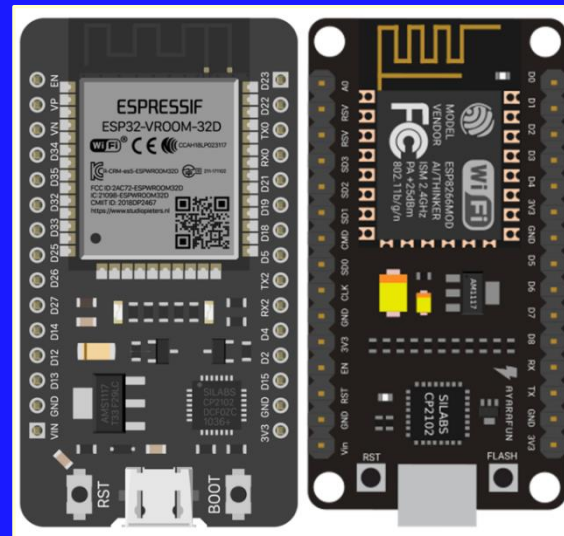


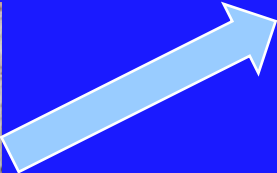
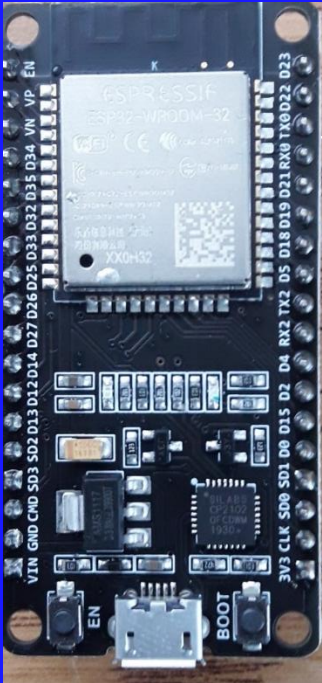
# Aplicaciones de sistemas embebidos ESP32 e implementación de un módulo de medición de Ph basado en NodeMCU



Parte II – Aplicaciones - Ing. Rafael Oliva - Ing. Néstor Cortez  
Instituto de Tecnología Aplicada (ITA-UARG)

JORNADAS DE INFORMÁTICA 2024 – UNPA /UARG  
12 de Noviembre de 2024

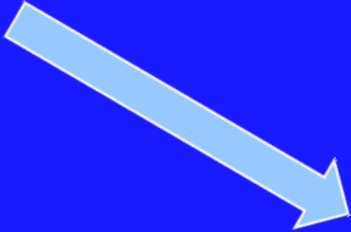
NodeMCU ESP32



Poncho PLC EDU CIA



Notebook



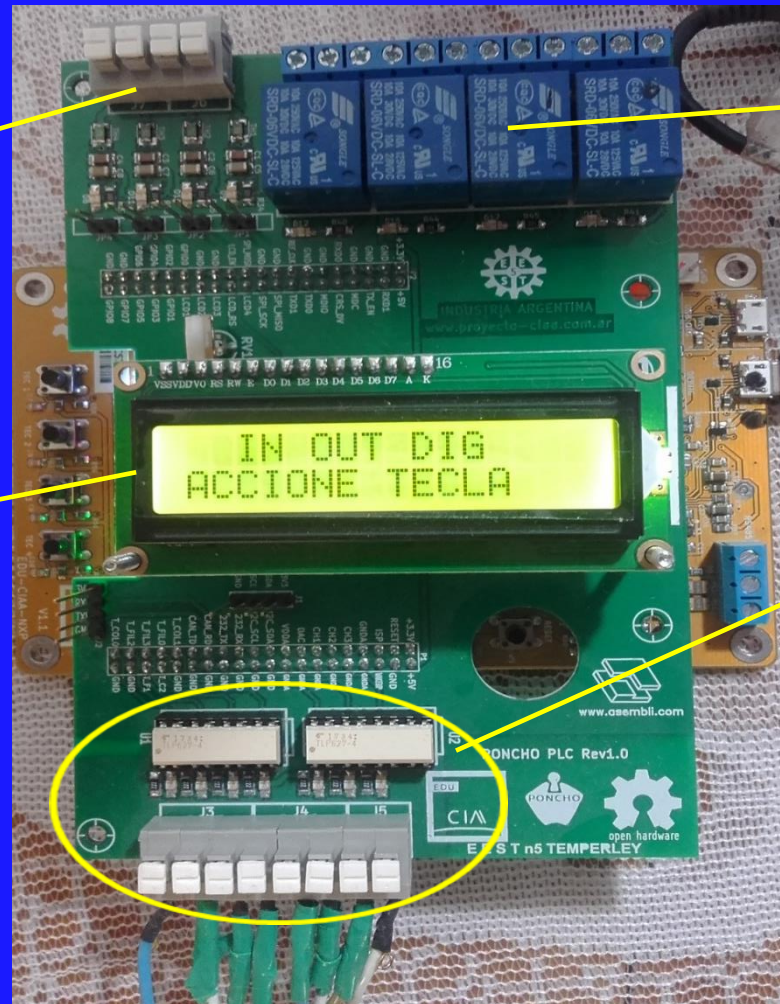
Módulo PH 452



## CONTENIDOS :

- Conexión ESP32 - Sistema Embebido Edu CIAA – poncho PLC
- Interfaz Web - módulos ESP (Aplicación)
- Conexión ESP32 – sensor de temperatura y humedad ambiente DHT22 – Interfaz web – Módulo adquisidor de datos
- Conexión Notebook - ESP32 - Modulo didáctico
- Adquisición de datos de temperatura y humedad de suelo
- ESP32 Módulo PH4502 - Medidor de PH

# PONCHO P.L.C



SALIDAS DIGITALES  
RELÉ

SALIDAS DIGITALES  
TRANSISTOR

ENTRADAS DIGITALES  
OPTOACOPLADAS

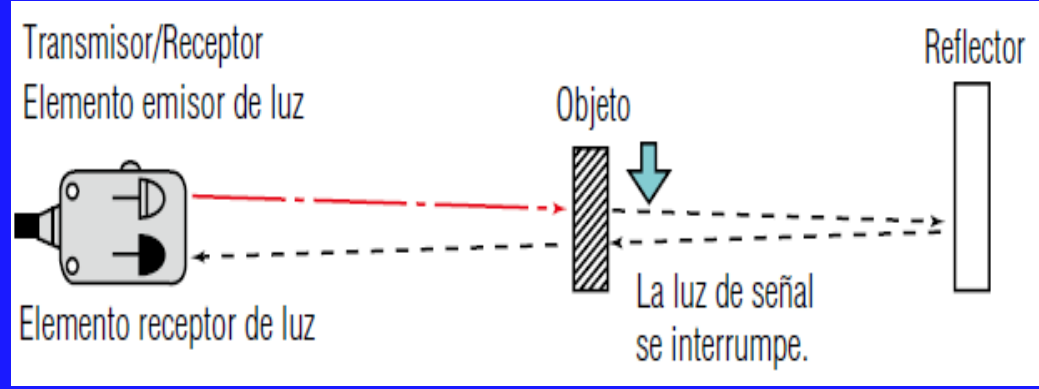
DISPLAY LCD

# ENTRADAS - SALIDAS - DIGITALES

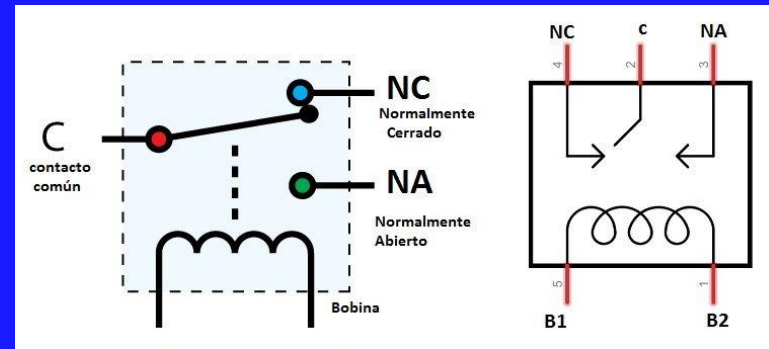
## SENSORES CAPACITIVOS - INDUCTIVOS



## SENSOR ÓPTICO RETROREFLECTIVO



## SALIDA RELÉ





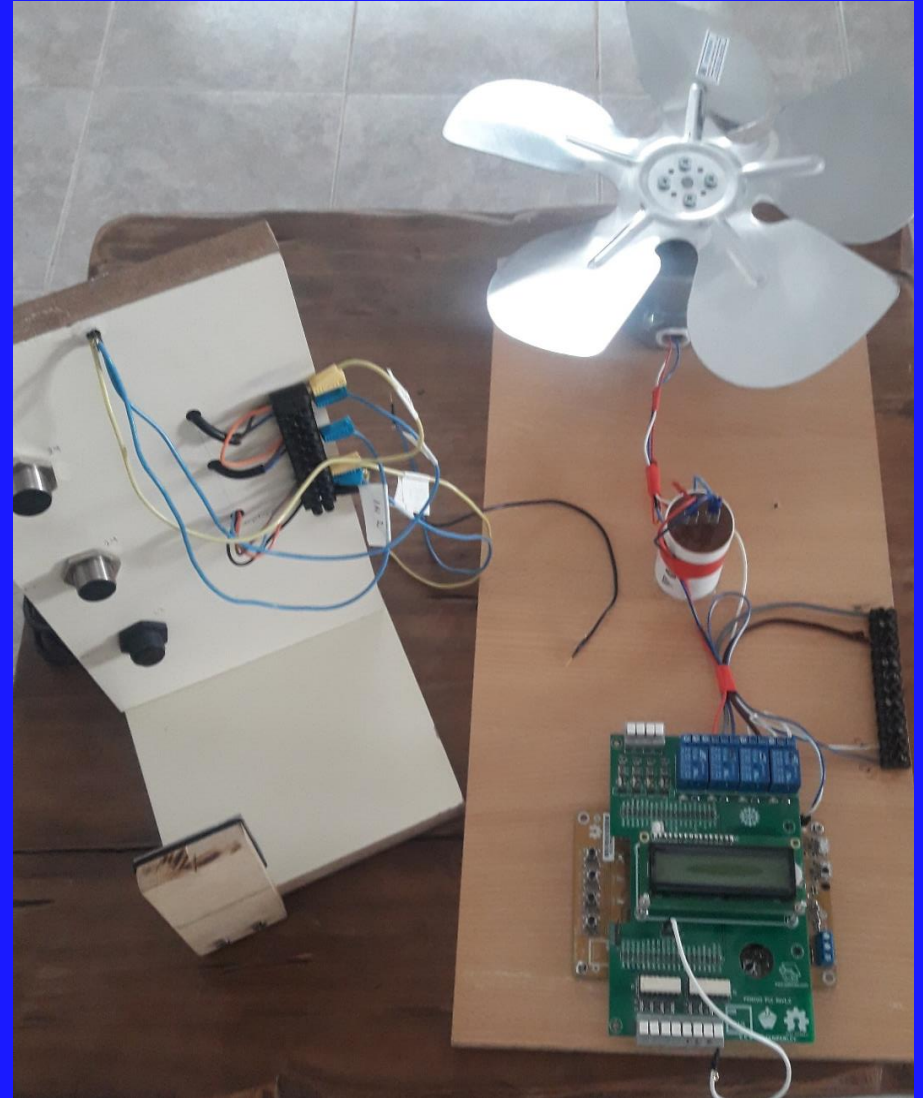
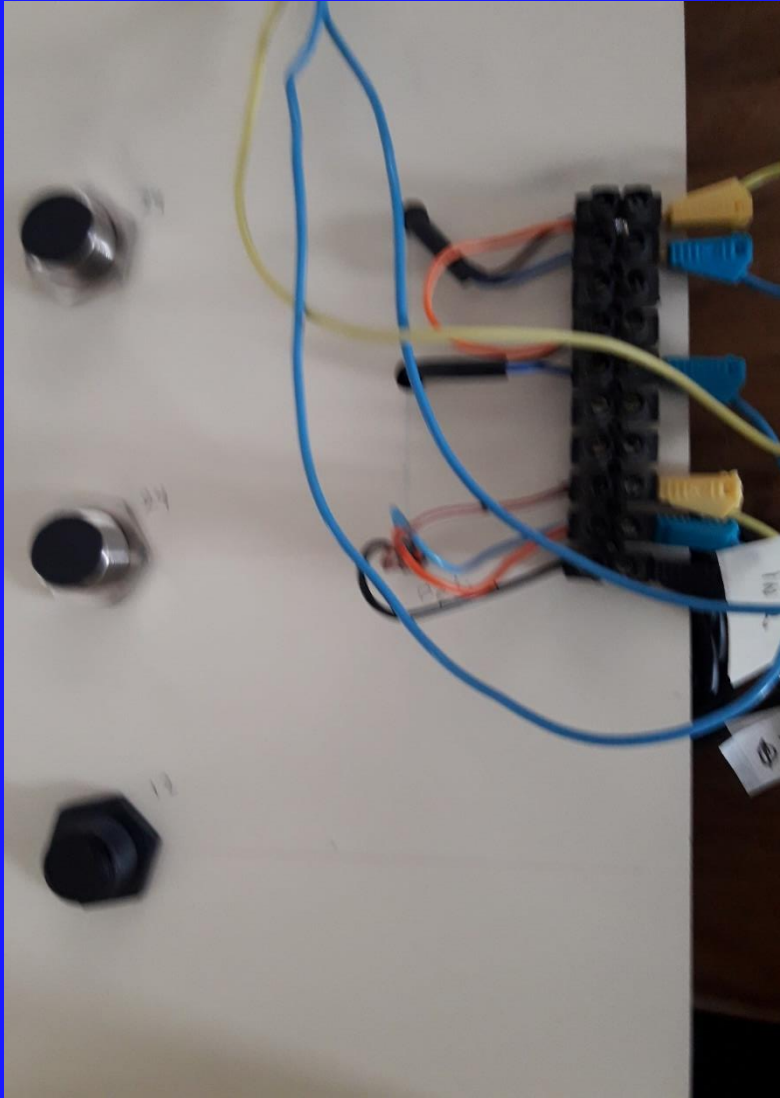
SISTEMAS  
EMBEBIDOS  
2024

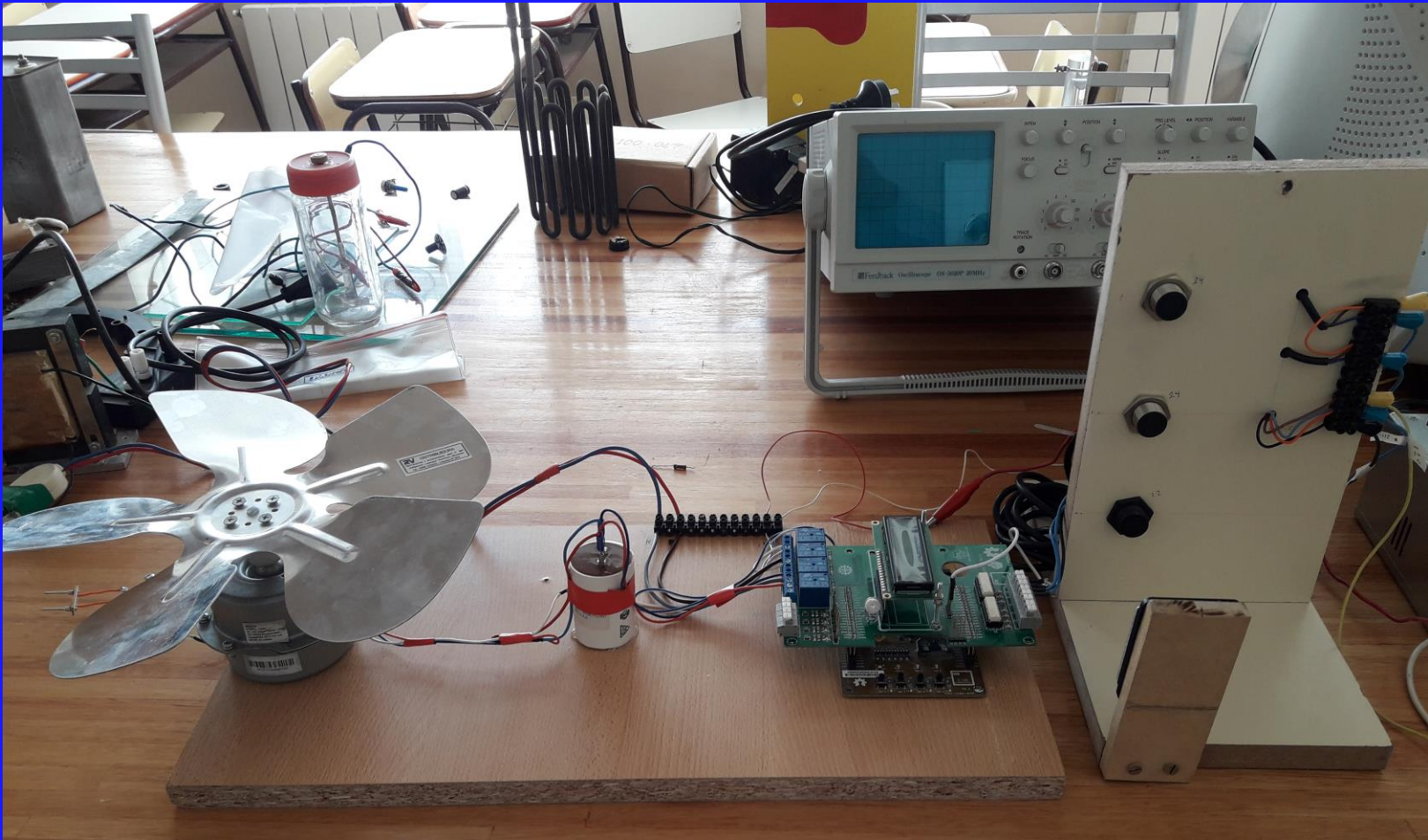


Computadora Industrial  
Abierta Argentina  
Desarrollo colectivo



ESPRESSIF







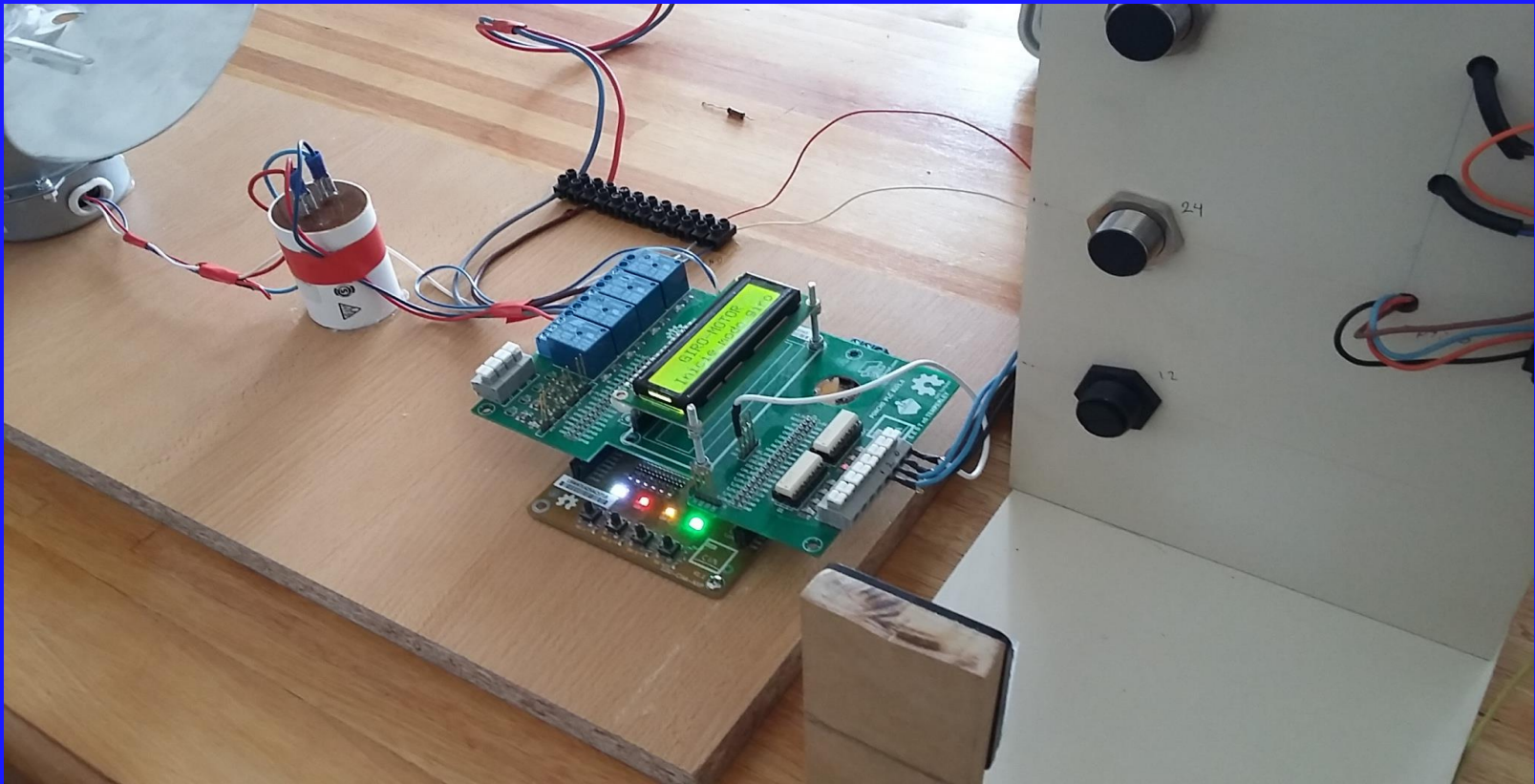
SISTEMAS  
EMBEBIDOS  
2024



Computadora Industrial  
Abierta Argentina  
Desarrollo colectivo



ESPRESSIF

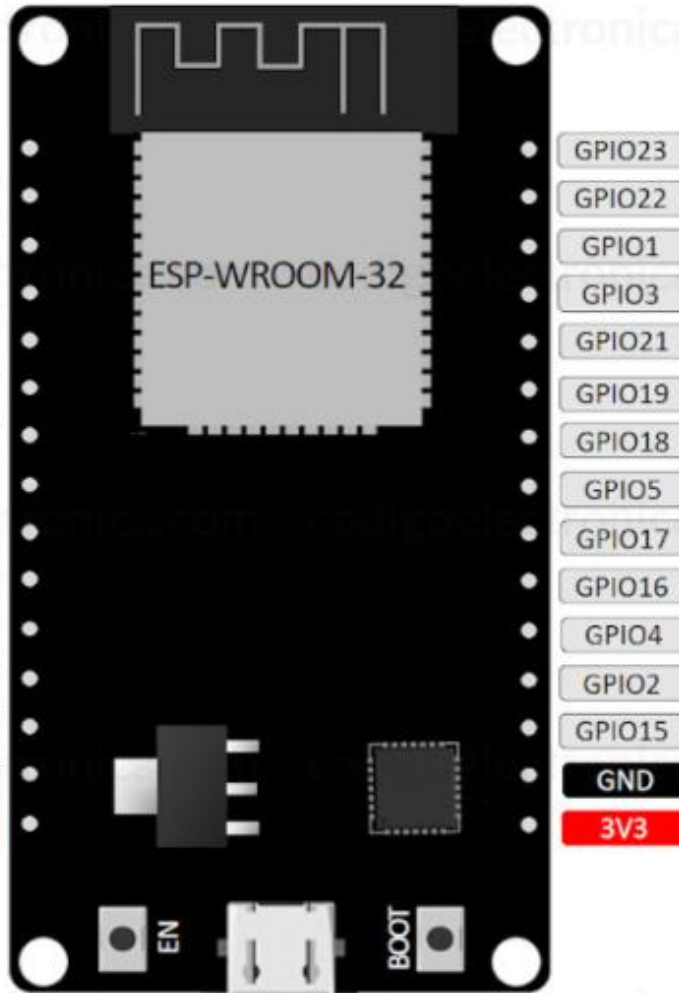


<https://www.youtube.com/watch?v=Jr6z9eIGYnc>



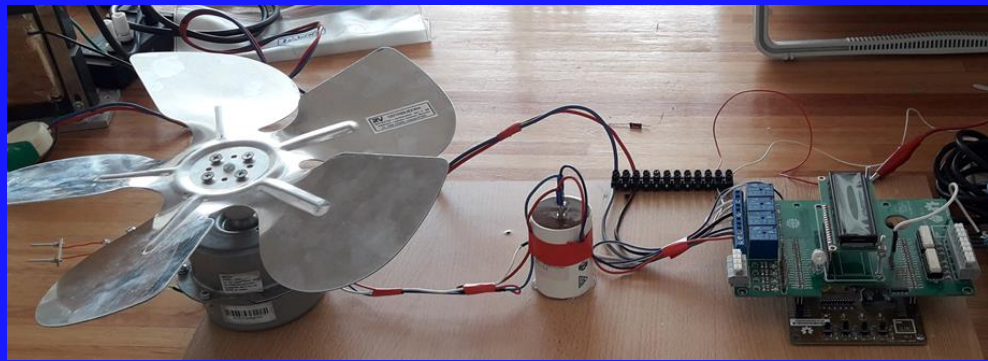
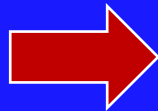
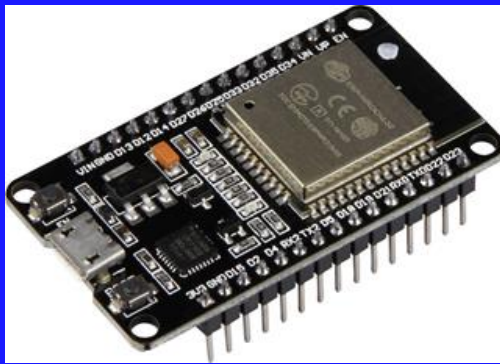
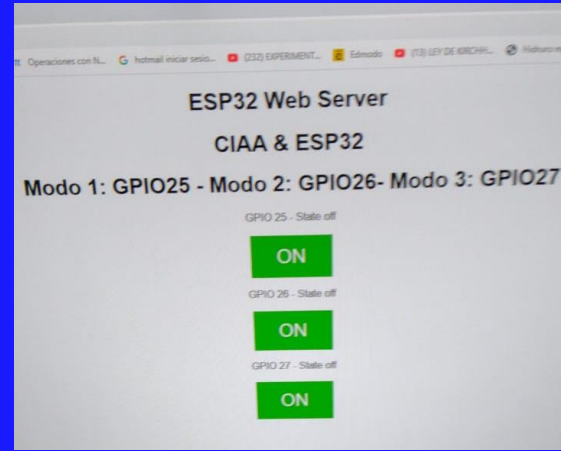
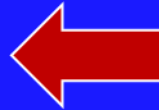
## ESP- WROOM-32

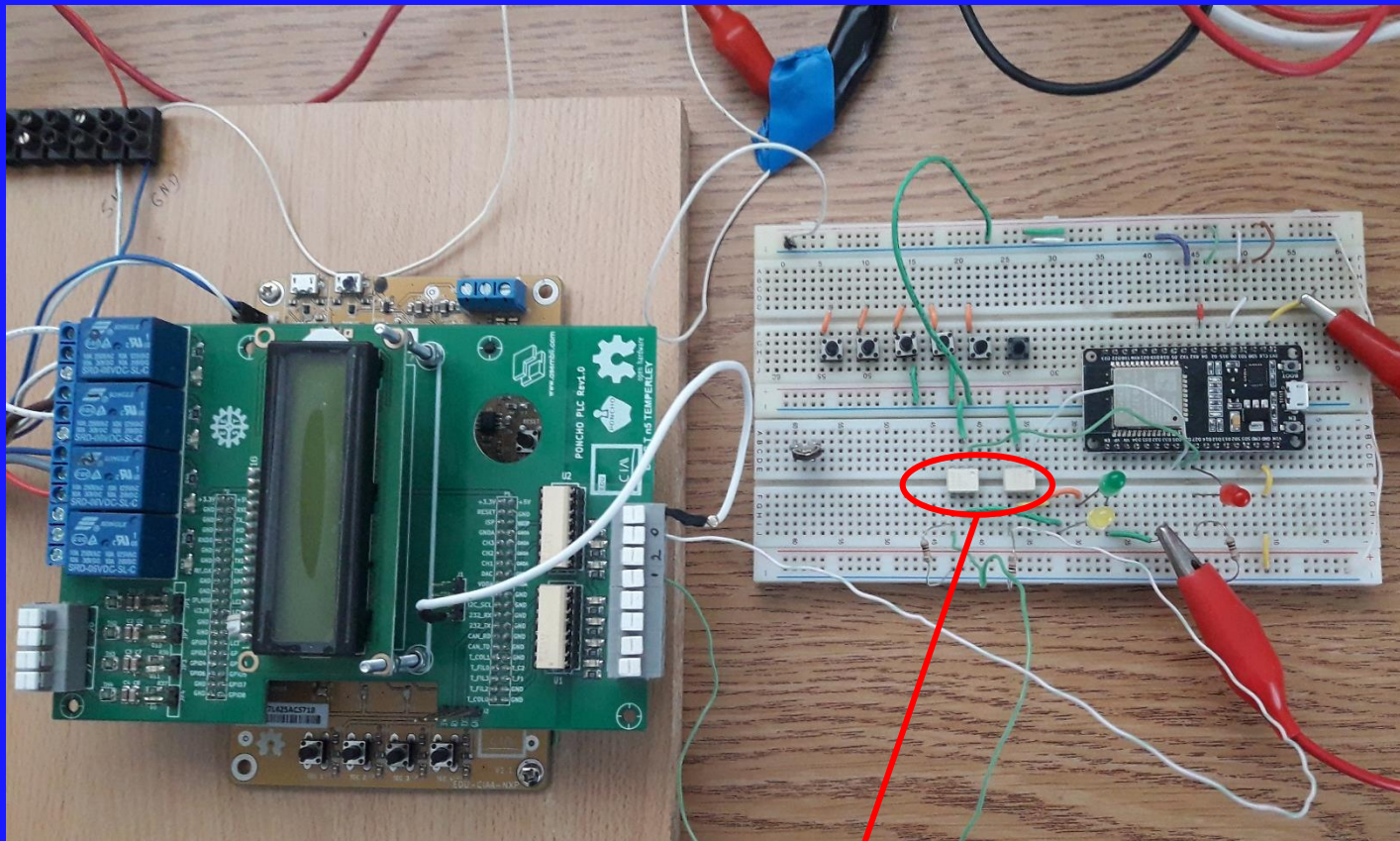
- Microprocesador de 32 – bits
- Interfaz serie-USB y regulador de voltaje
- Comunicación inalámbrica Wifi – Bluetooth
- Interfases periféricas: I2C, SPI, CAN, PWM, etc
- Entradas digitales y analógicas de 12 bit
- Salidas Analógicas PWM
- Controlador SD/MMC
- Controlador remoto infrarrojo
- Algunos modelos tienen integrados: Cámaras, pantallas Oled, Módulos LoRa, etc.



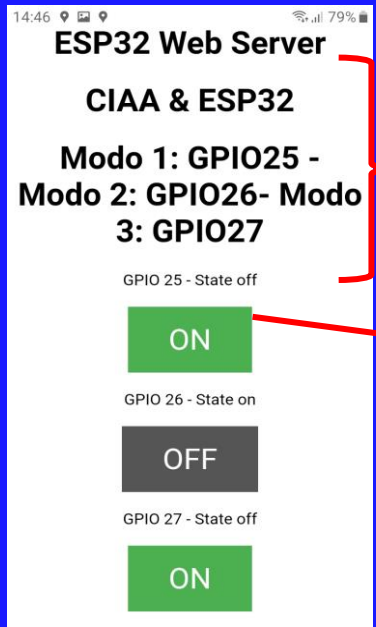
### Celular

### Notebook





Optoacopladores



```
/ Mostrar la página web html
client.println("<!DOCTYPE html><html>");
client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
client.println("<link rel=\"icon\" href=\"data:,\">");

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: ce
client.println(".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;");
client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;});
client.println(".button2 {background-color: #555555;}</style></head>");

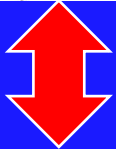
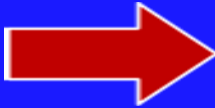
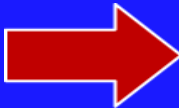
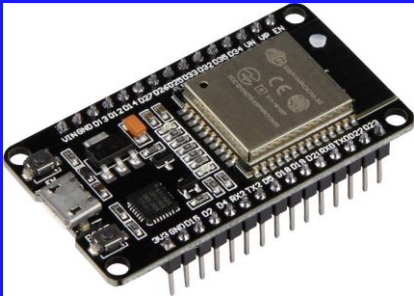
// Cabecera de la página web
client.println("<body><h1>ESP32 Web Server</h1>");
client.println("<body><h1>CIAA & ESP32 </h1>"); //agregado el 22/7
client.println("<body><h1>Modo 1: GPIO25 - Modo 2: GPIO26- Modo 3: GPIO27 </h1>");//agregado 23/7

//Muestra el estado actual y los botones on/off del gpio 25
client.println("<p>GPIO 25 - State " + output25State + "</p>");
// Si el estado de 25 es off muestra el boton on
if (output25State=="off") {
  client.println("<p><a href=\"/25/on\"><button class=\"button\">ON</button></a></p>");
} else {
  client.println("<p><a href=\"/25/off\"><button class=\"button button2\">OFF</button></a></p>");
}

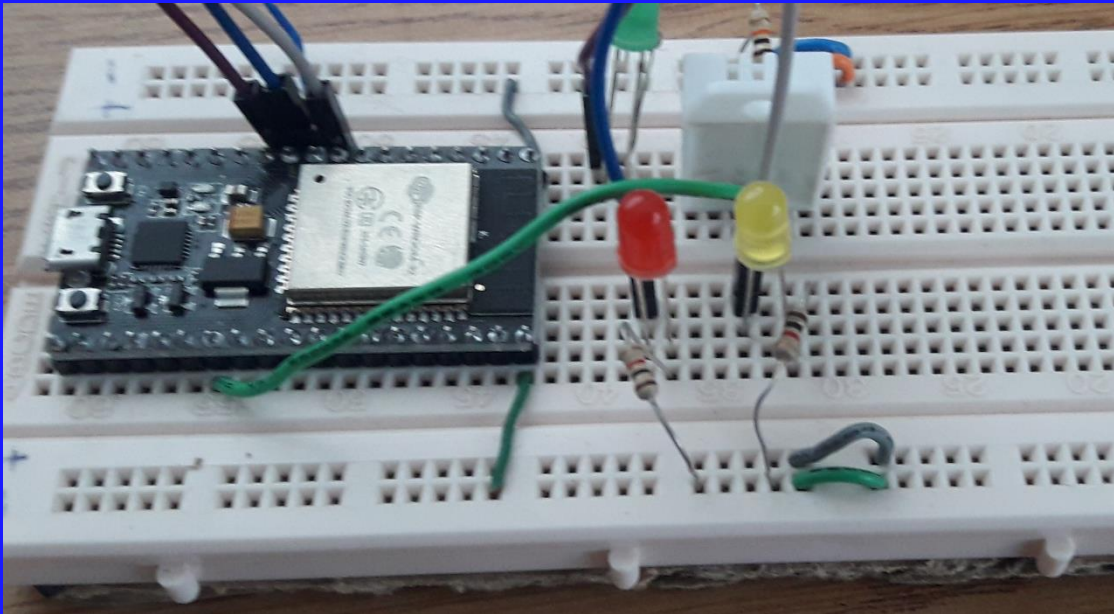
//Muestra el estado actual y los botones on/off del gpio 26
client.println("<p>GPIO 26 - State " + output26State + "</p>");
// muestra el estado actual y los botones on/off del gpio 26
if (output26State=="off") {
```

<https://www.youtube.com/watch?v=RAaM173FSd4>

# ESP32 MCU - DHT 22

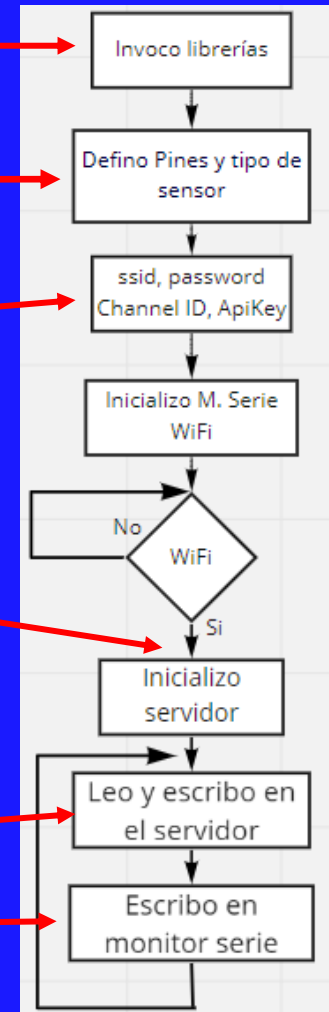


Sensor  
DHT22



```
1 #include "ThingSpeak.h" // agregado para comunicación thingspeak
2 #include "WiFi.h" // agregado para comunicación thingspeak
3 #include "DHT.h"
4
5
6 #define DHTPIN 4 // defino el pin donde estará conectado el
7 #define DHTTYPE DHT22 // defino el tipo de sensor
8
9
10 const char* ssid = "xxxxxxxxx"; // agregado para comunicac
11 const char* password = "xxxxxxxxx"; // agregado para comuni
12
13 unsigned long channelID =144965; // agregado para comun
14 const char* WriteAPIKey = "1ZLADKSNDS30OU"; // agregado para con
15
```

```
.....WIFI CONECTADO!
Temperatura DHT22: 23.00 °C.
Humedad DHT22: 36.10 %.
-----
datos enviados a Write APIKey
Temperatura DHT22: 23.10 °C.
Humedad DHT22: 35.90 %.
-----
datos enviados a Write APIKey
Temperatura DHT22: 23.50 °C.
Humedad DHT22: 35.90 %.
-----
datos enviados a Write APIKey
```



ThingSpeak™ Canales Aplicaciones Apoyo

Uso comercial Cómo comprar

# ThingSpeak para proyectos de IoT

Recopilación de datos en la nube con análisis de datos avanzado usando MATLAB

Empiece gratis [Aprende más](#)




## ThingSpeak para estudiantes y educadores

Implemente proyectos de investigación de IoT rápidamente con herramientas integradas de análisis de datos de MATLAB y recopilación de datos de sensores en tiempo real

## Acerca de ThingSpeak

ThingSpeak es un servicio de plataforma de análisis de IoT que le permite agregar, visualizar y analizar flujos de datos en vivo en la nube. Puede enviar datos a ThingSpeak desde sus dispositivos, crear una visualización instantánea de datos en vivo y enviar alertas.



### Recoger

Envíe los datos del sensor de forma privada a la nube.



### Analizar

Analice y visualice sus datos con MATLAB.



### actuar

Desencadena una reacción.



To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

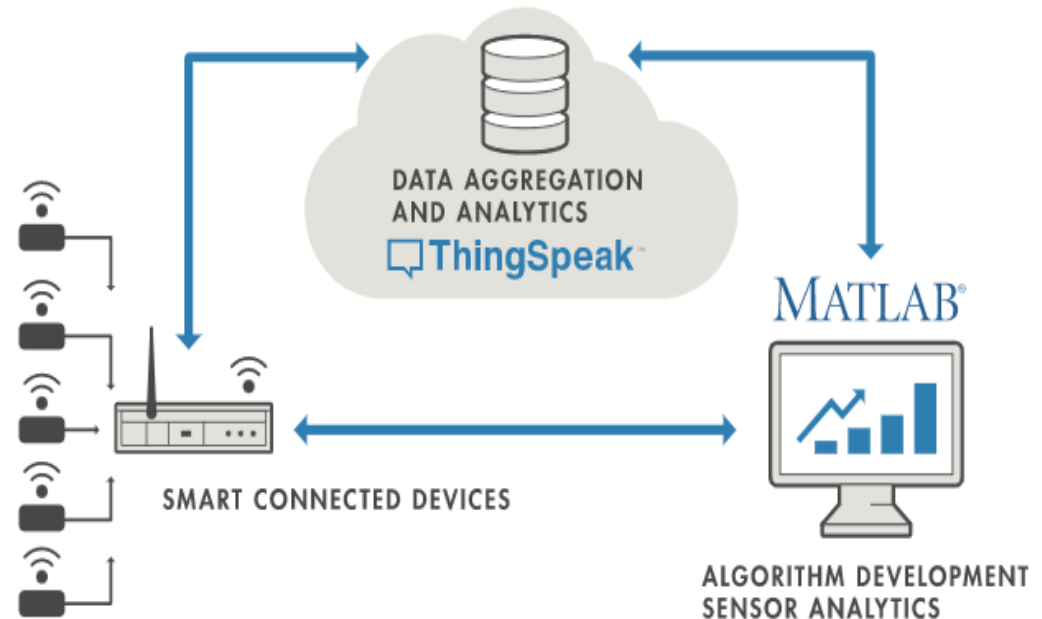


Email

No account? [Create one!](#)

By signing in you agree to our [privacy policy](#).

Next







Author: mwa0000023124537  
Access: Private

Private View | Public View | Channel Settings | Sharing | API Keys | Data Import / Export

+ Add Visualizations | + Add Widgets | Export recent data | MATLAB Analysis | MATLAB Visualization

### Channel Stats

Created: [3 months ago](#)  
Last entry: [a day ago](#)

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾ Commercial Use How to Buy NC

#### Estación de Medición - Temperatura

Date	Temperatura DHT22
12:30	29.5
12:32	29.0
12:34	30.0
12:36	30.5
12:38	29.5
12:40	29.5
12:42	29.5
12:44	29.5
12:46	30.5
12:48	31.5
12:50	31.5

#### Estación de Medición - Humedad

Date	Humedad DHT22
12:30	28.0
12:32	27.5
12:34	27.5
12:36	28.5
12:38	27.5
12:40	28.5
12:42	29.5
12:44	28.5
12:46	28.5
12:48	30.5
12:50	30.5

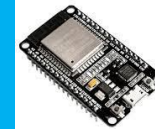
Temperatura DHT22 | Humedad



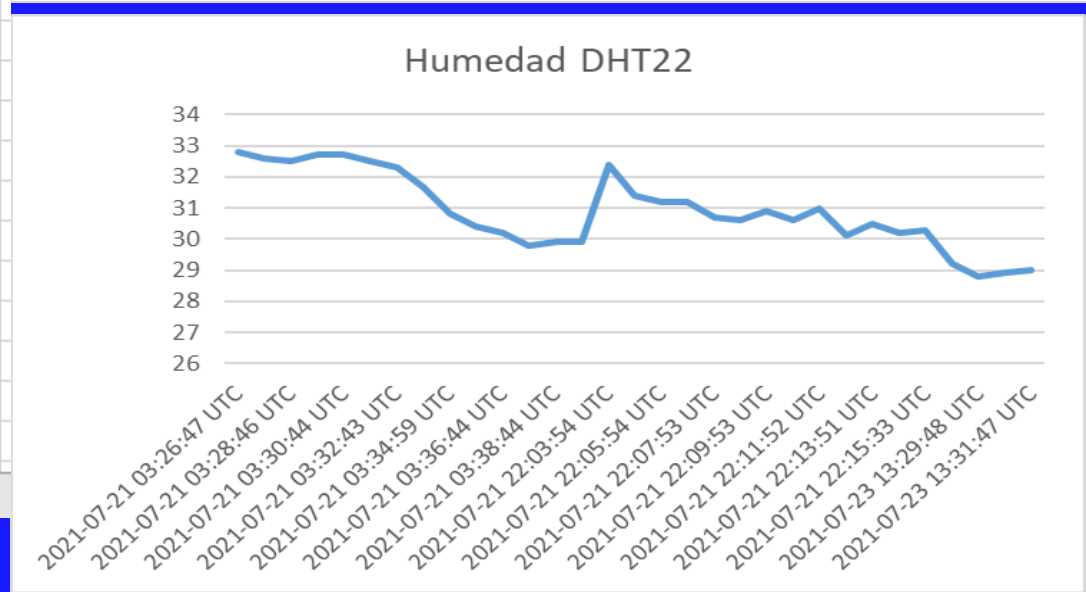
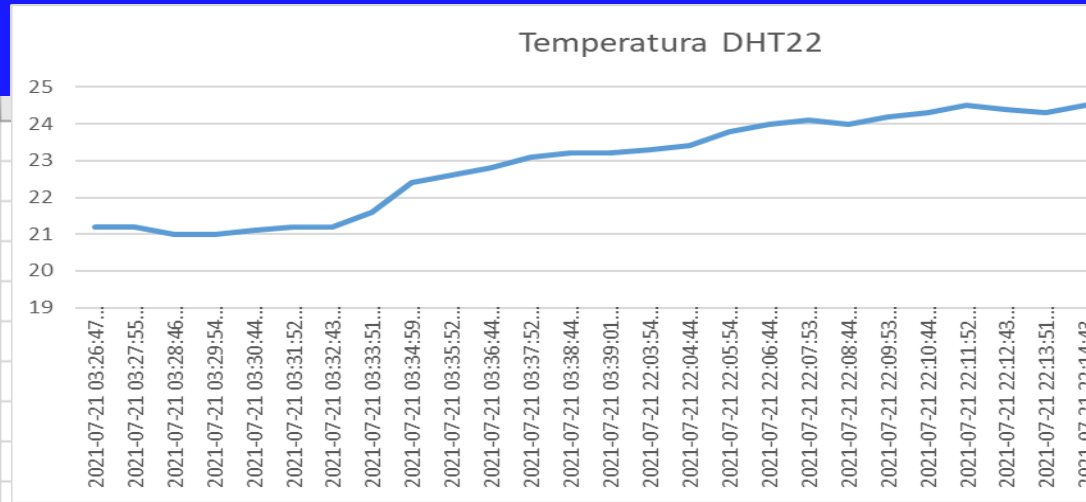
SISTEMAS  
EMBEBIDOS  
2024



Computadora Industrial  
Abierta Argentina  
Desarrollo colectivo



	A	B	C
1	created_at	field1	field2
2			
3	2021-07-21 03:26:47 UTC	21.2	32.8
4	2021-07-21 03:27:55 UTC	21.2	32.6
5	2021-07-21 03:28:46 UTC	21	32.5
6	2021-07-21 03:29:54 UTC	21	32.7
7	2021-07-21 03:30:44 UTC	21.1	32.7
8	2021-07-21 03:31:52 UTC	21.2	32.5
9	2021-07-21 03:32:43 UTC	21.2	32.3
10	2021-07-21 03:33:51 UTC	21.6	31.7
11	2021-07-21 03:34:59 UTC	22.4	30.8
12	2021-07-21 03:35:52 UTC	22.6	30.4
13	2021-07-21 03:36:44 UTC	22.8	30.2
14	2021-07-21 03:37:52 UTC	23.1	29.8
15	2021-07-21 03:38:44 UTC	23.2	29.9
16	2021-07-21 03:39:01 UTC	23.2	29.9
17	2021-07-21 22:03:54 UTC	23.3	32.4
18	2021-07-21 22:04:44 UTC	23.4	31.4
19	2021-07-21 22:05:54 UTC	23.8	31.2
20	2021-07-21 22:06:44 UTC	24	31.2
21	2021-07-21 22:07:53 UTC	24.1	30.7





SISTEMAS  
EMBEBIDOS  
2024



Computadora Industrial  
Abierta Argentina  
Desarrollo colectivo

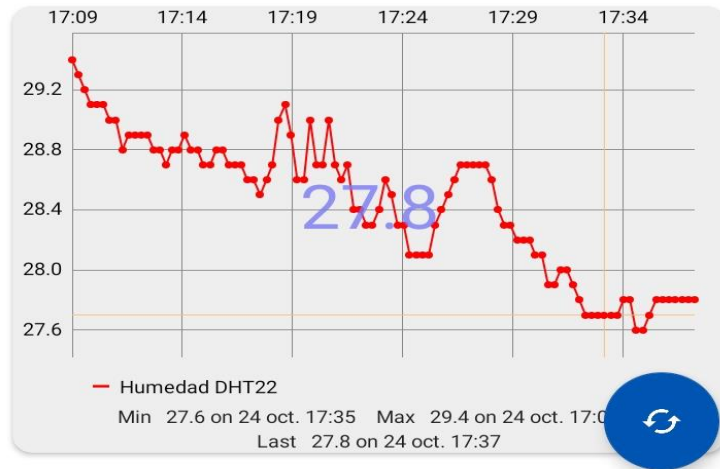
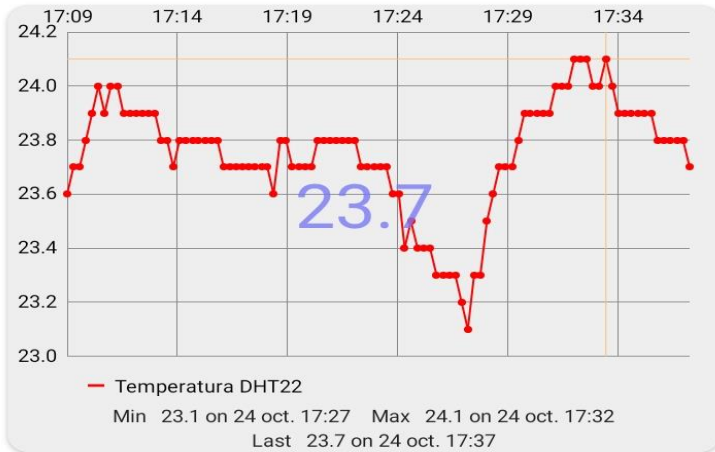


Aplicación desde  
el celular

00:04 40%



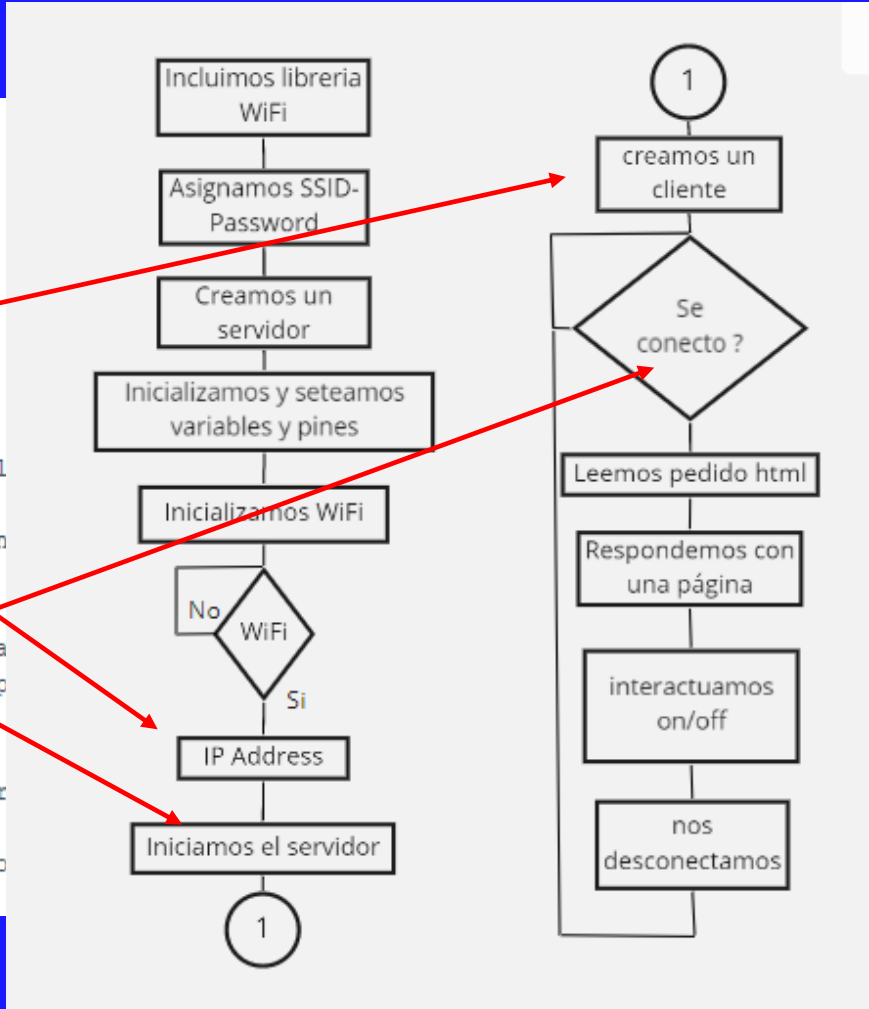
Estación de medición



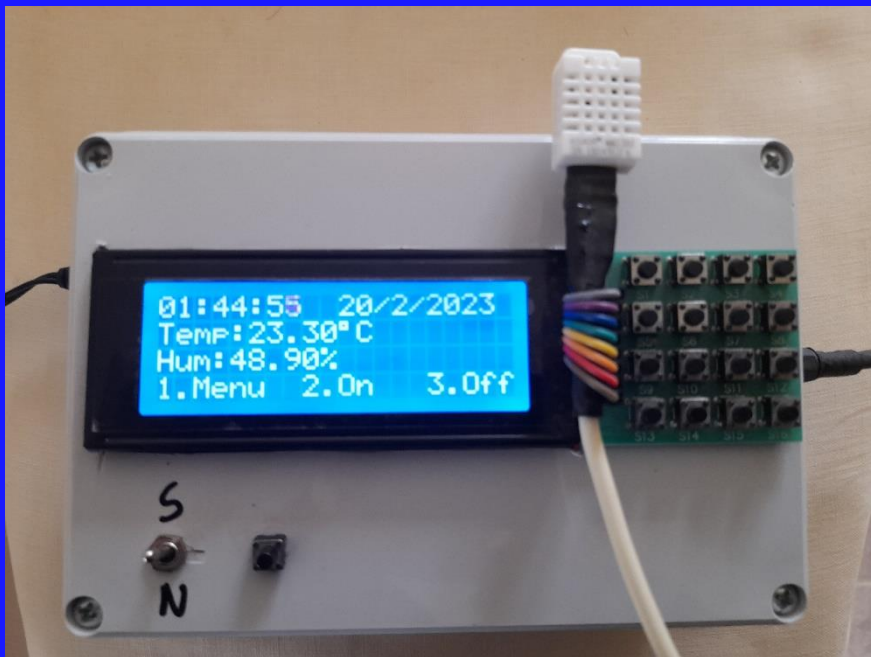
```

50 }
51 // Imprima la dirección IP y active el servidor |
52 Serial.println("");
53 Serial.println("WiFi connected.");
54 Serial.println("IP address: ");
55 Serial.println(WiFi.localIP());
56 server.begin();
57 }
58
59 void loop(){
60   WiFiClient client = server.available(); // Escucha a los cl
61
62   if (client) { // si se conecta un
63     currentTime = millis();
64     previousTime = currentTime;
65     Serial.println("New Client."); // escribe un mensa
66     String currentLine = ""; //Hace una cadena p
67     while (client.connected() && currentTime - previousTime <=
68           currentTime = millis();
69           if (client.available()) { // Si hay datos par
70             char c = client.read(); // lea un byte
71             Serial.write(c); // imprima en el mo
72             header += c;

```



## Módulo adquisidor de datos de temperatura y humedad ambiente





# APLICACIONES EN DOCENCIA

# Aplicaciones en Docencia

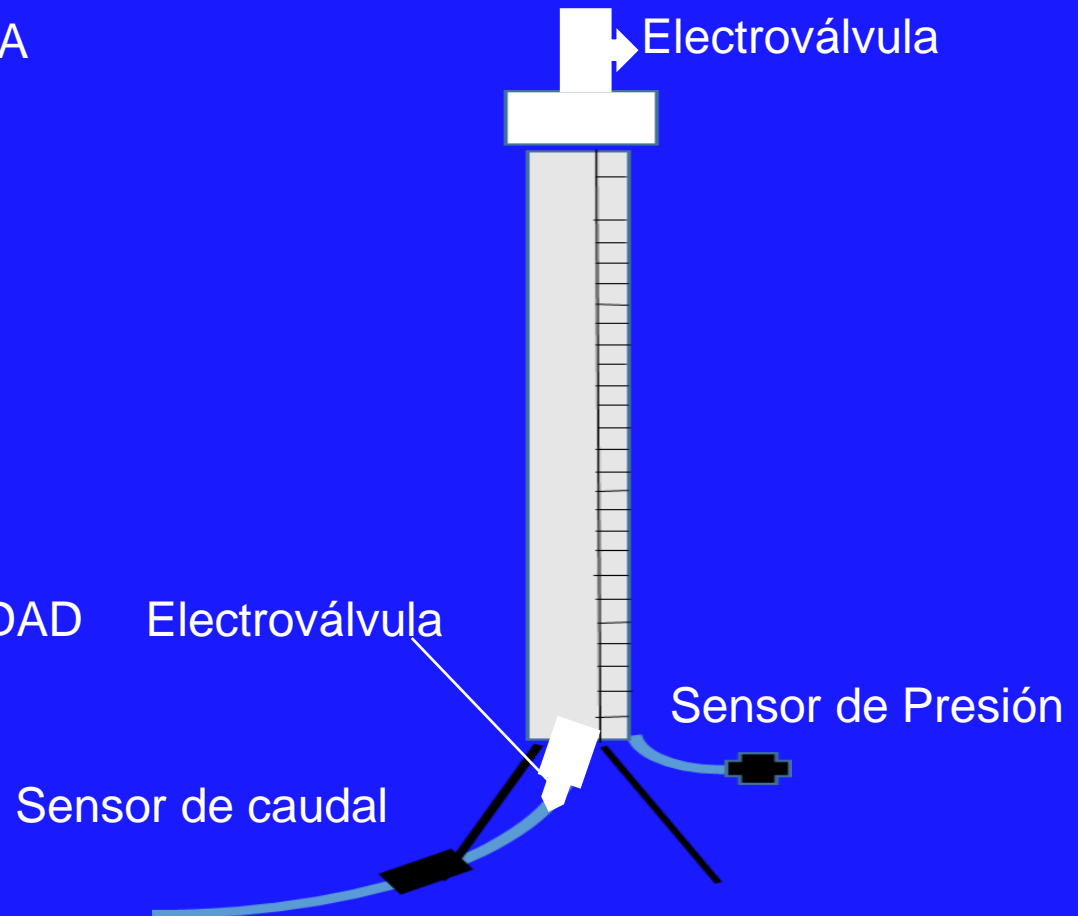
## LEY FUNDAMENTAL DE LA HIDROSTÁTICA

Presión absoluta  
 $P = P_0 + \rho * g * h$

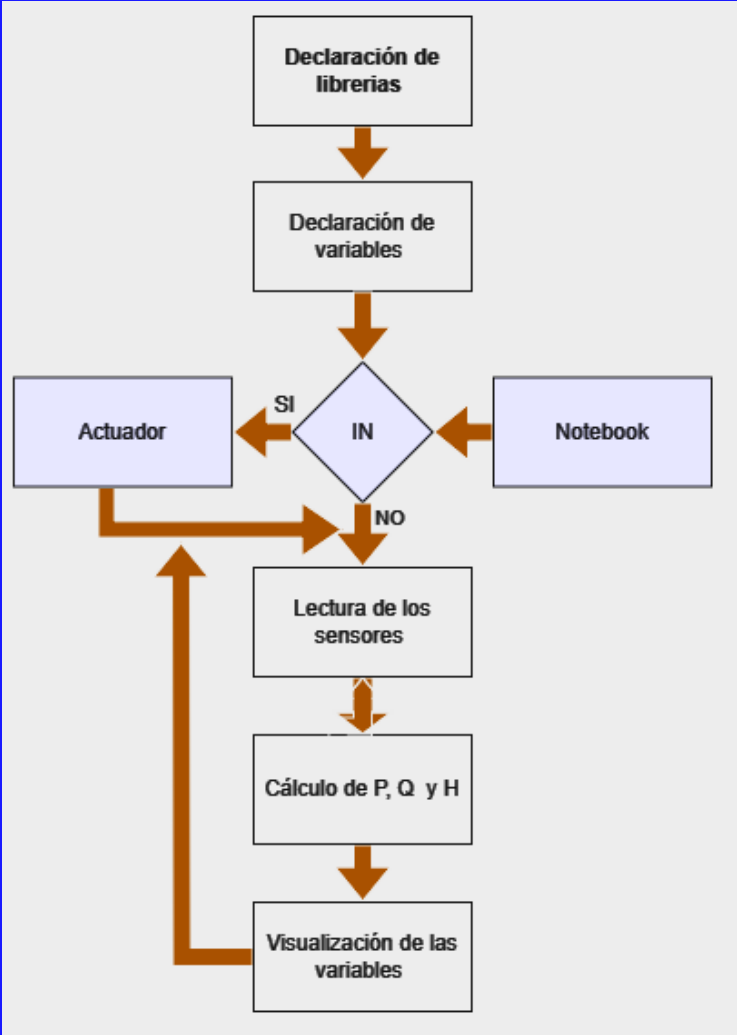
Presión manométrica  
 $P - P_0 = \rho * g * h$

## ECUACIÓN DE CONTINUIDAD

$$Q = A_1 * V_1 = A_2 * V_2$$



**Diagrama de flujo del programa**



**Sensor de Presión**



**Display 20x4**



**Sensor de Caudal**







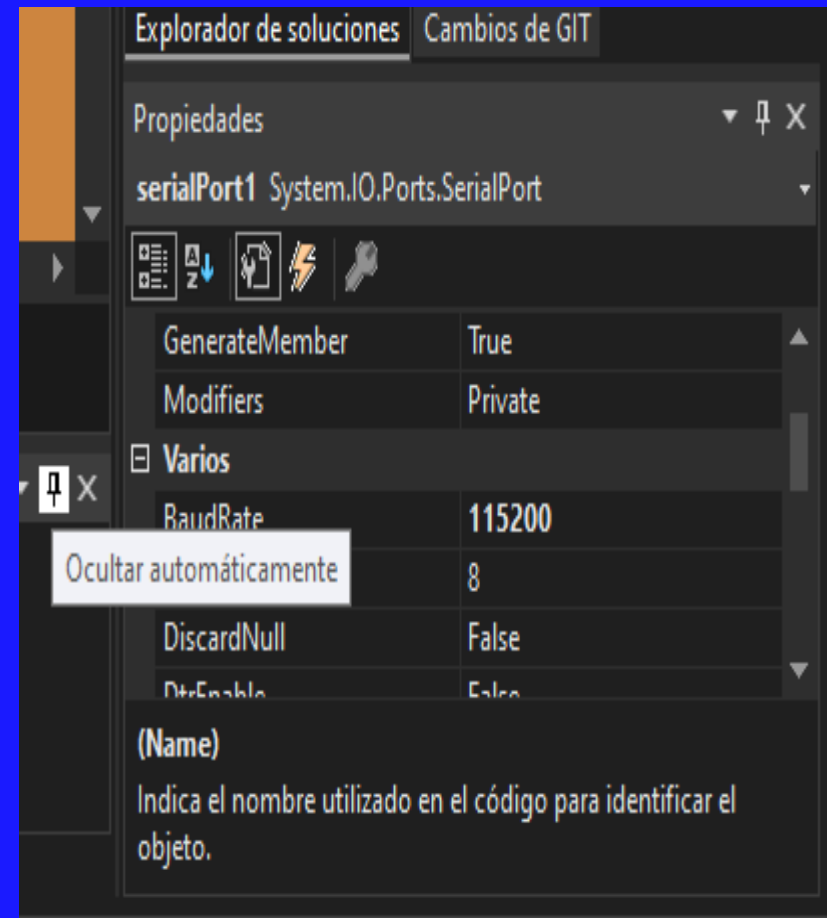
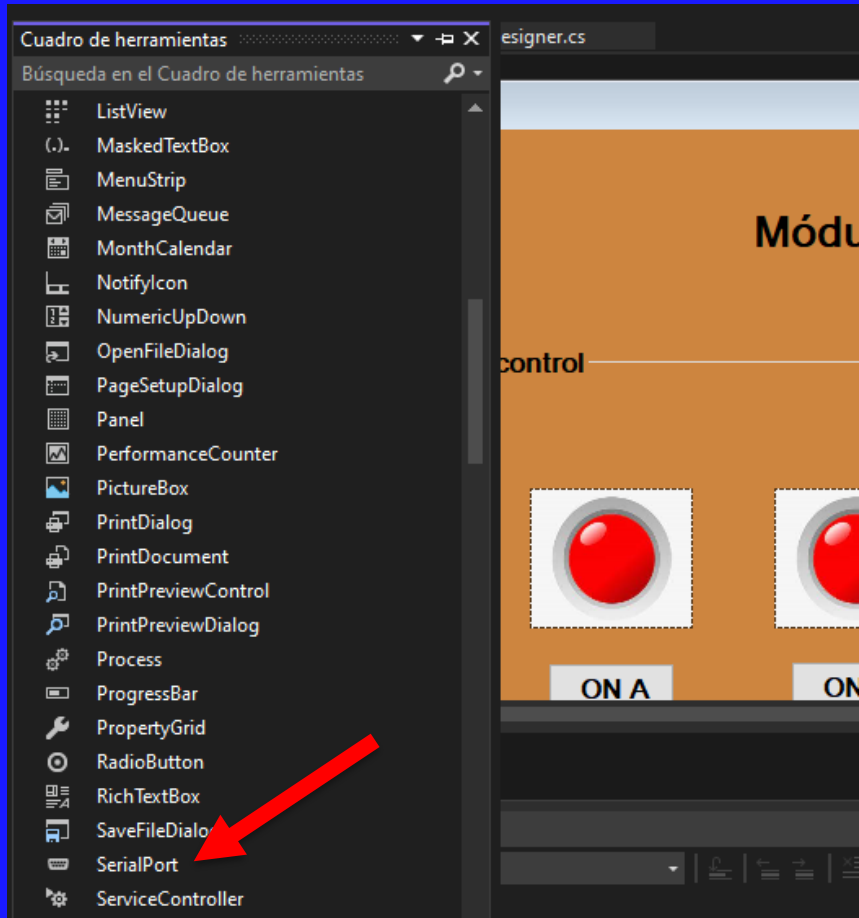
Parte del programa que reside en el ESP32

Parte del programa en VSC

```
51 Serial.print("AD = ");
52 Serial.println(promedio); // promedio de todas la
53
54
55
56 lcd.setCursor(0,0); //
57 lcd.print("Presion Manometrica ");
58
59 lcd.setCursor(0,1); //
60 lcd.print (" P= ");
61 lcd.print(manoMetrica);
62 lcd.print(" Kpa ");
63
64 lcd.setCursor(0,2); //
65 lcd.print ("Altura H=");
66 lcd.print (Altura);
67 lcd.print (" m ");
68
69 lcd.setCursor (0,3);
70 lcd.print ("V=");
71 lcd.print( promedio );
72 lcd.print(" Volts ");
73
```

```
110 try
111 {
112     pictureBoxLedA.Image = Properties.Resources.led3;
113     serialPort1.WriteLine("$AOn");
114 }
115 catch (Exception error)
116 {
117     MessageBox.Show(error.Message);
118 }
119
120
121
122 1 referencia
123 private void buttonOffA_Click(object sender, EventArgs e)
124 {
125     if (serialPort1.IsOpen)
126     {
127         try
128         {
129             pictureBoxLedA.Image = Properties.Resources.led2;
130             serialPort1.WriteLine("$AOff");
131         }
132         catch (Exception error)
133         {
134             MessageBox.Show(error.Message);
135         }
136     }
137 }
```

## CUADRO DE HERRAMIENTAS Y CONFIGURACIÓN

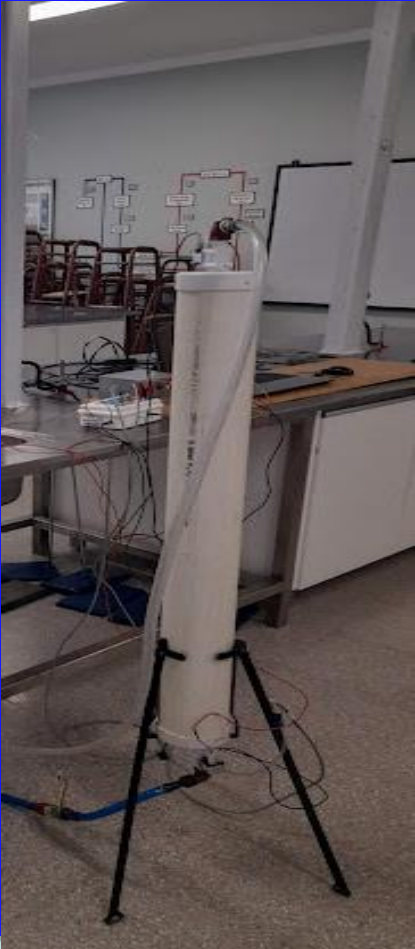
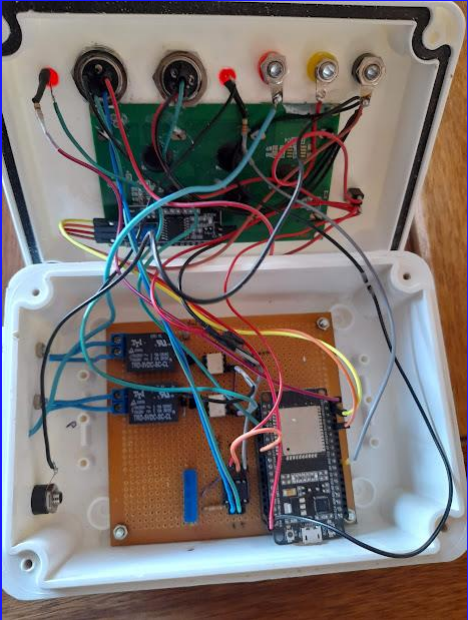




## Entorno de desarrollo del VSC

The screenshot shows the Visual Studio Code IDE environment. The main workspace displays a Windows Form titled "Control de Leds" with a brown background. The form contains three red LED indicators, each with a label "ON A", "ON B", and "ON C" below it. The text "Módulo didáctico para medir Presión Hi" is visible at the top of the form. The Solution Explorer on the right shows the project structure for "HMITresleds", including files like Properties, Referencias, Recursos, App.config, Form1.cs, and Program.cs. The Properties window at the bottom right shows the properties for the "Controldeleds" control, including Font (Microsoft Sans Serif, 8,25pt), ForeColor (ControlText), FormBorderStyle (Sizable), RightToLeft (No), RightToLeftLayout (False), and Text (Control de Leds).

# MÓDULO DIDÁCTICO PARA MEDIR PRESIÓN, CAUDAL DE SALIDA Y ALTURA DE UNA COLUMNA DE AGUA



# MÓDULO DIDÁCTICO PARA MEDIR PRESIÓN, CAUDAL DE SALIDA Y ALTURA DE UNA COLUMNA DE AGUA

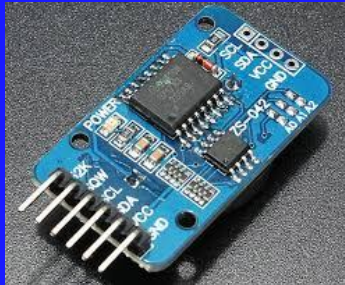
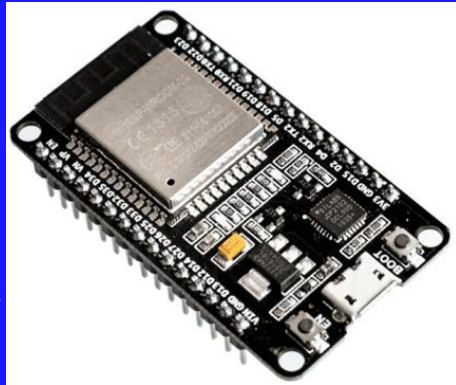


# Prototipo adquirente de datos de Temperatura y Humedad de tierra rtc

Sensor de temperatura



ESP WROOM 32



rtc

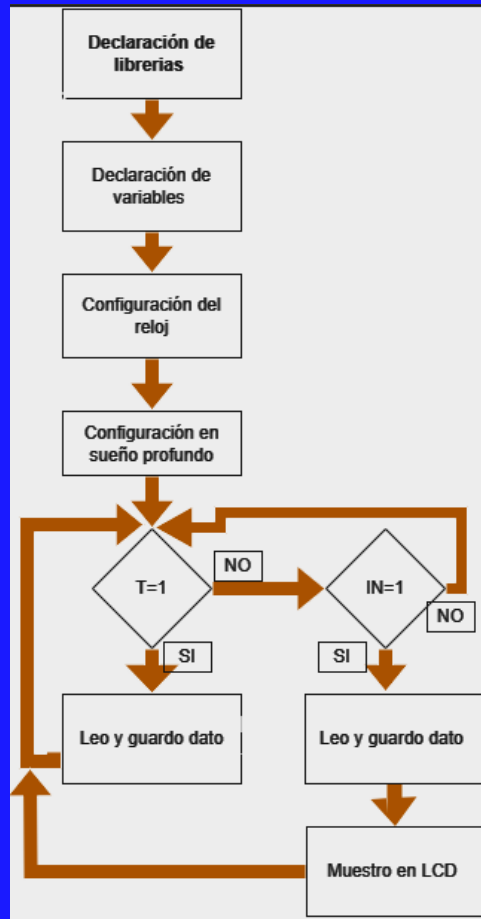
Sensor de humedad



Memoria SD



# Prototipo adquirente de datos de Temperatura y Humedad de tierra - Diagrama de Flujo

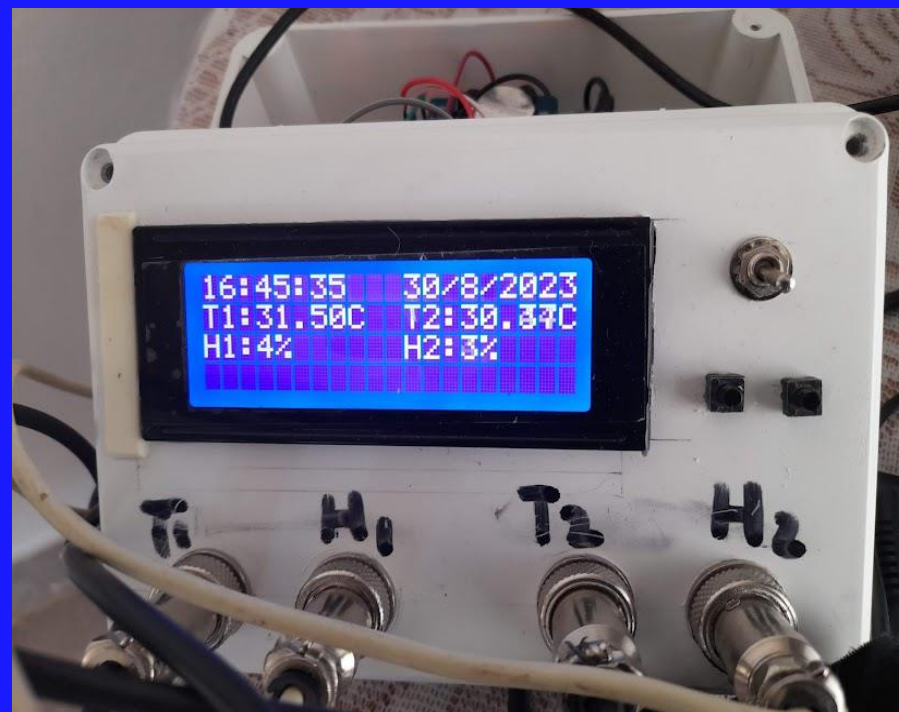


## Parte del programa residente en el esp32

```

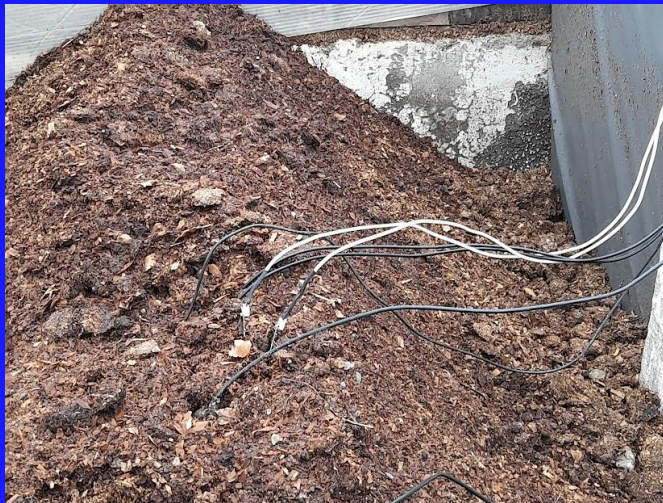
95 lcd.setCursor(10,0); //adopto la posi
96 lcd.print(now.day());
97 lcd.print("/");
98 lcd.print(now.month());
99 lcd.print("/");
00 lcd.print(now.year());
01
02 //-----Temperaturas
03 sensors1.requestTemperatures(); //Se
04 temp1= sensors1.getTempCByIndex(0);
05 lcd.setCursor(0,1); //primera columna-
06 lcd.print("T1:");
07 lcd.print(temp1);
08 lcd.print("C");
09
10 sensors2.requestTemperatures(); //Se
11 temp2= sensors2.getTempCByIndex(0);
12 lcd.setCursor(11,1); //12 columna - seg
13 lcd.print("T2:");
14 lcd.print(temp2);
15 lcd.print("C");
16
  
```

# Prototipo adquirente de datos de Temperatura y Humedad de tierra - Calibración





# Adquisidor de datos de Temperatura y Humedad de tierra Mediciones en campo





# Adquisidor de datos de Temperatura y Humedad de tierra

## Datos adquiridos en la memoria SD

```

Reading ID,Date,Time,Temp1,Temp2,Hum1,Hum2,
0,2023-09-16,12:51:54,19.12,19.87,1,0
1,2023-09-16,12:52:00,19.00,19.81,0,0
2,2023-09-16,13:07:24,15.50,16.12,1,0
3,2023-09-16,13:22:22,15.31,16.12,1,0
4,2023-09-16,13:37:21,15.50,16.12,1,0
5,2023-09-16,13:52:19,15.69,16.31,1,0
6,2023-09-16,14:07:18,15.50,16.19,1,0
7,2023-09-16,14:22:16,15.63,16.25,1,0
8,2023-09-16,14:37:15,15.81,16.44,1,0
9,2023-09-16,14:52:14,15.75,16.50,1,0
10,2023-09-16,15:07:14,15.81,16.50,1,0
11,2023-09-16,15:22:13,15.94,16.56,1,0
12,2023-09-16,15:22:39,15.88,16.62,1,0
13,2023-09-16,15:38:02,17.19,18.12,0,0
14,2023-09-16,15:53:01,17.69,18.75,0,0
15,2023-09-16,16:08:00,17.94,19.06,0,0
16,2023-09-16,16:22:59,18.00,19.31,0,0
17,2023-09-16,16:37:59,18.12,19.37,0,0
18,2023-09-16,16:52:58,18.37,19.62,0,0
19,2023-09-16,17:07:57,18.00,19.56,0,0
20,2023-09-16,17:22:55,18.00,19.62,0,0
21,2023-09-16,17:37:54,18.06,19.69,0,0
22,2023-09-16,17:52:52,17.94,19.62,0,0
23,2023-09-16,18:07:51,17.87,19.56,0,0
    
```

## Datos adquiridos y procesados en Excel

3	27/9/2023	10:45:28	13.75	11.81	39	34
4	27/9/2023	11:00:50	14.25	12.13	38	33
5	27/9/2023	11:15:48	14.44	12.31	37	30
6	27/9/2023	11:30:47	14.50	12.56	39	31
7	27/9/2023	11:45:46	14.69	12.69	36	29
8	27/9/2023	12:00:46	14.81	12.94	37	31
9	27/9/2023	12:15:45	15.06	13.13	36	30
10	27/9/2023	12:30:44	15.19	13.31	36	30
11	27/9/2023	12:45:43	15.31	13.56	38	32
12	27/9/2023	13:00:42	15.50	13.69	37	32
13	27/9/2023	13:15:41	15.75	13.94	37	32
14	27/9/2023	13:30:40	15.88	14.06	40	35
15	27/9/2023	13:45:39	16.12	14.31	40	33
16	27/9/2023	14:00:38	16.19	14.44	42	36
17	27/9/2023	14:15:37	16.31	14.50	39	32
18	27/9/2023	14:30:36	16.37	14.56	39	32
19	27/9/2023	14:45:34	16.44	14.56	38	32
20	27/9/2023	15:00:33	16.56	14.63	42	36
21	27/9/2023	15:15:33	16.62	14.69	38	30
22	27/9/2023	15:30:31	16.75	14.75	37	30
23	27/9/2023	15:45:30	16.81	14.81	35	29
24	27/9/2023	16:00:29	16.94	14.75	37	29



# Medidor de PH Módulo PH 4502 & NODEMCU- ESP 32S

## ¿Qué es un medidor de PH?

El medidor de pH es un instrumento utilizado para medir la acidez o la alcalinidad de una solución, también llamado de pH. El pH es la unidad de medida que describe el grado de acidez o alcalinidad y es medido en una escala que va de 0 a 14.

el valor de PH de determinada sustancia está directamente relacionado a la proporción de el valor de PH de determinada sustancia está directamente relacionado a la proporción de las concentraciones de los iones de hidrógeno  $[H^+]$  e hidroxilo  $[OH^-]$

Solución ácida —  $[H^+] > [OH^-]$

Solución alcalina —  $[H^+] < [OH^-]$

# Antecedentes de medición de PH

## Papel Tornasol



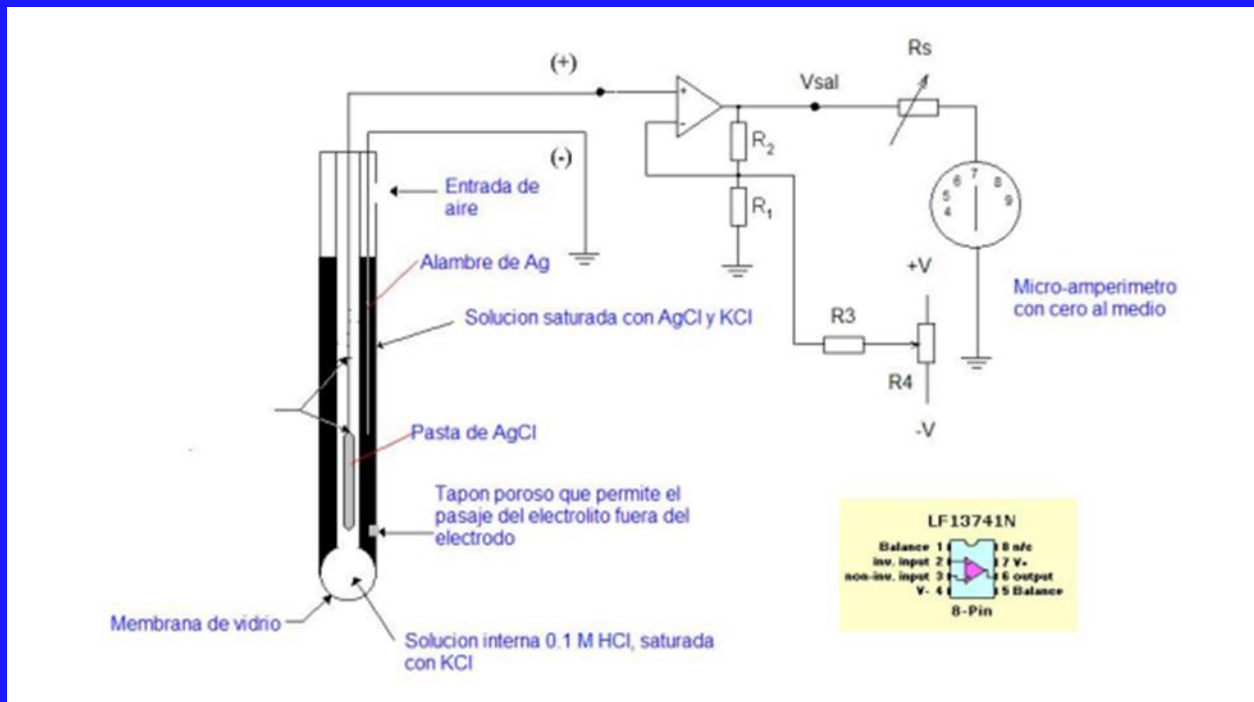
### Compuestos Ácidos

- Ácido de baterías 0-1
- Lluvia ácida 2-5
- Limón 2-3 (comestible)
- Café 5

### Compuestos Alcalinos

- Leche de Magnesio 10 -11 (comestible)
- Cloro 13
- Jabón y detergentes
- Bicarbonato de sodio
- Productos de limpieza

## Esquema eléctrico básico de una medición potenciométrica de PH



## Características técnicas del módulo PH 4502

- Amplio rango de mediciones. 0 - 14
- Interfase sencilla
- Salida analógica y digital
- Calibración sencilla
- Amplio espectro de aplicaciones



## Características técnicas de la Sonda E201

- Rango de medición: 0-14 PH
- Punto cero: 7 +- 0.5 PH
- Tiempo de respuesta: menor a 2 min
- Temp. de funcionamiento 0 - 60°C
- Conector: BNC
- Longitud: 70 cm aproxim.
- Resistencia interna: 250 Megaohms
- Humedad ambiente: menor al 85%
- Temperatura Ambiente: 5 - 40 °C





**PINOUT**  
Sensor de PH Liquido PH-4502C  
con electrodo E201-BNC



LED ROJO: Limite de PH  
LED VERDE: Alimentación

Conector a BNC

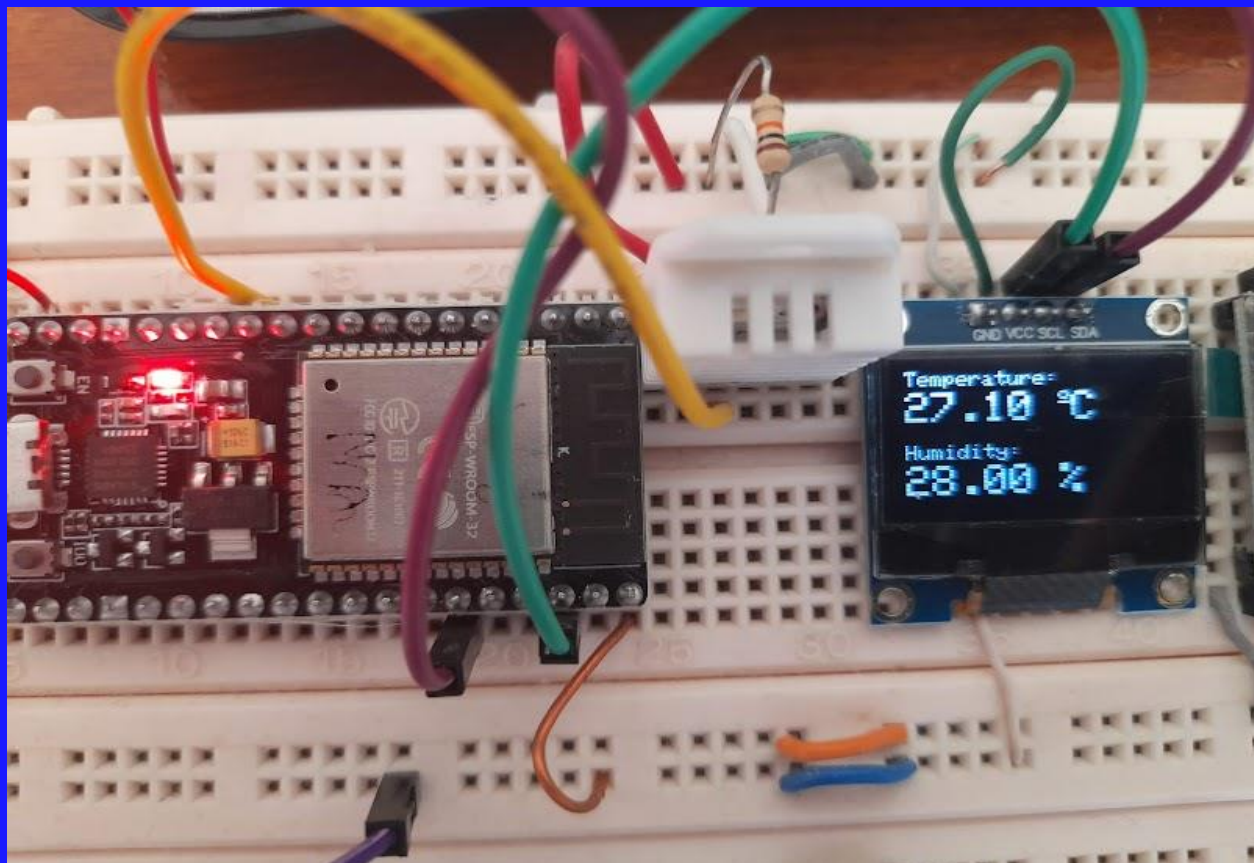
POT 1 Lectura analógica(offset)  
POT2 Limite de medicion PH

TO: Salida Analogica de Temperatura  
DO: Salida Digital de PH Limite (3.3V)  
PO: Salida Analogica de PH(V)

GND: Masa del Circuito Analógico  
GND: Masa de Alimentación  
VCC: 5V



## PRIMEROS ENSAYOS ESP32 – DISPLAY OLED – DHT22



## Preparación de muestras con diferentes PH





Buffer PH 4  
Acido

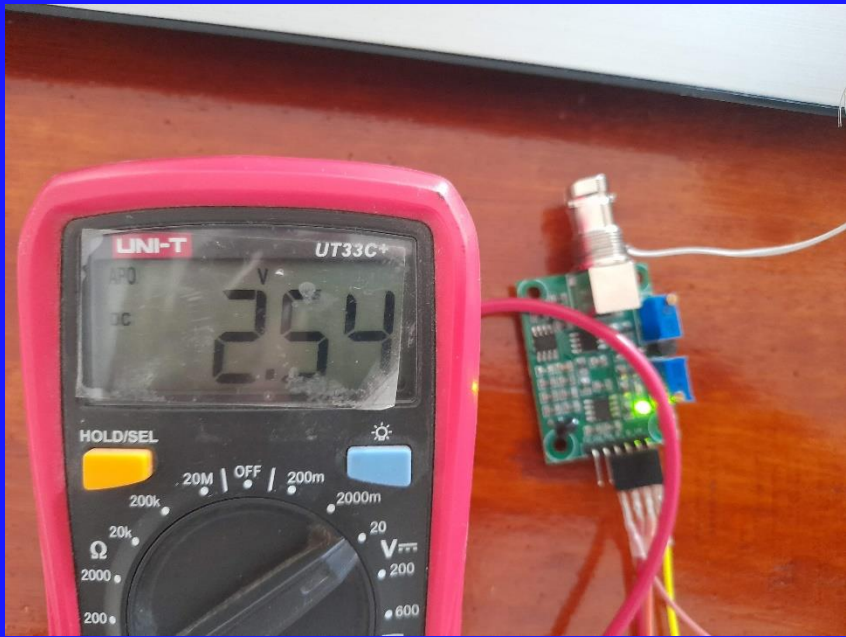


Buffer PH 6,86  
Neutro



Buffer PH 9,18  
Alcalino

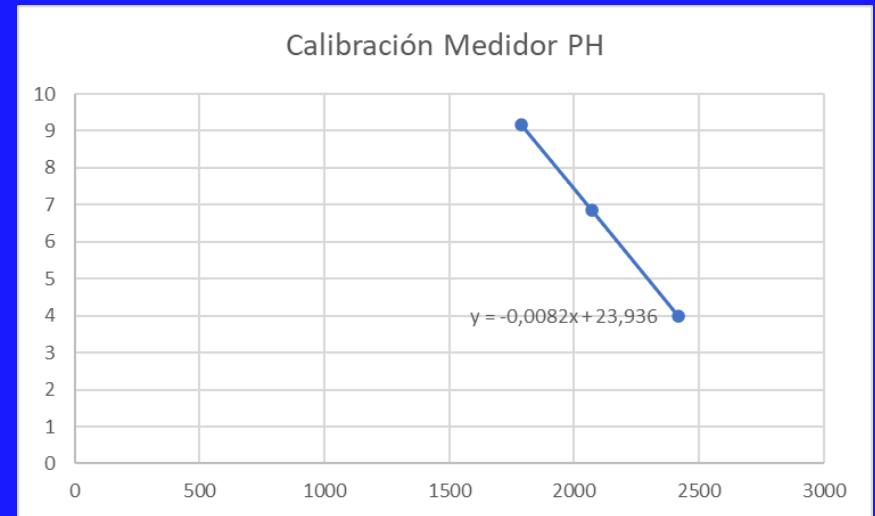
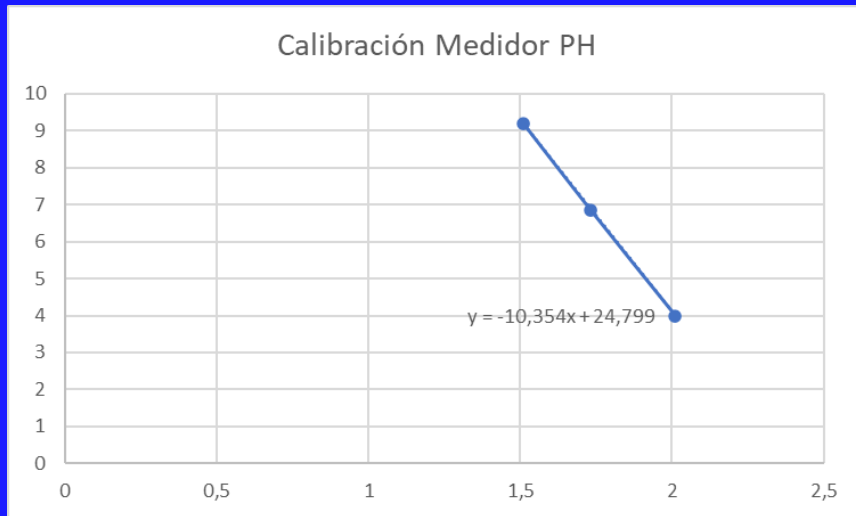
### Acondicionamiento de la señal



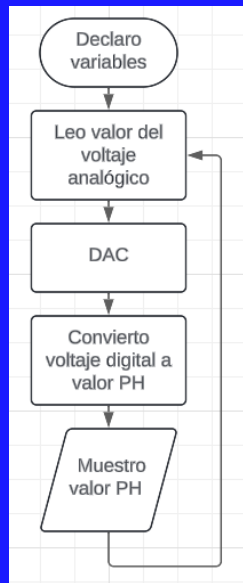
### Lectura del voltaje generado por cada una de las muestras



## Curvas de Calibración



## Diagrama de flujo del programa



## Parte del programa de calibración

```

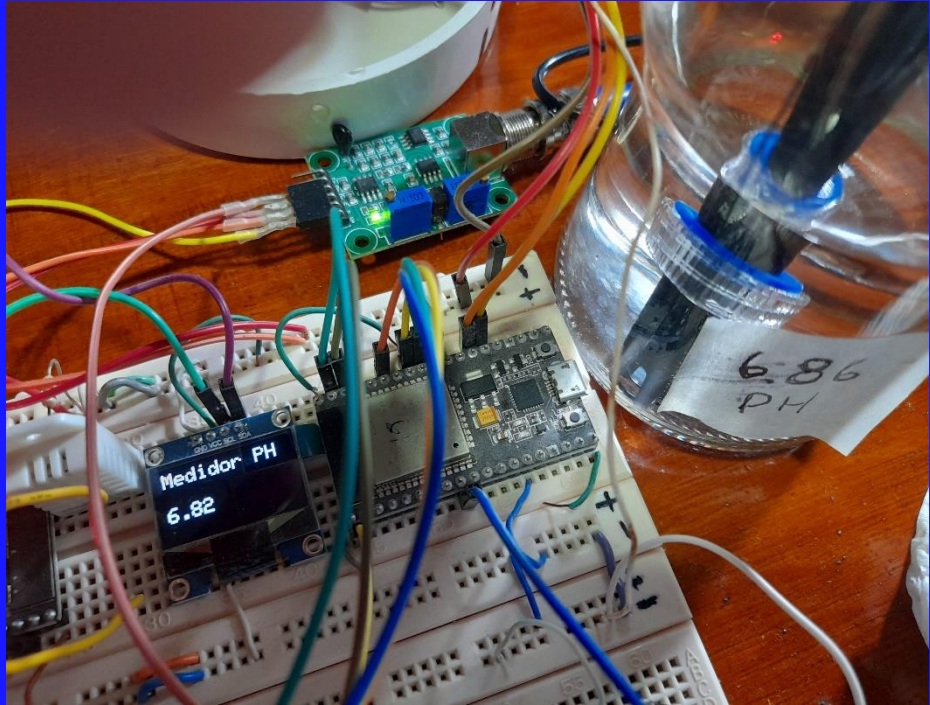
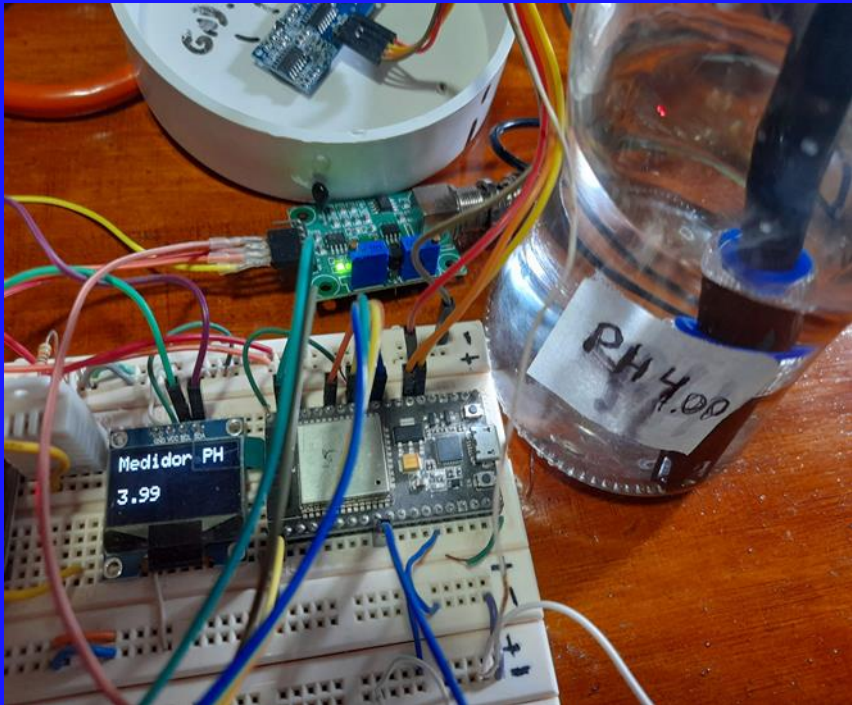
void setup () {
  Serial. begin ( 9600 ); //Iniciar el Monitor Ser
}

void loop () {
  float measure = analogRead (A0); //Leer pin A
  double voltage = measure* 5 / 1024 ; //Conversio
  Serial. println (voltage); //Imprimir el valor
}
  
```

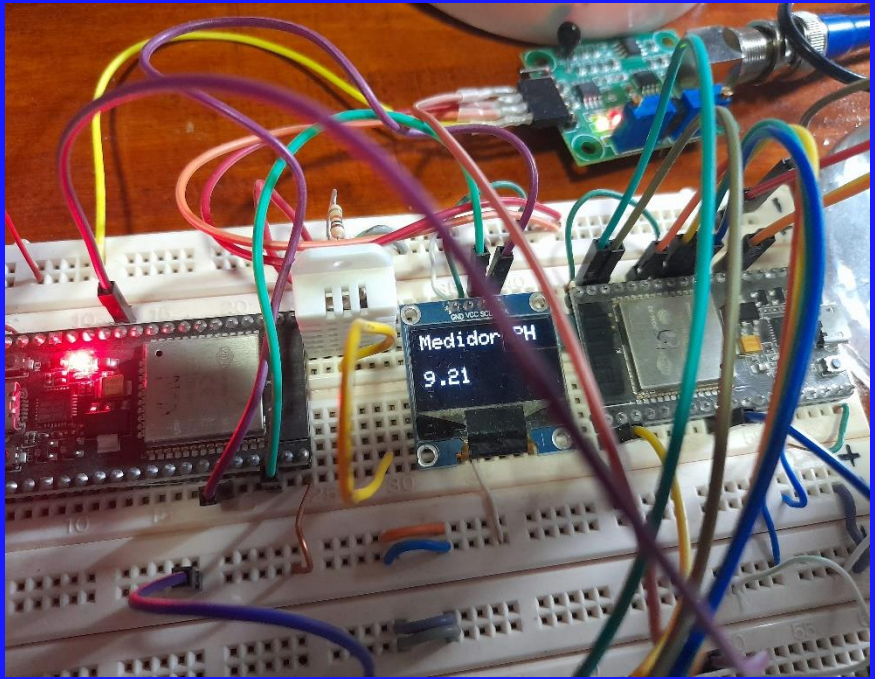
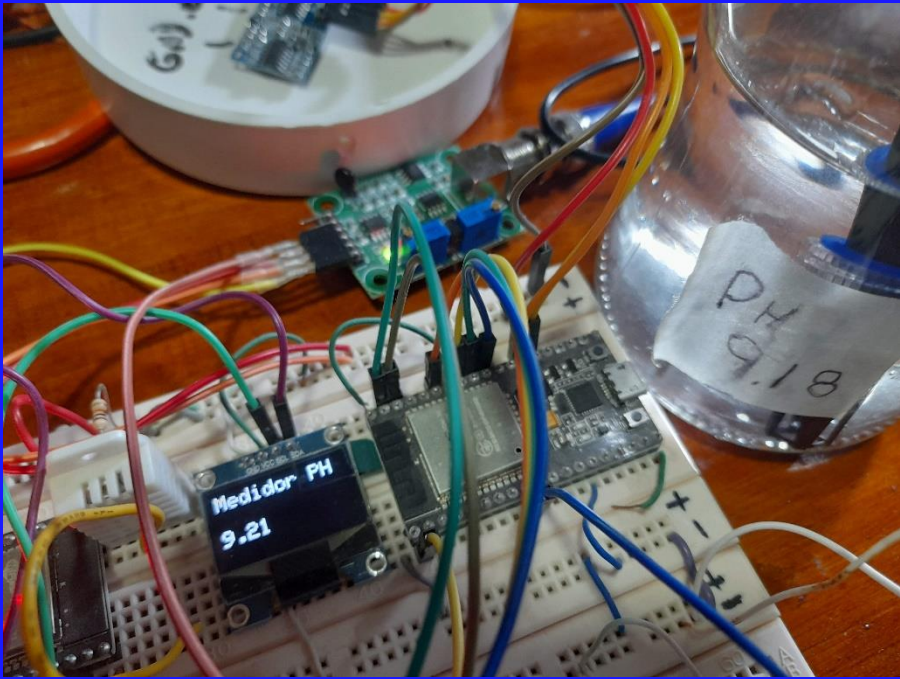
## Parte del programa de medida de PH

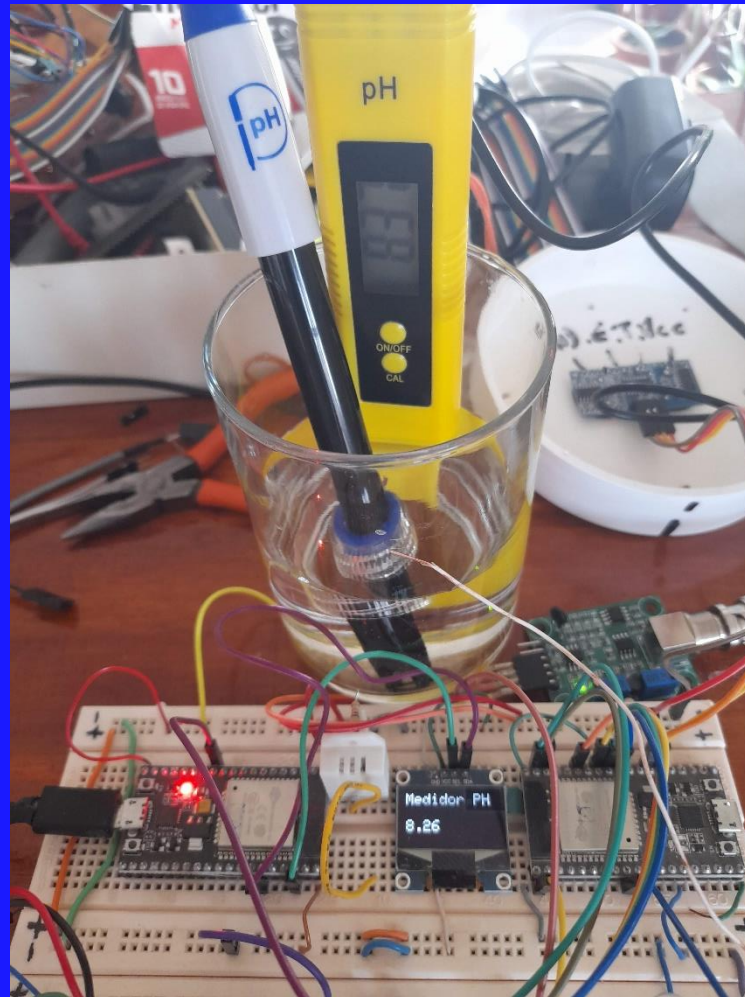
```

2   De serie.comenzar(9600); //Iniciar monitor serial
3   }
4
5   vacío bucle() {
6     flotarmedida =Lectura analógica(A0); //Leer el pin A0
7     doblevoltaje = medida*5/1024; //Conversión de analógico
8
9     //PH_step (unidad de voltaje/pH) = (voltaje @ PH7 - vo
10    flotarPH =7+((2.5- tegangan)/0,1841); //PH_sonda = PH7-
11    De serie.imprimir("PH:"); //Imprimir palabra pH en Seri
12    De serie.imprimirln(pH); //Imprimir el valor de pH en e
13
14    demora(1000); //Da un retraso de 1 segundo
15  }
  
```









Medición del PH de agua  
de consumo domiciliario



MUCHAS GRACIAS  
POR SU ATENCIÓN