Proyecto #1
Introducción a los Microcontroladores

Fernando Aguilar Sánchez

Nelly Arlet Bautista Hernández
Jorge Gómez Reus

Índice

1. Protoboard

2. Código

```c
#include <mega8535.h>
#include <delay.h>
#define botonIzq PINB.0
#define botonDer PINB.1

bit botonpIzq;
bit botonaIzq;
bit botonpDer;
bit botonaDer;
bit der;
bit izq;

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) | (1<<DDA1) | (1<<DDA0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<
    PORTA0);

// Port B initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<
    PORTB0);

// Port C initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<
    PORTC0);

// Port D initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<
    PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
```

```
56  // Mode: Normal top=0xFFFF
57  // OC1A output: Disconnected
58  // OC1B output: Disconnected
59  // Noise Canceler: Off
60  // Input Capture on Falling Edge
61  // Timer1 Overflow Interrupt: Off
62  // Input Capture Interrupt: Off
63  // Compare A Match Interrupt: Off
64  // Compare B Match Interrupt: Off
65  TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
66  TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
67  TCNT1H=0x00;
68  TCNT1L=0x00;
69  ICR1H=0x00;
70  ICR1L=0x00;
71  OCR1AH=0x00;
72  OCR1AL=0x00;
73  OCR1BH=0x00;
74  OCR1BL=0x00;
75
76  // Timer/Counter 2 initialization
77  // Clock source: System Clock
78  // Clock value: Timer2 Stopped
79  // Mode: Normal top=0xFF
80  // OC2 output: Disconnected
81  ASSR=0<<AS2;
82  TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
83  TCNT2=0x00;
84  OCR2=0x00;
85
86  // Timer(s)/Counter(s) Interrupt(s) initialization
87  TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<
        TOIE0);
88
89  // External Interrupt(s) initialization
90  // INT0: Off
91  // INT1: Off
92  // INT2: Off
93  MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
94  MCUCSR=(0<<ISC2);
95
96  // USART initialization
97  // USART disabled
98  UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
99
100 // Analog Comparator initialization
101 // Analog Comparator: Off
102 // The Analog Comparator's positive input is
103 // connected to the AIN0 pin
104 // The Analog Comparator's negative input is
105 // connected to the AIN1 pin
106 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
107 SFIOR=(0<<ACME);
108
109 // ADC initialization
110 // ADC disabled
111 ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);
112
113 // SPI initialization
114 // SPI disabled
115 SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
116
```

```c
117  // TWI initialization
118  // TWI disabled
119  TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
120
121  while (1)
122          {
123              if (botonIzq==0)
124                botonaIzq=0;
125              else
126                botonaIzq=1;
127
128              if (botonDer==0)
129                botonaDer=0;
130              else
131                botonaDer=1;
132              //Cambio de flanco de 1 a 0 para el boton izquierdo
133              if ((botonpIzq==1)&&(botonaIzq==0)){
134                //PORTA.0=1;
135                izq=1;
136                //Retardo de 40 ms para quitar rebote para el boton izquierdo
137                delay_ms(40);
138              }
139              //Cambio de flanco de 0 a 1   para el boton izquierdo
140              if ((botonpIzq==0)&&(botonaIzq==1))
141               //PORTA.0=0;
142               izq=0;
143               delay_ms(40);
144               //Retardo de 40mS para quitar rebote del boton izquierdo
145
146
147               //Cambio de flanco de 1 a 0 para el boton Derecho
148              if ((botonpDer==1)&&(botonaDer==0)){
149                //PORTA.1=1;
150                der=1;
151                //Retardo de 40 ms para quitar rebote para el boton derecho
152                delay_ms(40);
153              }
154              //Cambio de flanco de 0 a 1 para el boton Derecho
155              if ((botonpDer==0)&&(botonaDer==1))
156              //PORTA.1=0;
157              der=0;
158                delay_ms(40);
159                //Retardo de 40mS para quitar rebote del boton derecho
160                if (der == 1 && izq == 1) {
161                   PORTA.0=0;
162                   PORTA.1=0;
163                } else if (der == 0 && izq == 0) {
164                   PORTA.0=0;
165                   PORTA.1=0;
166                }  else if (der == 0 && izq == 1) {
167                   PORTA.0=1;
168                   PORTA.1=0;
169                } else if (der == 1 && izq == 0) {
170                   PORTA.0=0;
171                   PORTA.1=1;
172                }
173                botonpIzq=botonaIzq;
174                botonpDer=botonaDer;
175
176
177          }
178  }
```

3. Circuito

Figura 1: Simulación