

Description for hipecs IRQ functions

This is the description for the frenzel + berg electronic demo application for using interrupts at the hipecs PLC or CORE10 modules.

Introduction

For this demo application you need at least a hipecs core module or a hipecs PLC. In order to trigger an interrupt, you can either use an external signal or you can connect an output (DOUTB0.2) of the hipecs with a configured IRQ input (DINB1.2) and click a button in the visualization to toggle the output pin.

Preparation

There are only few preparations necessary to use this project.

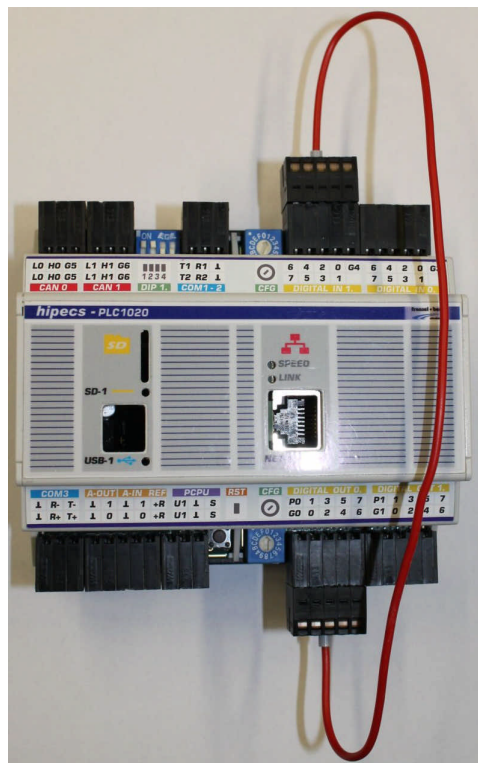
hipecs setup

The "app_hipecsplc_comtest_v1r1_en.prj" file is the CODESYS PLC program for this example. The communication parameters are COM15 and 57600 Baud. **This must be changed to your specific PC settings!**

When using a frenzel + berg Visu-P Panel, you have to use the COM3 (RS422) interface for the SGI connection to the panel. Baudrate is 460.000 Baud. Check Visu-Panel DIP switches, too.

It is also possible to use the CODESYS internal visualization.

In order to trigger the IRQ by clicking the button in the CODESYS visualization, you have to connect the digital output byte 0.2 with the digital input byte 1.2. It is also possible to use an external signal.



Description for hipecs IRQ functions

Starting and using the demo project

It is recommended to use the target visualisation to operate the demo application. In the "VISU_PRG" visualization, there are several grey buttons. These buttons trigger the corresponding function once. With the blue buttons, you can change parameters or variables.

The different CoDeSys POU's and functions

PLC_PRG

The programm is running in this POU. By clicking the corresponding button in the VISU, the variable IRQ_Step is set to a specific value. The function will then be called and the variable is set back to 0.

call_me_from_interrupt

This is the POU triggered by the interrupt. A variable is incremented to check if the program was executed.

Attention! Functions or POU's triggered by an interrupt should be as short as possible.

To use an IRQ:

1. Register Service
2. Then enable interrupt

To delete an IRQ:

1. Disable interrupt first
2. Then delete service

It is possible to set a priority for the IRQs. It is emphasized, that there must be **one IRQ per priority level only**. It is not possible to set the same priority for different IRQ channels.

IrqNr in register function	digital input of hipecs hardware
2	DINB1.2
3	DINB 1.3
4	DINB 1.4
5	DINB 1.5
6	DINB 1.6
7	DINB 1.7

Visu description on page 3.

Description for hipecs IRQ functions

Visu

IRQ Demo

register Service @ Input 1.2 for POU "call_me_from_interrupt"

priority: 22 priority + priority -

Edge: both rising falling both

execute function successful registered

enable Service @ Input 1.2 execute function enabled

Trigger interrupt with OUT0.2

Trigger interrupt by setting flag

disable Service @ Input 1.2 execute function

delete Service @ Input 1.2 execute function

Interrupt counter:

4

Button	Function
	register service (must be done first)
priority + / priority -	increments or decrements the priority of the IRQ from 0...32 where 0 is lowest and 32 is highest priority
rising / falling / both	sets the edge detection of the IRQ
execute function	registers the IRQ service in the operating system with the previously chosen parameters. You have to use digital input byte 1.2 !!
-> SysInterrupt_RegService()	
	enable service @ input (only after successful registration)
execute function	calls function: SysInterrupt_Enable(). IRQ can now be used.
Trigger interrupt with OUT0.2	sets output byte 0.2, which should be connected with input byte 1.2. So an interrupt should be triggered.
Trigger interrupt by setting flag	set the interrupt request flag in the operating system. An interrupt will be triggered.
disable service	calls function: SysInterrupt_Disable()
delete service	calls function: SysInterrupt_DeleteService(). After that, the inputs works as regular input again.