Universidad Intercontinental de la Empresa

Tecnologías de Automatización y Robotización para Empresas

IoT

Placa Arduino Uno

Práctica de Laboratorio 03

Entradas Analógicas

y

Salidas a una Pantalla LCD

Profesor Eladio Dapena Gonzalez

Curso 2024-2025

Práctica 03: Entradas Analógicas y Salidas a una Pantalla LCD.

1 Introducción

Durante la actividad de laboratorio se experimenta con el funcionamiento y gestión de los pines de entrada analógicas de la placa Arduino Uno.

Adicionalmente, se enviarán datos a una pantalla de cristal líquido LCD.

2 Objetivos

a) Leer el valores de entradas analógicas de la placa Arduino Uno.

b) Conocer el funcionamiento de un potenciómetro y realizar sus conexiones correctamente.

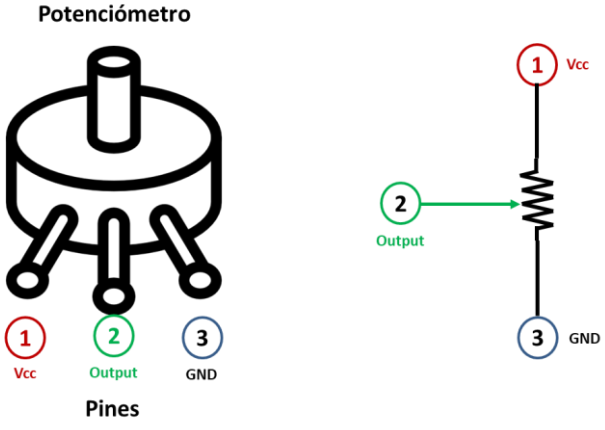
3 Materiales Requeridos

- 1 Placa Arduino UNO.
- 1 Protoboard.
- 1 Potenciómetro de 10K Ω.
- 1 Pantalla LCD 16x2.
- 6 Diodos LED.
- 7 Resistencias de 220 Ω.
- Cables para realizar las conexiones.

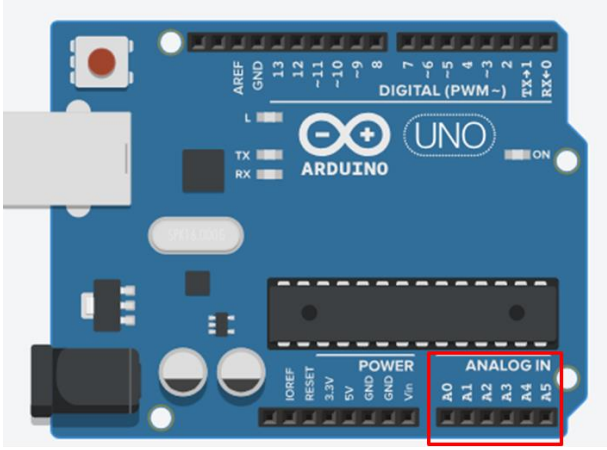
4 Marco teórico

4.1 Potenciómetro

Es una **resistencia variable** de tres terminales, cuyo comportamiento se basa en el divisor de tensión, donde la resistencia cambia su valor mediante la perilla frontal.



4.2 Pines Analógicos Placa Arduino



Pines Analógicos

Placa	Voltaje de Operación	Pines Analógicos	Máxima Resolución
UNO R3	5 Volts	A0 to A5	10 bits

4.3 Funciones.

Funciones	Descripción
analogRead(pin)	Lectura del valor de un pin analógico especificado. Las placas Arduino contienen un **convertidor de analógico a digital** multicanal de **10 bits**. Esto significa que convertirá los voltajes de entrada entre **0** y el voltaje de funcionamiento (**5 V o 3,3 V**) en valores enteros entre **0** y **1023**. En un Arduino UNO, por ejemplo, esto produce una resolución entre lecturas de: 5 voltios / 1024 unidades o 0,0049 voltios (4,9 mV) por unidad.
analogWrite(pin, valor)	Escribe un valor analógico PWM (Modulación de Ancho de Pulso) en un pin. Se puede utilizar para encender un LED con distintos niveles de brillo o para accionar un motor a distintas velocidades. Después de una llamada a **analogWrite()**, el pin generará una onda rectangular constante del ciclo de trabajo especificado hasta la siguiente llamada a **analogWrite()** (o una llamada a digitalRead() o **digitalWrite()**) en el mismo pin. **Parámetros:** ***pin*:** el pin de Arduino en el que se escribirá. Tipos de datos permitidos: **int** (Enteros). ***valor*:** el ciclo de trabajo: entre **0** (siempre apagado) y **255** (siempre encendido).

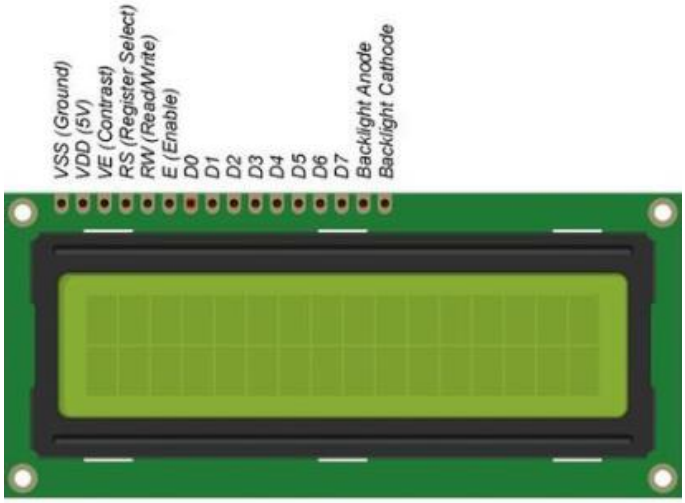
Para más información y funciones visitar el sitio web:

https://www.arduino.cc/reference/en/

4.4 Pantalla LCD 16x2

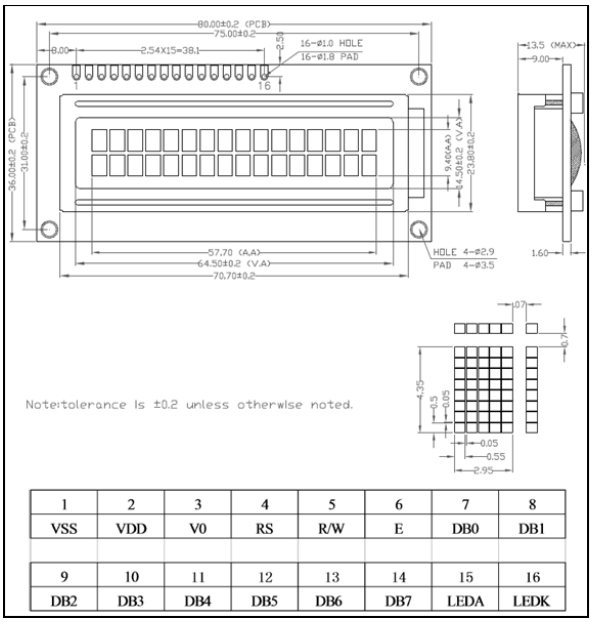
LCD (*Liquid Crystal Display*) es una pantalla electrónica basada en módulos de **siete segmentos**, que permiten mostrar caracteres alfanuméricos.

Las más comunes utilizan el driver de control **HITACHI HD44780**, el cual permite una fácil comunicación con diferentes microcontroladores.



Arduino cuenta con una biblioteca con la clase ***LiquidCrystal*** basada en el driver HITACHI HD44780.

Descripción de Pines



PIN NO.	SYMBOL	DESCRIPTION	FUNCTION
1	VSS	GROUND	0V (GND)
2	VDD	POWER SUPPLY FOR LOGIC CIRCUIT	+5V
3	V0	LCD CONTRAST ADJUSTMENT	
4	RS	INSTRUCTION/DATA REGISTER SELECTION	RS = 0 : INSTRUCTION REGISTER RS = 1 : DATA REGISTER
5	R/W	READ/WRITE SELECTION	R/W = 0 : REGISTER WRITE R/W = 1 : REGISTER READ
6	E	ENABLE SIGNAL	
7	DB0	DATA INPUT/OUTPUT LINES	8 BIT: DB0-DB7
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	LEDA	SUPPLY VOLTAGE FOR LED+	+5V
16	LEDK	SUPPLY VOLTAGE FOR LED-	0V

4.5 Funciones biblioteca Clase *LiquidCrystal* Pantalla LCD 16x2

Métodos	Descripción
LiquidCrystal()	Crea una variable de tipo ***LiquidCrystal***. La pantalla se puede controlar mediante 4 u 8 líneas de datos. Si es lo primero, omita los números de pin de d0 a d3 y deje esas líneas desconectadas. El pin RW se puede conectar a tierra en lugar de conectarlo a un pin del Arduino; en ese caso se excluye de los parámetros de esta función. Sintaxis: ***LiquidCrystal***(rs, enable, d4, d5, d6, d7) **rs:** número del pin Arduino que está conectado al pin RS en la pantalla LCD **rw:** el número del pin Arduino que está conectado al pin RW en la pantalla LCD (opcional) **enable:** el número del pin Arduino que está conectado al pin de habilitación en la pantalla LCD **d0, d1, d2, d3, d4, d5, d6, d7**: los números de los pines Arduino que están conectados a los pines de datos correspondientes en la pantalla LCD. d0, d1, d2 y d3 son opcionales; si se omite, la pantalla LCD se controlará utilizando sólo las cuatro líneas de datos (d4, d5, d6, d7). **Ejemplo:** ***LiquidCrystal Idnetiicador***(rs, enable, d4, d5, d6, d7); ***LiquidCrystal*** **LCD1**(12, 11, 5, 4, 3, 2);
var.begin(cols, rows, charsize)	Inicializa la interfaz de la pantalla LCD y especifica las dimensiones (ancho y alto) de la pantalla. Es necesario llamar al método begin() antes que cualquier otro comando de la biblioteca LCD. **var**: identificador de una variable de tipo ***LiquidCrystal***. **cols:** el número de columnas que tiene la pantalla. **filas:** el número de filas que tiene la pantalla. **charsize (opcional):** el número de puntos que tiene la pantalla por carácter: LCD_5x8DOTS para 5x8, LCD_5x10DOTS para 5x10. (predeterminado: LCD_5x8DOTS). **Ejemplo:** **LCD1.*begin***(16, 2);
var.clear()	Borra la pantalla LCD y coloca el cursor en la esquina superior izquierda. **var:** identificador de una variable de tipo ***LiquidCrystal***. **Ejemplo:** **LCD1.*clear***( );
var.setCursor(col, row)	Coloca el cursor LCD; es decir, establezca la ubicación en la que se mostrará el texto subsiguiente escrito en la pantalla LCD. **var**: una variable de tipo ***LiquidCrystal***. **col:** la columna en la que colocar el cursor (siendo 0 la primera columna) fila: la fila en la que colocar el cursor (siendo 0 la primera fila). **Ejemplo:** **LCD1.*setCursor***(3, 0);
var.print(data, BASE)	Imprime texto en la pantalla. **var**: Una variable del tipo ***LiquidCrystal***. **data:** Los datos a imprimir (char, byte, int, long o string). **BASE (opcional):** La base en la que imprimir números: BIN para binario (base 2), DEC para decimal (base 10), OCT para octal (base 8), HEX para hexadecimal (base 16). **Ejemplo:** **LCD1.*print***("UIE");

Para más información y funciones de la biblioteca visitar el sitio web:

https://www.arduino.cc/reference/en/libraries/liquidcrystal/

5 Ejercicio 01. Controlar el tiempo de parpadeo de un Led con un potenciómetro. (1 punto)

El objetivo de este ejercicio es controlar el tiempo de parpadeo de un LED conectado al **Pin 7** utilizando un potenciómetro.

El potenciómetro se conecta al pin analógico **A0** cuyo valor de voltaje de entrada se obtiene con la instrucción ***analogRead***(**A0**).

El valor retornado se utilizará como parámetro de la función ***delay*()** para generar un retardo variable en el encendido y apagado del **Led**.

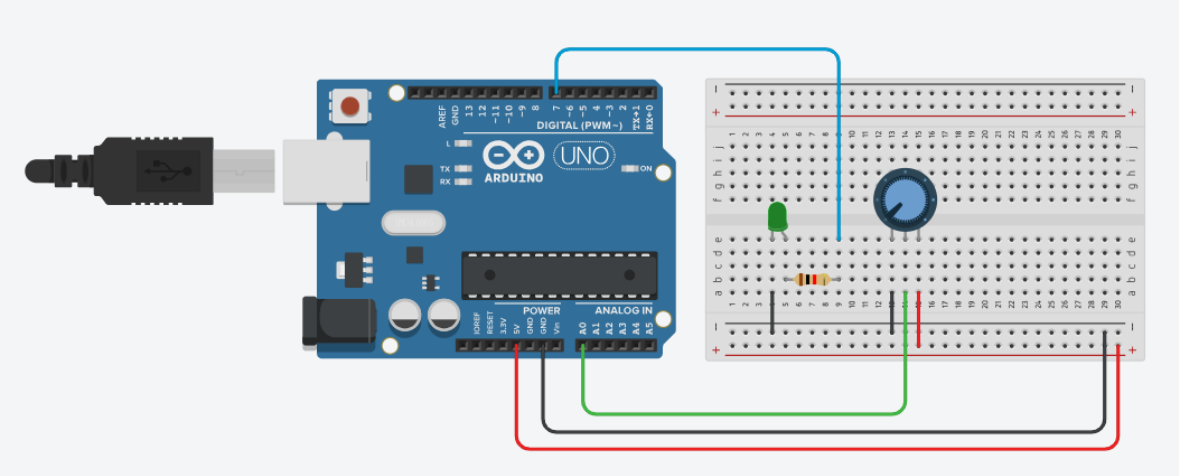
Además se utilizará la comunicación serial con el ordenador para mostrar en la terminal el valor asociado con la lectura del pin A0 conectado al potenciómetro.

a) Leer el valor de una entrada analógica.

b) Conocer el funcionamiento de un potenciómetro.

c) Conectar correctamente el potenciómetro.

5.1 Conectar los componentes como se muestra en el siguiente esquema:



- Realice las conexiones correspondientes según el esquema mostrado.

Nota: **Verifique** sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad.

5.2 Programa.

Abrir en su entorno de desarrollo (IDE Arduino) el programa **IoTP03E01.ino** descargado desde el material de la práctica para el ejercicio 01.

5.3 Prueba.

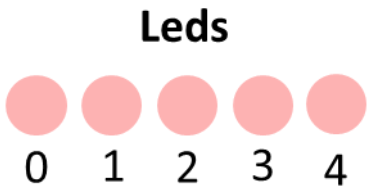
Cargar el programa en la placa Arduino para su ejecución.

5.4 Validar Ejercicio.

Mostrar al técnico del laboratorio el funcionamiento del modelo para el **registro en la plantilla de valoración** de la práctica**.**

6 Ejercicio 2. Encendido progresivo de Leds. (2 puntos)

El objetivo de este ejercicio es el encendido y apagado progresivo de una línea de 5 leds como se muestra en la figura.



Las entradas analógicas convierten un valor de voltaje de entrada en rango 0 y 5V, en el caso del Arduino Uno cuyo voltaje de operación es de 5 voltios, a un valor entero en el rango [0, 1023]

El potenciómetro conectado al pin analógico **A0** y cuyo valor de voltaje de entrada obtenido con la instrucción ***analogRead***(**A0**) y convertido a un valor entero en el rango [0,1024] será utilizado para determinar cuáles leds deben estar encendidos y cuáles no, en función de la siguiente tabla.

LED	Umbral Encendido ≥
0	170
1	340
2	510
3	680
4	850

Modelo Sistemas a Eventos Discretos.

Sea e_i el evento de encendido del Led i ocurre cuando se alcanza el umbral de encendido y sea a_i el evento de apagado ocurre cuando el umbral alcanza un valor menor al umbral de encendido.

Se desea construir un modelo y maqueta operativa que simule el comportamiento del sistema en función de la cantidad de leds encendidos.

Es decir, el **Estado del Sistema** es un valor entero en el rango [0 , 5].

Adicionalmente, se debe mostrar el estado actual del sistema en la terminal.

Definición del autómata

- Conjunto de Estados X = { 0, 1, 2, 3, 4, 5 } //Leds Encendidos
- Conjunto de eventos E = { e_0 , e_1 , e_2 , e_3 , e_4 , a_0 , a_1 , a_2 , a_3 , a_4 }
- Función de Transición de Estados.

  f(0, e_0) → 1

  f(1, e_1) → 2 ; f(1, a_0) → 0

  f(2, e_2) → 3 ; f(2, a_1) → 1

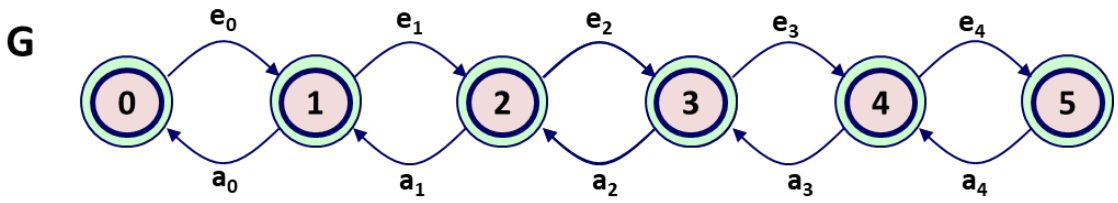
  f(3, e_3) → 4 ; f(3, a_2) → 2

  f(4, e_4) → 5 ; f(4, a_3) → 3

  f(5, a_4) → 4
- Estado Inicial

  X_0 Es desconocido. Cualquier estado puede ser el inicial. Al iniciar la ejecución el sistema modelado reconoce el estado inicial.
- Estados Marcados. { 0, 1, 2, 3, 4, 5 } (No hay un objetivo de finalización)

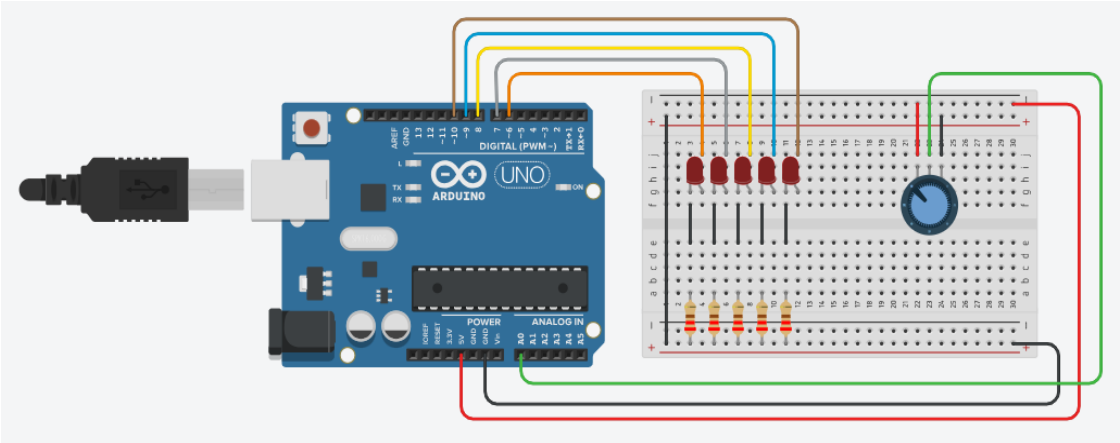
Autómata



El valor retornado por ***analogRead***(**A0**) será utilizado como parámetro de una rutina para determinar en cada momento el estado del sistema, es decir la función de transición de estado del autómata.

Además se muestra en la terminal el valor el potenciómetro.

6.1 Conectar los componentes como se muestra en el siguiente esquema:



- Realice las conexiones correspondientes según el esquema mostrado.

Nota: **Verifique** sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad.

6.2 Programa.

Abrir en su entorno de desarrollo (IDE Arduino) el programa **IoTP03E02.ino** descargado desde el material de la práctica 03.

6.3 Prueba.

Cargar el programa en la placa Arduino para su ejecución.

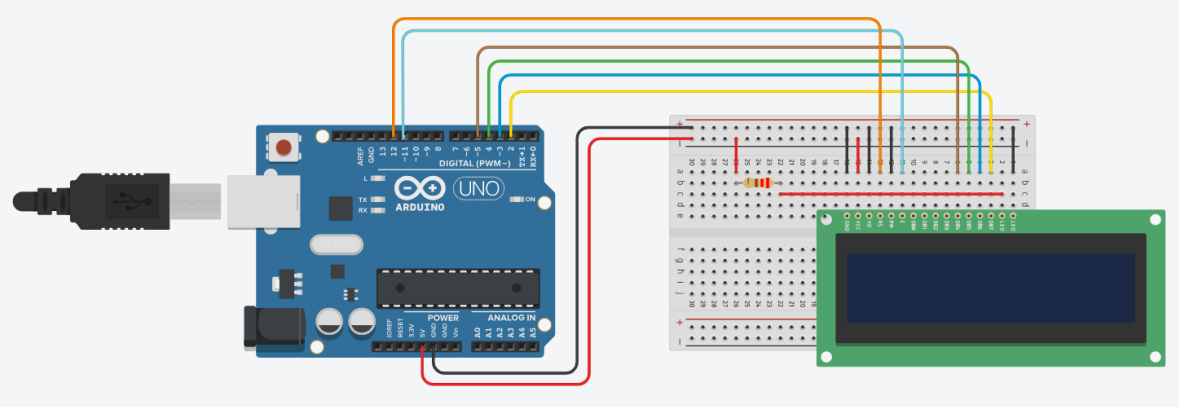
6.4 Validar Ejercicio.

Mostrar al técnico del laboratorio el funcionamiento del modelo para el **registro en la plantilla de valoración** de la práctica**.**

7 Ejercicio 03. Pantalla LCD 16x2. (2 puntos)

En el siguiente ejercicio se muestra el funcionamiento de una pantalla LCD 16x2.

7.1 Conectar los componentes y la placa Arduino como se muestra en el siguiente esquema:



- Realice las conexiones correspondientes según el esquema mostrado.

Nota: **Verifique** sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad práctica.

7.2 Programa.

Abrir en su entorno de desarrollo (IDE Arduino) el programa **IoTP03E03.ino** descargado desde el material de la práctica 03.

7.3 Prueba.

Cargar el programa en la placa Arduino para su ejecución.

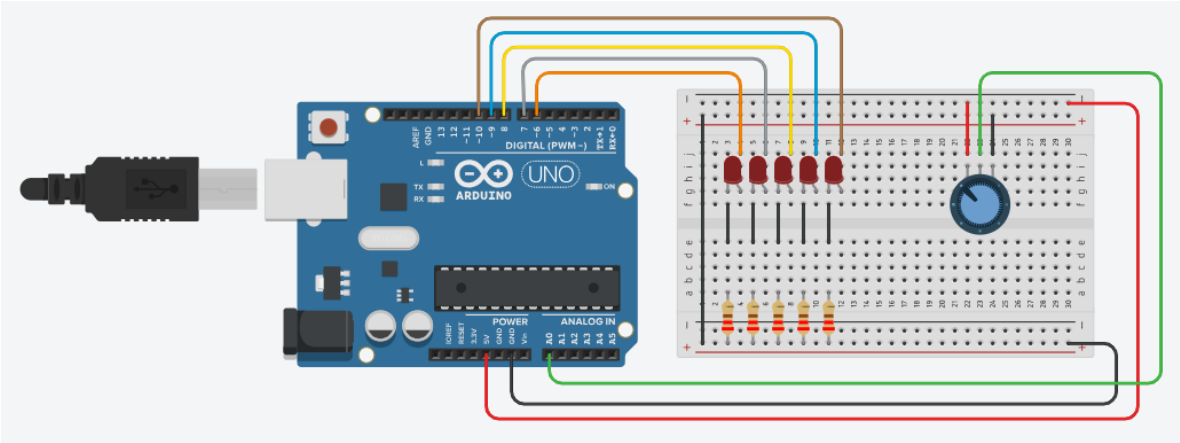
7.4 Validar Ejercicio.

Mostrar al técnico del laboratorio el funcionamiento del modelo para el **registro en la plantilla de valoración** de la práctica**.**

8 Actividad Práctica Autónoma. (5 puntos)

Objetivo.

- Mostrar el estado del Sistema del Ejercicio 02 en una pantalla LCD.



Instrucciones

- Añada al esquema del Ejercicio 02 de esta práctica una pantalla LCD realizando las conexiones de manera similar al esquema del Ejercicio 03.
- Realice las conexiones correspondientes al nuevo esquema.
- Modifique el programa del Ejercicio 02, incluyendo las instrucciones necesarias para mostrar en la pantalla LCD el estado del sistema y el valor de la entrada analógica.

Nota: Verifique sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad práctica.

8.1 Prueba.

Cargar el programa en la placa Arduino para su ejecución.

8.2 Validar Ejercicio.

Mostrar al técnico del laboratorio el funcionamiento del modelo para el **registro en la plantilla de valoración** de la práctica**.**