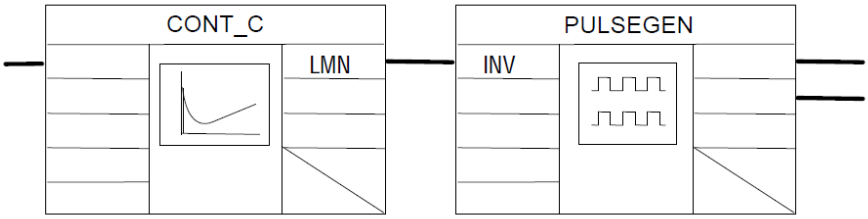
PV_IN : مقدار فیدبک اسکیل شده(خروجی FC105)

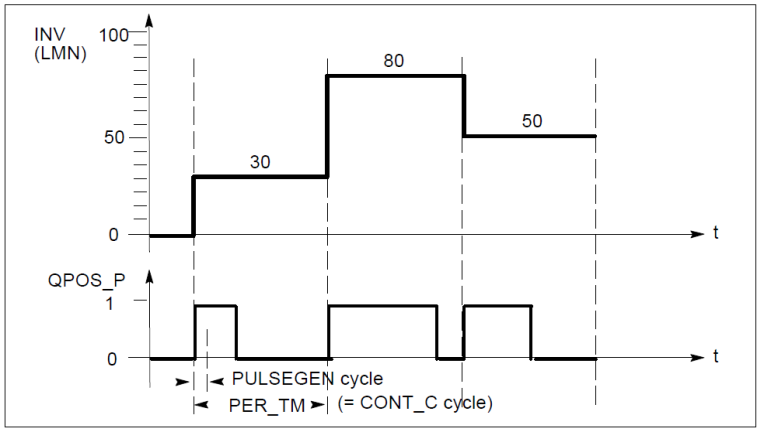
QLMNUP : فرمان باز شدن عملگر

QLMNDN : فرمان بسته شدن عملگر

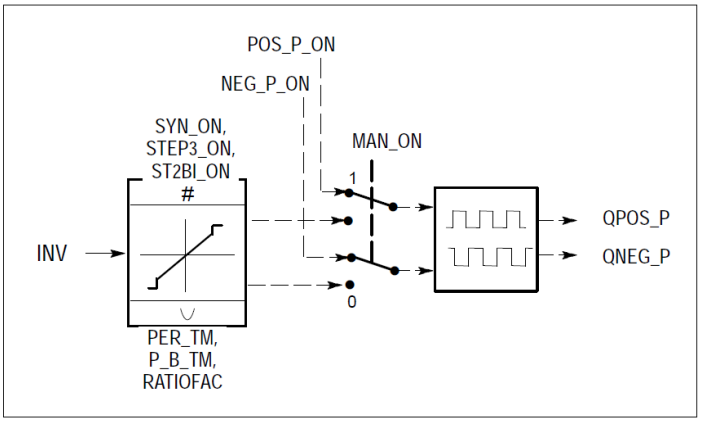
(PULSEGEN) FB43

توسط این فانکشن می توان یک مقدار پیوسته(Real) را به پالس تبدیل نمود. این فانکشن به تنهایی کاربرد ندارد. خروجی FB41 می تواند به ورودی این فانکشن اعمال می شود. در ادامه بسته به مقدار درصد فرمان سیگنال پیوسته، عرض پالس تغییر می کند.

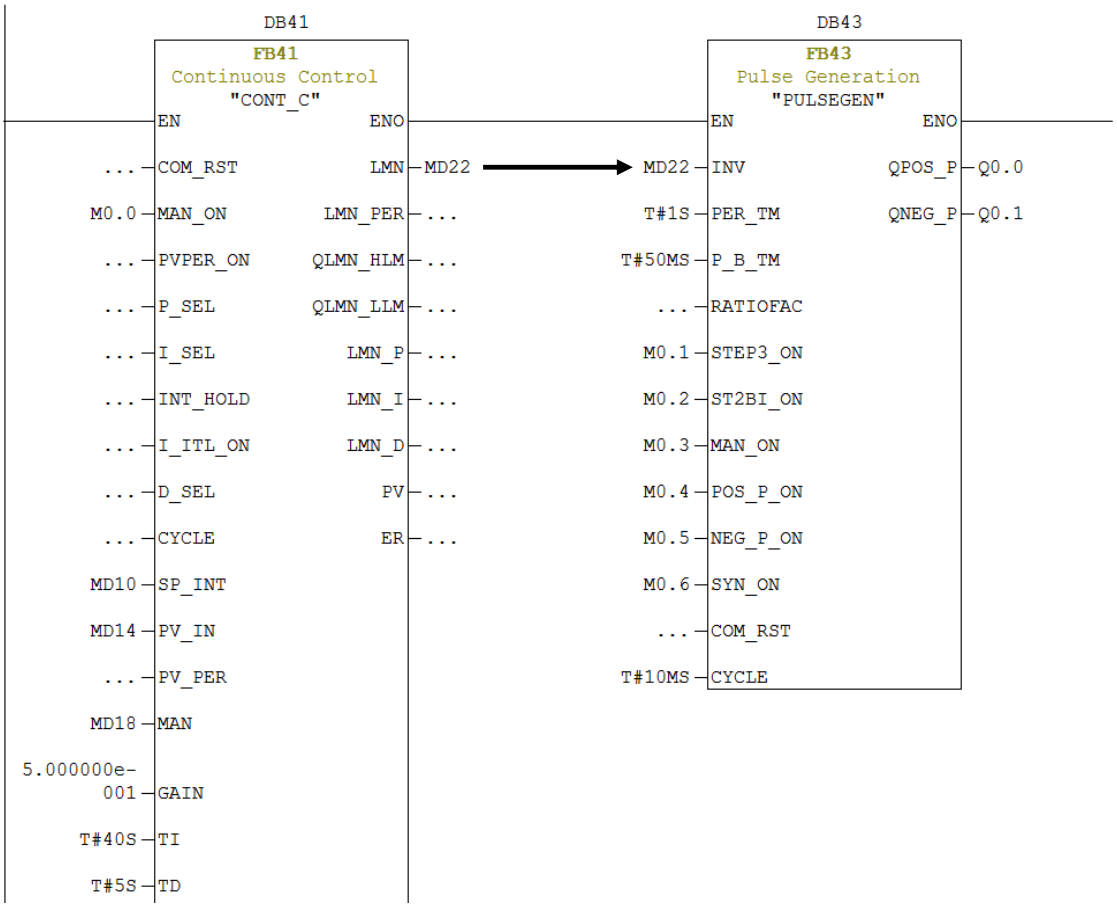




در شکل فوق نحوه تبدیل یک سیگنال پیوسته به پالس را ملاحظه می کنید. به عنوان مثال زمانی که مقدار سیگنال پیوسته برابر 30 درصد می باشد، خروجی پالس به اندازه 3 سیکل فراخوانی FB43 یک می باشد. یا اگر مقدار سیگنال 50 درصد باشد، خروجی پالس به اندازه 5 سیکل فراخوانی FB43یک می باشد. پس در نتیجه هر چقدر مقدار درصد LMN خروجی FB41 بیشترشود، مدت زمان یک بودن پالس نیز بیشتر می شود. در واقع در این حالت پالس نیز به سمت یک شدن کامل در طول مدت زمان تعیین شده برای پارامتر PER_TM می رود. از ترکیب FB41+FB43 می توان جهت کنترل دو مرحله ای یا سه مرحله ای با دو عملگر به صورت مجزا استفاده کرد. این فانکشن نیز دو خروجی جهت تولید پالس دارد.

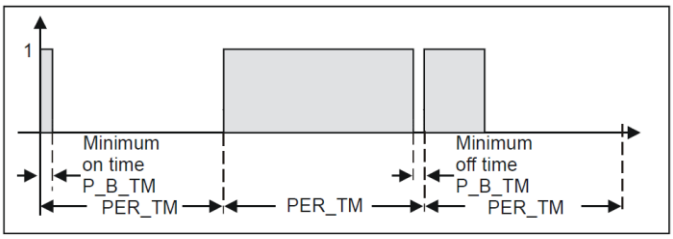


فانکشن های FB41+FB43



همانطور که در شکل فوق ملاحظه می کنید، خروجی LMN تابع FB41 به عنوان ورودی تابع FB43 (INV)می باشد.

INV : ورودی مربوط به سیگنال پیوسته



PER_TM : زمان پریود پالس خروجی

P_B_TM : حداقل طول پالس خروجی/حداقل زمان بین دو پالس(Minimum Pulse/Break Time)

STEP3_ON : ورودی فعال ساز حالت کنترل به صورت سه مرحله ای

ST2BI_ON : ورودی فعال ساز حالت کنترل به صورت دو مرحله ای

MAN_ON : ورودی مربوط به تغییر مدکاری (Automatic/Manual)

POS_P_ON : فرمان 1 شدن پالس خروجی(QPOS_P) در حالت Manual

Novin Fanavaran Arya Co.

NEG_P_ON : فرمان 1 شدن پالس خروجی(QNEG_P) در حالت Manual

SYN_ON : ورودی مربوط به فعال کردن عملیات سنکرون سازی

COM_RST : ورودی ریست کردن فانکشن

CYCLE : زمان فراخوانی بلاک FB43

QPOS_P : خروجی پالس مثبت

QNEG_P : خروجی پالس منفی

مدهای کاری FB43

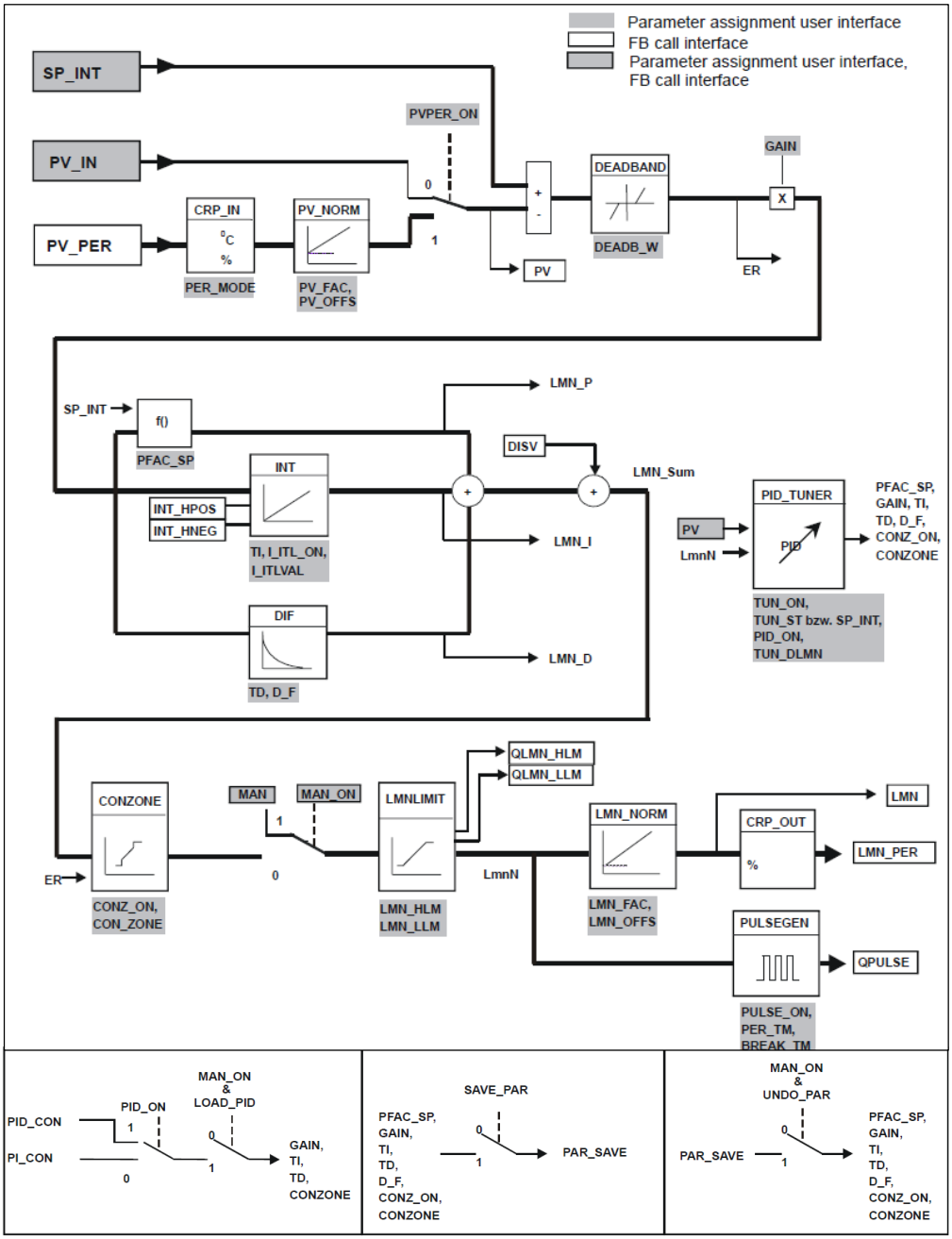
Mode \ Switch	MAN_ON	STEP3_ON	ST2BI_ON
Three-step control	FALSE	TRUE	Any
Two-step control with bipolar control range (-100 % to +100 %)	FALSE	FALSE	TRUE
Two-step control with monopolar control range (0 % ... 100 %)	FALSE	FALSE	FALSE
Manual mode	TRUE	Any	Any

وضعیت خروجی های FB43 در حالت کنترل ۳ مرحله ای برای کنترل دمای یک مخزن به صورت جدول شکل زیر می باشد.

Output signal \ Actuator	Heat	Off	Cool
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

(TCONT_CP) FB58

این فانکشن مختص کنترل لوپ دما (سرمایشی/گرمایشی) با قابلیت و امکانات بیشتر نسبت به فانکشن های بررسی شده (FB41+FB43)می باشد. FB58 برای کنترل پیوسته یا دو مرحله ای(پالسی) دما استفاده می شود.

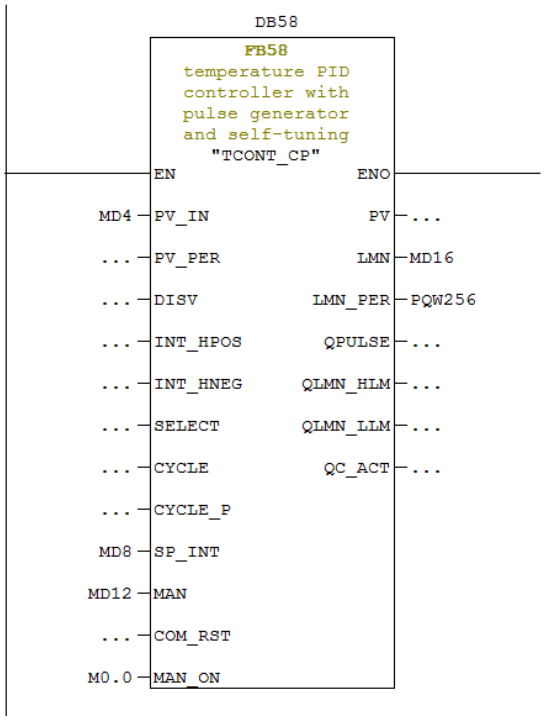


Network 1: Title:

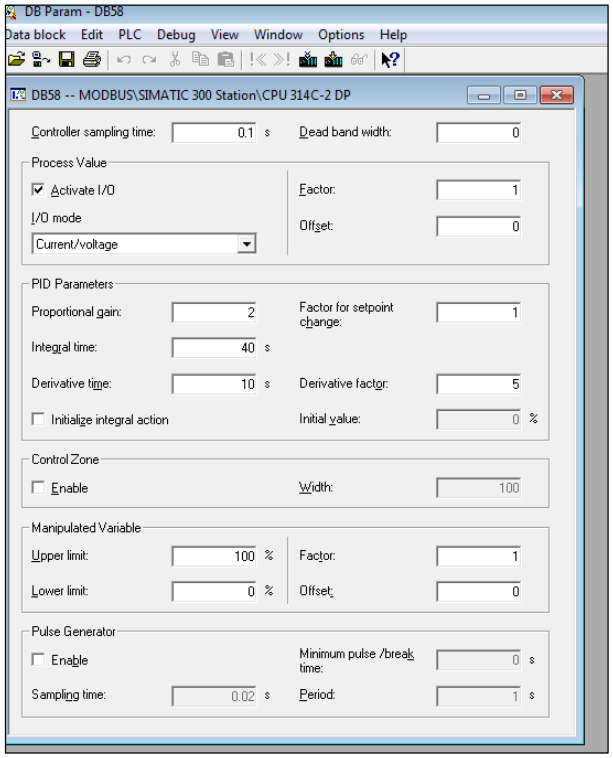
Comment:

فانکشن FB58

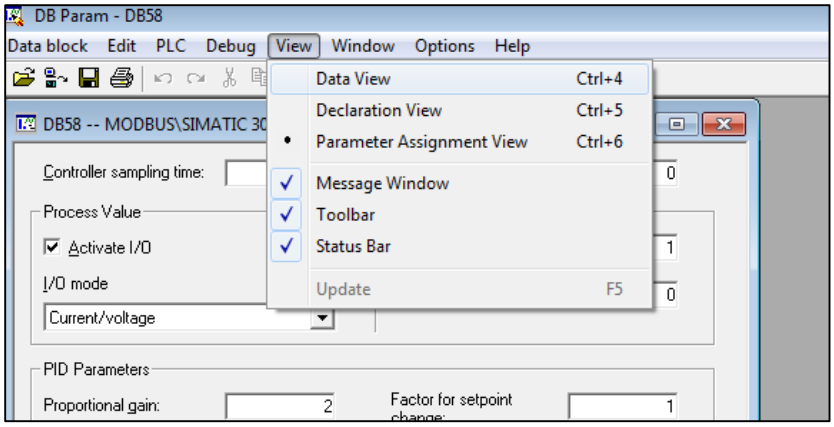
این فانکشن دارای پارامترهای دیگری نیز می باشد که با مراجعه به DB اختصاص داده شده در دسترس می باشند. با دابل کلیک بر روی DB مربوطه، پنجره شکل زیر نمایان می شود.



توسط این محیط می توان برخی از پارامترهای FB58 را تنظیم و دانلود نمود. در واقع با دانلود کردن این محیط به CPU، مقادیر دیتابلاک تغییر می کند.



جهت دسترسی به کلیه پارامترهای FB58، حالت نمایش DB را در حالت Data View قرار می دهیم.



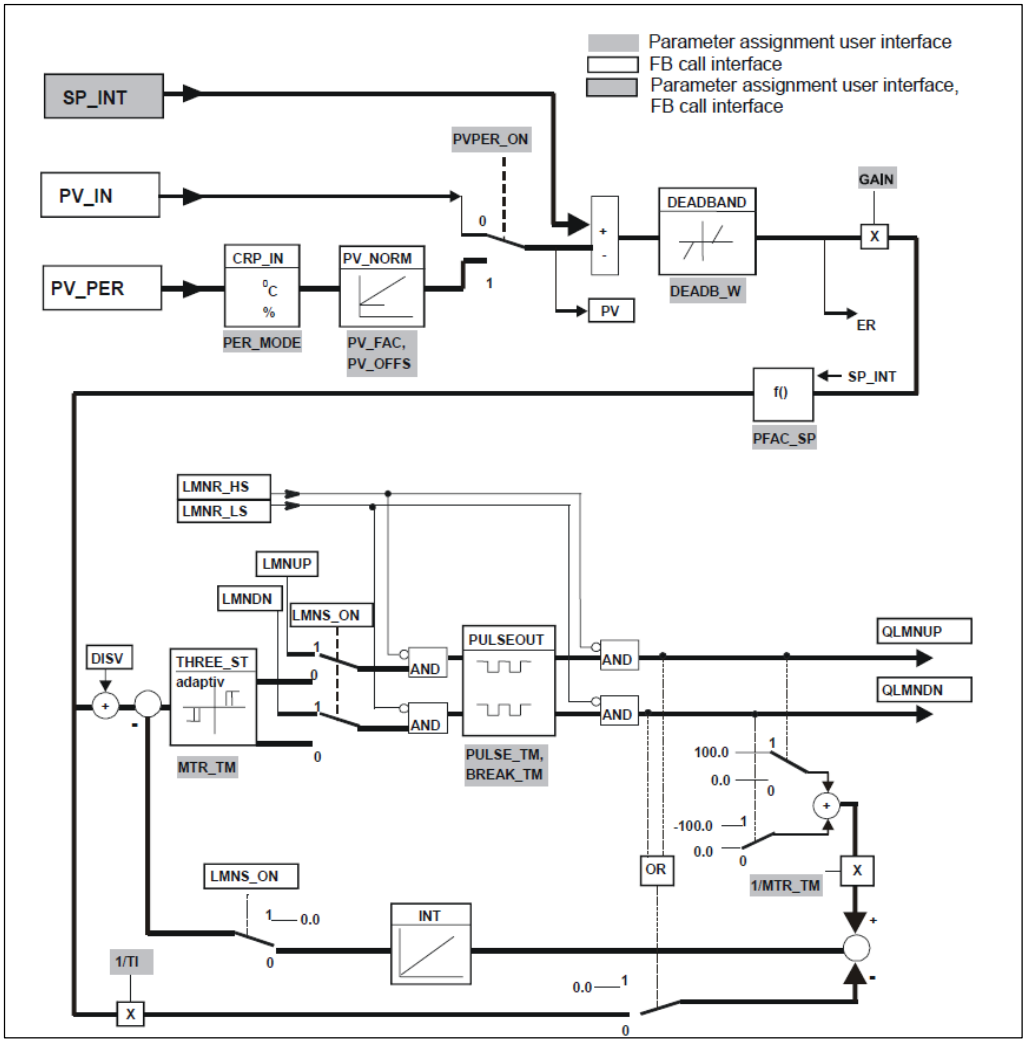
با کلیک بر روی گزینه Data View، تمامی پارامترهای DB نمایش داده می شوند.

Data block Edit PLC Debug View Window Help

	Address	Declaration	Name	Type	Initial value	Actual value	Comment
1	0.0	in	PV_IN	REAL	0.000000e+000	0.000000e...	process variable in
2	4.0	in	PV_PER	INT	0	0	process variable peripherie
3	6.0	in	DISV	REAL	0.000000e+000	0.000000e...	disturbance variable
4	10.0	in	INT_HPOS	BOOL	FALSE	FALSE	integral action hold in positive direction
5	10.1	in	INT_HNEG	BOOL	FALSE	FALSE	integral action hold in negative direction
6	12.0	in	SELECT	INT	0	0	selection of call PID and pulse generator
7	14.0	out	PV	REAL	0.000000e+000	0.000000e...	process variable
8	18.0	out	LMN	REAL	0.000000e+000	0.000000e...	manipulated variable
9	22.0	out	LMN_PER	INT	0	0	manipulated variable peripherie
10	24.0	out	QPULSE	BOOL	FALSE	FALSE	output pulse signal
11	24.1	out	QLMN_HLM	BOOL	FALSE	FALSE	high limit of manipulated variable reached
12	24.2	out	QLMN_LLM	BOOL	FALSE	FALSE	low limit of manipulated variable reached
13	24.3	out	QC_ACT	BOOL	TRUE	TRUE	next cycle, the continuous controller is working
14	26.0	in_out	CYCLE	REAL	1.000000e-001	1.000000e...	sample time of continuous controller [s]
15	30.0	in_out	CYCLE_P	REAL	2.000000e-002	2.000000e...	sample time of pulse generator [s]
16	34.0	in_out	SP_INT	REAL	0.000000e+000	0.000000e...	internal setpoint
17	38.0	in_out	MAN	REAL	0.000000e+000	0.000000e...	manual value
18	42.0	in_out	COM_RST	BOOL	FALSE	FALSE	complete restart
19	42.1	in_out	MAN_ON	BOOL	TRUE	TRUE	manual operation on
20	44.0	stat	DEADB_W	REAL	0.000000e+000	0.000000e...	dead band width
21	48.0	stat	I_ITLVAL	REAL	0.000000e+000	0.000000e...	initialization value of the integral action
22	52.0	stat	LMN_HLM	REAL	1.000000e+002	1.000000e...	manipulated variable high limit
23	56.0	stat	LMN_LLM	REAL	0.000000e+000	0.000000e...	manipulated variable low limit
24	60.0	stat	PV_FAC	REAL	1.000000e+000	1.000000e...	process variable factor
25	64.0	stat	PV_OFFS	REAL	0.000000e+000	0.000000e...	process variable offset
26	68.0	stat	LMN_FAC	REAL	1.000000e+000	1.000000e...	manipulated variable factor
27	72.0	stat	LMN_OFFS	REAL	0.000000e+000	0.000000e...	manipulated variable offset
28	76.0	stat	PER_TM	REAL	1.000000e+000	1.000000e...	period time [s]
29	80.0	stat	P_B_TM	REAL	0.000000e+000	0.000000e...	minimum pulse/break time [s]
30	84.0	stat	TUN_DLMN	REAL	2.000000e+001	2.000000e...	delta manipulated variable for process excitation
31	88.0	stat	PER_MODE	INT	0	2	periphery mode: 0=standard, 1=climate, 2=voltage/current
32	90.0	stat	PVPER_ON	BOOL	FALSE	TRUE	process variable periphery on
33	90.1	stat	I_ITL_ON	BOOL	FALSE	FALSE	initialization of the integral action on
34	90.2	stat	PULSE_ON	BOOL	FALSE	FALSE	pulse generator on
35	90.3	stat	TUN_KEEP	BOOL	FALSE	FALSE	keep tuning on (common tuning with other controllers)
36	92.0	stat	ER	REAL	0.000000e+000	0.000000e...	error signal
37	96.0	stat	LMN_P	REAL	0.000000e+000	0.000000e...	proportionality component
38	100.0	stat	LMN_I	REAL	0.000000e+000	0.000000e...	integral component
39	104.0	stat	LMN_D	REAL	0.000000e+000	0.000000e...	derivative component
40	108.0	stat	PHASE	INT	0	0	phase of self tuning (0..7)
41	110.0	stat	STATUS_H	INT	0	0	status heating of self tuning
42	112.0	stat	STATUS_D	INT	0	0	status controller design of self tuning

FB59 (TCONT_S)

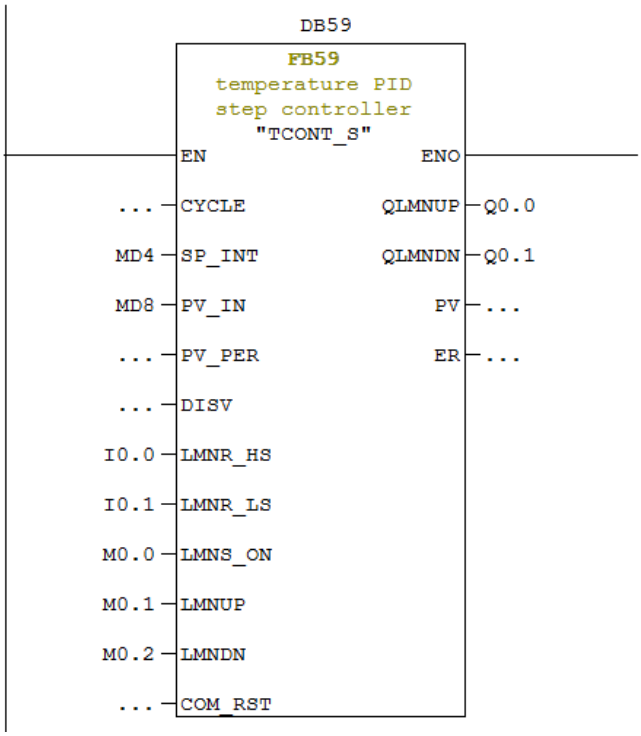
این فانکشن نیز مختص کنترل لوپ دما به صورت PI، با قابلیت و امکانات بیشتر نسبت به فانکشن FB42 در حالت کنترل سه مرحله ای می باشد. FB59 برای کنترل یک عملگر سه حالته همانند ولوهای موتوری(Open/Close/Off) استفاده می شود.



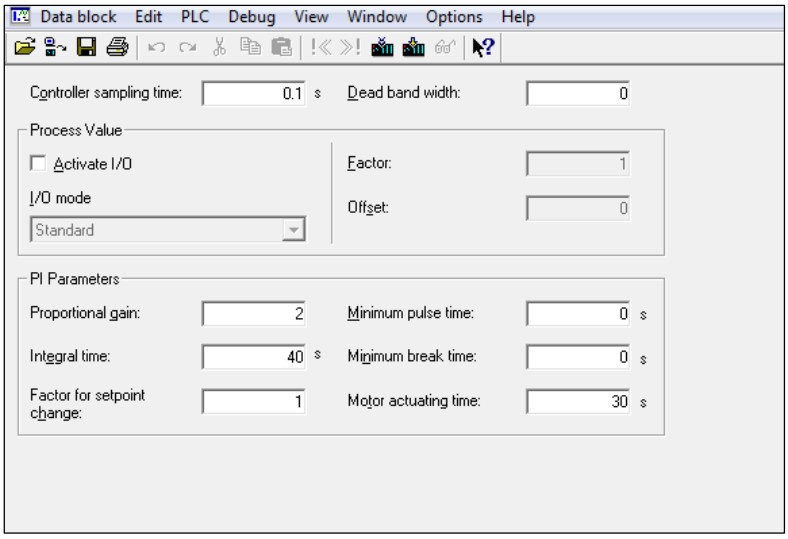
فانکشن FB59

Network 1 : Title:

Comment:



توسط محیط شکل زیر می توان پارامترهای FB59 را تنظیم و دانلود نمود.



لیست فانکشن های سیستمی

Overview of the System Functions (Part 1)

Function Group	Function	Block	S7-300	S7-400
Copy and Block Functions	Blockmove	SFC 20	X	X
	Preset field	SFC 21	X	X
	Generate DB	SFC 22	X	X
	Delete DB	SFC 23	-	X
	Test DB	SFC 24	-	X
	Compress	SFC 25	-	X
	Substitute value in Accu 1	SFC 44	X[1)]	X
Program Control	Multicomputing interrupt	SFC 35	-	X[2)]
	Trigger cycle time	SFC 43	X	X
	Stop state	SFC 46	X	X
	Delay (Wait)	SFC 47	X[1)]	X
Handling the Clock	Set clock time	SFC 0	X	X
	Read clock time	SFC 1	X	X
	Synchronize	SFC 48	-	X
Operating Hours Counter	Set the counter	SFC 2	X[1)]	X
	Start and stop	SFC 3	X[1)]	X
	Read out	SFC 4	X[1)]	X
	Read system time	SFC 64	X	X

Copy Functions and Block Functions

- SFC 20 copies the contents of a memory area (source) into another memory area (destination)
- SFC 21 fills a memory area (destination field) with the contents of a specified memory area (source field)
- SFC 22 creates a DB without preset values in the work memory
- SFC 23 deletes a DB in the work memory and possibly in the load memory.
- SFC 24 determines if a DB is present in the work memory (with length).
- SFC 25 compresses the memory. When blocks are corrected, gaps occur in the memory that are removed during compressing.
- SFC 44 (call in OB 122) saves a substitute value for a faulty input module in the Accu

Program Control

- SFC 35 triggers, with multicomputing, the synchronized start of OB 60 on all CPUs
- SFC 43 starts CPU scan cycle monitoring anew
- SFC 46 brings the CPU to the Stop state
- SFC 47 implements wait times in the user program up to 32767 µs

Handling the Clock

- SFC 0 sets the date and time-of-day for the real-time clock on the CPU
- SFC 1 reads the current CPU date and time-of-day
- SFC 48 synchronizes all slave clocks present on a bus segment. The CPU with the call must be assigned parameters as the master clock

Operating Hours Counter

The CPU has a specific number of operating hours counters with which you can record the operating duration of the operating equipment.

- SFC 2 sets the operating hours counter to a specified value
- SFC 3 starts and stops the operating hours counter
- SFC 4 reads the current number of operating hours and the status
- SFC 64 reads the CPU's system time. The system time is a free-running time counter that counts up every 10 ms (S7-300) or 1 ms (S7-400)..

Overview of the System Functions (Part 2)

Function Group	Function	Block	S7-300	S7-400
Transfter data records	Write dynamic parameters Write defined parameters Assign parameters to modules Write data record Read data record	SFC 55 SFC 56 SFC 57 SFC 58 SFC 59	X X X X X	X X X X X
Time interrupts	Set Cancel Activate Scan	SFC 28 SFC 29 SFC 30 SFC 31	X1) X1) X1) X1)	X X X X
Delay interrupt	Start Cancel Scan	SFC 32 SFC 33 SFC 34	X1) X1) X1)	X X X
Synchronous errors	Mask errors Demask errors Read status register	SFC 36 SFC 37 SFC 38	X X X	X X X
Interrupt errors and Asynchronous errors	Cancel new interrupts Enable new interrupts Delay new interrupts Enable higher priority interrupts	SFC 39 SFC 40 SFC 41 SFC 42	X X X X	X X X X

Transfer Data Records

There is a system data area with parameter and diagnostics data for the parameter-assignable modules. This area contains data records from 0 to 255 that can be read or written.

- SFC 55 transfers the dynamic parameters to the addressed module. The parameters in the SDB on the CPU are not overwritten.
- SFC 56 transfers the parameters (data record RECNUM) to the module.
- SFC 57 transfers all data records from the SDB to the module.
- SFC 58 transfers the RECORD data record to the module.
- SFC 59 reads out the RECORD data record from the module.

Time Interrupts

The blocks are used for time-of-day controlled processing (OB 10 to 17). You can determine the starting point either with the STEP 7 software or with the following system functions.

- SFC 28 sets the start date and time-of-day of a time-of-day controlled OB.
- SFC 29 deletes the start date and time-of-day of an OB (OB 10 to OB 17).
- SFC 30 activates the specified time interrupt OB.
- SFC 31 scans the status of a time interrupt OB.

Delay Interrupt

- SFC 32 starts in a delayed manner a delay interrupt (OB 20 to 27).
- SFC 33 cancels a delay interrupt.
- SFC 34 scans the state of a delay interrupt.

Synchronous Errors

- SFC 36 masks a synchronous error, i.e. a faulty instruction does not lead to the call of the associated synchronous error OB.
- SFC 37 demasks the synchronous error
- SFC 38 reads out the Error Register.

Interrupt and Asynchronous Errors

- SFC 39 disables the processing of interrupt and asynchronous error events.
- SFC 40 enables again the processing of interrupt and asynchronous errors.
- SFC 41 delays the processing of interrupt and asynchronous errors.
- SFC 42 enables again the processing of delayed interrupt and asychronous errors

Overview of the System Functions (Part 3)

Function Group	Function	Block	S7-300	S7-400
System Diagnostics	Read start info. Read partial system status list Write diagnostics buffer	SFC 6 SFC 51 SFC 52	- X X	X X X
Process image, I/O area	Update PII inputs Update PIQ outputs Set bit field in I/O Reset bit field in I/O	SFC 26 SFC 27 SFC 79 SFC 80	- - - -	X X X X
Addressing of Modules	Determine logical address Determine slot Determine all logical addresses	SFC 5 SFC 49 SFC 50	- X X	X X X
Distributed I/O	Trigger hardware interrupt Synchronize DP Slaves Read diagnostic interrupt Read user data Write user data	SFC 7 SFC 11 SFC 13 SFC 14 SFC 15	1) 1) 1) 1) 1)	1) 1) 1) 1)
Global Data Communication	Send GD package Receive GD package	SFC 60 SFC 61	- -	X X

System Diagnostics	• SFC 6 reads out the start information of the OB called last and the start-up OB • SFC 51 reads out a part of the system status list. The list contains: system data, diagnostics state data, diagnostics data, and the diagnostics buffer. • SFC 52 writes a user entry in the diagnostics buffer
Process Image, I/O Area	• SFC 26 updates the entire or a partial process-image input table • SFC 27 transfers the entire or a partial process-image to the output modules • SFC 79/ 80 are used to set and reset bit fields in the I/O area in conjunction with the Master Control Relay function
Addressing of Modules	• SFC 5 provides the logical address for a geographical address • SFC 49 determines the geographical address from the logical address • SFC 50 provides all the logical addresses of a module
Distributed I/O	• SFC 7 triggers a hardware interrupt at the DP master. The SFC 7 is called in the user program of an intelligent slave (CPU 315-2DP) • SFC 11 synchronizes one or several groups of DP slaves • SFC 13 reads the diagnostics data of a DP slave • SFC 14 reads consistent data from a DP slave. • SFC 15 writes consistent data to a DP slave
Global Data Communication	The global data are transferred cyclically (such as every 8th cycle) without using the SFC. With the help of the SFC 60 and 61 system function, sending and receiving data packages in the user program can be triggered. • SFC 60 sends a global data package • SFC 61 receives a global data package

Overview of the System Functions (Part 4)

Function Group	Function	Block	S7-300	S7-400
Data Exchange using SFB, configured connection	Scan state	SFC 62	-	X
	Uncoordinated send	SFB 8	-	X
	Uncoordinated receive	SFB 9	-	X
	Send block	SFB 12	-	X
	receive block	SFB 13	-	X
	Read data from remote CPU	SFB 14	-	X
	Write data to remote CPU	SFB 15	-	X
	Send to printer	SFB 16	-	X
	Carry out complete restart	SFB 19	-	X
	Stop state	SFB 20	-	X
	Carry out restart	SFB 21	-	X
	Scan device status	SFB 22	-	X
	Receive device status	SFB 23	-	X
Data Exchange using SFC, non-configured connection	Send data externally	SFC 65	1)	1)
	Recevie data externally	SFC 66	1)	1)
	Read data externally	SFC 67	1)	1)
	Write data externally	SFC 68	1)	1)
	Cancel connection externally	SFC 69	1)	1)
	Read data internally	SFC 72	1)	1)
	Write data internally	SFC 73	1)	1)
	Cancel connection internally	SFC 74	1)	1)

1) only for innovated CPUs

Data Exchange using SFBs

The SFBs are used to exchange data and to manage programs using configured connections. Depending on whether SFB calls are necessary at only one communications partner or at both, reference is made to one-sided or two-sided communication. SFBs exist only in the S7-400 operating system.

- SFC 62 determines the state of a local SFB instance and the state of the associated connection
- SFB 8 sends data to a remote partner without coordination
- SFB 9 is the counterpart to the SFB 8
- SFB 12 sends data (up to 64 KByte) to the remote partner with an ack.
- SFB 13 receives data from the remote partner with an acknowledgement
- SFB 14 reads data from a remote CPU (one-sided communication)
- SFB 15 writes data to a remote CPU (one-sided communication)
- SFB 16 sends data with formatting to a remote printer
- SFB 19 triggers a complete restart at the remote partner
- SFB 20 transfers the remote partner to the STOP state.
- SFB 21 carries out a restart at a remote partner
- SFB 22 provides the device status (operating state, error information) of the remote partner.
- SFB 23 receives the device status of a remote partner.

Data Exchange using SFCs

This communication - also referred to as basic communication - is implemented with S7- 300 as well as with S7-400. In comparison to SFB communication the following differences emerge:

- no connection configuration necessary
- no instance data blocks required
- maximum user data length 76 bytes
- dynamic connection configuration
- communication via MPI or K bus

Overview of the System Functions (Part 5)

Function Group	Function	Block	S7-300	S7-400
Integrated Closed-loop Control	Continuous control Step control Pulse shaping	SFB 41 SFB 42 SFB 43	3) 3) 3)	- - -
Plastics Technology	Call assembler block	SFC 63	1)	-
Integrated Functions	High speed counter Frequency meter A/B counter Positioning	SFB 29 SFB 30 SFB 38 SFB 39	2) 2) 3) 3)	- - - -
IEC Timer and IEC Counter	Pulse On delay Off delay Count up Count down Count up/down	SFB 3 SFB 4 SFB 5 SFB 0 SFB 1 SFB 2	X X X X X X	X X X X X X
Block Referenced Messages	Message without acknowledgement Message with acknowledgement Message with 8 accompanying values Message without accompanying values Send archive data Disable archive data Enable messages	SFB 36 SFB 33 SFB 35 SFB 34 SFB 37 SFC 10 SFC 9	- - - - - - -	X X X X X X X

1) only for CPU 614 2) only for CPU 312 IFM 3) only for CPU 314IFM

Integrated Closed-Loop Control

These blocks will be integrated in a later CPU version.

Plastics Technology

For the CPU 614 (S7-300), individual blocks can be created in the “C” language. The SFC 63 system function is used to call such blocks.

Integrated Functions

These blocks exist only for the CPU 312 IFM (S7-300).
You will find a description in the *Integrated Functions* manual.

- SFB 29 counts pulses on the integrated CPU inputs.
- SFB 30 is used to measure frequencies using the integrated inputs.

IEC Timer and Counter	This makes timer and counter functions available that correspond to the IEC 1131-3 standard. The remaining timer and counter functions are implemented as for SIMATIC S5, due to compatibility reasons. The IEC timers and counters differ in a larger value range for the timer and counter values.
Block-Referenced Messages	These blocks are used to implement message systems for HMI systems, such as process control systems. The messages are generated in the S7-CPU with this procedure and the respective messages including process variables are sent to the logged-on display devices. A central acknowledgement concept is used. That is, when you acknowledge a message on the display device, a response is sent to the originating CPU. The information is distributed to all logged-on users from the CPU. The messages are triggered for an edge change on the signal input.

- اکبر اویسی فر
- کارشناس ارشد برق-الکترونیک
- متخصص سیستم های اتوماسیون صنعتی
- ایمیل:Akb_Oveisifar@yahoo.com

ردیف	عنوان
۱	تسلط بر PLC های S5 ، S7-200 ، S7-300 و S7-400 ، S7-1200 ، S7-1500 (برنامه نویسی و عیب یابی)
۲	تسلط بر PLC های شرکت Allen Bradley (SLC 500، ControlLogix ، CompactLogix) STUDIO 5000
۳	تسلط بر سیستم DCS شرکت زیمنس (PCS7)-S7-400H ، S7-400FH
۴	تسلط بر سیستم های مانیتورینگ و طراحی و راه اندازی آنها توسط نرم افزارهای WinCC ، Protool و WinCC Flexible
۵	تسلط بر شبکه های صنعتی Profibus، Profinet و Ethernet (نصب ، برنامه نویسی و عیب یابی)
۶	تسلط بر نصب، برنامه نویسی و راه اندازی ET های زیمنس در شبکه Profibus و Profinet
۷	تسلط بر نصب، برنامه نویسی و راه اندازی درایوهای ABB ، SIEMENS، SEW و LENZE در شبکه Profibus
۸	تسلط بر سرو درایوهای YASKAWA ، SIEMENS ، MITSUBISHI
۹	تسلط کامل بر نرم افزارهای طراحی تابلوهای فرمان ، قدرت و کنترل(EPLAN ، CAD)
۱۰	تسلط کامل بر تجهیزات ابزاردقیق(سنسورهای دما، فشار، لودسل،کنترل سطح و...)،نصب و برنامه نویسی توسط PLC
۱۱	تسلط بر طراحی، اجراء و عیب یابی مدارات پنیوماتیک ، هیدرولیک
۱۲	توانایی راه اندازی انواع لوپ های کنترلی PID توسط PLC
۱۳	توانایی از بین بردن پسوردهای نرم افزاری و سخت افزاری PLC
۱۴	توانایی برنامه نویسی و طراحی بردهای صنعتی توسط میکروکنترلرهای AVR و ARM ، FPGA
۱۵	تسلط بر برنامه نویسی تراشه های FPGA و کار با نرم افزارهای XILINX ISE ، QUARTUS و ModelSim

برخی از کتاب های تالیف شده

شرکت فنی و مهندسی

نوین فن آوران آریا

(مشاور، طراح و مجری پروژه های اتوماسیون صنعتی)

Consulting, designing and executing of Industrial Automation Projects

SIEMENS

Allen-Bradley

YOKOGAWA

EMERSON

OMRON

MITSUBISHI

- طراحی و ساخت تابلوهای توزیع، قدرت و کنترل
- طراحی و برنامه نویسی PLC و DCS مطابق استانداردهای روز دنیا
- طراحی و برنامه نویسی سیستم های Redundant و Fail Safe
- طراحی سیستم های SCADA کاملا حرفه ای
- طراحی و راه اندازی شبکه های صنعتی در سطوح مختلف
- نصب و راه اندازی انواع درایوهای صنعتی
- تعمیرات تخصصی بردهای صنعتی
- ارائه خدمات مشاوره، بهینه سازی خطوط تولید و نگهداری
- تامین کننده تجهیزات اتوماسیون صنعتی و ابزاردقیق با قیمت رقابتی
- آموزش دوره های تخصصی اتوماسیون صنعتی
- تالیف و چاپ کتاب های تخصصی اتوماسیون صنعتی

Tel:021-44451668 0919-698-7875 www.FanavaranAria.Com

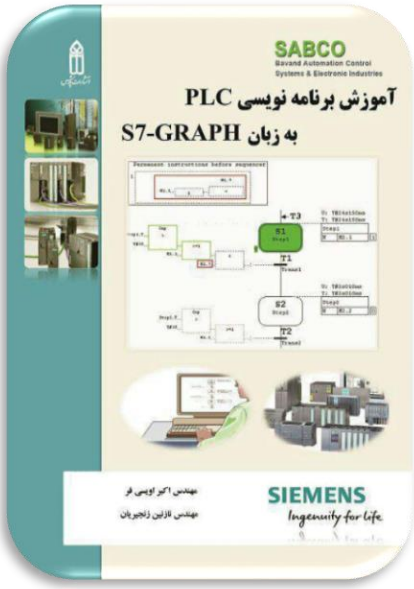
شرکت **نوین فن آوران آریا** مجری کلیه پروژه های اتوماسیون صنعتی، ابزار دقیق و اسکادا در سطح کشور می باشد. این شرکت با اتکاء به تجربه ۲۰ ساله موسسین در زمینه ساخت تابلوهای کنترل، تابلوهای محلی RIO ، تابلوهای فشار ضعیف و متوسط و همچنین طراحی و اجرای پروژه های اتوماسیون صنعتی و ابزاردقیق، بنیان نهاده شده است. تیم اجرایی این شرکت با بهره گیری از کادر مهندسی توانمند و با تجربه، توانایی اجرای کلیه پروژه های EPC در صنایع نفت و گاز، خودروسازی، شیشه، صنایع غذایی و ... را به صورت کاملا استاندارد دارا می باشد.

دوره های تخصصی اتوماسیون صنعتی

(ویژه صنایع)

واحد آموزش این شرکت با در اختیار داشتن اساتید با تجربه در عرصه علم و صنعت، اقدام به برگزاری دوره های تخصصی اتوماسیون صنعتی SIEMENS و Allen Bradley نموده است. تمامی دوره های این شرکت از لحاظ محتوا و کار عملی مطابق با استاندارد بوده و در هر یک از دوره ها شرکت کننده ها به طور کاملا تخصصی با سیستم های کنترلی آشنا می شوند. موسسین این شرکت در سال های فعالیت خود، افتخار برگزاری دوره های آموزشی برای بسیاری از مهندسین و تکنسینهای شرکت ها و صنایع مختلف در سطح کشور را دارا می باشند. مدرسین این شرکت مورد تایید شرکت نفت، گاز و صنایع خودرو سازی می باشند.

- Studio 5000-ControlLogix
- Studio 5000_PanelView 5000
- RSLogix500_SLC500
- Factory Talk View Studio
- PLC- S5
- S7-400F/FH
- PCS7
- S7-300,400
- S7-400F/FH
- S7-1200(TIA)
- S7-300,400(TIA)
- WinCC
- Protool
- WinCC Flexible
- Ethernet - Profibus -Profinet
- SINAMICS Servo Drive
- Micromaster Drive
- Simotion Scout
- S7 PID
- Eplan

تالیف کتاب های تخصصی اتوماسیون صنعتی

این شرکت جهت پیشبرد اهداف آموزشی خود اقدام به چاپ کتاب های تخصصی اتوماسیون و همچنین توزیع مجموعه های آموزشی جهت استفاده مهندسین و دانشجویان نموده است. برخی از کتاب های تالیف شده توسط دپارتمان آموزش این شرکت به شرح زیر می باشند.