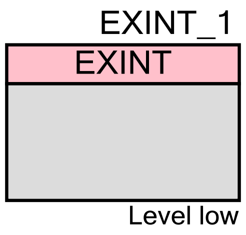
External Interrupt (PDL_EXINT)

1.0



Features

- Up to 32 external interrupt and one NMI input pin mounted
- Five types of detection condition
- Possible to use an EXINT component to return from standby mode
- Possible to enable/disable NMI interrupt in TYPE3-M0+ products

General Description

The Peripheral Driver Library (PDL) External Interrupt (PDL_EXINT) component control section outputs an external interrupt request to the interrupt controller in the following procedure:

1. The signal input to pin detects the edge or level specified in the External Interrupt Level Register. The edge or level to be detected can be selected from the following five types:
   - High level
   - Low level
   - Rising edge
   - Falling edge
   - Both rising and falling edge
2. The detected interrupt input is held in the interrupt factor F/F. It is read with the External Interrupt Factor Register.
3. If an external interrupt is enabled with the External Interrupt Enable Register, an external interrupt request to the interrupt controller is asserted.
4. The held interrupt factor is cleared with the External Interrupt Factor Clear Register, an external interrupt request to the interrupt controller is negated.

This component uses firmware drivers from the PDL_EXINT module, which is automatically added to your project after a successful build.

When to Use a PDL_EXINT Component

Use the PDL_EXINT component when you need to trigger the external signals in your firmware.

Quick Start

1. Drag a PDL_EXINT component from the Component Catalog FMx/Ports and Pins folder onto your schematic. The placed instance takes the name EXINT_1.
2. Double-click to open the component's Configure dialog.
3. On the **Basic** tab, set the following parameters:
   - select interrupt trigger condition
   - specify the callback function
4. You can use PDL_EXINT to configure NMI interrupt by setting NMI parameter in the component's Configure dialog.
5. Assign the pin in your device using the Pin Editor. If you are using a pin to read a SW button state on a development kit, refer the kit User Guide for suitable pin assignments.
6. Build the project to verify the correctness of your design. This will add the required PDL modules to the Workspace Explorer and generate configuration data for the EXINT_1 instance.
7. In the *main.c* file, initialize the peripheral and start the application.

```
stc_exint_config config;
config.abEnable[EINT_1_Index] = 1u;
config.aenLevel[EINT_1_Index] = EXINT_1_Level;
config.apfnExintCallback[EINT_1_Index] = EINT_1_ExintCallback;
config.bTouchNvic = 1;

Exint_Init(&config);
EXINT_1_SetPinFunc_INT();
```

8. Build and program the device.

Component Parameters

The PDL_EXINT component Configure dialog allows you to edit the configuration parameters for the component instance.

Basic Tab

This tab contains the component parameters used in the basic peripheral initialization settings.

Parameter Name	Data Structure Type	Description
enLevel	en_exint_level_t	Interrupt trigger condition.
NMI	stc_mfs_uart_config	Use NMI pin for EXINT.
pfnExintCallback	bool	Callback routine for external interrupt. Note: this generates a declaration only - USER must implement the function

Component Usage

After a successful build, firmware drivers from the PDL_EXINT module are added to your project in the pdl/drivers/exint folder. Pass the generated data structures to the associated PDL functions in your application initialization code to configure the peripheral.

Generated Data

The PDL_EXINT component populates the following peripheral initialization data structure(s). The generated code is placed in C source and header files that are named after the instance of the component (e.g. *EXINT_1_config.c*). Each variable is also prefixed with the instance name of the component.

Once the component is initialized, the application code should use the peripheral functions provided in the referenced PDL files. Refer to the PDL documentation for the list of provided API functions. To access this document, right-click on the component symbol on the schematic and choose "**Open API Documentation…**" option in the drop-down menu.

Preprocessor Macros

The PDL_EXINT component generates the following preprocessor macro(s). Note that each macro is prefixed with the instance name of the component (e.g. "EXINT_1").

Macro	Description
EXINT_1_SetPinFunc_NMIX()	Pin function macros. Visible when non masked interrupt mode is selected.
EXINT_1_SetPinFunc_INT()	Pin function macros. Visible when non masked interrupt mode isn't selected.
EXINT_1 _Index	Number of the EXINT line
EXINT_1 _Level	Indicate detection mode

Data in RAM

The generated data may be placed in flash memory (const) or RAM. The former is the more memory-efficient choice if you do not wish to modify the configuration data at run-time. Under the **Built-In** tab of the Configure dialog set the parameter CONST_CONFIG to make your selection. The default option is to place the data in flash.

Interrupt Support

If the PDL_EXINT component is specified to trigger interrupts. These can then be passed to the associated interrupt initialization functions along with your ISR callback function.

Code Examples and Application Notes

There are numerous code examples that include schematics and example code available online at the Cypress Code Examples web page.

Cypress also provides a number of application notes describing how FMx devices can be integrated into your design. You can access the Cypress Application Notes search web page at www.cypress.com/appnotes.

Resources

The PDL_EXINT component uses the EXINT (External Interrupt) peripheral block.

References

- FM0+ Family of 32-bit ARM® Cortex®-M0+ Microcontrollers Peripheral Manuals
- Cypress FM0+ Family of 32-bit ARM® Cortex®-M0+ Microcontrollers

Component Errata

This section lists known problems with the component.

Cypress ID	Component Version	Problem	Workaround
252934	1.0	EXINT is not supported on S6E1A devices. There is a defect in the EXINT interrupt handler that prevents its use on S6E1A devices. The defect does not impact S6E1B or S6E1C devices.	None. Contact Cypress technical support (http://www.cypress.com/mycases) for possible firmware updates and help with implementing external interrupts on S6E1A devices.

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
1.0	Initial Version	