**電通二甲微處理器實驗 實驗預報**

|  |  |  |  |
| --- | --- | --- | --- |
| **實驗名稱** | Lab 04 - 四合一七段顯示器 | | |
| **組別** | 27 | **組員** | 04050015陳姿華、04052502許書瑜 |

1. **實驗目的**

\* 了解七段顯示器之顯示原理

1.共陰極與共陽極七段顯示器之不同?

2.如何讓七段顯示器顯示0 –F?

3.如何控制四合一七段顯示器?

4.如何讓四合一七段顯示器顯示四位數字?

(實驗內容-時鐘)

\* 四合一七段顯示器每隔0.3秒鐘會跳一個數字,從0000->0059->0100->跳至2359 （60 分)

\* 於Pin 10 接一個按鍵, 按下後歸零顯示(10 分)

\* 於Pin 11 接一個按鍵, 按下後倒數計時(30 分)

1. **實驗步驟**

**實驗步驟1:**

\* Arduino 接好四合一七段顯示器之電路 \* 四合一七段顯示器每隔0.3 秒鐘會跳一個數字, 從 0000 -> 0059 -> 0100 -> 跳至2359 後歸零。

**實驗步驟2:**

\* 於Pin 10 接一個按鍵, 按下後歸零顯示

\* 於Pin 11 接一個按鍵, 按下後倒數計時

**實驗步驟3:**

合併歸零、倒數計時

1. **程式碼**

\* Ardunio 7-seg lib

#include “SevSeg.h”

SevSeg sevseg;

void setup()

{

byte numDigits = 4;

byte digtPins[] = {2,3,4,5};

byte segmenPins[] = {6,7,8,9,10,11,12,13};

sevseg.begin(COMMON\_ANODE,numDigits,digitPins,segmentPins);

.......

}

……

\*Code (實驗步驟1)

sevseg.setNumber(Hour \* 100 + Min, 0);

Min = Min + 1;

If(Min == 60)

{

Min = 0,

If (Hour == 23)

Hour=0else Hour++

}

sevseg.refreshDisplay();

sleep 0.3 second

\*Code (實驗步驟2)

Pin10= DigitalRead(D10) // Reset

Pin11 = DigitalRead (D11) // Reverse

If (Pin10 == LOW)

Min = Hour = 0;

If (Pin11 ==LOW)

step = -1 // Reverse

else step = 1

sevseg.setNumber(Hour \* 100 + Min, 0)

Min = Min + step;

If Min == 60

{

…

} // overflow

If (Min == -1)

{

Min = 59;

If (Hour == -1)

Hour = 23

else

Hour—

} // underflow sevseg.refreshDisplay(); sleep 0.3 second

1. **電路圖**