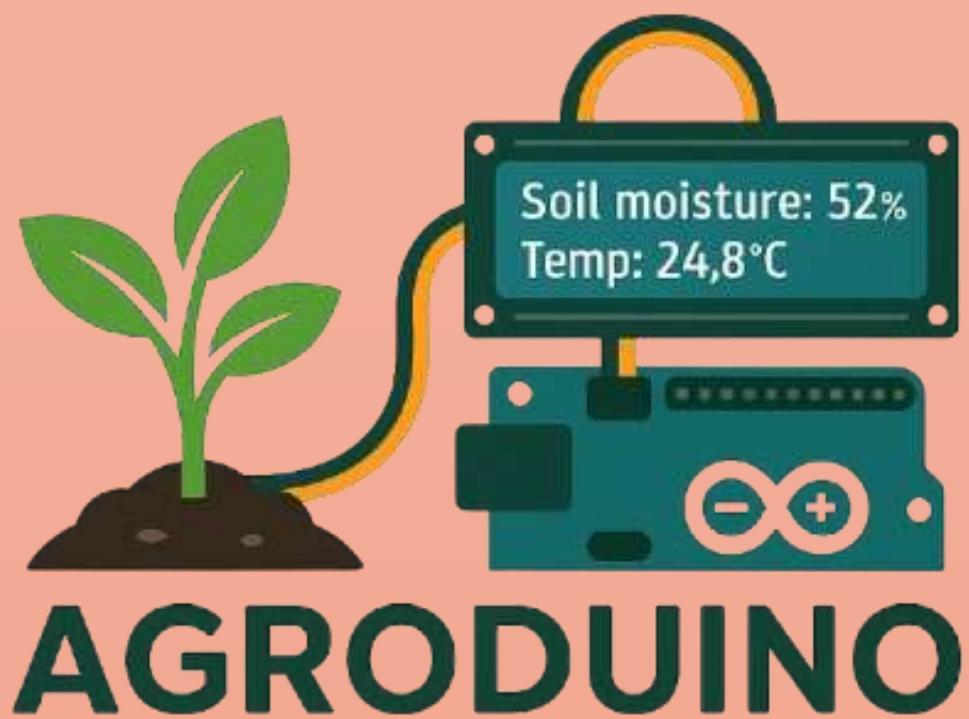


MANUAL DE USUARIO



**Sistema de monitoreo en tiempo real con
Arduino y visualización web interactiva**

CONTENIDO

Introducción	2
 Objetivo	2
 Requerimientos	2
Opciones del sistema	3
❖ Detección automática del puerto Arduino	3
❖ Lectura en tiempo real del sensor de humedad del suelo.....	3
❖ Reconexión automática	4
❖ Visualización de humedad del suelo (%)	4
❖ Visualización del porcentaje de agua	4
❖ Visualización de la temperatura ambiental.....	5
❖ Generación de alertas inteligentes	5
❖ Historial de datos	6
❖ Gráfica de barras de tendencia	7
❖ Estimación del tiempo de riego.....	8
❖ Indicador de lecturas óptimas	8
❖ Interfaz web responsiva	9
❖ Actualización automática.....	9
Conclusión	14
Recomendaciones finales	14
Autoría	14

Introducción

Objetivo

Crear una aplicación interactiva que facilite el seguimiento en tiempo real de la humedad del suelo mediante un sensor vinculado a un Arduino, con una presentación de los datos en un panel web dinámico que incorpora indicadores visuales, alertas automáticas, un registro de lecturas pasadas y análisis gráficos, para mejorar la toma de decisiones sobre riego en contextos agrícolas.

Requerimientos

- ❖ Hardware
 - Computadora con sistema operativo Windows 10 o superior.
 - Placa Arduino UNO, Nano o compatible con puerto USB.
 - Sensor de humedad del suelo (ej. YL-69, FC-28 o capacitivo).
 - Cable USB para conexión entre el Arduino y la computadora.
 - Fuente de alimentación (opcional, si se desea uso autónomo del Arduino).
 - Resistencias o módulos de protección (si se requiere según el tipo de sensor).
- ❖ Software
 - Python 3.8 o superior instalado en el sistema.
 - Dependencias de Python instaladas mediante pip:

```
pip install pyserial dash dash-bootstrap-components plotly numpy
```

- IDE o editor de texto (recomendado: VS Code, PyCharm o similar).
- Arduino IDE (opcional, para subir el código al microcontrolador si se modifica).
- Drivers USB para Arduino (ej. CH340, si se usa una placa clon).

Opciones del sistema

❖ Detección automática del puerto Arduino

El sistema identifica automáticamente el puerto al que está conectado el Arduino.

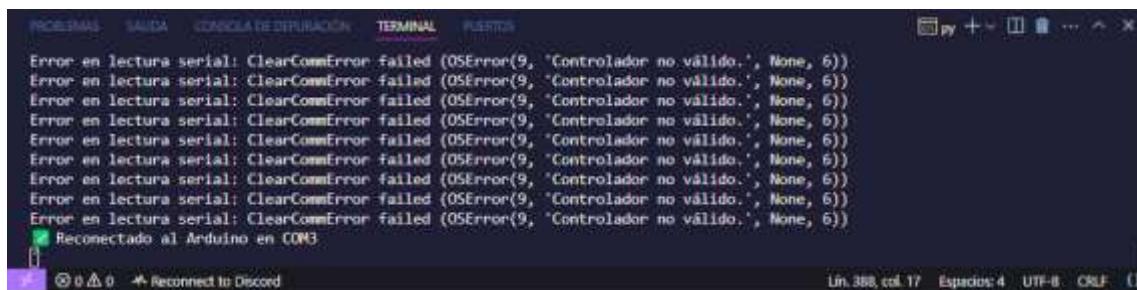


❖ Lectura en tiempo real del sensor de humedad del suelo

Recibe datos constantemente desde el sensor a través del Arduino.

❖ Reconexión automática

Si el Arduino se desconecta, el sistema intenta reconectarse de forma automática.



```
PROBLEMAS SALIDA CONSOLA DE DIBUJO TERMINAL PUERTOS
```

Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Error en lectura serial: ClearCommError failed (OSError(9, "Controlador no válido.", None, 6))
Reconectado al Arduino en COM3

Lin. 388, col. 17 Espacio: 4 UTF-8 ORF: 0

❖ Visualización de humedad del suelo (%)

Mediante un medidor gráfico tipo gauge actualizado en tiempo real.



❖ Visualización del porcentaje de agua

Segundo medidor que refleja el mismo valor de humedad como una aproximación al nivel hídrico.



❖ Visualización de la temperatura ambiental

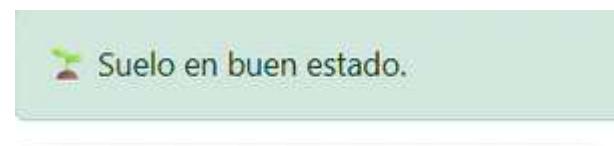
El sistema integra un sensor DHT11 conectado al pin A1 del Arduino, el cual mide la temperatura ambiente en grados Celsius. Esta información es leída constantemente y enviada al monitor serial, además de ser mostrada en la pantalla LCD junto al nivel de humedad.



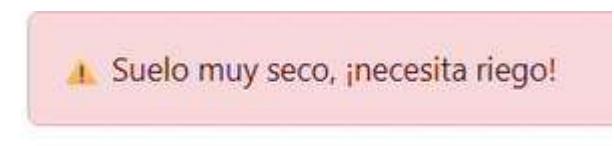
❖ Generación de alertas inteligentes

El sistema muestra notificaciones visuales según el nivel de humedad:

-  Estado óptimo



-  Suelo seco (riego necesario)



-  Suelo húmedo (no se requiere riego)



➤ ⚠ Arduino desconectado

⚠ Arduino no está conectado. Conéctalo para ver datos.

➤ Notificaciones de Temperatura

El sistema monitorea la temperatura ambiente y muestra mensajes que indican el estado térmico ideal para las plantas

-  Temperatura óptima.
-  Temperatura demasiado alta, riesgo para las plantas.
-  Temperatura demasiado baja, riesgo para las plantas.

❖ Historial de datos

Tabla interactiva con los últimos valores registrados de humedad y agua, junto con la hora de cada medición.

Hora	Humedad (%)	Temperatura (°C)
17:53:24	92	29.3
17:53:26	91	29.3
17:53:28	91	29.2
17:53:31	91	29.4
17:53:33	91	29.6
17:53:35	91	29.8
17:53:37	49	29.9
17:53:39	4	29.9
17:53:41	4	30

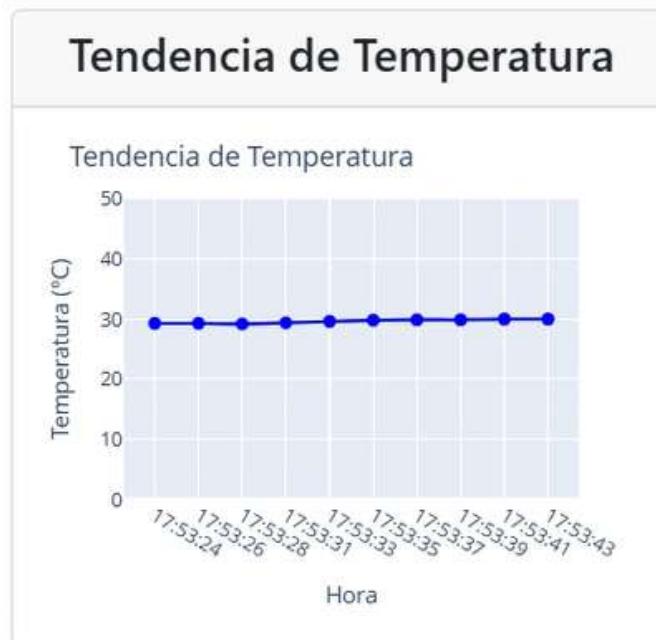
❖ Gráfica de barras de tendencia

Muestra cómo ha variado la humedad en las últimas lecturas (hasta 10), con colores según el estado.



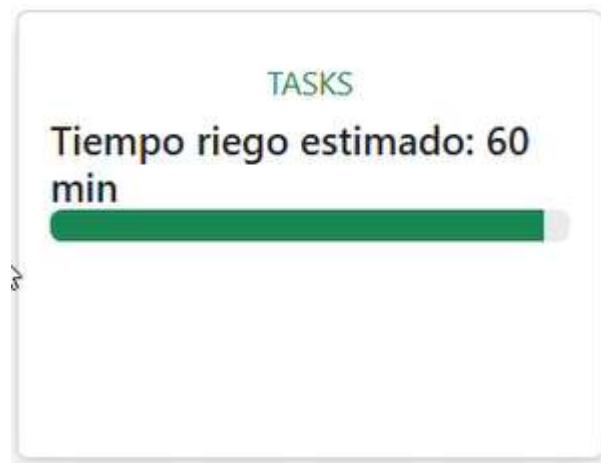
❖ Tendencia de la temperatura

el valor de temperatura puede representarse gráficamente como una curva o gráfico de líneas (si se integra). Aunque el código que compartiste es del lado de Arduino, puedes modificar el sistema en Dash para visualizar una gráfica de tendencia similar a la de humedad, basada en los datos que llegan por puerto serial.



❖ Estimación del tiempo de riego

Calcula en minutos el tiempo estimado necesario para llevar la humedad a niveles adecuados.



❖ Indicador de lecturas óptimas

Visualiza cuántas mediciones recientes estuvieron dentro del rango ideal de humedad (30% - 80%).

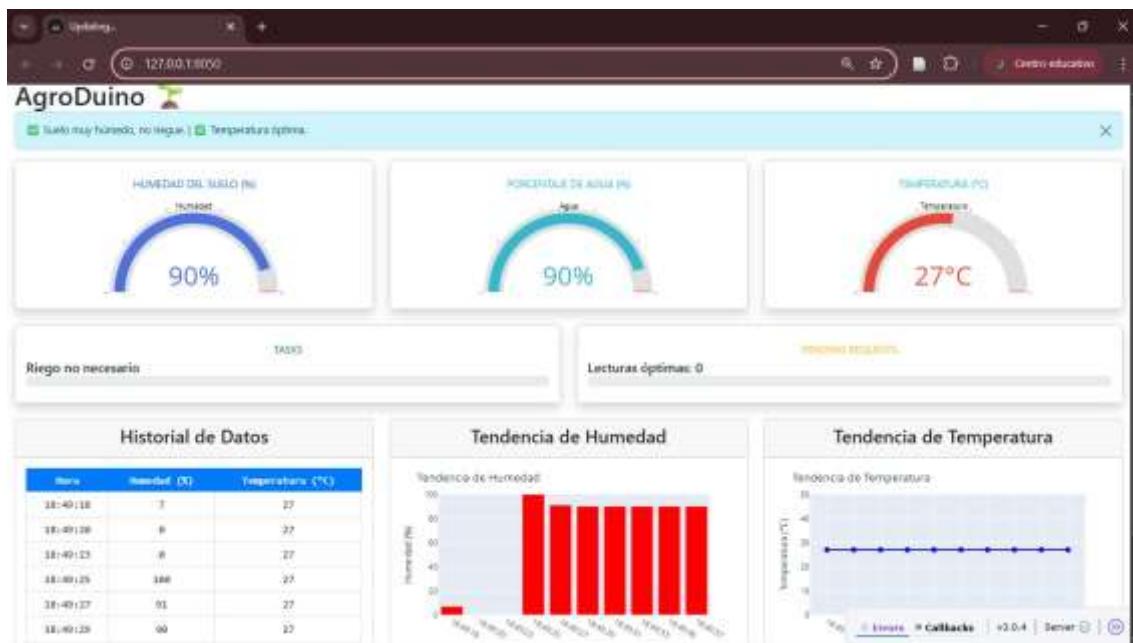


❖ **Interfaz web responsiva**

Compatible con navegadores modernos gracias a Dash y Bootstrap.

❖ **Actualización automática**

Todos los datos se actualizan sin necesidad de recargar la página, usando `dcc.interval`

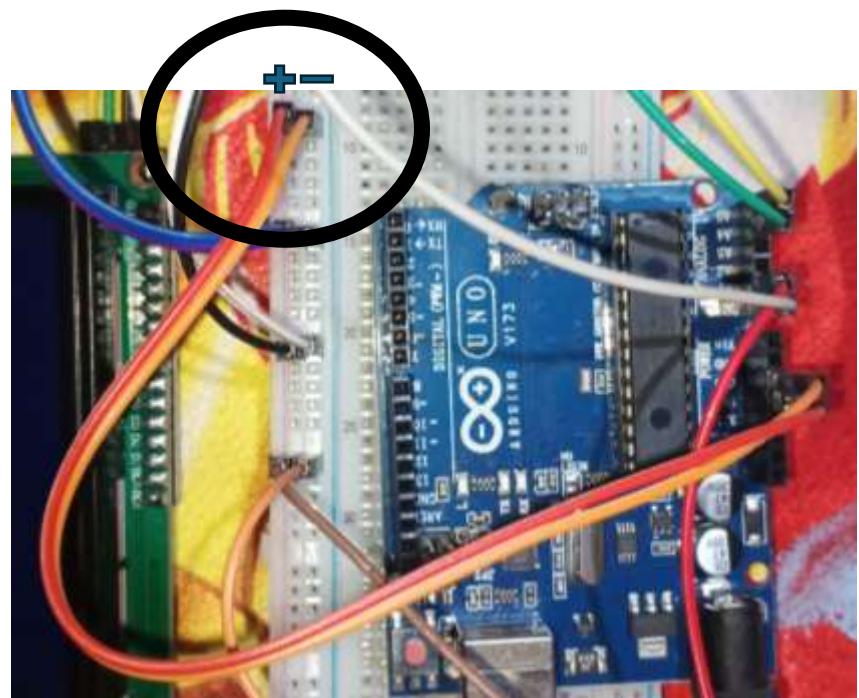


Guía de conexión del hardware

Seguidamente se enumeran los pasos para llevar a cabo de forma correcta las conexiones del sistema AgroDuino, garantizando así el correcto funcionamiento de los sensores y la pantalla LCD:

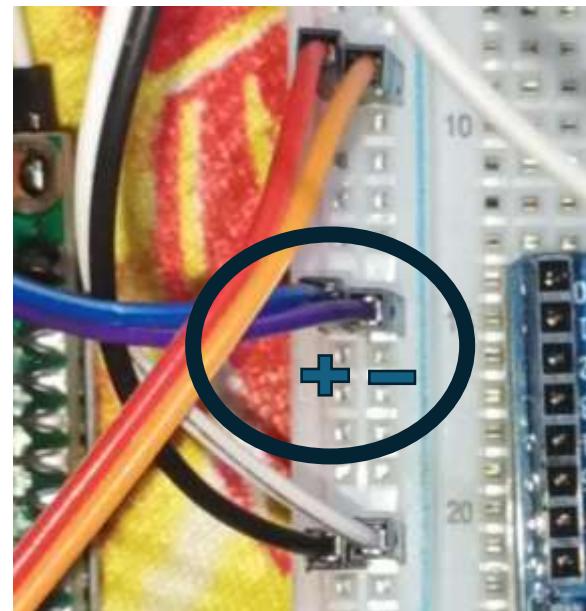
Conexión de alimentación general (Protoboard):

- ❖ Conectar el pin 5V del Arduino a la línea positiva (+) del protoboard.
- ❖ Conectar el pin GND del Arduino a la línea negativa (-) del protoboard.

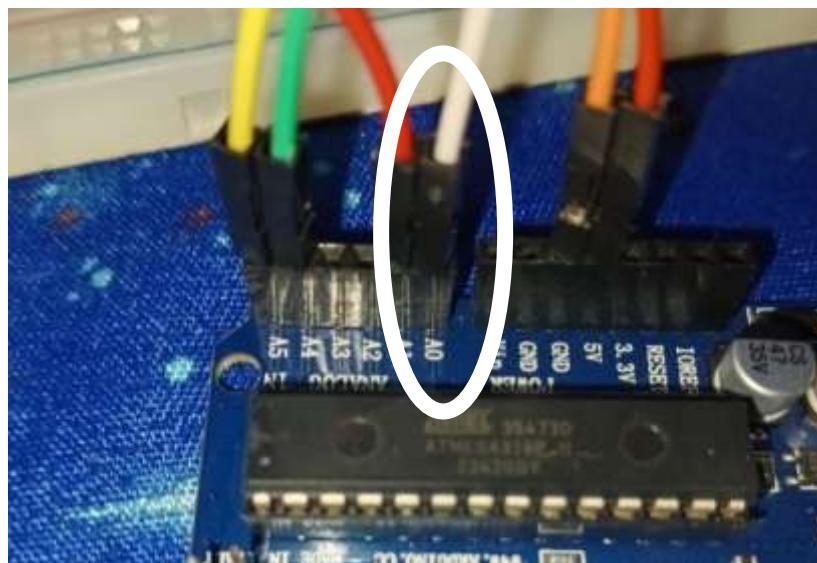


Conexión del sensor de humedad del suelo:

- ❖ VCC del módulo → a la línea positiva (+) del protoboard.
- ❖ GND del módulo → a la línea negativa (-) del protoboard.

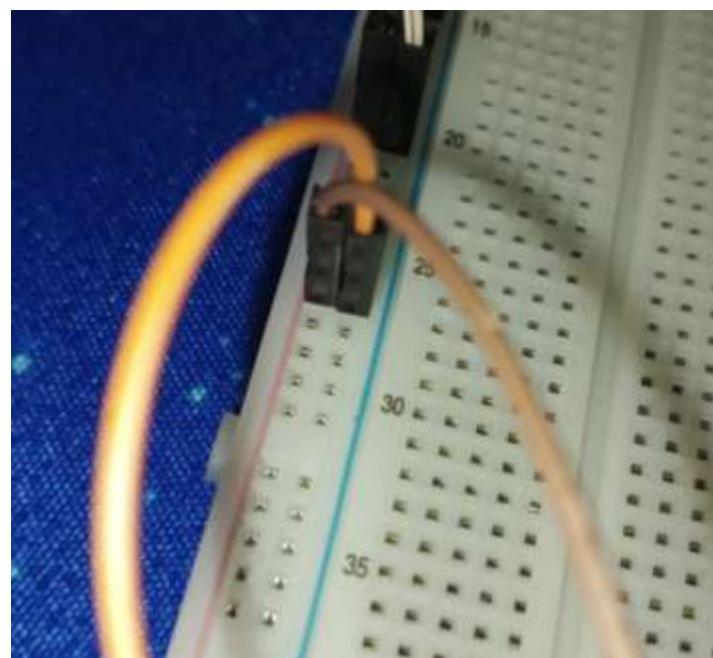


- ❖ A0 del módulo → al pin A0 del Arduino.

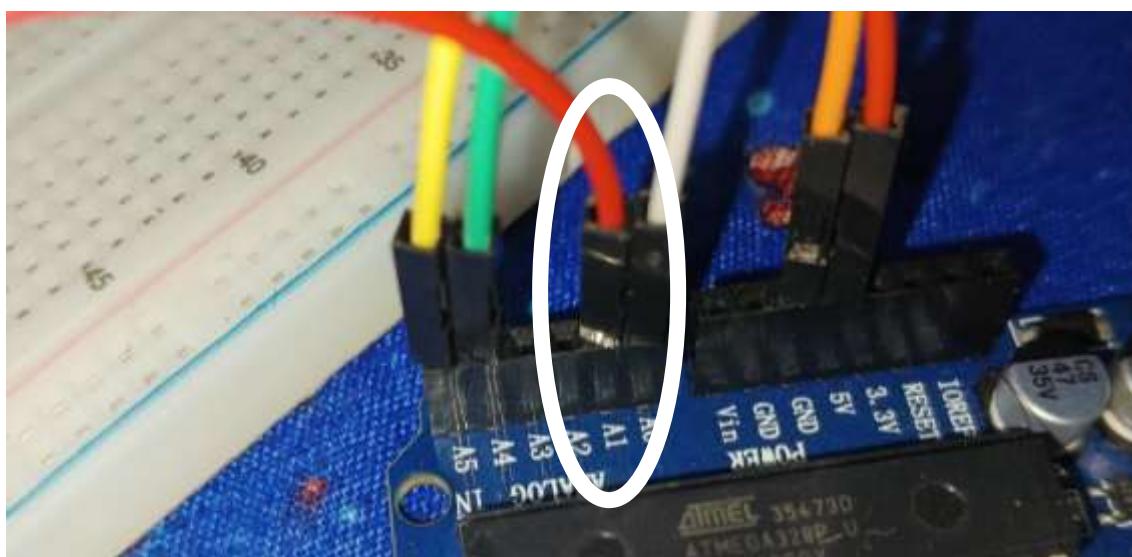


Conexión del sensor de temperatura:

- ❖ Pin + (alimentación) → a la línea positiva (+) del protoboard.
- ❖ Pin - (GND) → a la línea negativa (-) del protoboard.

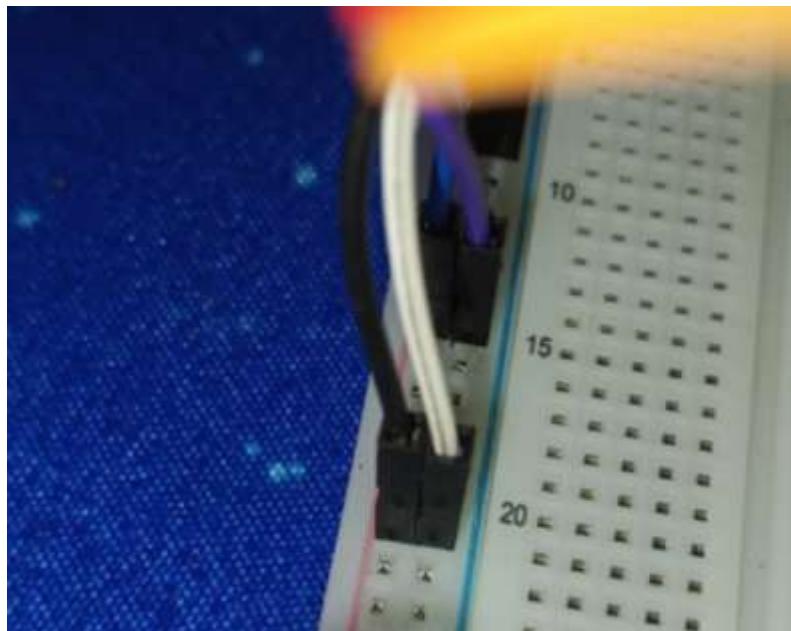


- ❖ Pin OUT (señal) → al pin A1 del Arduino.

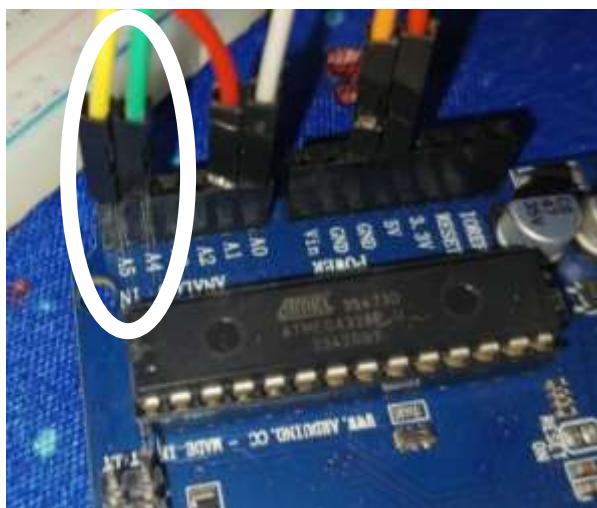


Conexión de la pantalla LCD (con interfaz I2C):

- ❖ VCC de la pantalla LCD → a la línea positiva (+) del protoboard.
- ❖ GND de la pantalla LCD → a la línea negativa (-) del protoboard.



- ❖ SDA (datos) → al pin A4 del Arduino.
- ❖ SCL (reloj) → al pin A5 del Arduino.



Conexión del sistema al computador:

- ❖ Una vez completadas todas las conexiones anteriores, conectar el Arduino a la computadora mediante su cable USB.
- ❖ Abrir el Arduino IDE para subir el programa correspondiente.

Conclusión

AgroDuino es un sistema que ofrece una solución eficaz y asequible para el control en tiempo real de la humedad del suelo, facilitando así una mejor gestión del riego en contextos agrícolas o experimentales. Mediante la integración de hardware (Arduino y sensor de humedad) y software (dashboard web con Python), el usuario tiene la capacidad de ver datos actualizados, recibir alertas automáticas, analizar tendencias y tomar decisiones informadas. Contribuye a una agricultura más inteligente al ahorrar agua y optimizar recursos.

Recomendaciones finales

- ❖ Verificar siempre que el sensor de humedad esté correctamente conectado y calibrado para obtener datos precisos.
- ❖ Mantener el Arduino en un entorno seco y protegido para evitar daños por humedad.
- ❖ Evitar exponer el sensor por largos períodos si no se encuentra en uso, para prolongar su vida útil.
- ❖ Ejecutar el sistema en un entorno virtual de Python para mantener aisladas las dependencias.
- ❖ Revisar periódicamente las conexiones físicas del sistema para evitar lecturas erróneas.
- ❖ Si se planea escalar el proyecto, considerar la integración de otros sensores (temperatura, luz, pH, etc.) y el almacenamiento de datos en la nube para un análisis más profundo.

Autoría

- ❖ **Nombre del proyecto:** AGRODUINO
- ❖ **Desarrollado por:**
 - Adán Alí Escandón Roca
 - Abraham Natanael Pérez Chávez
 - José Andrés Gómez molina
- ❖ **Lenguaje y Framework:** Python 3 + Dash(Desarrollado por Plotly para aplicaciones web interactivas)
- ❖ **Versión:** 1.0
- ❖ **Carrera:** ingeniería en software
- ❖ **Tutor del proyecto:** Mtr. Fabricio Morales
- ❖ **Universidad:** Universidad Estatal De Milagro
- ❖ **Fecha de entrega:** 03 de julio del 2025