

Práctica 1 – Microcontroladores

Instituto Tecnológico Autónomo de México
Departamento Académico de Sistemas Digitales
Laboratorio de Principios de Mecatrónica

Primavera 2022

1 Objetivos

- Conocer las plataformas, herramientas y ambientes más comunes para el desarrollo de sistemas mecatrónicos basados en microcontroladores.
- Identificar los elementos básicos de la arquitectura de un microcontrolador digital.
- Identificar los módulos que componen a la tarjeta de desarrollo Arduino y sus interconexiones básicas.
- Realizar una comparativa entre las implementaciones en código de bajo y alto nivel, destacando las ventajas y pertinencia de cada una de las estrategias de programación de microcontroladores.

2 Arduino MEGA 2560

■ Especificaciones técnicas

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

3 Recursos de la práctica

3.1 Material y Equipo

- 1 Arduino MEGA
- 1 Cable USB A/B
- 2 LED rojo
- 2 LED amarillo
- 2 LED verde
- 1 Push Button
- 6 Resistor 220 Ω
- 1 Resistor 10 k Ω

4 Procedimientos

4.1 Blink (encendido y apagado de un diodo LED)

1. Ejecutar una instancia ambiente integrado de desarrollo (IDE) de Arduino y descargar el código de ejemplo Blink (*Archivo > Ejemplos > 01. Basics > Blink*).
2. Identificar las partes y estructura del código y determinar la función que tiene cada una de las instrucciones presentes en el código. Señalar ciclos, comandos, argumentos, palabras reservadas, comentarios, etc.

```
void setup()
{
    pinMode(LED_BUILTIN, OUTPUT);    // initialize digital pin LED_BUILTIN as an output.
}

void loop()
{
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000);                     // wait for a second
    digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
    delay(1000);                     // wait for a second
}
```

3. Realizar las conexiones que se muestran para incorporar un LED y un botón externos a la tarjeta Arduino (Figura 1). Identificar los puertos digitales D12 y D13, el puerto de alimentación **+5V** y los nodos de tierra (GND) (Figura 2). Alimentar el sistema mediante el cable USB tipo A/B conectado a la PC.

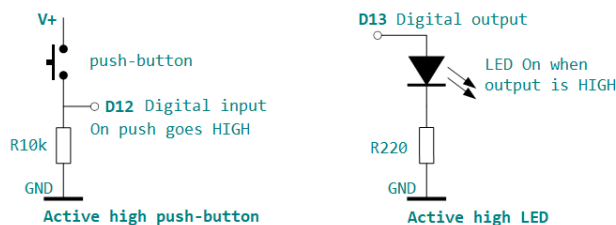


Figura 1: Conexión de LED y botón externos.

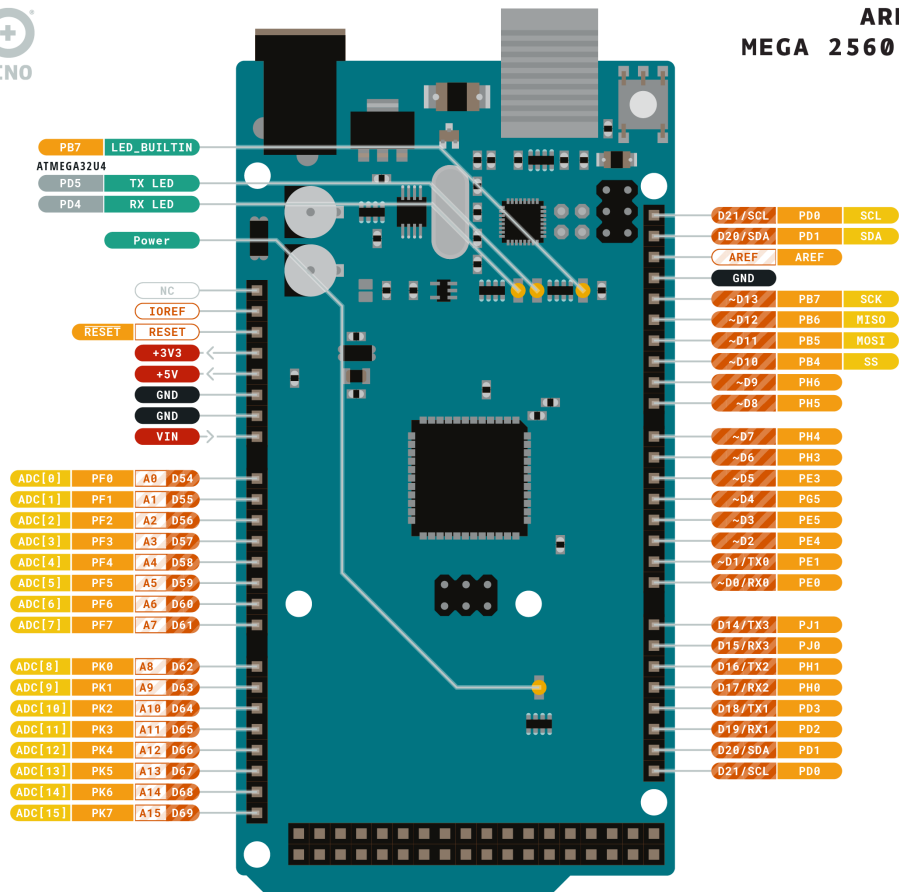
4. Compilar/verificar el código (**Ctrl+R**)
5. Verificar la selección de tarjeta y puerto COM en el menú **Herramientas** y cargar el programa a la tarjeta Arduino (**Ctrl+U**). Analizar el comportamiento del sistema en función del código suministrado.
6. Realizar la programación necesaria para incorporar funcionalidad en el push-button conforme en el Cuadro 1 y comprobar su funcionamiento.

Estado del PB	Frecuencia del LED
LOW	1 Hz
HIGH	0.5 Hz

Cuadro 1: Cambio de frecuencia en parpadeo.



ARDUINO MEGA 2560 REV3



Ground	Internal Pin	Digital Pin	Microcontroller's Port
Power	SWD Pin	Analog Pin	
LED	Other Pin	Default	

ARDUINO.CC



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Figura 2: Esquema de puertos Arduino MEGA.

4.2 Blink ASM

1. Sustituir el contenido del ciclo **Void Loop** () por el siguiente código:

```
void setup()
{
    DDRB = DDRB | B10000000; // Data Direction Register B: Inputs 0-6, Output 7
}
void loop()
{
    asm (
        "inicio: \n\t"
        "sbi 0x05,0x07 \n\t"
        "call tiempo \n\t"
        "cbi 0x05,0x07 \n\t"
        "call tiempo \n\t"
        "jmp main \n\t"

        "tiempo: \n\t"
        "LDI r22, 45 \n\t"
        "LOOP_3: \n\t"
        "LDI r21, 255 \n\t"
        "LOOP_2: \n\t"
        "LDI r20, 255 \n\t"
        "LOOP_1: \n\t"
        "DEC r20 \n\t"
        "BRNE LOOP_1 \n\t"
        "DEC r21 \n\t"
        "BRNE LOOP_2 \n\t"
        "DEC r22 \n\t"
        "BRNE LOOP_3 \n\t"
        "ret \n\t"
    );
}
```

2. Analizar la función de cada uno de los comandos mnemónicos e identificar la estructura de la rutina de tiempo.
3. Ajustar los parámetros variables de la rutina de tiempo y añadir los comandos necesarios para lograr la misma funcionalidad que en el punto 4.1.6 dentro del bloque de código ASM. Comprobar el funcionamiento.

4.3 Semáforo digital Arduino

1. Realizar las conexiones necesarias para la implementación de un semáforo digital (Figura 3). Tres LEDs de diferentes colores por cada uno de las direcciones del tránsito (seis puertos digitales de salida en total).
2. Realizar la programación para lograr los estados del semáforo conforme al siguiente cuadro:

Estado semáforo 1			Estado semáforo 2			Duración (s)
rojo	amarillo	verde	rojo	amarillo	verde	
HIGH	LOW	LOW	LOW	LOW	HIGH	5
HIGH	LOW	LOW	LOW	HIGH	LOW	1
LOW	LOW	HIGH	HIGH	LOW	LOW	5
LOW	HIGH	LOW	HIGH	LOW	LOW	1

Cuadro 2: Secuencia de estados del semáforo.

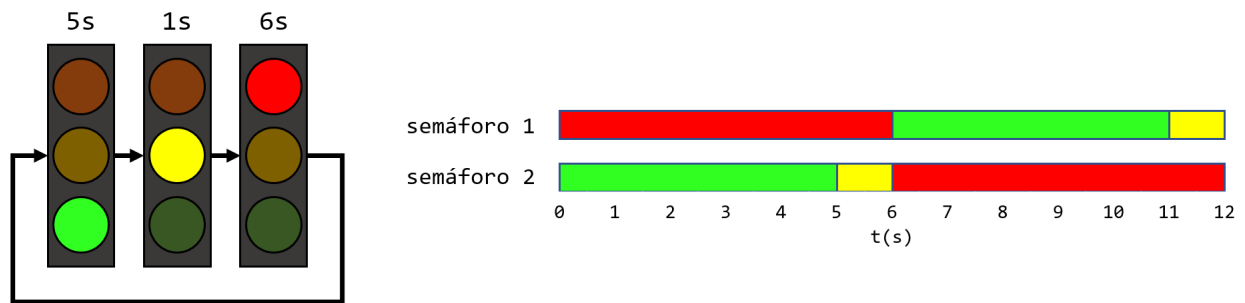


Figura 3: Conexión de LED y botón externos.

Referencias Recomendadas

Arduino reference <https://www.arduino.cc/reference/en/>

ATMEGA2560 <https://www.microchip.com/wwwproducts/en/ATmega2560>

Arduino MEGA <https://store.arduino.cc/usa/mega-2560-r3>