//Pat McMahon 18/3/2021

//A037-MFS7

//This sketch is a Test Sketch, with no external components added. Just piggy back the Multi-Purpose Shield carefully onto the Uno.

//Important-Make sure the jumper is moved from J1 to J2 for this sketch to work.

//on power up, buzzer will sound continuously, push button 2 to stop.

//Rotating the blue potentiometer, small brass slotted screw, it displays the value of the analogue revolving potentiometer, between 0-1024.

//pressing Push Button Switch 1, momentarily shows 0,1,2,3 on the LED 7 Segment Displays

//pressing Push Button Switch 2 momentarily has the buzzer ringing, keep pushed for beeping.

//pressing Push Button Switch 3 momentarily has the 4 LED's running down, while the 7 segment display stops displaying.

//After releasing Push Button Switch 1,2 & 3 as above, it will return to the previous Analog Value on the 7 Segment Display.

//defining three pins of 74HC595

int latchPin = 4;//ST\_CP

int clockPin = 7;//SH\_CP

int dataPin = 8; //DS

//defining three key input

int key1 = A1; //Push Button Switch 1

int key2 = A2; //Push Button Switch 2

int key3 = A3; //Push Button Switch 3

//buzzer pin

int buzzer = 3;

//pin definition of flowing light

int led1 = 13;

int led2 = 12;

int led3 = 11;

int led4 = 10;

int led5 = 9;

int led6 = 8;

int dat\_wei[4]={0x01,0x02,0x04,0x08}; //LED Segment Displays

//showing 1--4

int dat\_duan[10]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90}; //LED Segment Displays showing 0--9

char i=0;

void setup ()

{

 pinMode(latchPin,OUTPUT);

 pinMode(clockPin,OUTPUT);

 pinMode(dataPin,OUTPUT);

 pinMode(key1,INPUT);

 pinMode(key2,INPUT);

 pinMode(key3,INPUT);

 pinMode(buzzer,OUTPUT);

 pinMode(led1,OUTPUT);

 pinMode(led2,OUTPUT);

 pinMode(led3,OUTPUT);

 pinMode(led4,OUTPUT);

 pinMode(led5,OUTPUT);

 pinMode(led6,OUTPUT);

 for(char i=8;i<14;i++)

 digitalWrite(i,HIGH);

}

void loop()

{

if(digitalRead(key1)==LOW )

 SMG(); //testing LED Segment Displays

if(digitalRead(key2)==LOW )

 buzzer\_(); //testing buzzer

if(digitalRead(key3)==LOW)

 led\_display(); //testing LED

if(digitalRead(key1)==HIGH & digitalRead(key2)==HIGH & digitalRead(key3)==HIGH)

 analog(); //testing analog input

}

void SMG(void)

{

 digitalWrite(latchPin,LOW); //clear LED Segment Displays

 shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

 shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

 digitalWrite(latchPin,HIGH);

while(1)

 {

 digitalWrite(latchPin,LOW);

 //MSBFIRST,transmitting binary bit from high to low,74HC595 starts from first piece, and displaces data from Q0 to Q7. If there is data, it will start from the second piece like this.

 shiftOut(dataPin, clockPin, MSBFIRST ,dat\_duan[i]); //data about second piece

 shiftOut(dataPin, clockPin, MSBFIRST ,dat\_wei[i]); //way of MSBFIRST,data about first piece

 digitalWrite(latchPin,HIGH);

 i++;

 if(i==4){i=0;}

 if(digitalRead(key1)==HIGH)

 {

 digitalWrite(latchPin,LOW); //clear LED Segment Displays

 shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

 shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

 digitalWrite(latchPin,HIGH);

 break;

 }

 }

}

void buzzer\_(void)

{

 char i;

 digitalWrite(latchPin,LOW); //clear LED Segment Displays

 shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

 shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

 digitalWrite(latchPin,HIGH);

 while(1)

 {

 for(i=0;i<80;i++)// output a frequency sound

 {

 digitalWrite(buzzer,LOW);// sound

 delay(1);//delay1ms

 digitalWrite(buzzer,HIGH);//not sound

 delay(1);//ms delay

 }

 for(i=0;i<100;i++)// output a frequency sound

 {

 digitalWrite(buzzer,LOW);// sound

 digitalWrite(buzzer,HIGH);//not sound

 delay(2);//2ms delay

 }

 if(digitalRead(key2)==HIGH)

 {

 digitalWrite(latchPin,LOW); //clear LED Segment Displays

 shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

 shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

 digitalWrite(latchPin,HIGH);

 break;

 }

 }

}

void led\_display()

{

digitalWrite(latchPin,LOW); //clear LED Segment Displays

shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

shiftOut(dataPin, clockPin, MSBFIRST ,0x00);

digitalWrite(latchPin,HIGH);

while(1)

{

 digitalWrite(led1,LOW);

 delay(100);

 digitalWrite(led1,HIGH);

 digitalWrite(led2,LOW);

 delay(100);

 digitalWrite(led2,HIGH);

 digitalWrite(led3,LOW);

 delay(100);

 digitalWrite(led3,HIGH);

 digitalWrite(led4,LOW);

 delay(100);

 digitalWrite(led4,HIGH);

 digitalWrite(led5,LOW);

 delay(100);

 digitalWrite(led5,HIGH);

 digitalWrite(led6,LOW);

 delay(100);

 digitalWrite(led6,HIGH);

 if(digitalRead(key3)==HIGH)

 {

 break;

 }

}

}

void analog()

{

int val,qian,bai,shi,ge;

val=analogRead(A0);

qian=val/1000;

bai=val%1000;

bai=bai/100;

shi=val%100;

shi=shi/10;

ge=val%10;

digitalWrite(latchPin,LOW);

shiftOut(dataPin, clockPin, MSBFIRST ,dat\_duan[qian]);

shiftOut(dataPin, clockPin, MSBFIRST ,0x01);

digitalWrite(latchPin,HIGH);

digitalWrite(latchPin,LOW);

shiftOut(dataPin, clockPin, MSBFIRST ,dat\_duan[bai]);

shiftOut(dataPin, clockPin, MSBFIRST ,0x02);

digitalWrite(latchPin,HIGH);

digitalWrite(latchPin,LOW);

shiftOut(dataPin, clockPin, MSBFIRST ,dat\_duan[shi]);

shiftOut(dataPin, clockPin, MSBFIRST ,0x04);

digitalWrite(latchPin,HIGH);

digitalWrite(latchPin,LOW);

shiftOut(dataPin, clockPin, MSBFIRST ,dat\_duan[ge]);

shiftOut(dataPin, clockPin, MSBFIRST ,0x08);

digitalWrite(latchPin,HIGH);

}