

DCS Experts

توسعه دانش سیستم های کنترل صنعتی



SIEMENS

آموزش طراحی و پیاده سازی سیستم S7400FH

SIMATIC Fail safe & fault Tolerrant

**PROFI
NET**

Microsoft
Windows



industrial ethernet

**PROFI
BUS**



poweren@yahoo.com

کارشناس ارشد برق

کارشناس و مدرس دوره های زیرمنس - 2734

۱	فصل ۱ - مقدمه
۱	۱-۱- دلایل بکارگیری سیستم های H-PLC (Motivation)
۱	۱-1-1- حوزه های کاربردی اتوماسیون S7-400H
۲	۱-2- ایستگاه اتوماسیون (AS)
۴	۱-2-2- سامانه اتوماسیون استاندارد (سری AS400)
۴	۲-2-2- سامانه اتوماسیون برای دسترسی پذیری بالا (AS400H)
۶	۳-2-2- سامانه اتوماسیون ایمن در خطا (Fail-Safe AS400FH)
۷	فصل ۲ - ۲ - سامانه AS-400 استاندارد
۷	۲-1- اجزای AS-400 استاندارد
۸	۲-۲- انواع رک S7-400
۹	۲-3- انواع باس در رک S7-400
۱۱	فصل ۳ - ۳ - ساختار و معماری سامانه S7-400H
۱۲	۱-۳- مقدمه
۱۲	۲-۳- معماری سیستم (System architecture)
۱۳	۳-۳- اجزای سخت افزاری یک سیستم S7-400H (Minimum Configuration)
۱۴	۱-3-3- ماژول کنترل کننده (CPU)
۱۷	۲-3-3- ماژول های سنکرون سازی (SYNC Modules)
۱۸	۳-3-3- کابل فیبر نوری
۱۸	۴-۳- مشخصه های افزونگی (Redundancy features)
۱۸	۱-4-3- تعویض بی درنگ بین دو کنترل کننده (Bumpless Master-Stand-by switchover)
۱۸	۲-4-3- سنکرون سازی بین دو CPU (Synchronization Procedure)
۲۱	۳-4-3- توابع خودآزمایی (self-test)
۲۲	۴-4-3- برنامه نویسی آنلاین (Online programming)
۲۲	۵-4-3- اتصال به PG
۲۳	۶-4-3- قابلیت تعمیر آنلاین (Online Repair)
۲۴	۳-4-7- قابلیت CiR (Configuration Changes in Run Mode)
۲۵	۸-4-3- پیکربندی کنترل کننده (Configuration)

- ۲۷..... (System integration) 400H پیاده سازی سیستم 3-5-3
- ۲۷ برنامه نویسی 1-5-3
- ۲۸ (Hardware configuration) سخت افزاری 3-5-2
- ۲۸..... (Redundancy principle) افزونگی ۶-۳
- ۲۹ (Hardware Redundancy) سخت افزاری 1-6-3
- ۳۰ (homogeneous redundancy) افزونگی با اجزای یکسان 2-6-3
- ۳۰ (Passive redundancy) افزونگی پسینو 3-6-3
- ۳۰ (Active redundancy) افزونگی فعال 4-6-3
- ۳۰..... (I/O configuration) H در سیستم I/O پیکر بندی و اتصال ۷-۳
- ۳۲ اصول I/O افزونه 1-7-3
- ۳۲ (FMR-Flexible Modular Redundancy) افزونگی با I/O های مبتنی بر شبکه پروفی باس 2-7-3
- ۳۳ (Single Mode) افزونه ساختار افزونه 3-7-3
- ۳۳ (H-CPU in single mode) استفاده از ماژول های I/O افزونه با یک سیستم H بدون افزونه 3-7-4
- ۳۳ (Distributed Switched I/O) CPU مشترک بین دو I/O استفاده از ماژول های I/O افزونه در رک مرکزی (Central I/O Modules) دو سیستم H افزونه 6-7-3
- ۳۴
- ۳۵ (Distributed I/O modules) افزونه توزیع شده 3-7-7
- ۳۵ ساختار افزونه کامل برای I/O برای دسترس پذیری خیلی بالا 3-7-8
- ۳۸..... (Wiring) سیستم بندی کانال های افزونه ۸-۳
- ۳۸ (Wiring) سیستم بندی کانال های ورودی دیجیتال افزونه 1-8-3
- ۳۹ سیستم بندی کانال های خروجی دیجیتال افزونه 2-8-3
- ۴۰ سیستم بندی کانال های ورودی آنالوگ افزونه 3-8-3
- ۴۰ سیستم بندی کانال های خروجی های آنالوگ افزونه 3-8-4
- ۴۱ (Redundant quality stages) I/O سطوح کیفیت افزونگی 5-8-3
- ۴۱..... یکپارچگی با برنامه کاربر ۹-۳
- ۴۱ (Redundant quality stages) I/O پیکر بندی سخت افزاری برای I/O افزونه 3-9-1
- ۴۲ software Redundancy در station دو ET200M (IM153-3) کردن یک Redundant نحوه 3-9-2
- Error! Bookmark not HWCConfig** در 3-10 شماره سفارش ماژول، رک و باس یونیتها در **defined.**

۴۳ (Block library Functional I/O Redundancy) افزونه I/O بلوک های

۴۳ اتصال PROFIBUS PA از طریق یک لینک PA به سیستم H

۴۴ ماژول Y-Link

۴۵ واسط یا گذرگاه ارتباطی I/O

۴۶ (Redundant communication Principle) ارتباطات افزونه

۴۸ (Fault-tolerant S7 Connections) ارتباطات از طریق اتصالات تحمل پذیر خطا

۴۹ (Reliability of modules) قابلیت اطمینان ماژول های زیرمنس

۴۹ نصب و راه اندازی یک سیستم S7400H

۵۱ مانیتورینگ سگنال های سیستم H

۵۴ توصیف فانکشن بلاک ها

۵۴ فانکشن بلاک RedAn02

۵۵ فصل ۴ - ساختار و معماری سیستم های FH

۵۵ 4-1- مقدمه

۵۵ 1-1-4 تاریخچه سیستم های ایمنی زیرمنس (History of Siemens Safety Systems)

۵۶ 2-1-4 تفاوت بین سیستم SIS و سیستم کنترل پایه BPCS

۵۶ 3-1-4 هدف از سیستم های SIS

۵۷ 4-1-4 ریسک چیست و چه کسی تصمیم می گیرد که چه ریسکی قابل قبول است؟

۵۸ 5-1-4 استانداردهای مهم مرتبط با ایمنی

۵۸ ۲-۴ سخت افزار سازگار با سطوح SIL 1 , SIL2, SIL3

۵۹ 1-2-4 انتخاب یک CPU برای یک پلنت فرآیندی

۵۹ 4-3- سیستم های اتوماسیون خرابی امن FH (Fail Safe Automation System)

۶۰ 1-3-4 اصول پشت پرده عملکرد ایمنی (Principle behind the Safety Function)

۶۰ 2-3-4 پیکربندی های پایه برای تامین ایمنی و در دسترس پذیری (Basic Configuration Variants)

۶۲ ۴-۴ معماری های سیستم Fail Safe

۶۳ 1-4-4 افزودنی مورد نیاز

۶۳ 4-4-2 معماری سیستم تک سنسور یا ارزیابی 1001 (1 Out of 1)

۶۴ 3-4-4 ارزیابی 1002 (1 Out of 2)

۶۶ 4-4-4 معماری سیستم 2 Out of 2 (2002)

۶۶ 1-4-4 معماری سیستم (2003-2 Out of 3)

۶۸ 2-4-4 لاجیک ارزیابی 2003 بدون در نظر گرفتن خطای کانال

۶۹ 3-4-4 پیکربندی 2003 با در نظر گرفتن خطای کانال (2003D)

۶۹ 4-4-4 معماری استفاده از سه سنسور با دو ماژول افزونه (Three sensors with redundant F-AI)

۷۰ 5-4-4 معماری 1001D (with diagnostics – Non Redundant Safety System)

۷۱ 6-4-4 معماری 1002D (with diagnostics)

۷۱ 7-4-4 معماری طراحی سیستم S7-400F (Non-Redundant Setup)

۷۱ 8-4-4 معماری طراحی سیستم S7-400HF (Fully-Redundant Setup)

۷۲ ۵-۴-۵ مثال هایی از پاسخ سیستم تحمل خطا (*Fault-Tolerant*) به یک فالت

۷۲ 4-5-1-1 مثال 1- توقف CPU یا ماژول تغذیه

۷۲ 2-5-4-2 مثال 2- خطا یا نقص در کابل فیبر نوری (Failure of a fiber-optic)

۷۳ ۶-۴-۶ انواع سیستم *F*

۷۳ ۷-۴-۷ نصب بسته *F System*

۷۳ 1-7-4-1 نصب بسته F system مربوط به HMI

۷۴ ۸-۴-۸ تنظیمات پیشرفته در PCS7 (*Advanced PCS 7 ES settings*)

۷۴ 1-8-4-1 حفاظت از دسترسی (Access Protection)

۷۴ 2-8-4-2 تنظیمات CFC برای کامپایل و دانلود

۷۵ ۹-۴-۹ پیکربندی سخت افزار S7F/FH

۷۵ 1-9-4-1 تطبیق پارامترهای CPU

۷۵ 2-9-4-2 سربرگ Cyclic interrupts

۷۶ ۱۰-۴-۱۰ تنظیمات پارامترهای ماژول های سیگنال *F*

۷۷ 1-10-4-1 تنظیمات سخت افزاری (DIP switch settings)

۷۸ 2-10-4-2 زمان مانیتورینگ F

۷۹ 3-10-4-3 مفهوم Passivation در حالت کلی (غیرفعال شدن کانال/ماژول های ورودی/خروجی)

۸۱ 4-10-4-4 پسو شدن گروهی (Group passivation)

۸۲ 5-10-4-5 فعال کردن کانال ها (Activating channels)

۸۴ ۱۱-۴-۱۱ ارزیابی سنسور (Sensor evaluation)

۸۴ 1-11-4-1 معماری های سیم بندی و انتخاب برای ET 200M (Wiring and Voting)

۱۶۲.....(2003 Voting with F-AI) F ورودی و آنالوگ با ماژول انتخاب 2003 -2-11-4

۸۶..... فصل - ۵ - برنامه نویسی بلاک های F (Programming) 1-1-5

۸۶..... ۱-۵ - مقدمه 1-1-5

۸۶..... (Configuring the safety program) F ایجاد برنامه/لاچیک 1-1-5

۸۷..... (safety program) ایمنی ساختار برنامه 2-1-5

۸۹..... (Rules governing program structure) F قوانین ساختار برنامه 1-1-5

۸۹..... (Run sequence of F-blocks) F اجرای بلاک های 2-1-5

۹۰..... (Data exchange) تبادل داده بین لاجیک F و برنامه استاندارد 3-1-5

۹۰..... روند پایه برای ایجاد برنامه ایمنی 4-1-5

۹۲..... (F-cycle monitoring) F زمان مانیتورینگ سیکل 5-1-5

۹۳..... قرار دادن گروه های F-Runtime در چارت CFC 6-1-5

۹۳..... گروه های F-Shutdown برنامه ایمنی 7-1-5

۹۳..... ۲-۵ - درج بلاک های F و اتصال آنها به همدیگر 2-5

۷۴..... ۳-۵ - مکانیزم های ایمنی 3-5

۱۲۷..... CPU در دسترسی از حفاظت 1-3-5

۹۴..... 5-4- پروسه *F startup & (re)start Protection* 5-4-

۹۴..... (F-Startup) F راه اندازی 5-4-1-

۹۴..... (Re)Start Protection-2-4-5

۹۶..... فصل - ۶ - کتابخانه سیستم F 6-6

۹۶..... ۱-۶ - مقدمه 6-6

۹۶..... (F-Data types) F نوع داده 1-1-6

۹۷..... (F-I/O access) F دسترسی به ورودی/خروجی های نوع 2-1-6

۹۸..... ۲-۶ - مجموعه بلاک های دایور کانال F (*F-Channel drivers*) 2-6

۹۹..... F_CH_DI: دایور کانال دیجیتال ورودی: 1-2-6

۱۰۲..... (F_CH_DO) دایور کانال دیجیتال خروجی 6-2-2-

۱۰۳..... F_CH_BI دایور 3-2-6

۱۰۴..... (F_CH_DII) DP Slave درایور قرائت مقدار DINT از یک آدرس 4-2-6

۱۰۴..... (F_CH_DIO) DP Slave درایور نوشتن مقدار DINT در یک آدرس 5-2-6

- ۱۰۵..... F_CH_AI - 6-2-6 بلاک درایور کانال ورودی آنالوگ
- ۱۰۸..... 7-2-6 بلاک درایور کانال آنالوگ خروجی
- ۱۰۹..... ۳-۶ مجموعه *BIT_LGC* - بلاک های لاجیک *F* با نوع داده *BOOL*
- ۱۰۹..... 1-3-6 بلاک *F_2OUT3*
- ۱۱۰..... 2-3-6 بلاک *F_XOUTY*
- ۱۱۰..... ۴-۶ - بلاک های مقایسه کننده (*comparing two input values*)
- ۱۱۱..... 6-4-1-1 بلاک مقایسه کننده *F_CMP_R*
- ۱۱۲..... ۵-۶ - بلاک های *Voter* برای نوع داده *REAL* و *BOOL* (*Voter blocks*)
- ۱۱۲..... 6-5-1-1 بلاک ارزیابی 2003 برای ورودی های *BOOL* (*F_2003DI*)
- ۱۱۳..... 2-5-6 بلاک *F_2003AI*
- ۱۱۳..... 3-5-6 بلاک *RED_IN*
- ۱۱۳..... ۶-۶ - مجموعه *MATH_FP* - بلاک های محاسباتی
- ۱۱۴..... 1-6-6 بلاک *F_ABS_R* (*F_ABS_R: Absolute value of a REAL value*)
- ۱۱۵..... 2-6-6 بلاک (*F_1002_R*)
- ۱۱۶..... 3-6-6 بلاک انتخاب حداکثر مقدار بین سه ورودی (*F_MAX3_R*)
- ۱۱۷..... 4-6-6 بلاک انتگرال گیر *F_INT_P*
- Error! Bookmark not defined.** ۷-۶ - بلاک های تبدیل داده (*data conversion*)
- ۱۱۸..... 1-7-6 بلاک *F_SWC_BO*
- ۱۱۹..... 2-7-6 بلاک *F_FR_FDI*
- ۱۱۹..... 3-7-6 بلاک *F_TON* - تایمر تاخیر در روشن (*F_TON: Timer switch-on delay*)
- ۱۱۸..... 6-8- بلاک های مبدل (*Convert*)
- ۱۱۸..... 1-8-6 بلاک *F_QUITES*
- ۱۲۶..... ۹-۶ - بلاک های فلیپ فلاپ
- ۱۱۹..... 1-9-6 تصدیق خطا توسط کاربر (*Acknowledgement by ACK_REI*)
- ۱۳۶..... فصل ۷ - پیکربندی ماژول های *IO* ریدادانت در *PCS7*
- ۱۳۷..... 1-1-7-1-1- دیگرام اتصالات ماژول خروجی آنالوگ
- ۱۴۷..... 7-2- پیکربندی *IO* ریداندانت در *Step7*
- ۱۴۷..... 7-2-1-1 تنظیم پارامترهای *CPU*

۱۴۸.....	فصل ۸ -
۱۶۲.....	1-1-8- فانکشن بلاک های سیستمی پشتیبانی شده در ماژول CPU (Supported SFB & SFC)
۱۶۳.....	۲-۸- اتصال S7 REDCONNECT
۱۶۴.....	1-2-8- پیکربندی اتصالات RedConnect
۱۶۸.....	8-3- Using SFC 14 and SFC15 (for Consistent data Transfer)
۱۶۸.....	8-4- فضای حافظه CPU (S7 System Memory Areas)
۱۶۹.....	1-4-8- عمل غیر فعال سازی ماژول های ورودی/خروجی (Passivation)
۱۷۰.....	۵-۸- جدول اختصارات و تعاریف
۱۷۰.....	1-5-8- قابلیت های کارت های Fail Safe
۱۷۳.....	فصل ۹ - نکات مربوط به پیکربندی در Step 7
۱۷۳.....	9-1- آدرس پایه تبادل داده برای CP343
۱۷۴.....	۲-۹-
۱۷۴.....	1- Fail-safe modules
۱۷۴.....	1-0-1- 1oo1 and 1oo2 evaluation with the fail safe signal module SM 326F
۱۷۶.....	F Systems: Wiring and Voting Architectures for ET200M F-DIs and F-Dos-2-0-1
۱۷۷.....	1-1- کنترل کننده های Fail-Safe شرکت Siemens
۱۷۷.....	1-1-1- چگونگی پردازش در یک کنترلر Fail-Safe شرکت Siemens
۱۷۸.....	2-1-1- PROFIBUS با پروتکل PROFIsafe برای ارتباطات Safety
۱۷۹.....	3-1-1- مزایای استفاده از کنترلر های Fail-Safe شرکت زیمنس
۱۸۰.....	4-1-1- کاربردها
۱۸۰.....	5-1-1- استانداردها و گواهینامه ها
۱۸۱.....	فصل ۱۰ - نصب لایسنس در PCS7
۱۸۲.....	۱-۱۰- سیم بندی ماژول های S7300
۱۸۲.....	1-1-10- سیم بندی ماژول های دیجیتال ورودی (DI)
۱۸۳.....	10-1-2- سیم بندی ماژول های دیجیتال خروجی (DO)
۱۸۴.....	3-1-10- سیم بندی ماژول های آنالوگ ورودی (AI)
۱۸۶.....	4-1-10- سیم بندی ماژول های آنالوگ خروجی (AO)
۱۸۷.....	فصل ۱۱ - ماژول های SIPLUS زیمنس
۱۹۳.....	1-1-11- فعال سازی کانال های IO (Reintegration)

فصل - ۱۲ - مدیریت خطاها ۱۹۳

Local data stack مقدار از عبور از خطا در 1-1-12 ۱۹۳

خطای ۱۹۵ -2-1-12

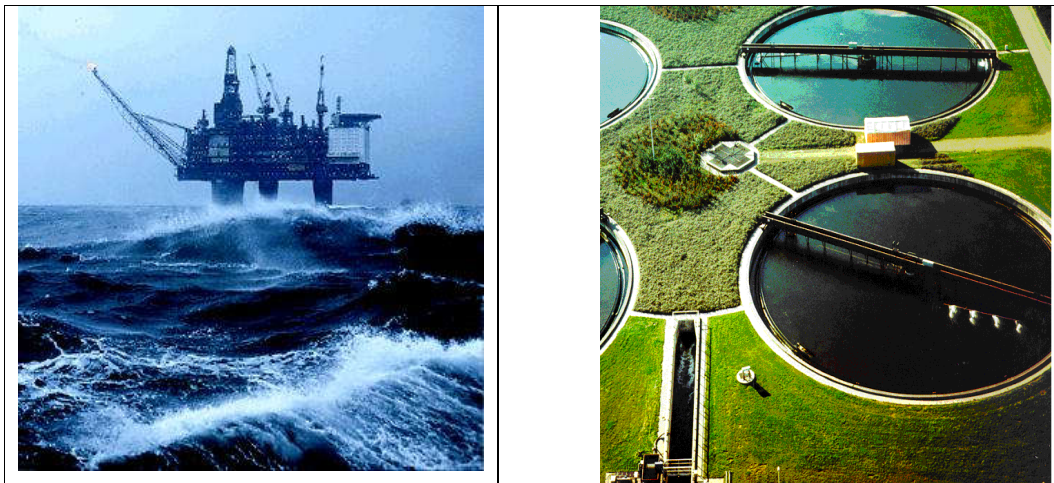
فصل ۱ – مقدمه

۱-۱- دلایل بکارگیری سیستم های H-PLC (Motivation)

- پرهیز از وقوع خرابی های پرهزینه در پردازش مواد ارزشمند
- هزینه های بالا در راه اندازی مجدد پس از وقوع نقص و توقف سیستم کنترل (Control Failure)
- عملیات بدون نظارت کارکنان تعمیر و نگهداری

۱-۱-۱- حوزه های کاربردی اتوماسیون S7-400H

- انرژی : خطوط لوله، سازه های داخل دریا
- استخراج معدن
- مواد شیمیایی، پتروشیمی
- تولید و توزیع برق (Power generation and distribution)
 - برای پلنت های نفت، گاز
- حمل و نقل (Transport)
 - مدیریت تونل
 - کشتی سازی
- صنعت شیشه
- صنایع غذایی



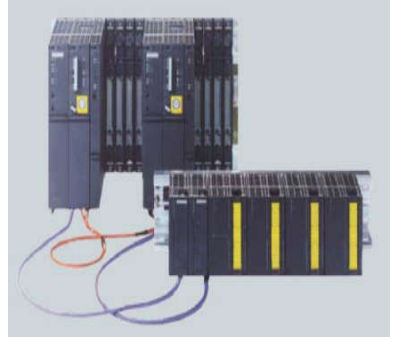


شکل ۱-۱- صنایع کاربردی سیستم های FH

۱-۲- ایستگاه اتوماسیون (AS)

یک ایستگاه اتوماسیون به یک مجموعه ماژول های سخت افزاری کنترل سیستم اتوماسیون پلنت مبتنی بر سامانه DCS فرآیندی زیمنس (PCS7) اطلاق می شود، که شامل ماژول های تغذیه (PS)، ماژول های CPU، ماژول شبکه ارتباطی (CP) و ماژول های ورودی/خروجی مرکزی و توزیع شده (RIO) است. هسته مرکزی سامانه اتوماسیون PCS7 را CPU های سری S7-400 تشکیل می دهد و لذا سامانه اتوماسیون بسته به نوع پردازنده سری S7400 به کار رفته در آن شناخته می شود. سامانه اتوماسیون در PCS7 براساس یکی از سه نوع کنترل کننده S7-400 است. که عبارت انداز:

- سامانه اتوماسیون استاندارد AS400 (Standard Systems)؛
- سامانه اتوماسیون افزونه AS400H (Redundant System)؛
- سامانه اتوماسیون ایمن در نقص AS400FH (Failsafe)؛

شکل ۱-۲ نمایشی از سامانه های اتوماسیون مبتنی بر CPU های S7-400 را نشان می دهد.

		
<p>ج- AS400FH</p> <ul style="list-style-type: none"> ▪ فراهم کردن ایمنی برای افراد ▪ سازگار با استانداردهای: ▪ EN 954 (up to category 4) ▪ IEC 61508 (up to SIL 3) ▪ قابلیت برنامه نویسی Fail-safe ▪ تحمل پذیر در نقص Faulttolerant 	<p>ب- AS400H</p> <ul style="list-style-type: none"> ▪ اجتناب از هزینه های بالای راه اندازی مجدد پلنت ▪ اجتناب از زمان های توقف (down times) ▪ سوئیچ بدون مکث بین دو CPU 	<p>الف- AS400</p> <ul style="list-style-type: none"> ▪ دارای کارایی و سرعت بالا ▪ قابلیت تعویض آنلاین در ماژول ها ▪ دارای زمان های سیکل از 10 ms به بالا

شکل ۱-۲- انواع سامانه اتوماسیون مبتنی بر S7-400

جدول ۱-۱- فهرست اجزای اصلی سامانه های اتوماسیون AS در PCS7

SIMATIC 400	Modules	Parts
CP – 400	Industrial Ethernet	CP 443-1 6GK7 443-1EX11-0XE0, V2.0 or higher
	PROFIBUS	CP 443-5 Extended 6GK7 443-5DX03-0XE0, V5.0 or higher
CPU – 400	CPU 414-4H	6ES7 414-4HJ00-0AB0, V3.1 or higher
	CPU 417-4H	6ES7 417-4HL01-0AB0, V3.1 or higher
	CPU 414-3 DP	6ES7 414-3XJ00-0AB0, V3.1 or higher
	CPU 416-2 DP	6ES7 416-2XK02-0AB0, V3.1 or higher
	CPU 416-3 DP	6ES7 416-3XL00-0AB0, V3.1 or higher
	CPU 417-4	6ES7 417-4XL00-0AB0, V3.1 or higher
PS-400	Redundant PS-400	PS 405 10A, 6ES7 405-KR00-0AA0
		PS 407 10A, 6ES7 407-KR00-0AA0
	Standard PS-400	PS 405 10A, 6ES7 405-KA01-0AA0
		PS 405 20A, 6ES7 405-RA01-0AA0
		PS 407 10A, 6ES7 407-KA01-0AA0
		PS 407 20A, 6ES7 407-RA01-0AA0
RACK – 400	9 slots, UR2	6ES7 400-1JA00-0AA0 6ES7 400-1JA01-0AA0
	18 slots, UR1	6ES7 400-1TA00-0AA0 6ES7 400-1TA01-0AA0
	H system, UR2-H	6ES7 400-2JA00-0AA0

The S7-400 is available in three versions:

S7-400

- The power PLC for the mid to high-end performance ranges
- The solution for even the most demanding tasks
- With a comprehensive range of modules and performance-graded CPUs for optimal adaptation to the automation task
- Flexible in use through simple implementation of distributed structures and extensive communications capabilities
- User-friendly handling and uncomplicated design without a fan
- Can be expanded without problems when the tasks increase

S7-400H

- Fault-tolerant automation system with redundant design
- For applications with high fail-safety requirements
Processes with high restart costs, expensive downtimes, little supervision, and few maintenance options.
- Redundant central functions
- Increases availability of I/O: Switched I/O configuration
- Also possible to use I/Os with standard availability: Single-sided configuration
- Hot stand-by: Automatic reaction-free switching to the standby unit in the event of a fault
- Configuration with two separate or one divided central rack
- Connection of switched I/O via redundant PROFIBUS DP

S7-400F/FH

- ◀ Failsafe automation system for systems with increased safety requirements

- ◀ If required, also fault tolerant through redundant design
- ◀ Without additional wiring of the safety-related I/O:
Safety-relevant communication via PROFIBUS DP with PROFIsafe profile
- ◀ Based on S7-400H and ET 200M with safety modules
- ◀ Standard modules for non-safety-related applications can also be used in the automation system

۱-۲-۱- سامانه اتوماسیون استاندارد (سری AS400)

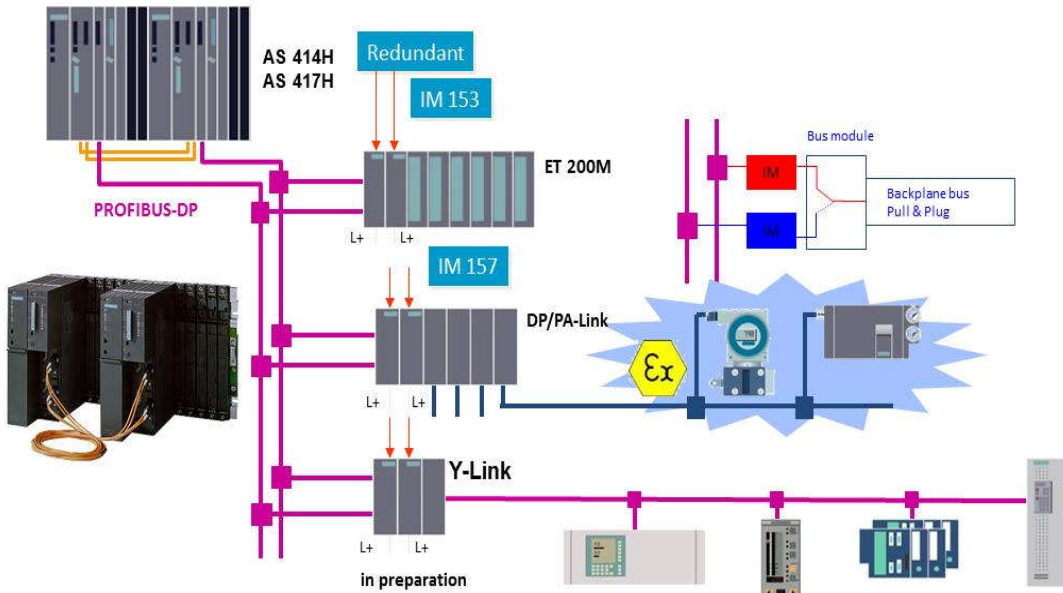
در این پیکربندی سامانه اتوماسیون، تنها دارای یک پردازنده S7-400 بوده و در کاربردهای معمولی که فاقد محیط خطرزا است و همچنین به مؤلفه «دسترس پذیری بالا» (افزونگی) نیازی وجود نداشته باشد. از این سری استفاده می شود. ماژول های کنترل کننده این نوع سامانه شامل مدل های زیر است:

➤ Standard CPU Modules

- CPU 414-3DP;
- CPU 418-3DP;
- CPU 418-2DP;
- CPU 414-3DP;
- CPU 417-4;

۱-۲-۲- سامانه اتوماسیون برای دسترسی پذیری بالا (AS400H)

در این نوع سامانه از ماژول های CPU نوع H استفاده می شود. حرف H در این نوع سامانه معرف قابلیت دسترسی پذیری (High Availability) بالا یا «افزونگی داغ» (Hot Redundant or Hot Standby) می باشد. این سامانه در فرایندهایی کاربرد دارد که در آنها توقف عملیات فرایند غیرمجاز است. شکل ۱-۳ یک پیکربندی نمونه از یک سیستم H با مدل پردازنده CPU 414/417 را نشان می دهد.



شکل ۱-۳- پیکربندی نمونه از یک سیستم H با CPU 414/417



شکل ۱-۴- نمونه یک سیستم S7-400H

سیستم های H، سیستم هایی هستند که با قابلیت افزونگی در سطوح مختلف سیستم کنترل (کنترل کننده، ماژول های I/O، شبکه، منابع تغذیه) به صورت پشتیبان یکدیگر می توانند یک فرایند را کنترل کنند. به طور معمول در ساختار افزونگی سیستم های H زیمنس، از یک جفت CPU استفاده می شود. که در آن فرایند موردنظر همواره تحت کنترل یکی از آنها که Master نامیده می شود، است. سیستم دیگر نیز که Standby نامیده می شود، در حالت آماده باش یا آماده به کار بوده و در آن لحظه نقشی در کنترل فرایند ندارد. ولی در صورت بروز نقص در Master، سیستم پشتیبان یا Standby در زمان بسیار کوتاهی وارد مدار شده و کنترل فرایند را ادامه می دهد. وقتی در یک سیستم کنترل، یکی از جفت کنترل کننده در حالت کار و دیگری آماده به کار است، مد کاری را افزونه (Redundant) می گویند. وقتی یکی از کنترل کننده ها به دلیل خطا متوقف می شود و دیگری در مدار می آید، مد کاری را Single می نامند. ساختاری که به این صورت ایجاد می شود افزونگی سخت افزاری خوانده می شود و نباید آن را با افزونگی نرم افزاری اشتباه گرفت. در واقع مطابق جدول ۱-۱ بین این دو سیستم از نظر ساختار و از نظر عملکرد تفاوت های اساسی زیر وجود دارد:

جدول ۱-۱- تفاوت های بین افزونگی سخت افزاری و نرم افزاری

افزونگی سخت افزاری	افزونگی نرم افزاری
فقط با CPU های خاصی از خانواده S7400 قابل پیاده سازی است.	با CPU های خانواده S7300 و S7400 قابل پیاده سازی است.
عمل سنکرون سازی حافظه بین دو CPU به طریق سخت افزاری و از طریق ماژول های SYNC که با فیبر نوری به هم متصل اند انجام می شود.	سنکرون سازی حافظه دو CPU با برنامه نویسی بلوک توابع خاص و از طریق کابل شبکه (MPI-Profibus-Ethernet) انجام می شود.
قابلیت اطمینان بالاتر دارد و زمان جابجایی از سیستم اصلی به سیستم پشتیبان خیلی کوتاه هست.	قابلیت اطمینان آن به اندازه سیستم H نبوده و زمان جابجایی آن نیز به کوتاهی سیستم های H نیست

هزینه آن زیاد است.	هزینه بالایی ندارد.
--------------------	---------------------

انتخاب بین دو روش فوق بر اساس نیاز فرآیند صورت می گیرد. ممکن است در فرایندی خسارات ناشی از توقف آن قدر زیاد باشد که سرمایه گذاری اولیه برای داشتن یک سیستم H در مقابل آن ناچیز محسوب شود. سیستم های H در S7 فقط توسط S7-400H قابل پیاده سازی هستند.

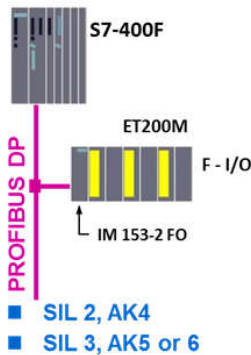
۱-۲-۳- سامانه اتوماسیون ایمن در خطا (Fail-Safe AS400FH)

سامانه اتوماسیون AS400FH یک سیستم ترکیبی با مشخصه افزونه و تحمل پذیر خطا (Fail-Safe) است. به طوری که سامانه PCS7 برای داشتن یک سیستم با میزان دسترسی بالا و در عین حال ایمن، از سری CPU های با ساختارهای ترکیبی FH استفاده می کند. معماری S7-400FH، در دسترس پذیری بالا و فن آوری ایمنی را در یک سیستم اتوماسیون واحد ترکیب می کند و با حفظ ایمنی (PROFIsafe) در برابر نقص ها تحمل پذیر است.

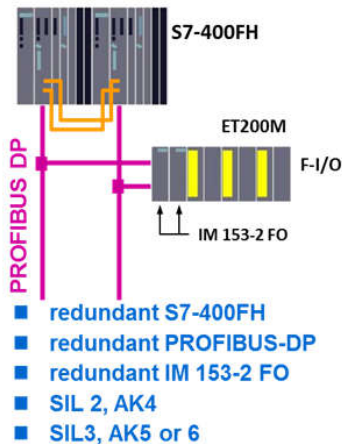
این سیستم دارای گواهی نامه TÜV بوده و مطابق با تمام استانداردهای مربوط به تولید و اتوماسیون فرایند است. این سامانه در محیط های پرخطر که نیاز به ایمنی بالا و افزونگی دارد، به کار گرفته می شود. شکل ۵-۱ پیکربندی های نمونه برای سیستم های FH را نشان می دهد. ماژول های CPU این نوع سامانه شامل مدل های زیر است:

- AS 414-2H;
- AS 417-4H;
- AS 417-5H;

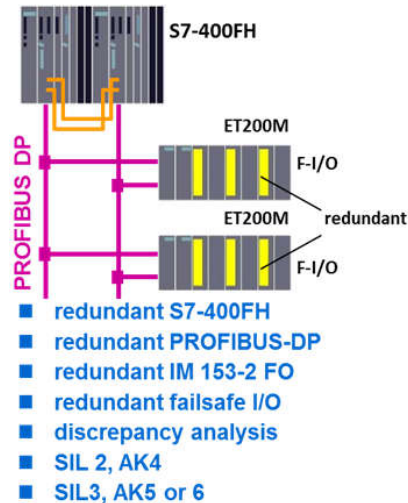
Assembly with single-channel, one-sided I/O



Assembly with single-channel, switched I/O



Assembly with redundant, switched I/O

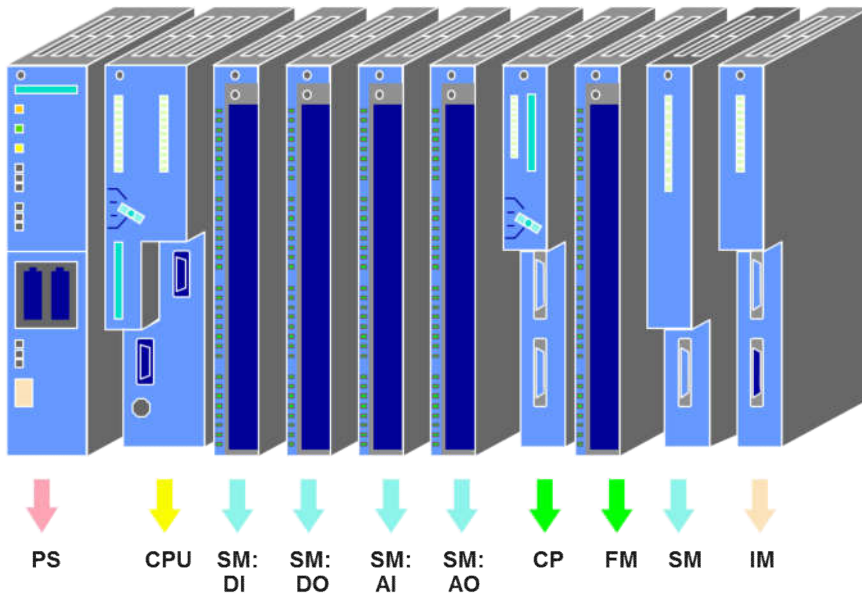
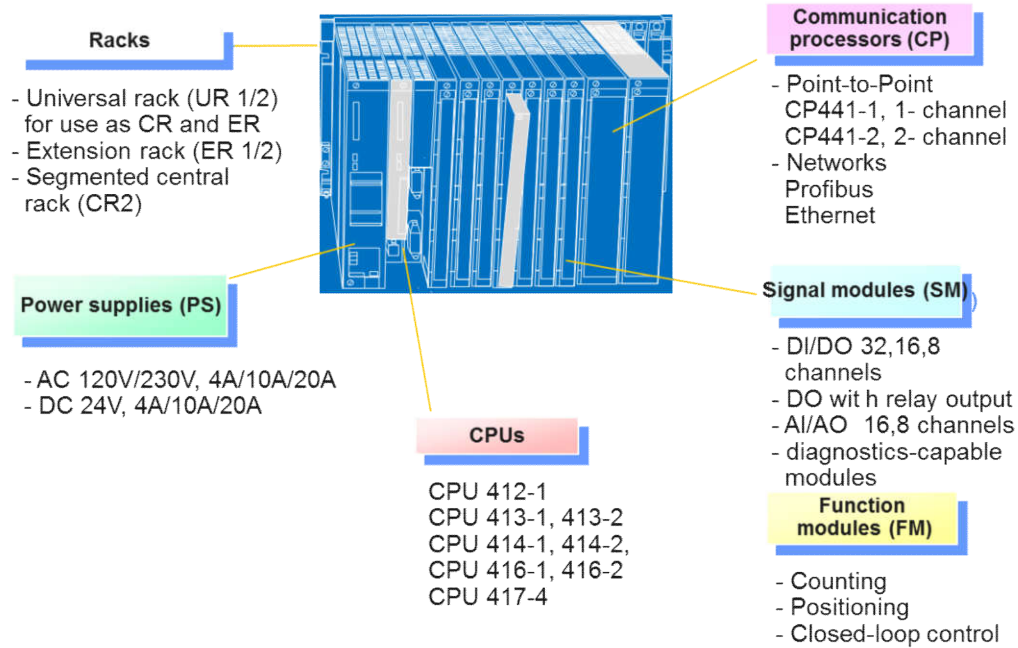


شکل ۵-۱- نمونه پیکربندی های تحول پذیر نقص سیستم های S7-400FH

فصل ۲ - سامانه AS-400 استاندارد

۲-۱- اجزای AS-400 استاندارد

شکل زیر اجزای یک سیستم AS-400 استاندارد را به تصویر کشیده است.



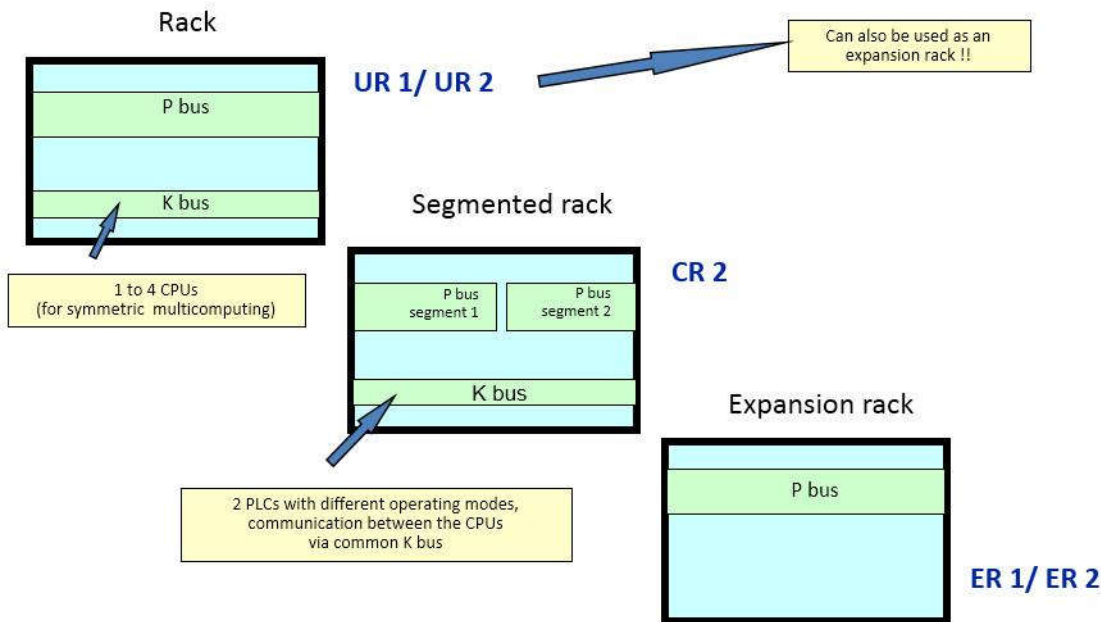
شکل ۲-۱- اجزای یک سیستم S7-400 استاندارد

۲-۲- انواع رک S7-400

برخلاف سیستم S7-300 که تنها یک نوع رک (ریل آلومینیومی) دارد. سیستم S7-400 دارای سه نوع رک می باشد. که هر یک دارای مشخصات و مزایای خاص خود می باشد.

- CR : *Central Rack*
- ER : *Expansion Rack*
- UR : *Universal Rack*

عنوان تمام رک های به کاررفته برای هر سه نوع سامانه AS400، UR (Universal rack) نام دارد که در چند مدل به شرح زیر استفاده می شود. شکل زیر انواع رک در S7-400 را به تصویر کشیده است.



شکل ۲-۲- انواع رک در S7-400

- UR2: Universal rack, 9 slots, not suitable for redundant Power Supply Modules
- UR2: Universal rack, 9 slots
- UR2ALU: Universal aluminium rack, 9 slots
- UR1: Universal rack, 18 slots, not suitable for redundant power supply modules
- UR1: Universal rack, 18 slots
- UR1ALU: Aluminium universal rack, 18 slots
- UR2-H: Central rack, 2*9 slots, split backplane bus, suitable for compact configuration of standard and redundant PLCs
- UR1ALU: Central aluminium rack, 2*9 slots, split backplane bus, suitable for compact configuration of standard and redundant PLCs

رک UR2-H نصب و راه اندازی دو زیرسیستم جداگانه را به صورت افزونه با ۹ اسلات پشتیبانی می کند که برای نصب در کابینت ۱۹ اینچی مناسب است. همچنین می توان S7-400H را با دو رک جداگانه UR1 و UR2 پیکربندی کرد.

جدول ۱-۲- انواع رک در S7-400

عنوان	نوع	کاربرد	تعداد اسلات	توضیح
UR1	Universal	بعنوان رک اصلی بعنوان رک بسط I/O	۱۸	دارای دو نوع است که در یک نوع آن می توان دو منبع تغذیه از نوع ریداندانت قرار داد.
UR2	"	"	۹	"
ER1	Expansion	به عنوان رک بسط I/O	۱۸	"
ER2	"	"	۹	"
CR2	Segmented	به عنوان رک اصلی	۱+۸ ۰	دارای دو بخش است که هر بخش آن می تواند یک رک اصلی مستقل باشد.
UR2-H	Universal	برای PLC های ریداندانت به کار می رود	۲*۹	شبهه دو رک اصلی UR2 است.

۲-۳- انواع باس در رک S7-400

بر روی رک ۴۰۰ دو نوع باس وجود دارد

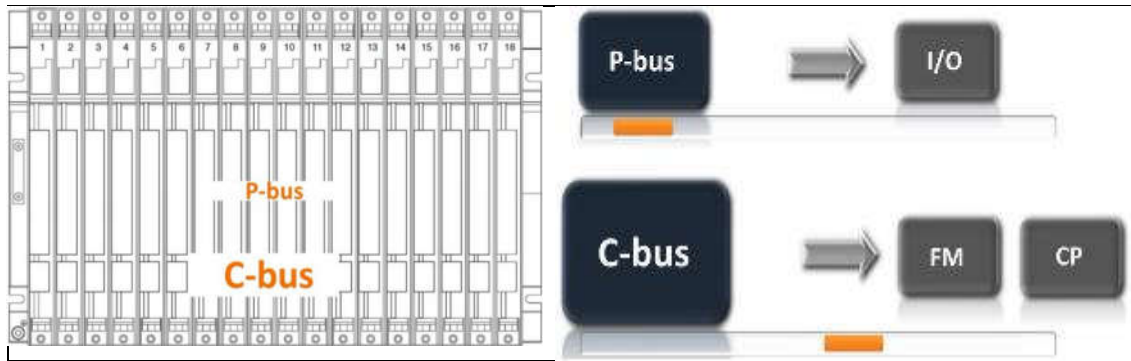
▪ گذرگاه P-bus یا IO Bus

حرف P معرف Perioheral می باشد. این باس تبادل داده بین CPU و ماژول های I/O را برعهده دارد.

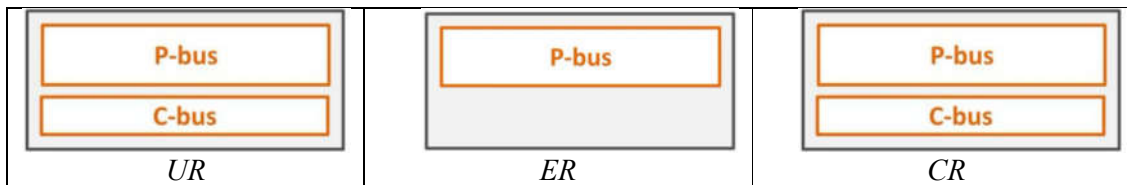
▪ گذرگاه C-bus یا K-bus (اول حرف عبارت آلمانی ارتباط Kpmmunikation)

حرف C معرف Communication می باشد. این باس تبادل داده بین CPU و ماژول های CP و FM را

بر عهده دارد. سرعت تبادل داده در C-bus بیشتر از P-bus می باشد.



شکل ۳-۲- دو نوع گذرگاه بر روی رک S7-400



شکل ۴-۲- باس های موجود بر روی رک های S7-400

- رک مرکزی CR، رکی است که تنها به عنوان Main Rack استفاده می شود و بایستی به عنوان اولین رک نصب شود. این نوع رک در پیکربندی چند لایه (Multi tire) به عنوان رک Expansion نمی تواند مورد استفاده قرار گیرد. رک CR دارای هر دو گذرگاه P و C می باشد.
- رک ER فقط به عنوان باس Expansion استفاده می شود و تنها دارای باس P می باشد. لذا تنها ماژول های I/O بر روی این باس قابل نصب هستند.
- رک UR به طور وسیعی به هر دو صورت رک Main و Expansion استفاده می شود.
- رک UR دارای هر دو باس P و C می باشد. لذا کلیه ماژول های I/O، CP و FM قابل نصب بر روی این رک می باشد.

فصل - ۳ - ساختار و معماری سامانه S7-400H

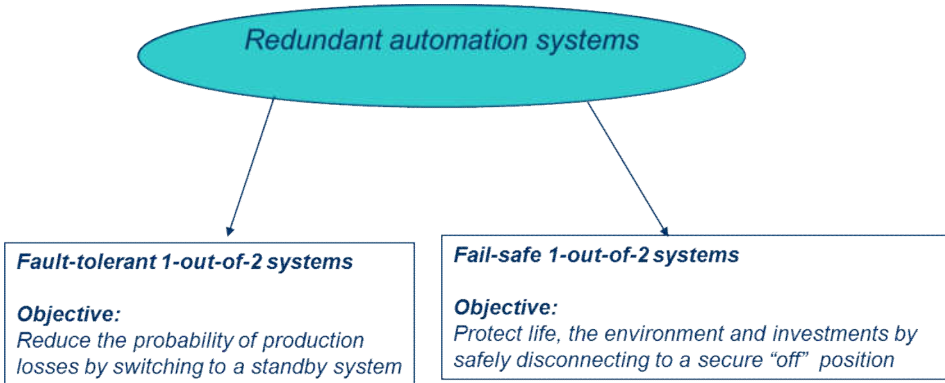


۳-۱- مقدمه

در سیستم های اتوماسیون افزونه دو مفهوم مهم و اساسی مدنظر می باشد که عبارت انداز:

1. *Fault-tolerant*
2. *Fail-safe*

هدف از سیستم های نوع اول کاهش احتمال از دست دادن تولید محصول از طریق سوچ به سیستم یدکی می باشد. و هدف از سیستم های نوع دوم حفاظت از جان افراد و تجهیزات محیط کار از طریق هدایت سیستم به موقعیت ایمن در مواقع بروز خط و اشکال می باشد.



چرا بایستی از یک سیستم کنترل PLC با مشخصه تحمل پذیر خطا استفاده کنیم؟

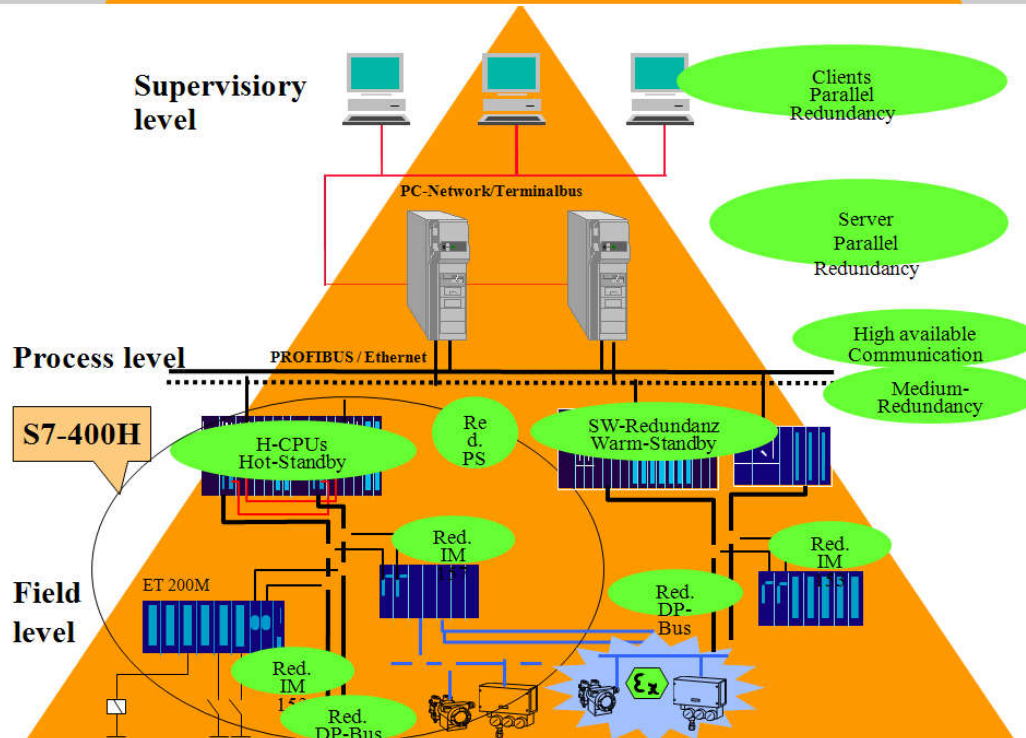
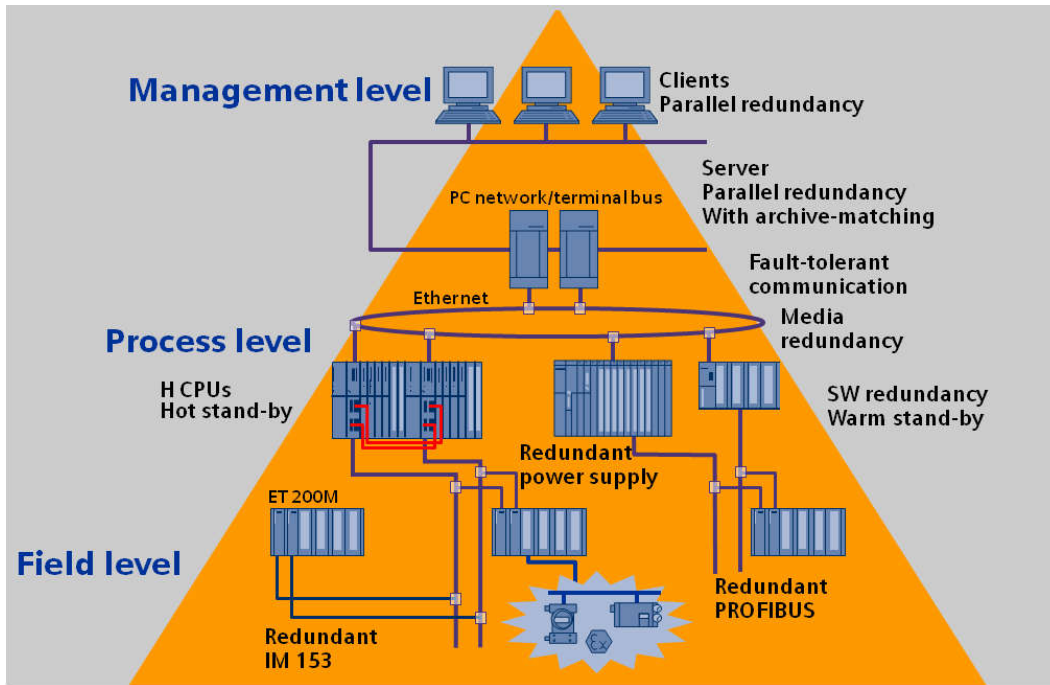
- هدف استفاده از کنترل کننده های PLC با قابلیت در دسترس پذیری بالا، کاهش تلفات تولید است. مهم نیست که آیا از دست دادن محصول ناشی از یک خطا است و یا در نتیجه کار تعمیر و نگهداری ایجاد می شود. هر چقدر هزینه های توقف بالاتر باشد، استفاده از یک سیستم تحمل پذیر خطا ارزشمند تر است. به طور کلی هزینه های سرمایه گذاری بالا در سیستم های تحمل پذیر خطا با اجتناب از کاهش تولید جبران می شود.

به طور کلی برای سیستم های اتوماسیون به دو صورت افزونگی قابل پیاده سازی می باشد:

- افزونگی نرم افزاری (*Software Redundancy*) - برای فرایندهای کند استفاده می شود.
- افزونگی سخت افزاری (*Hardware Redundancy*) - برای فرایندهای سریع استفاده می شود.

۳-۲- معماری سیستم (System architecture)

معماری سیستم S7-400H، به گونه ای است که ساختار افزونگی در تمام سطوح هرم اتوماسیون و اجزای سخت افزاری و نرم افزاری را پشتیبانی می کند. شکل زیر معماری نوعی را برای سیستم S7-400H به تصویر کشیده است.



شکل ۱-۳- افزونگی سامانه S7-400H در سطوح هرم اتوماسیون برای دسترس پذیری بالا (High Availability)

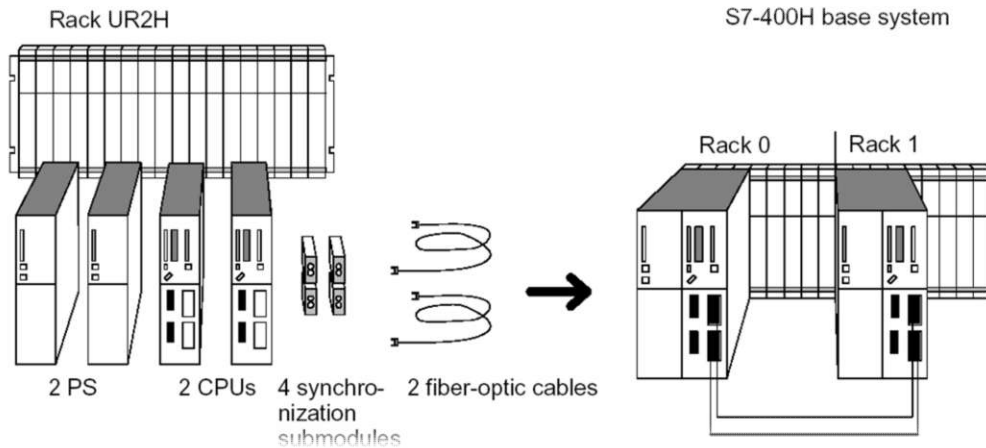
۳-۳- اجزای سخت افزاری یک سیستم S7-400H (Minimum Configuration)

یک کابینت سیستم AS400H حداقل دارای اجزای زیر است.

۱. دو عدد ماژول CPU؛

۲. دو عدد کابل فیبر نوری؛
۳. دو عدد منبع تغذیه؛
۴. جفت رک؛

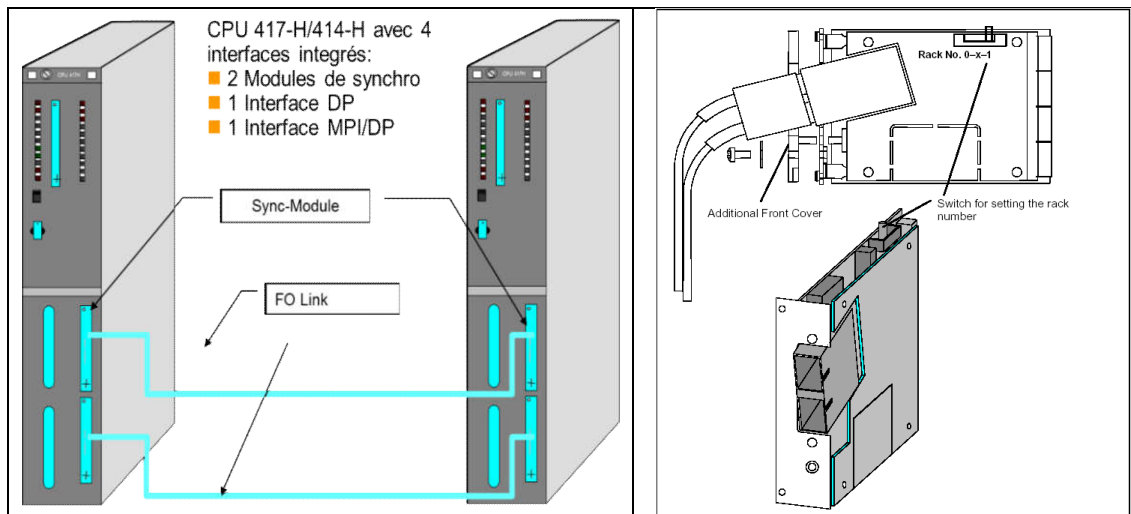
۵. چهار عدد ماژول سنکرون سازی (دو عدد برای هر CPU برای دو کابل افزونه فیبر نوری)



حداقل اجزای سخت افزاری یک سیستم H

۳-۳-۱- ماژول کنترل کننده (CPU)

در مرکز یک سیستم S7-400H، دو واحد CPU می باشد. تنظیم ماژول های سنکرون سازی، که به CPU متصل می شوند، شماره رک برای CPUها را تعریف می کند. در سیستم H از CPU در رک صفر، CPU 0 و از CPU نصب شده در رک ۱ با عنوان CPU1 یاد می شود.



شکل ۳-۲- ساختار کنترل کننده S7-400H

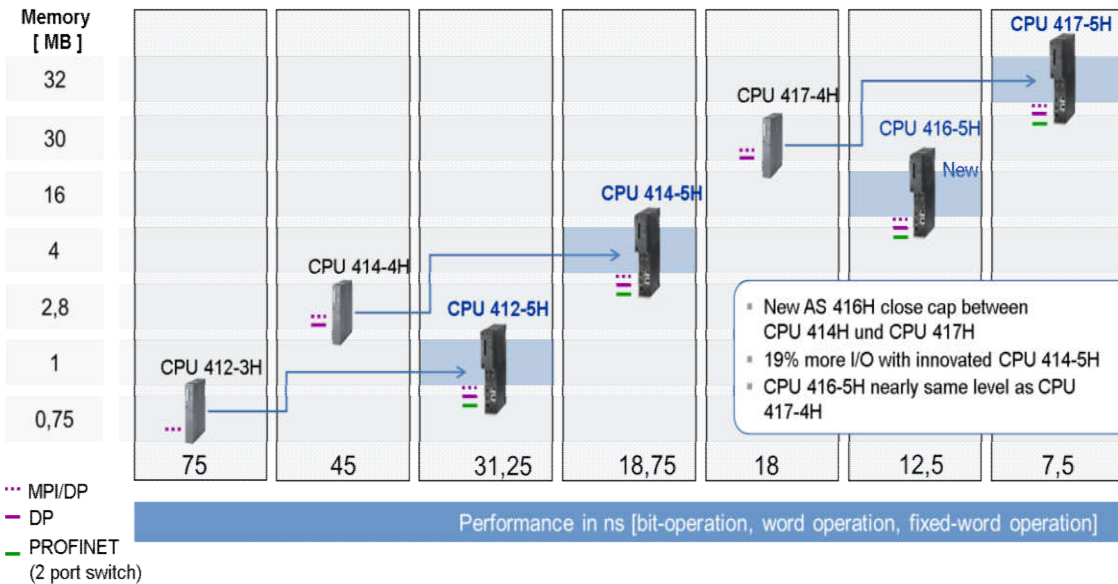
Switch Position	Meaning
	The CPU has rack number 1
	The CPU has not been assigned a rack number
	The CPU has rack number 0

شکل ۳-۳- تنظیم شماره رک بر روی H-CPU

شکل زیر مشخصات مدل های مختلف ماژول CPU سامانه اتوماسیون S7-400H را نشان می دهد.

➤ Fault-tolerant CPUs

- CPU 412-3H; CPU 412-5H; CPU 414-4H; CPU 414-5H; CPU 416-5H; CPU 417-4H;
- CPU 417-5H;



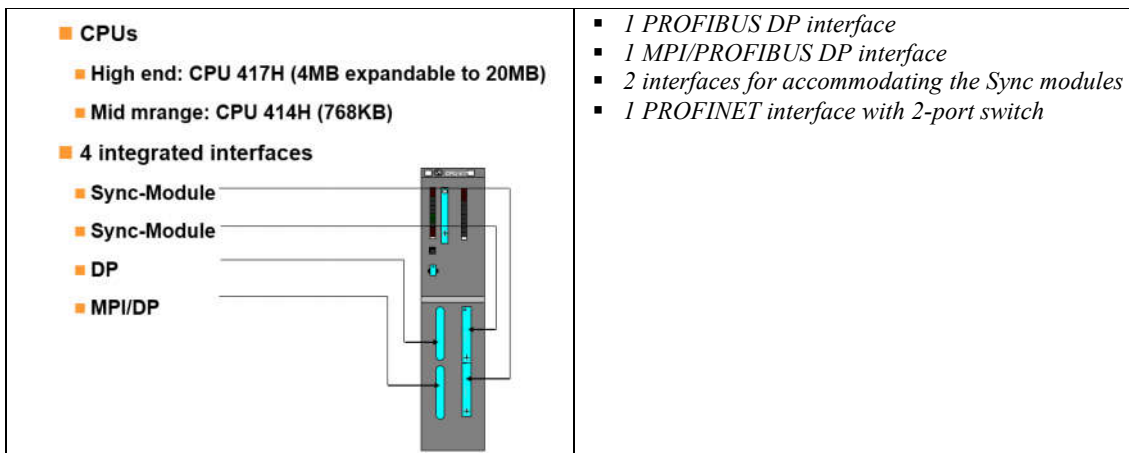
شکل ۳-۴- مدل های مختلف ماژول S7400 H-CPU

شکل ۳-۵- مشخصات فنی ماژول های H CPU را به تصویر کشیده است.

Technical data: H-CPU				
CPU	CPU 412-5H	CPU 414-5H	CPU 416-5H	CPU 417-5H
Dimensions (mm)	50x290x219			
Slots	2			
Order No. group: 6ES7	412-5HK.	414-5HM.	416-5HS.	417-5HT.
Firmware	V6.0			
Work memory				
Integrated	1 MB	4 MB	16 MB	32 MB
For program	0.5 MB	2 MB	6 MB	16 MB
For data	0.5 MB	2 MB	10 MB	16 MB
Processing times				
Bit operation	31.25 ns	18.75 ns	12.5 ns	7.5 ns
Word operation	31.25 ns	18.75 ns	12.5 ns	7.5 ns
Fixed-point operation	31.25 ns	18.75 ns	12.5 ns	7.5 ns
Floating-point operation	62.5 ns	37.5 ns	25 ns	15 ns
Bit memories, timers, counters				
Bit memory	8192			
S7 timers/counters	2048/2048			
IEC timers/counters	Unlimited (limited only by the work memory)			
Address ranges				
I/O	8192 bytes			
I/O process image	256 bytes		1 024 bytes	
DP interfaces				
Number of MPI/DP interfaces	1			
Number of DP interfaces	1			
Plug-in interfaces	-			
PN interfaces				
Number of PN interfaces	1			
PROFINET IO	Yes			
PROFINET with IRT	No			
PROFINET CBA	No			
TCP/IP	Yes			
UDP	Yes			
ISO-on-TCP (RFC 1006)	Yes			
Web server	No			
Data set gateway	Yes			

شکل ۳-۵- مشخصات فنی ماژول های H CPU

نکته: عناوینی چون 3H یا 4H و غیره به تعداد پورت های موجود بر روی CPU اشاره می کند. به عنوان مثال CPU 414-4H دارای دو پورت SYNC، یک پورت DP و یک پورت MPI/DP می باشد.



شکل ۳-۶- واسط های ارتباطی برای CPU 414-4H

جدول ۳-۱- مقدار حافظه و تعداد پورت های روی CPU های مختلف نوع H

CPU 412-5H	CPU 414-5H	CPU 416-5H	CPU 417-5H
------------	------------	------------	------------

(Code/Data)	1MB 512kB/512KB	4MB 2MB/2MB	16MB 6MB/10MB	32MB 16MB/ 16MB
DP/MPI	1	1	1	1
DP	1	1	1	1
PN (inkl. 2port-Switch)	1	1	1	1
Safety (optional)	✓	✓	✓	✓
Max. length	10 km	10 km	10 km	10 km

مشخصات نسخه FW V6.0 یک CPU از مدل CPU 412-5H عبارت انداز:

- New middle-class controller CPU 416H
- Increased performance
- Increased resources for communication
- Ethernet integrated (without additional CP)
- Routing for integrated PROFIBUS interface
- PROFINET System Redundancy
- CPU 412-5H: Synchronization up to 10 Km

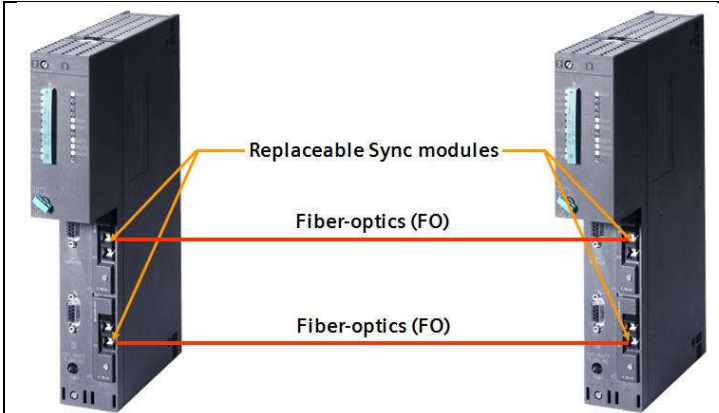
ماژول های I/O نیز به صورت ایستگاه های ET200M به سیستم H متصل می شوند.

➤ IO – ET 200M

- ET 200M : IM 153-4 PN (6ES7153-4BA00-0XB0)
- ET 200MP

۳-۲-۳- مازول های سنکرون سازی (SYNC Modules)

این ماژول ها برای اتصال CPU های نوع H به یکدیگر از طریق دو کابل فیبر نوری به صورت افزونه به کار می روند. روی هر CPU دو ماژول سنکرون سازی (SYNC) قرار می گیرد. روی ماژول های SYNC یک DIP Switch وجود دارد که دارای وضعیت ۰ و ۱ است. برای رک صفر آن را روی صفر و برای رک دیگر آن را روی یک تنظیم می کنیم. با روشن شدن سیستم ابتدا آنکه روی صفر تنظیم شده به عنوان Master کار خواهد کرد. بسته به کابل Sync یا نوع پیکربندی رک برای فواصل ۱۰ متر یا ۱۰ کیلومتر دو نوع ماژول SYNC وجود دارد.



- For Sync cables up to 10 m in length
- For Sync cables up to 10 km in length with the CPUs 414-4H or 417-4H for applications in which the subunits have to be set up at some distance

شکل ۷-۳- نمایش از درگاه CPU برای ماژول های سنکرون سازی

۳-۳-۳- کابل فیبر نوری

بین هر دو ماژول SYNC یک زوج فیبر برای برقراری ارتباط افزونه متصل می شود.

۳-۴-۲- مشخصه های افزونگی (Redundancy features)

افزونگی سیستم های H دارای ویژگی های زیر است:

۳-۴-۱- تعویض بی درنگ بین دو کنترل کننده (Bumpless Master-Stand-by switchover)

سری S7-400H تعویض آنی و بدون مکث (Bumpless) بین Master و Stand-by را با مشخصات زیر فراهم می کند.

▪ زمان تعویض (Switchover time):

- زمان تعویض زیر ۱۰۰ میلی ثانیه؛
- حفظ وضعیت خروجی ها در طول زمان تعویض (Outputs are Retained)؛
- از دست نرفتن هیچ گونه اطلاعات آلام یا وقفه (Alarm/Interrupt)؛

▪ معیار تعویض (Switchover Criteria):

معیار تعویض از Master به Stand-by وجود خرابی در Master می باشد. در صورت خرابی در Master که ناشی از نقص در یکی از اجزای زیر است، عمل تعویض به سیستم Stand By صورت می گیرد.

- Power supply;
- Rack;
- Sync module;
- Sync cable;
- CPU;

نقص یا خرابی (Failure) در ماژول واسط DP Slave موجب تعویض نمی گردد.

۳-۴-۲- سنکرون سازی بین دو CPU (Synchronization Procedure)

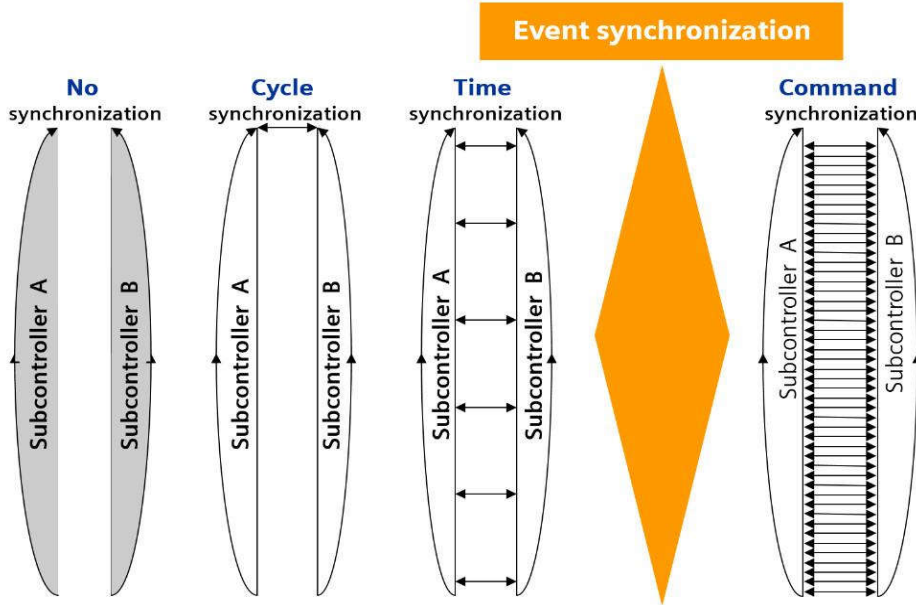
برای انجام سوییچ بدون مکث بین دو CPU در زمان بروز خرابی در یکی از آنها، سنکرون سازی بین دو سیستم Master و Stand-By ضروری است. مطابق شکل زیر سه روش سنکرون سازی برای عملیات دو CPU قابل تعریف است:

- سنکرون سازی در انتهای هر سیکل اسکن (Cycle Synchronization)؛
- سنکرون سازی زمانی (Time synchronization) - عمل سنکرون کردن در بازه های زمانی به خصوصی صورت می گیرد؛
- سنکرون سازی با اجرای هر دستور لاجیک (Command synchronization)؛

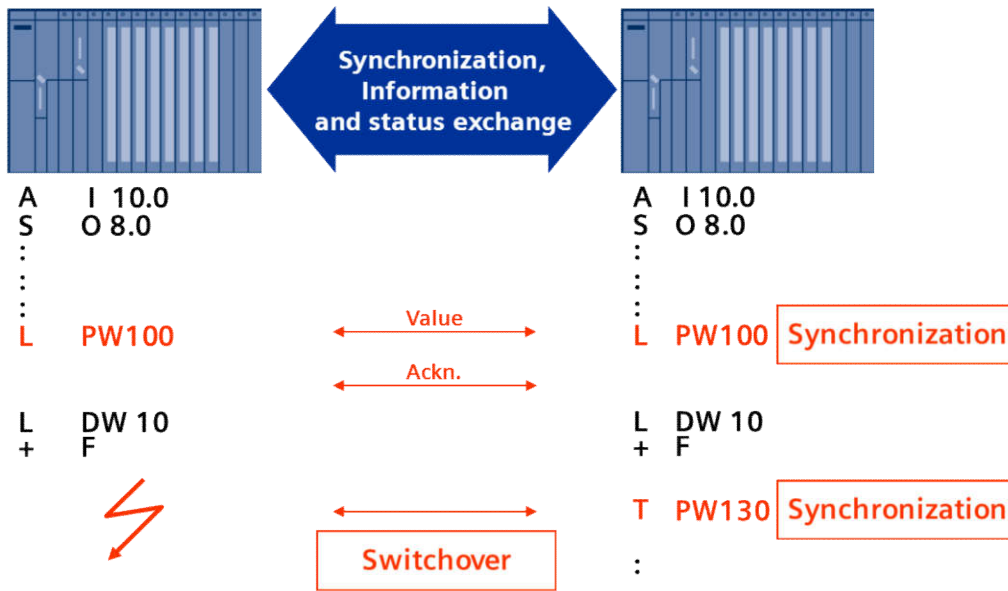
این روش همان تکنیک سنکرون سازی مبتنی بر رخداد (*Event Driven Synchronization*) است که CPU های H زیمنس از آن استفاده می کنند و حق امتیاز (*Patent*) آن برای زیمنس ثبت شده است. در این روش عمل سنکرون کردن داده بین دو CPU در زمان اجرای برنامه لاجیک (*User Program*) صورت می گیرد. به عبارت دیگر هر زمان که رخدادها در CPU مستر، منجر به وضعیت داخلی مختلف شد، یک عملیات هماهنگ سازی صورت می گیرد. عمل سنکرون بین دو CPU در مواقع زیر صورت می گیرد:

- در زمان دسترسی مستقیم به I/O؛
- در موقع وقوع وقفه ها و آلامها؛
- به روز رسانی زمانها در برنامه ی کاربر؛
- اصلاح یا تغییر داده ها با استفاده از توابع شبکه یا ارتباطات؛

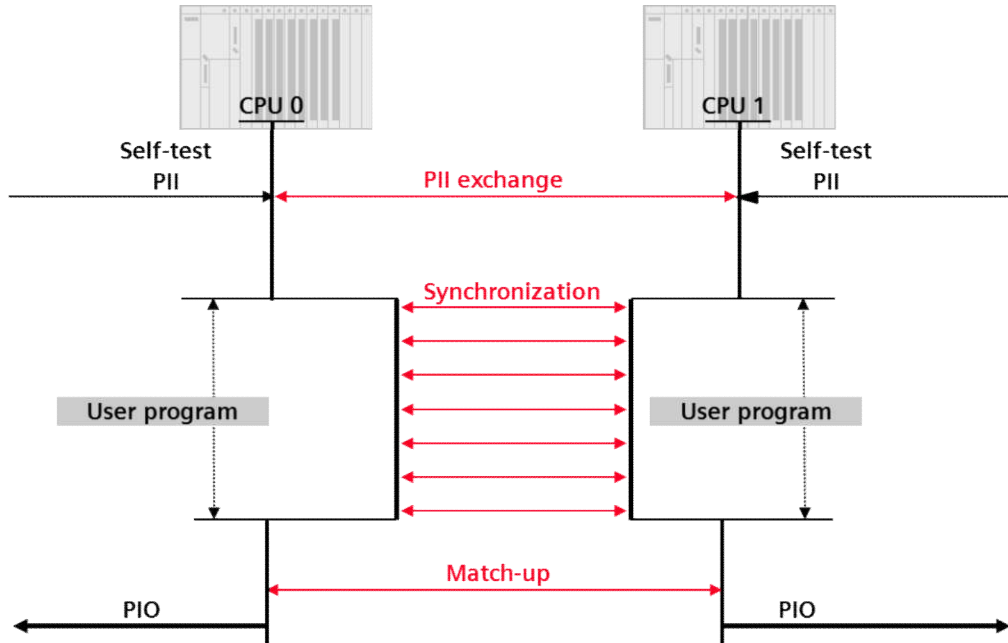
عمل هماهنگ سازی به طور خودکار و توسط سیستم عامل صورت می گیرد و نیاز به برنامه نویسی توسط کاربر ندارد.



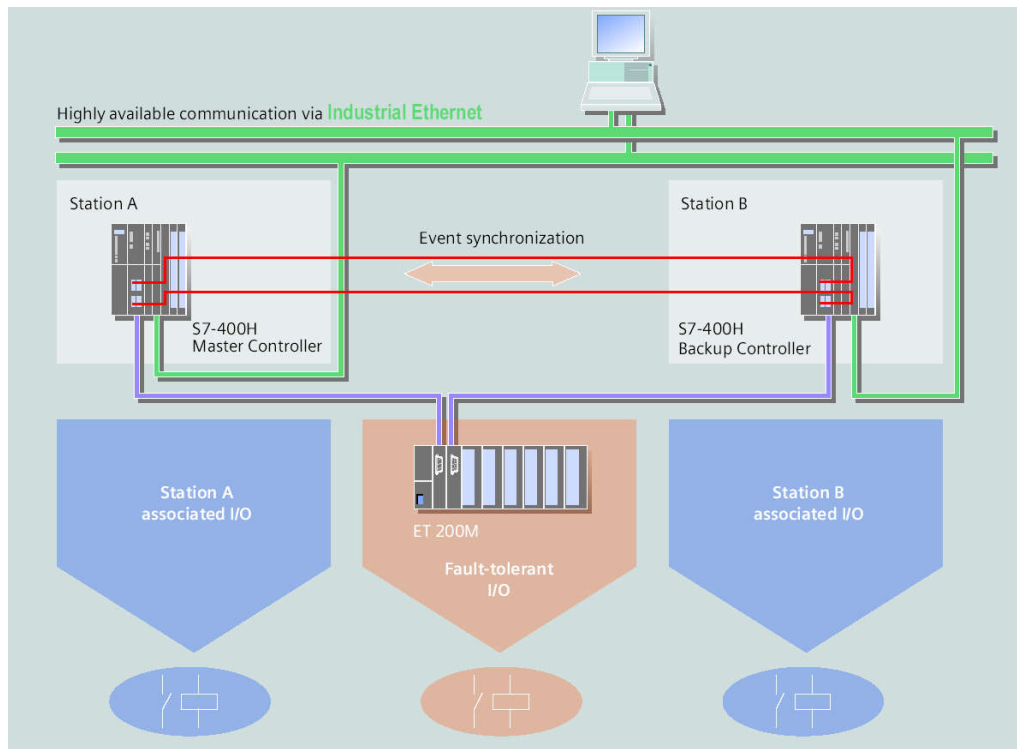
شکل ۳-۸- پروسه سنکرون سازی بین دو CPU



شکل ۳-۹- مفهوم سنکرون سازی بین دو CPU در سیستم H (Principle)



شکل ۳-۱۰- سیکل سنکرون سازی بین دو CPU در سیستم H



سنکرون سازی بین دو CPU به صورت Event Synchronization

۳-۴-۳- توابع خودآزمایی (self-test)

CPU های سری S7-400H دارای توابع خودآزمایی برای اجزای سخت افزاری (Scope) زیر هستند.

- CPU;
- Memory;
- Synchronization link or connection;

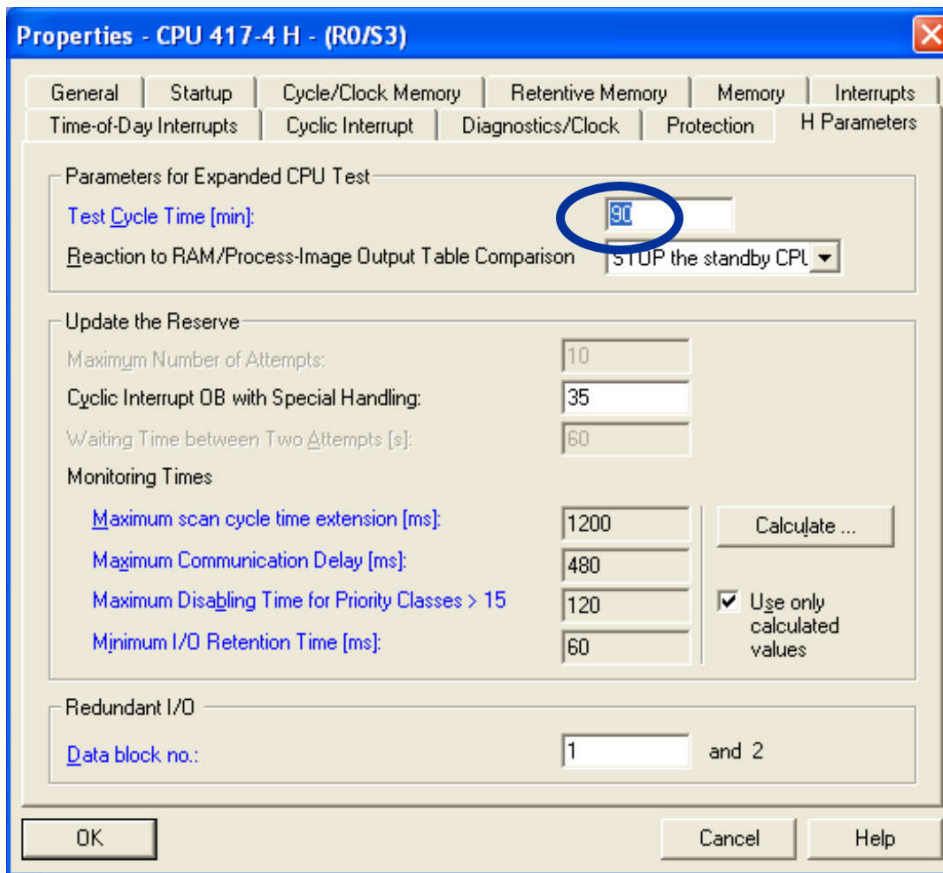
توابع آزمون به دو صورت اجرا می شوند (ساختار آزمون)

۱- خودآزمایی در زمان راه اندازی (Start-up self test)

- آزمون کاملی بوده و در زمان راه اندازی (Startup self-test) اجرا می شوند. زمانی که خطایی رخ می دهد، CPU به حالت defect می رود.

۲- خودآزمایی در زمان اجرای برنامه سیکلیک (Self test in cycle operation)

- این آزمون به صورت دائمی و به عنوان یک وظیفه (Task) در پس زمینه اجرا می شود.
- اجرای کل آزمون می تواند بین سیکل های مختلف توزیع شود و یا می تواند به طور کامل در یک دوره زمانی قابل تعریف در پارامترهای CPU (مثال ۹۰ دقیقه) اجرا شود.



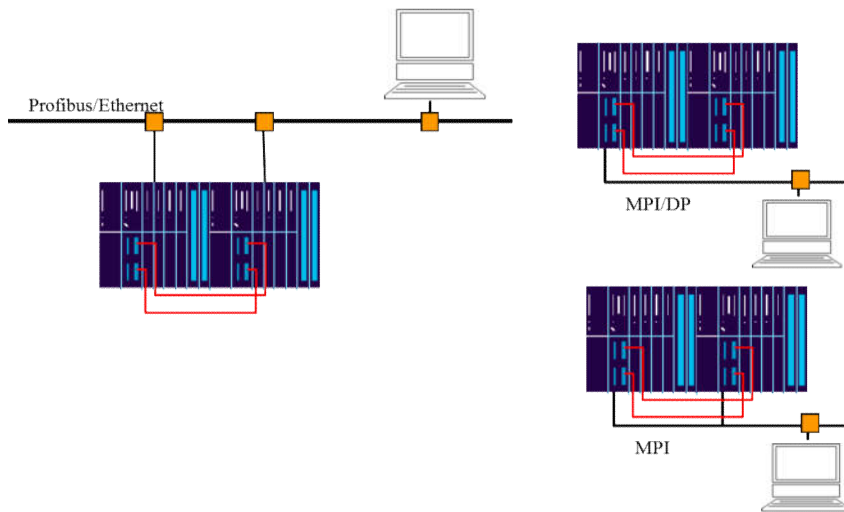
شکل ۳-۱۱- تنظیم دوره زمانی خودآزمایی در پارامترهای CPU

۳-۴-۴- برنامه نویسی آنلاین (Online programming)

همانند سیستم های استاندارد S7-400، قابلیت برنامه نویسی کامل آنلاین برای این سیستم ها نیز وجود دارد؛ که شامل تغییر برنامه، تخصیص پارامترها و تغییر پیکربندی سخت افزاری است. کلیه تغییرات و اصلاحات به صورت خودکار در هر دو CPU کپی می شود.

۳-۴-۵- اتصال به PG

برای اتصال سیستم H به کامپیوتر مهندسی (PG) و دانلود پیکربندی به آن می توان، از آداپتور MPI یا یکی از گذرگاه های شبکه اترنت و پروفیباس استفاده کرد.



شکل ۱۲-۳- اتصال سیستم H به کامپیوتر مهندسی (PG)

۳-۴-۶- قابلیت تعمیر آنلاین (Online Repair)

اجزایی از S7-400 که به صورت آنلاین قابل تعمیر (حذف و اضافه کردن ماژول) هستند، عبارت‌اند از:

- ۱- ماژول‌های I/O و CP در مد RUN قابل تعویض هستند.
- ۲- ماژول‌های IM (واقع در رک مرکزی) و CPU از طریق قطع تغذیه، قابل تعویض هستند (via Power off).

۳- پس از روشن کردن CPU تعویض شده، به صورت خودکار بروزآوری می‌شود

۴- ماژول‌های Sync

۵- ماژول‌های تغذیه (PS)

- ماژول‌های PS استاندارد از طریق قطع تغذیه (Power off)

- ماژول‌های PS افزونه در مد RUN؛

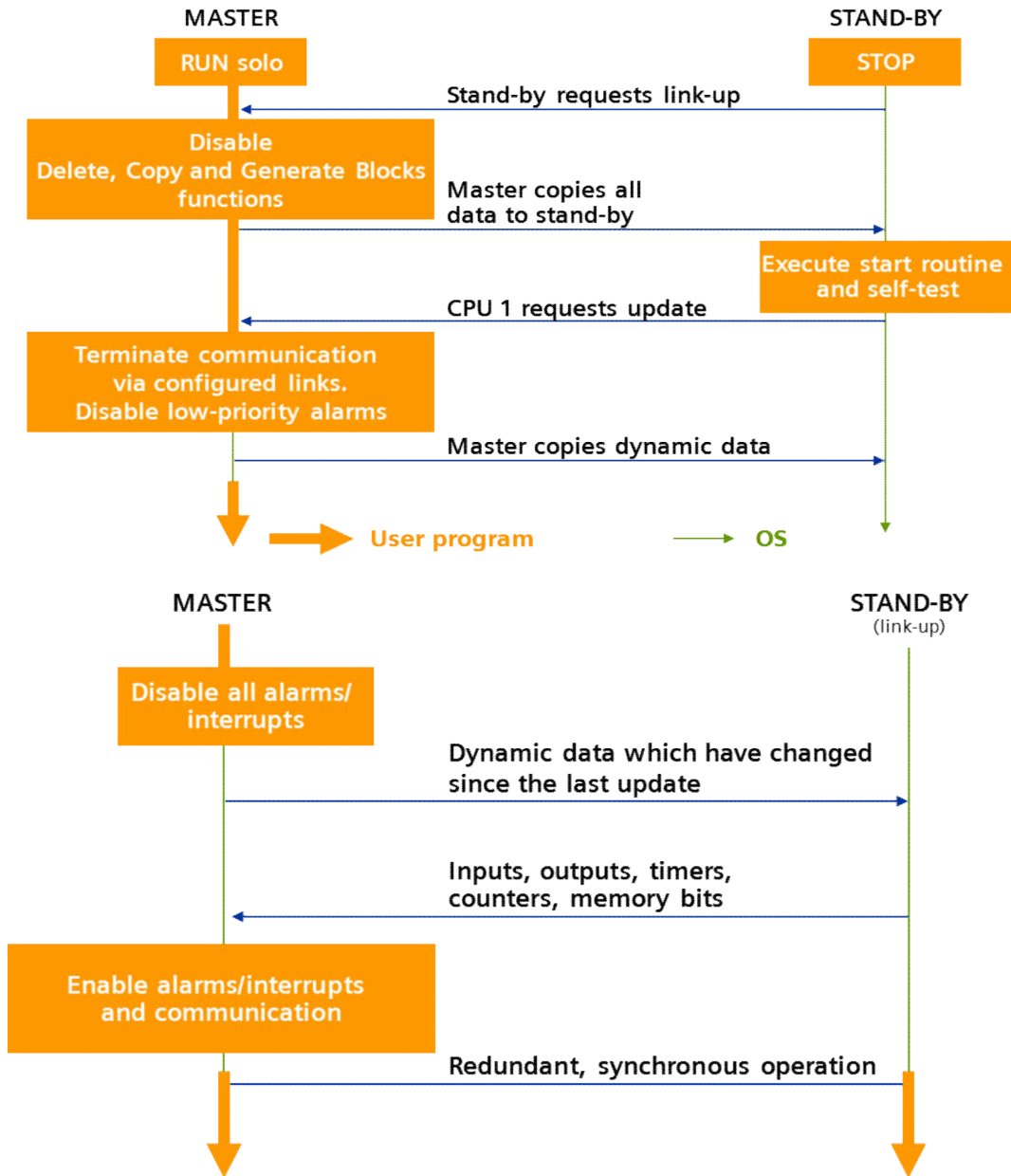
۶- اجزاء DP (DP Components)

- تعویض ماژول IM 153-2 در مد RUN؛

- تعویض ماژول‌های I/O در مد RUN؛

مطابق شکل ۱۳-۳، CPU‌های سری S7-400H پس از تعمیر سیستم Standby و قرار دادن ماژول در

رک افزونه برای برگردان آن به حالت افزونه و بروزآوری آن، پروسه بازگشت مجدد خودکاری را فراهم می‌کنند.



شکل ۳-۱۳- بازگشت خودکار سیستم Standby به پیکربندی افزونه پس از تعمیر

۳-۴-۷- قابلیت CiR (Configuration Changes in Run Mode)

برای پیکربندی سخت افزار یک سیستم H شامل کنترل کننده ها، ماژول های I/O و غیره از ابزار HWConfig استفاده می شود. در هنگام دانلود تنظیمات پیکربندی در محیط HW-Config به طور معمول CPUها به منظور اعمال تغییرات متوقف می شوند. لذا از آنجایی که توقف سیستم های کنترل دلچسب نبوده یا در برخی موارد به خاطر پیچیدگی سیستم، غیرممکن است، تغییرات جزئی یا اضافه کردن چند ماژول I/O باید بدون توقف سیستم های کنترل امکان پذیر باشد. برای این منظور سری S7400-H برای نصب یا برداشتن ماژول های

سخت افزاری در سیستم کنترل در زمانی که کنترل کننده در حال اجرا است از روش *CiR* یا تغییرات پیکربندی سیستم در حالت اجرا استفاده می کند. در صورتی که یک ایستگاه *I/O* در حالت *Runtime* به پیکربندی اضافه شود یا از قبل در یک ایستگاه *ET200M*، اسلاتی برای ماژول های *CiR* رزرو شده باشد. این ایستگاه ها در محیط *HWConfig* به رنگ نارنجی دیده می شوند. *CPU* های سری *S7400-H* قابلیت های پیکربندی آنلاین زیر را فراهم می کنند.

- پیکربندی حافظه *CPU* (*CPU memory configuration*)؛
- تغییر پارامترهای *CPU*؛
- حذف یا اضافه کردن ماژول های:
 - ماژول های *I/O or CP*؛
 - *DP slaves*
 - *PA interface and PA slaves*
 - *Y-link and slaves*
 - ماژول های *Sync*؛
 - *Modules in modular DP slaves*
 - *Redundant IM 153-2*
 - *Redundant Power Supplies*

اجزای افزونه که می توانند به صورت تغذیه خاموش، جایگذاری شوند عبارت اند از:

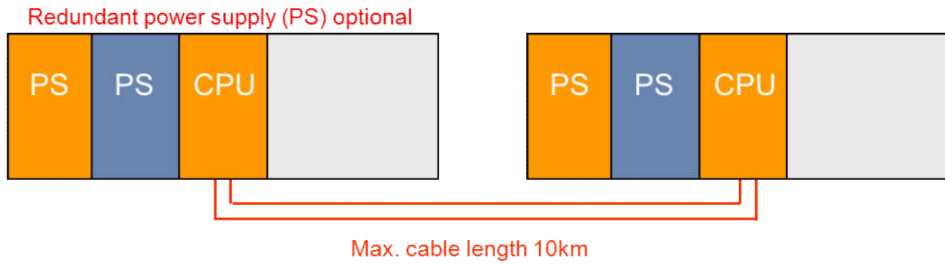
- *Standard power supplies*;
- *Central IM*;
- *CPU*;

ماژول *CPU* به محض جایگذاری در رک بروز آوری (*Program and Data*) می شود.

۳-۴-۸- پیکربندی کنترل کننده (Configuration)

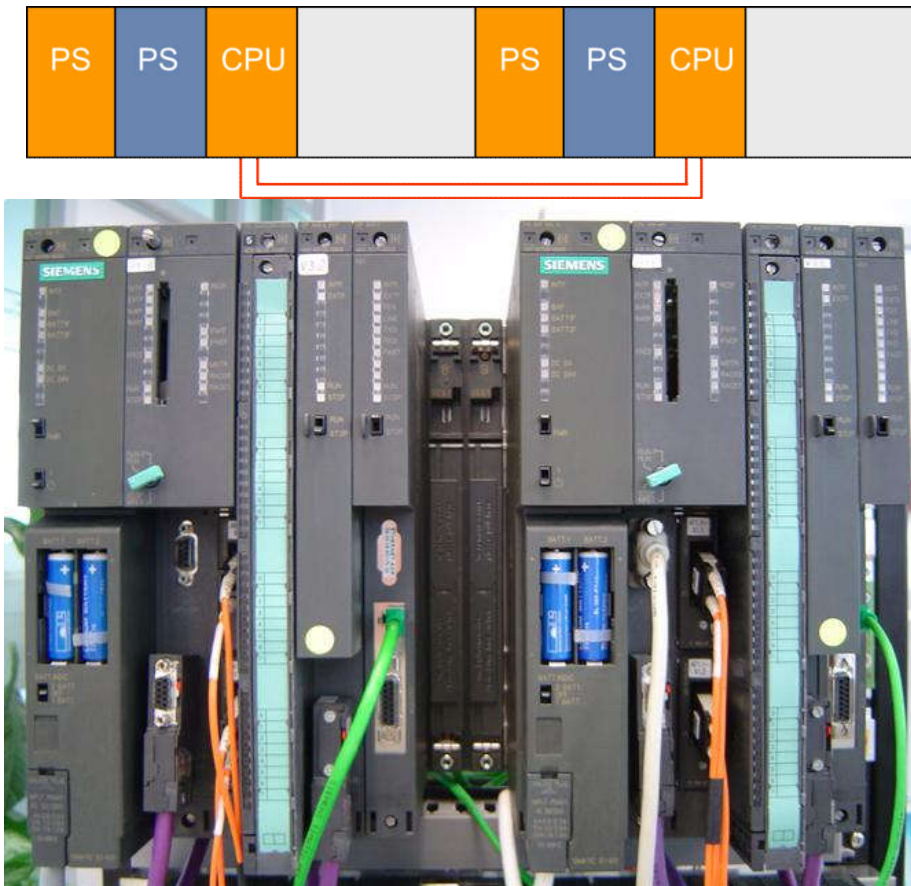
کنترل کننده های مرکزی را می توان به دو روش مختلف پیکربندی کرد:

۱- اگر قرار است که دو زیرسیستم افزونه به دلایل در دسترس بودن به طور کامل از هم جدا شوند، مناسب است که از دو رک استاندارد (*UR1* و *UR2*) استفاده شود. در این روش در هر یک از رک ها یک *CPU* و یک منبع تغذیه (*PS*) قرار می گیرد. همچنین اگر میزان در دسترس پذیری بالایی لازم باشد، دو ماژول منبع تغذیه می تواند به صورت افزونه پیکربندی شود. در این آرایش فاصله بین دو رک می تواند تا ۱۰ کیلومتر باشد.



شکل ۳-۱۴- پیکربندی CPU های H با دو رک استاندارد به فاصله دور از همدیگر

۲- در پیکربندی دیگر، دو CPU هر یک با یک منبع تغذیه تکی یا افزونه، به رک نوع UR2-H با یک گذرگاه Backplane جدا متصل می شود.



شکل ۳-۱۵- نمونه پیکربندی سیستم H بر روی یک backplane با رک UR2H

مشخصات برخی از ماژول های CPU به شرح زیر می باشد:

- میزان حافظه:
- 417-4H from 4 MB to 20MB;
- 414-4H from 768KB to 1,4MB;
- فراهم کردن قابلیت اطمینان بالا (Higher Reliability)

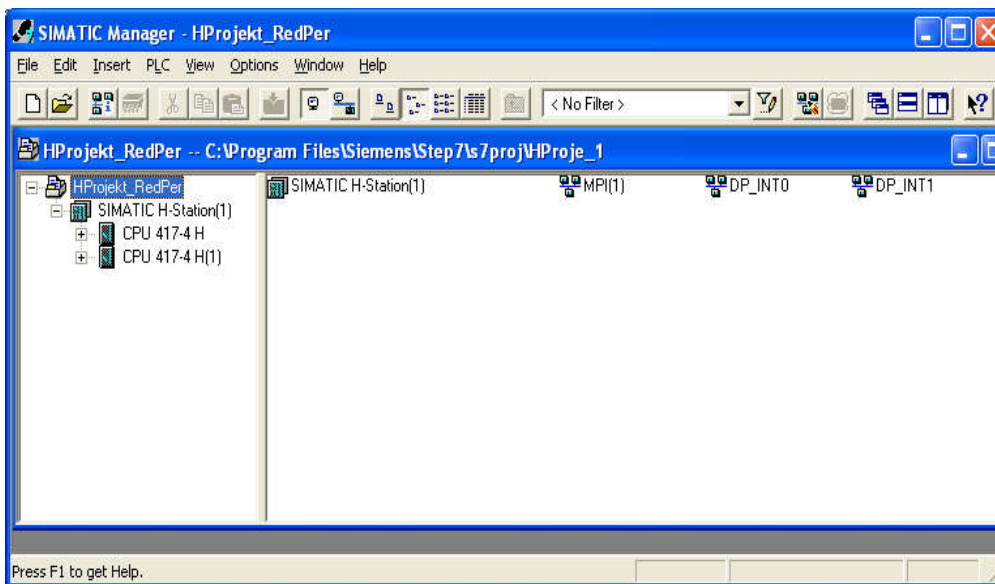
حافظه CPU دارای فناوری EDC (*Error Detection and Correction*) برای تشخیص و تصحیح خودکار خطا است.

- پشتیبانی از فاصله بین کنترل کننده ها تا ۱۰ کیلومتر (در CPU های قبلی تا ۵۰۰ متر پشتیبانی می شد)
- فاصله بین کنترل کننده های H بسته به نوع رک و کابل فیبر نوری دارای پیکربندی های زیر است:
 - FO-Cable 1m;
 - FO-Cable 2m;
 - FO-Cable 10m;
 - Monomode FO-Cable LC/LC Duplex crossed 9/125 μ : 10km;
- با دو رک استاندارد جداگانه (10km)
- با یک رک نوع H و گذرگاه پشتی جداگانه (with split backplane bus) – (1m, 2m, 10m)
- دارای ۴ واسط (Integrated interfaces):
 - دو واسط ماژول های Sync؛
 - یک واسط DP؛
 - یک واسط MPI/DP؛

۳-۵- پیاده سازی سیستم 400H (System integration)

۳-۵-۱- برنامه نویسی

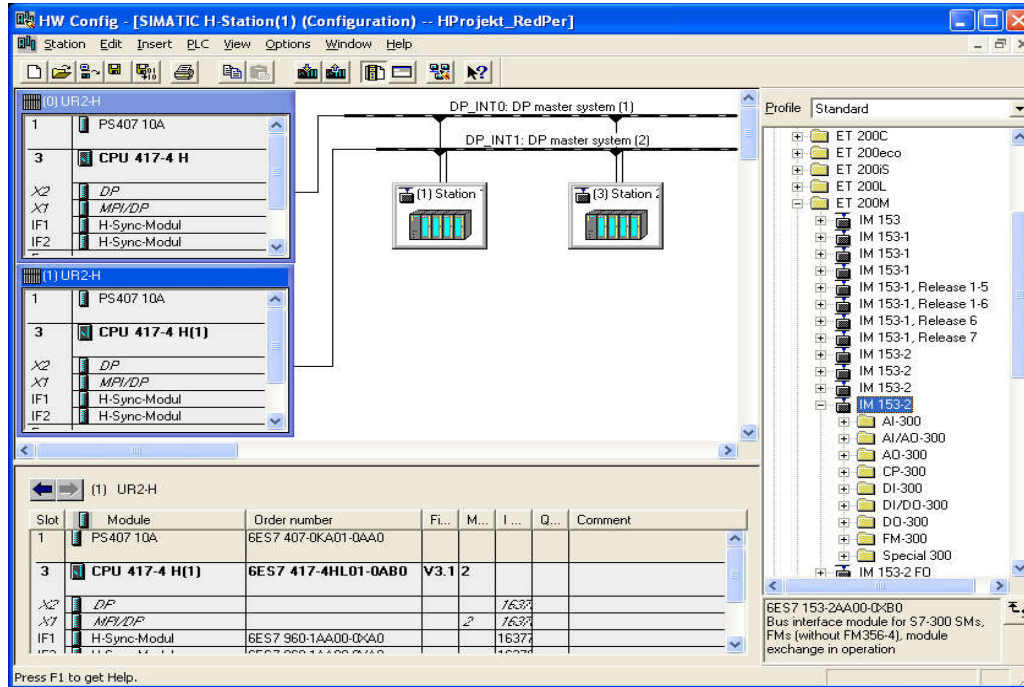
برای برنامه نویسی الگوریتم کنترل سیستم H از همان ابزار سیستم استاندارد یعنی *Simatic Manager* به همراه کتابخانه های مختلف استفاده می شود. به طوری که زیرمجموعه آن تحت عنوان *Transparent Programming* یاد می کند. لذا از کلیه زبان های برنامه نویسی بکار رفته در سیستم استاندارد بدون هیچ محدودیتی می توان استفاده کرد.



شکل ۱۶-۳- نمایی از پروژه S7-400H در محیط Simatic Manager

۳-۵-۲- پیکربندی سخت افزاری (Hardware configuration)

برای پیکربندی سیستم H از همان ابزار HWConfig بکار رفته برای سیستم استاندارد استفاده می کند. سخت افزار یک سیستم H پلات فرم پایه برای سیستم های F و FH می باشد.



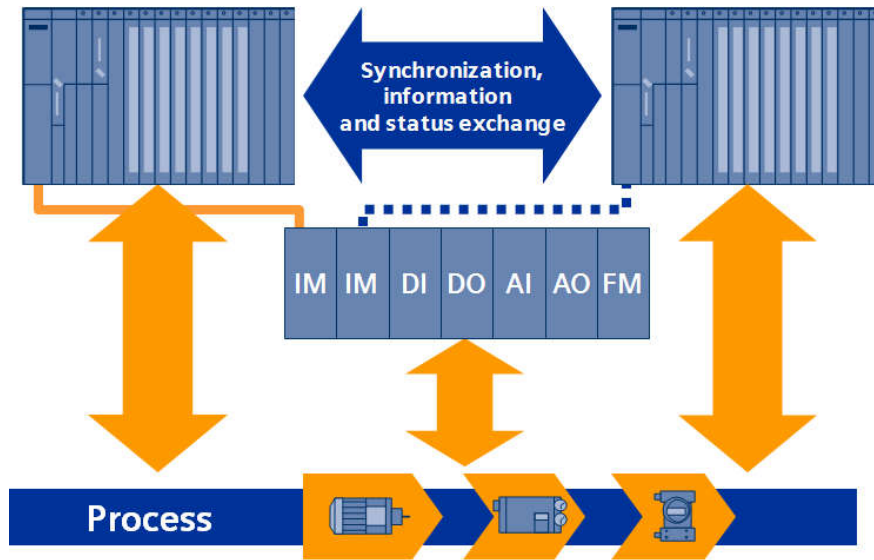
شکل ۱۷-۳- نمایی از پیکربندی سیستم H در محیط HwConfig

۳-۶- افزونگی (Redundancy principle)

افزونگی سیستم های کنترل اتوماسیون برای دو هدف کلی زیر صورت می گیرد.

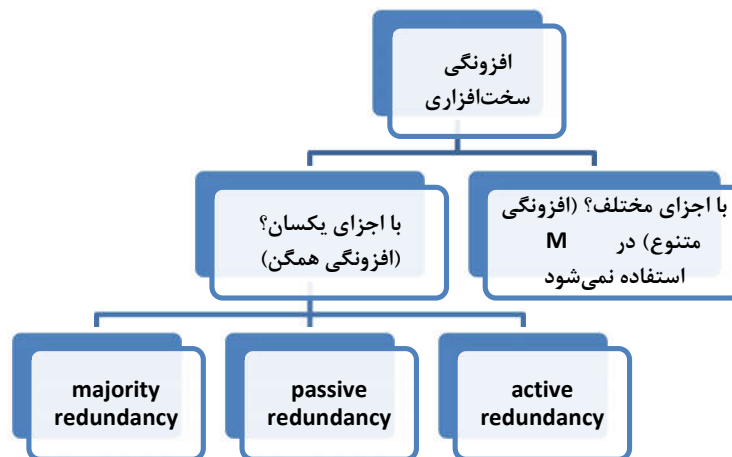
- داشتن یک سیستم کنترل با مشخصه تحمل خطا (Fault-tolerant) برای دسترس پذیری بالا (High Availability) است. به طوری که هدف نهایی، کاهش احتمال تلفات تولید از طریق سوئیچ به یک سیستم آماده به کار می باشد.
 - داشتن یک سیستم کنترل با مشخصه ایمن در نقص در سیستم (Fail-safe) با هدف حفاظت از زندگی افراد، محیط زیست و سرمایه ها از طریق قطع امن تجهیزات
- در سیستم های کنترل دو نوع افزونگی در سطح کنترل کننده وجود دارد که عبارت انداز:
- افزونگی سخت افزاری - برای فرآیندهای سریع- زمان سوئیچ از سیستم اصلی به رزرو در حد میلی ثانیه می باشد.

- افزونگی نرم افزاری - برای فرایندهای کند- زمان سوئیچ از سیستم اصلی به رزرو در حد ثانیه می باشد.

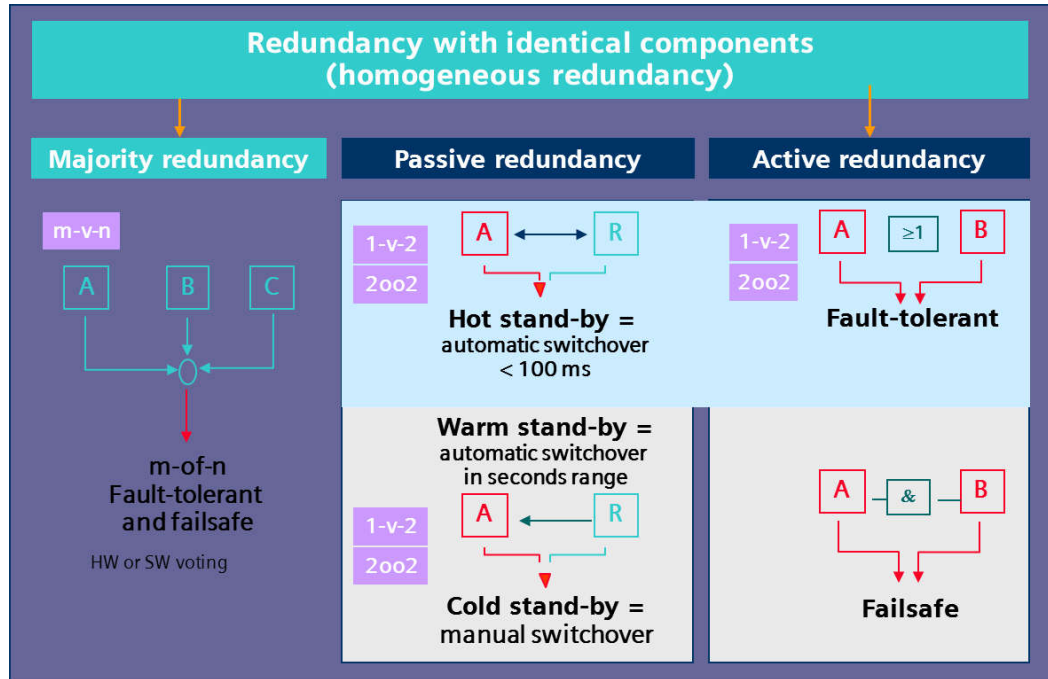


شکل ۱۸-۳- افزونگی در سیستم S7-400H

۳-۶-۱- افزونگی سخت افزاری (Hardware Redundancy)



۳-۶-۲- افزونگی با اجزای یکسان (homogeneous redundancy)



انواع افزونگی سخت افزاری با اجزای یکسان

۳-۶-۳- افزونگی پسیو (Passive redundancy)

به سه صورت پیاده سازی می شود. که عبارت انداز:

Passive Redundancy (1-v-2 , 2oo2: 2 out of 2)

- o Hot Standby : automatic switchover < 100 ms
- o Warm Standby : automatic switchover in seconds range
- o Cold Standby : manual switchover

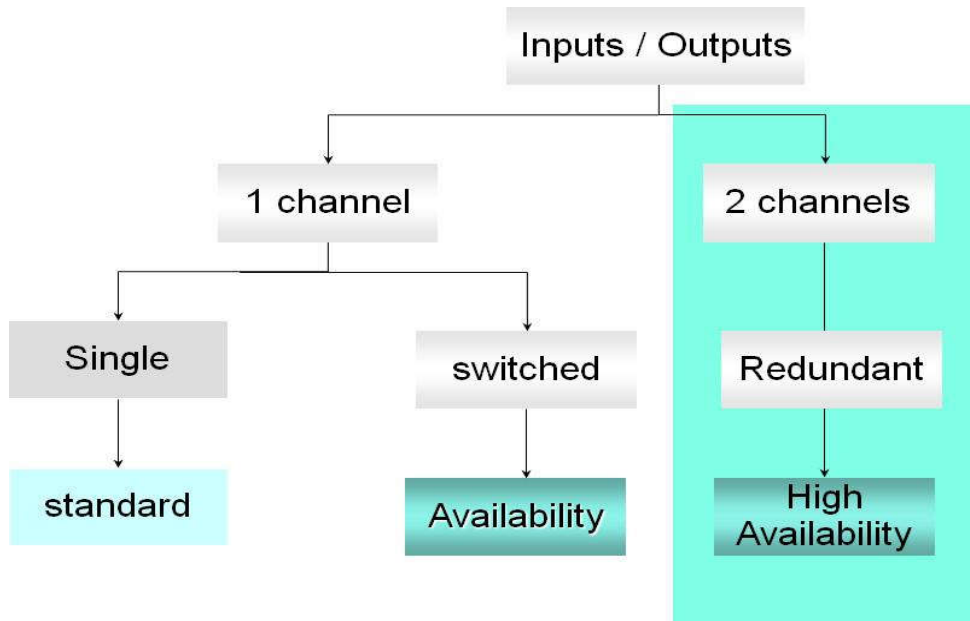
۳-۶-۴- افزونگی فعال (Active redundancy)

برای سیستم های تحمل پذیر خطا تنها یکی از CPUها نتایج پردازش را در خروجی می نویسند، در واقع عمل OR بین CPUها صورت می گیرد. ولی در مقابل برای سیستم های ایمن در خطا (Failsafe)، نتیجه پردازش توسط هر دو CPU در خروجی نوشته می شود.

۳-۷- پیکربندی و اتصال I/O در سیستم H (I/O configuration)

مطابق شکل ۳-۱۹ اتصال ماژول های ورودی/خروجی در سیستم H بسته به میزان «دسترس پذیری» مورد نیاز به چند روش می تواند، انجام شود. در این روش ها ماژول های I/O می توانند به صورت مرکزی در کنار خود CPU نصب شوند و یا بر روی رک RIO در شبکه Profibus قرار بگیرند. ■ استفاده از یک کانال I/O واحد - برای دسترسی پذیری استاندارد،

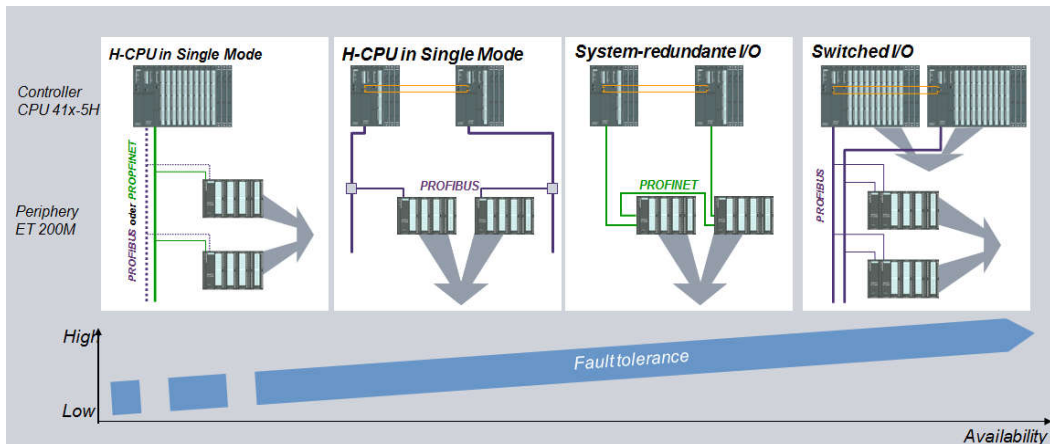
- استفاده از یک کانال I/O واحد با دو ماژول واسط IM افزونه که بین دو CPU سویچ می شود (پیکربندی Switched I/O) برای دسترس پذیری بالا؛
- استفاده از دو کانال I/O افزونه:



شکل ۳-۱۹- میزان حصول دسترس پذیری I/O در سیستم های S7-400H

ساختارهای افزونه ممکن برای ماژول های ورودی/خروجی I/O عبارت انداز:

- H-CPU in single mode with Redundant I/O Modules
- Central I/O Modules
- Distributed I/O Modules
- Distributed Switched I/O Modules
- Full Redundant I/O



شکل ۳-۲۰- میزان دسترس پذیری با پیکربندی های مختلف سیستم H با ET200M

۳-۷-۱- اصول I/O افزونه

I/O های افزونه ماژول های ورودی و خروجی هستند که به صورت افزونه پیکربندی و کار می کنند. در این حالت حداکثر میزان در دسترس پذیری حاصل می شود بدلیل این که در این روش نقص در CPU، خرابی در خط PROFIBUS و یا حلقه PROFINET و ماژول سیگنال جبران می شود.

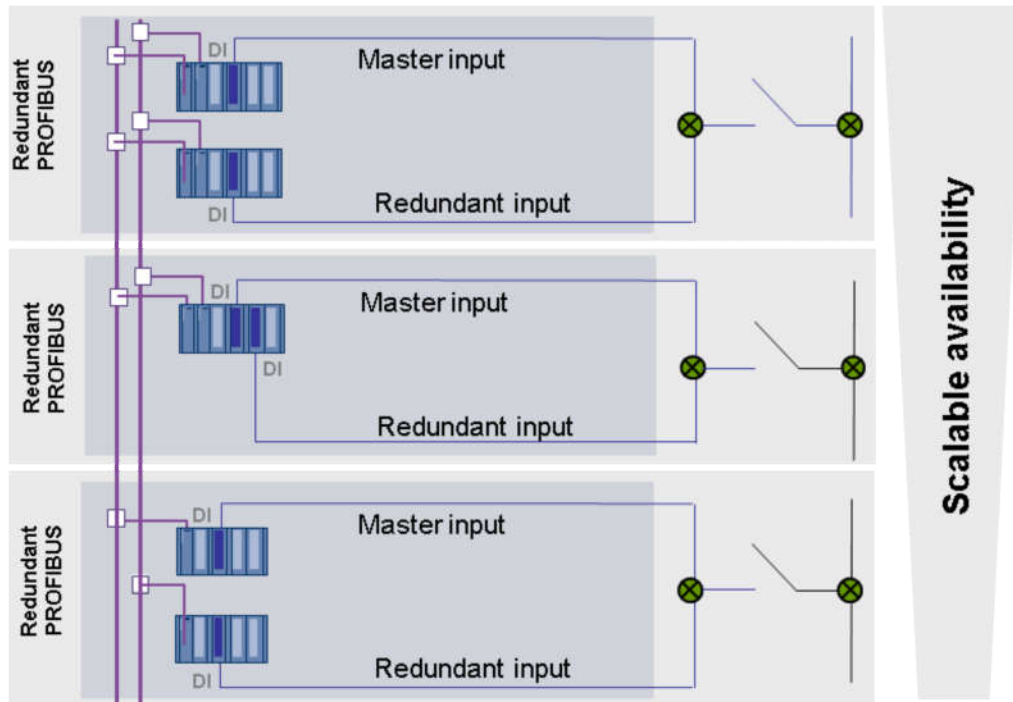
در طول عملکرد عادی، هر دو ماژول فعال هستند. در ماژول های ورودی افزونه، مقدار سنسور به طور مشترک توسط دو ماژول قرائت شده، نتیجه قرائت دو ورودی مقایسه شده و به عنوان یک مقدار واحد برای پردازش بیشتر در در اختیار برنامه لاجیک قرار می گیرد.

در مورد خروجی های افزونه نیز مقدار محاسبه شده توسط برنامه کاربر به خروجی های هر دو ماژول ارسال می شود. ولی یکی از ماژول ها در وسیله خروجی می نویسد.

در صورت وقوع خطا، به عنوان مثال خراب شدن یکی از دو ماژول ورودی، ماژول معیوب، دیگر آدرس دهی نمی شود. وقوع خطا اطلاع داده شده و پردازش تنها با ماژول بی عیب همچنان ادامه می یابد. بعد از تعمیر که ممکن است به صورت آنلاین انجام گیرد، هر دو ماژول می توان دوباره آدرس دهی شود.

۳-۷-۲- افزونگی با I/O های مبتنی بر شبکه پروفی باس (FMR—Flexible Modular Redundancy)

- Scalable fault tolerance
 - Single or redundant connection
 - Can be mixed in one and same system
- Advantage
 - Redundancy has to only be made available where it is actually needed .
 - More cost-efficient solutions than conventional architectures with single design.



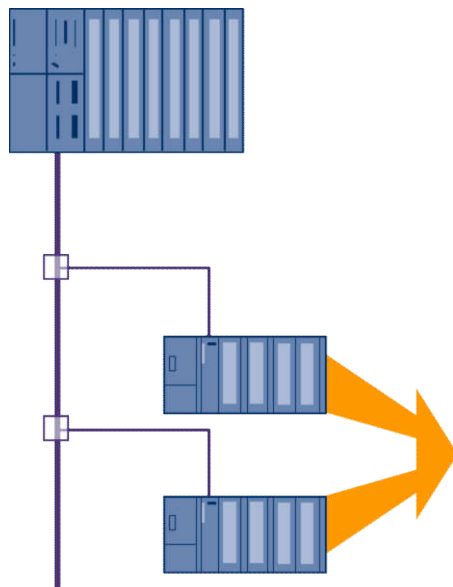
شکل ۳-۲۱-

۳-۷-۳- پیکربندی استفاده از یک کانال I/O واحد بدون ساختار افزونه (Single Mode)

در این روش کنترل کننده S7-400H در یک ساختار افزونه پیکربندی نشده است. به طوری که هر سنسور به یک کانال تنها از یک کارت I/O متصل می گردد.

۳-۷-۴- استفاده از ماژول های I/O افزونه با یک سیستم H بدون افزونه (H-CPU in single mode)

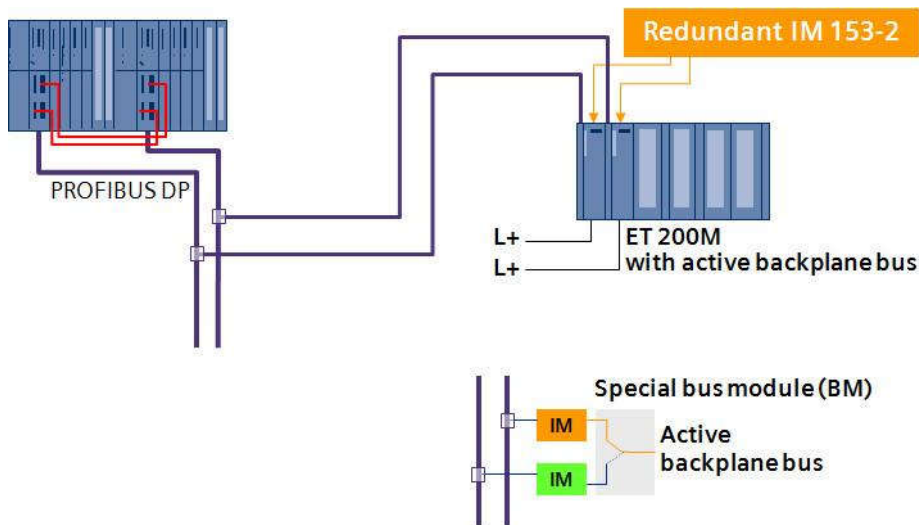
در این پیکربندی، هر سنسور به دو کانال I/O که در دو رک RIO جداگانه قرار گرفته است، متصل می شود. به طوری که کارت های I/O توسط یک H-CPU در یک ساختار غیر افزونه مورد دسترسی قرار می گیرد.



شکل ۳-۲۲- استفاده از دو کانال I/O افزونه در دو رک RIO جداگانه با یک سیستم H بدون افزونه

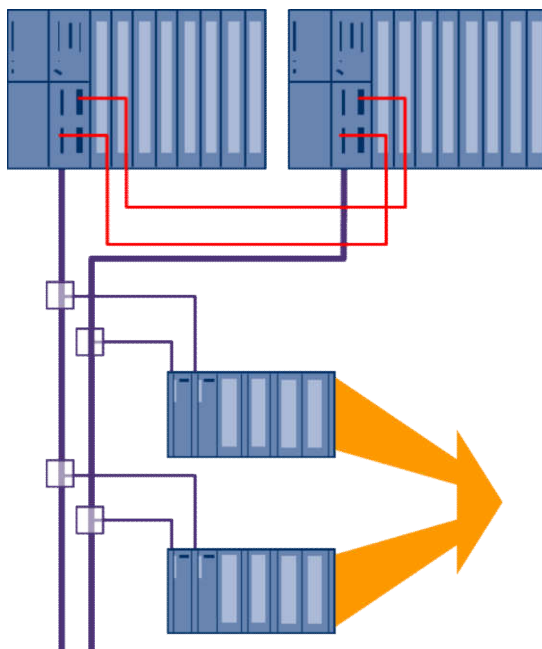
۳-۷-۵- استفاده از ماژول های I/O مشترک بین دو CPU (Distributed Switched I/O)

در این روش مطابق شکل ۳-۲۳ یک کارت I/O بین دو CPU مشترک بوده و توسط دو ماژول IM و از طریق یک Active backplane bus سویچ می شود. در «ساختار افزونه ماژول های I/O قابل سویچ بین دو سیستم H افزونه» یک کانال از ماژول های I/O، در رک های RIO جداگانه به صورت افزونه به یک سنسور متصل شده و بین دو سیستم H افزونه سویچ می شود.



شکل ۳-۲۳- ساختار Switched I/O

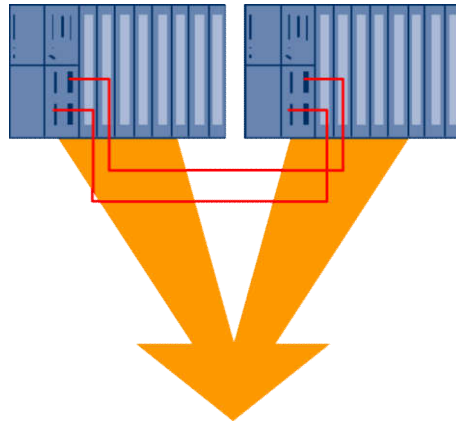
باس ماژول استفاده برای جاگذاری ماژول های IM دارای شماره سفارش 7HD10 می باشد.



شکل ۳-۲۴- اتصال دو کارت I/O در دو رک RIO جداگانه در یک ساختار افزونه سیستم H

۳-۷-۶- استفاده از ماژول های I/O افزونه در رک مرکزی (Central I/O Modules) دو سیستم H افزونه

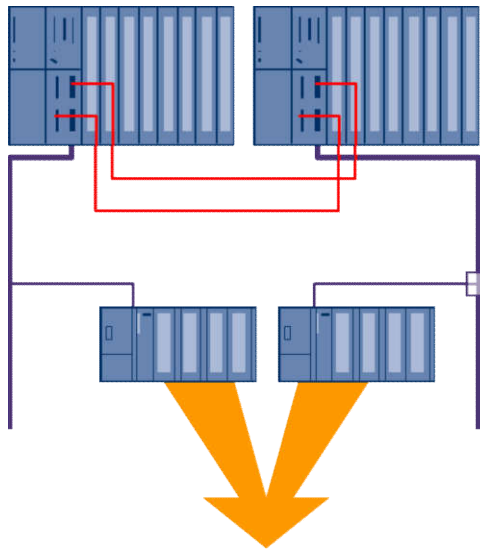
در این پیکربندی دو ماژول I/O قرار گرفته در رک مرکزی به صورت ساختار افزونه نصب می شوند.



شکل ۲۵-۳- ساختار افزونه ماژول های I/O در رک مرکزی دو سیستم H افزونه

۳-۷-۷- ساختار I/O افزونه توزیع شده (Distributed I/O modules)

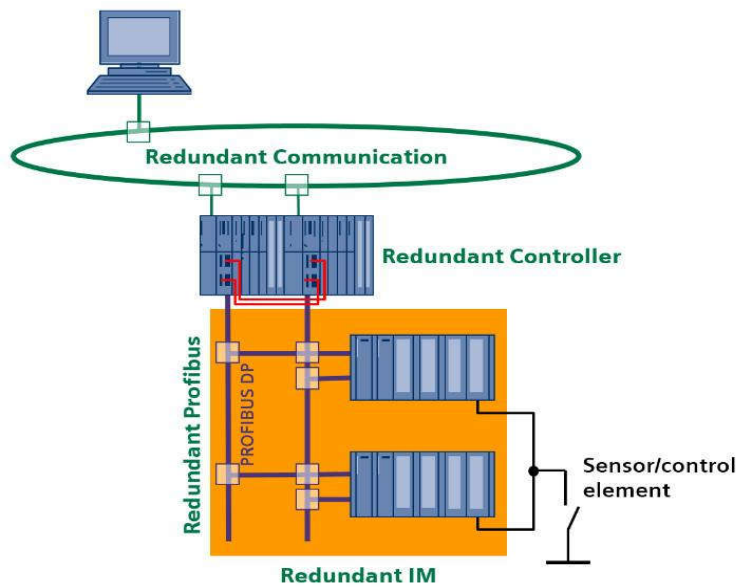
دو ماژول I/O در رک های جداگانه RIO (توزیع شده) به صورت افزونه باهم، نصب می شوند.



شکل ۲۶-۳- ساختار افزونه I/O در رک های جداگانه RIO در یک سیستم H افزونه

۳-۷-۸- ساختار افزونه کامل برای I/O برای دسترس پذیری خیلی بالا

در این روش که افزونگی کاملی را فراهم می کند. اجزای I/O نظیر کارت I/O، کارت IM و کابل ارتباطی دارای ساختار افزونه دوتایی است. شکل ۲۷-۳ ساختار افزونگی کارت های I/O و گذرگاه I/O را به همراه افزونگی شبکه و کنترل کننده نشان می دهد.



شکل ۳-۲۷- ساختار افزونگی کارت‌ها و گذرگاه I/O در PCS7 برای دسترسی بالا

توصیف اجزای ایستگاه RIO (ET200M) به شرح زیر می باشد.

۱- استفاده از یک رک آلومینیومی با گودی بیشتر از یک رک معمولی

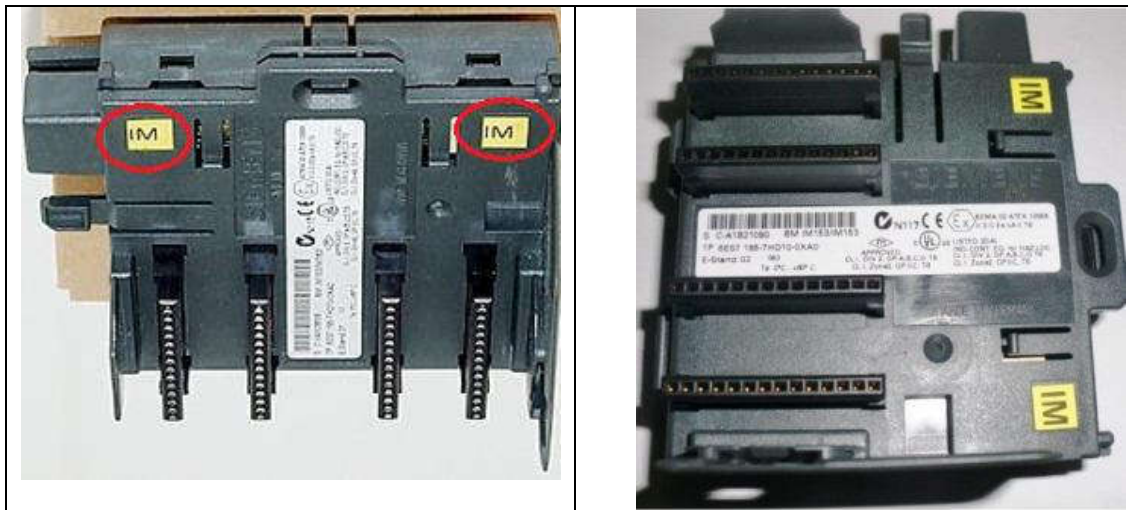
6ES7 195-1GF30-0XA0	SIMATIC DP, RAIL FOR ET 200M 530 MM LONG F. MAX. 5 BUS MOD. FUNCTION: INSERT/REMOVE
---------------------	---



شکل ۳-۲۸- رک برای نصب و مونتاژ ایستگاه Switched I/O

۲- استفاده از قطعات Bus Unit مخصوص نصب ماژول‌های IM153-2 با مشخصات زیر:

6ES7 195-7HD10-0XA0	SIMATIC DP, BUS Unit for ET200M for Integrating 2 IM153-2 RED. function: Insert/Remove While Operating Mode Run
---------------------	---

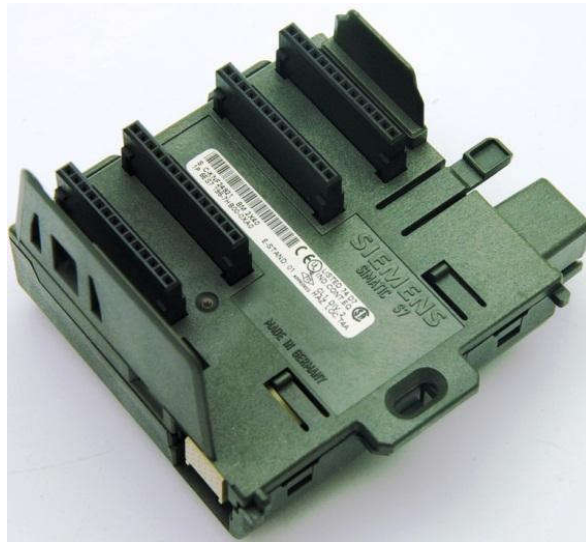


شکل ۲۹-۳ - Bus Unit برای نصب ماژول های IM153-2

۳- استفاده از قطعات Bus Unit مخصوص نصب ماژول های I/O با مشخصات زیر:

6ES7195-7HB00-0XA0	SIMATIC DP, BUS UNIT FOR ET200M F. THE INTEGR. OF TWO 40 MM WIDE I/O SUBMODULES FOR INSERT/REMOVE
--------------------	---





شکل ۳-۳۰- Bus Unit برای نصب ماژول های I/O

۴- ماژول های واسط IM 153-2

6ES7 153-2BA00-0XB0	SIMATIC DP, ET 200M INTERFACE IM 153-2 HIGH FEATURE FOR MAX. 8 S7-300 MODULES, WITH REDUNDANCY, TIME STAMPING FIT FOR ISOCHRONOUS MODE
---------------------	--

در این مد هر دو DP Master فعال بوده و به درستی عمل می کنند؛ و عملیات I/O به شرح زیر صورت می گیرد:

- قرائت ورودی (Reading inputs): ورودی ها تنها از کانال IM فعال خوانده می شوند
- نوشتن خروجی ها (Writing outputs): داده توسط هر دو کانال دریافت می شود ولی تنها از کانال IM فعال به خروجی ها ارسال می شود.

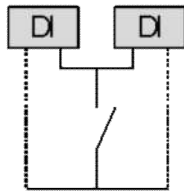
۳-۸- سیستم بندی کانال های افزونه (Wiring)

۳-۸-۱- سیستم بندی کانال های ورودی دیجیتال افزونه (Wiring)

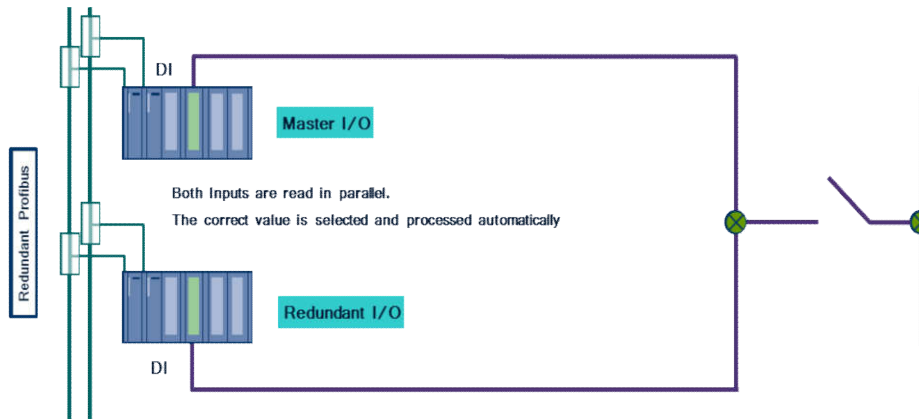
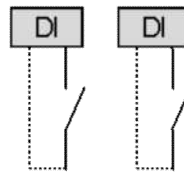
مطابق نحوه سیم بندی کانال های ورودی افزونه دیجیتال به دو صورت می باشد

- اتصال کانال های افزونه به یک سنسور
- اتصال کانال های افزونه به دو عدد سنسور

With one sensor



With two sensors

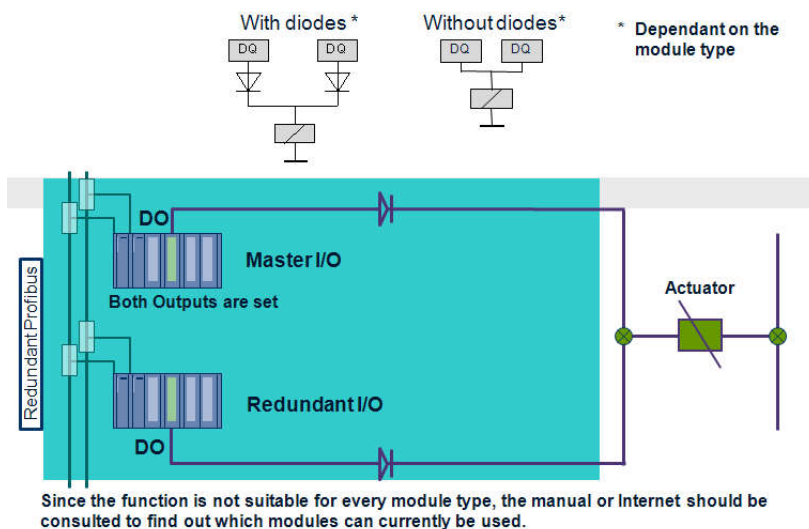


Since the function is not suitable for all module types, the manual or Internet should be consulted to find out which modules can currently be used.

شکل ۳-۳۱- سیستم بندی کانال های ورودی دیجیتال افزونه

مطابق شکل بالا هر دو ورودی قرائت می شود. مقدار درست انتخاب و پردازش می شود. نکته: از آنجایی که ارایش نشان داده شده در شکل بالا برای همه ماژول های ورودی مناسب نمی باشد. لذا برای هر نوع ماژول I/O بایستی به فایل راهنمای مربوطه مراجعه شود.

۳-۸-۲- سیستم بندی کانال های خروجی دیجیتال افزونه



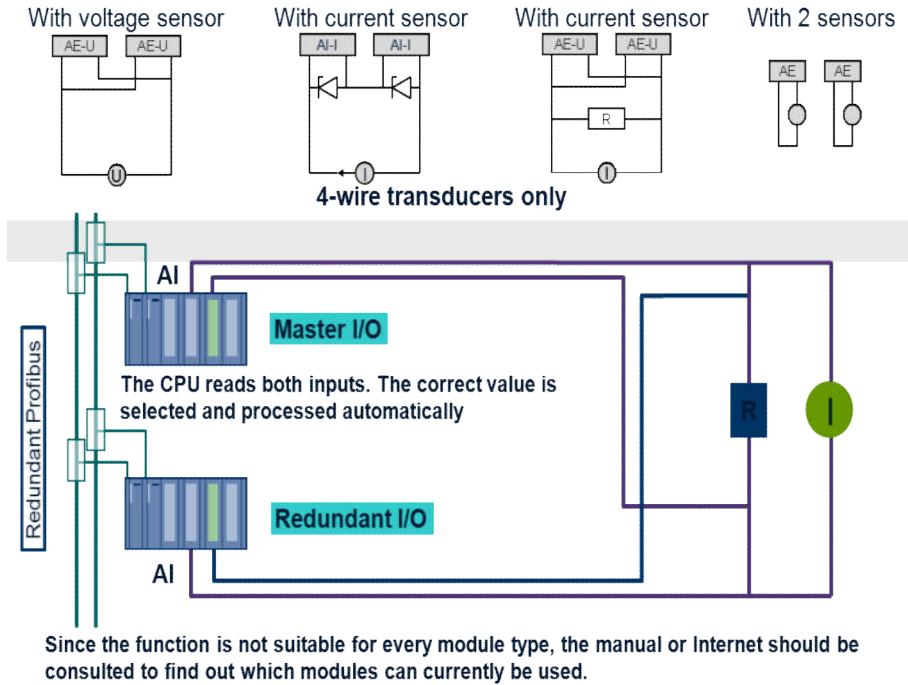
Since the function is not suitable for every module type, the manual or Internet should be consulted to find out which modules can currently be used.

شکل ۳-۳۲- سیستم بندی خروجی های دیجیتال افزونه

نکته: از آنجایی که ارایش نشان داده شده در شکل بالا برای همه ماژول های ورودی مناسب نمی باشد. لذا برای هر نوع ماژول I/O بایستی به فایل راهنمای مربوطه مراجعه شود.

۳-۸-۳- سیم بندی کانال های ورودی آنالوگ افزونه

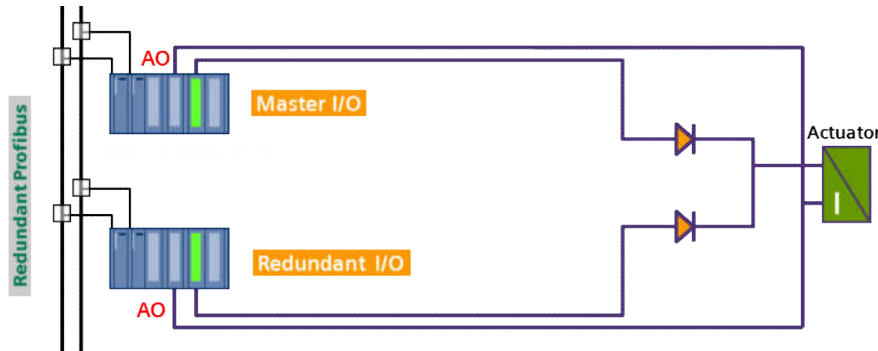
در کانال های آنالوگ بسته به نوع سنسور که ولتاژی باشد یا جریانی و یا از یک سنسور یا دو عدد سنسور استفاده شده باشد، سیم بندی متفاوت می باشد.



شکل ۳-۳-۳- سیم بندی ورودی های آنالوگ افزونه

۳-۸-۴- سیم بندی کانال های خروجی های آنالوگ افزونه

هر کانال خروجی، نیمی از مقدار خروجی را فراهم می کند. هنگامی که یکی از ماژول ها نقص پیدا می کند، کانال خروجی که سالم است مقدار کامل خروجی را فراهم می کند.



شکل ۳-۴-۳- سیم بندی خروجی های آنالوگ افزونه

۳-۸-۵ - سطوح کیفیت در افزونگی I/O (Redundant quality stages)

بسته به این که زیر سیستم I/O به چه صورت پیکربندی شود، سه سطح از کیفیت برای افزونگی I/O فراهم می شود که عبارت انداز:

- سطح کیفیت بالا (Highest quality level) در صورتی که در پیکربندی I/O یک سیستم H، از ماژول های F-I/O، (Failsafe I/O) با توابع تشخیصی (Diagnostic Functions) استفاده شود، سطح کیفیت بالا برای افزونگی سیستم تعریف می شود. این سطح کیفیت، برای عملیات ایمن (failsafe) در یک پلنت مورد نیاز می باشد.
- سطح کیفیت متوسط (Medium quality level) در این سطح کیفیت، از ماژول های استاندارد (نه از نوع Fail-safe) با عملکرد تشخیصی (Diagnostic Functions) استفاده می شود.
- سطح کیفیت کم هزینه (Low-cost quality level) از ماژول های استاندارد بدون عملکرد تشخیصی استفاده کنید.

۳-۹-۹ - یکپارچگی با برنامه کاربر

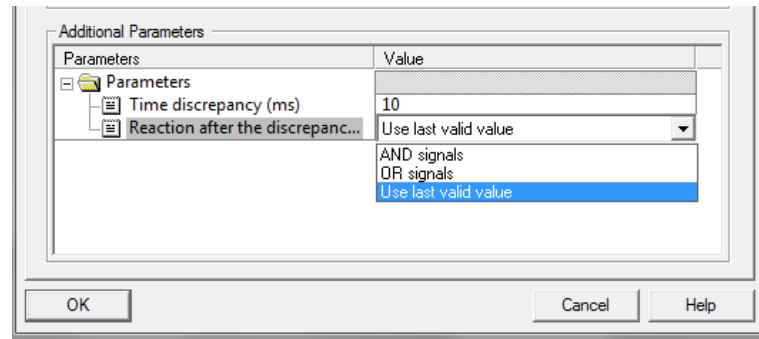
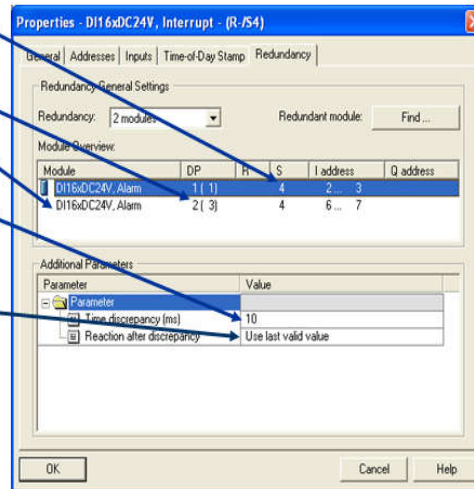
برای پیاده سازی برنامه کاربر یا لاجیک I/O های افزونه، در برنامه Step 7 از کتابخانه ی افزونگی که شامل فانکشن بلاک های مختلف می باشد، استفاده می شود. قوانین موجود بیان می کند که در سیستم H با I/O افزونه، همیشه پایین ترین آدرس باید برای برنامه نویسی استفاده شود. روش کار (Method of operation) به شرح زیر می باشد:

- ورودی توسط بلاک FB RED_IN خوانده شده و پس از آنالیز وجود اختلاف بین دو ورودی (discrepancy) مقدار خوانده شده به POI کپی می شود.
- در برنامه لاجیک کاربر نیز، یک خروجی تولید شده در برنامه، توسط بلاک FB RED_OUT به پایین ترین آدرس یک کانال خروجی به روش معمول نوشته می شود. سپس بلاک FB RED_OUT به طور خودکار مقدار مربوط را به آدرس دوم کپی می کند.

۳-۹-۱ - پیکربندی I/O افزونه

شکل زیر پنجره پراپرتی یک ماژول I/O پیکربندی شده به صورت افزونه را در محیط HW Config نشان می دهد.

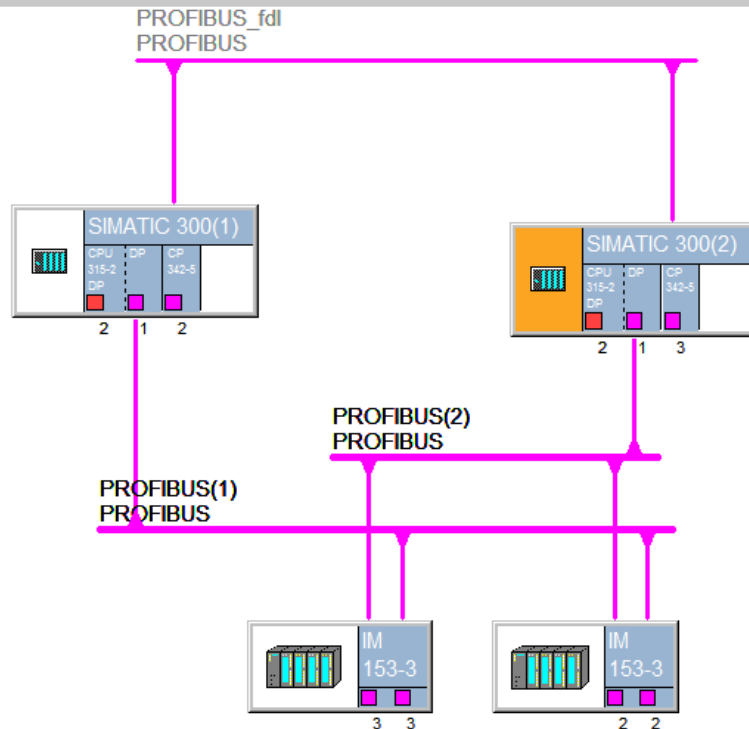
- Slot
- DP address
- Redundant DI
- Time discrepancy in ms
- Response time following discrepancy
- Possible options:
 - AND gate
 - OR gate
 - Use last valid value



شکل ۳-۳-۳- پیکربندی سخت افزاری I/O افزونه

۳-۹-۲- نحوه Redundant کردن یک ET200M (IM153-3) بین دو Remote I/O station در software Redundancy

۱. دو تا Station یکسان ایجاد می کنیم.
۲. در هر دو Station پورت DP مربوط به CPU را به DP Master تنظیم می کنیم.
۳. در Station A یک ET200 M (IM 153) قرار می دهیم. سپس ماژول های IO را در این ET جایگذاری می کنیم.
۴. در Station A از ET200 M (IM 153) کپی می گیریم و پنجره HWConfig مربوطه را می بندیم.
۵. پنجره HWConfig مربوط به Station B را باز می کنیم.
۶. باس DP Master مربوط به CPU Station B را انتخاب و از منوی EDIT گزینه Insert Redundant را اجرا می کنیم.
۷. پس از کامپایل پروژه ، پیکربندی کلی به شکل زیر خواهد بود.



در این شکل دو تا باس PROFIBUS برای ارتباط Redundant با ET200 IO و یک گذرگاه PROFIBUS برای ارتباط FDL بین دو CPU جهت سنکرون کردن در Software Redundancy در نظر گرفته شده است.

۳-۹-۳- کتابخانه بلوک های I/O افزونه (Block library Functional I/O Redundancy)

کتابخانه مربوط به بلوک های توابع I/O افزونه در مسیر زیر قرار دارد

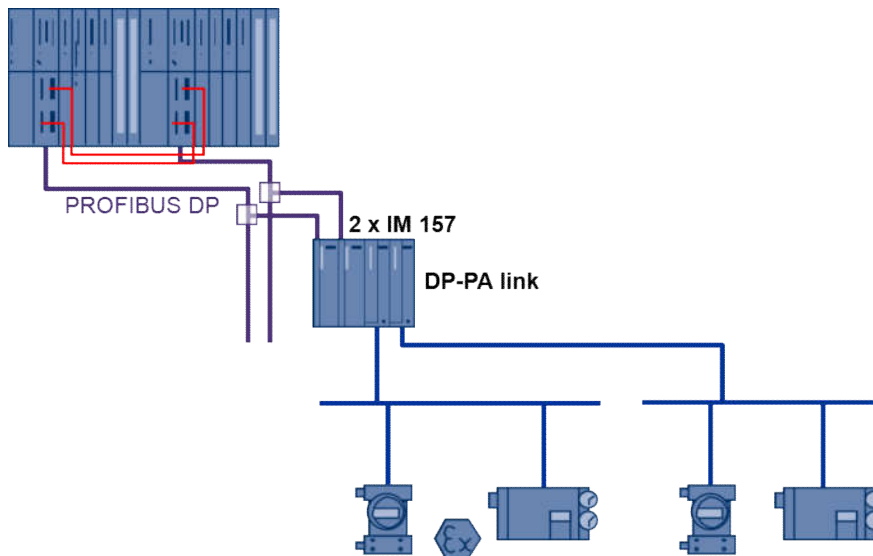
STEP 7\S7_LIBS\RED_IO

- FC 450 "RED_INIT": Initialization function
- FC 451 "RED_DEPA": Trigger depassivation
- FB 450 "RED_IN": Function block for reading redundant inputs
- FB 451 "RED_OUT": Function block for Writing redundant outputs
- FB 452 "RED_DIAG": Function block for diagnostics of redundant I/O
- FB 453 "RED_STATUS": Function block for redundancy status information

۳-۹-۴- اتصال PROFIBUS PA از طریق یک لینک PA به سیستم H

از آن جایی که باس PROFIBUS PA به طور مستقیم به CPU متصل نمی شود، اتصال این باس از طریق باس DP صورت می گیرد. لذا برای این کار از ماژول واسط IM 157 (DP-PA Link) استفاده می شود.

- IM 157: 6ES7 157-0AA82-0XA0
- Bus module BM IM 157 : 6ES7 195-7HD80-0XA0

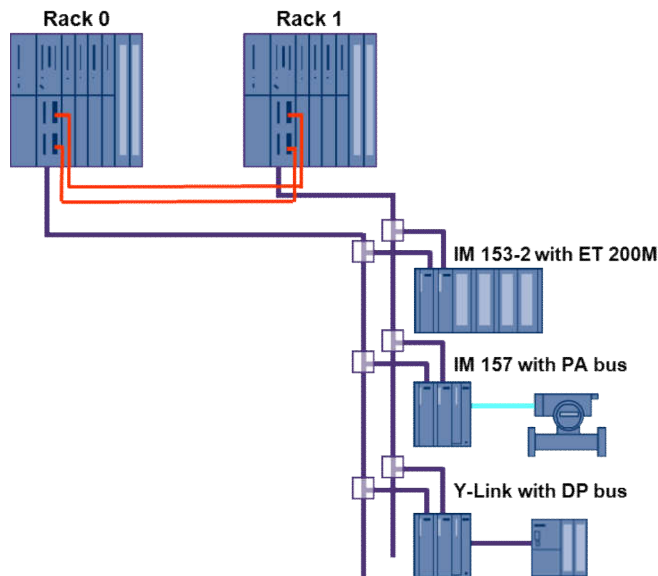


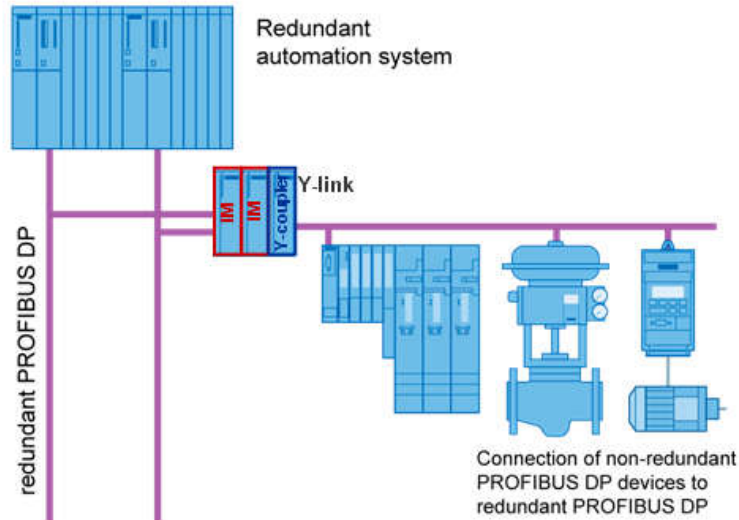
شکل ۳-۳۶- اتصال PROFIBUS PA از طریق یک لینک PA به سیستم H

۳-۹-۵- ماژول Y-Link

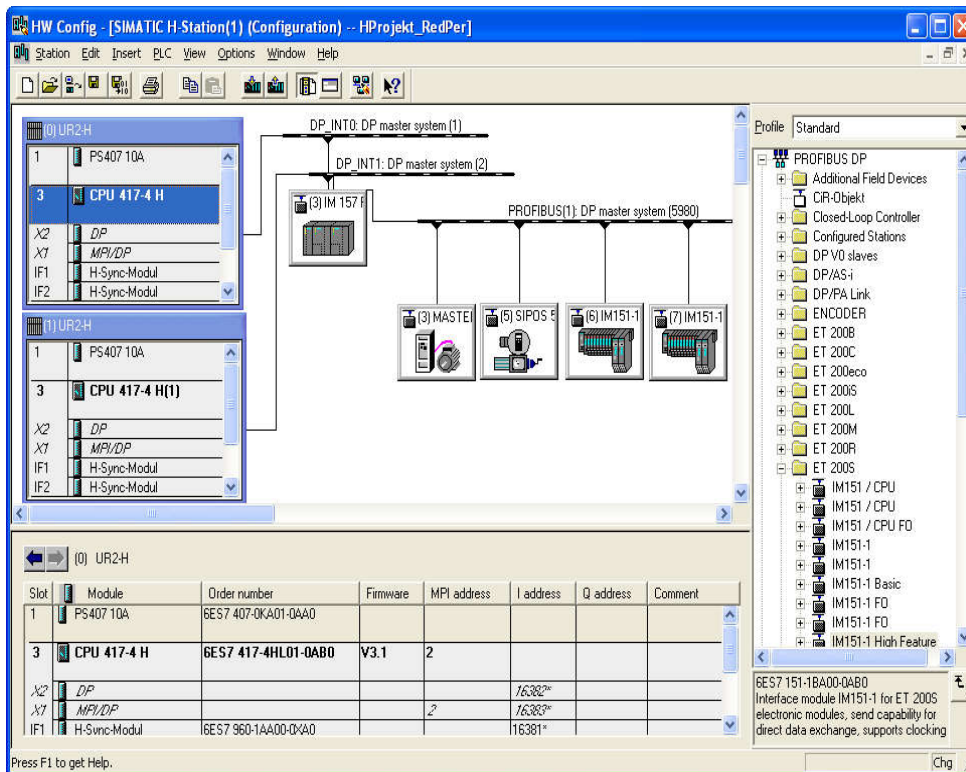
کار این ماژول همانند DP-PA Link می باشد. ماژول Y-Link زمانی استفاده می شود که بخواهیم یک شبکه DP غیر افزونه (تک کاناله) را به یک شبکه افزونه (دو کاناله) متصل کنیم.

- Y-Link: **6ES7 197-1LB00-0XA0**
- Bus module BM Y-Link : **6ES7 654-7HY00-0XA0**





شکل ۳-۳۷- اتصال شبکه DP تکی به یک شبکه DP افزونه از طریق واسط Y-Link



شکل ۳-۳۸- نمونه پیکربندی I/O از طریق واسط Y-Link در سیستم H

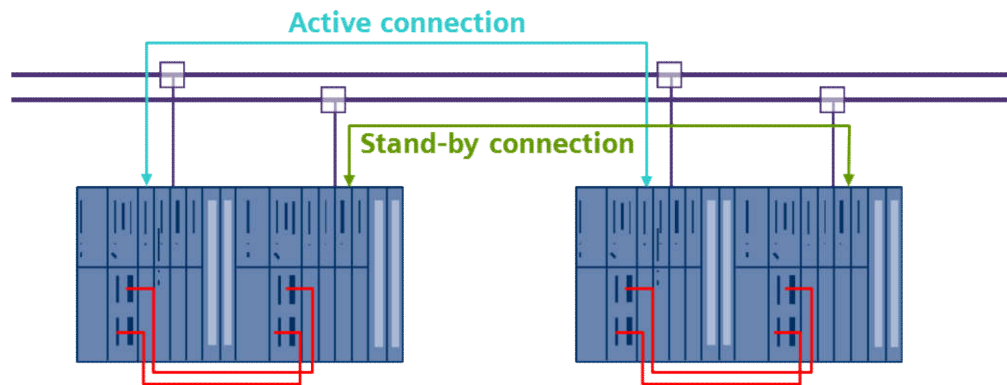
۳-۹-۶- واسط یا گذرگاه ارتباطی I/O

سیگنال های I/O می توانند به دو روش از طریق گذرگاه PROFIBUS و PROFINET متصل شوند. پیکربندی PROFIBUS و PROFINET را می توان با یکدیگر ترکیب کرد. واسط PROFIBUS پیکربندی های مختلف زیر را می تواند فراهم نماید.

- واسط یک طرفه – در دسترس پذیری عادی (*Normal availability*)
 - یک اتصال تک کانال (*a single-channel Connection*) – در دسترس پذیری عادی
 - واسط قابل سوییچ (*Switched interface*) – در دسترس پذیری افزونه (*Increased Availability*)
- واسط PROFINET یک رابط یک طرفه (دسترس پذیری عادی) یا واسط سیستم افزونه از طریق حلقه باز (*open ring*) فراهم می کند. دسترس پذیری در نوع حلقه باز زمانی که دستگاهها از افزونگی سیستم I/O PROFINET، مانند I/O توزیع شده ET200M پشتیبانی می کنند، افزایش می یابد.

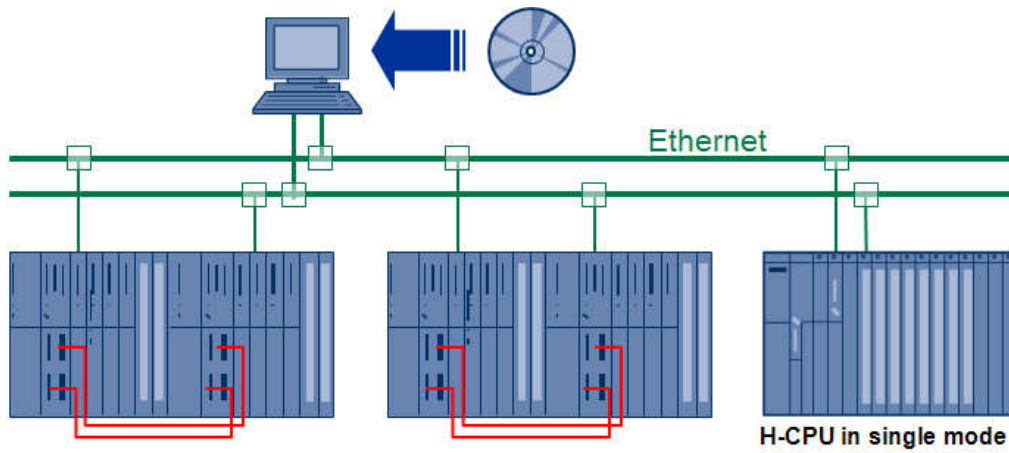
۳-۱۰- ارتباطات افزونه (*Redundant communication Principle*)

- ارتباطات افزونه بین دو یا چند سیستم از طریق اتصالات افزونه حاصل می شود. اتصالات افزونه (*Redundant connections*) را می توان از ایستگاههای H به ایستگاههای زیر ایجاد کرد.
- *Other H stations (one- or two-channel)*
 - *HMI PCs (software Redconnect required)*
- نوع اول جهت اتصال به دیگر ایستگاههای اتوماسیون نوع H و اتصال نوع دوم جهت اتصال به کامپیوترهای HMI استفاده می شود. در اتصال نوع دوم نصب نرم افزار *Redconnect* مورد نیاز است.



اتصالات افزونه بین دو سیستم افزونه H

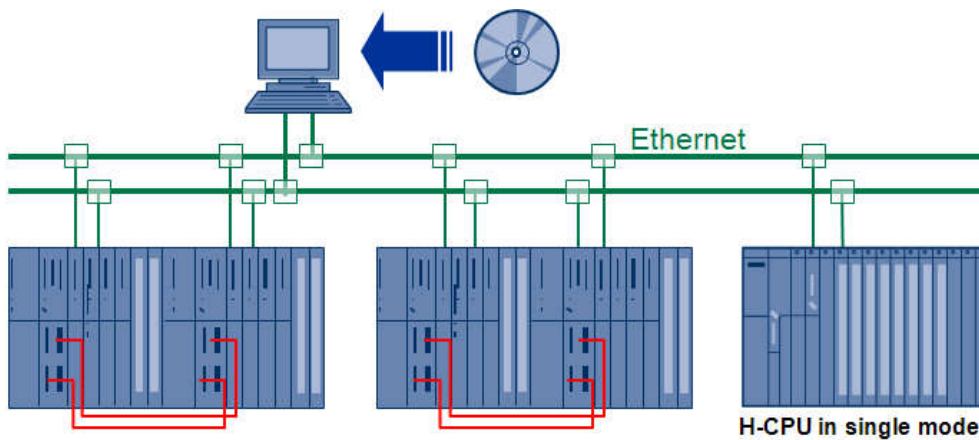
سامانه اتوماسیون S7-400H از طریق «پردازندههای ارتباطی یا ماژولهای CP» به گذرگاه پلنت متصل می شود. برای رسیدن به سطوح بالاتر افزونگی، می توان از دو عدد CP استفاده کرد.



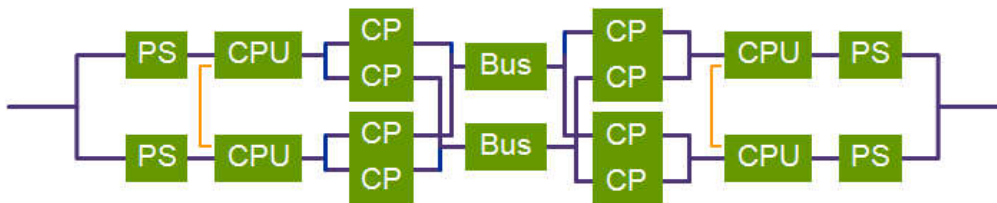
Equivalent circuit diagram:



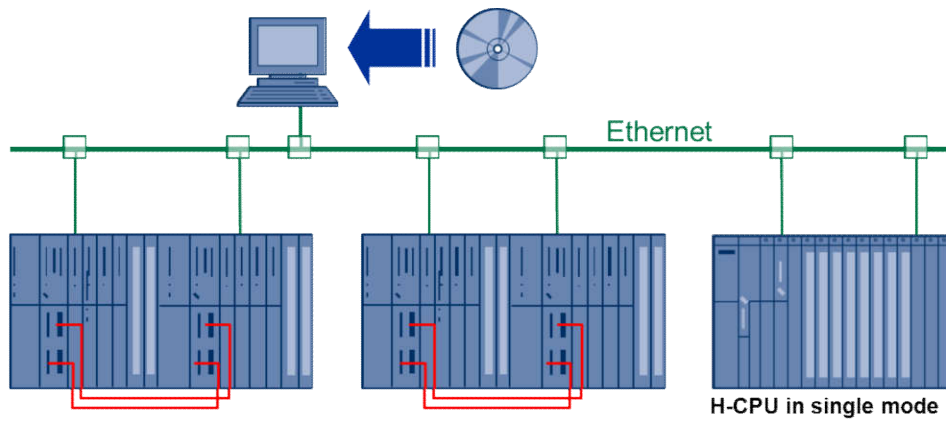
شکل ۳-۳۹- پیکربندی ارتباطات افزونه با باس افزونه



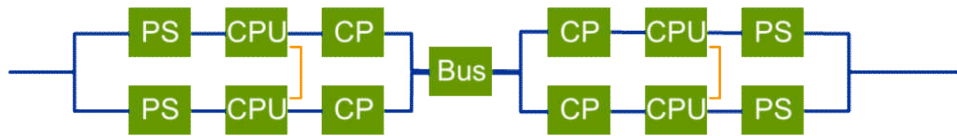
Equivalent circuit diagram:



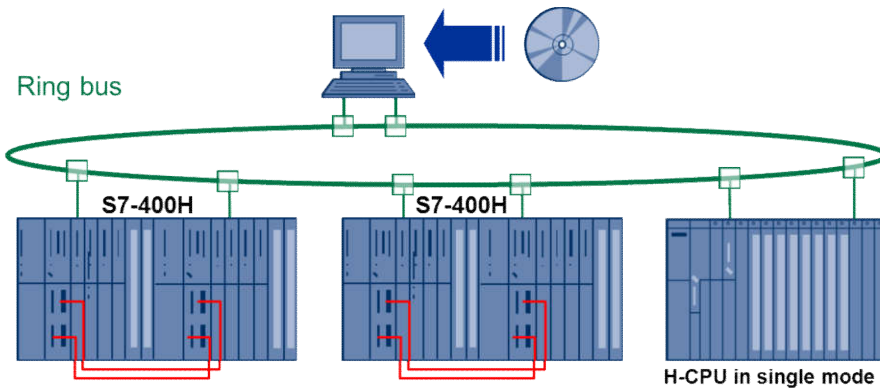
شکل ۳-۴۰- پیکربندی ارتباطات افزونه با باس افزونه و کارت CP افزونه



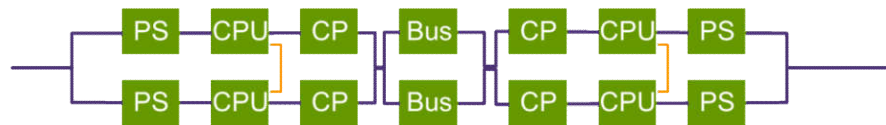
Equivalent circuit diagram:



شکل ۳-۴۱- پیکربندی سیستم H با باس تکی



Equivalent circuit diagram:



شکل ۳-۴۲- پیکربندی سیستم H با باس Ring

۳-۱۰-۱- ارتباطات از طریق اتصالات تحمل پذیر خطا (Fault-tolerant S7 Connections)

پیکربندی سخت افزار هر دو زیر سیستم که به صورت یکپارچه یک سیستم مقاوم در برابر خطا را ایجاد می کنند، بایستی یکسان باشد. این امر به ویژه برای اسلات ها (slots) درست است. بسته به شبکه ای که استفاده می شود، ماژول های CP زیر را می توان برای ارتباطات تحمل پذیر خطا استفاده کرد:

- Industrial Ethernet >> S7: CP 443-1

- PROFIBUS >> S7: CP 443-5 Extended (not configured as DP master system)
نکته: برای اینکه بتوانه از اتصالات fault-tolerant S7 connections بین یک سیستم مقاوم در برابر خطا (fault-tolerant system) و یک کامپیوتر (PC) استفاده کرد، بایستی بسته نرم افزار S7-REDCONNECT را بر روی کامپیوتر نصب کرد.

۳-۱۱- قابلیت اطمینان ماژول های زیمنس (Reliability of modules)

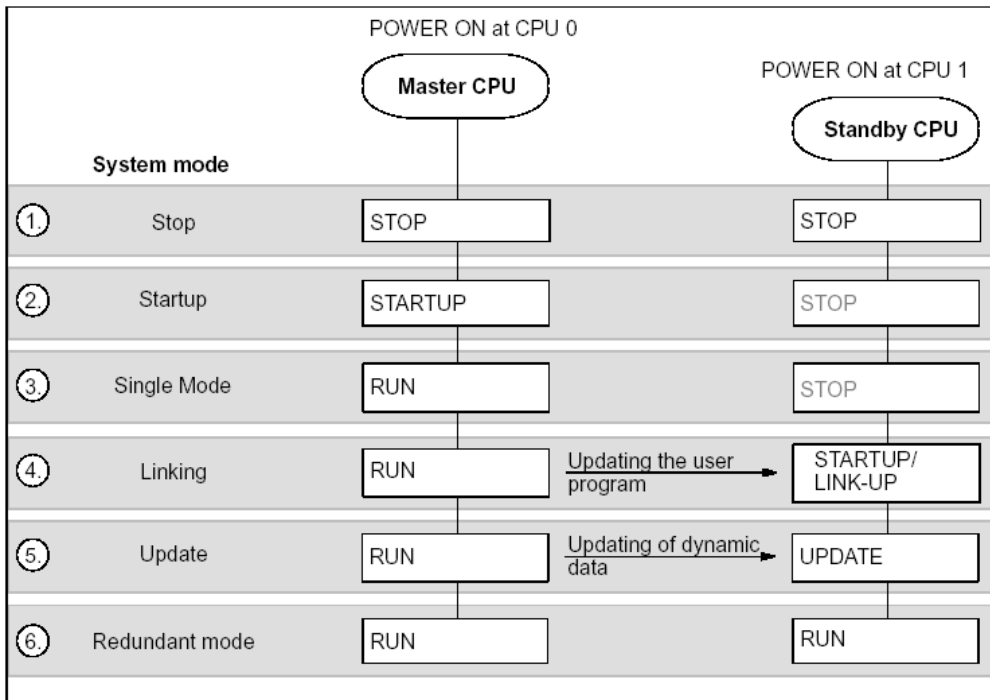
قابلیت اطمینان اجزای سیماتیک در نتیجه اقدامات تضمین کیفیت گسترده در توسعه و تولید محصولات سیماتیک، بسیار بالا است. فاکتورهای ارزیابی قابلیت اطمینان و دسترس پذیری بالا (High Availability & Reliability) عبارت انداز:

- Mean Time Between Failure (MTBF)
 - Mean Time To Repair (MTTR) that depends on:
 - Being Hot Plug
 - Fault Diagnostic Ability
 - Maintenance engineer reaction
- مقادیر متوسط زیر به قابلیت اطمینان ماژول سیماتیک اعمال می شود:
- MTBF of a central processing unit: 15 years
 - MTBF of an I/O module: 50 years

۳-۱۲- نصب و راه اندازی یک سیستم S7400H

۱- پس از نصب سخت افزارهای سیستم S7400H برای راه اندازی آن ابتدا منبع تغذیه واقع در رک ۰ و رک ۱ را روشن می کنیم. چراغ LED (DC 24V و DC 5V) مربوط به Power به رنگ سبز روشن می گردند. در این حالت چراغ Stop روی CPUها به رنگ زرد و چشمک زن می باشد. لذا تا زمانی که چراغ زرد Stop به صورت چشمک زن بوده و قطع نشده است. نمی توان با CPUها ارتباط داشت و آنها را برنامه ریزی کرد. معمولاً مدت زمان چشمک زن چراغ زرد در حدود ۱۰ الی ۱۵ دقیقه می باشد. در این حالت CPU در مد اجرای «خود آزمون راه اندازی» (Self Test) می باشد و اصطلاحاً CPU تمام قسمت های مختلف را چک می کند. این حالت همواره برای اولین بار که سیستم H مونتاژ و روشن می شود، اتفاق می افتد.

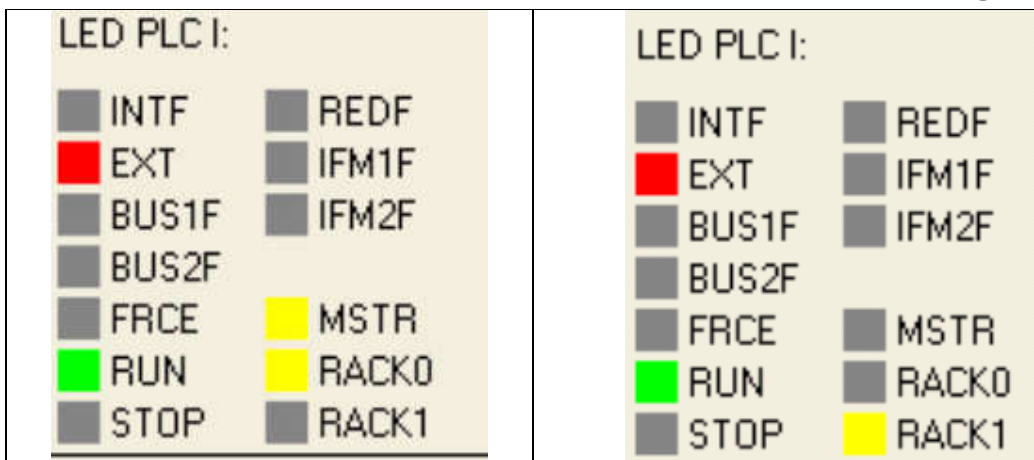
۲- بعد از این که سیستم یک بار تست شد و به عبارت دیگر عمل «خود آزمون» را انجام داد. در دفعات بعد که برق قطع و وصل شود، اگر باتری روی ماژول تغذیه PS موجود بوده و سالم باشد، عمل «خود آزمون» کوتاه خواهد بود. ولی اگر باتری نباشد و یا سویچ مربوطه در وضعیت خاموش باشد، با قطع و وصل شدن برق عمل «خود آزمون دوباره از نو شروع می شود. به این خاطر باتری نقش مهمی را در سیستم های S7400H بازی می کند.



شکل ۳-۴- راه اندازی سیستم S7 400H

همچنین اگر باتری وجود داشته باشد و عمل «خود آزمون» یک بار انجام شده باشد، ولی اگر ماژول CPU از رک جدا شده و دوباره نصب شود، عمل «خود آزمون» از نو تکرار خواهد شد.

۱- بعد از اتمام عمل «خود آزمون» چراغ Stop دیگر چشمک نمی زند و به رنگ زرد ثابت روشن می شود. برخی از چراغها به صورت قرمز روشن هستند. توضیحات مربوط به چراغهای سیستم S7 400H به شرح جدول زیر می باشد.



شکل ۳-۴- چراغهای ماژولهای CPU در سیستم S7 400H

جدول ۳-۲- توضیحات چراغهای روی ماژول منبع تغذیه (PS) در S7 400H

LED	توضیحات چراغهای روی ماژول منبع تغذیه (PS)
-----	---

<i>INTF</i>	<i>Internal fault</i> در شرایط نرمال نباید روشن باشد. به عنوان مثال اگر منبع تغذیه سالم باشد و یا مشکلی در داخل CPU نباشد، خاموش می باشد.
<i>BAF</i> <i>BATTF</i>	چراغ های مربوط به باتری هستند. اگر باتری هادر سوکت مربوطه نباشند روشن می شوند.
<i>DC 5V</i> <i>DC 24V</i>	چراغ روشن بودن تغذیه بوده و در صورت برقراری و سالم بودن با رنگ سبز روشن می شوند.

جدول ۳-۳- توضیحات چراغ های روی ماژول CPU در S7 400H

LED	توضیحات چراغ های روی CPU
<i>INTF</i>	<i>Internal fault</i> در صورتی که در برنامه CPU یا پردازش آن مشکلی باشد، روشن می شود.
<i>EXTF</i> <i>External Fault</i>	در صورت اشکال در ماژول های I/O روی رک مرکزی یا RIO این چراغ روشن می شود. به عنوان مثال اگر پیکربندی غلطی برای ماژول I/O روی ایستگاه ET200M برنامه ریزی شود.
<i>BUSIF</i> <i>Bus Fault</i>	در صورت اشکال در شبکه پروفیباس این چراغ روشن می شود.
<i>IFM1F</i> <i>IFM2F</i>	<i>Interface Module 1 & 2 → SYNC Modules</i> اگر ارتباط فیبر نوری ماژول سینک بالایی مشکل داشته باشد (قطع باشد)، چراغ IFM1F روی هر دو CPU روشن می شود. اگر ارتباط فیبر نوری ماژول سینک پایینی مشکل داشته باشد (قطع باشد)، چراغ IFM2F روی هر دو CPU روشن می شود.
<i>FRCE</i>	در صورتی که یکی از کانال های I/O به یک مقداری Force شده باشد این چراغ روشن می شود.
<i>REDF</i>	در صورت بروز اشکال در ارتباط افزونه بین دو CPU از طریق فیبر نوری، (کابل فیبر نوری قطع شود) و یا وقوع اشکال در پروفیباس مربوط به ایستگاه های ET200M، این چراغ روشن می شود

۳-۱۲-۱- مانیتورینگ سگنال های سیستم H

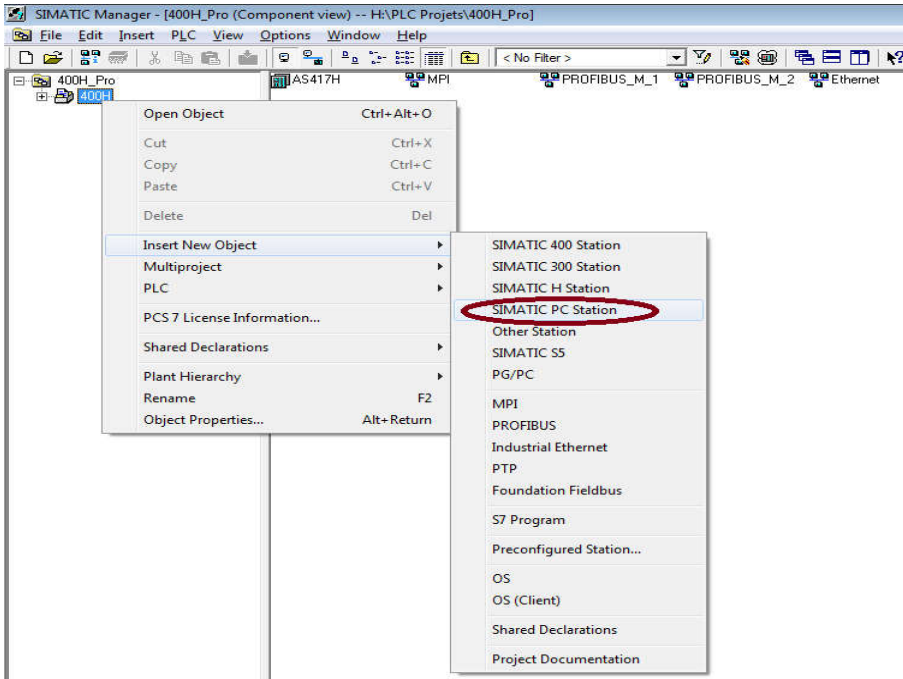
در سیستم های H در کنار هر CPU یک کارت اترنت وجود دارد و به روش معمولی نمی توان عمل مانیتورینگ را از طریق اترنت انجام داد و همواره با مشکلاتی همراه است. لذا باید به یک روش خاصی این کار را انجام داد.

۱- اگر از طریق MAC address به کارت اترنت CPU Master وصل شویم، در صورت خاموش شدن کارت اترنت Standby، اختلالی در ارتباط بین مانیتورینگ و PLC رخ نمی دهد. ولی اگر کارت اترنت واحد Master خاموش شود، وقتی سیستم به مد Solo (Single) می رود، اون سیستمی که کارت اترنت آن را در Wincc معرفی کردیم، خاموش بوده و مانیتورینگ از دست می رود. کارت اترنت زیمنس CP1613، قابلیت مانیتورینگ را با یک پیکربندی خاص در سیستم H فراهم می کند.

ایجاد Connection موسوم به Named Connection

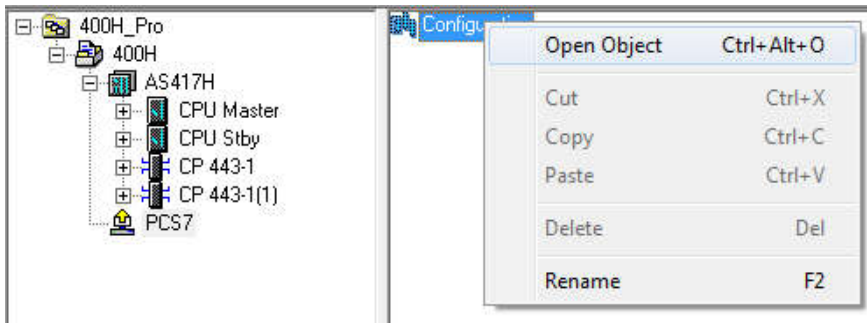
در این روش همانند سیستم H، تنظیمات خود PC را در محیط HwConfig را انجام می دهیم از کاتالوگ کارت CP1613 را انتخاب و در اسلات قرار می دهیم.

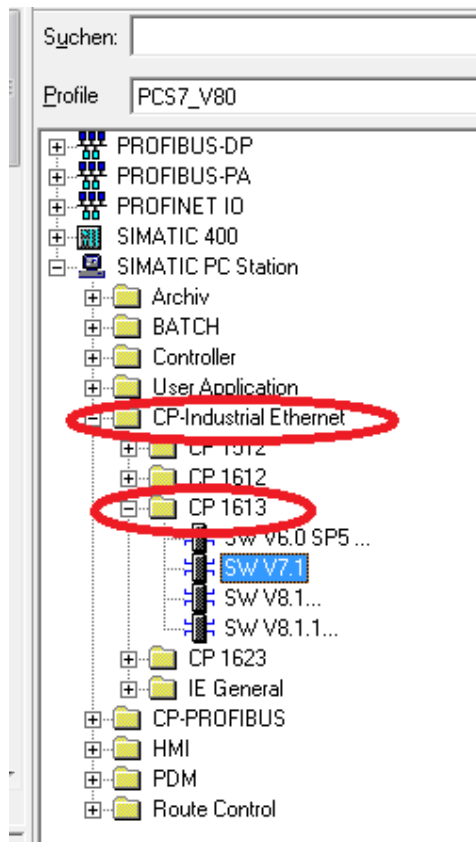
از طریق نرم افزار *SIMATIC Net* می توان مشخصات کارت *CP1613* را استخراج کرد.
از بخش *Version* باید نسخه مربوطه را استخراج کرد.



شکل ۳-۴۵- ایجاد یک *PC Station* برای سیستم مانیتورینگ

اسم *PC Station* را به اسم کامپیوتر (*PCS7*) تغییر می دهیم.





شکل ۳-۶-۴-

بایستی یک *Wincc Application* هم معرفی کنیم. برای این کار پس از اتمام کار بایستی عمل *Save & Compile* انجام داد. در *Wincc* یک پروژه ایجاد می گردد. برنامه *NetPro* را باز می کنیم. فایل *XDB* را باید از مسیر پروژه *Import* کنیم. در سیستم های *H* کارت های *I/O* به ندرت در کنار سیستم *PLC* نصب می شوند. بیشتر به بر روی یک رک *RIO* نوع *Active Bus* (با استفاده از *Bus Module* یا *Bus unit*) نصب می شوند. اتصال بین این ماژول های از طریق دو ماژول *IM 153*، کابل و کانکتور ۹ پین پروفیباس صورت می گیرد. بین دو قسمت (۹ اسلات در هر قسمت) رک *UR2H* هیچ ارتباط الکتریکی وجود ندارد. باس های موجود در هر دو طرف جدا از هم هستند. منابع تغذیه را در اسلات های ۱ هر قسمت نصب می کنیم بعد از نصب منبع تغذیه، ماژول *CPU* می تواند در هر اسلاتی نصب شود. در پشت ماژول *CPU* یک *Dip Switch* است. که در کنار آن بسته به حالت قرار گیری سویچ در بالا یا پایین عبارت های *Rack 0* و *Rack 1* قرار گرفته است. در یک سمت یکی از سویچ ها بر روی *Rack 0* قرار گیرد و در سمت

دیگر بایستی بر روی *Rack 1* قرار گیرد. این تنظیم فرقی نمی کند که در کدام طرف روی یک یا صفر قرار گیرد. لازم به یادآوری است که هر دو *CPU* بایستی در هر دو طرف بر روی یک اسلات با شماره یکسان نصب شوند.

بر روی *Bus Module* مربوط به کارت های واسط *IM 153* عبارت *IM* در دو طرف آن نوشته شده است. بر روی هر *Bus Module* دو عدد ماژول *IM 153* قابل نصب می باشد.

برای نصب کارت های *I/O* معمولی از یک *Bus Module* دیگر استفاده می شود. بر روی این *Bus Module* نیز دو عدد کارت *I/O* نصب می شود

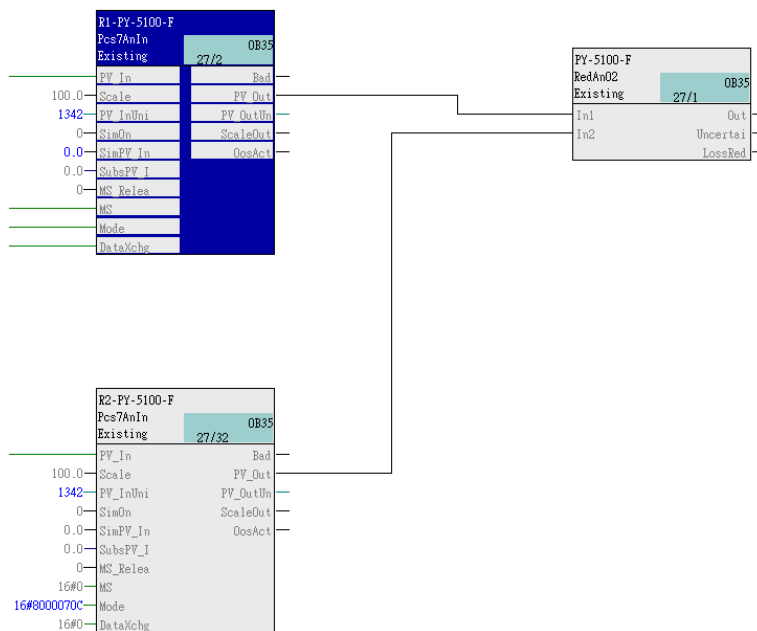
برای نصب کارت های *I/O* نوع *Fail safe* از یک *Bus Module* دیگر که عرض آن بیشتر از کارت های معمولی است، استفاده می شود. بر روی این *Bus Module* یک عدد کارت *I/O* نوع *Fail safe* نصب می شود.

۳-۱۳- توصیف فانکشن بلاک ها

۳-۱۳-۱- فانکشن بلاک *RedAn02*

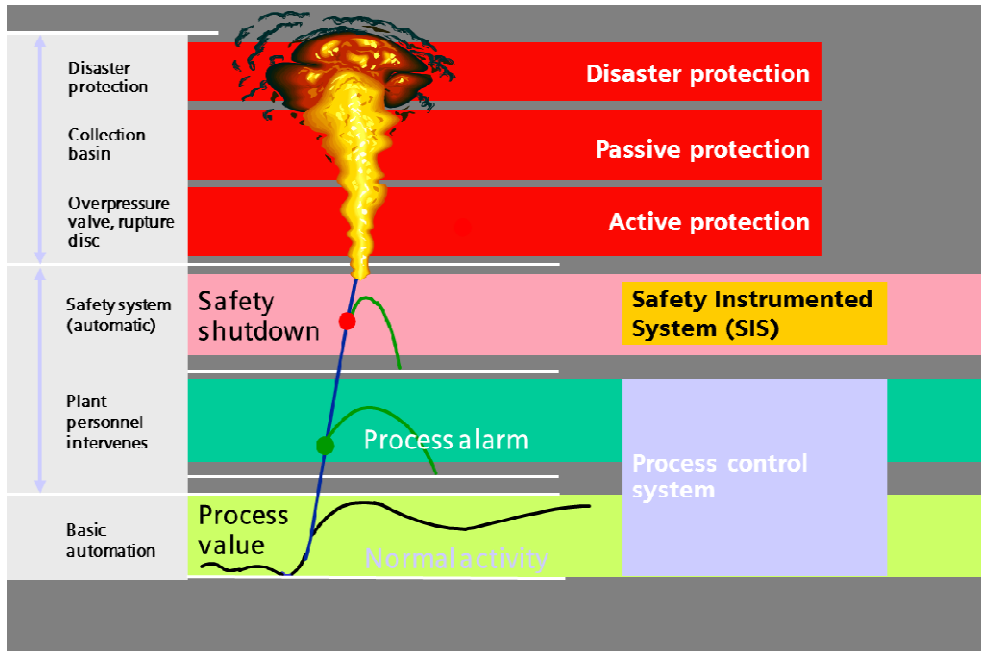
این فانکشن بلاک برای عملیات انتخاب مقدار سیگنال آنالوگ بین دو ورودی افزونه استفاده می شود.

- 1 out of 2 selection for redundant analog values



شکل ۳-۴۷- پیاده سازی درایور کانال ریداندانت در محیط CFC

فصل ۴ - ساختار و معماری سیستم های FH

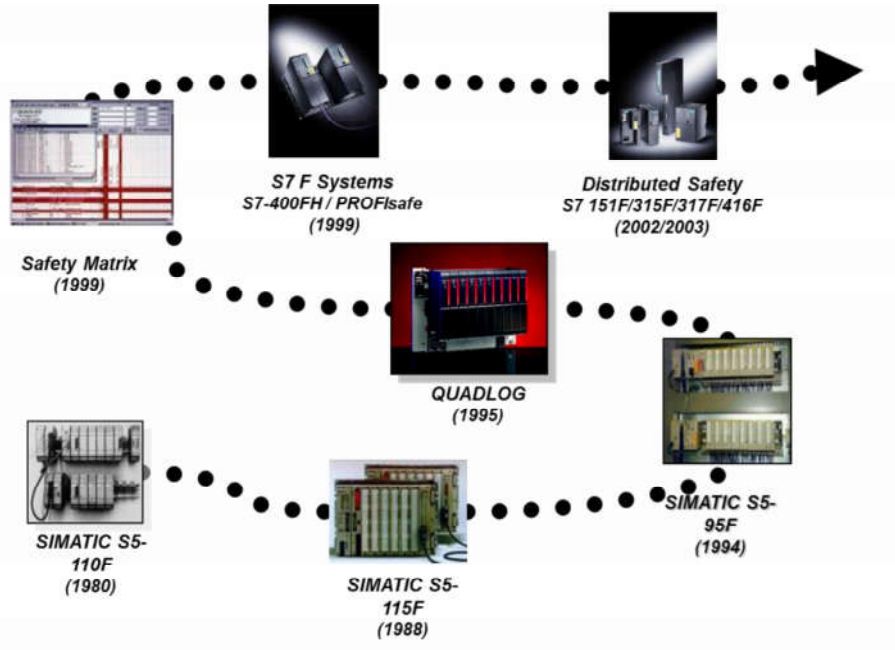


جایگاه سیستم SIS در لایه های حفاظت

۴-۱-۱- مقدمه

۴-۱-۱-۱- تاریخچه سیستم های ایمنی زیمنس (History of Siemens Safety Systems)

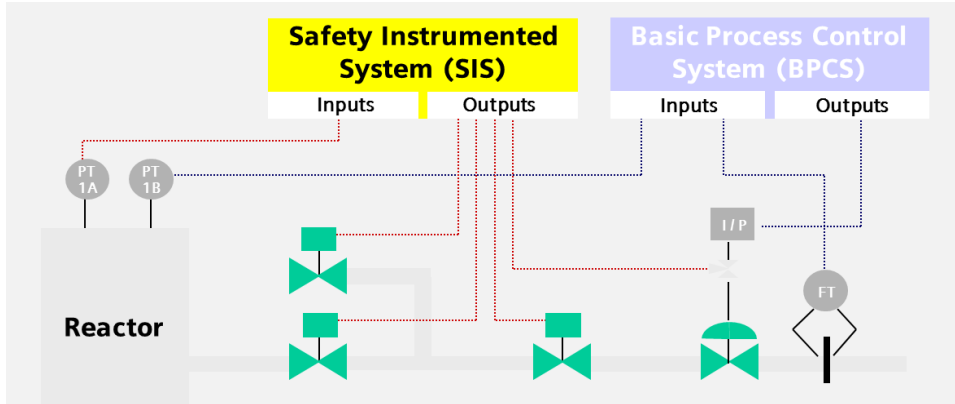
شکل زیر تاریخچه سیستم های کنترل ایمن در نقص زیمنس را به تصویر کشیده است.



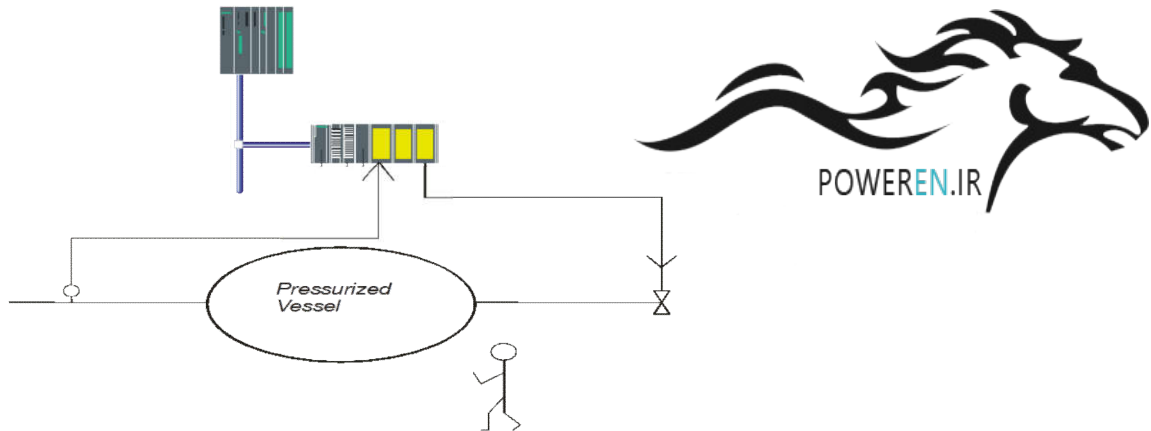
شکل ۴-۱- تاریخچه سیستم های کنترل ایمن در نقص زمین

۴-۱-۲- تفاوت بین سیستم SIS و سیستم کنترل پایه BPCS

به طور معمول در پلنت هایی همچون پتروشیمی که نیاز به ایمنی همزمان با کنترل حلقه های آنالوگ نیاز است از یک سیستم SIS برای کنترل حلقه های کنترل کننده ایمنی سیستم و از یک سیستم کنترل با عنوان DCS یا BPCS برای کنترل حلقه های آنالوگ استفاده می شود.



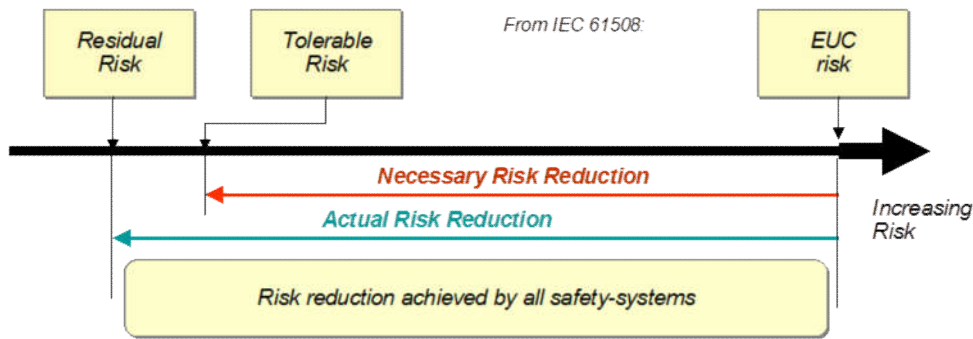
شکل ۴-۲- حلقه های کنترل SIS و BPCS برای پلنت راکتور



شکل ۴-۳- نمونه کاربرد سیستم F

۴-۱-۳- هدف از سیستم های SIS

هدف از بکارگیری سیستم های SIS، کاهش ریسک از طریق سیستم SIS، می باشد.



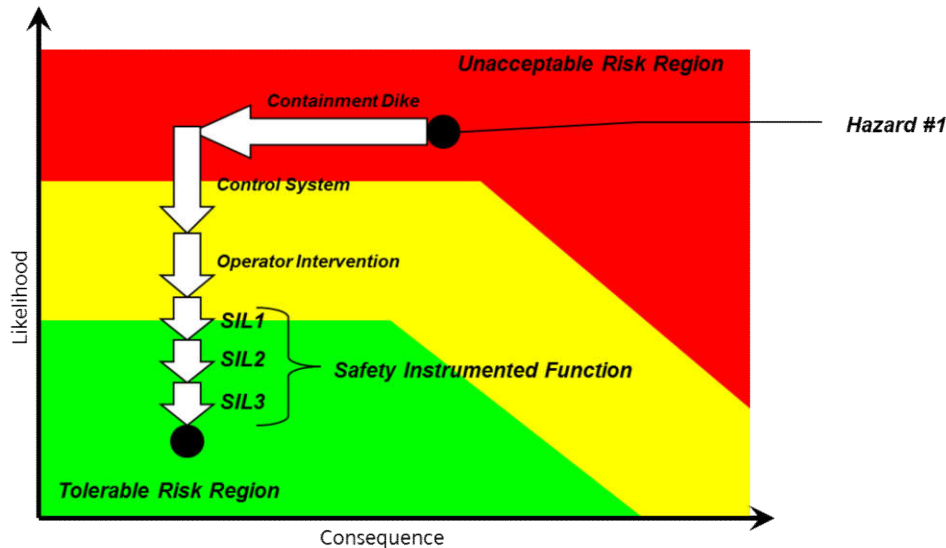
۴-۱-۴- ریسک چیست و چه کسی تصمیم می گیرد که چه ریسکی قابل قبول است؟

نمونه هایی از شکل های مختلف احتمال مرگ عبارت انداز:

- Road accident → 100cpm $1.0 \times 10^{-4}/yr$ 1 av 100 (at life 100 years)
- Car accident 150cpm $1.5 \times 10^{-4}/yr$ 1,5 av 100
- Accident at work 10cpm $1.0 \times 10^{-5}/yr$ 1 av 1000
- Falling Aircraft 0.02 cpm $2.0 \times 10^{-8}/yr$ 2 av 1000 000
- Lightning strike 0.1cpm $1.0 \times 10^{-7}/yr$ 1 av 100 000
- Insect/Snake bite 0.1cpm $1.0 \times 10^{-7}/yr$ 1 av 100 000
- Smoking 20 per day 5000 cpm $5.0 \times 10^{-3}/yr$ 1 av 2

که در آن cpm عبارت است از احتمال وقوع در بین یک میلیون نفر در سال:

$cpm = \text{chances per million of the population (per year)}$



برخی از کاربردهای سیستم های SIS عبارت انداز:

- ESD, Emergency Shutdown
- F&G, Fire & Gas Detection, Fire-fighting
- Process Shutdown
- Fire-pump Logic
- Ballast Control
- Blow-down
- Riser release / Anchor Release
- Fire Dampers, Active Smoke Control

- *HIPPS, High Integrity Pressure Protection System*

۴-۱-۵- استانداردهای مهم مرتبط با ایمنی

- *IEC 61508 (Application-independent standard*)
Function safety of electrical / electronic / programmable electronic safety-related systems*
- *DIN V VDE 0801 Fundamentals for computers in systems with safety tasks.*
- *VDI/VDE 2180 Ensuring the safety of process plants using instrumentation and control*
- *prEN50156 Standard for burner controls Electrical equipment for burner systems...” (en)*
- *EN 298 Standard for burner controls “Automatic gas burner control system...”*
- *EN 954 Standard for safety of machinery
Safety related parts of control systems”*
- *DIN V 19250 “Fundamental safety considerations for protection facilities in measuring and control technology”*

۴-۲- سخت افزار سازگار با سطوح SIL 1 , SIL2, SIL3

در شکل زیر اطلاعات لازم برای درک سطوح SIL 1 , SIL2, SIL3 نمایش داده شده است. هر یک از سطوح SIL وابسته به سه مقدار HFT (hardware fault tolerance)، مقدار DC (diagnostic coverage) و SFF (safe failure fraction) می باشد. اجزای الکترومکانیکی به عنوان مثال سوئیچ های موقعیت یا کنتاکتورها دارای $HFT=0$ هستند. اگر از دو جزء استفاده می کنید، لذا $HFT=1$ خواهد بود. با توجه SIL2 یا SIL3 در ترکیب با قطعات الکترو مکانیکی باید از دو جزء استفاده کنید. قطعات الکترونیکی به طور معمول یک مقدار SIL CL ثابت دارند. این مقدار از تولید کننده دریافت می شود.

Table 5 – Architectural constraints on subsystems: maximum SIL that can be claimed for a SRCF using this subsystem

Safe failure fraction	Hardware fault tolerance (see Note 1)		
	0	1	2
< 60 %	Not allowed (for exceptions see Note 3)	SIL1	SIL2
60 % – < 90 %	SIL1	SIL2	SIL3
90 % – < 99 %	SIL2	SIL3	SIL3 (see Note 2)
≥ 99 %	SIL3	SIL3 (see Note 2)	SIL3 (see Note 2)

NOTE 1 A hardware fault tolerance of N means that $N+1$ faults could cause a loss of the safety-related control function.
NOTE 2 A SIL 4 claim limit is not considered in this standard. For SIL 4 see IEC 61508-1.
NOTE 3 See 6.7.6.4 or for subsystems where fault exclusions have been applied to faults that could lead to a dangerous failure, see 6.7.7.

۴-۲-۱- انتخاب یک CPU برای یک پلنت فرآیندی

در انتخاب یک CPU برای یک پلنت، معیارهای بسیاری وجود دارد که باید در نظر گرفته شود. در ابتدا باید مشخص کنید که CPU را برای چه نوع پلنت فرآیندی انتخاب می کنید. جزئیات زیر را می توان برای انتخاب یک CPU برای یک پلنت فرآیندی در نظر گرفت:

- ۱- تعداد ورودی/خروجی - به طور معمول پلنت های فرآیندی دارای تعداد I/O از ۵۰۰ تا ۱۰۰۰۰۰ هستند.
- ۲- تعداد حلقه های بسته (Closed loops). اگر تعداد حلقه های کنترل آنالوگ بیشتر باشد. در آن صورت به سیکل اسکن سریع تر نیاز هست.
- ۳- تعداد حلقه های PID - بیشتر قدرت پردازش در حلقه های بسته صرف می شود.
- ۴- نوع پلنت (نوع فرآیند). اگر یک فرآیند از نوع حیاتی است. پس نیاز است که یک CPU با قابلیت دسترس پذیری بالا (high availability) را انتخاب کنید. به عنوان مثال CPU 417H
- ۵- نوع سیستم ابزار دقیق - روش اتصال سنسورها به سیستم کنترل چیست.

- 4-20mA
- HART
- Profibus
- MODBUS
- Foundation Fieldbus

- ۶- نوع معماری سیستم کنترل (توزیع شده یا کنترل متمرکز)
- ۷- همچنین در برآورد واقع بینانه از عملکرد مورد نیاز برای واسط HM دقت خاصی باید شود.
- ۸- آیا CPU برای کنترل پایه فرآیند (DCS) است یا برای کاربردهای مربوط به SIL مورد نیاز است اینها برخی از مهم ترین ملاحظات است که در انتخاب یک CPU باید مدنظر باشد.

۴-۳- سیستم های اتوماسیون خرابی امن FH (Fail Safe Automation System)

برای طراحی سیستم کنترل پلنت هایی که نیاز به قابلیت اطمینان و ایمنی بالایی دارند. به عبارت دیگر در سیستم هایی که هر دو قابلیت ایمنی (Safety) حداکثر و در دسترس پذیری بالا (Availability) مورد نیاز است، سیستم S7-400FH یکی از گزینه ها می باشد. در چنین فرآیندهایی به هنگام وقوع خطا سیستم کنترل باید فرآیند را به شرایط ایمن از پیش تعریف شده برود تا خسارتی متوجه افراد، محیط زیست، دستگاهها و فرآیند نگردد. این PLC ها هر دو مشخصات سری H و F را دارا می باشند و علاوه بر نرم افزار استاندارد Step7 به دو بسته F-System و H-System نیز نیاز است.

سیستم های F فرآیندهای صنعتی را کنترل و یا از آنها محافظت می کنند. به طوری که یک توقف فوری، جان پرسنل یا محیط زیست را به خطر نمی اندازد.

۴-۳-۱- اصول پشت پرده عملکرد ایمنی (Principle behind the Safety Function)

- رفتار خرابی امن با استفاده از توابع ایمنی در درجه اول در نرم افزار حاصل می شود.
- هنگامی که یک رویداد خطرناک رخ می دهد، توابع ایمنی (Safety functions) سیستم را به حالت امن هدایت می کنند و یا اینکه سیستم را در یک حالت امن نگه می دارند.
- در صورت بروز یک فالت، سیستم F دیگر نمی تواند تابع ایمنی تعریف شده را اجرا کند. لذا تابع مربوط به عکس العمل در برابر فالت را اجرا می کند. به عنوان مثال، خروجی های مرتبط خاموش (switched off) شده و برنامه ایمنی یا بخش هایی از برنامه ایمنی، در صورت لزوم غیر فعال می شود. مثالی از توابع ایمنی کاربر و توابع واکنش در برابر فالت عبارت انداز:
 - در صورت وقوع یک رخداد فشار بیش از حد، سیستم های F، یک ولو را باز می کنند.
 - در صورت وقوع یک فالت خطرناک در CPU-F، تمام خروجی ها، خاموش می شود. به این قابلیت تابع عکس العمل فالت یا F shutdown گفته می شود.
- به طور معمول قابلیت های توابع ایمنی در اجزای زیر گنجانیده می شود:
 - در برنامه لاجیک ریخته شده در CPU از طریق بلاک های کتابخانه F (F Library)
 - در ماژول های ورودی / خروجی Fail-Safe از طریق تنظیم پارامترهای ماژول F



شکل ۴-۴- پیاده سازی توابع ایمنی

۴-۳-۲- پیکربندی های پایه برای تامین ایمنی و در دسترس پذیری (Basic Configuration Variants)

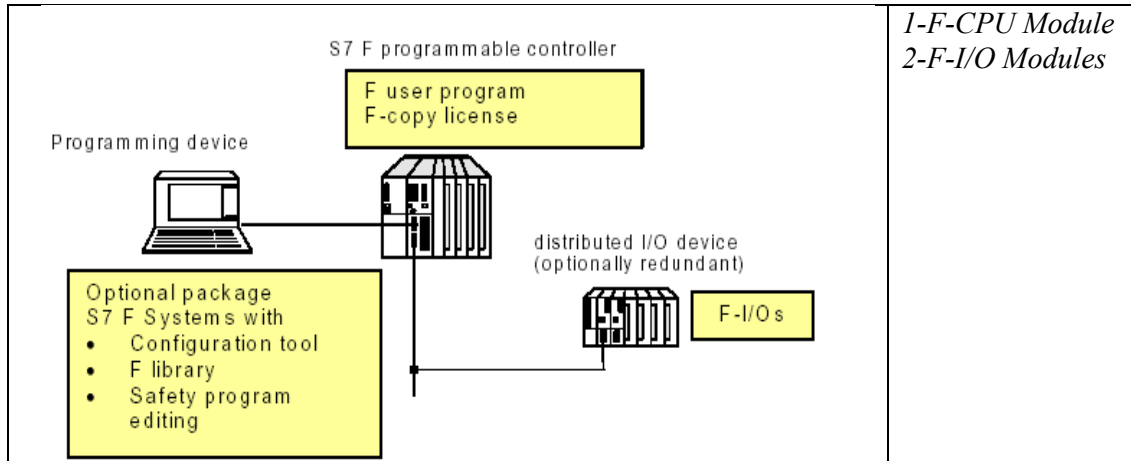
شرکت زیمنس سیستم های مختلفی برای پیکربندی های مستقل و ترکیبی از Fail safe و fault-tolerant به بازار عرضه کرده است.

- سیستم های مستقل خرابی امن (Fail-safe S7 F System)
- سیستم های ترکیبی از نوع Fail-safe و fault-tolerant که به سیستم های FH معروفند.
- Safety : Fail-safe S7 F System
 - CPU 315-F
 - CPU 317-F
 - CPU 416-F
- Safety & Availability: Fail-safe, fault-tolerant S7 FH System
 - CPU 412-F/H
 - CPU 414-F/H

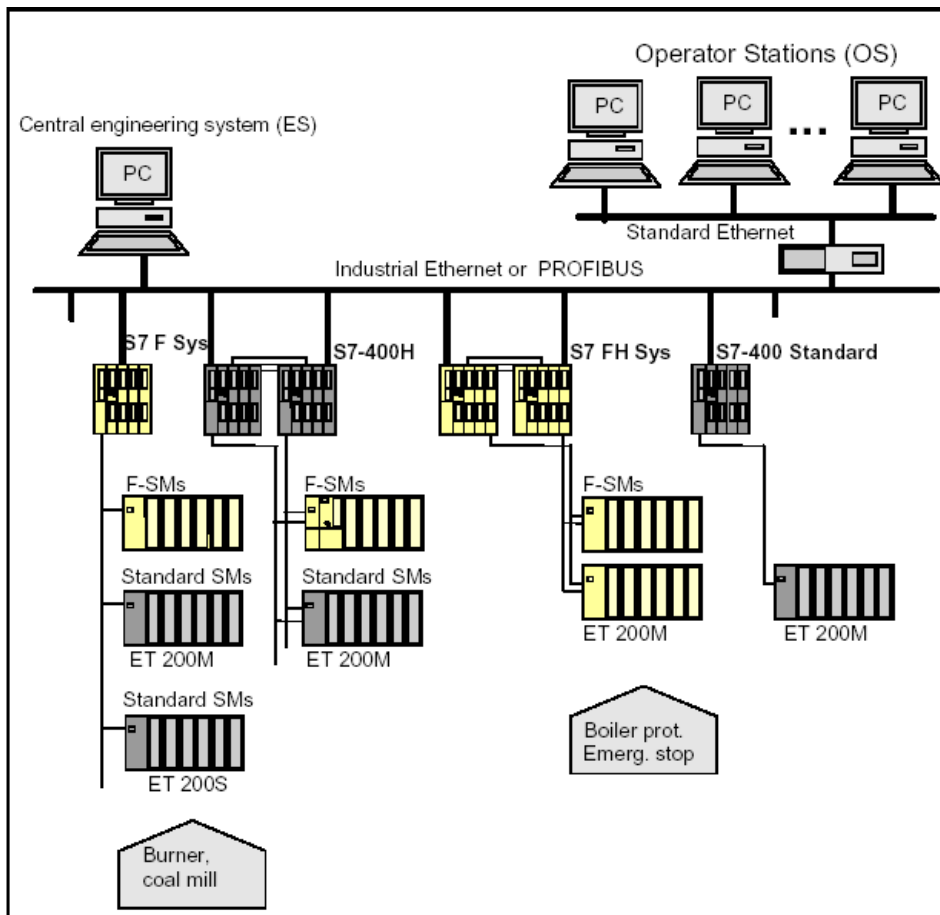
▪ CPU 417-F/H

سخت افزار یک سیستم اتوماسیون امن در خرابی S7 F متشکل از حداقل اجزای زیر است:

- یک ماژول CPU با قابلیت اجرای برنامه F، مانند CPU 417-4 H، که می تواند یک برنامه لاجیک از نوع خرابی امن (F) را اجرا کند.
- یک یا چند ورودی/خروجی از نوع خرابی امن (F-I/Os) در یک رک I/O توزیع شده (با قابلیت افزونگی اختیاری)



شکل ۴-۵- مولفه های یک سیستم FH

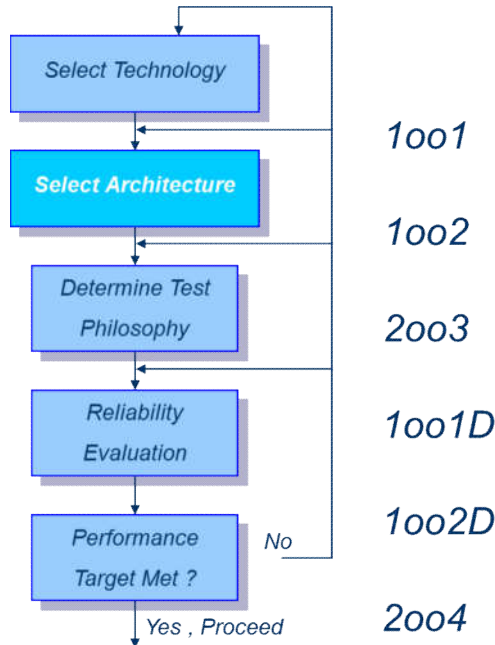


شکل ۴-۶- مثالی از پیکربندی سیستم FH زیمنس

۴-۴- معماری های سیستم Fail Safe

Safety Integrity Level	High demand or continuous mode of operation (probability of a dangerous failure per hour)	Low demand mode of operation (average probability of failure to perform its design function on demand)
4	$\geq 10^{-9}$ to $< 10^{-8}$	$\geq 10^{-5}$ to $< 10^{-4}$
3	$\geq 10^{-8}$ to $< 10^{-7}$	$\geq 10^{-4}$ to $< 10^{-3}$
2	$\geq 10^{-7}$ to $< 10^{-6}$	$\geq 10^{-3}$ to $< 10^{-2}$
1	$\geq 10^{-6}$ to $< 10^{-5}$	$\geq 10^{-2}$ to $< 10^{-1}$

۴-۴-۱- افزونگی مورد نیاز

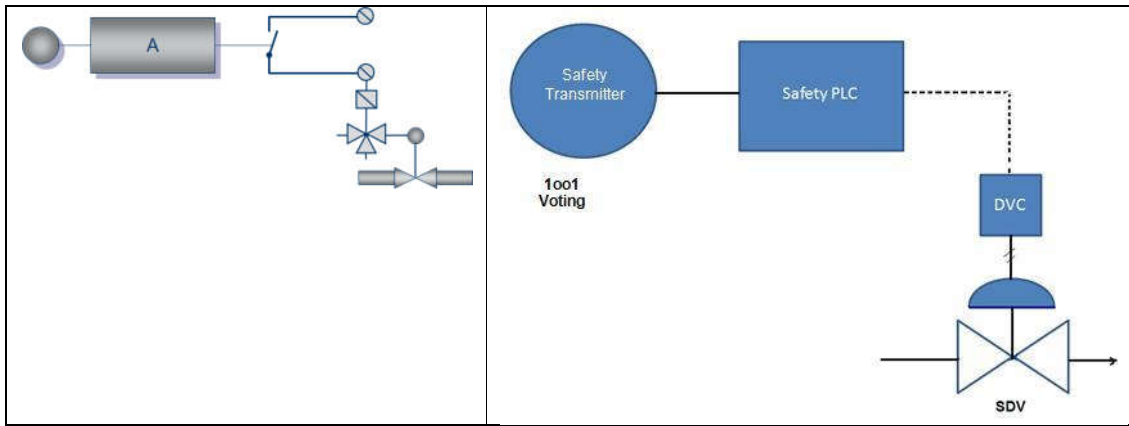


۴-۴-۲- معماری سیستم تک سنسور یا ارزیابی 1001 (1 Out of 1)

سیستم تک کانال

در این نوع ارزیابی تنها یک سنسور از طریق کانال سیگنال به ماژول سیگنال F متصل می شود. این معماری متشکل از یک سنسور واحد می باشد، که در آن هر نقص یا فالت خطرناک منجر به شکست تابع ایمنی (safety function) در مواقع نیاز می شود. به عنوان مثال عملکرد ایمنی (safety function) در شکل زیر باز کردن یا بستن یک شیر می باشد.

SF یا SIF (Safety Function or Safety instrumented Function) همان حلقه ایمن ابزار دقیق می باشد شکست امن (Safe failure): در حالتی که سیستم با شکست مواجه و ایمنی سیستم و افراد تامین می شود، یعنی سیستم پس از شکست به یک وضعیت امن می رود، به این معنی است که با عملکرد برنامه ایمنی (SF) کنتاکت های رله باز شده و برق سیستم (تابع ایمنی که در اینجا همان ولو می باشد) را قطع می کند. یعنی با وقوع شکست در سیستم، سیستم به یک حالت امن سوق داده می شود. شکست خطرناک (Dangerous failure): در این نوع شکست با فرمان از طرف کنترل کننده، کنتاکت های رله جوش خورده و به هنگام نیاز عمل نمی کنند.



شکل ۴-۷- معماری سیستم (1 Out of 2) (1001) (SIF – SIL 2)

۴-۳- ارزیابی 1002 (1 Out of 2)

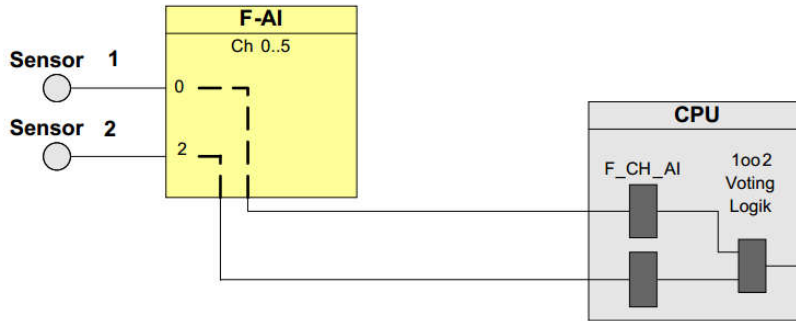
در این نوع ارزیابی برای یک سیگنال فرآیند، یک یا دو سنسور به دو ورودی مقابل هم (opposite inputs) در یک ماژول سیگنال F متصل می شود. وضعیت سیگنال ورودی ها (هم ارز یا غیر هم ارز) به صورت داخلی مقایسه می شوند. با این ارزیابی کلاس های ایمنی زیر قابل دستیابی می باشد.

- سیستم تک کانال- SIL2: در مورد چند کانال، کلاس SIL3 را می توان از طریق سیستم Voting در CPU بدست آورد.
- دو کانال- SIL3: عمل Voting در ماژول انجام می شود.

سیستم دو کانال (Dual Channel system)

- در این معماری از دو عدد سنسور برای ارزیابی سطوح ایمنی مورد نیاز استفاده می شود. استفاده از دو سنسور یا طرح ارزیابی 1002 به کاربردهایی اشاره می کند که برای رسیدن به سطح SIL (Safety Integration Level) مورد نیاز به دو سنسور نیاز دارد. ارزیابی 1002 بدان معنی است که برای صدور فرمان تریپ (trip) یا تریگر شدن لاجیک ایمنی (safety logic) تنها یکی از دو سنسور نیاز است که Fail کند. به عبارت دیگر اگر یکی از دو سنسور خراب شود، مدار تریپ فعال می شود. لذا در زمانی که یکی از دو زیر سیستم با شکست مواجه شده است، سیستم دیگر عملیاتی نمی باشد.
 - همچنین می توان گفت که در معماری 1002 تنها نیاز است که یک کانال عمل خاموش کردن تجهیز را انجام دهد.
 - در این معماری خروجی های دو کانال به صورت سری باهم بسته می شوند. (1002-1 Out of 2)
 - 2 von 2 (2v2): یعنی تنها زمانی که هر دو کانال عملیاتی باشند، سیستم فعال باقی می ماند.
- برای ارزیابی 1002 دو معماری پایه قابل انتخاب، توصیه می شود:

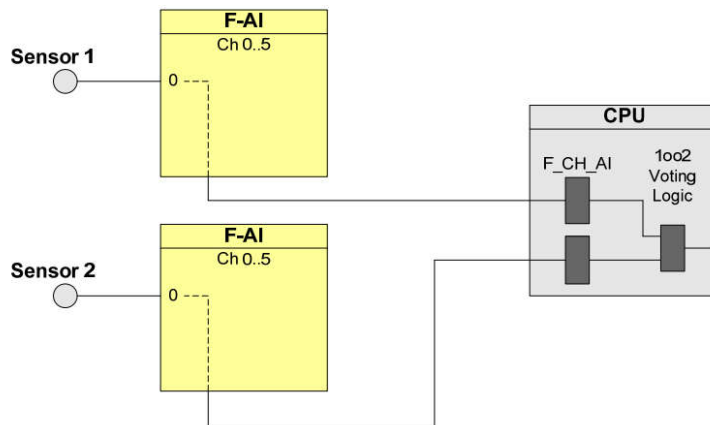
- انتخاب ۱: مطابق شکل ۱- هر دو سنسور به یک ماژول F-AI سیم بندی می شود. در این شکل یک سنسور به کانال ۰ و سنسور دیگر به کانال ۲ ماژول F-AI سیم بندی می شود.
- انتخاب ۲: اولین سنسور به یک کارت F-AI، و دومی سنسور به یک کارت F-AI دوم سیم بندی می شود. همانطور که در شکل ۲- نشان داده شده. در این شکل یک سنسور به کانال ۰ اولین کارت F-AI و سنسور دوم به کانال ۰ از دومین کارت F-AI سیم بندی می شود.



شکل ۴-۸- انتخاب ۱- ارزیابی 1oo2 در برنامه لاجیک با یک ماژول F-AI

پیکرندی مطابق شکل انتخاب اول برای رسیدن به SIL3 مناسب می باشد. اگر یک سنسور یا ماژول F-AI فالت دهد، تابع واکنش خطا عملکرد ایمنی (safety function) را فراهم می کند (از طریق سیستم failsafe). جدول زیر زمان فعال شدن تابع واکنش خطا (error reaction function) را نشان می دهد.

Component has failed ?			Error reaction function has been triggered?
Sensor 1	Sensor 2	F-AI	
no	no	no	no
X	yes	X	yes
yes	X	X	yes
X	X	yes	yes



انتخاب ۲- ارزیابی 1002 در برنامه لاجیک با استفاده از دو کارت F-AI

پیکربندی انتخاب دوم برای رسیدن به SIL3 مناسب می باشد.

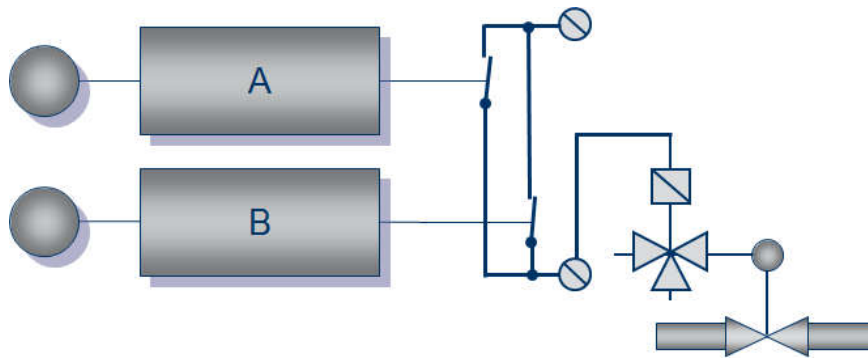
اگر یک سنسور یا یکی از ماژول های F-AI فالت دهد، تابع واکنش خطا، عملکرد ایمنی (safety function) را فراهم می کند (از طریق سیستم failsafe). جدول زیر زمان فعال شدن تابع واکنش خطا (error reaction function) را نشان می دهد.

Component has failed ?				Error reaction function has been triggered?
Sensor 1	Sensor 2	F-AI 1	F-AI 2	
no	no	no	no	no
X	X	X	yes	yes
X	X	yes	X	yes
X	yes	X	X	yes
yes	X	X	X	yes

۴-۴-۴ معماری سیستم 2 Out of 2 (2002)

سیستم دو کانال (Dual Channel system)

- خروجی کانالها به صورت موازی سیمی بندی می شوند. (2 Out of 2 : 2002)
- برای انجام عمل قطع یا خاموش نمودن تجهیز، انرژی هر دو کانال باید قطع شود.
- هنگامی که هر دو کانال یا زیرسیستم از کار بیافتند، سیستم دیگر کار نمی کند (1v2-1 von 2)
- در زمانی که تنها یکی از دو زیر سیستم عملیاتی است سیستم فعال باقی می ماند

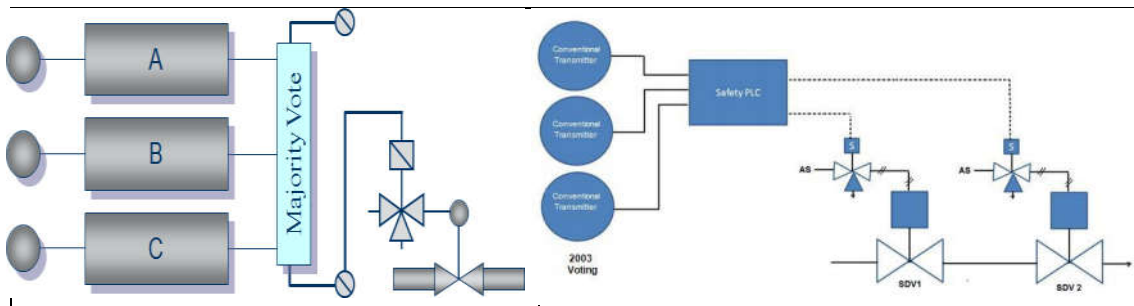


شکل ۴-۹- معماری سیستم (2 Out of 2 2002)

۴-۴-۱ معماری سیستم 2 Out of 3 (2003)

سیستم سه نسخه (TMR)

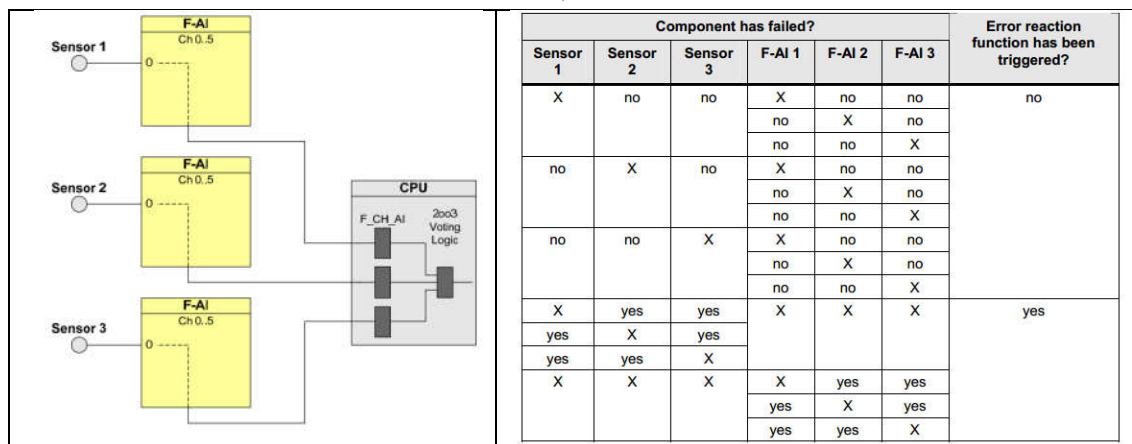
هنگامی که دو تا از سه زیرسیستم نقص داشته باشند سیستم دیگر عملیاتی نمی باشد. (3 von 1) یعنی سیستم تنها از دست دادن یکی از سه کانال را تحمل می کند و اگر دو تا از سه زیرسیستم عمل کند، فعال باقی می ماند.



شکل ۴-۱۰-۱- معماری سیستم 2 Out of 3

معماری سه سنسور (یا طرح ارزیابی 2003) به کاربردهایی اشاره می کند که برای رسیدن به سطح SIL موردنیاز به سه سنسور برای افزایش در دسترس بودن نیاز دارد. ارزیابی 2003 بدان معنی است که دو سنسور از سه سنسور تریپ دهد، مدار تریپ فعال می شود. اگر حداقل دو تا از سنسورها تریپ دهد. برنامه لاجیک ایمنی تریپ خواهد شد.

در سیستم F زمینس، معماری پایه 2003 با ارزیابی در برنامه لاجیک، از سه سنسور و سه F-AI استفاده می کند. در شکل ۵-۱ یک بلوک دیاگرام از این پیکربندی نشان داده شده است. در این شکل هر سنسور به کانال ۰ یک ماژول F-AI نصب شده در ET200M، سیم بندی شده است.



شکل ۴-۱۱- پیاده سازی معماری ارزیابی 2003 در سیستم F زمینس

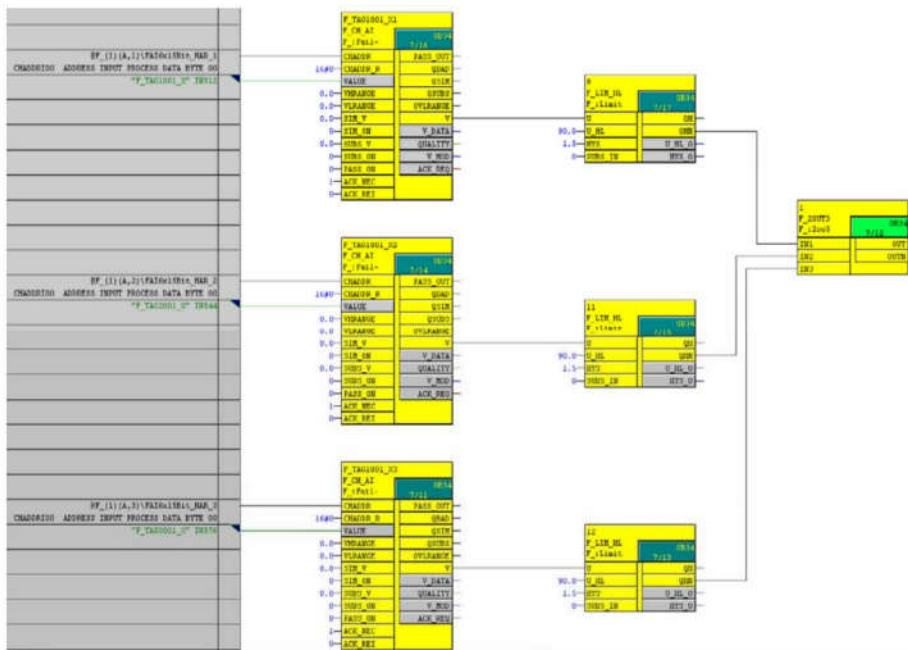
اگر دو سنسور یا دو ماژول F-AI شکست بخورد، تابع واکنش خطا از طریق سیستم failsafe عملکرد ایمنی را فراهم می کند.

توجه شود که با توجه به انعطاف پذیری سیستم، ساختارهای دیگری نیز امکان پذیر است. که با توجه به در دسترس بودن ماژولها در رک ET 200M، متفاوت از نمونه توصیف شده می باشد. برای مثال:

- برای کاربردهای در دسترس پذیری پایین (*Lower availability*)، تمامی هر سه سنسور به یک ماژول متصل می شود.
- برای کاربردهای در دسترس پذیری یکسان ماژول ها (*Similar availability of the modules*)، هر سه سنسور به دو ماژول که با هم به صورت افزونه هستند، متصل می شود. هر دو ماژول در یک رک *ET200M* نصب شده اند.
- برای کاربردهای در دسترس پذیری بالا در ماژول ها و رک های *ET200M* (*High availability*) هر سه سنسور به دو ماژول که باهم به صورت افزونه هستند، و دو ماژول در دو رک *ET200M* متفاوت نصب شده اند، متصل می شود.
- برای کاربردهای در دسترس پذیری بالا در ماژول ها و رک های *ET200M* (*High availability*)، هر سنسور به یک ماژول متصل می شود. ماژول ها در رک های *ET200M* متفاوت نصب شده اند.

۴-۴-۲- لاجیک ارزیابی 2003 بدون در نظر گرفتن خطای کانال

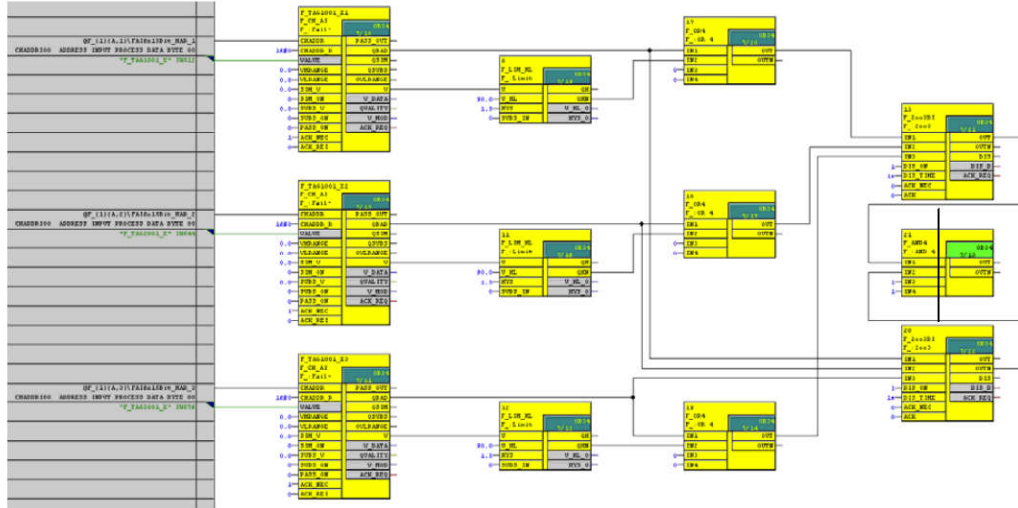
شکل زیر یک مثال از پیکر بندی ارزیابی 2003 را در محیط ویرایشگر *CFC* نشان می دهد. که در آن خطای کانال (پایه خروجی *BAD* از بلاک درایور کانال آنالوگ) در نظر گرفته نشده است. توجه شود که در این مثال یک مقدار حد ماکزیمم فرض کرده و خروجی لاجیک ارزیابی برای دستیابی به وضعیت *failsafe* به صفر سوئیچ می کند. ($normal\ state = 1, failsafe\ state = 0$)



شکل ۴-۱۲- نمونه لاجیک ارزیابی 2003

۴-۴-۳- پیکربندی 2003 با در نظر گرفتن خطای کانال (2003D)

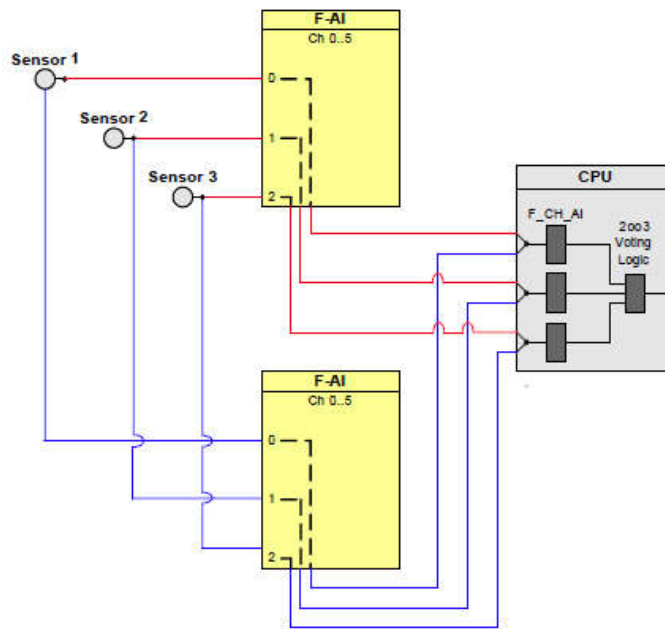
شکل زیر یک پیکربندی نمونه از ارزیابی 2003D (With Diagnostic) در ویرایشگر CFC را نشان می دهد. که در آن خطای کانال در لاجیک ارزیابی در نظر گرفته شده است. توجه شود که در این مثال یک در ارزیابی مقدار سنسور آنالوگ، یک مقدار حد MAX فرض شده و خروجی لاجیک ارزیابی برای دستیابی به وضعیت failsafe به صفر سویچ می کند. (normal state = 1, failsafe state = 0)



شکل ۴-۱۳- نمونه لاجیک ارزیابی 2003D

۴-۴-۴- معماری استفاده از سه سنسور با دو ماژول افزونه (Three sensors with redundant F-AI)

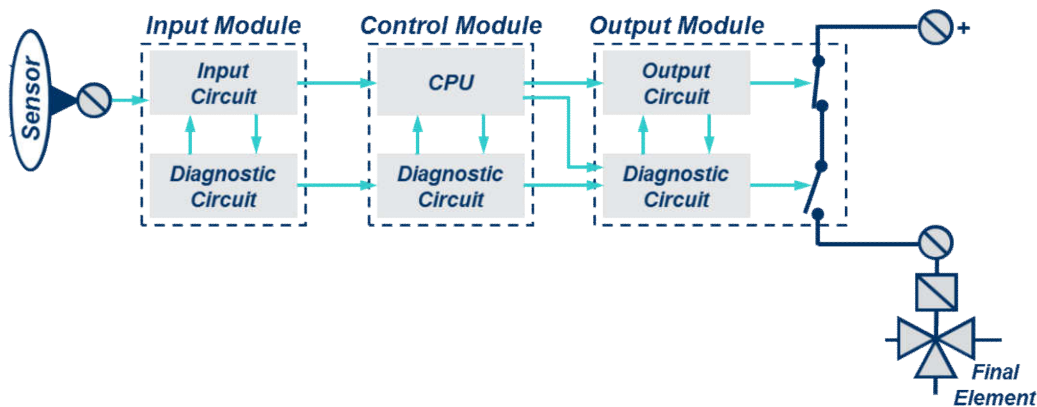
در این معماری که ارزیابی (2003) نامیده می شود. سه سنسور به دو ماژول F-AI افزونه متصل می شود. همانند معماری قبلی این طرح ارزیابی 2003 نیز به کاربردهایی اشاره می کند که در آنها برای رسیدن به سطح ایمنی موردنیاز به دو سنسور نیاز دارد. سنسور سوم در دسترس بودن (availability) را افزایش می دهد. به طوری که دو تا از سه سنسور بایستی کار بکنند. اگر حداقل دو تا از سه سنسور به شرایط تریپ برود، لاجیک تریپ را فعال خواهند کرد.



شکل ۴-۱۴- اتصال دو سنسور بخ دو ماژول ریداندانت در معماری 2003

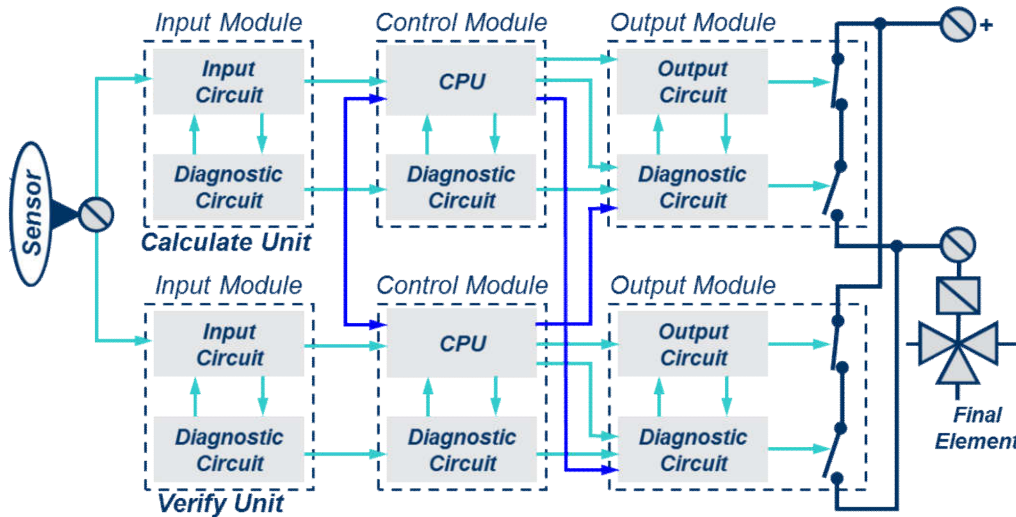
۴-۴-۵- معماری 1oo1D (with diagnostics – Non Redundant Safety System)

این معماری مطابق شکل زیر یک معماری تک کانال با قابلیت خودآزمون (Self-Tests) جامع می باشد. که دارای یک مسیر ثانویه Shutdown کنترل شده با مدار diagnostics می باشد. معماری 1oo1D برای تبدیل شکست های خطرناک به شکست های امن (safe failures) از طریق قطع انرژی خروجی (de-energizing) از مدار تشخیص خاص بهره گیری می کند. مدار I/O طراحی شده برای عملکرد به صورت خرابی امن دارای قابلیت تشخیص یکپارچه (built-in diagnostics) می باشد. پردازنده نیز دارای مدار تشخیص برای حافظه، عملیات CPU، تایمر watchdog و کلیه سیستم های ارتباطی می باشد.



۴-۴-۶ معماری 1oo2D (with diagnostics)

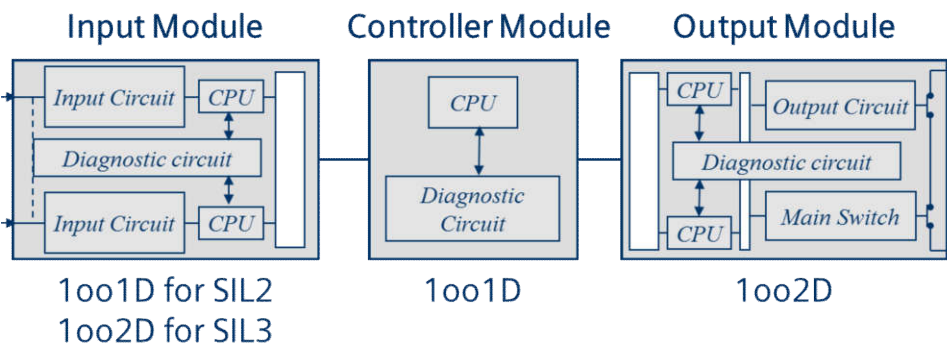
این معماری مطابق شکل زیر یک سیستم دو کانال (1oo2D) با قابلیت خودآزمون (Self-Tests) جامع می باشد. که دارای یک مسیر ثانویه Shutdown کنترل شده با مدار diagnostics می باشد.



در این معماری همان شکست های موجود در معماری IOID نیز وجود دارد. با این تفاوت که ماژول های IO به صورت پیکربندی افزونه بسته می شوند. اگر مدار تشخیص، یک شکست در یک کانال از سیستم (ورودی/ پردازش گرها/ خروجی) را تشخیص دهد. طرف دیگر کنترل را حفظ و سیستم در دسترس باقی می ماند.

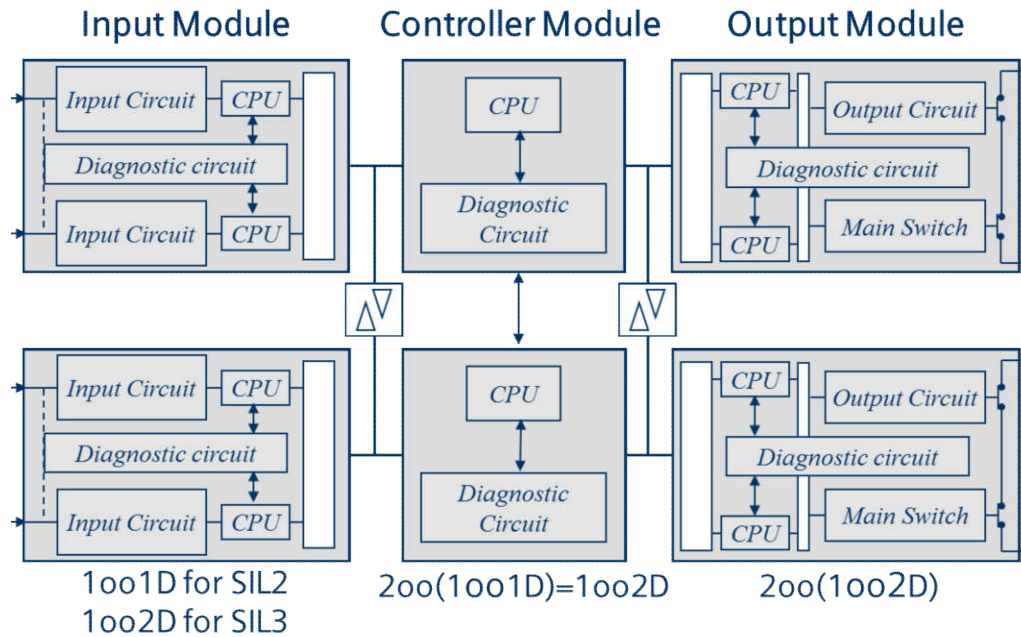
۴-۴-۷ معماری طراحی سیستم S7-400F (Non-Redundant Setup)

- Single or Dual Sensors for inputs
- Redundant circuitry within I/O modules
- Redundant output circuits



۴-۴-۸ معماری طراحی سیستم S7-400HF (Fully-Redundant Setup)

- Single, Dual, triple or quad Sensors for inputs
- Quad output circuits



۴-۵- مثال هایی از پاسخ سیستم تحمل خطا (Fault-Tolerant) به یک فالت

۴-۵-۱- مثال ۱- توقف CPU یا ماژول تغذیه

وضعیت اولیه: سیستم S7-400H در حالت افزونه می باشد.

۱- با خاموش کردن ماژول تغذیه CPU0 را به حالت توقف می بریم.

نتیجه: LED های REDF, IFM1F و IFM2F در CPU1 روشن می شوند. ماژول CPU1 به حالت عملکرد

تکی (solo mode) می رود و اجرای برنامه کاربر ادامه می یابد.

۲- منبع تغذیه رک صفر مربوط به CPU0 را دوباره روشن کنید.

نتیجه:

- CPU0 عمل LINK-UP و UPDATE را به صورت خودکار انجام می دهد.
- CPU0 به حالت Run می رود و به عنوان CPU آماده به کار (standby) عمل می کند.
- حال سیستم S7-400H در مد سیستم افزونه می باشد.

۴-۵-۲- مثال ۲- خطا یا نقص در کابل فیبر نوری (Failure of a fiber-optic)

وضعیت اولیه: سیستم S7-400H در مد افزونه می باشد. سویچ انتخاب مد عملیاتی در هر دو CPU در

مد RUN یا RUN-P می باشد.

۱- کانکتور یکی از کابل های فیبر نوری را قطع کنید.

نتیجه: LEDهای REDF و IFM1F یا IFM2F (بسته به این که کدام کابل فیبر نوری قطع می شود) در هر دو CPU روشن می گردد. CPU اصلی (CPU0) به حالت تکی سوئیچ می کند و برنامه کاربر همچنان اجرا می شود.

۲- کابل فیبر نوری که قطع شده بوده دوباره وصل کنید.

۳- CPU آماده به کار (CPU1) را که در حال حاضر در مد STOP می باشد، با استفاده از STEP7 راه اندازی مجدد کنید.

نتیجه:

- CPU1 عمل LINK-UP و UPDATE را به صورت خودکار انجام می دهد.
- سیستم S7-400H به حالت افزونه می رود.

۴-۶- انواع سیستم F

برای سیستم های S7 دو نوع سیستم ایمنی (Safety Systems) یا بسته نرم افزار توسعه برنامه ایمنی وجود دارد. که عبارت انداز:

- Distributed Safety برای محیط های اتوماسیون کارخانه
- S7 F Systems برای محیط های اتوماسیون فرآیند

این دو دارای مشترکات و تفاوت هایی هستند. برای محاسبات زمان مانیتورینگ f برای هر دو این سیستم ها از سوی زیمنس دو فایل اکسل معرفی شده است.

فایل S7ftimeb.xls برای S7 F Systems

فایل S7fcotib.xls برای Distributed Safety

۴-۷- نصب بسته F System

برای پیاده سازی برنامه یا لاجیک F در یک پروژه PCS7 بایستی بسته های نرم افزاری زیر روی سیستم ایستگاه مهندسی (ES) نصب گردد:

- S7 F Configuration Pack V5.5 SP9 Upd. 1
- S7 F Systems V6.1 SP1
- S7 F Systems Lib V1_3 SP1 Upd.1

نکته: بسته S7 F Systems V6.1 SP1 را می توان با کتابخانه F نسخه Lib V1_2 نیز به کار برد.

۴-۷-۱- نصب بسته F system مربوط به HMI

در صورتی که از SDW یا MOS استفاده می شود، بایستی بسته S7 F Systems HMI V6.1 SP1 روی کامپیوتر OS نصب شود.

- *SDW : Safety Data Write*
 - *MOS : Maintenance Override Switch*
- هم چنین بسته *Safety Matrix Viewer V6.2 SPI* نیز روی *PC* مربوط به *OS* نصب شود.

۴-۸- تنظیمات پیشرفته برای سیستم های FH در PCS7 (Advanced PCS 7 ES settings)

۴-۸-۱- مکانیزم های ایمنی

برای پیاده سازی توابع ایمنی به صورت *Fail safe* در سیستم *S7* در *CPU* های *S7* دو نوع مکانیزم ایمنی پیش بینی شده است. که عبارت است از:

- حفاظت از برنامه کاربر در مقابل دسترسی های غیرمجاز به سیستم *F*، که این کار از وقوع خطاهای عمدی و سهی پیشگیری می کند.
- اجرای توابع خودآزمون (*SelfTest*) در *CPU* که به تشخیص خطا در ماژول ها کمک می کند

۴-۸-۲- حفاظت از دسترسی (Access Protection)

یک ایستگاه *S7-400F/FH* که به عنوان یک سیستم *safety (SIS or ESD)* عمل می کند توسط دو تا پسوورد محافظت می شود.

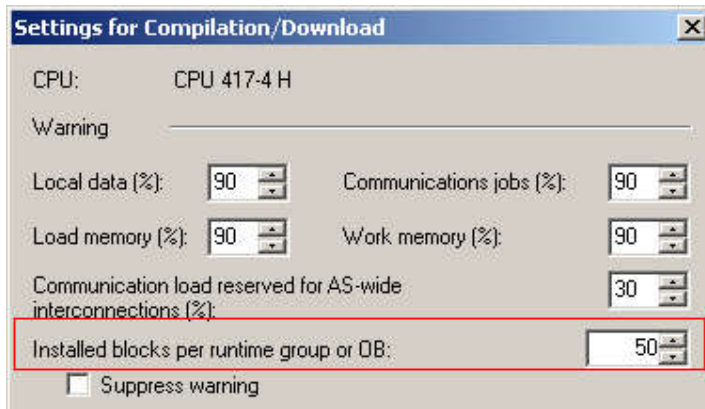
۱. یکی، پسوورد دسترسی به *CPU* می باشد. که با هدف محافظت از پارامترهای *CPU* در مقابل دانلود تصادفی یا دانلود اشتباهی برنامه می باشد. این پسوورد در *HWConfig* پیکربندی می شود.
۲. پسوورد دیگر، برای ویرایش برنامه *F* می باشد که در پنجره *Edit Safety Program* پیکربندی می شود. به طوری که از برنامه یا لاجیک *fail-safe*، پارامترهای پیکربندی شده برای ماژول های سیگنال و پارامترهای *fail-safe* ماژول *CPU* (گزینه ای در سربرگ *Protection* مربوط به *CPU*) محافظت می کند.

نکته: در خصوص *PCS 7 V7.0* و بالاتر با *SIMATIC logon*، امکان تعریف رمز عبور برای حفاظت از پروژه *PCS 7* نیز وجود دارد. دسترسی به *ES* نیز توسط یک رمز عبور در سطح سیستم عامل (ورود به ویندوز، محافظ صفحه نمایش) محافظت می شود.

۴-۸-۳- تنظیمات CFC برای کامپایل و دانلود

مطابق شکل زیر برای کامپایل صحیح چارت های *CFC* دارای بلوک های *F*، مقدار آستانه هشدار مبتنی بر تعداد بلوک ها به ازای هر *Runtime group* را تنظیم کنید. مقدار پیش فرض ۵۰ است.

CFC Editor > Options > Cusomize > Compile/download



۴-۹- پیکربندی سخت افزار S7F/FH

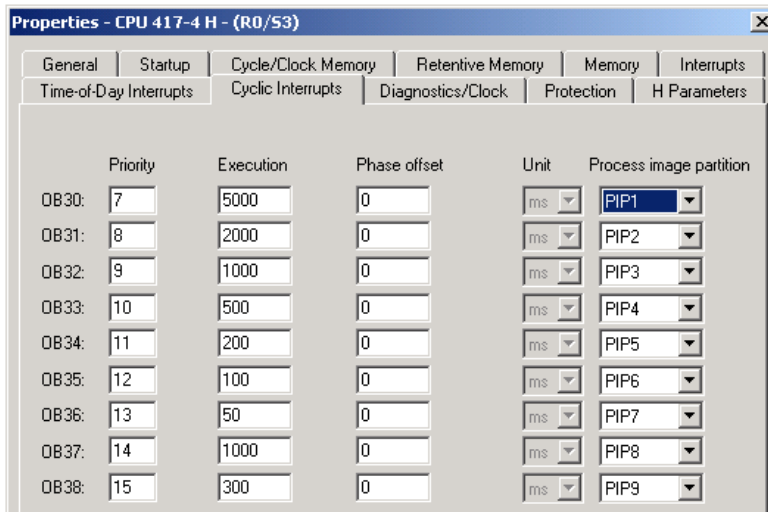
انواع ماژول های CPU نوع H که برای بکارگیری به عنوان سیستم S7 F در PCS7 ارائه شده است، عبارت انداز:

- CPU 412H , CPU 414H, CPU 416H, CPU 417H

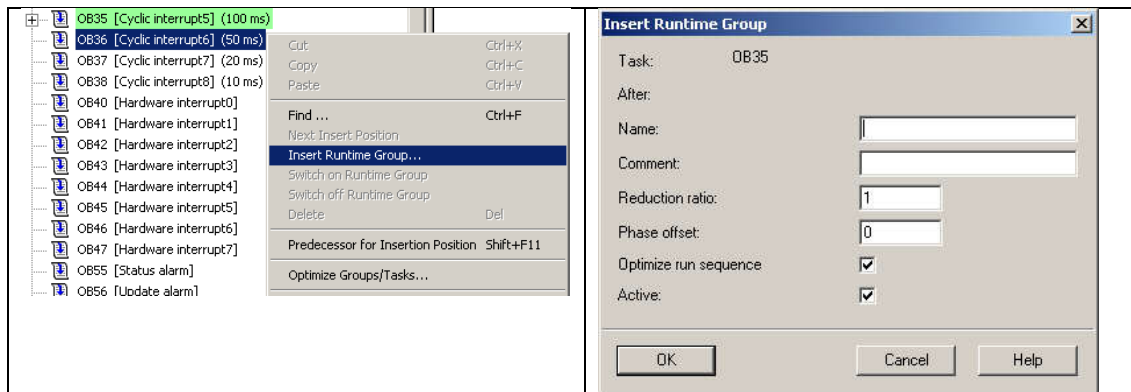
۴-۱۰- تطبیق پارامترهای CPU

۴-۱۰-۱- سربرگ Cyclic interrupts

برای بخش هایی از برنامه لاجیک CPU که از نوع F هستند، نیازی به پیکربندی پارتیشن های Process image در پنجره پراپرتی CPU نمی باشد. به این دلیل که داده استخراج شده توسط ماژول های سیگنال F در سیستم های نوع S7 F/FH از طریق حافظه process image مورد دسترسی واقع نمی شوند. بلکه دسترسی به آنها به صورت مستقیم از طریق درایور ماژول F صورت می گیرد. با این حال، اگر قطعات برنامه استاندارد در OB های سیکلیک مورد استفاده قرار می گیرد. تعریف بخش های Process image در سربرگ Cyclic Interrupts نیاز می باشد.

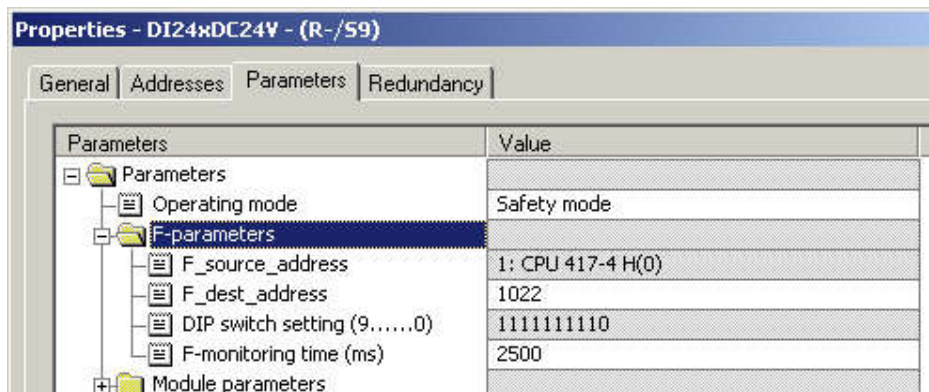


نکته: پس از درج یک *Runtime group* در پنجره *Run Sequence* گزینه های *Reduction ratio* و *Phase offset* مرتبط با *OB* های وقفه سیکلک در برنامه *F* پیکربندی نمی شود.



۴-۱۱- تنظیمات پارامترهای ماژول های سیگنال F

شبهه ماژول های سیگنال استاندارد، ماژول های سیگنال *F* نیز در محیط *HW Config* پیکربندی می شوند. برای این کار بایستی بسته *F Configuration Pack* نصب شود. در سیستم *F* کانال های استفاده نشده را در زمانی که سیستم کار می کند، را می توان به برنامه اضافه کرد. مشروط بر این که در طول اولین راه اندازی، در *HW Config* فعال شده و به منظور جلوگیری از صدور فالت، کانال مربوطه به یک مقاومت مجهز شده باشد. همین که ماژول های سیگنال *F* را به ایستگاه *ET200M* در *HwConfig* اضافه کردید، با دوبار کلیک روی ماژول سیگنال و یا از طریق منوی *Edit > Object Properties* می توان به پنجره پیکربندی ماژول *F* دسترسی داشت.



شکل ۴-۱۵- پنجره پراپرتی یک ماژول سیگنال *FDI*

▪ مد کاری (*Operating mode*):


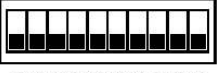
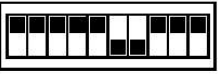
مطابق شکل بالا جهت عملکرد به عنوان ماژول F نه به عنوان ماژول استاندارد، بایستی مد کاری *Safety* برای ماژول های سیگنال F دیجیتال فعال شود. برای ماژول های هارت آنالوگ F این گزینه وجود ندارد.

▪ آدرس های **PROFIsafe**:

آدرس های مبدا و مقصد پروتکل *PROFIsafe* جهت ارتباط *PROFIsafe* بین *CPU* و ماژول می باشد. که عبارت است از F_SOURCE_ADD و F_DEST_ADD . این آدرس ها به صورت منحصر به فرد برای شناسایی مبدا و مقصد در هنگام برقراری ارتباط *PROFIsafe* استفاده می شود. آدرس F_DEST_ADD مقصد *PROFIsafe* (ماژول) را شناسایی می کند. این آدرس باید، در هر دو، یعنی شبکه و ایستگاه منحصر به فرد باشد. آدرس F_SOURCE_ADD به طور دائمی به *CPU* اختصاص داده می شود. برای جلوگیری از وقوع اشتباهات در انتساب پارامتر، آدرس های مبدا و مقصد پروفی سیف در پارامترهای ماژول F به طور خودکار اختصاص داده می شوند.

۴-۱۱-۱- تنظیمات سخت افزاری (DIP switch settings)

آدرس های تولید شده در بخش *Parameters* ماژول بایستی به صورت سخت افزاری در کارت تنظیم شوند. برای تنظیم آدرس F_DEST_ADD ، در هر یک از ماژول های F یک دیپ سویچ تعبیه شده است، که یک نمایش باینری از همان آدرس می باشد. از آن جایی که این دیپ سویچ در پشت ماژول است، این تنظیم بایستی قبل از نصب ماژول سیگنال F بر روی رک انجام شود.

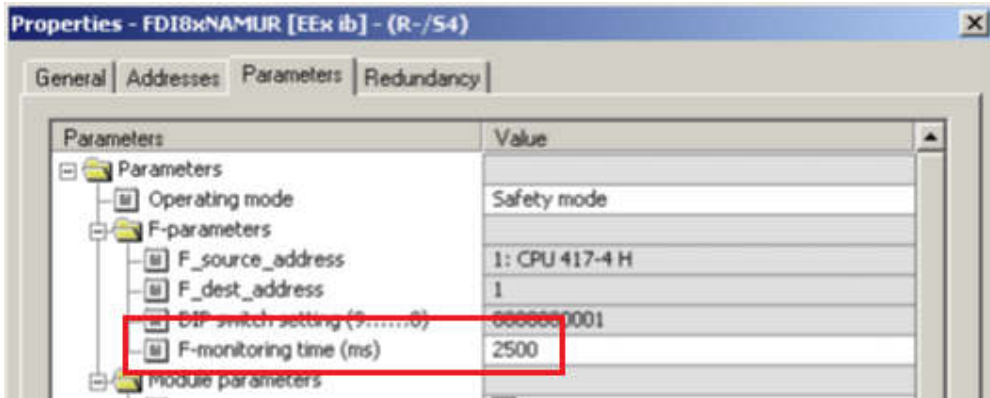
Standard mode	Safety mode
 <p>ON</p> <p>4096 2048 1024 512 256 128 64 32 16 8</p> <p>or</p>  <p>ON</p> <p>4096 2048 1024 512 256 128 64 32 16 8</p>	<p>All possible combinations that do not correspond to standard mode</p>  <p>ON</p> <p>4096 2048 1024 512 256 128 64 32 16 8</p> <p>9 8 7 6 5 4 3 2 1 0</p> <p>Module start address 192 corresponds to F_DEST_ADD 24</p>

زمانی که این ماژول ها در مد استاندارد کار می کنند. آدرس F_DEST_ADD همواره به مقدار صفر (*delivery condition*) تنظیم می شود.

در مورد نسخه های قدیمی ماژول های نوع F آدرس F_DEST_ADD وابسته به آدرس شروع ماژول می باشد. که عبارت است از: $F_DEST_ADD = \text{Module start address} / 8$

۴-۱۱-۲- زمان مانیتورینگ F

اگر ماژول های سیگنال F در مد کاری Safety کار می کنند، بایستی در پنجره پراپرتی ماژول F، زمان مانیتورینگ F، برای برقراری ارتباطات Safety بین F CPU و ماژول سیگنال F تنظیم شود. (زمان مانیتورینگ PROFIsafe). مقدار پیش فرض این زمان ۲۵۰۰ میلی ثانیه است. که معمولاً از همین مقدار استفاده می شود.



شکل ۴-۱۶- زمان مانیتورینگ ارتباطات F در Profisafe

1. Diagnostic interrupt

برای فعال کردن قابلیت گزارش وقفه در بلوک های داریور PCS7، وقفه تشخیصی برای ماژول سیگنال F باید همیشه در حالت safety فعال شود.

رویدادهای مختلف خطا، که ماژول سیگنال خرابی امن می تواند با استفاده از تابع تشخیص تعریف کند، یک وقفه تشخیصی را تریگر می کند. رویدادهای تشخیصی که رخ می دهد توسط ماژول F CPU در دسترس قرار می گیرد.

2. Group diagnostics

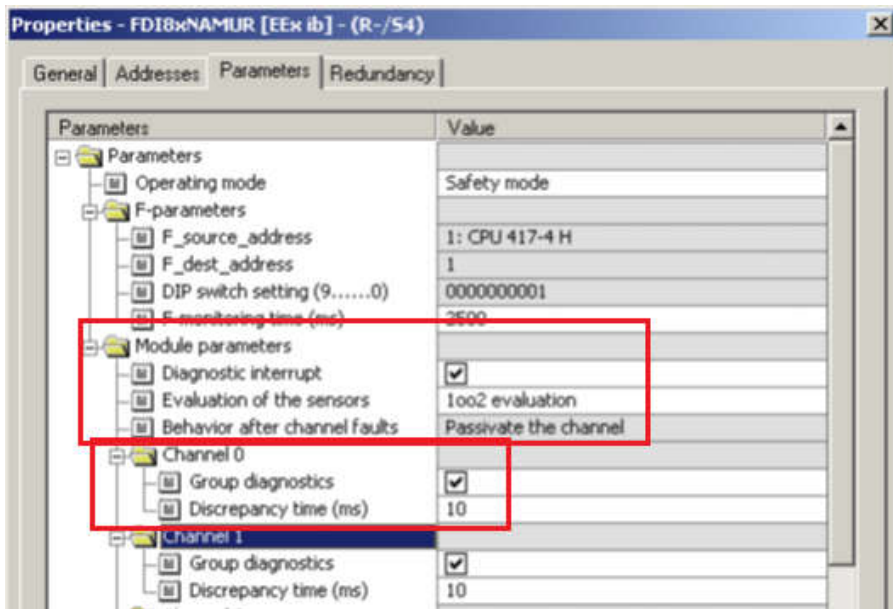
اگر مطابق شکل زیر باکس مربوط به یک کانال خاص را تیک بزنید، وقوع یک رویداد خاص یک کانال (مانند قطعی سیم) یک واکنش خطا در برنامه امنیتی را تریگر می کند (مقدار جایگزین در بلاک داریور کانال فعال شده و QBAD ست می شود). اگر گزینه «Enable diagnostics interrupt» انتخاب شود، یک اینترپت تشخیصی در CPU تریگر شده و مطابق آن یک از نوع process control message به سیستم OS ارسال می شود.

پارامتر Group diagnostics برای فعال و غیرفعال کردن پیام های channel-specific diagnostic (به عنوان مثال قطع سیم یا اتصال کوتاه) روی ماژول های سیگنال F به CPU استفاده می شود. در ماژول های سیگنال F هر موقع که کانال فعال شود (یعنی Spare نباشد)، گزینه group diagnostics نیاز است که فعال شود.

برخی از ماژول های ورودی/خروجی نوع F زیرممنس عبارت انداز:

- SM 326; DI 8 x NAMUR (order no. 6ES7326-1RF00-0AB0)
- SM 326; DI 24 x DC 24 V (order no. 6ES7326-1BK00-0AB0)

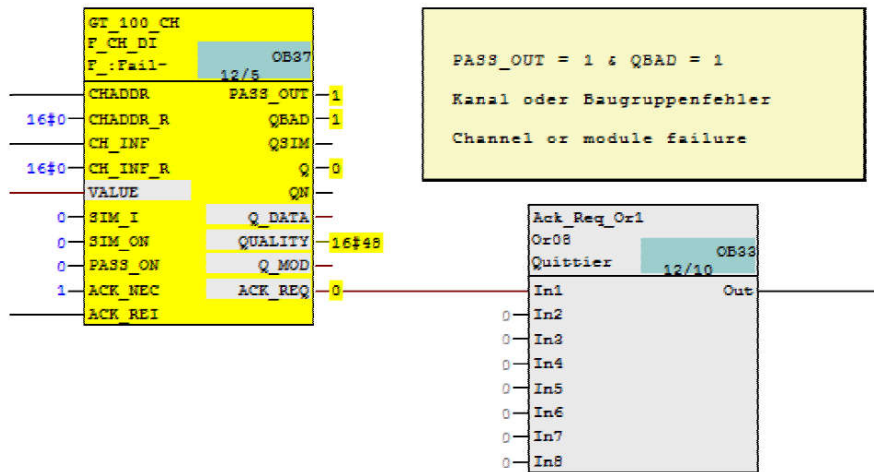
- SM 336; DO 10 x DC 24V/2A (order no. 6ES7326-2BF00-0AB0)
- SM 326; AI 6 x 13 Bit (order no. 6ES7336-1HE00-0AB0)



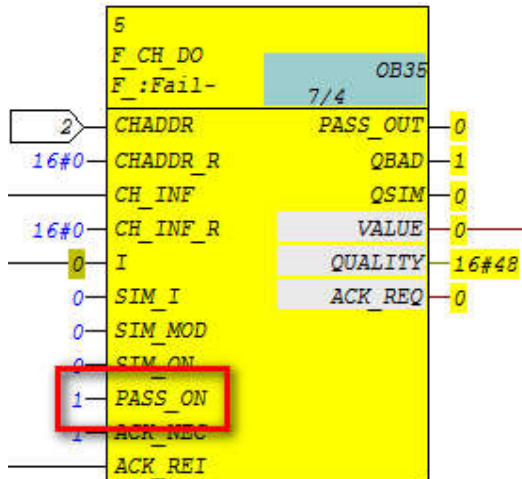
شکل ۴-۱۷- فعال سازی قابلیت Diagnostic Interrupt در ماژول FDI

۴-۱۱-۳- مفهوم Passivation در حالت کلی (غیر فعال شدن کانال/ماژول های ورودی/خروجی)

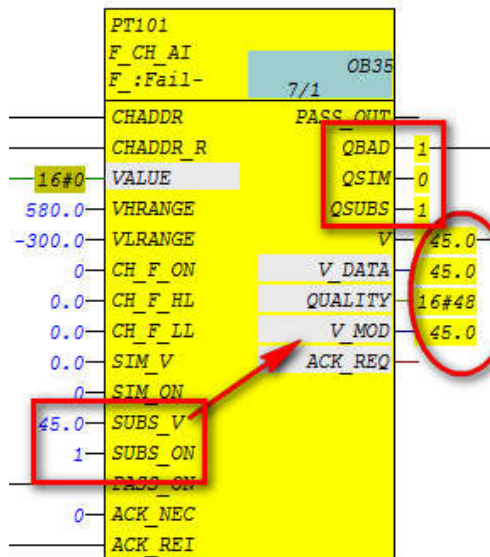
Passivation بدان معنی است که در صورت وقوع یک خطا، یک یا تعدادی از کانال های یک ماژول سیگنال F ، به وضعیت امن (منطق صفر) (*safe state*) سویچ می کند. در صورت وقوع خطا در یک کانال (برای مثال معیوب شدن یک سنسور)، تنها کانال های آسیب دیده پسیو/غیر فعال می شوند. در صورت وقوع خطا در ماژول (به عنوان مثال یک خطای ارتباطی)، تمام کانال ها بر روی ماژول $F-I/O$ پسیو می شوند. اگر یک ماژول سیگنال F خطایی تشخیص دهد، کانال های آسیب دیده و یا همه کانال های آن را به حالت امن سوئیچ می کند. به عبارت دیگر، کانال های این ماژول غیر فعال می شوند. ماژول سیگنال F برای نشان دادن این که خطایی شناسایی شده است، یک پیام به درایور کانال F و $PCS 7 OS$ می فرستد.



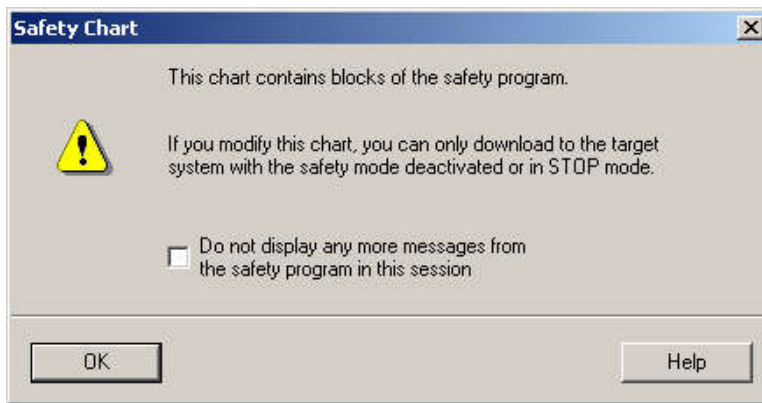
همچنین از ورودی $PASS_ON$ در یک بلاک درایور کانال نیز می توان برای فعال و غیرفعال کردن $passivate$ شدن کانال در برنامه F استفاده کرد. به عنوان مثال، با استفاده از یک شرایط خاص در ترتیب زمانی برنامه ($program\ sequence$) و یا راه اندازی مجدد حفاظت ($restart\ protection$). هنگامی که کانال های خروجی پسیو می شوند، تغذیه خروجی ها قطع می شود ($de-energized$) (به جریان صفر و یا ولتاژ صفر تنظیم می شوند). بلاک درایور یک کانال خروجی دیجیتال F پسیو شده، یک مقدار جایگزین ($substitute$) با کد کیفیت $16\#48$ را صادر و خروجی $QBAD$ را به 1 تنظیم می کند.



بلاک درایور یک کانال ورودی دیجیتال F پسیو شده، مقدار جایگزین 0 را با کد با کیفیت $16\#48$ در خروجی قرار داده و خروجی $QBAD$ را یک می کند. با توجه به انتساب مقدار در پارامتر ورودی $SUBS_ON$ ، درایور یک کانال ورودی آنالوگ F ، هم یک مقدار جایگزین با کد کیفیت $16\#48$ و یا آخرین مقدار معتبر با کد با کیفیت $16\#44$ را در خروجی قرار داده و خروجی $QBAD$ را نیز به یک تنظیم می کند. در صورتی که یک مقدار جایگزین وارد شده در ورودی $SUBS_V$ در خروجی بلاک ظاهر شود، در آن صورت خروجی $QSUBS$ نیز یک می شود.



در صورتی که یک چارت *F-CFC* برای اولین بار باز می شود، پیغام زیر نمایش داده می شود. که محتوای پیغام این است که عمل داندلود چارت ها به *CPU* تنها در مد *Safety* غیرفعال امکان پذیر است.



۴-۱۱-۴ - پسیو شدن گروهی (Group passivation)

اگر در حین پسیو شدن یک ماژول *F-I/O* یا یک کانال *F-I/O* بخواهید ماژول های *F-I/O* را پسیو کنید، برای این کار می توان از خروجی *PASS_OUT* و ورودی *PASS_ON* جهت پسیو کردن گروهی استفاده کرد. عمل پسیو کردن گروهی از طریق ورودی/خروجی های *PASS_OUT/PASS_ON* به عنوان مثال برای *Force* کردن عمل *reintegration* همزمان تمام *F-I/O* ها بعد از این که سیستم *F* راه اندازی گردید، می تواند مورد استفاده قرار گیرد.

برای فعال کردن *Group passivation* بایستی تمام پایه های خروجی *PASS_OUT* بلاک های درایور کانال *F* در یک گروه را توسط یک بلاک *F_OR4* به هم متصل نموده و خروجی *OUT* از بلاک *F_OR4* را به ورودی های *PASS_ON* بلاک های درایور کانال در این گروه متصل کنید.

۴-۱۱-۵- فعال کردن کانالها (Activating channels)

به واسطه ساختار ماژول های سیگنال F ، بدون $Passivate$ کردن ماژول، امکان ایجاد تغییرات در پیکربندی سخت افزار یا دانلود به آنها وجود ندارد. $Passivate$ کردن ماژول های خروجی شامل ایجاد یک وضعیت ایمن ($safe\ state$) روی تمام خروجی ها (صفر کردن مقادیر خروجی) می باشد. ولی $passivate$ کردن ماژول های ورودی شامل صفر کردن خروجی بلاک های درایور می باشد.

برای داشتن امکان فعال کردن کانال های استفاده نشده ($free\ channels$) در ماژول های سیگنال F در حین کار، بایستی کانالها را در $HW\ Config$ فعال کرد. ولی با توجه به قابلیت تشخیص ($diagnostics$) در ماژول های سیگنال F ، کانال های فعال شده که بدون استفاده هستند، منجر به خطا می شوند، که می توانید با تجهیز کانال به یک مقاومت معادل از بروز خطا جلوگیری کرد.

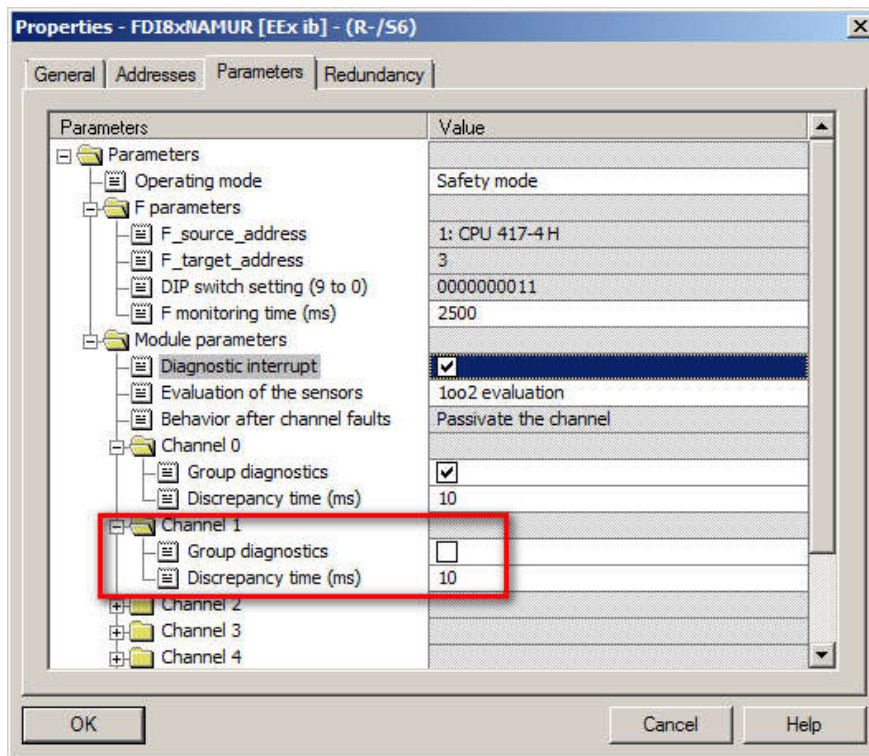
▪ پروسه فعال کردن ماژول سیگنال ورودی دیجیتال در حین کار

$SM326; DI\ 8\ x\ NAMUR\ [EEx\ ib]\ (6ES7\ 326-1RF00-0AB0 / 6ES7\ 326-1RF01-0AB0)$

۱. مد کاری ($Safety\ mode$) را تعریف کنید.
۲. پارامترهای مد کاری را تعریف کنید
۳. پارامترهای ماژول را تعریف کنید.
۴. گزینه $Group\ Diagnostics$ را برای کانال فعال کنید.
۵. کانال را به یک مقاومت $1\ kohm$ متصل کنید.

مطابق شکل زیر می توان با برداشتن تیک مربوط به $group\ diagnostics$ در یک کانال، آن کانال را غیرفعال کرد.





▪ **پروسه فعال کردن ماژول سیگنال خروجی دیجیتال در حین کار**

SM326; DO 10 x DC 24V/2A (6ES7 326-2BF01-0AB0)

۱. تعیین مد کاری (Safety mode)،
۲. تعریف پارامترهای ماژول،
۳. فعال کردن گزینه *Group Diagnostics* برای کانال،
۴. برای شبیه سازی یک عملگر (actuator) خروجی کانال را به یک مقاومت (مثال 2.7 kilohms) و سر دیگر مقاومت را به زمین متصل کنید.

▪ **پروسه فعال کردن ماژول سیگنال خروجی دیجیتال F در حین کار**

SM326; F-DO 10 x DC 24V/2A PP (6ES7 326-2BF10-0AB0)

۱. تعریف پارامترهای ماژول،
۲. فعال کردن گزینه *Group Diagnostics* برای کانال،
۳. برای شبیه سازی یک عملگر (actuator) خروجی کانال را به یک مقاومت (مثال 2.7 kilohms) و سر دیگر مقاومت را به زمین متصل کنید.

▪ **پروسه فعال کردن ماژول سیگنال ورودی آنالوگ F در حین کار**

SM 336; F-AI 6 x 13 Bit (6ES7 336-1HE00-0AB0)

۱. تعریف پارامترهای ماژول،
۲. فعال کردن گزینه *Group Diagnostics* برای کانال،

۳. انجام تنظیمات خاص کانال
۴. ورودی مثبت (*plus*) کانال را از طریق یک مقاومت (به عنوان مثال $3.9\ kilohms$) به منبع ولتاژ ($L+$) و پایه منفی کانال ورودی را به زمین متصل کنید.

▪ **پروسه فعال کردن ماژول سیگنال ورودی آنالوگ F هارت در حین کار**

SM 336; F-AI 6 x 0/4...20mA HART (6ES7 336-4GE00 0AB0)

۱. تعریف پارامترهای ماژول
۲. تعیین نوع اتصال سنسور (دوسیمه یا چهار سیمه)
۳. انجام تنظیمات خاص کانال
۴. انجام تنظیمات مربوط به ارتباطات هارت
۵. ورودی مثبت (*plus*) کانال را از طریق یک مقاومت (به عنوان مثال $3.9\ kilohms$) به منبع ولتاژ ($L+$) و پایه منفی کانال ورودی را به زمین متصل کنید.

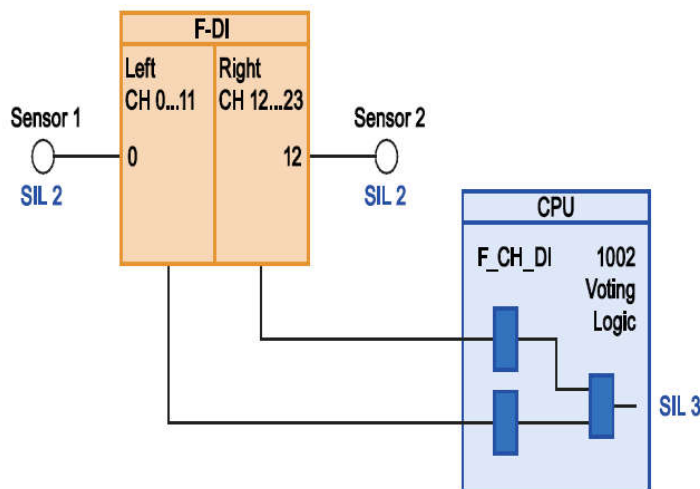
۴-۱۲- ارزیابی سنسور (Sensor evaluation)

۴-۱۲-۱- معماری های سیم بندی و انتخاب برای ET 200M (Wiring and Voting)

در ساختار معماری 1002 از دو سنسور استفاده می شود. در سیستم کنترل زیمنس برای حصول به سطوح $SIL3$ عمل *Voting* سیگنال های ورودی *fail-safe* را می توان به دو صورت انجام داد که عبارت انداز:

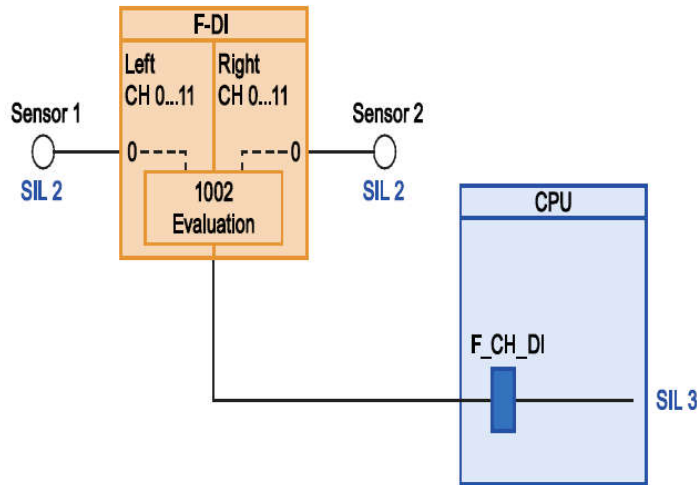
- انجام *Voting* در داخل ماژول سیگنال F ؛
- پیاده سازی *Voting* در برنامه لاجیک (چارت های CFC)؛

شکل زیر حالتی را نشان می دهد که در آن از دو سنسور استفاده شده و این دو سنسور به دو کانال یک ماژول $F-IO$ متصل شده است. همچنین عمل *voting* در برنامه F برای پیاده سازی کلاس $SIL 3$ صورت گرفته است.



شکل ۴-۱۸- انجام عمل *voting* به صورت برنامه *cfc* در CPU

شکل زیر نیز حالتی را نشان می دهد. که در آن برای پیاده سازی کلاس SIL 3، عمل *voting* بین دو سیگنال در داخل ماژول F انجام می شود.



شکل ۴-۱۹- انجام عمل *voting* به صورت سخت افزاری در داخل کارت

نکته: برای دستیابی به این نوع سیم بندی SIL3 سنسورهای مناسبی نیاز است.

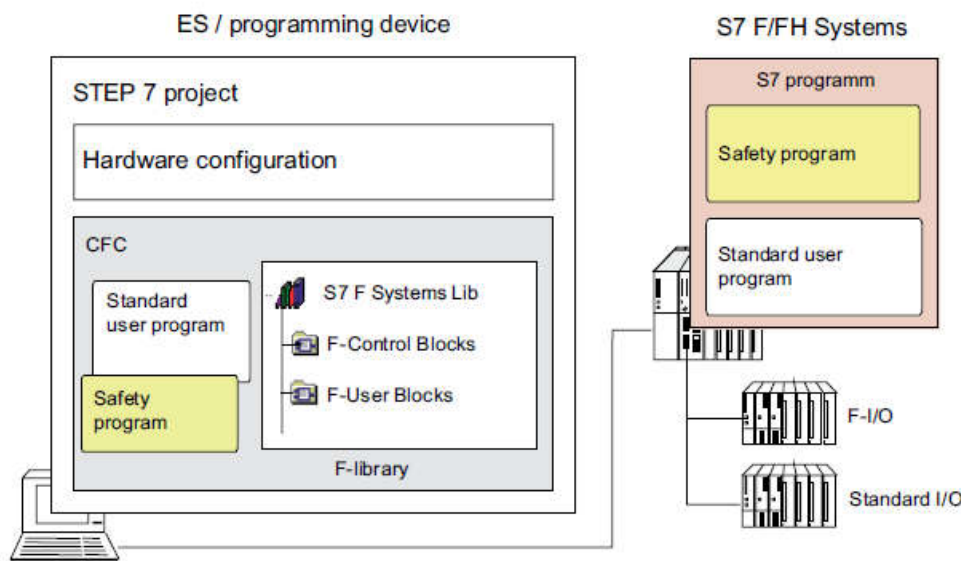
فصل ۵ - برنامه نویسی بلاک های F (Programming)

۵-۱- مقدمه

یک برنامه F متشکل از بلوک های F می باشد. که از یک کتابخانه F انتخاب شده اند. بلاک های F با استفاده از چارت های CFC به هم متصل شده و به طور خودکار در زمان کامپایل برنامه ایمنی به پوشه Blocks اضافه می شوند. در طول کامپایل، اقدامات کنترل خطا (فالت) نیز به طور خودکار به برنامه لاجیک افزوده می شود و آزمون های مرتبط با ایمنی انجام می شود.

شکل ۱-۵ ساختار شماتیکی یک پروژه مبتنی بر برنامه S7 که حاوی هر دو لاجیک استاندارد و لاجیک ایمنی می باشد را به تصویر کشیده است. به طور کلی یک برنامه S7 ماژول های FH متشکل از دو نوع لاجیک می باشد:

- برنامه نوشته شده با بلاک های استاندارد که برای توابع یا عملکرد ایمنی ضروری نمی باشد.
- برنامه F که توابع ایمنی را پیاده سازی می کند.



شکل ۱-۵- ساختار شماتیکی یک پروژه مبتنی بر برنامه S7

۵-۱-۱- ایجاد برنامه / لاجیک F (Configuring the safety program)

برنامه لاجیک در CPU را می توان با استفاده از بلوک های F و بلوک های غیر F (بلاک های استاندارد) ایجاد کرد. بلاک های F در گروه های زمان اجراء جداگانه (runtime groups) پیکربندی می شود. با توجه به این که یک سیستم FH هر دو برنامه استاندارد و F را در قالب یک برنامه S7 می تواند اجرا کند، بایستی یک روش امن در تبادل داده بین بخش F و بخش استاندارد برنامه S7 وجود داشته باشد. لذا انتقال داده ها بین برنامه استاندارد و F با استفاده از بلوک های تبدیل (Convert) واقع در کتابخانه F هندل

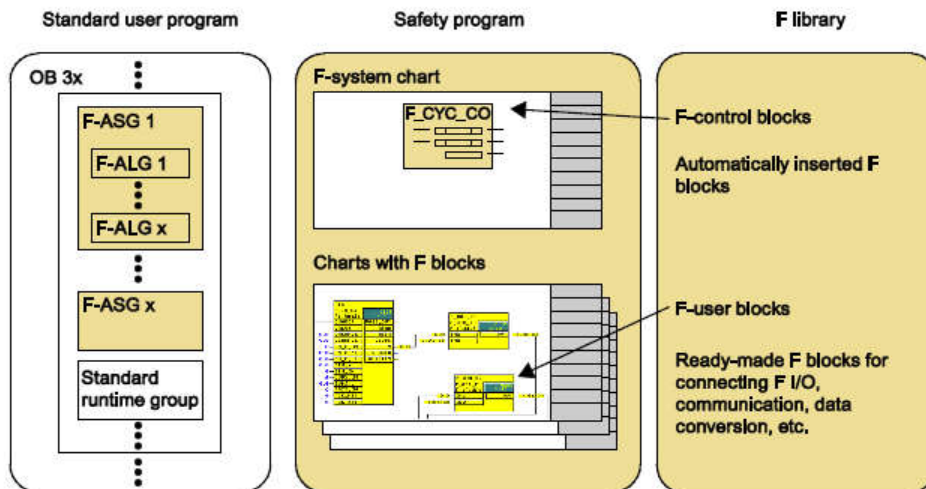
می شود. توجه داشته باشید که پیاده سازی عملکردهای ایمنی (مثلاً پیاده سازی یک حلقه کنترل ایمن) بایستی از نقطه قرائت سیگنال های ورودی (توسط بلاک های درایور ورودی) تا نقطه نوشتن فرامین در خروجی ها (توسط بلاک های درایور خروجی) به صورت پیوسته با بلوک های F پیاده سازی شود. به این معنی که بین این ها نباید از بلاک های استاندارد استفاده کرد.

علاوه بر توابع پیاده سازی کننده عملکرد ایمنی، بلوک های F شامل توابعی برای تشخیص و واکنش به خطاها است. این فانکشن بلاک های تشخیص فالت، تضمین می کنند که شکست ها و خطاها شناسایی شده و برای حفظ سیستم F در حالت امن و یا هدایت آن به یک وضعیت امن، واکنش مناسب را تریگر می کنند. در طول کامپایل، توابع خاص تشخیص خطا و واکنش به خطاها به طور خودکار به برنامه F اضافه می شوند.

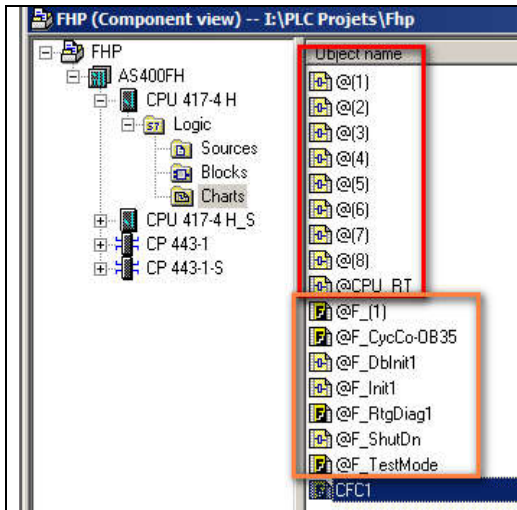
نکته: به طور کلی وضعیت *fail-safe* در توابع F منطبق صفر می باشد. به این معنی که حالت نرمال سیگنال یک بوده و وضعیت صفر، در یک سیگنال ورودی یا خروجی منجر به تریپ شده و سیستم متوقف می شود.

۵-۱-۲- ساختار برنامه ایمنی (safety program)

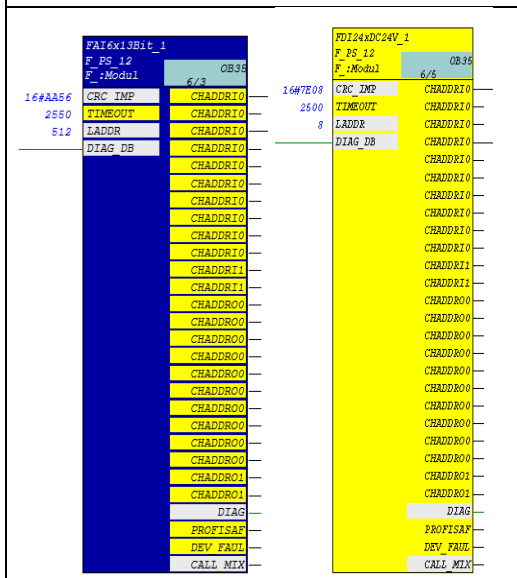
شکل زیر یک دیاگرام از ساختار یک برنامه F متشکل از چارت های CFC با بلاک های F را که جهت اجرا توسط CPU به گروه های F -runtime تخصیص داده شده است را به تصویر کشیده است.



شکل ۵-۲- دیاگرام/ساختار یک برنامه ایمنی



همانطور که از شکل دیده می شود یک برنامه F متشکل از چارت های F_CFC می باشد که شامل بلاک های F می باشد. خود این بلاک های F نیز دو دسته می باشند. دسته اول که به بلاک های کنترل معروف هستند، به صورت اتوماتیک در زمان کامپایل برنامه F در چارت های CFC با پیشوند $@$ اضافه می شوند. دسته دیگر که به بلاک های کاربر معروف هستند، برحسب نیاز و کد لاجیک برنامه F توسط کاربر از طریق اتصال آنها در چارت های CFC به همدیگر، برنامه نویسی می شوند.



به عنوان مثال یکی از این بلاک های کنترل بلاک درایور ماژول می باشد. که مسئول ارتباطات $PROFIsafe$ بین برنامه F و ماژول $F-I/O$ می باشد. درایور ماژول F با فعال کردن تیک گزینه $Generator Module Drivers$ در پنجره تنظیم کامپایل CFC به طور خودکار در چارت $@F_I$ قرار داده شده و به کانال های مربوطه در برنامه F متصل می شود.

برخی از مشخصات اجزای مختلف برنامه F عبارتند از:

۱. برنامه F شامل $runtime group$ های نوع F و چارت های تعریف شده به آنها می باشد.
۲. چارت ها شامل بلوک های F با تنظیمات پارامتری و اتصالات بین هم می باشد.
۳. برای اجرای $runtime group$ های نوع F توسط CPU ، آنها به یک یا چند OB وقفه سیکلینگ اضافه می شوند.
۴. $Runtime group$ های نوع F می تواند در $F-shutdown group$ ها ($F-SG$) ترکیب شود.
۵. OB وقفه سیکلینگ همچنین می تواند شامل گروه های زمان اجرای استاندارد باشد.
۶. بلوک های F در کتابخانه $S7 F$ سیستم F به رنگ زرد می باشد.
۷. چارت های CFC و گروه های زمان اجرای F به رنگ زرد ظاهر می شود.

۸. چارت های *CFC* و گروه های زمان اجرا نوع *F* با بلوک های *F* به رنگ زرد ظاهر می شوند و به منظور تمایز آنها از چارت ها و گروه های زمان اجرا مربوط به برنامه لاجیک استاندارد با حرف *F* مشخص می شوند.

۵-۱-۱- قوانین ساختار برنامه *F* (Rules governing program structure)

- موقع طراحی و آماده سازی پیش نویس یک برنامه *S7*، برای یک سیستم *S7 F/FH* بایستی به سوالات زیر در مقایسه برنامه *F* با برنامه استاندارد پاسخ داده شود.
- چه بخش هایی از برنامه *S7* بایستی به صورت *F* باشد.
 - به چه پاسخ زمانی هایی نیاز دارید. لذا بر این اساس بایستی برنامه *S7* را به *OB 3x* های مختلف تقسیم کنید.
 - تنها فقط می توان *F-Shutdown groups* های حاوی بلاک های *F* را به *OB 3X* اختصاص داد (*OB 30* تا *OB 38*) به عبارت دیگر گروه های زمان اجرا نوع *F* دارای بلوک های *F* را تنها می توان به *OB* های وقفه *30* تا *38* اختصاص داد.
 - یک برنامه *S7* می تواند شامل هر دو نوع بلوک *F* و بلوک استاندارد باشد؛ ولی نیاز است که این چارت ها در *Runtime group* های جداگانه وارد شوند. به عبارت دیگر بلاک های *F* و بلاک های استاندارد بایستی در چارت های جداگانه برنامه نویسی شوند. همچنین از این چارت ها نمی توان به عنوان بلاک تایپ *F* استفاده کرد.
 - داده *F-I/O* تنها از طریق درایور *F*-کانال قابل دسترسی است. یعنی دسترسی به ماژول های سیگنال *F* تنها از طریق درایور کانال *F (F_CH_xx)* مجاز است.
 - برای اتصال بلوک های *F* به هم از نوع داده بخصوص *F (F data types)* استفاده می شود. فرمت داده *F* قادر می سازد که خطاهای داده و آدرس شناسایی شود.

۵-۱-۲- اجرای بلاک های *F* (Run sequence of F-blocks)

سیستم اجرایی *S7*، برای اجرای برنامه کاربر از روش ساختار یافته استفاده می کند به این طریق که رابط اجرایی برنامه کاربر و سیستم عامل *CPU* بلاک هایی بنام *OB* می باشد که دارای سیکل های اجرایی متفاوت و با اولویت های اجرایی از قبل تعریف شده می باشد. به این معنی که هر یک از این بلاک های *OB* به صورت سیکلیک با اینتروال های زمانی تعریف شده در پنجره پراپرتی *CPU* فراخوانی شده و برنامه داخل آنها اجرا می شود به عنوان مثال *OB35* توسط سیستم عامل *CPU* هر ۱۰۰ میلی ثانیه فراخوانی می شود. زمینس از بلاک های *OB* تحت عنوان سطوح اجرایی *CPU (runtime level)* یاد می کند.

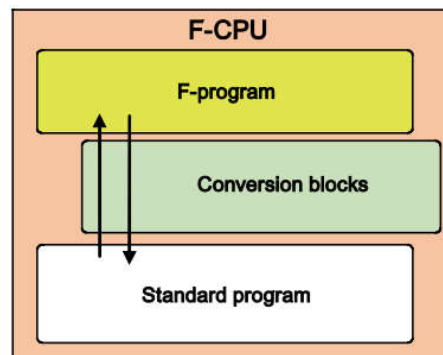
همچنین از آنجایی که برنامه کاربر توسط برنامه نویسی چارت های *CFC* صورت می گیرد، سیستم *S7* برای رعایت ترتیب زمانی اجرای بلاک های واقع در چارت ها، آنها (چارت های *CFC*) را به گروه های اجرایی

تحت عنوان *Runtime Group* تقسیم می کند. لذا زمانی که یک چارت جدید *CFC* در *PCS 7* ایجاد می شود، سیستم به طور خودکار یک *Runtime Group* به نام *CFC*، که بلوک های برنامه در آن قرار داده شده است را ایجاد می کند.

به همان روش برنامه استاندارد، موقع درج بلاک های *F* در چارت های *CFC* یک *Runtime Group* نوع *F* ایجاد می شود. یعنی به محض قرار دادن یک بلاک *F* در یک چارت *CFC*، آن چارت تبدیل به چارت *F* می شود. لذا یک *Runtime group* تنها زمانی *F-runtime group* می شود که در آن بلوک *F* فراخوانی شود. مشخصه *Run sequence* در چارت های *F* نیز به همان روش برنامه استاندارد، در ویرایشگر *CFC* تعریف می شود. تغییر در *Run sequence* همچنین موجب تغییر در *collective signature* می شود. نکته: در طول برنامه نویسی برنامه ایمنی، بلوک های *F* را نمی توان به طور مستقیم در بلاک های *OB* وقفه سیکلیک درج کرد.

۵-۱-۳- تبادل داده بین لاجیک *F* و برنامه استاندارد (Data exchange)

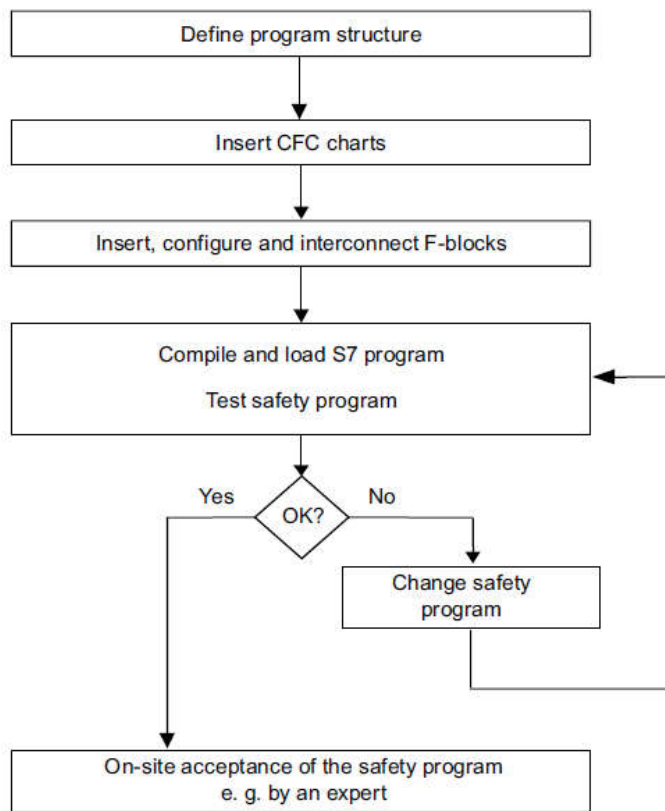
برنامه استاندارد و برنامه *F* از فرمت های داده متفاوت استفاده می کند. بر این اساس، بلوک های تبدیل خاص برای تبادل داده بین آنها استفاده می شود. اگر نیاز باشد که برنامه استاندارد، داده هایی از برنامه *F* را پردازش کند (مثال برای مانیتورینگ در *OS*)، از یک بلوک *F_Fdatatype_data* برای تبدیل نوع داده استفاده خواهد شد. بلاک *F* که برای تبدیل نوع داده *F* به نوع داده استاندارد استفاده می شود، بایستی در گروه اجرایی استاندارد فراخوانی شود (*standard runtime group*).



شکل ۳-۵- تبادل داده بین لاجیک *F* و برنامه استاندارد توسط بلاک های تبدیل صورت می گیرد

۵-۱-۴- روند پایه برای ایجاد برنامه ایمنی

شکل ۴-۵ یک روند پایه برای ایجاد یک برنامه *F* را به تصویر کشیده است. قبل از شروع به ایجاد لاجیک *F* کارهای زیر بایستی صورت گیرد.



شکل ۵-۴- روند پایه برای ایجاد برنامه F

	<p>۱. یک پروژه با نام دلخواه را در <i>Simatic Manager</i> ایجاد کنید.</p>
	<p>۲. سخت افزار <i>F-CPU</i> و <i>F-IO</i> در پنجره <i>HWConfig</i> پیکربندی شده باشد. به این معنی که <i>CPU</i> ای که قادر به اجرای برنامه <i>F</i> باشد (مانند <i>CPU 412-3H</i>, <i>CPU 414-4H</i>, or <i>CPU 417-4H</i>) و اجزای مورد نیاز یعنی رک، ماژول تغذیه، ماژول <i>CP</i> و غیره را پیکربندی شود. <i>ET200M > IM153-2 > 6ES7 153-2BA10-0XB0</i> <i>F-AI Module : 6ES7 336-1HE00-0AB0</i></p>

	<p>۳. در سربرگ <i>Protection</i> از پنجره پراپرتی <i>CPU</i> گزینه <i>CPU Contains safety program</i> را تیک زده و برای محافظت از برنامه <i>CPU</i> یک رمز تعریف کنید.</p>
	<p>۴. در پنجره پراپرتی ماژول سیگنال <i>F</i>، در قسمت <i>Parameters</i> مد کاری را به <i>Safety mode</i> تنظیم کنید. در غیر این صورت موقع کامپایل برنامه <i>F</i> خطا ظاهر خواهد شد.</p>

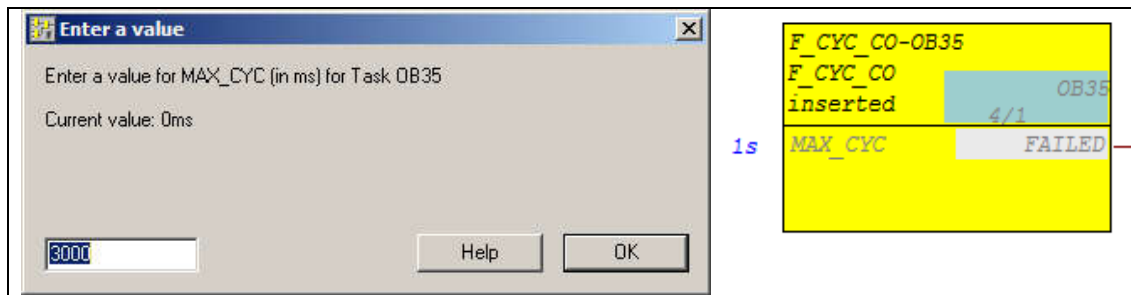
۵-۱-۵- زمان مانیتورینگ سیکل *F* (*F-cycle monitoring*)

فانکشن بلاک *F_CYC_CO*

هنگام استفاده از بسته *S7 F system*، این بلوک کنترلی *F* یعنی "*F_CYC_CO*" به طور خودکار در اولین کامپایل چارت های *CFC* در داخل یک چارت *@F_CycCo-OB3x* (مانند *@F_CycCo-OB35*) قرار داده شده و پیکربندی می شود. بلوک "*F_CYC_CO*" دارای نام سطح اجرایی *CPU* (مانند *OB35*) مربوطه می باشد. همچنین به صورت اتوماتیک در یک گروه *F-Runtime group* ایجاد شده به صورت خودکار، با شناسه ("*@F_*") در طول کامپایل برنامه *S7*، به منظور ایجاد برنامه ایمنی اجرایی از برنامه ایمنی کاربر درج می شود. ماژول *F-CPU*، زمان زمان *F-Cycle* را برای هر بار اجرای *OB 3x* که حاوی گروه های *F-Runtime* می باشد را مانیتور می کند.

وقتی برای اولین بار در برنامه *S7-F*، چارت های *CFC* کامپایل می شود، پنجره ای برای ورود یک مقدار برای زمان سیکل حداکثر (*MAX_CYC*) که می تواند بین دو فراخوانی *OB* صرف شود، نمایان می شود. اگر بعد از اولین کامپایل برنامه *S7* نیاز باشد که حداکثر زمان *F-Cycle* را تغییر دهید. برای تغییر مقدار *F-Cycle time* بایستی پارامتر *MAX_CYC* را در بلوک *F_CYC_CO-OB3x* در چارت *@F_CycCo-OB3x* تغییر دهید.

در صورتی که در کامپایل های بعدی، اگر مقدار پارامتر "*MAX_CYC*" نامعتبر باشد، یک باکس محاوره ای باز می شود که در آن می توان مقدار معتبر را وارد کرد. مقدار پیش فرض داده شده برای این پارامتر برابر ۳۰۰۰ میلی ثانیه (حداکثر زمان مانیتورینگ) است.



دلایل نیاز به تعریف زمان *F-runtime-group-time*

یک دلیل برای تعریف زمان *F-runtime-group* این است که مجبور نیستی که برنامه *F* را در هر سیکل *OB3x* فراخوانی کنید. نیاز به این که هر چند وقت یا با چه سرعتی برنامه *F* باید فراخوانی شود، فقط به فرآیند بستگی دارد. اگر هیچ الزامات خاصی در مورد زمان اجرا ندارید، می توانید به عنوان مثال سیکل زمانی *OB35* را برای *F-runtime-group* دو برابر فرض کنید.

۵-۱-۶- قرار دادن گروه های *F-Runtime* در چارت *CFC*

۱. موقع ترکیب گروه های *F-Runtime* و *standard runtime* در یک *OB* سیکلیک اینترپت، به منظور جلوگیری از طولانی تر شدن زمان اجرای گروه *F-Shutdown* و تاثیرگذاری در زمان پاسخ، گروه های *standard runtime* را قبل از گروه های *F-Runtime* اجرا کنید.

۲. برای یک گروه *F-Runtime* بایستی تنظیم پیش فرض پارامترهای *Phase shift* و *Reduction ratio* ($Reduction\ ratio = 1, Phase\ shift = 0$) حفظ شود. این مقادیر بایستی تغییر داده شود.

۳. گروه های *F-Runtime* که به صورت خودکار ایجاد شده اند نباید شیفت داده شوند و تغییرات نباید در داخل گروه *F-Runtime* انجام شود.

۵-۱-۷- گروه های *F-Shutdown* برنامه ایمنی

به محض این که بلوک های *F* در ویرایشگر *CFC* برای اولین بار قرار داده می شوند، همه گروه های *F-Runtime* در هر یک *OB 3x* ها یک گروه *F-Shutdown* شکل می دهند.
نکته: کاربر مجاز به اتصال بلوک های *F* متعلق به گروه های *F-Shutdown* به هم نمی باشد.

۵-۲- درج بلاک های *F* و اتصال آنها به همدیگر

بلوک های موجود در پوشه *F-Control Blocks* به طور خودکار در زمان کامپایل برنامه *S7* درج می شوند و کاربر مجاز به درج این بلوک ها به صورت دستی نمی باشد. همچنین کاربر مجاز به درج یک بلاک *F-Block* *instance* در چند گروه *F-Runtime* نمی باشد.

۵-۳-۲- F startup & (re)start Protection پروسه

در صورت توقف CPU اجرای لاجیک Safety، نیز متوقف می شود. لذا به این نوع توقف F-Stop گفته می شود.

۵-۳-۱- راه اندازی F (F-Startup)

در سیستم های S7-F، تمایزی بین راه اندازی مجدد سرد و راه اندازی مجدد گرم (Warm restart) برای ماژول CPU وجود ندارد. ولی بلاک های F_CHG_BO، F_CHG_R (بخشی از تابع نوشتن داده Safety) و F_MOV_R در این مورد استثناء می باشند.

در هر دو نوع راه اندازی مجدد سرد و گرم CPU، یک راه اندازی F صورت می گیرد. لذا در صورت وقوع راه اندازی F، اجرای برنامه ایمنی به طور خودکار با مقادیر اولیه شروع می شود. راه اندازی F در موارد زیر انجام می شود.

- اگر پس از توقف CPU، F-CPU راه اندازی مجدد Warm شود.
 - پس از F-STOP، اگر بلوک F_SHUTDOWN لبه مثبت در ورودی RESTART تشخیص دهد.
- پس از partial shutdown برنامه ایمنی، تنها گروه های F-shutdown که در F-STOP سهیم بودند، یک عمل F startup را انجام می دهند. گروه های F-shutdown دارای خطا در F-STOP باقی می ماند.

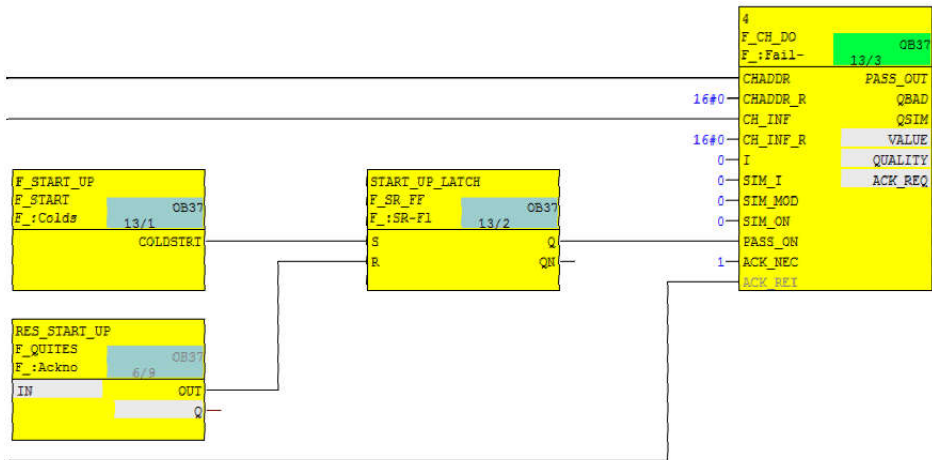
۵-۳-۲- (Re)Start Protection

اگر فرایند اجازه نمی دهد که برنامه ایمنی (safety program) به صورت خودکار با مقادیر اولیه راه اندازی (start up) شود، نیاز خواهید داشت که یک پاسخ به F-startup را برنامه نویسی کند.

بلاک F_START در برنامه ایمنی برای انجام F-startup با مقادیر اولیه استفاده می شود. پارامتر خروجی COLDSTRT نشان می دهد که یک F-startup تریگر شده است.

در مثال زیر، موقعی که برنامه F راه اندازی (started up) می شود (یعنی ماژول CPU از Stop به Run می رود)، خروجی فلیپ فلاپ یک می شود. این خروجی به ورودی Pass_ON بلاک های درایور متصل می شود. به طوری که تمام درایورهای خروجی را غیرفعال (passivate) می کند. همین که تمام شرایط فرآیند برآورده شد، اپراتور می تواند با استفاده از بلاک F_QUITES (F acknowledgment) و با ریست کردن فلیپ فلاپ سیگنال های خروجی را از حالت passivate خارج کند.

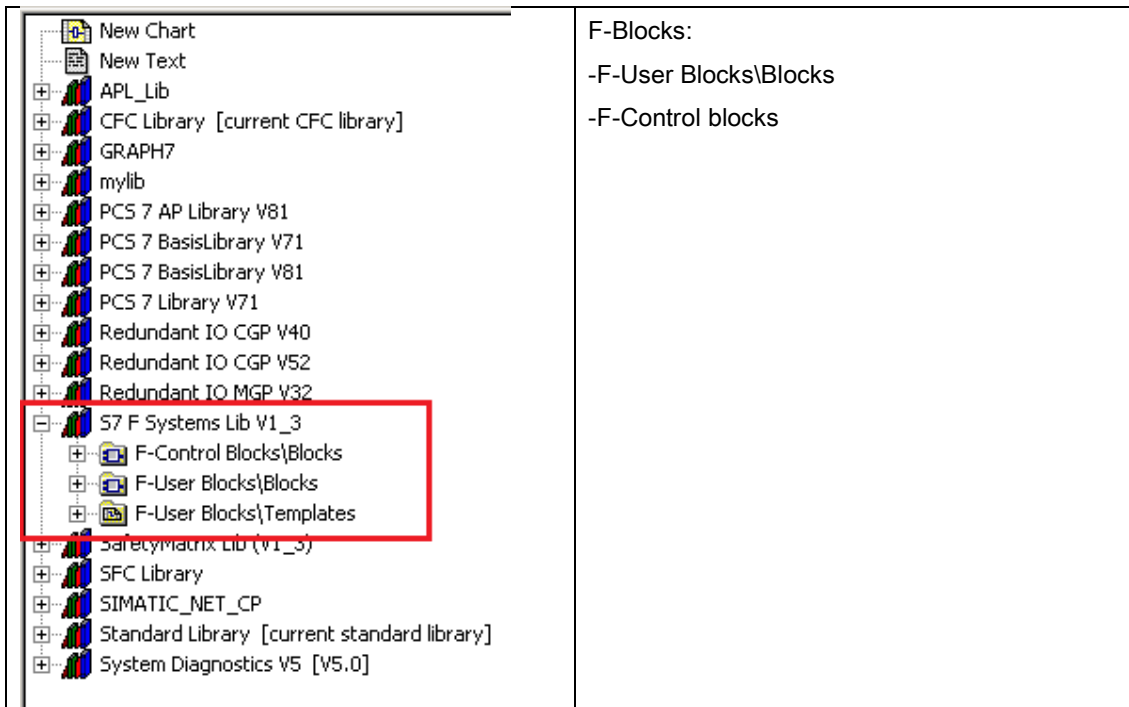
در این مثال در هنگام راه اندازی CPU خروجی COLDSTRT بلاک F_START یک شده و فلیپ فلاپ را Set می کند. و در نهایت موجب غیرفعال شدن بلاک های درایور متصل به خروجی فلیپ فلاپ می شود.



فصل ۶ - کتابخانه سیستم F

۶-۱- مقدمه

پس از نصب بسته F-System کتابخانه F-library V1.3 در کنار کتابخانه های دیگر در محیط CFC ظاهر می گردد. توابع این کتابخانه بر حسب تکنولوژی در پوشه های مختلف دسته بندی شده است.



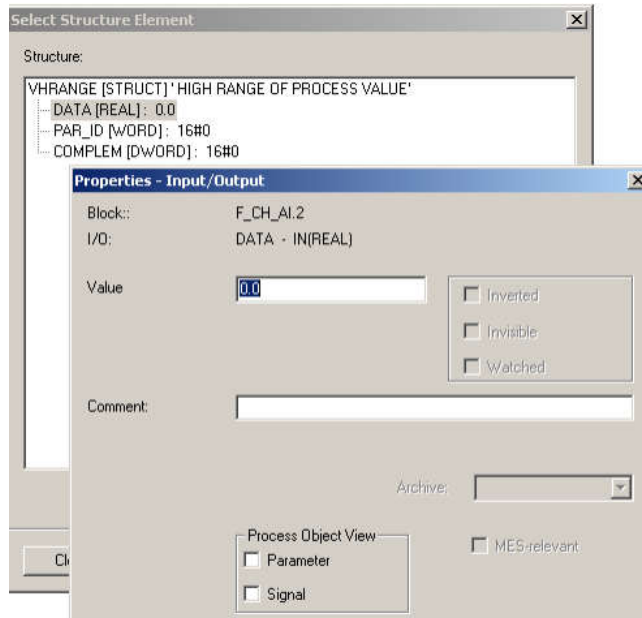
نکته: اسم کتابخانه F نبایستی تغییر داده شود. همچنین نبایستی شماره بلاک های F تغییر داده شود.

نکته: بلاک های F بایستی پیش از کامپایل در داخل CFC به یک آدرس درست و معتبر متصل شوند، در غیر این صورت خطا ظاهر می شود.

همچنین پیش از کامپایل بایستی در HW Config پارامترهای ماژول های F به مقادیر مناسبی تنظیم شوند. به عنوان مثال مد کاری ماژول به Safety mode تنظیم شود. در غیر این صورت خطا خواهد داد.

۶-۱-۱- نوع داده F (F-Data types)

در کتابخانه S7 F-system برای پارامترهای ورودی/خروجی بلاک های F از نوع داده های بخصوص در عملیات محاسباتی و منطقی استفاده می شود. نوع داده های بخصوص F در یک قالب داده ایمنی برای واسطه های بلوک های fail safe استفاده می شود. قالب داده F (safety data format) برای نمایش داده و آدرس دهی خطاها استفاده می شود. از نظر برنامه نویسی نوع داده های F به صورت structures پیاده سازی می شوند به طوری که در آن تنها جزء DATA به کاربر مربوطه می شود.



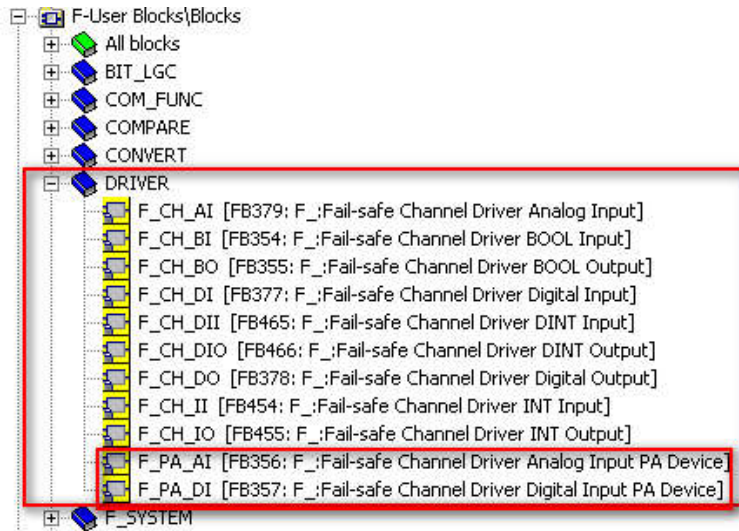
شکل ۶-۱- استفاده از نوع داده Structure برای پارامترهای واسط بلاک

۶-۱-۲- دسترسی به ورودی/خروجی های نوع F (F-I/O access)

در سیستم های نوع S7-FH، دسترسی به داده ماژول های سیگنال F، به جای دسترسی از طریق حافظه تصویر فرآیند ورودی (Process Image) از طریق بلاک های درایور F صورت می گیرد. برای این منظور دو نوع درایور یعنی درایور ماژول سیگنال و درایور کانال سیگنال برای دسترسی به مقادیر F-I/O استفاده می شود. به این معنی که به ازای هر ماژول F I/O یک درایور ماژول F و به ازای هر کانال ورودی/خروجی F از یک بلاک درایور کانال F استفاده می شود.

◀ درایور کانال F (F channel driver): به عنوان مثال F_CH_xx برای هر کانال ورودی/خروجی F در کانال های سیگنال ریداندانت تنها از یک درایور کانال سیگنال استفاده می شود.

◀ درایور ماژول (module driver): درایور ماژول F مسئول ارتباطات PROFIsafe بین برنامه F و ماژول F-I/O می باشد. به ازای هر ماژول F-I/O یک درایور ماژول F (F-Module driver) با فعال کردن تیک گزینه Generator Module Drivers در پنجره تنظیم کامپایل CFC به طور خودکار در چارت های CFC با پیشوند @ قرار داده شده و به کانال های مربوطه در برنامه F متصل می شود.

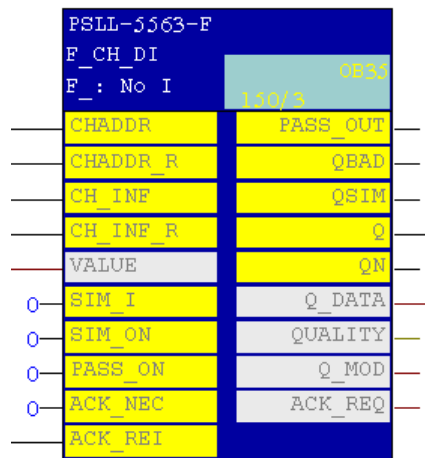


شکل ۳-۶-۳- بلوک های دایور کانال های ورودی/خروجی F

۶-۲-۱- بلوک درایور کانال دیجیتال ورودی (F_CH_DI)

Fail-safe channel driver for digital inputs of F-I/O (except fail-safe DP standard slaves)

این بلاک برای دریافت داده از یک کانال دیجیتال F استفاده می شود.



شکل ۶-۴- بلاک درایور کانال دیجیتال ورودی معمولی (نه برای پروفیباس DP)

ورودی/خروجی	نوع داده	توضیح
ADR_CODE	DWORD	
VALUE	BOOL	آدرس سیگنال کانال ورودی دیجیتال کارت به این ورودی متصل می شود. به عنوان مثال ورودی دیجیتال با سیمبل ZSL-5450-I-G. نوع داده این پایه به جای F_BOOL از نوع BOOL معمولی می باشد.

<i>SIM_I</i>	<i>F_BOOL</i>	در صورتی که بخواهیم سیگنال این کانال را در نبود آن شبیه سازی کنیم، مقدار ۱ به این پایه داده می شود.
<i>SIM_ON</i>	<i>F_BOOL</i>	امکان شبیه سازی سیگنال را برای کانال فعال می کند.
<i>PASS_ON</i>	<i>F_BOOL</i>	این پایه برای غیرفعال کردن کانال از طریق برنامه استفاده می شود. با یک کردن این پایه، کانال مربوطه غیرفعال می گردد. با یک شدن این ورودی، خروجی <i>QBAD</i> یک می شود. <i>I = Activate Passivation</i>
<i>ACK_NEC</i>	<i>F_BOOL</i>	در صورت یک شدن این پایه، <i>Ack</i> کردن خطای <i>Passivation</i> ضروری می باشد. <i>I = Acknowledgement Necessary</i>
<i>ACK_REI</i>	<i>F_BOOL</i>	<i>Acknowledgment Reintegration</i> با یک کردن این ورودی عمل <i>Depassivate</i> کانال ها صورت می گیرد.
<i>ACK_REQ = I</i>		یک شدن این خروجی به این معنی است که خطا از بین رفته است و تصدیق کاربر در ورودی <i>ACK_REI</i> برای فعال سازی مجدد کانال (<i>reintegration</i>) لازم است.
<i>QUALITY</i>	<i>BYTE</i>	کد کیفیت سیگنال ورودی را نشان می دهد. اگر مقدار ورودی دیجیتال دریافت شده از <i>F-I/O</i> معتبر باشد، مقدار دریافت شده از کانال با کد کیفیت <i>16#80</i> در خروجی <i>QUALITY</i> در خروجی <i>Q</i> قرار می گیرد.
<i>QN</i>	<i>F_BOOL</i>	<i>Negated Process Value</i> <i>Not</i> شده خروجی <i>Q</i> می باشد.
<i>Q_DATA</i>	<i>BOOL</i>	<i>PROCESS VALUE DATA (for monitoring)</i> مقدار دریافت شده از کانال دیجیتال ورودی در صورت معتبر بودن در سه پایه <i>Q</i> ، <i>QN</i> (به صورت معکوس) و <i>Q_DATA</i> ظاهر می شود. از آنجایی که نوع داده این پایه خروجی از نوع <i>BOOL</i> می باشد، می توان به طور مستقیم به یک بلاک مانیتورینگ دیجیتال متصل کرد.
<i>DISCF</i>	<i>BOOL</i>	<i>Discrepancy Error Module</i> این خروجی زمانی معنی پیدا می کند. که کانال های ماژول به صورت ریداندانت پیکربندی شده باشند. در صورت وقوع خطای <i>Discrepancy</i> این خروجی یک می شود.

▪ فعال سازی مجدد کانال پس از حذف خطا (*Reintegration after error elimination*)

بعد از این که خطا رفع شد، قرائت مقدار ورودی دیجیتال دریافت شده از کانال *F-I/O* می تواند به طور خودکار و یا پس از *Ack* کردن توسط کاربر فعال شود.

اگر ورودی *ACK_NEC = I* باشد، عمل تصدیق کاربر (*user acknowledgement*) از طریق ورودی *ACK_REI*، بعد از رفع خطا ضروری است و اگر *ACK_NEC = 0* باشد، فعال سازی (*reintegration*) کانال

به صورت خودکار (*automatic reintegration*) انجام می گیرد. برای حالتی که $PASS_ON = 1$ است و یا راه اندازی F (*F-Startup*) پس از یک $CPU-STOP$ به تصدیق کاربر نیاز نیست.

Fail-safe value

در صورت بروز خطا مقدار صفر در خروجی Q بلاک قرار می گیرد که **Fail-safe value** نامیده می شود. مقدار صفر در شرایط زیر در خروجی Q قرار می گیرد.

- مقدار دیجیتال کانال به دلیل خطای ارتباطی (*PROFIsafe*) غیر معتبر باشد.
- مقدار دیجیتال کانال به دلیل فالت در مازول یا کانال (مانند قطعی سیم) غیر معتبر باشد.
- یک پروسه *F-Startup* در حال اجرا باشد.
- یک حالت *Passivation* با $PASS_ON = 1$ وجود داشته باشد.

▪ آنالیز اختلاف بین دو کانال ریداندانت (*Discrepancy analysis*)

در خصوص مازول های $F-I/O$ پیکربندی شده به صورت افزونه، اگر زمان اختلاف (*discrepancy time*) تنظیم شده در *HWConfig* مخالف صفر باشد، بلوک F تجزیه و تحلیل اختلاف مقدار بین دو کانال افزونه را انجام می دهد.

اگر اختلاف بین کانال های ورودی دیجیتال ریداندانت بیش از زمان اختلاف تعریف شده باشد، یک خطای اختلاف تشخیص داده می شود. در این صورت بلوک F خروجی *DISCF* را یک می کند. اگر کانال ورودی دیجیتال که در ورودی *VALUE* آدرس دهی شده است، سیگنال صفر را اعمال کند، بلاک F خروجی *DISCF* را یک می کند. همچنین اگر مقدار کانال ریداندانت نیز صفر باشد، بلوک F خروجی *DISCF_R* را یک می کند خروجی های *DISCF / DISCF_R* به محض رفع شدن اختلاف ریست می شوند.

به عنوان مثال، عمل تجزیه و تحلیل اختلاف، تشخیص سنسور معیوب را فراهم می سازد. به این دلیل که فرض می کند که وقتی سنسور F مقدار صفر را تولید می کند، معیوب می باشد. این کار می تواند در دسترس بودن سیستم را افزایش دهد.

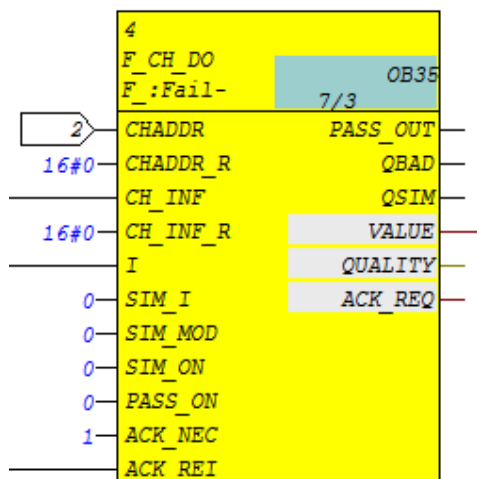
نکته- خطاهای اختلاف هیچ تاثیری بر روی خروجی های Q ، $QBAD$ یا $PASS_OUT$ ندارد. خروجی های *DISCF/DISCF_R* را که از نوع *non-fail-safe* هستند را می توان توسط یک *OS* (مانیتورینگ) برای فعالیت هایی مانند ارزیابی در برنامه استاندارد قرائت کرد.

رفتار درایور کانال در زمان *F-STOP*

در صورت وقوع یک *F-STOP*، کد کیفیت $16\#00$ در خروجی *QUALITY* و $QBAD.DATA = 1$ می شود. دیگر متغیرهای کانال فریز (*frozen*) می شوند.

۶-۲-۲- بلاک درایور کانال دیجیتال خروجی (F_CH_DO)

این بلاک برای نوشتن مقادیر دیجیتال Fail safe در کانال خروجی F-I/O استفاده می شود.



شکل ۶-۵- نمایشی از بلاک درایور کانال دیجیتال خروجی (F_CH_DO)

پایه ورودی/خروجی	نوع داده	توضیح
I	F_BOOL	بیت خروجی (Process Value) که قرار است به یک خروجی منتقل شود، به این پایه داده می شود.
SIM_MOD	F_BOOL	I=Simulation Has Priority با یک شدن این پایه، اولویت با شبیه سازی خواهد بود.
SIM_I	F_BOOL	در صورتی که بخواهیم سیگنال این کانال را در نبود آن شبیه سازی کنیم، مقدار ۰، ۱ به این پایه داده می شود.
SIM_ON	F_BOOL	شبیه سازی سیگنال را برای کانال فعال می کند.
PASS_ON	F_BOOL	I = Activate Passivation این پایه برای غیر فعال کردن کانال از طریق برنامه استفاده می شود. با یک کردن این پایه، کانال مربوطه غیرفعال می گردد.
ACK_NEC	F_BOOL	در صورت یک شدن این پایه، Ack کردن خطای Passivation ضروری می باشد. I = Acknowledgement Necessary
ACK_REI	F_BOOL	Acknowledgment reintegration
ACK_REQ=I		به این معنی است که خطا از بین رفته است و تصدیق کاربر در ورودی ACK_REI برای فعال سازی مجدد کانال (Reintegration) لازم است.
QUALITY Quality Code Of Process Value	BYTE	کد کیفیت سیگنال ورودی را نشان می دهد. اگر مقدار ورودی دیجیتال دریافت شده معتبر باشد، کد کیفیت 16#80 در خروجی QUALITY قرار می گیرد.
QSIM	F_BOOL	I = simulation active

		فعال بودن مد شبیه سازی در کانال را نشان می دهد.
QBAD	F_BOOL	1 = process value invalid
Normal value		در صورت عدم وجود خطا در کانال، داده دریافت شده از I با کد کیفیت 16#80 در ماژول F-I/O نوشته می شود.
PASS_OUT	F_BOOL	1 = Passivation Because Of Error در صورت وقوع Passivation در کانال به دلیل خطا این خروجی یک می شود.

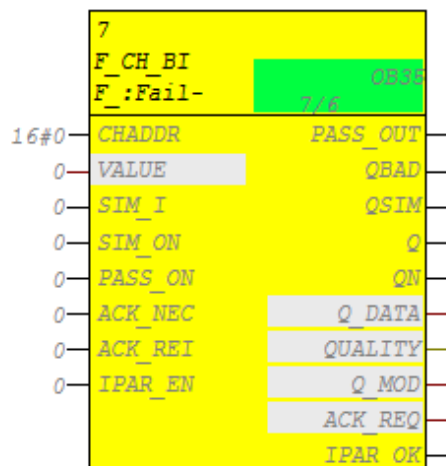
شبیه سازی

- اگر ورودی $SIM_ON=1$ و $SIM_MOD=0$ باشد، مقدار SIM_I در کانال F-I/O نوشته شده و به شرطی که هیچ خطای ارتباطی (PROFIsafe) و هیچ فالتی در کانال یا ماژول وجود نداشته باشد، در خروجی VALUE ظاهر می شود.
- اگر ورودی $SIM_ON=1$ و $SIM_MOD=1$ باشد، در صورت وقوع خطای ارتباطی (PROFIsafe) و فالت در کانال یا ماژول، مقدار SIM_I در خروجی VALUE ظاهر می شود.
- در هر دو حالت بالا خروجی QUALITY به مقدار 16#60 تنظیم شده و $QSIM = 1$ می شود.

۶-۲-۳- بلاک درایور برای ورودی باینری از شبکه DP (F_CH_BI)

F_CH_BI: F-Channel driver for inputs of data type BOOL of fail-safe DP standard slaves and fail-safe standard I/O devices

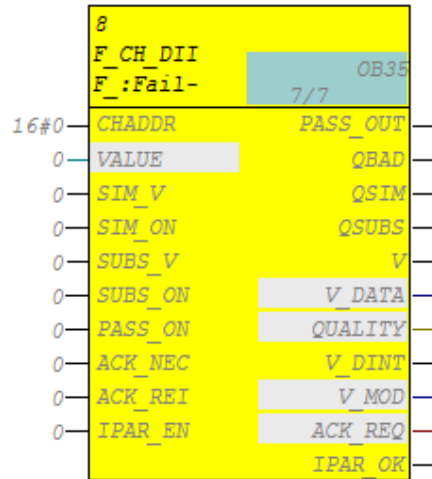
این بلاک برای پردازش یک سیگنال ورودی باینری (BOOL) از Slave های fail-safe DP standard و وسایل fail-safe standard I/O استفاده می شود. این بلوک F، به صورت سیکلیک مقدار ورودی BOOL را از یک Slave نوع fail-safe DP آدرس دهی شده در ورودی VALUE را می خواند مقدار ورودی از طریق درایور ماژول خرابی امن F_PS_12 که با fail-safe DP standard تبادل داده می کند، بر اساس پروفایل PROFIsafe دریافت می شود. اگر مقدار ورودی دیجیتال معتبر باشد، در خروجی Q قرار می گیرد.



شکل ۶-۶-۶- نمایشی از بلاک درایور کانال ورودی F_CH_BI

۶-۲-۴- بلاک درایور قرائت مقدار $DINT$ از یک آدرس $DP Slave$ (F_CH_DII)

F-Channel driver for inputs of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices

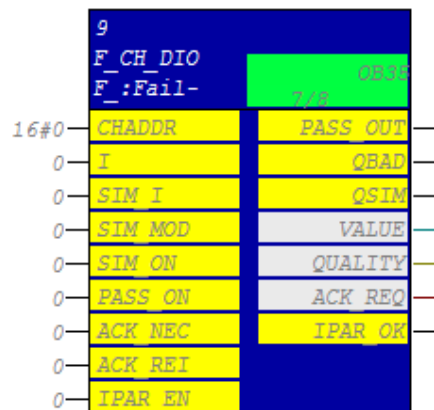


شکل ۶-۶-۷- بلاک درایور قرائت مقدار $DINT$ از یک آدرس $DP Slave$ (F_CH_DII)

۶-۲-۵- بلاک درایور نوشتن مقدار $DINT$ در یک آدرس $DP Slave$ (F_CH_DIO)

F-Channel driver for outputs of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices

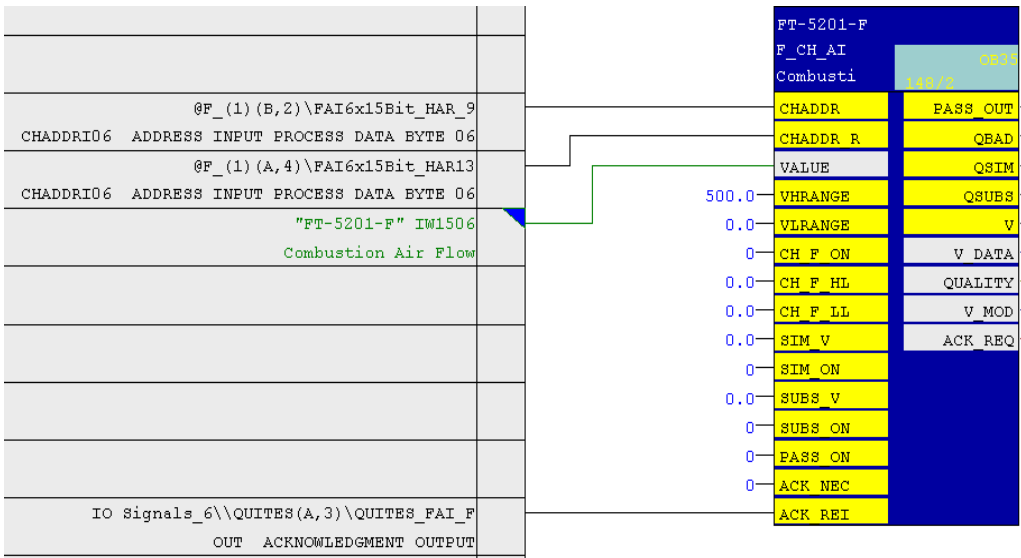
این بلوک F برای پردازش سیگنال یک مقدار خروجی از نوع $DINT$ در slave های $fail-safe DP$ $standard$ یا دستگاههای $fail-safe standard I/O$ استفاده می شود.



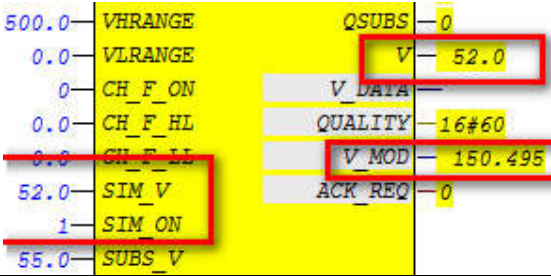
شکل ۶-۶-۸- بلاک درایور نوشتن مقدار $DINT$ در یک آدرس $DP Slave$

۶-۲-۶- بلاک درایور کانال ورودی آنالوگ F_CH_AI

Fail-safe channel drivers for analog inputs of F-I/O (except fail-safe DP standard slaves)



پایه ورودی/خروجی	نوع داده	توضیح
CHADDR	STRUCT	CHANNEL-ADDRESS IN F MOD-BLOCK
CHADDR_R	STRUCT	REDUNDANT CHANNEL-ADDRESS IN F_MOD-BLOCK
VALUE	WORD	به آدرس IW کانال آنالوگ متصل می شود.
VHRANGE	F_REAL	رنج بالای متغیر ورودی فرآیند (PV) را مشخص می کند.
VLRANGE	F_REAL	رنج پایین متغیر ورودی فرآیند (PV) را مشخص می کند.
CH_F_ON	F_BOOL	I = Enable Limit Value Monitoring با یک شدن این ورودی مقدار PV با حدود مقادیر تعریف شده در ورودی های CH_F_LL و CH_F_HL مانیتور می شود.
CH_F_HL	F_REAL	Overrange limit of input value (mA)
CH_F_LL	F_REAL	Underrange Limit Of Input Value (mA)
SIM_V	F_REAL	مقدار ورودی برای شبیه سازی می باشد.
SIM_ON	F_BOOL	امکان شبیه سازی مقدار ورودی را فعال می کند.
SUBS_V	F_REAL	مقدار جایگزین در مواقع بروز خطا به این ورودی داده می شود.
SUBS_ON	F_BOOL	امکان جایگزینی مقدار ورودی با مقدار پایه SUBS_V را فعال می کند.
PASS_ON	F_BOOL	این پایه برای غیر فعال کردن کانال از طریق برنامه استفاده می شود. با یک کردن این پایه، کانال مربوطه غیرفعال می گردد. خروجی QBAD یک می شود. I = Activate Passivation

<i>ACK_NEC</i>	<i>F_BOOL</i>	در صورت یک شدن این پایه، <i>Ack</i> کردن خطای <i>Passivation</i> ضروری می باشد. <i>I = Acknowledgement Necessary</i>
<i>ACK_REI</i>	<i>F_BOOL</i>	<i>Acknowledgment Reintegration</i>
<i>Normal value</i>		اگر عدد دیجیتال دریافت شده از <i>F-I/O</i> در ورودی <i>VALUE</i> معتبر باشد، آن را به کمیت فیزیکی بر اساس مقادیر ورودی های <i>VLRANGE</i> و <i>VHRANGE</i> اسکیل کرده و در خروجی <i>V</i> با کد کیفیت <i>16#80</i> قرار می دهد. برای فعال کردن امکان انتقال مقادیر <i>VLRANGE</i> و <i>VHRANGE</i> به پارامترهای سایر بلوک ها، مقادیر ورودی های <i>VLRANGE</i> و <i>VHRANGE</i> در خروجی های <i>OVHRANGE</i> و <i>OVLRange</i> نیز نوشته می شود. الگوریتم تبدیل ورودی را یک سیگنال خطی فرض می کند هنگامی که <i>VLRANGE = 0.0</i> و <i>VHRANGE = 100.0</i> است، یک مقدار درصد در خروجی تولید می شود. اگر <i>VHRANGE = VLRANGE</i> تنظیم شود،
<i>V_MOD</i>	<i>REAL</i>	<i>Value From Module</i> این خروجی مقدار آنالوگ خوانده شده از کارت را نشان می دهد. در صورتی که مد شبیه سازی با <i>SIM_ON=1</i> فعال شده باشد. خروجی <i>V</i> مقدار وارد شده در <i>SIM_V</i> را و پایه <i>V_MOD</i> مقدار خوانده شده از <i>F-I/O</i> را به عنوان متغیر فرآیندی نشان خواهد داد. 
<i>V_DATA</i>	<i>REAL</i>	<i>Process Value Data (For Monitoring)</i> نوع داده این خروجی از نوع معمولی (<i>F_REAL</i> نیست) بوده و می توان به صورت مستقیم به بلاک های مانیتورینگ متصل کرد.
<i>ACK_REQ</i>	<i>BOOL</i>	<i>Acknowledgement Request</i> یک شدن این خروجی نشان می دهد. که خطا رفع شده و نیاز به <i>Ack</i> توسط کاربر می باشد.

PT1001		
F_CH_AI		
F_:Fail-		
CHADDR	PASS_OUT	
CHADDR_R	QBAD	1
VALUE	QSIM	0
500.0	VHRANGE	QSUBS
0.0	VLRANGE	V
0	CH_F_ON	V_DATA
0.0	CH_F_HL	QUALITY
0.0	CH_F_LL	V_MOD
0.0	SIM_V	ACK_REQ
0	SIM_ON	
0.0	SUBS_V	
0	SUBS_ON	
1	PASS_ON	
0	ACK_NEC	
0	ACK_REI	

در صورتی که به صورت دستی ورودی $PASS_ON$ را یک کنیم، کانال غیرفعال شده و مطابق شکل خروجی $BAD=1$ یک شده و مقدار خروجی V در آخرین مقدار خود ثابت می ماند.

PT1001		
F_CH_AI		
F_:Fail-		
CHADDR	PASS_OUT	
CHADDR_R	QBAD	1
VALUE	QSIM	0
500.0	VHRANGE	QSUBS
0.0	VLRANGE	V
0	CH_F_ON	V_DATA
0.0	CH_F_HL	QUALITY
0.0	CH_F_LL	V_MOD
0.0	SIM_V	ACK_REQ
0	SIM_ON	
55.0	SUBS_V	
1	SUBS_ON	
1	PASS_ON	
0	ACK_NEC	
0	ACK_REI	

Fail-safe value → $SUBS_V$

اگر ورودی $SUBS_ON=1$ باشد و به ورودی $SUBS_V$ مقدار داده باشیم. در آن صورت با وقوع خطا و *passivate* شدن کانال، درحالت های زیر مقدار $SUBS_V$ در خروجی V قرار می گیرد. همانطور که مشاهده می شود. خروجی $ACK_REQ=1$ نیز یک می شود.

- مقدار آنالوگ ورودی به دلیل خطای ارتباطی (*PROFIsafe*) غیر معتبر باشد.
- مقدار آنالوگ ورودی به دلیل فالت در مازول یا کانال (مانند قطعی سیم) غیر معتبر باشد.
- مقدار آنالوگ ورودی به دلیل *ovrflow* یا *underflow* غیر معتبر باشد.
- مقدار آنالوگ ورودی به دلیل فالت تخطی از حدودد (*limit violation*) در کانال فعال (برای رنج اندازه گیری 4-20mA)، غیر معتبر باشد.
- یک حالت *Passivation* با $PASS_ON = 1$ وجود داشته باشد.
- یک پروسه *F-Startup* در حال اجرا باشد.

فعال سازی کانال پس از رفع خطا (Reintegration after error elimination)

بعد از رفع خطا، مقدار ورودی آنالوگ دریافت شده از $F-I/O$ می تواند به طور خودکار و یا پس از تصدیق کاربر از طریق ACK_REI مجدد تولید (*Reintegrate*) شود.

در صورتی که ورودی $ACK_NEC=1$ باشد، بعد از رفع خطا تصدیق کاربر از طریق ورودی ACK_REI لازم است. ولی اگر $ACK_NEC = 0$ باشد، تصدیق به صورت خودکار انجام می گیرد.

(NUMUR) limit value check in the 4 to 20 mA measuring range

در دستورالعمل NUMUR برای پردازش سیگنال آنالوگ، برای سیگنال آنالوگ نوع *life zero* یعنی سیگنال $4-20\text{ mA}$ ، در جایی که فالت در کانال وجود دارد، حدود تعریف شده است. که عبارت انداز: $3.6\text{ mA} < \text{analog signal} < 21\text{ mA}$ به طور پیش فرض، این حدود تعریف شده در NAMUR، برای چک کردن مقادیر حدی (*limit value*) (*checking*) استفاده می شود. برای چک کردن حدود فالت جدید بایستی ورودی $CH_F_ON=1$ تنظیم شود و مقادیر حدود جدید در ورودی های CH_F_HL و CH_F_LL برحسب میلی آمپر (mA) وارد شود. $CH_F_LL < \text{analog signal} < CH_F_HL$ در صورت وقوع سرریز (*overflow*) یا *underflow* از حدود فالت کانال فعال، خروجی $QBAD$ نیز یک شده و بسته به مقدار ورودی $SUBS_ON$ ، مقدار خرابی امن $SUBS_V$ یا آخرین مقدار معتبر خروجی در خروجی قرار می گیرد.

۶-۲-۷- مفاهیم *Live zero* و *Dead zero*

عبارت *Live zero* به کمترین مقدار رنج سیگنال (*lowest end or 0% of the signal range*) اشاره می کند. و عبارت *Dead zero* به یک وضعیت شکست (*signal failure or dead signal*) در سیگنال اشاره می کند. به عنوان مثال اگر یک سیگنال جریان $4-20\text{ mA}$ را در نظر بگیرید، مقدار 4mA نمایش دهنده *live zero* یا 0% سیگنال بوده و مقدار 0mA (*no current*) نمایش دهنده یک *dead-zero* می باشد. سیگنال $4-20\text{ mA}$ یک مثال کلاسیک از یک سیستم سیگنالینگ است که بین یک سیگنال 1% و یک شکست در سیگنال، تفاوت قائل می شود. در نتیجه امکان تشخیص خرابی در حلقه را فراهم می کند. به تعریف دیگر یک *live zero* یک سیگنال در یک حلقه (*loop signal*) می باشد که در آن مقدار صفر سیگنال بالاتر از صفر واقعی می باشد. به عنوان مثال $10-50\text{mV}$ ، $1-5\text{ VDC}$ ، $4-20\text{ mADC}$ و غیره. یک *dead zero* سیگنالی است که در آن مقدار صفر سیگنال، خودش مقدار صفر حلقه می باشد. مانند سیگنال $0-5\text{ vdc}$ ، $0-100\text{ mv}$ ، $0-20\text{ mADC}$ و غیره.

۶-۲-۸- بلاک درایور کانال آنالوگ خروجی

به طور معمول در سیستم های *Fail-safe* تنها ماژول $F-AI$ وجود دارد و از ماژول $F-AO$ استفاده نمی شود. با توجه به این که کارت های *Fail safe* برای EMS و سیستم های $SAFETY$ استفاده می شوند. خروجی آنالوگ تو این دسته کارتها وجود ندارد.

همان طوری که از عنوان «قطع اضطراری» بر می آید، در سیستم های ESD یا fail-safe هدف غایی، هدایت پلنت به یک وضعیت ایمن (fail-safe) از طریق توقف یا تریپ سریع یک زیر سیستم پس از وقوع یک رخداد خطرناک می باشد این کار با استفاده از عملگرهای باینری که به صورت On/Off هستند انجام می شود.

۶-۳-۳- مجموعه BIT_LGC

BIT_LGC Family: Logic blocks with the BOOL data type

بلاک های این مجموعه برای پیاده سازی عملیات بیتی (نوع داده BOOL) لاجیک F استفاده می شود. گروه

BIT_LGC شامل شش بلاک برای عملیات باینری می باشد:

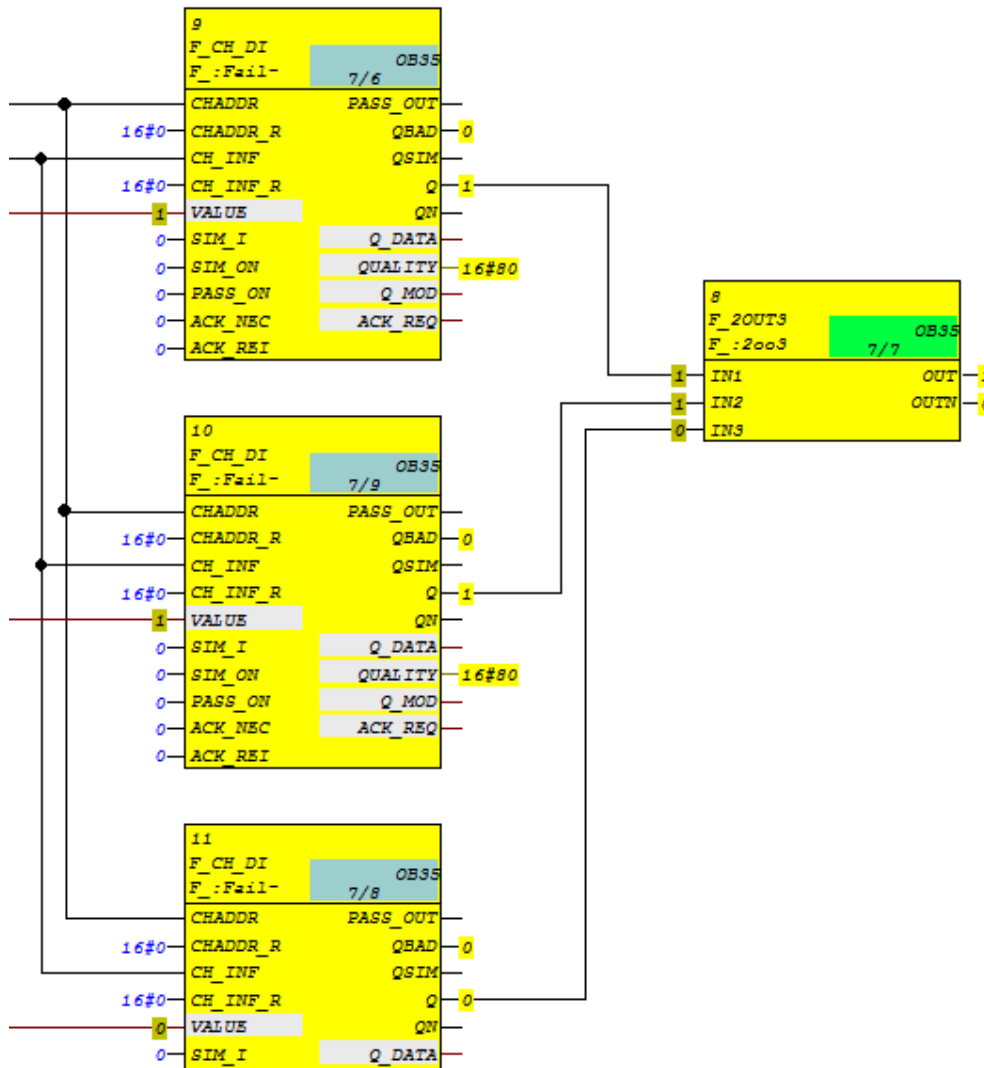
BIT_LGC	
F_2OUT3 [FB305: F_:2oo3 evaluation of BOOL Inputs]	
F_AND4 [FB301: F_:AND 4 Inputs]	
F_NOT [FB304: F_:Inverter]	
F_OR4 [FB302: F_:OR 4 Inputs]	
F_XOR2 [FB303: F_:EXOR 2 Inputs]	
F_XOUTY [FB306: F_:XooY evaluation of BOOL Inputs]	
F_AND4	چهار ورودی F_BOOL را با هم AND می کند.
F_OR4	چهار ورودی F_BOOL را با هم OR می کند.
F_XOR2	دو ورودی F_BOOL را با هم XOR می کند.
F_NOT	ورودی نوع F را Not می کند.
F_2OUT3	ارزیابی 2oo3 را بر روی ورودی های نوع داده F_BOOL انجام می دهد.
F_XOUTY	ارزیابی XooY را بر روی ورودی های نوع داده F_BOOL انجام می دهد.

۶-۳-۱- بلاک F_2OUT3

F_2OUT3: 2oo3 evaluation of inputs of data type BOOL

این بلاک وضعیت سه ورودی باینری را برای مقدار یک مانیتور می کند. تا زمانی که دو ورودی از سه ورودی یک باشد. خروجی Q یک بوده و در غیر این صورت خروجی Q صفر خواهد بود. مقدار پیش فرض پایه های بلاک صفر می باشد.

فرق این بلاک با بلاک F_2OO3DI این است که این بلاک آنالیز اختلاف زمانی (Discrepancy Analysis) را بین ورودی ها انجام نمی دهد.



شکل ۶-۹- مثال از پیاده سازی لاجیک F_2OUT3

۶-۳-۲- بلاک F_XOUTY

این بلاک تا ۱۶ ورودی باینری را برای وضعیت سیگنال یک در ورودی های IN1 تا ورودی INY را مانیتور می کند. تعداد ورودی های باینری که بایستی مانیتور شود، توسط پارامتر Y تعیین می شود. خروجی Q زمانی یک است که حداقل به تعداد X ورودی از ۱۶ ورودی یک باشد. نکته: در شرایط $X > Y$ یا $X \leq 0$ ، $X > 16$ ، $X \leq 0$ یا $Y \leq 0$ خروجی Q صفر می باشد.

۶-۴- بلاک های مقایسه کننده (comparing two input values)

بلاک های مقایسه کننده که در گروه COMPARE قرار دارند. دو ورودی را بایکدیگر مقایسه می کنند.

COMPARE	
F_CMP_R	[FB313: F_:REAL Comparator for two REAL]
F_LIM_HL	[FB314: F_:Limit Monitoring high Level]
F_LIM_LL	[FB315: F_:Limit Monitoring low Level]
F_CMP_R	دو عدد اعشاری را با هم مقایسه می کند
F_LIM_HL	حد بالای یک مقدار Real را مانیتور می کند.
F_LIM_LL	حد پایین یک مقدار Real را مانیتور می کند.

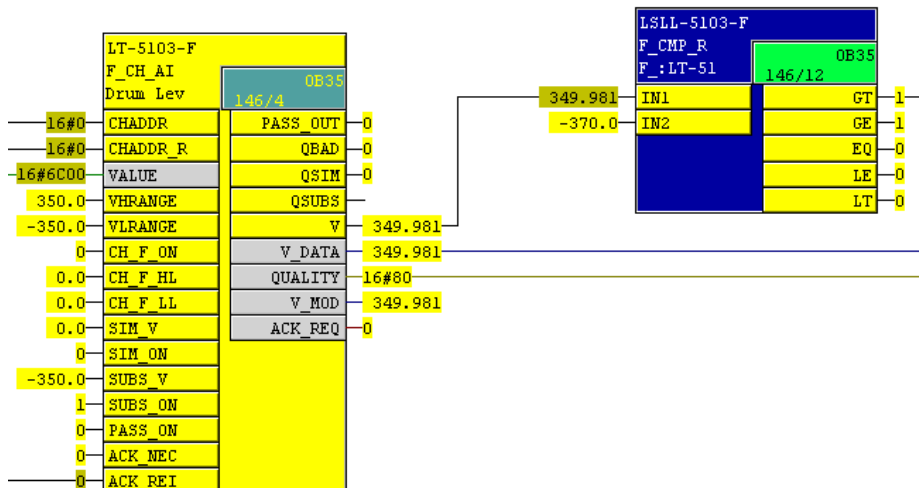
- در بلاک F_CMP_R در صورتی که یکی از ورودی های IN1 یا IN2 یک عدد نامعتبر (NaN) باشد. خروجی های GT و LT به یک تنظیم می شود.

۶-۴-۱- بلاک مقایسه کننده دو عدد اعشاری (F_CMP_R)

F_CMP_R Comparator for two REAL values

دو ورودی IN1 و IN2 با نوع داده F_STRUCT را باهم مقایسه کرده و چهار خروجی F_BOOL را تولید می کنند. مقدار پیش فرض پایه ها صفر می باشد.

Name	Data type	Description
IN1	F_REAL	Input 1
IN2	F_REAL	Input 2
GT	F_BOOL	IN1 > IN2
GE	F_BOOL	IN1 ≥ IN2
EQ	F_BOOL	IN1 = IN2
LT	F_BOOL	IN1 < IN2
LE	F_BOOL	IN1 ≤ IN2



شکل ۶-۱۰- مثالی از پیاده سازی بلاک F_CMP_R

۶-۵- بلاک های Voter (Voter blocks)

این بلاک ها مطابق جدول زیر در گروه Voter از کتابخانه F قرار دارند.

VOTER	
F_1oo2AI [FB317: F_1oo2 evaluation of REAL Inputs with Discrepancy Analysis]	
F_2oo3AI [FB318: F_2oo3 evaluation of REAL Inputs with Discrepancy Analysis]	
F_2oo3DI [FB316: F_2oo3 evaluation of BOOL Inputs with Discrepancy Analysis]	
F_2oo3DI	ارزیابی 2oo3 را برای ورودی های BOOL به همراه آنالیز discrepancy انجام می دهد.
F_2oo3AI	ارزیابی 2oo3 را برای ورودی های REAL به همراه آنالیز discrepancy انجام می دهد.
F_1oo2AI	ارزیابی 1oo2 را برای ورودی های REAL به همراه آنالیز discrepancy انجام می دهد.

۶-۵-۱- بلاک ارزیابی 2oo3 برای ورودی های BOOL (F_2oo3DI)

2oo3 evaluation of inputs of data type BOOL with discrepancy analysis

این بلوک سه سیگنال باینری F_BOOL را برای وضعیت یک مانیتور می کند. اگر حداقل دو ورودی از سه ورودی INX یک باشد، خروجی OUT یک می شود. در غیر این صورت خروجی OUT صفر می شود. خروجی OUTN، معکوس شده (Not) خروجی OUT است.

اگر ورودی DIS_ON به یک تنظیم شود، در آن صورت ورودی ها بایستی در یک زمان مشخص برسند. به این معنی که آنالیز اختلاف زمانی بین ورودی ها انجام می گیرد. اگر اختلاف بین ورودی INX و دو ورودی دیگر INy طولانی تر از زمان اختلاف اختصاص داده شده در ورودی DIS_TIME باشد، یک خطای اختلاف زمانی تشخیص داده شده و در خروجی DIS و DIS_D با مقدار ۱ ذخیره می شود.

در صورتی که ورودی ACK_NEC=0 باشد و خطای اختلاف زمانی تشخیص داده شود، در آن صورت خطای اختلاف زمانی به صورت اتوماتیک تصدیق می شود.

خروجی ACK_REQ = 1 به این معنی است که خطا رفع شده و کاربر بایستی از طریق ورودی ACK عمل صدیق خطا را باید انجام دهد.



PY-5307-F F_2003DI 50 ms F :2003 152/16 IN1 OUT IN2 OUTN IN3 DIS 0 DIS ON DIS D 1s DIS TIME ACK REQ 0 ACK NEC 0 ACK	Inputs/outputs <table border="1"> <thead> <tr> <th></th> <th>Name</th> <th>Data type</th> <th>Description</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td rowspan="6">Inputs:</td> <td>IN1</td> <td>F_BOOL</td> <td>Input 1</td> <td>0</td> </tr> <tr> <td>IN2</td> <td>F_BOOL</td> <td>Input 2</td> <td>0</td> </tr> <tr> <td>IN3</td> <td>F_BOOL</td> <td>Input 3</td> <td>0</td> </tr> <tr> <td>DIS_ON</td> <td>F_BOOL</td> <td>1 = Discrepancy analysis</td> <td>0</td> </tr> <tr> <td>DIS_TIME</td> <td>F_TIME</td> <td>Discrepancy time in ms</td> <td>1000</td> </tr> <tr> <td>ACK_NEC</td> <td>F_BOOL</td> <td>1 = Acknowledgment necessary</td> <td>0</td> </tr> <tr> <td rowspan="7">Outputs:</td> <td>ACK</td> <td>F_BOOL</td> <td>Acknowledgment</td> <td>0</td> </tr> <tr> <td>OUT</td> <td>F_BOOL</td> <td>Output</td> <td>0</td> </tr> <tr> <td>OUTN</td> <td>F_BOOL</td> <td>NEGATING OUTPUT</td> <td>1</td> </tr> <tr> <td>DIS</td> <td>F_BOOL</td> <td>DISCREPANCY ERROR</td> <td>0</td> </tr> <tr> <td>DIS_D</td> <td>BOOL</td> <td>DISCREPANCY ERROR DATA</td> <td>0</td> </tr> <tr> <td>ACK_REQ</td> <td>BOOL</td> <td>ACKNOWLEDGMENT REQUEST</td> <td>0</td> </tr> </tbody> </table>		Name	Data type	Description	Default	Inputs:	IN1	F_BOOL	Input 1	0	IN2	F_BOOL	Input 2	0	IN3	F_BOOL	Input 3	0	DIS_ON	F_BOOL	1 = Discrepancy analysis	0	DIS_TIME	F_TIME	Discrepancy time in ms	1000	ACK_NEC	F_BOOL	1 = Acknowledgment necessary	0	Outputs:	ACK	F_BOOL	Acknowledgment	0	OUT	F_BOOL	Output	0	OUTN	F_BOOL	NEGATING OUTPUT	1	DIS	F_BOOL	DISCREPANCY ERROR	0	DIS_D	BOOL	DISCREPANCY ERROR DATA	0	ACK_REQ	BOOL	ACKNOWLEDGMENT REQUEST	0
	Name	Data type	Description	Default																																																				
Inputs:	IN1	F_BOOL	Input 1	0																																																				
	IN2	F_BOOL	Input 2	0																																																				
	IN3	F_BOOL	Input 3	0																																																				
	DIS_ON	F_BOOL	1 = Discrepancy analysis	0																																																				
	DIS_TIME	F_TIME	Discrepancy time in ms	1000																																																				
	ACK_NEC	F_BOOL	1 = Acknowledgment necessary	0																																																				
Outputs:	ACK	F_BOOL	Acknowledgment	0																																																				
	OUT	F_BOOL	Output	0																																																				
	OUTN	F_BOOL	NEGATING OUTPUT	1																																																				
	DIS	F_BOOL	DISCREPANCY ERROR	0																																																				
	DIS_D	BOOL	DISCREPANCY ERROR DATA	0																																																				
	ACK_REQ	BOOL	ACKNOWLEDGMENT REQUEST	0																																																				

ACK_NEC	F_BOOL	Acknowledgment necessary=1 با یک شدن این ورودی عمل تصدیق فالت در بلاک ضروری می باشد.
---------	--------	---

نکته: در صورت وقوع یک خطا در فرمت داده ایمنی (safety data format) بلاک داده مربوطه (instance DB) یک F-STOP تریگر می شود. سپس یک پیغام تشخیص خطا (iagnostic event) در بافر F-CPU وارد می شود.

"Safety program: Error in safety data format in DB" (Event ID 16#75DA)

۶-۵-۲- بلاک F_2003AI

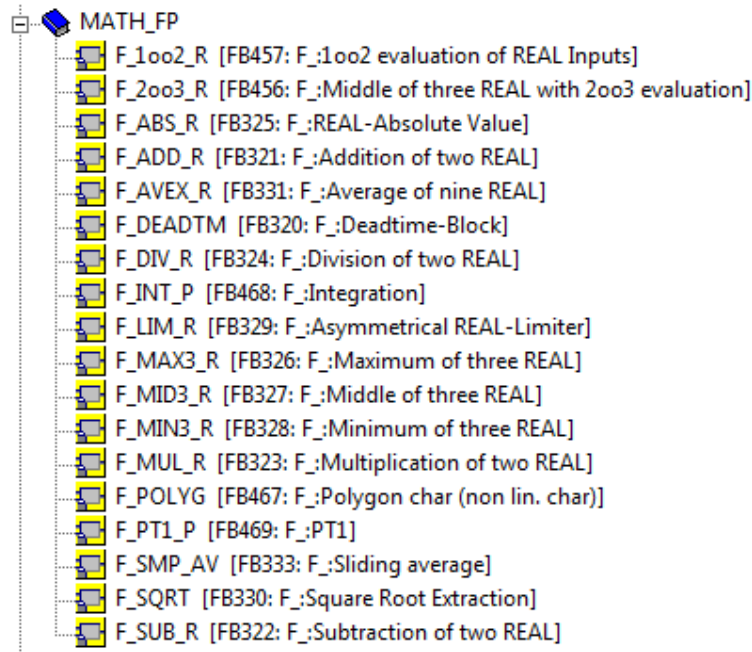
در صورتی که بخواهیم خروجی V یک بلاک درایور F_CH_AI را به یکی از ورودی های بلاک F_2003AI متصل کنیم. بایستی ورودی $QBADx$ بلاک F_2003AI به خروجی $QBAD$ بلاک F_CH_AI وصل کنیم.

۶-۵-۳- بلاک افزونه کردن ورودی ها (RED_IN)

این بلاک برای تنظیم کانال های ماژول های ورودی به صورت افزونه به کار می شود. به صورت اتوماتیک با کامپایل چارت و در صورت افزونه تعریف شدن کارت های I/O در چارت های @ درج می شود.

۶-۶- مجموعه بلاک های محاسباتی F اعشاری (MATH_FP)

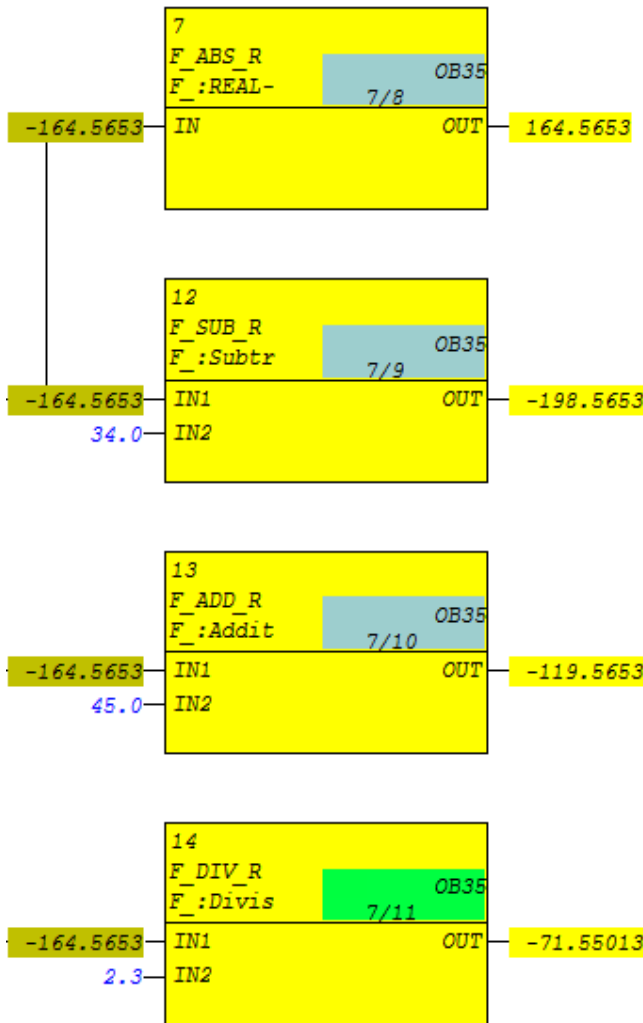
این بلاک های برای عملیاتی محاسباتی بر روی ورودی های نوع F به کار می رود.



شکل ۶-۱۱- مجموعه *MATH_FP* - بلاک های محاسباتی

۶-۶-۱ - بلاک *F_ABS_R* (Absolute value of a REAL value)

این بلاک مقدار مطلق *F_REAL* در ورودی *IN* را در خروجی *OUT* تولید می کند.



شکل 126-- مثالی از بلاک های محاسباتی

۶-۲-۶ بلاک (F_1002_R)

1002 evaluation of inputs of data type REAL

این بلاک بسته به مقدار ورودی $QBAD1$ ، داده دریافت شده از یکی از ورودی های $IN1$ و $IN2$ را به خروجی OUT منتقل می کند.

$$QBAD1 = 0 \gg OUT = IN1$$

$$QBAD1 = 1 \gg OUT = IN2$$

اگر هر دو ورودی $IN1$ و $IN2$ نامعتبر باشد ($QBAD1=1$ و $QBAD2=1$)، خروجی OUT نیز نامعتبر و خروجی $QBAD=1$ خواهد بود.

اگر اختلاف بین ورودی $IN1$ و $IN2$ بیش از مقدار تلورانس $DELTA$ اختصاص داده شده باشد، یک خطای اختلاف ($discrepancy\ error$) را تشخیص داده و خروجی های $DIS1$ و $DIS2$ به شرح زیر فعال می شوند.

$DIS1 = 1$ if $IN2$ is output at the OUT output.

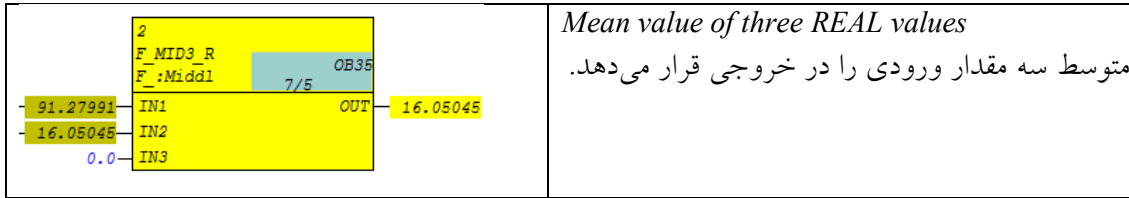
$DIS2 = 1$ if $IN1$ is output at the OUT output.

<i>DELTA</i>	<i>F_REAL</i>	مقدار تولورانس در اختلاف بین دو ورودی را مشخص می کند.

	<p>$IN1 < IN2, IN1 >>>> Out, DIS2=1$</p>
<p>$QBAD1=1 >>>> IN2 >>> OUT, DIS1=1$</p>	

۶-۶-۳- بلاک انتخاب حداکثر مقدار بین سه ورودی (F_MAX3_R)

	<p><i>Maximum of three REAL values</i></p> <p>این بلوک ورودی های $IN1$، $IN2$ و $IN3$ را مقایسه و حداکثر آنها را در خروجی OUT انتقال می دهد. مقدار پیش فرض برای همه ورودی ها برابر مقدار - $3,402823e+38$ (بزرگترین عدد حقیقی منفی) است. لذا می توان مقدار حداکثر، بین دو ورودی را نیز حساب کرد.</p>
	<p><i>Square root of a REAL value</i></p> <p>ریشه مربع ورودی را محاسبه و در خروجی OUT قرار می دهد.</p>
	<p><i>Minimum of three REAL values</i></p> <p>حداقل بین سه ورودی را در خروجی قرار می دهد.</p>



۶-۶-۴ - بلوک انتگرال گیر (F_INT_P)

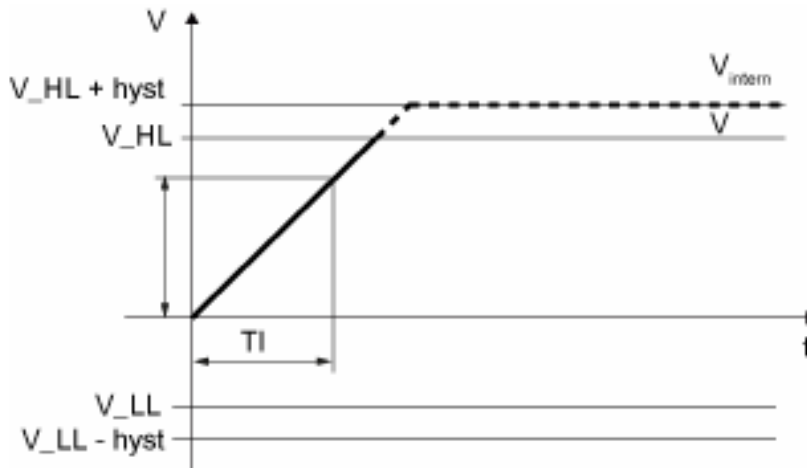
Integration function with integration and track mode

بلوک F_INT_P از زیرمجموعه $MATH_FP$ معادل F بلوک انتگرال گیر معمولی در کتابخانه PCS7 می باشد. این بلوک می تواند در دو مد متفاوت زیر کار کند.

- Integration mode
- TRACK mode

۱. مد انتگرالی

در مد انتگرالی، خروجی V با یک سیگنال مثبت در ورودی U افزایش و با سیگنال منفی در ورودی U کاهش می یابد. این بلوک با تشکیل جمع کل مقادیر ورودی با توجه به قانون دوزنقه ای برای هر فاصله نمونه برداری (TS) در مد انتگرالی کار می کند. نتیجه $V_{internal}$ مطابق شکل زیر در محدوده $V_{HL} + hyst$ تا $V_{LL} - hyst$ قرار می گیرد. نتیجه حاصله پس از این که به مقادیر V_{HL} و V_{LL} محدود می شود، در خروجی V نوشته می شود.



$$Hyst = HYS / 100 * (V_{HL} - V_{LL})$$

شکل ۶-۱۳ - پاسخ شیب بلوک F_INT_P

خروجی V مطابق فرمول زیر محاسبه می شود.

$$V_x = V_{x-1} + U_x \cdot \frac{T_s}{T_I}$$

VX Current internal output value

VX-1 Last internal output value (*Vinternal*)

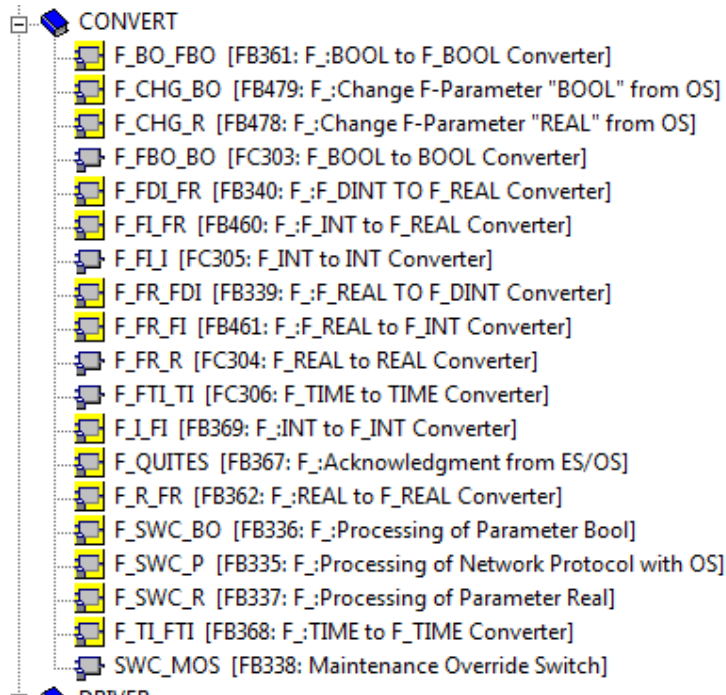
Ts Sampling time (time elapsed between two F-Block processing cycles) in seconds

TI Integration time in seconds

UX Current input value

نکته: می توان با دادن یک مقدار ثابت به ورودی بلوک *F_INT_P*، افزایش سطح یک مخزن را تا ۱۰۰ درصد شبیه سازی کرد.

۶-۷- بلاک های مبدل (Convert)



۶-۷-۱- بلاک *F_SWC_BO*

F_SWC_BO: Processing of a parameter of data type *F_BOOL* for operator input via the OS
 بلاک *F_SWC_BO* امکان ایجاد تغییرات در پارامترهای *F* نوع داده *BOOL* را در برنامه ایمنی از سیستم مانیتورینگ ایستگاه اپراتوری فراهم می کند. به عبارت دیگر برای ورود یا دریافت یک داده *Bool* از *OS* در برنامه *F* بکار می رود. خروجی *OUT* به ورودی که در برنامه *F* که بایستی تغییر داده شود متصل می گردد. خروجی های *OUT* و *AKT_VAL* می تواند به طور مستقل از طریق ورودی های *S* و *R*، *Set* یا *Reset* شوند. خروجی های *OUT* و *AKT_VAL* با یک لبه مثبت در *S*، *Set* می شوند. اولویت با ورودی *Reset* است، در نتیجه، تا زمانی که ورودی *R* برابر با ۱ باشد ($R = 1$)، این خروجی ها *Reset* خواهند شد.

Inputs/outputs

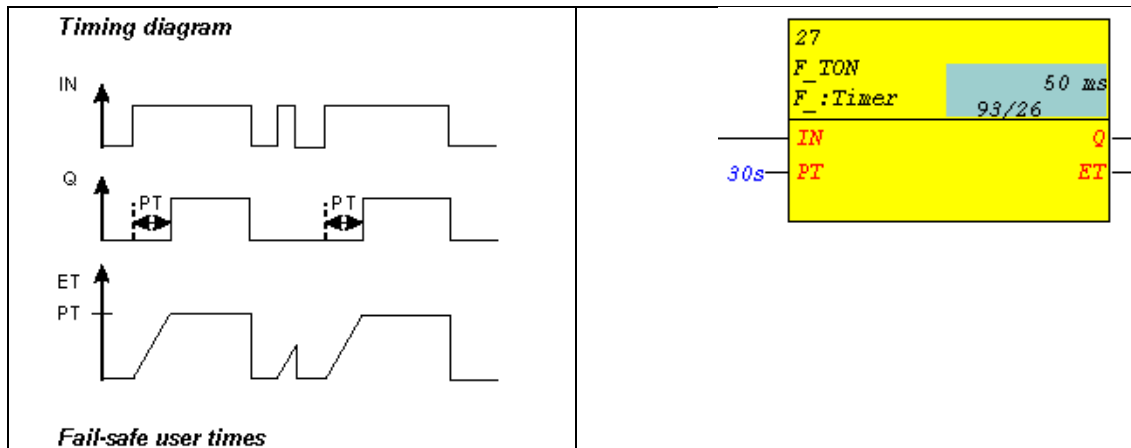
	<u>Name</u>	<u>Data type</u>	<u>Description</u>	<u>Default</u>
Inputs:	S	F_BOOL	Set input	0
	R	F_BOOL	Reset input	0
	CS_VAL	F_BOOL	Cold restart	0
Outputs:	OUT	F_BOOL	Current value of the operated parameter	0
	AKT_VAL	BOOL	Current value of the operated parameter for the OS	0

۶-۷-۲- بلاک F_FR_FDI

تبدیل داده از F_REAL به F_DINT را انجام می دهد.

۶-۷-۳- بلاک تایمر تاخیر در روشن (F_TON: Timer switch-on delay)

با لبه بالا رونده سیگنال در ورودی IN تایمر شروع و پس از زمان PT خروجی Q یک می شود. این تایمر با صفر شدن ورودی IN ریست می شود.



۶-۷-۴- تصدیق خطا توسط کاربر (ACK_REI)

Regeneration of F-I/O Modules or depassivation of F- I/O modules

برای فعال کردن یک کانال پس از *Passivation* و یا به اصطلاح فعال سازی *Regenerate* کردن کانال، بایستی پس از رفع خطا، یک تصدیق از طریق پایه *ACK_REI* بلاک های درایور کانال آنالوگ و دیجیتال انجام داد. برای این منظور از فانکشن بلاک *F_QUITES* کتابخانه *F* استفاده می کنیم. به این ترتیب که به پایه ورودی *IN* بلاک *F_QUITES* مقدار ۶ و پس از حداکثر یک دقیقه مقدار ۹ را اعمال می کنیم. با این کار در خروجی بلاک *F_QUITES* (متصل به ورودی *ACK_REI* بلاک های درایور) به مدت یک سیکل مقدار ۱ ایجاد شده

و خطا ریست می شود. با این کار بلاک به عملیات عادی یعنی تولید مجدد مقدار از روی سیگنال کانال (Regenerate) ادامه می دهد.

عمل فعال سازی ACK_REI را می توان از طریق ورود مقدار ۶ و ۹ در یکی از صفحات HMI پیاده سازی کرد. که در ادامه تشریح می شود.

۶-۷-۵- بلاک F_QUITES

Fail-safe acknowledgement via the ES/OS

بلاک F_QUITES امکان *fail-safe acknowledgement* را از یک کامپیوتر *non-fail-safe ES/OS* فراهم می کند. این کار امکان کنترل فعال سازی دوباره (*reintegration*) کانال های I/O پس از *Passivate* شدن را فراهم می کند. عمل *Acknowledgment* در دو مرحله انجام می شود.

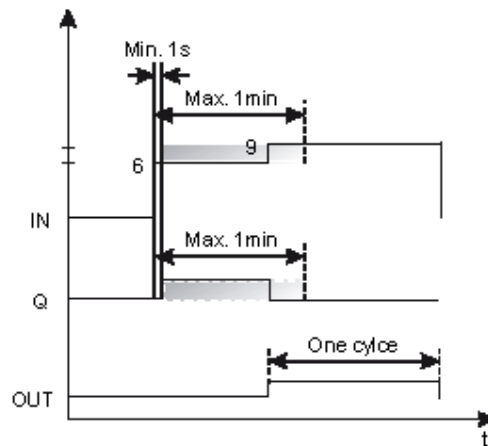
Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	INT	Input	0
Outputs:	OUT	F_BOOL	ACKNOWLEDGMENT OUTPUT	0
	Q	BOOL	Status of the time evaluation	0

۱- ابتدا مقدار ۶ به ورودی IN اعمال می شود.

۲- سپس در مدت کمتر از یک دقیقه ورودی IN بلاک از مقدار "۶" به مقدار "۹" تغییر می یابد.

بلوک F پس از این که مقدار ورودی به ۶ تغییر یافت ارزیابی می کند. که آیا مقدار ورودی IN در حداقل زمان ۱ ثانیه یا در حداکثر زمان ۱ دقیقه به مقدار ۹ تغییر یافته است یا نه. سپس مقدار یک را در خروجی OUT بلاک (خروجی برای *acknowledgment*) برای مدت زمان یک سیکل قرار می دهد.



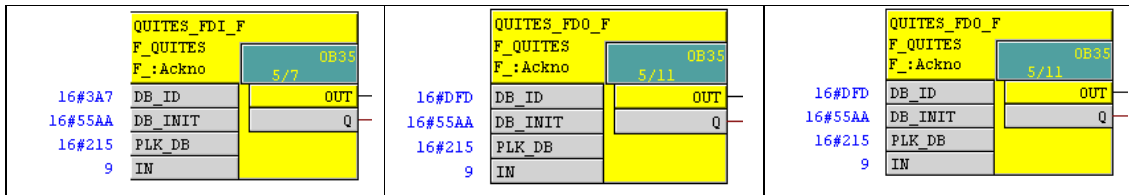
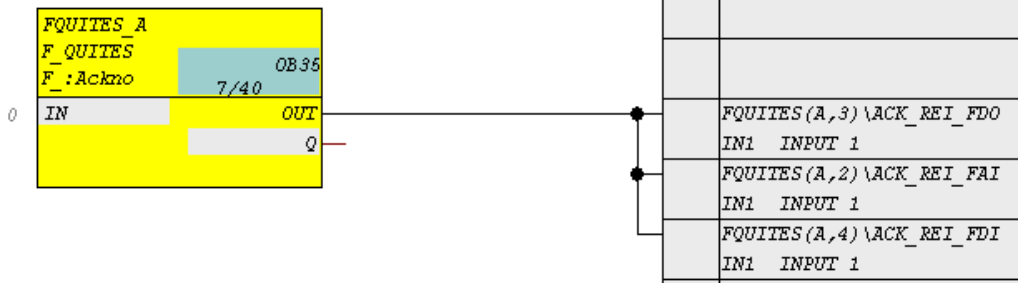
اگر یک مقدار نامعتبر وارد شود و یا اگر تغییر به مقدار "۹" قبل از ۱ ثانیه و یا پس از ۱ دقیقه رخ دهد، ورودی IN به صفر ست شده و دو مرحله بالا باید تکرار شود.

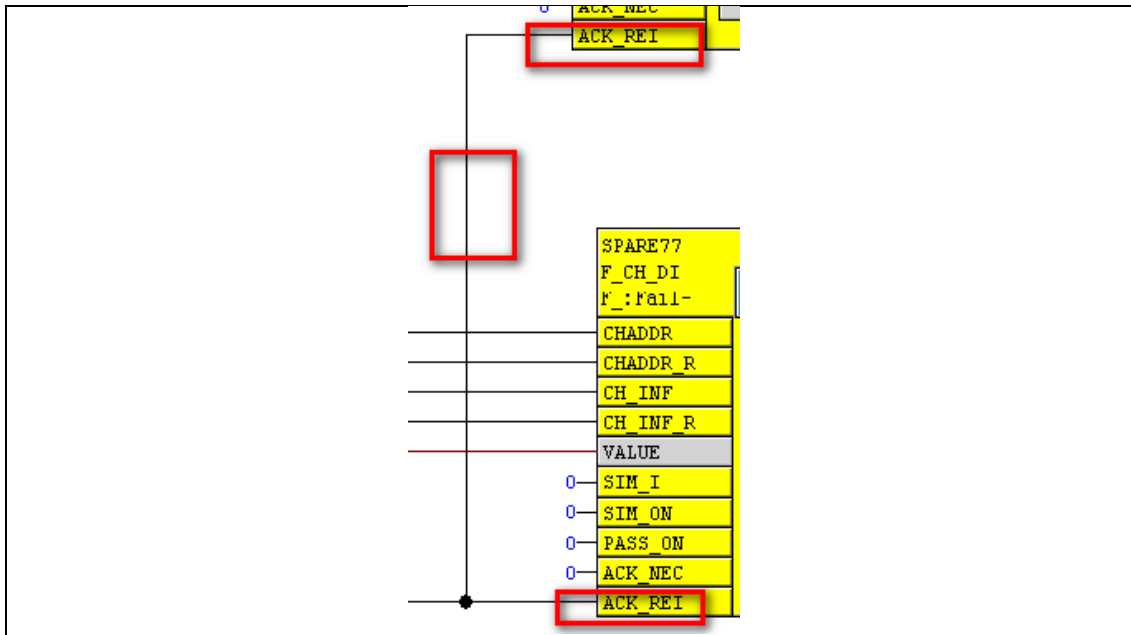
در طول زمانی که مقدار ورودی *IN* از "۶" به "۹" باید تغییر کند، خروجی *Q* *non-fail-safe* بلاک به یک ست می شود. در دیگر موارد *Q* دارای یک مقدار صفر می باشد.

دو پارامتر ورودی *IN* و خروجی *Q* بلاک دارای *attribute S7_m_c* می باشد. به این معنی که این دو پارامتر در *OS* قابل دسترسی می باشد. برای این منظور معمولاً برای هر یک از دسته ماژول های ورودی/خروجی دیجیتال و آنالوگ (*F_DI, F_DO, F_AI*) یک بلاک *F_QUITES* در یک چارت *CFC* با عنوان *QUITES* قرار داده و خروجی *OUT* هر یک از بلاک های *F_QUITES* را به ورودی *ACK_REI* بلاک های درایور کانال متصل می کنیم. مطابق شکل زیر در یک پروژه نمونه در چارت *QUITES* سه بلاک *F_QUITES* با عناوین زیر برای عمل *Acknowledgement* استفاده شده است.

- *QUITES_FDI_F*
- *QUITES_FDO_F*
- *QUITES_FAI_F*

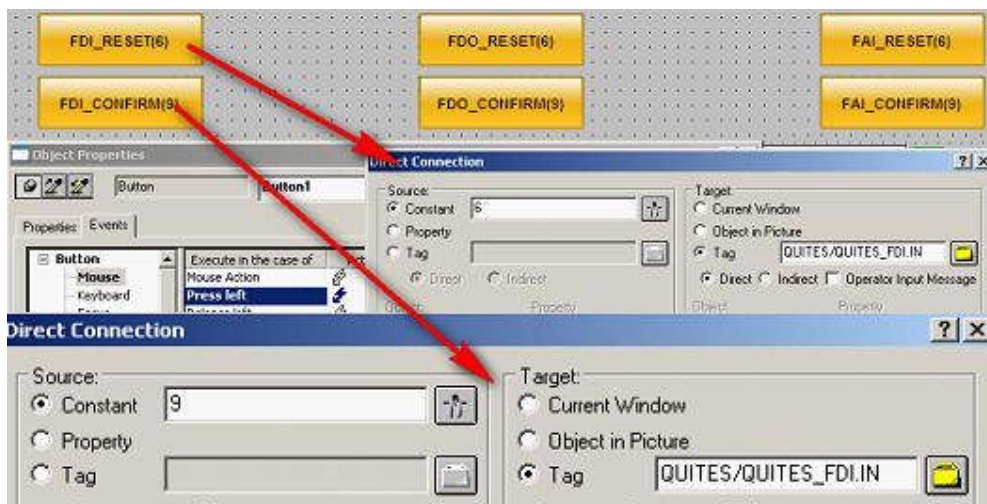
در شکل زیر از بلاک *F_QUITES*





۶-۷-۶- پیاده سازی user acknowledgment از طریق سیستم OS

۱. سه عدد بلاک F_QUITES برای بلاک های DI ، DO و DI به برنامه F در داخل یک چارت CFC با نام $QUITES$ اضافه می کنیم.
۲. خروجی OUT بلاک های F_QUITES را به ورودی ACK_REI درایورهای کانال سیگنال متصل کنید.
۳. یک $Button$ برای نوشتن مقدار ۶ ($Acknowledge\ value$) و یک $Button$ دیگر برای نوشتن مقدار ۹ ($second\ acknowledgment\ step$) در یکی از OS های $Picture$ پیکربندی کنید.

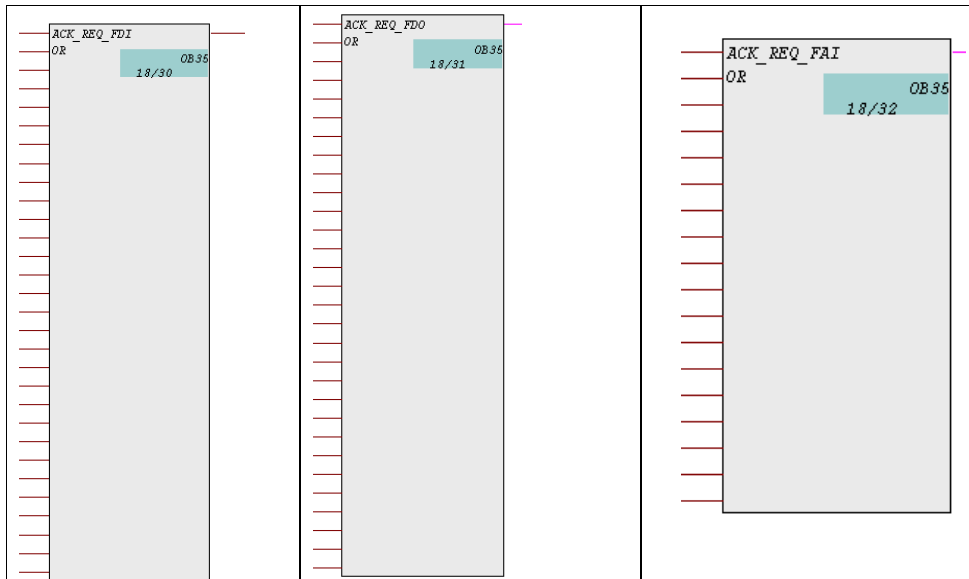


شکل ۶-۱۴- اعمال مقادیر ۶ و ۹ به ورودی IN بلاک F_QUITES

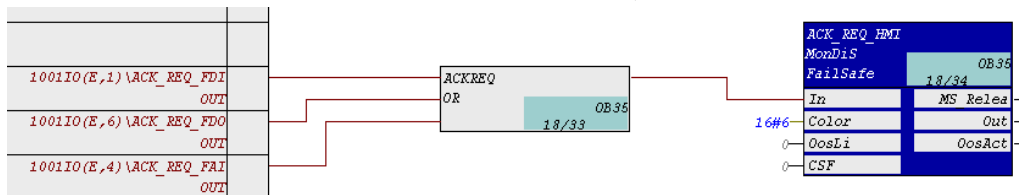
۶-۷-۷- پیاده سازی F-user acknowledgment در OS

برای این که پیغام درخواست acknowledgment را برای Depassivation ورودی ها را در OS نمایش دهیم. تمام پایه های خروجی ACK_REQ بلاک های درایور کانال های ورودی /خروجی را با استفاده از یک بلاک OR در بخش استاندارد برنامه کاربر، گروه بندی کرده و مقدار خروجی OR را از طریق یک بلاک DIG_MON در OS قابل دسترس می کنیم. به عنوان مثال در در یک برنامه به صورت زیر عمل می کنیم:

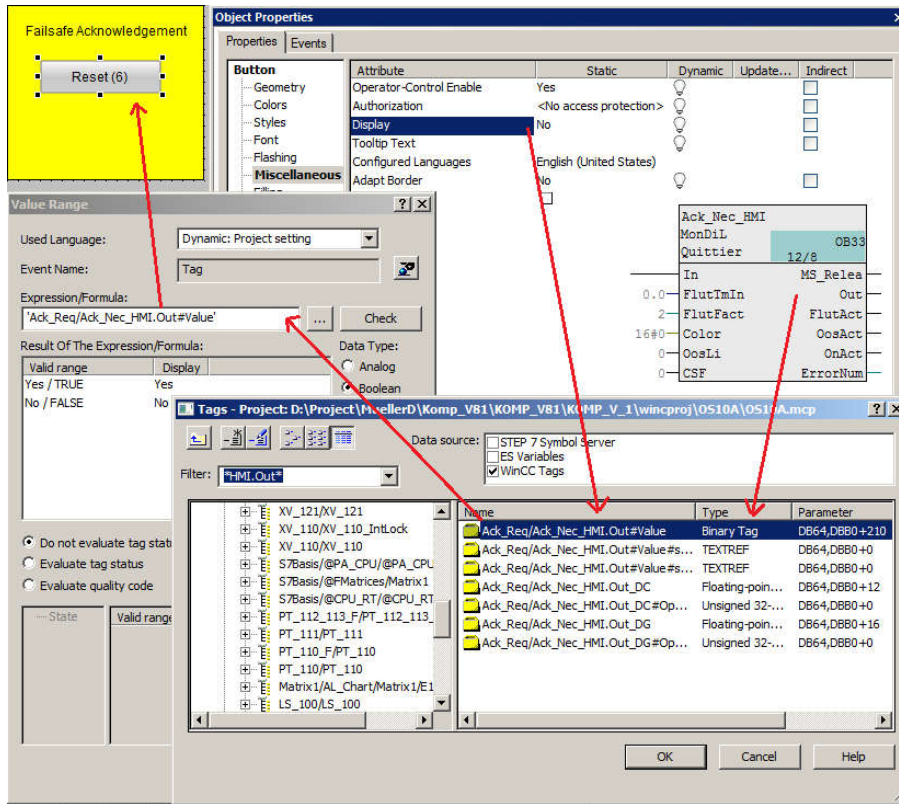
۱. تمام خروجی های ACK_REQ بلاک های درایور کانال دیجیتال ورودی را به یک بلاک OR با عنوان ACK_REQ_FDI متصل می کنیم.
۲. تمام خروجی های ACK_REQ بلاک های درایور کانال دیجیتال خروجی را به یک بلاک OR با عنوان ACK_REQ_FDO متصل می کنیم.
۳. تمام خروجی های ACK_REQ بلاک های درایور کانال آنالوگ ورودی را به یک بلاک OR با عنوان ACK_REQ_FAI متصل می کنیم.



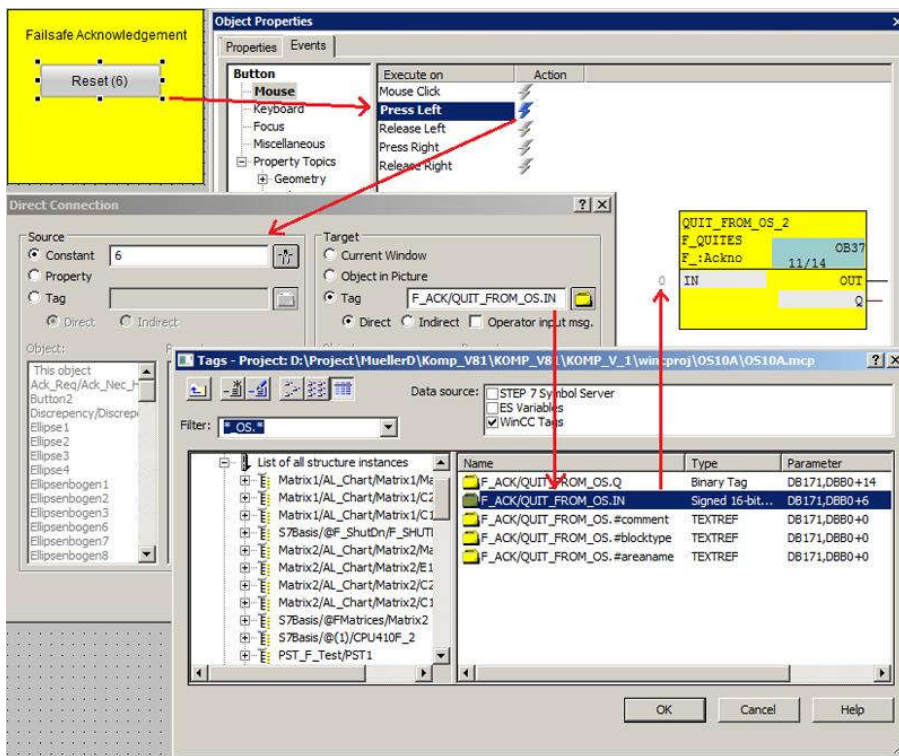
۴. سپس خروجی این سه بلاک OR را به یک بلاک OR دیگر و خروجی آن را به یک بلاک مانیتورینگ سیگنال دیجیتال MonDis متصل می کنیم.



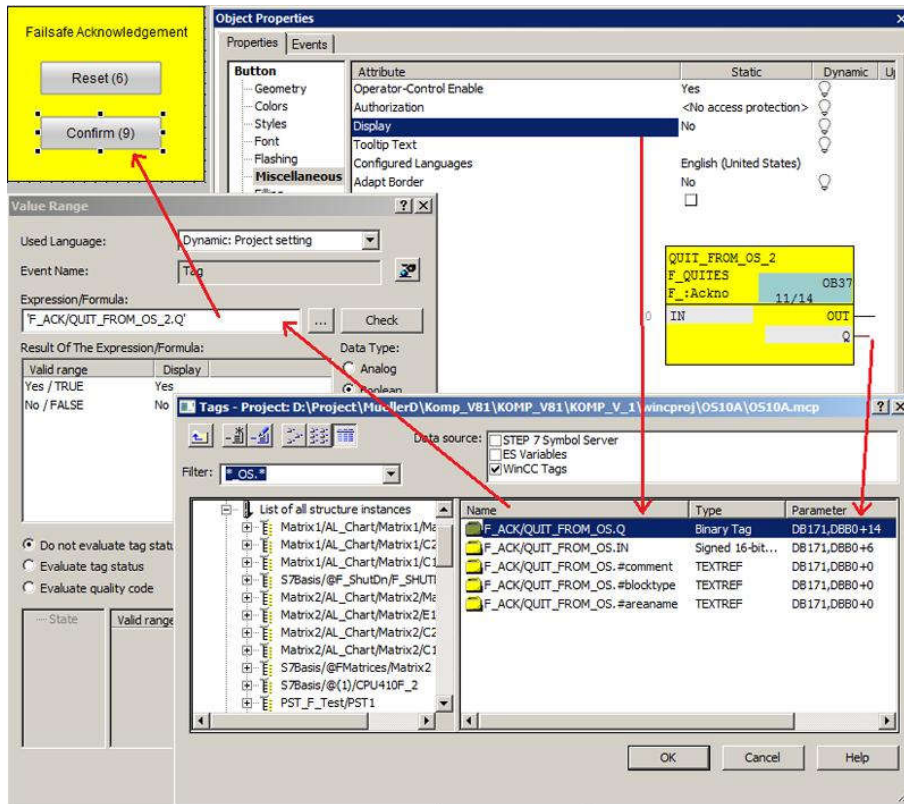
در صورت فعال شدن یک acknowledgment (ACK_REQ=1)، فیلد acknowledge زرد (yellow) شده و دکمه (6) reset در OS ظاهر می گردد.



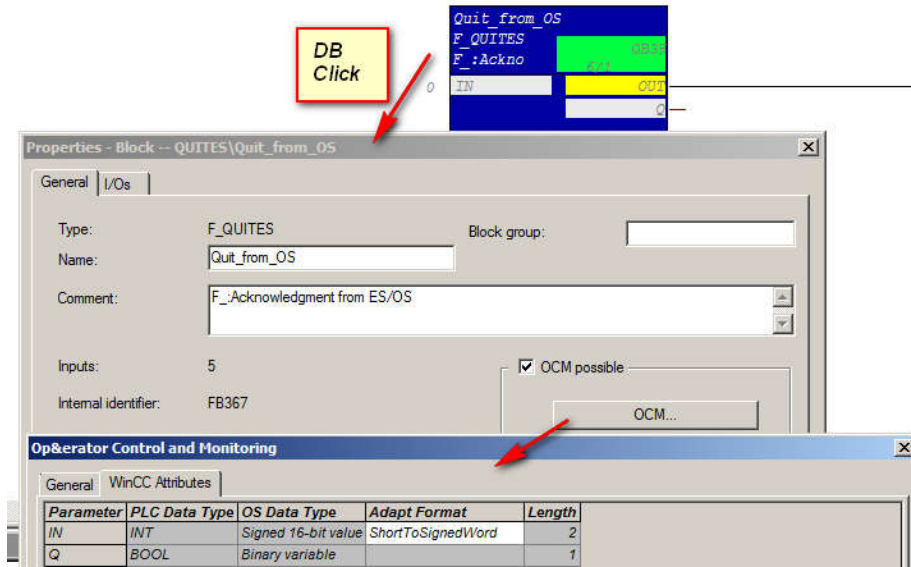
۵. برای نوشتن مقدار ۶ در ورودی IN بلاک F_QUITES بر روی دکمه (6) کلیک کنید.



۶. اگر خروجی Q بلاک F_QUITES ، ست شود، دکمه $acknowledge$ دوم یعنی (9) $Reset$ ظاهر می شود. این خروجی برای مدت ۶۰ ثانیه یک می ماند.

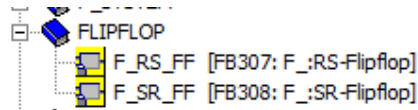


اگر مقدار ۹ در بازه یک تا ۶۰ ثانیه به ورودی IN بلاک F_QUITES نوشته شود، خروجی OUT بلاک به مدت یک سیکل به یک ست می شود. در آن صورت بلاک های درایورهای کانال که ورودی ACK_REI آنها به این خروجی متصل شده است، تولید دوباره (reintegrate) مقدار در خروجی بلاک را شروع می کنند. اگر هیچ درخواستی (acknowledgment prompt) برای تصدیق نباشد، یعنی ورودی بلاک ACK_REQ_HMI یک نشده باشد. دکمه های طراحی شده در OS مخفی خواهد شد.
نکته: برای این که بتوان از OS به مقادیر خروجی و ورودی بلاک F_QUITES دسترسی داشت، بایستی در پنجره پراپرتی این بلاک گزینه OCM را تیک زد.



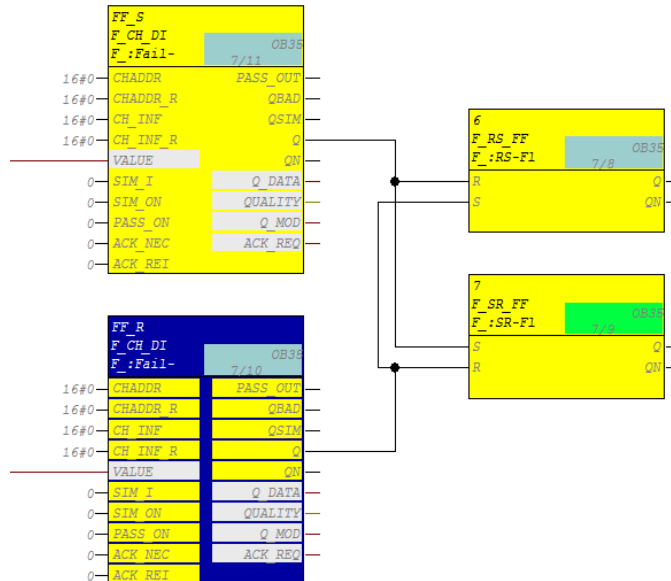
۶-۸- بلاک های فلیپ فلاپ

این دسته بلاک شامل دو نوع بلاک فلیپ فلاپ *RS* و *SR* می باشد.



در بلاک *RS* اولویت با ورودی ریست و در بلاک *SR* اولویت با ورودی *S* می باشد.

- *F_RS_FF*: *RS Flip-Flop, resetting dominant*
- *F_SR_FF*: *SR Flip-Flop, setting dominant*



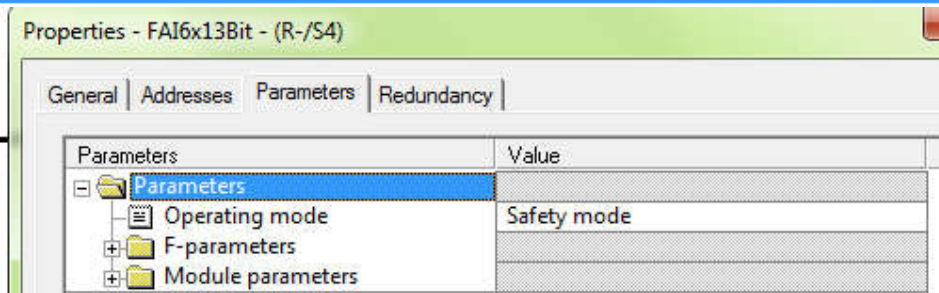
شکل ۶-۱۵- بلاک های فلیپ فلاپ نوع *F*

۶-۹- کامپایل برنامه F

اگر یک برنامه S7 حاوی چارت های CFC دارای بلوک های F باشد، این بلاک های F در زمان کامپایل چارت های CFC، کامپایل شده و اقدامات لازم برای از بین بردن خطاها پیش بینی و همچنین بررسی های اضافی تر در ارتباط با برنامه ایمنی (safetyrelevant checks) انجام می گیرد.

نکته: پس از درج ماژول F-I/O بایستی در پنجره پراپرتی آن پارامترهای F تنظیم گردد. به عنوان مثال در صورتی که گزینه Operating Mode برای ماژول F برای مد Safety mode تنظیم نشود. در زمان کامپایل خطای زیر ظاهر خواهد شد.

A: Safety program compiler: S7F_FSYSTEMS_BUILD_06.01.01.00_03.10.00.01 - 15.May.2016 11:23:57
E: Block 'PT1001': Module not configured or parameter VALUE not supplied [Configure a module for this driver in HW Config and link it in SIMATIC Manager with

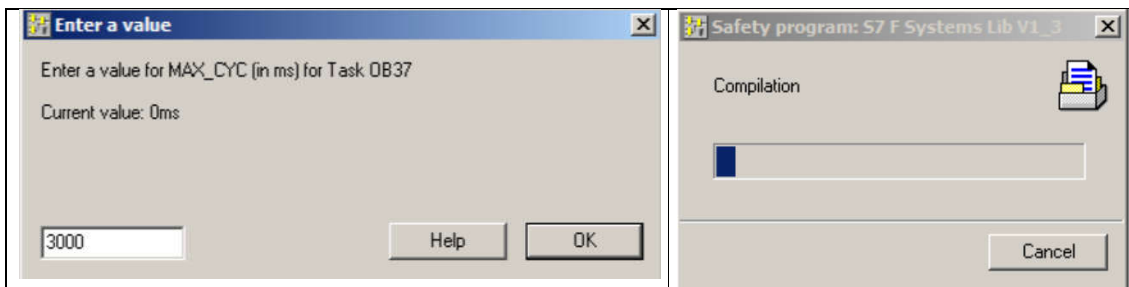


نمایش خطای کامپایل برای ماژولی که پارامترهای آن تنظیم نشده است.

۶-۹-۱- مقداردهی به حداکثر زمان مانیتورینگ سیکل F

Parameterizing the maximum F cycle monitoring time

ماژول F-CPU زمان F-cycle را در هر سیکل اینترپت OB که حاوی F-runtime group ها می باشد، مانیتور می کند. وقتی برای اولین بار برنامه S7-F، کامپایل می شود، برای هر OB اینترپت سیکلیک، پنجره ای برای ورود یک مقدار برای حداکثر سیکل زمان (maximum cycle time: MAX_CYC) که می تواند بین دو فراخوانی OB سپری شود، نمایش داده می شود.

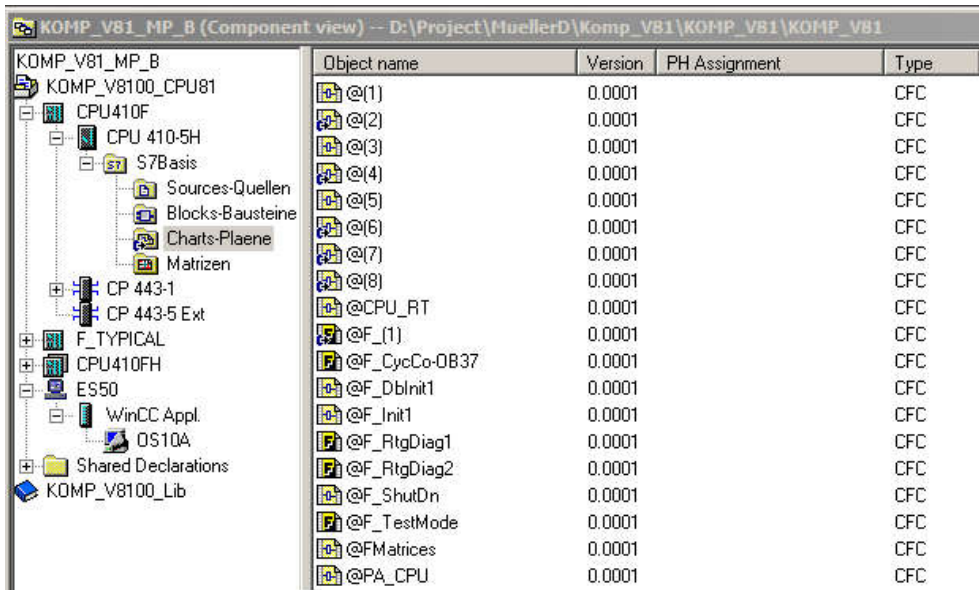


مقدار پیش فرض برای زمان F-Cycle، ۳۰۰۰ میلی ثانیه (حداکثر زمان مانیتورینگ) است.

نکته: مقدار پیش فرض F-Cycle time را می توان توسط ورودی MAX_CYC، بلاک F_CYC_CO-OB3x در چارت @F_CycCo-OB3x تغییر داد.

۶-۹-۲- کامپایل برنامه S7

در زمان کامپایل، برای برنامه S7، به صورت خودکار بلاک های درایور تشخیص خطا (*diagnostics drivers*) در چارت های سیستمی @ و بخش های خاص F درج می شود. بلاک های F-System در چارت های @F_XXXX ذخیره می شود.



درج بلاک های @F.. در زمان کامپایل برنامه

نکته: درج، اتصال بلاک ها به هم و تخصیص پارامتر برای بلوک های سیستمی نوع F، به طور خودکار در طول فرایند تکمیل می شود و نباید تغییر کند. همچنین نباید بلاک های F درج شده به صورت خودکار را در داخل چارت ها تغییر داده و یا حذف کنید.

۶-۹-۳- دانلود برنامه به CPU و مد ایمنی (Safety mode and downloading the safety program)

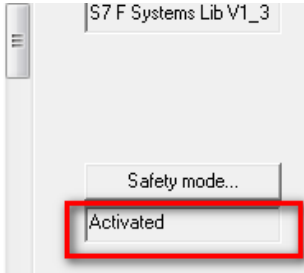
مد ایمنی برنامه F در F-CPU را می توان به طور موقت غیرفعال و دوباره فعال کرد. این مشخصه امکان ایجاد تغییرات در برنامه ایمنی را در مد اجرا (RUN) فراهم می کند.
یک سیستم S7-400 F/FH که حاوی یک برنامه fail-safe می باشد. موقع راه اندازی به طور خودکار به مد ایمنی می رود. در مد ایمنی تمام توابع تشخیص خطا در سیستم حاضر بوده و برنامه fail-safe کاربر فعال می شود. در این وضعیت امکان تغییر یا اصلاح در برنامه ایمنی در زمانی که مد کاری RUN می باشد، وجود ندارد.

به منظور فراهم کردن امکان ایجاد تغییرات در پارامترهای *fail-safe*، از مدآنلاین *CFC* یا دانلود تغییرات ایجاد شده در برنامه *fail-safe* به *CPU*، بخشی از توابع تشخیصی (*diagnostics*) بایستی خاموش شود. لذا مد ایمنی بایستی قبل از اینکه تغییرات به صورت آنلاین اعمال شود و یا دانلود شود، غیرفعال شود. قبل از غیرفعال کردن مد ایمنی، باید اطمینان حاصل شود که فرآیند در حالت خطرناک نبوده و توسط یک اپراتور در طول این مدت مانیتور می شود. برای دانلود تغییرات انجام شده در برنامه در «مد ایمنی غیرفعال شده» بخش های مانیتورینگ خاموش، تغییرات نرم افزاری شناسایی شده و *F-STOP* تریگر می شود. شناسایی فالت های تصادفی در سخت افزار ادامه می یابد و توابع تشخیصی برای ماژول ها فعال می ماند. پردازش برنامه ایمنی همچنان ادامه می یابد، تا اطمینان حاصل شود که تقاضای توقف از فیلد منجر به توقف ایمنی می شود. همین که تغییرات انجام شد و یا در پایان پروسه دانلود، مد ایمنی باید بلافاصله فعال شود.

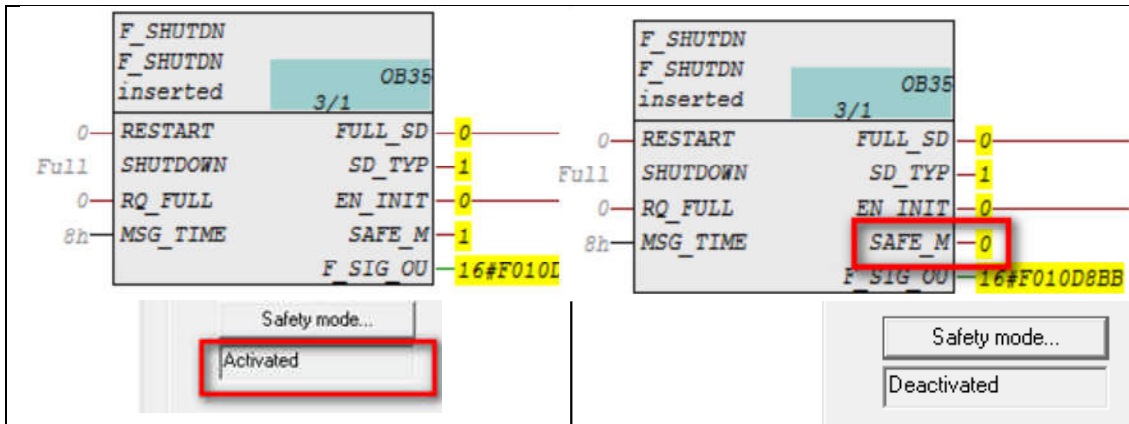
۶-۹-۴- غیرفعال کردن مد ایمنی (Deactivating safety mode)

در زمان کامپایل یا دانلود مد ایمنی می تواند از طریق یک پنجره که اتوماتیک نمایش داده می شود، غیرفعال/فعال شود. این کار همچنین از طریق برنامه *SIMATIC Manager* نیز قابل انجام است. برای این کار، در پنجره *SIMATIC Manager* پس از انتخاب *CPU* که حاوی برنامه ایمنی می باشد، گزینه *Options > Edit Safety Program* را اجرا می کنیم. فیلد زیر دکمه «*Safety Mode...*» را در پنجره باز شده نشان می دهد که می توان از طریق آن مد ایمنی را فعال یا غیر فعال کرد.

FC55	FC55	4ee9	
FC774	FC774	a88d	
FC775	FC775	54d5	
FC776	FC776	f739	
FC777	FC777	2a11	
FC796	FC796	eb72	
FB308	F_SR_FF	9ebe	b56d
FB325	F_ABS_R	7e9d	4885
FB326	F_MAX3_R	c14f	f93f
FB327	F MID3 R	ec2c	ea98



نکته: اگر برنامه ایمنی با برنامه ایمنی موجود در *F-CPU* مطابقت نداشته باشد و یا ارتباط با *F-CPU* قطع شده باشد، در فیلد ذکر شده ظاهر عبارت *unknown* خواهد شد. در صورتی که بین ایستگاه مهندسی و *CPU* اتصال وجود داشته باشد، وضعیت فعلی برنامه ایمنی نمایش داده می شود و می توان با کلیک روی دکمه *Safety Mode...* می توان مد ایمنی را تغییر داد. ولی قبل از تغییر مد ایمنی پنجره دیگری باز می شود که تغییر وضعیت را اعلان می کند. مد ایمنی در بافر تشخیص عیب *CPU* وارد شده و به *OS* اعلام می شود و می توان در چارت *F_Shutdn* در خروجی *SAFE_M* بلوک *F_SHUTDOWN* چک کرد.



۶-۹-۵- دانلود برنامه F (Downloading the safety program)

پس از کامپایل برنامه F، می توان برنامه را به CPU دانلود کرد. بسته به این که مد ایمنی فعال یا غیرفعال شده باشد، تغییرات برنامه را به مطابق جدول زیر می توان به CPU دانلود کرد.

Download	AS in STOP Single/H-system	AS in RUN Safety mode active	AS in RUN Safety mode inactive
... of the entire program	Possible/Possible	Not possible	Not possible
... of changes in the standard program	Possible/Not possible	Possible	Possible
... of changes in the safety program	Possible/Not possible	Not possible	Possible

قبل از این که بتوان کل برنامه یا تغییرات آن را به CPU دانلود کرد، بایستی شرایط زیر فراهم شود:

- بایستی پیکربندی سخت افزار از پنجره HW config به CPU دانلود شده باشد.
- برنامه کاربر بدون خطا کامپایل شده باشد.
- کاربر حق دسترسی لازم را داشته باشد. یعنی پسورد CPU و برنامه F را داشته باشد.
- یک اتصال آنلاین بین CPU و ایستگاه EWS وجود داشته باشد.

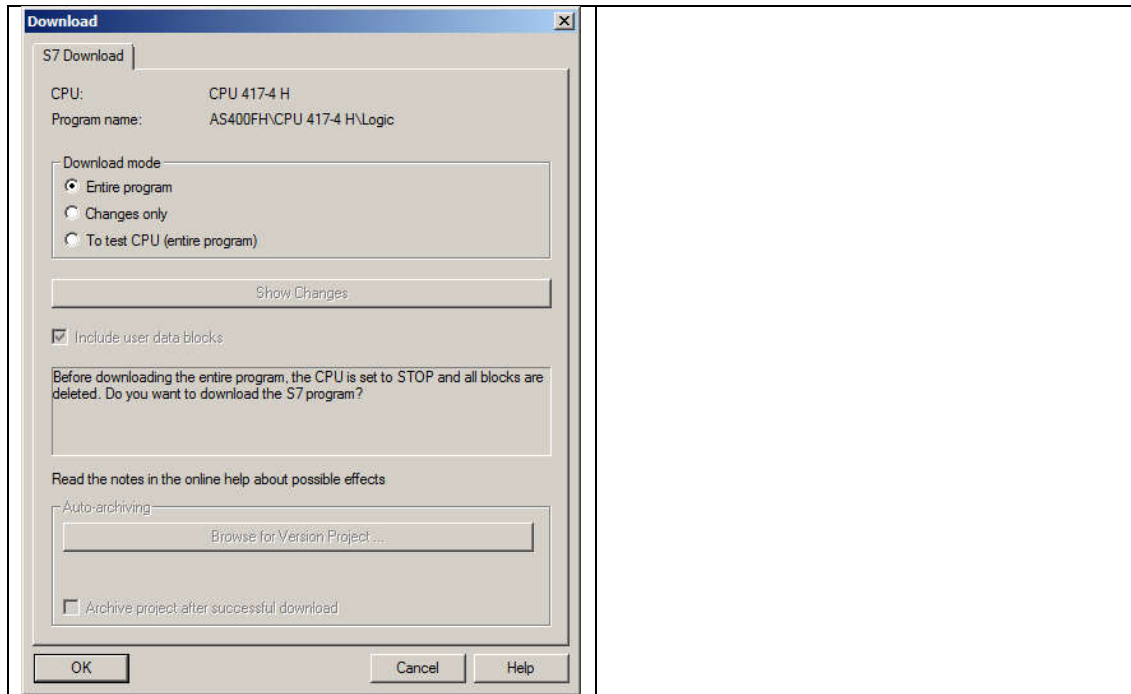
قوانین دانلود

۱. قبل از دانلود برنامه ایمنی، آزمون consistency را انجام دهید. هر دو Signature، در بخش اطلاعات برنامه و در فولدر ایمنی باید یکسان باشد.
۲. F-runtime group های خالی ممکن است پس از حذف توابع ایمنی ایجاد شوند. قبل از کامپایل آنها را حذف کنید. برای انجام این کار در PCS 7 V7.0 و بالاتر، فرمان زیر را در ویرایشگر CFC اجرا کنید:

Edit > Delete Empty Runtime Groups

۳. برنامه ایمنی را تنها می توان از ویرایشگر CFC یا از SIMATIC Manager از پوشه Chart دانلود کرد.

۴. همین که برنامه F را دانلود کردید. نیاز است که Collective signature بررسی شود.

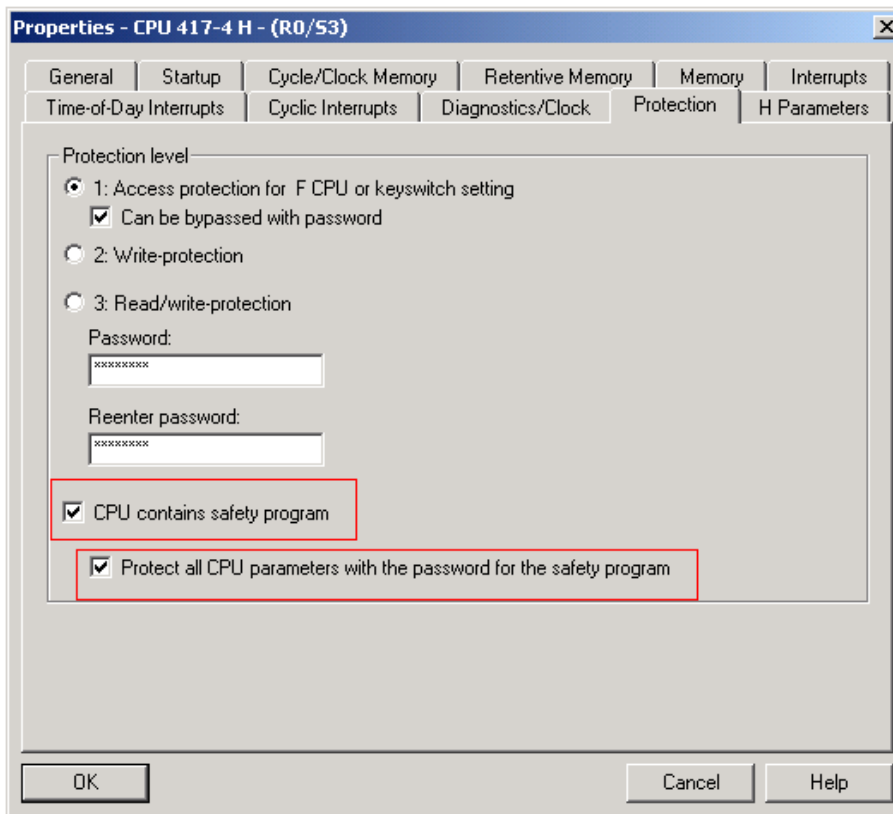


۶-۹-۶- رمز حفاظتی برای توابع مرتبط با ایمنی و تنظیمات CPU

اگر در یک پروژه هر دو بخش استاندارد و F برنامه کاربر در یک CPU اجرا می شود، تغییر در بخش استاندارد را می توان بدون نیاز به وارد کردن رمز عبور F -password کامپایل کرد. زیرا فرض می کند که هیچ تغییری در برنامه ایمنی صورت نگرفته است.

برای فعال شدن توابع ایمنی موجود در سیستم عامل H -CPU در زمان دانلود برنامه به CPU، پنجره ای باز می شود که نیاز است رمز عبور تعریف شده در پنجره پراپرتی برای دسترسی به CPU وارد شود. همانطور که گفته شد این رمز برنامه F را در مقابل دسترسی های غیرمجاز محافظت می کند.

برای تعریف این رمز قبل از برنامه نویسی بلاک های F و در زمان پیکربندی سیستم F ، مطابق شکل زیر تعریف کرده و همچنین گزینه *CPU contains safety program* را نیز برای فعال کردن قابلیت اجرای بلاک های F توسط CPU تیک می زنیم. همچنین گزینه حفاظت از پارامترهای CPU با رمز عبور را برای برنامه ایمنی فعال می کنیم.



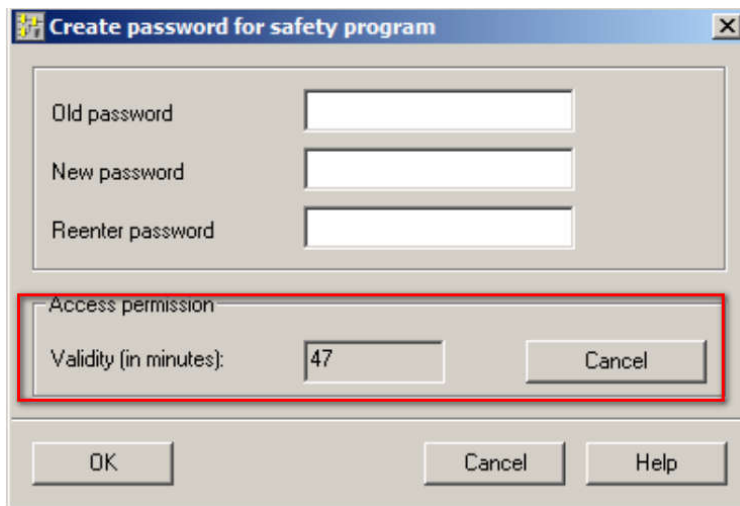
وقتی برای اولین بار برنامه F کامپایل می شود، پنجره تعریف رمز عبور برای برنامه F نمایش داده می شود. این رمز عبور در هنگام باز کردن چارت ها و بلاک های F درخواست می شود. به عبارت دیگر در CPU های S7 که قادر به اجرای برنامه fail-safe هستند، برنامه F و پارامترهای ماژول های F در برابر تغییرات غیر مجاز، با یک رمز عبور محافظت می شود. این رمز موقع دسترسی به بخش های fail-safe سیستم، به طور اتوماتیک پرسیده می شود.

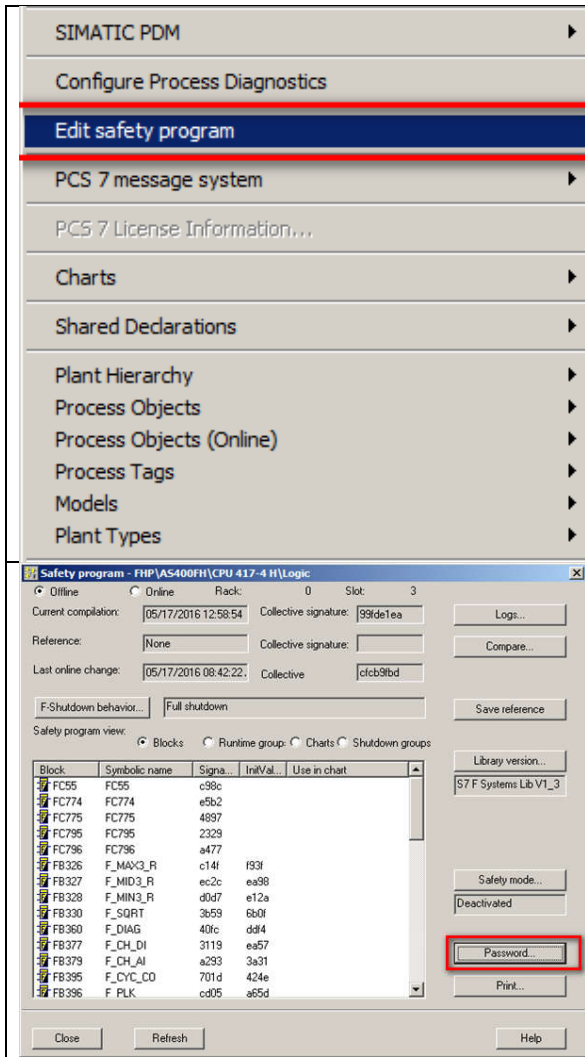
برای مثال :


- هنگام باز کردن یک چارت *fail-safe CFC*؛
- هنگام کامپایل تغییرات در برنامه ایمنی؛
- و یا در هنگام باز کردن خواص یک ماژول F؛

<p>درخواست ورود رمز برای اولین بار در زمان کامپایل، در صورتی که در پنجره <i>Edit safety program</i> تعریف نشده باشد.</p>	<p>درخواست رمز موقع دسترسی به چارت ها یا بخش های مختلف برنامه <i>fail-safe</i></p>

اگر پس از نمایش پنجره درخواست رمز، کلمه عبور وارد شود، به مدت یک ساعت همچنان در سیستم معتبر بوده و در دفعات بعد پرسیده نمی شود. در صورت اتمام کار توصیه می شود که پرپود زمانی اعتبار رمز عبور ریست بشود. برای این کار، مطابق شکل زیر در پنجره *SIMATIC Manager* پس از انتخاب *CPU* که حاوی برنامه ایمنی می باشد، گزینه *Options > Edit Safety Program* را اجرا می کنیم. از پنجره باز شده با کلیک روی دکمه *Password* پنجره *Create Password for Safety Program* باز کنید. سپس در این پنجره با کلیک روی دکمه *Cancel* بازه اعتبار رمز به صفر تنظیم می شود. در این پنجره همچنین می توانید رمز عبور را تغییر دهید. در شکل زیر مقدار زمان باقی مانده برای اعتبار پسورد، ۴۷ دقیق می باشد.



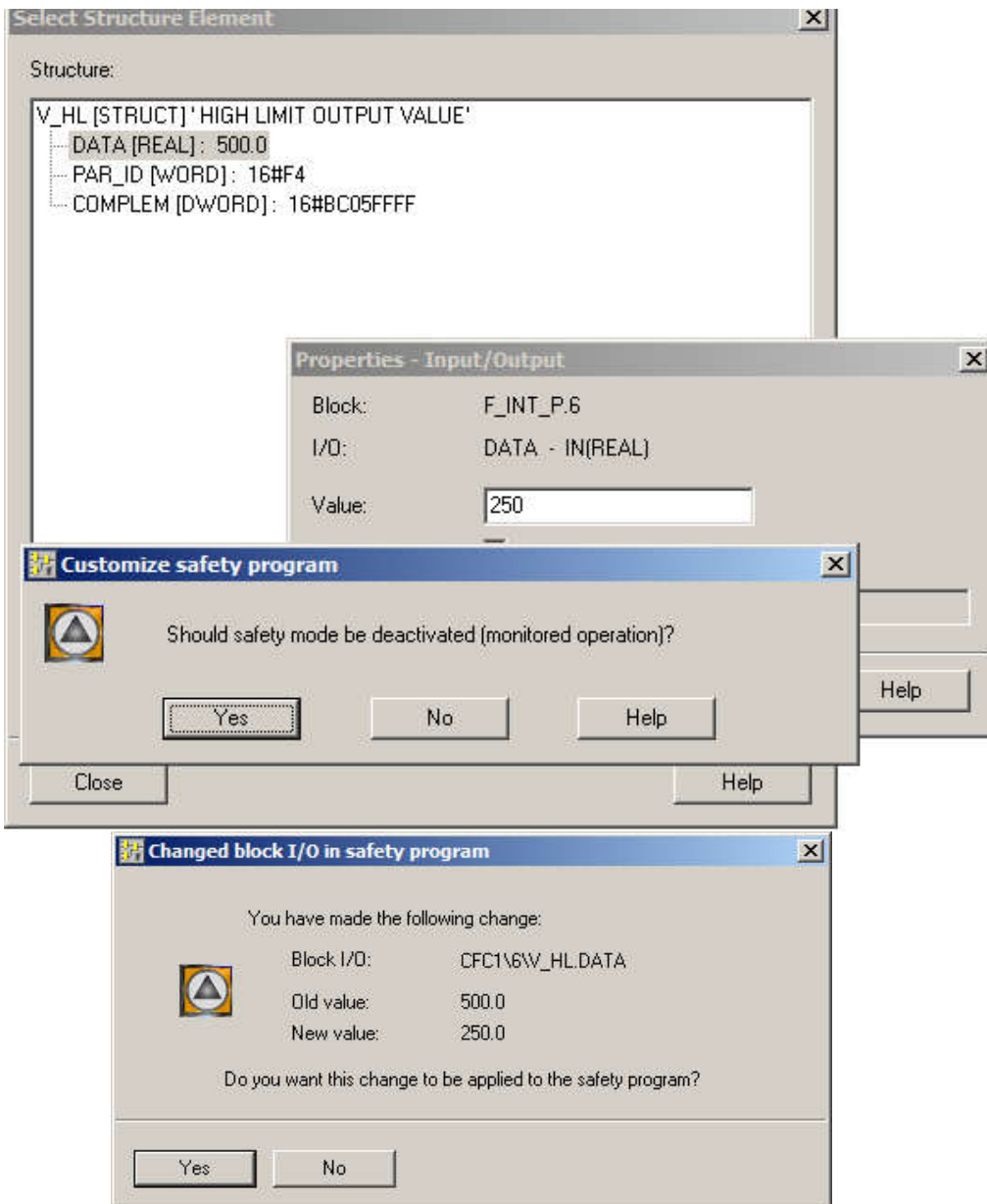


در صورتی که بخواهیم این رمز را تغییر دهیم، در پنجره *Simatic Manager* از منوی *Option* گزینه *Edit safety program* را انتخاب می کنیم. و یا از *Tool bar* آیکون  را کلیک می کنیم

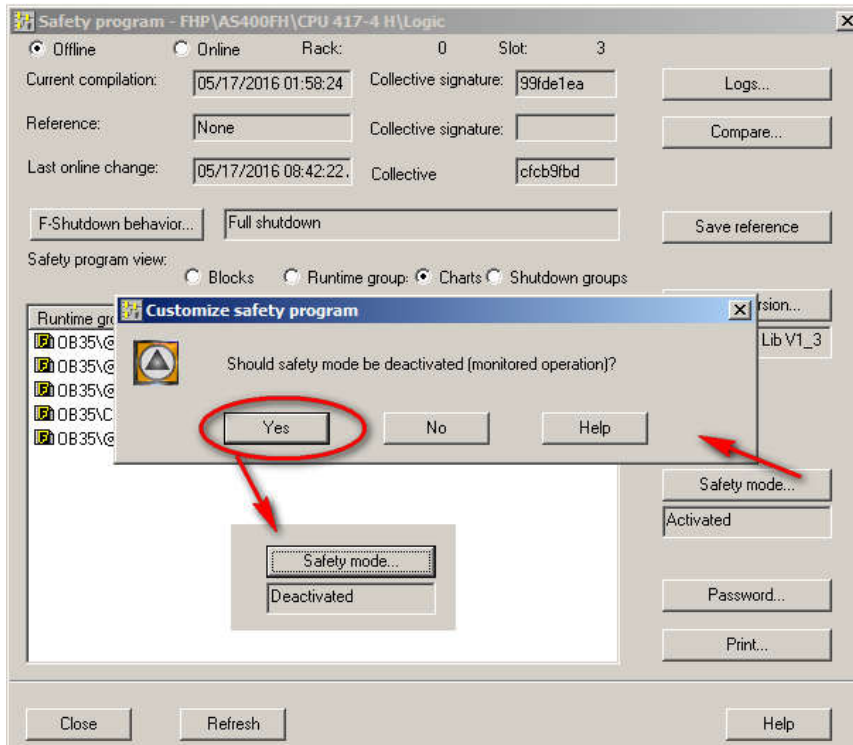
در پنجره باز شده از طریق دکمه *Password* پنجره تغییر رمز را باز کرده و رمز جدید را تعریف می کنیم.



در این حالت با هر بار ورود مقدار به یک پارامتر و اعمال آن به *CPU* در حالت *Runtime* در زمان دیباگ برنامه، پنجره ای باز می شود که اعلام می کند که در زمان تغییر بایستی *safety mode* بایستی غیرفعال شود.



در صورتی که بخواهیم در زمان تست برنامه، مدام پنجره پرسش غیرفعال کردن مد ایمنی نمایش داده نشود، از پنجره *Edit safety program* مد ایمنی را مطابق شکل زیر غیر فعال می کنیم.



موقع دانلود برنامه به CPU رمز تعریف شده در پنجره پراپرتی CPU پرسیده می شود و موقع باز کردن چارت F-CFC پنجره ورود رمز تعریف شده برای برنامه ایمنی پرسیده می شود.



فصل ۷ - پیکربندی ماژول های IO ریدادانت در PCS7

نکات

۱. کارت های Fail Safe امکان مقایسه یک سیگنال را با دو کانال دارند. مثلا اگر DI مد نظر باشد سیگنال آن به دو ترمینال از کارت متصل شده و مقایسه می شود.

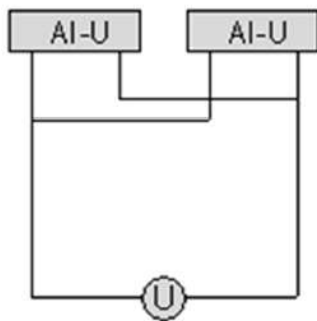
۲. نحوه *Wiring* کارت های *Fail Safe* نسبت به کارت های استاندارد متفاوت است. از نظر ظاهر نیز مقداری عریض تر از کارت های استاندارد هستند.
۳. ۱. کارت های *Fail Safe* به رنگ زرد می باشند.

۷-۱-۱- ورودی آنالوگ

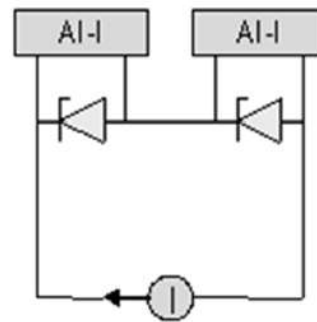
اتصال انکودر (Encoder connectio)

ماژول رودی آنالوگ *SM331; AI 8x16 bit* می تواند در یک پیکربندی ریداندانت با یک یا دو انکودر (*2oo2 structure*) استفاده شود.

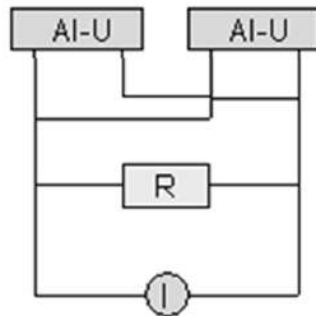
Voltage measurement



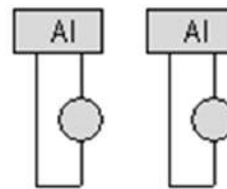
Direct current measurement



Indirect current measurement



With two sensors



به منظور رسیدن به حداکثر در دسترس پذیری توصیه می شود که از انکودرهای ریداندانت استفاده شود. نکته: هنگام استفاده از ماژول های آنالوگ *HART* ریداندانت، تنها یک ترنسmitter اندازه گیری (*Encoder*) می تواند متصل شود.

اتصال سنسور/عملگر در مد ریداندانت (*Connecting sensors/actuators in redundant mode*)

بکار گیری ماژول های آنالوگ ورودی/خروجی هارت در مد ریداندانت تنها در ساختار *Remote I/o* امکان پذیر است. ماژول های هارت دارای پارامترهای «*Primary Master*» و «*Secondary Master*» برای این

منظور هستند. این نوع تخصیص پارامتر برای فعال کردن ارتباطات هارت همزمان به یک وسیله فیلد با استفاده از طریق هر دو ماژول می باشد. ماژول با آدرس پایین همیشه *Primary Maste* می باشد.

۱. در مد ریداندانت، مسترهای دیگر مانند دستگاه هارت دستی، نمی تواند وصل شود.

۲. مد *HART-Fast-Mode* در مد ریداندانت امکان پذیر نیست.

۳. کالیبراسیون کاربر در مد ریداندانت امکان پذیر نیست.

۴. در مد ریداندانت، گزینه *substitute value behavior of the current outputs* به طور خودکار به

جریان و ولتاژ صفر (*OCV*) تنظیم می شود. به هر حال، اگر *CPU* در مد توقف باشد، جریانی حدود

۱۱۵ میکروآمپر در خروجی هر کانال خواهد بود.

۵. زمانی که ماژول های آنالوگ هارت در مد ریداندانت عمل می کنند، یک ترنسدیوسر دو سیمه باید به

صورت یک ترنسدیوسر چهار سیمه متصل شده و همچنین به صورت یک ترنسدیوسر چهار سیمه

در *HW Config* پیکربندی شود. ترمینال های ۱۰ و ۱۱ در *Front panel connector* نباید متصل

شوند.

۶. در مد ریداندانت، افت ولتاژ در هر دو ماژول باید رعایت شود. برای اطمینان از تامین ولتاژ کافی به

ترنسدیوسر، بایستی افت ولتاژ در هر دو ماژول و افت ولتاژ در سیم کشی و خود ترنسدیوسر باید

رعایت شود (اتصال سری). با یک جریان سنسور ۲۲ میلی آمپر، افت ولتاژ حدود $3,3 V$ در هر

ماژول رخ می دهد. اگر شما از مدار با دیود زنر که در شکل زیر نشان داده شده است، استفاده می کنید.

اگر دارید ماژول را جایگزین می کنید، توجه داشته باشید که افت ولتاژ در ماژول برداشته شده از

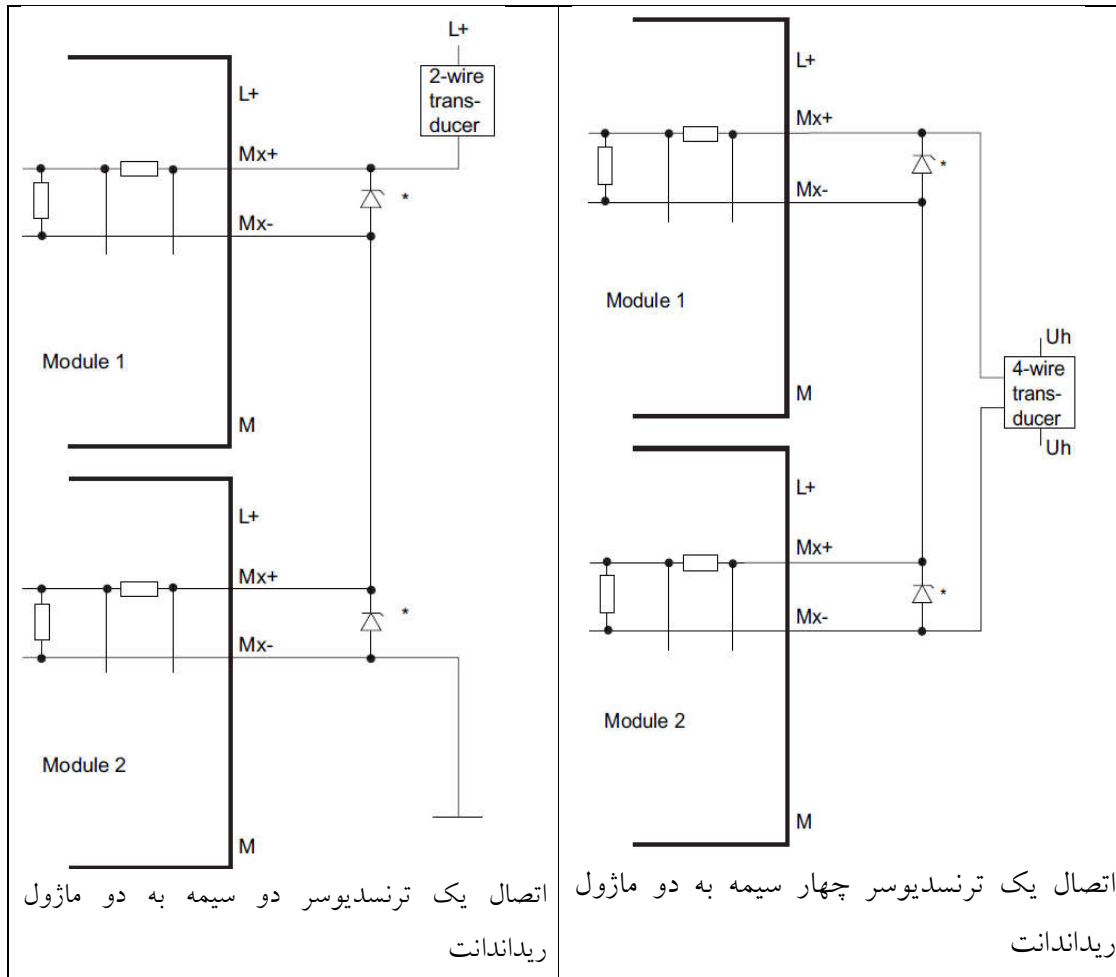
اسلات (*removed module*) برابر ولتاژ دیود زنر ($5.1V$) و افت ولتاژ در ماژول قرار داده شده در

اسلات $3.3V$ است.

۷. وقفه های سخت افزاری توسط بلوک های کتابخانه "*RedLib*" پشتیبانی نمی شود. اگر مایل به استفاده

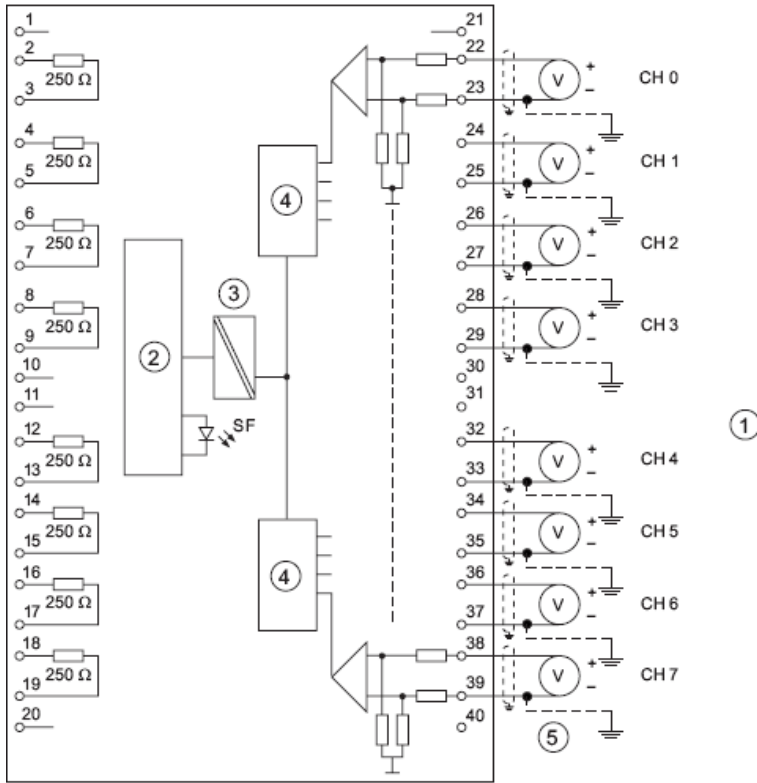
از وقفه های سخت افزاری باشید، بایستی ارزیابی در سطح کاربر را پیاده سازی کنید.



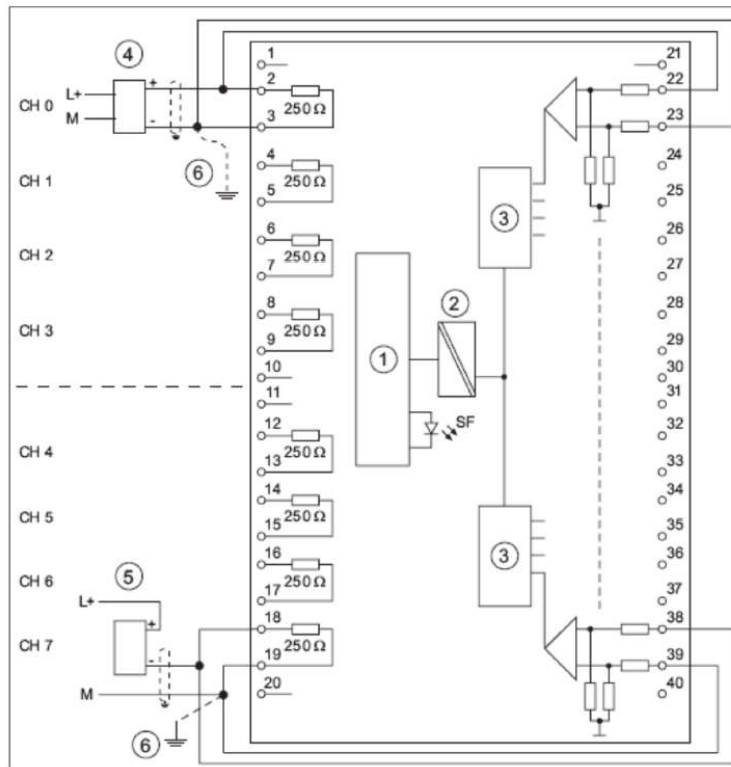


- Zener diode 5.1 V (e.g., BZX85C5V1)

دیاگرام مداری اتصالات ماژول SM331 (Connection and principle circuit diagram)

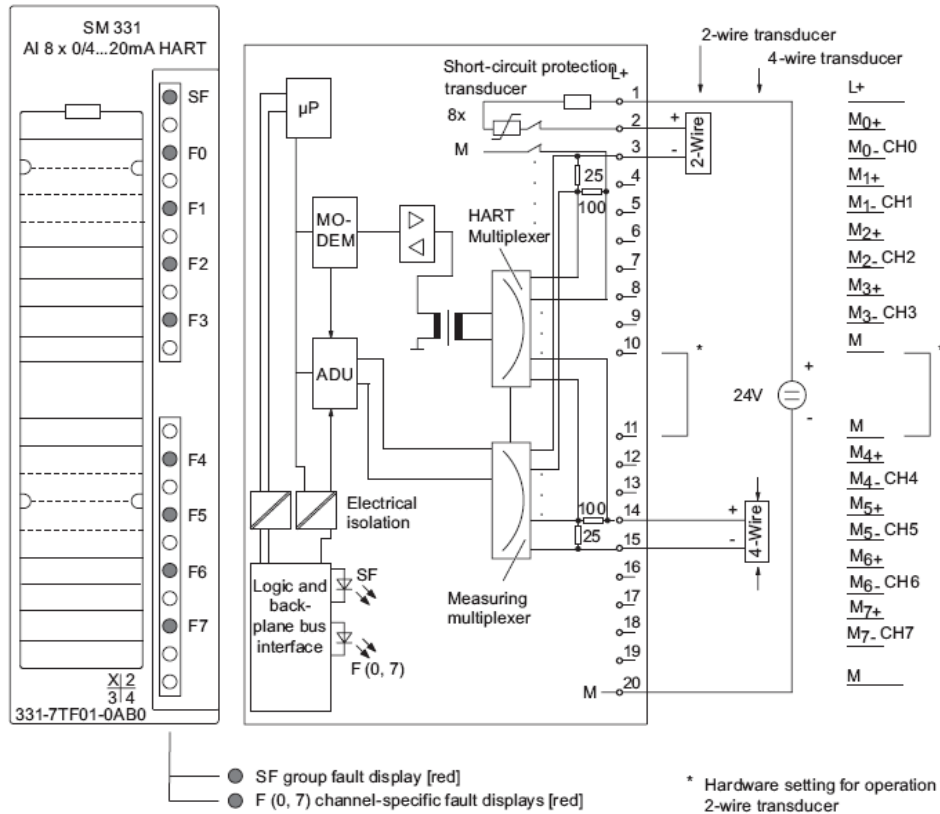


دیاگرام مداری اتصالات ماژول (6ES7 331-7NF00-0AB0) برای اتصال ولتاژی (SM331 AI 8x16Bit)



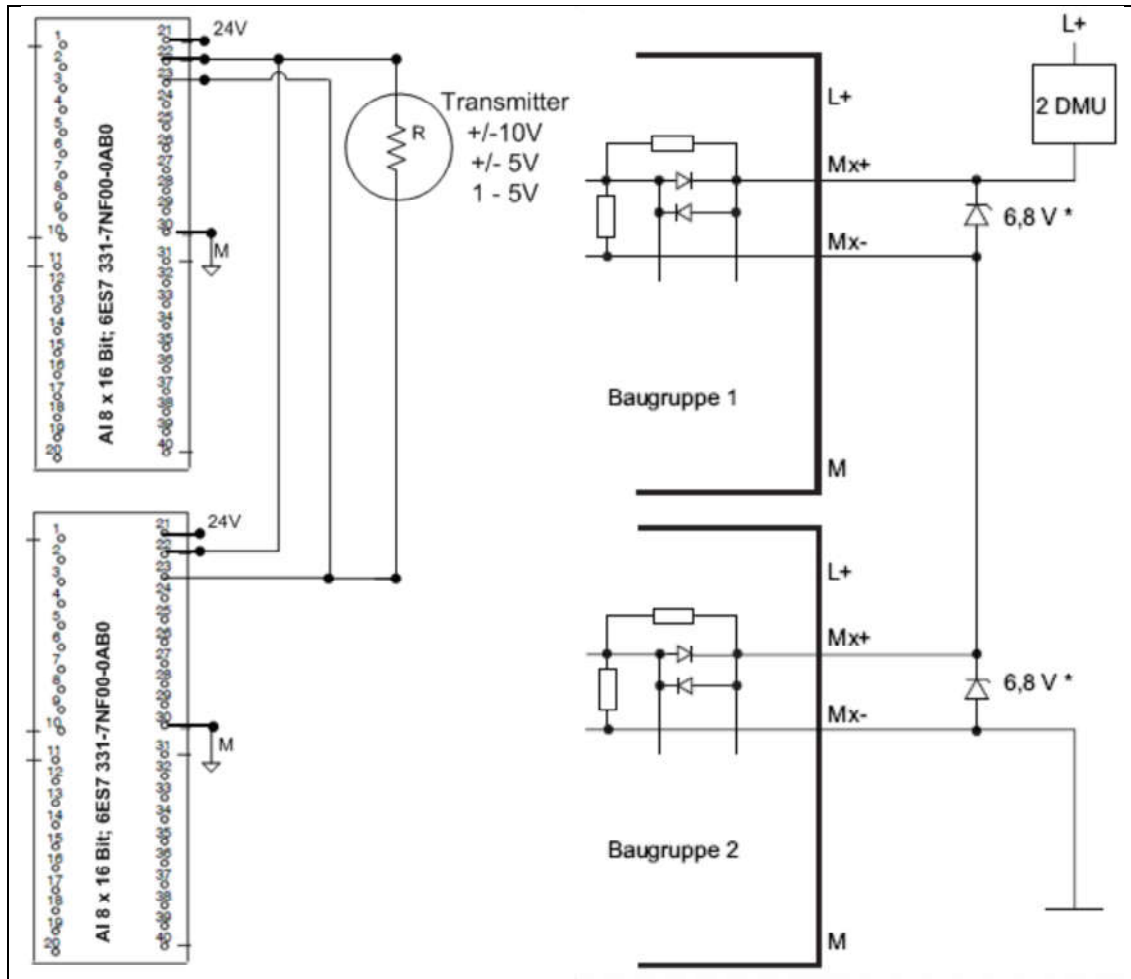
Number	Description
1	Backplane bus connection
2	Electrical isolation
3	Analog-digital converter (ADU)
4	CH 0 for 4-wire measuring transformer
5	CH 7 for 2-wire measuring transformer (with external supply)
6	Equipotential bonding

دیگرام مداری اتصالات ماژول (SM331 AI 8x16Bit (6ES7 331-7NF00-0AB0) برای اتصال جریان



دیگرام مداری اتصالات ماژول هارت (SM331 AI 8x0/4...20 mA (6ES7 331-7TF01-0AB0)

نکته: در حالت ریدادانت نباید دو پایه ۱۰ و ۱۱ به هم متصل شوند.

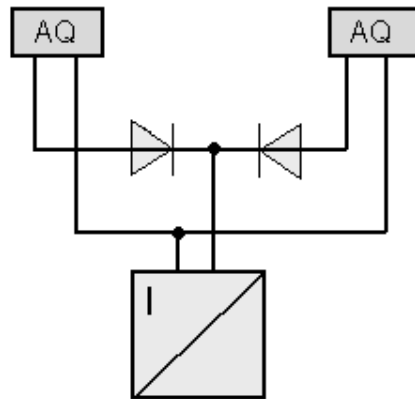


The wired Z-diodes are necessary if after pulling a module the system shall continue to run.

۷-۱-۲- خروجی آنالوگ

کنترل المان کنترل نهایی (Controlling the final control element)

برای کنترل تحمل پذیر خطا (*fault-tolerant*) در یک عنصر کنترل نهایی، دو خروجی به صورت موازی (ساختار 2002) توسط دو ماژول *SM 332*؛ *AO 8x12bit* و از طریق دو دیود روشن (*switched*) می شود. توجه شود که تنها ماژول های آنالوگ خروجی با خروجی جریان می تواند به صورت ریداندانت عمل کند (۰ تا ۲۰ میلی آمپر یا ۴ تا ۲۰ میلی آمپر).



دو خروجی آنالوگ، نیمی از مقدار خروجی را تولید و در پایه خروجی شان قرار می دهند. اگر یکی از ماژول ها با شکست مواجه (*fails*) شود، باز هم خروجی غیر فعال مقدار کامل را تحویل می دهد.
Both outputs output half of the value. If one of the modules fails, the still inactive output delivers the complete value.

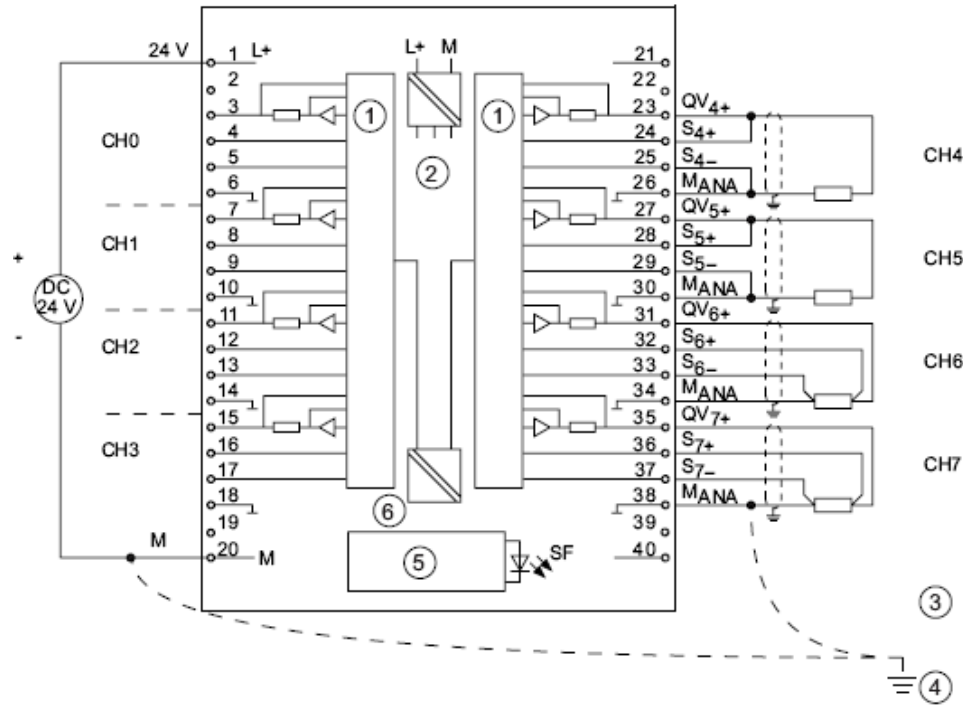
دیاگرام اتصالات ماژول خروجی آنالوگ

Connection and principle circuit diagram of SM 332; AO 8 x 12 bit (voltage output)

شکل زیر دیاگرام مدار یک ماژول خروجی آنالوگ SM332 AO 8x16Bit (6ES7 332-5HF00-0AB0) را

برای خروجی ولتاژ در دو حالت نشان می دهد.

- اتصال ۲ خط بدون مقاومت جبران ساز خط (compensation of line resistors)
- اتصال ۴ خط با مقاومت جبران ساز خط



Number	Description
1	DAU
2	Internal supply
3	Equipotential bonding
4	Functional earth
5	Backplane connection
6	Electrical isolation

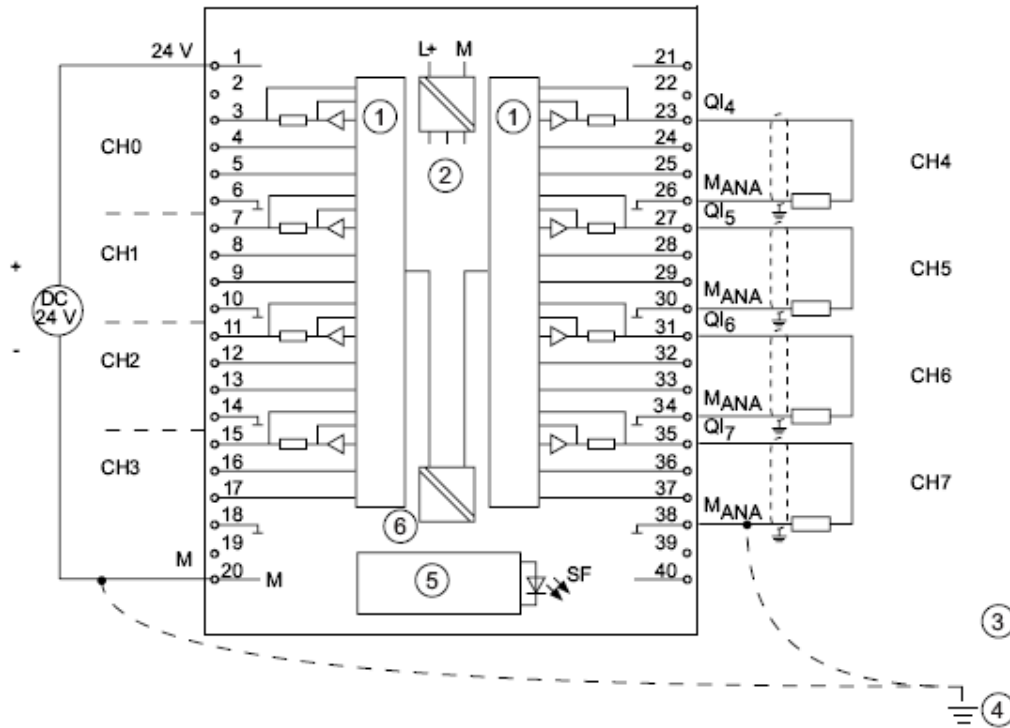
شکل ۷-۱- دیاگرام اتصالات ماژول خروجی آنالوگ 6ES7 332-5HF00-0AB0

نکته : ارت تمیز و ارت کثیف (Protective & Clean earth)

۱. زمین حفاظتی که زمین کثیف (*dirty earth*) نیز نامیده می شود، برای محافظت از پرسنل و تجهیزات استفاده می شود. زمین حفاظتی برای حفاظت از پرسنل در برابر شوک الکتریکی در زمان وقوع یک فالت زمین در نظر گرفته شده است. همواره از لوازم الکتریکی از طریق هادی زمین یک اتصال به ترمینال زمین اصلی در اتاق تامین برق وجود دارد. به طوری که جریان ناشی از فالت از طریق این هادی به جرم زمین جاری خواهد شد.

۲. در این جا زمین *Functional* همان زمین *Clean* است. که به عنوان یک زمین مرجع برای سیگنال های دیجیتال و آنالوگ استفاده می شود. این زمین همچنین برای کاهش اعوجاج در سیگنال لازم است. زمین کاربردی (*Functional earth*) برای مواردی چون کاهش نویز فرکانس رادیویی، فیلتر کامپیوتر به منظور

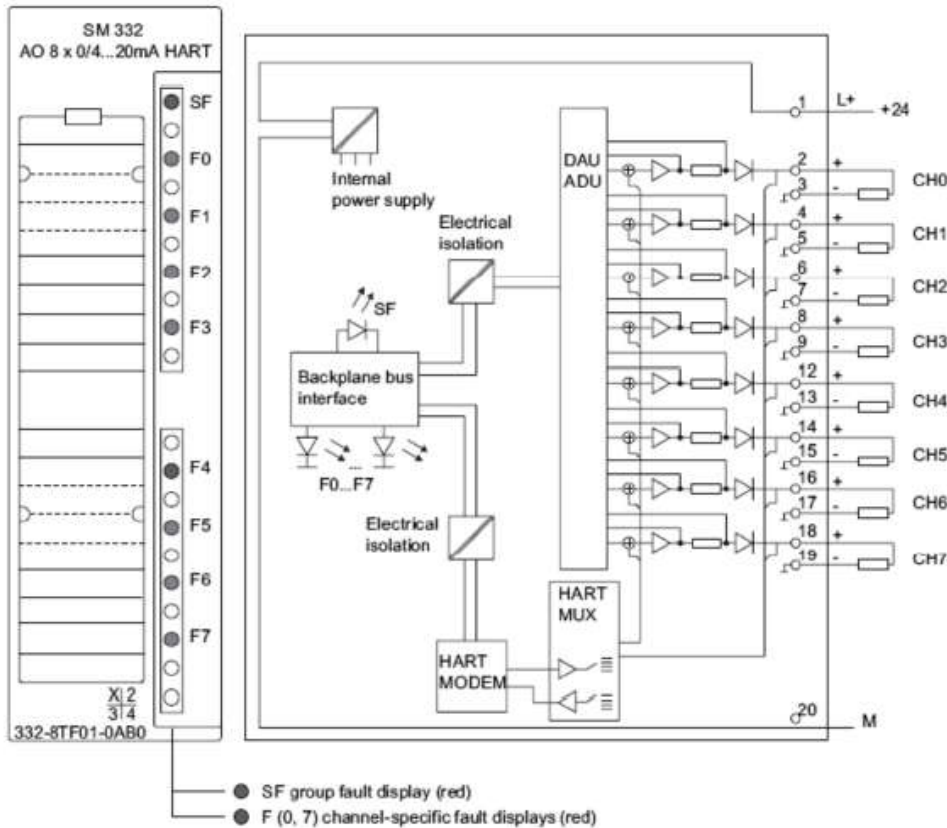
افزایش وضوح سیگنال و یا دیگر تجهیزات که همواره جریان نشتی $10mA$ دارند، در نظر گرفته شده است. زمین کاربردی باید از زمین حفاظت به جز در نقطه زمین اتصال ترمینال اصلی جدا شوند.
شکل زیر دیاگرام مدار یک ماژول خروجی آنالوگ $SM332 AO 8x16Bit$ را برای خروجی جریان نشان می دهد.



شکل ۷-۲- دیاگرام سیم بندی ماژول آنالوگ خروجی جریان (6ES7 332-5HF00-0AB0)

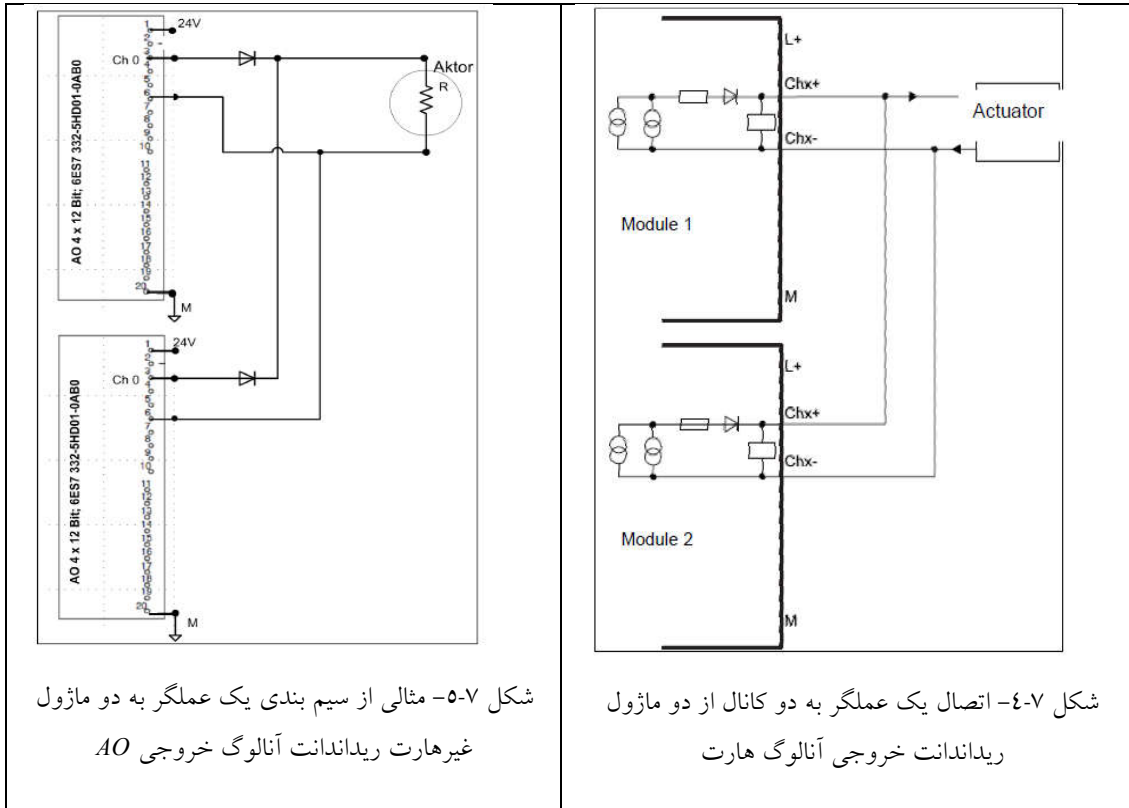
دیاگرام اتصالات ماژول آنالوگ خروجی هارت $0/4...20mA$

Connection and principle circuit diagram SM 332; AO 8 x 16Bit, 0/4...20mA HART



شکل ۳-۷- دیاگرام اتصالات ماژول آنالوگ خروجی هارت (6ES7 332-8TF01-0AB0)

شکل ۵-۷ مثالی از سیم بندی یک عملگر به دو ماژول غیرهارت ریداندانت آنالوگ خروجی (AO) نوع جریان یا ولتاژ و شکل ۴-۷ اتصال یک عملگر به دو کانال از دو ماژول ریداندانت آنالوگ خروجی هارت را نشان می دهد.



شکل ۵-۷- مثالی از سیم بندی یک عملگر به دو ماژول
غیرهارت ریداندانت آنالوگ خروجی AO

شکل ۴-۷- اتصال یک عملگر به دو کانال از دو ماژول
ریداندانت خروجی آنالوگ هارت

دیودهای مناسب برای این منظور به عنوان مثال سری $IN4003 \dots IN4007$ می باشد. یا هر دیود با مشخصه $I_F \geq 1 A$ و $U_r \geq 200 V$ را می توان استفاده کرد.

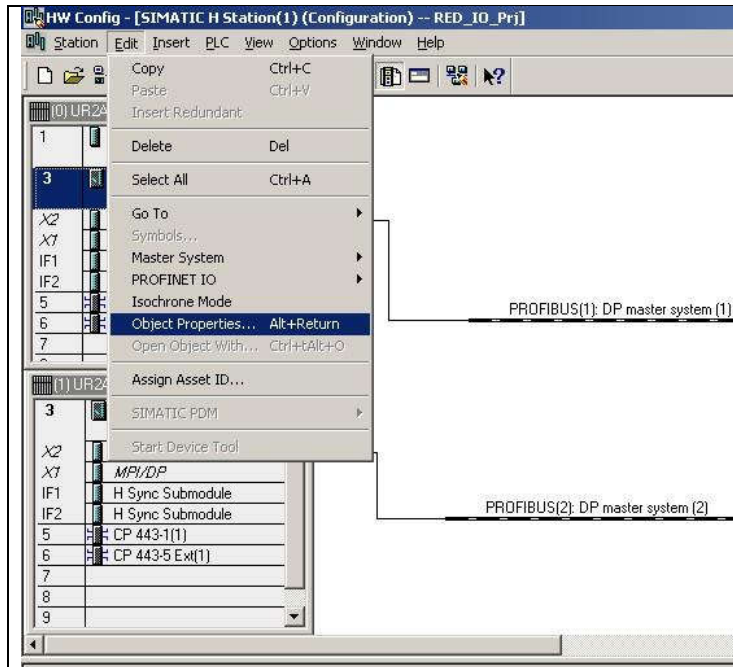
۷-۲- پیکربندی IO ریداندانت در Step7

این بخش به توصیف پیکربندی اجزاء مختلف برای پیکربندی ریداندانسی IO در PCS 7 پرداخته است. فرض بر این است که یک پروژه PCS 7 از نوع H Station با استفاده از ویزارد پروژه، ایجاد شده است. مراحل پیکربندی زیر تشریح می شود:

- Setting the CPU (H parameters)
- Configuring the first ET 200M (interface module IM 153-2)
- Configuration of the individual signal module IOs
- Configuring the second ET 200M by copying
- Redundancy settings at the signal modules
- Symbolic name assignment
- CFC configuration

۷-۲-۱- تنظیم پارامترهای CPU

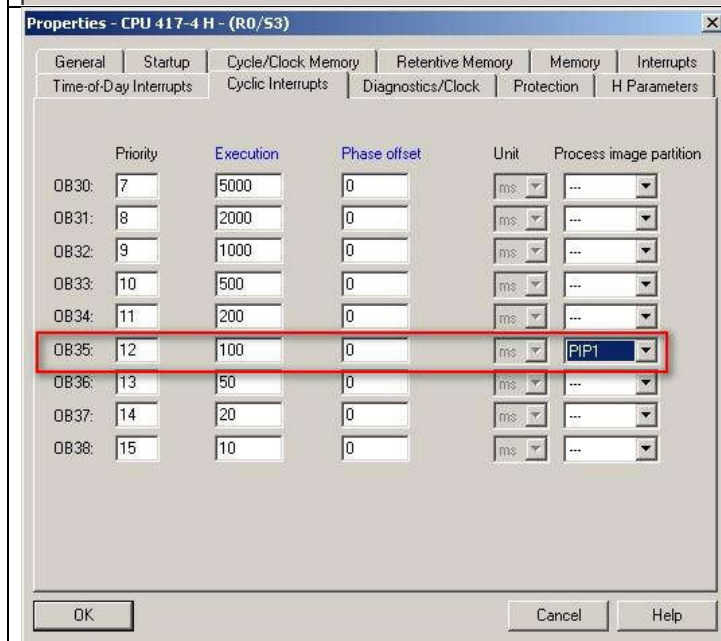
به منظور فعال کردن عملکرد ریداندانت بین AS و I/O توزیع شده، نیاز است که در CPU تنظیمات خاصی انجام شود.



۱- SIMATIC H Station را در HW

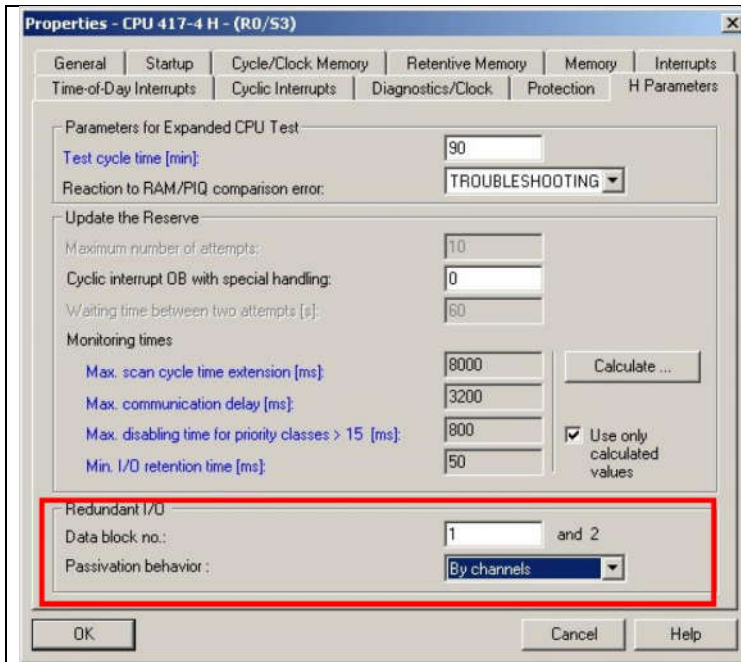
Config باز کنید.

۲- اولین CPU در اسلات ۳، رک (۰) را انتخاب و پنجره پراپرتی آن را باز کنید.



۳- سربرگ *Cyclic Interrupts* را با

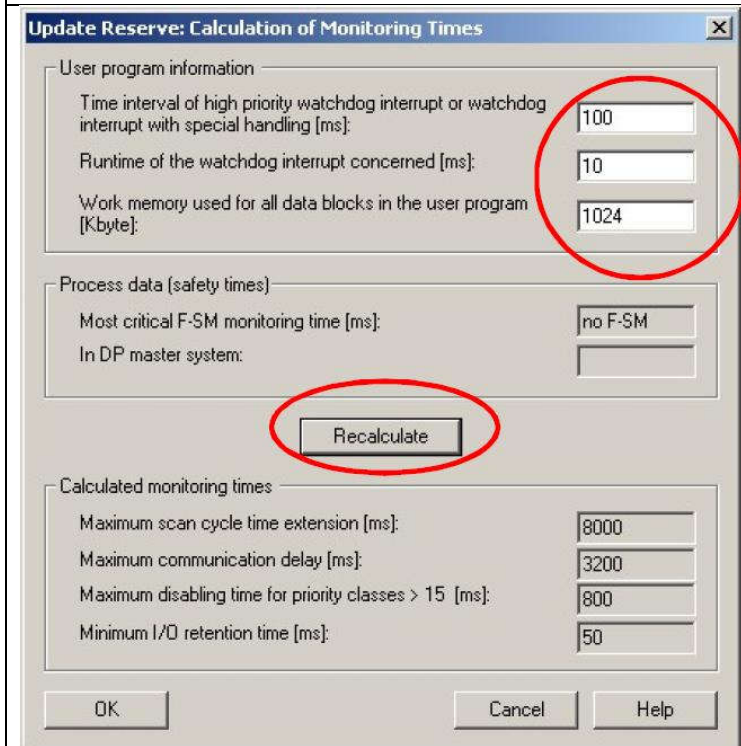
کرده و از آن جا از طریق یک لیست *drop-down* فضای *process image partition* را برای *OB* سیکلیک (*OB35*) که برنامه در آن اجرا می شود را انتخاب کنیدید.



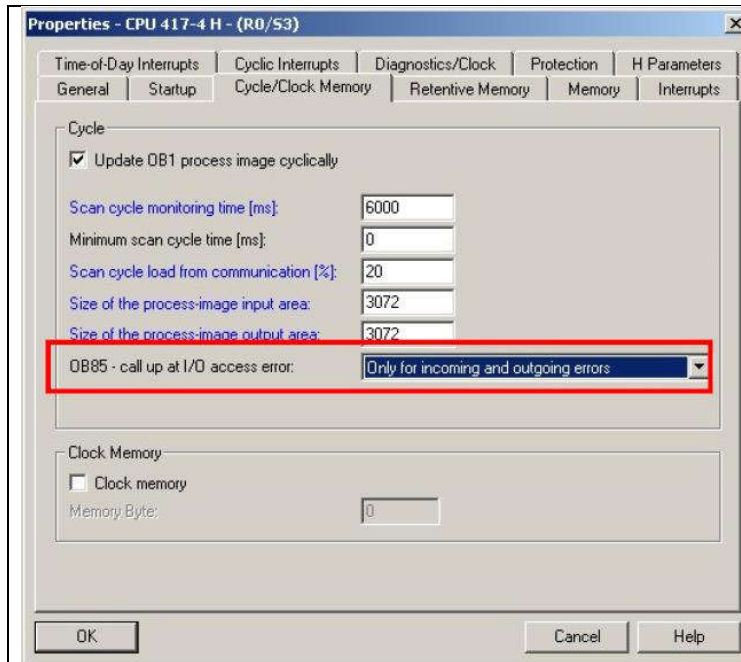
۴- سربرگ *H Parameters* را باز کنید. در این سربرگ دقت کنید که شماره *DB* که در فیلد « *Data block* » وارد می شود. نایستی در برنامه یا در پیکربندی استفاده شده باشد. در قسمت *Passivation behavior* یکی از دو گزینه زیر را برای زمانی که خطا رخ می دهد، انتخاب کنید.

- *By channel*
- *By module*

این تنظیمات به تمامی ماژول هایی که به صورت ریدانداست به *CPU* متصل شده اند اعمال می شود. برای محاسبه زمان مانیتورینگ دگمه *Calculate* را از قسمت *Update the Reserve* کلیک کنید.



۵- پس از کلیک روی *Calculate* در پنجره باز شده، مقادیر داده های مربوط به برنامه را در فیلدهای مربوطه وارد کرده و دگمه *Recalculate* را کلیک کنید. پس از بستن این پنجره، از اطلاعات محاسبه شده جدید در فیلدهای مربوطه ظاهر می شود.



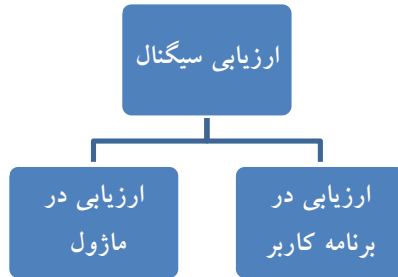
۶- سربرگ Cycle/Clock Memory
باز کرده و در قسمت *OB 85 call for*
گزینه *Only for incoming I/O error*
and outgoing errors را انتخاب
کنید.

۷-۲-۲- پیکربندی ET200M (Configuring the first ET 200M)

فصل ۸ - سیم بندی و معماری ارزیابی ایمنی برای ماژول های آنالوگ ورودی

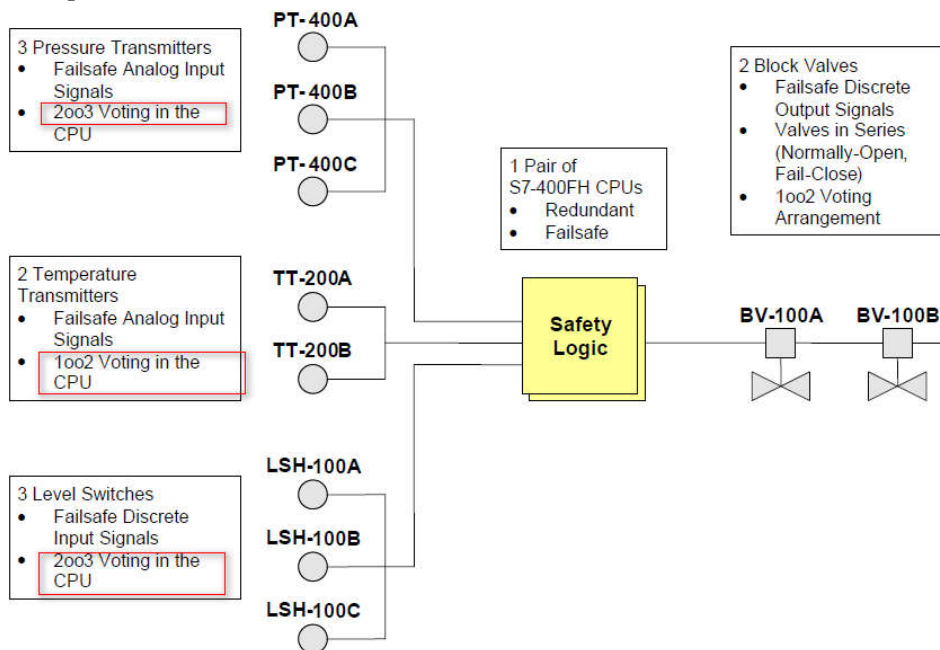
۸-۱-۱ مثال

فرض کنید چندین سیگنال آنالوگ بایستی برای هدف *failsafe* در یک پلنت مانیتور شود. بسته به اهمیت و خطر وقوع نقص یا شکست در سیستم، چندین گزینه، برای سیم بندی و ارزیابی ایمنی سیگنال ها وجود دارد. ارزیابی ممکن است، برای مثال، در ماژول ورودی آنالوگ و یا در برنامه کاربر انجام شود.



شکل ۸-۱ نشان می دهد یک مثال از سیم بندی ارزیابی ایمنی از سیگنال های ورودی آنالوگ را به تصویر کشیده است، که در آن شیرهای سولونوئید (*BV-100A* و *BV-100B*) بسته به مقادیر سیگنال های زیر باید به صورت ایمن بسته شوند.

- *Pressure*
- *filling level*
- *temperature*



شکل ۸-۱- یک مثال از سیم بندی ارزیابی ایمنی از سیگنال های ورودی آنالوگ

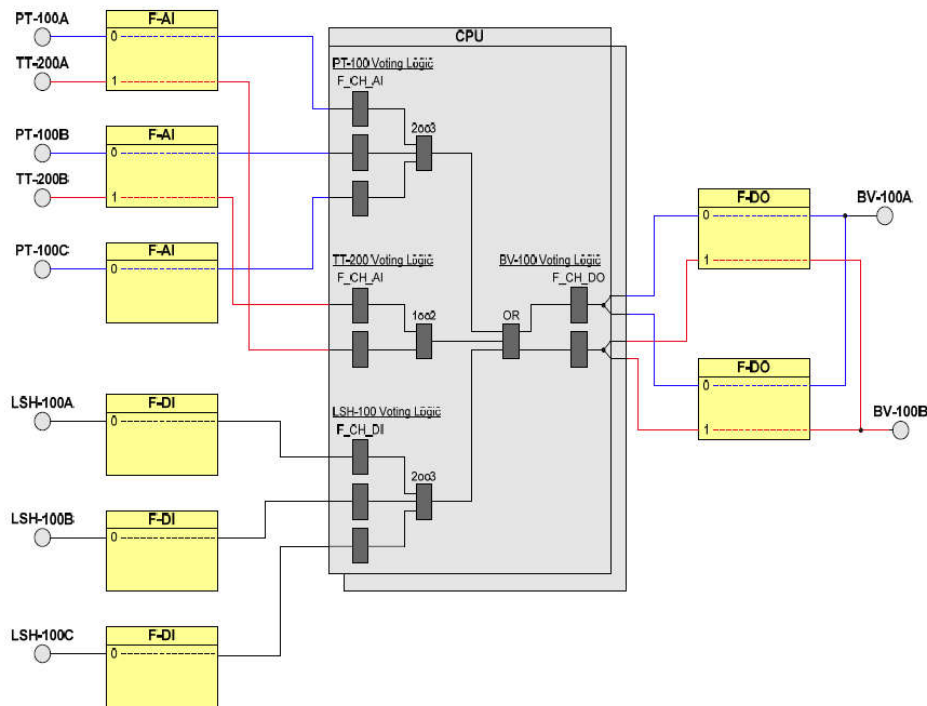
مطابق شکل بالا عمل *voting* در ماژول CPU صورت می گیرد.

نکته: در تمام مثال های این بخش، ماژول ورودی آنالوگ *F* از نوع هارت زیر در نظر گرفته شده است.

- *F-AI 6 x 0/4 ... 20 mA HART- 6ES7 336-4GE00-0AB0*

۸-۱-۱- راه حل

شکل زیر یک تحقق احتمالی از این اجرای پلنت را نشان می دهد که در آن کانکشن و معماری های آزمون مختلف از سیگنال های آنالوگ استفاده شده است.



۸-۱-۲- توصیف معماری های ارزیابی

معماری های ارزیابی که برای این مثال توصیه می شود عبارت است از:

۱. معماری تک سنسور (1001)

کاربردهای نوعی این مورد در جایی است که در آن یک سنسور تکی، سطح یکپارچه ایمنی (*SIL*) مورد نیاز را دارا بوده و در آن در دسترس بودن بالا (*increased availability*) لازم نیست.

۲. ارزیابی دو سنسور (1002) در *F-AI*

کاربردهای نوعی این مورد در جایی است که در آن یک سنسور سطح مورد نیاز ایمنی را دارا نبوده و در دسترس بودن بالا (*increased availability*) لازم نیست.

۳. ارزیابی دو سنسور در برنامه کاربر

کاربردهای نوعی این مورد در جایی است که در آن در آن یک سنسور سطح مورد نیاز ایمنی را دارا نبوده و مشاهده داده از هر دو سنسور (*visibility of the data of both sensors*) مورد نیاز است. این معماری می تواند برای افزایش در دسترس بودن، به صورت 2002 نیز پیگیری شود.

۴. ارزیابی سه سنسور (2003) در برنامه کاربر

کاربردهای نوعی این مورد در جایی است که در آن برای رسیدن به سطح ایمنی مورد نیاز، چند سنسور لازم بوده و افزایش در دسترس بودن مدنظر است.

۸-۲- مشخصات یک ماژول ورودی آنالوگ F

مشخصات یک ماژول ورودی آنالوگ F به شرح زیر می باشد:

- 6 analog inputs with electrical isolation between channels and backplane bus
- Input ranges:
 - 0 to 20 mA
 - 4 to 20 mA
- Short-circuit resistant power supply of 2- to 4-wire measuring transmitter via the module
- External encoder supply possible
- Collective error display (SF)
- Display safety mode (SAFE)
- Display for channel-specific error (Ex)
- Display for HART status (Hx) (If you have switched on measuring type HART for a channel, and the HART communication is running, the green HART status display lights up.)
- Parameterizable diagnostics
- Parameterizable diagnostic alarm only in safety-mode
- HART communication
- Firmware update via HW Config
- Identification data

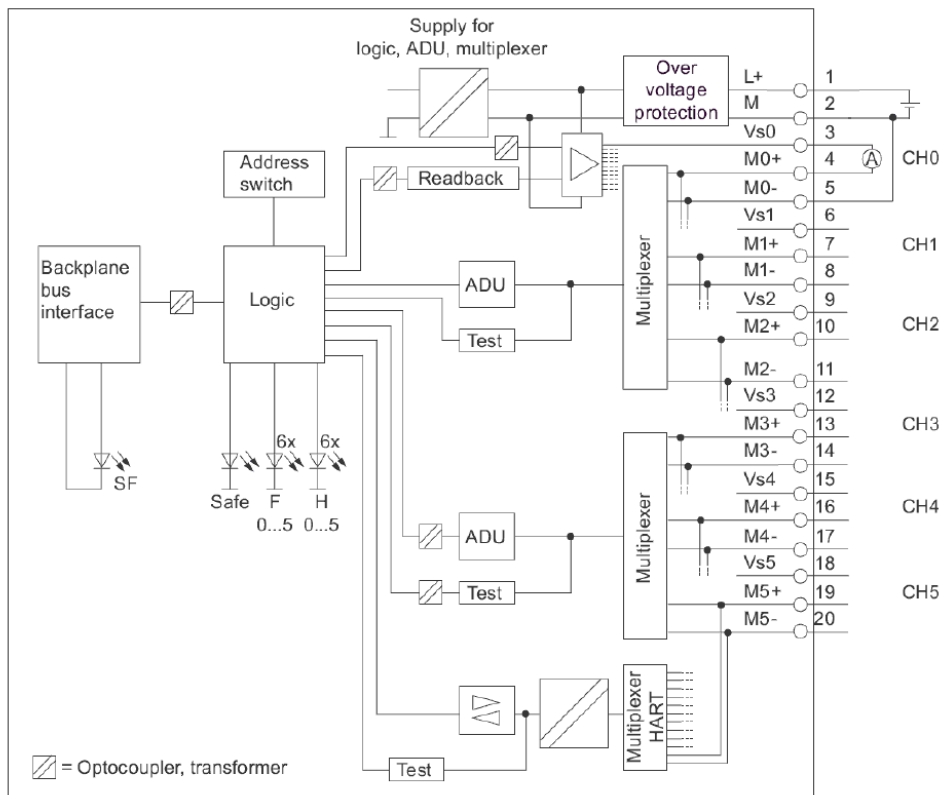
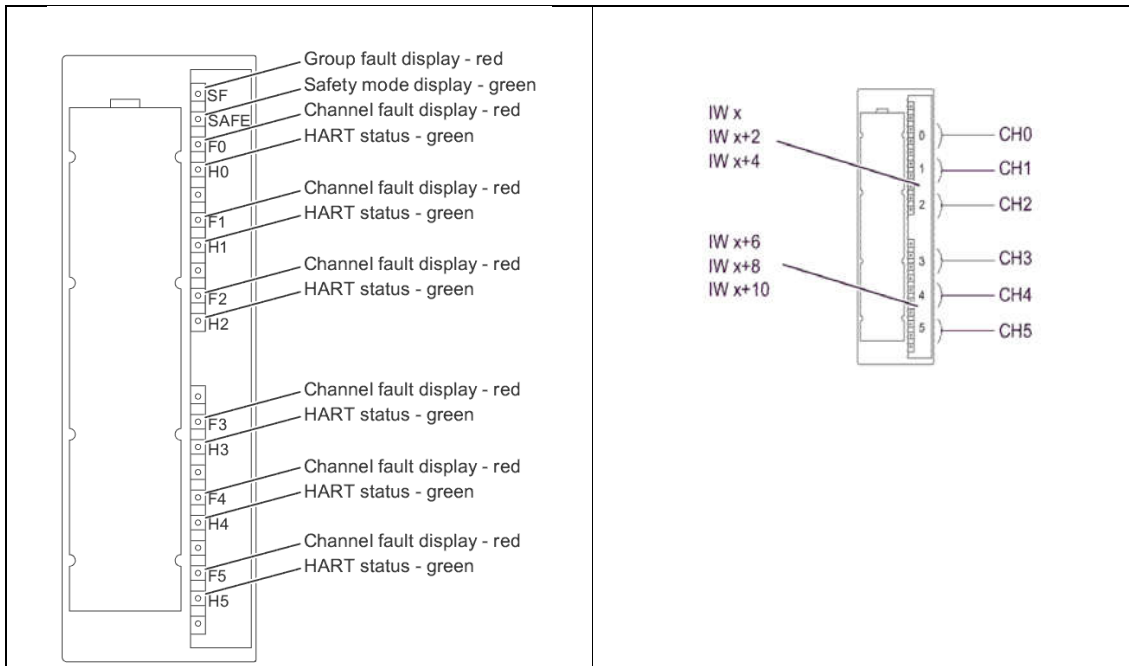
هر یک از شش کانال آنالوگ ورودی را می توان برای اندازه گیری جریان استفاده کرد:

- 0 to 20 mA (without using HART)
- 4 to 20 mA (with/without using HART)

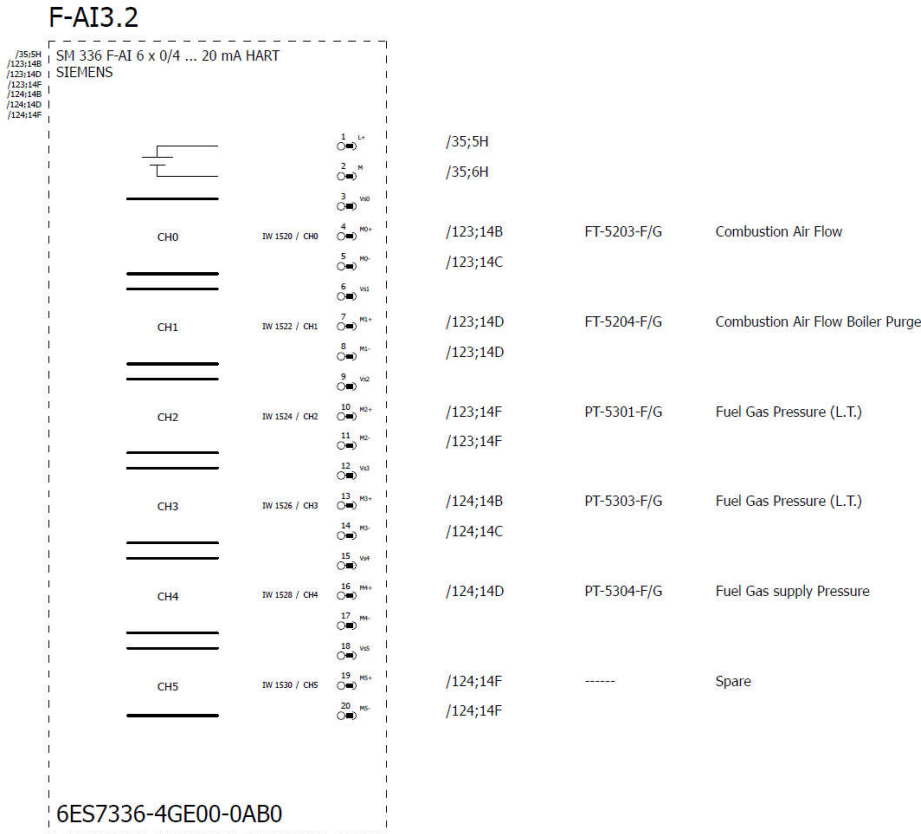
▪ این کارت بیست پین می باشد.

▪ توصیه می شود که همیشه از short-circuit resistant internal encoder supply استفاده شود.

▪



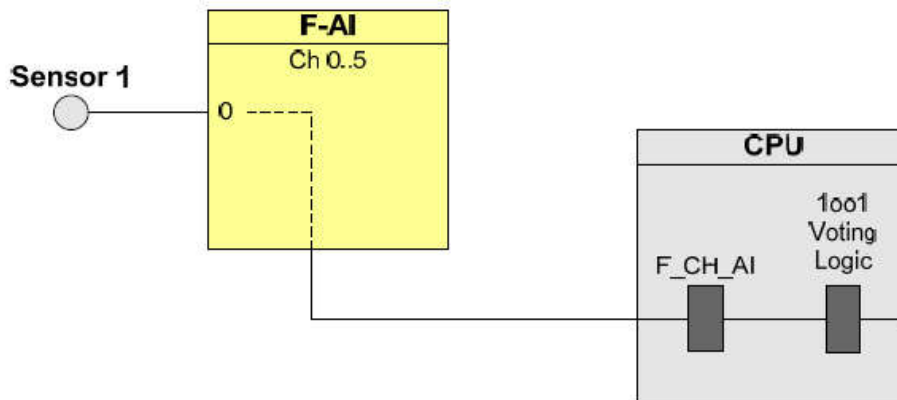
دیگرام مدار اتصالات کارت آنالوگ HART 6 x 0/4...mA F-AI SM 336



شکل ۸-۲- نمونه اتصال سیگنال های آنالوگ هارت به کارت

۸-۳- سیم بندی یک سنسور آنالوگ (1001)

طرح ارزیابی یک سنسور (یا 1001) به کاربردهایی مربوط می شود که در آن به افزودنی سنسور نیاز نمی باشد. ارزیابی 1001 بدان معنی است که تنها یک سنسور وجود دارد. اگر در سنسور شرایط تریپ رخ دهد، لاجیک ایمنی تریپ خواهد کرد. شکل زیر معماری پایه 1001 را نشان می دهد که در آن یک سنسور به یک کانال از ماژول F-AI متصل شده است.



شکل ۸-۳- سیم‌بندی یک سنسور به یک کانال F-AI (1001)

جدول زیر شرایط تریگر شدن تابع واکنش خطا (error reaction function) را نشان می‌دهد.

Component has failed ?		Error reaction function has been triggered?
Sensor 1	F-AI	
No	No	No
X	Yes	Yes
Yes	X	Yes

در این معماری اگر سنسور یا F-AI معیوب شود، تابع واکنش خطا، عملکرد ایمنی (از طریق سیستم failsafe) فراهم می‌کند.

۸-۳-۱- محاسبه PFD (Probability of Failure on Demand)

مقدار PFD احتمال شکست یک تابع failsafe را توصیف می‌کند.

$$PFD(1001) = PFD_{Sensor} + PFD_{F-AI} + PFD_{CPU}$$

برای یک سنسور با معماری 1001 مقدار PFD_{Sensor} با استفاده از فرمول زیر محاسبه می‌شود

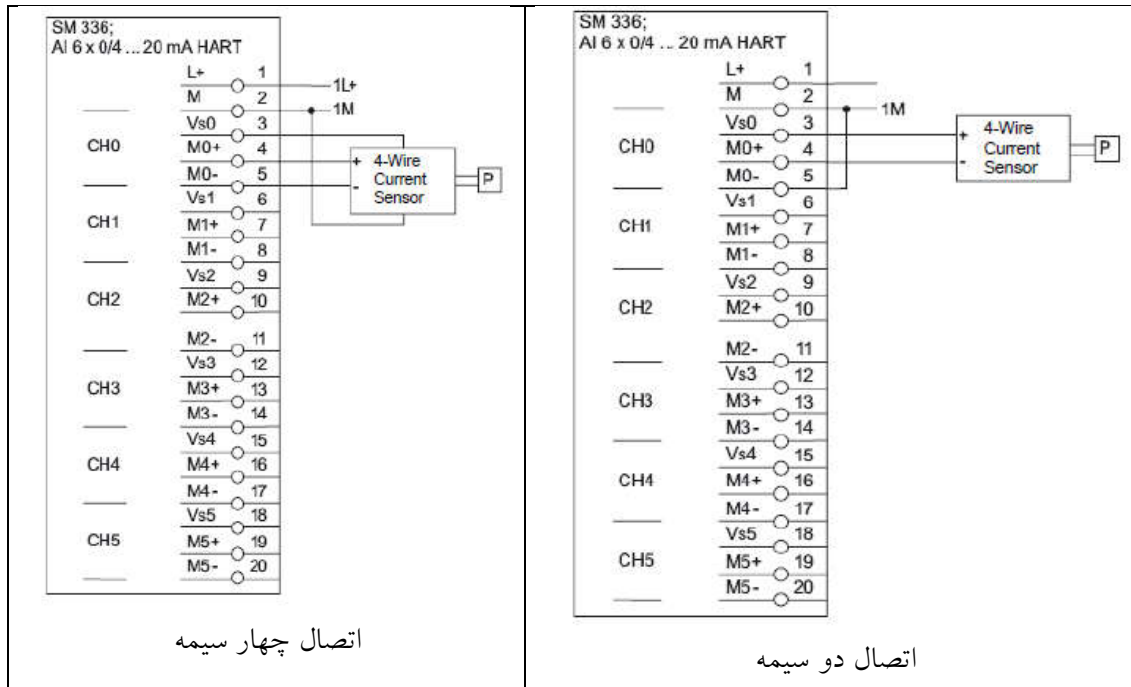
$$PFD_{1001} \approx \lambda_{DU} \cdot \frac{T_I}{2}$$

۸-۳-۲- سیم‌بندی (Wiring)

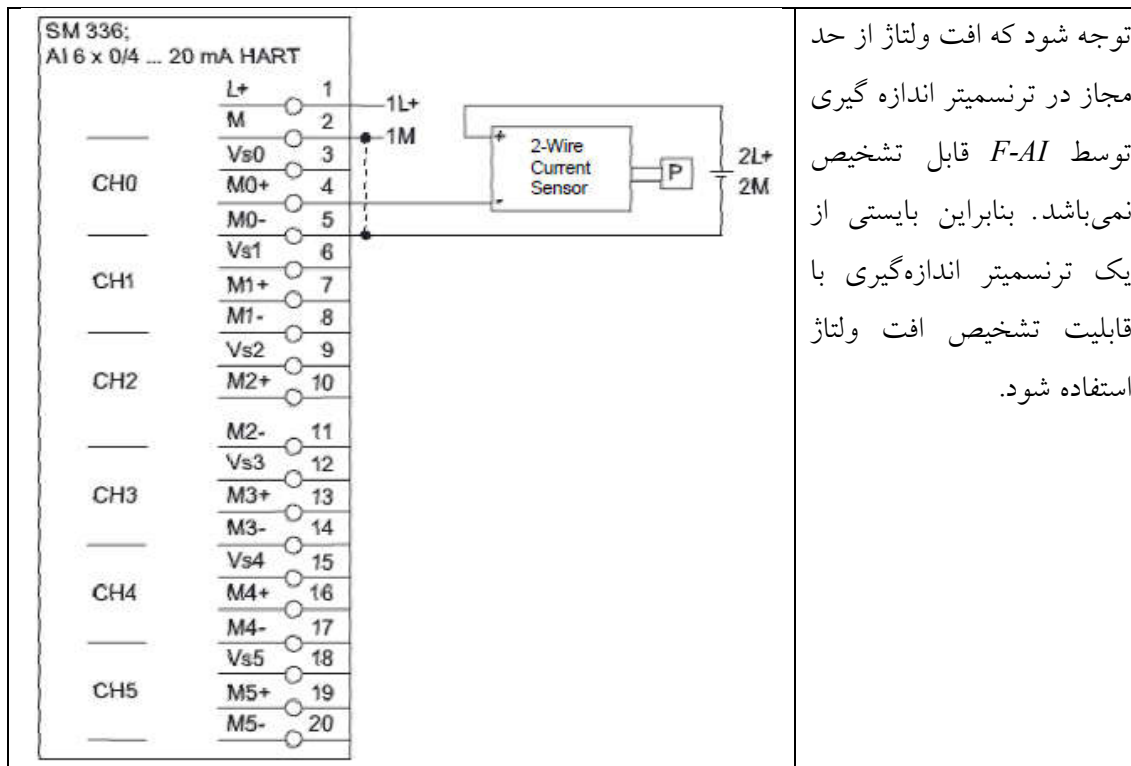
در این طرح ارزیابی 1001، سنسور را می‌توان به دو صورت زیر تغذیه کرد:

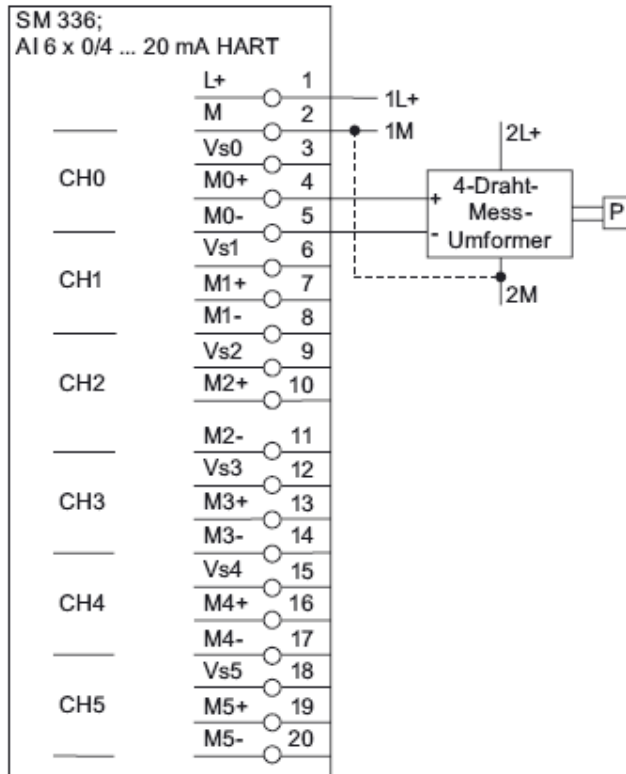
- از طریق منبع داخلی ماژول F-AI (internally)
- از طریق یک منبع بیرونی (external power supply)

شکل زیر یک مثال از سیم‌بندی یک سنسور دو سیمه و چهار سیمه را به تصویر کشیده است. که در هر دو شکل ترنسمیتر اندازه‌گیری به کانال ۰ (ترمینال ۳، ۴، ۵) متصل شده و با ماژول F-AI تغذیه شده است. لازم به ذکر است که با توجه به هارت بودن کارت سیگنال از نوع جریان می‌باشد. همچنین در شکل های زیر، حرف P معرف یک سنسور می‌باشد.



شکل زیر یک مثال از سیم بندی یک سنسور دو سیمه را با تغذیه بیرونی به تصویر کشیده است. که در آن ترنس میتر اندازه گیری به کانال ۰ (ترمینال ۲، ۴، ۵) متصل شده است.

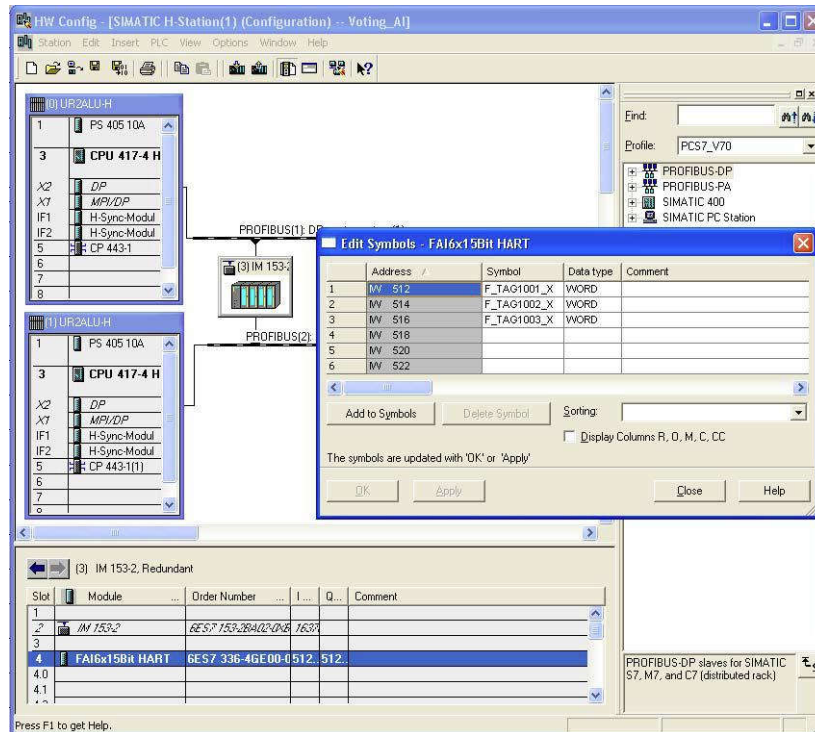




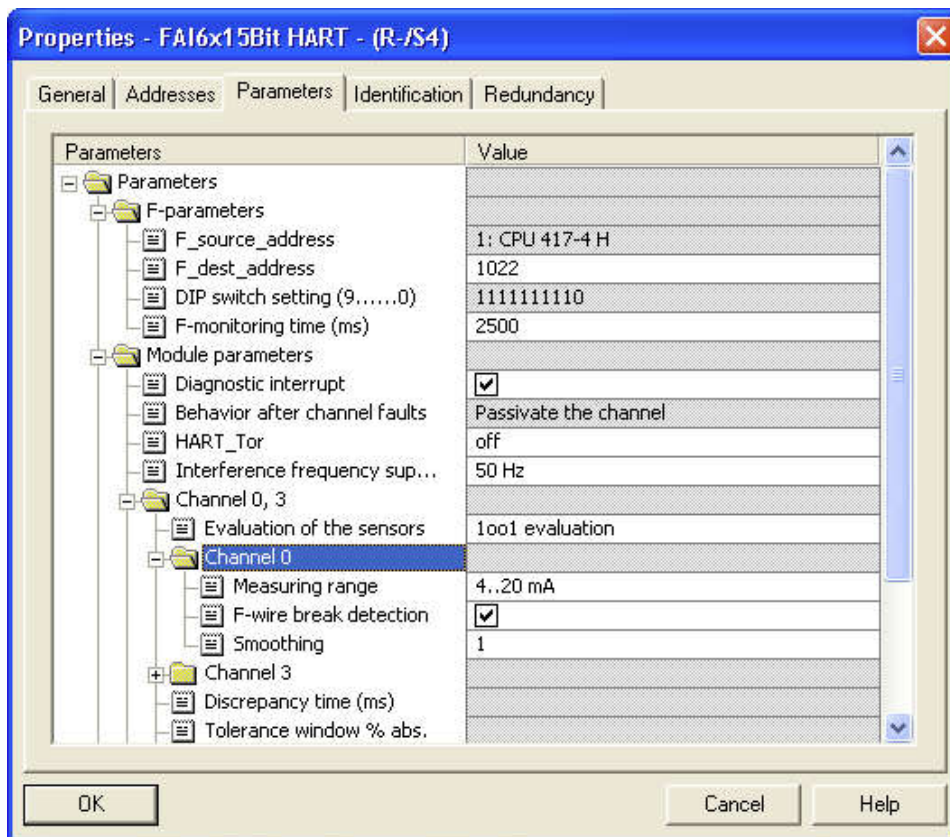
شکل ۸-۴- اتصال سنسور چهار سیمهبا تغذیه بیرونی به یک کارت F-AI

۸-۳-۳- پیکربندی ماژول در Step7

پیکربندی سخت افزاری ماژول F-AI در STEP 7 (HW Config) همانند هر ماژول F نوع ET 200M می باشد. یک مثال از پیکربندی سخت افزار F-AI در شکل زیر نشان داده شده است. در این مثال سیگنال سنسور به کانال ۰ از F-AI متصل شده است.



در پنجره *object properties* ماژول *F-AI* پارامترهای مورد نیاز برای عملکرد *F-AI* تنظیم می شود. پارامترها در جدول ۲-۲ خلاصه شده است.



پنجره پراپرتی یک ماژول F-AI

۸-۳-۴- پیکربندی معماری Iool در CFC

دو روش ممکن است برای پیاده سازی منطق CFC وجود دارد:

- بدون ارزیابی خطای کانال (Iool)
- با ارزیابی خطای کانال (IoolD)

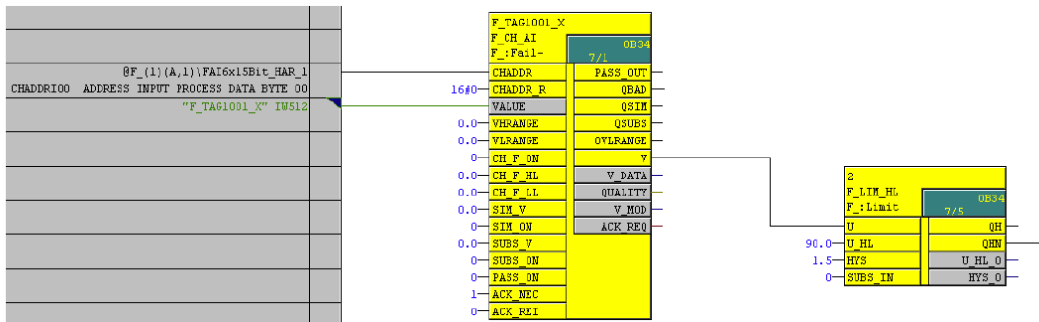
پیکربندی بدون ارزیابی خطای کانال (Iool)

شکل زیر یک پیکربندی نمونه ایجاد شده در ویرایشگر CFC را برای قرائت یک سیگنال ورودی که در آن خطای کانال در نظر گرفته نشده است، نشان می دهد. توجه داشته باشید که در این مثال که یک مقدار حد MAX فرض شده است. لذا مقدار قرائت شده از کانال توسط بلاک F_Limit برای تجاوز از حد بالا (۹۰ در این مثال) ارزیابی شده و خروجی لاجیک برای رسیدن به وضعیت ایمن، به وضعیت خاموش سوئیچ می کند. یعنی تا زمانی که مقدار کانال از حد مقدار ۹۰ تجاوز نکرده است، خروجی بلاک ۱ و به محض تجاوز از حد ۹۰ با تولورانس ۱,۵ خروجی لاجیک ۰ می شود. یعنی فرمان تریپ صادر می شود. (normal state failsafe state = 0,=1)

برای پیاده سازی لاجیک CFC بلاک های زیر استفاده می شود:

۱- از یک بلاک F_CH_AI برای قرائت سیگنال کانال

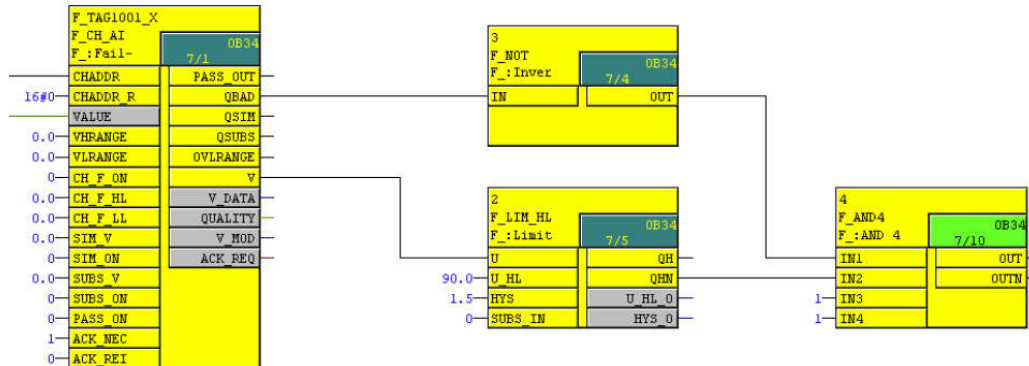
۲- از یک بلاک limit value مانند F_LIM_HL یا F_LIM_LL برای ارزیابی مقدار کانال



پیکربندی با ارزیابی خطای کانال (IoolD)

شکل زیر یک پیکربندی نمونه ایجاد شده در ویرایشگر CFC را برای قرائت یک سیگنال ورودی که در آن خطای کانال در نظر گرفته شده است، نشان می دهد. توجه داشته باشید که در این مثال که یک مقدار حد MAX فرض شده است. لذا مقدار قرائت شده از کانال توسط بلاک F_Limit برای تجاوز از حد بالا (۹۰ در این مثال) ارزیابی شده و خروجی لاجیک برای رسیدن به وضعیت ایمن، به وضعیت خاموش سوئیچ می کند. یعنی تا زمانی که مقدار کانال از حد مقدار ۹۰ تجاوز نکرده است، خروجی بلاک ۱ و به محض تجاوز از حد ۹۰ با تولورانس ۱,۵ خروجی لاجیک ۰ می شود. یعنی فرمان تریپ صادر می شود. (normal state failsafe state = 0,=1)

این مثال) ارزیابی شده و خروجی لاجیک ارزیابی برای رسیدن به وضعیت ایمن، به وضعیت خاموش سوئیچ می کند.



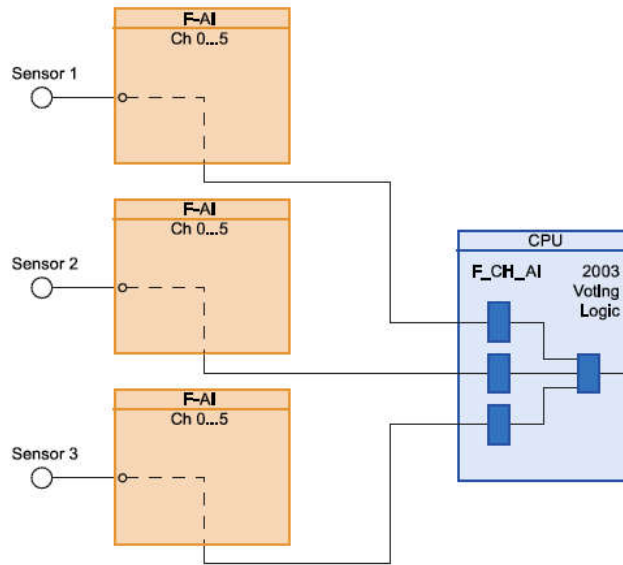
در این پیکربندی:

- برای رنج نرمال یعنی تا زمانی که مقدار کانال از حد مقدار ۹۰ تجاوز نکرده است، خروجی بلاک ارزیابی (بلاک های F_LIM_HL , AND , NOT) ۱ می باشد
- به محض تجاوز از حد ۹۰ با تولرانس ۱,۵ خروجی لاجیک ۰ می شود. یعنی فرمان تریپ صادر می شود. ($failsafe\ state = 0, normal\ state = 1$)
- اگر در قرائت مقدار PV ، خطا در کانال گزارش شود، خروجی لاجیک ۰ می شود. یعنی فرمان تریپ صادر می شود.

۸-۴- یک

۸-۵- سیستم انتخاب 2003 با ماژول آنالوگ ورودی F (2003 Voting with F-AI)

انتخاب دو از سه (2-out-of-3) از سه سنسور و به عنوان مثال از سه ماژول F-AI استفاده می کند. در این مثال هر سنسور به کانال صفر یک ماژول F-AI سیم بندی می شود. سپس تک تک سیگنال های حاصل شده از کانال ها در برنامه لاجیک ارزیابی می شود. در این نوع معماری با یک پیکربندی دو کاناله سطح SIL3 نیز قابل دستیابی است.

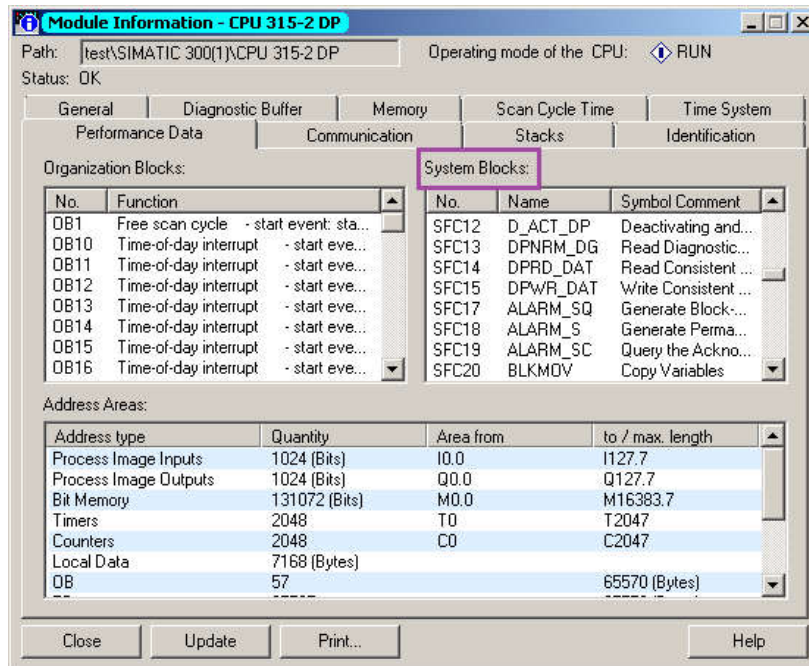


شکل ۵-۸-

۸-۵-۱- فانکشن بلاک های سیستمی پشتیبانی شده در ماژول CPU (Supported SFB & SFC)

برای تشخیص این که یک ماژول CPU چه فانکشن بلاک های سیستمی را پشتیبانی می کند. به دو روش زیر می توان عمل کرد.

- ۱- پروژه را در *Simatic Manager* باز کرده و مد آنلاین را فعال می کنیم. لذا کلیه *SFC* ها و *SFB* های که CPU پشتیبانی می کند را می توان مشاهده کرد.
- ۲- مد آنلاین را فعال کرده و از پنجره *module information* مطابق شکل زیر از سربرگ *System locks*, بلاک های پشتیبانی شده قابل مشاهده است.



۸-۶- اتصال S7 REDCONNECT

اتصال S7 REDCONNECT یک لایسنس نرم افزار برای ارتباط بین سیستم S7-400H AS و سیستم مانیتورینگ WinCC/PCS7 OS (WinCC Redundancy/PCS7 OS Redundancy) از طریق دو عدد CP1613 A2 یا CP1623 A2 می باشد. این لایسنس در سیستم H در صورت وقوع نقص در یکی از کنترل کننده ها و یا با از دست دادن ارتباط بین AS و OS، به طور خودکار اتصال از یک CP به دیگر سوئیچ می کند. برای استفاده از این قابلیت بایستی در برنامه NETPRO بین AS و OS یک اتصال از نوع Fault tolerant تعریف کنید.

اتصال ایستگاه S7 400H به یک OS را از طریق یک CP-1613 به صورت زیر انجام دهید:

- ۱- در پنجره NETPRO بر روی CPU-41X H راست کلیک کرده و گزینه Insert New Connection را انتخاب کنید.
- ۲- در پنجره جدید باز شده گزینه S7-connection fault-tolerant را از قسمت Connection type انتخاب کنید.

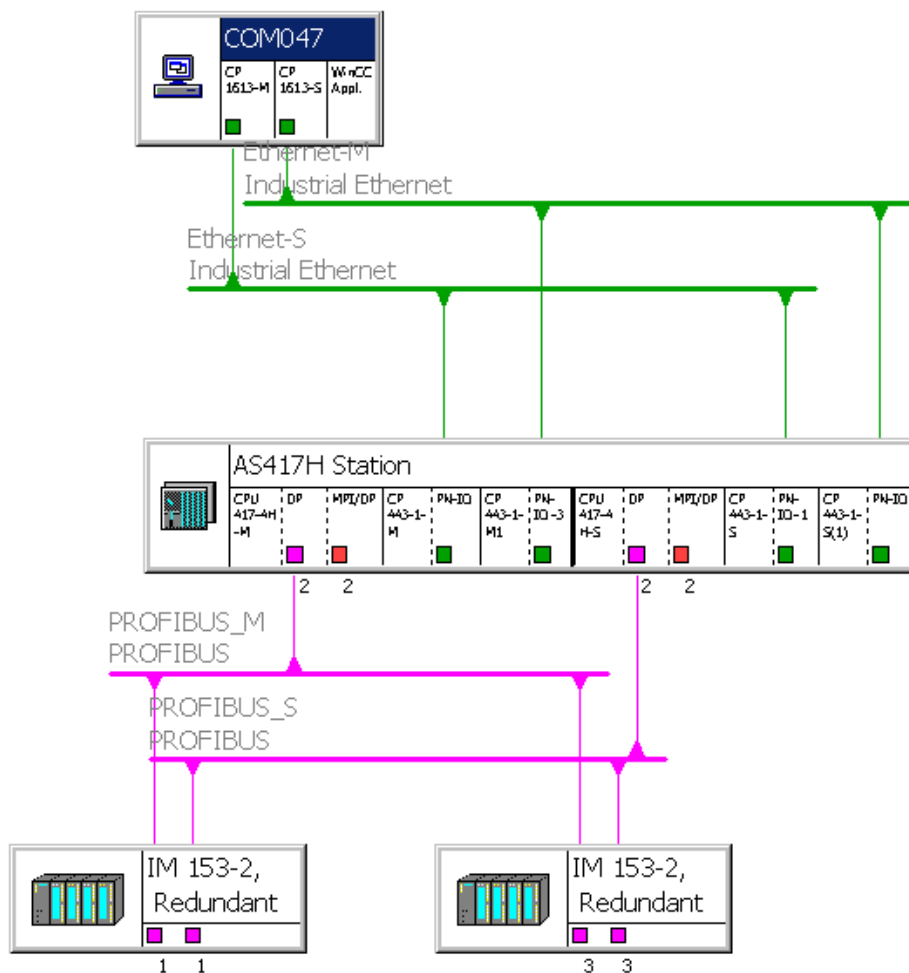
۸-۶-۱- پیکربندی اتصالات RedConnect

نکته: برقراری اتصالات *Fault-tolerant S7 connections* تنها با استفاده از کارت های *CP1613* و *CP1623* امکان پذیر می باشد. امکان پیاده سازی *Fault-tolerant S7 connections* با کارت های شبکه استاندارد وجود ندارد.

همچنین نیاز است که یکی از دو ابزار پیکربندی زیر در ایستگاه مهندسی نصب شود.

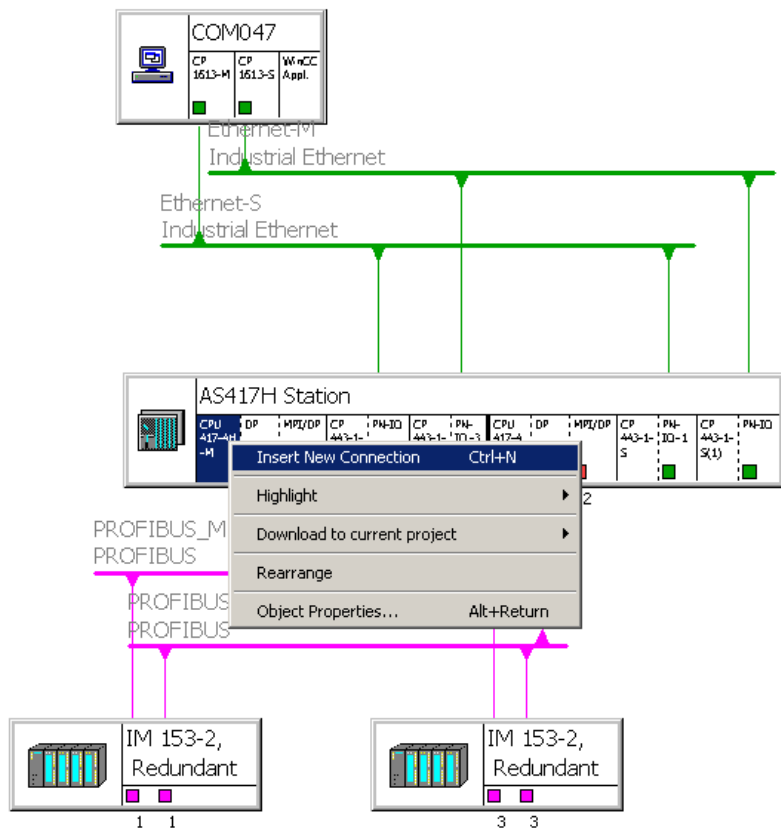
You need the license for the SIMATIC NET PC software S7-REDCONNECT on the PC station that communicates with the S7400-H system via fault-tolerant S7 connections.

در شکل زیر یک اتصال *S7 Fault-tolerant* بین سیستم *H* و کامپیوتر *HMI* برقرار شده است. که در آن خود شبکه اترنت به صورت افزونه می باشد. لذا در بخش *master* و *Slave* ایستگاه *AS* از دو عدد *CP* اترنت استفاده شده است.

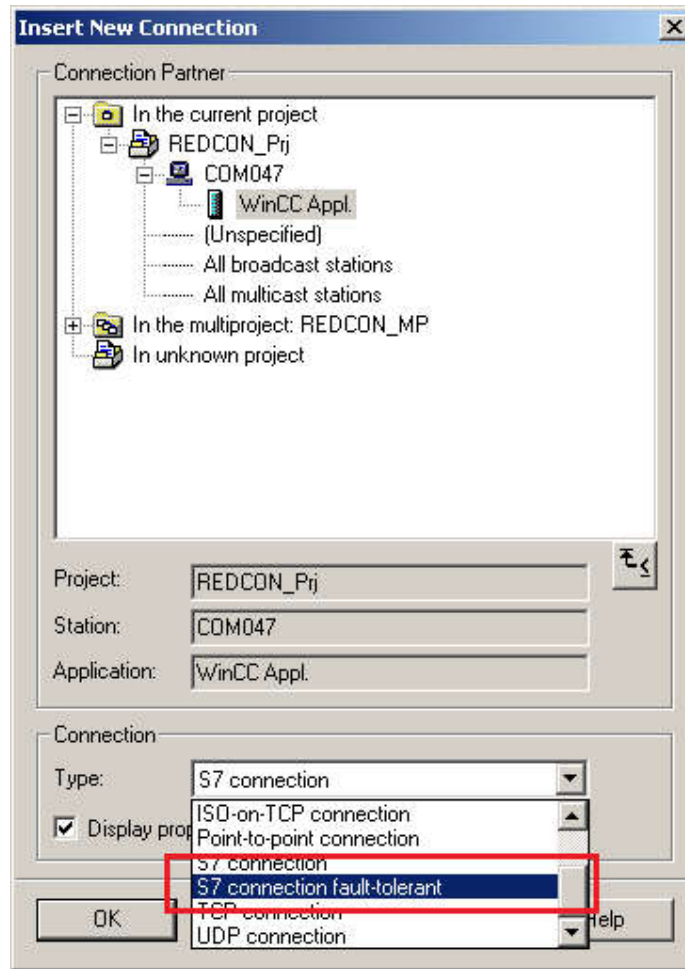


مراحل ایجاد *named Connection* به شرح زیر می باشد.

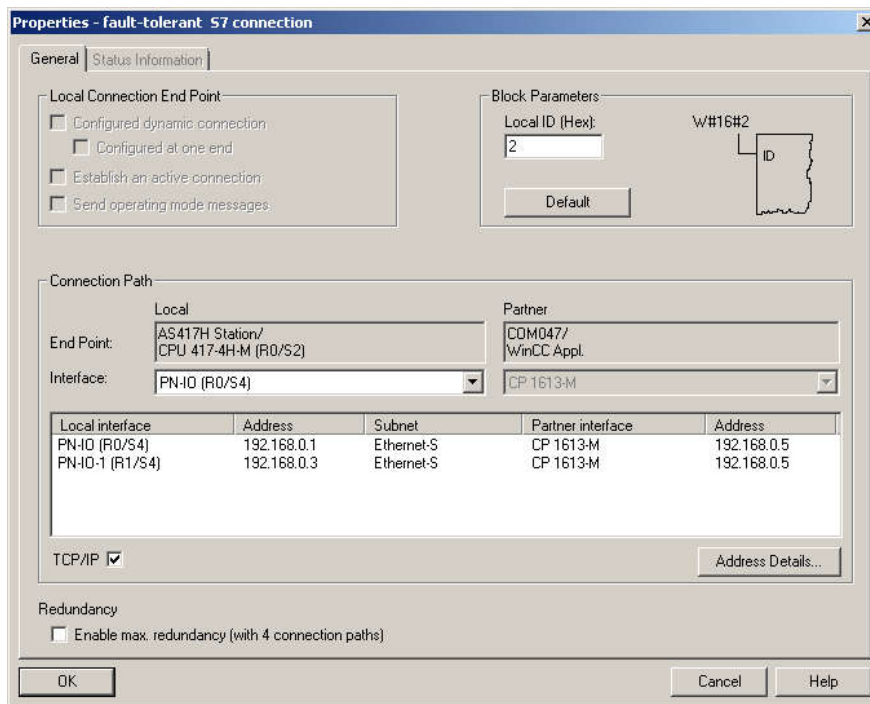
- در پنجره *NetPro* مطابق شکل زیر بر روی *CPU 417H* مستر راست کلیک کرده و گزینه *Insert New Connection* را انتخاب کنید.



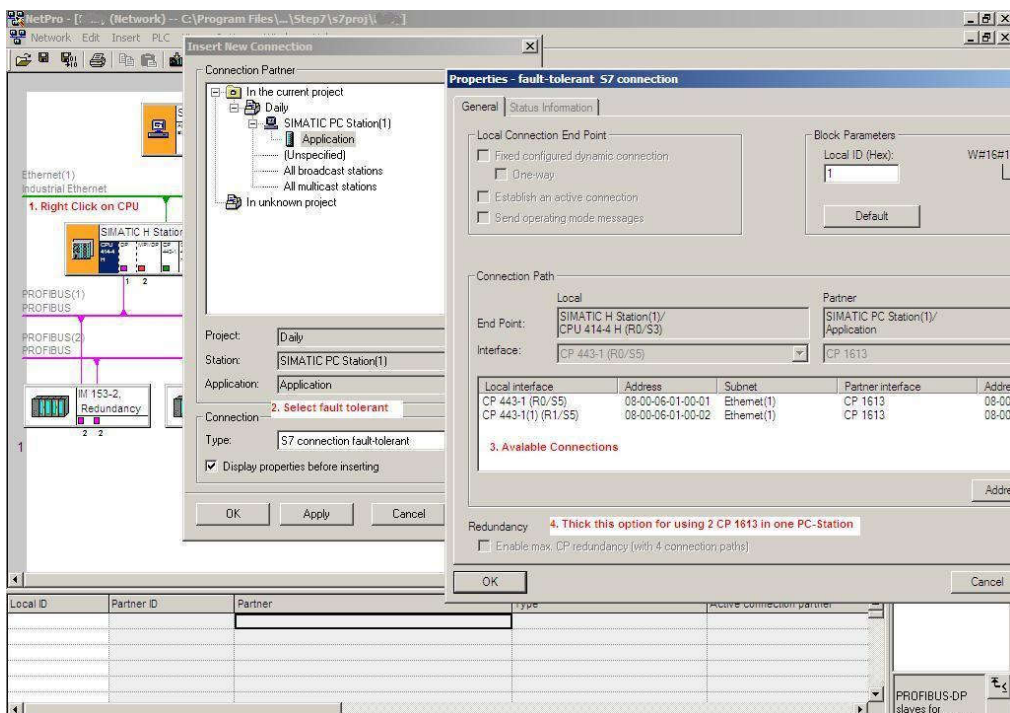
- در پایین پنجره ظاهر شده مطابق شکل زیر گزینه *S7 Connection fault tolerant* را انتخاب کرده و *Ok* می کنیم.



۳- با کلیک روی *OK* در پنجره قبلی شکل زیر ظاهر می شود.



۴- در صورتی که تیک *Enable max redundancy* را بزیم. عمل ریداندنسی از طریق ۴ مسیر برقرار می شود.



8-7- Using SFC 14 and SFC15 (for Consistent data Transfer)


Consistent data" basically means that you are transferring large amounts of data as a single unit, in a single communication cycle.

Consistent Data with the size of 1, 2 or 4 byte should be read / write via a direct access without the SFC's.

۸-۸- فضای حافظه CPU (S7 System Memory Areas)

The system memory of the S7 CPUs is divided into address areas (see table below). Using instructions in your program, you address the data directly in the corresponding address area.

Address Area	Access via Units of Following Size	S7 Notation (IEC)	Description
Process image input table	Input (bit)	I	At the beginning of the scan cycle, the CPU reads the inputs from the input modules and records the values in this area.
	Input byte	IB	
	Input word	IW	
	Input double word	ID	
Process image output table	Output (bit)	Q	During the scan cycle, the program calculates output values and places them in this area. At the end of the scan cycle, the CPU sends the calculated output values to the output modules.
	Output byte	QB	
	Output word	QW	
	Output double word	QD	
Bit memory	Memory (bit)	M	This area provides storage for interim results calculated in the program.
	Memory byte	MB	
	Memory word	MW	
	Memory double word	MD	
Timers	Timer (T)	T	This area provides storage for timers.
Counters	Counter (C)	C	This area provides storage for counters.
Data block	Data block, opened with "OPN DB":	DB	Data blocks contain information for the program. They can be defined for general use by all logic blocks (shared DBs) or they are assigned to a specific FB or SFB (instance DB).
	Data bit	DBX	
	Data byte	DBB	
	Data word	DBW	

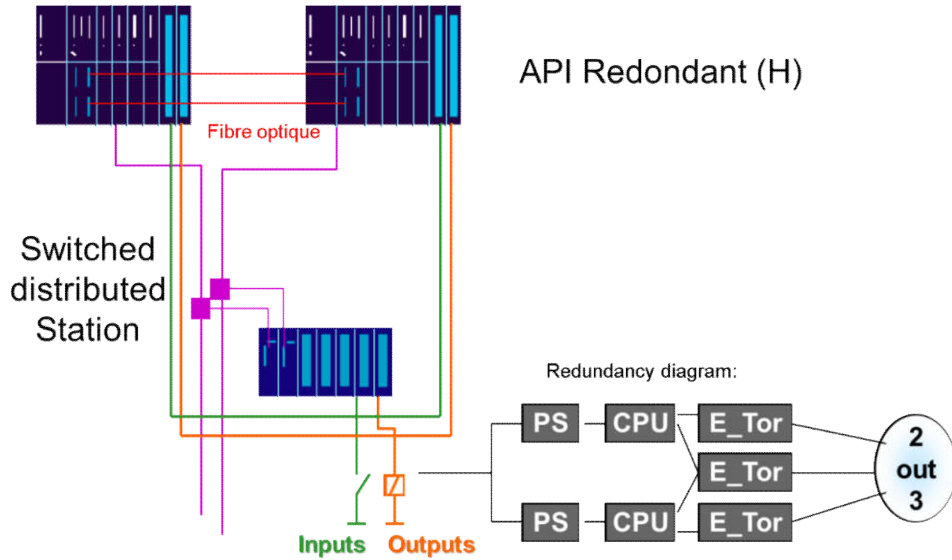
	Data double word	DBD	
	Data block, opened with "OPN DI":	DI	
	Data bit	DIX	
	Data byte	DIB	
	Data word	DIW	
	Data double word	DID	
Local data	Local data bit	L	This area contains the temporary data of a block while the block is being executed. The L stack also provides memory for transferring block parameters and for recording interim results from Ladder Logic networks.
	Local data byte	LB	
	Local data word	LW	
	Local data double word	LD	
Peripheral (I/O) area: inputs	Peripheral input byte	PIB	The peripheral input and output areas allow direct access to central and distributed input and output modules (DP).
	Peripheral input word	PIW	
	Peripheral input double word	PID	
Peripheral (I/O) area: outputs	Peripheral output byte	PQB	
	Peripheral output word	PQW	
	Peripheral output double word	PQD	

۸-۸-۱- عمل غیر فعال سازی مازول های ورودی / خروجی (Passivation)

غیرفعال کردن بدان معنی است که در صورت وقوع یک خطا، یک یا تعدادی از کانال ها در یک مازول سیگنال F^+ ، به یک وضعیت امن سویچ می کند. در صورت خطا در کانال (برای مثال یک سنسور معیوب)، تنها کانال های آسیب دیده پسو می شوند.

در صورت وقوع خطا در مازول (به عنوان مثال یک خطای ارتباطی)، تمام کانال ها بر روی مازول I/O خرابی امن پسو می شوند.

اگر یک ماژول سیگنال F خطایی تشخیص دهد، کانال های آسیب دیده و یا همع کانال های آن را به حالت امن سوئیچ می کند. به عبارت دیگر، کانال های این ماژول غیرفعال می شود. ماژول سیگنال F برای نشان دادن این که خطایی شناسایی شده است، یک پیام به درایور کانال F و $PCS 7 OS$ می فرستد.



۸-۹- جدول اختصارات و تعاریف

<i>HFT</i>	<i>hardware fault tolerant</i>
<i>DC</i>	<i>diagnostic coverage</i>
<i>SFF</i>	<i>safe failure fraction</i>
<i>EUC</i>	<i>Equipment Under Control</i>

۸-۹-۱- قابلیت های کارت های Fail Safe

کارت های *Fail Safe* دارای این قابلیت هستند که با نصب بسته *S7 F system* امکان تعریف سیستم ارزیابی یا *Voting*، ۲ از ۳ برای ورودی های آنالوگ فراهم می شود. مثلاً تو خط ساکشن کمپرسور گاز سه عدد *PT* نصب می کنند. به طوری که اگر حداقل دو عدد از سه ترانسسمیتر از حد تریپ گذشتند کمپرسور تریپ می خورد.

قابلیت دیگر این است که برای ورودی دیجیتال هم ولتاژ، ۲۴ ولت که قرار است سر سوئیچ برود می تواند از تو خود کارت (*VS*) ایجاد بشود و این ولتاژ مانیتور میشه که اگه تغذیه سر سوئیچها (یا با تنها مانند امرجنسی) مشکلی پیدا کرد آلارم ایجاد بشه.

و در آخر چون فقط برای EMS و سیستمهای SAFETY استفاده می شوند خروجی آنالوگ تو این دسته کارتها نیست.

برداشت من از مفهوم Fail Safe ارتباط اون تا اندازه ای با درجه SIL بود و کاملا فکر میکنم به خود کارت مربوط باشه و نوع ساخت و سخت افزار اون و رفتاری که در هنگام بروز خطا کارت از خودش نشون میده

همانطوری که گفتید در بحث SIL، مفهومی بنام pfd یا probability of failure ondemand وجود دارد که احتمال بروز خطای منجر به حادثه را بیان میکند و همین مقدار نشان دهنده سطح SIL است بطوریکه هرچقدر pfd کمتر باشه سطح SIL بالاتر میره

حالا کارتها یا تجهیزاتی که احتمال بروز خطای حاد در آنها وجود دارد، SIL تو اونها در سطح تعریف شده باشه اون تجهیز اون سطح از SIL رو خواهد داشت و اصطلاحا به این تجهیزات fail safe گفته میشه این تجهیزات معمولا در زمان بروز خطا عملکردی از خودشون نشان میدن تا safety فرایند حفظ بشه حالا این تشخیص خطا و شرایط fail هم بصورت سخت افزاری و هم بصورت نرم افزاری و با استفاده از الگوریتم های خاص صورت میگیره

در مورد redundancy هم باید بگم که به تنهایی تضمین کننده safety نیست و تنها availability رو بالاتر میبره اما در مورد تجهیزات fail safe که سطح SIL مشخصی رو دارند redundancy میتونه سطح SIL رو بالاتر ببره

برای مثال کارتهای ۲۴ کاناله fail safr زمینس دارای سطح سیل ۲ است که با ریداندت کردن دو کانال روبروی هم، این سطح SIL به ۳ ارتقا پیدا می کنه و عملا کارت ۱۲ کاناله میشه.

اگر محاسبات pfd رو توجه کنید می بینید که redundancy باعث کاهش pfd میشه و میتونه سطح sil رو افزایش بده (رجوع شود به کتاب sil شرکت gmi)

If you are using Manufacturer’s prior use data because a selected product does not reach the required level under FMEDA analysis, be aware that there are significant requirements on the end user. A mature product must generally be used to have the required field experience, and the design and assembly must be “frozen in time” in such a way that no upgrades, modifications or even configuration changes may be allowed that may render the “Proven In Use” data useless.

A key result of the analyses is establishing a Safe Failure Fraction (SFF) for a product. Figure F below shows the relationship of SFF values, SIL ratings and the effects of redundancy.

Figure F

Safe Failure Fraction (SFF) <i>(for Type B, microprocessor-based devices)</i>	No Redundancy	Single Redundancy	Double Redundancy
<60%	Not Allowed	SIL 1	SIL 2
60%<90% <i>(typical competitor)</i>	SIL 1	SIL 2	SIL 3
90%<99% <i>(Eclipse, Jupiter, E3)</i>	SIL 2	SIL 3	SIL 4
>99%	SIL 3	SIL 4	SIL 4

While two SIL 1 devices can be used together to achieve SIL 2 and two SIL 2 devices to achieve SIL 3 (as suggested by the chart above), it is not automatic. Using redundancy to attain a higher SIL rating has additional requirements beyond hardware. It has an additional requirement of systematic safety which includes software integrity.

It is important to note that the most conservative approach to redundancy is to use dissimilar technologies. This reduces failures due to application issues.

فصل ۹ - نکات مربوط به پیکربندی در Step 7

۹-۱ - آدرس پایه تبادل داده برای CP343

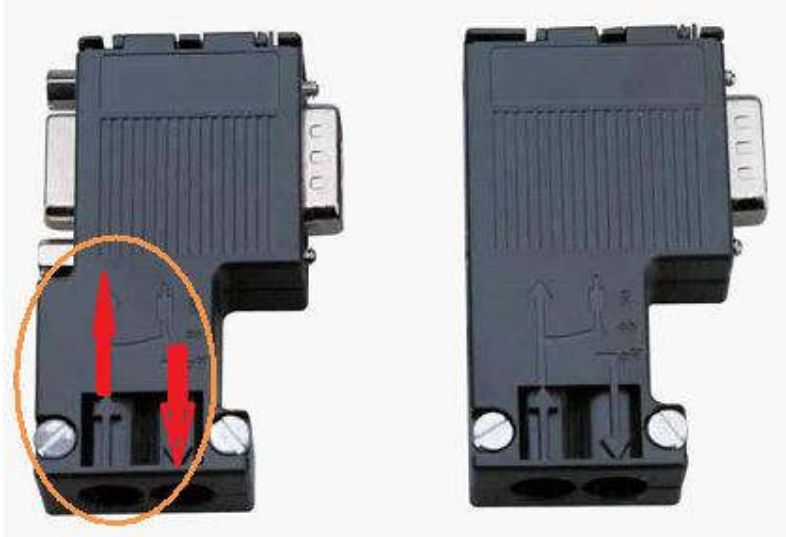
The image shows the configuration of a CP 343-1 module in SIMATIC Manager. In the HW Config window, slot 4 is assigned to CP 343-1. Below it, a table lists the module details:

Slot	Module	Order number	Firmware	MPI address	I address	Q address	Comment
1	PS 307 10A	6ES7 307-1EA01-0AA0					
2	CPU 315-2 DP	6ES7 315-2AH14-0AB0	V3.3	10			
3	DP Master				204.7*		
4	CP 343-1	6GK7 343-1EX30-0XE0	V3.0	5	260...275	260...275	
X1	PN-IO				1023*		
X1A	Port 1				1022		
X1B	Port 2				1021*		

The Properties dialog for the ISO-on-TCP connection shows the following settings:

- Local Endpoint ID (hex): 0001 A050
- Name: ISO-on-TCP connection 1
- Via CP: CP 343-1, PN-IO (R0/S4)
- Block Parameters: ID (circled in red) and LADDR (W#16#0104). A red arrow points from the ID field to the text "Verb_ID" below it.

نکته: نحوه اتصال کانکتور پروفی باس در سمت CPU از مسیر ورود کابل و در ETها از مسیر خروج کابل می باشد.



–۲–۹

1- Fail-safe modules

- The major difference between fail-safe modules and standard ET 200S modules is that failsafe modules have a two-channel internal design.
- Both integrated processors monitor each other, automatically test the I/O circuits, and set the F-module to safe state in the event of a fault. The F-CPU communicates with the fail-safe module using the PROFIsafe safetyrelated bus profile.

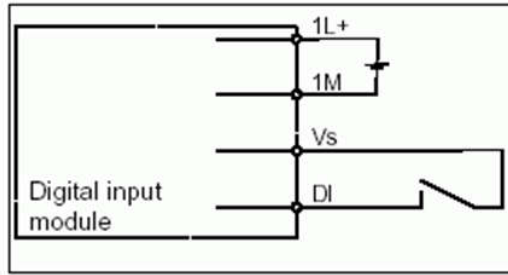
1-0-1- Ioo1 and Ioo2 evaluation with the fail safe signal module SM 326F

پرسش:

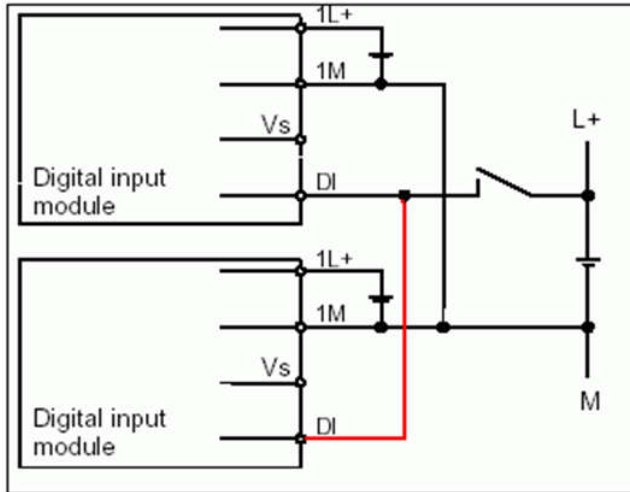
معنای آنچه که توسط ارزیابی Ioo1 و ارزیابی Ioo2 با ماژول سیگنال fail-safe (SM 326F) صورت می گیرد چیست دارند؟

پاسخ:

برای ارزیابی Ioo1، به ازای هر سیگنال فرآیند یک سنسور تک کانال (one single-channel sensor) به ماژول های دیجیتال متصل می شود. با این حال، کاربردهای متعددی برای سنسورهای تک کانال وجود دارد. نحوه اتصال بستگی به کلاس ایمنی مورد نیاز است.



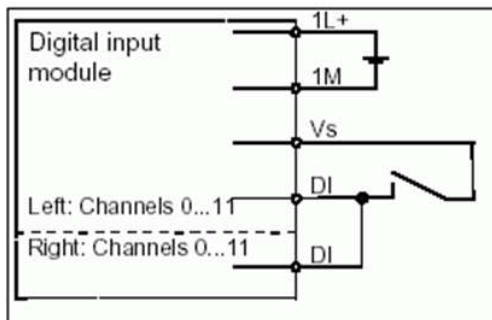
اتصال یک ماژول



اتصال دو ماژول

نکته:

مطمئن شوید که برای اتصالی مانند شکل ۲، جامپر برای ورودی دیجیتال از ماژول دوم انجام شده باشد. در ارزیابی 1002، یک حسگر توسط یک ماژول بررسی می شود. یک سنسور به دو کانال مختلف که در آن کانال چپ و راست باید همیشه شماره کانال یکسانی داشته باشند، متصل می شود. یک نسخه دیگر از ارزیابی 1002 این است که در آن دو سنسور و هر یک به کانال های سمت راست و چپ کانال متصل می شود. در این جا نیز شماره کانال ها نیز باید یکسان باشد.



اتصال یک سنسور به دو کانال

Let us take an example of a Ioo2: an Emergency push button.

The push button is equipped with two contacts. Both contacts opens or close nearly at the same time (a few milliseconds difference is possible).

A Ioo2 input has two physical channels (let us say the left and the right channel).

One contact is connected to the left channel, the second to the right one.

But in your program you interrogate only one bit, because it is the input card that evaluates if both contacts deliver the same information.

In case of discrepancy (the contacts deliver a different status) the failsafe input board will deliver a "0" (once the discrepancy timeout is elapsed). You will get a "1" only if both contacts deliver an ON signal.

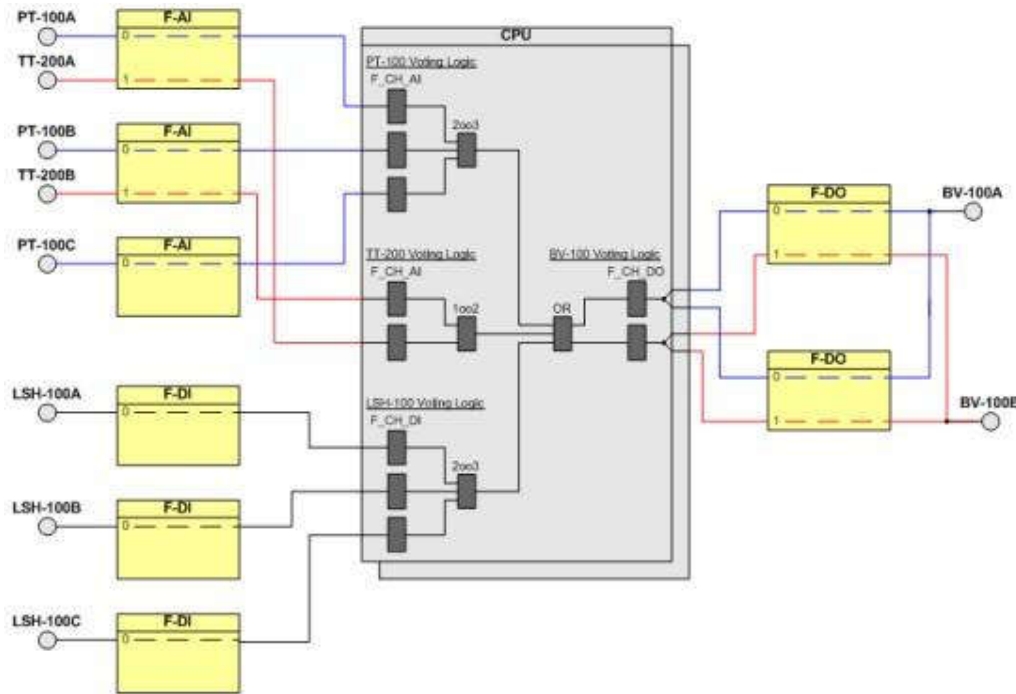
1-0-2- F Systems: Wiring and Voting Architectures for ET200M F-DIs and F-Dos

هدف:

چند سیگنال دیجیتال *failsafe* بایستی مانیتور شده و چند عملگر *failsafe* باید در یک پلنت کنترل شود. بسته به اهمیت و خطر شکست (*failure risk*)، گزینه های متعددی برای سیم بندی و داوری (*Voting*) سیگنال وجود دارد.

راه حل:

عمل داوری می تواند برای مثال، در ماژول ورودی دیجیتال و یا در *CPU* تحقق یابد. امکان های مختلف از سیم بندی و داوری سیگنال *failsafe* در این مثال عملکردی نشان داده شده است. همچنین سیم بندی و داوری خروجی های *failsafe* نیز نشان داده شده است.



۱-۱- کنترل کننده های Fail-Safe شرکت Siemens

شرکت زیمنس با نوآوری سیستم های ایمنی یکپارچه، راه حلی جدید برای طراحی کنترلرهای Fail-Safe ارائه کرده است که تمامی مشخصات یک کنترلر ایمن را در بر می گیرد.

شکل ۱- کنترلرهای Fail-Safe

این کنترلرها جهت کنترل فرایندها و همچنین هدایت و نگاه داشتن آنها در حالت ایمن در هنگام وقوع نقص و خطا در سیستم بکار می رود. در کنار کارایی بالای این کنترلرها، یکپارچه بودن نیز به خوبی رعایت شده است. با بکارگیری این کنترلرها می توان از عوارض و خسارات ناشی از نقص در عملکرد سیستم کنترل، جلوگیری نمود. بهبود در موضع یابی محل نقص به دلیل وجود اطلاعات جامع تشخیص خطا، از سرگیری تولید را پس از یک وقفه در سیستم Safety، قادر می سازد.

۱-۱-۱- چگونه پردازش در یک کنترلر Fail-Safe شرکت Siemens

یکی از مشخصات اصلی یک سیستم Fail-Safe، تشخیص خطا می باشد. تا کنون شرکت های سازنده این نوع از کنترلرها، با بکارگیری دو عدد CPU که صحت عملیات را بازمینی می کردند، این مشخصه را محقق می ساختند. برخلاف دسترس پذیری بالای سیستم در جایی که redundancy ساختاری برای افزایش

دسترس پذیری سیستم اتوماسیون استفاده می شود، این دو کانال *redundancy* جهت آشکار کردن خطا بکار می رود.

بجای استفاده از روش تجاری فوق، راه کار ابتکاری *redundant* و *diverse* برای پردازش دستورات، در کنترلرهای *Fail-Safe* شرکت زیمنس مورد استفاده قرار می گیرد. در این طراحی با وجود بهره گیری از یک *CPU* (که دو پردازشگر درون آن جاسازی شده است)، می توان به همان قابلیت تشخیص خطا که در روش قدیمی بکارگیری دو *CPU* را ایجاد می کرد، دست یافت. شکل ۲ به خوبی اصول این روش را نمایش می دهد.

در این روش، برنامه کاربردی دو بار به صورت همزمان در دو پردازشگر مختلف در یک *CPU*، یکبار در منطق مثبت و بار دیگر در منطق متمم مورد پردازش قرار می گیرد. خطاهای احتمالی از مقایسه این دو نتیجه آشکار می شود. مزیت این روش به وضوح دیده می شود: تنها یک *CPU* برای حصول *Safety* مورد استفاده قرار می گیرد و بنابراین یک واسطه پیچیده برای تحقق سیستم *Fail-Safe* حذف می شود. در پروسه هایی که از نظر *Safety* بایستی دسترس پذیری بیشتری داشته باشند، دو *CPU* با کابل نوری بهم متصل می شوند. در صورتیکه یک *CPU* دچار مشکل شود، *CPU* دیگر کنترل پروسه را ادامه می دهد. برای دستیابی به استانداردهای ایمنی از *SIL1* تا *SIL3*، *Category4* و *AK6*، صرف نظر از پردازش به صورت *Diverse*، روتین های بازبینی و تشخیص خطا در این کنترلرها اعمال می گردند.

۱-۱-۲ PROFIBUS با پروتکل PROFIsafe برای ارتباطات Safety

PROFIsafe به عنوان اولین پروتکل مورد تایید *IEC 61508* که اجازه انتقال اطلاعات *Safety* و استاندارد را بر روی یک باس داده می دهد، شناخته شده است.

PROFIsafe از همان اجزاء و سرویس های *PROFIBUS* برای ارتباطات *Fail-Safe* استفاده می کند. یک *PLC* از نوع *Fail-Safe* با زیر مجموعه ای از ماژول های *Fail-Safe*، اطلاعات لازم را بدون نیاز به سخت افزار اضافی برای این ارتباط تبادل می کند.

بنابراین *PROFIBUS* جهت کاربردهای *Safety*، بدون مجبور شدن به تغییر *Cabling* گسترش یافته است و راه حل های موجود بدون هیچ تغییری در سخت افزار قابل استفاده می باشند.

PROFIBUS با *PROFIsafe* دارای گواهی نامه های استانداردهای زیر می باشد:

IEC 61508 (Up to SIL3)

EN954 (Up to Category4)

NFPA 79 - 2002

NFPA 85

شکل ۲- چگونگی استفاده از روش *redundant* و *diverse* برای پردازش دستورات در PLC های *Fail-Safe* شرکت زیمنس

شکل ۳- تا کنون سیستم های اتوماسیون *Fail-Safe* و استاندارد به صورت مجزا بودند!
شکل ۴- در حال حاضر با استفاده از تکنولوژی ایمنی یکپارچه شرکت *Siemens* سیستم های اتوماسیون استاندارد و *Fail-Safe* در یک سیستم تلفیق شده اند.

۱-۱-۳- مزایای استفاده از کنترلرهای *Fail-Safe* شرکت زیمنس

- ۱- کاهش تجهیزات مورد استفاده
 - ۲- به کارگیری یک کنترلر برای اتوماسیون استاندارد و *Fail-Safe*
 - ۳- امکان قرارگیری برنامه های استاندارد و *Fail-Safe* در یک کنترلر
 - ۴- امکان استفاده از یک کابل *PROFIBUS* برای ارتباطات استاندارد و *Fail-Safe* تا *SIL3/Category4/ AK6*
 - ۵- امکان استفاده از یک *I/O* توزیع شده برای ماژول های استاندارد و *Fail-Safe*
 - ۶- پیکربندی و راه اندازی سریع سیستم
- ۷- یکپارچگی بهتر سیستم؛ بدلیل یکسان بودن عملیات مهندسی برای اجزای کنترلی استاندارد و *Fail-Safe*

- ۳- انعطاف پذیری بیشتر
- ۴- به جای سیم بندی های سخت افزاری، اساس کار بر نرم افزار می باشد.
- ۵- سهولت اعمال تغییرات، توسعه و مستند سازی
- ۶- پیاده سازی کارهای پیچیده مربوط به سیستم *Safety* با سهولت بیشتر
- ۷- قابلیت تکثیر سریعتر
- ۸- سادگی بیشتر اعمال راه حل های نرم افزاری در قیاس با راه حل های سخت افزاری
- ۹- امکان بهره گیری از بلوک های نرم افزاری مجاز و استاندارد
- ۱۰- تعمیر و نگهداری [۱۳] ساده تر
- ۱۱- تعمیر و نگهداری ساده تر به علت وجود سیستم تشخیص خطای جامع

۱-۱-۴- کاربردها

تا کنون، کنترلرهای *Fail-Safe* شرکت *Siemens* در صنایع گوناگونی بکار رفته اند که تعدادی از آنها در ذیل آمده است:

- *Processing industry*
- *Chemical Industry*
- *Pharmaceutical industry*
- *Automobile*
- *Mechanical Engineering*
- *Glass Industry*
- *Food and Luxury Industry*
- *Conveying Engineering*

همچنین وجود کتابخانه *F* که به صورت انحصاری برای سیستم های *Burner* طراحی شده است، استفاده از این نوع کنترلر ها را به صورت گسترده ای در صنایع مرتبط با *Burner* سبب شده است.

۱-۱-۵- استانداردها و گواهینامه ها

برای تایید کارکرد مناسب مربوط به ایمنی یک ماشین یا فرایند، اجزای حفاظتی مربوط به سیستم *Safety* و تجهیزات کنترل بایستی به گونه ای قابل اعتماد کار کرده و در هنگام بروز خطا فرایند مورد کنترل را به حالت ایمنی هدایت کنند.

به همین منظور هر محصول مربوط به این نوع از محصولات، بایستی برخی از شرایط و آزمون های خاص را با موفقیت پشت سر بگذارند. این شرایط در استانداردهای مختلفی گرد آوری شده اند. داشتن گواهینامه مربوط به یک استاندارد خاص برای یک محصول، سندی از برآورده شدن شرایط احراز استاندارد مورد نظر برای آن محصول می باشد. تمامی ماژول های *Fail-Safe* شرکت زیمنس شرایط استانداردهای زیر را بر آورده می سازند:

تا *IEC 61508 (SIL3)*

تا *EN 954 (Category4)*

NFPA 79 - 2002

NFPA 85

TÜV certificate no: Z2 02 03 20411 009

UL 1998, UL 508, UL 991

- Safety integrated* -۱
- Safe* -۲
- Availability* -۳
- Complementary* -۴
- Distributed I/O* -۵
- Configuration* -۶
- Commissioning* -۷
- Flexibility* -۸
- Modification* -۹
- Expansion* -۱۰
- Documentation* -۱۱
- Reproducibility* -۱۲
- Maintenance* -۱۳

فصل - ۱۰ نصب لایسنس در PCS7

برای راه اندازی قابلیت *maintenance* در PCS7 نسخه 8.1 نیاز است که لایسنس های زیر نصب شود.

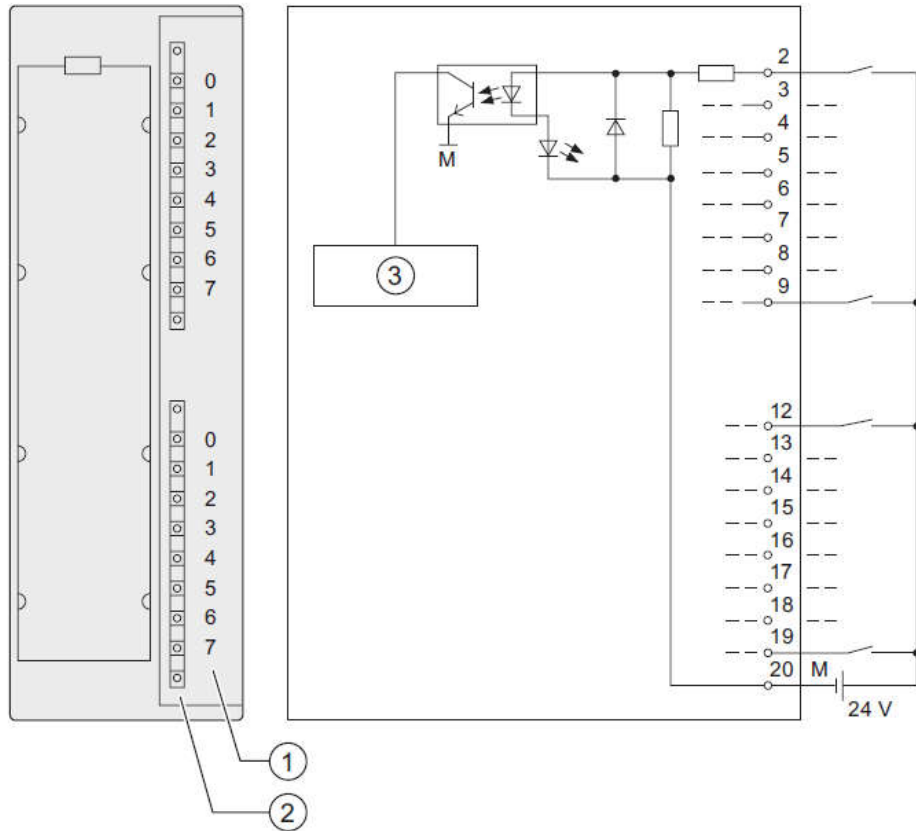
Select	Short name	Long name	License name	Version
<input type="checkbox"/>	2072	SIFLMAINES0801	Maintenance ES	v8.1
<input type="checkbox"/>	1353	SIFLYMAINR93991	Maintenance RT V7.1 (unlimited)	v99.9
<input type="checkbox"/>	1354	SIFCYMAINR93992	Maintenance RT V7.1 (10)	v99.9
<input type="checkbox"/>	1355	SIFCYMAINR93994	Maintenance RT V7.1 (100)	v99.9
<input type="checkbox"/>	1356	SIFCYMAINR93997	Maintenance RT V7.1 (1000)	v99.9

Status	Family	Product	Version	Number of license keys	License key	License number	Standard license type	License type	Validity
✓	SIMATIC HMI	WinCC RT PO (unlimited)	7.2	1	SIFLA9WRP90702	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC HMI	WinCC RT (1500000)	7.2	1	SIFLA9WRTA0702	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC HMI	WinCC RC PO (2000)	7.2	1	SIFLA9WCP30702	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC HMI	WinCC RC (1500000)	7.2	1	SIFLA9WRC0702	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC HMI	WinCC ES PO (unlimited)	7.2	1	SIFLA9WCP90702	00411000002015006...	Floating	Unlimited	Unlimited
→	SIMATIC HMI	WinCC Arch (10000)	-	1	SIFCY9WGTG99995	00411000002015006...	Floating	Count relevant	1000
→	SIMATIC STEP 7	Version Trail	8.1	1	SIFLA1VERT0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	TH	8.1	1	SIFLA1THP0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC STEP 7	STEP 7 Professional Edition 2010	5.5	1	SIFLSTPROF0505	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	SFC-Visualization	8.0	1	SIFLA1SFC0800	00411000002015007...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	SFC	8.1	1	SIFLA1SFC_0801	00411000002015006...	Floating	Unlimited	Unlimited
→	SIMATIC PCS 7	OS Server Redundancy	8.1	1	SIFLP7OSRE0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	OS Server Basic	8.1	1	SIFLP7SERB0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	OS Server	8.1	1	SIFLP7OSSV0801	00411000002015006...	Floating	Unlimited	Unlimited
→	SIMATIC PCS 7	OS RT PO(unlimited)	-	1	SIFLUP7OR19999	00411000002015006...	Floating	Unlimited	Unlimited
→	SIMATIC PCS 7	OS Engineering	8.1	1	SIFLP7OSE0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	OS Client	8.1	1	SIFLP7CLIE0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	OPC DBA	8.1	1	SIFLA1DBA0801	00411000002015006...	Floating	Unlimited	Unlimited
→	SIMATIC PCS 7	Maintenance RT (1000)	-	1	SIFCYMAINR99997	00411000002015007...	Floating	Count relevant	1000
✓	SIMATIC PCS 7	Maintenance ES	8.1	1	SIFLMAINES0801	00411000002015007...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	IEA	8.1	1	SIFLA1IEA_0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC STEP 7	CFC STEP 7 Option	8.1	1	SIFLA1CFC0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	CFC - unlimited	8.1	1	SIFLA1CFC0801	00411000002015006...	Floating	Unlimited	Unlimited
✓	SIMATIC PCS 7	AS RT PO (10000)	-	1	SIFPZASRP9999A	00411000002015006...	Floating	Countable objects	10000 (10000)
→	SIMATIC PCS 7	AS Engineering	8.1	1	SIFLP7ASE0801	00411000002015006...	Floating	Unlimited	Unlimited

۱۰-۱-۱- سیم بندی ماژول های S7300

۱۰-۱-۱-۱- سیم بندی ماژول های دیجیتال ورودی (DI)

➤ 6ES7 321-1BH02-0AA0



- ① Channel number
- ② Status display - green
- ③ Backplane bus interface

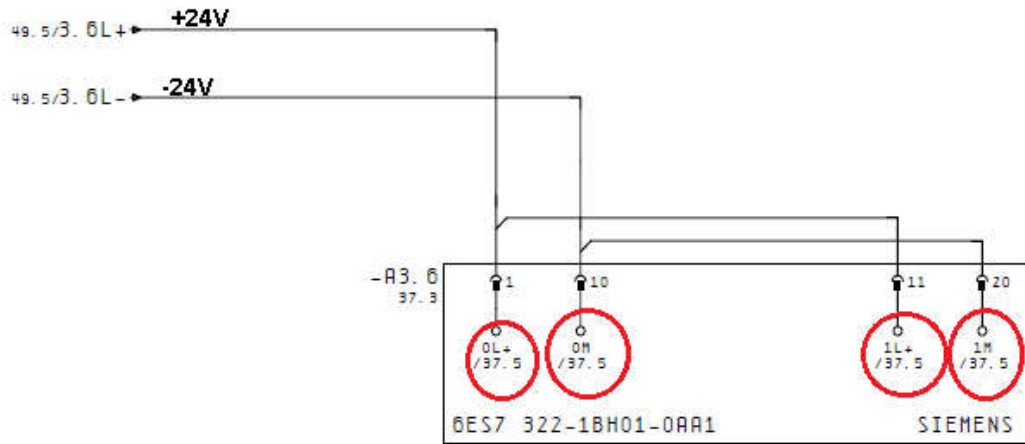
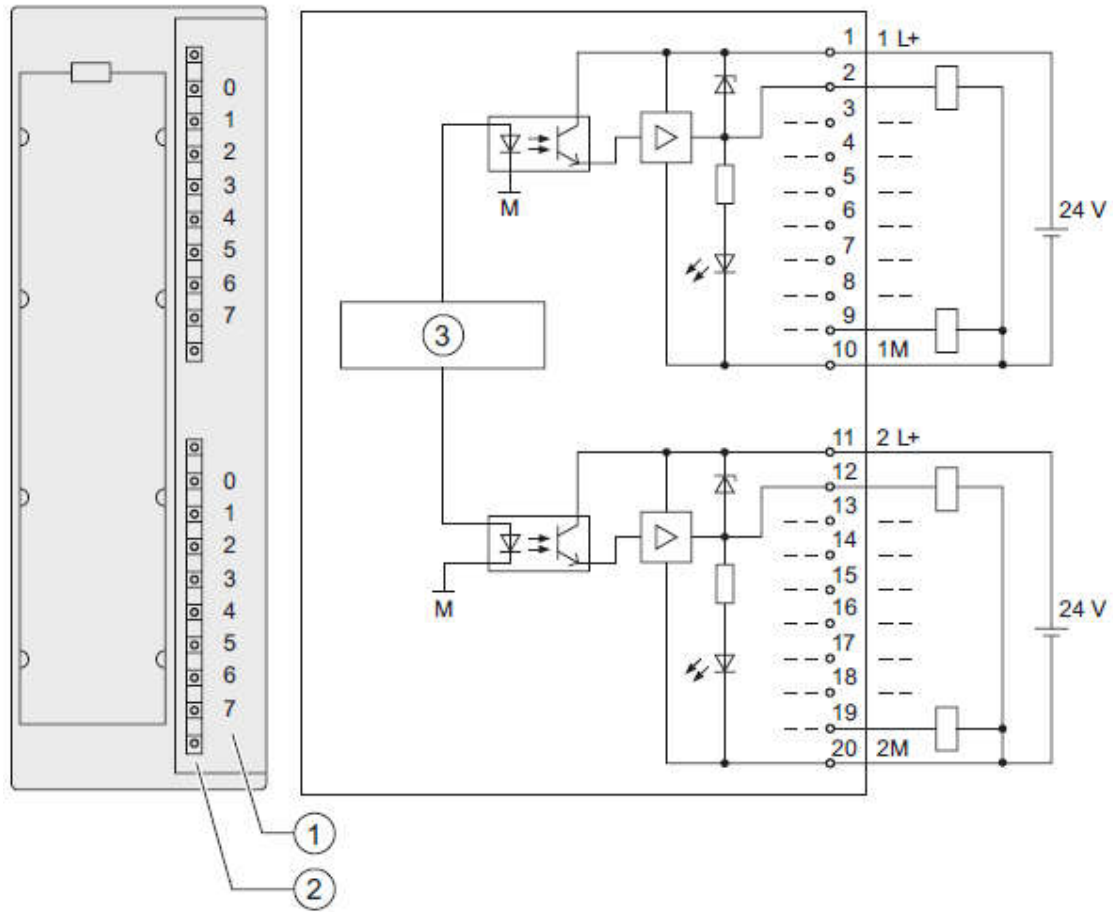
ماژول های *SIPLUS S7-300* را می توان در شرایط محیطی توسعه یافته (*extended environmental*)

مورد استفاده قرار داد

- مناسب برای عملیات در بازه دمایی $C - 25$ تا $C + 60$
- فشار مکانیکی مجاز افزایش یافته (*increased mechanical stress*)

۱۰-۱-۲- سیم بندی ماژول های دیجیتال خروجی (DO)

Digital output module SM 322; DO 16 x DC 24 V/ 0,5 A;(6ES7322-1BH01-0AA0)



DO Power Supply

۱۰-۱-۳- سیم‌بندی ماژول‌های آنالوگ ورودی (AI)

Analog input module SM 331; AI 8 x 13 Bit; (6ES7 331-1KF01-0AB0)

این ماژول برای اندازه‌گیری انواع سیگنال‌های زیر استفاده می‌شود

- Voltage
- Current
- Resistance
- Temperature

Wiring: 2-wire and 4-wire transducers for current measurement

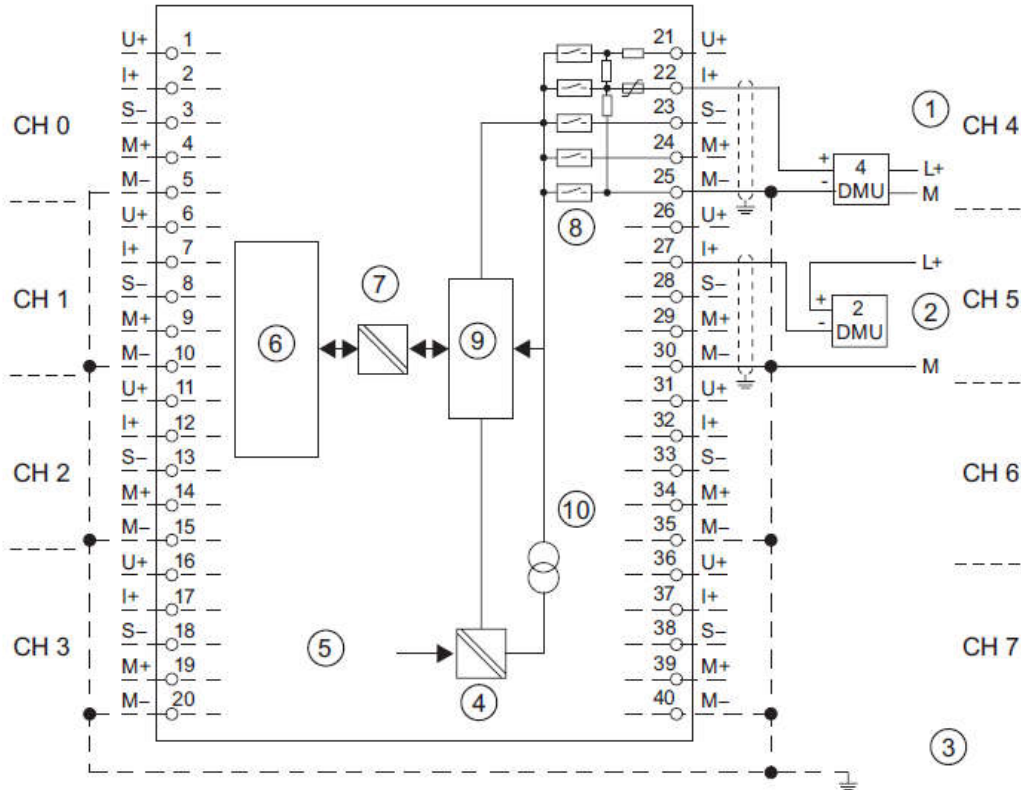
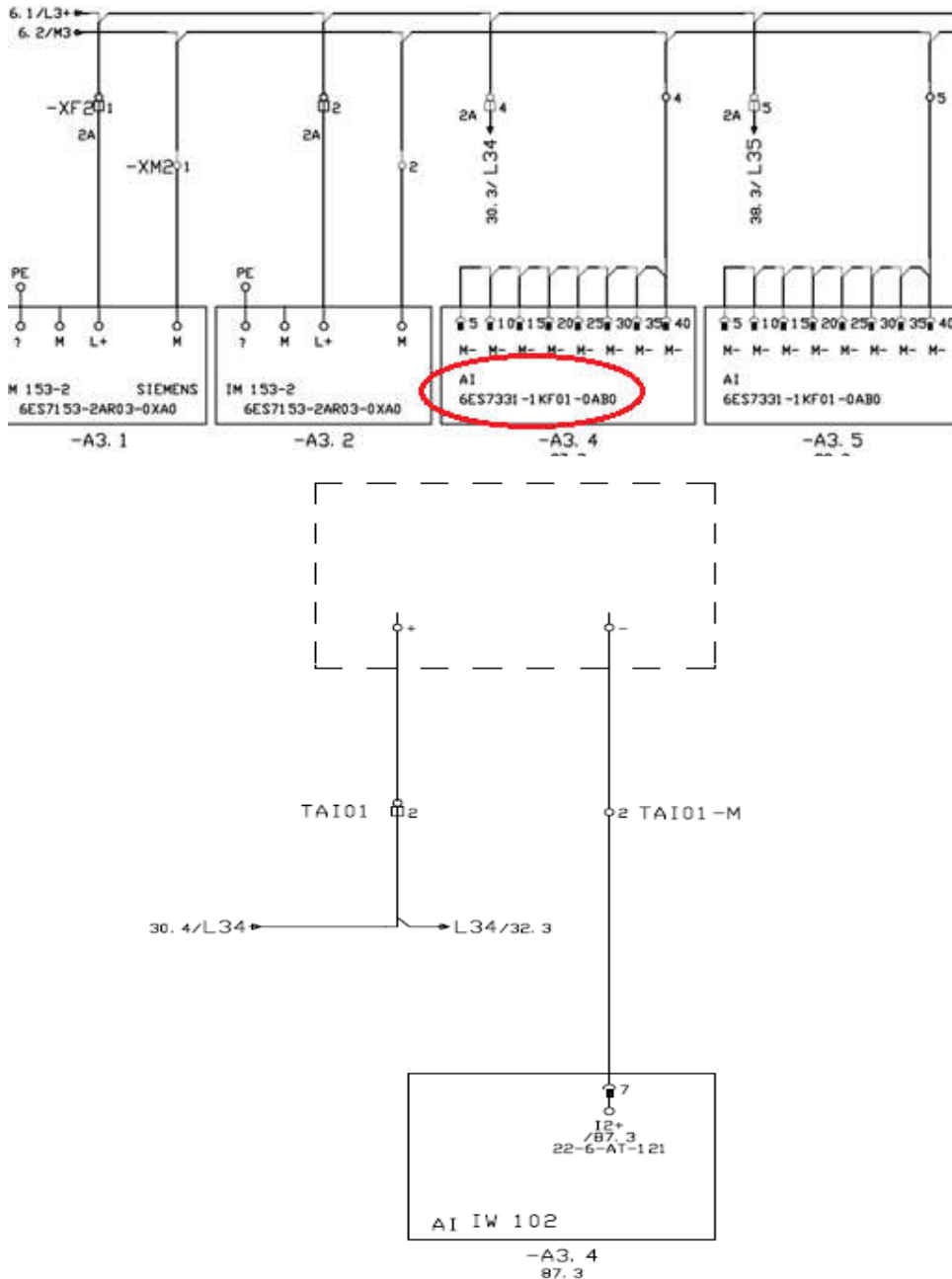


Figure 6-12 Block diagram and wiring diagram

- ① 4-wire transducer (0/4 mA to 20 mA or ± 20 mA)
- ② 2-wire transducer (4 mA to 20 mA)
- ③ Equipotential bonding
- ④ Internal supply
- ⑤ + 5 V from backplane bus
- ⑥ Logic and backplane bus interface
- ⑦ Electrical isolation
- ⑧ Multiplexer
- ⑨ Analog to Digital Converter (ADC)
- ⑩ Current source

برای اتصال سنسور جریان به AI (6ES7 331-1KF01-0AB0) تنها زمین (-) 24 ولت به پایه های M مازول که به هم متصل شده اند وصل می گردد.
 خط 24+ به سر سنور رفته و خط - برگشت از سنسور به ورودی منفی - مازول می رود.



۱۰-۱-۴- سیم بندی ماژول های آنالوگ خروجی (AO)

Analog output module SM 332; AO 8 x 12 Bit; (6ES7332-5HF00-0AB0)

Wiring: Current output

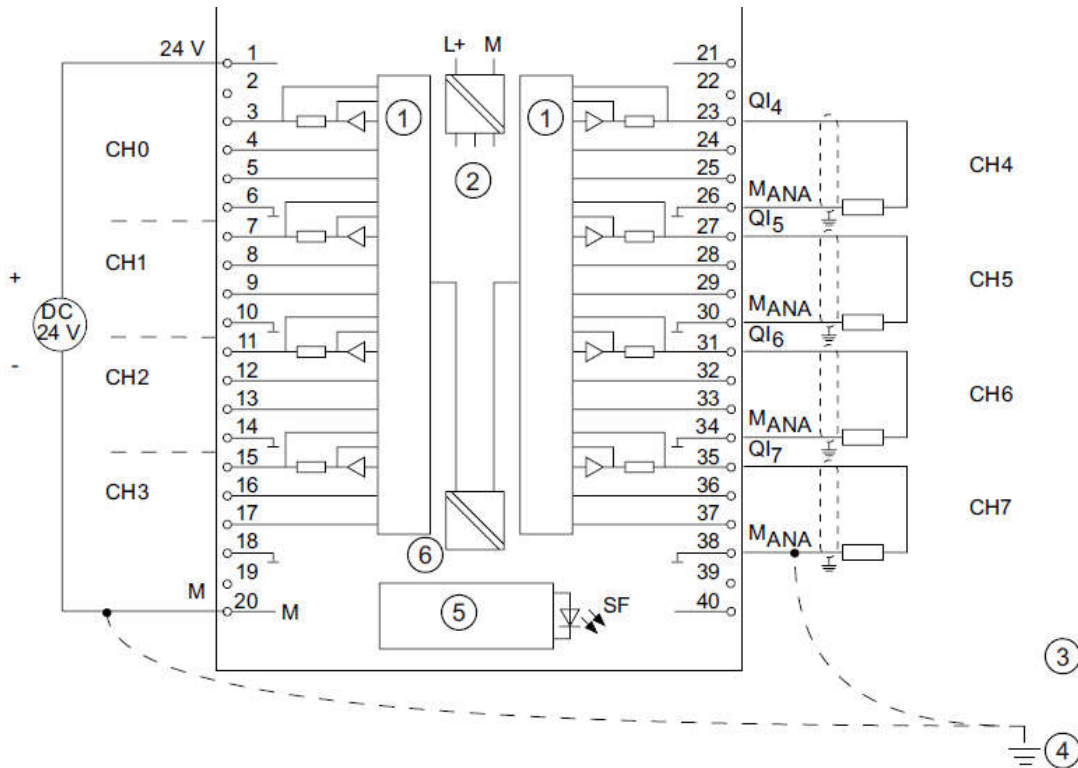


Figure 6-33 Wiring and block diagrams

- | Numeral | Description |
|---------|-------------------------|
| ① | DAC |
| ② | Internal supply |
| ③ | Equipotential bonding |
| ④ | Functional earth |
| ⑤ | Backplane bus interface |
| ⑥ | Electrical isolation |

Chanel	Connection number	Chanel	Connection number
Ch0	3,6 →+,-	Ch4	23,26 →+,-
Ch1	7,10 →+,-	Ch5	27,30 →+,-
Ch2	11,14 →+,-	Ch6	31,34 →+,-
Ch3	15,18 →+,-	Ch7	35,38 →+,-

فصل ۱۱ – ماژول های SIPLUS زیمنس

q- کسی اطلاع داره که PLC های معمولی زیمنس (و نه *siplus*) یا *ABB* چقدر در مقابل خوردگی گازهایی مثل *H2S* مقاومتند؟ مثلاً *S7-400* تا میتونه بطور دایم زیر چه حدی از گاز برحسب *ppb* کار کنه؟ یا تو چه گروهی قرار می گیره؟ (*G1-G4*)

A- تا جایی که اطلاع دارم PLC های زمينس Outdoor نيست و بايد در داخل تابلو كنترلي و در اتاق كنترل با شرايط تعريف شده باشه.

A- البته بنظرم اين مقاومت در برابر خوردگي كه مد نظر شماست هم مربوط به تابلو ميشه نه خود كنترل كننده

A- شما ميتوانيد از سري cpu هاي siplus زمينس استفاده كنيد كه در محيط هاي اسيدى مورد استفاده قرار مي گيرند. كه مي توانند خارج از تابلو هاي كنترلي قرار بگيرند.

A- اما گازهاي خوردنده ميتونند indoor هم باشند كه داخل تابلو، اولين جاي است كه اسيد ميرسونه، كنترلره. بر اساس ميزان خوردگي، شرايط indoor ميتونه G1 تا Gx باشه.

Ambient Climatic Conditions

You can use the S7-400 under the following ambient climatic conditions:

Table 1-10 Ambient Climatic Conditions

Climatic Conditions	Permitted Range	Remark
Temperature	0 to +60 °C	
Temperature change	Max. 10 °C/h	
Relative humidity	Max. 95 % at +25 °C	No condensation, corresponds to RH stressing level 2 in accordance with IEC 61131-2
Atmospheric pressure	1080 to 795 hPa (corresponds to a height of -1000 to 2000 m)	
Concentration of contaminants	SO ₂ : < 0.5 ppm; RH < 60 %, no condensation	Test: 10 ppm; 4 days
	H ₂ S: < 0.1 ppm; RH < 60 %, no condensation	Test: 1 ppm; 4 days

Climatic environmental conditions

The S7-300 may be operated on following environmental conditions:

Environmental conditions	Permissible range	Comments
Temperature: horizontal mounting position: vertical mounting position:	0°C to 60°C 0°C to 40°C	
Relative humidity	10 % to 95 %	No condensation, corresponds to relative humidity (RH) Class 2 to IEC 61131, Part 2
Barometric pressure	1080 hPa to 795 hPa	Corresponds with an altitude of -1000 m to 2000 m
Concentration of pollutants	SO ₂ : < 0.5 ppm; RH < 60 %, no condensation	Test: 10 ppm; 4 days
	H ₂ S: < 0.1 ppm; RH < 60 %, no condensation	Test: 1 ppm; 4 days

SIPLUS extreme Designed for demanding conditions

Industrial applications that have to withstand extreme environmental conditions require special product features to function cost-effectively and reliably. Sewage treatment plant digester towers or desalination plants are typical areas which illustrate the higher requirements that must be met by electronic components and accessories. SIPLUS extreme products are extremely robust and are therefore predestined for use in these difficult environmental conditions:

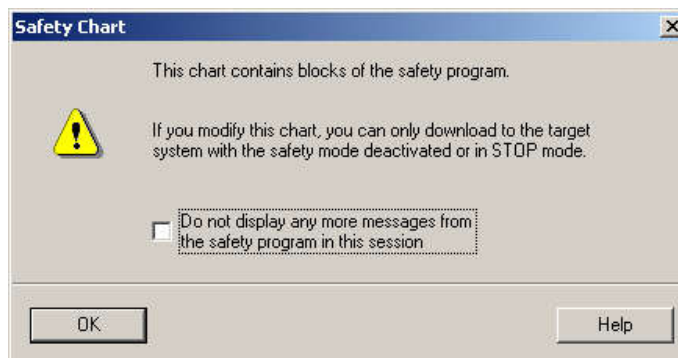
Environmental conditions	SIPLUS extreme		
Temperature range	From -40/-25 to +60/+70 °C ¹⁾		
Relative humidity	100% humidity, condensation and ice formation are permitted		
Pollutant concentrations	EN60721-3-3 3C4 and ISA 571.04 G1, G2, G3, GX		
	Permanent load	Limit value ²⁾	
	H ₂ S	9.9 ppm	49.7 ppm
	Cl	0.2 ppm	1.0 ppm
	HCl	0.66 ppm	3.3 ppm
	At RH < 75%, condensation allowed		
Salt mist	Salt mist test (EN 60068-2-52)		
Mechanically active substances	EN60721-3-3 3S4 including conductive sand/dust ("Arizona dust")		
- Dust (total suspended solids)	4.0 mg/m ³		
- Dust (deposit)	40 mg/(m ² .h)		
Biologically active substances	EN60721-3-3 3B2 mold growth, fungus (excluding fauna)		
¹⁾ For specific product families	²⁾ 30 min/day		

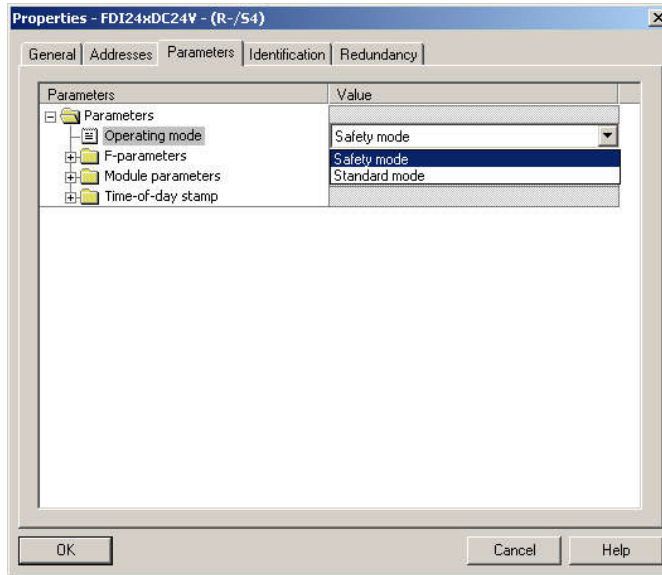
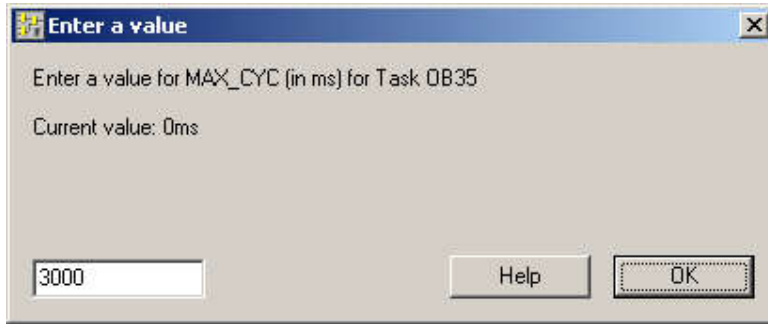
برخی از کارت های ورودی دیجیتال نوع F عبارتند از:

6ES7 321-7TH00-0AB0	Digital input module DI 16xNAMUR, with diagnostics and value status, reconfigurable online
---------------------	--

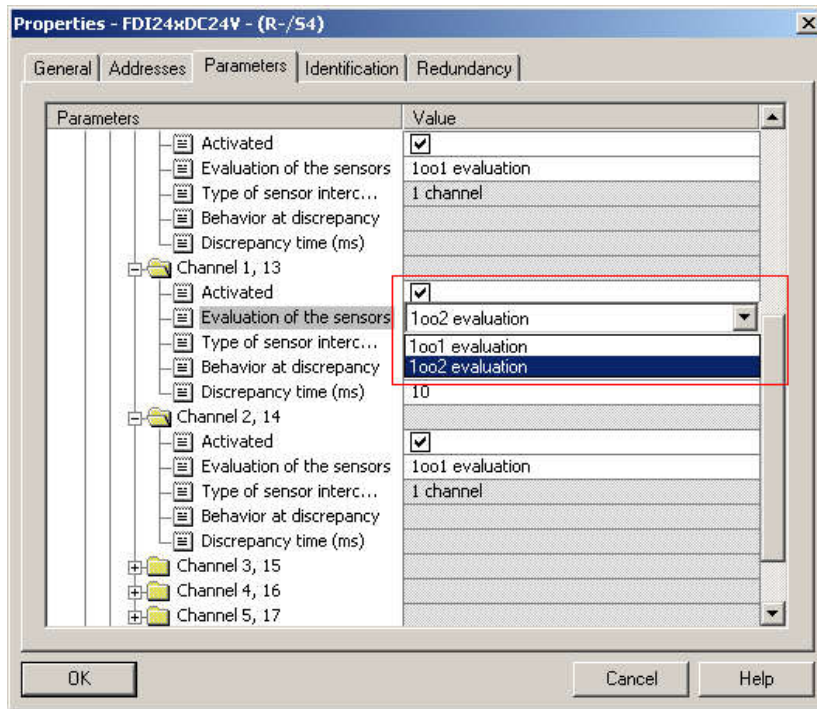
موقع درج کردن بلاک های F به محیط CFC پنجره ظاهر می شود که در آن می توان برای برنامه f پسورد انتخاب کرد

موقع دابل کلیک روی یک سیگنال که به برنامه F مرتبط است پنجره زیر آشکار می شود.

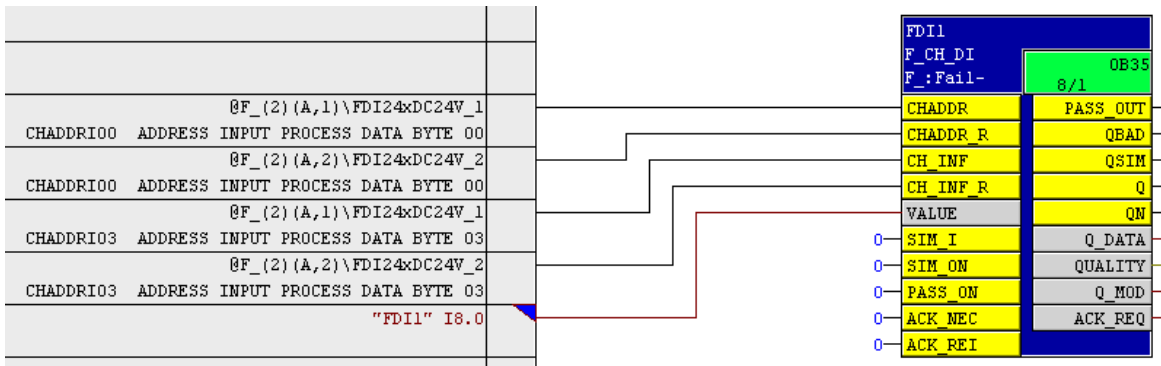




تنظیم کارت FDI برای مد Safety

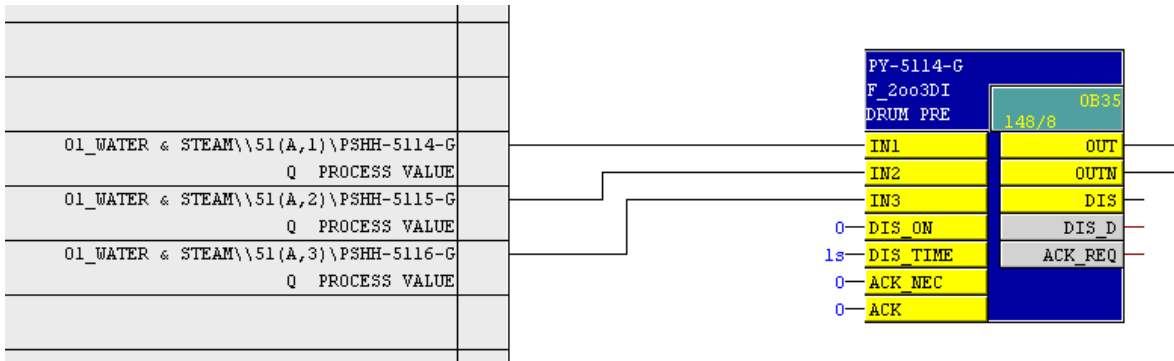


تعیین مد ارزیابی 1001 یا 1002

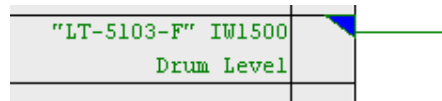


ماژول ارزیابی سه ورودی دیجیتال (F_2003DI)

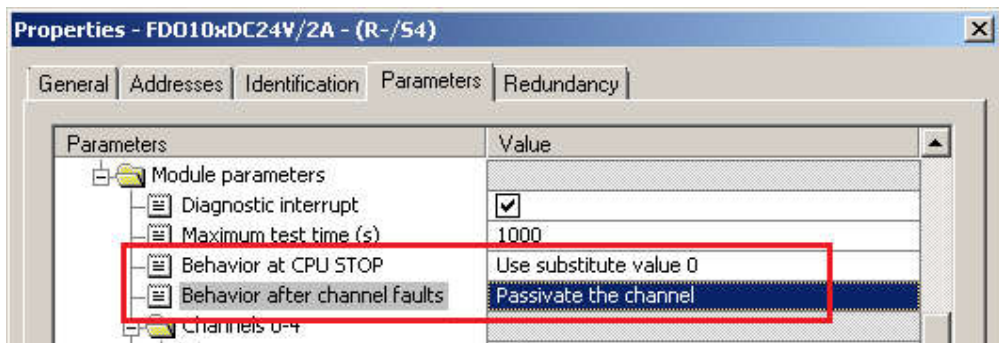
*F_2003DI: 2003 evaluation of inputs of data type BOOL with discrepancy analysis
This block monitors three binary inputs for signal state 1. The OUT output is 1 when at least two inputs INx are 1. Otherwise the output is OUT 0. The OUTN output corresponds to the negated OUT output.*



نکته: آدرس های واقع در کارت های I/O با مثلث آبی رنگ ظاهر می شوند



نکته: در پراپرتی یک بلاک F-DO و F-DI رفتار کانال پس از وقوع فالت به *Passivate the channel* تعیین شده است.

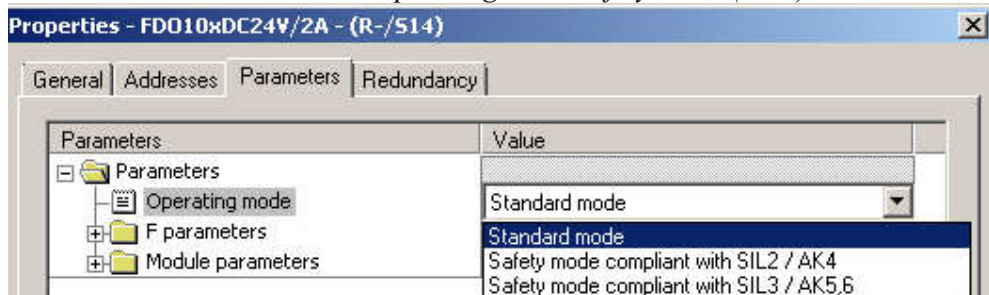


شکل ۱-۱۱-

← مدکاری ماژول های سیگنال fail-safe

برای ماژول های سیگنال fail-safe می توان مد کاری را از طریق پارامتر *Operating mode* در سربرگ *Parameters* پنجره پراپرتی ماژول سیگنال تعیین کرد که در یکی از مدهای استاندارد یا F کار کند. ماژول های زیر مستثنی می باشند.

- 6ES7 326-2BF10-0AB0 → *Operating mode:safety mode (SIL3)*



اطلاعات مربوط به صفحات *Diagnostic* از چارت های @ که به صورت خودکار با کامپایل چارت ایجاد شده اند، گرفته می شود.

۱۱-۱-۱- فعال سازی کانال های IO (Reintegration)

با توجه به اینکه موقع وقوع یک فالت در برنامه یا سخت افزار، مقدار *failsafe* تعریف شده برای کانال های I/O در ورودی/خروجی ها قرار می گیرد. لذا برای فعال سازی مجدد کانال های نیاز به عمل *Depassivation* یا *Reintegration* می باشد.

تعویض از مقادیر قرار گرفته در خروجی های درایور کانال در زمان غیرفعال شدن (fail-safe) به مقادیر فرایند (reintegration of an F-I/O) یا به طور خودکار و یا پس از عمل *user acknowledgment* روی درایور *F-Channel* اجرا می شود.

روش *Reintegration* به موارد زیر بستگی دارد:

- عامل غیرفعال کردن کانال های F-I/O یک F-I/O
- پارامترهای اختصاص داده شده به درایور کانال F

نکته: اگر با فشردن کلید FI بر روی یک بلاک از کتابخانه F، فایل *Help* باز نشود، بایستی برنامه *Html*

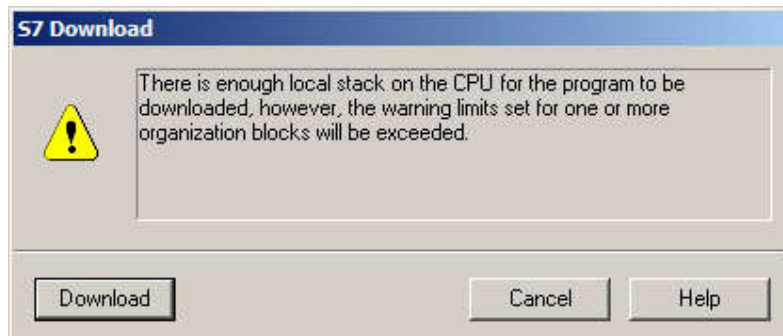
help ویندوز را *Update* کرد. (Windows6.1-KB917607-x64)

8. *max reaction time (max time between change at a physical input and reaction on the physical output).*

فصل ۱۲ - مدیریت خطاها

۱۲-۱-۱- خطا در عبور از مقدار Local data stack

Error: There is enough local stack on the CPU for the program to be downloaded, however, the warning limits set for one or more organization blocks will be exceeded.



here's two ways to solve your problem:

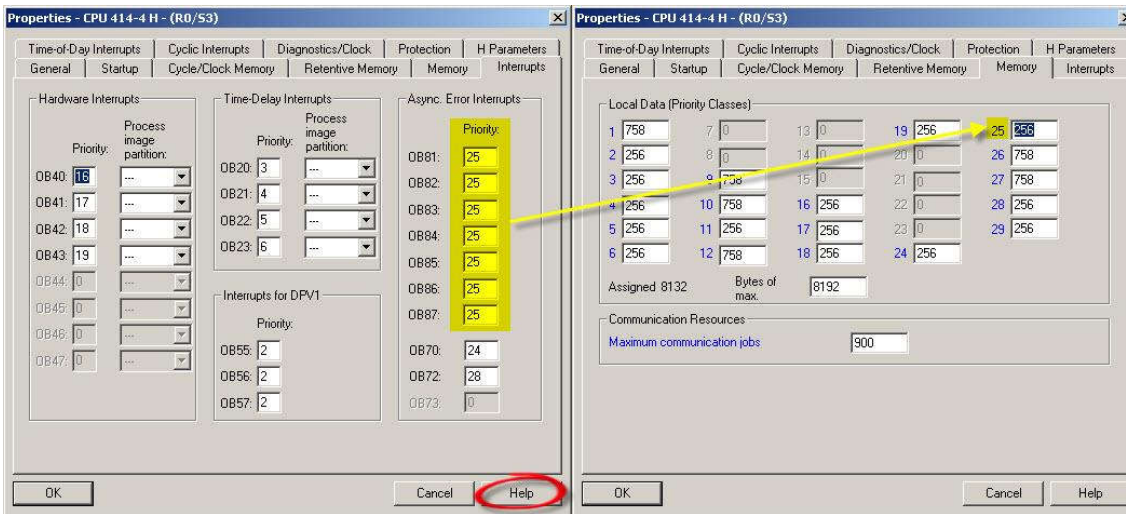
1.) Reduce the number of TEMP variables for the Blocks that are called from the OB's in question to reduce the local data requirements.

2.) Change the amount of local data that is available for the error OB's (this is what I would do).

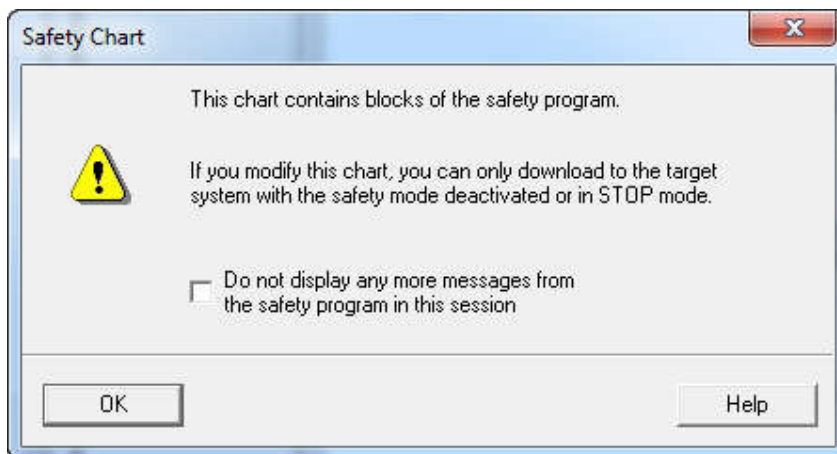
To do so, go into HWconfig double click on your CPU and adjust the Local data size for the priority class to which your OB's belong (better explained in the attached pic).

Please note: Do yourself a favour and use the HELP button on the "Memory" tab and read up on the ramifications of changing the size of the Local data!

I hope this helps



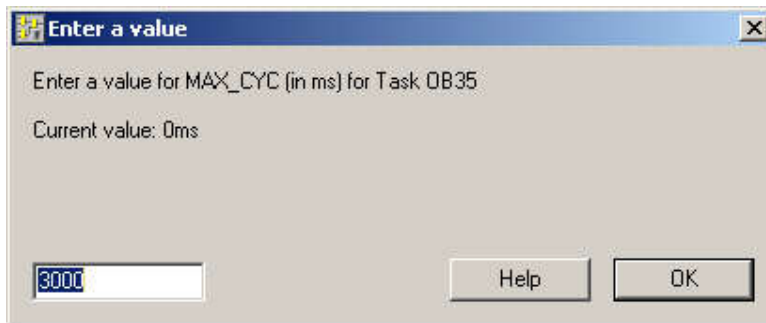
در صورت باز کردن یک چارت حاوی برنامه F پیغام زیر ظاهر می شود. به طوری که به کاربر هشدار می دهد که در صورت تغییر در لاجیک F، بایستی برای دانلود تغییرات به CPU مد Safety در CPU غیرفعال شود.



۱۲-۱-۲- خطای

این خطا مربوط به بلاک های محاسباتی می باشد. به عنوان مثال در یک چارت از یک بلاک *SQRT* استفاده شده بود. این بلاک مربع یک عدد مثبت را محاسبه می کند. ولی از آنجایی که در ورودی این بلاک مطابق شکل زیر یک عدد منفی ظاهر می شد منجر به خطای *s7 safety program : invalid real number in db109* شده و چراغ *SF* ماژول *CPU* روشن می شود.

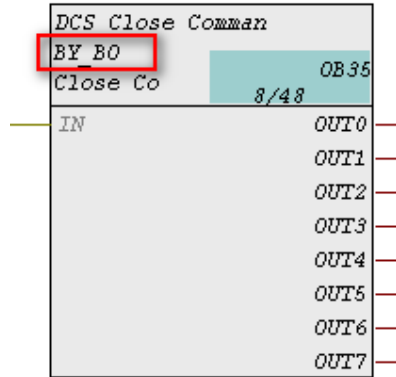
No.	Time of day	Date	Event
12	03:22:57.392 PM	05/20/2016	Safety Program: Invalid REAL Number in DB 109
13	03:22:57.292 PM	05/20/2016	Safety Program: Invalid REAL Number in DB 109
14	03:22:57.192 PM	05/20/2016	Safety Program: Invalid REAL Number in DB 109
15	03:22:57.092 PM	05/20/2016	Safety Program: Invalid REAL Number in DB 109
16	03:22:56.992 PM	05/20/2016	Safety Program: Invalid REAL Number in DB 109
17	03:22:56.892 PM	05/20/2016	Safety Program: Invalid REAL Number in DB 109
18	03:22:56.792 PM	05/20/2016	Safety Program: Invalid REAL Number in DB 109



۱۲-۲- توابع استاندارد

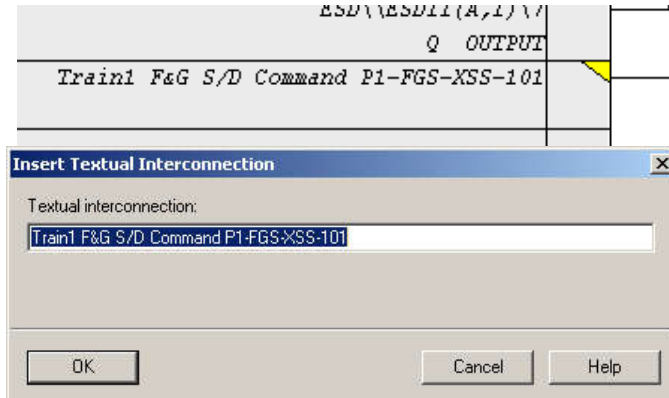
۱۲-۱-۲- فانکشن بلاک *BY_BO*

این تابع یک بایت (*BYTE*) ورودی را به هشت بیت (*BOOL*) جدا می کند.



۱۲-۳- علائم در چارت های CFC

۱. رنگ زرد برای اتصالات از نوع *Textual Interconnection* می باشد.



۲. رنگ سفید برای *Chart I/O* می باشد.

