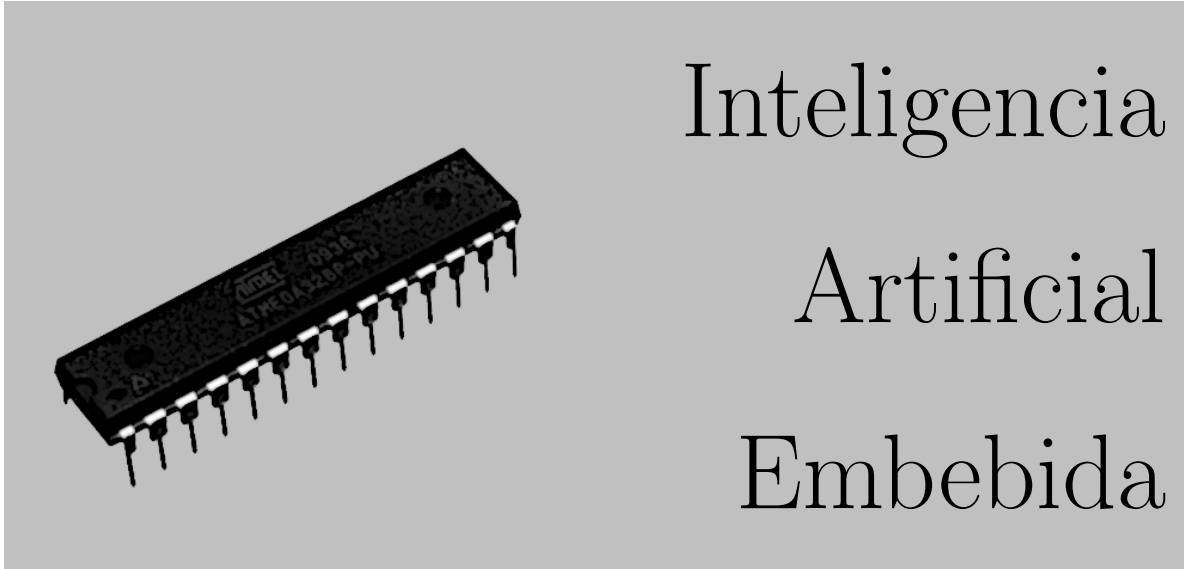




Inteligencia Artificial Embebida

Nombre: _____ Grupo: _____

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

Práctica 1

Salidas digitales

1.1. Material

El material enlistado es necesario para la realización de la práctica 1.

Ct	Dispositivo	Descripción	Eti.
1	ATmega328P-PU	Microcontrolador AVR RISC 8-bit 20Mhz	U1
1	Regulador L7805CV	Regulador de voltaje 5V 1 A	IC1
1	Capacitor cerámico de 0.1 μ F 50V	Código: 104	C1
1	Capacitor electrolítico de 470 μ F 25V	Tolerancia $\pm$ 20 %	C2
1	Capacitor electrolítico de 220 μ F 25V	Tolerancia $\pm$ 20 %	C3
9	Resistencia de 330 Ω 1/4W	Código: Naranja, naranja, café, oro	R1 ... R9
1	Resistencia de 10k Ω 1/4W	Código: Café, negro, naranja, oro	R10
1	Push button (Microswitch)	Tipo push, 4 o 2 terminales	S1
9	Led 5mm difuso	Color rojo	
2	Metro de alambre para protoboard		
1	Protoboard		
1	Grabador Usbasp	Grabador microcontroladores AVR 8-bit	J1
1	Fuente de alimentación de 12V 2A	Eliminador de voltaje	J2
1	Computadora con puerto USB		

1.2. Conexión de los componentes

- Conecte el siguiente diagrama:

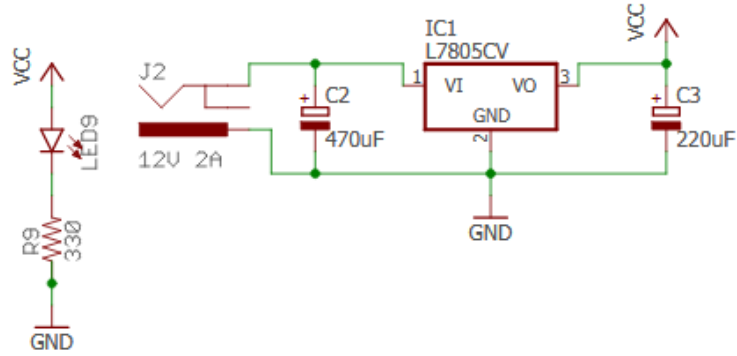


Figura 1.1: Conexión de la fuente de alimentación.

- Conecte su fuente de alimentación. Con un multímetro cheque que realmente **VCC** tenga el valor de **5.0V**. Si no es así, revise sus conexiones.
- Desconecte la fuente de alimentación. Complete el circuito anterior de acuerdo al siguiente diagrama, a este circuito lo llamamos el *Circuito base*:

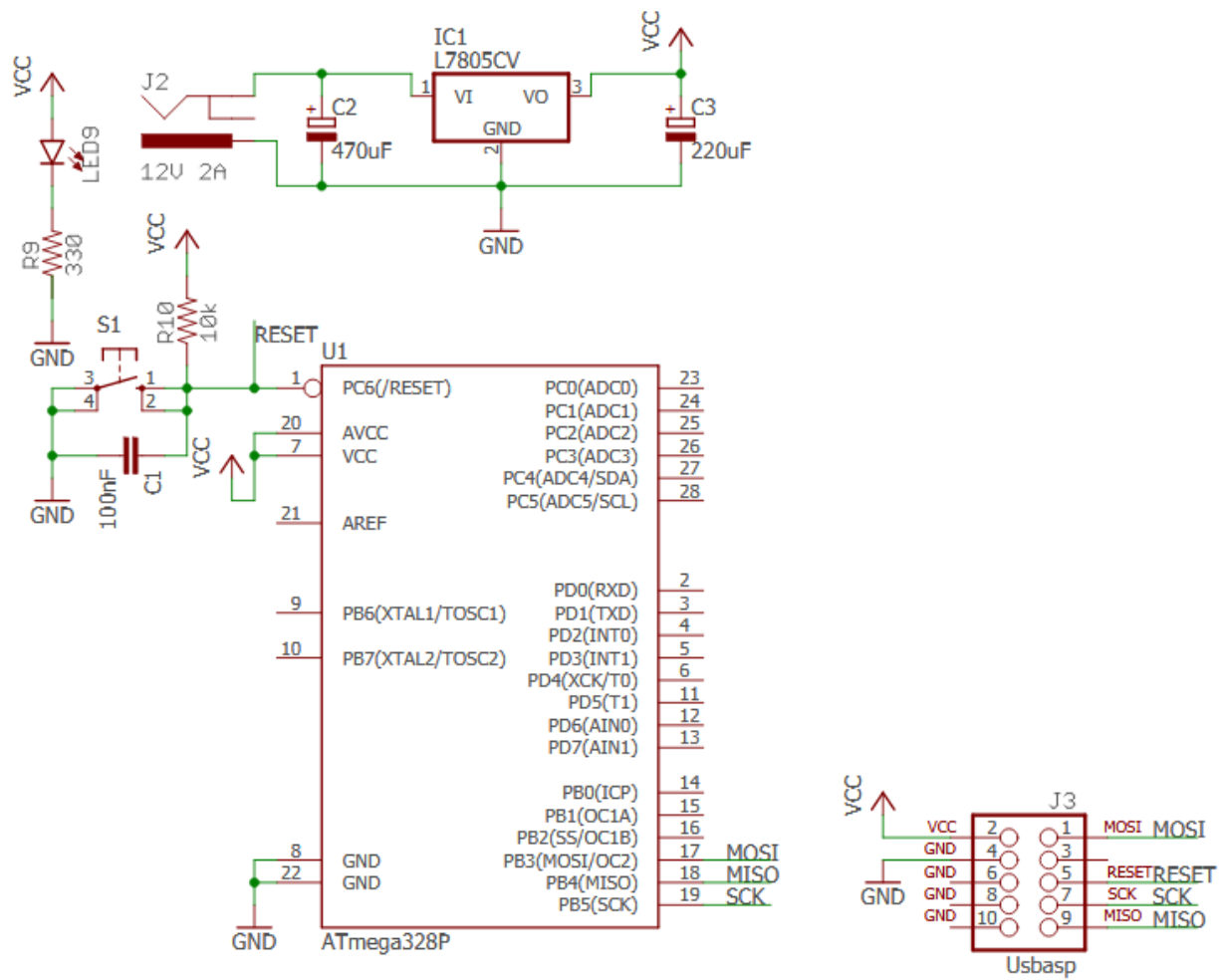


Figura 1.2: Conexión del circuito base .

- Los pines del conector de 10-pin de la tarjeta Usbasp tienen las siguientes señales, tenga cuidado al conectar, guíese por la muesca del conector.

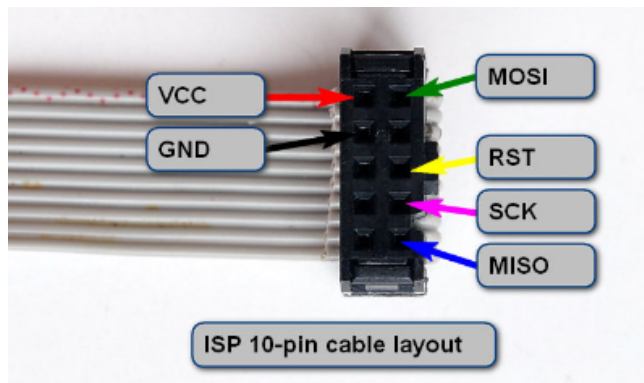


Figura 1.3: Descripción de los pines de la tarjeta Usbasp.

- Revise de nuevo sus conexiones. Conecte de nuevo la fuente de alimentación e inicie Extreme Burner AVR. En el menú seleccione [Chip>ATmega328P](#). Después elija [Erase>Chip Erase](#). Extreme Burner AVR borrará la memoria del microcontrolador:

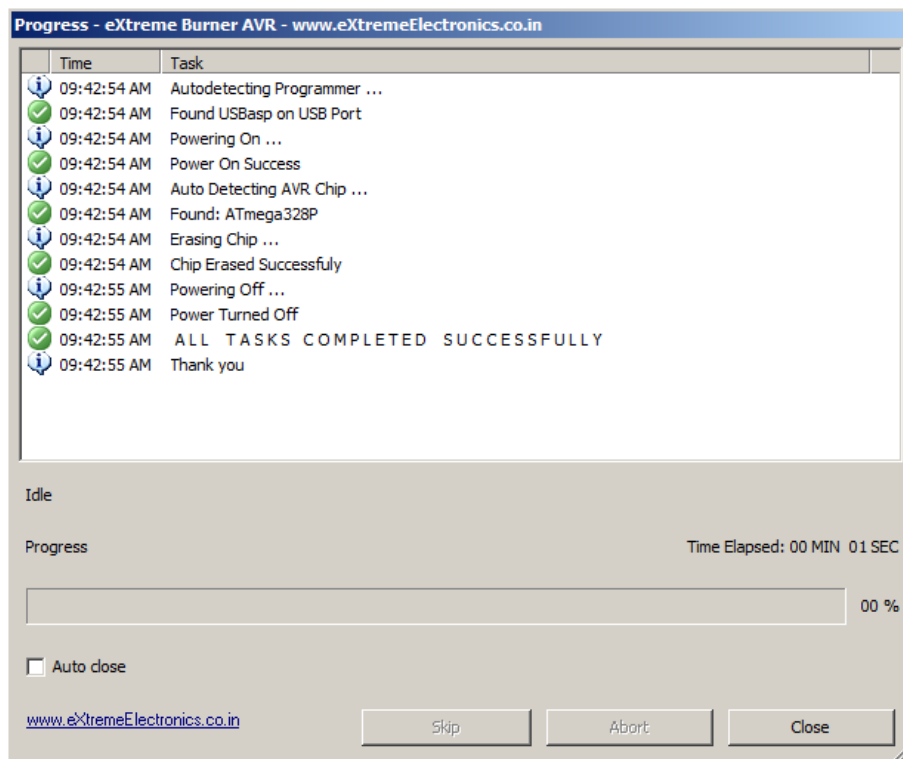


Figura 1.4: Borrado exitoso de la memoria del microcontrolador

1.3. Preparación del proyecto

Ejecute los siguientes pasos:

- Conecte el siguiente diagrama:

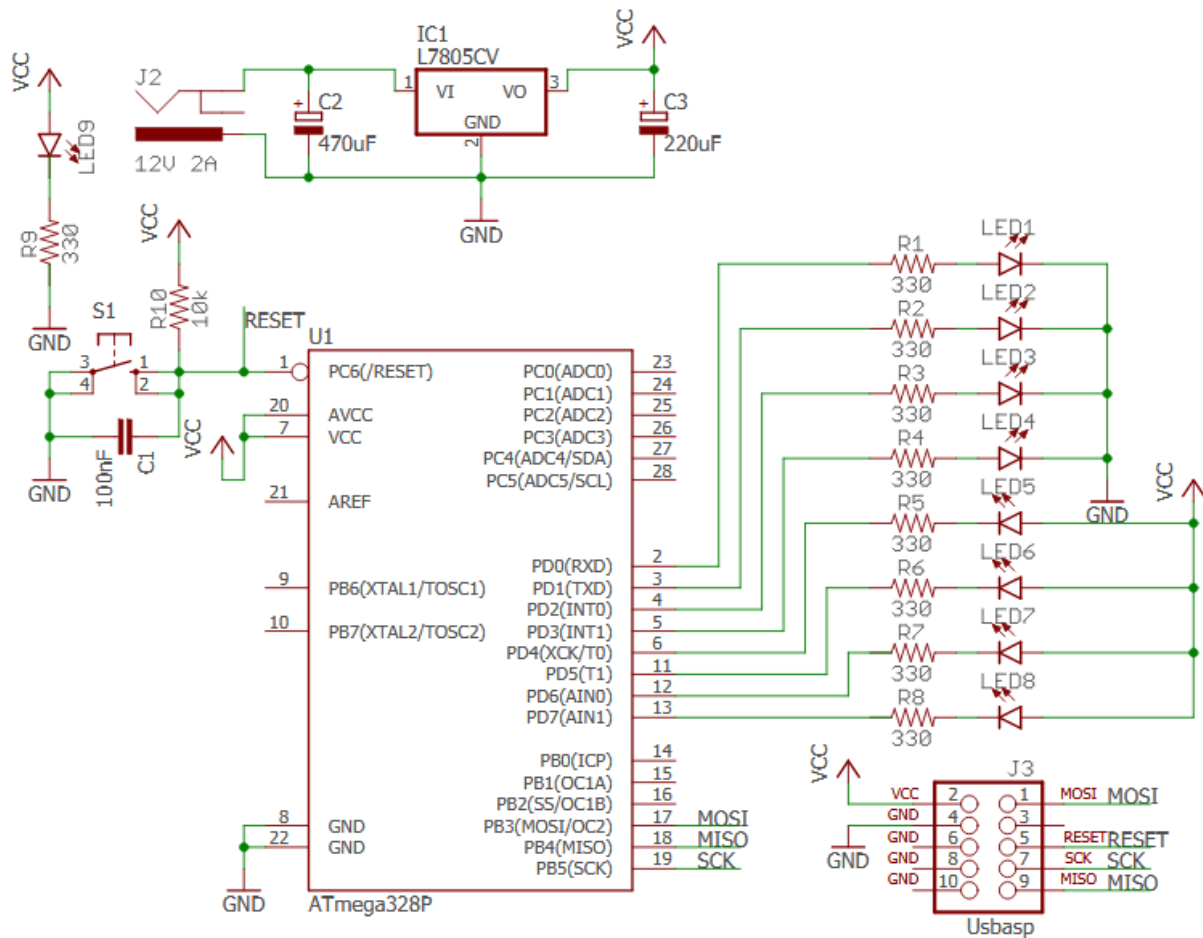


Figura 1.5: Conexión de la fuente de alimentación.

- Inicie el software Codevision AVR. Cree un nuevo proyecto dando click en el menú **New>Project**.

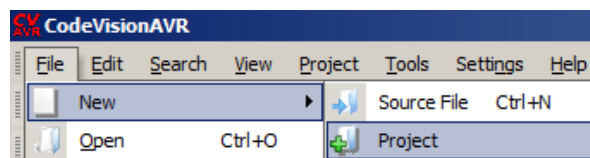


Figura 1.6: Creación de un nuevo proyecto.

- Cuando el software pregunte si queremos usar el asistente [CodeWizardAVR](#) le indicamos que [No](#).

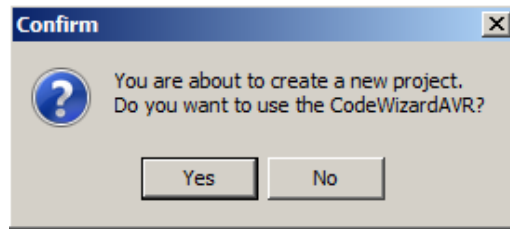


Figura 1.7: Seleccionemos [No](#) en esta ventana.

- Posteriormente debemos dar el nombre [prac01](#) al proyecto.

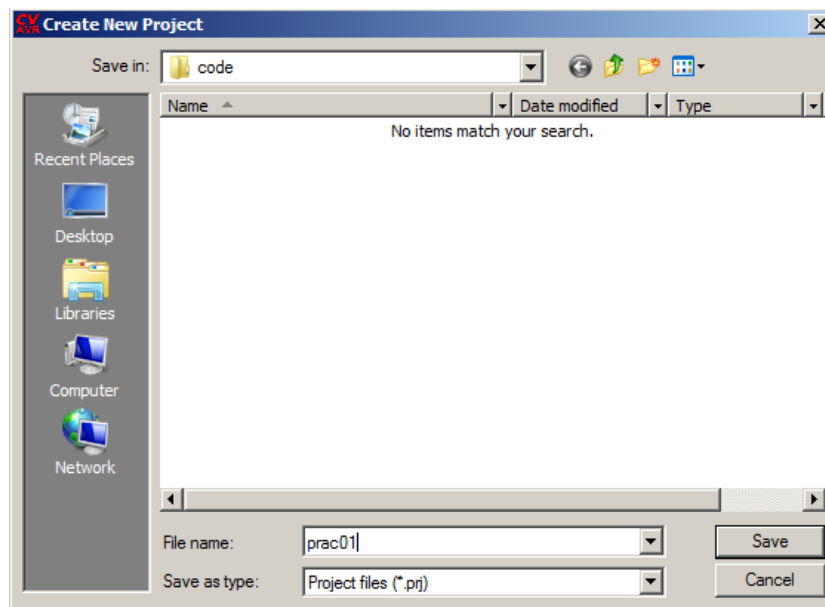


Figura 1.8: Estableciendo el nombre del proyecto.

- Posteriormente seleccionemos el modelo del microcontrolador de la lista: [ATmega328P](#). Al final presionamos [OK](#).

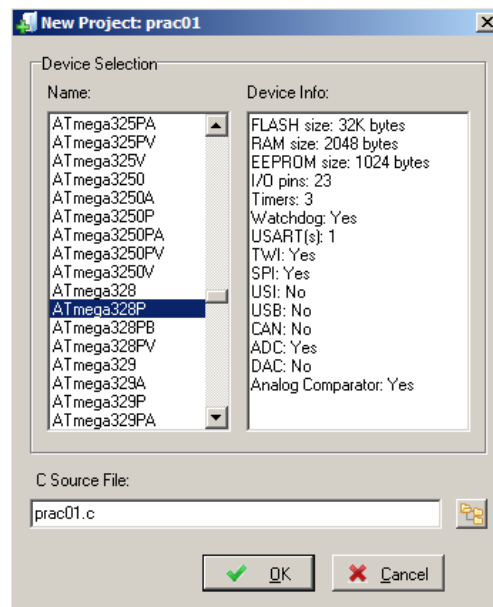


Figura 1.9: Estableciendo el modelo del microcontrolador.

- En la siguiente ventana que se muestra, configuremos la velocidad del oscilador en la ficha **C Compiler** y en **Clock**: establecemos 1.000 Mhz. Al final presionamos OK.

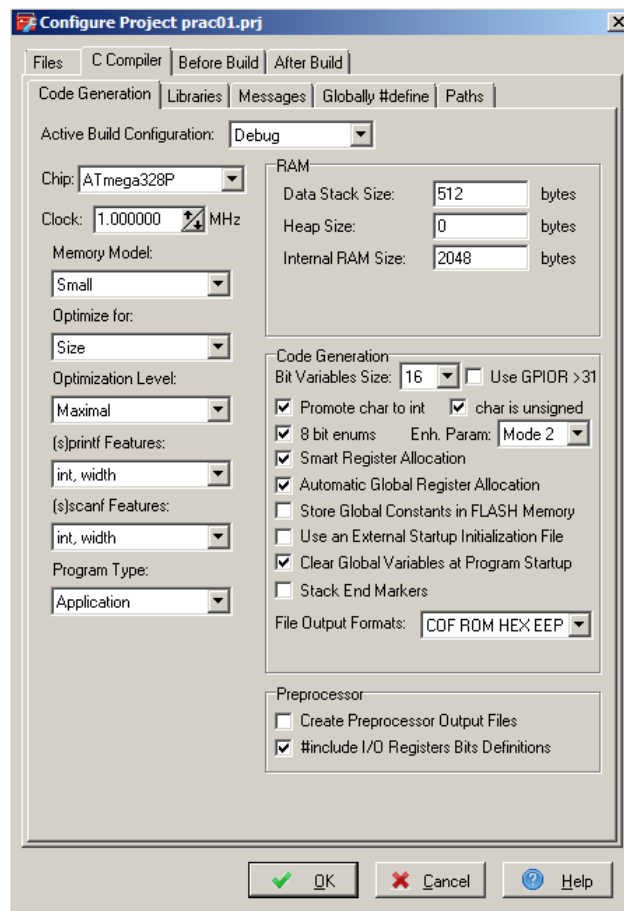


Figura 1.10: Estableciendo la velocidad del oscilador.

- Modifique el código como se muestra a continuación:

```
//Prctica 1, ejercicio 1
#include <mega328p.h>

void main(void)
{
    //Todo el puerto D como salida
    //Bit 0 al 4 sacan GND
    //Bit 5 al 8 sacan VCC
    DDRD=0xFF;
    PORTD=0xF0;

    while(1)
```

```
{  
    //Seales individuales  
    PORTD.0=1;  
    PORTD.1=1;  
    PORTD.2=1;  
    PORTD.3=1;  
    PORTD.4=0;  
    PORTD.5=0;  
    PORTD.6=0;  
    PORTD.7=0;  
}
```

- Compile el proyecto eligiendo desde el menú **Project>Build All**. Una compilación correcta arrojará la información *No errors, No warnings*

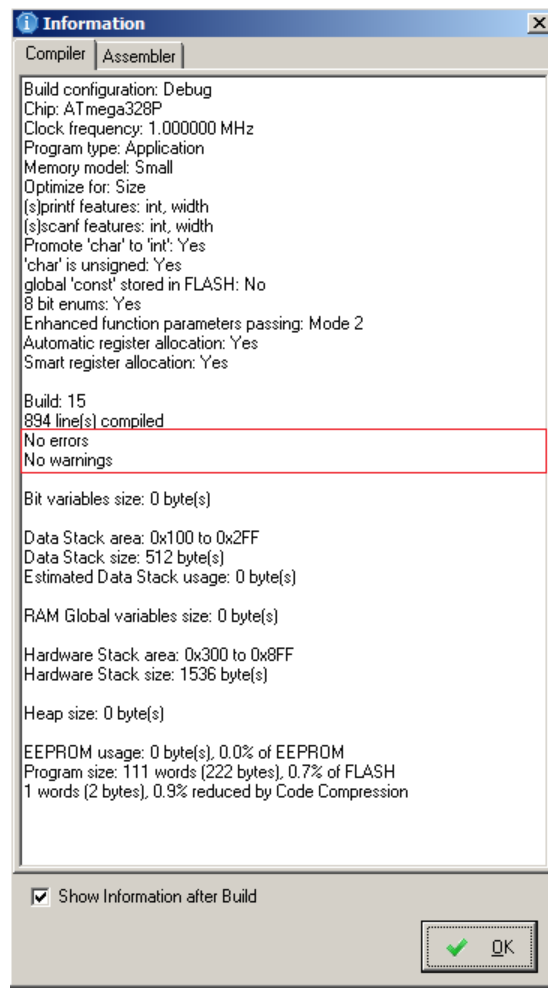


Figura 1.11: Compilación correcta

- De nuevo ejecute el software Extreme Burner AVR, en el menú seleccione **File>Open Flash** y busque la carpeta **Debug/Exe** en el directorio donde guardó su proyecto y seleccione el archivo **prac01.hex**. Extreme Burner AVR cargará el ejecutable producto de la compilación.

	00 - 01	02 - 03	04 - 05	06 - 07	08 - 09	0A - 0B	0C - 0D	0E - 0F
000000	940C	0034	940C	0000	940C	0000	940C	0000
000010	940C	0000	940C	0000	940C	0000	940C	0000
000020	940C	0000	940C	0000	940C	0000	940C	0000
000030	940C	0000	940C	0000	940C	0000	940C	0000
000040	940C	0000	940C	0000	940C	0000	940C	0000
000050	940C	0000	940C	0000	940C	0000	940C	0000
000060	940C	0000	940C	0000	94F8	27EE	BBEF	E0F1
000070	BFF5	BFE5	E08D	E0A2	27BB	93ED	958A	F7E9
000080	E080	E098	E0A0	E0B1	93ED	9701	F7E9	E0E0
000090	BBEE	EFEF	BFED	E0E8	BFEE	E0C0	E0D3	940C
0000A0	0051	EFEF	B9EA	EFE0	B9EB	9A58	9A59	9A5A
0000B0	9A5B	985C	985D	985E	985F	CFF7	CFFF	FFFF
0000C0	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
0000D0	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
0000E0	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
0000F0	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
000100	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
000110	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
000120	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF

Figura 1.12: Compilación correcta

- Grabe el microcontrolador mediante la opción **Write>Flash**
- Muestre el circuito funcionando al profesor, para que le sea tomado en cuenta.

1.4. Ejercicio 2.

- Modifique el código como se muestra a continuación:

```
//Prctica 1, ejercicio 2
#include <mega328p.h>

void main(void)
{
```

```

//Todo el puerto D como salida
//Bit 0 al 4 sacan GND
//Bit 5 al 8 sacan VCC
DDRD=0xFF;
PORTD=0xF0;

while(1)
{
    //Sacar datos por el puerto D
    //al mismo tiempo
    PORTD=0x0F;
}
}

```

- Compile, grabe y muestre el resultado al profesor para su evaluación.

1.5. Ejercicio 3.

- Modifique el código como se muestra a continuación:

```

//Prctica 1, ejercicio 3
#include <mega328p.h>

void main(void)
{
    //Todo el puerto D como salida
    //Bit 0 al 4 sacan GND
    //Bit 5 al 8 sacan VCC
    DDRD=0xFF;
    PORTD=0xF0;

    while(1)
    {
        //Led1 = prendido
        //Led2 = apagado
        //Led3 = prendido
        //Led4 = apagado
        //Led5 = prendido
        //Led6 = apagado
        //Led7 = prendido
        //Led8 = apagado
        PORTD=0xA5;
    }
}

```

- Compile, grabe y muestre el resultado al profesor para su evaluación.

1.6. Ejercicio 4.

- Modifique el código como se muestra a continuación:

```
//Prctica 1, ejercicio 4
#include <mega328p.h>
#include <delay.h>

void main(void)
{
    //Todo el puerto D como salida
    //Bit 0 al 4 sacan GND
    //Bit 5 al 8 sacan VCC
    DDRD=0xFF;
    PORTD=0xF0;

    while(1)
    {
        //Alternando las seales a los Leds
        //con retardo de 200ms
        PORTD=0xA5;
        delay_ms(200);
        PORTD=0x5A;
        delay_ms(200);
    }
}
```

- Compile, grabe y muestre el resultado al profesor para su evaluación.