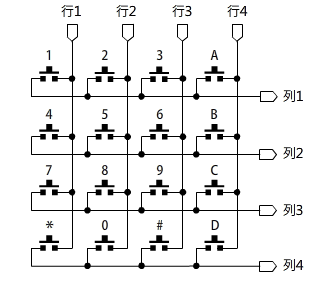
4X4矩陣鍵盤控制一位數七段顯示器

4X4鍵盤分為4個行和4個列。每行(ROW)中的所有按鍵連接在一起，而每列(COL)中的所有按鍵連接在一起。每當我按下按鍵時，行(ROW)與列(COL)之間就會連接，利用中斷配置就知道按下了甚麼按鍵



程式碼如下：

1.定義七段顯示器的顯示編碼

uint32\_t segment[]={0x06, 0x5b, 0x4f, 0x77, // 1 2 3 A

0x66, 0x6d, 0x7D, 0x7c, // 4 5 6 b

0x07, 0x7f, 0x6F, 0x39, // 7 8 9 C

0x79, 0x3f, 0x71, 0x5e}; // E 0 F d

2.定義4X4矩陣鍵盤的腳位

void keypad\_config(void)

{

rcu\_periph\_clock\_enable(RCU\_GPIOB);

gpio\_init(GPIOB, GPIO\_MODE\_IPU, GPIO\_OSPEED\_50MHZ, GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7);

gpio\_init(GPIOB, GPIO\_MODE\_OUT\_PP, GPIO\_OSPEED\_50MHZ, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_8|GPIO\_PIN\_3);

}

3.定義七段顯示器的腳位

void display\_config(void)

{

rcu\_periph\_clock\_enable(RCU\_GPIOD);

gpio\_init(GPIOD, GPIO\_MODE\_OUT\_PP, GPIO\_OSPEED\_50MHZ, GPIO\_PIN\_ALL);

}

4.主程式

int main(void)

{

初始化按鍵

keypad\_config();

初始化七段顯示器

display\_config();

配置SysTick時鐘定時器

systick\_config();

while(1)

{

// 1 2 3 A

偵測第一列

gpio\_port\_write(GPIOB, 0xfe);

將PB0拉低，其餘拉高

gpio\_bit\_write (GPIOB, GPIO\_PIN\_0, RESET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_1, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_8, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_3, SET);

查看按鍵的狀態

if(RESET == gpio\_output\_bit\_get(GPIOB, GPIO\_PIN\_0)){

偵測第一行

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4)){

延遲50ms來消除抖動

delay\_1ms(50);

再次偵測第一行

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4)){

輸出對應的七段顯示器編碼

gpio\_port\_write(GPIOD, segment[0]);

等待按鍵被鬆開

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4));

}

}

偵測第二行

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5)){

gpio\_port\_write(GPIOD, segment[1]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5));

}

}

偵測第三行

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6)){

gpio\_port\_write(GPIOD, segment[2]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6));

}

}

偵測第四行

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7)){

gpio\_port\_write(GPIOD, segment[3]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7));

}

}

}

// 4 5 6 b

偵測第二列

gpio\_port\_write(GPIOB, 0xfd);

將PB1拉低，其餘拉高

gpio\_bit\_write (GPIOB, GPIO\_PIN\_0, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_1, RESET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_8, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_3, SET);

if(RESET == gpio\_output\_bit\_get(GPIOB, GPIO\_PIN\_1)){

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4)){

gpio\_port\_write(GPIOD, segment[4]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5)){

gpio\_port\_write(GPIOD, segment[5]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6)){

gpio\_port\_write(GPIOD, segment[6]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7)){

gpio\_port\_write(GPIOD, segment[7]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7));

}

}

}

// 7 8 9 C

偵測第三列

gpio\_port\_write(GPIOB, 0xfb);

將PB8拉低，其餘拉高

gpio\_bit\_write (GPIOB, GPIO\_PIN\_0, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_1, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_8, RESET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_3, SET);

if(RESET == gpio\_output\_bit\_get(GPIOB, GPIO\_PIN\_8)){

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4)){

gpio\_port\_write(GPIOD, segment[8]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5)){

gpio\_port\_write(GPIOD, segment[9]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6)){

gpio\_port\_write(GPIOD, segment[10]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7)){

gpio\_port\_write(GPIOD, segment[11]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7));

}

}

}

// E 0 F d

偵測第四列

gpio\_port\_write(GPIOB, 0xf7);

將PB3拉低，其餘拉高

gpio\_bit\_write (GPIOB, GPIO\_PIN\_0, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_1, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_8, SET);

gpio\_bit\_write (GPIOB, GPIO\_PIN\_3, RESET);

if(RESET == gpio\_output\_bit\_get(GPIOB, GPIO\_PIN\_3)){

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4)){

gpio\_port\_write(GPIOD, segment[12]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_4));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5)){

gpio\_port\_write(GPIOD, segment[13]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_5));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6)){

gpio\_port\_write(GPIOD, segment[14]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_6));

}

}

else if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7)){

delay\_1ms(50);

if(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7)){

gpio\_port\_write(GPIOD, segment[15]);

while(RESET == gpio\_input\_bit\_get(GPIOB, GPIO\_PIN\_7));

}

}

}

}

}