

# 組合語言與微處理機實習 實驗報告

## LAB 3

### 撰寫 C 語言讓 ZedBoard 執行程式且控制 LED 燈

106/10/13

學生一：范真瑋

學生二：陳奕元

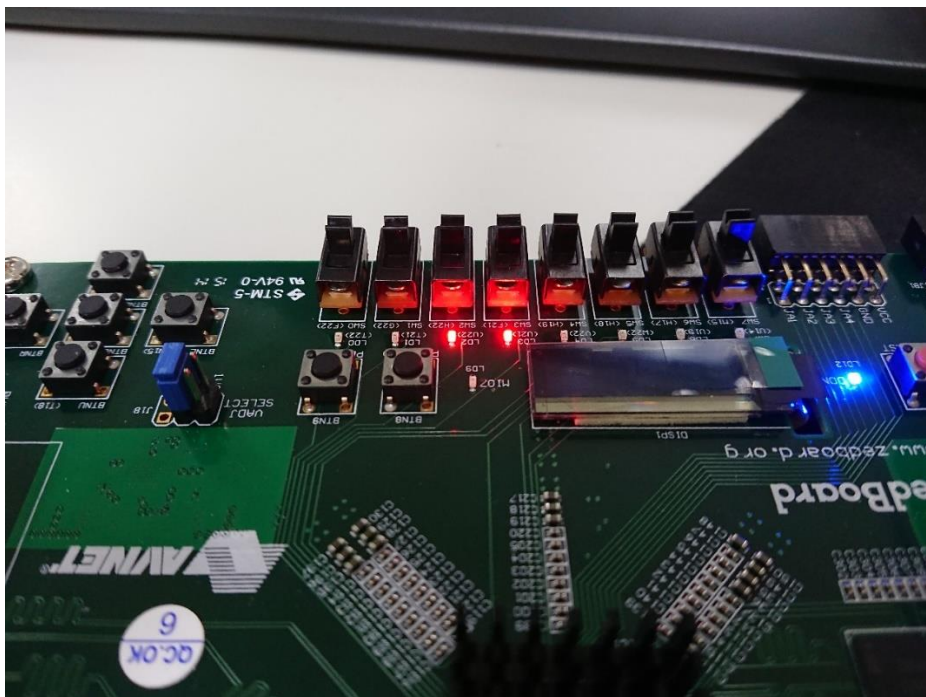
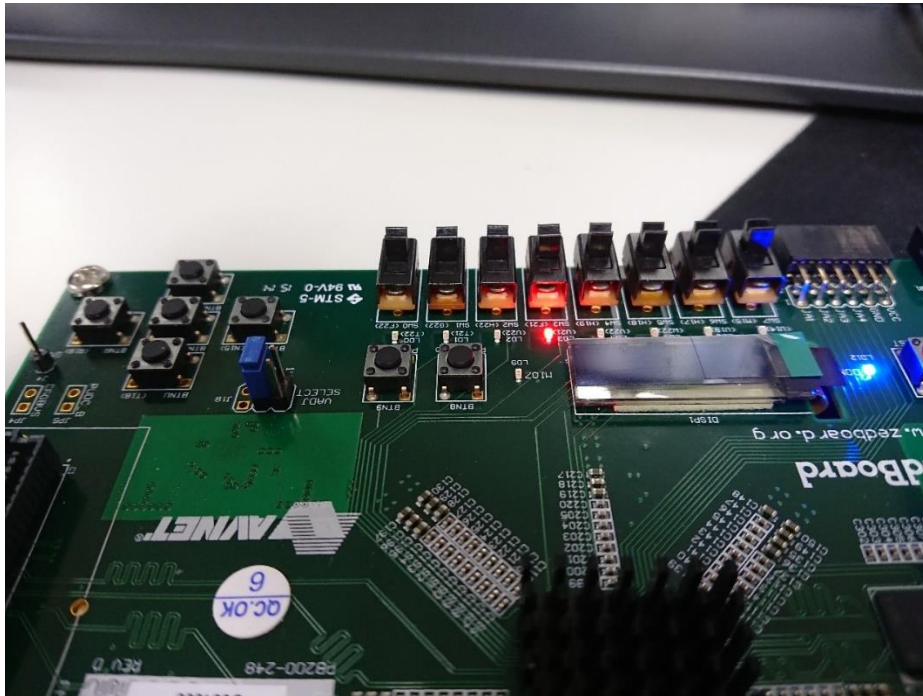
學生三：林建業

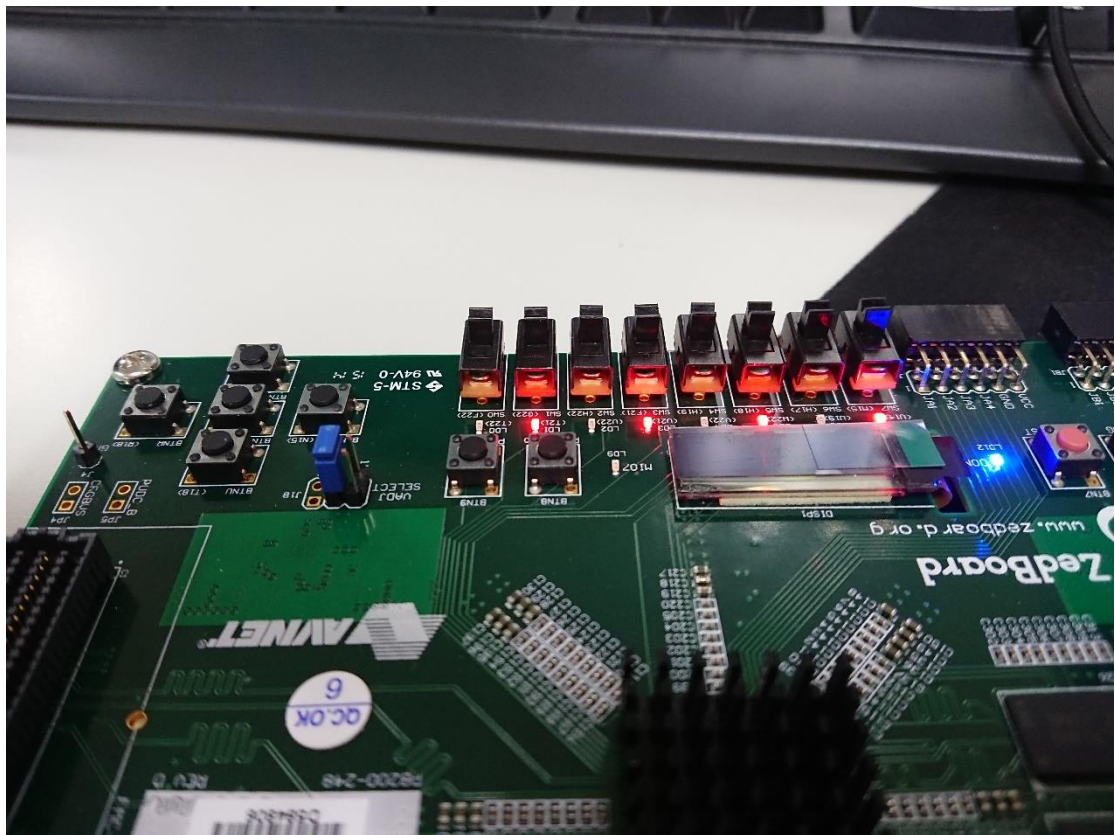
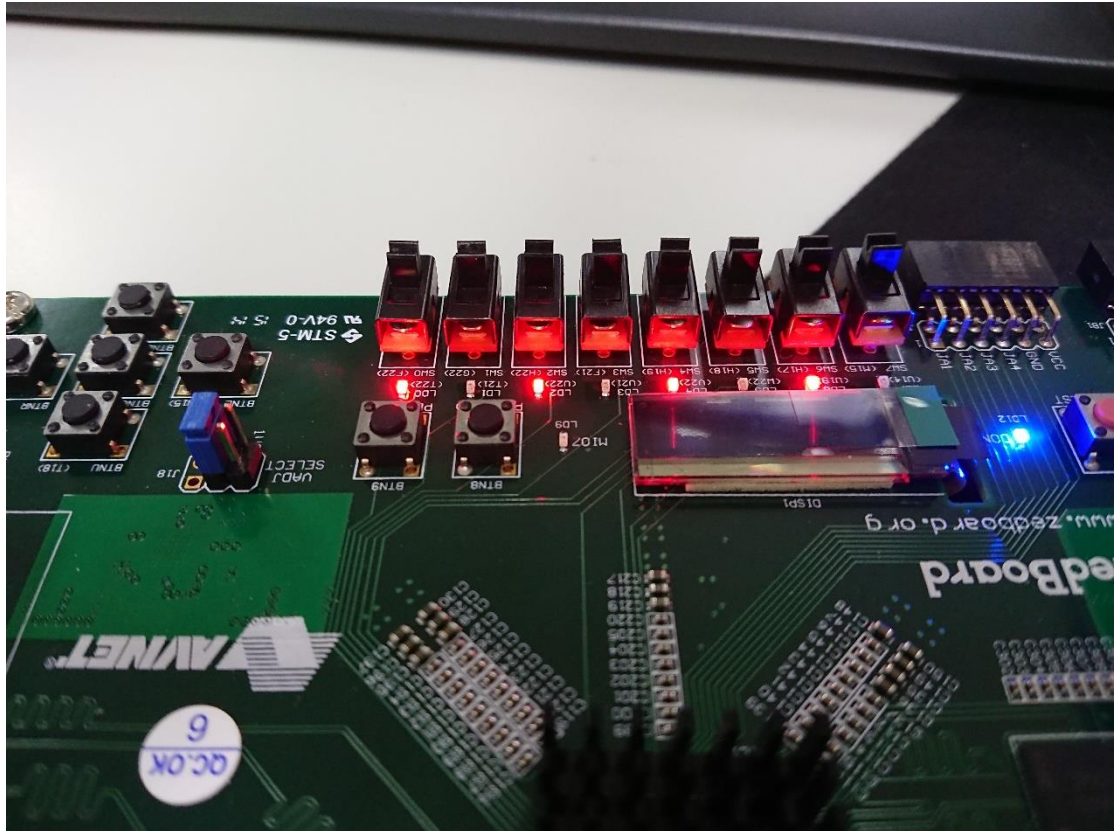
# 練習題

## 1. 目的

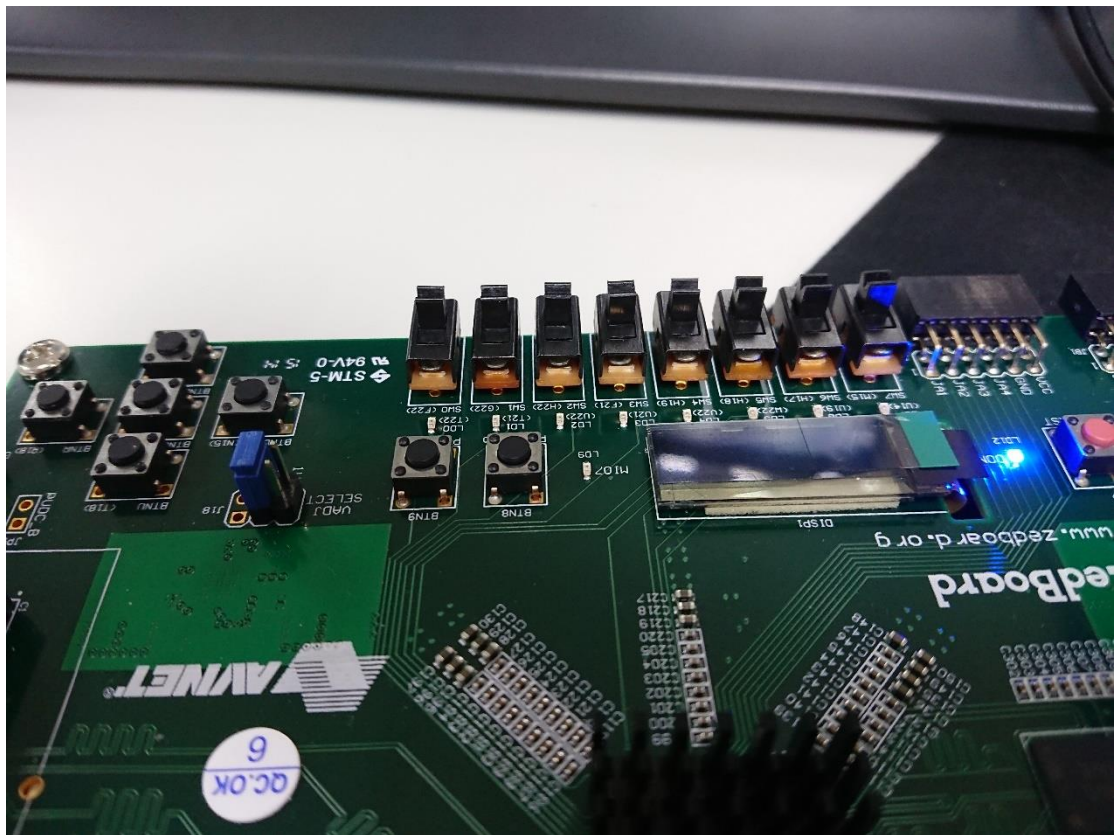
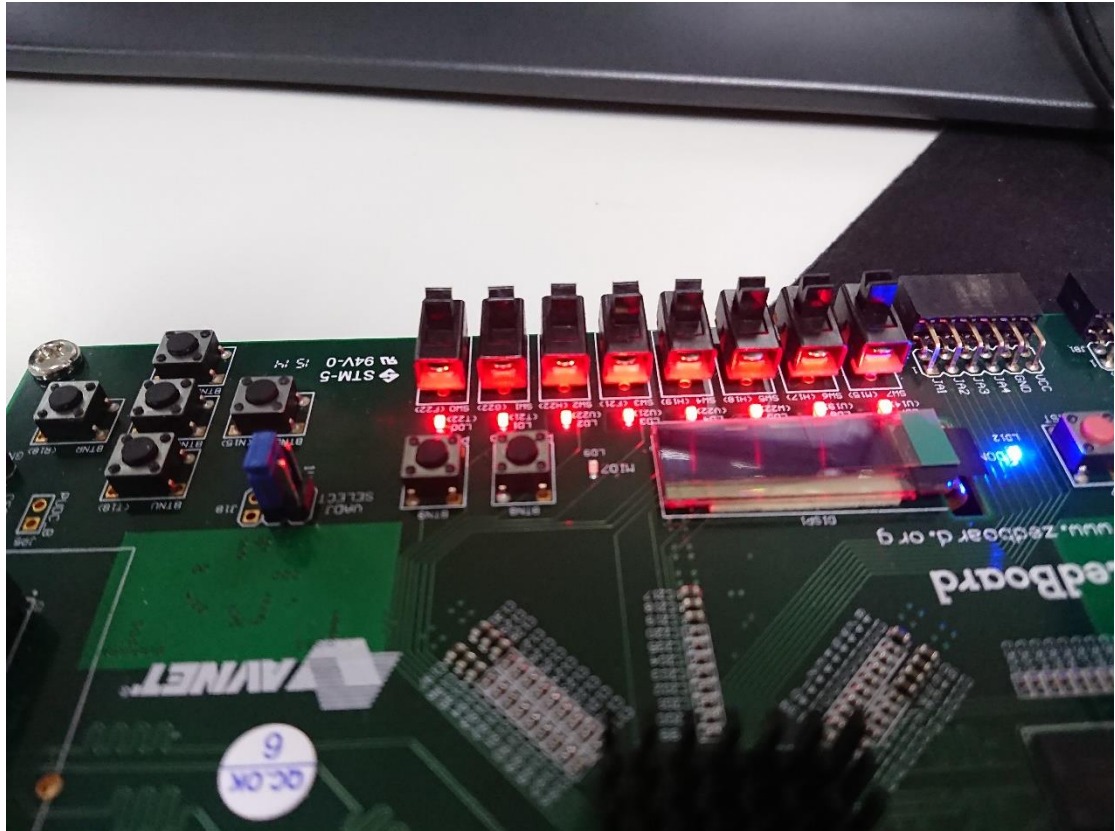
使用 CPU 控制 GPIO，自由發揮做出創意的 LED 燈。

## 2. 實驗結果









Youtube 影片連結：  
<https://youtu.be/SEnZ9XoBPjk>

### 3. 遇到的問題與解決方法

(1) 練習 for 迴圈完成九九乘法表時，有時 terminal 上會顯示多餘的換行  
解決方法：更改 printf 的輸出格式

(2) 裝置管理員中沒有看到如投影片上的 USB Serial Port

解決方法：因為沒有插上 UART 的接線，與電腦連接後即可

(3) 一開始不知如何控制 LED 亮燈位置

解決方法：助教向大家仔細講解後，我們便學會利用程式碼控制

### 4. 心得與討論

這一次的實驗我們使用了 SDK，

在完成基本題後，我們組員一起集思廣益想出了一套 LED 燈的變化。

首先我們決定讓一顆燈從左到右( $2^0 \sim 2^7$ )來回亮燈，

於是我們寫一個迴圈讓控制燈號的變數 i 從  $2^0$  每次乘 2 一直到  $2^7$

```
int i;
for(i = 1; i <= 128; i*=2)
{
    XGpio_DiscreteWrite(&GpioOutput, LED_CHANNEL, i);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
    //
    XGpio_DiscreteClear(&GpioOutput, LED_CHANNEL, i);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
}
```

再讓它從  $2^7$  每次除以 2 一直到  $2^0$

```
for(i = 128; i >= 1; i/=2)
{
    XGpio_DiscreteWrite(&GpioOutput, LED_CHANNEL, i);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
    //
    XGpio_DiscreteClear(&GpioOutput, LED_CHANNEL, i);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
}
```

接著我們讓連續的兩顆燈依次從左至右來回亮燈

( $11_2 \rightarrow 1100_2 \rightarrow 110000_2 \rightarrow 11000000_2$ )

我們又寫一個迴圈讓變數 i 從 3 每次乘 4 一直到 192

```
for(i = 3; i <= 192; i*=4)
{
    XGpio_DiscreteWrite(&GpioOutput, LED_CHANNEL, i);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
    //
    XGpio_DiscreteClear(&GpioOutput, LED_CHANNEL, i);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
}
```

再讓它從 192 每次除以 4 一直到 3

```
for(i = 192; i >= 3; i/=4)
{
    XGpio_DiscreteWrite(&GpioOutput, LED_CHANNEL, i);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
    //
    XGpio_DiscreteClear(&GpioOutput, LED_CHANNEL, i);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
}
```

接著我們又設計讓每顆燈號以間隔的方式閃爍 10 次( $10101010_2 \Leftrightarrow 01010101_2$ )

我們用 170 和 85 來控制亮燈位置達到我們預期的亮燈方式

```
for(i = 1; i <= 10; i++)
{
    XGpio_DiscreteWrite(&GpioOutput, LED_CHANNEL, 170);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
    //
    XGpio_DiscreteClear(&GpioOutput, LED_CHANNEL, 170);
    for(Delay = 0; Delay < LED_DELAY; Delay++);

    XGpio_DiscreteWrite(&GpioOutput, LED_CHANNEL, 85);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
    //
    XGpio_DiscreteClear(&GpioOutput, LED_CHANNEL, 85);
    for(Delay = 0; Delay < LED_DELAY; Delay++);
}
```

最後我們決定用全部亮燈(11111111<sub>2</sub>)一次並熄燈來代表結束

```
XGpio_DiscreteWrite(&GpioOutput, LED_CHANNEL, 255);  
for(Delay = 0; Delay<100000000; Delay++);  
//  
XGpio_DiscreteClear( &GpioOutput, LED_CHANNEL, 255);  
for(Delay = 0; Delay<100000000; Delay++);
```

為了加快亮燈的節奏，我們還修改了 Delay 的時間為原本的一半

```
#define LED_DELAY 5000000
```