SIEMENS

常问问题 • 1/ 2015

如何读取 410H 控制器状态灯

How to read the LEDs of 410H controller

http://support.automation.siemens.com/CN/view/zh/109095268

目 录

可靠、耐用的全能系统 AS 410 是 SIMATIC S7- 400 系列一款独特的新产品，特别针对 SIMATIC PCS 7 过程控制系统设计。该系统具有优秀的通用性，适用于所有领域。

CPU 410-5H 过程控制器涵盖了传统 AS 412 至 AS 417 自动化系统的全部应用领域。其自动化性能可根据 SIMATIC PCS 7 过程对象 (PO) 数量灵活调整。

在系统维护时，需要知道 CPU 是否有故障，包括冗余的状态是否完好、IO 卡件是否有故障，如果维护工程师定期打开控制柜巡检，这样比较直观，但是比较费时间且效率低，如果在 OS 画面上显示出 CPU 的指示灯，就可以实时了解 CPU 的工作状态。

1 读取 410－5H故障灯的原理介绍

系统控制器运行过程中，控制器内部的各种不同信息都被保存在 CPU 的内部存储器中，并根据运行情况由控制器内部的操作系统实时进行更新。在 410H 中，这些内部信息也包含了 410H 的状态灯信息。

在系统提供的系统功能 SFC 中，功能 SFC51（RDSYSST）专门用于读取系统的状态信息。该功能块提供的 SSL-ID 功能码输入管脚用于设置需要读取的信息类型，其中功能码 16#0074 可以用于读取控制器的状态灯（包括单 CPU 和冗余 CPU）。

关于 SFC51 的详细帮助请参考 Step7 在向帮助或有个系统提供的系统功能介绍。

参 数	声 明	数据类型	描 述
REQ	INPUT	BOOL	REQ = 1：启动处理
SSL-ID	INPUT	WORD	需要读取的系统状态功能码
INDEX	INPUT	WORD	部分功能码中对象的类型或编号
RET_VAL	OUTPUT	INT	如果执行 SFC 时出错，则 RET_VAL 将包含出错代码
BUSY	OUTPUT	BOOL	TRUE：尚未完成读取
SSL_HEADER	OUTPUT	STRUCT	数据记录信息： LENTHDR：单条数据记录长度 N_DR：DR中记录的数据记录的条数
DR	OUTPUT	ANY	读取的数据记录存储区域： • 如果仅读取了SSL列表的单条信息，则不能评估DR的值，而只能评估SSL_HEADER的值。 • 否则，LENTHDR 和 N_DR 的乘积为已在 DR 中存储的字节数

图 1-1 SFC51 各引脚的说明

也就是说，通过 SFC51（功能码 16#0074）读取的数据记录存储在 DR 中，每条数据记录都拥有上述的结构，每条数据记录通过 CPU_LED_ID 字节 1（LED 标识符）来标识该记录对应的具体 LED，通过评估记录的后两字节，即可判断该 LED 的当前状态。下面是在试验中得到的数据记录：

Address	Name	Type	Initial value	Actual value	Comment
0.0	SZL_Header.LENGTHDR	WORD	W#16#0	W#16#0004	
2.0	SZL_Header.N_DR	WORD	W#16#0	W#16#001C	
4.0	CPU_LEDS[1].Data_record.index	WORD	W#16#0	W#16#F802	
6.0	CPU_LEDS[1].Data_record.led_on	BYTE	B#16#0	B#16#00	
7.0	CPU_LEDS[1].Data_record.led_blink	BYTE	B#16#0	B#16#00	
8.0	CPU_LEDS[2].Data_record.index	WORD	W#16#0	W#16#F803	
10.0	CPU_LEDS[2].Data_record.led_on	BYTE	B#16#0	B#16#00	
11.0	CPU_LEDS[2].Data_record.led_blink	BYTE	B#16#0	B#16#00	
12.0	CPU_LEDS[3].Data_record.index	WORD	W#16#0	W#16#F804	
14.0	CPU_LEDS[3].Data_record.led_on	BYTE	B#16#0	B#16#01	
15.0	CPU_LEDS[3].Data_record.led_blink	BYTE	B#16#0	B#16#00	
16.0	CPU_LEDS[4].Data_record.index	WORD	W#16#0	W#16#F805	
18.0	CPU_LEDS[4].Data_record.led_on	BYTE	B#16#0	B#16#00	
19.0	CPU_LEDS[4].Data_record.led_blink	BYTE	B#16#0	B#16#00	
107.0	CPU_LEDS[26].Data_record.led_blink	BYTE	B#16#0	B#16#00	
108.0	CPU_LEDS[27].Data_record.index	WORD	W#16#0	W#16#F117	
110.0	CPU_LEDS[27].Data_record.led_on	BYTE	B#16#0	B#16#00	
111.0	CPU_LEDS[27].Data_record.led_blink	BYTE	B#16#0	B#16#00	
112.0	CPU_LEDS[28].Data_record.index	WORD	W#16#0	W#16#F118	
114.0	CPU_LEDS[28].Data_record.led_on	BYTE	B#16#0	B#16#00	
115.0	CPU_LEDS[28].Data_record.led_blink	BYTE	B#16#0	B#16#00	
116.0	CPU_LEDS[29].Data_rec		#16#0	W#16#0000	
118.0	CPU_LEDS[29].Data_rec		#16#0	B#16#00	

图 1-2 SFC51 各引脚的说明

在上图里可以看到，第一个红色方框里的数据表示一个灯的状态，其中“INLEX”代表指示灯的类型，INDEX=“W#16#F802”中的“F8”表示 RACK0 CPU 的指示灯，“02”表示是“INTF 内部故障”指示灯，“LED_ON= B#16#0”表示该灯没有亮，“LED_BLINK= B#16#0”表示该指示灯没有闪烁。 与 410－5H 相关 LED 标识符如下所示：

W#16#0002：INTF(内部出错)

W#16#0003：EXTF(外部出错)

W#16#0004：RUN

W#16#0005：STOP

W#16#000B：BUS1F(总线接口 1 故障)

W#16#000D：REDF(冗余出错)

W#16#000E：MSTR(主站)

W#16#000F：RACK0(机架号 0)

W#16#0010：RACK1(机架号 1)

W#16#0011：RACK2(机架号 2)

W#16#0012：IFM1F(接口出错接口模块 1)

W#16#0013：IFM2F(接口出错接口模块 2)

W#16#0015：MAINT(维护请求)

W#16#0017：BUS5F(总线接口 5 故障)

W#16#0018：BUS8F(总线接口 8 故障)

使用 SSL-ID=16#0074 读取冗余 CPU 的状态灯时，SSL_HEADER 数据结构中 N_DR= 16#001C，即每次读取并存储在 DR 中的数据记录条数最大为 28 条。因此，设置接受 DR 的数据区间长度应该至少大于等于：

LENTHDR×N_DR=4×28＝112 bytes

根据上述的 DR 数据记录结构来分析每条数据记录，并将其显示在 OS 上即可。

2 例子程序的使用介绍

由于 CFC 编程语言的特点，要用 SFC51 读取并分析数据记录比较困难，推荐使用结构化编程语言 SCL 来实现，为了方便用户，本文将相关的 SCL 源程序例子和 OS 的面板附在文档后面，对于 SCL 源程序的编程不做介绍。附件里内容介绍如下：

- @SFC51_LED_410H：采用 SCL 编写的读取 LED 的源代码，支持冗余 CPU 和单 CPU 两种类型的控制器，已经在 410-5H V8.1 CPU 测试通过；
- @*.PDL 文件：专门为 PCS7 开发的上位显示图标和面板，需结合上述源代码一起使用，需要将它们拷贝到项目下所有 OS 站的画面目录下：

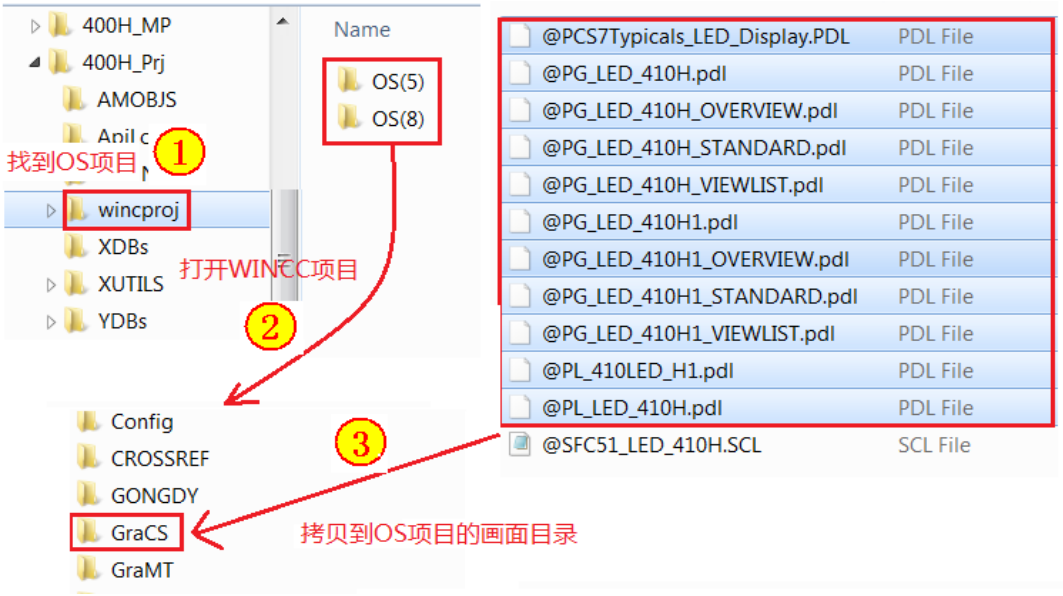

图 2-1 将画面文件拷贝到 OS 项目里

例子程序可以支持 410－5H 控制器做为冗余控制器或者单 CPU 使用，在 OS 编译时自动生成图标和面板，用户下面对例子程序的用法介绍如下：

2.1 导入SCL数据源，生成功能块

在项目的设备视图下，选中 CPU 下的“Sources”文件夹，在右侧的空白处点击鼠标右键，在快捷菜单里依次选择“Insert New Project”→ “External Source”，在打开的窗口里找到解压缩的 SCL 源文件，选择“Open”后，即可成功导入 SCL 源程序。

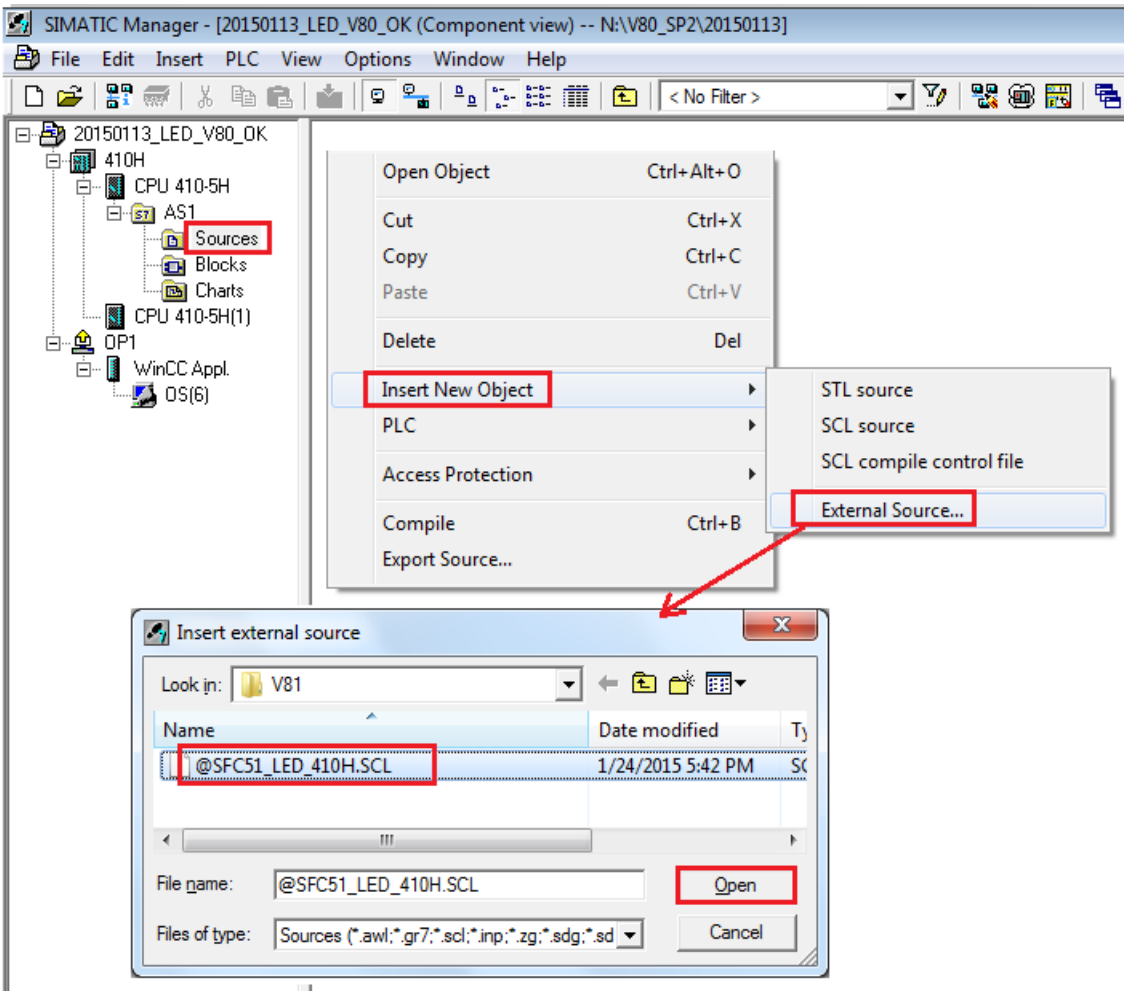

图 2-2 插入 SCL 程序源文件

打开 SCL 源程序后，找到功能块的符号名，在符号表里插入符号表和对应的 FB 块号，本文以 FB501 为例，块号推荐使用 500 以后的数字，不要与项目里已经存在的块号冲突。如下所示：

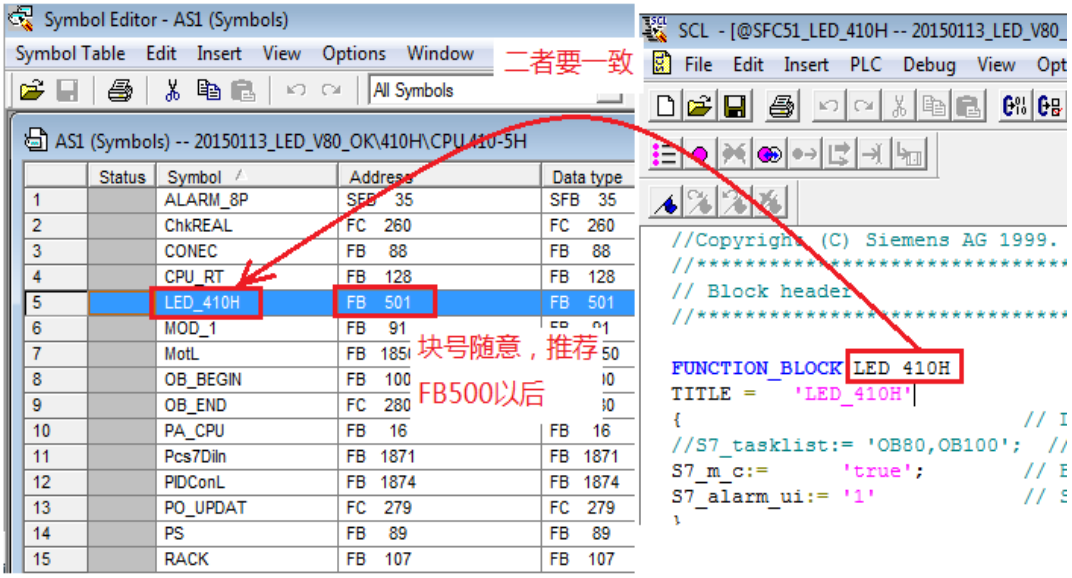

图 2-3 在符号表里定义 LED_410H

接下来，打开导入的 SCL 源文件，鼠标左键点击编译按钮，系统会生成 FB501 功能块。

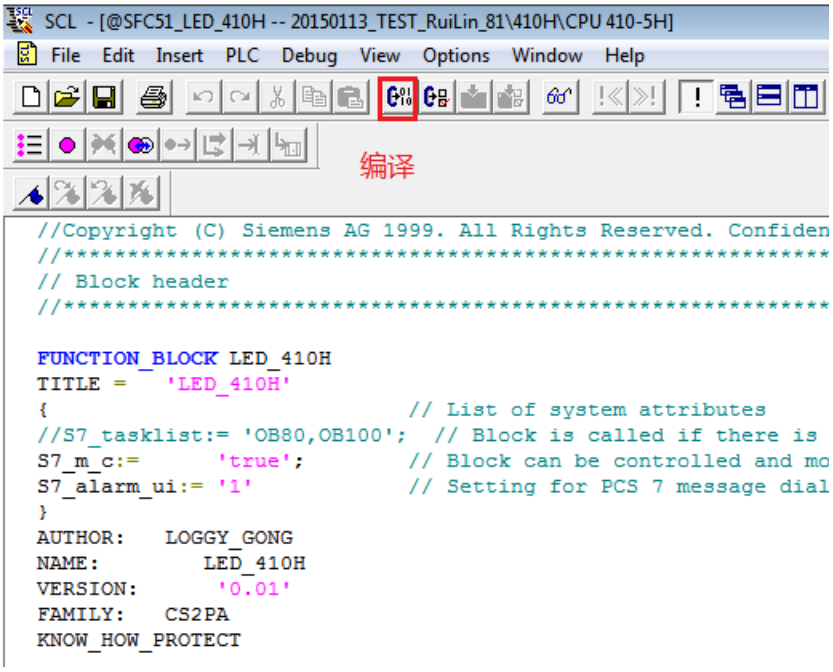

图 2-4 编译 SCL 源生成 FB 块

2.2 组态相关的程序

在工厂视图里找到合适的层级，插入一个新的 CFC，在 Blocks 里找到编译生成的功能块(本例是 FB501)，将它拖入到 CFC 里调用，将块的名字改成一个有意义的名字，这样在画面上生成图标时便于理解。

注意调用这个块的 CFC 要在某一层级下，并且层级下是有画面，否则无法自动生成图标。

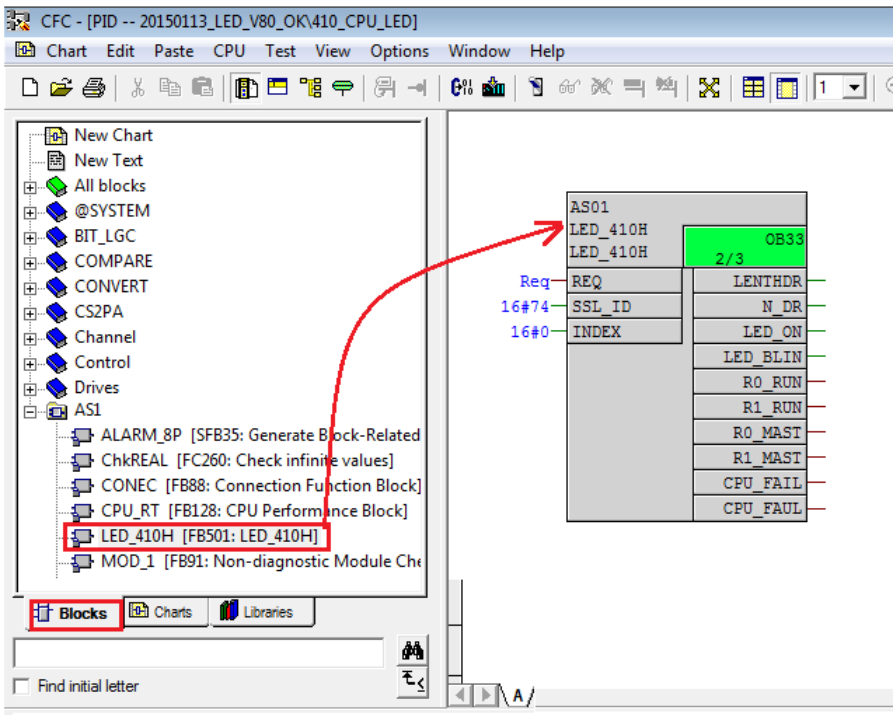

图 2-5 CFC 里调用功能块

注意：由于调用 SFC51 将占用 CPU 系统资源，建议采用较慢循环周期的 OB 来调用该 SFC51，例如 OB32，1 秒钟调用一次或更慢。

接下来，对报警进行相关设定，报警文本可以根据需要自己设定，相关的步骤和解释如下图：

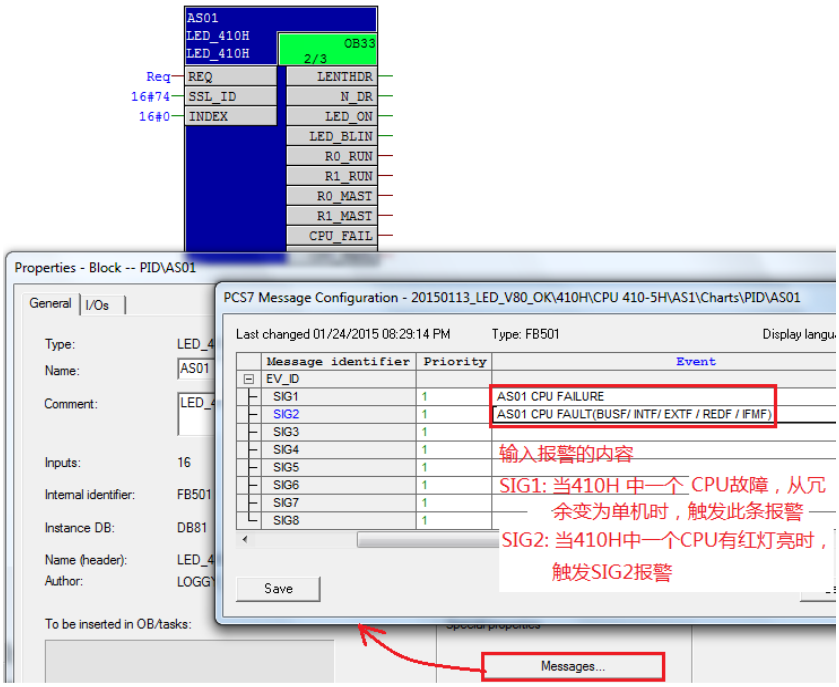

图 2-6 组态块的报警

接下来，对 CFC 进行编译、下载，对 OS 项目进行编译，这时在相关的画面上和下载，运行 OS 项目，最终上位显示效果如下图所示：

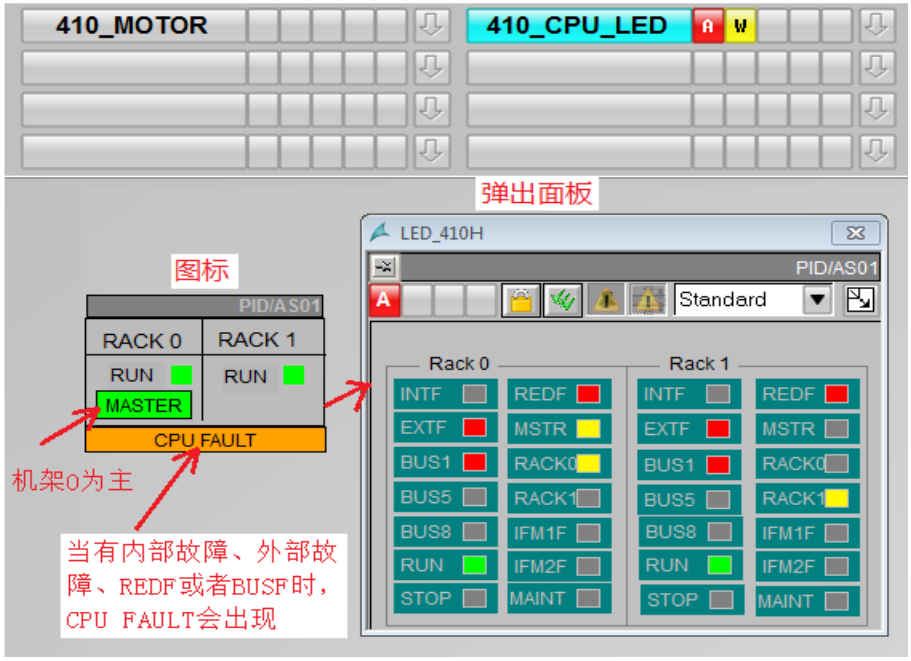

图 2-7 画面运行后的效果

2.3 修改图标和面板

在项目执行过程中，用户可能对图标和面板中对象的颜色有自己的要求，这时需要做一些改动，例如：用户觉得机架的指示灯是黄颜色不合适，可以打开“PG_LED_410H_STANDARD”画面进行修改，如果 PCS7 是中文版，也可以将注释文本改为中文，如下图所示：

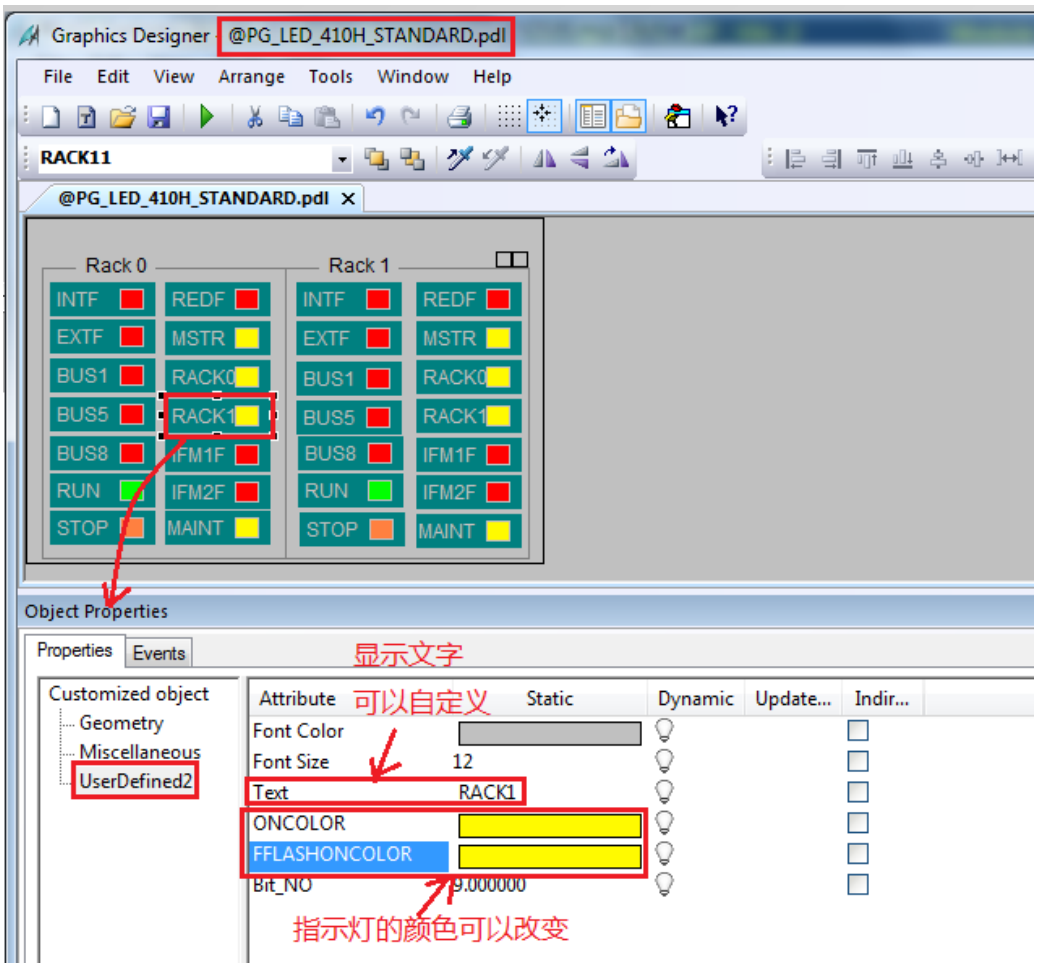

图 2-8 自定义指示灯的文本和颜色

如果还觉得图标里对象的颜色也需要修改，就需要打开@Pcs7typical 可以参考下面的 FAQ 进行自定义：

http://support.automation.siemens.com/CN/view/zh/16514590

如何创建自己的块图标自动的集成到过程画面中？

http://support.automation.siemens.com/CN/view/zh/26697820

如何使用模板画面 "@PCS7Typicals*.pdl" 来创建块图标 (block icon)？

2.4 单 410-5H应用的设置

如果将附件的程序在单 410－5H 的应用时，需要将图标的类型设置为 2，如下图所示：

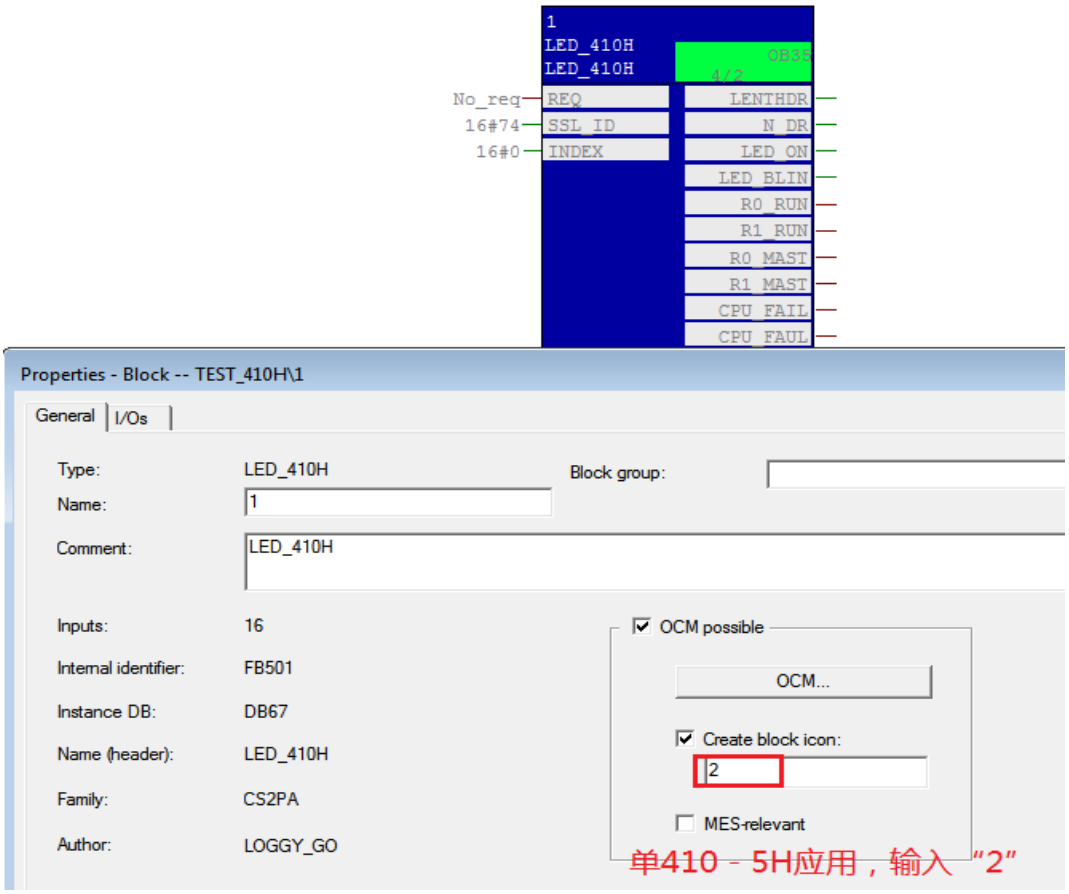

图 2-9 设置图标的类型

另外，还要把 SIG1 的报警取消，因为功能在检测到读回的数据长度只有 14 条时会触发此报警，方法是在 Message class 里把它设置为“no message”。

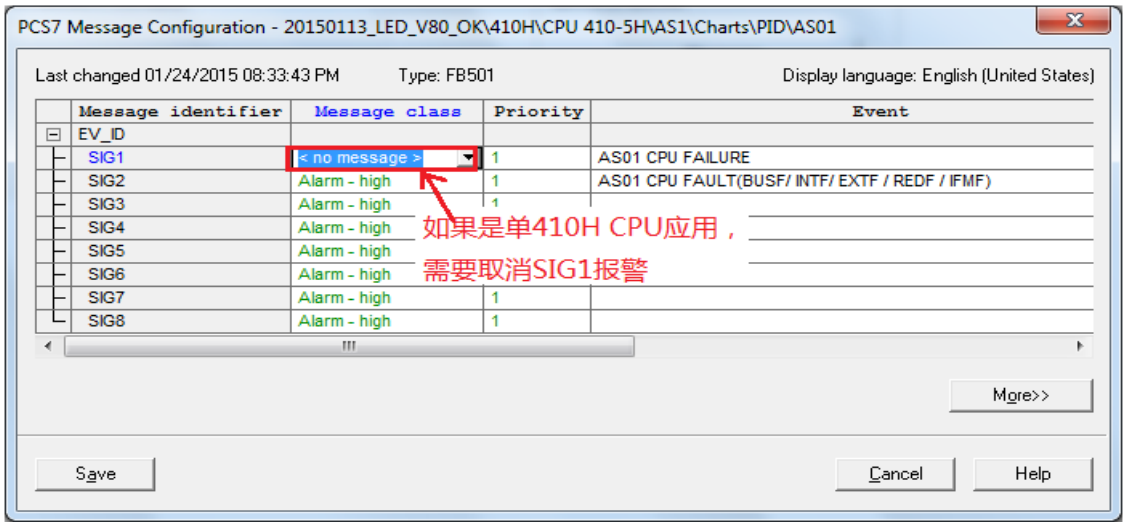

图 2-10 取消单机运行的报警