

電路裝配實習

上數器

學生：田捷林
老師：呂景富、李裕誠 老師

實習日期：112年度第1學期
製作日期：01/29(一)/2024

程式解說：

一開始這個程式的設計只有上數功能而已，
之後做完自己才決定將上數跟下數結合在一起，並且用按鈕開關做控制。
本次用到的有6個按鈕和一個七段顯示器。

在Arduino模組的右下腳有6個按鈕，由左到右所設定的功能為：
上數、暫停、下數、數字加一、數字減一和歸零。
而程式執行初始直為”0000”，利用上述功能控制七段顯示器顯示的數字大小。

上數器程式：

```
1 // 4位數7段顯示器
2 // 1 4 5 12 對應:千、百、十、個位數。
3 // 9 小數點
4 // 2 a
5 // 3 f
6 // 6 b
7 // 7 e
8 // 8 d
9 // 10 c
10 // 11 g
11 //-----
12
13 int sega[] = {0, 1, 2, 3, 12, 5, 6, 7};
14 int seg4[] = {8, 9, 10, 11};
15 int i, j, a, b, c, x;
16 int d3, d2, d1, d0;
17 byte num[] = {B11000000, B11111001, B10100100, B10110000, B10011001, B10010010, B10000010, B11111000, B10000000, B10010000};
```

因為這次是用外接七段顯示器去做，所以1 ~ 11 行為七段顯示器的腳位。
接續的13 ~ 17行為變數設定。

上數器程式：

```
124 }  
125 h3{  
126   font-size: 22px;  
127   color: #8e8e8e;  
128   font-family: 'montserratregular';  
129 }  
130 }  
  
19 void setup() {  
20   for (i = 0; i < 8; i++) {  
21     pinMode(sega[i], OUTPUT);  
22     pinMode(seg4[i], OUTPUT);  
23   }  
24   d3 = 0; d2 = 0; d1 = 0; d0 = 0;  
25  
26   pinMode(47, INPUT);  
27   pinMode(46, INPUT);  
28   pinMode(45, INPUT);  
29   pinMode(44, INPUT);  
30   pinMode(43, INPUT);  
31   pinMode(42, INPUT);  
32 }  
33
```

19行開始的[void setup(){}]為程式初始設定，分別為
20～23行：設定LED狀態(熄滅)；
24行：七段顯示器四位數顯示”0000”；
26～31行：宣告並啟用按鈕。

上數器程式：

34行開始到之後的程式結尾皆為[void loop(){}],
代表程式重複執行的地方。
35是一個函數，在之後程式尾端會看到[scan()]的用途。

36 ~ 38簡單說就是控制程式上數的速度，
控制方法為設定一變數，用[while()]的特性讓程式再回圈裡多跑幾次，
拖延執行速度。

39 ~ 58 為上數功能的程式，利用[if()]來決定是否要執行裡面的程式，
第一次按下按鈕，if判斷式的”digitalRead(47)==0”成立，
執行內部程式一次。
又因程式要按一次自動上數，所以在if內部的程式加一個變數b，
判斷程式加上“or b>=1”，
則當b變數+1時，將會達成另外一個執行條件”b>=1”，
這麼一來這段程式程式將會無限循環執行下去。

40行的”c=0”是暫停下數(後面會講到)。

```
34 void loop() {  
35   scan();  
36   a++;  
37   while (a > 2) {  
38     a = 0;  
39     if (digitalRead(47) == 0 or b >= 1) {  
40       c=0;  
41       b++;  
42       d0++;  
43       if (d0 > 9) {  
44         d1++;  
45         d0 = 0;  
46       }  
47       if (d1 > 9) {  
48         d2++;  
49         d1 = 0;  
50       }  
51       if (d2 > 9) {  
52         d3++;  
53         d2 = 0;  
54       }  
55       if (d3 > 9) {  
56         d3 = 0;  
57       }  
58     }
```


上數器程式：

42 行：個位數字(d0)加一。

43 ~ 46 行：判斷個位數是否大於9；

如果有，則將十位數(d1)加一，個位數歸零，
並跳脫if，執行其餘程式。

如果沒有則繼續判斷下一個條件。

接續的47 ~ 57 皆為相同的邏輯。

```
34 void loop() {  
35   scan();  
36   a++;  
37   while (a > 2) {  
38     a = 0;  
39     if (digitalRead(47) == 0 or b >= 1) {  
40       c=0;  
41       b++;  
42       d0++;  
43       if (d0 > 9) {  
44         d1++;  
45         d0 = 0;  
46       }  
47       if (d1 > 9) {  
48         d2++;  
49         d1 = 0;  
50       }  
51       if (d2 > 9) {  
52         d3++;  
53         d2 = 0;  
54       }  
55       if (d3 > 9) {  
56         d3 = 0;  
57       }  
58     }
```


上數器程式：

這段為下數程式，
邏輯與上數程式幾乎相同，差別只有：
所有的”加”要變成”減”，
且判斷式的>9要改成<0，
與在if裡的dX=0要改成dX=9

```
124 }
125 h3{
126   font-size: 22px;
127   color: #8e8e8e;
128   font-family: 'montserratregular';
129 }
130 }
131
132 em.mail{
133   background: url(../img/mailico.png) no-repeat center;
134   display: inline-block;
135   width: 12px;
136   height: 14px;
137   float: left;
138   margin: 2px 7px 0 0;
139 }
140 em.phone{
141   background: url(../img/phonetico.png) no-repeat center;
142   display: inline-block;
143   width: 20px;
144   height: 18px;
145   float: left;
146   margin: 3px 8px 0 0;
147 }
```

```
59   if (digitalRead(45) == 0 or c >= 1) {
60       b = 0;
61       c++;
62       d0--;
63       if (d0 < 0) {
64           d1--;
65           d0 = 9;
66       }
67       if (d1 < 0) {
68           d2--;
69           d1 = 9;
70       }
71       if (d2 < 0) {
72           d3--;
73           d2 = 9;
74       }
75       if (d3 < 0) {
76           d3 = 9;
77       }
78   }
```


上數器程式：

79 ~ 82 行為暫停按鈕的程式，因為前兩個的自動上下數都有設定變數，好讓if可以一直判斷為“符合條件”，並執行if內部的程式。然而這種程式只要不小心把變數歸零，他就無法繼續執行了，所以暫停的部分就看好這個程式的弱點“變數”，在變數上動手腳；令46號這顆按鈕按下時，將b(自動上數變數)與c(自動下數變數)歸零，則上下數程式就會直接暫停掉了。(因為if的條件不再成立)

```
79  if (digitalRead(46) == 0) {
80      b = 0;
81      c = 0;
82  }
83
84
85  if (digitalRead(44) == 0) {
86      d0++;
87      if (d0 > 9) {
88          d1++;
89          d0 = 0;
90      }
91      if (d1 > 9) {
92          d2++;
93          d1 = 0;
94      }
95      if (d2 > 9) {
96          d3++;
97          d2 = 0;
98      }
99      if (d3 > 9) {
100          d3 = 0;
101      }
102  }
```


上數器程式：

85 ~ 102行為數字加一
原理與上數一模一樣，只是我們將變數拿掉，不再讓if條件一直成立。
這麼一來，就只有按下按鈕時才會執行該段的程式。

```
124 }
125 h3{
126   font-size: 22px;
127   color: #8e8e8e;
128   font-family: 'montserratregular';
129 }
130 }
131
132 em.mail{
133   background: url(../img/mailico.png) no-repeat center;
134   display: inline-block;
135   width: 12px;
136   height: 14px;
137   float: left;
138   margin: 2px 7px 0 0;
139 }
140 em.phone{
141   background: url(../img/phonetico.png) no-repeat center;
142   display: inline-block;
143   width: 20px;
144   height: 18px;
145   float: left;
146   margin: 3px 8px 0 0;
```

```
79   if (digitalRead(46) == 0) {
80     b = 0;
81     c = 0;
82   }
83
84
85   if (digitalRead(44) == 0) {
86     d0++;
87     if (d0 > 9) {
88       d1++;
89       d0 = 0;
90     }
91     if (d1 > 9) {
92       d2++;
93       d1 = 0;
94     }
95     if (d2 > 9) {
96       d3++;
97       d2 = 0;
98     }
99     if (d3 > 9) {
100       d3 = 0;
101     }
102   }
```


上數器程式：

103 ~ 125行為數字減一
原理也與下數一模一樣，我們也將變數拿掉，不再讓if條件一直成立。
這麼一來，就只有按下按鈕時才會執行該段的程式。

```
103 if (digitalRead(43) == 0) {
104     d0--;
105     if (d0 < 0) {
106         d1--;
107         d0 = 9;
108     }
109     if (d1 < 0) {
110         d2--;
111         d1 = 9;
112     }
113     if (d2 < 0) {
114         d3--;
115         d2 = 9;
116     }
117     if (d3 < 0) {
118         d3 = 9;
119     }
120 }
121 if (digitalRead(42) == 0) {
122     d3 = 0; d2 = 0; d1 = 0; d0 = 0;
123 }
124 }
125 }
```


上數器程式：

這段開始到結束就是前面有提到的
[scan()]函數，
主要是在設定七段顯示器的動作

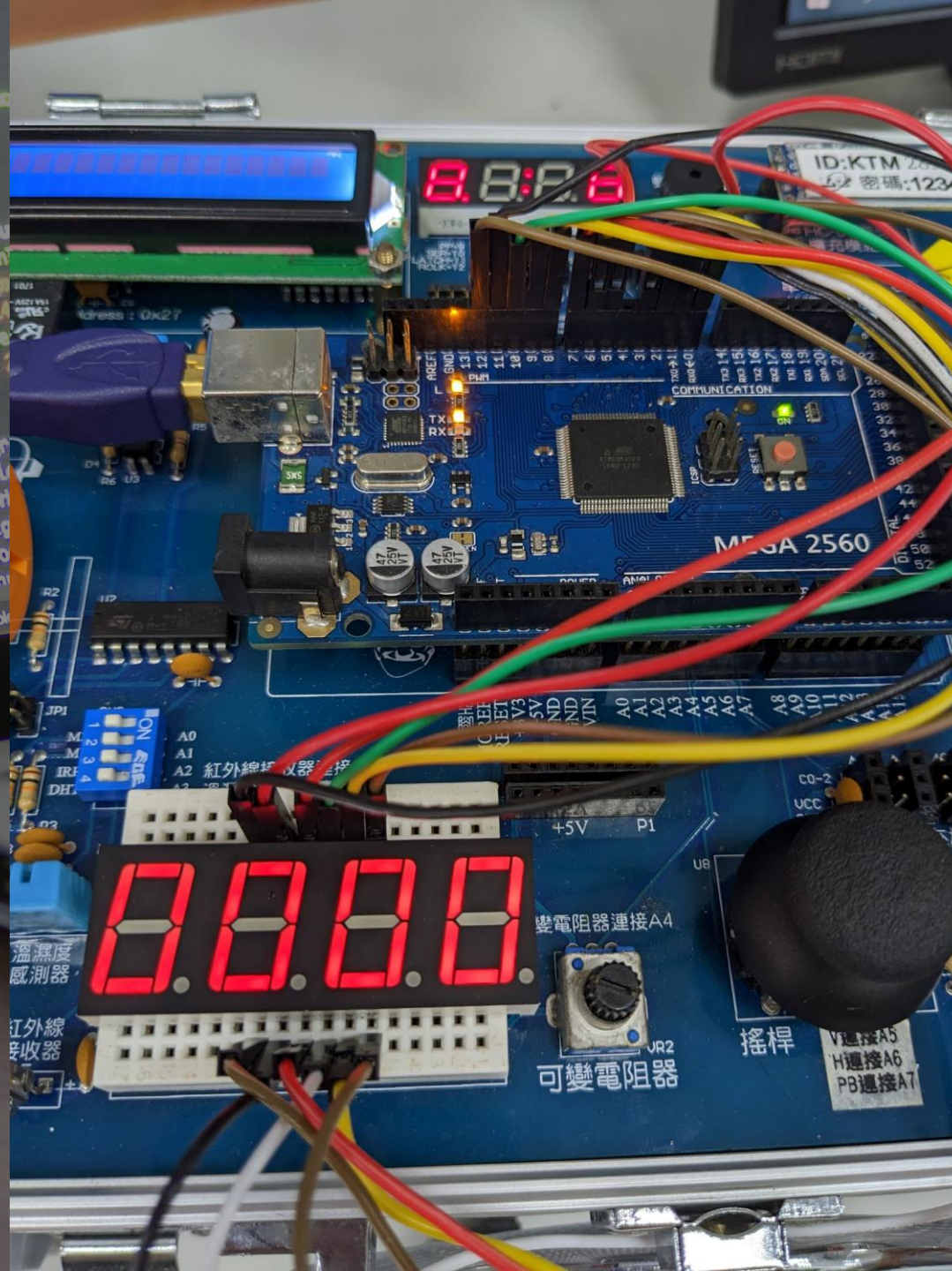
```
124 }
125 h3{
126   font-size: 22px;
127   color: #808080;
128   font-family: 'montserratregular';
129 }
130 }
131
132 em.mail{
133   background: white;
134   display: inline-block;
135   width: 120px;
136   height: 40px;
137   float: left;
138   margin: 2px 7px;
139 }
140 em.phone{
141   background: white;
142   display: inline-block;
143   width: 120px;
144   height: 40px;
145   float: left;
146   margin: 2px 7px;
147 }
```

```
128 void scan() {
129   digitalWrite(8, 1);
130   for (j = 0; j < 8; j++) {
131     if (bitRead(num[d0], j) == 1)
132       digitalWrite(sega[j], 1);
133     else
134       digitalWrite(sega[j], 0);
135   }
136   delay(5);
137   digitalWrite(8, 0);
138   digitalWrite(9, 1);
139   for (j = 0; j < 8; j++) {
140     if (bitRead(num[d1], j) == 1)
141       digitalWrite(sega[j], 1);
142     else
143       digitalWrite(sega[j], 0);
144   }
145   delay(5);
146   digitalWrite(9, 0);
147   digitalWrite(10, 1);
```

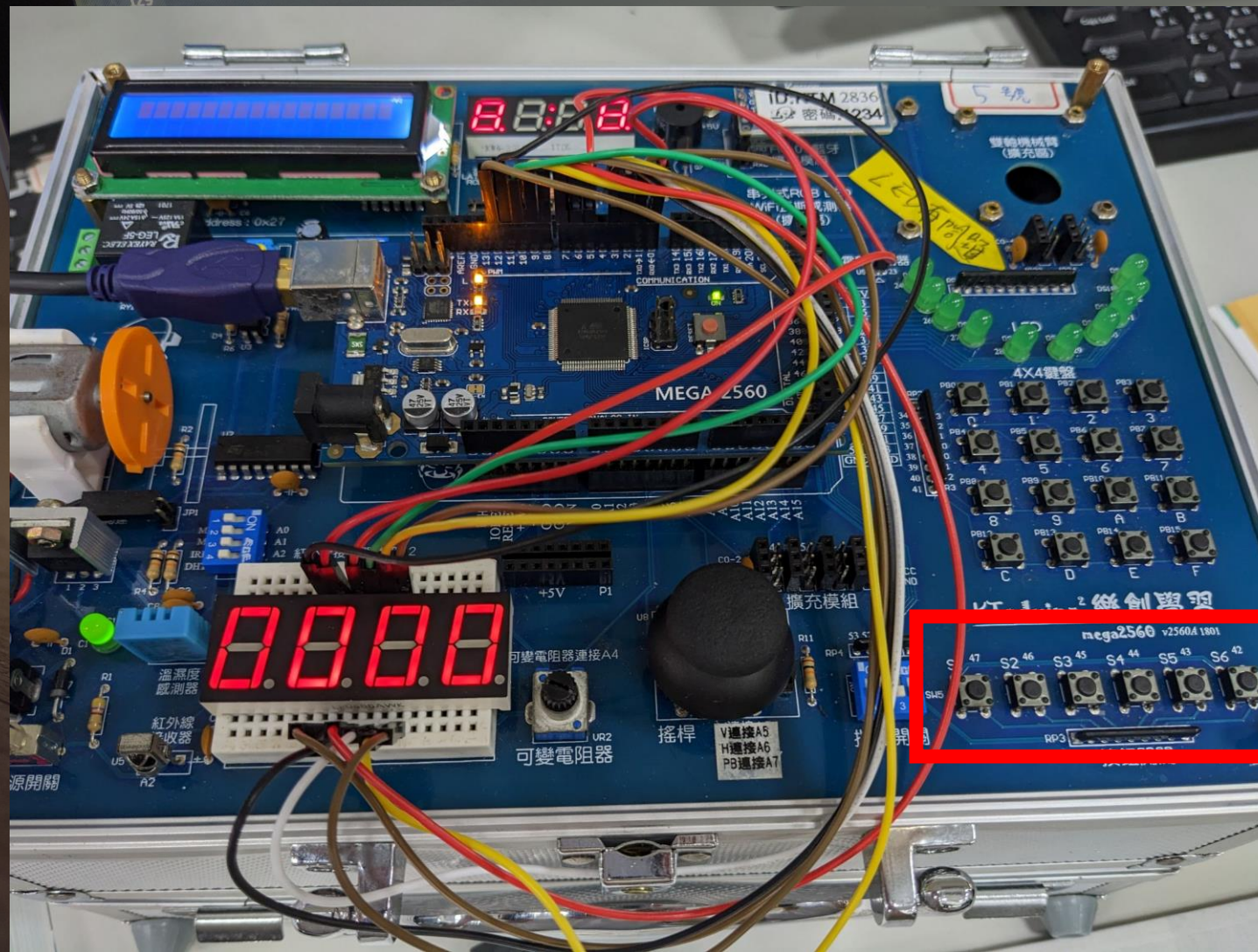
```
148   for (j = 0; j < 8; j++) {
149     if (bitRead(num[d2], j) == 1)
150       digitalWrite(sega[j], 1);
151     else
152       digitalWrite(sega[j], 0);
153   }
154   delay(5);
155   digitalWrite(10, 0);
156   digitalWrite(11, 1);
157   for (j = 0; j < 8; j++) {
158     if (bitRead(num[d3], j) == 1)
159       digitalWrite(sega[j], 1);
160     else
161       digitalWrite(sega[j], 0);
162   }
163   delay(5);
164   digitalWrite(11, 0);
165 }
```


上數器成果：

外接七段顯示器



上數器成果：



按鈕位置

模組全貌

上數器成果(影片)



影片請點下面網址：

可程式邏輯設計實習 上數器1

<https://youtu.be/IzSj2RqzArI>

可程式邏輯設計實習 上數器2

<https://youtu.be/Z-8tss8FwUQ>

心得：

其實這個難在於最一開始的階段，如果最一開始有成功寫出來的話，後面基本上都是複製，然後改一些些地方而已。至少我的感覺是這樣；當時在寫自動上數的時候花了蠻常一段時間的，但寫完後就變順很多！基本上後面寫的時間加起來還不到第一段自動上數的時間。

總之，這次這個程式算是簡單一些，用到的都是基本的for迴圈和if判斷式，但行數會比較多一點，因為程式很多都長得幾乎差不多，當時修改時我常常會看錯行數，導致越改越錯的情況出現……所以要看好修改的地方是不是正確的，不然會容易改錯地方。