راهنمای جامع

LOGO!

با مثالهای کاربردی

قابل استفاده برای دانشجویان و هنر جویان رشته برق و کنترل

ویراست اول

به همراه DVD شامل:

- آخرین نسخه نرم افزار LOGO V 6.0
- فایلهای آموزشی به صورت مالتی مدیا مختص مثال‌های کتاب
- فایل‌های اجرایی مثال‌های کتاب
- فیلم‌های آموزشی سنسورها
- اسلایدهای آموزشی جهت تدریس
- سیمیلاتور LOGO V 3.0

تالیف: مهندس احمد طهماسبی

پیشگفتار مولف

کنترل کننده های منطقی قابل برنامه ریزی PLC نقش بسیار مهمی در اتوماسیون صنایع بر عهده دارند و در اکثر مراکز صنعتی جدید از آنها استفاده می شود و امروزه در کارخانجات صنعتی، PLC ها جایگزین بسیار مناسبی برای مدارات رله کنتاکتوری گردیده اند. به کارگیری PLC ها در اکثر پروژه های صنعتی همانند دستگاههای تزریق، کوره ها، دستگاه های چاپ، سیستمهای بالابرو ... نمونه هایی از کاربرد و ضرورت آشنایی با این سیستمها می باشد.

به دلیل نبودن مرجعی کامل که پاسخگوی سوالات فراوان دانشجویان، متخصصین، تکنسینها و کارگران ماهر در این زمینه باشد، اینجانب حقیر را بر آن داشت تا با تهیه مجموعه ای کامل در زمینه *miniPLC* لوگو راه را برای کاربران این سیستمها هموار سازم.

با توجه به اینکه اینجانب در مراکز صنعتی و آموزشی مشغول تدریس می باشم، در تالیف این کتاب سعی نموده ام مطالب به گونه‌ای بیان شود که در حین ساده بودن، همراه با مثالهای کاربردی باشد تا خواننده ذهنیت بهتری نسبت به مطالب کتاب داشته باشد.

فصل اول کتاب در مورد آشنایی با مدارات فرمان و نحوه راه اندازی موتورهای صنعتی می باشد که توصیه می نمایم دوستانی که در مورد مدارات فرمان آشنایی چندانی ندارند حتما این فصل را مطالعه فرمایند، چرا که دید بهتری نسبت به کاربرد PLC پیدا خواهید کرد.

فصل دوم کتاب در مورد مفاهیم اولیه سنسور وکاربرد انواع سنسورهای بدون تماس می باشد. در DVD همراه کتاب فیلمهایی از کاربرد این سنسورها وجود دارند که دیدن آنها خالی از لطف نیست.

فصل سوم کتاب در مورد تاریخچه PLC، مزایای PLC و انواع PLCهای زیمنس می باشد. فصل چهارم در مورد سخت افزار لوگو و نحوه سیم بندی آن می باشد. فصل پنجم کتاب محیط نرم افزاری لوگو را شرح می دهد. البته در DVD همراه کتاب فایلهای تصویری که محتوای مثالهای فصل ششم کتاب را بیان می کنند، قرار داده ایم تا به شما کمک بیشتری در مورد آشنایی با محیط کار نرم افزار کنند. توصیه می کنم که اگر در محیط کار نرم افزار با مشکل مواجهید حتما از این فایلها استفاده کنید.

فصل ششم کتاب درباره دستورات لوگو ورژن ۶ می باشد. در نگارش این فصل سعی شده که برای هر دستور یک مثال ساده ای از کاربرد همان دستور آورده شود تا خواننده براحتی با کاربرد آن دستور آشنا شود. به دلیل ملزومات مثالها ترتیب شرح توابع لوگو رعایت نشده است. فایلهای اجرایی مثالهای این فصل در لوح فشرده این کتاب موجود می باشند که توصیه می نمایم جهت یادگیری بهتر این مثالها، فایل اجرایی آنها را درنرم افزار لوگو شبیه سازی کنید. همچنین فایلهای مالتی مدیا که مختص مثالهای این فصل است، نحوه آوردن عناصر به صفحه و شبیه سازی برنامه‌ها را نشان می‌دهند.

فصل هفتم کتاب درباره برنامه نویسی بر روی لوگو می باشد. در این فصل مختصری در مورد نحوه برنامه نویسی بر روی سخت افزار لوگو شرح داده شده است. برای یادگیری این فصل نرم افزار سیمیلاتور لوگو را که در لوح فشرده همراه کتاب موجود می باشد را بر روی کامپیوتر خود نصب کنید و گام به گام با دستورات ذکر شده، این مراحل را در این سمیلاتور انجام دهید. این سیمیلاتور در واقع شمای سه بعدی سخت افزار را نشان می دهد. از آنجایی که نسخه جدید این نرم افزار در دسترس نبود، لذا این فصل متناسب با نسخه قدیمی نگارش شده است ولی در لابلای مطالب تفاوتهای آن با نسخه جدید بیان گردیده است. یک فایل تصویری درباره یک مثال که نحوه کار با این سیمیلاتور را نشان می دهد، در DVD همراه کتاب وجود دارد. بعد از یک بار مطالعه فصل هفتم این فیلم را ببینید.

فصل هشتم کتاب در مورد مثالهای کاربردی لوگو در صنعت می‌باشد که عمدتا از کتابخانه خود زیمنس برداشته شده است. می‌توانید این مثالها را بعد از آنکه با دستورات و محیط نرم افزار آشنا شدید، تمرین کنید و نسبت به نیاز خود در صنعت آنها را بسط دهید.

بر خود لازم می دانم از مدیریت محترم انتشارات آشینا جناب آقای مهندس افشین میراب که با اشتیاق در چاپ و نشر این کتاب همکاری نمودند تشکر کنم.

در پایان از دوست خوبم آقای مهندس نیل کار که زحمت مطالعه کتاب را جهت ویراستاری تقبل نمودند، تشکر می کنم. همچنین از آقای میلاد پرتوی دانشجوی رشته برق دانشکده فنی مهندسی تبریز که همیاری صمیمانه ای در تدوین DVD همراه کتاب داشتند، کمال تشکر را دارم. از همکار عزیزم آقای رحیمی که انگیزه نگارش کتاب را برای اینجانب فراهم نمودند، سپاسگزارم.

امید است با چاپ این کتاب گامی در جهت ارتقا دانش فنی کشور برداشته باشم. هر چند در گردآوری این کتاب با مراجعه به منابع معتبر و استفاده از تجربه آموزش خود تلاش بیشتری نموده ام، اما ارائه نظرات و پیشنهادات شما بنده حقیر را در ارائه بهتر این مجموعه یاری رسان خواهد بود.

احمد طهماسبی
بهار ۱۳۹۰- تبریز
Tahmasebi_ah@yahoo.com

انتشارات آشینا-تبریز

۰۴۱۱-۵۵۳۶۱۹۶

تقدیم به:

پدر عزیزم

شمع پر فروغ وجودم که خود سوخت تا روشنی بخش راه زندگی ام شود. مویش سپیدی گرفت تا روسپید شوم. سپاسی باشد از دریای بیکران زحماتش.

مادر فداکارم

این دریای عشق و محبت، کوه صبر و استقامت، وجود پاک مقدسی که شمع هستی اش را برای روشنی وجودم افروخته و کسی که همیشه مدیون زحماتش خواهم بود.

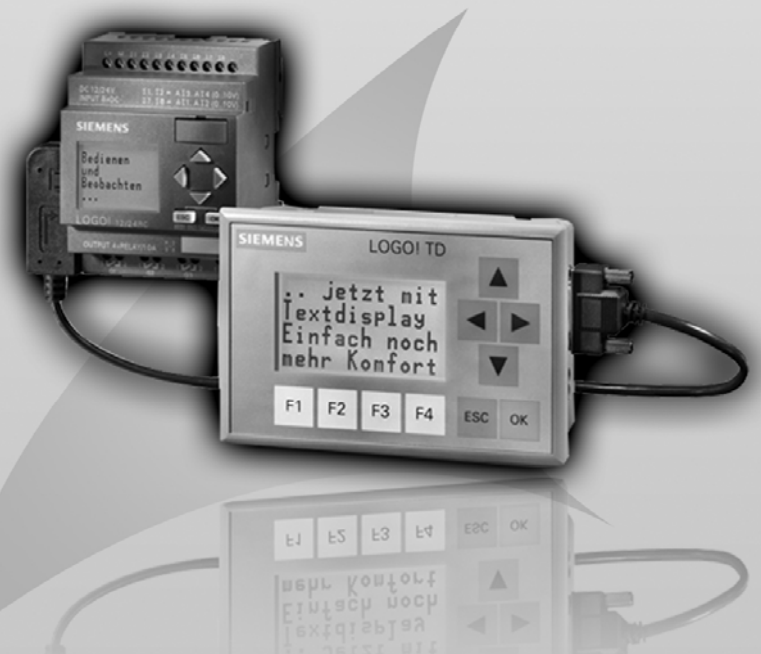
فهرست مطالب

LOGO!

۶

آشنایی با دستورات LOGO!

در این فصل با دستورات LOGO!V6.0 آشنا می شوید. برای هر دستور یک مثال ساده وجود دارد. فایلهای اجرایی مثالهای این فصل به همراه فیلمهای آموزشی که نحوه آوردن عناصر هر مثال به صفحه برنامه نویسی را نشان می دهند، در DVD همراه کتاب موجود می باشد.



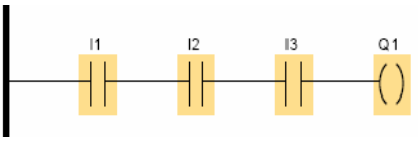
این فصل دستورات لوگو را به زبان ساده همراه با مثالهای کاربردی برای هر دستور بیان می کند. می توانید جهت سهولت یادگیری مثالهای این فصل را که بصورت فایلهای اجرایی در DVD همراه کتاب موجود می باشند را در نرم افزار اجرا کنید . همچنین جهت دیدن چگونگی شبیه سازی مثالها و نحوه آوردن عناصر به صفحه برنامه نویسی از فایلهای تصویر ی که مختص مثالهای این فصل است، استفاده کنید. لازم به ذکر است که با توجه ملزومات مثالها ترتیب شرح دستورات به طریقی که در نرم افزار چیده شده، رعایت نگردیده است.

زبانهای برنامه نویسی در لوگو

دو نوع زبان برای برنامه نویسی در لوگو وجود دارد:

LAD (ladder diagram):

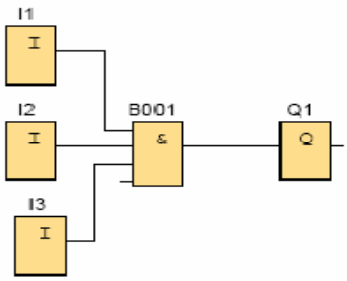
زبان نردبانی که با توجه به شبیه بودن به مدل مدار فرمان بیشتر برای طراحی مدارات فرمان مورد استفاده قرار می گیرد. نمونه‌ای از این نوع برنامه نویسی را در شکل (۶-۱) مشاهده می کنید.



شکل (۶-۱): نمونه ای از زبان برنامه نویسی LAD

FBD (function block diagram):

زبان بلوکی که برای درک بهتر مسئله مناسب است. نمونه ای از این نوع برنامه نویسی را در شکل(۶-۲) مشاهده می کنید.



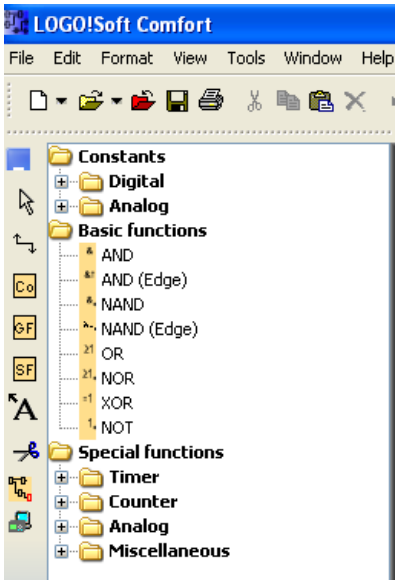
شکل (۶-۲): نمونه ای از زبان برنامه نویسی FBD

عناصر برنامه نویسی در لوگو به سه قسمت کلی تقسیم می شوند:

۱- ثابتها- (Constants)

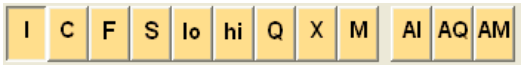
۲- توابع بیسیک (پایه)- (Basic function)

۳- توابع ویژه- (Special Function)



شکل (۶-۳): عناصر برنامه نویسی

ثابتها-Constants

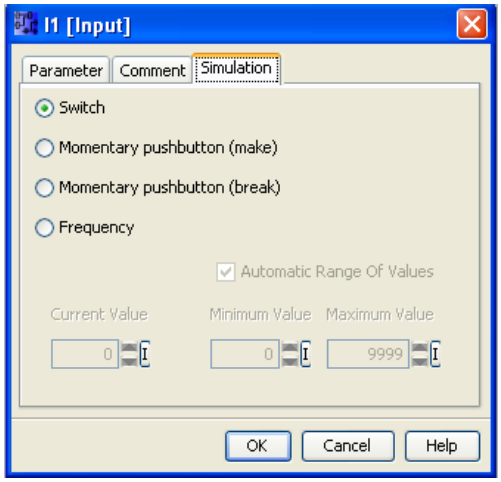


شامل ورودی و خروجی و فلگ های (پرچم) دیجیتال و آنالوگ می باشد.

الف) دیجیتال

ورودی-Input

ورودیهای دیجیتال می توانند دارای مقدار صفر و یک باشند. لوگو دارای ۲۴ ورودی دیجیتال می باشد. با دو بار کلیک کردن بـر روی هر ورودی دیجیتال در محیط برنامه نویسی پنجره مشخصات آن به شکل (۶-۴) باز می شود.



شکل (۶-۴): پنجره تنظیمات ورودی دیجیتال

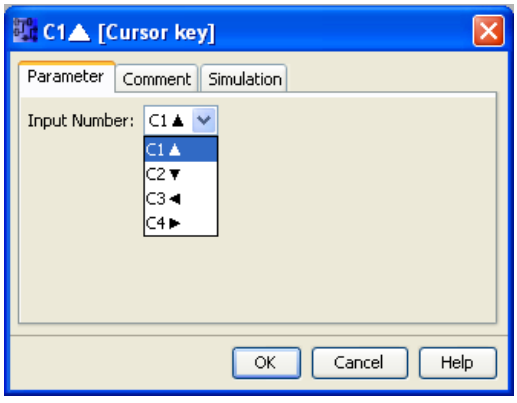
در سربرگ Parameter از این پنجره می توان شماره مربوط به ورودی را که حداکثر ۲۴ می باشد، انتخاب نمود.

در سربرگ Comment از این پنجره می توان توضیحاتی در مورد این ورودی ارائه داد.

در سربرگ Simulation از این پنجره می توان نوع این ورودی برای شبیه سازی را تعیین کرد. یک ورودی دیجیتال می تواند از نوع کلید، شستی استارت، شستی استوپ و یا فرکانسی باشد.

کلید مکان نما- Curser Key

از کلیدهای مکان نما (چهار جهته) که بر روی بعضی از لوگوها وجود دارند می توان به عنوان ورودی دیجیتال استفاده نمـود. از این کلیدها بر روی سخت افزار لوگو موقع برنامه نویسی استفاده می شود. در نرم افزار نیز می توان شماره آنها را با توجه بـه جهـت نشان داده شده، از پنجره مشخصات آن انتخاب کرد شکل (۶-۵).



شکل (۶-۵): پنجره تنظیمات کلید مکان نما

بیتهای شیفت رجیستر- Shift register bits

بیتهای شیفت رجیسترکه ۸ عدد می باشند، به همراه تابع شیفت رجیستر استفاده می شوند. بهتر است برای توضیح بیشتر به توضیح تابع شیفت رجیستر مراجعه کنید.

یک –High

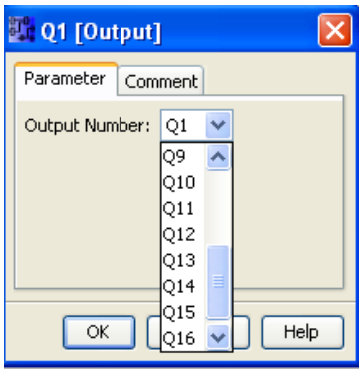
به عنوان یک ورودی ثابت با مقدار یک در داخل برنامه می باشد و نیاز به اتصال کلیدی از خارج به PLC برای این ورودی نمی‌باشد و تعداد آن محدود است.

صفر –Low

به عنوان یک ورودی ثابت با مقدار صفر در داخل برنامه می باشد و نیاز به اتصال کلیدی از خارج به PLC برای این ورودی نمی باشد و تعداد آن محدود است.

خروجی–Output

در لوگو ۱۶ خروجی دیجیتال وجود دارد که روشن و خاموش بودن آنها مبین عملکرد وصل یا قطع بودن رله ها در خروجی لوگو می باشد.



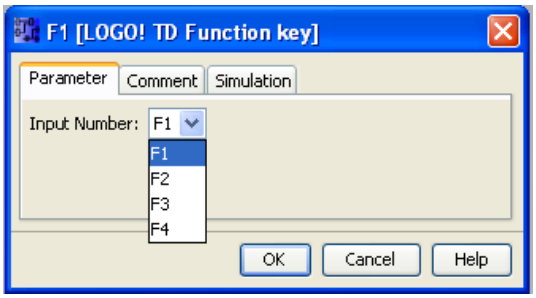
شکل (۶-۶): پنجره تنظیمات خروجی دیجیتال

اتصال دهنده های باز–Open connectors

جهت نشان دادن متن پیام در بلوک Massage text به خروجی بلوک Massage text وصل می شود. تعداد آن با تعداد خروجی دیجیتال برابر است.

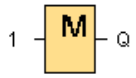
کلید تابع -LOGO! TD function key

LOGO! TD چهار کلید دارد که شما می توانید از آنها همانند ورودیهای دیگر در برنامه استفاده کنید. این کلیدها با نام های F1-F4 مشخص می باشند. بدین طریق می توانید با فشار دادن هریک از کلیدهای LOGO! TD فرمانهائی را همانند سایر ورودی‌ها صادر کنید.



شکل (۶-۷): پنجره تنظیمات کلید LOGO! TD

فلگ ها–Flags



فلگ یا پرچم جهت نشان دادن وضعیت نقاط مختلف مدار در داخل لوگو به کار می رود. عملکرد آن مثل خروجی است ولی در سخت افزار به رله های خروجی وصل نیست و می توان آنرا به عنوان رله داخلی مدار فرمان دانست. همچنین وقتی مشاهده کردید که در اتصال از خروجی تابعی به ورودی تابعی دیگر با مشکل مواجهید در مابین این دو تابع از فلگ استفاده کنید. همچنین در طراحی مدارات فرمان می توان از آن به عنوان کنتاکتور کمکی استفاده کرد.

تعداد فلگ ها به تفکیک ورژن های لوگو به صورت زیر می باشد:

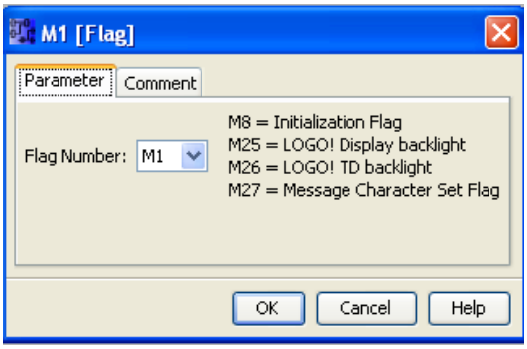
0BA6: 27 digital flags M1... M27
0BA4, 0BA5: 24 digital flags M1... M24
0BA3, 0BA2: 8 digital flags M1... M8
0BA1: 4 digital flags M1... M4
0BA0: 0 flags

در ورژن ۶ لوگو ۲۷ فلگ وجود دارد که در این میان، می توان ازفلگ M8 به عنوان راه انداز اولیه استفاده کرد. چرا که این فلگ بدون نیاز به ورودی در ابتدای شروع به کار لوگو یک پالس می فرستد که با استفاده مناسب از آن، می توان موقعی که برق قطع شده و بایستی بعد از وصل برق سیستم دوباره به راه بیافتد، برنامه را راه اندازی کرد.

فلگ M25 برای کنترل نور پس زمینه نمایشگر لوگو به کار می رود. فلگ M26 نیز برای کنترل نور پس زمینه نمایشگر LOGO!TD مورد استفاده قرار می گیرد. در واقع وقتی که این فلگها در برنامه باشند و برنامه به داخل لوگو ریخته شود نمایشگر

لوگو یا LOGO! TD تا زمانیکه این فلگ ها فعال باشند روشن خواهد بود. این فلگ ها می توانند دربرنامه نویسی در خروجی تابع Message text قرار گیرند.

فلگ M27 از بین دو مجموعه کاراکتر یکی را برای نمایش در صفحه نمایشگر انتخاب می کند. اگر ورودی M27 یک باشد، مجموعه کاراکتر ۱ و اگر صفر باشد، مجموعه کاراکتر ۲ برای نمایشگر نمایش داده خواهد شد.

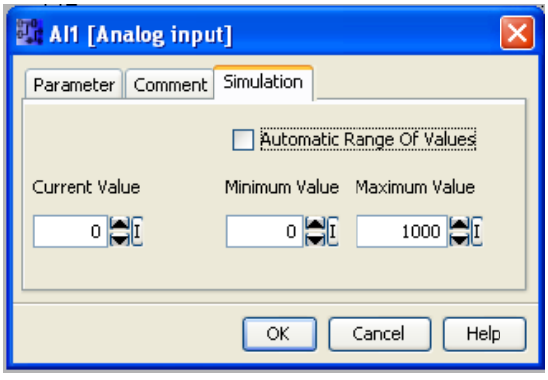


شکل (۶-۸): پنجره تنظیمات فلگ دیجیتال

آنالوگ-Analog

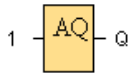
ورودیهای آنالوگ- Analog Inputs

به عنوان ورودی آنالوگ می توان از این ورودیها استفاده کرد. ۸ ورودی آنالوگ در نسخه های جدید لوگو موجود می باشند. همچنین می توان محدوده کاری آنرا تنظیم کرد به عنوان مثال(۰تا ۱۰ ولت). در لوگو ورژن ۶ ورودیهای I1،I2،I7 و I8 می توانند به عنوان ورودیهای آنالوگ به ترتیب با آدرسهای AI1،AI4،AI3 و AI2 مورد استفاده قرار گیرند. با این حال در لوگو جمعا ۸ ورودی آنالوگ داریم.



شکل (۶-۹): پنجره تنظیمات ورودی آنالوگ

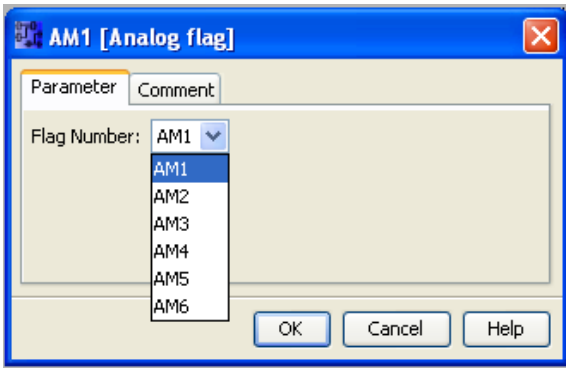
خروجیهای آنالوگ- Analog outputs



لوگو دو خروجی آنالوگ دارد که در خروجی سیگنال آنالوگ تولید می کند. سیگنال های آنالوگ رنج ۰ تا ۱۰ ولت و ۰تا ۲۰ میلی آمپر و ۴ تا ۲۰ میلی آمپر را دارا هستند.

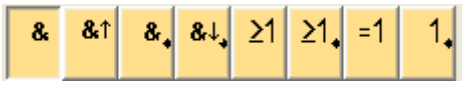
فلگ های آنالوگ–Analog flags

لوگو دارای ۶ فلگ آنالوگ می باشد، که می توانند یک مقدار آنالوگ را دریافت کرده و همان را به خروجی خود ارسال کنند.



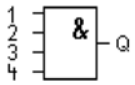
شکل (۶–۱۰): پنجره تنظیمات فلگ آنالوگ

توابع بیسیک (پایه) (Basic function)



منظور از توابع مبنا، توابع منطقی مانند AND، OR، XOR و... می باشند. که شرح آن به صورت زیر است.

AND



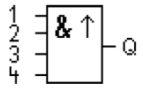
مشابه با اتصال سری در یک حلقه است. خروجی آن وقتی 1 است که تمام ورودیها 1 باشد.

ورودی X(ورودی بی اهمیت و یا متصل نشده) در این تابع 1 در نظر گرفته می شود.

1	2	3	4	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

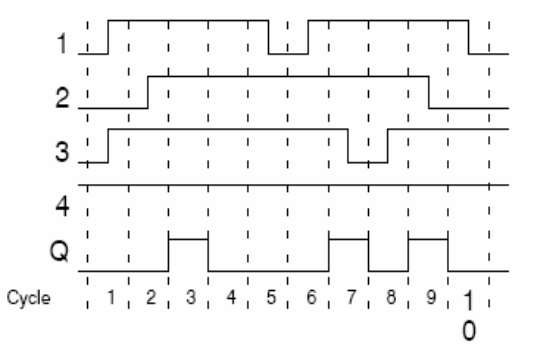
جدول (۶–۱): جدول منطقی گیت AND

AND تحریک شده با لبه بالا رونده



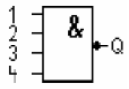
خروجی فقط هنگامی 1 است که همه ورودیها 1 بوده و حداقل یکی از ورودیها در سیکل قبل 0 بوده باشد.

ورودی X (ورودی بی اهمیت و یا متصل نشده) در این تابع 1 در نظر گرفته می شود.



شکل (۶-۱۱): نمایش سیکل های زمانی گیت AND تحریک شده با لبه بالا رونده

NAND

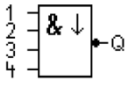


خروجی فقط وقتی 0 است که همه ورودیها 1 باشند. ورودی X (ورودی بی اهمیت و یا متصل نشده) در این تابع 1 در نظر گرفته می شود.

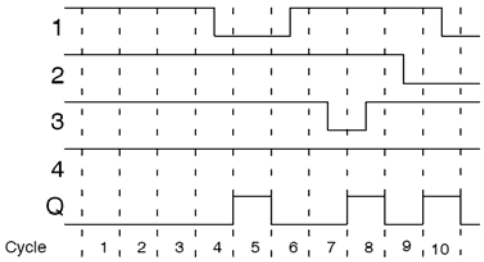
1	2	3	4	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

جدول (۶-۱): جدول منطقی گیت AND

NAND با لبه پایین رونده

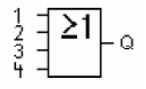


خروجی ها با تغییر ورودیها از 1 به 0 برای یک لحظه یک می شوند.



شکل (۶-۱۲): نمایش سیکل های زمانی گیت NAND با لبه پایین رونده

OR

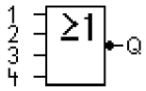


مشابه با اتصال موازی در یک حلقه است که خروجی آن وقتی 0 است که تمام ورودیها 0 باشند و خروجی آن وقتی 1 است که حداقل یکی از ورودیها 1 باشد.

1	2	3	4	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

جدول (۶-۲): جدول منطقی گیت OR

NOR

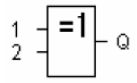


خروجی فقط هنگامی 1 است که همه ورودیها 0 باشند. ورودی X (ورودی بی اهمیت و یا متصل نشده) در این تابع 0 در نظر گرفته می شود. در تغییر 0 به 1 در یکی از ورودیها، خروجی 0 می شود.

1	2	3	4	Q
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

جدول (۶-۳): جدول منطقی گیت NOR

XOR

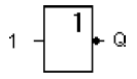


خروجی فقط وقتی 1 است که ورودیها غیریکسان باشند. ورودی X (ورودی بی اهمیت و یا متصل نشده) در این تابع 0 در نظر گرفته می شود.

1	2	Q
0	0	0
0	1	1
1	0	1
1	1	0

جدول (۶-۴): جدول منطقی گیت XOR

منفی کننده- NOT (Negation, Inverter)



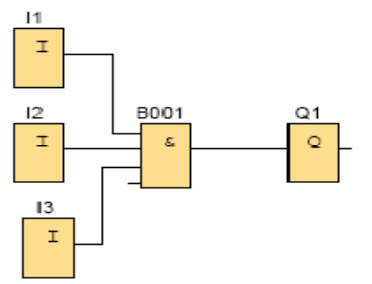
این تابع معکوس کننده ورودی است، یعنی 0 را به 1 و 1 رابه 0 تغییر می دهد.

1	Q
0	1
1	0

جدول (۶-۵): جدول منطقی گیت NOT

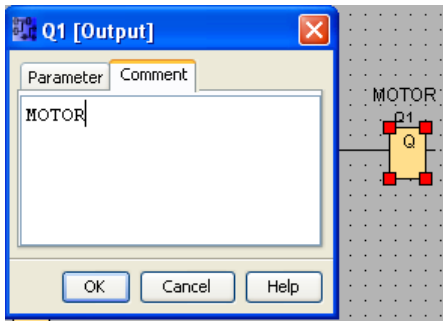
مثال(۶-۱): دستگاهی هنگامی کار می کند که هر سه کلید آن فعال باشند. برنامه مربوطه را به زبانهای LAD وFBD بنویسید.

جواب: اینکه هر سه کلید باید بطور همزمان فعال باشند، تا دستگاه کار کند، به معنی گیت AND می باشد. بنابراین در برنامه نویسی به روش FBD کافی است یک گیت AND از توابع پایه به صفحه برنامه نویسی آورده و input ها را به ورودیهای آن وصل کنیم. یک خروجی نیز به نشانه موتور دستگاه به گیت AND وصل می کنیم.



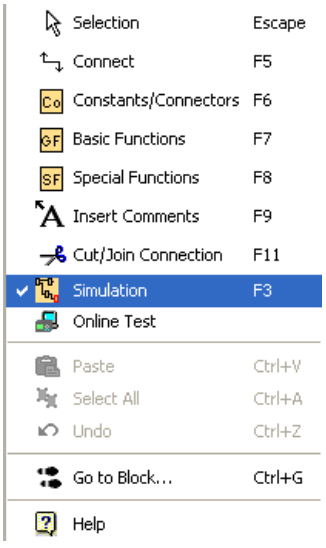
شکل (۶-۱۳): مدار مربوط به مثال (۶-۱) – FBD

می توان با راست کلیک کردن بر روی خروجی Q1 وانتخاب Block properties و سپس انتخاب comment بـر روی خروجـی اسم نوشت.



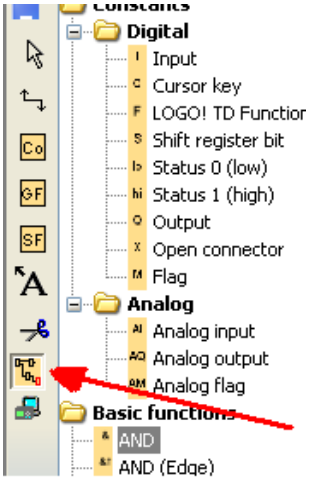
شکل (۶-۱۴): برچسب گذاری بر روی خروجی

برای شبیه سازی می توانید به دو روش این کار را انجام دهید. در روش اول در صفحه برنامه نویسی کلیک راست کرده و گزینـه Simulation را انتخاب کنید. این کار را می توانید مستقیماً با زدن کلیدF3 نیز انجام دهید.



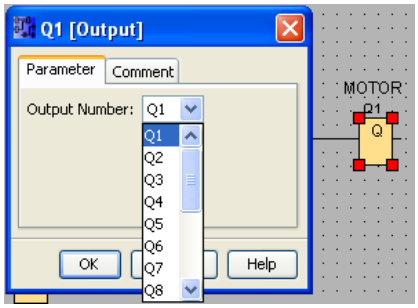
شکل (۶-۱۵): نحوه شبیه سازی

در روش دوم آیکن مربوط به شبیه سازی در سمت چپ صفحه را فعال کنید.



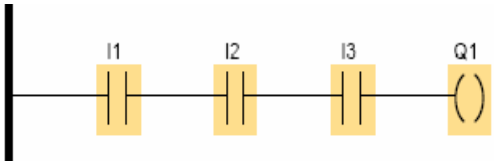
شکل (۶-۱۶): شبیه سازی با استفاده از آیکن Simulation

برای تغییر آدرس ورودیها و یا خروجیها می توانید بعد از انتخاب Block properties در کادر Parameter آدرس ها را تغییردهید. مشخص است که ورودیهای دیجیتال تا I24 و خروجیها نیز تا Q16 موجود می باشند.



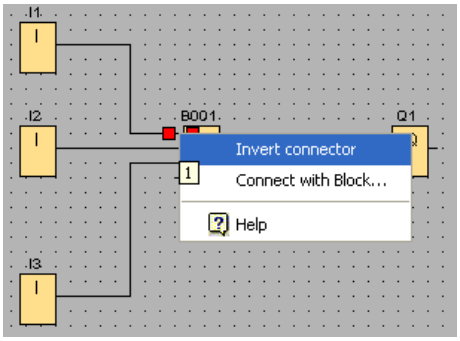
شکل (۶-۱۷): نحوه تغییر آدرس خروجیها

در زبان LAD دیگر گیت AND یا OR وجود ندارد. به جای AND ورودیها را سری می کنیم و به جای OR ورودیها را موازی می کنیم. در زبان LAD ورودیهای NO (در حالت عادی باز- normal open) را از Make contact و ورودیهای NC (normalcolse) را از Break contact می آوریم. خروجی نیز با نام Relay coil در سمت چپ صفحه موجود می باشد. لازم به یادآوری است که برنامه نوشته شده به هریک از زبانها قابل تبدیل به یکدیگر می باشند. برای اینکار در بالای صفحه آیکن convert to LAD/FBD را فشار دهید.



شکل (۶-۱۸): مدار مربوط به مثال (۶-۱) –LAD

جهت not کردن ورودی در زبان بلوکی(FBD)، بایستی در ورودی گیت ها کلیک راست کرده و گزینه Invert convertor را انتخاب کنید. در این حالت ورودی not می شود. برای برگشت به حال قبلی بایستی همان مراحل را تکرار کنید. در زبان LAD ورودیهای not شده یا به عبارتی NC را از بسته Break contact به صفحه بیاورید.

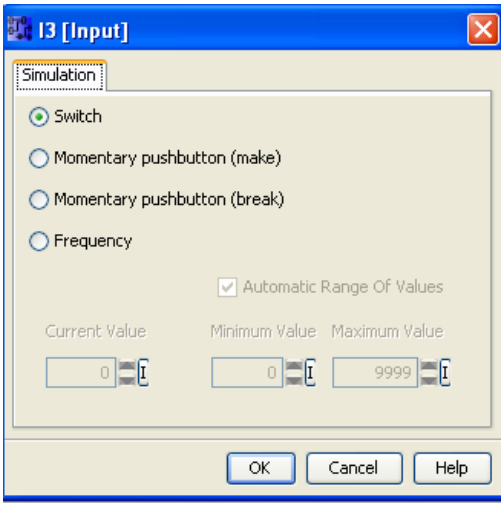


شکل (۶-۱۹): نحوه not کردن ورودی در زبان FBD

در صورتیکه بخواهید وضعیت ورودیها را از حالت سویچ به شستی تغییر دهید، می توانید به یکی از دو روش زیر اینکار را انجام دهید. در روش اول در مد شبیه سازی بر روی ورودیها در نوار سیمیلاتور کلیک راست کرده و گزینه Simulation parameters را انتخاب کنید.

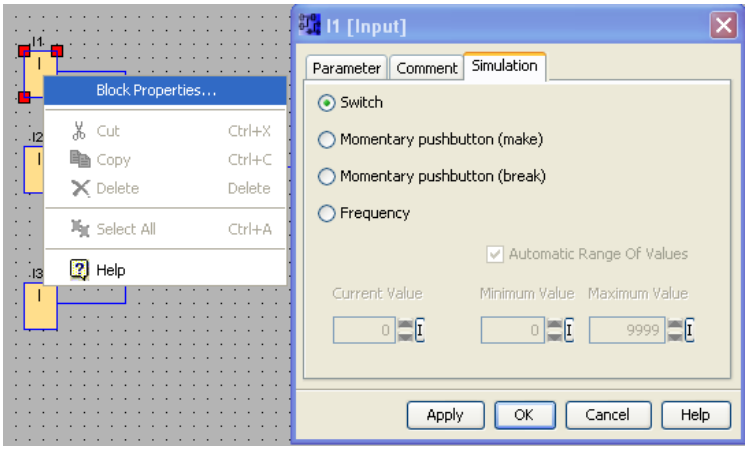
شکل (۶-۲۰): نحوه رفتن به پنجره انتخاب نوع ورودی

در این حالت پنجره ای به شکل (۶-۲۱) باز می شود. گزینه اول حالت سویچ می باشد. گزینه دوم شستی NO (در حالت عادی باز) و گزینه سوم شستی NC (در حالت عادی بسته) می باشد. گزینه چهارم ورودی فرکانسی می باشد.



شکل (۶-۲۱): پنجره تنظیمات حالات ورودی

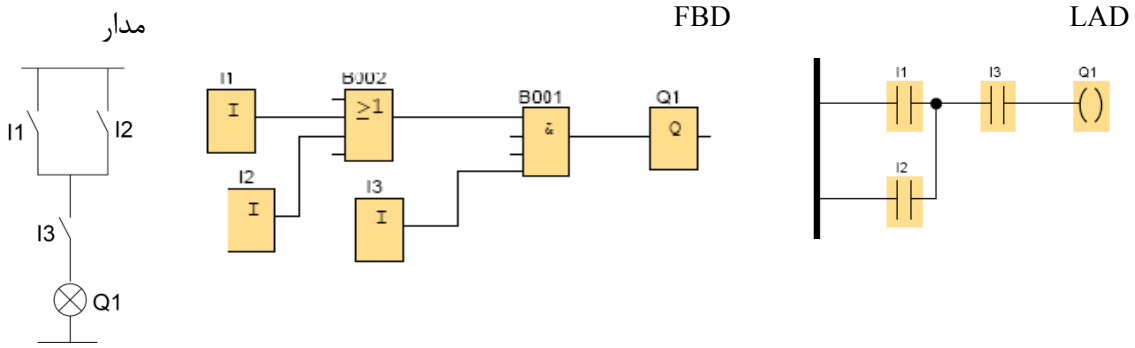
همچنین می توانید بدون فعال کردن شبیه سازی در صفحه برنامه نویسی بر روی هر یک از ورودیها کلیک راست کرده و با انتخاب Block properties و سپس انتخاب کادر simulations از پنجره باز شده همان تنظیمات فوق را انجام دهید. (البته با دو بار کلیک کردن بر روی ورودی نیز می توانید وارد پنجره مشخصات ورودی شوید.)



شکل (۶-۲۲): نحوه رفتن به پنجره تنظیمات ورودی

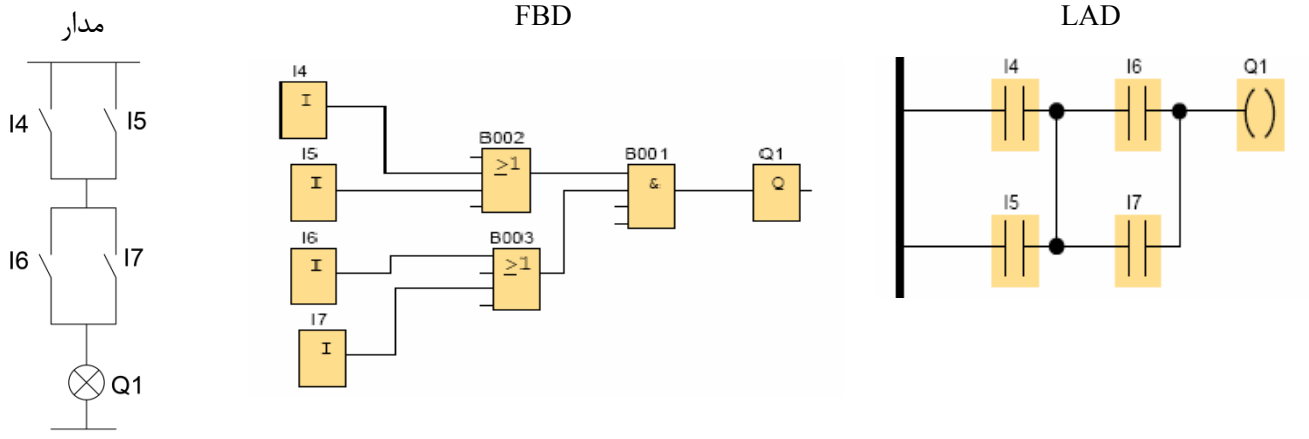
برنامه مدارهای زیر را به هر دو زبان LAD وFED بنویسید.

مثال(۶-۲): با توجه به مدار زیر مشخص است که ورودیهای I1 وI2 موازی هستند، لذا در زبان بلوکی از گیت OR استفاده می کنیم. نتیجه OR نیز با ورودی I3 سری می باشد، لذا از AND استفاده می کنیم.



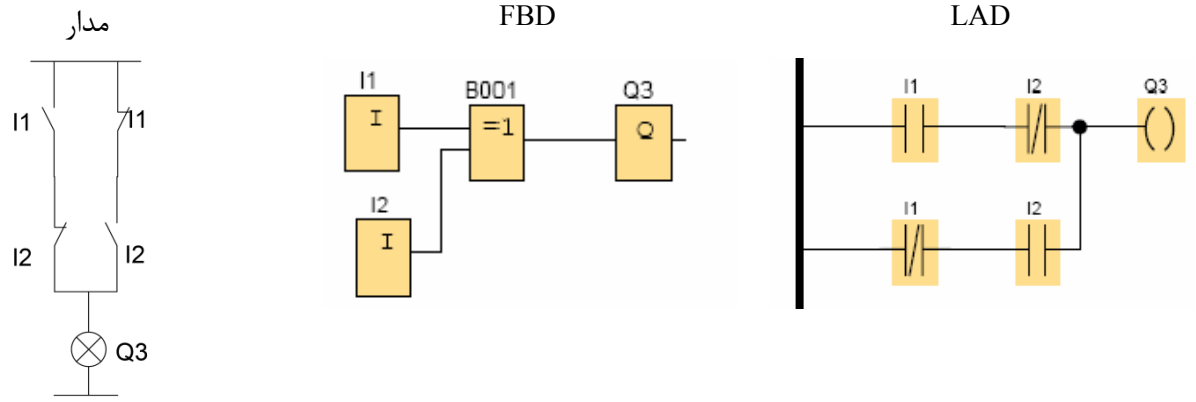
شکل (۶-۲۳): مدار مربوط به مثال (۶-۲)

مثال(۶-۳): در این مدار ورودیهای I4 و I5 با یکدیگر و ورودیهای I6 و I7 نیز باهم موازی می باشند. و تر کیب این موازی‌ها با یکدیگر سری می باشند. لذا به دو گیت OR ویک گیت AND در زبان FBD نیاز می باشد.



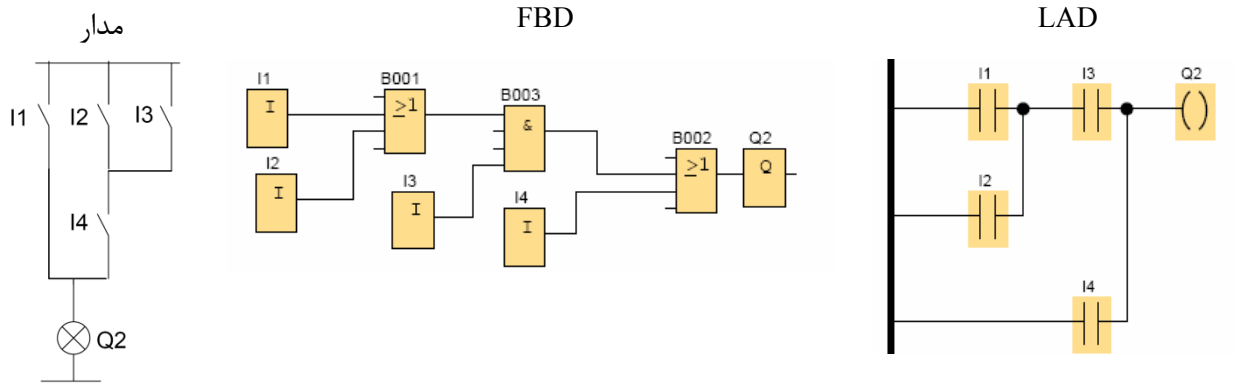
شکل (۶-۲۴): مدار مربوط به مثال (۶-۳)

مثال(۶-۴): با دقت در مدار این مثال متوجه می شوید که این مدار از دو شستی دوبل استفاده کرده است. گرچه می توان در زبان بلوکی از ترکیب دو AND و یک OR مدار فوق را ساخت، ولی این مدار همان گیتXOR می باشد که در بین توابع پایه موجود می باشد.



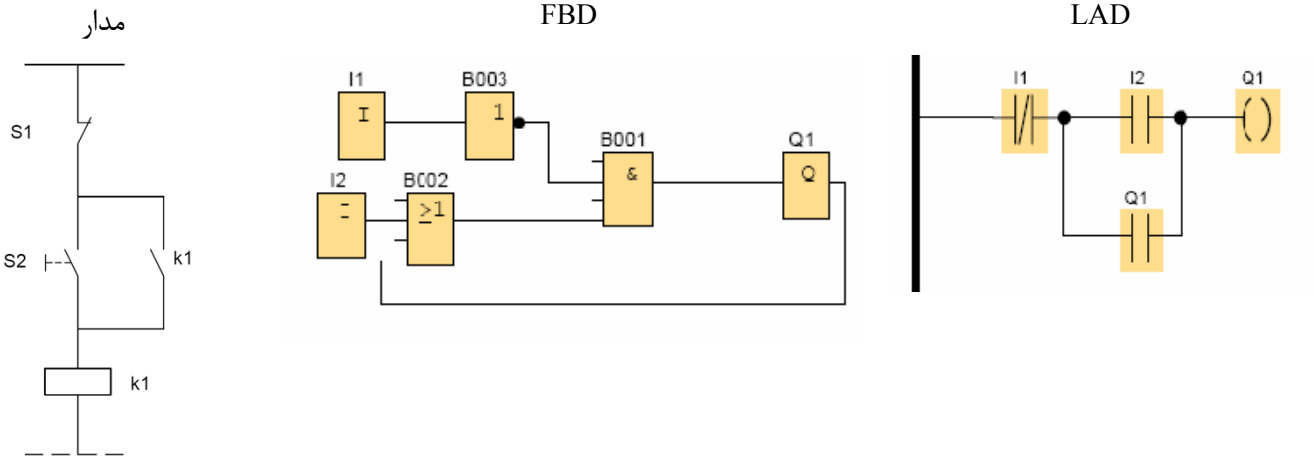
شکل (۶-۲۵): مدار مربوط به مثال (۶-۴)

مثال(۶-۵):



شکل (۶-۲۶): مدار مربوط به مثال (۶-۵)

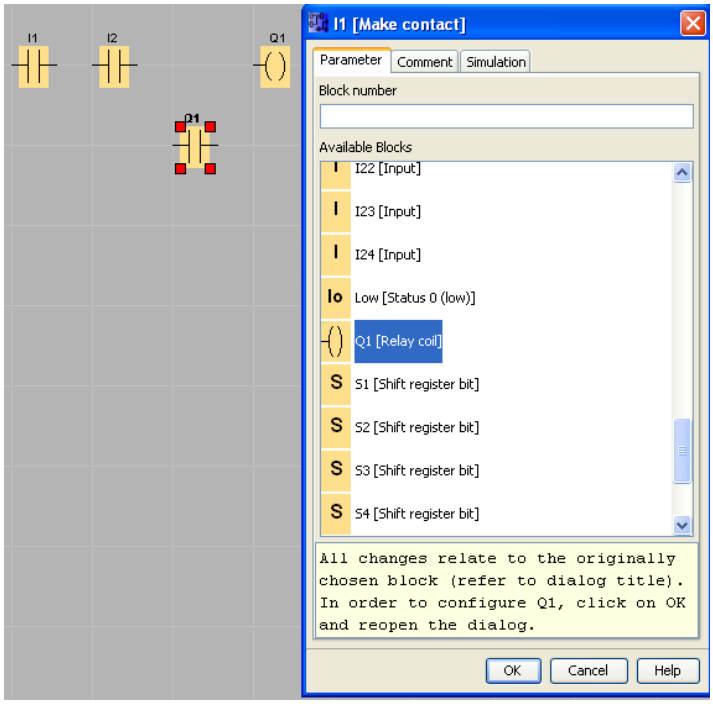
مثال(۶-۶): مدار زیر یک مدار فرمان ساده راه اندازی موتور سه فاز می باشد.



شکل (۶-۲۷): مدار مربوط به مثال (۶-۶)

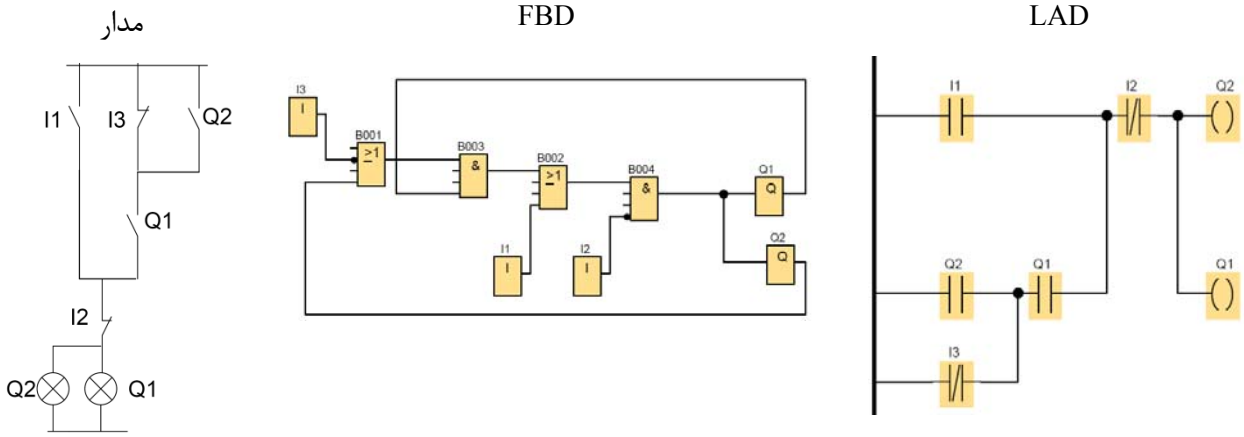
از آنجایی که خروجی Q1 با I2 موازی است، در زبان بلوکی از گیت OR استفاده می کنیم. برای استوپ نیز بایستی ورودی I1 منفی شود.

در روش LAD برای آنکه کنتاکت باز Q1 را که بطور موازی با استارت قرار گرفته است، به صفحه برنامه نویسی بیاورید، کافی است ابتدا خروجی Q1 را به صفحه برنامه نویسی بیاورید. سپس به سراغ Make contact رفته و در صفحه برنامه نویسی کلیک کنید. در این حالت در پنجره باز شده Make contact به انتهای صفحه رفته و Q1 را انتخاب کنید. مشاهده می کنید که کنتاکت باز Q1 در صفحه ظاهر می شود. لازم به یاد آوری است که برای آنکه بخواهید کنتاکت بسته تابعی را در صفحه ظاهر کنید، بایستی این بار پس از آوردن تابع مورد نظر به صفحه برنامه نویسی، به سراغ Break contact رفته و همان مراحل فوق را تکرار کنید.



شکل (۶-۲۸): نحوه آوردن کنتاکت باز Q1

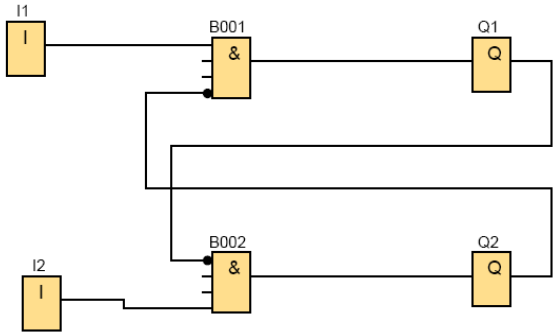
مثال(۶-۷):



شکل (۶-۲۹): مدار مربوط به مثال (۶-۷)

مثال(۶-۸): مدار Interlock

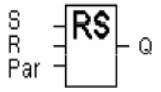
دو ورودی داریم که هرکدام یک خروجی را فعال می کنند. برنامه ای بنویسید که در صورت فعال شدن هر دو ورودی به صورت همزمان فقط یکی از خروجیها فعال باشد.



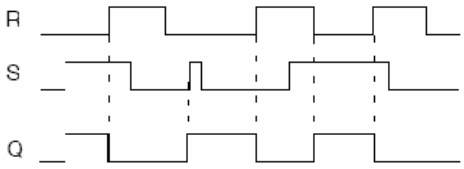
شکل (۶-۳۰): مدار مربوط به مثال (۶-۸)

لازم به توضیح است که برخی توابعی که در ذیل می آیند، به جهت کاربرد در مثالها زودتر از موعد مقرر و خارج از ترتیب ذکر شده در دستورات لوگو می آیند.

رله نگهدارنده- Latching Relay

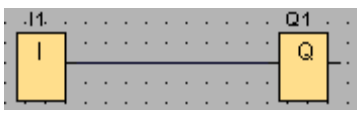


خروجی با لبه بالا رونده S (ست) فعال می شود و با لبه پایین رونده S خاموش نمی شود، مبادا اینکه ورودی R (ریست) فعال شود. اگر گزینه Retentivity را فعال کنید، در این صورت وقتی برق قطع شد با وصل مجدد برق، وضعیت رله در همان حالت باقی می ماند. اگر هردو ورودی با هم فعال شوند در این حالت R بر S ارجعیت خواهد داشت. یعنی در صورتی که هر دو ورودی همزمان یک شوند، خروجی خاموش خواهد بود.

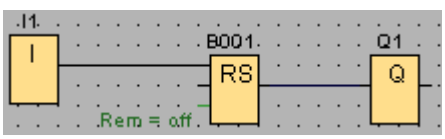


شکل (۶-۳۱): نمودار زمانی رله نگهدارنده RS

حال با توجه به توضیحات فوق، فرق دو مدار زیر چیست؟



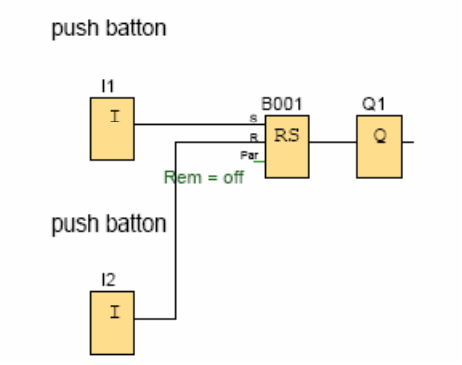
الف



ب

جواب: هر دو خروجی با فعال کردن ورودی I1 فعال می شوند. حال اگر ورودی را غیر فعال کنیم، خروجی در مدار الف خاموش می شود، ولی در مدار ب روشن می ماند و زمانی خاموش می شود که RESET فعال شود.

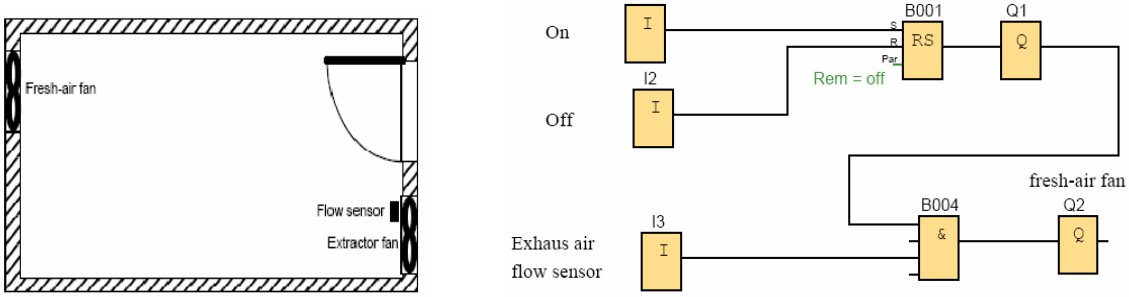
مثال(۶-۹): مدار شماره ۶ را با رله RS انجام دهید.



شکل (۶-۳۲): مدار مربوط به مثال (۶-۹)

مثال(۶-۱۰): سیستم تهویه هوای اتاق

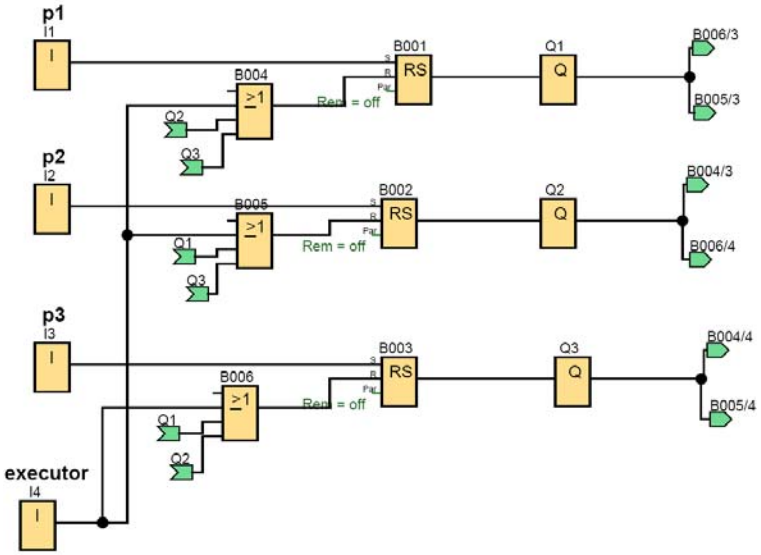
اتاقی مجهز به یک فن تخلیه هوا EXHAUST و یک فن هوای تازه FRESH-AIR FAN می باشد. فن تخلیه توسط شستی ON روشن می شود و فن هوای تازه زمانی روشن می شود که قبل از آن فن تخلیه به همراه سنسور جریان تخلیه هوا فعال شده باشد. سیستم با زدن شستی OFF خاموش می شود.



شکل (۶-۳۳): مدار مربوط به مثال (۶-۱۰)

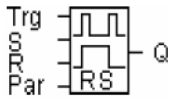
مثال(۶-۱۱): میز مسابقه ۳ نفره

برنامه میز مسابقه ۳ نفره را بنویسید. در این برنامه در صورتیکه هر کدام از شرکت کنندگان زودتر زنگ بزند، شرکت کنندگان دیگر نمی توانند زنگ بزنند. یک شستی نیز برای مجری برنامه در نظر بگیرید که هر موقع خواست چراغهای شرکت کنندگان را خاموش کند.

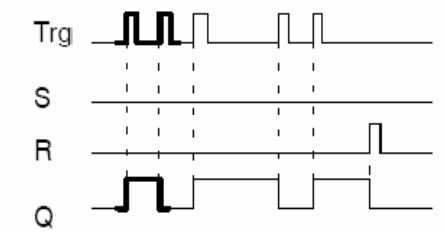


شکل (۶-۳۴): مدار مربوط به مثال (۶-۱۱)

رله پالسی- Pulse Relay

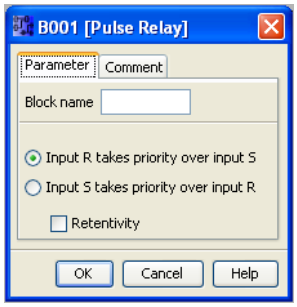


در این رله، خروجی با لبه بالا رونده پالس اول روشن و با لبه بالا رونده پالس دوم خاموش می شود. عملکرد این رله همانند شستی درب اتوبوس ها می باشد که با یکبار زدن آن در باز می شود و با زدن دوباره شستی، در بسته می شود. دیاگرام زمانی این تابع به شکل (۶-۳۵) می باشد.



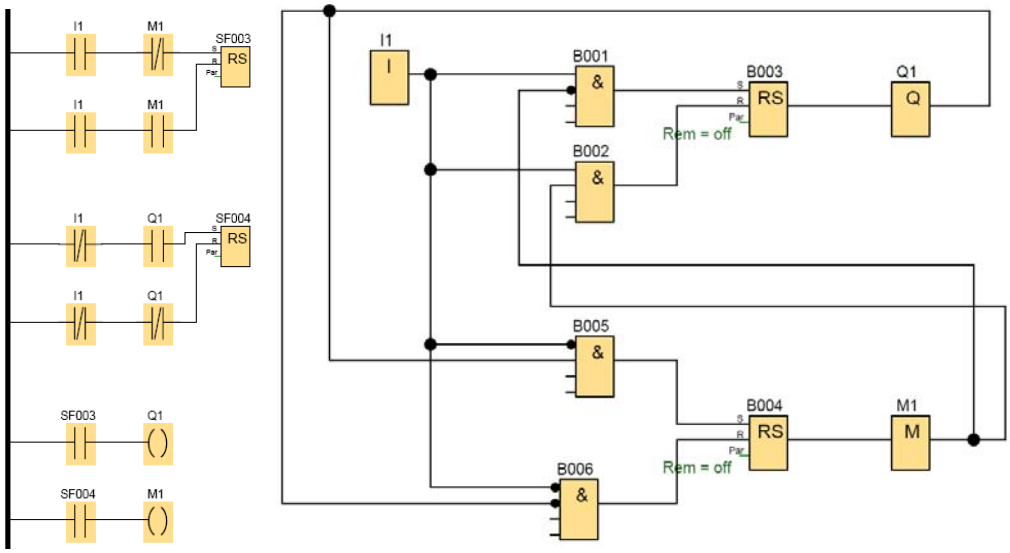
شکل (۶-۳۵): نمودار زمانی رله پالسی

اگر پایه Trg فعال نباشد، رله مثل فلیپ فلاپ RS (البته با ویژگی ارجحیت دادن) عمل می کند. در این حالت فقط از ورودیهای Rو S استفاده می شود. پنجره تنظیمات این تابع در شکل (۶-۳۶) نمایش داده شده است. با انتخاب گزینه اول ورودی R به ورودی S ارجحیت خواهد داشت، و با انتخاب گزینه دوم برعکس.



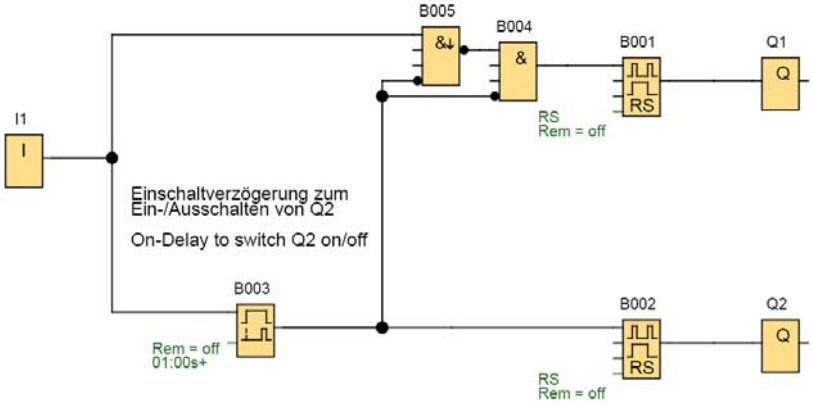
شکل (۶-۳۶): پنجره تنظیمات رله پالسی

مثال(۶-۱۲): بدون استفاده از رله پالسی برنامه ای بنویسید که با اولین لبه بالا رونده خروجی روشن و با دومین لبه بالا رونده خروجی خاموش شود.



شکل (۶-۳۷): مدار مربوط به مثال (۶-۱۲)

مثال(۶-۱۳): مداری طراحی کنید که اگر کاربر دست خود را کمتر از یک ثانیه روی شستی نگه دارد Q1 روشن و اگر دوباره دست خود را روی شستی کمتر از یک ثانیه نگه دارد Q1 خاموش شود. همچنین اگر کاربر دست خود را بیش از یک ثانیه نگه دارد، خروجی Q2 روشن شود و اگر دوباره دست خود را بیش از یک ثانیه روی شستی نگه دارد Q2 خاموش شود.

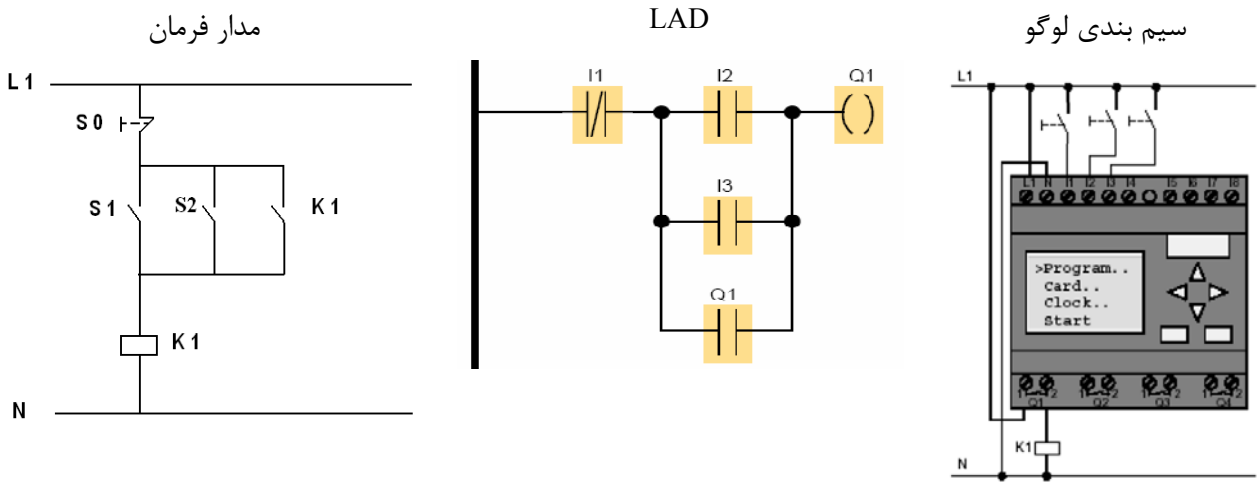


شکل (۶-۳۸): مدار مربوط به مثال (۶-۱۳)

تایمر استفاده شده در این مدار از نوع on delay می باشد که در ادامه این فصل آن را توضیح خواهیم داد. زمان تاخیر در روشن شدن این تایمر ۱ ثانیه تنظیم شده است.

برنامه مدارات فرمان زیر را به روش LAD بنویسید.

مثال(۶-۱۴): راه اندازی الکترو موتور سه فاز روتور قفسه ای بصورت دائم با استارت از دو محل و استوپ از یک محل

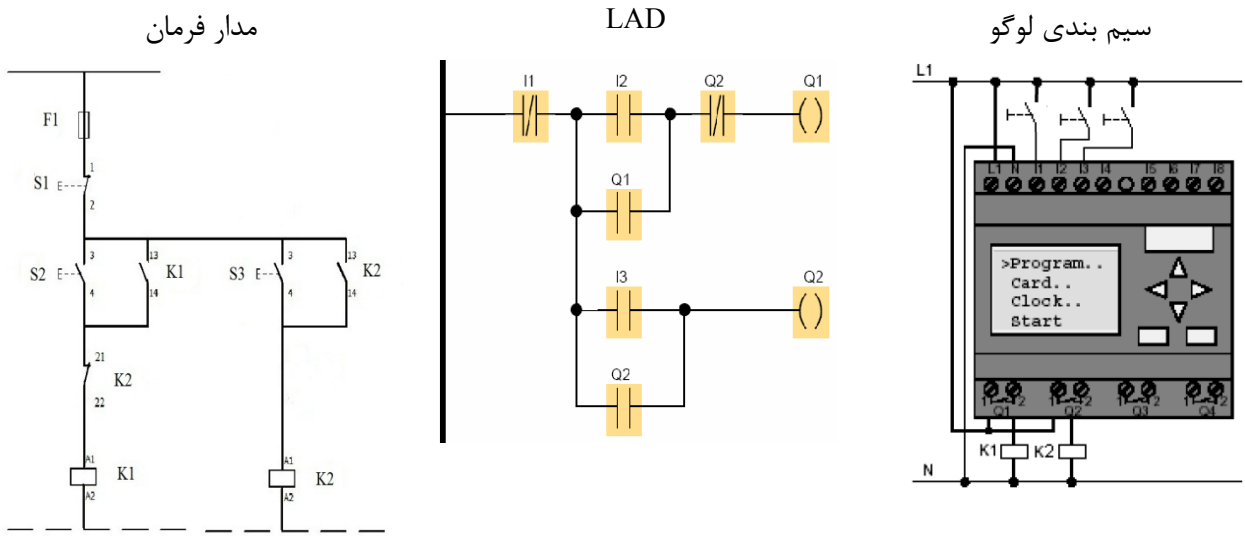


شکل (۶-۳۹): مدار مربوط به مثال (۶-۱۴)

همانطور که مشاهده می کنید در سیم بندی ورودیها تماماً از شستی استارت استفاده شده است، در حالیکه در مدار فرمان یک استوپ وجود دارد. در اینجا ما چون استوپ را به صورت نرم افزاری تعریف کرده ایم، لذا دیگر لازم نیست در سیم بندی سخت افزار از شستی استوپ استفاده کنیم. لازم به تذکر است که در صورت بروز چنین مسئله ای (یعنی استفاده از استوپ در ورودی I1) با زدن استارت Q1 هرگز روشن نخواهد شد. زیرا قبلا مسیر برق به Q1 قطع شده است. (چرا؟)

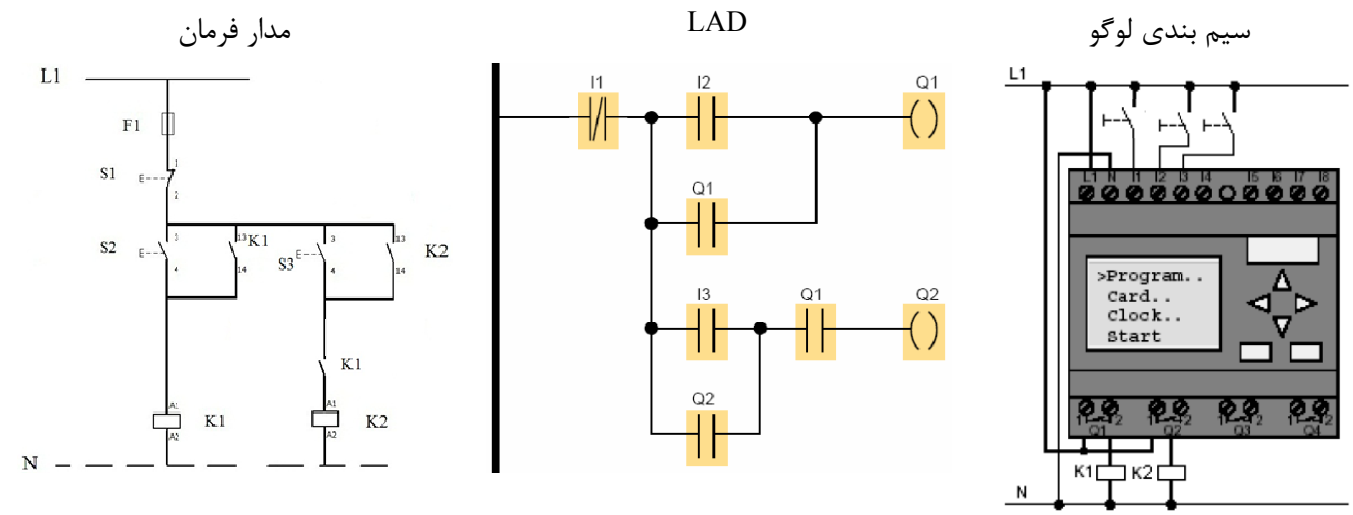
بنابراین یکی از مزیت های PLC این است که می توان استوپ را به صورت نرم افزاری تعریف کرد، بنابراین می توان تمام ورودیهای PLC را از نوع شستی باز انتخاب کرد.

مثال(۶-۱۵): راه اندازی دو الکترو موتور سه فاز به صورت یکی به جای دیگری به صورت دستی



شکل (۶-۴۰): مدار مربوط به مثال (۶-۱۵)

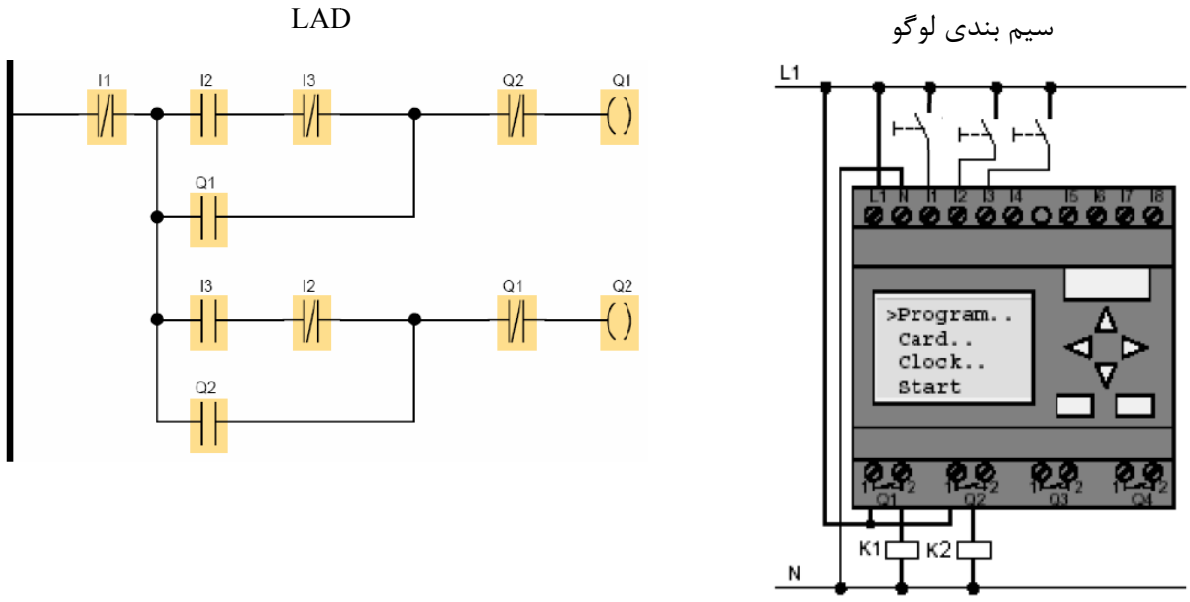
مثال(۶-۱۶): راه اندازی دو الکترو موتور سه فاز به صورت یکی پس از دیگری به صورت دستی



شکل (۶-۴۱): مدار مربوط به مثال (۶-۱۶)

از لحاظ سیم بندی، مدار همانند قبلی می باشد. مشخص است که با وجود PLC کافی است که فقط در برنامه مدار فرمان تغییر داده شود، بدون آنکه از لحاظ سخت افزاری(مدار قدرت) تغییری داده شود.

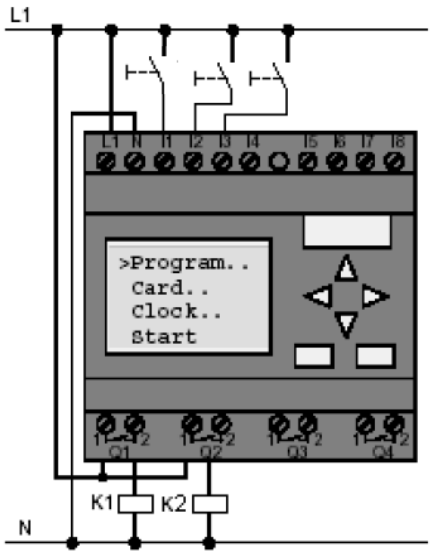
مثال(۶-۱۷): راه اندازی الکترو موتور سه فاز روتور قفسه ای به صورت چپگرد و راستگرد دور کند



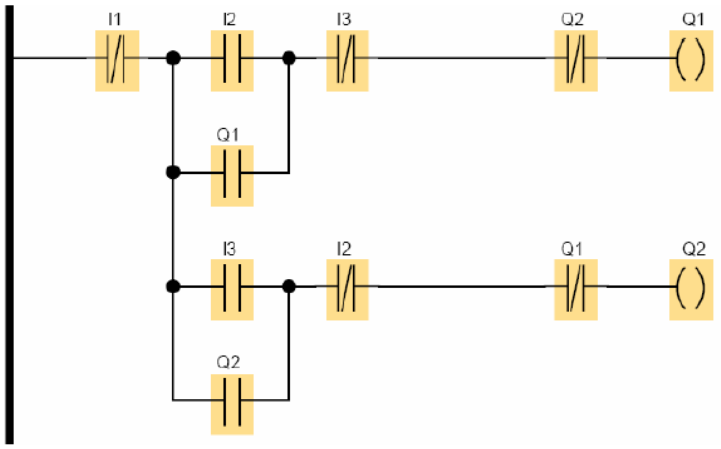
شکل (۶-۴۲): مدار مربوط به مثال (۶-۱۷)

مثال(۶-۱۸): راه اندازی الکترو موتور سه فاز روتور قفسه ای به صورت چپگرد و راستگرد دورتند

سیم بندی لوگو



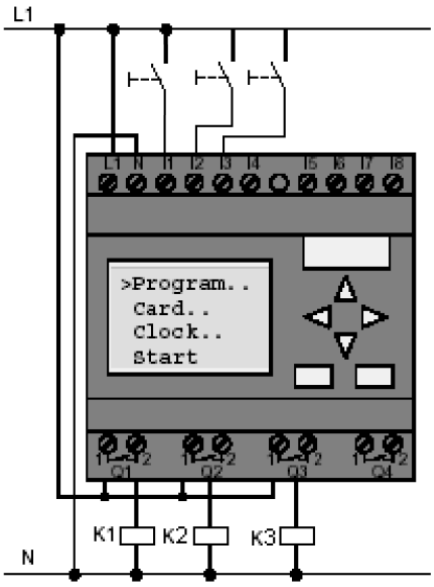
LAD



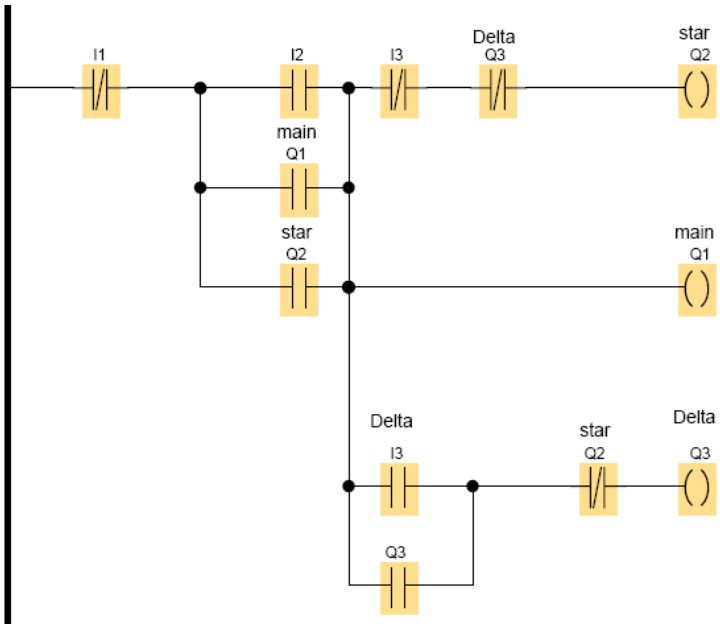
شکل (۶-۴۳): مدار مربوط به مثال (۶-۱۸)

مثال(۶-۱۹): راه اندازی الکترو موتور سه فاز روتور قفسه ای به صورت ستاره و مثلث به صورت دستی

سیم بندی لوگو



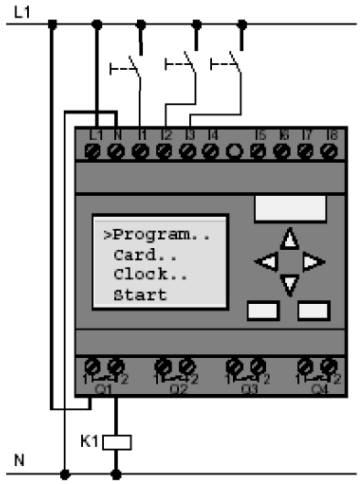
LAD



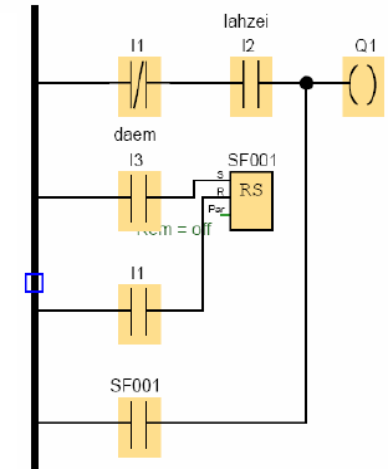
شکل (۶-۴۴): مدار مربوط به مثال (۶-۱۹)

مثال(۶-۲۰): راه اندازی الکترو موتور سه فاز روتور قفسه ای به صورت لحظه ای و دائم کنترل از یک محل

سیم بندی لوگو



LAD

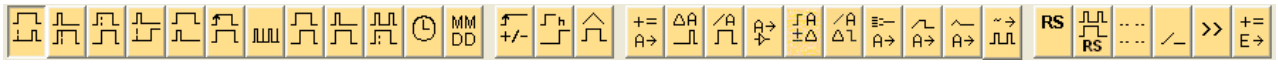


شکل (۶-۴۵): مدار مربوط به مثال (۶-۲۰)

با توجه به اینکه در مدارات فرمان رله ای کنتاکت بسته زودتر از کنتاکت باز عمل می کند، ولی در PLC تمامی کنتاکت ها بصورت همزمان عمل می کنند، بنابراین نمی توان از مدار فرمان ذکر شده در فصل اول برای زبان LAD استفاده کرد. لذا برای برنامه نویسی از یک تابع RS استفاده شده است.

با وجود PLC، مدار فرمان حذف می شود، ولی مدار قدرت بر سر جای خود باقی است. یعنی، فرمان کنترل را بوسیله PLC، انجام خواهیم داد. در مدارات فرمان رله کنتاکتوری، باز یا بسته بودن کنتاکت های مدار قدرت را، مدار فرمان تعیین می کرد. که مدار فرمان اینکار را با برق دار یا غیر برق دار کردن بوبین کنتاکتور، انجام می داد. اکنون در حالت اتوماسیون اینکار را با وصل کردن خروجی های LOGO (البته مدلهای رله ای) به بوبین کنتاکتور ها، انجام خواهیم داد. در فصل قبل نیز با نحوه سیم بندی این خروجی ها آشنا شده اید.

توابع ویژه-(Special Function)



هنگام برنامه ریزی LOGO علاوه بر توابع مبنا از توابع ویژه نیز می توان استفاده نمود. ابتدا اتصالات و یا پارامترهایی که به سایر ماژولها و یا ورودیها متصل می‌شود را تعریف کرده و سپس لیست این توابع آورده می‌شود و به دنبال آن مشروح هر یک نیز می آید.

S (set): می تواند برای یک کردن خروجی استفاده شود.

R (reset): این ورودی نسبت به سایر ورودی ها تقدم و ارجحیت داشته و برای صفر کردن خروجی استفاده می شود.

TRG (trigger): این ورودی برای تحریک و آغاز به کار سیکل تابع استفاده می شود.

CNT (count): این ورودی برای شمارش پالسها استفاده می شود.

DIR (direction): این ورودی برای تعیین جهت شمارش (صعودی یا نزولی بودن) استفاده می شود.

EN (enable): این ورودی تابع را فعال می نماید. وقتی این ورودی 0 باشد، بلوک تمام سیگنالهای دیگر را نادیده می گیرد.

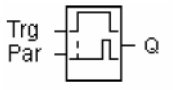
INV (invert): هنگامی که این ورودی set باشد، خروجی بلوک معکوس می گردد.

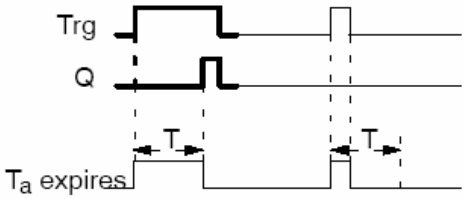
RAL (reset all): تمام مقادیر داخلی reset می شوند.

PAR (parameter): از این ورودی برای تنظیم پارامترها استفاده می شود.

T (time): از این ورودی برای تنظیم زمان استفاده می شود.

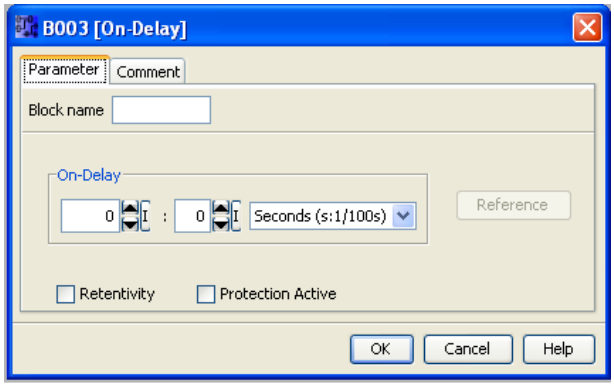
تایمر تاخیر در وصل- on – delay





شکل (۶-۴۶): نمودار زمانی تایمر تاخیر در وصل

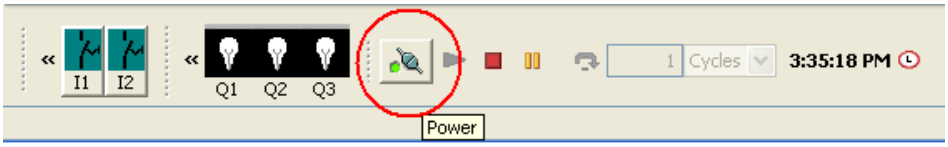
در این تابع خروجی زمانی فعال می شود که زمان تعریف شده Ta بعد از فعال شدن Trg سپری شده باشد. همچنین خروجی با لبه پایین رونده ورودی غیر فعال می شود. پنجره تنظیمات این تایمر در شکل (۶-۴۷) نشان داده شده است. می توانید مقیاس زمان را برحسب ثانیه، دقیقه و ساعت تنظیم کنید.



شکل (۶-۴۷): پنجره تنظیمات تایمر تاخیر در وصل

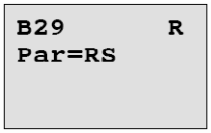
Retentivity: با انتخاب این گزینه داده های جاری بر روی بلوک تابع ویژه بعد از قطع برق تغییر نمی کنند. فرض کنید تایمر تا ۲ ثانیه شمرده، زمانی که برق قطع می شود، در صورتیکه این گزینه فعال نباشد بعد از آمدن برق زمان مجدداً از صفر شمارش خواهد کرد. ولی اگرگزینه Retentivity فعال باشد، زمان از همان جائی که قبل از قطع برق مانده بود شروع به شمارش خواهد کرد. لازم به توضیح است که توابع ویژه شمارنده ساعتی، تایمر هفتگی، تایمر سالیانه و کنترل کننده PI همیشه Retentive می باشند. می توانید این حالت را به صورت زیر در نرم افزار شبیه سازی کنید.

برای قطع برق آیکن نشان داده شده (power) در نوار شبیه سازی را موقع شمارش تایمر برای چند ثانیه فشار داده و رها کنید. این کار را یکبار برای حالتی که Retentivity فعال است و بار دیگر برای حالتی که این گزینه فعال نیست انجام دهید.



شکل (۶-۴۸): نحوه شبیه سازی قطع بودن برق

در هنگام برنامه نویسی توسط سخت افزار نیز برای حفظ اطلاعات این گزینه را که با حرف R مشخص است، فعال کنید.



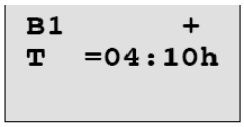
شکل (۶-۴۹): نمایش حالت Rtentivity در نمایشگر لوگو

نـکـتـه : در ورژن های قدیمی، فقط کانترها دارای این گزینه هستند.

در یک LOGO بعد از قطع برق باید اطلاعات بدون هیچ تغییری ثابت باقی بمانند تا در زمان استارت مجدد، برنامه بـه درسـتی اجرا شود. مقدار زمانی که LOGO می تواند بعد از قطع برق بوسیله باطری های داخلی اطلاعات یک برنامه را بدون تغییر نگه دارد ۸۰ ساعت است. اگر قبل از اتمام این زمان برق LOGO وصل نشود، در آن صورت اطلاعات از روی حافظه پاک می شود.

البته اصل برنامه بعد از گذشت این زمان از بین نمی رود، بلکه مقادیری که در زمان اجرا متغیـر بـوده(ماننـد مقـادیر کانترهـا و تایمرها و تاریخ سخت افزار) و از حافظه های پایدار Retentive استفاده کرده اند، ریست می شوند.

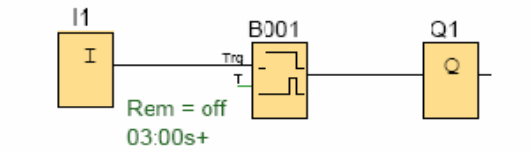
Protection Active: با انتخاب این گزینه به اپراتور این اجازه را نمی دهیم که در زمان اجرای برنامه وارد پارامتر برنامه شـود و مقادیر بلوک زمانها و... را تغییر بدهد. در سخت افزار این کار بوسیله علامت(-) صورت می گیرد. با انتخاب علامت (+) که پـیش فرض دستگاه است کاربر می تواند پارامترها را بدون متوقف کردن برنامه از منوی SET PARM تغییر دهد.



شکل (۶-۵۰): نمایش حالت protection active در نمایشگر لوگو

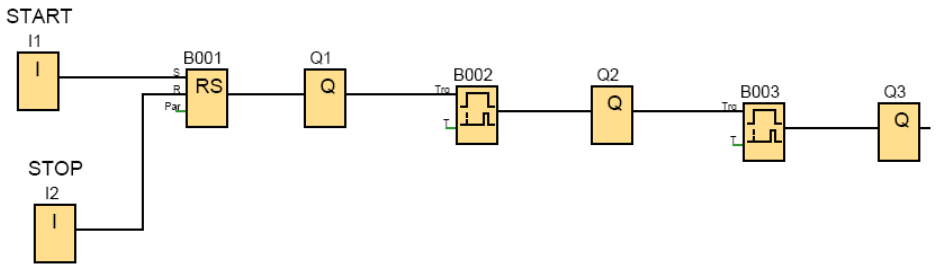
دقت زمان T: تلرانس در تایمرها برابر ۵ +/ - ثانیه در یک ۲۴ ساعت می باشد.

مثال(۶-۲۱): برنامه ای بنویسید که، خروجی 3 ثانیه بعد از فعال شدن ورودی I1 روشن شود.



شکل (۶-۵۱): مدار مربوط به مثال (۶-۲۱)

مثال(۶-۲۲): برنامه ای بنویسید که با زدن Start، ابتدا Q1 روشن و پس از ۲ ثانیه Q2 نیز روشن و پس از ۲ ثانیه بعد از روشن شدن Q2 خروجی Q3 نیز روشن شود. هر گاه Stop زده شد همگی خاموش شوند.



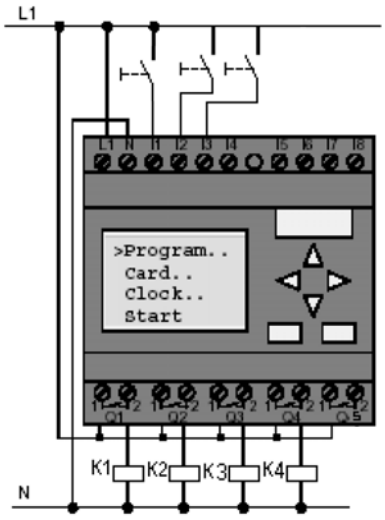
شکل (۶-۵۲): مدار مربوط به مثال (۶-۲۲)

در برنامه فوق چون گفته ورودیها شستی باشند، بایستی برای نگهدارندگی ورودی تایمر تأخیر در روشن حتماً از RS استفاده کنیم، چرا که این تایمر به لبه پایین رونده ورودی حساس می باشد و با قطع برق ورودی، خروجی این تایمر صفر می شود.

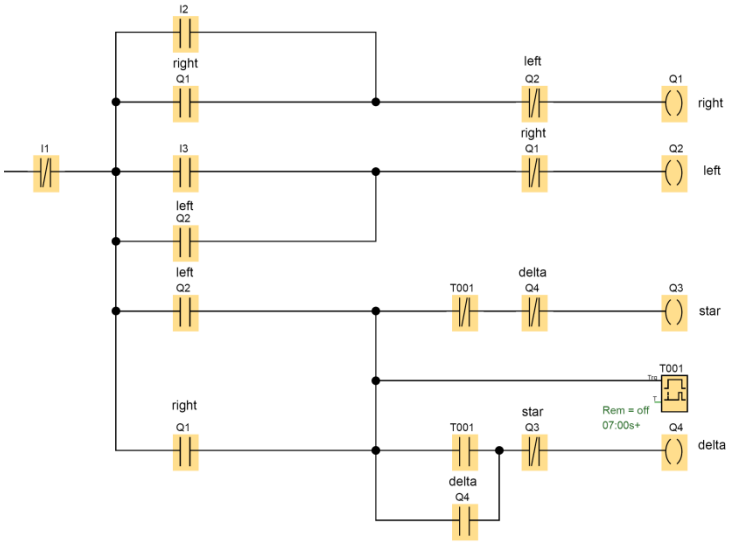
مثال(۶-۲۳): برنامه راه اندازی موتور سه فاز به صورت ستاره مثلث توام با راستگرد و چپگرد را بنویسید. مدت زمان تغییر از ستاره به مثلث را ۷ ثانیه در نظر بگیرید.

ابتدا بایستی وضعیت راستگرد و چپگرد بودن تایید شود. یعنی به عنوان مثال ابتدا کنتاکتور راستگرد به همراه کنتاکتور ستاره کار کند سپس کنتاکتور راستگرد به همراه مثلث کار کند. برای چپگرد نیز چنین وضعیتی وجود داشته باشد.

سیم بندی لوگو

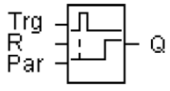


LAD

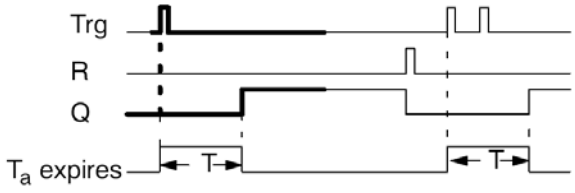


شکل (۶-۵۳): مدار مربوط به مثال (۶-۲۳)

تایمر تاخیر در روشن شدن با خاصیت نگهدارندگی-Retentive on-delay

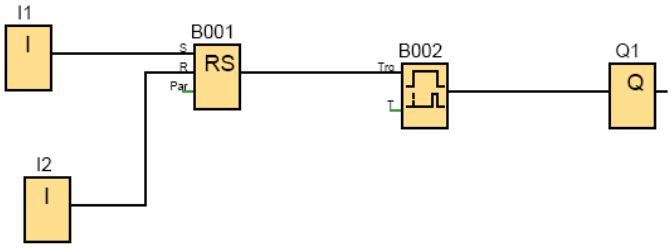


همانند تایمر **on-delay** است، با این تفاوت که خروجی به لبه پایین رونده ورودی حساس نیست. همچنین پایه Reset نیز دارد. شکل (۶-۵۴) دیاگرام عملکرد این تابع را نشان می دهد.



شکل (۶-۵۴): نمودار زمانی تایمر Retentive on-delay

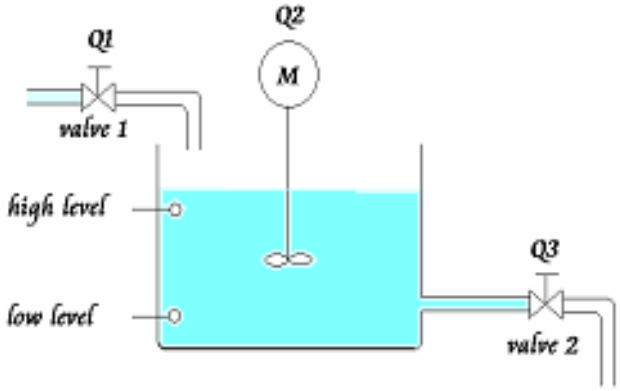
مثال(۶-۲۴): با استفاده از تایمر on- delay تایمر Retentive on-delay بسازید.

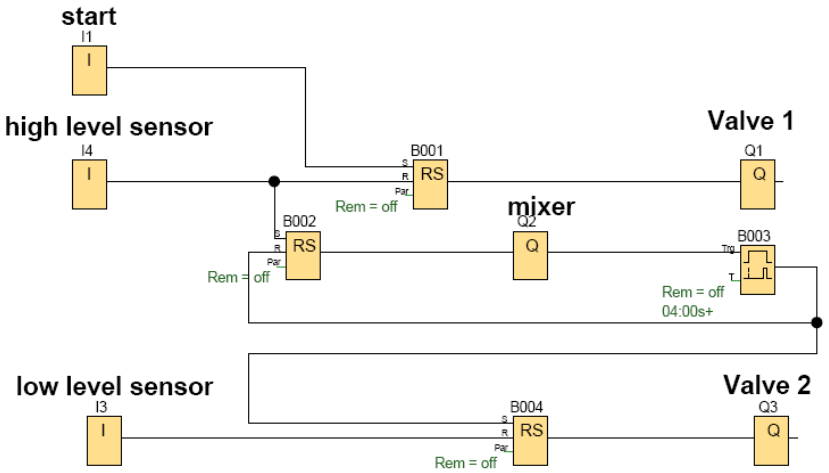


شکل (۶-۵۵): مدار مربوط به مثال (۶-۲۴)

مثال(۶-۲۵): میکسر

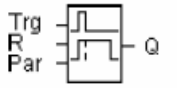
با زدن شستی استارت شیر ۱ باز می شود تا مایع وارد مخزن شود. وقتی مخزن به اندازه High level پر شد شیر ۱ غیر فعال شده، موتور همزن روشن می شود و به مدت ۱۰ ثانیه mixer را می چرخاند. سپس شیر ۲ باز شده مخزن را خالی می کند. زمانیکه سنسور Low level فعال شد، شیر دو نیز بسته می شود.



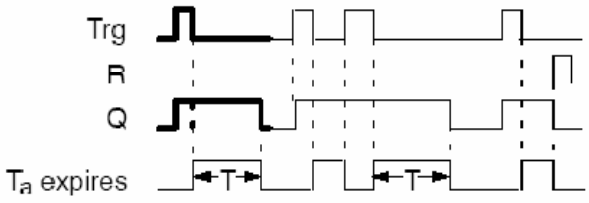


شکل (۶-۵۶): مدار مربوط به مثال (۶-۲۵)

تایمر تاخیر در خاموشی- off – delay



در این تابع، خروجی با لبه بالا رونده ورودی Trg روشن می شود زمان T نیز با لبه پایین رونده Trg شروع به شمارش می کند. خروجی تا زمانی که زمان تنظیم شده Ta سپری نشده روشن باقی می ماند. شکل زیر دیاگرام عملکرد این تابع را نشان می دهد.



شکل (۶-۵۷): نمودار زمانی تایمر تاخیر در خاموشی

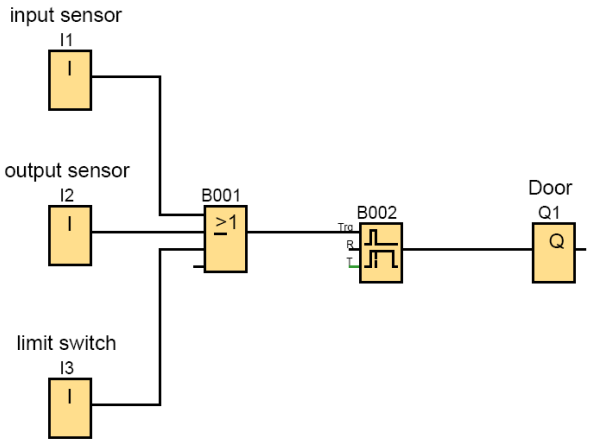
خروجی با لبه بالا رونده R خاموش می شود.

ضمناً این تایمر به آخرین لبه پایین رونده ورودی حساس است. مثال زیر عملکرد تایمر فوق را به خوبی نشان می دهد.

مثال(۶-۲۶): سیستم درهای اتوماتیک بانکها و فروشگاه ها

دو آشکار ساز حرکتی I1 و I2 یکی در داخل و دیگری در خارج، نزدیک شدن اشخاص به در را حس می کنند و یک پالس را به LOGO ارسال می کنند. در همین حال در به مدت ۵ ثانیه بطور اتوماتیک باز می شود و اگر قبل از اتمام ۵ ثانیه شخص دیگری وارد شود بایستی این زمان دوباره تمدید شود. اگر شخصی در لای در بماند لیمیت سویچ تعبیه شده حضور وی را به LOGO گزارش می دهد و تا شخص از در فاصله نگرفته در بسته نمی شود تا به وی آسیبی نرسد. برنامه پروسه فوق را بنویسید.

برای آنکه با هر بار وارد شدن شخص، باز بودن در به مدت ۵ ثانیه دیگر تمدید شود، می توان از تایمر تاخیر در خاموشی استفاده کرد، زیرا مبنای شمارش زمان در این تایمر آخرین لبه پایین رونده ورودی است.



شکل (۶-۵۸): مدار مربوط به مثال (۶-۲۶)

مثال(۶-۲۷): کنترل نوار نقاله جهت تخلیه بار

مدار بایستی طبق خواسته های زیر عمل کند:

۱- با زدن شستی I1 مدار به مدت ۵ ثانیه شروع به بارگیری نماید.

۲- بعد از اتمام ۵ ثانیه بارگیری، موتور M1 برای انتقال بار فعال می شود و همزمان با آن ۲۰ ثانیه رفت نیز آغاز می شود.

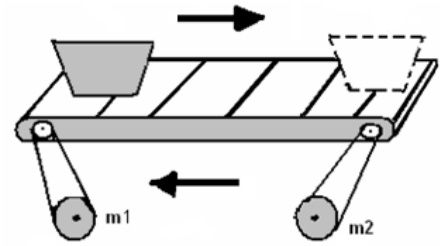
۳- بعد از سپری شدن ۲۰ ثانیه رفت، موتور M1 خاموش شده و همزمان با آن زمان ۵ ثانیه برای تخلیه بار شروع می شود. سپس بعد از اتمام ۵ ثانیه موتور M2 برای برگشت واگن شروع به کار می کند و بعد از سپری شدن ۲۰ ثانیه برگشت، موتور M2 خاموش شده و برای اجرای سیکل بعدی، زمان ۵ ثانیه برای بارگیری آغاز می شود.

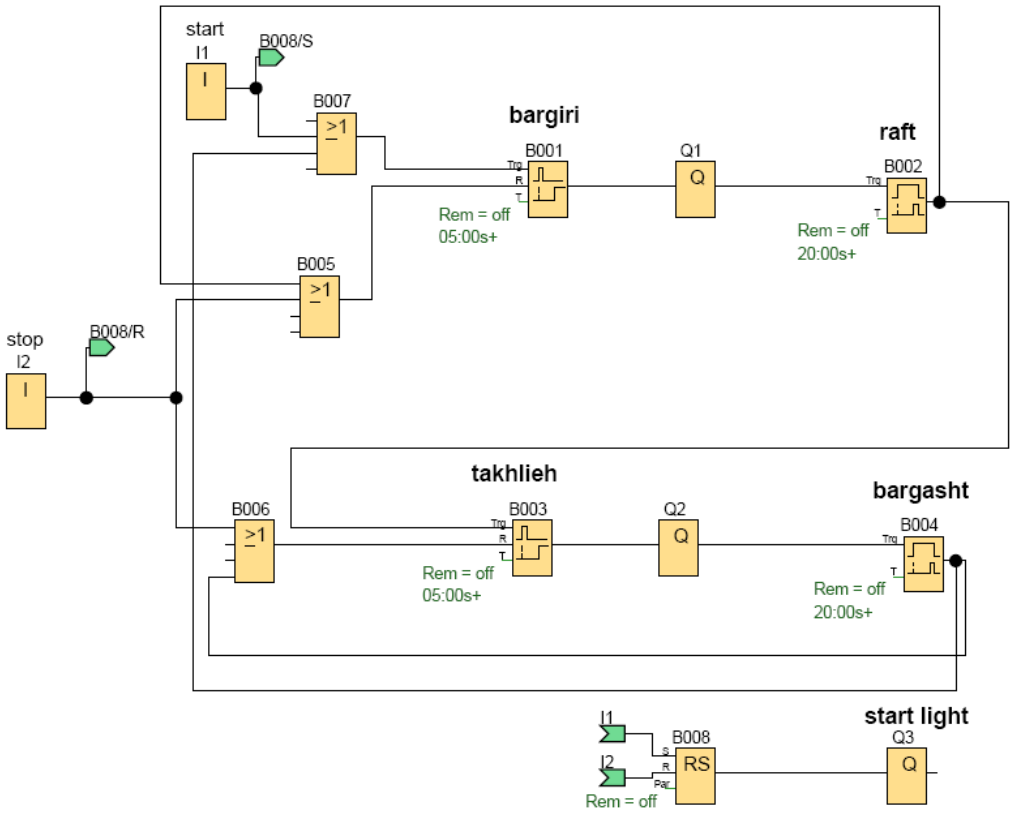
۴- با زدن شستی I2 کل مدار غیر فعال شده و عملیات متوقف می شود.

۵- هر گاه دو شستی I1 و I2 به هر علتی فعال شوند در آن صورت نباید مدار عملیات خود را آغاز کند.

۶- یک چراغ به معنای فعال بودن سیستم روشن شود.

I2→STOP, I1→START
Q2→M2, Q1→M1
Q3→START LIGHT





شکل (۶-۵۹): مدار مربوط به مثال (۶-۲۷)

مثال(۶-۲۸): کنترل سه نوار نقاله

می خواهیم برنامه عملکرد سه نقاله را مطابق شکل زیر در یک پروسه صنعتی کنترل کنیم. مدار بایستی خواسته های زیر را برآورده کند:

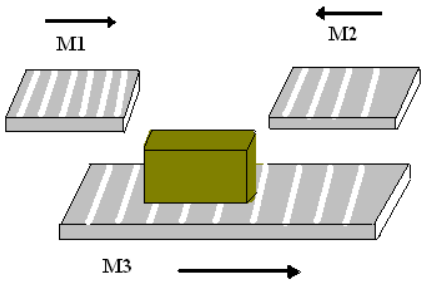
۱- نقاله های شماره ۱ و ۲ هر دو توسط دو پوش باتن روشن و خاموش می شوند. (برای هرنقاله یک استارت و استوپ جداگانه وجود دارد.)

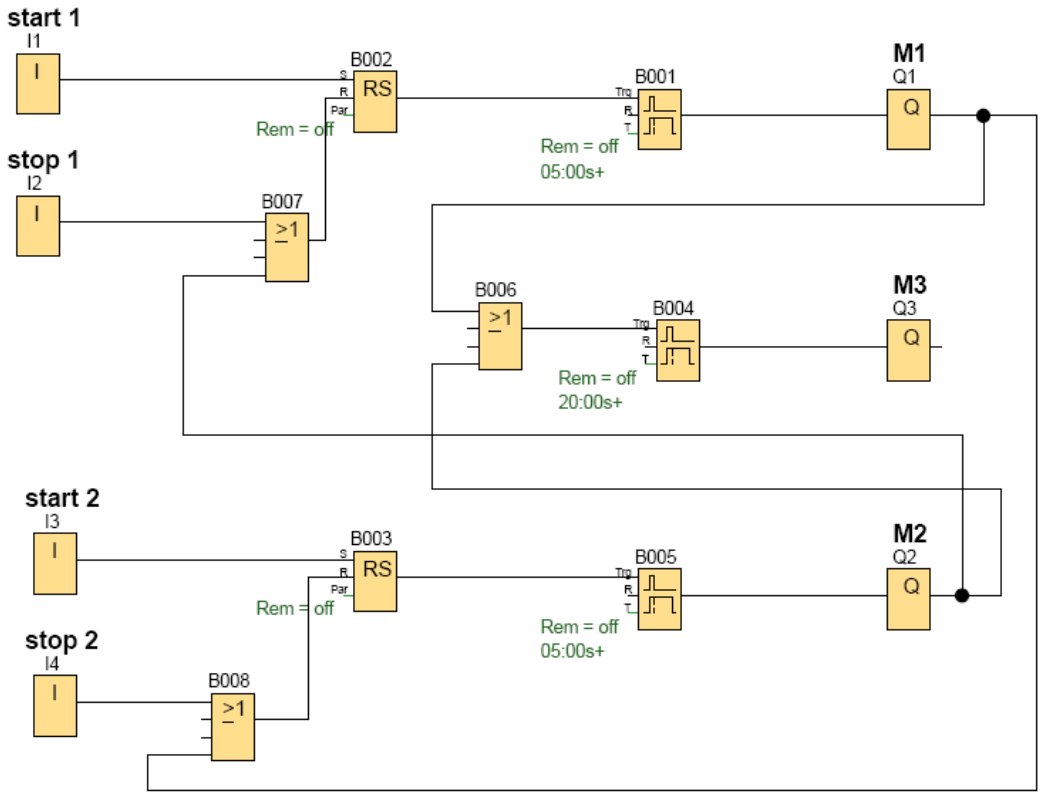
۲- با فعالیت هر یک از نقاله های ۱ یا ۲ نقاله شماره ۳ روشن می شود.

۳- نقاله های ۱ و ۲ نباید بطور همزمان روشن شوند.

۴- هنگامی که فرمان خاموش کردن نقاله ۱ یا ۲ داده می شود، نقاله مربوطه بعد از ۵ ثانیه خاموش می شود و سپس به دنبال آن نقاله شماره ۳، ۲۰ ثانیه بعد خاموش گردد.

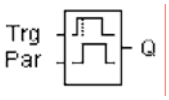
M1 - START	I1
STOP-M1	I2
START-M2	I3
STOP-M2	I4
MOTOR1	Q1
MOTOR2	Q2
MOTOR3	Q3



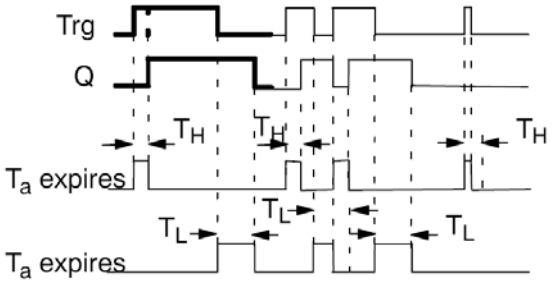


شکل (۶-۶۰): مدار مربوط به مثال (۶-۲۸)

تایمر تاخیر در وصل و قطع - On/Off-delay



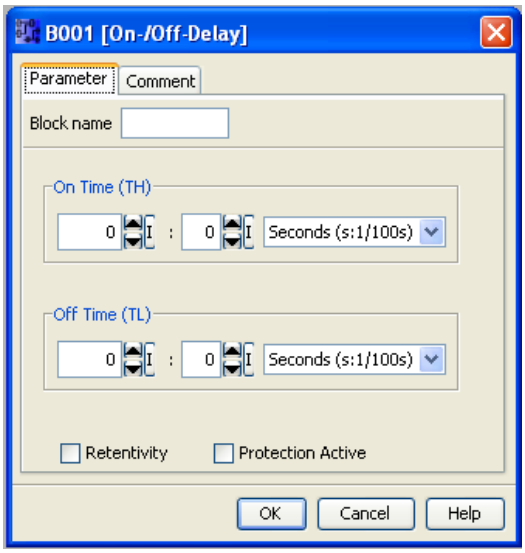
شکل (۶-۶۱) دیاگرام عملکرد این تابع را نشان می دهد.



شکل (۶-۶۱): نمودار زمانی تایمر تاخیر در وصل و قطع

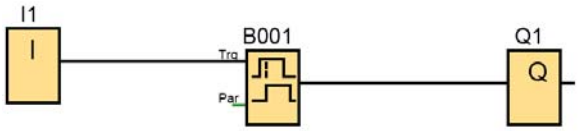
TH = زمان تاخیر در وصل و TL = زمان تاخیر در قطع

این تابع یک تایمر تاخیر در وصل و تاخیر در قطع می باشد. در این تابع خروجی بعد از سپری شدن اولین زمان TH روشن می شود و بعد از سپری شدن دومین زمان تنظیم شده TL خاموش می شود. زمان TH با لبه بالا رونده و زمان TL با لبه پایین رونده ورودی Trg شروع به شمارش می کند.



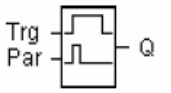
شکل (۶-۶۲): پنجره تنظیمات تایمر تاخیر در وصل و قطع

مثال(۶-۲۹): برنامه ای بنویسید که خروجی ۳ ثانیه بعد از فعال شدن ورودی روشن و ۳ ثانیه پس از غیر فعال شدن ورودی خاموش شود.

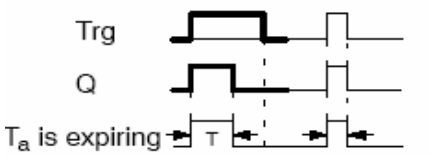


شکل (۶-۱۳): مدار مربوط به مثال (۶-۲۹)

خروجی پالسی -Wiping relay (pulse output)



در این تابع، با فعال شدن ورودی Trg، خروجی به اندازه زمان تعیین شده فعال می شود. اگر ورودی Trg قبل از سپری شدن زمان Ta صفر شود در این صورت خروجی فوراً غیر فعال می شود و اگر دوباره فعال شود، زمان Ta نیز از صفر شروع به شمارش خواهد کرد.



شکل (۶-۶۳): نمودار زمانی تابع خروجی پالسی

مثال(۶-۳۰): چراغ راهنمایی به صورت قراردادی

شرایط برنامه:

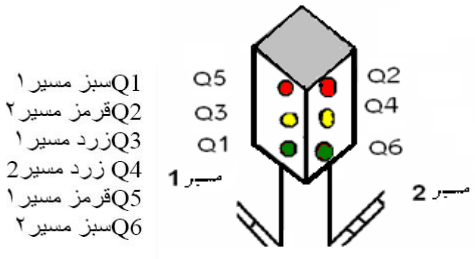
۱- هنگامی که کلید فعال می شود سیستم کنترل شروع به کار نماید.

۲- زمانیکه چراغ سبز برای مسیر ۱ روشن است چراغ قرمز برای مسیر ۲ روشن باشد.

۳- زمانیکه چراغ زرد برای مسیر ۱ روشن است چراغ زرد برای مسیر ۲ روشن باشد. (برای راحتی مسئله این طور فرض شده است)

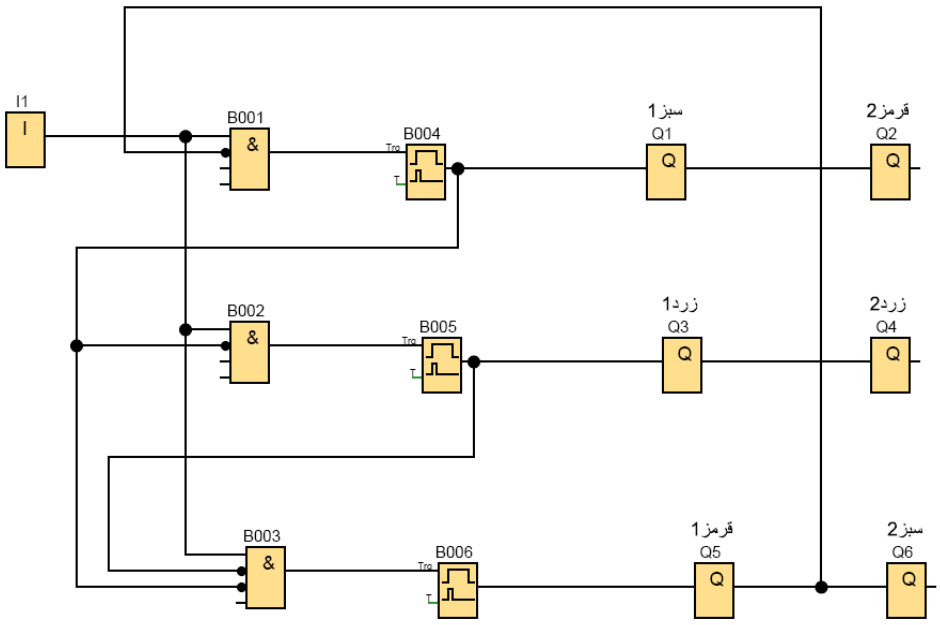
۴- زمانیکه چرغ سبز برای مسیر ۲ روشن است چراغ قرمز برای مسیر ۱ روشن باشد.

مدت زمان روشن ماندن چراغهای سبز و قرمز ۲۰ ثانیه و برای زرد ۵ ثانیه باشد.



جواب: همانطوری که از جدول زیر مشخص است در این برنامه نیاز به سه تایمر pulse output داریم، زیرا حالت چهارم تکرار حالت اول می باشد.

	مسیر۲	مسیر۱
حالت اول	سبز	قرمز
حالت دوم	زرد	زرد
حالت سوم	قرمز	سبز
تکرار حالت اول	سبز	قرمز



شکل (۶-۶۴): مدار مربوط به مثال (۶-۳۰)

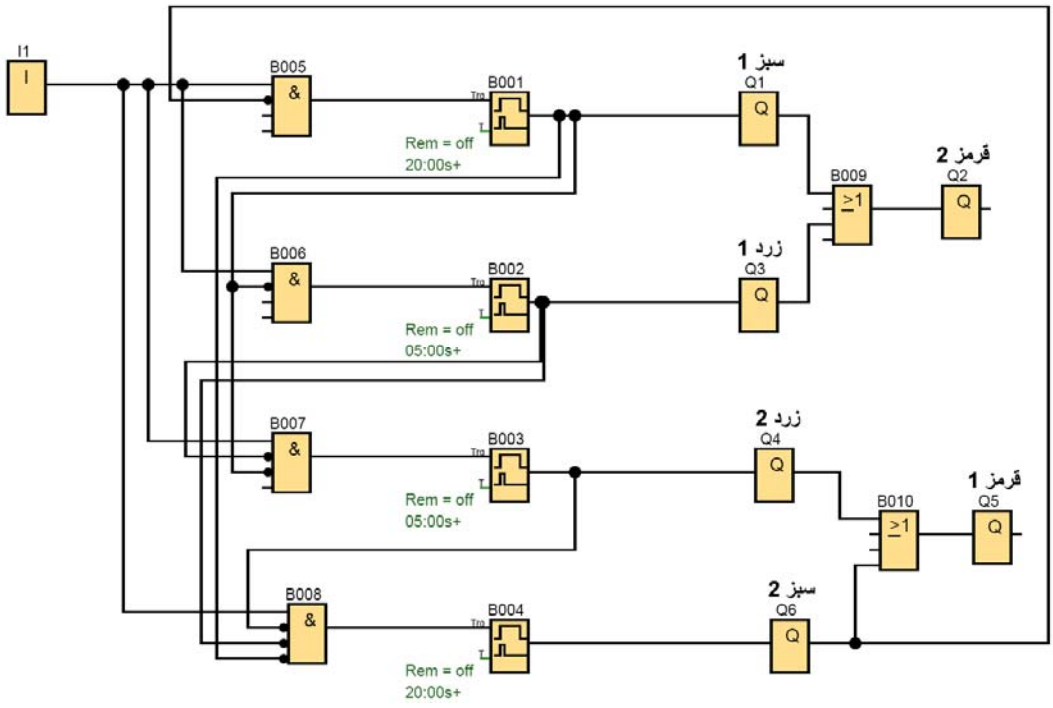
در مدار مثال (۶-۳۰) ورودی به تمام AND ها وارد شده است. علت این کار این است که هیچ کدام از ردیف ها بدون فرمان ورودی به کار نیافتند، به این کار اصطلاحاً وابسته کردن به ورودی می گوییم. قرار است که خط دوم بعد از خط اول بکار بیافتد، لذا منفی شده تایمر خط اول را به AND خط دوم داده ایم. همین روال را برای خط سوم نیز انجام داده ایم. برای به حالت چرخش درآمدن سیستم بایستی منفی خروجی تایمر خط سوم را به AND خط اول وصل کنیم. در این هنگام زمانی که عملیات حالت سوم تمام شد، با توجه به اینکه ورودی I1 قبلا یک شده، خروجی AND خط اول یک می شود، لذا خط اول دوباره به راه می افتد و همین روند تا زمانیکه ورودی I1 فعال باشد ادامه می یابد. در صورتیکه I1 غیر فعال شود، با توجه به اینکه شرط ANDها به هم می خورد کل سیستم متوقف می شود. بدیهی است که در سیستم فوق بایستی ورودی از نوع کلید باشد. درصورت شستی بودن بایستی از رله RS استفاده کنیم.

مثال(۶-۳۱): چراغ راهنمایی به صورت واقعی

اکنون برنامه فوق برای ۲ مسیر با در نظر گرفتن شرایط واقعی بنویسید.

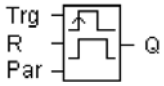
جواب: از جدول زیر پیداست که ۴ حالت وجود دارد، لذا نیاز به چهار تایمرخروجی پالسی داریم.

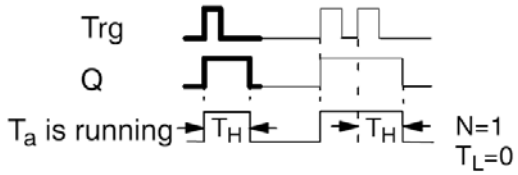
	مسیر۲	مسیر۱
حالت اول	سبز	قرمز
حالت دوم	زرد	قرمز
حالت سوم	قرمز	سبز
حالت چهارم	قرمز	زرد
تکرار حالت اول	سبز	قرمز



شکل (۶-۶۵): مدار مربوط به مثال (۶-۳۱)

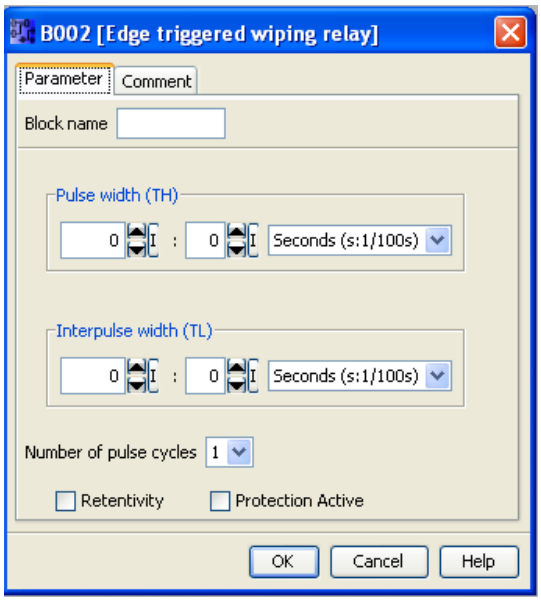
رله لغزان با لبه راه انداز –Edge triggered Wiping relay





شکل (۶-۶۶): نمودار زمانی تایمر Edge triggered Wiping relay

در این تایمر خروجی فقط با لبه بالارونده ورودی روشن می شود. با لبه بالارونده ورودی، خروجی بعد از مدت زمان تنظیم شده TL روشن می شود. و بعد از سپری شدن زمان TH خاموش می شود. اینکار به اندازه N بار که در پنجره مشخصات تابع قابل تعریف است می تواند صورت بگیرد (حداکثر N=9). در مثال نشان داده شده در دیاگرام TL=0 می باشد.

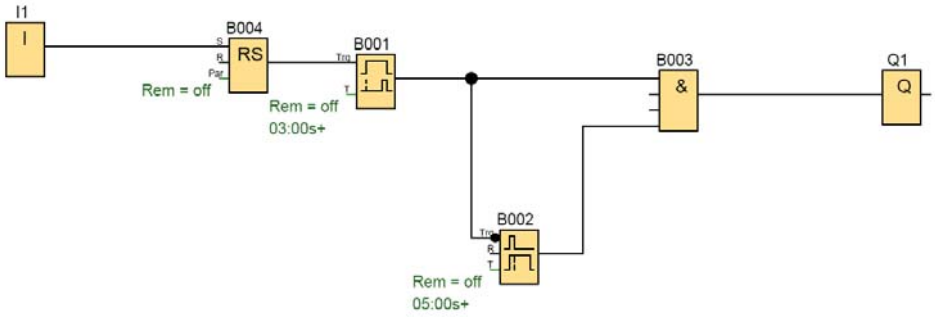


شکل (۶-۶۷): پنجره تنظیمات Edge triggered Wiping relay

نکته : در سخت افزار سری OBA3 , OBA2 فقط پارامتر TH موجود می باشد. همچنین ورودی R نیز موجود نیست. اگر N = 1 باشد، از این تایمر می توان به عنوان تاخیر در خاموش مستقل از ورودی استفاده کرد.

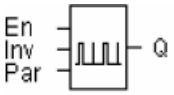
مثال(۶-۳۲): بدون استفاده از تایمر Edge triggered Wiping relay برنامه ای بنویسید که خروجی،3 ثانیه بعد از فعال شدن ورودی به مدت 5 ثانیه روشن مانده، سپس خاموش گردد.

این برنامه را می توان براحتی با تایمر Edge triggered انجام داد، اما مسئله گفته بدون استفاده از این تایمر. بنابراین بایستی از تایمرهای اولیه استفاده کنیم.

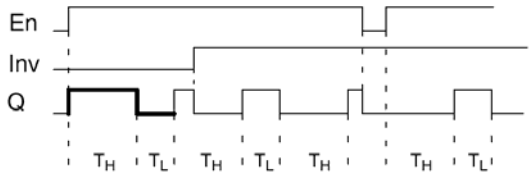


شکل (۶-۶۸): مدار مربوط به مثال (۶-۳۲)

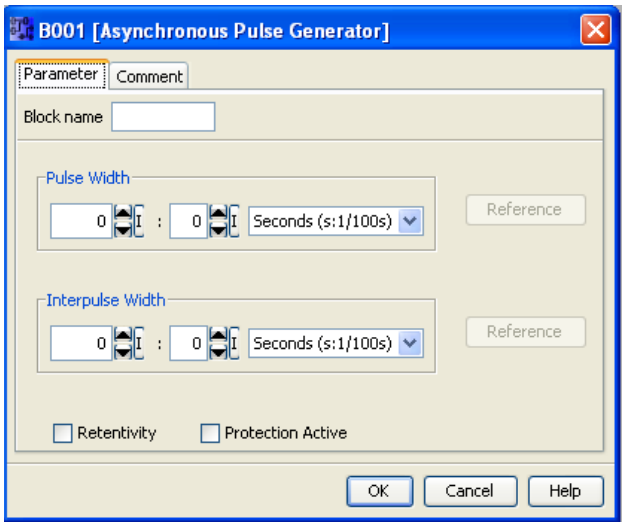
مولد پالس غیر همزمان یا آسنکرون-Asynchronous pulse generator



از این تابع برای تولید پالسهای غیرمتقارن استفاده می شود. کاربرد این تابع در ایجاد چراغهای چشمک زن در موقع بـروز خطـا در سیستم و همچنین برای شبیه سازی شمارنده ها و... می باشد. برای راه اندازی این تابع ورودی En حتماً بایستی یک باشد.



شکل (۶-۶۹): نمودار زمانی مولد پالس آسنکرون



شکل (۶-۷۰): پنجره تنظیمات مولد پالس آسنکرون

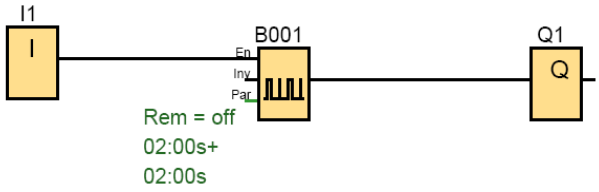
TH: عرض پالس و TL: عرض وقفه پالس

برای معکوس کردن عملیات (یعنی تعویض زمان TH با TL) می توانید ورودی INV را فعال کنید.

مثال(۶-۳۳): چراغ چشمک زن

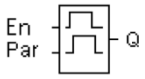
برنامه ای بنویسید که از هر ۴ ثانیه یک بار خروجی روشن شود.

می توان زمان عرض پالس و و قفه پالس را هرکدام ۲ ثانیه انتخاب کرد.

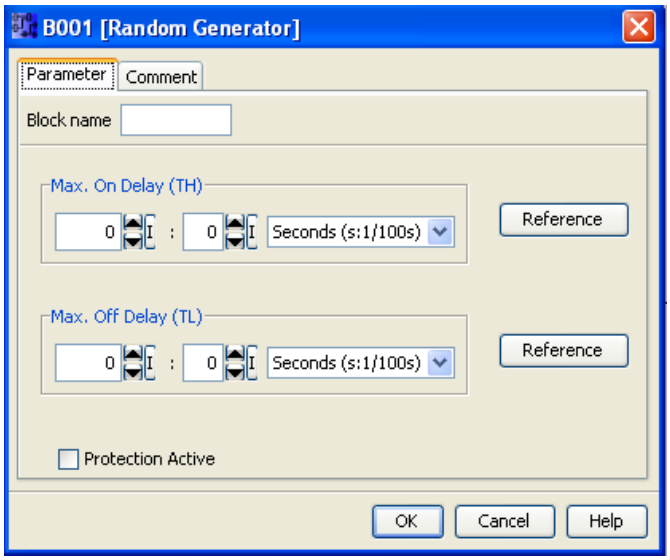


شکل (۶-۷۱): مدار مربوط به مثال (۶-۳۳)

مولد تصادفی- Random generator

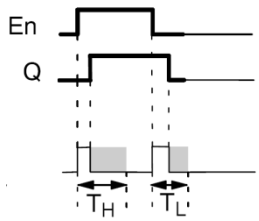


مولد پالس با پهنای صفر ویک تصادفی، به طوریکه مقادیر ماکزیمم زمانهای TH (زمان رسیدن خروجی از خاموشی به روشنی) و TL (زمان رسیدن خروجی از روشنی به خاموشی) را در پنجره مشخصات بلوک تنظیم می کنیم.



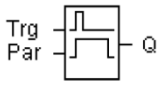
شکل (۶-۷۲): پنجره تنظیمات مولد تصادفی

اکنون با فعالسازی پایه En، این بلوک راه اندازی شده و به طور تصادفی یک TH را در محدوده ای که تعریف کرده ایم، اختیار می کند. و خروجی را پس از آن روشن می کند. حال که خروجی روشن است با غیر فعال کردن پایه En دوباره بلوک راه اندازی شده و به طور تصادفی یک TL را در محدوده ای که ما تعریف کرده ایم، انتخاب کرده و پس از گذشت این زمان خروجی را خاموش می کند. نحوه این تغییرات به وضوح در دیاگرام زمانی (۶-۷۳) مشهود است.



شکل (۶-۷۳): نمودار زمانی مولد تصادفی

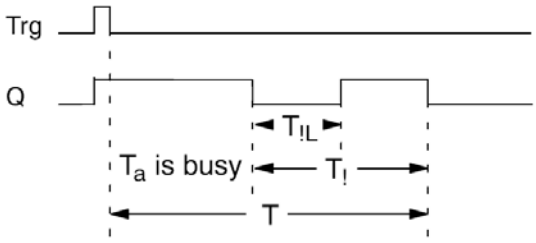
کلید روشنائی راه پله – Stairway lighting switch



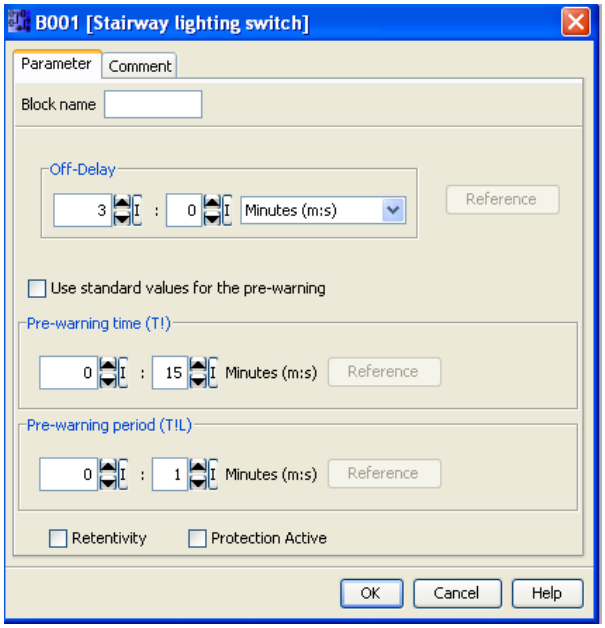
این تابع همان تایمر راه پله می باشد که در آپارتمانها موجود می باشد. خروجی با لبه بالا رونده پالس ورودی Trg فعال می شود و بعد از سپری شدن زمان T خاموش می شود. لازم به توضیح است که زمان کل با لبه پایین رونده Trg فعال می شود و هر بار که ورودی را فعال کنیم، زمان از صفر شروع به شمارش می کند، یعنی مبنای زمان آخرین لبه پالس ورودی است. همچنین این تابع مجهز به یک هشدار می باشد. در پنجره تنظیمات این تابع پارامترهای هشدار به صورت زیر تعریف می شوند:

Pre – warning time (T!): زمانی است که قبل از زمان اتمام کل، هشدار شروع می شود.

Pre – warning period (TL!): زمان خاموشی بعد از هشدار

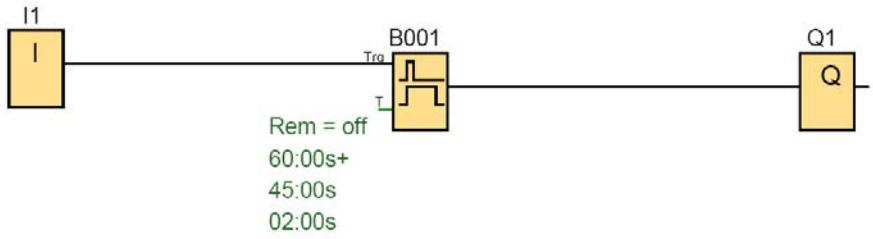


شکل (۶-۷۴): نمودار زمانی کلید روشنائی راه پله



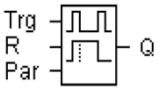
شکل (۶-۷۵): پنجره تنظیمات کلید روشنائی راه پله

مثال(۶-۳۴): اگر کل زمان روشن ماندن خروجی،۱ دقیقه باشد و زمان (T!) = ۱۵ ثانیه و (TL!) را برابر ۲ ثانیه فرض کنیم. در آن صورت خروجی در ثانیه ۴۵ ام به مدت ۲ ثانیه خاموش شده سپس روشن می شود تا اینکه کل زمان ۱ دقیقه تمام شود.

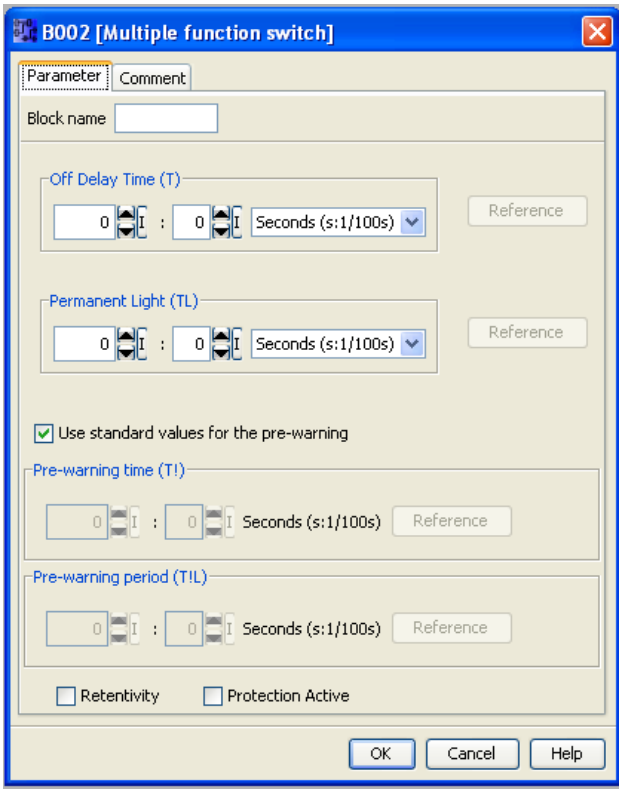


شکل (۶-۷۶): مدار مربوط به مثال (۶-۳۴)

کلید با عملکرد چند گانه Multiple function switch



این تابع در واقع چراغ راه پله بصورت دائم و معمولی می باشد. پنجره تنظیمات این تابع را در شکل زیر مشاهده می کنید. برای راحتی بیان مسئله فرض کنید هشدار وجود ندارد.

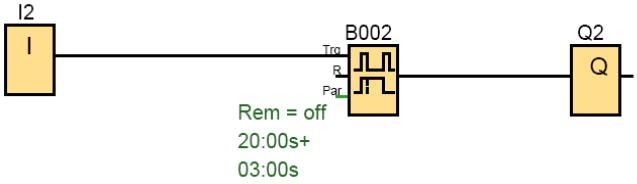


شکل (۶-۷۷): پنجره تنظیمات کلید با عملکرد چند گانه

در حالت اول در صورتیکه زمان فعال بودن ورودی کمتر از زمان(Permanent Light (TL باشد، این حالت مثل چراغ راه پله معمولی که در بالا ذکر شد، به کار خود ادامه داده و پس از تمام شدن زمان کل T خاموش می شود.

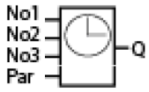
در حالت دوم در صورتیکه زمان فعال بودن ورودی بیش از زمان(Permanent Light (TL باشد، در این حالت چراغ راه پله بطور دائم روشن خواهد ماند و در صورتیکه شستی ورودی دوباره فشار داده شود و یا ریست فعال شود، خروجی خاموش خواهد شد.

مثال(۶-۳۵): چراغ راه پله را با تابع Multiple function switch به گونه ای طراحی کنید که اگر فرد ۳ ثانیه دست خود را بر روی کلید نگه دارد چراغ دائم روشن بماند. و اگر زودتر از ۳ ثانیه دست خود را از روی شستی بردارد چراغ بعد از ۲۰ ثانیه خاموش شود.

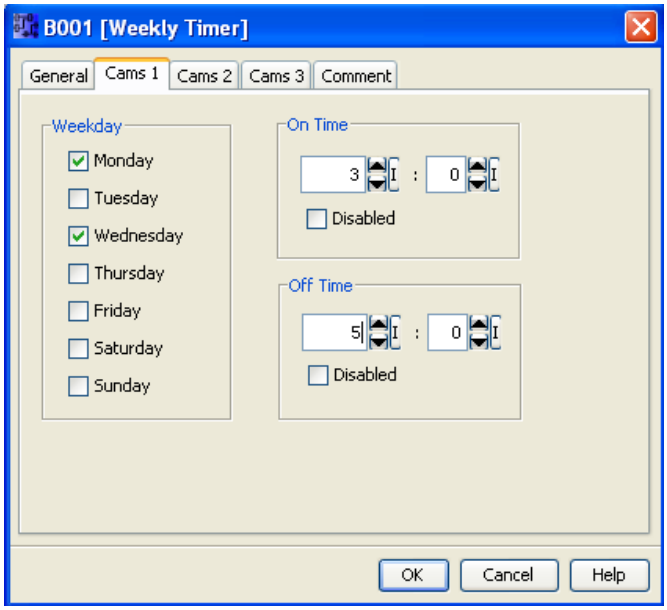


شکل (۶-۷۸): مدار مربوط به مثال (۶-۳۵)

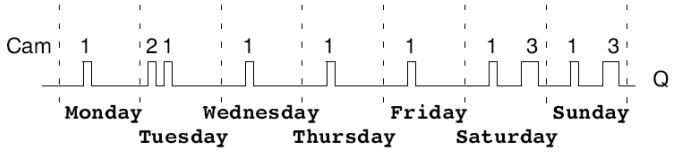
تایمر هفتگی – Weekly timer



این تایمر یک تایمر هفتگی می باشد که در ساعات و روزهای قابل تنظیم خروجی آن فعال خواهد شد. این تایمر ورودی نداشته، و ۳ قسمت برای تنظیم زمانهای دلخواه دارد. بدین معنی که می توان برای یک یا چند روز هفته سه ساعت شروع و پایان مشخص کرد. لازم به یاد آوری است که برای این تایمر مقیاس زمانی براساس ساعت و دقیقه می باشد.



شکل (۶-۷۹): پنجره تنظیمات تایمر هفتگی



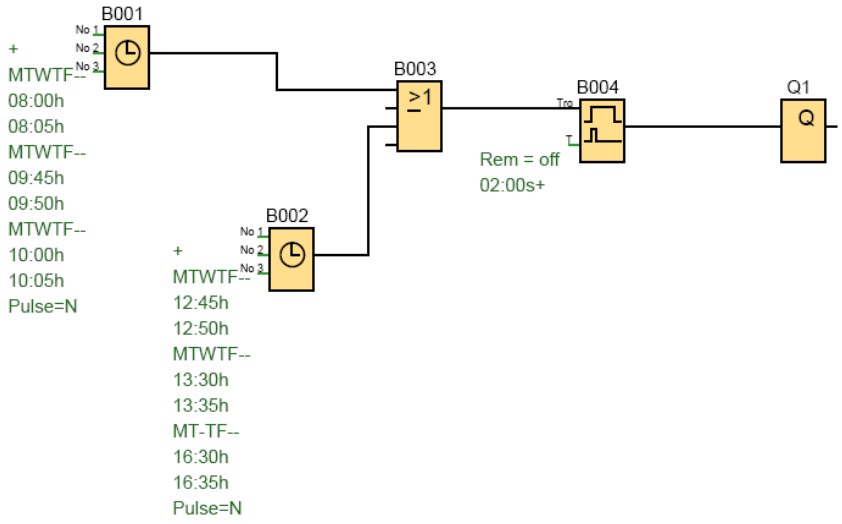
Cam 1:	Daily:	06:30 h to 8:00 h
Cam 2:	Tuesday:	03:10 h to 04:15 h
Cam 3:	Saturday and Sunday:	16:30 h to 23:10 h

شکل (۶-۸۰): نمودار زمانی تایمر هفتگی

ذکر این نکته ضروری است که قطع شدن برق هیچ تاثیری بر روی عملکرد تابع زمان سنج ندارد.

مثال(۶-۳۶): سیستم زنگ مدرسه

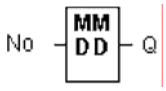
بوسیله تایمر هفتگی زنگ مدرسه را برای شروع و زنگ سیاحت و پایان مدرسه طراحی کنید، بطوری که در روزهای دوشنبه تا جمعه در ساعتهای ۸-۹:۴۵ – ۱۰- ۱۲:۴۵ -۱۳:۳۰ - ۱۶:۳۰ زده شود و مدت زمان زنگ زدن را تا ۲ ثانیه در نظر بگیرید.



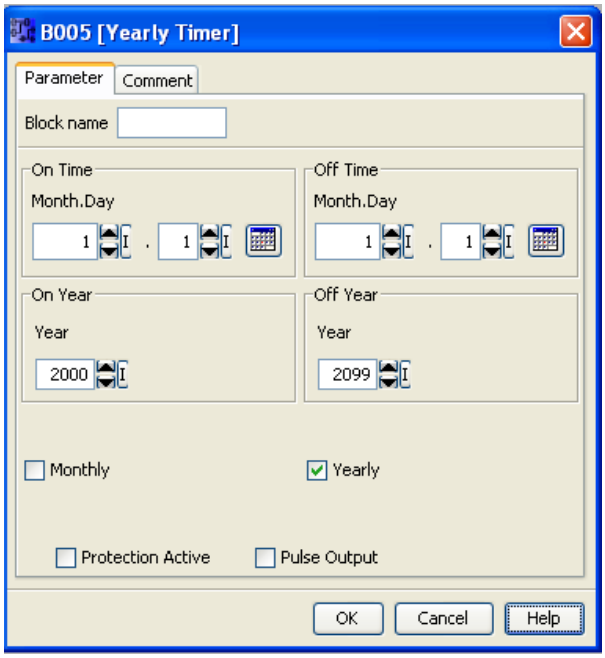
شکل (۶-۸۱): مدار مربوط به مثال (۶-۳۶)

با توجه به این که برای ایام هفته ۶ زمان وجود دارد، و با در نظر گرفتن اینکه در هر تایمر حداکثر۳ زمان برای ایام هفته قابل تنظیم است، لذا دو تایمر را با یکدیگر موازی کرده ایم. همچنین با توجه به اینکه این تایمرها مقیاس زمانی ثانیه ندارند، لذا در مسیر زنگ یک تایمر Pulse output گذاشته ایم.

تایمر سالیانه - Yearly timer



در این تابع خروجی توسط تاریخ روشن و خاموش می شود. یک تاریخ برای روشن شدن و یک تاریخ برای خاموش شدن، از ۱۲ ماه تعیین می شود. خروجی از یک تاریخ تنظیم شده روشن می شود تا تاریخ بعدی، که نشان دهنده خاموش شدن خروجی می باشد.



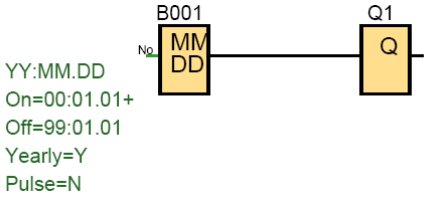
شکل (۶-۸۲): پنجره تنظیمات تایمر سالیانه

در صورتیکه گزینه yearly فعال باشد، ازسال مبدا تا سال مقصد(به عنوان مثال ۲۰۰۰ تا سال ۲۰۹۹) در ماه و روز تنظیم شده خروجی روشن و در ماه و روز تنظیم شده، خاموش می شود. و در صورتیکه گزینه monthly فعال باشد، هر ماه در روز تنظیم شده خروجی روشن و در روز تنظیم شده خروجی خاموش می شود.

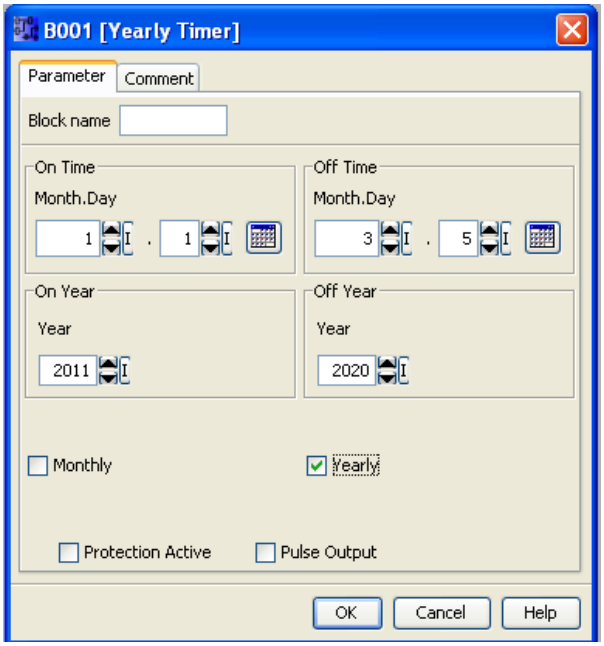
در صورتیکه گزینه Pulse output فعال باشد، خروجی در تاریخ On Time تنظیم شده به اندازه یک سیکل زمانی روشن وسپس خاموش می شود. مسلم است که در این حالت دیگر زمان تنظیم شده Off Time معنی نخواهد داشت، چرا که خروجی خود به خود بعد از یک سیکل خاموش خواهد شد.

مثال(۶-۳۷): برنامه ای بنویسید که خروجی Q از سال ۲۰۱۱ تا سال ۲۰۲۰ هر سال در اول ژانویه روشن و در پنجم مارس خاموش شود.

برنامه این مسئله در شکل زیر آورده شده است. پنجره تنظیمات تایمر سالیانه نیزدر شکل(۶-۸۴) نشان داده شده است.



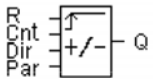
شکل (۶-۸۳): مدار مربوط به مثال (۶-۳۷)



شکل (۶-۸۴): پنجره تنظیمات تایمر سالیانه مربوط به مثال(۶-۳۷)

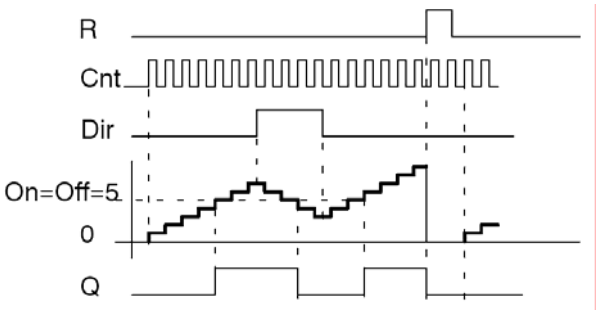
شمارنده ها- counters

شمارنده بالا/ پایین شمار- Up/down counter



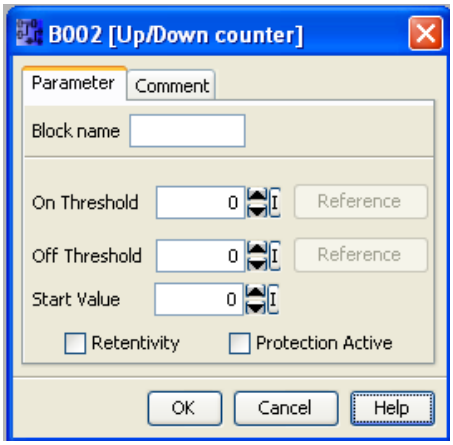
شمارش قطعات روی نقاله ها، شیشه های نوشابه و.... در صنعت از اهمیت ویژه ای برخوردار است. برای این منظور از شمارنده‌ها استفاده می شود.

شمارنده بالا/ پایین شمار یک شمارنده دو جهته می باشد که می تواند پالسهائی کـه ورودی آن توسـط سنسـورها و غیـره وارد می‌شود، را شمارش کند و در مقدار تنظیم شده، خروجی را فعال و غیر فعال کند.



شکل (۶-۸۵): نمودار زمانی شمارنده بالا شمار/ پایین شمار

خروجی این شمارنده زمانی فعال می شود که مقدار شمارش بیشتر یا برابر تنظیم شده شود.



شکل (۶-۸۶): پنجره تنظیمات شمارنده بالا شمار/ پایین شمار

On threshold : آستانه روشن شدن

Off threshold : آستانه خاموش شدن

Start value: مقدار اولیه که شمارش می تواند از آنجا شروع شود.

پایه های این شمارنده به شرح زیر می باشد:

ورودی cnt: این پایه مقدار شمارش را با توجه به هر پالس ورودی و با لبه بالا رونده تنظیم می کند.

Reset: با فعال شدن Reset تمامی مقادیر پاک می شوند.

Dir: جهت شمارش را نیز می توانید با ورودی Dir تنظیم کنید.

Dir=1 شمارش به طرف پایین

Dir=0 شمارش به طرف بالا۱-۲-۳...

قوانین محاسبه این شمارنده به شرح زیرمی باشد:

- اگر On threshold(آستانه روشن شدن)، بزرگتر یا مساوی Off threshold (آستانه خاموش شدن) باشد:

Q=1 اگر cnt ≥ on

Q=0 اگر cnt < off

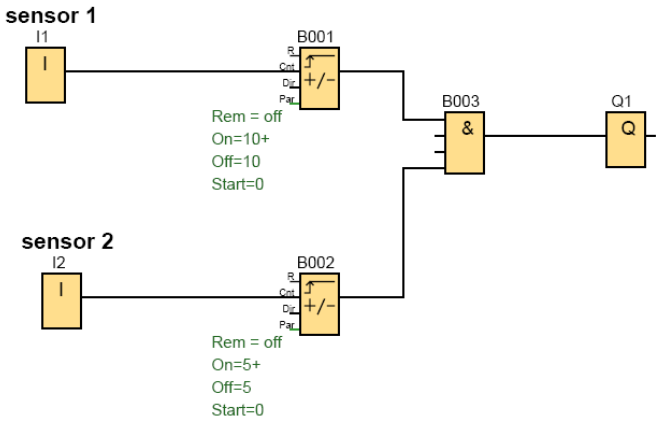
- اگر On threshold آستانه روشن شدن، کوچکتر از Off threshold آستانه خاموش شدن باشد:

Q=1 اگر on ≤ cnt < off

در بقیه جاها خروجی صفر می باشد.

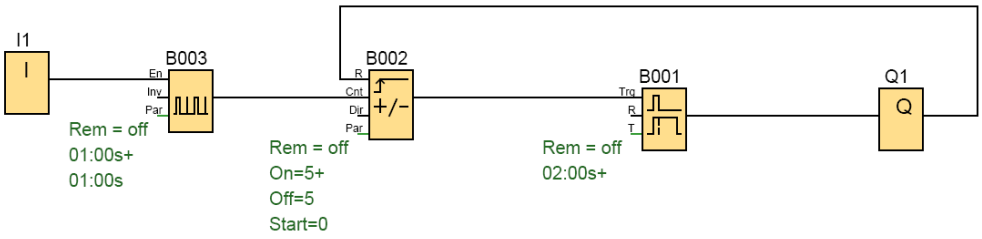
نـکـتـه : اگر بخواهید مقدار شمارش، با قطع شدن برق تغییر نکند گزینه Retentivity را از پنجره تنظیمات فعال کنید.

مثال(۶-۳۸): برنامه ای بنویسید که در آن دستگاه بسته بندی زمانی روشن شود که سنسور اول تعداد ده قطعه از نقالـه اول و سنسور دوم تعداد ۵ قطعه از نقاله دوم را بشمارد.



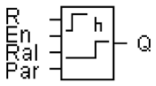
شکل (۶-۸۷): مدار مربوط به مثال (۶-۳۸)

مثال(۶-۳۹): برنامه ای بنویسید که شمارنده بسته های روی یک کانوایر را شمرده و زمانیکه تعداد بسته ها به عدد ۵ رسید شمارش را به مدت ۲ ثانیه متوقف کرده و سپس چرخه را تکرار کند. فرض کنید در این دو ثانیه موتور اهرمی، جهت جابجا کردن بسته ها از روی کانوایر فعال می شود.

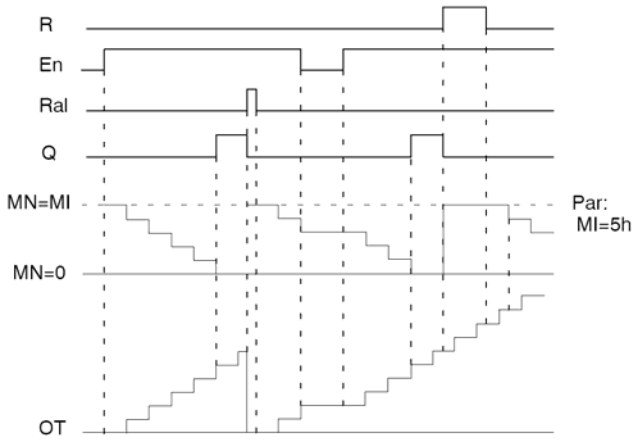


شکل (۶-۸۸): مدار مربوط به مثال (۶-۳۹)

شمارنده ساعتی Hours counter



این تابع، یک ساعت می باشد که برای ذخیره کردن مدت زمان کارکرد وسیله مورد نظر مورد استفاده قرار می گیرد. وقتی ورودی En فعال می شود پارامتر زمان تنظیم شده (MI) شروع به فعالیت می کند. خروجی زمانی فعال می شود که زمان باقی مانده MN به پایان برسد.



شکل (۶-۸۹): نمودار زمانی شمارنده ساعتی

MI: فاصله زمانی طی شده

MN: زمان باقی مانده

OT: کل زمان سپری شده بعد از آخرین سیگنال بر حسب ورودی Ral می باشد. یعنی زمان شروع برای شمارش ساعت که برحسب ساعت تعیین می گردد.

MN به صورت زیر تعیین می شود:

اگر MI>OT باشد آنگاه MN=MI-OT خواهد بود.

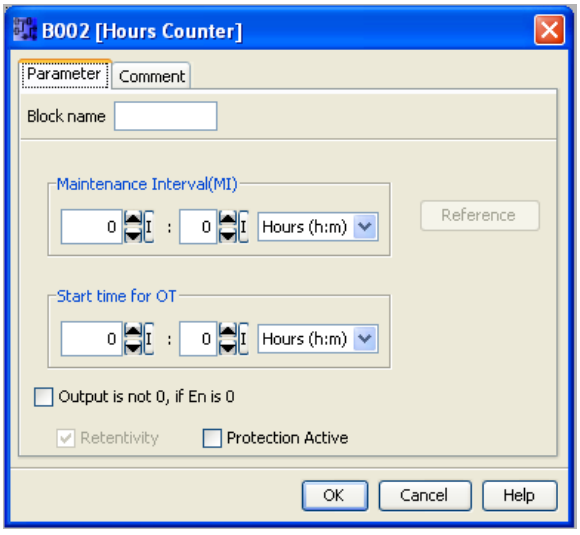
اگر OT>MI باشد آنگاه MN=2MI-OT خواهد بود.

مثال:

MN = 70 » = = MI = 100, OT = 130

در صورتیکه ورودی Ral(reset all) نشانده شود تمامی مقادیر صفر می شوند.

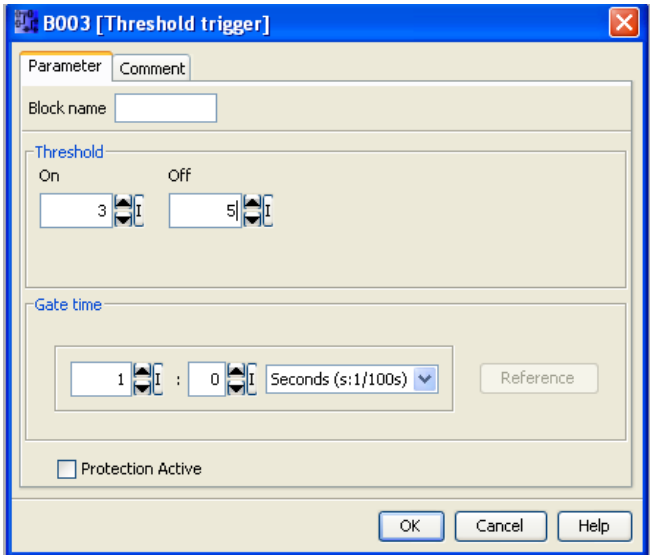
اگر ورودی R نشانده شود شمارش داخلی OT حالت اجرا را دنبال کرده و تغییر نمی کند، ولی خروجی خاموش می شود. به عبارتی MN=MI می شود.



شکل (۶-۹۰): پنجره تنظیمات شمارنده ساعتی

تحریک شونده با فرکانس-Threshold trigger

این تابع به عنوان یک شمارنده فرکانس به کار برده می شود. اگر تعداد شمارش در محدوده قابل شمارش ما بود، خروجی یک می شود. این تابع دارای یک پایه ورودی است که فرکانس ورودی یعنی تعداد تکرار پالسهای ایجاد شده توسط ورودی در واحد زمان، را می شمارد. در پنجره مشخصات این تابع Gate Time را به عنوان محدوده زمانی شمارش، تعیین می کنیم. تعداد لازم و به عبارتی فرکانسهای ابتدا و انتها در فاصله زمانی مورد نظر را که خروجی باید روشن یا خاموش شود، در قسمت Threshold تعیین می شوند.



شکل (۶-۹۱): پنجره تنظیمات تابع Threshold trigger

توجه به این نکته قابل اهمیت است که اگر از ورودیهای I5 و I6 استفاده شود، فرکانس ورودی می تواند تا حد 4KHz باشد، و اگر از بقیه ورودیها استفاده شود این عدد 2KHz خواهد بود.

قوانین محاسبه این شمارنده به شرح زیرمی باشد:

- اگر On threshold (آستانه روشن شدن)، بزرگتر یا مساوی Off threshold (آستانه خاموش شدن) باشد:

اگر $f_a > On$ آنگاه $Q = 1$

اگر $f_a \leq Off$ آنگاه $Q = 0$

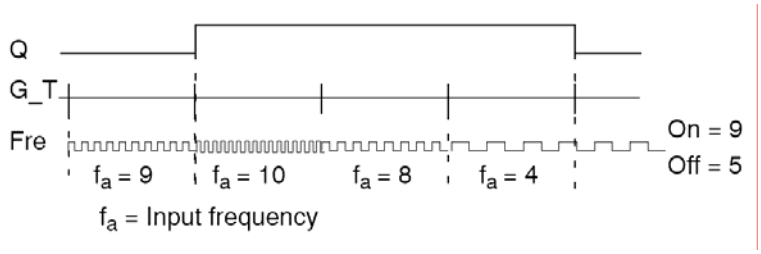
- اگر On threshold (آستانه روشن شدن)، کوچکتر از Off threshold (آستانه خاموش شدن) باشد:

اگر $On \leq f_a < Off$ آنگاه $Q=1$

در بقیه جاها خروجی صفر می باشد.

در مورد دوم یعنی اگر تعداد پالسهای ورودی در مدت زمان G_T بین مقادیر on و off باشد، روشن می شود.

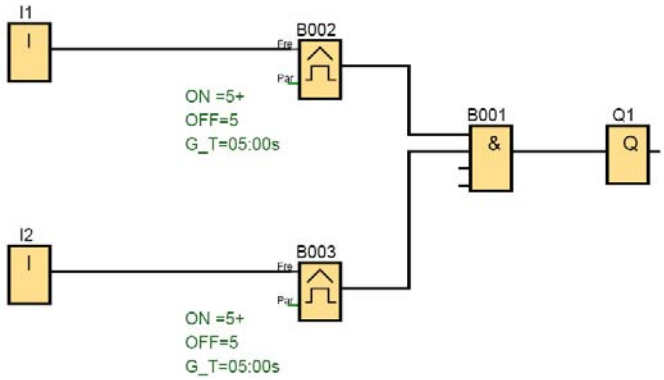
نحوه شمارش با توجه به حالت ورودی ها در دیاگرام زمانی این شمارنده در یک مثال بصورت شکل (۶-۹۲) نشان داده شده است.



شکل (۶-۹۲): نمودار زمانی تابع Threshold trigger

مثال(۶-۴۰): در یک کارخانه یک سنسور در ابتدای کانوایر اول وجود دارد که قطعات ورودی به کانوایر را می شمارد و سنسور دیگری نیز وجود دارد که قطعات ورودی به کانوایردوم را می شمارد. در هر ۵ ثانیه هر سنسور بایستی ۵ قطعه بشمارد. برنامه ای بنویسید که در صورت برابر بودن قطعات ورودی دو کانوایر چراغی روشن شود.

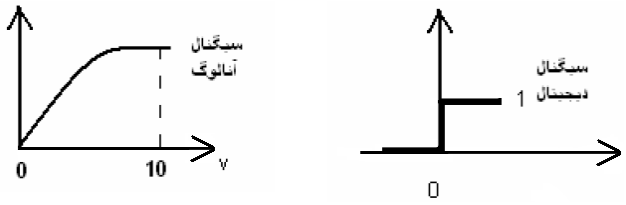
با استفاده از دو عدد شمارنده Threshold trigger این مسئله را طراحی می کنیم. مقدارG_T را برابر ۵ ثانیه و مقدار On=off 5= تنظیم می کنیم. این تابع هر ۵ ثانیه یکبار عمل شمارش را انجام می دهد. اگر این عدد برابر ۵ بود، خروجی هر تابع روشن می شود.



شکل (۶-۹۳): مدار مربوط به مثال (۶-۴۰)

توابع آنالوگ - Analog Functions

آنالوگ به سیگنالی گفته می شود که پیوستگی در آن حفظ شود و متغییر نیز باشد مانند سیگنال سنسور دما، فشار و سطح مایع یک مخزن.



شکل (۶-۹۴): سیگنال آنالوگ و دیجیتال

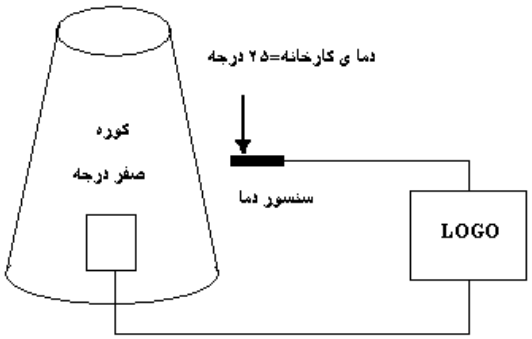
تحریگر آستانه آنالوگ-Analog threshold trigger

در این تابع که ورودی آن آنالوگ و خروجی آن دیجیتال می باشد، خروجی زمانی روشن می شود که مقدار آنالوگ از مقدار تنظیم شده برای آستانه روشن شدن تجاوز کند. هر گاه مقدار آنالوگ برابر یا کمتر از مقدار تنظیم شده برای آستانه خاموش شدن شود، خروجی خاموش می شود. در تمامی توابع آنالوگ از دو پارامتر gain و offset استفاده می شود. سیگنالی که از سنسور به لوگو وارد می شود بین ۰-۱۰ ولت و ۰-۲۰ میلی آمپر می باشد که این سیگنال برای لوگو ارزش ۰ تا ۱۰۰۰ را دارد. یعنی به عنوان مثال زمانیکه سطح یک مخزن خالی خالی است سیگنال صفر ولت به لوگو وارد می شود و زمانیکه پر پر است سیگنال ده ولت به لوگو وارد می شود. در این حالت هر گام برابر یک صدم خواهد بود. در صورتیکه رنج اندازه گیری از محدوده صفر تا ۱۰۰۰ خارج باشد یا انحرافی

در اندازه گیری صورت گیرد، از از دو پارامتر (بهره) gain و(انحراف) offset جهت کالیبره کردن رنج استفاده می شود. بدین صورت که مقدار واقعی سیگنال با توجه به مقدار اندازه گیری شده و پارامترهای تنظیم شده به شرح زیر محاسبه می گردند.

مقدار واقعی سیگنال آنالوگAx= offset + (Gain . مقدار اندازه گیری شده سیگنال آنالوگ Ax)

مثال: یک سنسور دما، دمای بین ۰ تا ۱۲۰۰ درجه سانتیگراد یک کوره را به سیگنال ۰ تا ۱۰ولت تبدیل می کند. قرار است که وقتی دمای کوره صفر درجه سانتی گراد شود درب کوره توسط لوگو باز شود تا آلیاژ وارد کوره شود. با فرض آنکه دمای کارخانه در جائی که سنسور دما قرار دارد برابر ۲۵ درجه باشد، برای آنکه لوگو کنترل مناسبی روی فرآیند انجام بدهد، مقادیر بهره وانحراف را تعیین کنید.



شکل (۶-۹۵)

با نسبت بندی اینکه بهره، زمانی که ماکزیمم مقدار ۱۰۰۰ است برابر یک می باشد، مقدار بهره جدید برای مقدار ۱۲۰۰ برابر ۱/۲ خواهد شد. بطوریکه ۱۲۰۰=۱۰۰۰*۱/۲

مسئله گفته دمای کارخانه ۲۵ درجه می باشد که بر کار سنسور دخالت می کند. یعنی زمانیکه دمای کوره صفر درجه است و طبق گفته مسئله بایستی در کوره باز شود، دمای ۲۵ درجه وارد کوره می شود، واین دما ناشی از دمای محیط است. بنابراین یک دمای مزاحم است. لذا برای آنکه دمای صفر درجه که دمای واقعی است به لوگو وارد شود، بایستی آفست را ۲۵- درجه تعریف کنیم. یعنی وقتی دمای اندازه گیری شده ۲۵ درجه می باشد، دمای واقعی صفر درجه خواهد بود. زیرا ۰=۲۵-۲۵

قوانین محاسبه روشن یا خاموش شدن خروجی به شرح زیر می باشد:

- اگر On threshold(آستانه روشن شدن)، بزرگتر یا مساوی Off threshold (آستانه خاموش شدن) باشد:

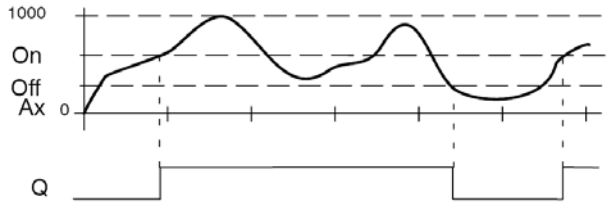
اگر مقدار واقعی On<Ax آنگاه Q = 1

اگر مقدار واقعی On≥Ax آنگاه Q = 0

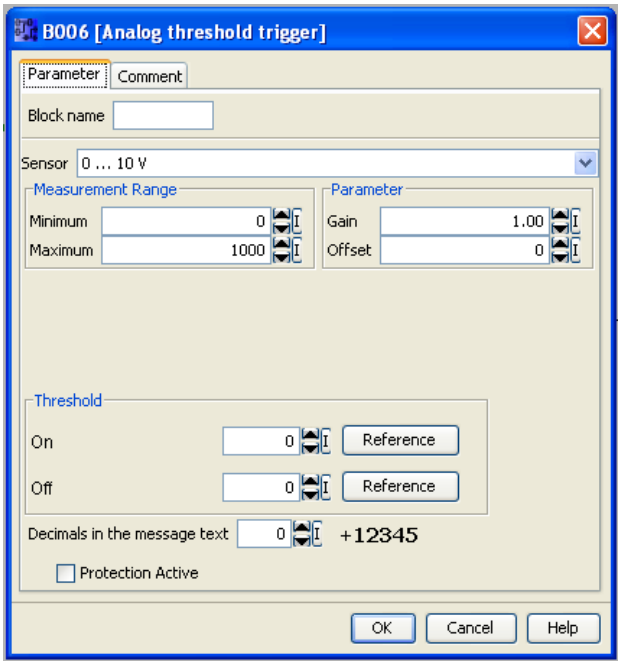
- اگر On threshold آستانه روشن شدن، کوچکتر از Off threshold آستانه خاموش شدن باشد:

اگر مقدار واقعی On ≤ Ax < Off آنگاه Q = 1

در بقیه جاها خروجی صفر می باشد.



شکل (۶-۹۶): نمودار زمانی Analog threshold trigger

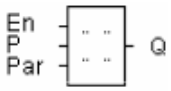


شکل (۶-۹۷): پنجره تنظیمات تابع Analog threshold trigger

اعداد اعشاری فقط در زمان نمایش On وOff و مقدار AX در یک متن پیام قابل اجرا می باشند. همچنین مقدار On وOff را برای مقایسه نمی توان اجرا کرد، زیرا در حین مقایسه از علامت اعشاری چشم پوشی می شود.

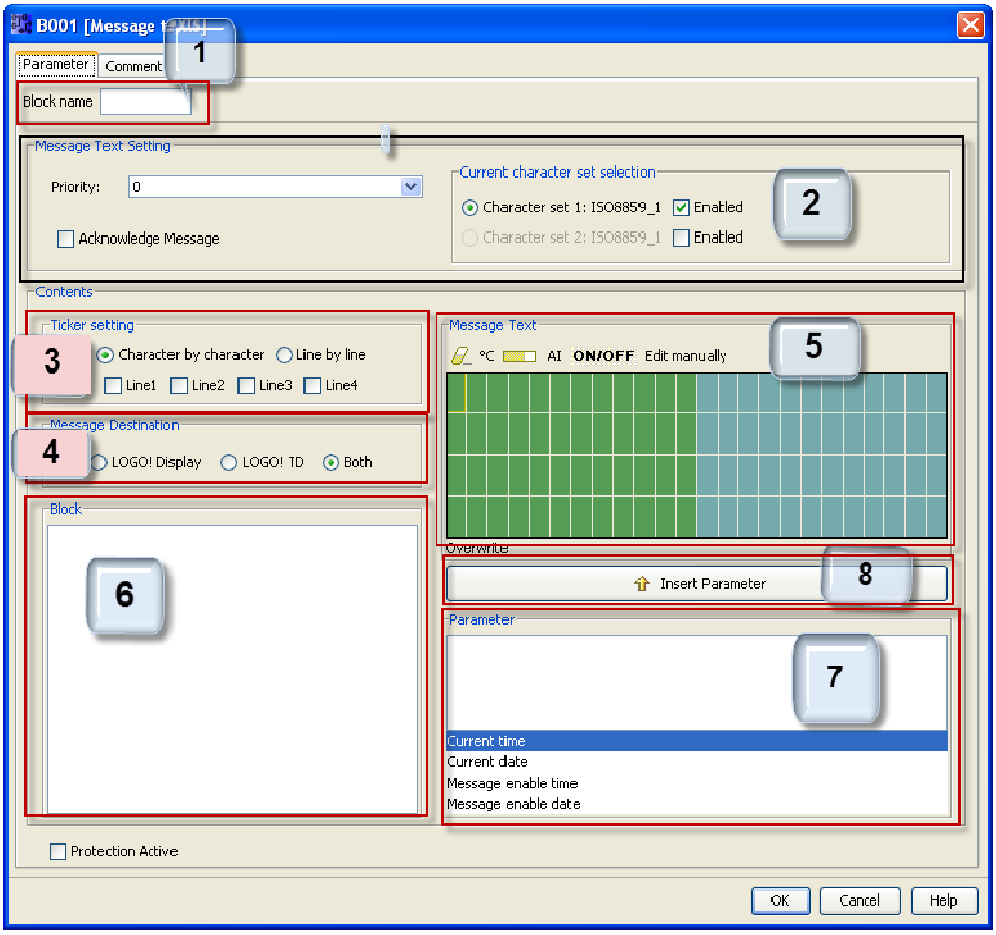
بدین منظور تابع Message texts را به همراه این تابع معرفی می کنیم.

متن های پیام-Message texts



این بلوک می تواند پیامهای متنی نوشته شده و پارامترهای تنظیم شده در مدت اجرا را، در صفحه نمایشی لوگو یا LOGO!TD نمایش دهد. طریقه عملکرد تابع به این صورت است که هرگاه وضعیت ورودی En از صفر به یک تغییر پیدا کند، متن پیام تنظیم شده در حالت اجرا نمایش داده می شود. خروجی این بلوک را می توانید با open connector (x) مسدود کنید و یا به ورودی توابع دیگر وصل کنید. در پنجره مشخصات این بلوک می توان لیست بلوک های استفاده شده در برنامه را در قسمت بلوک دید و آنها را به جدول Message text منتقل کرد. هرگاه چندین متن پیام داشته باشید، می توانید برای اولویت دادن نمایش پیام، شماره آنرا از جدول مشخصات در قسمت priority تعیین کنید. بطوریکه شماره بالاتر(۱۲۷) دارای بیشترین اولویت و شماره پایین تر(۰) دارای کمترین اولویت می باشد. پیام با بالاترین ارجحیت نمایش داده می شود. می توانید با استفاده از دکمه های ▲ و ▼ پیام های دیگر را در مد اجرا ببینید.

پنجره تنظیمات این بلوک به صورت زیر است. در پنجره تنظیمات در صورتیکه گزینه Acknowledgment Message فعال نباشد، متن پیام، زمانی که وضعیت سیگنال در ورودی En از یک به صفر تغییر می کند، پنهان می شود. ودر صورتیکه این گزینه فعال باشد، و وضعیت سیگنال ورودی En از یک به صفر تغییر یابد، در این حالت، متن پیام تا وقتی که توسط ok تایید شود نمایش داده می شود. زمانی که En=1 می باشد شما نمی توانید متن های پیام را تایید کنید.



شکل (۶-۹۸): پنجره تنظیمات تابع Message text

۱- **قسمت نام بلوک:** در این قسمت می توانید نامی برای بلوک بنویسید.

۲- **قسمت تنظیم**: در این قسمت می توانید تنظیمات زیر را انجام دهید:

- ارجحیت دادن به متن پیام
- جعبه چک کردن تایید پیام: اگر تیک زده شود در آنصورت برای بسته شدن یک پیام نیازمند تایید آن است.
- انتخاب مجموعه کاراکتر برای متن پیام

۳- **قسمت تیک:** می توانید تیک های مربوط به متن پیام را جهت نمایش دلخواه خود در این قسمت فعال کنید.

- تیک فرمت کاراکتر به کاراکتر
- تیک فرمت خط به خط
- تیک فعال سازی برای هر خط نمایشگر

۴- **قسمت مقصد پیام**: در این قسمت می توان مقصد پیام را تعیین کرد. به نمایشگر لوگو، LOGO! TD یا هر دو.

۵- **قسمت متن پیام-massage text**: در این قسمت می توان متن پیام را تنظیم کرد. اطلاعاتی که در این قسمت وارد می کنید، در نمایشگر نمایش داده خواهند شد. بالای این قسمت دکمه های اضافی زیر قرار دارند:

- **دکمه پاک کردن**: برای پاک کردن هر کاراکتری که در این قسمت وارد کرده اید.

- **دکمه علامتهای مخصوص**: برای قرار دادن علامتهای مخصوص در داخل قسمت پیام
- **دکمه بارگراف-Bar Graph**: برای قرار دادن بار گراف عمودی یا افقی در داخل قسمت متن پیام
- **دکمه AI:** برای جایگذاری یک مقدار ورودی آنالوگ در داخل ناحیه متن پیام
- **دکمه ON/OFF**: برای برچسب زدن به یک مقدار دیجیتال مطابق با وضعیت صفر ویک. برای مثال "OFF" و "ON" مطابق با ۰ و ۱.
- **دکمه Edit manually**: برای ویرایش یا حذف عناصر متن پیام بدون تغییر موقعیت عناصر دیگر مورد استفاده قرار می گیرد.

۶-قسمت بلوک-Block: در این قسمت می توانید بلوکهایی از بلوک های استفاده شده در مدار برنامه را جهت استفاده از پارامترهای آنها در ناحیه متن پیام انتخاب نمود.

۷- قسمت پارامترهای بلوک-Parameter: در این قسمت می توانید پارامترهای مربوط به بلوک انتخاب شده جهت نمایش در قسمت متن پیام را انتخاب کنید.

۸- دکمه Insert Parameter: این دکمه برای قرار دادن یک پارامتر بلوک انتخاب شده در داخل قسمت پیام استفاده می شود.

توضیحات:

Massage ticker

شما می توانید یک متن پیام را با تیک زدن یا نزدن در این قسمت پیکر بندی کنید. دو نوع تیک موجود می باشد:

- کاراکتر به کاراکتر-character by character
- خط به خط-Line by Line

هر خط متن پیام شامل ۲۴ کاراکتر می باشد، ولی در صفحه نمایشگر لوگو یا LOGO!TD امکان نمایش ۱۲ کاراکتر می باشد. برای آنکه این ۲۴ کاراکتر در در صفحه نمایش لوگو یا LOGO!TD قابل نمایش باشد، می توانید هر سطری را که می خواهید تمام کاراکتر هایش نمایش داده شود، با تیک زدن لاین مربوطه(Line1-Line4)، یکی از دو حالت نمایش کاراکتر به کاراکتر یا خط به خط را برای آن خط انتخاب کنید.

فرض کنید سطر اول مانند مثال زیر شامل ۲۴ کاراکتر می باشد. در حالت نمایش کاراکتر به کاراکتر ۱۲ کاراکتر سمت چپ ابتدا نمایش داده می شوند و بعد از گذشت مدت زمان ۰/۱ ثانیه اولین کاراکتر از سمت راست نمایش داده می شود و در واقع کاراکتر ها با فاصله زمانی ۰/۱ ثانیه به سمت چپ شیفت می یابند. البته ۰/۱ ثانیه مربوط به Time tick می باشد که در پنجره massage text setting قابل تنظیم است.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24

نمایش کارکترها در حالت کاراکتر به کاراکتر(ابتدا ۱۲ کاراکتر سمت چپ نمایش داده می شوند.)

X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 X21 X22 X23 X24

بعداز ۰/۱ ثانیه:

X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 X21 X22 X23 X24 X1

و این روند تا رسیدن به کاراکتر آخر ادامه می یابد.

برای همین مثال ذکر شده در حالت نمایش خط به خط ابتدا ۱۲ کاراکتر سمت چپ یک لاین نمایش داده مـی شـود و بعـد از گذشت مدت زمان ۱ ثانیه ۱۲ کاراکتر بعدی سمت راست نمایش داده می شوند.

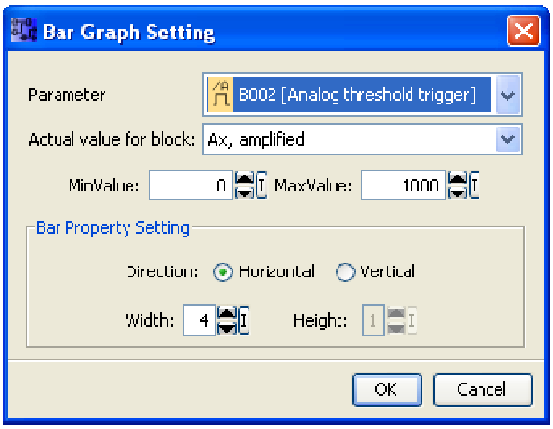
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 X21 X22 X23 X24

بعد از یک ثانیه:

X13 X14 X15 X16 X17 X18 X19 X20 X21 X22 X23 X24 X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12

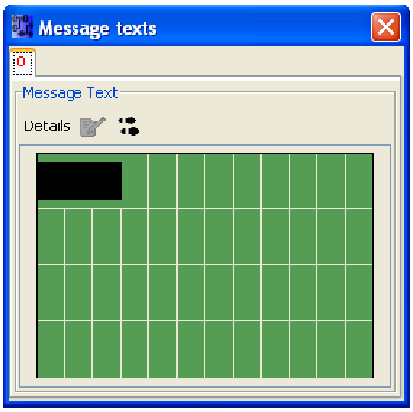
بار گراف– Bar Graph

بار گراف می تواند جهت نمایش یک سـیگنال آنـالوگ بـه صـورت بـار در دو حالـت افقـی یـا عمـودی روی نمایشـگر لوگـو یـا LOGO!TD مورد استفاده قرار گیرد. شما می توانید تا ۴ بار گراف در هر متن پیام تنظیم کنید. برای قرار دادن بار گراف، دکمه بار گراف را در قسمت متن پیام جهت قرار دادن بار گراف در این قسمت استفاده کنید. مقادیر حداقل و حداکثر بـارگراف را در پنجـره تنظیمات بارگراف Bar graph setting تعیین کنید. لوگو طول یا ارتفاع بار گراف را با درجه بندی کردن مقدار واقعی مـابین مقـادیر حداقل و حداکثر تعیین خواهد کرد. در این پنجره می توانید نوع نمایش بار گراف از لحاظ افقی یا عمـودی بـودن را تعیـین کنیـد. طول یا ارتفاع این بارگراف در این پنجره قابل تنظیم است.



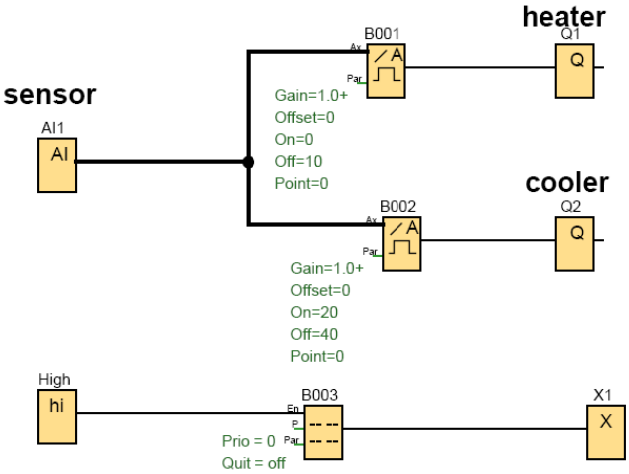
شکل (۶-۹۹): پنجره تنظیمات بارگراف

به عنوان مثال اگر طول بارگراف ۴ باشد و نمایش به صورت افقی باشد و مقادیر بین حداقل و حداکثر به ترتیب برابـر ۱۰۰۰ تـا ۲۰۰۰ باشند، با مقدار واقعی ۱۷۵۰، بار گراف به اندازه فاصله ۳ کاراکتر طول خواهد داشت.



شکل (۶-۱۰۰): نمایش بار گراف به اندازه ۳ کارکتر

مثال(۶-۴۱): برنامه کنترل دمای یک اتاق را بنویسید بطوریکه در بازه دمائی بین صفر تا ده درجه سانتیگراد هیتر روشن و در بازه دمائی بین ۲۰تا ۴۰ درجه کولر روشن شود. در بین ۱۰ تا ۲۰ درجه هیچ دستگاهی به نشانه ایده آل بودن دمای اتاق روشن نشود. می خواهیم در نمایشگر لوگو در سطر اول نام شرکت،در سطر دوم مقدار دمای اتاق، در سطر سوم دمای روشن شدن کولر و در سطر چهارم نیز تاریخ جاری نماش داده شود. پارامترهای دمائی فوق را به صورت پیام های متنی نمایش دهید.

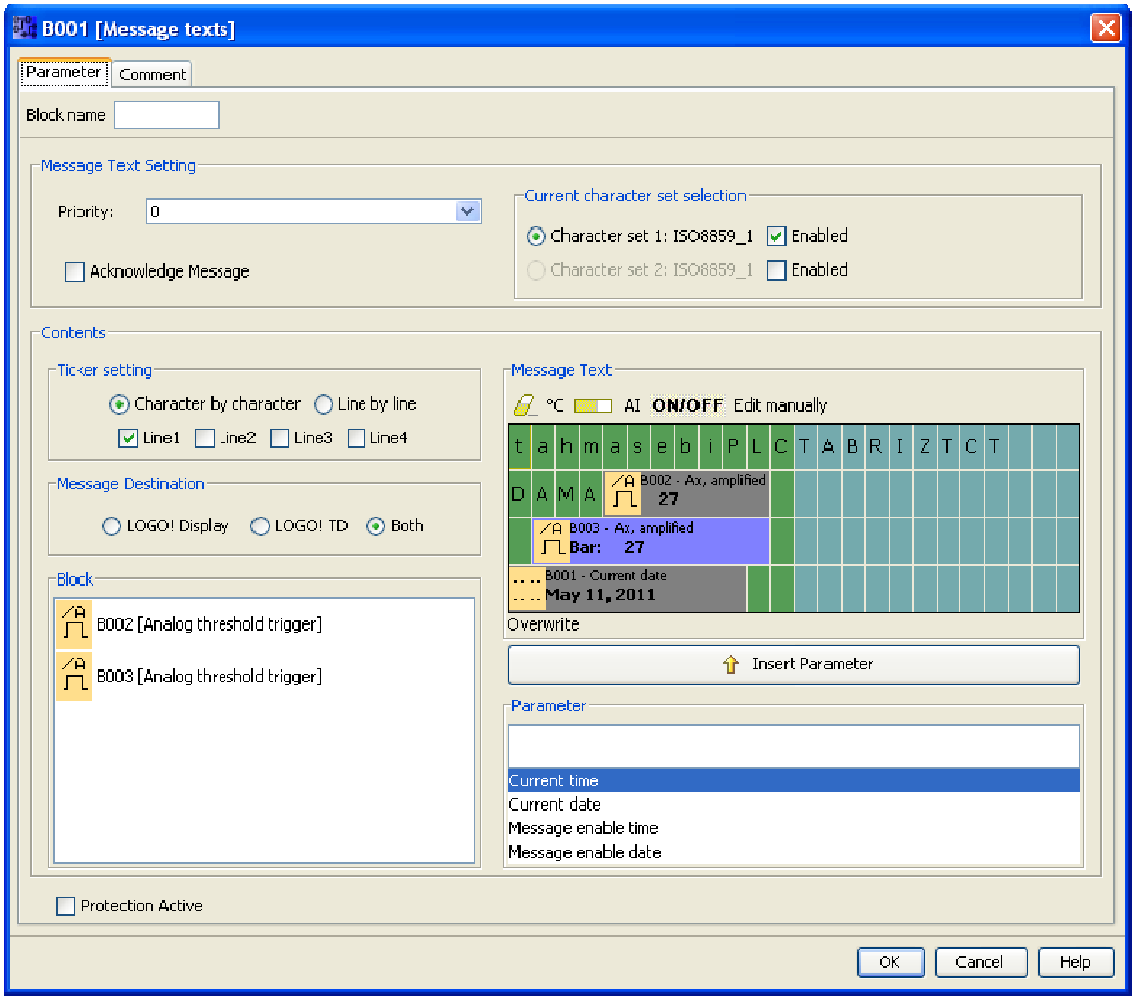


شکل (۶-۱۰۱): مدار مربوط به مثال (۶-۴۱)

در این مثال از یک ورودی آنالوگ برای سنسور دمائی و دو تابع آنالوگی Analog threshold trigger استفاده شده است. برای تابع اول مقادیر آستانه روشن و خاموش شدن به ترتیب برابر ۰ و ۱۰ و برای تابع دوم این مقادیر به ترتیب برابر ۲۰ و ۴۰ تنظیم شده اند. برای نشان دادن پیغام از تابع Message text استفاده کرده ایم. برای فعال کردن En از ورودیHigh به معنی ۱ و برای مسدود کردن خروجی از open connector استفاده کرده ایم. با دوبار کلیک کردن بر روی تابع Message text پنجره مشخصات این تابع ظاهر می شود که در سمت چپ لیست توابع استفاده شده در برنامه وجود دارد.

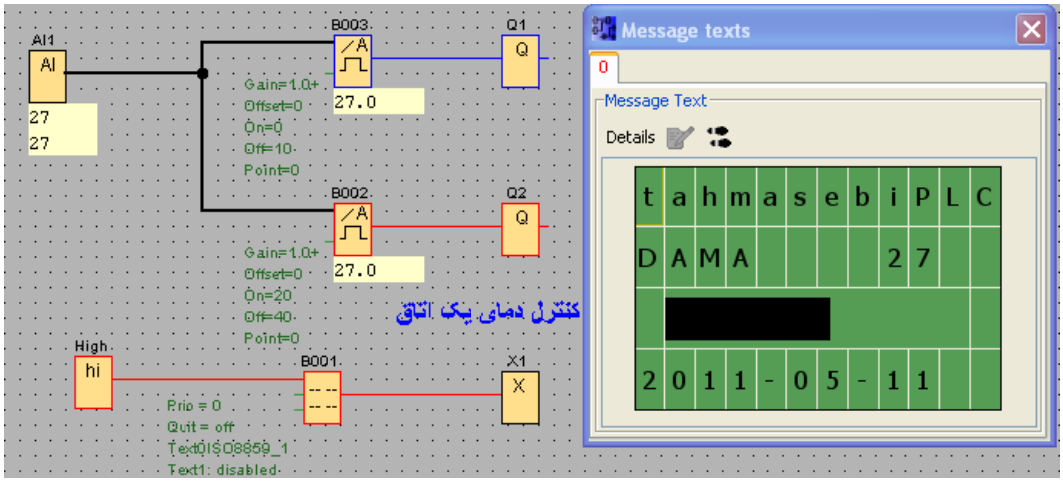
در سطر اول نام شرکت را که ما در اینجا بطور مثال Tahmasebi PLC Tabriz TCT انتخاب کرده ایم، تایپ می کنیم. دقت کنید که چون این نام بیش از ۱۲ کاراکتر می باشد بایستی حتما برای آنکه در یک سطر نمایش دهیم، از ویژگی جدید تابع متن پیام استفاده کنیم. برای استفاده از طرح جدید تابع متن پیام دکمه Enable new style… را در پنجره اولیه ای که باز می شود، فشار دهید. در پنجره باز شده Message text setting گزینه اول یعنی use new feature را تیک زده وok را بزنید. در این حالت سبک جدید تابع متن پیام به شکل (۶-۱۰۲) باز می شود. پس از انجام مراحل فوق متن ذکر شده را در سطر اول تایپ می کنیم. چون می

خواهیم از اطلاعات مربوط به تابع آنالوگی دوم که در سمت چپ پنجره متن پیام موجود است، استفاده کنیم (هر چند دراین مثال فرقی نمی کند از کدام استفاده کنیم) بایستی بر روی این تابع کلیک کنیم تا اطلاعات مربوط به این تابع در سمت راست در قسمت پارامتر نمایش داده شود. برای سطر دوم بعد از تایپ کردن کلمه DAMA بر روی Ax, amplified در قسمت parameter کلیک کرده و دکمه Insert parameter را می زنیم. برای آنکه دما بر حسب درجه سانتی گراد نمایش داده شود، علامت سانتیگراد را برای انتهای سطر دوم از قسمت Message text انتخاب می کنیم. برای سطر سوم بارگراف سیگنال را به صورت افقی تنظیم می کنیم. طول بار گراف را برای بازه ی ۰ تا ۵۰ برابر ۱۰ خانه انتخاب می کنیم. برای سطر چهارم پارمتر Current date را Insert می کنیم.



شکل (۶-۱۰۲): پنجره تنظیمات تابع متن پیام مربوط به مثال (۶-۴۱)

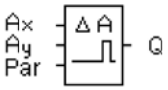
بعد از اتمام مراحل فوق برنامه را شبیه سازی می کنیم که در موقع شبیه سازی در نمایشگر پیام خبری از جزییات وجود ندارد.



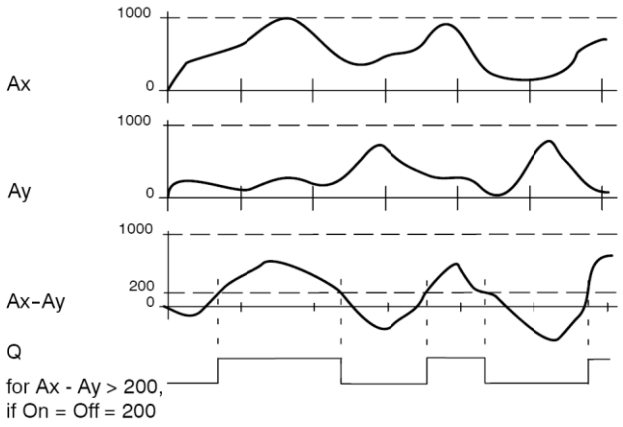
شکل (۶-۱۰۳): شبیه سازی مثال (۶-۴۱)

بر روی تابع Analog threshold trigger دوبارکلیک کنید تا پنجره مشخصات آن ظاهر شود. در قسمت Decimals in the massage text به طور مثال عدد ۲ و۳ تایپ کنید و نتیجه را در حین شبیه سازی مشاهده کنید. متوجه می شوید که در نمایشگر پیام متنی اعداد با دو رقم وسه رقم اعشار نمایش داده می شوند.

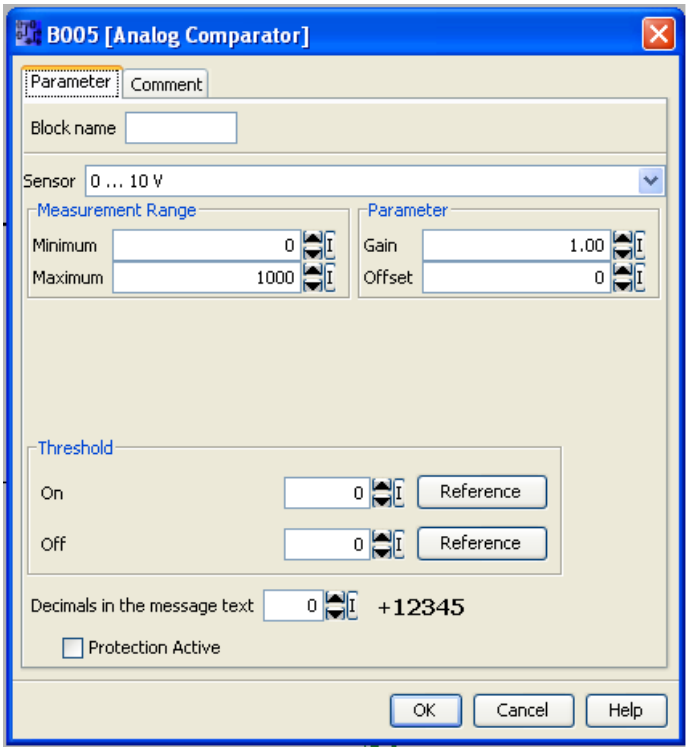
مقایسه کننده آنالوگ - Analog comparator



در این تابع، خروجی زمانی فعال است که تفاوت بین AX وAY از مقدار تعیین شده تجاوز کند.(AX – AY > ON)

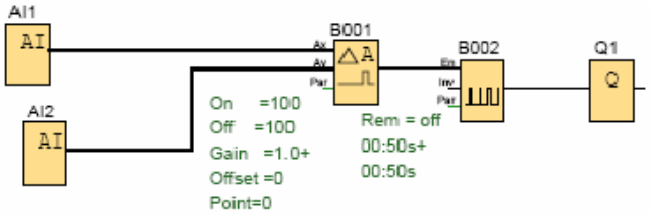


شکل (۶-۱۰۴): نمودار زمانی تابع مقایسه کننده آنالوگ



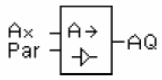
شکل (۶-۱۰۵): پنجره تنظیمات تابع مقایسه کننده آنالوگ

مثال(۶-۴۲): برای تولید یک آلیاژ مرغوب از دو کوره ذوب استفاده می کنیم. شرط تولید محصول با کیفیت آن است که دمای کوره دوم نباید بیش از ۱۰۰درجه با دمای کوره اول اختلاف داشته باشد. برنامه ای بنویسید که در صورت بروز چنین مسئله ای چراغ آلارم چشمک بزند.

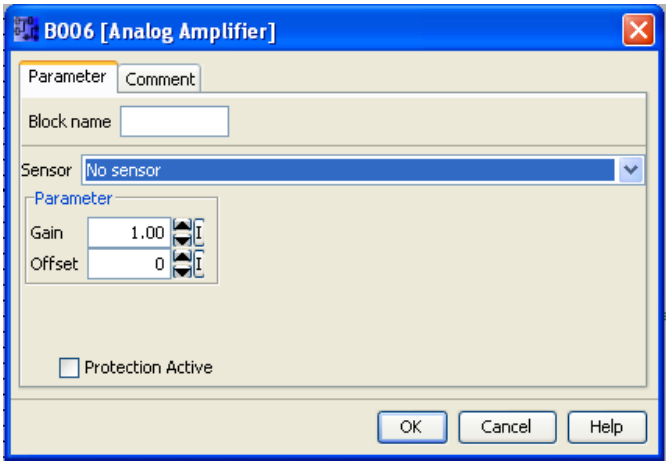


شکل (۶-۱۰۶): مدار مربوط به مثال (۶-۴۲)

تقویت کننده آنالوگ - Analog amplifier

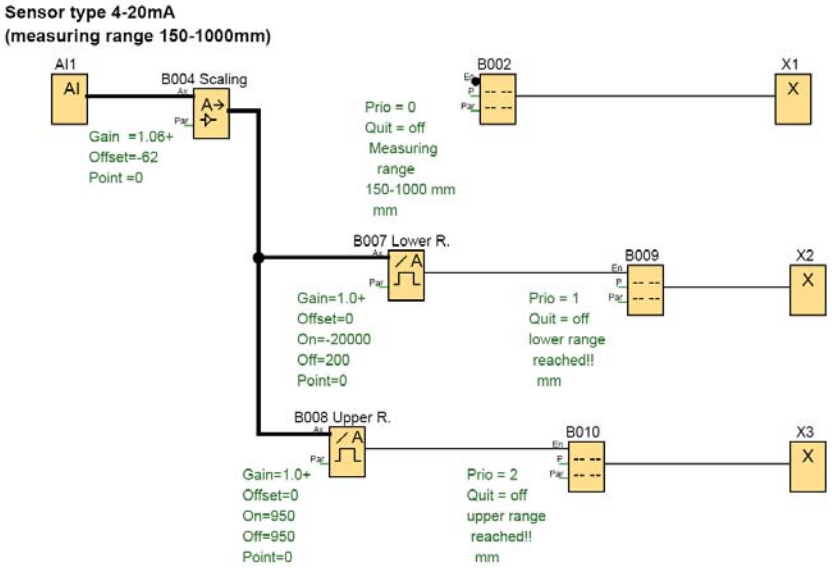


این بلوک یک مقدار ورودی آنالوگ را تقویت می کند و به یک خروجی آنالوگ باز گشت می دهد. خروجی آنالوگ تنها می تواند مقادیری از ۰ تا ۱۰۰۰ را پردازش کند. بنابراین برای آنکه دامنه خروجی تابع در محدوده ۰ تا ۱۰۰۰ باشد، باید از تقویت کننده استفاده کنیم. ورودی و خروجی این تابع آنالوگ می باشد.



شکل (۶-۱۰۷): پنجره تنظیمات تابع تقویت کننده آنالوگ

مثال(۶-۴۳): یک سنسور القائی فاصله ۱۵۰ تا ۹۹۸ میلی متر را اندازه گیری کرده و آن را به جریان ۴ تا ۲۰ میلی آمپر تبدیل می کند. برنامه ای بنویسید که در فاصله کمتر از ۲۰۰ میلی پیام lower range به معنای حداقل فاصله، در فاصله بین ۲۰۰ تا ۹۵۰ میلی پیام measuring range به معنای رنج اندازه گیری و در رنج بالاتر از ۹۵۰ پیام upper range به معنای بالاترین رنج نمایش داده شود.



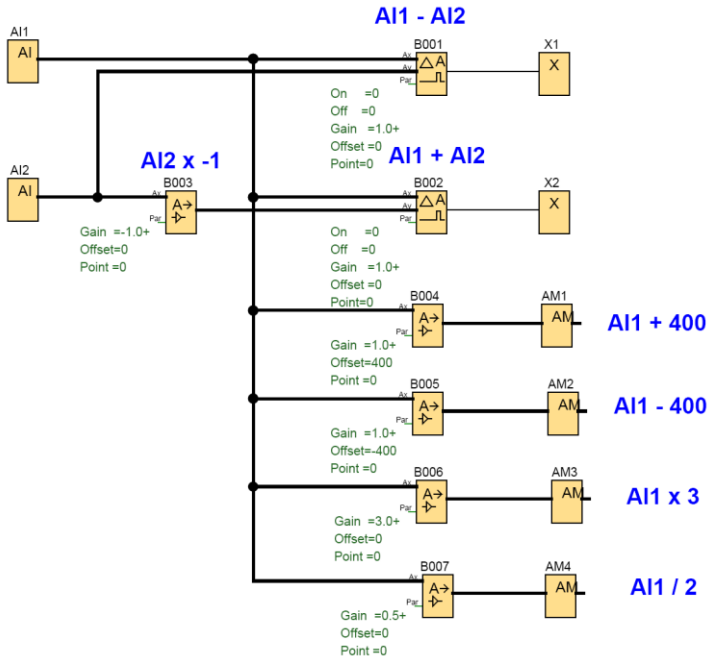
شکل (۶-۱۰۸): مدار مربوط به مثال (۶-۴۳)

در این برنامه بلوک های پیام متنی ارجحیت بندی شده اند. بطوریکه پیام measuring range با کمترین ارجحیت(priority) یعنی صفر، پیام lower range با ارجحیت یک و پیام upper range با ارجحیت ۲ تنظیم شده است. یعنی در صورتیکه به طور مثال دو تابع با ارجحیت های ۰و۱ همزمان فعال باشند، نمایشگر، پیام متنی با ارجحیت ۱ را نمایش خواهد داد.

مثال(۶-۴۴): محاسبات آنالوگی

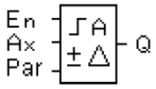
بدون استفاده از تابع Analog math اعمال زیر را برای دو سیگنال آنالوگ AI1 و AI2 انجام دهید.

AI1-AI2, AI1+AI2, AI1+400, AI1-400, AI1*3, AI1/2

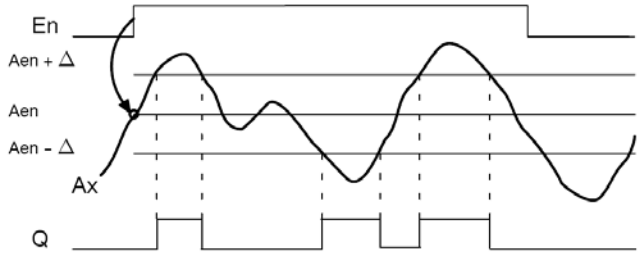


شکل (۶-۱۰۹): مدار مربوط به مثال (۶-۴۴)

سگ نگهبان آنالوگی- Analog value monitoring (Analog watchdog)

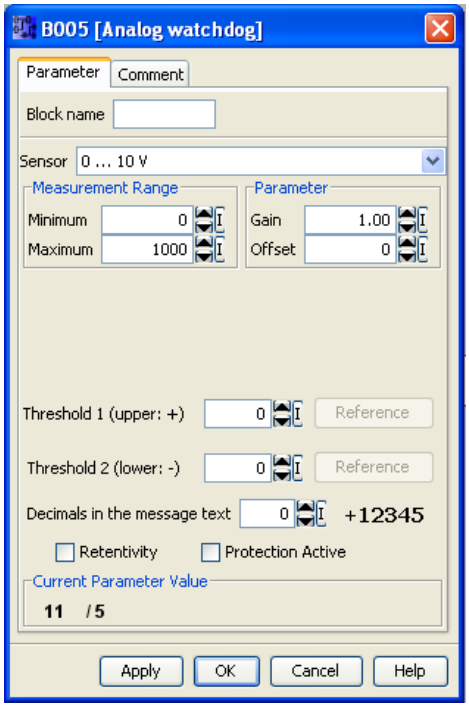


این تابع در واقع یک سگ نگهبان می باشد و زمانیکه سیگنال آنالوگ از محدوده تعیین شده خارج می شود، شروع به پارس کردن می نماید.



شکل (۶-۱۱۰): نمودار زمانی تابع سگ آنالوگی

در این تابع با فعال شدن En تابع آنالوگی یک نقطه را به عنوان مرز وسط Aen تعیین می نماید. بنابراین زمانی که سیگنال آنالوگ ازمحدوده Aen+Threshold1 و Aen-Threshold2 تجاوز می کند، خروجی روشن می شود. لازم به توضیح است که مقادیر آستانه مثبت و منفی را بایستی در پنجره مشخصات تابع تعریف کنیم.

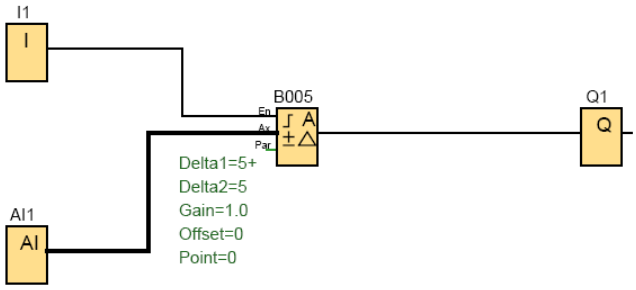


شکل (۶-۱۱۱): پنجره تنظیمات تابع سگ آنالوگی

مثال(۶-۴۵): ایجاد پایداری بعد ازتعیین نقطه مطلوب

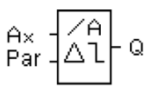
می خواهیم سطح یک مخزن در ۵۰ سانتی متر ثابت بماند. مقدار خطای ۵ سانتی متر قابل قبول می باشد. برنامه ای بنویسید که علیرغم تحقق شرایط فوق، امکان تعیین نقطه مطلوب به صورت اختیاری در دسترس باشد.

با توجه به صورت مسئله مقادیرThreshold1و Threshold2 را هر کدام را ۵ انتخاب می کنیم. زمانیکه مقدار سیگنال آنالوگ به ۵۰ رسید، En را جهت اسکن نقطه مطلوب فعال می کنیم. برنامه فوق به صورت زیر است.



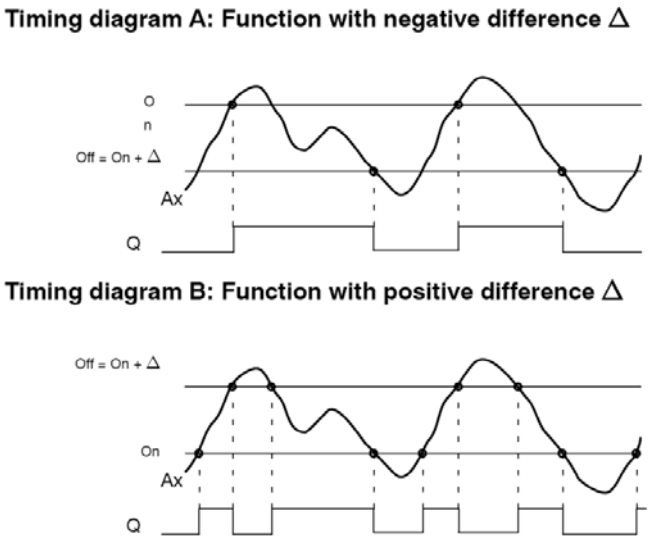
شکل (۶-۱۱۲): مدار مربوط به مثال (۶-۴۵)

راه انداز تفاضلی آنالوگ -Analog differential trigger



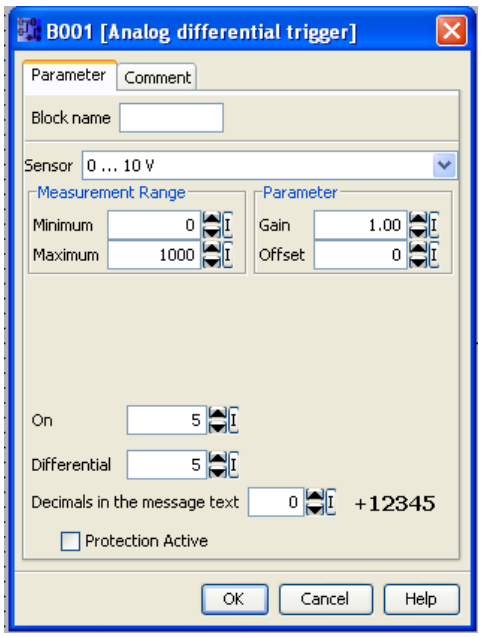
در این تابع خروجی متناسب با ارزش آستانه و یک مقدار تفاضلی فعال و غیر فعال می شود.

تابع بطور خودکار پارامترoff را بصورت Off = On + Δ محاسبه می کند که به موجب آن Δ ممکن است منفی یا مثبت باشد. دیاگرام زمانی این تابع با دلتای مثبت و دلتای منفی در شکل (۶-۱۱۳) آمده است.



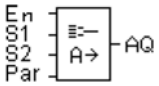
شکل (۶-۱۱۳): نمودار زمانی تابع راه انداز تفاضلی آنالوگ

بطور مثال اگر در پنجره مشخصات این تابع، Δ=5 وon=5 با شد، مقدارoff=10 خواهد شد. پنجره تنظیمات این تابع برای مثال ذکر شده به شکل (۶-۱۱۴) می باشد.



شکل (۶-۱۱۴): پنجره تنظیمات تابع راه انداز تفاضلی آنالوگ

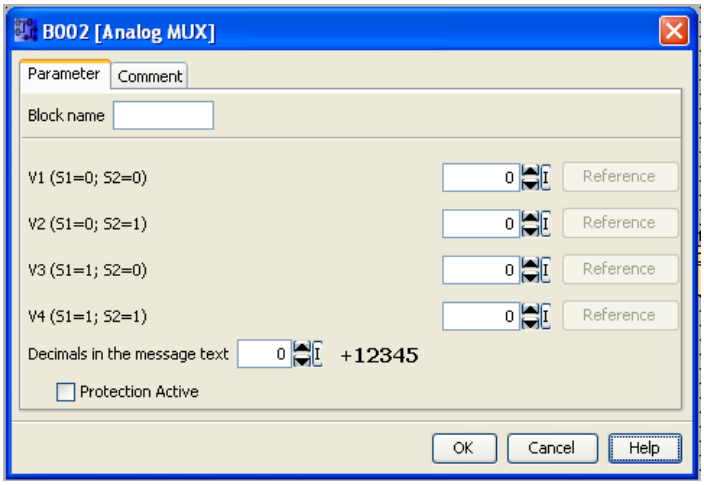
مالتی پلکسر آنالوگ–Analog MUX



این تابع دارای سه ورودی دیجیتال و یک خروجی آنالوگ می باشد. این بلوک در خروجی می تواند مقدار صفر یا یکی از مقدار آنالوگ ذخیره شده V1, V2, V3, V4 را که توسط پنجره مشخصات آماده سازی می شود به ما بدهد. زمانی که En=0 باشد خروجی برابر صفر خواهد شد. ودر صورتیکه En=1 باشد، با توجه به وضعیت S1 و S2 خروجی یکی از مقادیر V1، V2، V3 و V4 را خواهد داشت.

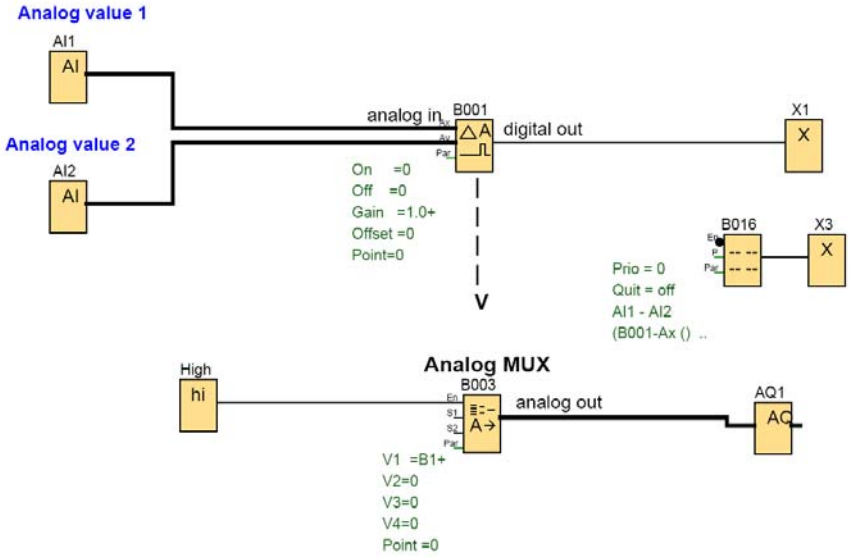
S1	S2	OUT
0	0	V1
0	1	V2
1	0	V3
1	1	V4

پنجره مشخصات این تابع به شکل زیر می باشد. البته می توانید برای مقادیر ذکر شده V1...V4 از توابع آنالوگی دیگر رفرنس دهید.



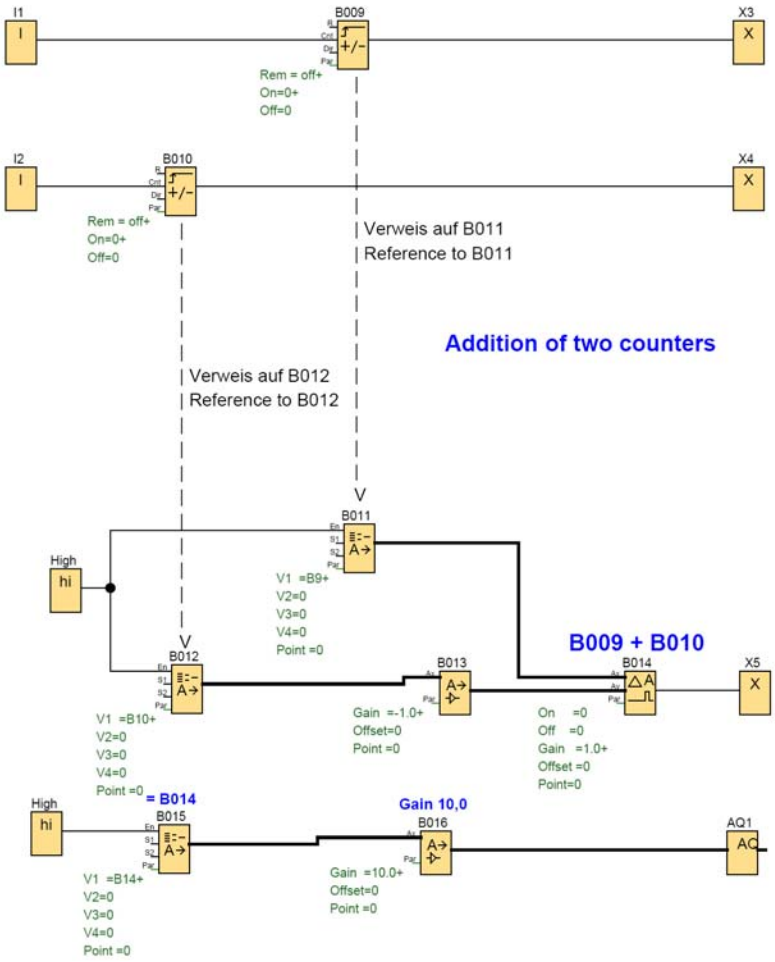
شکل (۶-۱۱۵): پنجره تنظیمات مالتی پلکسر آنالوگ

مثال(۶-۴۶): برنامه ای بنویسید که دو سیگنال آنالوگ را با هم مقایسه کرده و تفاضل آنها را به صورت آنالوگی نمایش دهد.



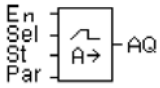
شکل (۶-۱۱۷): مدار مربوط به مثال (۶-۴۶)

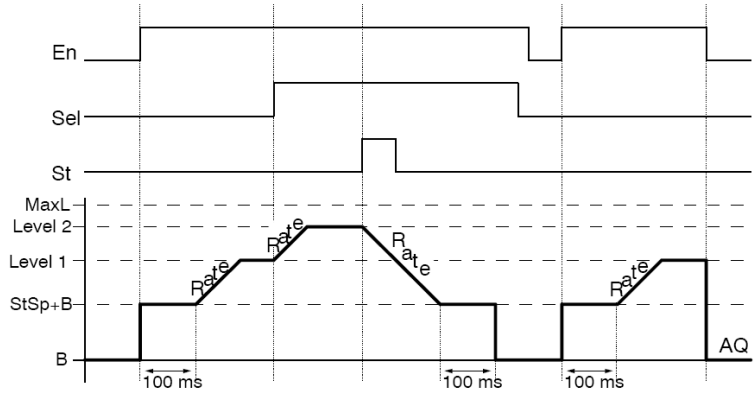
مثال(۶-۴۷): برنامه ای بنویسید که دو شمارنده دیجیتالی را با هم جمع کرده ونتیجه را در ۱۰ ضرب کند وآن را به صورت سیگنال آنالوگ به خروجی بدهد.



شکل (۶-۱۱۸): مدار مربوط به مثال (۶-۴۷)

کنترل شیب- Ramp control





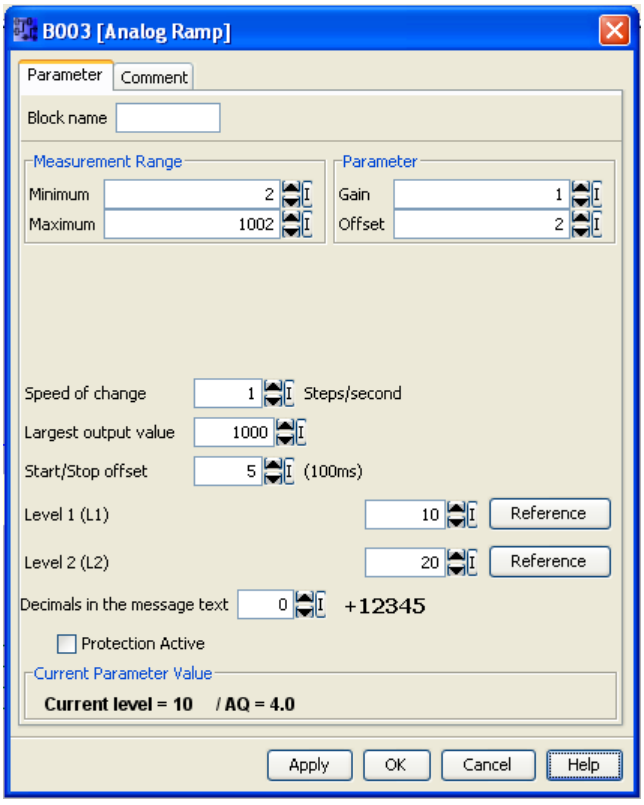
شکل (۶-۱۱۹): نمودار زمانی تابع کنترل شیب

این تابع دارای سه پایه ورودی دیجیتال و خروجی آنالوگ می باشد. با فعال شدن En، سطح ولتاژ خروجی از مقدار "StSp + Offset"B بصورت نرم و پیوسته به مقدار L1 می رسد و با فعال شدن پایه Sel سطح ولتاژ خروجی به صورت نرم و

پیوسته به L2 می رسد و با فعال کردن پایه St سطح ولتاژ بصورت معکوس از مقادیر L1 یا L2 به سمت صفر کاهش می یابد.
خروجی آنالوگ طبق فرمول زیر محاسبه می شود:

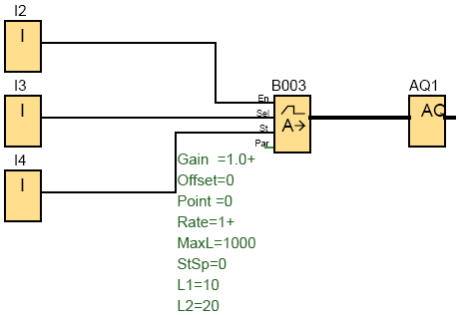
AQ= (Current Level – Offset"B")/ Gain"A"

سرعت این تغییرات با مقدارRate در پنجره مشخصات تابع تنظیم می شود.



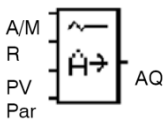
شکل (۶–۱۲۰): پنجره تنظیمات تابع کنترل شیب

مثال(۶–۴۸): مداری طراحی کنید که با زدن کلید اول نور لامپ بطور پیوسته به حالت نیمه روشن برسد و با زدن کلید دوم به نور کامل برسد و با زدن کلید سوم نور لامپ کم شده و سپس خاموش شود.



شکل (۶–۱۲۱): مدار مربوط به مثال (۶–۴۸)

کنترل کننده PI controller–PI



این تابع یک کنترل کننده انتگرالی- تناسبی می باشد، که دو پایه دیجیتالA/M و R و یک ورودی آنالوگ PV دارد، و خروجی آن نیز آنالوگ می باشد.

A/M=0 مد دستی و AM=1 مد اتوماتیک

R صفر کننده خروجی

PV (Process Value) مقدار پردازش

SP مقدار تنظیم شده

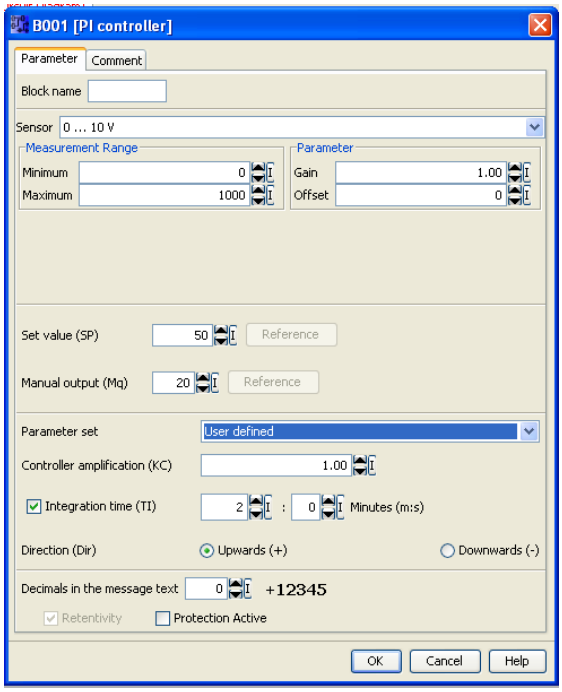
Mq مقدار خروجی در حالت مد دستی

Min حداقل مقدارPV

Max حداکثر مقدار PV

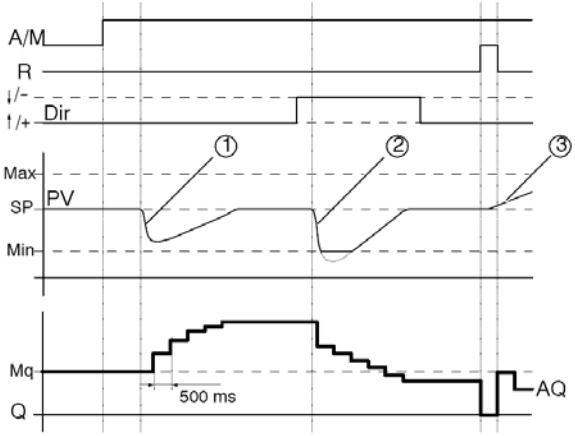
KC بهره (00.00 تا 99.99)، در صورتیکه صفر قرار داده شود، کنترل تناسبی اجرا نخواهد شد.

TI زمان انتگرال (00:01 تا 99:99)، در صورتیکه حداکثر مقدار قرار داده شود، کنترل انتگرالی اجرا نخواهد شد.



شکل (۶-۱۲۲): پنجره تنظیمات تابع کنترل کننده PI

رفتار و سرعت تغییرات AQ وابسته به پارامترهای KC و TI می باشد. دیاگرام نشان داده شده در شکل زیر یک مثال می باشد.



شکل (۶-۱۲۳): نمودار زمانی کنترل کننده PI

این تابع شامل دو وضعیت است:

الف) وضعیت دستی:

اگر ورودی A/M صفر باشد در اینصورت مقدار خروجی AQ با مقدار پارمتر Mq برابر می باشد.

ب) وضعیت اتوماتیک:

اگر ورودی A/M فعال شود در اینصورت تابع کنترلر از فرمول زیر مطابقت کرده و مقدار جدید PV محاسبه می شود.

PV(*new*)=(PV x gain) + offset

اگر مقدار جدید PV با SP برابر باشد، تابع کنترلر مقدار خروجی AQ را تغییر نمی دهد.

اگر مقدار جدید PV بیشتر از مقدار SP باشد، تابع کنترلر مقدار خروجی AQ را افزایش می دهد.

اگر مقدار جدید PV کمتر از مقدار SP باشد، تابع کنترلر مقدار خروجی AQ را کاهش می دهد.

با یک اغتشاش AQ به کاهش و یا افزایش ادامه می دهد تا اینکه مقدار جدید PV خود را دوباره با مقدار SP مطابقت دهد. سرعت تغییرات خروجی بستگی به پارامترهای KC و TI دارد.

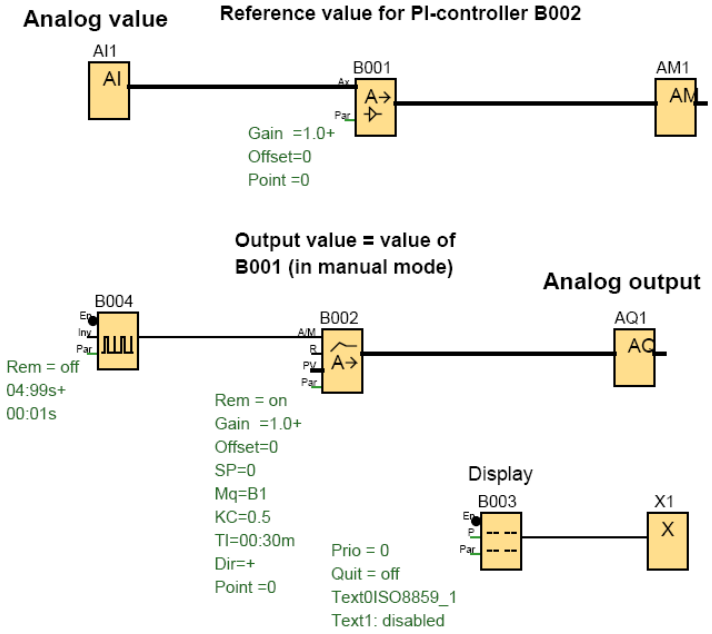
اگر ورودی PV از مقدار Max تجاوز نماید، در این صورت مقدار جدید PV به مقدار ماکزیمم Set می رسد.

اگر ورودی PV از مقدار min کمتر باشد، در اینصورت مقدار جدید PV به مقدار مینیمم Set می رسد.

برای غیرفعال نمودن مقادیر AQ و A/M می توان پایه Rرا فعال کرد.

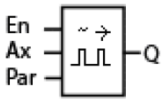
مثال(۶-۴۹): فیلتر آنالوگی

در یک فرآیند دمائی سریع سیگنال آنالوگ دارای تغییراتی است. می خواهیم این تغییرات از هر ۵ ثانیه یک بار در خروجی آنالوگی ظاهر شود و در طول این۵ ثانیه آخرین مقدار تغییریافته در ۵ ثانیه قبلی، همچنان در خروجی ثبت شود. برنامه فوق را با استفاده از کنترل کننده PI بنویسید.

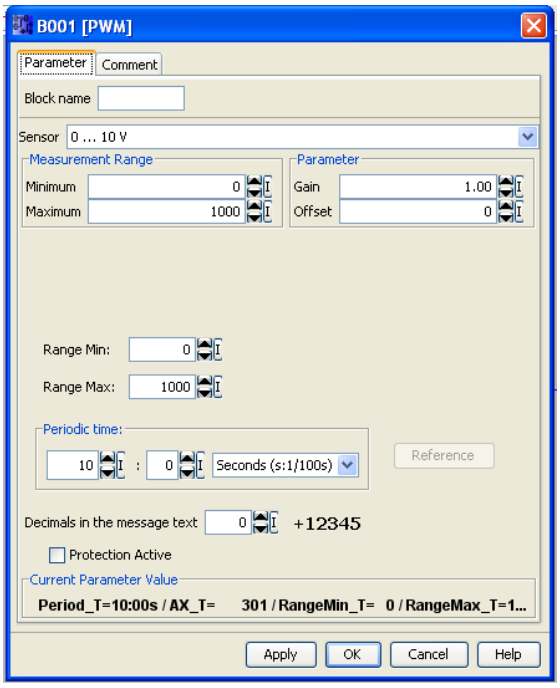


شکل (۶-۱۲۴): مدار مربوط به مثال (۶-۴۹)

مدولاسیون پهنای پالس-Pulse Width Modulator (PWM)

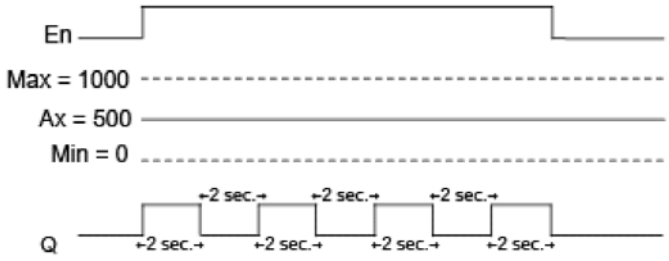


این تابع ورودی آنالوگ خود را با یک سیگنال مربعی در خروجی مدوله می کند. اندازه پهنای پالس متناسب با سیگنال آنـالوگ Ax می باشد.



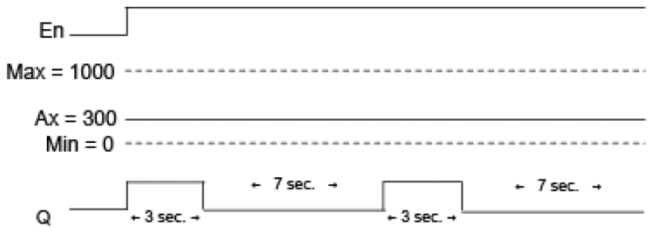
شکل (۶-۱۲۵): پنجره تنظیمات تابع مدولاسیون پهنای پالس

مثـال اول: اگر سیگنال آنالوگ برابر ۵۰۰ باشد (در رنج بین صفر تا ۱۰۰۰)و PT(periodic time) زمان تکـرار برابـر ۴ باشـد، موج مربعی حاصل دارای عرض پالس ۲ ثانیه و وقفه پالس ۲ ثانیه خواهد بود، البته به شرطی که ورودی En فعال باشد.



شکل (۶-۱۲۶): نمودار زمانی مدولاسیون پهنای پالس مثال اول

مثـال دوم: اگر سیگنال آنالوگ برابر ۳۰۰ باشد (در رنج بین صفر تا ۱۰۰۰) و PT(periodic time) زمان تکرار برابر ۱۰ باشـد، موج مربعی حاصل دارای عرض پالس ۳ ثانیه و وقفه پالس ۷ ثانیه خواهد بود، البته به شرطی که ورودی En فعال باشد.



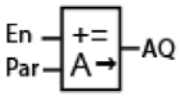
شکل (۶-۱۲۷): نمودار زمانی مدولاسیون پهنای پالس مثال دوم

قوانین محاسبه مدت زمان عرض پالس و وقفه پالس خروجی

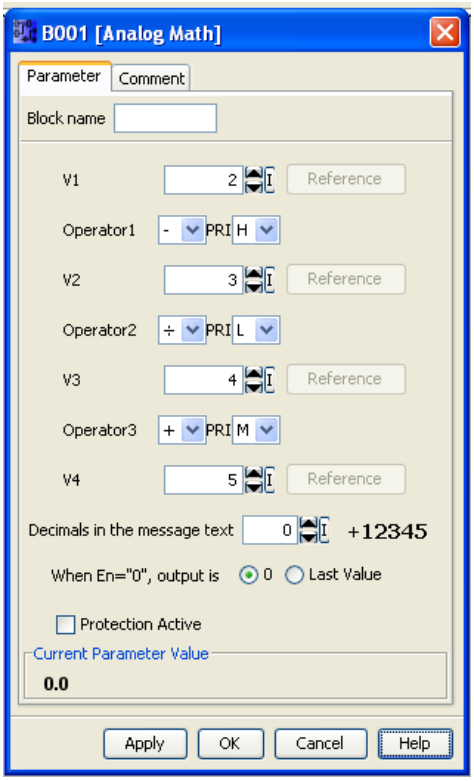
Q = 1, for (Ax – Min)/ (Max – Min) of time period T, زمانیکه Min < Ax < Max

Q = 0, for PT – [(Ax – Min) / (Max – Min)] of time period T.

تابع محاسبات آنالوگی– Analog math



این تابع یک محاسبه گر آنالوگی برای چهار سیگنال آنالوگ می باشد. محاسبه زمانی صورت می گیرد، که ورودی دیجیتالی En فعال باشد. خروجی این تابع به صورت آنالوگ می باشد.



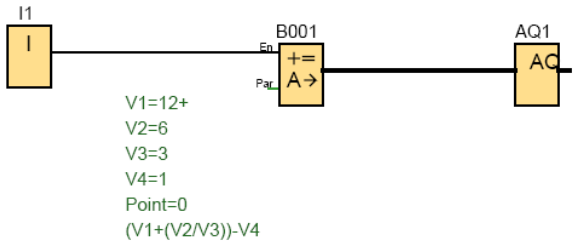
شکل (۶-۱۲۸): پنجره تنظیمات تابع محاسبات آنالوگی

چهار مقدار آنالوگی با نام های V1…V4 مشخص هستند، که می توانند از توابع آنالوگی دیگر نیز رفرنس بگیرند. عملگر های ریاضی جمع، تفریق، ضرب و تقسیم با نام های Operator در پنجره تنظیمات قابل مشاهده هستند. می توانید از نوار PRI برای عملگرها با انتخاب به ترتیب H(High)، M(Medium) و L (Low) الویت انجام تعیین کنید.

مثال(۶-۵۰): عمل ریاضی زیر را انجام دهید.

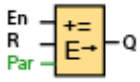
Equation: (12 + (6 / 3)) - 1 = 13

V1	Op1 (Pr1)	V2	Op2 (Pr2)	V3	Op3 (Pr3)	V4
12	+ (M)	6	/ (H)	3	- (L)	1

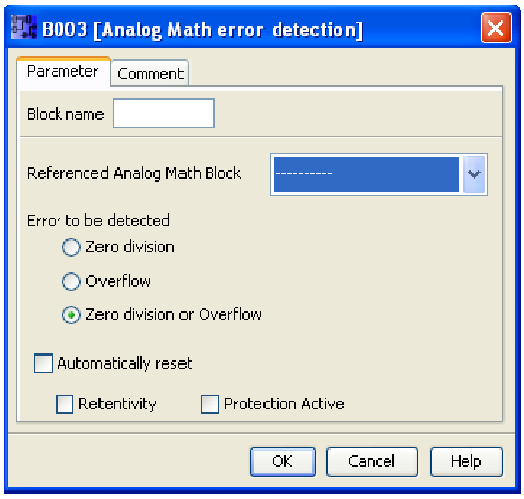


شکل (۶-۱۲۹): مدار مربوط به مثال (۶-۵۰)

آشکار سازی خطای محاسبات آنالوگی- Analog math error detection



این بلوک جهت آشکار سازی خطای اتفاق افتاده در بلوک محاسبه گر آنالوگی بکار می رود. این بلوک زمانی فعال می شود که پایه En آن یک باشد. خروجی این تابع زمانی فعال می شود که یکی از خطاهای تقسیم بر صفر (zero division) یا سرریز(overflow) و یا هر دوی آنها در بلوک محاسبه گر آنالوگی اتفاق بیافتد. در پنجره تنظیمات این تابع می توانید نوع خطا را که خروجی بایستی در آن روشن شود، تعیین نمایید. در صورتیکه گزینه Automatically reset فعال باشد، در آنصورت خروجی پیش از اجرای بعدی تابع محاسبه گر آنالوگی خاموش می شود، یعنی زمانی که خطا رفع شد خروجی غیر فعال می شود. و اگر این گزینه فعال نباشد، خروجی تا وقتی که تابع محاسبه گر آنالوگی ریست نشود، همچنان روشن می ماند.



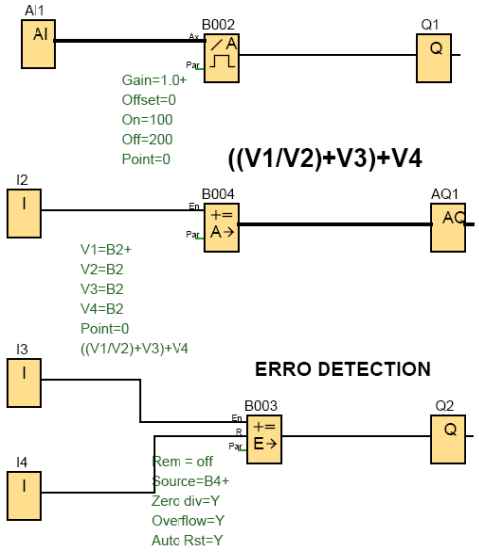
شکل (۶-۱۳۰): پنجره تنظیمات آشکار سازی خطای محاسبات آنالوگی

مثال(۶-۵۱): آشکار سازی خطای تقسیم بر صفر

مثال زیر را در محیط نرم افزار شبیه سازی کنید. می خواهیم محاسبه زیر را انجام دهیم:

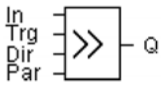
((V1/V2) +V3)+V4

رفرنس تمامی مقادیر آنالوگی را در تابع محاسبه گر آنالوگی از تابع Analog threshold trigger قرار دهید. یعنی در هر لحظه V3،V2،V1 و V4 دارای مقدار یکسان خواهند بود. مقدار AI1 را به آرامی از صفر شروع به افزایش دهید. در لحظه ای که سیگنال آنالوگ مقدار صفر دارد، خروجی Q2 به نشانه بروز خطای تقسیم بر صفر روشن خواهد شد. چنانچه در این حالت گزینه Automatically reset را فعال کرده باشید، به محض افزایش سیگنال از مقدار صفر خروجی Q2 خاموش خواهد شد، و اگر این گزینه را فعال نکنید بایستی برای خاموش کردن خروجی از پایه ریست (I4) تابع آشکار ساز خطا استفاده کنید.



شکل (۶-۱۳۱): مدار مربوط به مثال (۶-۵۱)

شیفت رجیستر- Shift register



تابع شیفت رجیستر می تواند برای خواندن مقدار یک ورودی و انتقال اطلاعات از چپ به راست و برعکس استفاده شود. مقدار خروجی با بیت ایجاد شده شیفت رجیستر مطابقت دارد. تابع شیفت رجیستر فقط برای یکبار در داخل مدار برنامه استفاده می شود.

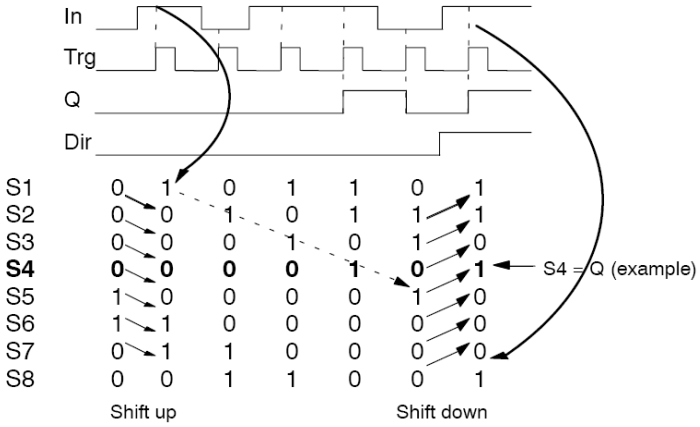
ورودیIN: مقدار دهی به اولین بیت یا آخرین بیت از هشت بیت توسط این پایه صورت می گیرد.

ورودیTrg: تابع با لبه بالا رونده Trg تحریک می شود.

ورودی Dir: انتقال از چپ به راست (در صورت صفر بودن) و انتقال از راست به چپ (در صورت یک بودن) را مشخص می کند.

تابع شیفت رجیستر به همراه بیت های شیفت رجیستر (shift register bit) که نهایتاً ۸ عدد می باشد بکار برده می شود. در پنجره مشخصات این تابع، بیت شیفت رجیستر تعیین کننده مقدار خروجی Q است. به عنوان مثال اگر ۴ را برای آن انتخاب کنیم، خروجی در بیت چهارم روشن می شود.

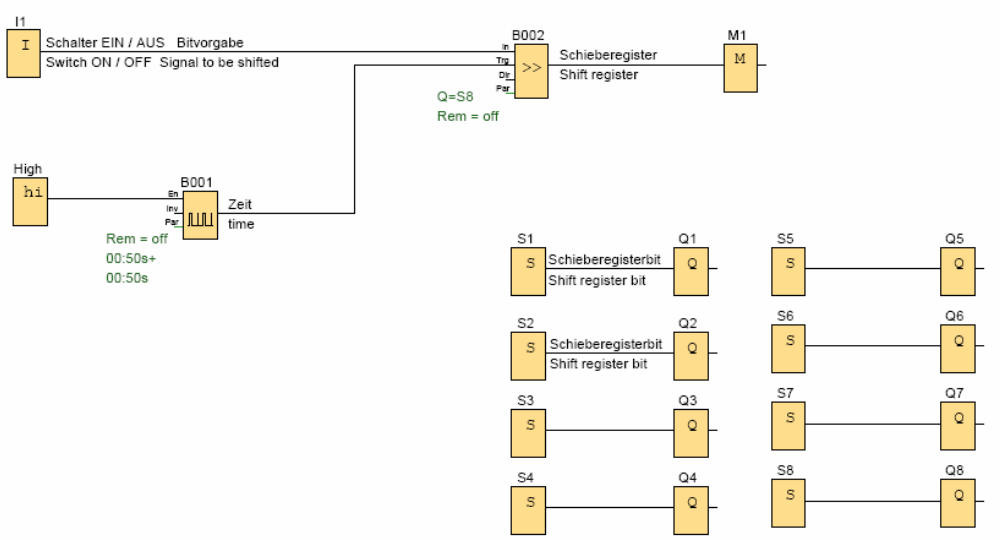
شکل (۶-۱۳۲): پنجره تنظیمات تابع شیفت رجیستر



شکل (۶-۱۳۳): نمودار زمانی تابع شیفت رجیستر

برای اینکه بطور کامل متوجه مسئله شوید مثال (۶-۵۲) را اجرا کنید.

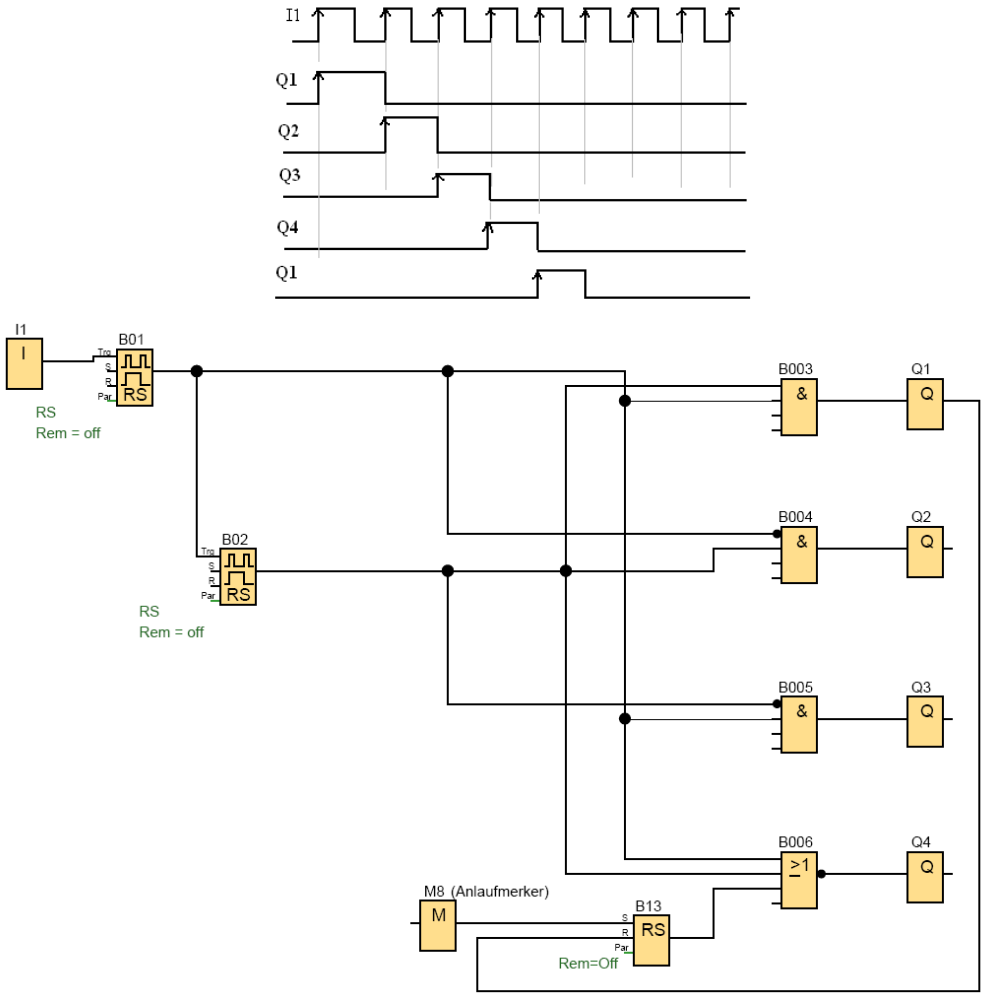
مثال(۶-۵۲): برنامه زیر را در لوگو اجرا کنید.



شکل (۶-۱۳۴): مدار مربوط به مثال (۶-۵۲)

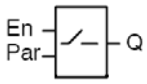
مثال (۶-۵۳): شمارنده جانسون

بدون استفاده از شیفت رجیستر مداری طراحی کنید که بتوان ۴ خروجی را به ترتیب از Q1 تا Q4 را روشن و خاموش کرد، بطوریکه در هر لحظه فقط یک خروجی روشن و بقیه خاموش باشند.

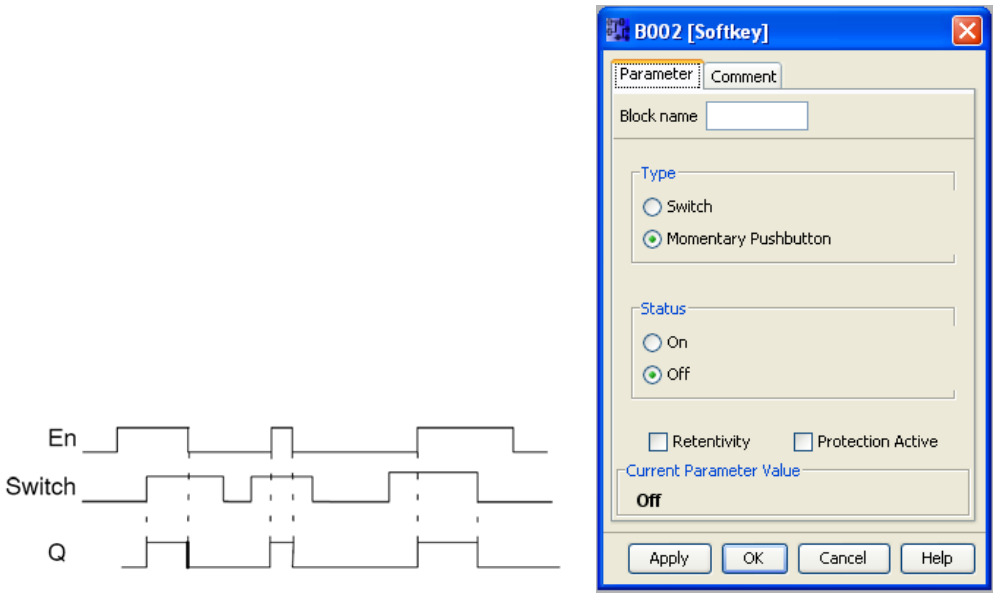


شکل (۶-۱۳۵): مدار مربوط به مثال (۶-۵۳)

کلید-Softkey



این تابع دارای یک ورودی En و یک خروجی می باشد. پنجره مشخصات این تابع به شکل زیر است.



شکل (۶-۱۳۶): پنجره تنظیمات و نمودار زمانی تابع Softkey

در صورتیکه نوع تابع switch و حالت آن بر روی on تنظیم شده باشد، تا زمانیکه ورودی En فعال باشد، خروجی روشن خواهد شد.

در صورتیکه نوع تابع pushbutton بوده و ورودی آن فعال باشد و حالت آنرا برروی on تنظیم کنیم، خروجی فقط به اندازه یک cycle time روشن خواهد بود.