Práctica 2 – Arduino

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

Primavera 2022

1 Objetivos

- Identificar las principales librerías y comandos del lenguaje Arduino, en particular lo relacionado con sensores y actuadores.
- Hacer uso de periféricos del microcontrolador tales como el Convertidor Analógico - Digital (ADC) y la Modulación por ancho de pulso (PWM).
- Interactuar con componentes externos a la tarjeta Arduino tanto para la adquisición de señales como para el desplegado de información.
- Manejo del monitor serial y del display LCD.

2 Arduino MEGA 2560

- **Especificaciones ADC**

Operating Voltage	5V
Analog Input Pins	A0 to A14
Resolution	10 bits (4.88 mV/LSB)

- **Especificaciones DAC (PWM)**

Analog Output Pins	2 to 13, 44 to 46
PWM Frequency	490 Hz (pins 4 and 13: 980 Hz)

- **Especificaciones Servo (PWM)**

Pulse width	544 to 2400 μs
Angle	0 to 180°

3 Recursos de la práctica

3.1 Material y Equipo

1 Arduino MEGA
1 Cable USB A/B
1 Potenciómetro 10 kΩ
1 Servo motor
1 Display LCD
1 LED
1 Resistor 220 Ω

4 Procedimientos

4.1 Convertidor Analógico Digital

1. Realizar la conexión de un potenciómetro de $10k\Omega$ a la terminal analógica A0 y añadir un LED al puerto digital D13. (Figura 1).

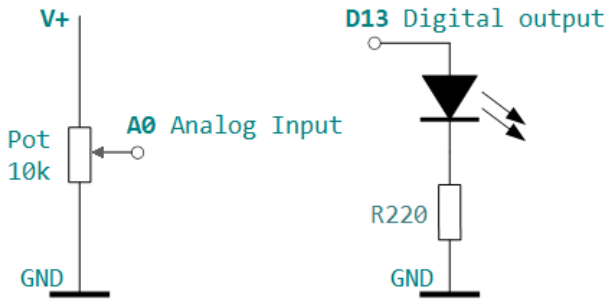

Figura 1: Conexión Convertidor Analógico-Digital (ADC).

2. Cargar el siguiente código, abrir el monitor serial en el IDE de Arduino y verificar el valor que se despliega conforme se hacen cambios en la posición del potenciómetro.

```
int analogPin = A0; // potentiometer wiper (middle terminal) connected to analog pin 3
                    // outside leads to ground and +5V
int val = 0;  // variable to store the value read

void setup() {
  Serial.begin(9600);           //  setup serial
}
void loop() {
  val = analogRead(analogPin);  // read the input pin
  Serial.print ("Conversión analógico-digital: ");
  Serial.println(val);          // debug value
}
```

3. Realizar los cambios necesarios para que el monitor despliegue el valor de voltaje (entre 0 y 5V) que corresponde con la posición del potenciómetro.

4. Realizar los cambios necesarios para que el LED encienda si y sólo si el voltaje que entra al puerto A0 es mayor a 3V.

5. Realizar los cambios necesarios para que el LED encienda con una intensidad proporcional al voltaje de entrada en el puerto A0 desde 0 hasta 5V. Utilizar la función **AnalogWrite()**.

4.2 Liquid-Crystal Display (LCD)

1. Identificar las posiciones de escritura en el display de cristal líquido (LCD) (Figura 2) y los caracteres que se pueden imprimir en las mismas (Figura 5).

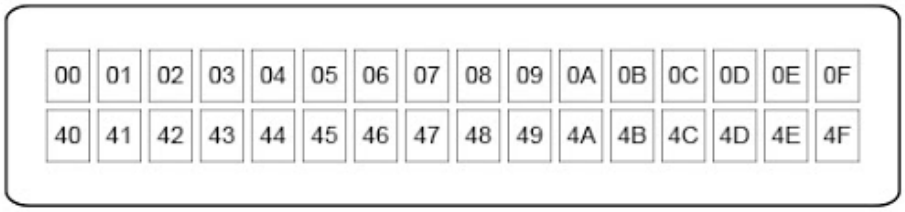

Figura 2: Posiciones de escritura en el LCD (direcciones de memoria).

2. Realizar las conexiones necesarias para operar el LCD (Figura 3).

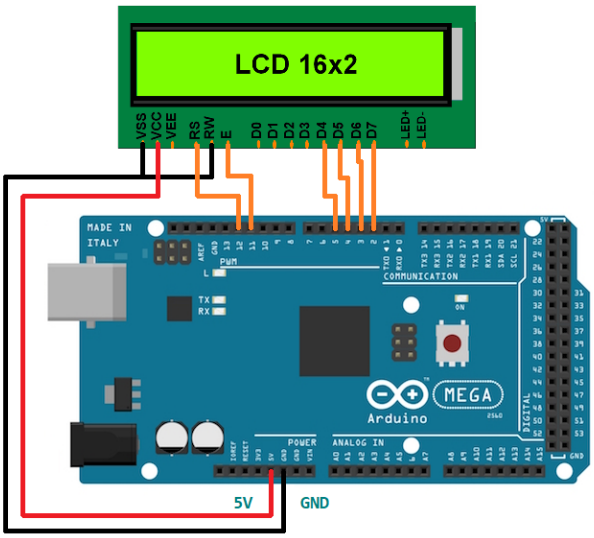

Figura 3: Conexión LCD.

3. Cargar el siguiente siguiente código en la tarjeta Arduino y hacer las modificaciones pertinentes para que el texto parpadee una vez por segundo. Sugerencia: $0.75s$ encendido, $0.25s$ apagado.

```
#include <LiquidCrystal.h>  // include the library code:
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // initialize the interface pins

void setup() {
  lcd.begin(16, 2);     // set up the LCD's number of columns and rows:
}
void loop() {
  lcd.setCursor(1, 0);
  lcd.print("Principios de");// Print a message to the LCD.
  lcd.setCursor(2, 1);
  lcd.print("Mecatronica");
}
```

4. Realizar un programa en Arduino que despliegue el nombre completo del estudiante en el LCD. El nombre debe comenzar a aparecer por el margen derecho del display y detenerse cuando se llegue a la extrema izquierda, pueden usarse ambas filas para nombres y apellidos, respectivamente.

4.3 Servomotor

1. Realizar las conexiones necesarias para operar un servo-motor (Figura 4).

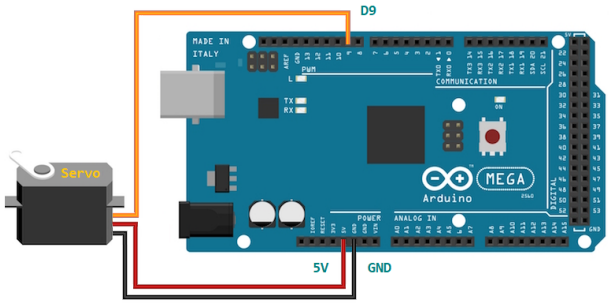

Figura 4: Conexión de un servo-motor.

2. Cargar el siguiente programa, describir su comportamiento y verificar su funcionamiento.

```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo

int val = 0;    // variable to read the value from the analog pin
void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (int i = 0; i <= 180; i++)
  {
    myservo.write(i);                  // sets the servo position according to the scaled value
    delay(20);                           // waits for the servo to get there
  }
  for (int i = 180; i >= 0; i--)
  {
    myservo.write(i);                  // sets the servo position according to the scaled value
    delay(20);                           // waits for the servo to get there
  }
```

3. Realizar los cambios necesarios para que la posición del servo-motor se corresponda con la posición del potenciómetro. Sugerencia: utilizar función **map()**.

4. Mostrar en el LCD los valores de voltaje (lectura ADC) y ángulo enviado al servo-motor.

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

Figura 5: Caracteres LCD.