作業四： Key_Matix 控制實驗

資工三乙 406262199 陳奕帆
資工三乙 406262216 劉品萱

繳交日期：_____

1．問題

Q1：ADP- WT58F2C9 實驗板上有兩個 Key Matrices？其維度(dimension)分別為多少？ (20%)

一個為二維，是由 KEY0~KEY7 所組成

另一個為一維，是由 KEY8~KEY11 所組成

Q2：同上，兩個 Key Matrices 對應之 Switch 編號之範圍為何？(20%)

4x4 key Matrices 所對應到的 Switch 為 SW2~SW17

4x1 key Matrices 所對應到的 Switch 為 SW18~SW21

KEY0 對應的 SW 為 2、6、10、14　　KEY4 對應的 SW 為 2、3、4、5

KEY1 對應的 SW 為 3、7、11、15　　KEY5 對應的 SW 為 6、7、8、9

KEY2 對應的 SW 為 4、8、12、16　　KEY6 對應的 SW 為 10、11、12、13

KEY3 對應的 SW 為 5、9、13、17　　KEY7 對應的 SW 為 14、15、16、17

KEY8 對應的 SW 為 21　　KEY9 對應的 SW 為 20

KEY10 對應的 SW 為 19　　KEY11 對應的 SW 為 18

Q3：請說明 Push Button 如何作用？(10%)

未按下去時，線路是沒有接通的，呈現斷電狀態。

按下去時，線路是接通的，呈現通電狀態。

Q4： adp-wt58f2c9_v10_0714 文件之電路圖中，請找到匯流排(bus) key_[0:11]，請分別說明 KEY0~KEY_3 之作用為何？KEY4~KEY7 之作用為何？KEY8~KEY_11 之作用為何？ (30%)

原本電路都是 1

KEY_0~KEY_3 ➔ 輸出 0，按下後有電位差，所以電路接通

KEY_4~KEY_7 ➔ 當 Push button 被按下後，KEY0~3 會傳 0 到 KEY4~7。因為有電位差、電路接通，檢查為哪 row 的哪個 switch 被按下，KEY_4~KEY_7 輸出。

KEY_8~KEY_11 ➔ 當 Push button 被按下後，KEY0~3 會傳 0 到 KEY8~11。檢查為哪一 row 的 switch 被按下，KEY8~11 會輸出。

Q5：ADP- WT58F2C9 實驗板上有 key_[0:11] 匯流排(bus)連接到哪個 GPIO Port？使用哪些位元？ 其 Port 對應到的完整記憶體位址範圍？使用時哪些位元規畫為輸出？哪些位元規畫為輸入？ (20%)

連接到 PortA_GPIO，使用到的是 PortA 0~11 的位元(Key_[0:11]所對應的)

PortA 其完整記憶體位置範圍為 0x001F_6800~ 0x001F_681F。

PortA(0~3)位元規劃為輸出， Port(4~11)位元規劃為輸入 1

2．C 程式碼 – Key Matrix + Dip-Switch+ 7-Seg 控制

計算器程式，請自行設計可輸入兩個 unsigned byte number X 與 Y (0 <= X, Y <=99)可以選擇 + 、- 、* 與 % 的計算， 輸出必須顯示 X、Y 及計算結果。

```
int main()
{
        unsigned int tmp = 0;



        OS_PowerOnDriverInitial();



        DRV_Printf("=========================================\r\n", 0);
        DRV_Printf("      ADP-WT58F2C9 Key Matrix demo program      \r\n", 0);
        DRV_Printf("=========================================\r\n", 0);



        DRV_Printf("Key Matrix testing...\r\n", 0);
        DRV_Printf("Press SW17 then SW16 to EXIT.\r\n", 0);

        GPIO_PTA_FS = 0x0000;
        GPIO_PTA_PADINSEL = 0x0000;

        // Setting for 7LED select
        GPIO_PTA_DIR = 0x0000;
        GPIO_PTA_CFG = 0x0000;
        GPIO_PTA_GPIO = Digit_8;
        // Setting for 7LED number
        GPIO_PTD_DIR = 0x0000;
        GPIO_PTD_CFG = 0x0000;
        GPIO_PTD_GPIO = 0x0000;

        GPIO_PTC_DIR = 0xFFFF;
        GPIO_PTC_CFG = 0x0000;
        GPIO_PTC_FS = 0x0000;
        GPIO_PTC_PADINSEL = 0x0000;

        unsigned int col;
        unsigned int key;
```

```
36          unsigned int index_7LED_NUM[17] = {Number_0, Number_1, Number_2,
    Number_3, Number_4, Number_5, Number_6, Number_7,
37                      Number_8, Number_9, Number_A, Number_b, Number_C,
    Number_d, Number_E, Number_F, Number_Dot };
38          unsigned int index_7LED[8] = {Digit_1, Digit_2, Digit_3, Digit_4, Digit_5, Digit_6,
    Digit_7, Digit_8};
39          unsigned int operator_enter = 0;
40          unsigned int number_enter = 0;
41          unsigned int input[4];
42          unsigned int answer[4];
43          unsigned int output[4][4] = {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};
44          unsigned int operator = 0;
45          unsigned int check = 0;
46          unsigned int myi;
47          unsigned int i = 0, j;
48          unsigned int x = 0, y = 0, k = 0;
49          unsigned int negative = 0;
50          unsigned int tmp2;
51
52          while(1)
53          {
54                  tmp2 = (GPIO_PTC_PADIN >> 2) & 0x1;      //tmp 儲存 dip_switch 的值
                        if(tmp2 == 0)               //dip_switch1 為關  運算模式
55                  {
56                          GPIO_PTD_DIR = 0x0000;
57                          GPIO_PTD_CFG = 0x0000;
58                          GPIO_PTD_GPIO = 0x0000;
59
60                          unsigned int input[4];
61                          unsigned int answer[4];
62                          unsigned int output[4][4] = {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-
    1};
63
64                          key = 0xFF;
65                          GPIO_PTA_DIR = 0x0FF0;
66                          GPIO_PTA_CFG = 0x0000;
67                          for (col=0; col<4; col++)
68                          {
```

```
69                          GPIO_PTA_BS = 0x000F;
70                          GPIO_PTA_BR = 0x0000 | (1 << col);
71                          tmp = ((~GPIO_PTA_PADIN) & 0xFF0) >> 4;
72                          if (tmp > 0)
73                          {
74                                  if (tmp & 0x1)
75                                          key = 0*4 + col;
76                                  else if (tmp & 0x2)
77                                          key = 1*4 + col;
78                                  else if (tmp & 0x4)
79                                          key = 2*4 + col;
80                                  else if (tmp & 0x8)
81                                          key = 3*4 + col;
82                                  else if (tmp & 0x80)
83                                          key = 60; // +
84                                  else if (tmp & 0x40)
85                                          key = 61; // -
86                                  else if (tmp & 0x20)
87                                          key = 62; // *
88                                  else if (tmp & 0x10)
89                                          key = 63; // %
90                                  break;
91                          }
92                  }
93                  if(!number_enter)       //當還沒完成輸入
94                  {
95                          if (key != 0xFF)    //檢驗 key 是否被按下
96                          {
97                                  if(!check)
98                                  {
99                                          input[i] = key;             //每次輸入的 key 值存入
    array 中
100                                         i++;
101                                 }
102                                 check = 1;
103
104                         }
105                         else
```

```
106                             check = 0;
107                         if(i == 4)
108                             number_enter = 1;               //四個數值輸入完成
109                 }
110                 if(number_enter)
111                 {       //四個數字輸入完畢  儲存四種運算
112                     x = ((input[0]+1) % 10)*10 + ((input[1]+1) % 10);
113                     y = ((input[2]+1) % 10)*10 + ((input[3]+1) % 10);
114
115                     if(x < y)               //減法會為負  抓出來判斷
116                     {
117                         answer[0] = x + y;
118                         answer[1] = y - x;
119                         answer[2] = y * x;
120                         answer[3] = x % y;
121
122                         negative = 1;
123
124                         for(j = 0 ; j < 4 ; j++)
125                         {
126                             k = 0;
127                             if ( answer[j] == 0)                //若答案為零，也要
    輸出一個零
128                             {
129                                 output[j][0] = 0;
130                             }
131                             else
132                             {
133                                 while(answer[j] > 0)
134                                 {
135                                     output[j][k] = answer[j] % 10;
136                                     answer[j] = answer[j] / 10;
137                                     k++;            //計算答案儲存到哪一位
138                                 }
139                                 if(j == 1)              //減法答案為負  加上負號
140                                     output[j][k] = 0x4040;
141                             }
142                         }
```

```
143                     }
144                     else
145                     {
146                         answer[0] = x + y;
147                         answer[1] = x - y;
148                         answer[2] = x * y;
149                         answer[3] = x % y;
150 
151                         for(j = 0 ; j < 4 ; j++)
152                         {
153                             k = 0;
154                             if ( answer[j] == 0)
155                             {
156                                 output[j][0] = 0;
157                             }
158                             else
159                             {
160                                 while(answer[j] > 0)
161                                 {
162                                     output[j][k] = answer[j] % 10;
163                                     answer[j] = answer[j] / 10;
164                                     k++;
165                                 }
166                             }
167                         }
168                     }
169                 }
170                 if(number_enter)
171                 {
172                     if (key != 0xFF)
173                     {   //哪種 operator 的 key 被按下 之後要執行
174                         if(!check)
175                         {
176                             operator = key;
177                             operator_enter = operator_enter + 1;
178                         }
179                         check = 1;
180                     }
```

```
                        else
                                check = 0;
                }
                for(myi = 0 ; myi < 1000 ; myi++)
                {
                        for(j = 0 ; j < i ; j++)
                        {
                                GPIO_PTD_GPIO = index_7LED_NUM[(input[j]+1)%10];
                                GPIO_PTA_GPIO = index_7LED[7-j];
                                delay1(50);
                        }
//dip_switch 為 0  不會將結果清為零
                        if(((GPIO_PTC_PADIN >> 2) & 0x1) == 0)
                        {
                                if(operator_enter)
                                {       //判斷 key 被按下將進行哪一種運算
                                        if(operator == 60)
                                        {
                                                for(j = 0 ; j < 4 ; j++)
                                                {
                                                        if(output[0][j] == -1)
                                                                continue;
                                                        GPIO_PTD_GPIO =
index_7LED_NUM[(output[0][j])];             //7-seg 印出的數
                                                        GPIO_PTA_GPIO = index_7LED[j];
//7-seg 在哪個 digital  上顯示
                                                        delay1(50);
                                                }
                                        }
                                        else if(operator == 61)
                                        {
                                                for(j = 0 ; j < 4 ; j++)
                                                {
                                                        if(output[1][j] == -1)
                                                                continue;
                                                        if(output[1][j] == 0x4040)
                                                                GPIO_PTD_GPIO = output[1][j];
                                                        else
```

```
                                                GPIO_PTD_GPIO =
index_7LED_NUM[(output[1][j])];
                                                GPIO_PTA_GPIO = index_7LED[j];
                                                delay1(50);
                                        }
                                }
                                else if(operator == 62)
                                {
                                        for(j = 0 ; j < 4 ; j++)
                                        {
                                                if(output[2][j] == -1)
                                                        continue;
                                                GPIO_PTD_GPIO =
index_7LED_NUM[(output[2][j])];
                                                GPIO_PTA_GPIO = index_7LED[j];
                                                delay1(50);
                                        }
                                }
                                else if(operator == 63)
                                {
                                        for(j = 0 ; j < 4 ; j++)
                                        {
                                                if(output[3][j] == -1)
                                                        continue;
                                                GPIO_PTD_GPIO =
index_7LED_NUM[(output[3][j])];
                                                GPIO_PTA_GPIO = index_7LED[j];
                                                delay1(50);
                                        }
                                }
                                        }
                                }
                        }
                }
                else
                {       //當 dip_switch 切換為 1 時 將結果清為零
                        operator_enter = 0;
                        number_enter = 0;
```

```
253                     check = 0;
254                     operator = 0;
255                     GPIO_PTD_GPIO = 0x0000;
256                     x = 0;
257                     y = 0;
258                     i = 0;
259                     for(j = 0 ; j < 4 ; j++)
260                     {
261                             input[j] = -1;
262                     }
263                     delay1(50);
264             }
265         }
266
267         GPIO_PTA_GPIO = Digit_1;
268         GPIO_PTD_GPIO = Number_8 | Number_Dot;
269
270         DRV_Printf("=========================================\r\n", 0);
271
272         return 0;
273 }
274
```