Application Note

How To Enable Safety Power Off by on-board SuperCAP Module

EDA Technology Co., Ltd

April 2024

Revision History

Date	Revision	Description	Remark
2023-12-7	v00	Initial Version	
2023-4-2	v01		

Content

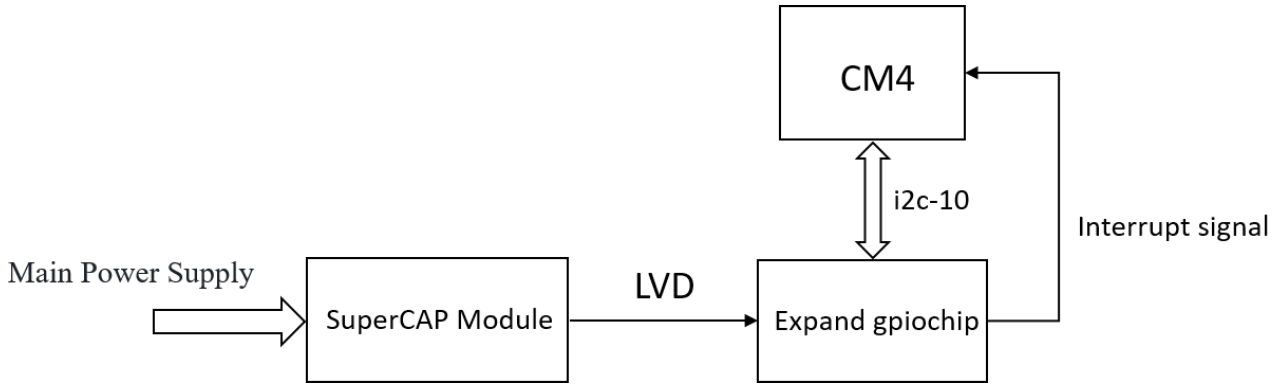
1 Overview

As we know, sudden loss of power may damage the file system and may result a boot failure. In some other application scenarios, it is necessary to save some key data before losing power. Our IPC and HMI serials products support an optional SuperCAP module, it can do it.

1.1 Scope

#	Product Name	Remark
1	ED-IPC2100 series	
2	ED-IPC2200 series	
3	ED-IPC2600 series	
4	ED-HMI2100 series	
5	ED-HMI2200 series	

1.2 System Block Diagram



When the main power supply is active, LVD is a high level. When the main power supply is inactive, LVD is a low level.
The LVD pin is connected to an expanded gpiochip which address is 0x20.

1.3 Working Mechanism

An interrupt is generated by the falling edge of the LVD input, which will trigger a GPIO interrupt event. By this method, we can monitor the level changes of LVD.

2 Basic Configuration and Usage

This git repository is an example of using GPIO interrupt triggering.

```
git clone https://github.com/edatec/peripheral-usage.git
```

2.1 Update dtoverlay

```
cd lvd/dtoverlay
dtc -@ -I dts -O dtb -o ed-pca953x.dtbo ed-pca953x-overlay.dts
sudo cp ./ed-pca953x.dtbo /boot/overlays/
```

2.2 Modify config.txt

```
sudo nano /boot/config.txt
#Modify dtoverlay=ed-pca953x according to the following parameters
#For IPC21xx,IPC26xx,HMI21xx series
dtoverlay=ed-pca953x,ipc2110,addr=0x20
#For IPC22xx,HMI22xx series
dtoverlay=ed-pca953x,ipc2210,addr=0x20
```

Reboot the device after completing the modifications.

2.3 Install gpiod library

```
sudo apt-get -y install libgpiod-dev
sudo apt-get install gpiod
```

2.4 Compile and Run

```
cd lvd
make
```

You will get an executable file lvd.

3 Monitor the Power Failure Event

3.1 Manual testing

```
./lvd
```

Turn off the 12V main power supply of the device, you will obtain the following output:

```
Low voltage
LVD trigger callback script
```

You can modify the macro LVD_ HOOK_ EXEC specifies a custom callback script.

3.2 Using the lvd-detect.service

```
sudo cp lvd /usr/sbin/
sudo cp lvd-callback.sh /usr/sbin/
sudo cp service/lvd-en.sh /usr/sbin/
sudo cp service/lvd-detect.service /lib/systemd/system/

sudo systemctl enable lvd-detect.service
sudo systemctl start lvd-detect.service
```

The output of sudo cat /sys/kernel/debug/gpio:

```
gpiochip2: GPIOs 488-503, parent: i2c/10-0020, 10-0020, can sleep:
 gpio-488 (5V_GOOD                )
 gpio-489 (LVD                    |falling edge          ) in  hi IRQ
 gpio-490 (BUZZER_EN              )
 gpio-491 (4G_RST                 )
 gpio-492 (4G_LED                 )
 gpio-493 (USER_LED               )
```

NOTE: For IPC2110 and IPC2210, the LVD signal is connected to the Pin1 of gpiochip2

Turn off the 12V main power supply of the device, run the following command line

```
tail /var/log/messages
```

You will obtain the following output:

```
raspberrypi root: LVD trigger callback script
```

NOTE: You can modify the content of the script lvd-callback.sh according to the needs of your application

4 SDK Library and Example Code

We use libgpiod to interact with the linux GPIO character device. You can refer to *2.3 Install gpiod library* for installation.

You can obtain LVD gpiochip through the following script:

```
gpiodetect | grep 10-0020 | sed 's/gpiochip//' | awk '{print $1}'
```

NOTE: Its return result is <THE_LVD_GPIOCHIP>

You can obtain LVD pin through the following script:

```
gpioinfo <THE_LVD_GPIOCHIP> | grep 'LVD' | cut -d ':' -f 1 | awk '{print $2}'
```

NOTE: Its return result is <THE_LVD_PIN>

You can use the following script to read the level of the LVD pin:

```
gpioget <THE_LVD_GPIOCHIP> <THE_LVD_PIN>
```

When the 12V main power supply is active, the return value is 1. When the 12V main power supply is inactive, the return value is 0.

NOTE: Before executing this command, you must stop lvd-detect.service:

```
sudo systemctl stop lvd-detect.service
```