

自主學習成果

# 軟硬兼施-Arduino的玩法

20729黎鎮緯

# 目錄

- 動機 P.2
- 目標 P.3
- 物件說明 P.4~6
- 實作過程 P.7~9
- 成果 P.10
- 心得 P.11
- 參考文獻 P.12



# 動機

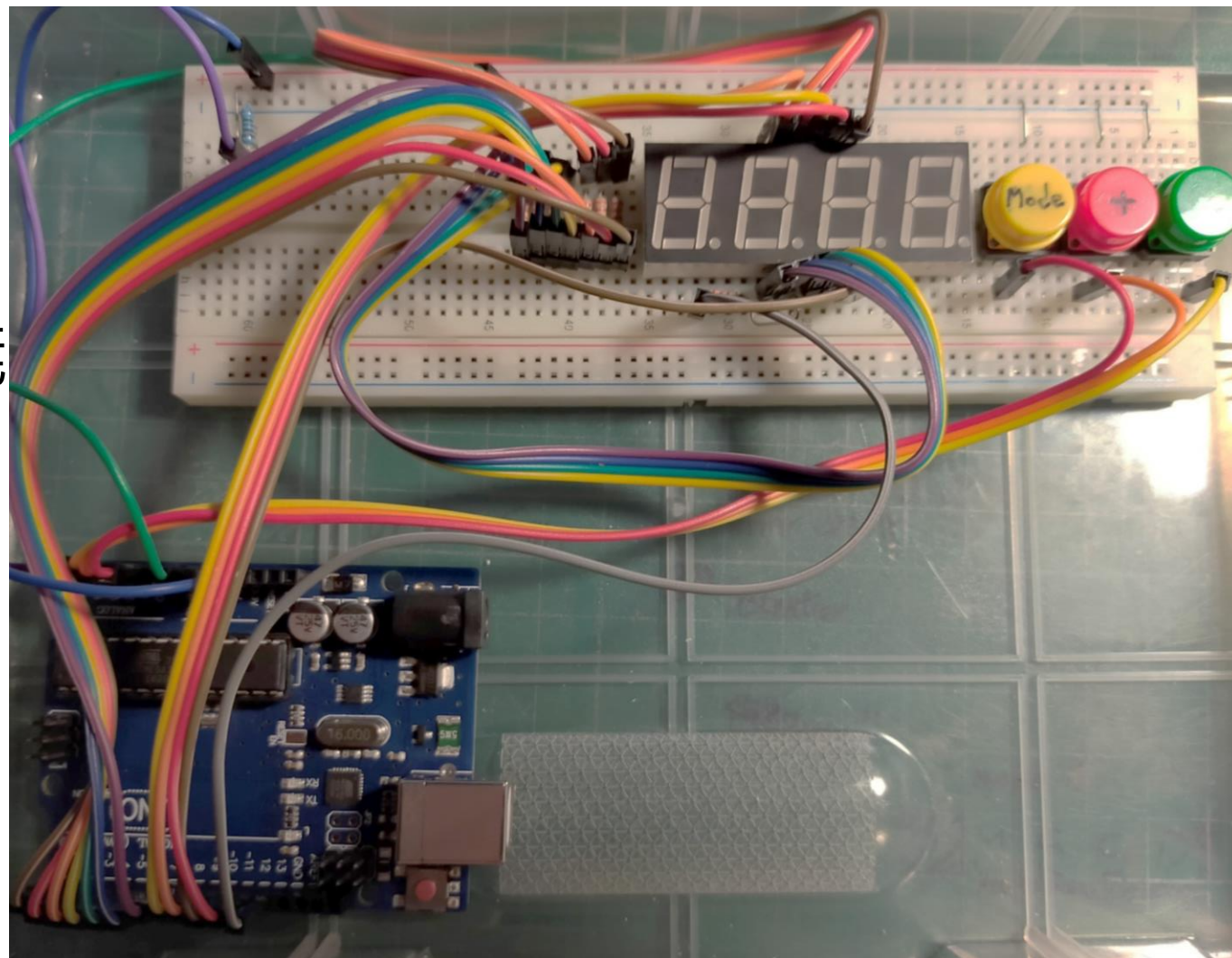
房間裡的鐘原本放在右圖的主機裡，但是會**阻礙顯卡散熱**，且時鐘還會有**光害影響睡眠**。因此我打算做一個時鐘，且能改善上述問題。





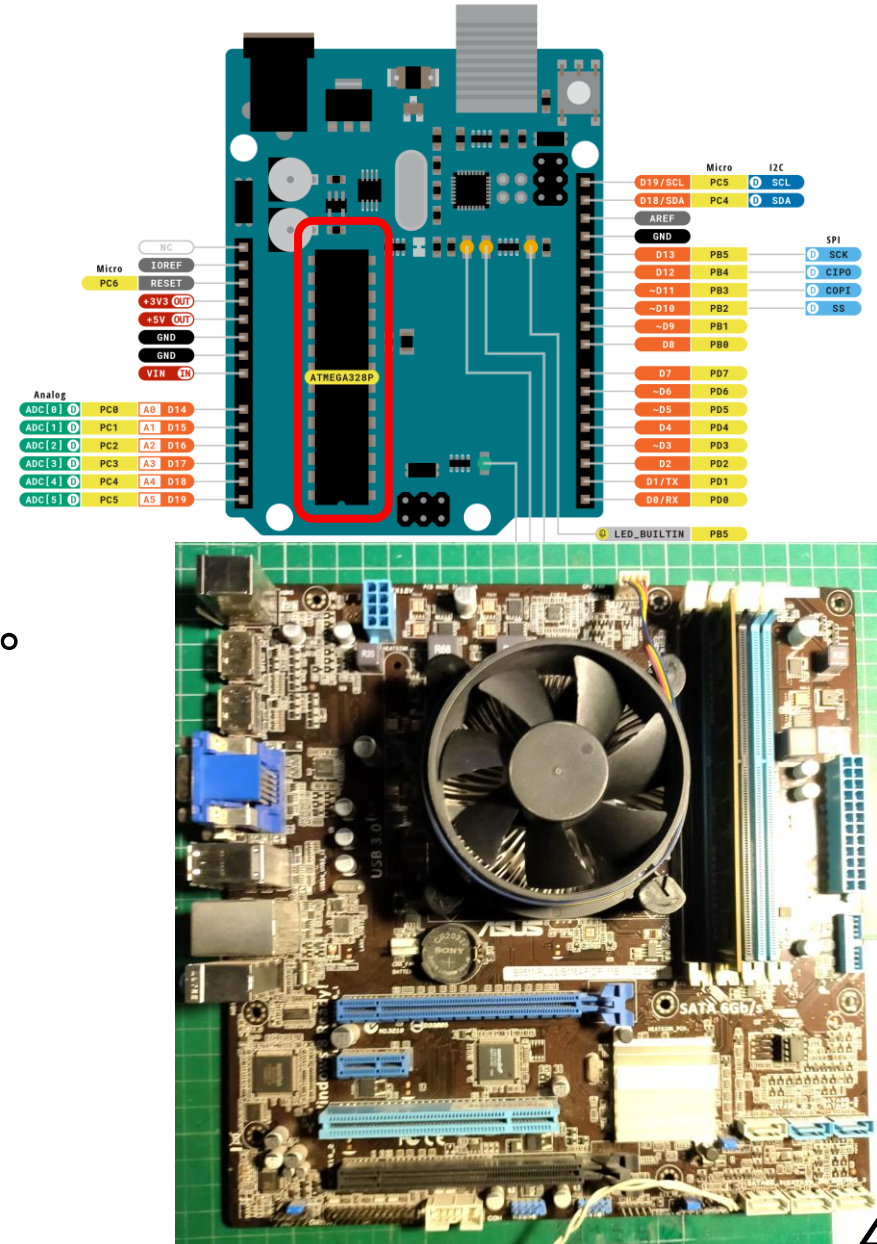
# 目標

- 做出一個時鐘
- 且有環境光檢測
- 具有人性化的設定功能



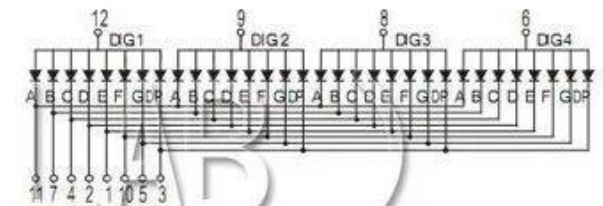
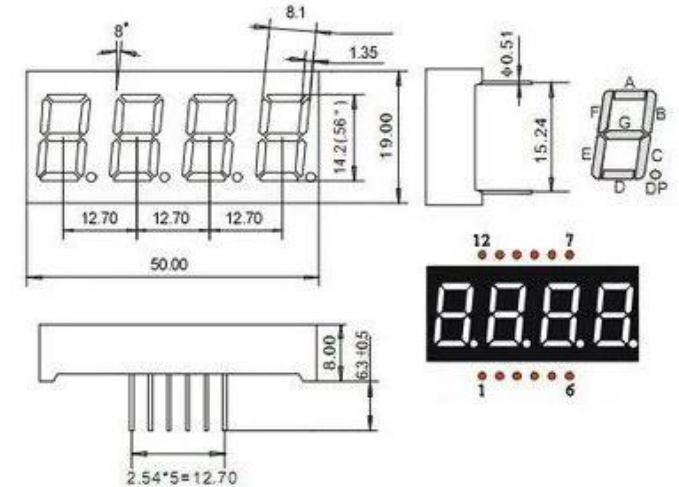
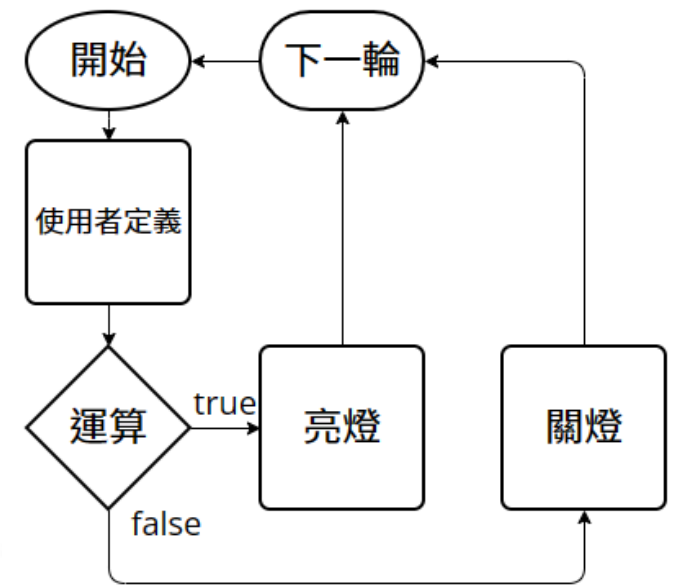
# Arduino說明

Arduino可以理解為下面的主板，其中圈起來的IC具有CPU、RAM和ROM的功能，而Arduino板子上的接點則是相當於下圖主機板的USB和其他I/O接口。



# 七段顯示器說明

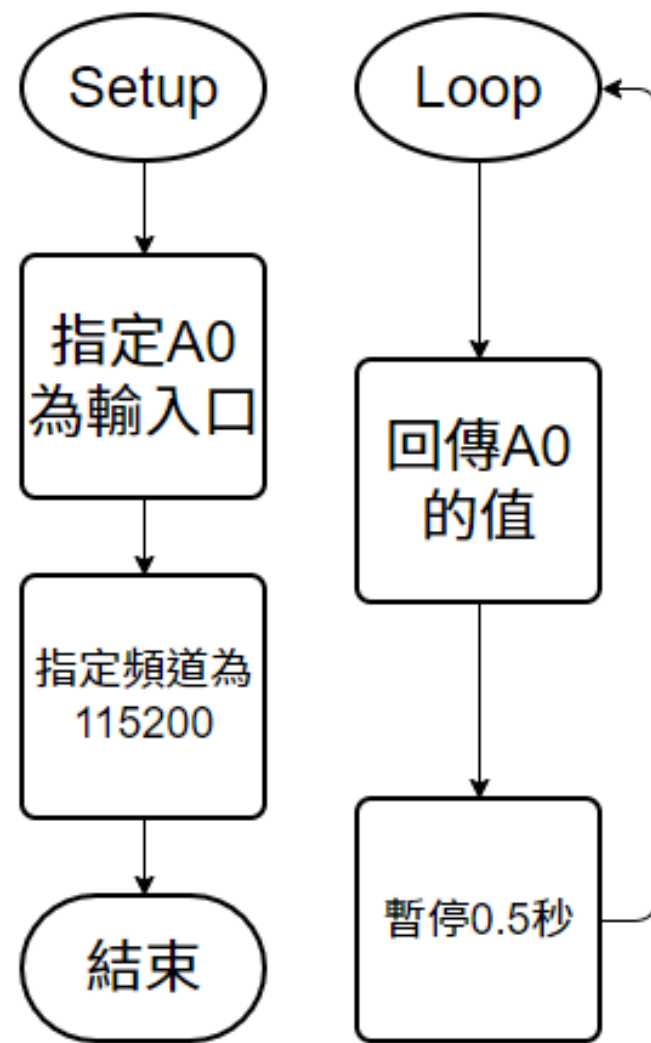
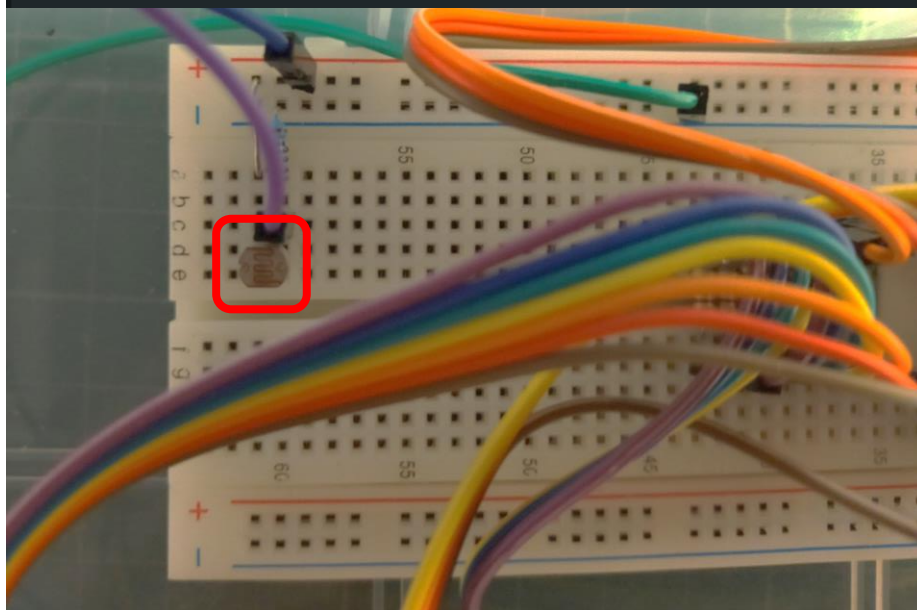
七段顯示器是多組LED結合而成，通常運行流程大致會像右上圖一樣。然而，四位七段顯示器是共用陰極的，這時就要讓陽極通電的時間錯開，並使用視覺暫留，使四塊區域看起來是同時點亮。



# 光敏電阻說明

圖中是光敏電阻，  
為可變電阻的一種  
可以用中上圖的指令  
檢測。右圖為中  
上圖程式執行過程

```
void setup()  
{  
  pinMode(A0, INPUT); //指定接口, 接口形式  
  Serial.begin(115200); //指定回傳頻道  
} // setup()  
  
void loop(){  
  Serial.println(analogRead(A0)); //監控視窗顯示值  
  delay(500); //暫停0.5sec  
} // loop()
```





# 實作過程

- 1~9字型的效果不佳。
- 後來上網查後，發現右邊的陣列方式會比較好，而且若之後線路更動要修改時會比較容易調整。至於亮度過低則是因為在正極就先限流了，改成在負笈裝電阻就修好了

```
void ledef(int num,int dig);  
const int pinState[10][7] = {  
    //2, 3, 4, 5, 6, 7, 8  
    //7,10,11, 5, 4, 2, 1  
    //B, F, A, G, C, D, E  
    {0, 0, 0, 1, 0, 0, 0}, // 0  
    {0, 1, 1, 1, 0, 1, 1}, // 1  
    {0, 1, 0, 0, 1, 0, 0}, // 2  
    {0, 1, 0, 0, 0, 0, 1}, // 3  
    {0, 0, 1, 0, 0, 1, 1}, // 4  
    {1, 0, 0, 0, 0, 0, 1}, // 5  
    {1, 0, 0, 0, 0, 0, 0}, // 6  
    {0, 1, 0, 1, 0, 1, 1}, // 7  
    {0, 0, 0, 0, 0, 0, 0}, // 8  
    {0, 0, 0, 0, 0, 0, 1} // 9  
};
```



# 實作過程

- 在設定時指標閃爍時遇到字元只會常亮的問題
- 在第二層if後面加上else之後就正常了

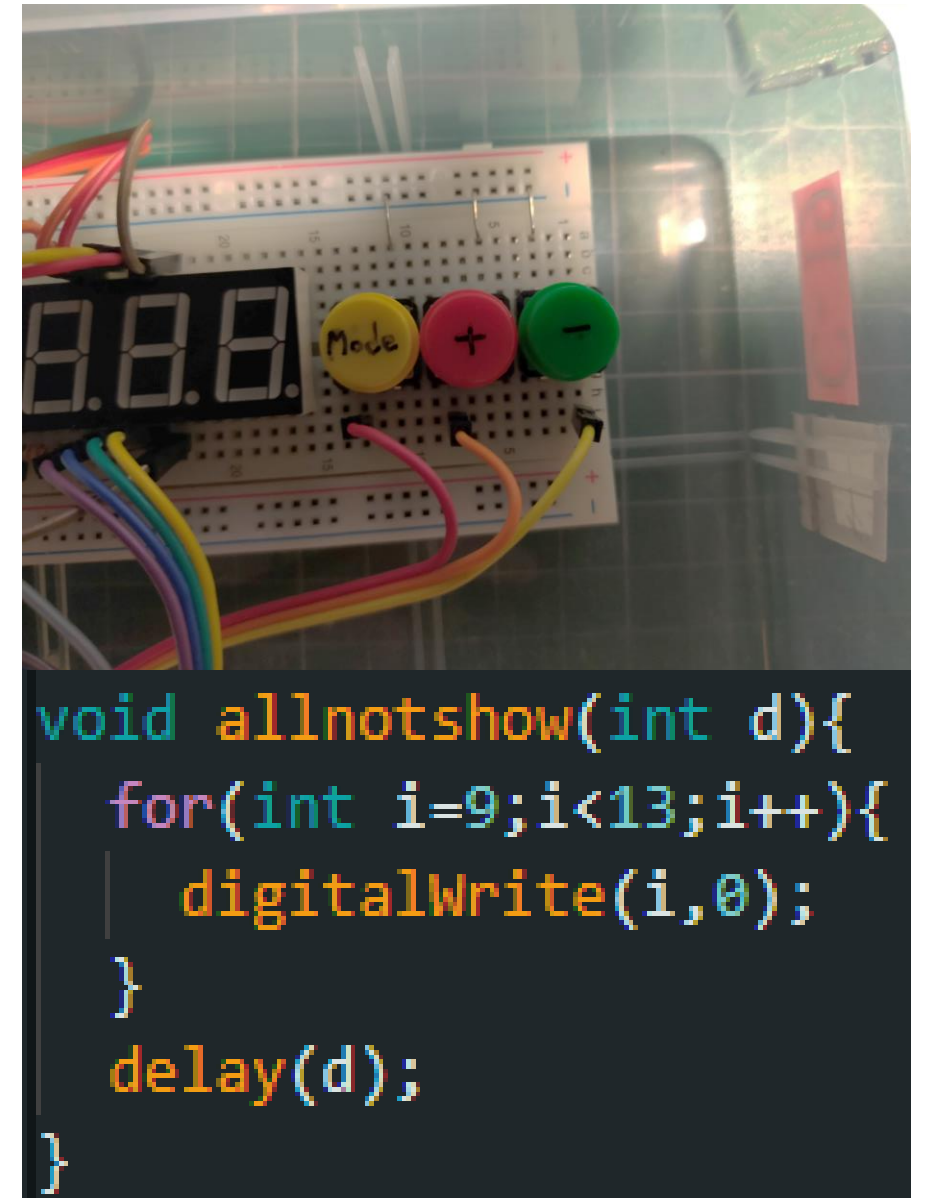
```
timer+=1;
if(timer%98==0){
    sec+=1;
    if(sec>1){
        sec=0;
    }
}
```

左圖為閃  
爍控制函式

```
if(st==1 or st==3){
    if(dig==0 or dig==1){
        digitalWrite(dig+9,sec);
    }else{
        digitalWrite(dig+9,1);
    }
}else if(st==2 or st==4){
    if(dig==2 or dig==3){
        digitalWrite(dig+9,sec);
    }else{
        digitalWrite(dig+9,1);
    }
}else if(st==5){
    digitalWrite(dig+9,sec);
}else if(light>=391){
    digitalWrite(dig+9,0);
}else{
    digitalWrite(dig+9,1);
}
delay(5);
```

# 實作過程

- 按下Mode後會導致顯示器會暫時失常，或是程式誤以為是長按。
- 我是用讓顯示器暫時關閉的方案，讓程式到達一定時間才顯示，也順便解決連點問題。



# 實作過程

- 原本設定區塊我打算用if執行，但是會造成跳出設定的問題。
- 之後設定這一塊我一律使用while讓程式在設定時持續運行在這個區塊。

```
while(st>0){  
    while(st==1){//hour  
        show=0;  
        time();  
        mobtn();  
        if(digitalRead(15)==0){  
            hour+=1;  
            allnotshow(200);  
        }  
        if(digitalRead(16)==0){  
            hour-=1;  
            allnotshow(200);  
        }  
        if(mode>5){  
            st=2;  
            allnotshow(200);  
            mode=0;  
        }  
    }  
    while(st==2){//min  
        show=0;  
        time();
```

成果(影片) 自主學習成果 (youtube.com)



# 心得

這次的自主學習項目帶給我許多的收穫，雖然很多時間都花在除錯上，但是也讓我感受到寫硬體和軟體的不同。因為程式不是在速度飛快的電腦上運行，所以要考慮到它的執行速度外，還要確保當它執行這一個函式時，另一個也有被正確地執行。在編程上要發揮更多的耐心，和能處理突發錯誤的能力。花最多時間的區塊是在計算時間的功能，因為我給的日期變數是有包含閏年和平年，還有日期和月份的關聯，要花時間設計和測試。

因為這程式是執行在Arduino上，因此從程式設計、測試到完成會比單純在電腦寫C++並執行所花的時間更長。

```
if(day>28){  
    if(month==2 and year%4!=0){  
        if(st==0){  
            month+=1;  
        }  
        day=1;  
    }  
    if(day>29 and month==2 and year%4==0){  
        if(st==0){
```

# 參考文獻

- [5641BH.pdf \(xlitx.com\)](#)
- [mBlock & Arduino \( 17 \) 四位數七段顯示器 \(openhome.cc\)](#)
- [Arduino - Home](#)

Thanks For Listening!