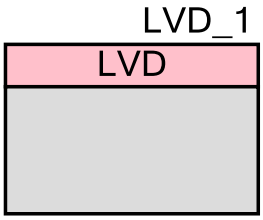
Low Voltage Detect (PDL_LVD)

1.0



Features

- Low-voltage Reset Circuit
- Low-voltage Interrupt Circuit

General Description

The Peripheral Driver Library (PDL) Low-voltage Detection Circuit (LDV) component monitors the power supply voltage and generates reset and interrupt signals when the power supply voltage falls below the detection voltage. There are three types of LVD mounted in different products. For details of each setting, refer to each chapter in the Peripheral Manual Main Part.

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

When to Use a PDL_LVD Component

The PDL_LVD component is specified to detect low supply voltage.

Quick Start

1. Drag a PDL_LVD component from the Component Catalog FMx/System/Low Voltage Detect folder onto your schematic. The placed instance takes the name LVD_1.
2. There is no need to open Configure dialog. This component doesn't provide any parameters.
3. 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 LVD_1 instance.
4. In the *main.c* file, initialize the peripheral and start the application.

```
stc_lvd_config_t stcLvdConfig;
stcLvdConfig.enLvd0IrqDetectVoltage = Lvd0IrqDetectVoltage270;
stcLvdConfig.enLvd0IrqReleaseVoltage = Lvd0IrqReleaseVoltage300;
stcLvdConfig.bLvd0ReleaseVoltageEnable = TRUE;
while(Ok != Lvd_Init(&stcLvdConfig)
{
}
Lvd_EnableIrqDetect(0u);
while(TRUE != Lvd_GetIrqOperationStatus(0u))
{
```

```
}

if(Lvd_GetIrqStatus(0u) == TRUE)
{
Lvd_ClrIrqStatus(0u);/* If low voltage detected then code will run here*/
}
```

5. Build and program the device.

Component Usage

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

Generated Data

The PDL_LVD component doesn't generate peripheral initialization data structures.

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…**" in the drop-down menu.

Preprocessor Macros

The LVD component generates the following preprocessor macro(s). Note that each macro is prefixed with the instance name of the component (e.g. "LVD_1").

Macro	Description
LVD_1_SetPinFunc_LVDI	Macro to initialize LVDI pin. This macro is generated only for Berrypecker devices. Other FM0 devices doesn't have LVDI pin.

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.

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_LVD component uses the Low Voltage Detect (LVD) 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 Changes

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

Version	Description of Changes	Reason for Changes / Impact
1.0.a	Minor datasheet edits.	
1.0	Initial Version	