**電通二甲微處理器實驗 實驗結報**

|  |  |  |  |
| --- | --- | --- | --- |
| **實驗名稱** | **Arduino lab04** | | |
| **組別** | **電通二甲** | **組員** | **06050136陳庭薇** |

1. **實驗目的**

* **了解七段顯示器之顯示原理**

1. **實驗步驟**

CHECK POINT 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秒會跳一個數字，從0000->0059->0100->跳至2359後歸零**

CHECK POINT 2

* **於PIN6按鍵，按下後歸零顯示**
* **於PIN16接一個按鍵，按下後到數計時**

CHECK POINT 3

* **於PIN6按鍵，按下後歸零顯示**
* **於PIN16按下後到數計時，放開後恢復正數**

1. **程式碼**

CHECK POINT 1

#include &quot;SevSeg.h&quot;

SevSeg sevseg;

int Hour = 0, Min = 1;

void setup()

{

byte numDigits = 4;

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

byte segmentPins[] = {8, 9, 10, 11, 12, 13, 14, 15};

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

sevseg.setBrightness(90);

}

void loop()

{

sevseg.setNumber(Hour \* 100 + Min, 4);

Min = Min + 1;

if (Min == 60)

{

Min = 0;

if (Hour == 23)

Hour = 0;

else

Hour++;

}

for(int i=0;i&lt;1990909;i++)

sevseg.refreshDisplay();

}

CHECK POINT 2

#include &quot;SevSeg.h&quot;

SevSeg sevseg;

int Hour = 0, Min = 1 , sw = 6;

void setup()

{

pinMode(sw, INPUT);

byte numDigits = 4;

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

byte segmentPins[] = {8, 9, 10, 11, 12, 13, 14, 15};

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

sevseg.setBrightness(90);

}

void loop()

{

if (digitalRead(6) == HIGH)

{

Hour = 0;

Min = 0;

}

else

{

sevseg.setNumber(Hour \* 100 + Min, 4);

Min = Min + 1;

if (Min == 60)

{

Min = 0;

if (Hour == 23)

Hour = 0;

else

Hour++;

}

}

for (int i = 0; i &lt; 1000000; i++)

sevseg.refreshDisplay();

}

CHECK POINT 3

#include "SevSeg.h"

SevSeg sevseg; //Instantiate a seven segment object

int Hour = 0, Min = 1,sw=6,x=16;

void setup()

{

pinMode(sw, INPUT);

pinMode(x,INPUT);

byte numDigits = 4;

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

byte segmentPins[] = {8, 9, 10, 11, 12, 13, 14, 15};

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

sevseg.setBrightness(90);

}

void loop()

{

if (digitalRead(6)==HIGH)

{

Hour=0;

Min=0;

}

else if(digitalRead(16)==LOW)

{

sevseg.setNumber(Hour \* 100 + Min , 4);

Min = Min - 1;

if ( Min == -1)

{

Min = 59;

if (Hour == -1)

Hour = 23;

else

Hour--;

}

}

else

{

sevseg.setNumber(Hour \* 100 + Min , 4);

Min = Min + 1;

if ( Min == 60)

{

Min = 0;

if (Hour == 23)

Hour = 0;

else

Hour++;

}

}

for (int i = 0; i < 1000000; i++)

sevseg.refreshDisplay();

}

1. **實驗結果及分析**

CHECK POINT **1**

**用if判斷數值跳到多少，如果小時的部分等於23時就跳0，分的位置等於60就跳0。**

CHECK POINT **2**

**用if判斷，如果PIN6按鍵按下，數值歸零顯示，若沒有按下即跟**CHECK POINT **1一樣。若PIN16接通，loop迴圈裡的值一次減一。**

CHECK POINT **3**

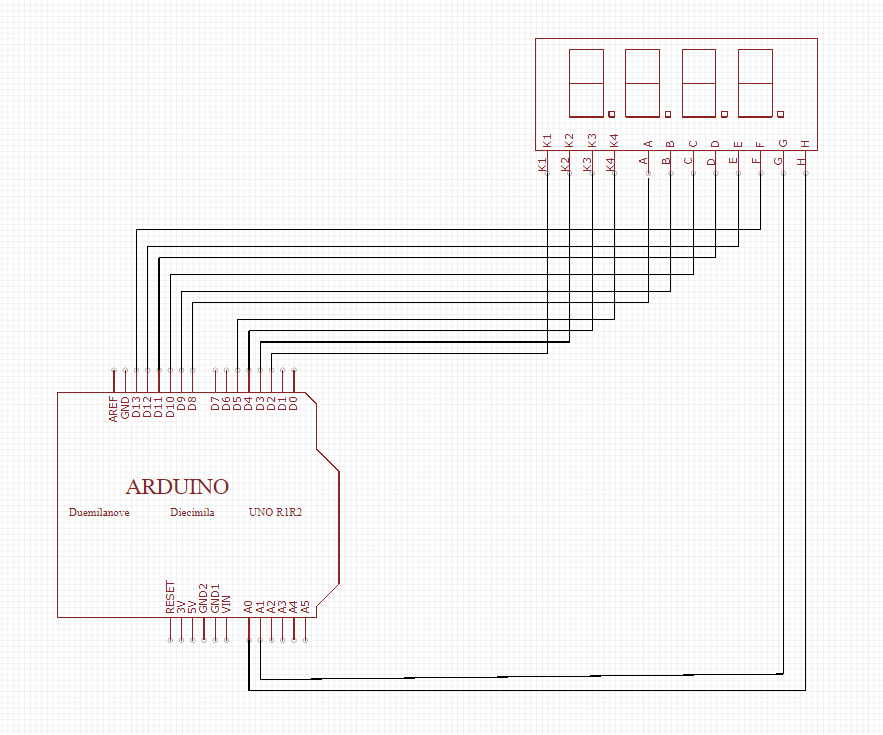
**用if判斷，如果PIN6按鍵按下，數值歸零顯示，若沒有按下小時的部分等於23時就跳0，分的位置等於60就跳0。若PIN16接通，loop迴圈裡的值一次減一。**

1. **心得討論**

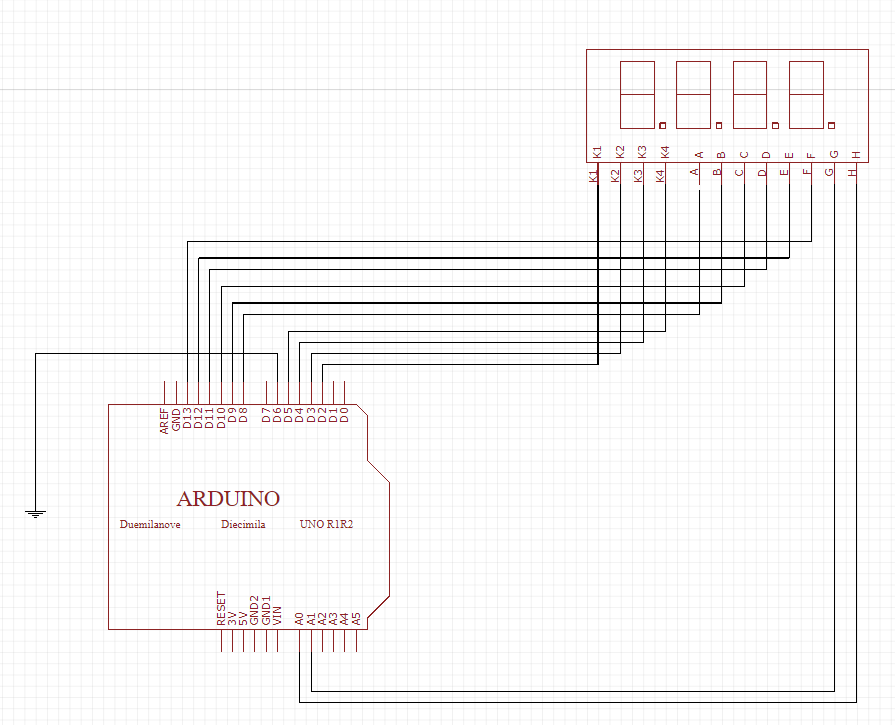
**這次的實驗好複雜，很多數值都要一個一個試，而且題目一樣很多，真的有變難的感覺，真的要認真聽認真做才能跟得上進度。但是其實做出一個奇搭剩下的都蠻像的，而且都做出來很有成就感。**

1. **修正電路圖**

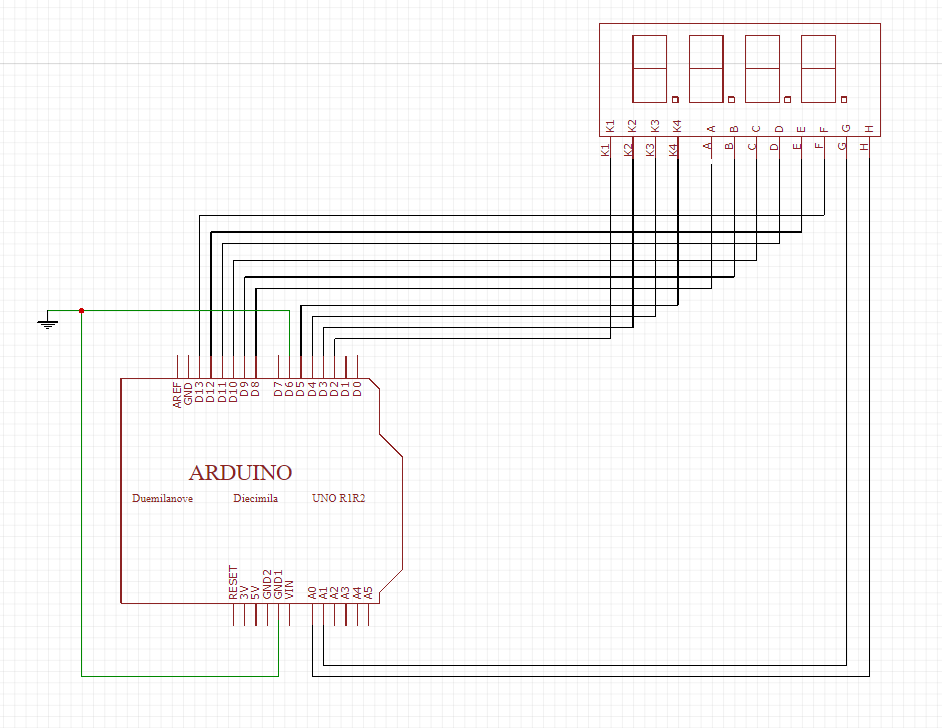
CHECK POINT 1



CHECK POINT 2



CHECK POINT 3

**修正程式碼**

CHECK POINT 1

#include &quot;SevSeg.h&quot;

SevSeg sevseg;

int Hour = 0, Min = 1;

void setup()

{

byte numDigits = 4;

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

byte segmentPins[] = {8, 9, 10, 11, 12, 13, 14, 15};

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

sevseg.setBrightness(90);

}

void loop()

{

sevseg.setNumber(Hour \* 100 + Min, 4);

Min = Min + 1;

if (Min == 60)

{

Min = 0;

if (Hour == 23)

Hour = 0;

else

Hour++;

}

for(int i=0;i&lt;1990909;i++)

sevseg.refreshDisplay();

CHECK POINT 2

#include &quot;SevSeg.h&quot;

SevSeg sevseg;

int Hour = 0, Min = 1 , sw = 6;

void setup()

{

pinMode(sw, INPUT);

byte numDigits = 4;

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

byte segmentPins[] = {8, 9, 10, 11, 12, 13, 14, 15};

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

sevseg.setBrightness(90);

}

void loop()

{

if (digitalRead(6) == HIGH)

{

Hour = 0;

Min = 0;

}

else

{

sevseg.setNumber(Hour \* 100 + Min, 4);

Min = Min + 1;

if (Min == 60)

{

Min = 0;

if (Hour == 23)

Hour = 0;

else

Hour++;

}

}

for (int i = 0; i &lt; 1000000; i++)

sevseg.refreshDisplay();

}

CHECK POINT 3

#include "SevSeg.h"

SevSeg sevseg; //Instantiate a seven segment object

int Hour = 0, Min = 1,sw=6,x=16;

void setup()

{

pinMode(sw, INPUT);

pinMode(x,INPUT);

byte numDigits = 4;

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

byte segmentPins[] = {8, 9, 10, 11, 12, 13, 14, 15};

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

sevseg.setBrightness(90);

}

void loop()

{

if (digitalRead(6)==HIGH)

{

Hour=0;

Min=0;

}

else if(digitalRead(16)==LOW)

{

sevseg.setNumber(Hour \* 100 + Min , 4);

Min = Min - 1;

if ( Min == -1)

{

Min = 59;

if (Hour == -1)

Hour = 23;

else

Hour--;

}

}

else

{

sevseg.setNumber(Hour \* 100 + Min , 4);

Min = Min + 1;

if ( Min == 60)

{

Min = 0;

if (Hour == 23)

Hour = 0;

else

Hour++;

}

}

for (int i = 0; i < 1000000; i++)

sevseg.refreshDisplay();

}