

Nombre: _____ Grupo: _____

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

Práctica 2

Entradas digitales

2.1. Material

El material enlistado es necesario para la realización de la práctica 2.

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
4	Resistencia de 220 Ω 1/4W	Código: Rojo, rojo, café, oro	R11, R13, R15, R17
7	Resistencia de 330 Ω 1/4W	Código: Naranja, naranja, café, oro	R1 ... R4, R9, R19, R20
2	Resistencia de 1k Ω 1/4W	Código: Café, negro, rojo, oro	R14, R18
7	Resistencia de 10k Ω 1/4W	Código: Café, negro, naranja, oro	R5 ... R8, R10, R12, R16
1	Push button (Microswitch)	Tipo push, 4 o 2 terminales	S1
1	Dip-switch	4 vías	SW1
7	Led 5mm difuso	Color rojo	LED1 ... LED7
2	Sensores CNY70	Sensor óptico reflexivo	U\$1, U\$2
2	Transistores BC547	Transistor NPN	Q1, Q2
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		

2.2. Conexión de los componentes

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

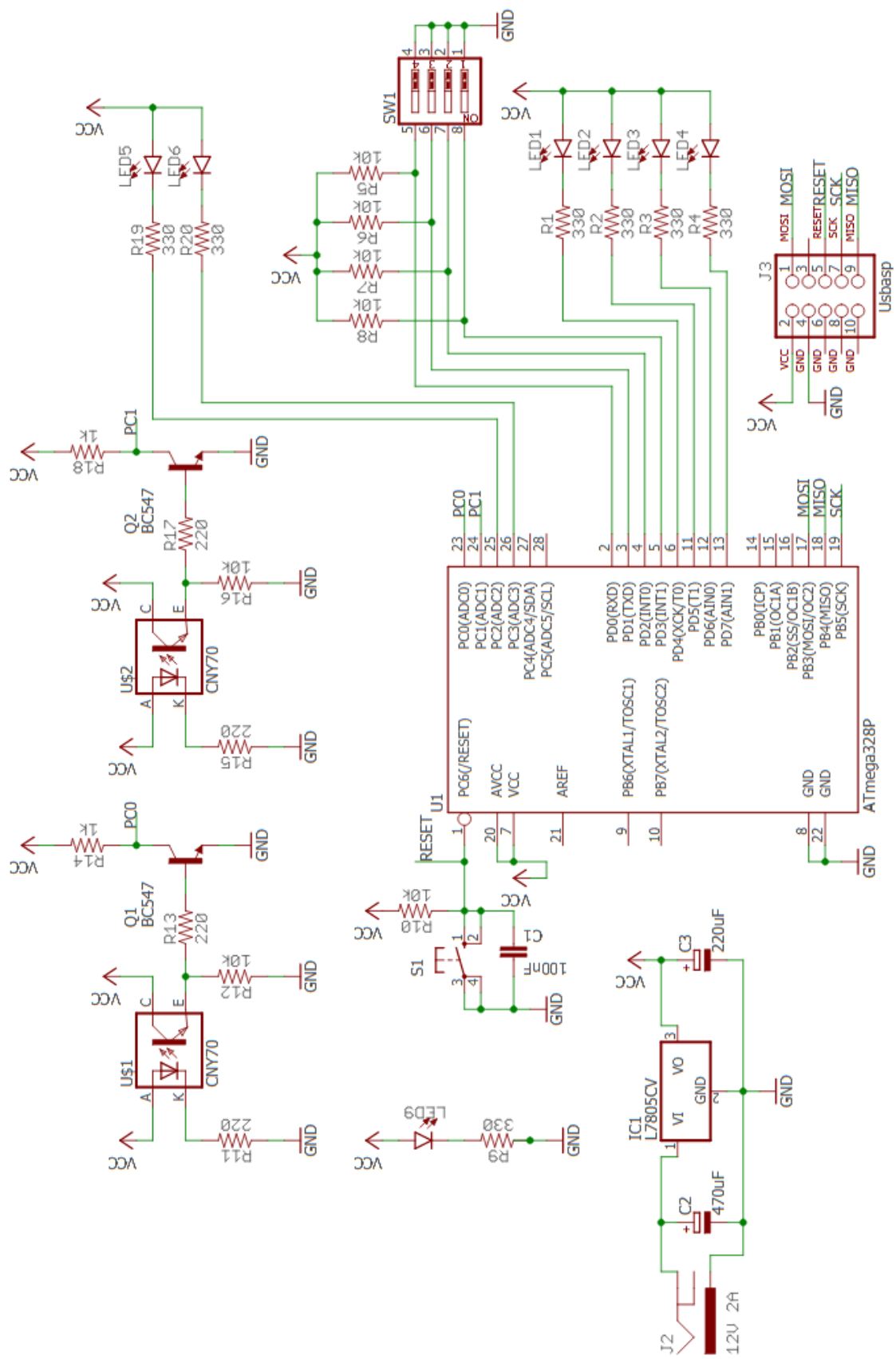


Figura 2.2: Conexión de la práctica 2.

2.3. 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 [prac02](#) 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.

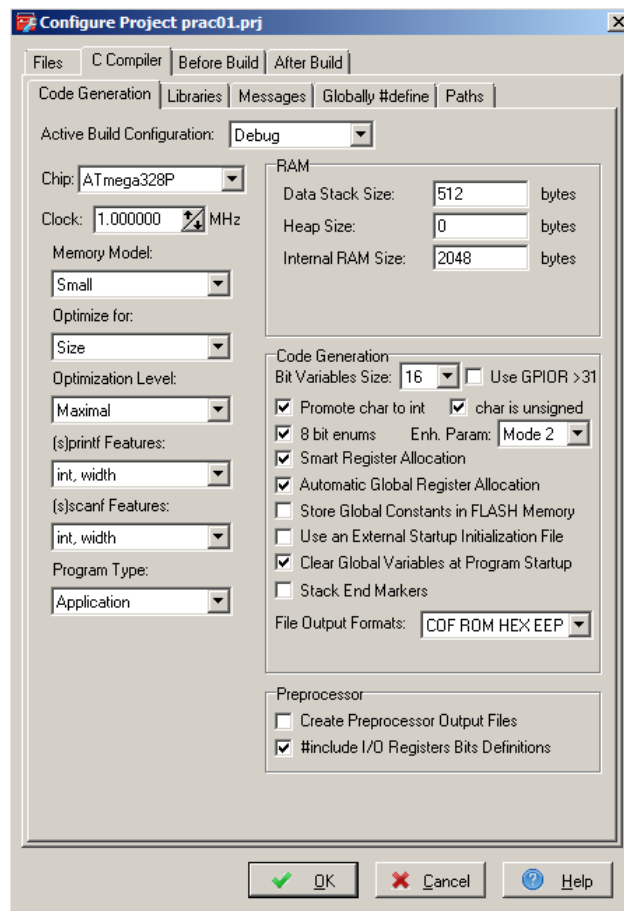


Figura 2.3: Estableciendo la velocidad del oscilador.

- Modifique el código como se muestra a continuación:

```
#include <mega328p.h>

void main(void)
{
    //Declaración de variables
    bit sw0, sw1, sw2, sw3;
    bit sens0, sens1;

    /* Configuración E/S
    Puerto D:
    PD7=Salida, Estado Inicial= Vcc
    PD6=Salida, Estado Inicial= Vcc
```

```

    PD5=Salida, Estado Inicial= Vcc
    PD4=Salida, Estado Inicial= Vcc
    PD3=Entrada, Pull-up
    PD2=Entrada, Pull-up
    PD1=Entrada, Pull-up
    PD0=Entrada, Pull-up

    Puerto C:
    PC3=Salida, Estado Inicial= Vcc
    PC2=Salida, Estado Inicial= Vcc
    PC1=Entrada, Pull-up
    PC0=Entrada, Pull-up
*/
DDRD=0xF0;
PORTD=0xFF;

DDRC=0x0C;
PORTC=0x0F;

while (1)
{
    //Lectura del estado del DipSwitch
    sw0=PIND.0;
    sw1=PIND.1;
    sw2=PIND.2;
    sw3=PIND.3;

    //Lectura del estado de los sensores
    sens0=PINC.0;
    sens1=PINC.1;

    //Pulso de salida para los Leds (PD)
    PORTD.4=sw0;
    PORTD.5=sw1;
    PORTD.6=sw2;
    PORTD.7=sw3;

    //Pulso de salida para los Leds (PC)
    PORTC.2=sens0;
    PORTC.3=sens1;
}
}

```

- 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 [prac02.hex](#) en la memoria Flash del microcontrolador.
- Muestre el circuito funcionando al profesor, para que le sea tomado en cuenta.

2.4. Ejercicio 2.

- Modifique el código como se muestra a continuación:

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

void main(void)
{
    //Declaración de variables
    unsigned char lecturaSW;
    const unsigned int tiempo = 500;
    bit sens0, sens1;

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

    Puerto C:
    PC3=Salida, Estado Inicial= Vcc
    PC2=Salida, Estado Inicial= Vcc
    PC1=Entrada, Pull-up
    PC0=Entrada, Pull-up
    */
    DDRD=0xF0;
    PORTD=0xFF;

    DDRC=0x0C;
    PORTC=0x0F;

    while (1)
    {
```

```

//Lectura de los estados de los sensores y Dip-Switch
sens0=PINC.0;
sens1=PINC.1;
lecturaSW=PIND & 0x0F;

if(lecturaSW==0x0E)
{
    //Prende todos los Leds
    PORTD=0x00;
}
else if(lecturaSW==0x0D)
{
    //Prende y apaga con temporización todos los Leds
    PORTD=0x00;
    delay_ms(tiempo);
    PORTD=0xF0;
    delay_ms(tiempo);
}
else if(lecturaSW==0x0B)
{
    //Efecto Walking
    PORTD=0xE0;
    delay_ms(tiempo);
    PORTD=0xD0;
    delay_ms(tiempo);
    PORTD=0xB0;
    delay_ms(tiempo);
    PORTD=0x70;
    delay_ms(tiempo);
}
else if(lecturaSW==0x07)
{
    //Efecto Walking Inverso
    PORTD=0x70;
    delay_ms(tiempo);
    PORTD=0xB0;
    delay_ms(tiempo);
    PORTD=0xD0;
    delay_ms(tiempo);
    PORTD=0xE0;
    delay_ms(tiempo);
}
else
{
    //Apaga todos los Leds
    PORTD=0xF0;
}

```

```
    }  
  
    if((sens0==0) && (sens1==0))  
    {  
        PORTC.2=0;  
        PORTC.3=0;  
    }  
    else  
    {  
        PORTC.2=1;  
        PORTC.3=1;  
    }  
}  
}
```

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