

iTE SDK

IO Expander 模組開發指南

V0.9

ITE TECH. INC.

修訂記錄

修訂日期	修訂說明	頁次
2014/10/30	初建版本 V0.9	
2016/04/11	版本 V1.0	

目錄

1.	前言.....	1
1.1	編寫目的.....	1
1.2	適用範圍.....	1
1.3	適用人員.....	1
2.	IO EXPANDER DRIVER 介紹.....	2
2.1	PCF8574 DRIVER 設定.....	2
2.1.1	INITIALIZED PCF8574.....	2
2.1.2	SET INPUT PORTPIN.....	2
2.1.3	SET OUTPUT PORTPIN.....	2
2.1.4	READ PORT.....	2
2.1.5	WRITE PORT.....	3
2.1.6	READ PORTPIN.....	3
2.1.7	SET PORTPIN.....	3
2.1.8	CLEAR PORTPIN.....	3
2.1.9	GET INPUTSTATUS.....	3
3.	IOEXPANDER GPIO 模組介紹.....	4
3.1	IOEXPANDER 模組.....	4
3.2	選用與啟動 IOEXPANDER GPIO.....	4
3.3	設定 INPUT/OUTPUT.....	4
3.4	設定 OUTPUT 值/讀取 INPUT 值.....	4
3.5	註冊 INTERRUPT HANDLER.....	5
4.	軟件配置說明.....	6
4.1	KCONFIG.....	6
4.1.1	IOEXPANDER SETTING OF KCONFIG.....	6
4.1.2	客製化.....	8
5.	GPIO 與 INTERRUPT HANDLER 範例.....	10
5.1	GPIO 範例 1 (OUTPUT).....	10
5.2	GPIO 範例 2(INPUT).....	10
5.3	INTERRUPT HANDLER 範例.....	11

1. 前言

ITE SOC 的 IO Expander 模組提供 8 組 GPIO pin，每個 GPIO pin 都能各自獨立設定成 input 或是 output pin。當 GPIO pin 設成 input 時，可支援 Active Low 中斷觸發條件。

1.1 編寫目的

介紹 IO Expander GPIO 模組之功能，說明如何使用 GPIO 各種 API 來達到各種輸入輸出的功能。

ITP IO Expander driver 已經包裝 GPIO 的硬體設定，提供各種設定 GPIO 的 API 讓上層其他驅動程式或是應用程式使用。使用者可以選定並啟動某個 GPIO pin、設定

1.2 適用範圍

提供 GPIO pin 的選定、啟動、設定 input/output、設定輸出值或是取得輸入值等功能。

1.3 適用人員

軟體應用程式、驅動程式開發者

2. IO Expander Driver介紹

參考函式庫原型 “sdk\include\gpio_ex\pcf8574.h

2.1 PCF8574 driver 設定

IO Expander 是使用 NXP-PCF8574 芯片，PCF8574 提供 8 bit I/O 擴展，以 I2C 為控制介面，並可把 8 根 I/O pin 各自獨立設定成 input 或是 output，另外提供 Active Low 的中斷訊號 (INT)，來通知 input 狀態改變，其中 I2C 的最高運作時脈為 100Khz，以下是基於這種規範所實做的函式庫。

2.1.1 Initialized PCF8574

使用 “IOExpanderDriver_initial” 初始化 PCF8574

```
{  
    IOExpanderDriver_initial (void);  
}
```

2.1.2 Set Input PortPin

使用 “IOExpander_SetInPortPin” 透過 I2C 介面來將其中一根 GPIO 設定為 input mode，輸入參數是 GPIO pin 編號(0~7)，其操作方法如下：

```
{  
    Byte pin=7;  
    IOExpander_SetInPortPin(pin);  
}
```

將 pin7 設定為 input mode，其餘 pin 保留上次狀態，無回傳值。

2.1.3 Set Output Portpin

使用 “IOExpander_SetOutPortPin” 透過 I2C 介面來將其中一根 GPIO 設定為 output mode，輸入參數是 GPIO pin 編號(0~7)，其操作方法如下：

```
{  
    Byte pin=2;  
    IOExpander_SetOutPortPin(pin);  
}
```

將 pin2 設定為 output mode，其餘 pin 保留上次狀態，無回傳值。

2.1.4 Read Port

使用 “IOExpander_ReadPort” 透過 I2C 介面來一次讀取 8 個 pins 的 input 值，其操作方法如下：

```
{  
    Byte R_data;  
    R_data = IOExpander_ReadPort();  
}
```

一次讀取 pin0~pin7 的 status，回傳值為 input 值。

2.1.5 Write Port

使用“IOExpander_WritePort”透過 I2C 介面來一次寫入 8 個 pins 的 output 值，其操作方法如下：

```
{  
    Byte W_data = 0xFF;  
    IOExpander_WritePort(W_data);  
}
```

一次寫入 pin0~pin7 的 output 值，無回傳值。

2.1.6 Read PortPin

使用“IOExpander_ReadPortPin”透過 I2C 介面來讀取其中一根 GPIO 的 input 值，輸入參數是 GPIO pin 編號(0~7)，其操作方法如下：

```
{  
    Byte pin=3;  
    Byte R_data;  
    R_data = IOExpander_ReadPortPin (pin);  
}
```

回傳值為 1 或 0。

2.1.7 Set PortPin

使用“IOExpander_SetPortPin”透過 I2C 介面來寫入其中一根 GPIO 的 output 值為 1，輸入參數是 GPIO pin 編號(0~7)，其操作方法如下：

```
{  
    Byte pin=5;  
    IOExpander_SetPortPin(pin);  
}
```

將 pin5 設為 1，其餘 pin 保留上次狀態，無回傳值。

2.1.8 Clear PortPin

使用“IOExpander_ClrPortPin”透過 I2C 介面來清除其中一根 GPIO 的 output 值為 0，輸入參數是 GPIO pin 編號(0~7)，其操作方法如下：

```
{  
    Byte pin=1;  
    IOExpander_ClrPortPin(pin);  
}
```

將 pin1 設為 0，其餘 pin 保留上次狀態，無回傳值。

2.1.9 Get InputStatus

使用“IOExpander_GetInputStatus”透過 I2C 介面來讀取 input 狀態，可以搭配 PCF8574 中斷訊號，當中斷訊號發生時，調用此函式來得知那些 pin 的狀態改變，其操作方法如下：

```
{  
    Byte status;  
    status = IOExpander_GetInputStatus();  
}
```

一次讀取 pin0~pin7 的 status，並判斷狀態是否改變，回傳值為 1 表示狀態改變。

3. IOExpander GPIO模組介紹

參考 API 原型 “sdk\include\ite\itp.h”

3.1 IOExpander 模組

IO Expander 是建構在 NXP-PCF8574 函式庫之上，提供相關函式，讓使用者很容易對個別 GPIO 進行控制，以下是 IOExpander ITP API。

3.2 選用與啟動 IOExpander GPIO

函式操作是以類似 ITE SOC GPIO 的操作方式，其原型如下：

```
/**
 * Initialized GPIO Expander
 *
 * @param none
 */
void itpIOExpanderInit(void);
```

無輸入參數

3.3 設定 INPUT/OUTPUT

設定 GPIO PIN 為 INPUT MODE，輸入參數是 GPIO pin 編號(0~7)，無回傳值。

```
/**
 * Sets GPIO Expander pin to input mode.
 *
 * @param pin the GPIO Expander pin to set to input mode.
 */
void itpIOExpanderSetIn(unsigned int pin);
```

設定 GPIO PIN 為 OUTPUT MODE，輸入參數是 GPIO pin 編號(0~7)，無回傳值。

```
/**
 * Sets GPIO Expander pin to output mode.
 *
 * @param pin the GPIO Expander pin to set to output mode.
 */
void itpIOExpanderSetOut(unsigned int pin);
```

3.4 設定 OUTPUT 值/讀取 INPUT 值

設定 GPIO 的 OUTPUT 值為 1，輸入參數是 GPIO pin 編號(0~7)，無回傳值。

```
/**
 * Sets GPIO Expander pin to 1.
 *
 * @param pin the GPIO Expander pin to set to 1.
```

```
*/  
void itpIOExpanderSet(unsigned int pin);  
  
設定 GPIO 的 OUTPUT 值為 0，輸入參數是 GPIO pin 編號(0~7)，無回傳值。  
/**  
 * Sets GPIO Expander pin to 0.  
 *  
 * @param pin the GPIO Expander pin to set to 0.  
 */  
void itpIOExpanderClear(unsigned int pin);
```

```
讀取 GPIO 的 input 值，輸入參數是 GPIO pin 編號(0~7)，回傳值為 input 值。  
/**  
 * Gets the value of GPIO Expander pin.  
 *  
 * @param pin the GPIO Expander pin to get.  
 */  
uint8_t itpIOExpanderGet(unsigned int pin);
```

```
讀取 GPIO pin0~pin8 的 status，回傳值為 input 值。  
/**  
 * Gets all the value of GPIO Expander pins.  
 */  
uint8_t itpIOExpanderGetAllPins(void);
```

3.5 註冊 Interrupt handler

每個 IO pin 可以獨立設置相對應的中斷處理程序，當中斷訊號被觸發時，狀態改變的 pin，會被調用對應的中斷處理程序

```
輸入參數是 GPIO pin 編號(0~7)及中斷處理程序，無回傳值。  
/**  
 * Register the interrupt handler of GPIO Expander pin.  
 *  
 * @ param pin the GPIO Expander pin to set.  
 * @ param interrupt handler  
 */  
void itpRegisterIOExpanderIntrHandler(int index, ITPGpioExpanderIntrHandler handler);
```

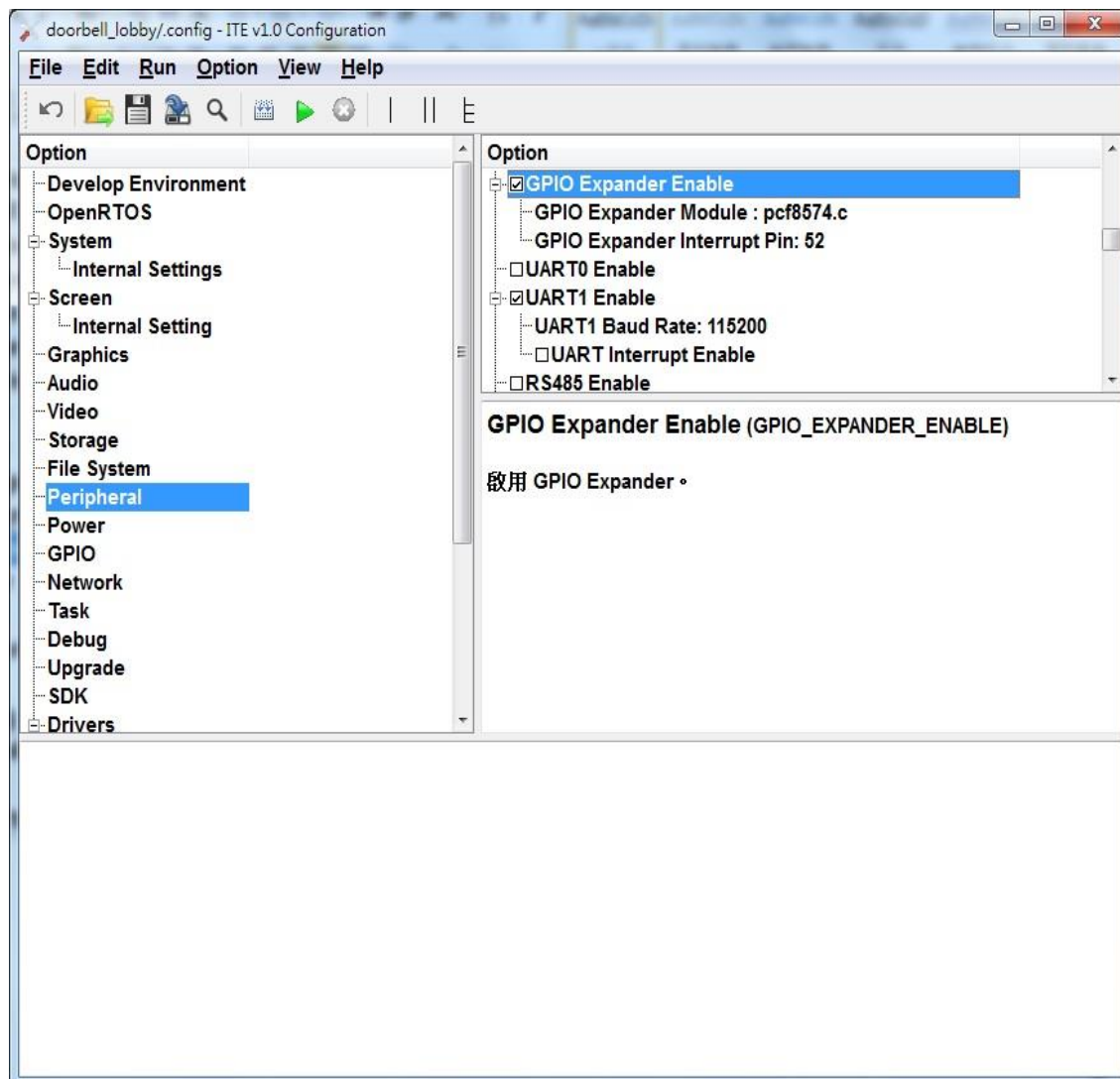

4. 軟件配置說明

4.1 KConfig

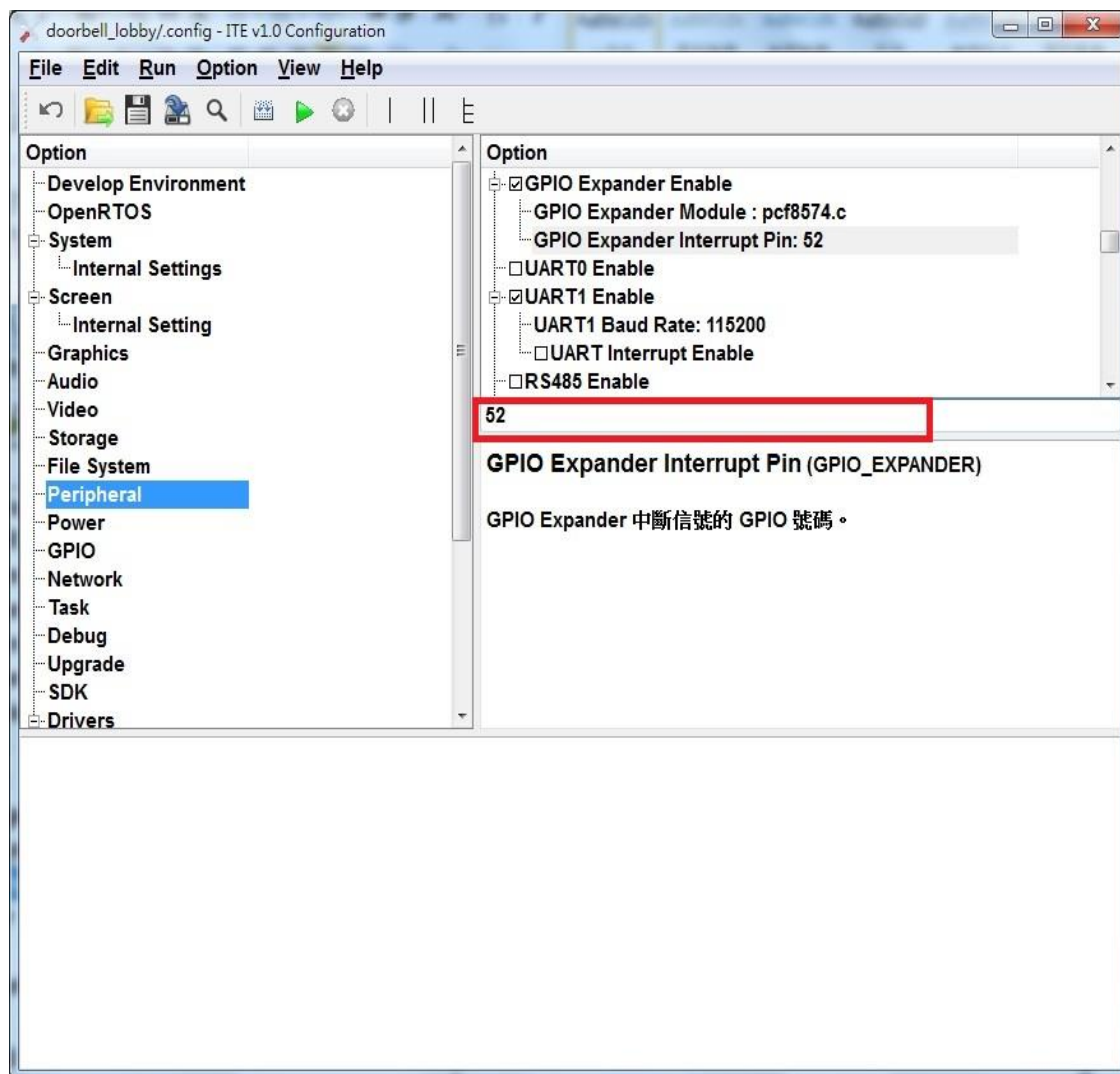
To set Kconfig Depend on necessary of project application

4.1.1 IOExpander Setting of KConfig

Option 選擇 Peripheral，根據使用者的系統應用，是否打開 GPIO Expander，如下圖所示



IO Expander 的實作是 interrupt 驅動，PCF8574 是以 Active Low 為觸發條件，所以需要設置 interrupt pin 的位置，當中斷訊號被觸發時，AP 可以即時得知狀態及做即時的處理利用。PCF8574 Interrupt pin 是接到 ITE SOC 的 GPIO pin，連接兩下“GPIO Expander Interrupt Pin:52”即可從紅色框框輸入對應的 ITE SOC GPIO pin。大門口機初始化設定 GPIO Expander Interrupt Pin 為 52，室內機 GPIO Expander Interrupt Pin 為 45。



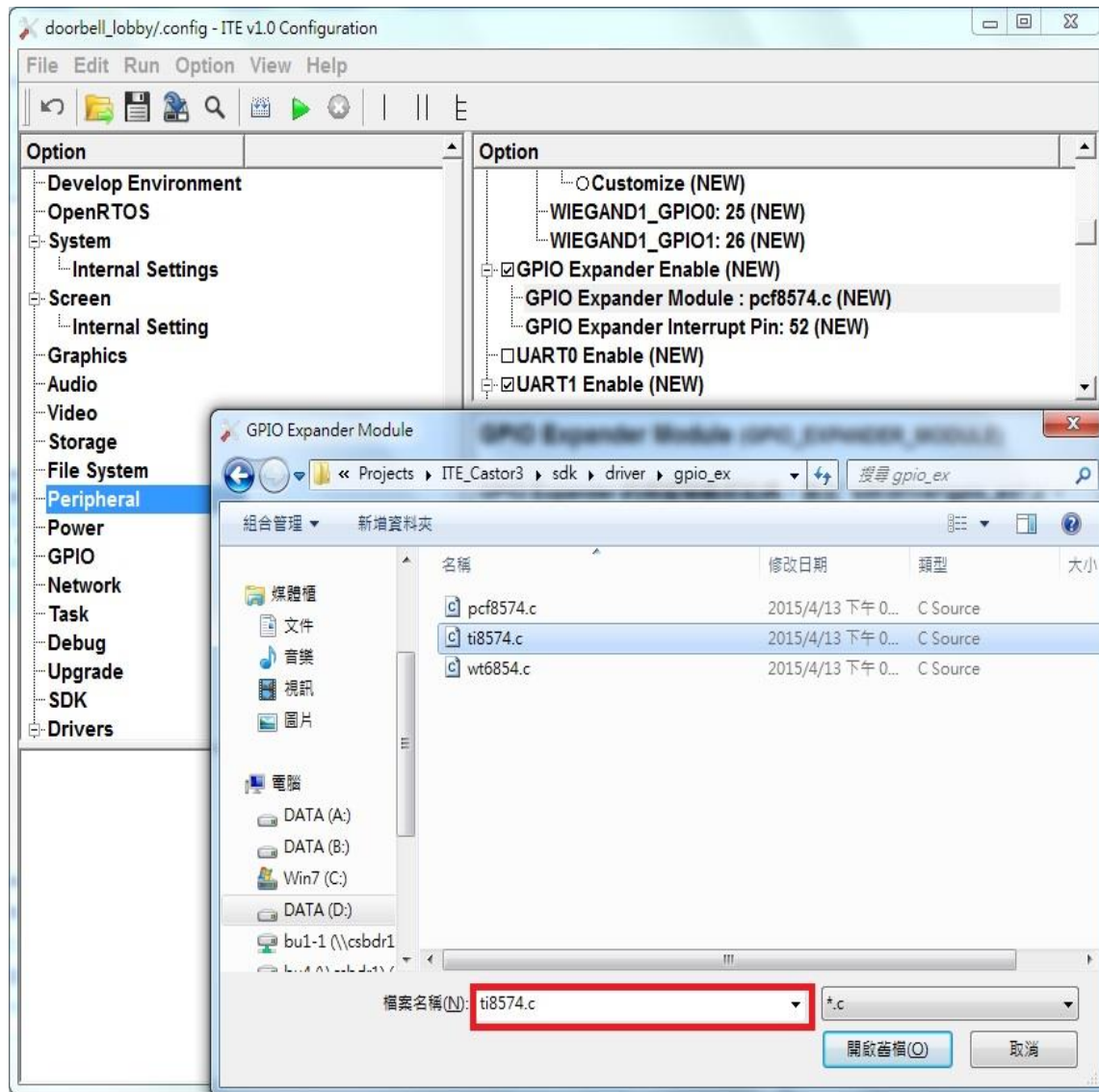
4.1.2 客製化

目前提供的是 NXP 的 PCF8574 drivers，如果使用者要更換替代 IC，需要修改對應的 PCF8574 drivers，實作第二章所列之函式。

Kconif 設置如下：

連按兩下“GPIO Expander Module:pcf8574”即可從紅色框框輸入檔名更換 sdk\driver\gpio_ex*.c 的 driver

Example: 若此 project 使用的 ti8574 的 driver，則連按兩下“GPIO Expander Module:pcf8574”，在紅色框框輸入 driver 的名稱,就可以替換，同時在 sdk\driver\gpio_ex 底下需有一 ti8574.c 的 driver.



5. GPIO與Interrupt handler範例

5.1 GPIO 範例 1 (output)

#include "ite/itp.h" //for all ith driver (include IOExpanderGPIO) & MACRO

```
void main(void)
{
    int gpioPin = 5;
    int i = 0;

    //initial GPIO
    itpIOExpanderSetOut(gpioPin);
    //
    while(1)
    {
        if(i++&0x1)
        {
            itpIOExpanderClear(gpioPin);
        }
        else
        {
            itpIOExpanderSet(gpioPin);
        }
        printf("current GPIO[%d] state=%x, index=%d\n",gpioPin, itpIOExpanderGet(gpioPin),i);
        printf("Get All the GPIO status = 0x%x\n", itpIOExpanderGetAllPins() );
        usleep(1000*1000); //wait for 1 second
    }

    return NULL;
}
```

5.2 GPIO 範例 2(input)

#include "ite/itp.h" //for all ith driver (include IOExpanderGPIO) & MACRO

```
static void _gpioPinInit(void)
{
    itpIOExpanderSetIn(gpioPin);
}
```

```
void main(void)
{
    int gpioPin =1;
    int lastPinStatus = 0;

    //initial GPIO
    _gpioPinInit();
```

```
//
while(1)
{
    //polling gpio pin
    if(itpIOExpanderGet(gpioPin) != lastPinStatus)
    {
        // GPIO status has changed
        lastPinStatus = itpIOExpanderGet(gpioPin);
        if(lastPinStatus)
        {
            printf("The lasted GPIO state is HIGH\n");
        }
        else
        {
            printf("The lasted GPIO state is LOW\n");
        }
    }
}

return NULL;
}
```

5.3 Interrupt handler 範例

```
#include "ite/itp.h"
static void GuardIntrHandler(unsigned int pin, void* arg)
{
    printf("The GPIO5 status has changed !!\n");
}

void main(void)
{
    int gpioPin = 5;
    int lastPinStatus = 0;

    //init GPIO
    itpIOExpanderSetIn(gpioPin);

    //Setting Interrupt handler
    itpRegisterIOExpanderintrHandler(gpioPin,GuardIntrHandler);

    while(1) //polling gpio pin
    {
        if(itpIOExpanderGet (gpioPin) != lastPinStatus)
        {
            // GPIO status has changed
            lastPinStatus = itpIOExpanderGet(gpioPin);
        }
    }
}
```