

iTE SDK

LCD back light 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.	BACKLIGHT 模組介紹	2
2.1	DESCRIPTION.....	2
2.2	BACKLIGHT 模組的 KCONFIG	2
2.3	BACKLIGHT 模組的初始化.....	3
2.4	BACKLIGHT 模組的 TURN ON/OFF.....	3
2.5	BACKLIGHT 模組的亮度設定.....	4
3.	BACKLIGHT 範例.....	5
3.1	BACKLIGHT 範例 1	5

1. 前言

介紹 backlight 模組是利用 PWM 原理，藉由控制 duty cycle(0%~100%)值，來達到調節 LCD backlight 亮度。但以下將著重 ITP driver 有關 backlight API 的部分以及如何應用此 APIs 來達到控制 backlight module。

1.1 編寫目的

介紹 Back Light(PWM) 模組之功能, 說明 Backlight API 的操作與使用。Backlight 的 ITP API 已包裝 PWM 硬體的控制，軟體開發者只需依據接下來的說明即可控制 backlight module。

1.2 適用範圍

backlight 的初始化，burn on/off，設定 backlight 亮度等功能。
設定適合的 GPIO，做為 backlight 的控制訊號。

1.3 適用人員

軟體應用程式開發者

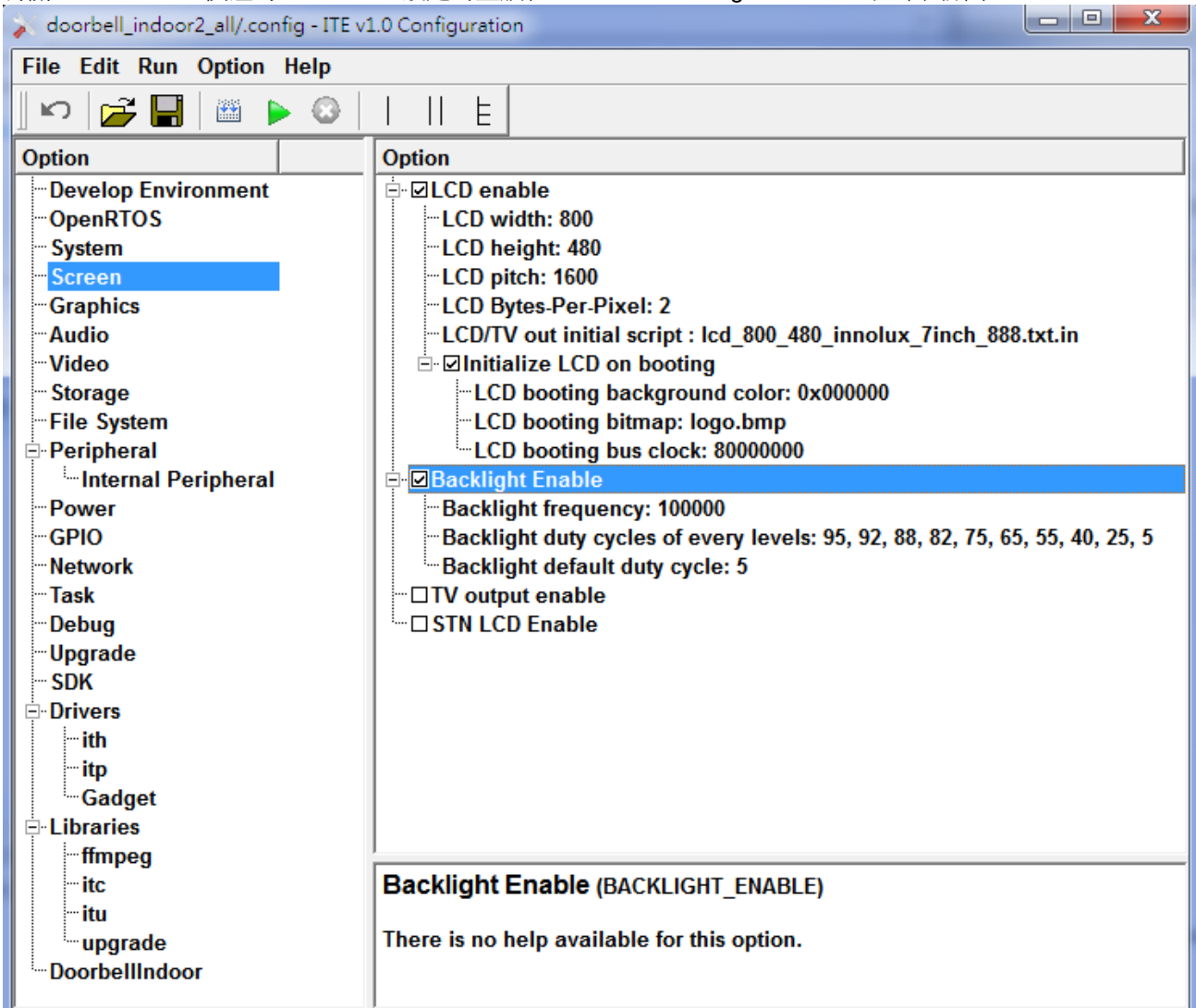
2. BackLight模組介紹

2.1 Description

ITP 的 driver 中，其中有一部份是跟控制 backlight 相關的 API。軟體開發者可以藉由這些 API 簡單地控制 backlight module。關於所有 BackLight 模組 ITP API 的原始碼，請參考“\sdk\driver\itp\itp_backlight.c”。控制 backlight 模組，通常伴隨 LCD 模組的控制，所以視需要參考 LCD 模組說明文件。

2.2 BackLight 模組的 Kconfig

有關 BACKLIGHT 模組的 KCONFIG 設定為置放在 Screen->BackLight Enable，如下圖所示。



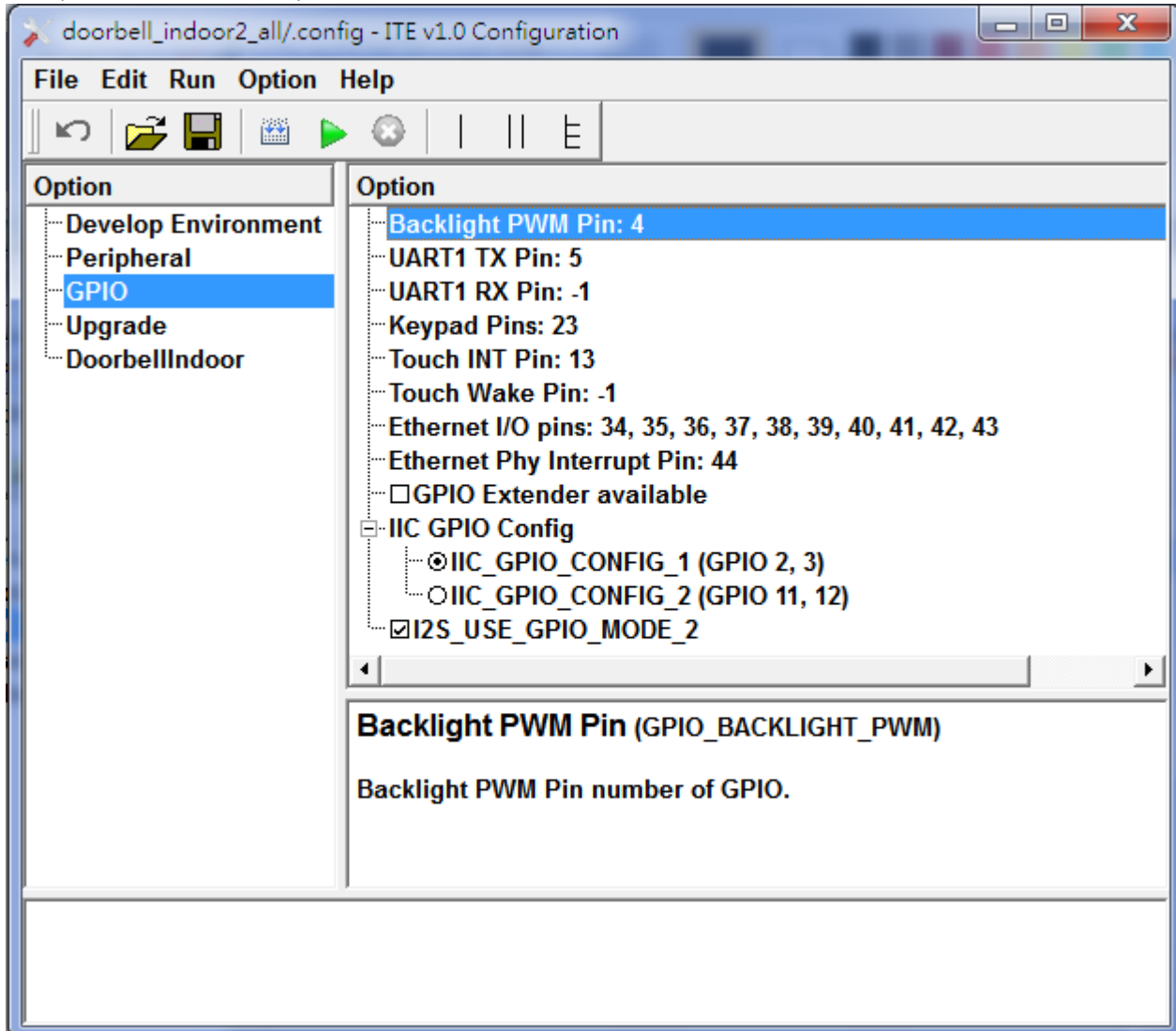
Backlight Enable：啟動 Backlight 模組。

Backlight frequency：設定 PWM 的頻率

Backlight duty cycles of every levels：設定 Backlight 亮度的階數以及各階的 duty cycle 值。

Backlight default duty cycle：設定 Backlight 亮度的預設值。

在 KCONFIG 選定一個 GPIO 做為 backlight 的控制訊號。目前 IT9070 提供 GPIO4~7 做為 backlight 的控制訊號。(如下圖是選用 GPIO4)



2.3 BackLight 模組的初始化

```
ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_INIT, NULL);
```

ioctl() : 參考“sdk\include\ite\itp.h”中的“Device ioctl method”

ITP_DEVICE_BACKLIGHT : 參考“sdk\include\ite\itp.h”中的“device types”

ITP_IOCTL_INIT : 參考“sdk\include\ite\itp.h”中的“IOCTL definitions”

2.4 BackLight 模組的 turn on/off

```
ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_ON, NULL);
```

```
ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_OFF, NULL);
```

ITP_IOCTL_ON: 對 ITP DEVICE 執行 TURN-ON 的動作

ITP_IOCTL_OFF: 對 ITP DEVICE 執行 TURN-OFF 的動作

2.5 BackLight 模組的亮度設定

Backlight 的 ITP API 提供以下兩個 API:

```
int = ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_GET_MAX_LEVEL, NULL);  
ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_SET_BRIGHTNESS, (void*)brightness);
```

ITP_IOCTL_GET_MAX_LEVEL: 取得 Backlight 模組的亮度階數，回傳值的資料型別為 int

ITP_IOCTL_SET_BRIGHTNESS: 設定亮度，其中第三個參數“brightness”為亮度設定值，範圍是 0~最大亮度階數(最大亮度階數即”ITP_IOCTL_GET_MAX_LEVEL”所取得的值)。可參考下一頁 Backlight 的範例程式。

3. Backlight範例

3.1 BackLight 範例 1

This example code will demonstrate how to turn on/off backlight, how to get the max level of brightness, and how to control the brightness. Also refer to the example code in “\project\test_lcd\test_backlight.c”.

```
#include "ite/itp.h"

int main(void)
{
    uint16_t* addr;
    uint32_t col;
    uint32_t row;
    uint32_t x, y, i = 0;
    int maxLevel, brightness = 0;

    itpInit();

    addr = (uint16_t*) ithLcdGetBaseAddrA();
    col = ithLcdGetPitch() / 2;
    row = ithLcdGetHeight();

    ioctl(ITP_DEVICE_SCREEN, ITP_IOCTL_POST_RESET, NULL);    //reset screen
    ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_ON, NULL);        //turn on backlight

    //use "ioctl" to get the value of max level of backlight
    maxLevel = ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_GET_MAX_LEVEL, NULL);

    for (;;)
    {
        uint16_t* base = ithMapVram((uint32_t) addr, ithLcdGetPitch() * ithLcdGetHeight(), ITH_VRAM_WRITE);
        uint16_t color = ITH_RGB565(255, 0, 0); //set LCD as red color
        uint16_t* ptr = base;

        if (++brightness > maxLevel)    brightness = 0;

        printf("BackLight brightness = %d, \n", brightness);

        ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_SET_BRIGHTNESS, (void*)brightness);

        //fill up the VRAM with "red color"
        for (y = 0; y < row; y++)
            for (x = 0; x < col; x++)
                *ptr++ = color;

        ithFlushDCacheRange(base, row * col * 2);    //flush VRAM data
        sleep(1);
    }
}
```



```
ioctl(ITP_DEVICE_BACKLIGHT, ITP_IOCTL_OFF, NULL); //turn off backlight  
return NULL;  
}
```