

Nombre: _____ Grupo: _____

Dr. Enrique García Trinidad
Tecnológico de Estudios Superiores de Huixquilucan
<https://enriquegarcia.xyz>
enrique.garcia@tesh.edu.mx

Práctica 8

Comunicación Serial

8.1. Material

El material enlistado es necesario para la realización de la práctica 8.

Ct	Dispositivo	Descripción	Eti.
1	ATmega328P-PU	Microcontrolador AVR RISC 8-bit 20Mhz	U1
1	MAX232	Conversor de señales RS232	IC2
1	Regulador L7805CV	Regulador de voltaje 5V 1 A	IC1
1	Capacitor cerámico de 0.1 μ F 50V	Código: 104	C1
2	Capacitor cerámico de 33pF 50V	Código: 33	C8, C9
1	Cristal de cuarzo 4Mhz	Oscilador	Y1
4	Capacitor electrolítico de 1 μ F 50V	Tolerancia $\pm 20\%$	C4, C5, C6, C7
1	Capacitor electrolítico de 470 μ F 25V	Tolerancia $\pm 20\%$	C2
1	Capacitor electrolítico de 220 μ F 25V	Tolerancia $\pm 20\%$	C3
2	Resistencia de 330 Ω 1/4W	Código: Naranja, naranja, café, oro	R1, R3
1	Resistencia de 10k Ω 1/4W	Código: Café, negro, naranja, oro	R4
1	Conector DB9	Conector DB9 Hembra	J1
1	Led 5mm difuso	Color rojo	LED1
1	Push button (Microswitch)	Tipo push, 4 o 2 terminales	S1
1	Pantalla LCD	16 caracteres, 2 líneas	U2
1	Potenciómetro de 10k	Tipo preset horizontal de 10 mm	R2
2	Metro de alambre para protoboard		
1	Protoboard		
1	Grabador Usbasp	Grabador microcontroladores AVR 8-bit	J3
1	Fuente de alimentación de 12V 2A	Eliminador de voltaje	J2
1	Computadora con dos puertos USB		

8.2. Conexión de los componentes

- Conecte el diagrama de la práctica 8 de acuerdo a la Figura 8.1:

8.3. Ejercicio 1

- Inicie el software Codevision AVR. Cree un nuevo proyecto dando click en el menú [New>Project](#).
- Cuando el software pregunte si queremos usar el asistente [CodeWizardAVR](#) le indicamos que [Si](#).

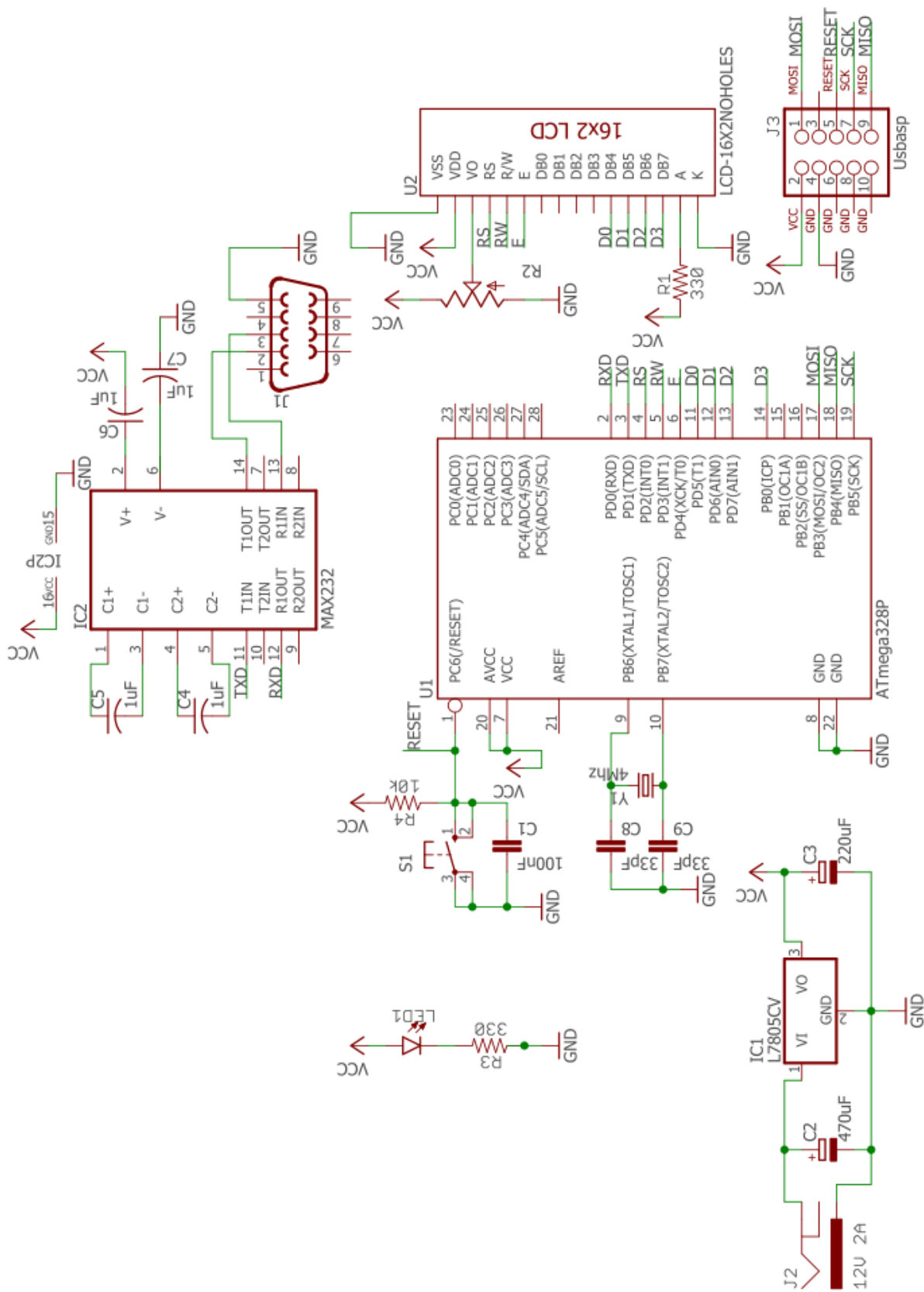


Figura 8.1: Conexión de la práctica 8.

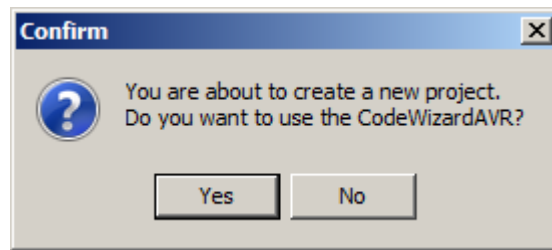


Figura 8.2: Elegir [Yes](#).

- Seleccionar en el tipo de microcontrolador a utilizar [AT90](#), [ATtiny](#), [ATmega](#)

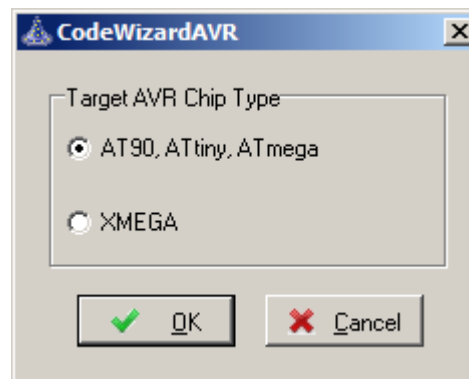


Figura 8.3: Al finalizar seleccionar [OK](#)

- En el asistente, en la ficha [Chip](#) dejar la configuración como sigue:

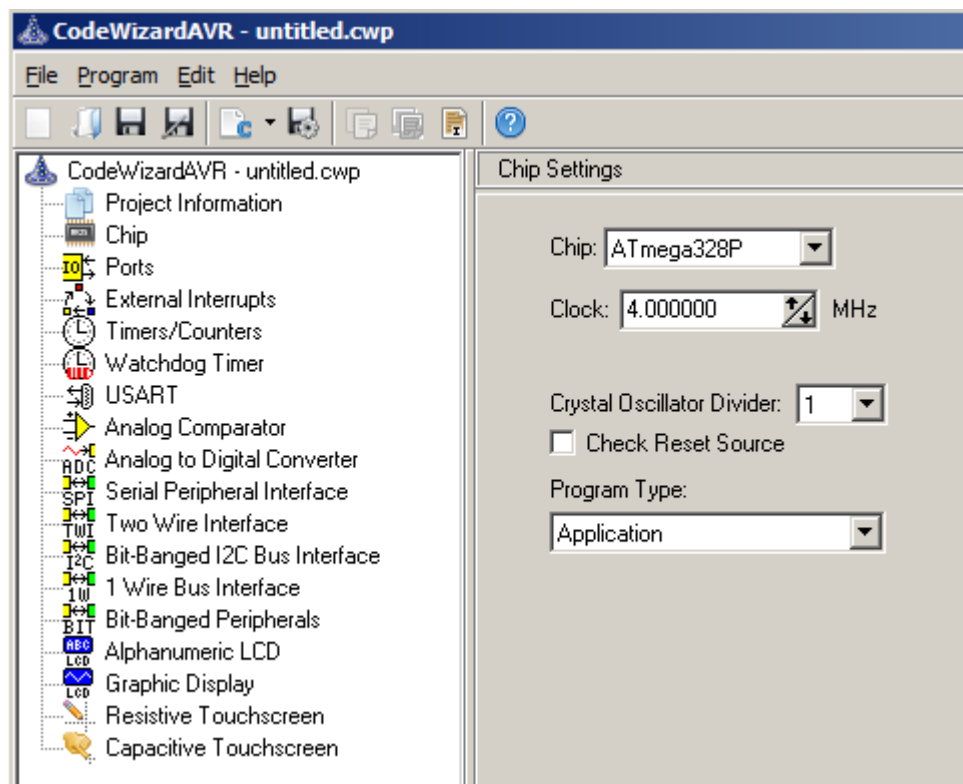


Figura 8.4: Configuración ficha **Chip**

- En el asistente, en la ficha **Ports**, en el Puerto B y el Puerto D dejar la configuración como sigue:

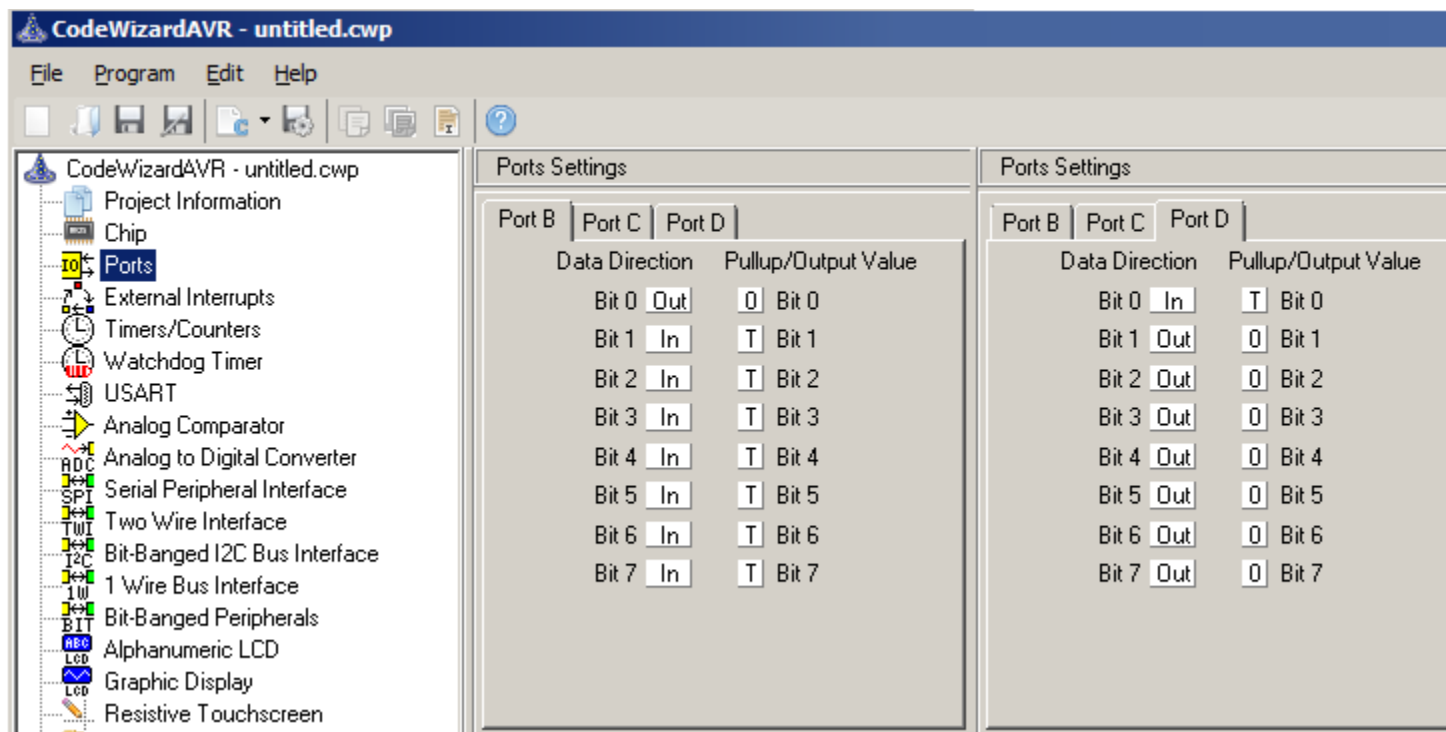


Figura 8.5: Configuración E/S Port B y Port D

- En el asistente, en la ficha **USART**, dejar la configuración como sigue:

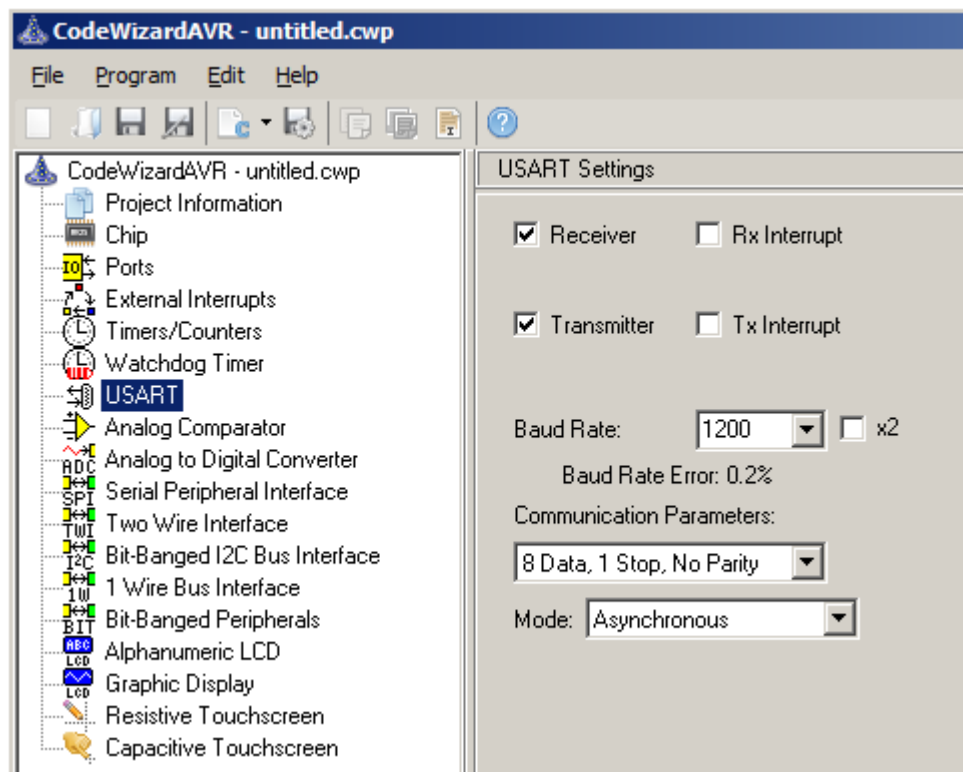


Figura 8.6: Configuración de la USART

- En el asistente, en la ficha **Alphanumeric LCD**, dejar la configuración como sigue:

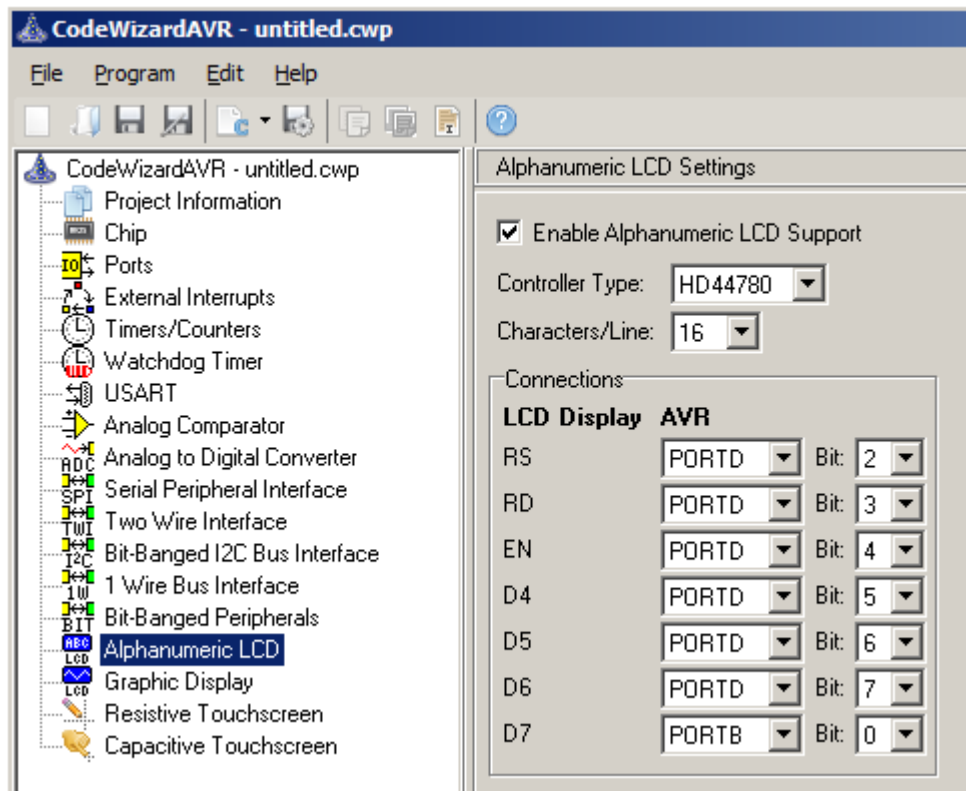


Figura 8.7: Configuración de la pantalla LCD

- Posteriormente debemos de generar el proyecto y el código, eligiendo **Program>Generate, Save and Exit**. Guarde el asistente, código y proyecto con el mismo nombre **prac08**.

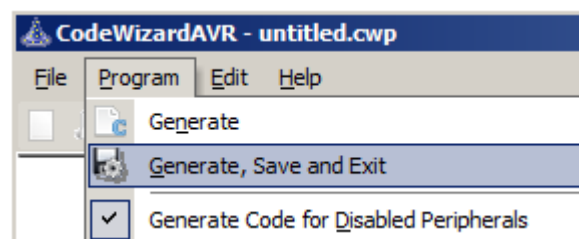


Figura 8.8: Generar el proyecto y el código.

- Mucho del código generado por el asistente no se utiliza en esta práctica. Modifique el código borrando algunas líneas para que sólo quede lo necesario para

usar la USART y la pantalla LCD. El código resultante se muestra a continuación:

```
#include <mega328p.h>

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

// Declare your global variables here

// Standard Input/Output functions
#include <stdio.h>

void main(void)
{
// Declare your local variables here
char texto[16];
char dato;
// Crystal Oscillator division factor: 1
#pragma optsize-
CLKPR=(1<<CLKPCE);
CLKPR=(0<<CLKPCE) | (0<<CLKPS3) | (0<<CLKPS2) | (0<<CLKPS1) |
    (0<<CLKPS0);
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif

// Input/Output Ports initialization
// Port B initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
//          Bit1=In Bit0=Out
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3)
    | (0<<DDB2) | (0<<DDB1) | (1<<DDB0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0
//        =0
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
    (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

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

```

// Port D initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=
//           Out Bit1=Out Bit0=In
DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3)
      | (1<<DDD2) | (1<<DDD1) | (0<<DDD0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 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
// OCOA output: Disconnected
// OCOB output: Disconnected
TCCR0A=(0<<COM0A1) | (0<<COM0A0) | (0<<COM0B1) | (0<<COM0B0) |
      (0<<WGM01) | (0<<WGM00);
TCCR0B=(0<<WGM02) | (0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0A=0x00;
OCR0B=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;
OCR1AH=0x00;
OCR1AL=0x00;

```

```

OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2A output: Disconnected
// OC2B output: Disconnected
ASSR=(0<<EXCLK) | (0<<AS2);
TCCR2A=(0<<COM2A1) | (0<<COM2A0) | (0<<COM2B1) | (0<<COM2B0) |
    (0<<WGM21) | (0<<WGM20);
TCCR2B=(0<<WGM22) | (0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2A=0x00;
OCR2B=0x00;

// Timer/Counter 0 Interrupt(s) initialization
TIMSK0=(0<<OCIE0B) | (0<<OCIE0A) | (0<<TOIE0);

// Timer/Counter 1 Interrupt(s) initialization
TIMSK1=(0<<ICIE1) | (0<<OCIE1B) | (0<<OCIE1A) | (0<<TOIE1);

// Timer/Counter 2 Interrupt(s) initialization
TIMSK2=(0<<OCIE2B) | (0<<OCIE2A) | (0<<TOIE2);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// Interrupt on any change on pins PCINT0-7: Off
// Interrupt on any change on pins PCINT8-14: Off
// Interrupt on any change on pins PCINT16-23: Off
EICRA=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
EIMSK=(0<<INT1) | (0<<INT0);
PCICR=(0<<PCIE2) | (0<<PCIE1) | (0<<PCIE0);

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 1200
UCSROA=(0<<RXCO) | (0<<TXCO) | (0<<UDRE0) | (0<<FE0) | (0<<DORO
    ) | (0<<UPE0) | (0<<U2X0) | (0<<MPCM0);
UCSROB=(0<<RXCIE0) | (0<<TXCIE0) | (0<<UDRIE0) | (1<<RXEN0) |
    (1<<TXEN0) | (0<<UCSZ02) | (0<<RXB80) | (0<<TXB80);

```

```

UCSR0C=(0<<UMSEL01) | (0<<UMSEL00) | (0<<UPM01) | (0<<UPM00) |
    (0<<USBS0) | (1<<UCSZ01) | (1<<UCSZ00) | (0<<UCPOL0);
UBRR0H=0x00;
UBRR0L=0xCF;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<AC0) | (0<<ACI) | (0<<ACIE) |
    (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
ADCSRB=(0<<ACME);
// Digital input buffer on AIN0: On
// Digital input buffer on AIN1: On
DIDR1=(0<<AINOD) | (0<<AIN1D);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<
    ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
    (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWST0) | (0<<TWEN) | (0<<TWIE
    );

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu
:
// RS - PORTD Bit 2
// RD - PORTD Bit 3
// EN - PORTD Bit 4
// D4 - PORTD Bit 5
// D5 - PORTD Bit 6
// D6 - PORTD Bit 7
// D7 - PORTB Bit 0
// Characters/line: 16
lcd_init(16);

```

```
while (1)
{
    dato=getchar();
    putchar(dato);
    sprintf(texto,"%c", dato);
    lcd_gotoxy(0,0);
    lcd_puts(texto);
}
```

- Compile el proyecto eligiendo desde el menú **Project>Build All**. Una compilación correcta arrojará la información *No errors, No warnings*
- Descargue el archivo que resulto de la compilación **prac08.hex** en la memoria Flash del microcontrolador.
- **¡Este es un paso importante!** Usted debe de habilitar el uso de un oscilador externo en el microcontrolador. Escriba en **Low Fuse** el número **EE** tal y como se muestra en la imagen.

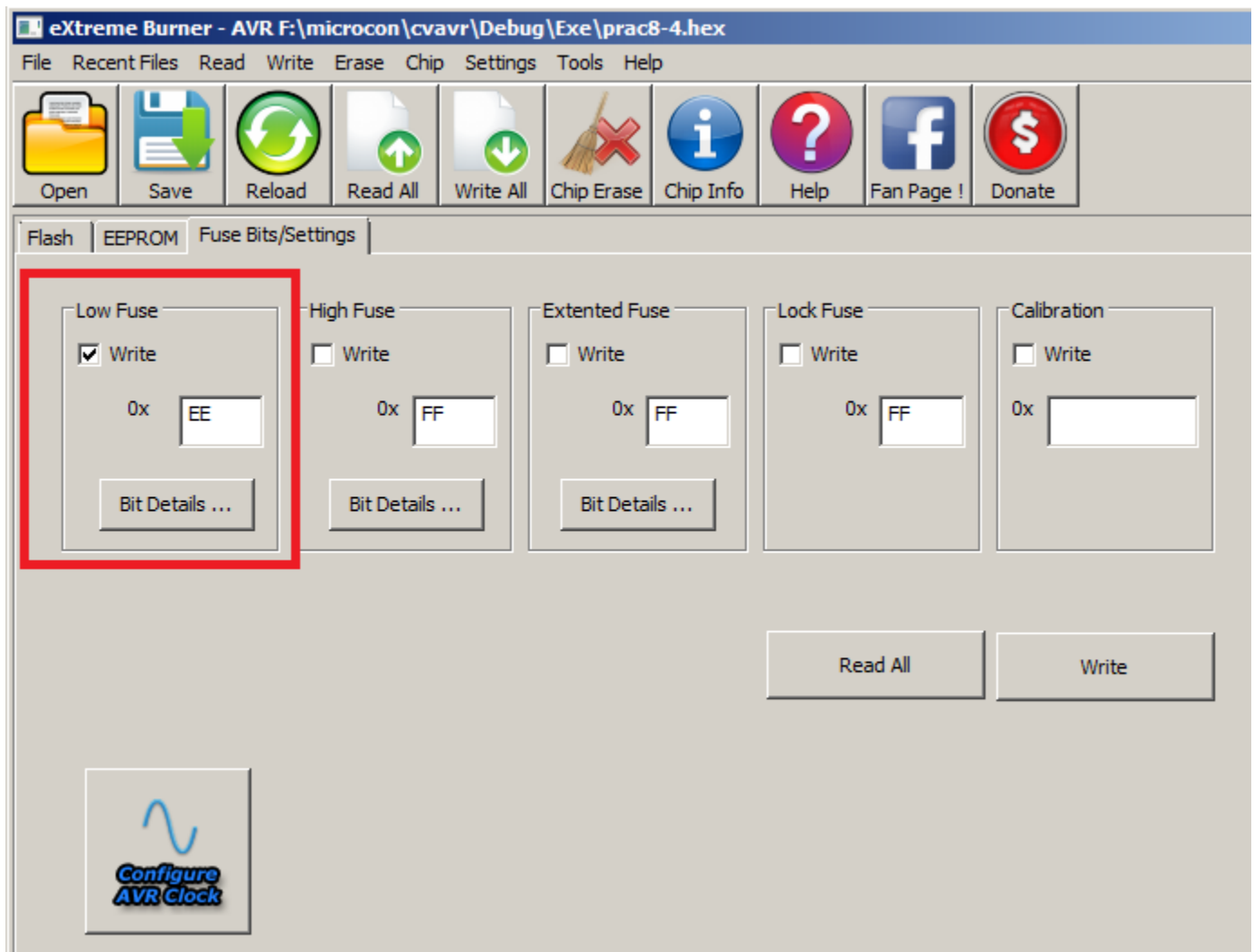


Figura 8.9: Habilitar oscilador externo.

- Conecte su convertidor USB a RS232 a cualquier puerto USB. Configure su emulación de puerto serial según los pasos de su fabricante.
- Configure la terminal integrada de Codevision AVR en [Settings>Terminal](#) como donde [COMX](#) es el puerto de comunicación serial que esta ocupando:

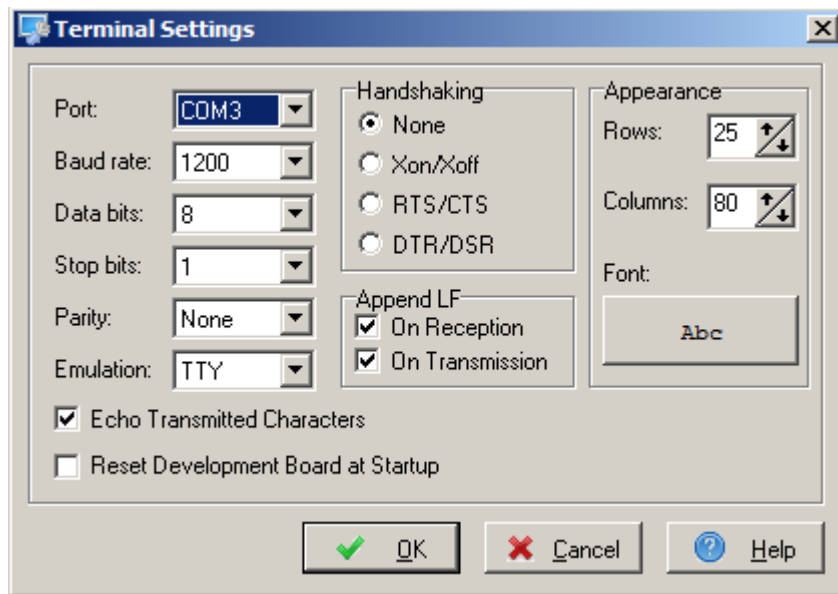


Figura 8.10: Configuración de la Terminal.

- Conecte la Terminal en **Tools>Terminal**, como se muestra en la siguiente figura:

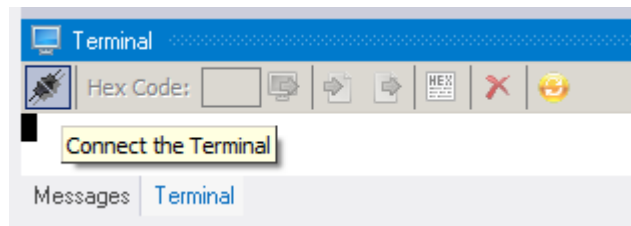


Figura 8.11: Configuración de la herramienta Terminal.

- Envíe datos en código ASCII tecleando letra a letra en la Terminal.
- Muestre el circuito funcionando al profesor, para que le sea tomado en cuenta. En la pantalla LCD se debe de mostrar las letras que el usuario esta enviando a la Terminal, así mismo en la Terminal se debe de ver la letra enviada, puesto que el microcontrolador regresa los datos que llegan. De la misma manera muestre la simulación.