

MIAT-STM32-EVB

軟硬體實驗模組與開發流程介紹



WU-YANG
Technology Co., Ltd.



Declared Version

Training Only

Declare

Document Version	1.00
Release Date	2009.06.20
Document Title	MIAT-STM32-EVB軟硬體實驗模組與開發流程介紹
Exercise Time	■ Lecture 30 minutes ■ Operating 60 minutes
Platform	■ MIAT_STM32 ■ MIAT_IOB
Peripheral	Key Switch, LED
Author	■ WU-YANG Technology Co., Ltd.



Outline

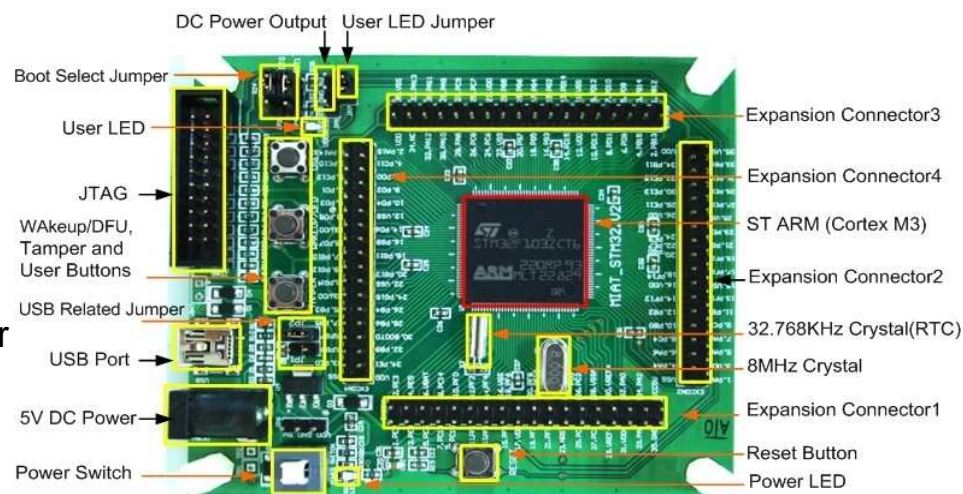
- MIAT_STM32實驗板
 - 主要功能介紹
 - 硬體環境設定
- MIAT_IOB實驗板
 - 各個週邊模組功能介紹
- 整合開發流程介紹
 - 以基本I/O驅動為例
 - 硬體電路配置
 - RVMDK軟體基本使用方法
 - DFU韌體燒錄
- 開發流程練習



MIAT_STM32實驗板主要功能介紹

□ 處理器編號STM32F103ZC

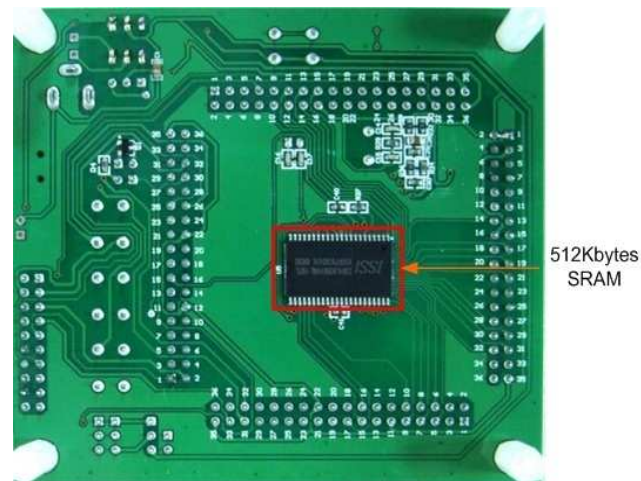
- 256Kbytes of Flash memory
- 48Kbytes of SRAM
- 7 × 16 pin fast I/O ports
- 3 × 12bit ADC
- 1 × 12bit DAC
- Flexible static memory controller
- DMA controller
- 2 × watchdog timers
- 4 × 16-bit timers
- 2 × I2C interfaces
- 5 × USARTs
- 3 × SPIs
- 1 × RTC
- CAN interface
- USB 2.0 full speed interface
- SDIO interface
- Serial wire debug (SWD) & JTAG interfaces
- Temperature sensor
- LQFP 144





MIAT_STM32實驗板主要功能介紹

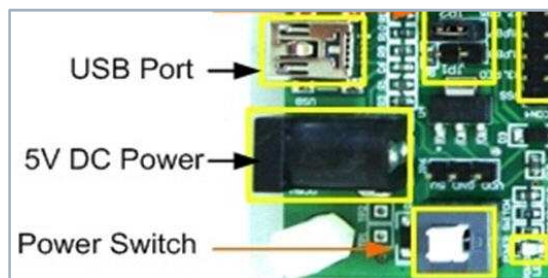
- ❑ 外部記憶體512Kbytes SRAM
(可支援至1Mbytes)
- ❑ 8MHz crystal系統時脈
- ❑ 32.768KHz crystal 即時時鐘(RTC)時脈
- ❑ 可由USB介面燒錄程式
- ❑ 標準JTAG ICE接頭
- ❑ 1個紅色的使用者測試 LED
- ❑ 3個按鈕開關
- ❑ 核心晶片之I/O接腳以4排2.54mm間距的2x18PIN排針連接
- ❑ 電源DC5V：可由DC 座或mini USB座提供
- ❑ 尺寸：92.5x85mm





MIAT_STM32實驗板環境設定

□ 電源輸入



□ 處理器啟動模式選擇

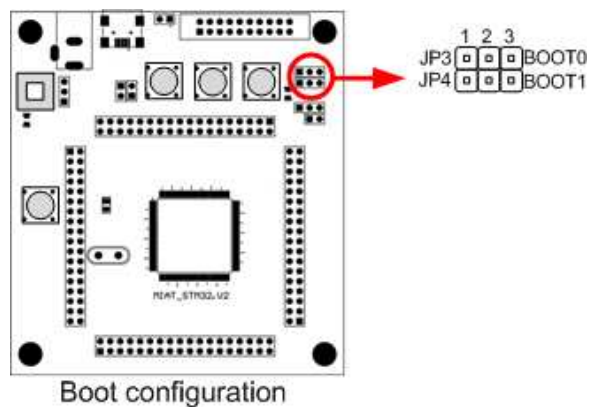






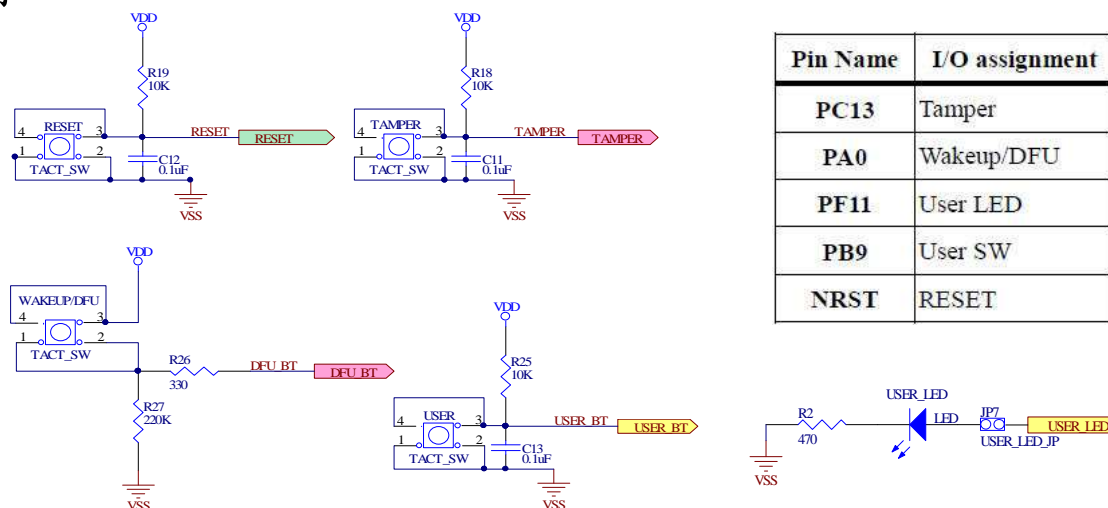
表 2.1 啟動模式選擇。

Jumper Configuration	Boot mode		Boot space
	BOOT1	BOOT0	
 or 	X	0	Embedded main Flash memory(user Flash)
	0	1	System memory with boot loader for ISP
	1	1	Embedded SRAM for debugging



MIAT_STM32實驗板環境設定

LED與按鈕開關



USB介面連接設定

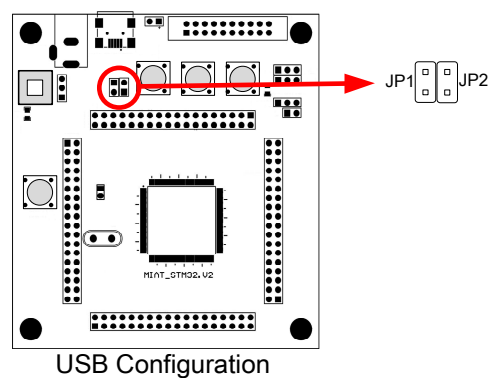


表 2.2 USB 介面裝置設定表。

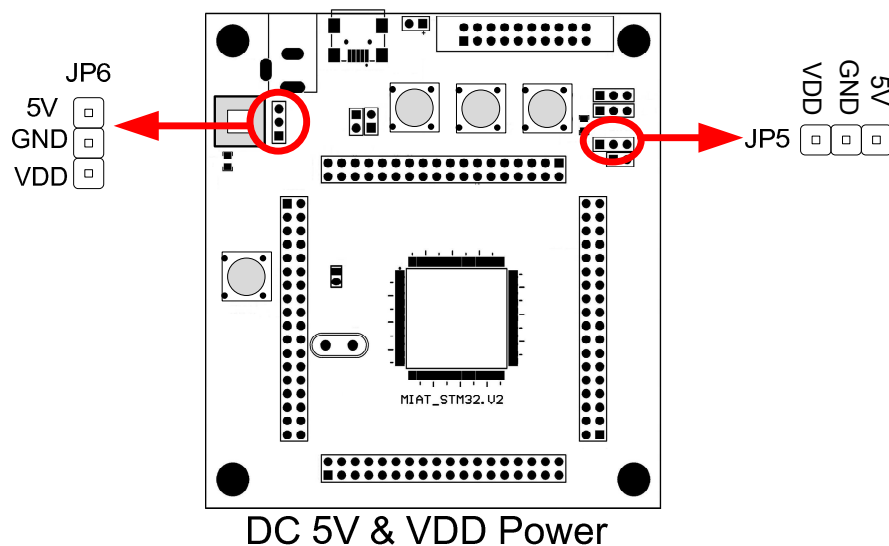
Jumper Configuration	JP1	JP2	功能
	開路	開路	關閉 USB 介面功能。
	開路	短路	啟動 USB 介面功能。
	短路	開路	由軟體透過 PD2 控制 USB 介面功能是否啟動。



MIAT_STM32實驗板環境設定

□ DC5V與VDD(3.0V)電源輸出連接器

- 一般使用PC之USB介面作為電源輸入可提供5V/500mA，若外接實驗週邊使得POWER LED有閃爍、偏暗或熄滅的情況發生時，表示超過可使用的電流量，建議將外接的實驗週邊另外連接獨立的電源或將USB供電改由DC座供電並使用具有較高輸出電流之變壓器。

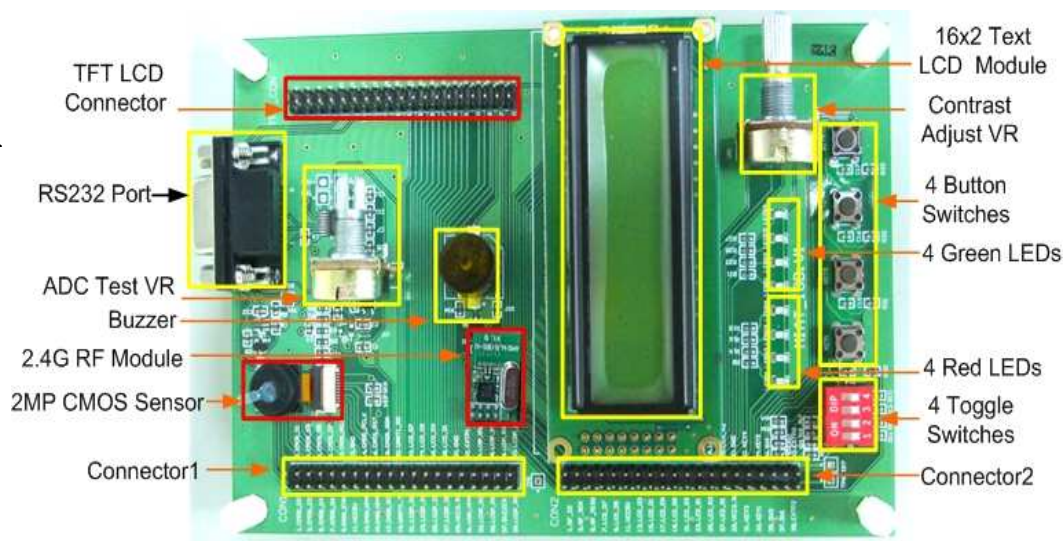






MIAT_IOB實驗週邊功能介紹

- ❑ 2MPixels CMOS Sensor模組
- ❑ 2.4G RF模組
- ❑ 2x16文字型LCD模組
(含明暗度調整電阻)
- ❑ RS232介面
- ❑ AD轉換測試模組
- ❑ 蜂鳴器
- ❑ 4個紅色LED
- ❑ 4個綠色LED
- ❑ 4個按鈕開關
- ❑ 4 P指撥開關
- ❑ TFT LCD 2.54mm連接器
- ❑ 尺寸：145x103mm



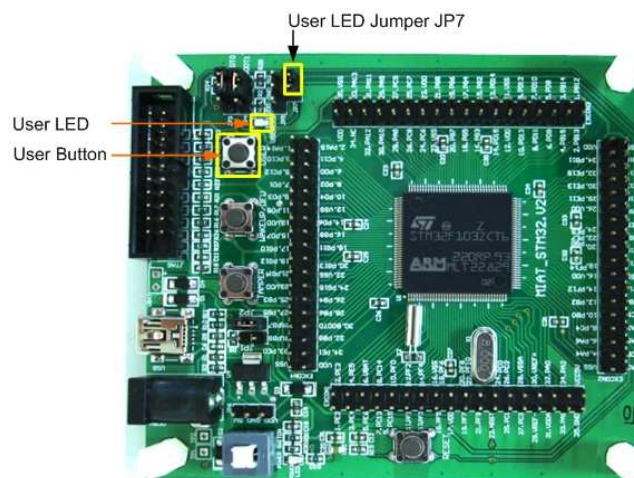
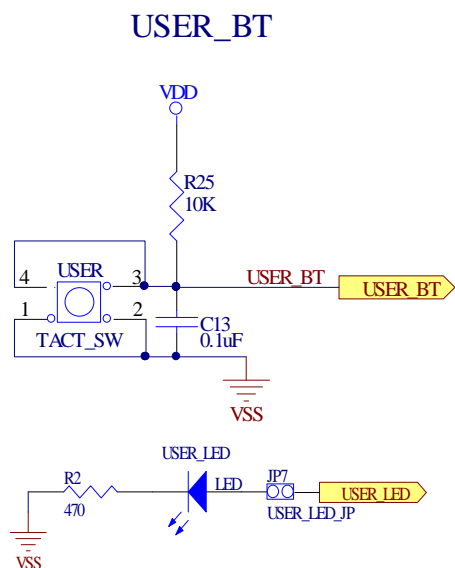


整合開發流程介紹－以基本I/O驅動為例

□ 實驗目的

- 應用MIAT_STM32實驗板上的User Button控制User LED是否閃爍，User Button未按下時User LED不閃爍，否則閃爍，達成MCU之I/O控制目的。

□ 硬體電路配置



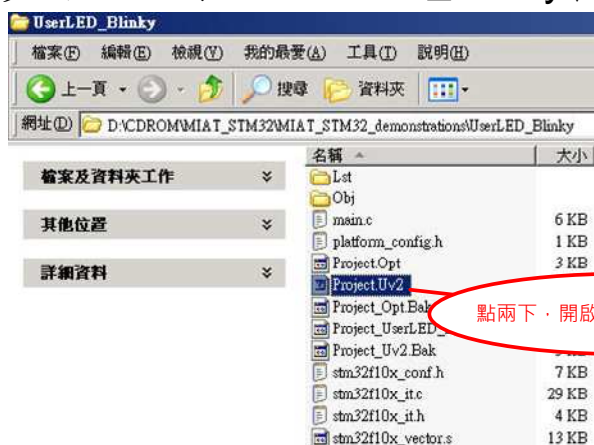
Pin Name	I/O assignment
PB9 (4.32)	User SW
PF11 (2.13)	User LED



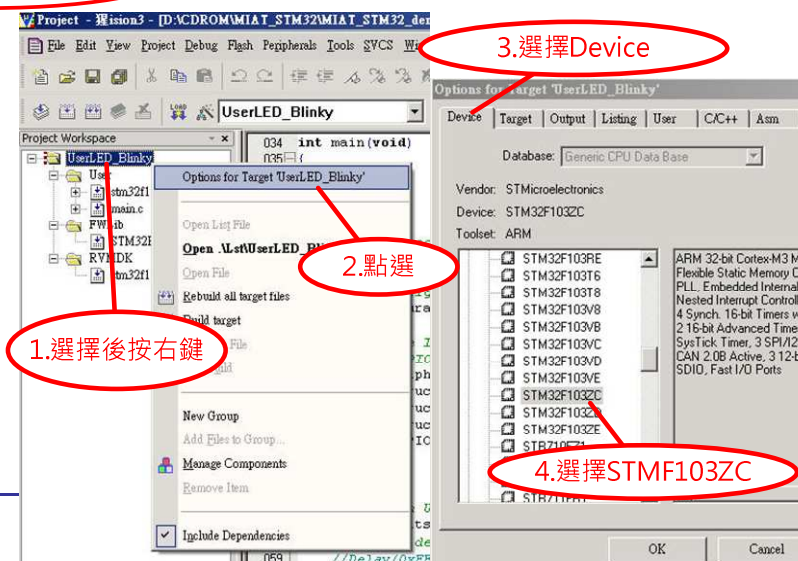
RVMDK軟體基本使用方法

- 假設使用者已安裝Keil ARM軟體工具，在此以UserLED_Blinky為範例，目錄於CDROM\MIAT_STM32_EVB_Demonstrations\UserLED_Blinky，編譯設定步驟如下：

步驟一、開啟UserLED_Blinky範例程式的專案檔



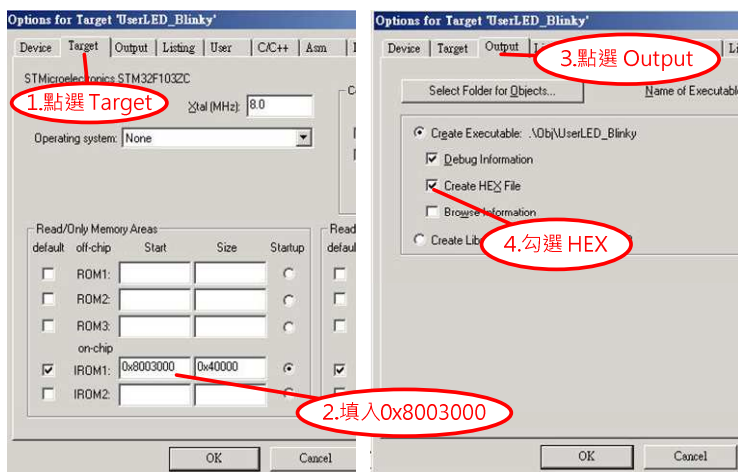
- 步驟二、設定晶片編號



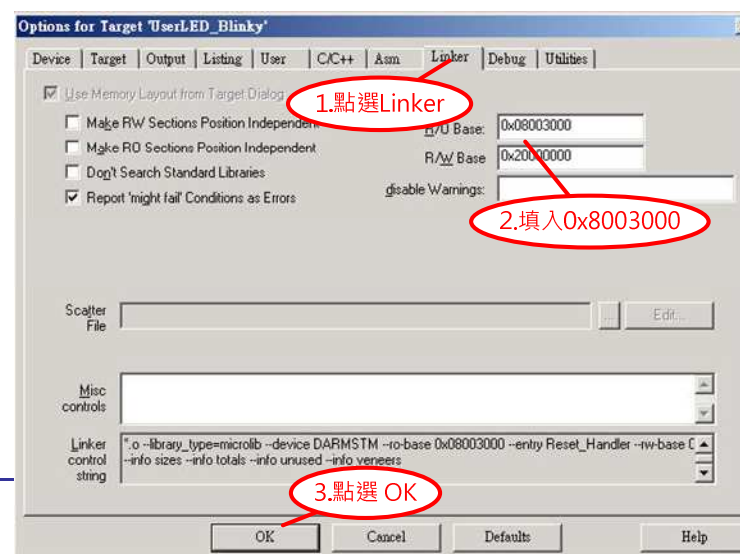


RVMDK軟體基本使用方法

- 步驟三、設定程式碼起始位置與勾選"Create HEX File"選項。



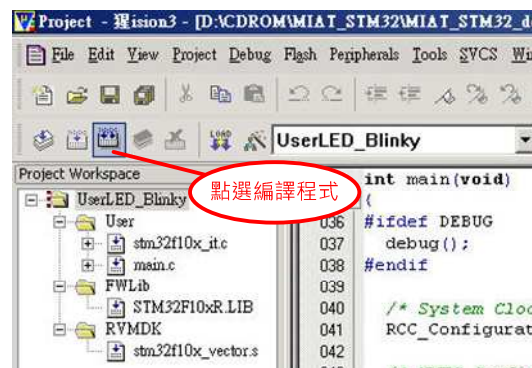
- 步驟四、設定R/O Base為0x8003000





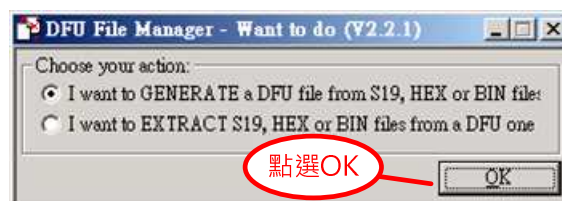
RVMDK軟體基本使用方法與.hex至.dfu轉檔

- 步驟五、重新編譯程式，產生.hex檔



- 步驟六、執行DFU File Manager(HEX轉DFU格式程式)

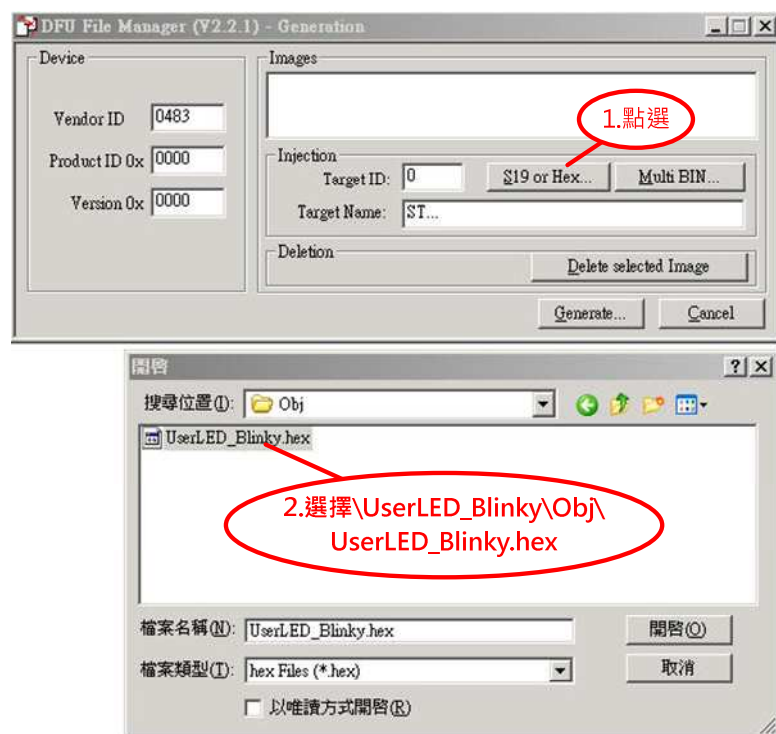
。



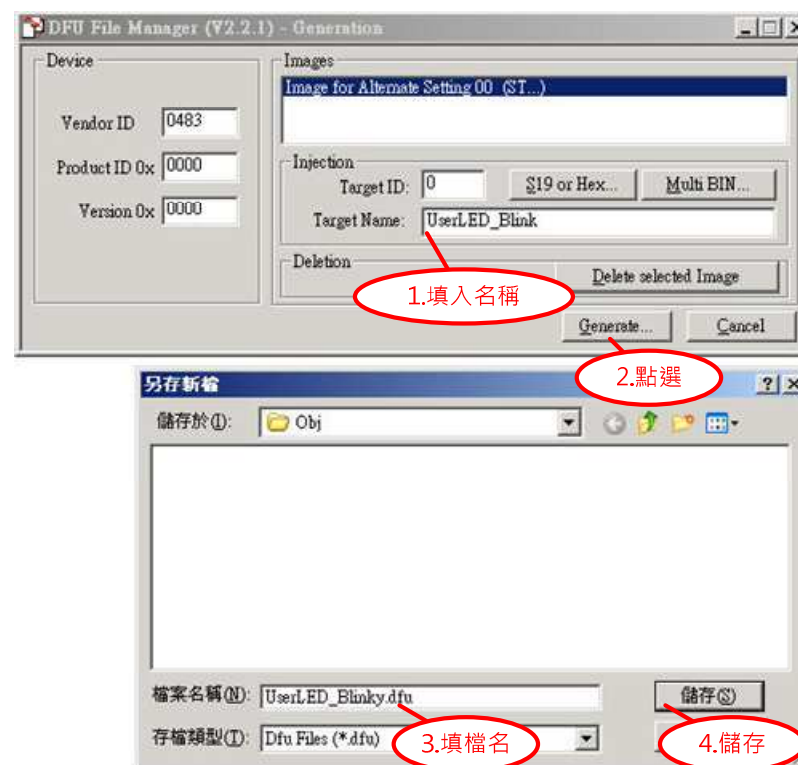


.hex至.dfu轉檔

- 步驟七、選擇來源檔(.hex) ，如 :”\obj\UserLED_Blinly.hex”。



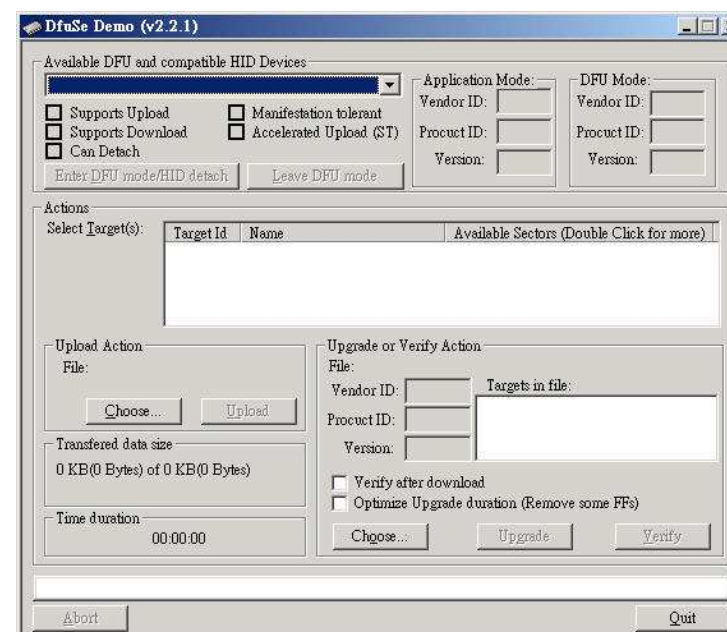
- 步驟八、產生.dfu檔，完成轉檔後即可將此檔透過DfuSe Demonstration 下載至實驗板。





DFU韌體燒錄

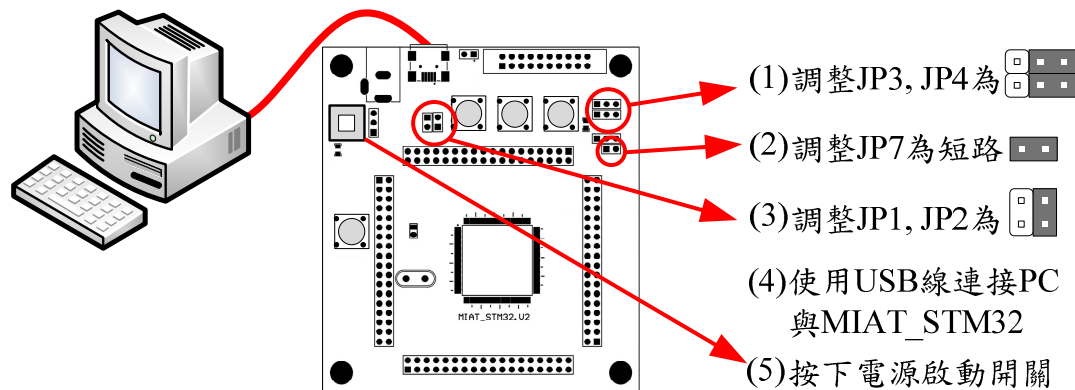
- ❑ 步驟一、安裝DfuSe_Demo_V2.2.1_Setup.exe，此檔案的目錄位置在光碟內的”\MIAT_STM32\MIAT_STM32_DfuSe\”。
- ❑ 步驟二、執行DfuSe Demonstration。



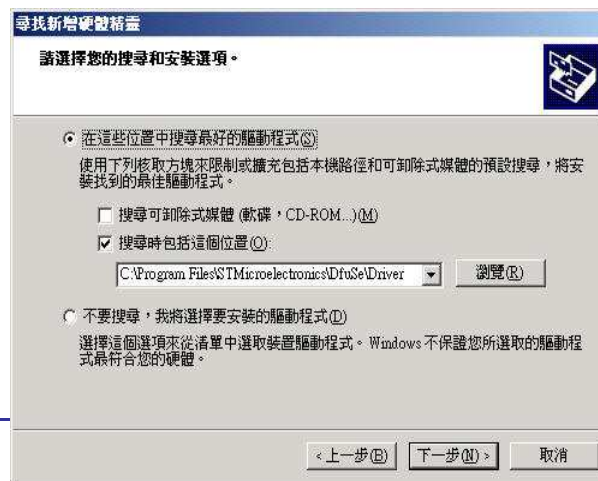


DFU韌體燒錄

□ 步驟三、硬體環境設定



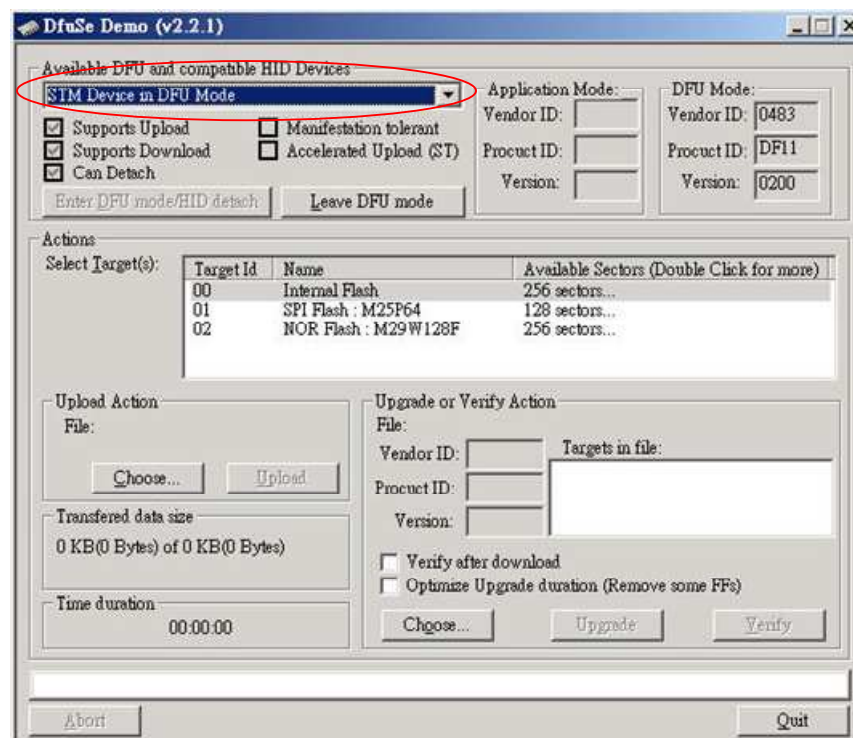
□ 步驟四、當實驗板第一次連接PC時，需安裝USB驅動程式，驅動程式目錄為C:\Program Files\STMicroelectronics\DfuSe\Driver。





DFU韌體燒錄

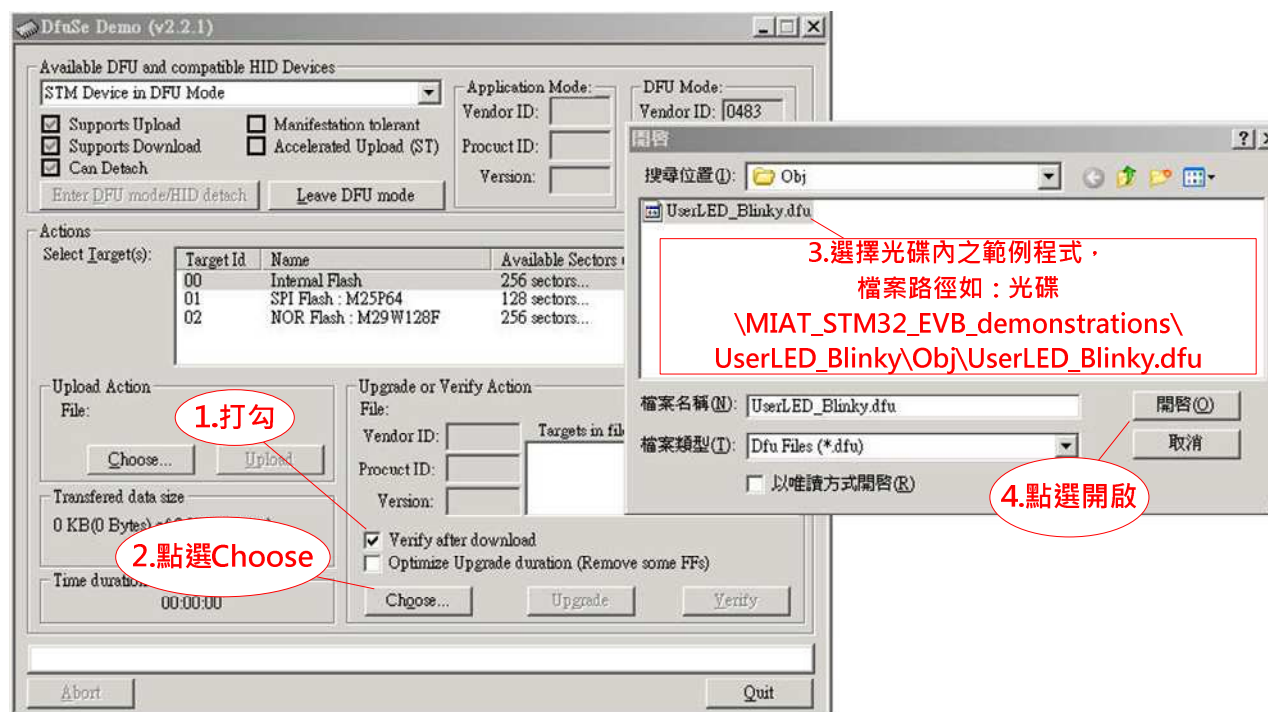
- 步驟五、同時按下RESET與DFU兩個按鈕，然後先放開RESET按鈕再放開DFU按鈕，DfuSe Demonstration視窗如下圖出現STM Device in DFU Mode選項。





DFU韌體燒錄

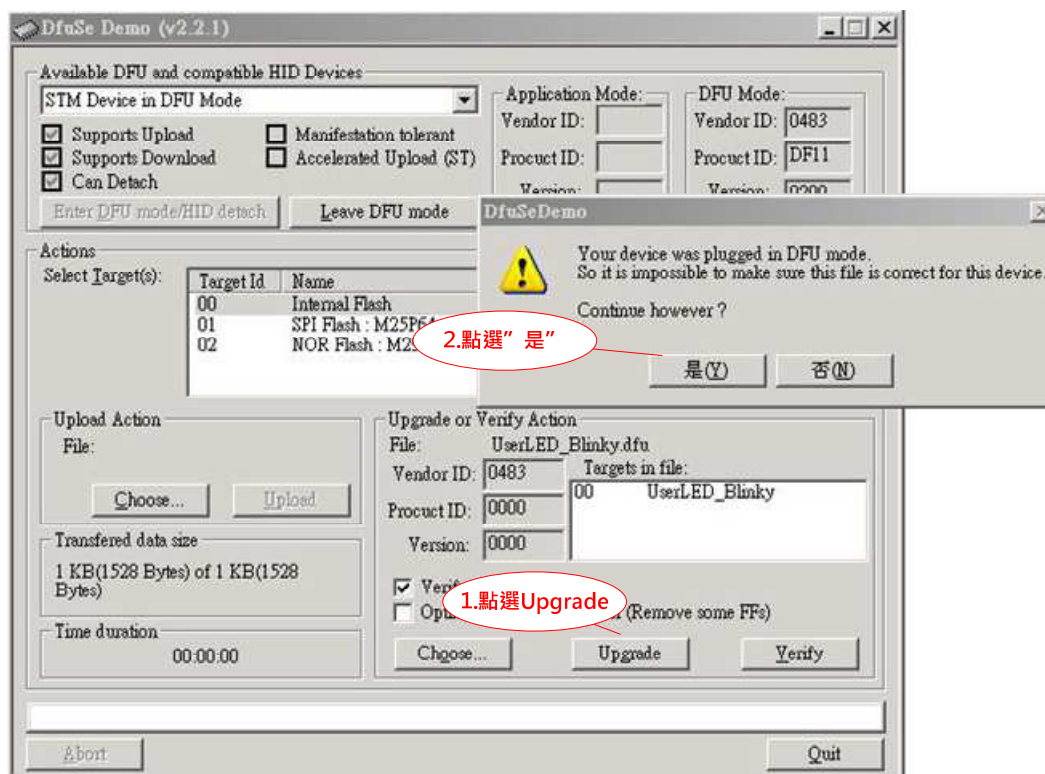
- ❑ 步驟六、開啟範例檔"UserLED_Blinky.dfu"。





DFU韌體燒錄

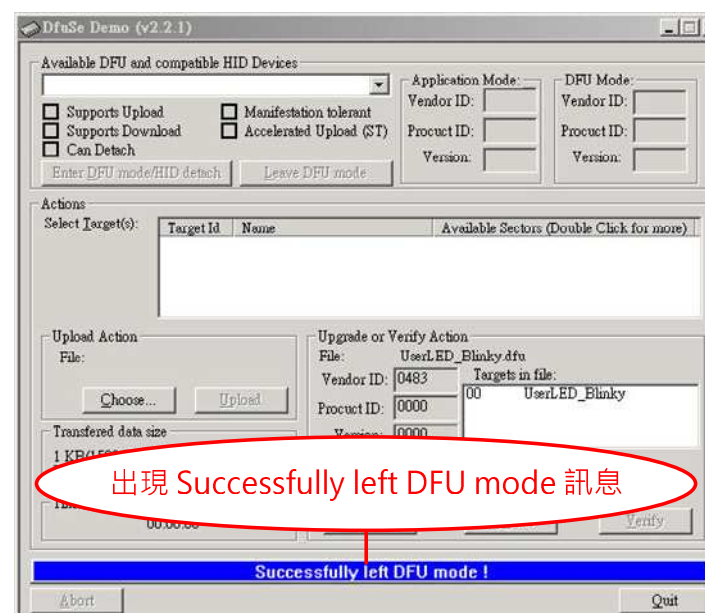
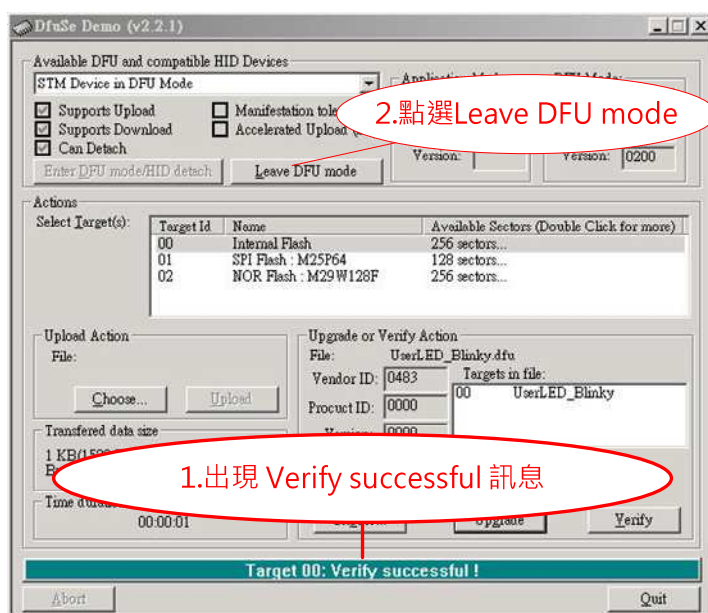
- ❑ 步驟七、執行軟體更新(下載)。





DFU韌體燒錄

- ❑ 步驟八、離開DFU 模式，範例程式開始動作，UserLED閃爍，可測試按下User Button，觀察User LED閃爍情形。





練習一、開發流程練習

□ 步驟一、修改程式碼如下

```
int main(void)
{
    ...
    while (1)
    {
        if(GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_9)==1)
        {
            /* Turn on User LED */
            GPIO_SetBits(GPIOF, GPIO_Pin_11);
            /* Insert delay */
            //減少Delay //Delay(0xFFFF); Delay(0xFFFF); Delay(0xFFFF); Delay(0xFFFF);
            Delay(0xFFFF); Delay(0xFFFF);

            /* Turn off User LED */
            GPIO_ResetBits(GPIOF, GPIO_Pin_11);
            /* Insert delay */
            //減少Delay //Delay(0xFFFF); Delay(0xFFFF); Delay(0xFFFF); Delay(0xFFFF);
            Delay(0xFFFF); Delay(0xFFFF);
        }
    }
}
```

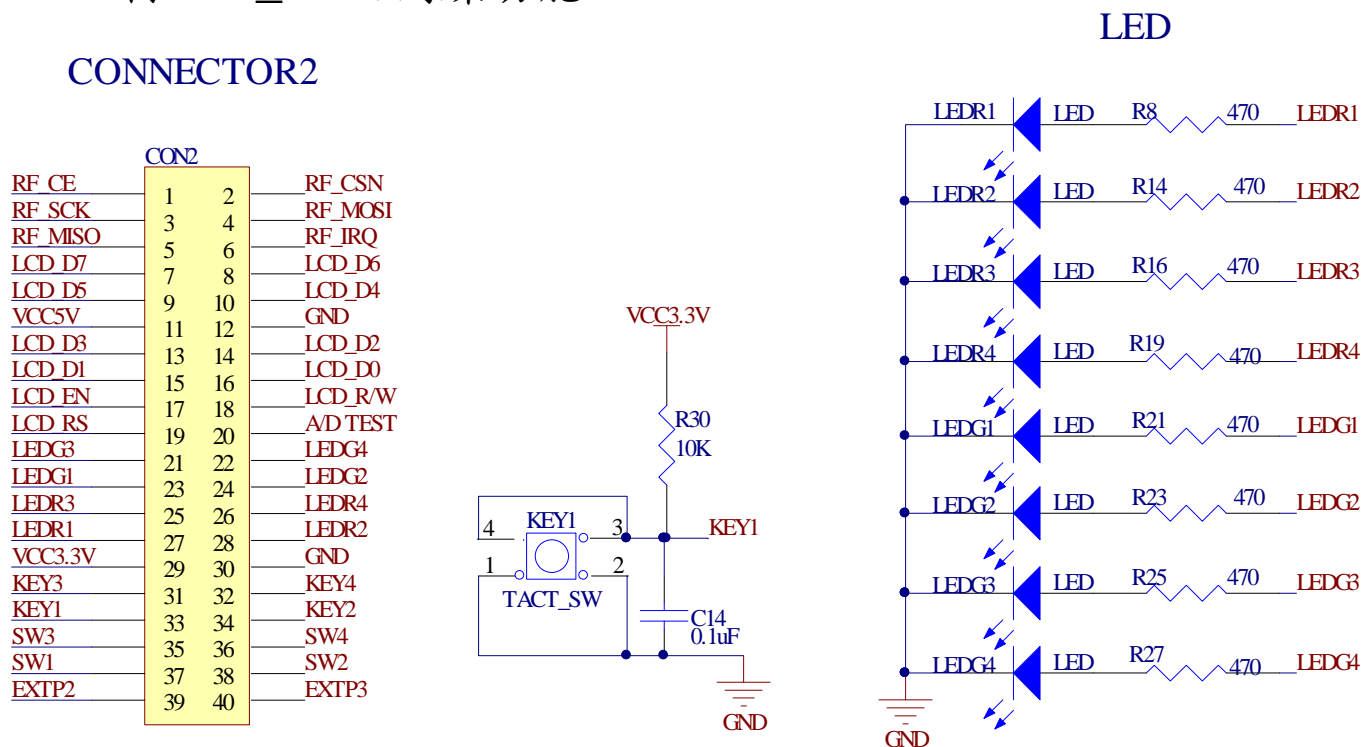
□ 步驟二、重新編譯、轉檔與燒錄至實驗板後重新執行，觀察是否為預期之結果。



練習二、軟硬體電路設計修改

□ 步驟一、硬體電路配置

- 將練習一修改為PF12連接LED_R1，PB13連接KEY1。使用KEY1控制LED_R1之閃爍功能。

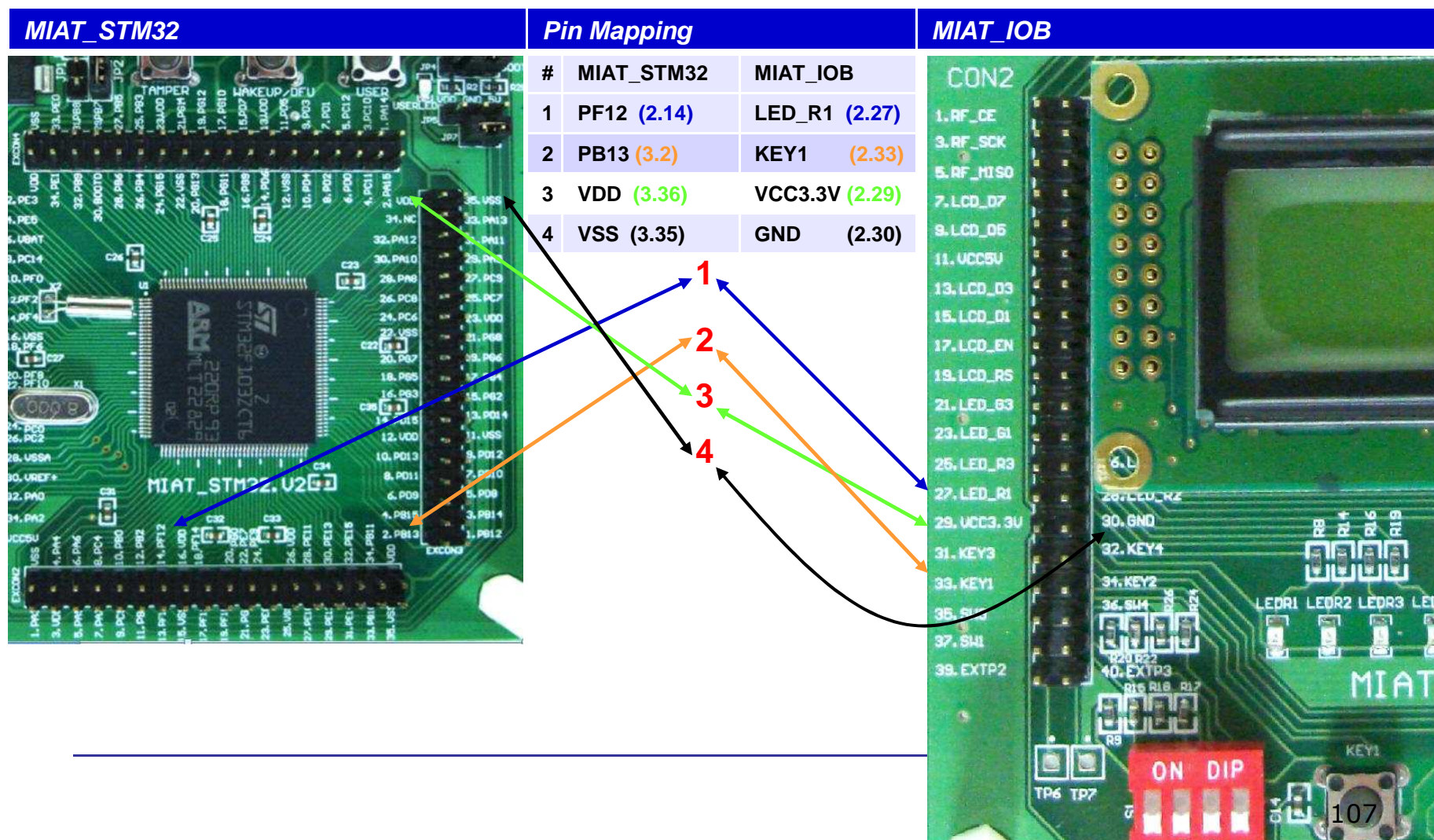


MIAT_IOB之KEY1與LED之電路圖



練習二、軟硬體電路設計修改

□ 線路連接





練習二、軟硬體電路設計修改

- 步驟二、修改程式碼後，重新編譯執行。以下為修改完成之程式碼。

```
//Example 2
// Configure PB13 (Key1) *****
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);

// Configure PF12 (LED_R1) *****
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOF, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOF, &GPIO_InitStructure);

while (1)
{
    if(GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_13)==1)
    {
        // Turn on LED_R1
        GPIO_SetBits(GPIOF, GPIO_Pin_12);
        // Insert delay
        Delay(0xFFFFF);      Delay(0xFFFFF);

        // Turn off LED_R1
        GPIO_ResetBits(GPIOF, GPIO_Pin_12);
        // Insert delay
        Delay(0xFFFFF);      Delay(0xFFFFF);
    }
}
```

Q & A



WU-YANG
Technology Co., Ltd.