SENSORES Y ACTUADORES – TST 2024

Tp8-e4 #58. Salida a una pantalla LCD de 16x2 ó 20x4. de este Proyecto

- [x] 1. Configurar la pantalla LCD en el entorno de simulación.
- [x] 2. Implementar el código necesario para mostrar los valores de los sensores.
- [x] 3. Validar la correcta visualización en la pantalla.

Para agregar la salida a una pantalla LCD de 16x2 o 20x4 al proyecto de riego inteligente, es importante integrar la visualización de los datos de los sensores (humedad del suelo, lluvia) y el estado del sistema (bomba de agua). Utilizando una pantalla LCD con I2C es una forma conveniente de ahorrar pines y simplificar las conexiones.

1. Materiales Necesarios

- **Pantalla LCD**: 16x2 o 20x4 con interfaz I2C.
- **Módulo I2C**: Adaptador I2C para la pantalla LCD (es muy común y facilita la conexión).
- **Librería LiquidCrystal_I2C**: Para controlar la pantalla desde el ESP32 o Arduino.

2. Conexiones para la Pantalla LCD con I2C

- **SDA**: Conectar al pin SDA del microcontrolador.
  - En **Arduino Uno**: Pin A4.

  - En **ESP32**: Generalmente el pin 21, pero puede variar.
- **SCL**: Conectar al pin SCL del microcontrolador.
  - En **Arduino Uno**: Pin A5.
  - En **ESP32**: Generalmente el pin 22.
- **VCC**: Conectar a 5V (o 3.3V si es compatible).
- **GND**: Conectar a GND.

3. Código para Mostrar los Datos en la Pantalla LCD

Una implementación para una pantalla LCD de **16x2** (puedes adaptarlo para una de 20x4 simplemente ajustando el número de columnas y filas en el código).

Paso 1: Instalar la librería LiquidCrystal_I2C

- Si se estás usando el **IDE de Arduino**, ve a **Sketch → Incluir Librería → Administrar Bibliotecas** y busca LiquidCrystal I2C.
- Instálala.

Paso 2: Código de Ejemplo con Visualización en LCD

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Definición de la dirección I2C y el tamaño de la pantalla LCD (16x2 o 20x4)
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Cambiar a (0x27, 20, 4) para una pantalla de 20x4

// Definición de pines de sensores y actuadores
const int pinHumedadResistivo = A0; // Sensor de humedad resistivo
const int pinHumedadCapacitivo = A1; // Sensor de humedad capacitivo
const int pinLluvia = 2;             // Sensor de lluvia
const int pinBomba = 3;              // Relé de la bomba

void setup() {
  // Inicializa la pantalla LCD
  lcd.begin();
  lcd.backlight();  // Activa la retroiluminación de la pantalla LCD

  // Configura los pines de los sensores y la bomba
  pinMode(pinLluvia, INPUT);
  pinMode(pinBomba, OUTPUT);
  digitalWrite(pinBomba, LOW); // Apaga la bomba al inicio

  // Inicia la comunicación serie para depuración
  Serial.begin(9600);
}
```

```
void loop() {
  // Lectura de sensores
  int humedadResistivo = analogRead(pinHumedadResistivo); // Lee el sensor resistivo
  int humedadCapacitivo = analogRead(pinHumedadCapacitivo); // Lee el sensor capacitivo
  int lluvia = digitalRead(pinLluvia); // Lee el sensor de lluvia (0 = sin lluvia, 1 = llu

  // Mostrar datos en la pantalla LCD
  lcd.clear();  // Limpia la pantalla antes de mostrar nuevos datos

  // Primera línea: mostrar humedad resistiva
  lcd.setCursor(0, 0);  // Posiciona el cursor en la primera línea
  lcd.print("Hum. Res.: ");
  lcd.print(humedadResistivo);

  // Segunda línea: mostrar estado de lluvia
  lcd.setCursor(0, 1);  // Posiciona el cursor en la segunda línea
  lcd.print("Lluvia: ");
  if (lluvia == HIGH) {
    lcd.print("SI");
  } else {
    lcd.print("NO");
  }
```

///

….

…./////

```
  // Decidir si activar o desactivar la bomba
  if (lluvia == LOW && humedadResistivo < 400) {  // Sin lluvia y el suelo está seco
    digitalWrite(pinBomba, HIGH);  // Enciende la bomba
    lcd.setCursor(10, 1);  // Posiciona para mostrar el estado de la bomba
    lcd.print("Riego: ON");
  } else {
    digitalWrite(pinBomba, LOW);  // Apaga la bomba
    lcd.setCursor(10, 1);
    lcd.print("Riego: OFF");
  }

  delay(1000); // Espera un segundo antes de la próxima lectura
}
```

4. Descripción del Código

1. **Inicialización de la Pantalla LCD**: La pantalla LCD se inicializa con la dirección I2C 0x27 y el tamaño 16x2. Si utilizas una pantalla de 20x4, cambia el tamaño a LiquidCrystal_I2C lcd(0x27, 20, 4);.
2. **Lectura de Sensores**: Se leen los valores del sensor resistivo de humedad del suelo, el sensor capacitivo (aunque en este caso no lo estamos utilizando para la lógica, pero puedes adaptarlo) y el sensor de lluvia.
3. **Visualización en la LCD**:
   - La primera línea muestra el valor de la **humedad resistiva**.

   - La segunda línea muestra el estado de **lluvia** y el **estado de la bomba** (si el riego está activado o no).
4. **Lógica del Riego**:
   - Si no está lloviendo y la humedad del suelo es baja (menor que un valor umbral, como 400), el sistema activa la bomba de agua.
   - Si está lloviendo o la humedad es suficiente, la bomba se apaga.

5. Adaptación para Pantalla LCD de 20x4

Si se utiliza una pantalla LCD de 20x4, solo necesitas cambiar el tamaño de la pantalla en el código y puedes mostrar más información debido a su mayor tamaño.

Ejemplo para Pantalla 20x4:

```
LiquidCrystal_I2C lcd(0x27, 20, 4);  // Definir la pantalla de 20x4
```

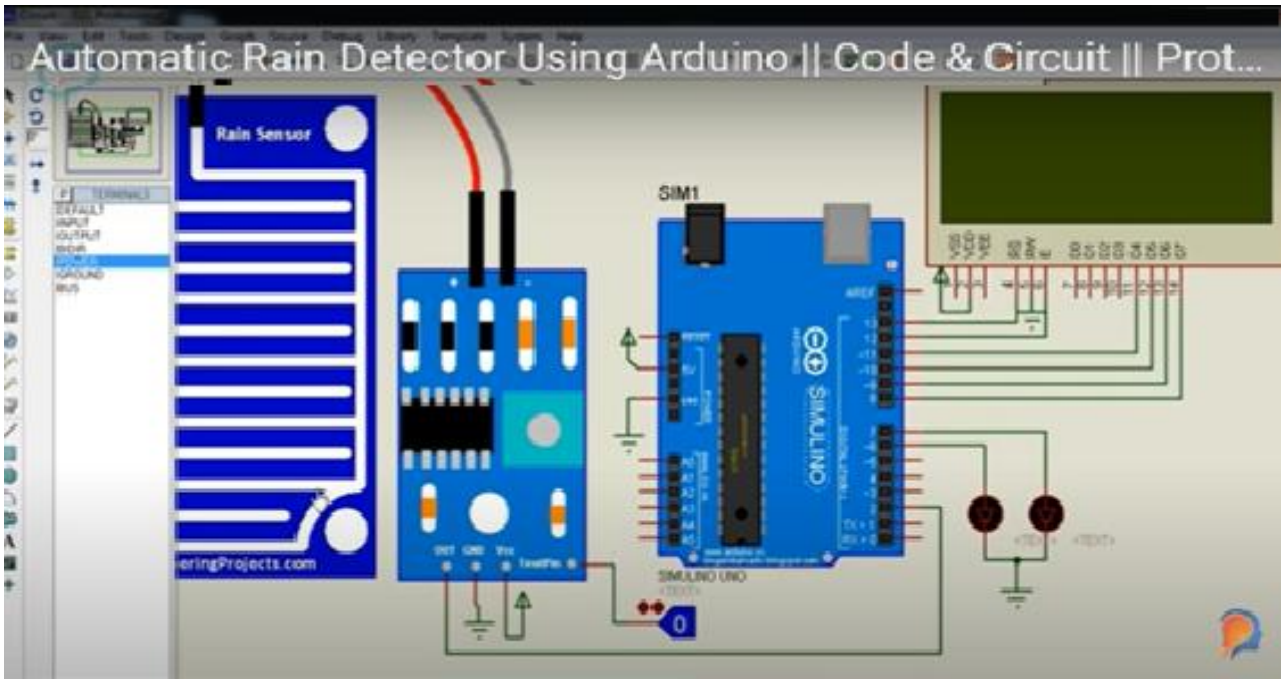
Se puede usar las cuatro líneas para mostrar información adicional:

```
lcd.setCursor(0, 0);  // Línea 1
lcd.print("Hum. Res.: ");
lcd.print(humedadResistivo);

lcd.setCursor(0, 1);  // Línea 2
lcd.print("Hum. Cap.: ");
lcd.print(humedadCapacitivo);

lcd.setCursor(0, 2);  // Línea 3
lcd.print("Lluvia: ");
lcd.print(lluvia == HIGH ? "SI" : "NO");

lcd.setCursor(0, 3);  // Línea 4
lcd.print("Riego: ");
lcd.print(digitalRead(pinBomba) == HIGH ? "ON" : "OFF");
```

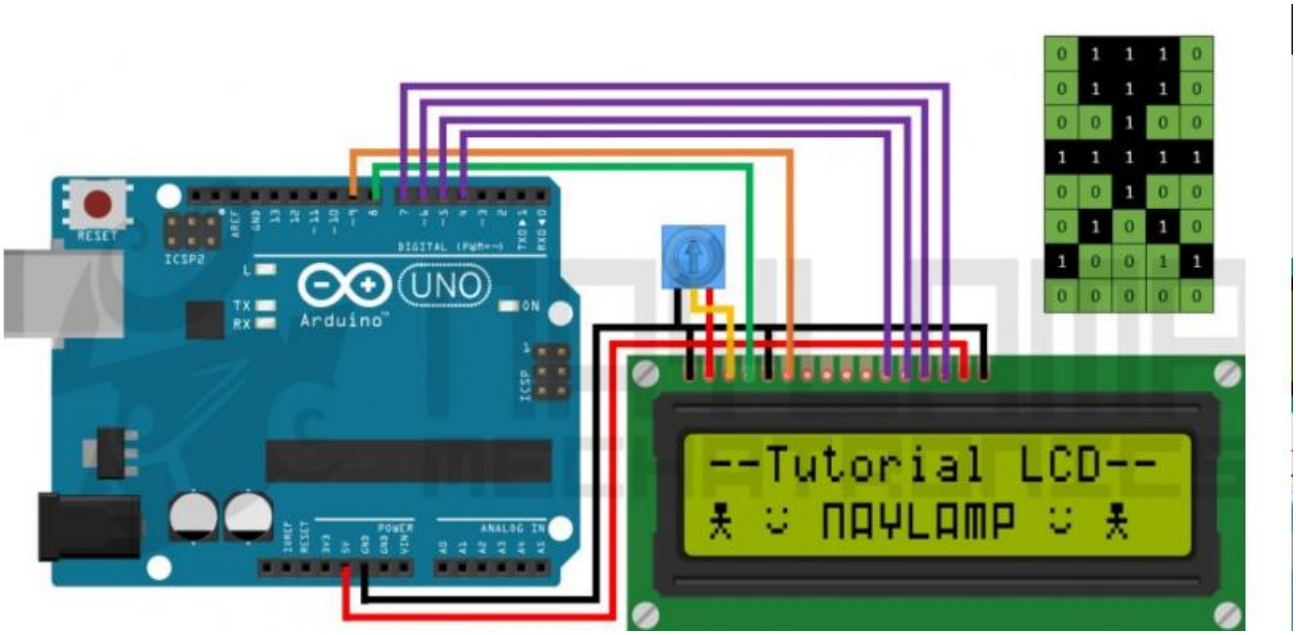
6. Pruebas y Calibración

1. **Umbral de Humedad**: Puedes ajustar el umbral de humedad del suelo (el valor 400 en el ejemplo). Este umbral se debe calibrar con base en las pruebas con el tipo de suelo donde se usará el sistema.
2. **Verificar Conexiones I2C**: Si la pantalla LCD no muestra información, asegúrate de que la dirección I2C es correcta (0x27 es la más común, pero algunas pantallas pueden tener 0x3F).

3. **Retroiluminación de la Pantalla**: Asegúrate de que la retroiluminación de la pantalla está activa con el comando lcd.backlight() para que sea visible en entornos con poca luz.



7. Tutorial LCD, Conectando Arduino a un LCD.



https://naylampmechatronics.com/blog/34_tutorial-lcd-conectando-tu-arduino-a-un-lcd1602-y-lcd2004.html

El funcionamiento de los LCD alfanuméricos, desarrolla varios ejemplos prácticos, para aprender a cómo usar los LCD en nuestros proyectos y como crear caracteres personalizados.

En muchos de nuestros proyectos se necesita visualizar o monitorear parámetros, si bien una solución es los display de 7 segmentos, pero solo se está limitados a valores numéricos e incluso si se desea poner varios dígitos a nivel de hardware aumenta nuestro diseño electrónico por lo que se necesita técnicas de multiplexado.
Los LCD alfanuméricos son la solución más práctica, para este problema, empezando por su bajo consumo, diferentes tamaños disponibles, y trabaja con caracteres alfanuméricos, por lo que se puede representar cualquier texto o variable.

Los LCD alfanuméricos más utilizados en el mercado son el LCD1602 y el LCD204 con tamaños de 16x2 y 20x4 caracteres respectivamente. Entre estos tamaños también hay diferentes modelos los cuales varían en color y sobretodo en la presencia o no de un Backlight (retro iluminador incorporado)

El LCD tiene una memoria interna en donde almacena todos sus caracteres alfanuméricos, los cuales podemos extender en ocho caracteres personalizados adicionales.

Los caracteres soportados por la mayoría de modelos de LCD son los siguientes:

Para la comunicación del LCD y un controlador utiliza un bus de 8 datos, pero se puede configurar para trabajar con un buz de 4 datos. También usa pines de control RS (chip select) RW (lectura/escritura) y E (enable).

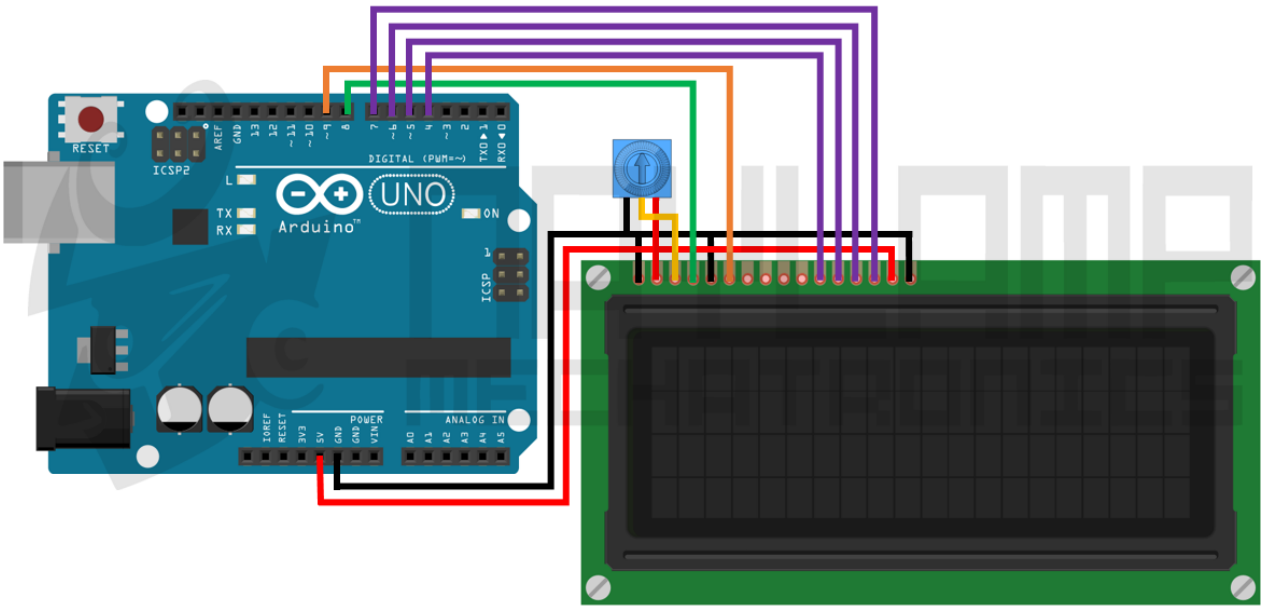
Para controlar el contrastes usa una entrada analógica VEE el cual por lo general se usa un potenciómetro para poder variar el contraste, y los LCD q traen Backlight tiene dos pines al final Led+ (A) y Led- (K) los cuáles se pueden alimentar directamente a 5V o a través de una resistencia si se tratase de un LED, variando su resistencia se pude variar la intensidad de luz.

Conexiones entre Arduino y display LCD1602

LCD1602 o LCD2004	Arduino Uno, Nano, Mega, etc.
1.VSS	GND
2. VDD	5V
3. VEE	Potenciómetro
4. RS	D8
5. RW	GND
6. EN	D9
11. D4	D4
12. D5	D5
13. D6	D6
14. D7	D7
15. Led+	VCC
16. Led-	GND

Conexiones entre Arduino y display LCD2004

Las conexiones para el LCD de 20 x 4 son las mismas:



Librería LiquidCrystal de Arduino

El IDE de Arduino ya viene con una librería que nos permite manejar diferentes tamaños de LCD's, La documentación completa la pueden encontrar en la página oficial de Arduino: LiquidCrystal

Explicaremos las funciones principales, las cuales se usaran en este tutorial.

LiquidCrystal(rs, en, d4, d5, d6, d7)
Función constructor, crea una variable de la clase LiquidCrystal, con los pines indicados.

begin(cols, rows)
Inicializa el LCD, es necesario especificar el número de columnas (cols) y filas (rows) del LCD.

clear()
Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda (posición (0,0)).

setCursor(col, row)
Posiciona el cursor del LCD en la posición indicada por col y row (x,y); es decir, establecer la ubicación en la que se mostrará posteriormente texto escrito para la pantalla LCD.

write()
Escribir un carácter en la pantalla LCD, en la ubicación actual del cursor.

print()
Escribe un texto o mensaje en el LCD, su uso es similar a un Serial.print

scrollDisplayLeft()
Se desplaza el contenido de la pantalla (texto y el cursor) un espacio hacia la izquierda.

scrollDisplayRight()
Se desplaza el contenido de la pantalla (texto y el cursor) un espacio a la derecha.

createChar (num, datos)
Crea un carácter personalizado para su uso en la pantalla LCD. Se admiten hasta ocho caracteres de 5x8 píxeles (numeradas del 0 al 7). Donde: **num** es el número de carácter y **datos** es una matriz que contienen los pixeles del carácter. Se verá un ejemplo de esto más adelante.

Explicado la librería veamos unos ejemplos:

Un Hola mundo con Arduino y LCD7

Se muestra un texto y un valor numérico en el LCD, para esto se debe cargar el siguiente sketch:

```
 #include <LiquidCrystal.h>

//Crear el objeto LCD con los números correspondientes (rs, en, d4, d5, d6, d7)
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

void setup() {
  // Inicializar el LCD con el número de  columnas y filas del LCD
  lcd.begin(16, 2);
  // Escribimos el Mensaje en el LCD.
  lcd.print("Hola Mundo");
}

void loop() {
   // Ubicamos el cursor en la primera posición(columna:0) de la segunda línea(fila:1)
  lcd.setCursor(0, 1);
   // Escribimos el número de segundos trascurridos
  lcd.print(millis()/1000);
  lcd.print(" Segundos");
  delay(100);
}
```

Como se observa con la función printf() escribimos el texto, y con setCursor(x,y) indicamos la poción en donde deseamos que se escriba el texto.Después de cargar, en su LCD deben obtener el siguiente resultado

Para el caso del LCD de 20x4 tienen que especificar este tamaño cundo inicializan el LCD.

```
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
void setup() {
  // Inicializar el LCD con el número de  columnas y filas del LCD
  lcd.begin(20,4);
  // Escribimos el Mensaje en el LCD
  lcd.print("Hola Mundo");
  lcd.setCursor(0,2);
  lcd.print("NAYLAMP MECHATRONICS");
  lcd.setCursor(0,3);
  lcd.print("      PERU  ");
```

```
}
void loop() {
  // Ubicamos el cursor en la primera posición (columna:0) de la segunda línea(fila:1)
  lcd.setCursor(0, 1);
  // Escribimos el número de segundos trascurridos
  lcd.print(millis()/1000);
  lcd.print(" segundos");
  delay(100);
}
```

Desplazando el texto en el LCD. En este ejemplo para desplazar el texto se usa la función scrollDisplayLeft();

```
 #include <LiquidCrystal.h>



//Crear el objeto LCD con los números correspondientes (rs, en, d4, d5, d6, d7)
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
void setup() {
  // Inicializar el LCD con el número de  columnas y filas del LCD
  lcd.begin(16, 2);
  // Escribimos el Mensaje en el LCD en una posición  central.
  lcd.setCursor(10, 0);
  lcd.print("WWW.NAYLAMPMECHATRONICS.COM");
  lcd.setCursor(5, 1);
  lcd.print("Tutorial LCD, Test de desplazamiento  ");
}
void loop() {
  //desplazamos una posición a la izquierda
  lcd.scrollDisplayLeft();
  delay(250);
}
```

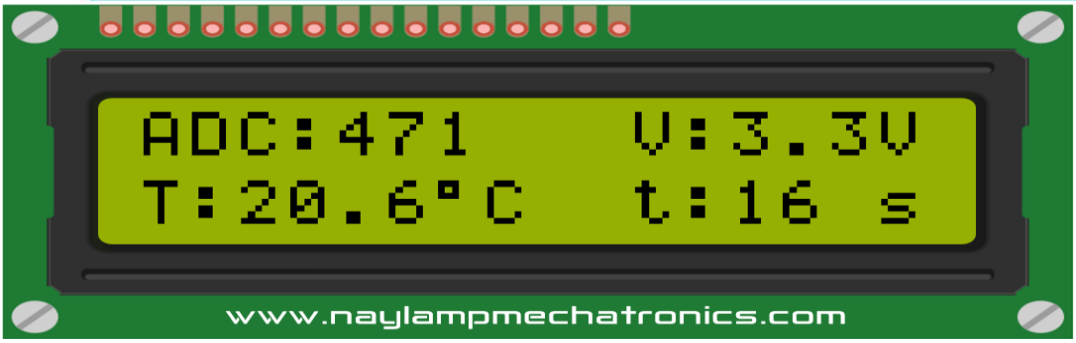
Como se observa en el código, inicialmente se escribe el texto, luego se desplaza el texto una posición por cada ciclo con una pausa de 250ms , tiempo que si se varía aumentará o disminuirá la velocidad de desplazamiento, se nota que el texto que se escribe es mayor a los 16 caracteres, pero igual el texto no se pierde, esto es porque el espacio de trabajo por cada fila en realidad es de 40 caracteres, y el LCD solo muestra los 16 primeros caracteres, pero al desplazarlo se logra ver los demás caracteres.

Mostrando datos de sensores o variables en el LCD

En este ejemplo mostramos en el LCD variables, que pueden representar valores de sensores u otros datos. Para simular los sensores se usa **potenciómetros conectados a los pines analógicos**.

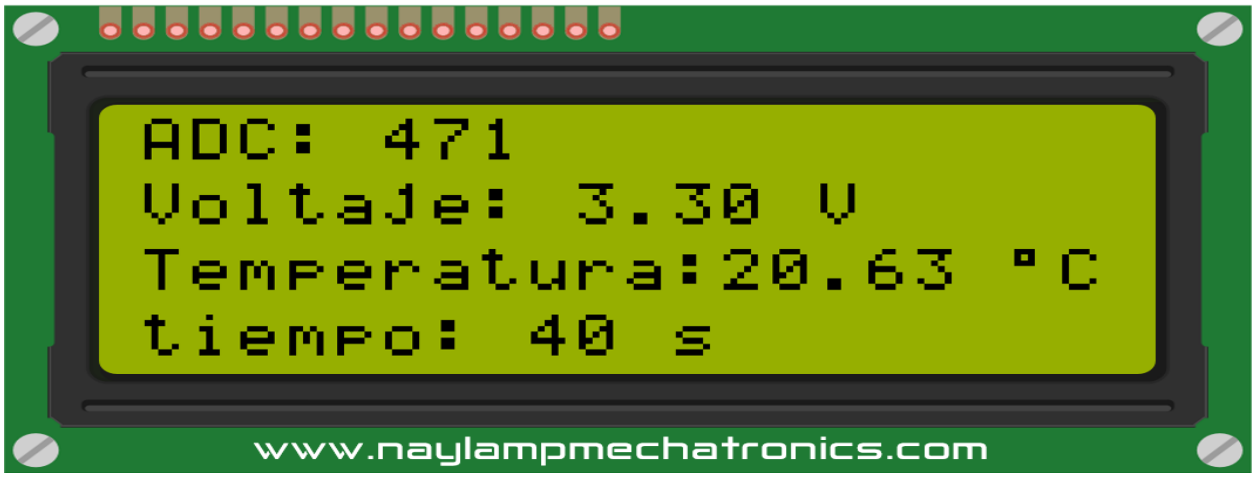
```
 #include <LiquidCrystal.h>
//Crear el objeto LCD con los números correspondientes (rs, en, d4, d5, d6, d7)
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
void setup() {
  // Inicializar el LCD con el número de  columnas y filas del LCD
  lcd.begin(16,2);
}
void loop() {
  int sen1=analogRead(A0);
  float sen2=analogRead(A1)*(5.0 / 1023.0);
  float sen3=analogRead(A2)*(100.0 / 1023.0);
  int tiempo=millis()/1000;
  // Cursor en la primera posición de la primera fila
  lcd.setCursor(0,0);
  lcd.print("ADC:");
  lcd.print(sen1);
  lcd.print("   ");
  // Cursor en la 11° posición de la primera fila
  lcd.setCursor(10,0);
  lcd.print("V:");
  lcd.print(sen2,1);//1 decimal
  lcd.print("V  ");
  // Cursor en la primera posición de la 2° fila
  lcd.setCursor(0,1);
  lcd.print("T:");
  lcd.print(sen3,1); //1 decimal
  lcd.print("337C  "); // "337" -> "°"
  // Cursor en la 11° posición de la 2° fila
  lcd.setCursor(10,1);
  lcd.print("t:");
  lcd.print(tiempo);
  lcd.print(" s  ");
    delay(200);
  }
```

Su resultado debe ser el siguiente.



De igual manera se trabaja con el LCD de 20x4, con la ventaja de tener más espacio para mostrar los valores.

```
#include <LiquidCrystal.h>
//Crear el objeto LCD con los números correspondientes (rs, en, d4, d5, d6, d7)
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

void setup() {
  // Inicializar el LCD con el número de  columnas y filas del LCD
  lcd.begin(20,4);
}
void loop() {
  int sen1=analogRead(A0);
  float sen2=analogRead(A1)*(5.0 / 1023.0);
  float sen3=analogRead(A2)*(100.0 / 1023.0);
  int tiempo=millis()/1000;
  // Cursor en la primera posición de la primera fila
  lcd.setCursor(0,0);
  lcd.print("ADC: ");
  lcd.print(sen1);
  lcd.print("   ");
  // Cursor en la 1° posición de la 2° fila
  lcd.setCursor(0,1);
  lcd.print("Voltaje: ");
  lcd.print(sen2);
  lcd.print(" V  ");
  // Cursor en la primera posición de la 3° fila
  lcd.setCursor(0,2);
  lcd.print("Temperatura:");
  lcd.print(sen3); //1 decimal
  lcd.print(" 337C  "); // "337" -> "°"
  // Cursor en la 1° posición de la 4° fila
  lcd.setCursor(0,3);
  lcd.print("tiempo: ");
  lcd.print(tiempo);
  lcd.print(" s  ");

  delay(200);

}
```



Agregando nuevos caracteres a nuestro LCD

En algunos casos el LCD no tiene los caracteres que deseamos, o necesitamos dibujar caracteres personalizados, en este caso usamos la función **createChar ()** pero antes expliquemos como está constituido un carácter: Un carácter está formado por 5x8 pixeles los cuales se representan por 8 bytes, uno para cada fila, los 5 bits menos

8. Conclusión

Este código te permitirá implementar la salida de información a una pantalla LCD de 16x2 o 20x4 en tu Sistema de Riego Inteligente. Recuerda que puedes ajustar la disposición de los datos mostrados según el tamaño de tu pantalla, y que es importante calibrar los sensores para obtener resultados óptimos en el monitoreo del suelo y el control del riego.

BIBLIOGRAFIA

- https://www.youtube.com/watch?v=cqJ5fXaShP0
- https://www.youtube.com/watch?v=6ODn1XrysGw
- https://www.youtube.com/watch?v=mH_h1LS9wMo
- https://www.youtube.com/watch?v=Wx1Vi0EPhQU
- https://naylampmechatronics.com/blog/34_tutorial-lcd-conectando-tu-arduino-a-un-lcd1602-y-lcd2004.html