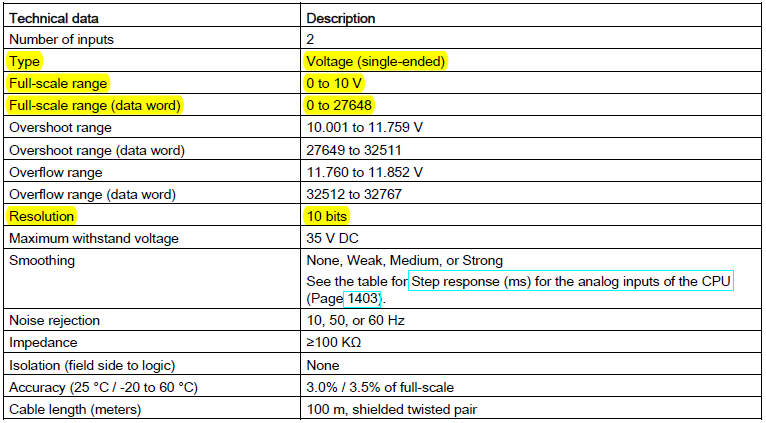
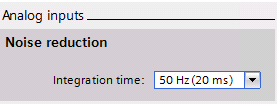
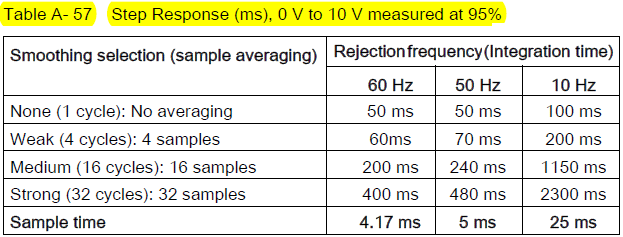
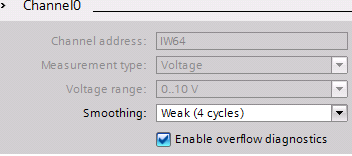
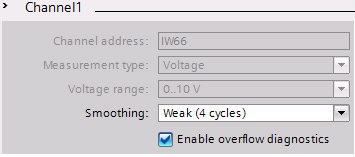
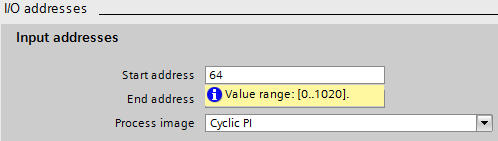
**Analog Girişler**

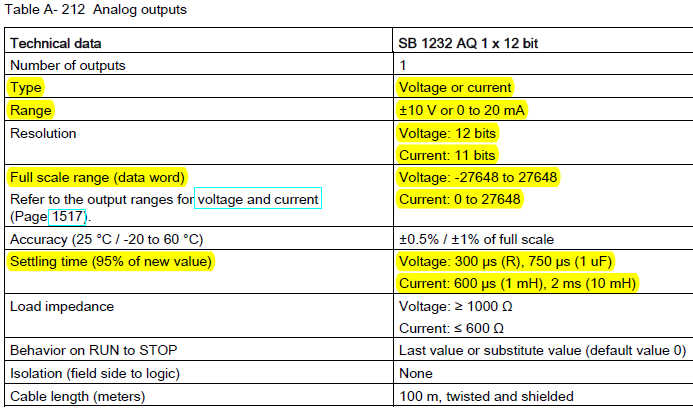


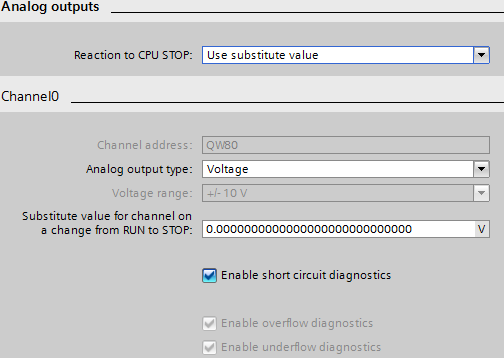
 

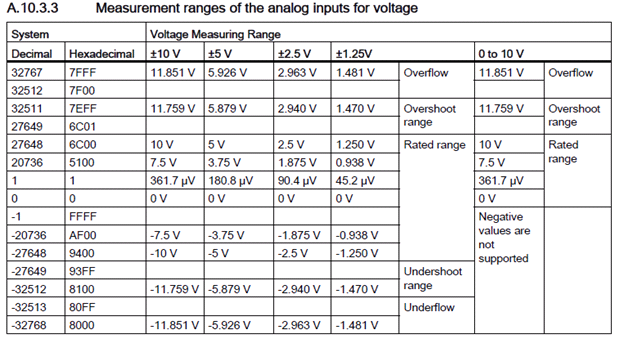
 

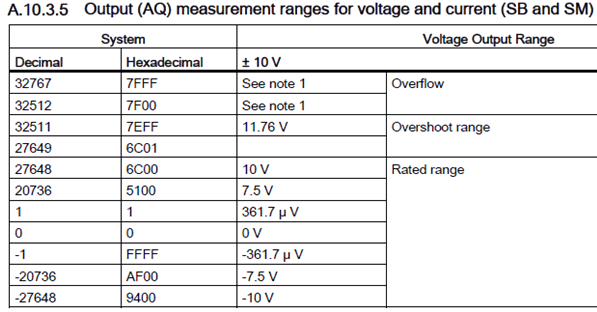


**Analog Çıkışlar**

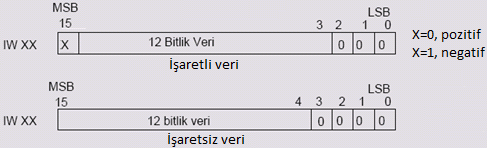






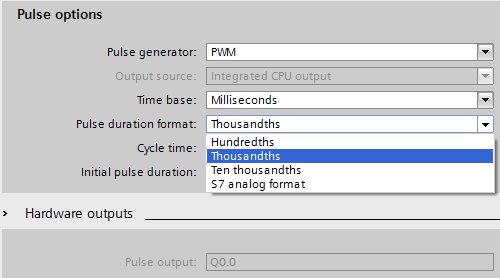
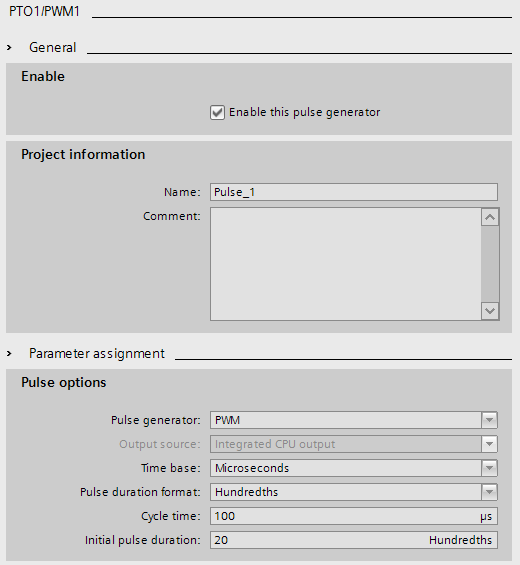


**S7 – Analog Format:**

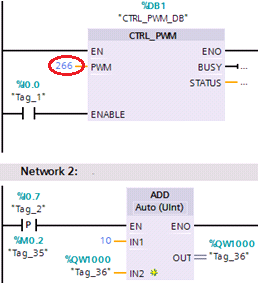


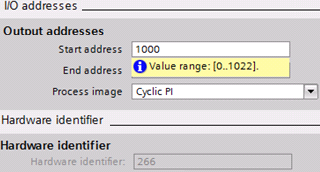
**4 adet PTO / PWM**

PTO (Pulse Train Output); %50 Duty, T değişken, Servo-Step Motor Uygulamaları

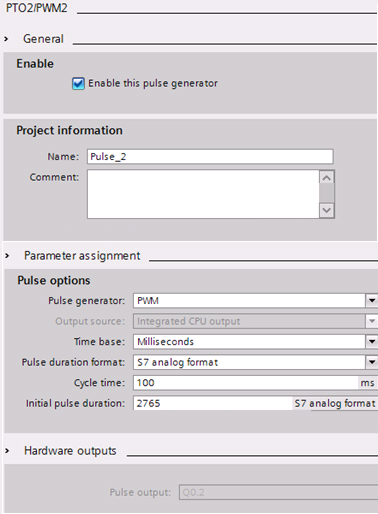
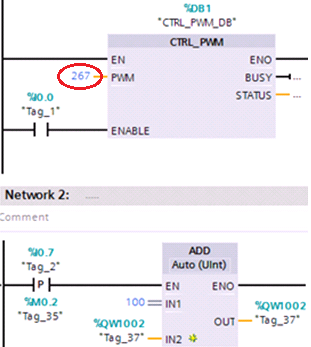


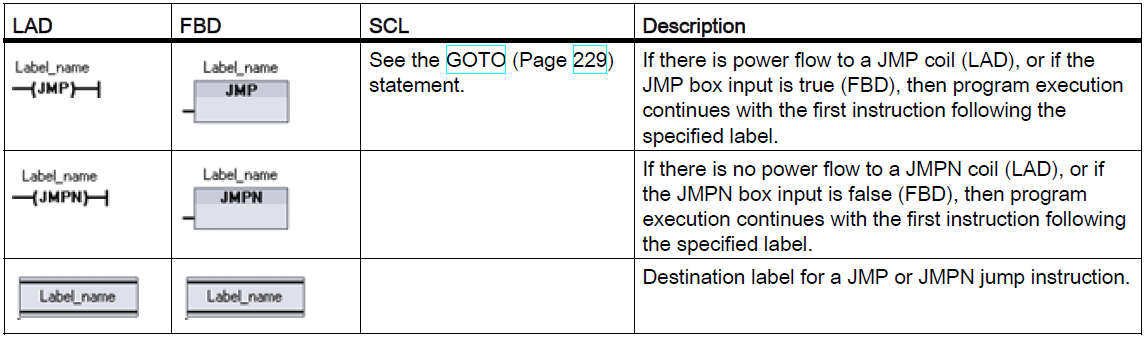






**Donanım ayarları değiştirildiğinde yazılıma ilave olarak donanım da yeniden derlenmelidir.**

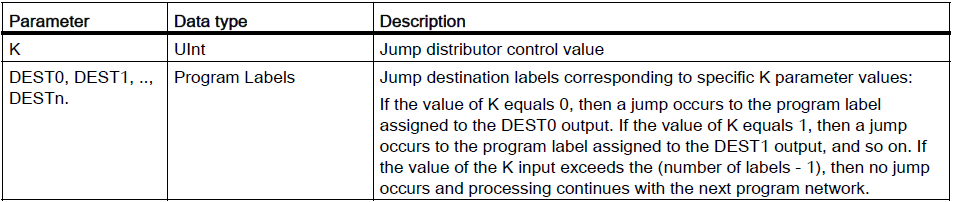
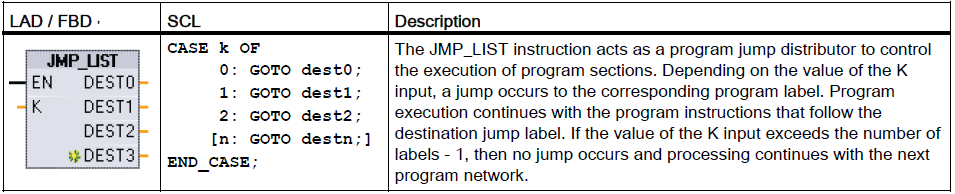
**Program Akış Kontrol Komutları**

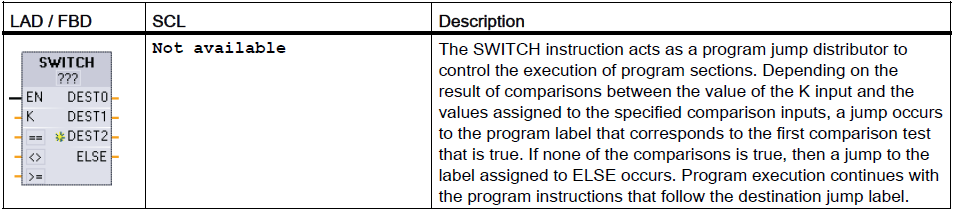
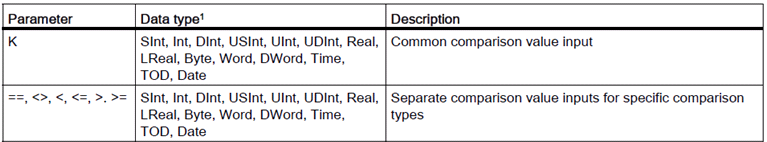
● You can jump forward or backward.

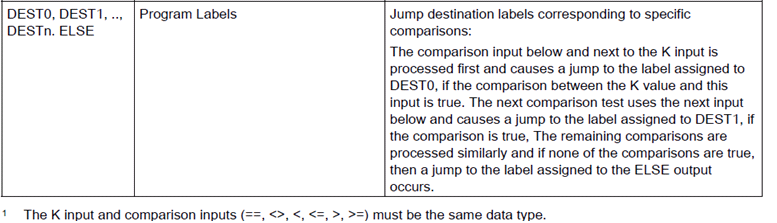
● Each label must be unique within a code block.

● You can jump to the same label from more than one place in the same code block.

● You can jump within a code block, but you cannot jump from one code block to another code block.

**JMP\_LIST instruction**

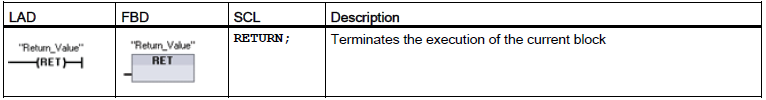
**SWITCH instruction** 



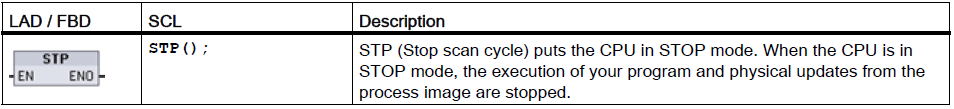
**RET execution control instruction**

The optional RET instruction is used to terminate the execution of the current block. If and only if there is power flow to the RET coil (LAD) or if the RET box input is true (FBD), then program execution of the current block will end at that point and instructions beyond the RET instruction will not be executed. If the current block is an OB, the "Return\_Value" parameter is ignored. If the current block is a FC or FB, the value of the "Return\_Value " parameter is passed back to the calling routine as the ENO value of the called box.

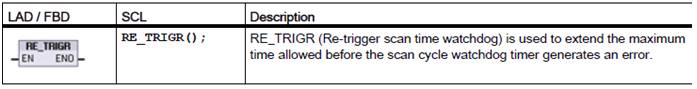
You are not required to use a RET instruction as the last instruction in a block; this is done automatically for you. You can have multiple RET instructions within a single block.



**Stop scan cycle instruction**



**Re-trigger scan cycle watchdog instruction**



Use the RE\_TRIGR instruction to restart the scan cycle monitoring timer during a single scan cycle. This has the effect of extending the allowed maximum scan cycle time by one maximum cycle time period, from the last execution of the RE\_TRIGR function.

**Note**

Prior to S7-1200 CPU firmware version 2.2, RE\_TRIGR was restricted to execution from a program cycle OB and could be used to extend the PLC scan time indefinitely. ENO = FALSE and the watchdog timer is not reset when RE\_TRIGR was executed from a start up OB, an interrupt OB, or an error OB.

For firmware version 2.2 and later, RE\_TRIGR can be executed from any OB (including start up, interrupt, and error OBs). However, the PLC scan can only be extended by a maximum of 10x the configured maximum cycle time.

Setting the PLC maximum cycle time Configure the value for maximum scan cycle time in the Device configuration for "Cycle time".



**Watchdog timeout**

If the maximum scan cycle timer expires before the scan cycle has been completed, an error is generated. If an error handling code block OB 80 is included in the user program, the CPU executes OB 80 where you may add program logic to create a special reaction. If OB 80 is not included, the first timeout condition is ignored and the CPU goes to STOP.

If a second maximum scan time timeout occurs in the same program scan (2 times the maximum cycle time value), an error is triggered that causes the CPU to transition to STOP mode.

In STOP mode, your program execution stops while CPU system communications and system diagnostics continue

**System and clock memory**

You can assign one byte in M memory for system memory. The byte of system memory provides the following four bits that can be referenced by your user program by the following tag names:

– First cycle: (Tag name "FirstScan") bit is set to1 for the duration of the first scan after the startup OB finishes. (After the execution of the first scan, the "first scan" bit is set to 0.)

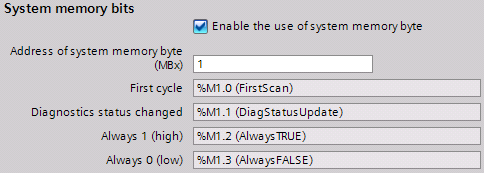
– Diagnostics status changed (Tag name: "DiagStatusUpdate") is set to 1 for one scan after the CPU logs a diagnostic event. Because the CPU does not set the "diagnostic graph changed" bit until the end of the first execution of the program cycle OBs, your user program cannot detect if there has been a diagnostic change either during the execution of the startup OBs or the first execution of the program cycle OBs.

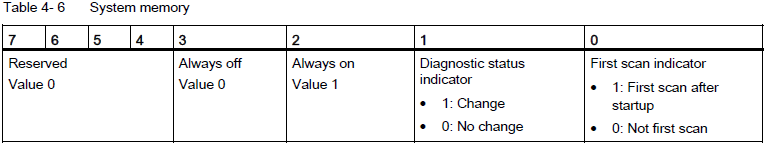
– Always 1 (high): (Tag name "AlwaysTRUE") bit is always set to 1.

– Always 0 (low): (Tag name "AlwaysFALSE") bit is always set to 0.

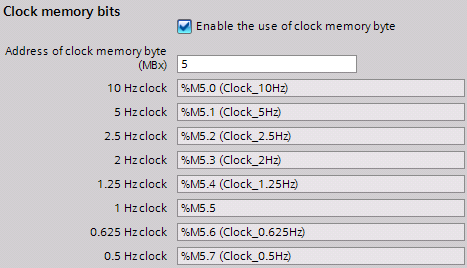
You can assign one byte in M memory for clock memory. Each bit of the byte configured as clock memory generates a square wave pulse. The byte of clock memory provides 8 different frequencies, from 0.5 Hz (slow) to 10 Hz (fast). You can use these bits as control bits, especially when combined with edge instructions, to trigger actions in the user program on a cyclic basis.

The CPU initializes these bytes on the transition from STOP mode to STARTUP mode. The bits of the clock memory change synchronously to the CPU clock throughout the STARTUP and RUN modes.

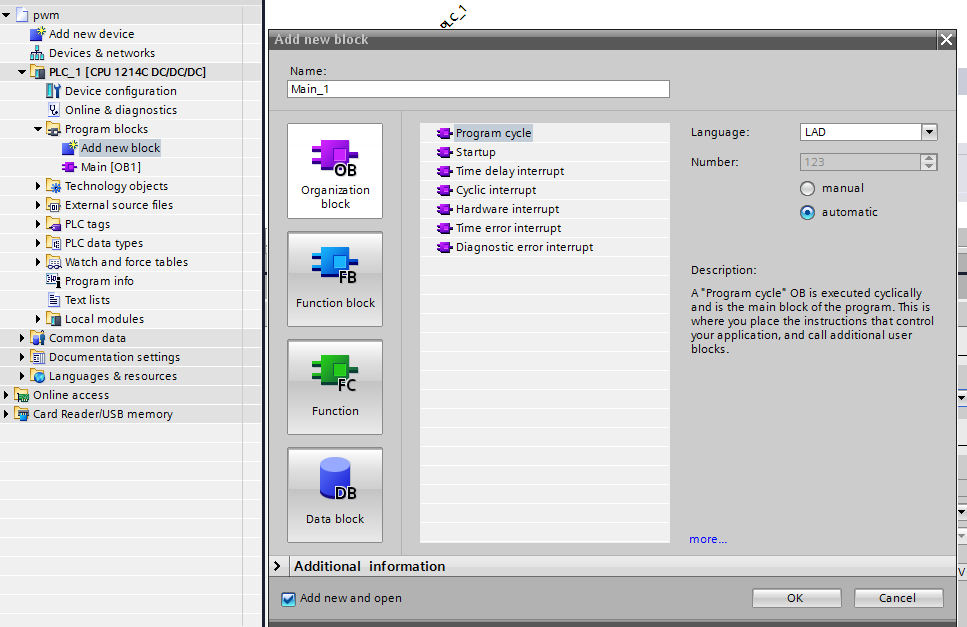


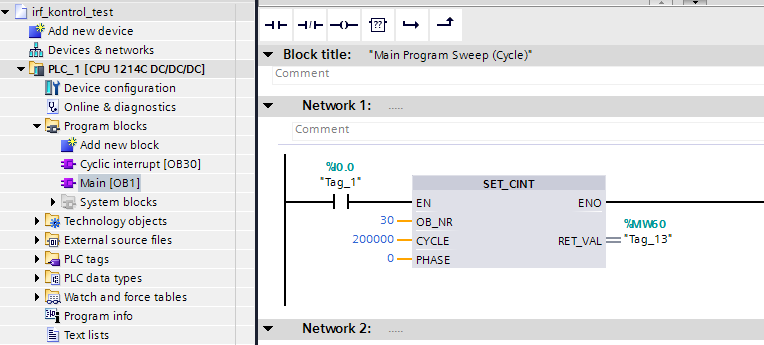


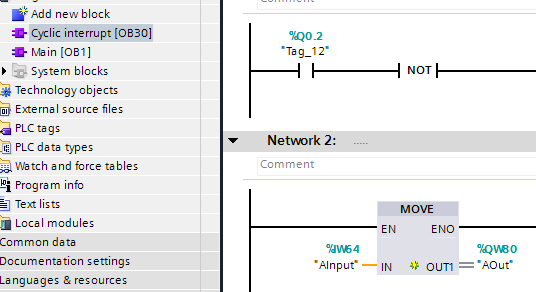
Clock memory configures a byte that cycles the individual bits on and off at fixed intervals. Each clock bit generates a square wave pulse on the corresponding M memory bit. These bits can be used as control bits, especially when combined with edge instructions, to trigger actions in the user code on a cyclic basis.



**Kesmeler (INTERRUPTs)**

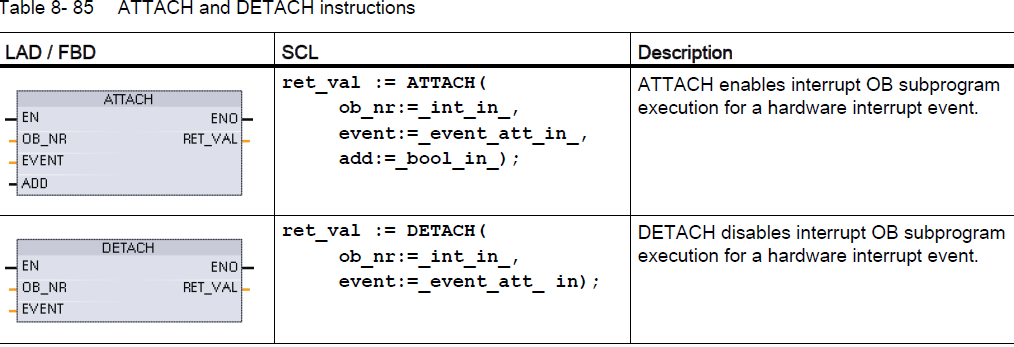


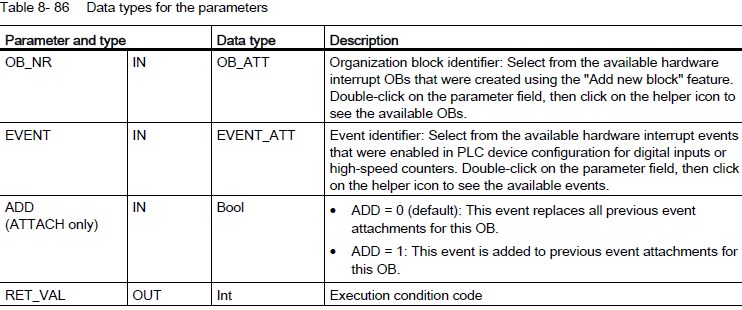




**8.4.1 Attach and detach instructions**

You can activate and deactivate interrupt event-driven subprograms with the ATTACH and DETACH instructions.





Hardware interrupt events

The following hardware interrupt events are supported by the CPU:

● Rising edge events (all built-in CPU digital inputs and SB digital inputs)

– A rising edge occurs when the digital input transitions from OFF to ON as a response to a change in the signal from a field device connected to the input.

● Falling edge events (all built-in CPU digital inputs and SB digital inputs)

– A falling edge occurs when the digital input transitions from ON to OFF.

● High-speed counter (HSC) current value = reference value (CV = RV) events (HSC 1 through 6)

– A CV = RV interrupt for a HSC is generated when the current count transitions from an adjacent value to the value that exactly matches a reference value that was previously established.

● HSC direction changed events (HSC 1 through 6)

– A direction changed event occurs when the HSC is detected to change from increasing to decreasing, or from decreasing to increasing.

● HSC external reset events (HSC 1 through 6)

– Certain HSC modes allow the assignment of a digital input as an external reset that is used to reset the HSC count value to zero. An external reset event occurs for such a HSC, when this input transitions from OFF to ON.

**Enabling hardware interrupt events in the device configuration**

Hardware interrupts must be enabled during the device configuration. You must check the enable-event box in the device configuration for a digital input channel or a HSC, if you want to attach this event during configuration or run time.

Check box options within the PLC device configuration:

● Digital input

– Enable rising edge detection

– Enable falling edge detection

● High-speed counter (HSC)

– Enable this high-speed counter for use

– Generate interrupt for counter value equals reference value count

– Generate interrupt for external reset event

– Generate interrupt for direction change event

**Adding new hardware interrupt OB code blocks to your program**

By default, no OB is attached to an event when the event is first enabled. This is indicated by the "HW interrupt:" device configuration "<not connected>" label. Only hardware-interrupt OBs can be attached to a hardware interrupt event. All existing hardware-interrupt Obs appear in the "HW interrupt:" drop-down list. If no OB is listed, then you must create an OB of type "Hardware interrupt" as follows. Under the project tree "Program blocks" branch:

1. Double-click "Add new block", select "Organization block (OB)" and choose "Hardware interrupt".

2. Optionally, you can rename the OB, select the programming language (LAD or FBD), and select the block number (switch to manual and choose a different block number than that suggested).

3. Edit the OB and add the programmed reaction that you want to execute when the event occurs. You can call FCs and FBs from this OB, to a nesting depth of four.

**OB\_NR parameter**

All existing hardware-interrupt OB names appear in the device configuration "HW interrupt:" drop-down list and in the ATTACH / DETACH parameter OB\_NR drop-list.

**EVENT parameter**

When a hardware interrupt event is enabled, a unique default event name is assigned to this particular event. You can change this event name by editing the "Event name:" edit box, but it must be a unique name. These event names become tag names in the "Constants" tag table, and appear on the EVENT parameter drop-down list for the ATTACH and DETACH instruction boxes. The value of the tag is an internal number used to identify the event.

**General operation**

Each hardware event can be attached to a hardware-interrupt OB which will be queued for execution when the hardware interrupt event occurs. The OB-event attachment can occur at configuration time or at run time.

You have the option to attach or detach an OB to an enabled event at configuration time. To attach an OB to an event at configuration time, you must use the "HW interrupt:" drop-down list (click on the down arrow on the right) and select an OB from the list of available hardware-interrupt OBs. Select the appropriate OB name from this list, or select "<not connected>" to remove the attachment.

You can also attach or detach an enabled hardware interrupt event during run time. Use the ATTACH or DETACH program instructions during run time (multiple times if you wish) to attach or detach an enabled interrupt event to the appropriate OB. If no OB is currently attached (either from a "<not connected>" selection in device configuration, or as a result of executing a DETACH instruction), the enabled hardware interrupt event is ignored.

DETACH operation

Use the DETACH instruction to detach either a particular event or all events from a particular OB. If an EVENT is specified, then only this one event is detached from the specified OB\_NR; any other events currently attached to this OB\_NR will remain attached. If no EVENT is specified, then all events currently attached to OB\_NR will be detached.

8.4.2 Cyclic interrupts

