

iTE SDK

GPIO 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
2014/10/01	初建版本 V0.9	

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.	GPIO 模組介紹.....	2
2.1	選用與啟動 GPIO.....	2
2.2	設定 INPUT/OUTPUT	2
2.3	設定 OUTPUT 值/讀取 INPUT 值	2
2.4	INPUT MODE 的中斷控制	3
2.5	GPIO PIN SHARE	5
3.	GPIO 範例.....	8
3.1	GPIO 範例 1 (OUTPUT)	8
3.2	GPIO 範例 2(INPUT)	8
3.3	GPIO 範例 3(使用中斷功能)	9

表格目錄

TABLE 2-1. GPIO PIN SHARE TABLE	5
---------------------------------------	---

1. 前言

IT9856 的 GPIO 模組提供 81 組 GPIO pin，每個 GPIO pin 都能各自獨立設定成 input 或是 output。當 GPIO pin 設成 input 時，還可以支援各種觸發條件(rising edge, falling edge, both edge, and high/low level)的中斷。

ITH 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 與設定

1.1 編寫目的

介紹 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。.

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、設定中斷與觸發條件、設定內部 pull up/down 等功能。

1.3 適用人員

軟體應用程式、驅動程式開發者

2. GPIO模組介紹

參考 API 原型 “sdk\include\ite\ith.h”

2.1 選用與啟動 GPIO

依據表 2-1，選定一個特定且可用的 GPIO PIN(通常會避開已經被其他功能占用的 GPIO PIN，例如 I2C(使用 GPIO2&GPIO3)、網路(GPIO34~44)...)，使用 ithGpioEnable()，API 的原型如下：

```
/**
 * Enables GPIO pin.
 *
 * @param pin the GPIO pin to enable.
 */
static inline void ithGpioEnable(unsigned int pin)
```

輸入參數是 GPIO pin 編號(0~81)，無回傳值。

2.2 設定 INPUT/OUTPUT

設定 GPIO PIN 為 INPUT MODE，輸入參數是 GPIO pin 編號(0~81)，無回傳值。

```
/**
 * Sets GPIO pin to input mode.
 *
 * @param pin the GPIO pin to set to input mode.
 */
static inline void ithGpioSetIn(unsigned int pin)
```

設定 GPIO PIN 為 OUTPUT MODE，輸入參數是 GPIO pin 編號(0~81)，無回傳值。

```
/**
 * Sets GPIO pin to output mode.
 *
 * @param pin the GPIO pin to set to output mode.
 */
static inline void ithGpioSetOut(unsigned int pin)
```

2.3 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(0~81)，無回傳值。

```
/**
 * Sets GPIO pin to 1.
 *
 * @param pin the GPIO pin to set to 1.
 */
static inline void ithGpioSet(unsigned int pin)
```

設定 GPIO 的 OUTPUT 值為 0，輸入參數是 GPIO pin 編號(0~81)，無回傳值。

```
/**
 * Sets GPIO pin to 0.
 *
 * @param pin the GPIO pin to set to 0.
 */
static inline void ithGpioClear(unsigned int pin)
```

讀取 GPIO 的 input 值，輸入參數是 GPIO pin 編號(0~81)，回傳值為 input 值。

```
/**
 * gets the value of gpio pin.
 *
 * @param pin the gpio pin to get.
 */
static inline uint32_t ithGpioGet(unsigned int pin)
```

2.4 INPUT MODE 的中斷控制

GPIO 的中斷控制的定義

```
/**
 * GPIO interrupt controls definition.
 */
typedef enum
{
    ITH_GPIO_PULL_ENABLE           = 0, ///< GPIO pin pull enable
    ITH_GPIO_PULL_UP               = 1, ///< GPIO pin pull up
    ITH_GPIO_INTR_LEVELTRIGGER     = 2, ///< GPIO interrupt trigger method
    ITH_GPIO_INTR_BOTHEDGE        = 3, ///< GPIO interrupt edge trigger by both edge
    ITH_GPIO_INTR_TRIGGERFALLING  = 4, ///< GPIO interrupt trigger by falling edge
    ITH_GPIO_INTR_TRIGGERLOW       = 4, ///< GPIO interrupt trigger by low level
} ITHGpioCtrl;
```

晶片內部 GPIO PULL 控制定義之說明:

ITH_GPIO_PULL_ENABLE：啟動或關閉晶片內部 GPIO PULL 功能。

啟動晶片內部 GPIO PULL 功能：ithGpioCtrlEnable(pin, ITH_GPIO_PULL_ENABLE)。

關閉晶片內部 GPIO PULL 功能：ithGpioCtrlDisable(pin, ITH_GPIO_PULL_ENABLE)。

ITH_GPIO_PULL_UP：設定晶片內部 GPIO pull 的方式是 pull up 或 pull down。

pull up：ithGpioCtrlEnable(pin, ITH_GPIO_PULL_UP)。

pull down：ithGpioCtrlDisable(pin, ITH_GPIO_PULL_UP)。

中斷控制定義之說明:

ITH_GPIO_INTR_LEVELTRIGGER：設定中斷的 trigger 方式。

level trigger：ithGpioCtrlEnable(pin, ITH_GPIO_INTR_LEVELTRIGGER)。

edge trigger：ithGpioCtrlDisable(pin, ITH_GPIO_INTR_LEVELTRIGGER)。

ITH_GPIO_INTR_BOTHEDGE：設定中斷的 edge trigger 方式為 both edge 或 single edge。

both edge : `ithGpioCtrlEnable(pin, ITH_GPIO_INTR_BOTHEDGE)` 。
single edge : `ithGpioCtrlDisable(pin, ITH_GPIO_INTR_BOTHEDGE)` 。

ITH_GPIO_INTR_TRIGGERFALLING : 設 edge trigger 的方式為 falling edge 或 rising edge 。
falling edge : `ithGpioCtrlEnable(pin, ITH_GPIO_INTR_TRIGGERFALLING)` 。
rising edge : `ithGpioCtrlDisable(pin, ITH_GPIO_INTR_TRIGGERFALLING)` 。

ITH_GPIO_INTR_TRIGGERLOW : 設定中斷的 level trigger 方式為 low level 或是 high level 。
low level : `ithGpioCtrlEnable(pin, ITH_GPIO_INTR_TRIGGERLOW)` 。
high level : `ithGpioCtrlDisable(pin, ITH_GPIO_INTR_TRIGGERLOW)` 。

啟動 GPIO input mode 的相關控制，包含 PULL ENABLE/PULL UP/中斷觸發型態等設定。輸入參數有 GPIO pin 編號(0~81)，以及控制的參數，無回傳值。

```
/**
 * Enables specified controls.
 *
 * @param pin the GPIO pin to enable.
 * @param ctrl the controls to enable.
 */
void ithGpioCtrlEnable(unsigned int pin, ITHGpioCtrl ctrl);
```

關閉 GPIO input mode 的相關控制，包含 PULL ENABLE/PULL UP/中斷觸發型態等設定。輸入參數有 GPIO pin 編號(0~45)，以及控制的參數，無回傳值。

```
/**
 * Disables specified controls.
 *
 * @param pin the GPIO pin to disable.
 * @param ctrl the controls to disable.
 */
void ithGpioCtrlDisable(unsigned int pin, ITHGpioCtrl ctrl);
```

註冊 GPIO 的 ISR，輸入參數有 GPIO pin 編號(0~81)，要註冊的 ISR 函式的位置，以及要傳入 ISR 的參數(void* arg)，無回傳值。

```
/**
 * GPIO interrupt handler.
 *
 * @pin The GPIO pin.
 * @arg Custom argument.
 */
void ithGpioRegisterIntrHandler(unsigned int pin, ITHGpioIntrHandler handler, void* arg)
```

啟動 GPIO 的中斷功能，輸入參數有 GPIO pin 編號(0~81)，無回傳值。

```
/**
 * enables the interrupt of gpio pin.
 *
 * @param pin the gpio pin to enable.
 */
static inline void ithGpioEnableIntr(unsigned int pin)
```


關閉 GPIO 的中斷功能，輸入參數有 GPIO pin 編號(0~81)，無回傳值。

```
/**
 * Disables the interrupt of GPIO pin.
 *
 * @param pin the GPIO pin to disable.
 */
static inline void ithGpioDisableIntr(unsigned int pin)
```

清除 GPIO 的中斷旗標，輸入參數有 GPIO pin 編號(0~81)，無回傳值。

```
/**
 * Clears the interrupt of GPIO pin.
 *
 * @param pin the GPIO pin to clear.
 */
static inline void ithGpioClearIntr(unsigned int pin)
```

中斷設定的流程

- 建議使用 `ithEnterCritical()`，避免被其他中斷干擾。
- 清除先前的中斷，使用 `ithGpioClearIntr(pin)` 清除此 GPIO pin 的中斷訊號。
- 註冊 ISR，使用 `ithGpioRegisterIntrHandler()` 來註冊 ISR。
- 設定中斷方式，使用 `ithGpioCtrlEnable()` 以及 `ithGpioCtrlDisable()` 來設定中斷的型式(如 edge trigger/level trigger, both edge/single edge, ...)。
- 啟動 GPIO 中斷的 IRQ，使用 `ithIntrEnableIrq(ITH_INTR_GPIO)` 來啟動 GPIO 的 IRQ。
- 啟動此 GPIO pin 的中斷功能，使用 `ithGpioEnableIntr()` 來啟動此 GPIO pin 的中斷功能。
- 使用 `ithExitCritical()`，離開 critical section。

2.5 GPIO pin share

GPIO PIN 為 multi-function pin 除了 GPIO pin(mode0)的功能外，每個 GPIO 還能設定成不同功能(mode1~mode3)，因此，ITH driver 有提供一組 API 來設定每個 GPIO pin 的 mode0~mode3(mode0 為 GPIO mode, 為開機預設值)。

設定 GPIO PIN 的 mode0~mode3，輸入參數有 GPIO pin 編號(0~81)，以及其 GPIO mode，無回傳值。

```
/**
 * Sets the mode of GPIO pin.
 *
 * @param pin the GPIO pin to set.
 * @param mode the mode to set.
 */
void ithGpioSetMode(unsigned int pin, ITHGpioMode mode);
```

Table 2-1. GPIO Pin Share Table

Pin	Mode 0	Mode 1	Mode 2	Mode 3
-----	--------	--------	--------	--------

Pin	Mode 0	Mode 1	Mode 2	Mode 3
GPIO0	GPIO0	UART0 DCD	IIC_0 SDA	PWM4
GPIO1	GPIO1	UART0 RTS	IIC_0 SCL	PWM5
GPIO2	GPIO2	IrDA_RXL	IIC_1 SDA	IIC_0 SDA
GPIO3	GPIO3	IrDA_TX	IIC_1 SCL	IIC_0 SCL
GPIO4	GPIO4	UART0 RI	Remote IR Out	PWM0
GPIO5	GPIO5	Remote IR Out	IIS ZDO 1	PWM1
GPIO6	GPIO6	UART0 CTS/IrDA_RXH	IIS ZDI 1	PWM2
GPIO7	GPIO7	Remote IR Out	IIS AMCLK	PWM3
GPIO8	GPIO8	Remote IR Out	PWM4	SDCLK2
GPIO9	GPIO9	Remote IR Out	IIS ZWS	UART1 CTS
GPIO10	GPIO10	SPI0 CS#2	IIS ZCLK	UART1 RTS
GPIO11	GPIO11	SPI1 CS#2	IIS AMCLK	IIC_0 SDA
GPIO12	GPIO12	UART0 DSR	IIS ZDI 1	IIC_0 SCL
GPIO13	GPIO13	UART0 DTR	IIS ZDO 1	SDCLK2
GPIO14	GPIO14	PWM1	-	SPI0 CS#1
GPIO15	GPIO15	PWM0	SPI0 CS#3	SPI1 CS#1
GPIO16	GPIO16	SDCLK1	SPI1 CS#3	SPDIF OUT
GPIO17	GPIO17	SDCMD	-	PWM2
GPIO18	GPIO18	SD0	-	SPI0 MOSI
GPIO19	GPIO19	SD1	-	SPI0 MISO
GPIO20	GPIO20	SD2	-	SPI0 CLK
GPIO21	GPIO21	SD3	-	PWM5
GPIO22	GPIO22	SD4	-	PWM0
GPIO23	GPIO23	SD5	-	IIC_0 SDA
GPIO24	GPIO24	SD6	-	IIC_0 SCL
GPIO25	GPIO25	SD7	-	IIC_1 SDA
GPIO26	GPIO26	PWM1	-	IIC_1 SCL
GPIO27	GPIO27	PWM2	-	REFCLK
GPIO28	GPIO28	Remote IR Out	-	TX_EN
GPIO29	GPIO29	SPI1 MOSI	-	TX_D1
GPIO30	GPIO30	SPI1 MISO	-	TX_D0
GPIO31	GPIO31	SP1 CLK	-	RXD1
GPIO32	GPIO32	IIS AMCLK	-	RXD0
GPIO33	GPIO33	IIS ZWS	-	RX_CRS_DV
GPIO34	GPIO34	IIS ZDO 1	-	MDIO
GPIO35	GPIO35	IIS ZCLK	-	MDC
GPIO36	GPIO36	IIS ZDI 1	Remote IR Out	RX_ER
GPIO37	GPIO37	LSCK	SPI0 CS#4	REF_25MHZ
GPIO38	GPIO38	LSA0	SPI1 CS#4	REF_25MHZ
GPIO39	GPIO39	LD23	PWM0	TXEN
GPIO40	GPIO40	LD22	PWM1	TXD1
GPIO41	GPIO41	LD21	PWM2	TXD0
GPIO42	GPIO42	LD20	PWM3	REFCLK
GPIO43	GPIO43	LD19	PWM4	RXD1
GPIO44	GPIO44	LD18	PWM5	RXD0
GPIO45	GPIO45	LD17	Wiegand 0 out	RX_CRS_DV
GPIO46	GPIO46	LD16	-	PWM0
GPIO47	GPIO47	LD15	-	PWM1
GPIO48	GPIO48	LD14	-	PWM2
GPIO49	GPIO49	LD13	-	PWM3
GPIO50	GPIO50	LD12	-	PWM4
GPIO51	GPIO51	LD11	-	PWM5
GPIO52	GPIO52	LD10	-	-

Pin	Mode 0	Mode 1	Mode 2	Mode 3
GPIO53	GPIO53	LD9	-	-
GPIO54	GPIO54	LD8	-	-
GPIO55	GPIO55	LD7	-	-
GPIO56	GPIO56	LD6	-	-
GPIO57	GPIO57	LD5	-	-
GPIO58	GPIO58	LD4	-	TSOCLK
GPIO59	GPIO59	LD3	-	TSOCSN
GPIO60	GPIO60	LD2	-	TSOSTR
GPIO61	GPIO61	LD1	-	TSODATA0
GPIO62	GPIO62	LD0	Wiegand 1 out	TSODATA1
GPIO63	GPIO63	LSDA	PWM5	TSODATA2
GPIO64	GPIO64	LCSN	PWM4	TSODATA3
GPIO65	GPIO65	LDEN	PWM3	TSODATA4
GPIO66	GPIO66	LDCLK	PWM2	TSODATA5
GPIO67	GPIO67	LHSYNC	PWM1	TSODATA6
GPIO68	GPIO68	LVSYN	PWM0	TSODATA7
GPIO69	GPIO69	CAP_MCLK	IIC_0 SDA	IIC_1 SDA
GPIO70	GPIO70	CAP_DE	IIC_0 SCL	IIC_1 SCL
GPIO71	GPIO71	CAP_PCLK	-	TSICLK
GPIO72	GPIO72	CAP_HSYNC	-	TSICSN
GPIO73	GPIO73	CAP_VSYNC	-	TSISTR
GPIO74	GPIO74	CAP_D0	-	TSIDATA0
GPIO75	GPIO75	CAP_D1	-	TSIDATA1
GPIO76	GPIO76	CAP_D2	PWM5	TSIDATA2
GPIO77	GPIO77	CAP_D3	PWM4	TSIDATA3
GPIO78	GPIO78	CAP_D4	PWM3	TSIDATA4
GPIO79	GPIO79	CAP_D5	PWM2	TSIDATA5
GPIO80	GPIO80	CAP_D6	PWM1	TSIDATA6
GPIO81	GPIO81	CAP_D7	PWM0	TSIDATA7

3. GPIO範例

3.1 GPIO 範例 1 (output)

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

void main(void)
{
    int gpioPin = 13;
    int i = 0;

    itpInit();

    //initial GPIO
    ithGpioSetOut(gpioPin);
    ithGpioSetMode(gpioPin, ITH_GPIO_MODE0);

    //
    while(1)
    {
        if(i++&0x1)
        {
            ithGpioClear(gpioPin);
        }
        else
        {
            ithGpioSet(gpioPin);
        }
        printf("current GPIO[%d] state=%x, index=%d\n",gpioPin,ithGpioGet(gpioPin),i);
        usleep(1000*1000); //wait for 1 second
    }

    return NULL;
}
```

3.2 GPIO 範例 2(input)

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

static void _gpioPinInit(void)
{
    //ithGpioCtrlEnable(pin, ITH_GPIO_PULL_ENABLE);    //enable internal pull function(depend on circuit)
    //ithGpioCtrlEnable(pin, ITH_GPIO_PULL_UP);        //set internal pull up(depend on circuit)

    ithGpioSetMode(gpioPin, ITH_GPIO_MODE0);
    ithGpioSetIn(gpioPin);
    ithGpioEnable(gpioPin);
}
```

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

    itpInit();

    //initial GPIO
    _gpioPinInit();

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

    return NULL;
}
```

3.3 GPIO 範例 3(使用中斷功能)

```
#include "ite/itp.h" //for ITH_GPIO_MODE0 & GPIO ith driver

#define INTR_GPIO_PIN 13
static char g_GPIO_INTR = 0;

void _gpio_isr(void* data)
{
    g_GPIO_INTR = 1;
}

void _initGpioPin(int pin)
{
    ithGpioSetMode(pin, ITH_GPIO_MODE0); //設定此 GPIO pin 是 GPIO mode(mode0)
    ithGpioSetIn(pin); //設定此 GPIO pin 為 input
    ithGpioCtrlEnable(pin, ITH_GPIO_PULL_ENABLE); //啟動此 GPIO pin 使用內部 PULL 功能
    ithGpioCtrlEnable(pin, ITH_GPIO_PULL_UP); //設定此 GPIO pin 使用內部 PULL UP 功能
}
```

```
    ithGpioEnable(pin); //啟動此 GPIO pin
}

void _initGpioIntr(int pin)
{
    ithEnterCritical(); //to prevent from interrupt
    ithGpioClearIntr(pin); //清除先前的中斷事件
    ithGpioRegisterIntrHandler(pin, _gpio_isr, (void*)pin); //註冊此 GPIO pin 的 ISR
    ithGpioCtrlDisable(pin, ITH_GPIO_INTR_LEVELTRIGGER); //設定中斷 trigger 方式是使用 edge trigger
    ithGpioCtrlEnable(pin, ITH_GPIO_INTR_BOTHEDGE); //設定中斷 edge trigger 方式是使用 both edge
    ithIntrEnableIrq(ITH_INTR_GPIO); //啟動 GPIO 的中斷功能(中斷控制器)
    ithGpioEnableIntr(pin); //啟動此 GPIO pin 的中斷功能
    ithExitCritical(); //unlock spinlock(建議使用)
}

void _initGpio(int pin)
{
    _initGpioPin(pin);
    _initGpioIntr(pin);
}

void main(void)
{
    int gpioPin = 13;
    int i = 0;

    iplInit();

    _initGpio(gpioPin);

    while(1)
    {
        //polling "g_GPIO_INTR"
        if(g_GPIO_INTR)
        {
            unsigned int gpioState = ithGpioGet(gpioPin) ? 1 : 0;
            printf("current GPIO[%d] state = %d \n", gpioPin, gpioState);
            g_GPIO_INTR = 0;
            ithGpioClearIntr(gpioPin);
        }
        usleep(1000); //sleep for 1 ms
    }

    return NULL;
}
```