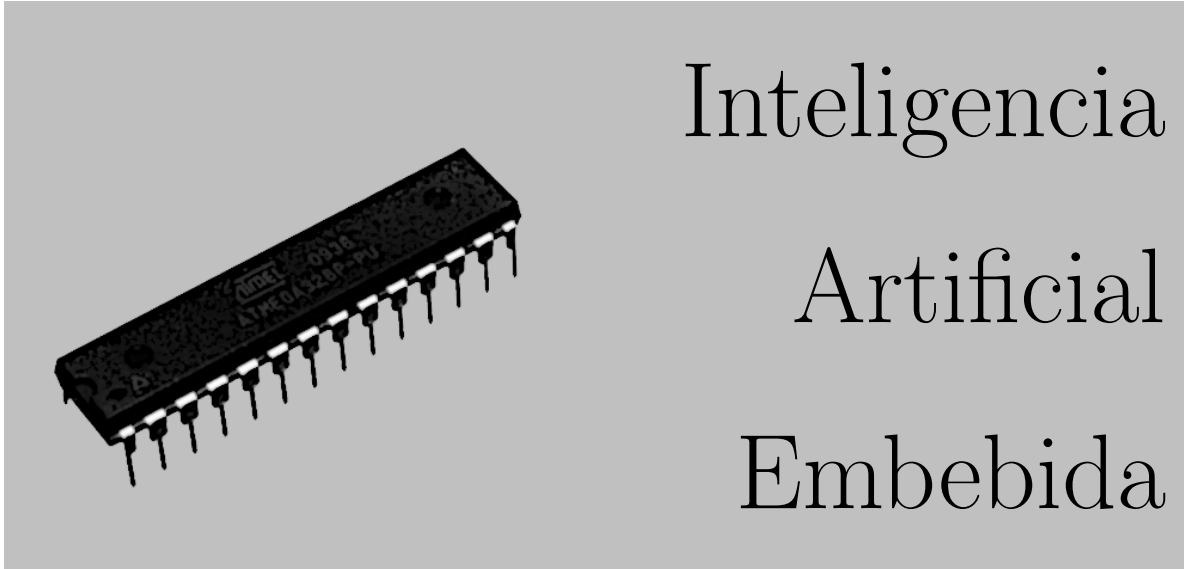




Inteligencia Artificial Embebida

Nombre: _____ Grupo: _____

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

Práctica 3

Arreglos

3.1. Material

El material enlistado es necesario para la realización de la práctica 3.

Ct	Dispositivo	Descripción	Eti.
1	ATmega328P-PU	Microcontrolador AVR RISC 8-bit 20Mhz	U1
1	Regulador L7805CV	Regulador de voltaje 5V 1 A	IC1
2	Capacitor cerámico de 0.1 μ F 50V	Código: 104	C1, C4
1	Capacitor electrolítico de 470 μ F 25V	Tolerancia $\pm$ 20 %	C2
1	Capacitor electrolítico de 220 μ F 25V	Tolerancia $\pm$ 20 %	C3
3	Resistencia de 330 Ω 1/4W	Código: Naranja, naranja, café, oro	R1, R2, R9
2	Resistencia de 10k Ω 1/4W	Código: Café, negro, naranja, oro	R10, R3
1	Led 5mm difuso	Color rojo	LED9
2	Push button (Microswitch)	Tipo push, 4 o 2 terminales	S1, S2
1	Display de 7 segmentos	Ánodo común	D1
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 puerto USB		

3.2. Conexión de los componentes

- Conecte el circuito base de acuerdo a la Figura 2.1:

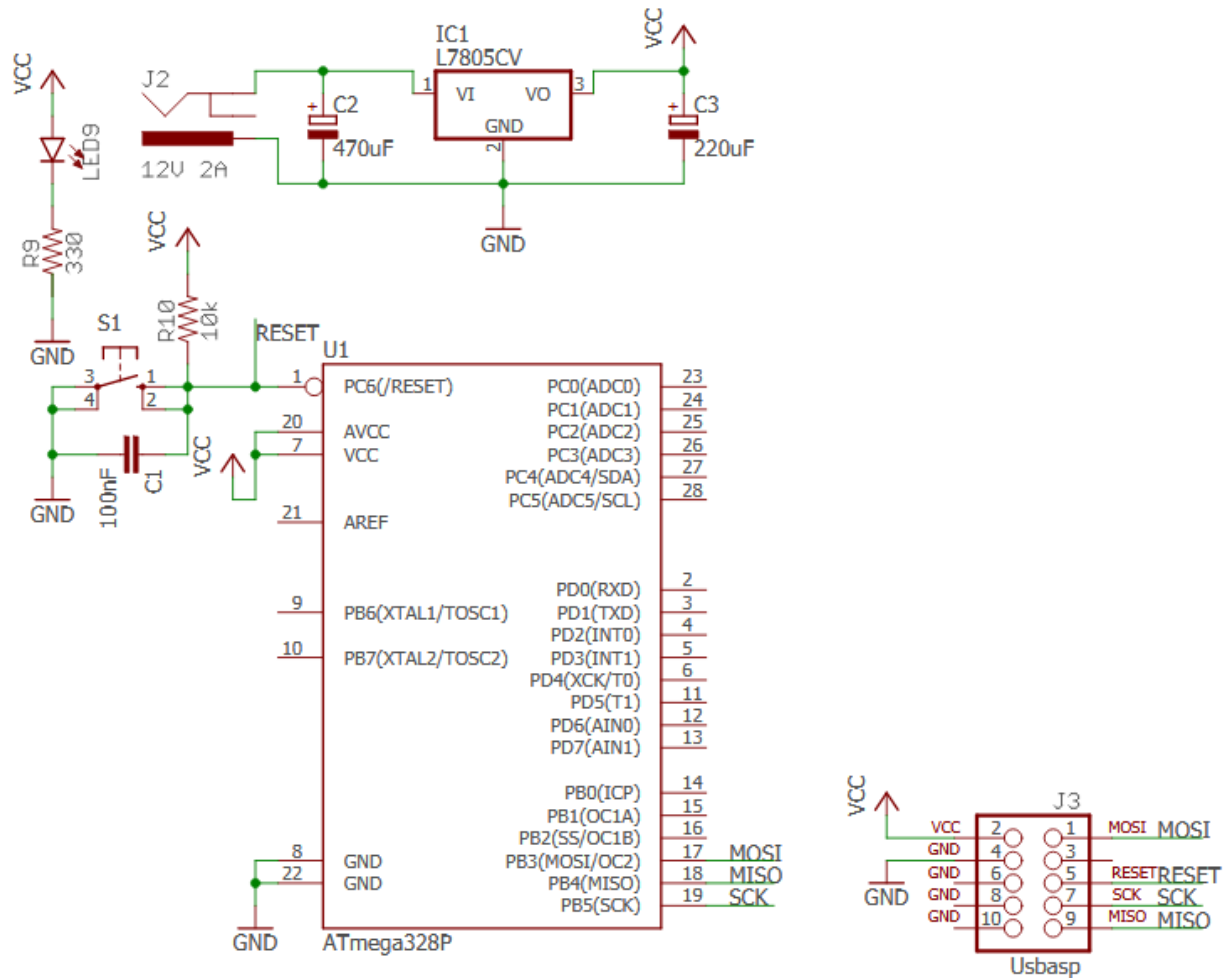


Figura 3.1: Conexión del circuito base .

- Conecte su fuente de alimentación. Con un multímetro cheque que realmente **VCC** tenga el valor de **5.0V**. Borre el contenido de la memoria del microcontrolador.
- Conecte el diagrama de la práctica 3 de acuerdo a la Figura ??:



Figura 3.2: Conexión de la práctica 3.

3.3. 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 [No](#).
- Posteriormente debemos dar el nombre [prac03](#) al proyecto.
- Después seleccionemos el modelo del microcontrolador de la lista: [ATmega328P](#). Al final presionamos [OK](#).
- 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.

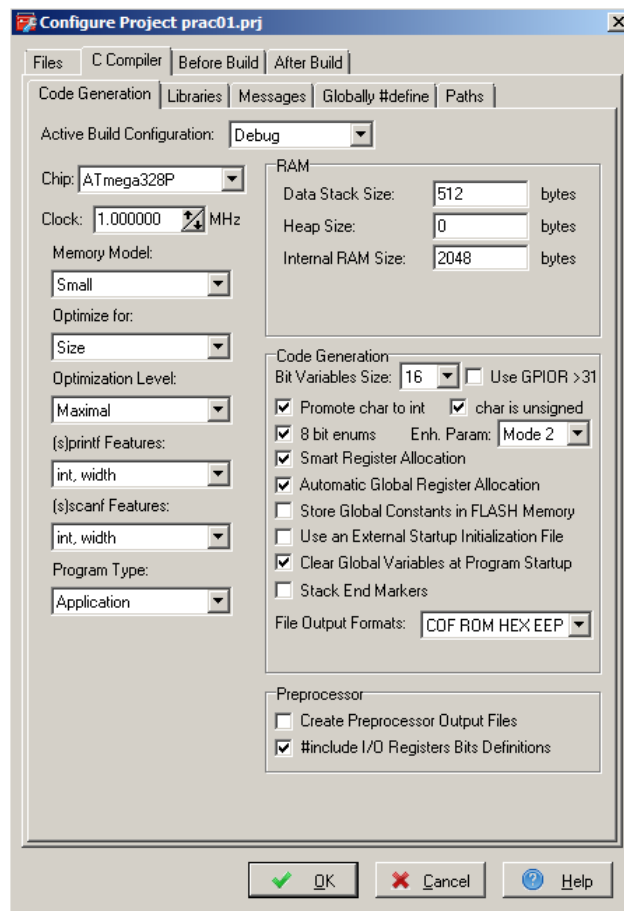


Figura 3.3: Estableciendo la velocidad del oscilador.

- Modifique el código como se muestra a continuación:

```
#include <mega328p.h>
#include <delay.h>
#define xtal 1000000L

/*
    Codificación de cada dígito para display de 7 segmentos
    Dig_0=0x40, Dig_1=0x79, Dig_2=0x24, Dig_3=0x30, Dig_4=0x19
    Dig_5=0x12, Dig_6=0x02, Dig_7=0x78, Dig_8=0x00, Dig_9=0x18
*/
const unsigned char tabla7segmentos[10]={0x40,0x79,0x24,0x30,0
    x19,0x12,0x02,0x78,0x00,0x18};
```

```

void main(void)
{
    //i=variable para conteo inicializada en 0
    unsigned char i=0;

    /* Configuración E/S
    Puerto D:
    PD7=Entrada, Pull-up
    PD6=Salida, Estado Inicial= Vcc
    PD5=Salida, Estado Inicial= Vcc
    PD4=Salida, Estado Inicial= Vcc
    PD3=Salida, Estado Inicial= Vcc
    PD2=Salida, Estado Inicial= Vcc
    PD1=Salida, Estado Inicial= Vcc
    PD0=Salida, Estado Inicial= Vcc
    */

    DDRD =0x7F;
    PORTD=0xFF;

    while (1)
    {
        PORTD=tabla7segmentos[i]; //Obtiene la codificación del
            dígito i
        delay_ms(500);             //Temporización 500ms
        i++;                       //Incrementa el valor de i en 1
        if(i>9)                    //Si i es mayor a 9 resetea su
            valor a 0
        {
            i=0;
        }
    }
}

```

- 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 **prac03.hex** en la memoria Flash del microcontrolador.
- Muestre el circuito funcionando al profesor, para que le sea tomado en cuenta. En el display deberá aparecer un conteo automático desde **0** hasta **9**.

3.4. Ejercicio 2.

- Modifique el código como se muestra a continuación:

```
#include <mega328p.h>
#include <delay.h>
#define xtal 1000000L

#define boton PIND.7

/*
  Codificación de cada dígito para display de 7 segmentos
  Dig_0=0x40, Dig_1=0x79, Dig_2=0x24, Dig_3=0x30, Dig_4=0x19
  Dig_5=0x12, Dig_6=0x02, Dig_7=0x78, Dig_8=0x00, Dig_9=0x18
*/
const unsigned char tabla7segmentos[10] = {0x40, 0x79, 0x24, 0x30, 0x19, 0x12, 0x02, 0x78, 0x00, 0x18};

void main(void)
{
    //i=variable para conteo inicializada en 0
    unsigned char i = 0;
    bit botonP = 0;
    bit botonA = 0;

    /* Configuración E/S
    Puerto D:
    PD7=Entrada, Pull-up
    PD6=Salida, Estado Inicial= Vcc
    PD5=Salida, Estado Inicial= Vcc
    PD4=Salida, Estado Inicial= Vcc
    PD3=Salida, Estado Inicial= Vcc
    PD2=Salida, Estado Inicial= Vcc
    PD1=Salida, Estado Inicial= Vcc
    PD0=Salida, Estado Inicial= Vcc
    */

    DDRD = 0x7F;
    PORTD = 0xFF;

    while (1)
    {
        if(boton == 0)
        {
            botonA = 0;
        }
    }
}
```



```

    }
    else
    {
        botonA = 1;
    }
    if((botonA == 0) && (botonP == 1))
    {
        i++;
        if(i > 9)
        {
            i = 0;
        }
        delay_ms(40);
    }
    if((botonA == 1) && (botonP == 0))
    {
        delay_ms(40);
        PORTD = tabla7segmentos[i];
        botonP = botonA;
    }
}

```

- Compile, grabe y muestre el resultado al profesor para su evaluación. Si se presiona **S2**, el dígito mostrado en el display deberá aumentar en uno.

3.5. Ejercicio 3.

- Modifique el código del Ejercicio 2 para que el conteo decimal se convierta en un conteo hexadecimal, es decir que se agreguen los siguientes dígitos al conteo:



Figura 3.4: Agregue los siguientes dígitos al conteo

- Compile, grabe y muestre el resultado al profesor para su evaluación. Si se presiona **S2**, el dígito mostrado en el display deberá aumentar en uno, desde **0** hasta **F**.

- Adjunte el código resultante en esta hoja: