**INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO**





Laboratorio de Principios de Mecatrónica

**Práctica 1. Microcontroladores**

Estudiantes:

* Plauchú Rodríguez Rodrigo 182671
* Castillejos Corzo Victor Hugo 182344

Asignatura: Laboratorio de Principios de Mecatrónica

Docente: M.I. Sergio Hernández Sánchez

Grupo: \_04\_

1. **Introducción**

Hoy en día, los microcontroladores son ocupados para cualquier aparato tecnológico, ya sea, cotidiano o científico. Por lo anterior, es muy importante, casi indispensable, conocer el funcionamiento de estos porque así podremos crear conocimiento que beneficie a la comunidad. Esta práctica consiste en aprender a utilizar el microcontrolador que pertenece a la placa de desarrollo Arduino MEGA.

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.

1. **Marco Teórico**

**Microcontrolador:** Es un equipo con las mismas características de una computadora, solo que su tamaño es más pequeño. Tiene un CPU, una memoria RAM y una memoria ROM. Es el cerebro de un sistema informático y el motor que activa el funcionamiento de un equipo. Su utilidad está presente en muchas áreas de la vida cotidiana, en la industria cumple una tarea fundamental, ya que es utilizado como complemento en la automatización de diversas operaciones.

**Placa de desarrollo Arduino MEGA:**

El [Arduino Mega 2560](https://www.mcielectronics.cl/shop/product/arduino-mega-2560-r3-10231?search=Mega+2560) es una placa de desarrollo basada en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 14 pueden ser usadas como salidas PWM), 16 entradas analógicas, voltaje de entrada de 7 a 12V, 256K de memoria flash, un reloj con un cristal de 16Mhz, [conexión USB](https://www.mcielectronics.cl/shop/product/cable-usb-a-macho-b-macho-6ft-1-82m-9602), jack para [alimentación DC](https://www.mcielectronics.cl/shop/product/transformador-ac-dc-9v-650ma-regulado-jack-dc-5-5x2-1mm-9591), conector ICSP, y un botón de reseteo. La placa Mega 2560 es compatible con la mayoría de shields compatibles para Arduino UNO.

Las resistencias pull-up y pull-down se utilizan para polarizar correctamente las entradas de las puertas digitales y evitar que floten de forma aleatoria cuando no hay ninguna condición de entrada.

1. **Material y equipo utilizado**
   1. 1 Arduino MEGA
   2. 1 cable USB A/B
   3. 2 LED rojo
   4. 2 LED amarillo
   5. 2 LED verde
   6. 1 Push-Button
   7. 6 resistor 220 Ω
   8. 1 resistor 10 k Ω
2. **Experimentos y simulaciones**

**Nota: No pudimos tomar foto en el lab de las primeras actividades**

* 1. **Actividad 1 – Blink**

**////4.1**

**//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**

**//}**

* 1. **Actividad 2 – Frecuencia**

**\*4.2**

**int ledPin = 13; // LED connected to digital pin 13**

**int inPin = 12; // push button connected to digital pin 7**

**int input = 0;**

**void setup()**

**{**

**pinMode(ledPin, OUTPUT); // initialize digital pin D13 as an output.**

**pinMode(inPin, INPUT); // initialize digital pin D12 as an input.**

**}**

**void loop()**

**{**

**input = digitalRead(inPin); // read the input pin**

**digitalWrite(ledPin, input); // sets the LED to the button's value**

**}**

* 1. **Actividad 3 – ASM**

**4.3**

**int ledPin = 13; // LED connected to digital pin 13**

**int inPin = 12; // push button connected to digital pin 7**

**int input = 0;**

**void setup()**

**{**

**pinMode(ledPin, OUTPUT); // initialize digital pin D13 as an output.**

**pinMode(inPin, INPUT); // initialize digital pin D12 as an input.**

**}**

**void loop()**

**{**

**input = digitalRead(inPin); // read the input pin**

**if(input == LOW){**

**digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)**

**delay(1000); // wait for a second**

**digitalWrite(ledPin, LOW); // turn the LED off by making the voltage LOW**

**delay(1000); // wait for a second**

**}else{**

**digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)**

**delay(2000); // wait for a second**

**digitalWrite(ledPin, LOW); // turn the LED off by making the voltage LOW**

**delay(2000); // wait for a second**

**}**

**}**

* 1. **Actividad 4 – Semáforo**

**//4.4**

**int sm1\_ledVerde = 8;**

**int sm1\_ledAmarillo = 9;**

**int sm1\_ledRojo = 10;**

**int sm2\_ledVerde = 11;**

**int sm2\_ledAmarillo = 12;**

**int sm2\_ledRojo = 13;**

**void setup()**

**{**

**pinMode(sm1\_ledVerde, OUTPUT);**

**pinMode(sm1\_ledAmarillo, OUTPUT);**

**pinMode(sm1\_ledRojo, OUTPUT);**

**pinMode(sm2\_ledVerde, OUTPUT);**

**pinMode(sm2\_ledAmarillo, OUTPUT);**

**pinMode(sm2\_ledRojo, OUTPUT);**

**}**

**void loop()**

**{**

**digitalWrite(sm1\_ledRojo, HIGH);**

**digitalWrite(sm2\_ledVerde, HIGH);**

**delay(5000); // wait for 5 seconds**

**digitalWrite(sm2\_ledAmarillo, HIGH);**

**digitalWrite(sm2\_ledVerde, LOW);**

**delay(1000); // wait for a second**

**digitalWrite(sm1\_ledRojo, LOW);**

**digitalWrite(sm2\_ledAmarillo, LOW);**

**digitalWrite(sm1\_ledVerde, HIGH);**

**digitalWrite(sm2\_ledRojo, HIGH);**

**delay(5000); // wait for 5 seconds**

**digitalWrite(sm1\_ledVerde, LOW);**

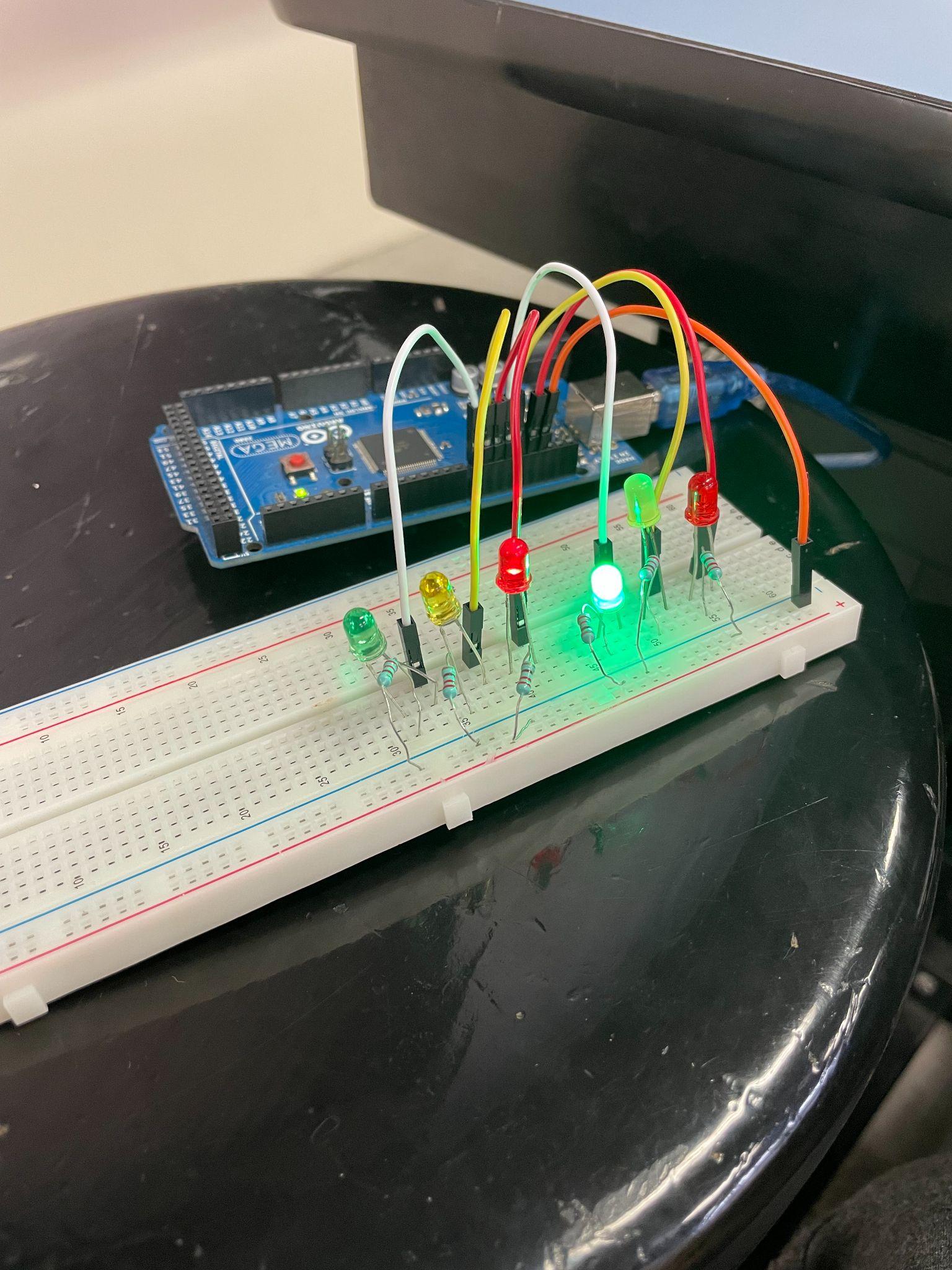
**digitalWrite(sm1\_ledAmarillo, HIGH);**

**delay(1000); // wait for a second**

**digitalWrite(sm1\_ledAmarillo, LOW);**

**digitalWrite(sm2\_ledRojo, LOW);**

**}**

****

1. **Conclusiones**

Después de realizar la práctica 1 con todas sus actividades aprendimos a trabajar con arduino, al principio tuvimos unos problemas, lo cual es normal ya que es la primera vez que trabajamos con ese software, esto se resolverá a lo largo de las siguientes prácticas. También es importante mencionar que en la actividad 3 y 4 tuvimos unos inconvenientes, ya que nuestra protoboard no funcionaba y nos tardamos en dar cuenta que era por eso; sin embargo, pudimos resolverlo y logramos ejecutar todos los códigos de forma exitosa. Finalmente, como equipo nos dimos cuenta que hay microcontroladores en casi todos lados y ahora tenemos una mucha mejor noción de cómo están implementados y de qué forma trabajan.

1. **Referencias**

Aguayo, P. (2019, enero 15). *Arduino Mega 2560*. Arduino.cl - Compra tu Arduino en Línea. <https://arduino.cl/arduino-mega-2560/>

Redacción. (2021, abril 7). *Resistencias pull-up y pull-down en Arduino*. Descubrearduino.com. <https://descubrearduino.com/resistencias-pull-up-y-pull-down-en-arduino/>

support support. (2021, julio 15). Industriasgsl.com / Venta de Suministros Industriales. *Industrias GSL*. <https://www.industriasgsl.com/blog/post/que_es_un_microcontrolador>

*Tarjeta de Desarrollo Arduino Mega 2560 R3*. (s/f). Mcielectronics.cl. Recuperado el 16 de febrero de 2022, de

<https://www.mcielectronics.cl/shop/product/arduino-mega-2560-r3-arduino-10231?search=Mega+2560>