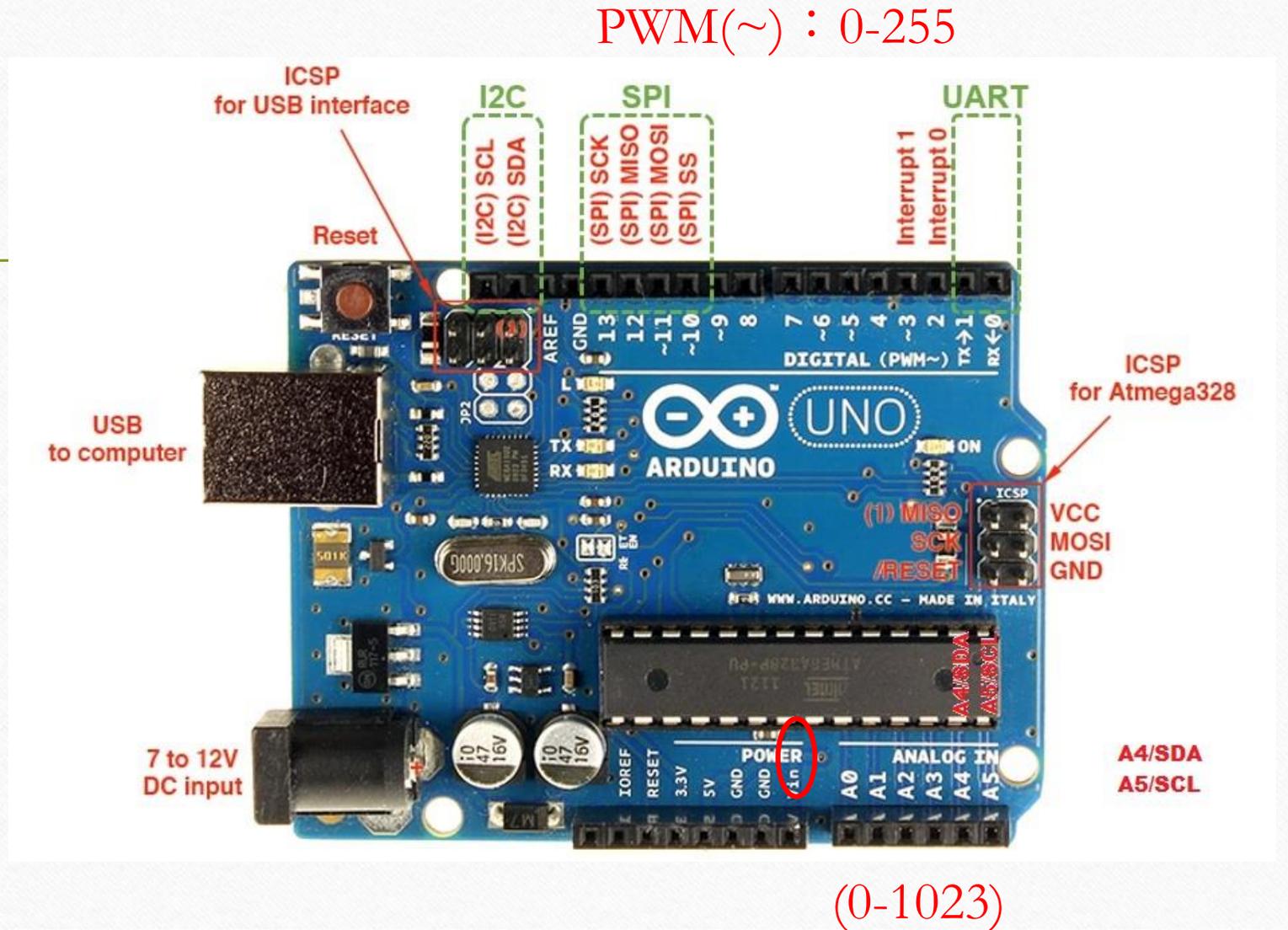


# 110新泰創資

2022.5.21

# 硬體介紹(備用)

## Arduino UNO



# Mega2560

## Mega 2560 Pinout

DC Power Jack 7-12VDC Input  
2.1mm x 5.5mm Male Center Positive

USB-B Port To Computer

No Connection  
I/O Reference Voltage for shields  
Reset Input  
3.3V Output @ 50mA  
5V Output or Input  
Ground  
Ground  
7-12V Output or Input

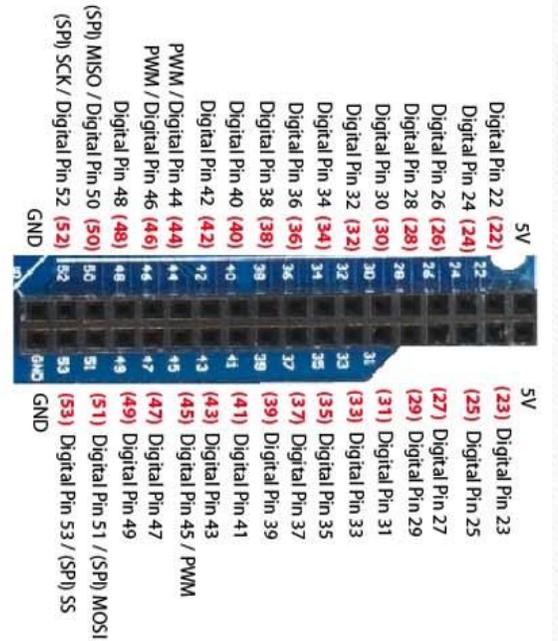
Analog Pin 0 / Digital Pin 54 (A0)  
Analog Pin 1 / Digital Pin 55 (A1)  
Analog Pin 2 / Digital Pin 56 (A2)  
Analog Pin 3 / Digital Pin 57 (A3)  
Analog Pin 4 / Digital Pin 58 (A4)  
Analog Pin 5 / Digital Pin 59 (A5)  
Analog Pin 6 / Digital Pin 60 (A6)  
Analog Pin 7 / Digital Pin 61 (A7)

Analog Pin 8 / Digital Pin 62 (A8)  
Analog Pin 9 / Digital Pin 63 (A9)  
Analog Pin 10 / Digital Pin 64 (A10)  
Analog Pin 11 / Digital Pin 65 (A11)  
Analog Pin 12 / Digital Pin 66 (A12)  
Analog Pin 13 / Digital Pin 67 (A13)  
Analog Pin 14 / Digital Pin 68 (A14)  
Analog Pin 15 / Digital Pin 69 (A15)



Red numbers in parenthesis are the name to use when referencing that pin.  
Analog pins are referenced as A0 thru A15 even when using as digital I/O

- (I2C) SCL - Serial Clock
- (I2C) SDA - Serial Data
- Analog Reference Voltage
- Ground
- (13) Digital Pin 13 / PWM / Connected to on-board LED
- (12) Digital Pin 12 / PWM
- (11) Digital Pin 11 / PWM
- (10) Digital Pin 10 / PWM
- (9) Digital Pin 9 / PWM
- (8) Digital Pin 8 / PWM
- (7) Digital Pin 7 / PWM
- (6) Digital Pin 6 / PWM
- (5) Digital Pin 5 / PWM
- (4) Digital Pin 4 / PWM
- (3) Digital Pin 3 / PWM / Ext Int 5
- (2) Digital Pin 2 / PWM / Ext Int 4
- (1) Digital Pin 1 / Serial Port 0 TXD (Main Serial Port)
- (0) Digital Pin 0 / Serial Port 0 RXD (Main Serial Port)
- (14) Digital Pin 14 / Serial Port 3 TXD
- (15) Digital Pin 15 / Serial Port 3 RXD
- (16) Digital Pin 16 / Serial Port 2 TXD
- (17) Digital Pin 17 / Serial Port 2 RXD
- (18) Digital Pin 18 / Serial Port 1 TXD / Ext Int 3
- (19) Digital Pin 19 / Serial Port 1 RXD / Ext Int 2
- (20) Digital Pin 20 / (I2C) SDA / Ext Int 1
- (21) Digital Pin 21 / (I2C) SCL / Ext Int 0



# Arduino pro-mini

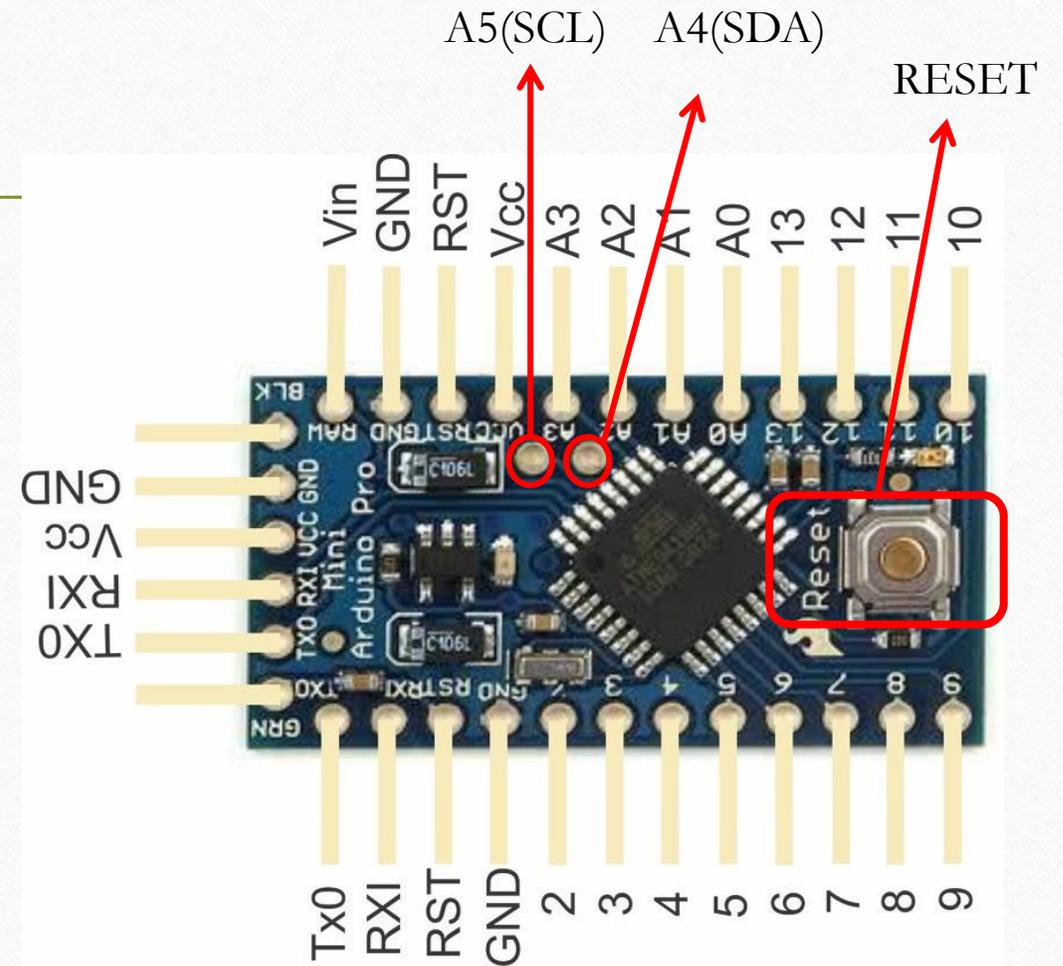
燒錄注意事項1：選定處理器

開發板：Arduino pro or pro mini

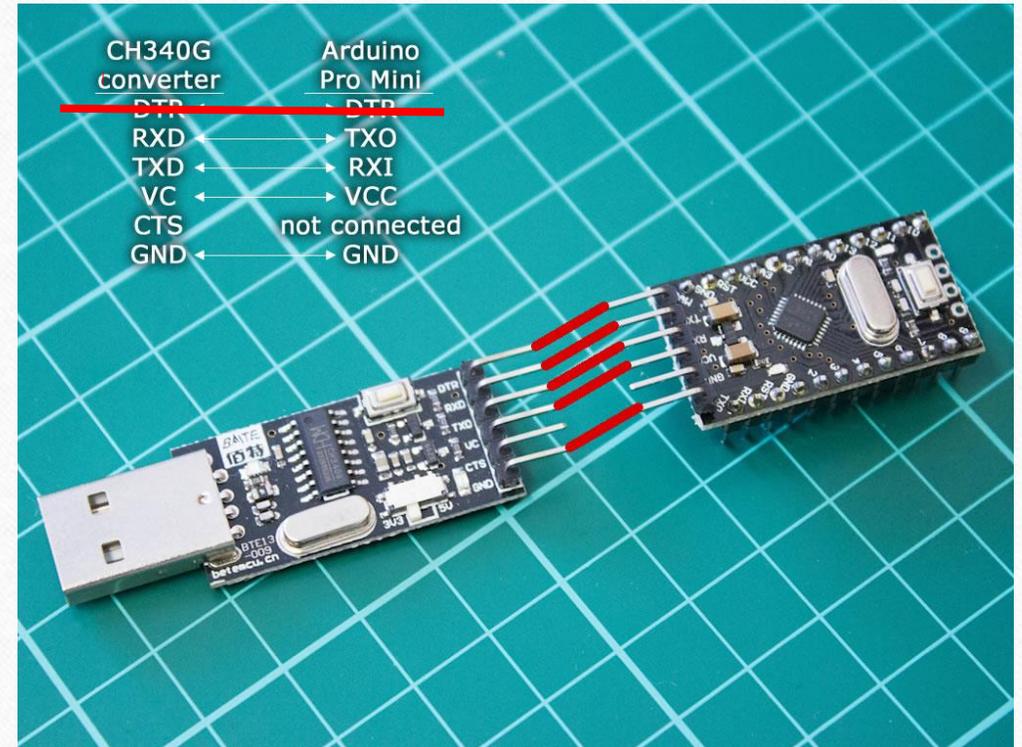
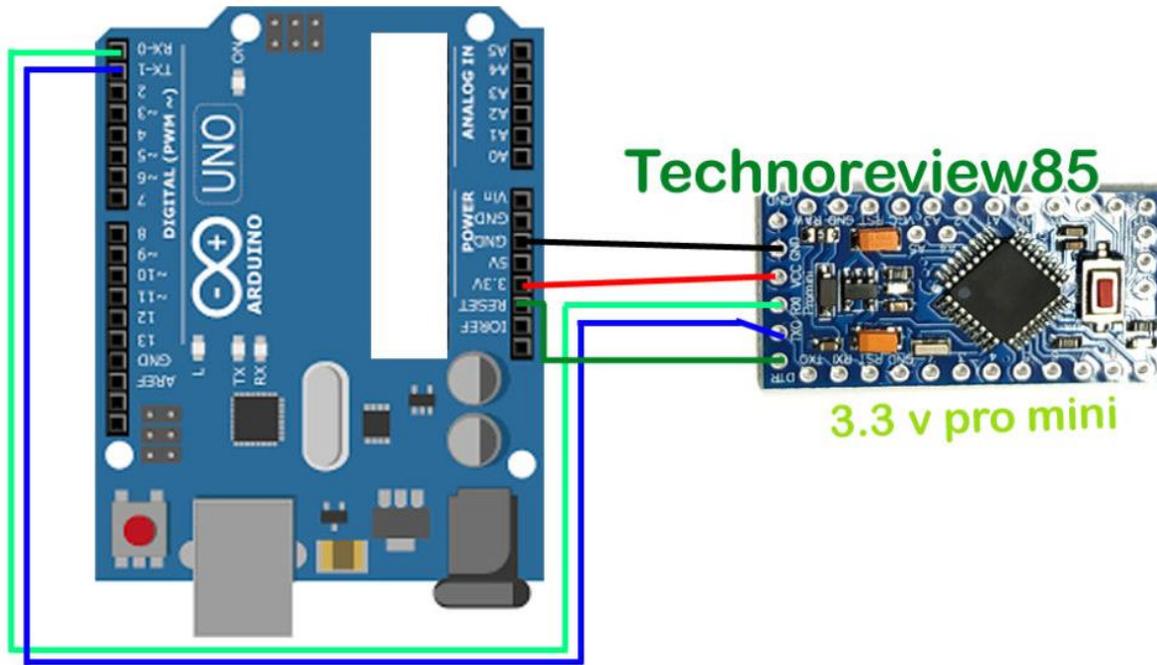
處理器：ATmega328P(5V,16MHz)

port：自訂

先空測試

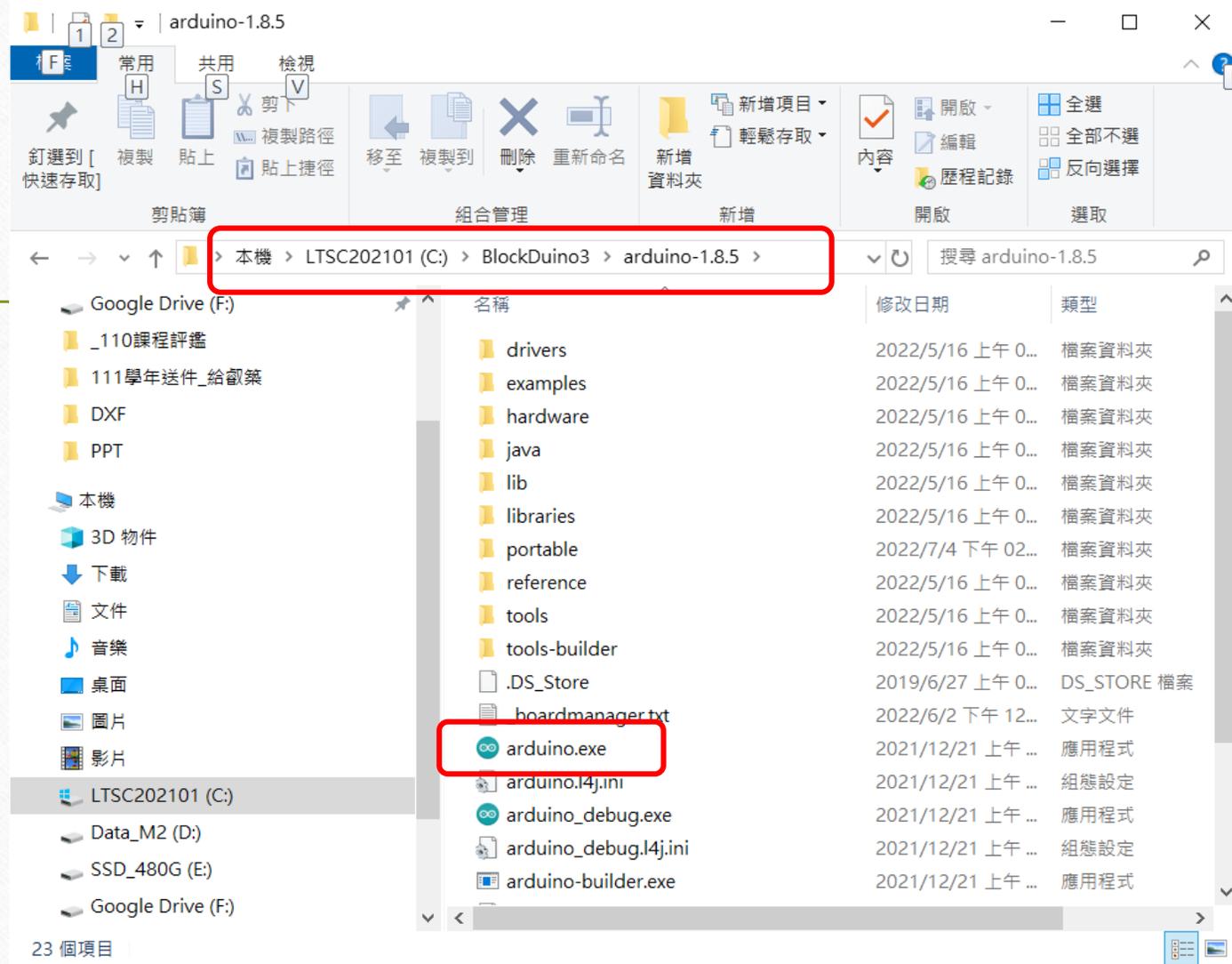


# Arduino pro-mini燒錄接線圖



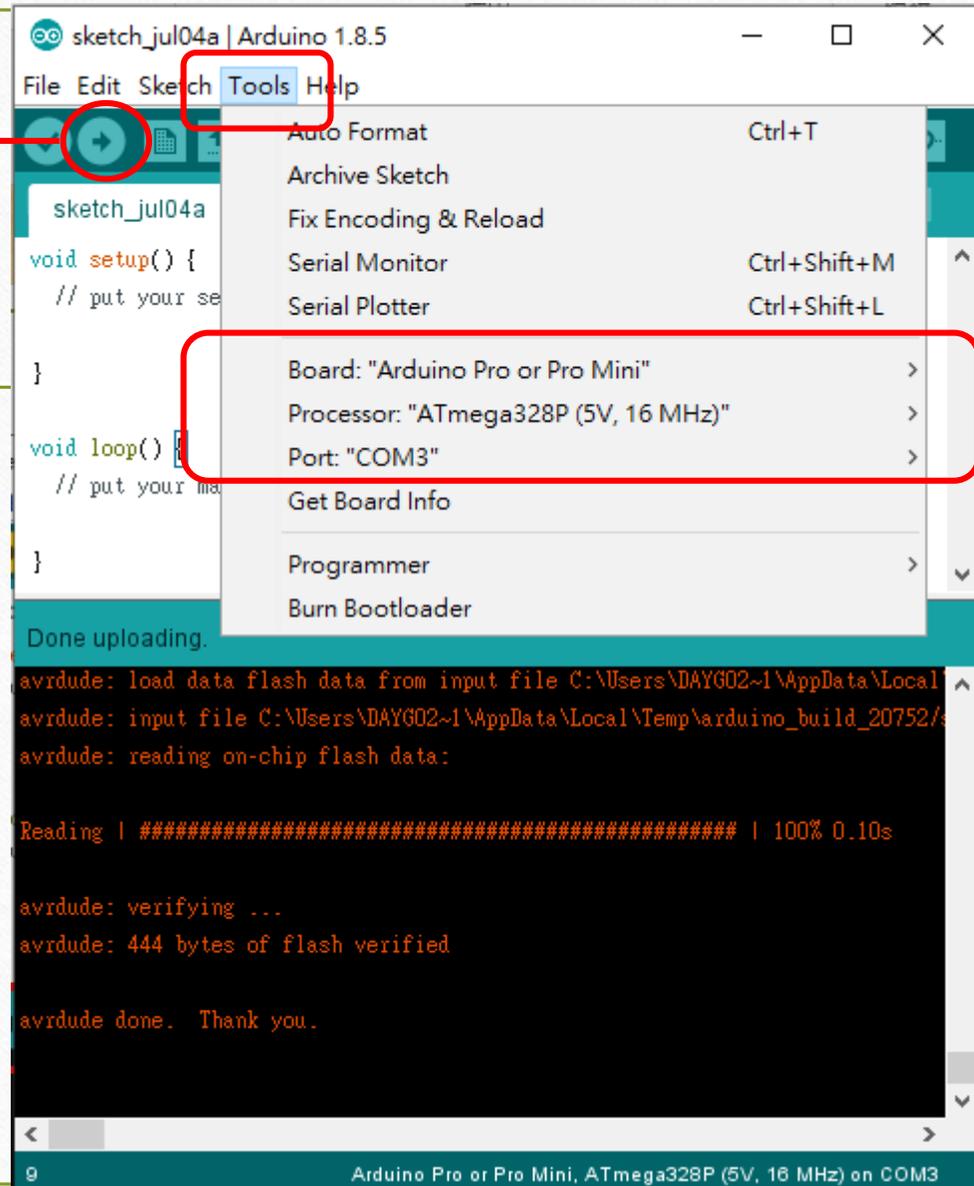
# 修改預設燒錄程式

- BD3預設是Linkit7697
- 請找到BD3資料夾內的Arduino，  
F1版本的Arduino是1.8.5，  
F2版本的Arduino是1.8.19。



- 依照右圖修改  
port請依照自己的編號

上傳



# 空測試

燒錄注意事項2：  
上傳前先按住RESET鍵  
出現上傳(Uploading)，  
再放開RESET鍵。

sketch\_jul04a | Arduino 1.8.5

File Edit Sketch Tools Help

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Uploading...

Done uploading.

上傳完成

avrduide: version 0.3-20190619  
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicr  
Copyright (c) 2007-2014 Joerg Wunsch  
  
System wide configuration file is "C:\Blocklyduino3F  
  
Using Port : COM3  
Using Programmer : arduino  
Overriding Baud Rate : 57600

avrduide: load data flash data from input file C:\Users\DAYG02~1\AppData\Local  
avrduide: input file C:\Users\DAYG02~1\AppData\Local\Temp\arduino\_build\_20752/  
avrduide: reading on-chip flash data:  
Reading | ##### | 100% 0.10s  
avrduide: verifying ...  
avrduide: 444 bytes of flash verified  
avrduide done. Thank you.

1 Arduino Pro or Pro Mini, ATmega328P (5V, 16 MHz) on COM3

# BD3-Arduino pro mini上傳注意事項

按住RESET

放開RESET

```
燒錄狀態  
Launching Arduino.exe...  
Loading configuration...  
Initializing packages...  
Preparing boards...  
Verifying...  
Build options changed, rebuilding all  
Archiving built core (caching) in: C:\Users\DAYG02~1\AppData\Local\Temp\arduino_cache_431477\core\cor  
ffff04d804_a  
Uploading...  
Arduino.exe exited with 0 and null
```

開發板 Arduino 初始化

- 設定紅外線接收器腳位 4
- 宣告 全域 整數 (int) bullet 值 30
- 宣告 全域 整數 (int) delay 值 200
- 宣告 全域 字串 (String) GunShow 值 “ ”
- 宣告 全域 整數 (int) GunType 值 0
- 初始化 液晶顯示 1602 訊號: I2C I2C位址 0x27
- 液晶顯示 背光 開

函數 single\_Shot 宣告參數

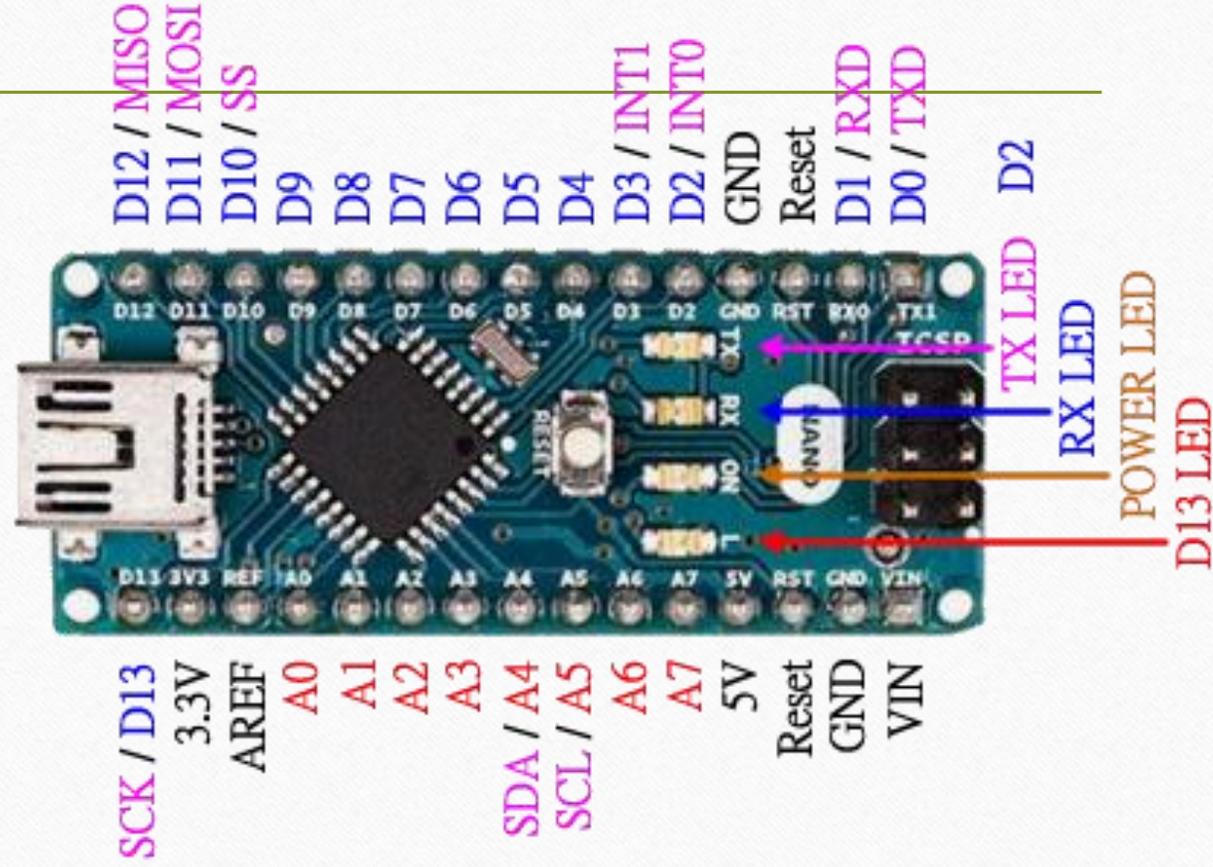
如果 數位讀取 Pin 5 且

執行

- 數位寫入 Pin 7 狀態 高電位
- 數位寫入 Pin 6 狀態 高電位
- 修改 bullet 增量 -1
- 呼叫 ShowLCD 參數值
- 延遲毫秒 30
- 數位寫入 Pin 7 狀態 低電位

# Arduino nano

老師現有的須使用  
Oldbootloader燒錄



# 程式範例

---

- 靶：

[http://163.26.205.2/~htaes/110%E5%89%B5%E9%80%A0%E5%8A%9B%E8%B3%87%E5%84%AA/bd3/target\\_10ok.zip](http://163.26.205.2/~htaes/110%E5%89%B5%E9%80%A0%E5%8A%9B%E8%B3%87%E5%84%AA/bd3/target_10ok.zip)

- 槍：

[http://163.26.205.2/~htaes/110%E5%89%B5%E9%80%A0%E5%8A%9B%E8%B3%87%E5%84%AA/bd3/gun\\_10ok.zip](http://163.26.205.2/~htaes/110%E5%89%B5%E9%80%A0%E5%8A%9B%E8%B3%87%E5%84%AA/bd3/gun_10ok.zip)

# 槍-程式說明

開發板 Arduino 初始化

設定紅外線接收器腳位 4

宣告 全域 整數 (int) bullet 值 30

宣告 全域 整數 (int) delay 值 200

宣告 全域 字串 (String) GunShow 值 “ ”

宣告 全域 整數 (int) GunType 值 0

初始化 液晶顯示 1602 訊號: I2C I2C位址 0x27

液晶顯示 背光 開

液晶顯示 清除畫面

液晶顯示 設定游標行: 0 列: 0

液晶顯示 顯示 “ Select Gun&Bull ”

液晶顯示 設定游標行: 0 列: 1

液晶顯示 顯示 “ Choice 1 2 3 4.. ”

子彈數

子彈發射間格時間

LCD顯示槍形式的文字說明

槍形式的代號

1：單發射擊

2：3發點放

3：連發1機關槍模式

4：連發2雷射模式



# 槍-程式說明

重複執行

當接收到紅外線訊號時

如果 接收到的紅外線編碼 = "7d7d847b" 單發射擊模式

執行

- 設定 GunShow 值 "single shot"
- 設定 delay 值 60
- 設定 GunType 值 1
- 呼叫 ShowLCD 參數值

如果 接收到的紅外線編碼 = "7d7d44bb" 3發點放模式

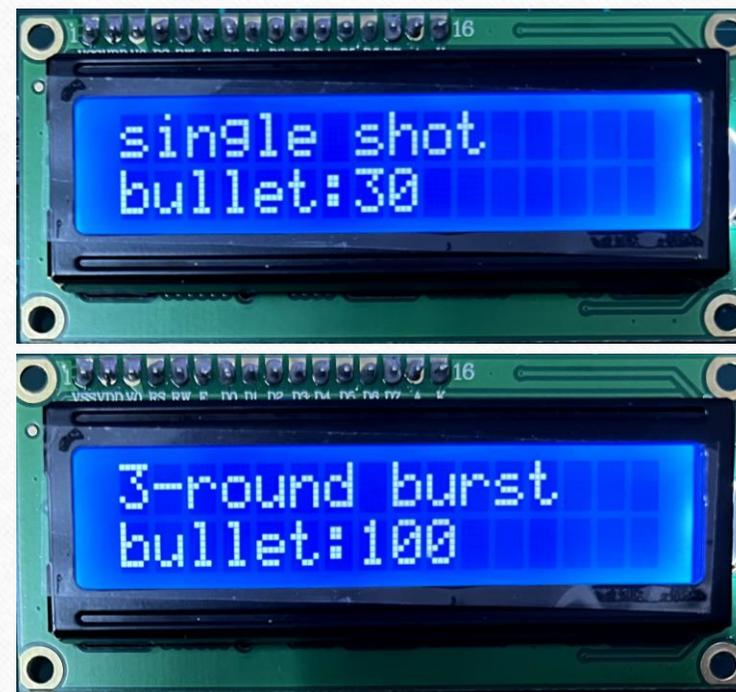
執行

- 設定 GunShow 值 "3-round burst"
- 設定 delay 值 30
- 設定 GunType 值 2
- 呼叫 ShowLCD 參數值

收到紅外線訊號後在LCD上顯示資訊

函數 ShowLCD 宣告參數

- 液晶顯示 清除畫面
- 液晶顯示 設定游標行：0 列：0
- 液晶顯示 顯示 GunShow
- 液晶顯示 設定游標行：0 列：1
- 液晶顯示 顯示 "bullet:"
- 液晶顯示 顯示 bullet



# 槍-程式說明

```
如果 接收到的紅外線編碼 = "7d7dc43b"
執行
  設定 GunShow 值 "Machine Gun"
  設定 delay 值 30
  設定 GunType 值 3
  呼叫 ShowLCD 參數值
如果 接收到的紅外線編碼 = "7d7d24db"
執行
  設定 GunShow 值 "Laser Gun"
  設定 delay 值 1
  設定 GunType 值 4
  呼叫 ShowLCD 參數值
```

連發:機關槍模式

連發:雷射模式



如果 接收到的紅外線編碼 = " 7d7da45b "

執行 設定 bullet 值 30  
呼叫 ShowLCD 參數值

設定子彈為30發

如果 接收到的紅外線編碼 = " 7d7d649b "

執行 設定 bullet 值 100  
呼叫 ShowLCD 參數值

設定子彈為100發

如果 接收到的紅外線編碼 = " 7d7de41b "

執行 設定 bullet 值 500  
呼叫 ShowLCD 參數值

設定子彈為500發

如果 接收到的紅外線編碼 = " 7d7d14eb "

執行 修改 bullet 增量 10  
呼叫 ShowLCD 參數值  
延遲毫秒 200

增加10發子彈

如果 接收到的紅外線編碼 = " 7d7d946b "

執行 修改 bullet 增量 100  
呼叫 ShowLCD 參數值  
延遲毫秒 200

增加100發子彈

# 槍-程式說明



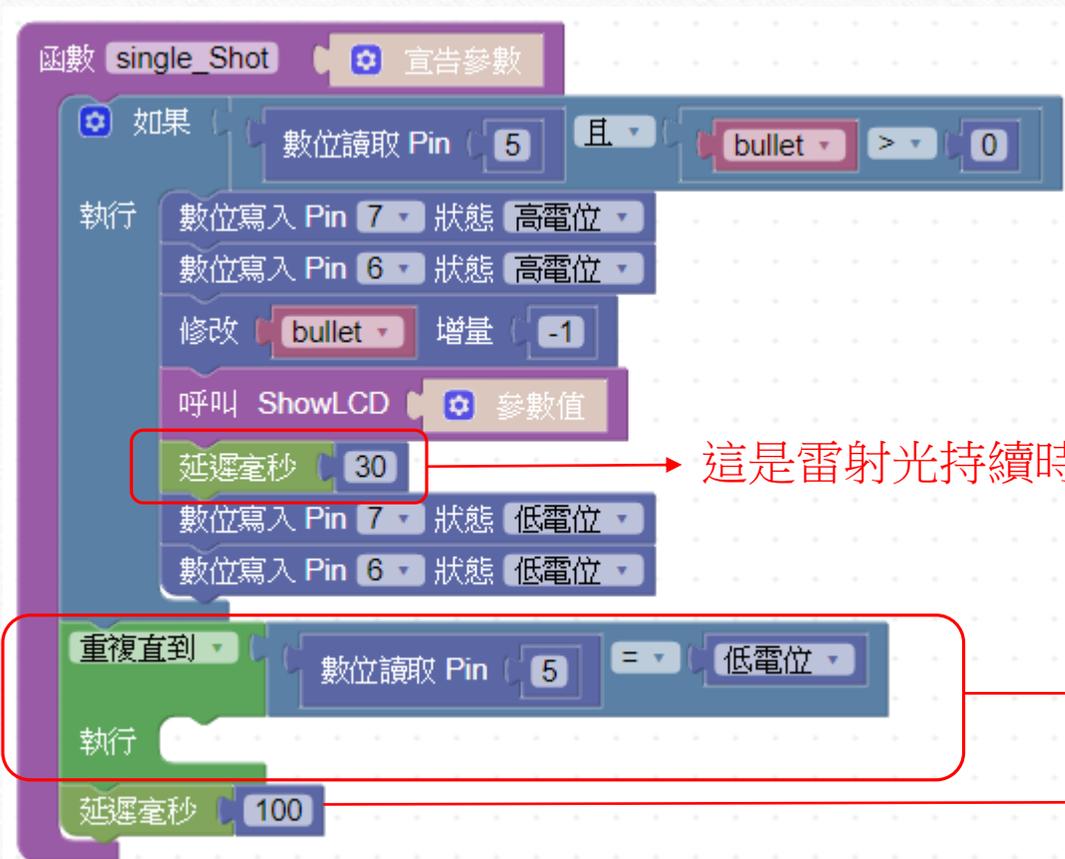
如果GunType=1執行single\_Shot副程式(函數)

如果GunType=2執行3 round burst副程式(函數)

如果GunType=3執行Machine\_Gun副程式(函數)

如果GunType=4執行Laser\_Gun副程式(函數)

# 槍-副程式(函數)-單發射擊



如果pin5的按鈕按下而且子彈數大於0，則

Pin6高電位(雷射光發射)  
pin7高電位(有源蜂鳴器叫)

這是雷射光持續時間，大於靶的檢查光線時間(20)

一直等到pin5低電位(放開按鈕)，再繼續執行。

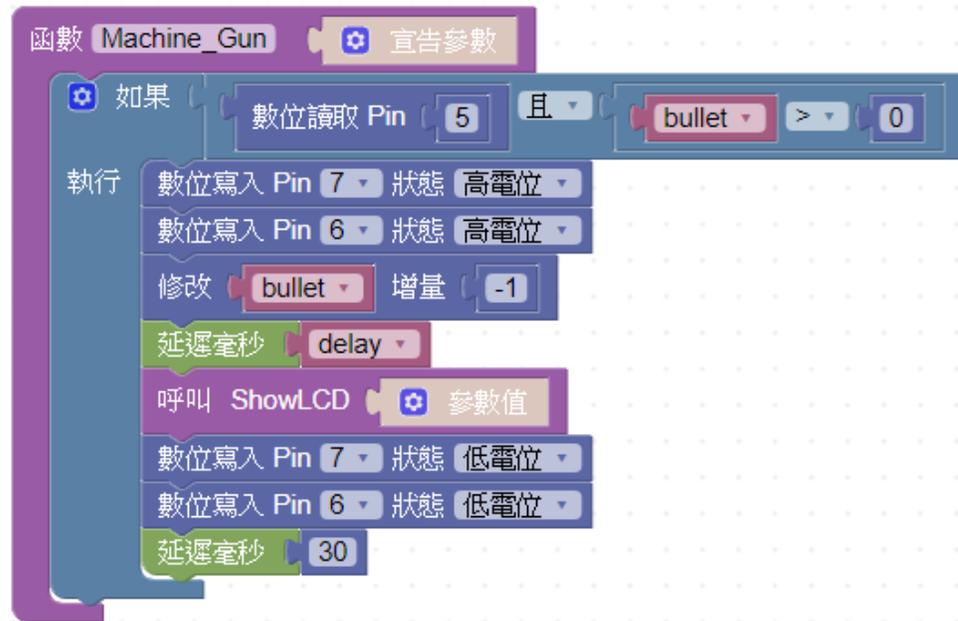
延遲100毫秒是避免接觸彈跳，建議20-100毫秒。

# 槍-副程式(函數)-3發點放



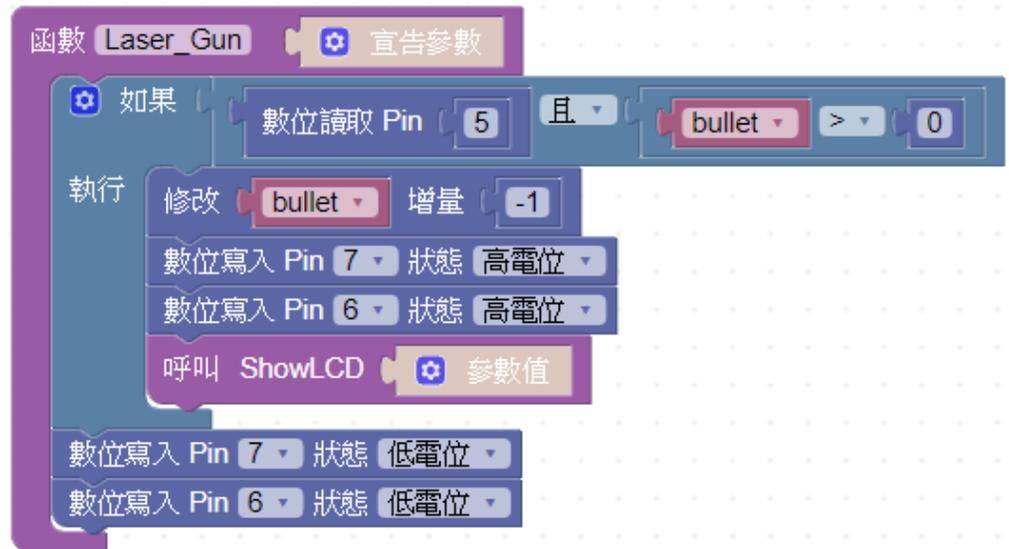
# 槍-副程式(函數)說明

機關槍模式，發射時(按下)，電位高低切換



```
function Machine_Gun()
  if (digitalRead(Pin 5) && bullet > 0)
    execute
      digitalWrite(Pin 7, HIGH)
      digitalWrite(Pin 6, HIGH)
      modify bullet, -1
      delay 100ms
      call ShowLCD
      digitalWrite(Pin 7, LOW)
      digitalWrite(Pin 6, LOW)
      delay 300ms
```

雷射模式：發射時(按下)持續高電位



```
function Laser_Gun()
  if (digitalRead(Pin 5) && bullet > 0)
    execute
      modify bullet, -1
      digitalWrite(Pin 7, HIGH)
      digitalWrite(Pin 6, HIGH)
      call ShowLCD
      digitalWrite(Pin 7, LOW)
      digitalWrite(Pin 6, LOW)
```

# 靶：程式說明1

初始化

宣告 全域 長整數 (long) T1 值 0

宣告 全域 浮點數 (float) 靶1角度 值 90

宣告 全域 浮點數 (float) 靶1角度改變 值 0

宣告 全域 長整數 (long) T2 值 0

宣告 全域 浮點數 (float) 靶2角度 值 90

宣告 全域 浮點數 (float) 靶2角度改變 值 0

宣告 全域 長整數 (long) T3 值 0

宣告 全域 浮點數 (float) 靶3角度 值 90

宣告 全域 浮點數 (float) 靶3角度改變 值 0

宣告 全域 長整數 (long) T4 值 0

宣告 全域 浮點數 (float) 靶4角度 值 90

宣告 全域 浮點數 (float) 靶4角度改變 值 0

宣告 全域 整數 (int) 定時間隔 值 20

設定紅外線接收器腳位 8 - I2C

靶1的時間間隔，因為是時間單位，所以使用長整數

靶1的角度，因為有小數點，所以使用浮點數

靶1的角度改變值

靶收集光線資料的時間間隔

# 靶設定1



4隻靶(舵機)全部轉到90度角位置(直立)  
角度不改變(改變值是0)

# 靶設定2



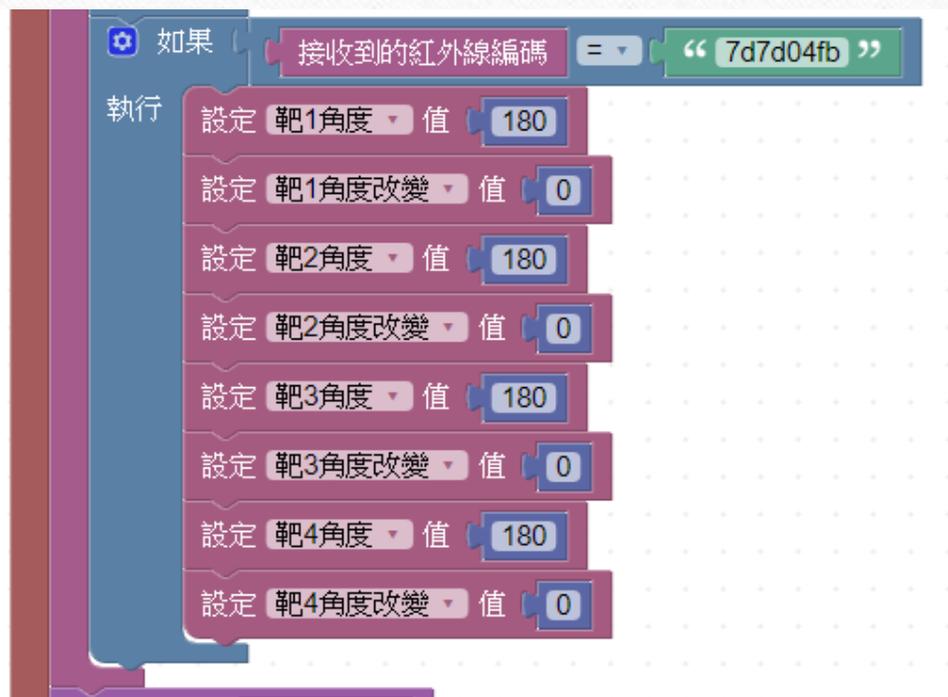
The image shows a Scratch script starting with an 'If' block that checks if the received infrared code is '7d7d44bb'. Inside the 'If' block, there are eight 'Set' blocks arranged in pairs for four targets. Each pair consists of a 'Set Target Angle' block and a 'Set Target Angle Change' block. The 'Set Target Angle' blocks are set to 'Random' with a minimum of 20 and a maximum of 160. The 'Set Target Angle Change' blocks are set to 0.2.

```
if (received infrared code = "7d7d44bb") {
  set target 1 angle to random (20 to 160)
  set target 1 angle change to 0.2
  set target 2 angle to random (20 to 160)
  set target 2 angle change to 0.2
  set target 3 angle to random (20 to 160)
  set target 3 angle change to 0.2
  set target 4 angle to random (20 to 160)
  set target 4 angle change to 0.2
}
```

隨機轉到20~160度角位置

角度改變0.2(較慢)，數字越大轉動越快。

# 靶設定3



4隻靶(舵機)全部轉到180度角位置(躺平)  
角度不改變(改變值是0)

# 靶1說明，其餘類推

宣告 全域 整數 (int) 定時間隔 值 20

```
function func1()
  if (current_millis() > T1 + interval)
    if (analogRead(A0) > 350)
      set_increment(0)
      set_Angel1(180)
      write_servo_motor(4, Angel1)
    else if (Angel1 < 0 || Angel1 > 180)
      set_increment(0 - increment)
      write_servo_motor(4, Angel1)
      modify_Angel1_increment(increment)
    set_T1(current_millis())
```

每隔[定時間隔]毫秒檢查，因為硬體是單工作業

當光敏電阻的數值大於350(自訂，看說明)  
靶1不轉動  
角度設為180，躺平。

0~180來回擺動

重設時間起始點，方便計算下一次的時間間隔

# 光敏電阻

---

- 每個光敏電阻接收到光源後，傳回的數值都不同(尤其不同廠商、不同批號)，建議針對不同光敏電阻做不同的設定。