



原创文章，转载请注明出处。

更多实用资料请登录官网：www.founderchip.com

作者：北岛李工

前面的文章《[西门子SCL编程入门教程连载\(5\)-沿信息检测指令](#)》，有小伙伴留言问：

SCL里使用沿检测必须使用DB块码？和梯形图不能一样么？

今天这篇文章，我来回答这个问题。

SIEMENS
Ingenuity for life

西门子PLC高级编程语言SCL入门教程

第17篇：自己编写沿信号代码

<https://www.founderchip.com>

第一种情况：

如果使用SCL语言指令列表中的**R_TRIG**或者**F_TRIG**指令，在添加这些指令的时候系统会自动添加背景数据块或者多重背景数据块，所以这种情况下必须使用数据块的；

第二种情况：

沿信号检测指令的关键之处在于要记住之前的信号状态，然后将当前的信号状态与之前的进行比较，从而来判断是否是上升沿或者下降沿。

基于这一原理，我们可以自己编写沿信号检测代码：

首先，创建功能块FB_EdgeDetect，声明变量如下：

1. request：用于检测的输入信号；
2. edgeUp：上升沿输出信号；
3. edgeDown：下降沿输出信号；
4. requestMemory：输入信号的记忆值（上一个扫描周期的状态值），注意是静态变量；

FB_EdgeDetect							
	名称	数据类型	默认值	保持	可从 HMI...	从 H...	在 HMI ...
1	▼ Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	request	Bool	false	非保持	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	<新增>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	▼ Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	edgeUp	Bool	false	非保持	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	edgeDown	Bool	false	非保持	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	<新增>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	▼ InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	<新增>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	▼ Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	requestMemory	Bool	false	非保持	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	<新增>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13	▼ Temp				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14	<新增>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	▼ Constant				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16	<新增>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

编写代码如下：

```

test ▶ PLC_1 [CPU 1515-2 PN] ▶ 程序块 ▶ FB_EdgeDetect [FB1]

```

块接口

```

IF... CASE... FOR... WHILE... (*...*) REGION
OF... TO DO... DO...

```

```

1  (*
2  编写自己的沿信号检测代码
3  北岛李工
4  2020-6-22
5  *)
6  //上升沿信号输出
7  #edgeUp := #request AND NOT #requestMemory;
8  //下降沿信号输出
9  #edgeDown := NOT #request AND #requestMemory;
10 //获取输入信号
11 #requestMemory := #request;
12

```

当然，在FB中实现该代码仍然需要背景数据块。所以，我也介绍下如何在FC中实现这个功能。

比如，创建一个功能FC_EdgeDetect，声明变量如下：

SCL_test ▶ PLC_1 [CPU 1515-2 PN] ▶ 程序块 ▶ FC_EdgeDetect [FC3]

FC_EdgeDetect

	名称	数据类型	默认值	监控	注释
1	▼ Input				
2	request	Bool			要检测的输入信号
3	▼ Output				
4	edgeUp	Bool			上升沿
5	edgeDown	Bool			下降沿
6	▼ InOut				
7	requestMemory	Bool			状态记忆值-up
8	▼ Temp				
9	<新增>				
10	▼ Constant				
11	<新增>				
12	▼ Return				
13	FC_EdgeDetect	Void			

代码不用变，如下：

SCL_test ▶ PLC_1 [CPU 1515-2 PN] ▶ 程序块 ▶ FC_EdgeDetect [FC3]

块接口

```

CASE... FOR... WHILE... (*...*) REGION
OF... TO DO... DO...

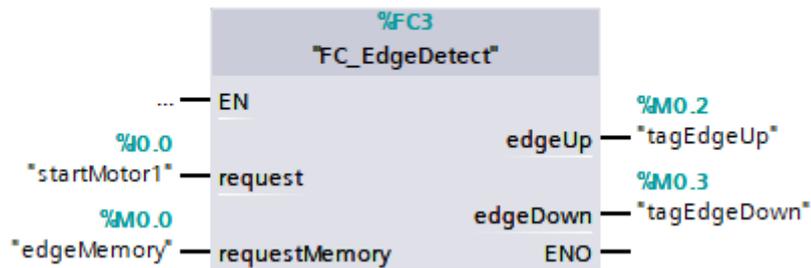
1 (*
2 编写自己的沿信号检测代码
3 北岛李工
4 2020-6-22
5 *)
6 //上升沿信号输出
7 #edgeUp := #request AND NOT #requestMemory;
8 //下降沿信号输出
9 #edgeDown := NOT #request AND #requestMemory;
10 //获取输入信号
11 #requestMemory := #request;
12

```

在OB1中调用FC_EdgeDetect的示例代码如下：

程序段 3：沿信号检测示例

注释



需要注意的是，requestMemory要使用全局变量或者静态变量，不能使用临时变量；

好了，关于自己编写沿信号检测的代码就先介绍到这里，有任何问题欢迎留言。我最近开通了微信视频号，以视频的方式分享一些工控小知识。具有内测权限的小伙伴欢迎扫描下方二维码体验：



扫码关注我的微信视频号