



# 微算機系統實習

## **MICROPROCESSOR SYSTEMS LAB.**

### **SPRING, 2021**

Instructor : Yen-Lin Chen(陳彥霖), Ph.D.

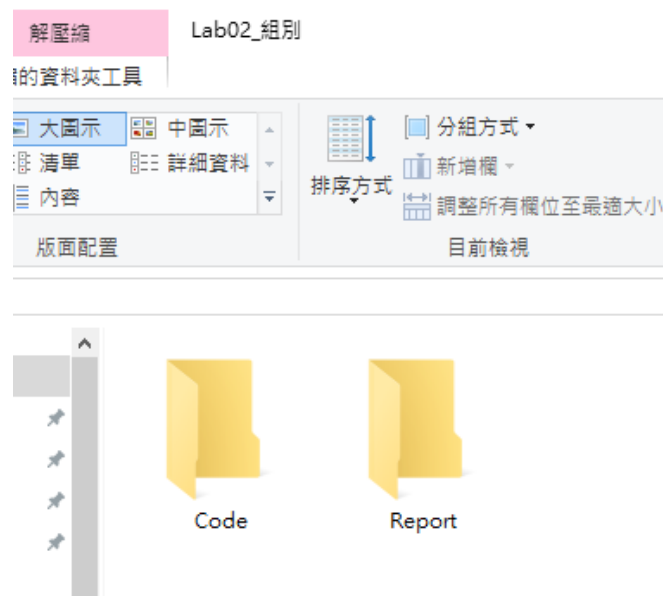
Professor

Dept. Computer Science and Information Engineering  
National Taipei University of Technology

# 作業繳交格式

- 檔名: Lab02\_組別.zip
- 其zip裡要包含如下資料夾
  1. -Code //存放專案程式碼
  2. -Report //存放報告

(不須上傳影片，但須說明程式檔各函式的作用)



# 作業繳交

- 基本繳交時間
  - 實驗：3/19 (18:00)上課結束前驗收
  - 報告：3/25 (23:59)以前上傳
  - \* 若有因為特殊原因繳交時間有變動助教會另外公布
  - 超過時間遲交每隔一週（含一週內）分數打8折，採累計連乘方式，實驗與報告打折是分開算的
    - 舉例：
      - 遲交三天 - 以遲交一週計算  $\text{<遲交的項目單獨分數>} * 0.8 = \text{該項目得到的分數}$
      - 遲交九天 - 以遲交兩週計算  $\text{<遲交的項目單獨分數>} * 0.8 * 0.8 = \text{該項目得到的分數}$
- 以上配分與注意事項有問題請聯絡助教

# 實驗二



2

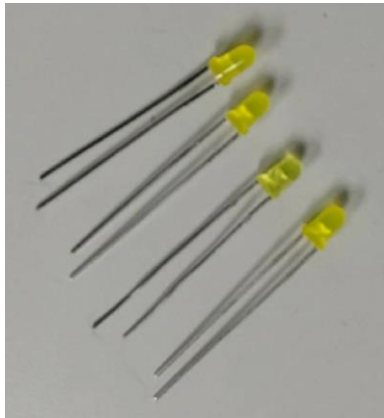
透過C/C++程式控制GPIO上的LED燈

# 本次實驗目標

- 學習如何透過GPIO與C/C++控制TX2上的LED燈。
- 了解在嵌入式系統上執行的檔案類型。
- 了解TX2 physical pin對應的sysfs filename
- 學習使用跨平台編譯工具發展嵌入式系統程式。
- 學習如何撰寫Makefile編譯程式。

# Lab2實驗說明

- 將彩虹排線插入TX2 J21腳位，4顆LED燈插入程式對應的Pin，使用C/C++寫一個程式控制GPIO上的4個LED (共70分)
- 基本要求：必須使用**跨平台編譯與Makefile進行編譯**(在電腦或Ubuntu虛擬機編譯出可在嵌入式平台上執行的執行檔，再放入嵌入式平台上執行)
  - `aarch64-linux-gnu-g++ -o <輸出的執行檔名稱> <cpp檔案名稱>`
- 可以參考上課講義的範例code。(Lecture 2 P.29-40)



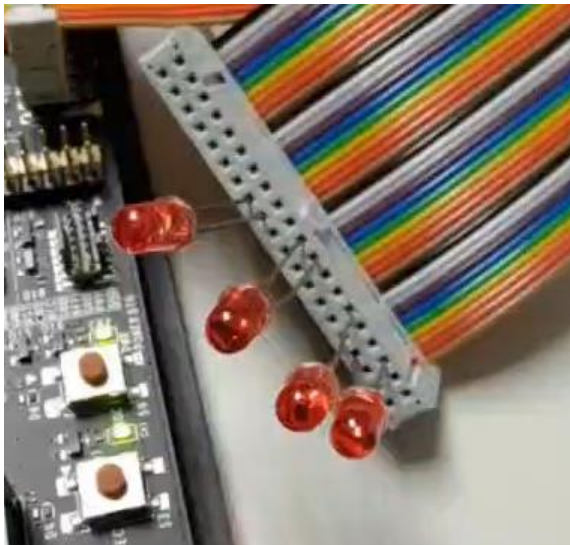
LED燈泡



彩虹排線

# Lab2實驗說明

- 程式功能說明
  - 寫一個C/C++ 程式透過輸入 **參數** 控制GPIO上的4顆LED燈狀態。
  - 1.基本題：
    - LED[編號] on/off 可以控制該編號的LED狀態，其他LED狀態不變(占40分)
    - Ex: **./L2Program LED1 on** (第一顆燈狀態改為亮，其他狀態不變)。



影片範例執行指令：

```
./L2Program LED3 on  
./L2Program LED3 off  
./L2Program LED1 on  
./L2Program LED1 off
```

# Lab2實驗說明

- 2. 進階題：
  - LED 1 & LED 2 為第一組而 LED 3 & LED 4 為第二組，第一組會同時亮或滅(同理第二組也是一樣的)，而第二組的狀態會跟第一組相反，最後可以指定閃爍次數(第一組亮完之後換第二組亮，這樣為一次閃爍) (占30分)

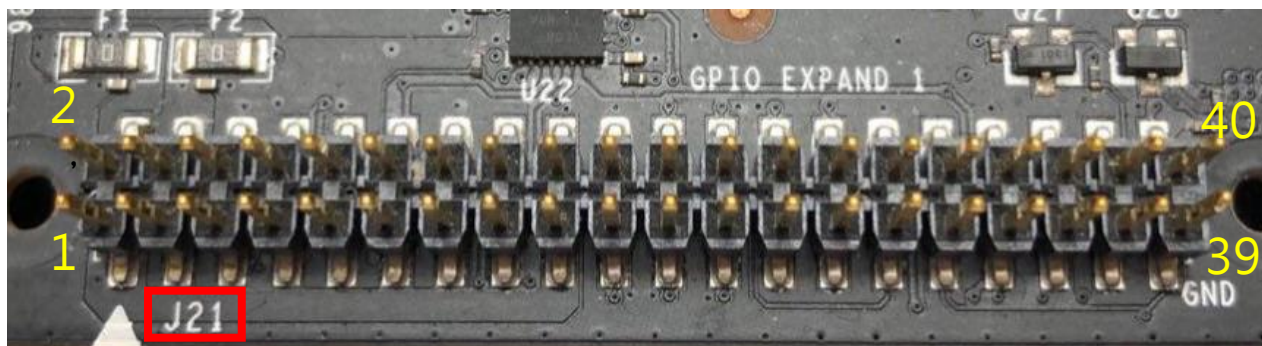


影片範例執行指令：  
`./L2Program Mode_Shine 5`  
(交替閃爍5次)



# Lab2實驗注意事項

- 原pin腳腳位如圖：



- 使用彩虹排線後，第一排跟第二排腳位相反

1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40

# TX2 的 GPIO 腳位參考(J21)

Jetson TX2 J21 Header					
Sysfs GPIO	Connector Label	Pin	Pin	Connector Label	Sysfs GPIO
	3.3 VDC Power	1	2	5.0 VDC Power	
	SDA1 General I2C Data 3.3.V, I2C Bus 1	3	4	5.0 VDC Power	
	SCL1 General I2C Clock 3.3.V, I2C Bus 1	5	6	GND	
gpio396	GPIO_GCLK Audio Master Clock (1.8/3.3.V)	7	8	TXD0 UART #0 Transmit	
	GND	9	10	RXD0 UART #0 Receive	
gpio466	GPIO_GEN0 UART #0 Request to Send	11	12	GPIO_GEN1 Audio I2S #0 Clock	gpio392
gpio397	GPIO_GEN2 Audio Code Interrupt	13	14	GND	
gpio255	GPIO_GEN3 From GPIO Expander (P17)	15	16	GPIO_GEN4 Unused	gpio296
	3.3 VDC Power	17	18	GPIO_GEN5 Modem Wake AP GPIO	gpio481
gpio429	SPI_MOSI SPI #1 Master Out/Slave In	19	20	GND	

gpio428	SPI1_MISO SPI #1 Master In/Slave Out	21	22	GPIO_GEN6 From GPIO Epander (P16)	gpio254
gpio427	SPI_SCLK SPI #1 Shift Clock	23	24	SPI_CE0_N SPI Chip Select #0	gpio430
	GND	25	26	SPI_CE1_N SPI #1 Chip Select #1	
	ID_SD General I2C #1 Data (3.3V), I2C Bus 0	27	28	ID_SC General I2C #1 Clock (3.3V), I2C Bus 0	
gpio398	GPIO5 Audio Reset (1.8/3.3.V)	29	30	GND	
gpio298	GPIO6 Motion Interrupt (3.3V)	31	32	GPIO12 Unused	gpio297
gpio389	GPIO13 AP Wake Bt GPIO	33	34	GND	
gpio395	GPIO19 AUDIO I2S #0 Left/Right Clock	35	36	GPIO16 UART #0 Clear to Send	gpio467
gpio388	GPIO26 (3.3V)	37	38	GPIO20 Audio I2S #0 Data in	gpio394
	GND	39	40	GPIO21 Audio I2S #0 Data in	gpio393