**電通二甲微處理器實驗 實驗結報**

|  |  |  |  |
| --- | --- | --- | --- |
| **實驗名稱** | 四合一七段顯示器 | | |
| **組別** | **6** | **組員** | **李哲儀04053096/沈毅03050654** |

1. **實驗目的:**

了解七段顯示器之顯示原理 ， 共陰極與共陽極七段顯示器之不同 ，還有讓七段顯示器顯示 0 – F ，控制四合一七段顯示，令四合一七段顯示器顯示四位數字。

1. **實驗步驟**

**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

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

1. **程式碼**

|  |
| --- |
| #define PIN\_0 2 #define PIN\_g 3 #define PIN\_c 4 #define PIN\_h 5 #define PIN\_d 6 #define PIN\_e 7 #define PIN\_b 8 #define PIN\_1 9 #define PIN\_2 10 #define PIN\_f 11 #define PIN\_a 12 #define PIN\_3 13 // 共有4個七段顯示器，分別由針腳PIN\_0、PIN\_1、PIN\_2、PIN\_3控制 // 七段顯示器裡有8個LED（包含小數點） #define POS\_NUM 4 #define SEG\_NUM 8 const byte pos\_pins[POS\_NUM] = {PIN\_0, PIN\_1, PIN\_2, PIN\_3}; const byte seg\_pins[SEG\_NUM] = {PIN\_a, PIN\_b, PIN\_c, PIN\_d, PIN\_e, PIN\_f, PIN\_g, PIN\_h}; // 底下定義由七段顯示器顯示數字時所需要的資料 #define t true #define f false const boolean data[10][SEG\_NUM] = { {t, t, t, t, t, t, f, f}, // 0 {f, t, t, f, f, f, f, f}, // 1 {t, t, f, t, t, f, t, f}, // 2 {t, t, t, t, f, f, t, f}, // 3 {f, t, t, f, f, t, t, f}, // 4 {t, f, t, t, f, t, t, f}, // 5 {t, f, t, t, t, t, t, f}, // 6 {t, t, t, f, f, f, f, f}, // 7 {t, t, t, t, t, t, t, f}, // 8 {t, t, t, t, f, t, t, f}, // 9 }; // 一支方便的函式，以格式字串輸出到序列埠 void pf(const char \*fmt, ... ){ char tmp[128]; // max is 128 chars va\_list args; va\_start (args, fmt ); vsnprintf(tmp, 128, fmt, args); va\_end (args); Serial.print(tmp); } // 設定某個七段顯示器所顯示的數字， // 參數pos為0~3，指出想要更新哪一個七段顯示器， // 參數n為0~9，顯示數字 void setDigit(int pos, int n){ if(pos < 0 || 3 < pos){ pf("error pos=%d\n", pos); return; } // 控制想要更新哪一個七段顯示器，將其腳位設為LOW // 其他腳位則設為HIGH，代表不更新。 for(int p = 0; p < POS\_NUM; p++){ if(p == pos) digitalWrite(pos\_pins[p], LOW); else digitalWrite(pos\_pins[p], HIGH); } // 寫入數字 if(0 <= n && n <= 9){ for(int i = 0; i < SEG\_NUM; i++){ digitalWrite(seg\_pins[i], data[n][i] == t ? HIGH : LOW); } } else{ for(int i = 0; i < SEG\_NUM; i++){ digitalWrite(seg\_pins[i], LOW); } digitalWrite(PIN\_h, HIGH); pf("error pos=%d, n=%d\n", pos, n); } } // 設定整個四合一型七段顯示器想要顯示的數字 // 參數number的範圍應是0~9999 void setNumber(int number) { int n0, n1, n2, n3; n3 = number / 1000; number %= 1000; n2 = number / 100; number %= 100; n1 = number / 10; n0 = number % 10; // 求出每個位數的值後，分別更新 // 注意，此處以delay(5)隔開每個位數的更新 setDigit(0, n0); delay(5); setDigit(1, n1); delay(5); setDigit(2, n2); delay(5); setDigit(3, n3); delay(5); } unsigned long time\_previous; void setup() { pinMode(0,INPUT); pinMode(1,INPUT); Serial.begin(115200); for(int i = 0; i < POS\_NUM; i++){ pinMode(pos\_pins[i], OUTPUT); digitalWrite(pos\_pins[i], HIGH); } for(int i = 0; i < SEG\_NUM; i++){ pinMode(seg\_pins[i], OUTPUT); digitalWrite(seg\_pins[i], LOW); } time\_previous = millis(); } int number = 0; int i = 0 ; void loop() { // 經過一秒後就讓number加1 unsigned long time\_now = millis(); if(digitalRead(0) == 0) { number = 0; i=0; } if(digitalRead(1) == 1){ if(time\_now - time\_previous > 1000){ if(number <2359){ if(i==0){ number+=59; time\_previous += 300; pf("number=%d\n", number); i=1; } else{ number+=41; i=0; time\_previous += 300; } } else{ number = 0; i=0; } } } else if(time\_now - time\_previous > 1000){ if(number >0){ if(i==0){ number-=41; time\_previous += 300; pf("number=%d\n", number); i=1; } else{ number-=59; i=0; time\_previous += 300; } } else{ number = 2359; i=1; } } // 不斷地寫入數字 setNumber(number); } |

1. **實驗結果及分析**
2. **心得討論**

**這一次的作業挺簡單的，之前上數位系統就有做過類似的，所以可以駕輕就熟。**

1. **修正電路圖**
2. **修正程式碼**