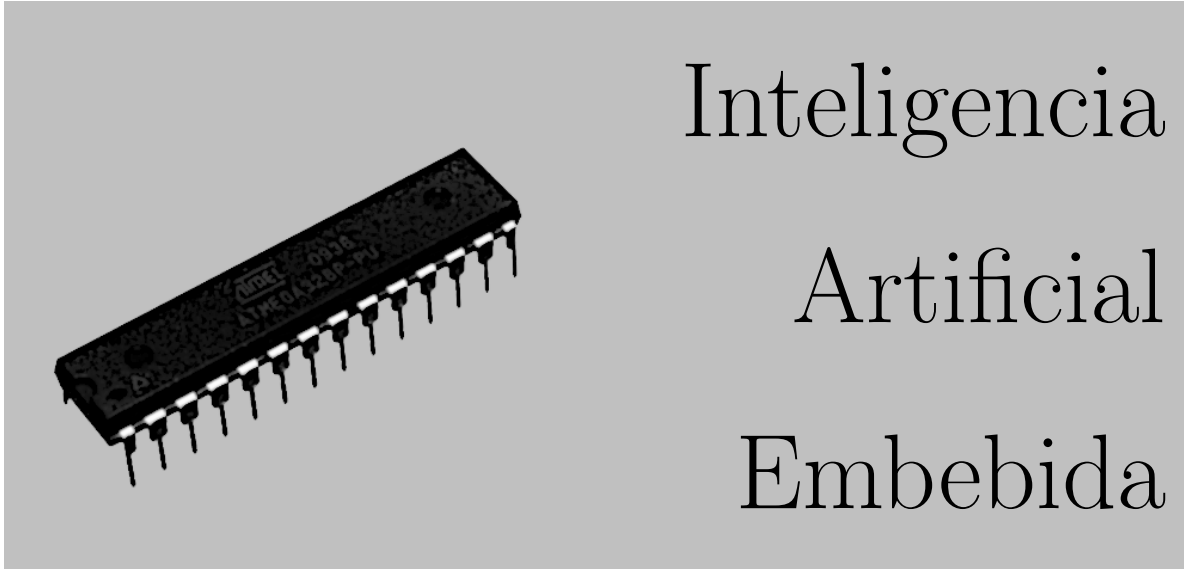




Inteligencia Artificial Embebida

Nombre: _____ Grupo: _____

Dr. Enrique García Trinidad
Tecnológico de Estudios Superiores de Huixquilucan
<https://enriquegarcia.xyz>
enrique.g.t@huixquilucan.tecnm.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 ??:

8.3. 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](#).

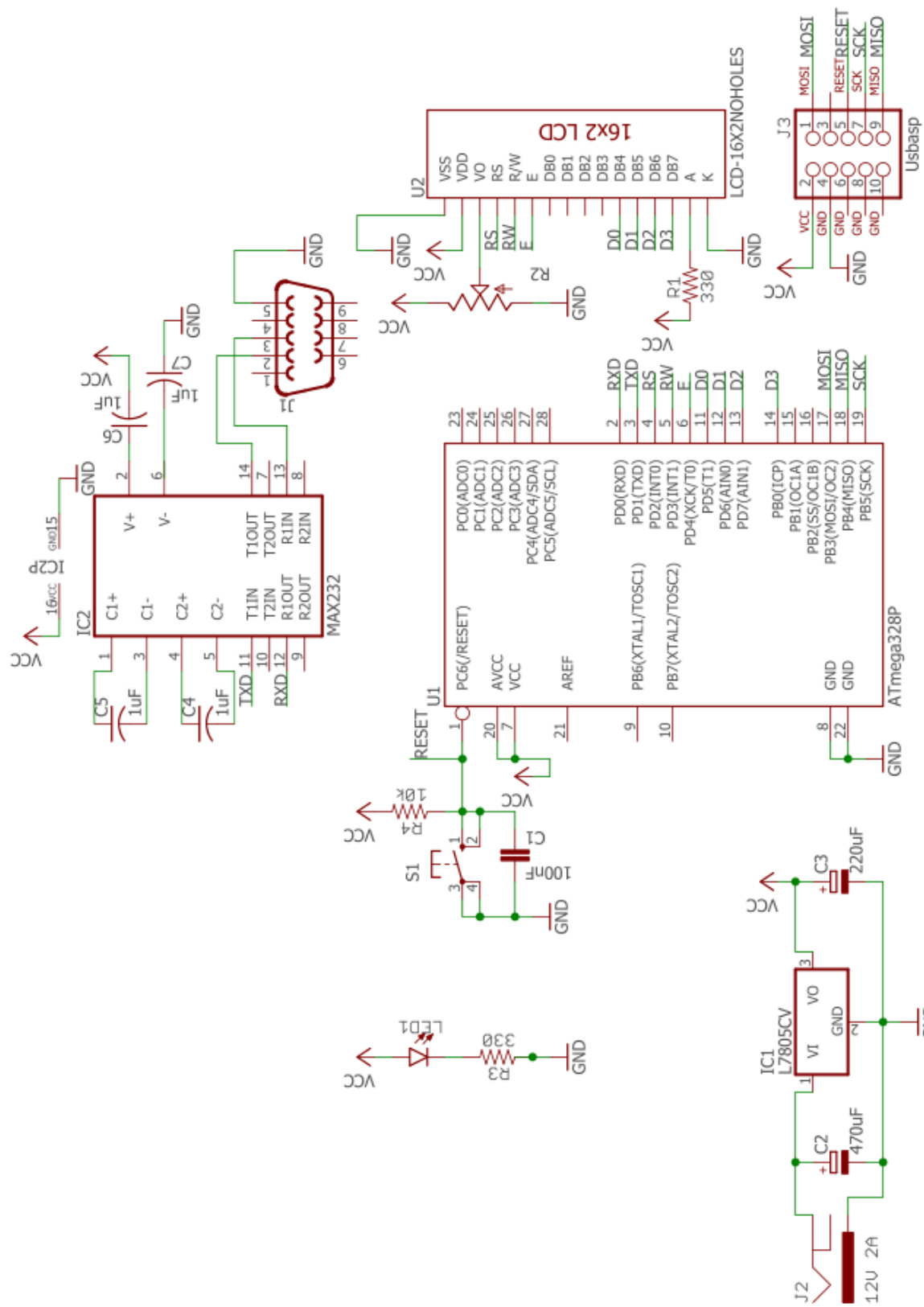


Figura 8.1: Conexión de la práctica 8.

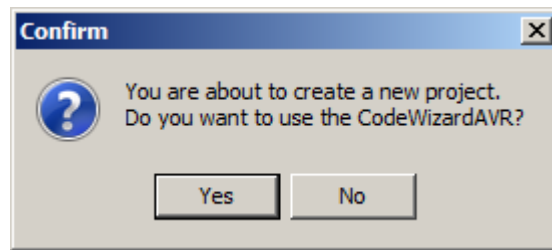


Figura 8.2: Elegir [Yes](#).

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

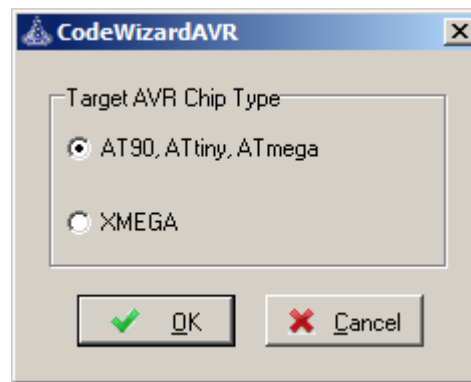


Figura 8.3: Al finalizar seleccionar [OK](#)

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

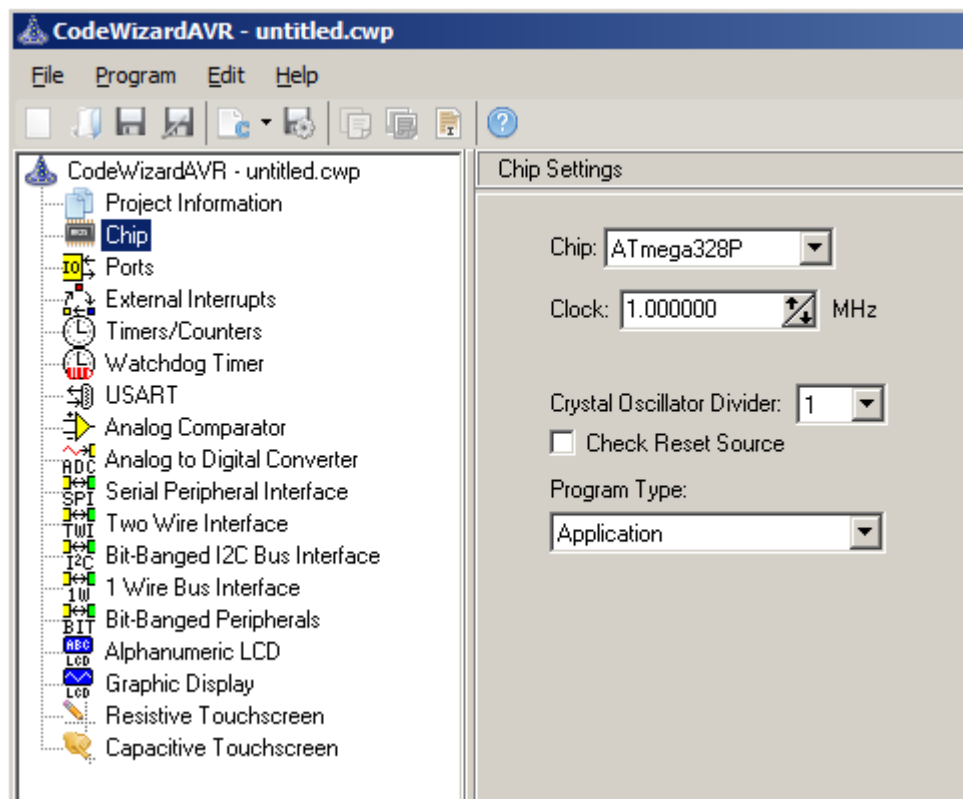


Figura 8.4: Configuración ficha **Chip**

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

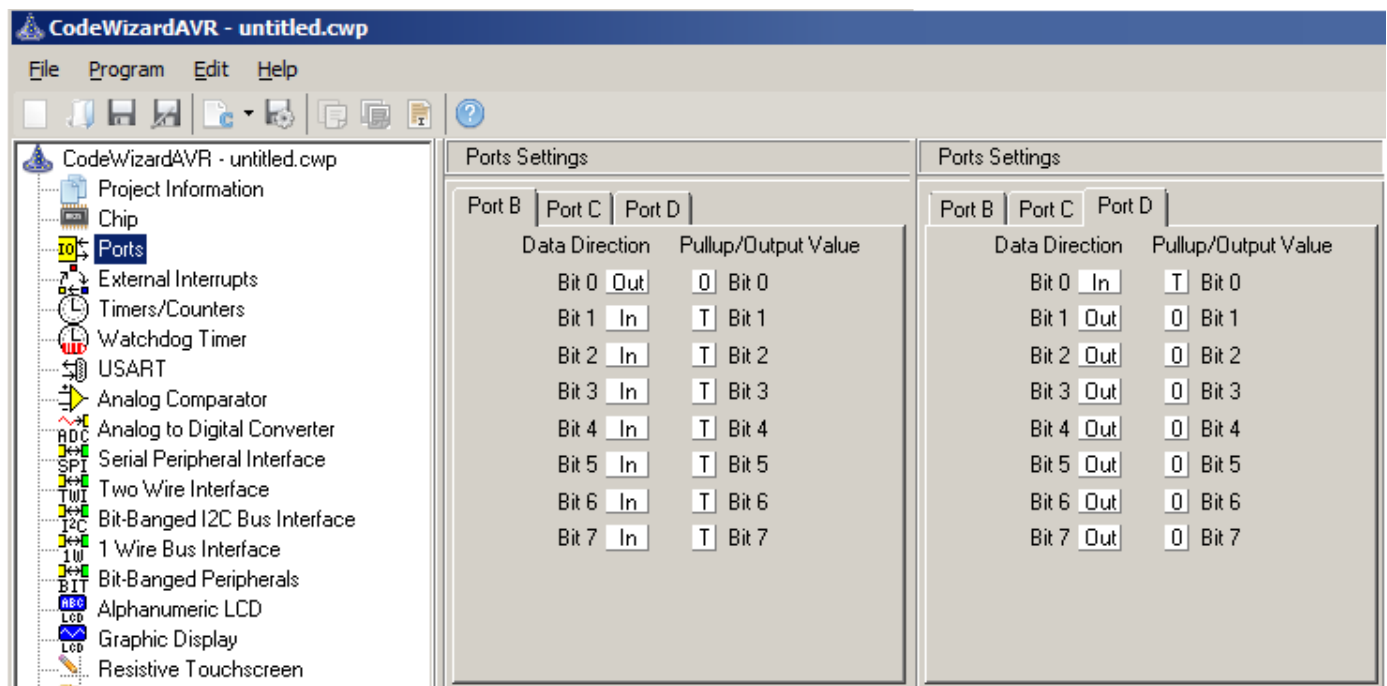


Figura 8.5: Configuración E/S **Port B** y **Port D**

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

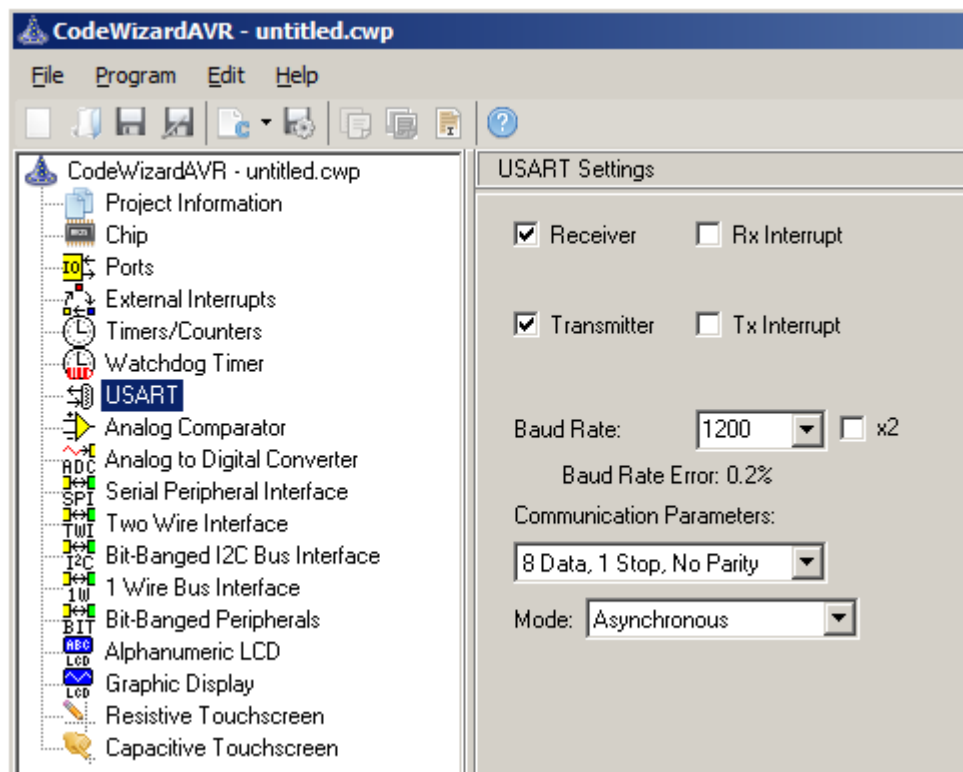


Figura 8.6: Configuración de la USART

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

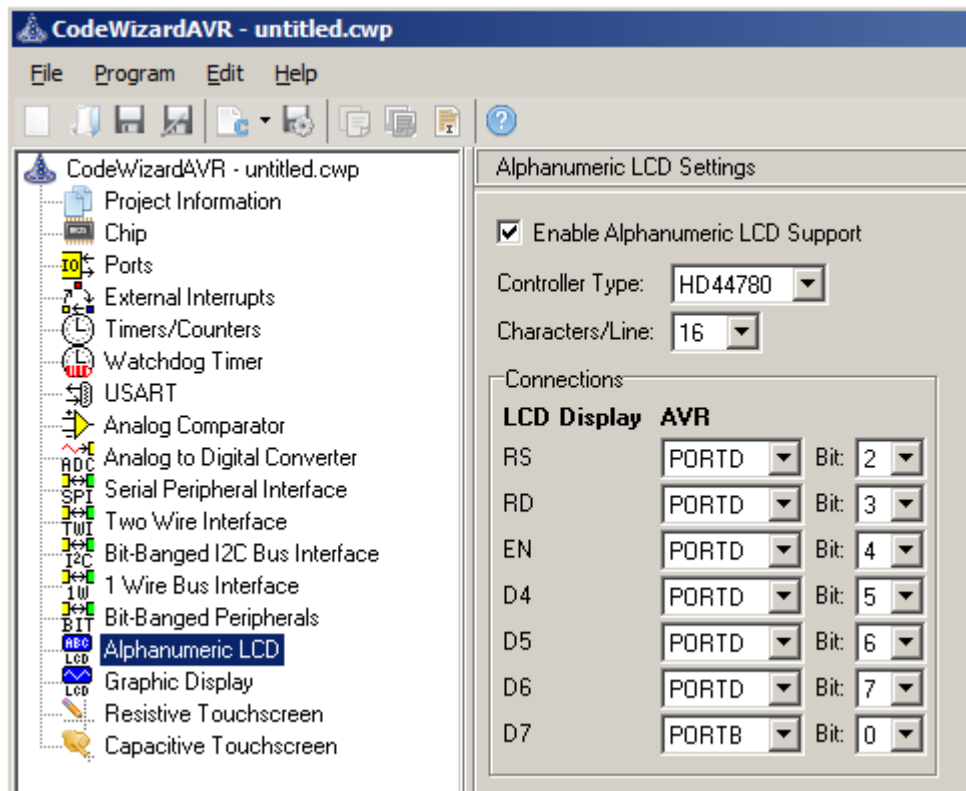


Figura 8.7: Configuración de la pantalla LCD

- 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**.

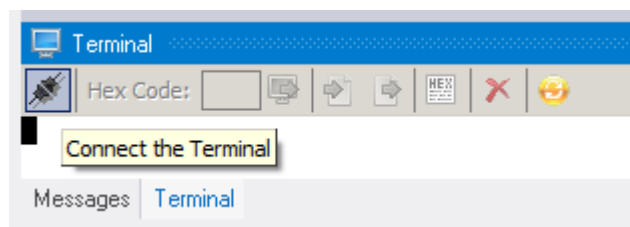


Figura 8.8: Generar el proyecto y el código.

- 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>
#include <alcd.h>
#include <stdio.h>

// Declare your global variables here
#define xtal 1000000L
#define baud 1200

void main(void)
{
    // Declare your local variables here
    char texto[16];
    char dato;

    DDRB=0x01;
    PORTB=0x00;
    DDRC=0x00;
    PORTC=0x00;
    DDRD=0xFE;
    PORTD=0x00;

    #pragma optsize-
    CLKPR=(1<<CLKPCE);
    CLKPR=(0<<CLKPCE) | (0<<CLKPS3) | (0<<CLKPS2) | (0<<CLKPS1) |
        (0<<CLKPS0);
    #ifdef _OPTIMIZE_SIZE_
    #pragma optsize+
    #endif

    TCCR0A=0x00;
    TCCR0B=0x00;
    TCNT0=0x00;
    OCR0A=0x00;
    OCR0B=0x00;

    TCCR1A=0x00;
    TCCR1B=0x00;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x00;
    ICR1L=0x00;
    OCR1AH=0x00;
```

```

OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

ASSR=0x00;
TCCR2A=0x00;
TCCR2B=0x00;
TCNT2=0x00;
OCR2A=0x00;
OCR2B=0x00;

TIMSK0=0x00;
TIMSK1=0x00;
TIMSK2=0x00;

EICRA=0x00;
EIMSK=0x00;
PCICR=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 1200
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x06;
UBRR0H=(xtal/16/1200)>>8;
UBRR0L=(xtal/16/1200)>>0xFF;

ACSR=0x80;
ADCSRB=0x00;
DIDR1=0x00;
ADCSRA=0x00;
SPCR=0x00;

TWCR=0x00;

lcd_init(16);

while (1)
{
    dato=getchar();
    putchar(dato);
    sprintf(texto, "Dato=%d", 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.

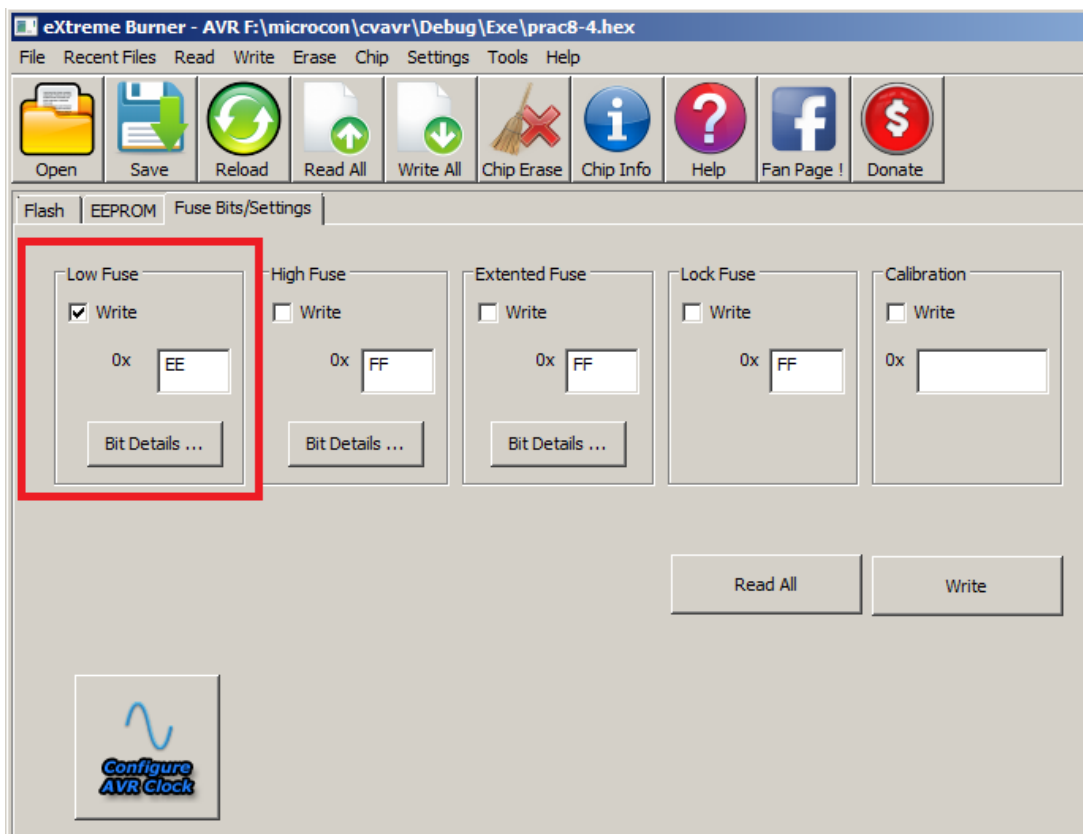


Figura 8.9: Habilitar oscilador externo.

- 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:

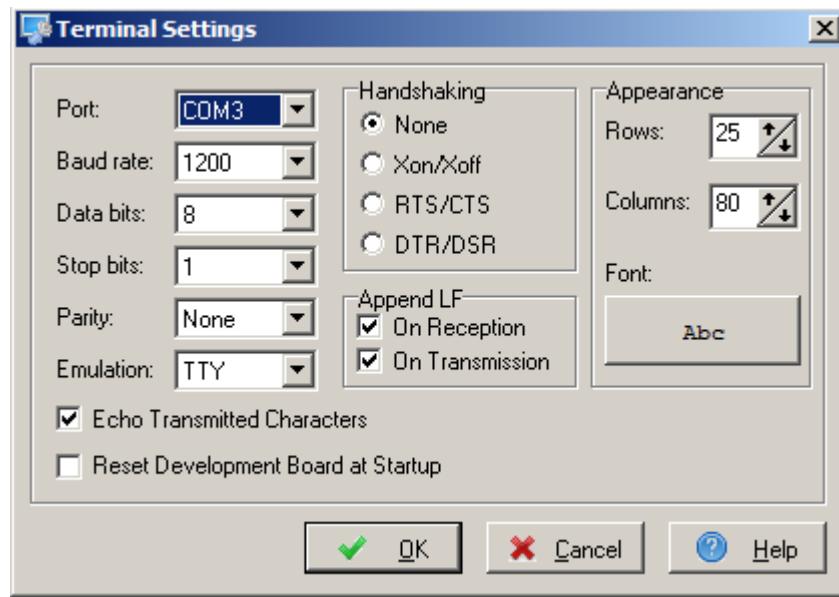


Figura 8.10: Configuración de la Terminal.

- Conecte la Terminal el [Tools>Terminal](#), como se muestra en la siguiente figura:

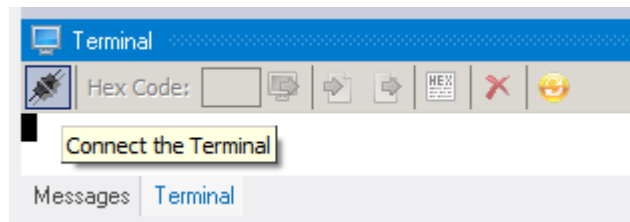


Figura 8.11: Configuración de la herramienta Terminal.

- 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.