Instituto Politécnico Nacional

Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

Alumnos:

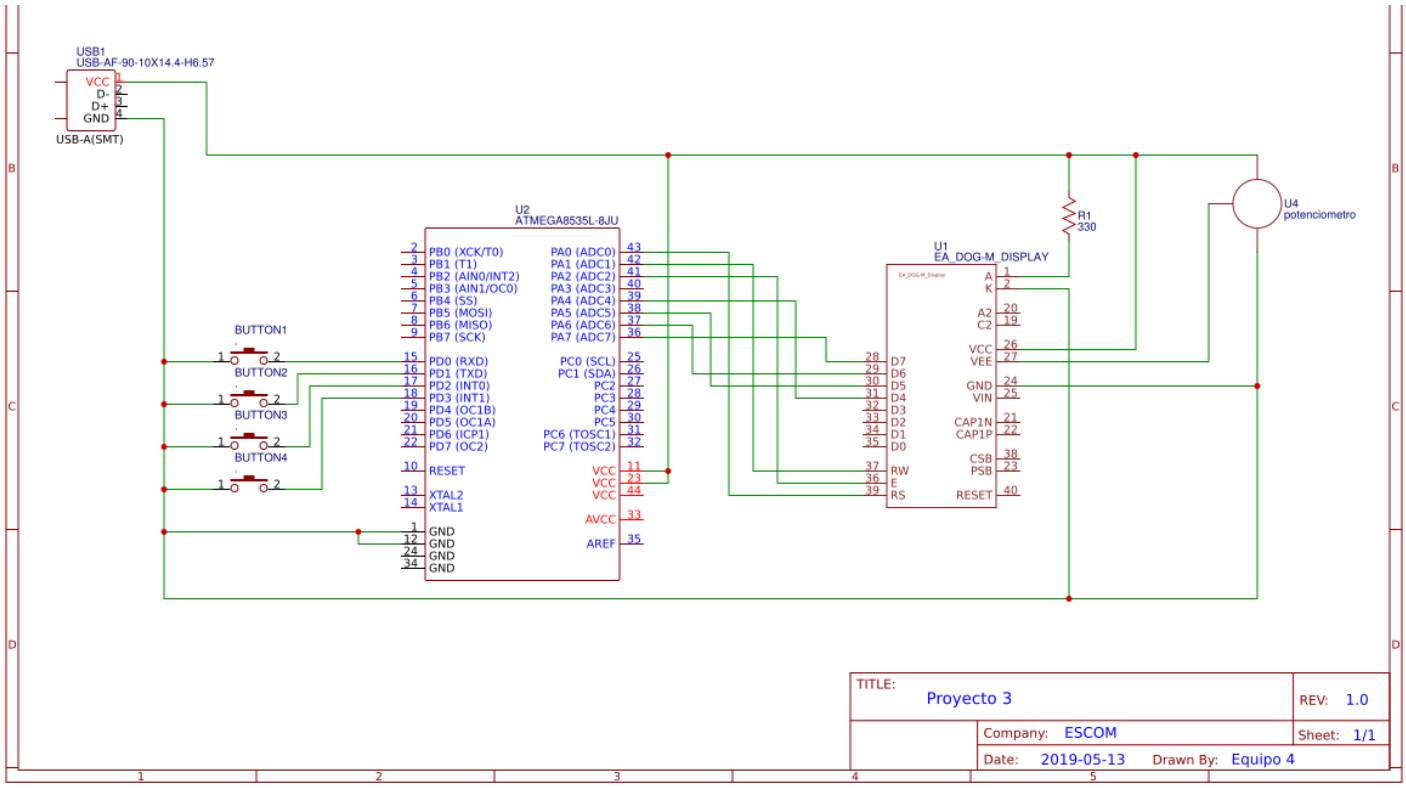
Aldavera Gallaga Iván

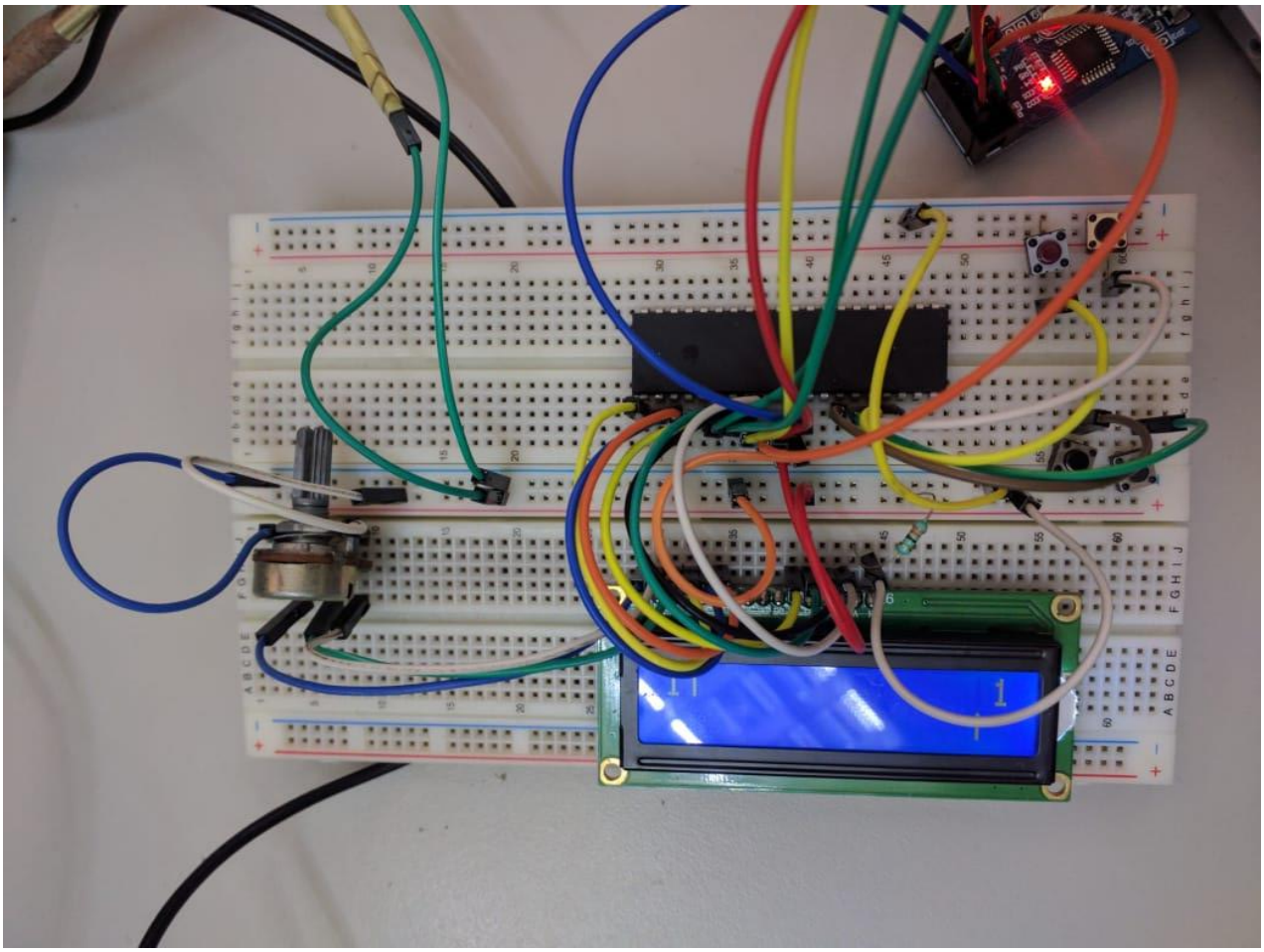
Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Proyecto 3

Ping pong


USB1
USB-AF-90-10X14.4-H6.57
VCC
D-
D+
GND
USB-A(SMT)
U2
ATMEGA8535L-8JU
PB0 (XCK/T0)
PB1 (T1)
PB2 (AIN0/INT2)
PB3 (AIN1/OC0)
PB4 (SS)
PB5 (MOSI)
PB6 (MISO)
PB7 (SCK)
PD0 (RXD)
PD1 (TXD)
PD2 (INT0)
PD3 (INT1)
PD4 (OC1B)
PD5 (OC1A)
PD6 (ICP1)
PD7 (OC2)
RESET
XTAL2
XTAL1
GND
GND
GND
GND
PA0 (ADC0)
PA1 (ADC1)
PA2 (ADC2)
PA3 (ADC3)
PA4 (ADC4)
PA5 (ADC5)
PA6 (ADC6)
PA7 (ADC7)
PC0 (SCL)
PC1 (SDA)
PC2
PC3
PC4
PC5
PC6 (TOSC1)
PC7 (TOSC2)
VCC
VCC
VCC
AVCC
AREF
BUTTON1
BUTTON2
BUTTON3
BUTTON4
U1
EA_DOG-M_DISPLAY
A
K
A2
C2
VCC
VEE
GND
VIN
CAP1N
CAP1P
CSB
PSB
RESET
D7
D6
D5
D4
D3
D2
D1
D0
RW
E
RS
R1
330
U4
potenciometro
TITLE:
Proyecto 3
REV: 1.0
Company: ESCOM
Sheet: 1/1
Date: 2019-05-13
Drawn By: Equipo 4



```
/*****************************************************
This program was created by the
CodeWizardAVR V2.60 Evaluation
Automatic Program Generator
© Copyright 1998-2012 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 16/04/2019
Author  : Equipo 4
Company :
Comments:


Chip type               : ATmega8535L
Program type            : Application
AVR Core Clock frequency: 1,000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 128
*****************************************************/

#include <mega8535.h>

#include <delay.h>
#include <stdlib.h>
// Alphanumeric LCD functions
#include <alcd.h>


#define movi1 PIND.0
#define movd1 PIND.1
#define movi2 PIND.2
#define movd2 PIND.3

int num=0;
int portero;
unsigned char jug1=0,jug2=0;
const char car=48; //codigo ascii

// Declare your global variables here

#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (1<<ADLAR))

// Read the 8 most significant bits
// of the AD conversion result


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

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
```

```
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
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=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) | (1<<PORTD2) | (1<<PORTD1) | (1<<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
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
      // Compare B Match Interrupt: Off
      TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
      TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
      TCNT1H=0x00;
      TCNT1L=0x00;
      ICR1H=0x00;
      ICR1L=0x00;
```

```
107.       OCR1AH=0x00;
108.       OCR1AL=0x00;
109.       OCR1BH=0x00;
110.       OCR1BL=0x00;
111.
112.       // Timer/Counter 2 initialization
113.       // Clock source: System Clock
114.       // Clock value: Timer2 Stopped
115.       // Mode: Normal top=0xFF
116.       // OC2 output: Disconnected
117.       ASSR=0<<AS2;
118.       TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<
    CS21) | (0<<CS20);
119.       TCNT2=0x00;
120.       OCR2=0x00;
121.
122.       // Timer(s)/Counter(s) Interrupt(s) initialization
123.       TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) |
    (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
124.
125.       // External Interrupt(s) initialization
126.       // INT0: Off
127.       // INT1: Off
128.       // INT2: Off
129.       MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
130.       MCUCSR=(0<<ISC2);
131.
132.       // USART initialization
133.       // USART disabled
134.       UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<U
    CSZ2) | (0<<RXB8) | (0<<TXB8);
135.
136.       // Analog Comparator initialization
137.       // Analog Comparator: Off
138.       ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
    (0<<ACIS1) | (0<<ACIS0);
139.
140.       // ADC initialization
141.       // ADC Clock frequency: 500,000 kHz
142.       // ADC Voltage Reference: AVCC pin
143.       // ADC High Speed Mode: Off
144.       // ADC Auto Trigger Source: ADC Stopped
145.       // Only the 8 most significant bits of
146.       // the AD conversion result are used
147.       ADMUX=ADC_VREF_TYPE;
148.       ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<AD
    PS2) | (0<<ADPS1) | (1<<ADPS0);
149.       SFIOR=(1<<ADHSM) | (0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
150.
151.       // SPI initialization
152.       // SPI disabled
153.       SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA)
    | (0<<SPR1) | (0<<SPR0);
154.
155.       // TWI initialization
156.       // TWI disabled
157.       TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
158.
159.       // Alphanumeric LCD initialization
160.       // Connections are specified in the
161.       // Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
```

```
      // RS - PORTB Bit 0
      // RD - PORTB Bit 1
      // EN - PORTB Bit 2
      // D4 - PORTB Bit 4
      // D5 - PORTB Bit 5
      // D6 - PORTB Bit 6
      // D7 - PORTB Bit 7
      // Characters/line: 16
      lcd_init(16);

      while (1)
            {
                jug1=0, jug2=0;
                lcd_clear();
                lcd_gotoxy(4,0);
                lcd_putsf("PING-PONG");
                lcd_gotoxy(6,1);
                lcd_putsf("FAS");
                delay_ms(1500);
                lcd_clear();

                lcd_gotoxy(1,0);
                lcd_putsf("INSTRUCCIONES");
                delay_ms(1000);
                lcd_clear();

                lcd_gotoxy(0,0);
                lcd_putsf("1.Gana quien");
                lcd_gotoxy(0,1);
                lcd_putsf("tenga 9 puntos");
                delay_ms(1800);
                lcd_clear();

                lcd_gotoxy(0,0);
                lcd_putsf("2.Diviertanse");
                delay_ms(1800);
                lcd_clear();

                lcd_gotoxy(2,0);
                lcd_putsf("Creado por:");
                delay_ms(1500);
                lcd_clear();

                lcd_gotoxy(0,0);
                lcd_putsf("Ruben Lara");
                delay_ms(1500);
                lcd_clear();

                lcd_gotoxy(0,0);
                lcd_putsf("Erik Morales");
                delay_ms(1500);
                lcd_clear();

                lcd_gotoxy(0,0);
                lcd_putsf("Ivan Aldavera");
                delay_ms(1500);
                lcd_clear();

                lcd_gotoxy(0,0);
                lcd_putsf("Comienza");
                delay_ms(1500);
```

```
                lcd_clear();

                do{
                int i,x=0,y=1;
                lcd_gotoxy(0,0);
                lcd_putchar(jug1+car);
                lcd_gotoxy(15,0);
                lcd_putchar(jug2+car);

                lcd_gotoxy(5,0);

                num++;
                if(num==4){
                num=0;
                }

                if(num==0){
                for(i=13; i>=2; i--){

                if(movi1==0){
                lcd_gotoxy(1,1);
                lcd_putchar(32);
                portero=1;
                delay_ms(300);
                lcd_gotoxy(1,portero);
                lcd_putchar(124);
                }

                else{
                lcd_gotoxy(1,0);
                lcd_putchar(32);
                portero=0;
                delay_ms(300);
                lcd_gotoxy(1,portero);
                lcd_putchar(124);
                }



                lcd_gotoxy(i,x);
                lcd_putchar(46);
                delay_ms(200);
                lcd_gotoxy(i,x);
                lcd_putchar(32);
                i--;
                lcd_gotoxy(i,y);
                lcd_putchar(46);
                delay_ms(200);
                lcd_gotoxy(i,y);
                lcd_putchar(32);
                if(i==2 && portero==0){
                jug2++;
                }
                }
                }



                else if(num==1){
                for(i=2; i<14; i++){

```

```
284.                if(movi2==0){
285.                lcd_gotoxy(14,1);
286.                lcd_putchar(32);
287.                portero=1;
288.                delay_ms(300);
289.                lcd_gotoxy(14,portero);
290.                lcd_putchar(124);
291.                }
292.
293.                else{
294.                lcd_gotoxy(14,0);
295.                lcd_putchar(32);
296.                portero=0;
297.                delay_ms(300);
298.                lcd_gotoxy(14,portero);
299.                lcd_putchar(124);
300.                }
301.
302.                lcd_gotoxy(i,x);
303.                lcd_putchar(46);
304.                delay_ms(200);
305.                lcd_gotoxy(i,x);
306.                lcd_putchar(32);
307.                i++;
308.                lcd_gotoxy(i,y);
309.                lcd_putchar(46);
310.                delay_ms(200);
311.                lcd_gotoxy(i,y);
312.                lcd_putchar(32);
313.
314.                if(i==13 && portero==0){
315.                jug1++;
316.                }
317.                }
318.                }
319.
320.
321.
322.                else if(num==2){
323.
324.                for(i=13; i>=2; i--){
325.                if(movd1==0){
326.                lcd_gotoxy(1,0);
327.                lcd_putchar(32);
328.                portero=0;
329.                delay_ms(300);
330.                lcd_gotoxy(1,portero);
331.                lcd_putchar(124);
332.                }
333.
334.                else{
335.                lcd_gotoxy(1,1);
336.                lcd_putchar(32);
337.                portero=1;
338.                delay_ms(300);
339.                lcd_gotoxy(1,portero);
340.                lcd_putchar(124);
341.                }
342.
343.                lcd_gotoxy(i,y);
344.                lcd_putchar(46);
```

```
                delay_ms(200);
                lcd_gotoxy(i,y);
                lcd_putchar(32);
                i--;
                lcd_gotoxy(i,x);
                lcd_putchar(46);
                delay_ms(200);
                lcd_gotoxy(i,x);
                lcd_putchar(32);

                if(i==2 && portero==1){
                jug2++;
                }
                }
                }


                else if(num==3){
                for(i=2; i<=13; i++){
                if(movd2==0){
                lcd_gotoxy(14,0);
                lcd_putchar(32);
                portero=0;
                delay_ms(300);
                lcd_gotoxy(14,portero);
                lcd_putchar(124);
                }

                else{
                lcd_gotoxy(14,1);
                lcd_putchar(32);
                portero=1;
                delay_ms(300);
                lcd_gotoxy(14,portero);
                lcd_putchar(124);
                }

                lcd_gotoxy(i,y);
                lcd_putchar(46);
                delay_ms(200);
                lcd_gotoxy(i,y);
                lcd_putchar(32);
                i++;
                lcd_gotoxy(i,x);
                lcd_putchar(46);
                delay_ms(200);
                lcd_gotoxy(i,x);
                lcd_putchar(32);
                if(i==13 && portero==1){
                jug1++;
                }
                }
                }

                   }while(jug2<=9 && jug1<=9);

            }
      }
```