

國立台北科技大學

第四次實習報告

學期：112學年度第2學期

科別：智慧自動化工程科

班級：三年級A班

學號：1102B0010

姓名：江冠儒

課程名稱：機電整合與實習

授課老師：李政宏博士

實習名稱：機電整合與實習

1、單元名稱：按鈕控制繼電器、LED、蜂鳴器與七段顯示器動作

2、實習設備

項次	品名	規格	數量
1	樂創學習 KTeduino 套件		1

3、實習材料：

項次	品名	規格	數量
1	Arduino	Uno R3	1
2	杜邦線	公對母	12
3	按鈕開關		4
3	LED		8
4	七段顯示器	四位數	1
5	74HC595		1
6	蜂鳴器		1
7	繼電器		1

4、實習步驟：

1、確認各電子元件運作情況：

在實驗前，應先將各設備與元件的運作情況進行簡單測試，以避免後續程式出錯時無從除錯。

2、進行接線：

在進行Arduino的程式撰寫前，應先將需用到之線路進行接線，並將相關的接腳名稱進行紀錄作為備用。接線方式為：

- (1) 杜邦線應選取相異顏色的線，若無法，則至少相鄰接腳應為不同顏色，以避免整線時混淆。
- (2) 取8條杜邦線，將公頭端插至Aduino 9~2 pin腳，並依序將母頭端接至樂創學習KTeduino套件電路板上的JP22-1 ~ JP22-8 腳位。
- (3) 取4條杜邦線，將公頭端插至Aduino A0~A3 pin腳，並依序將母頭端接至樂創學習KTeduino套件電路板上的JP24-5 ~ JP24-8腳位。
- (4) 接線時，應確保杜邦線接頭有確實插入腳位。
- (5) 因為七段顯示器、74HC595、蜂鳴器與繼電器在套件裡已經接好線路，故不需要額外接線。

3、 撰寫程式：

撰寫程式分為以下幾個區塊：

- (1) 變數設定與初始化
- (2) 關閉LED函式
- (3) 讀取按鈕函式與主程式選擇部分
- (4) 按鈕1 動作函式
- (5) 按鈕2 動作函式
- (6) 按鈕3 動作函式
- (7) 按鈕4 動作函式

後續會依序進行說明。

4、 程式 - 變數設定與初始化

變數設定與初始化，是在一開始進行之基本設定並進行初始化，如按鈕所使用之腳位，以及所需使用到的變數定義。步驟如下：

- (1) 設定按鈕相關使用接腳的變數
- (2) 設定LED相關使用接腳的變數
- (3) 設定七段顯示器相關使用接腳的變數
- (4) 設定蜂鳴器相關使用接腳的變數
- (5) 設定繼電器相關使用接腳的變數
- (6) 設定所需之變數，可先獨立出一個區塊，後續用到變數時可更方便的統一在此區塊設定變數及除錯。所命令的變數如下表一

表一 變數命令與用途

變數名稱	值	用途	備註
PB[4]	{A0,A1,A2,A3}	設定按鈕使用之接腳為一串列，PB[0]到PB[3]分別對應到Arduino的A0到A3腳位。	const int
LED[8]	{9,8,7,6,5,4,3,2}	設定LED使用之接腳為一串列，LED[0]到LED[7]分別對應到Arduino的9到2腳位。	const int
Off	1	由於此套件是屬於低態啟動，與常用之邏輯相反，故設定此常數以方便閱讀與除錯。	const int
On	0	由於此套件是屬於低態啟動，與常用之邏輯相反，故設定此常數以方便閱讀與除錯。	const int
scan[4]	{5,6,7,8}	設定掃描使用之接腳為一串列，scan[0]到scan[3]分別對應到Arduino的5到8腳位。	const int
dataPin	10	宣告10接腳連接串列資料	const int

latchPin	11	宣告11接腳連接栓鎖信號	const int
clockPin	12	宣告12接腳移位脈波信號	const int
barCode [8]	{0xFE, 0xFD, 0xFB , 0xF7, 0xEF, 0xDF , 0xBF, 0x7F}	依序為七段顯示器A~G以及DP 段的顯示代碼	const int
buzzer Pin	13	宣告13接腳為蜂鳴器使用	const int
relayPin	4	宣告4接腳為繼電器使用	const int

5、 程式 - 關閉LED函式

使LED全部關閉，以方便對LED進行後續動作。程式步驟如下：

- (1) 依序將LED[0]~LED[7]關閉

程式如下：

```
void turnOff() {
    for(int i=0;i<8;i++) {
        digitalWrite(LED[i], Off);    //關閉LED[0]~LED[7]
    }
}
```

6、 程式 - 讀取按鈕函式與主程式選擇部分

因為需要根據按下不同的按鈕而有不同的反應，故先需要讀取各按鈕的狀態，再根據按下不同的按鈕切換到不同的區塊，以達成目的。步驟如下：

- (1) 設定一個函式，根據讀取不同按鈕的狀態，回傳按下的按鈕編號。
- (2) 在主程式loop裡使用switch case判斷式，根據讀取按鈕函式的回傳值，選擇到不同的按鈕動作函式，以進行後續動作
- (3) 其中在按下按鈕1、按鈕2與未按下時因為後續動作不需要使用到LED，所以會將LED全部關閉，以避免其餘問題產生

程式如下：

```
int readPb(void){
    if(!digitalRead(PB[0])) return 1; //若按鈕PB[0]被按下，則回傳1
    else if(!digitalRead(PB[1])) return 2; //若按鈕PB[1]被按下，則回傳2
    else if(!digitalRead(PB[2])) return 3; //若按鈕PB[2]被按下，則回傳3
    else if(!digitalRead(PB[3])) return 4; //若按鈕PB[3]被按下，則回傳4
}
```

```

void loop() {
  switch(readPb()) { //讀取按鈕
    case 1:
      turnOff(); //將LED全部關閉
      S1();
      break;
    case 2:
      turnOff(); //將LED全部關閉
      S2();
      break;
    case 3:
      S3();
      break;
    case 4:
      S4();
      break;
    default:
      turnOff(); //將LED全部關閉
  }
}

```

7、 程式 - 按鈕1 動作函式

繼電器開始激磁，蜂鳴器開始響，同時四個七段顯示器開始顯示，並間隔一秒依序顯示完每一個段含DP。如：A-B-C-D-E-F-G-DP-A-B-C-...-DP-A-...。七段顯示器、蜂鳴器與繼電器要同步。程式步驟如下：

- (1) 開啟繼電器
- (2) 開啟蜂鳴器
- (3) 在四個七段顯示器顯示A段
- (4) 顯示一定時間
- (5) 切換至下一段並顯示，並從步驟(4)重複至DP段
- (6) 如此往複循環

程式如下：

```

void S1() {
  digitalWrite(relayPin, On); //開啟繼電器
  tone(buzzerPin, 1000); //使蜂鳴器發出聲音
  for(int j=0;j<8;j++) { //依序從七段顯示器的A段至DP段
    for(int k=0;k<250;k++) { //每組的顯示時間為 (k總數*i總數) 毫秒
      for(int i=0;i<4;i++) { //依序更新四個七段顯示器
        digitalWrite(latchPin, 0); //解除資料栓鎖
        //設定顯示內容為barCode[j]
        shiftOut(dataPin, clockPin, MSBFIRST, barCode[j]);
        //輸出顯示資料
        digitalWrite(latchPin, 1); //栓鎖資料
        digitalWrite(scan[i], 0); //輸出掃描信號
      }
    }
  }
}

```

```

        delay(1); //暫停1毫秒
        digitalWrite(scan[i], 1); //關閉掃描信號
    }
}
}
}

```

8、 程式 - 按鈕2 動作函式

暫時停止激磁、嗶聲與顯示。程式步驟如下：

- (1) 因為七段顯示器顯示是需要程式一直執行，所以只要此函式沒有七段顯示器的程式區塊，就不會顯示
- (2) 關閉繼電器
- (3) 關閉蜂鳴器

程式如下：

```

void S2() {
    digitalWrite(relayPin, Off); //關閉繼電器
    noTone(buzzerPin); //使蜂鳴器停止發出聲音
}

```

9、 程式 - 按鈕3 動作函式

LED由兩側往內開始亮(其餘LED保持亮)。程式步驟如下：

- (1) 開啟兩側的LED
- (2) delay一段時間
- (3) 兩邊各往內移一顆，並從步驟(2)重複，直到到最內側的LED
- (4) 將LED全部關閉
- (5) delay一段時間
- (6) 如此往複循環

程式如下：

```

void S3() {
    for(int i=0;i<=3;i++) {
        digitalWrite(LED[i], On); //開啟LED
        digitalWrite(LED[7-i], On); //開啟LED
        delay(200); //暫停0.2秒
    }
    delay(200); //暫停0.2秒
    turnOff(); //將LED全部關閉
    delay(200); //暫停0.2秒
}

```

10、 程式 - 按鈕4 動作函式

LED由內往外側開始亮(其餘LED皆滅)。程式步驟如下：

- (1) 開啟內側的LED
- (2) delay一段時間
- (3) 關閉LED
- (4) 兩邊各往外移一顆，並從步驟(2)重複，直到到最外側的LED
- (5) 將LED全部關閉
- (6) delay一段時間
- (7) 如此往複循環

程式如下：

```
void S4() {  
  for(int i=3;i>=0;i--) {  
    digitalWrite(LED[i], On); //開啟LED  
    digitalWrite(LED[7-i], On); //開啟LED  
    delay(200); //暫停0.2秒  
    digitalWrite(LED[i], Off); //關閉LED  
    digitalWrite(LED[7-i], Off); //關閉LED  
  }  
  turnOff(); //將LED全部關閉  
  delay(200); //暫停0.2秒  
}
```

11、 成品：



實驗結果影片：

5、 問題與討論：

1、 Arduino板接腳數目不足怎麼辦？

本次題目因為需要同時用到繼電器、LED、蜂鳴器與七段顯示器，不含類比輸入的按鈕，全部總共要17個腳位，而Arduino Uno板的輸出腳位只有14個，若每個腳位只能命令給一個用途，則腳位數一定不足。所以勢必需要有腳位的命令是重疊到的，而此題目中，LED與其他元件不會同時有動作，所以可以將LED所使用的腳位與其他元件所使用的腳位重疊，這樣就可以在有限的腳位數目裡將所有元件都設定完成。而此解決方案帶來的副作用就是會在有重疊的腳位動作時干擾到另一個元件，如在七段顯示器掃描顯示時，LED會輕微的發亮，又或是在LED動作時，繼電器會一直開開關關，雖然會干擾到其他的元件，但被干擾的元件並沒有需要在同一區塊動作，故仍可算是在腳位不足的權宜之計。

2、如何同時使七段顯示器、蜂鳴器與繼電器開始動作？

因為七段顯示器使用的是掃描顯示，故在掃描過程中需要連續不斷的執行，而開啟蜂鳴器與繼電器只需要對腳位的狀態進行一次改變，即可保持狀態，故只要先將蜂鳴器與繼電器開啟，後續再對七段顯示器進行掃描顯示，即可在開始的一瞬間將三個元件同時啟動。

6、心得與建議：

這次的實驗需要將從開始以來的所有知識與程式進行綜合利用，所以在之前的題目中有親手做過並了解原理的話，這次的實驗並不會非常困難。可惜的點在於繼電器通常適用於控制大電的開關，而我們並沒有一個可以實際控制的負載給繼電器使用，所以只能簡單的控制開與關，所幸在其他地方也有碰過繼電器的經驗，這次的實作搭配以往的經驗，能夠使我更加熟悉此元件，也期待在未來能有更多碰到繼電器的機會，也希望能有設備使我們實際的使用繼電器進行實作。

7、參考文獻：

- 1、張義和、程兆龍編著，KT eduino樂創學習（附範例光碟），新文京開發出版股份有限公司，2017年11月出版。