



# Bebedero inteligente

Proyecto de:  
Julián Sciammaro, Valentino Videla

Objetivo:  
*comprender el funcionamiento del Arduino y el ESP32*

*Docente/s*  
*Juan Rosendo, José Luis Maldonado*

**2  
0  
2  
4**

**E  
E  
S  
T  
N  
6**

Institución educativa:  
Escuela Albert Thomas, calle 1 entre 57 y 58, Buenos Aires, Argentina



## ***Informe de Bebedero Automático***

### Resumen del Proyecto:

El proyecto "Bebedero Automático" tiene como objetivo automatizar el proceso de abastecimiento de agua en un recipiente, utilizando sensores de nivel de agua y un sensor ultrasónico para medir el volumen del agua. A través de un microcontrolador ESP32, se conecta a Wi-Fi para enviar alertas vía WhatsApp cuando el agua en el plato es insuficiente. La bomba de agua se activa automáticamente para rellenar el plato cuando es necesario. Este sistema mejora la comodidad y eficiencia en el manejo del agua para animales u otros usos.

### Componentes del Proyecto

## ESP32



Definición breve:

El ESP32 es un microcontrolador de 32 bits con capacidades de Wi-Fi y Bluetooth. Se utiliza para controlar el sistema, leer los sensores y enviar datos a través de internet.

Datasheet:

Voltaje: 3.3V

Frecuencia: 160 MHz

Memoria Flash: 4 MB

Conexiones: Wi-Fi y Bluetooth

Puertos: 34 pines GPIO, 12 pines analógicos, 2 UARTs, 1 I2C y 1 SPI

Protocolos de comunicación: Wi-Fi 802.11 b/g/n, Bluetooth 4.2



Sensor Ultrasonico Hc-sr04 Para  
Arduino Pic Robotica

Visitar >

Sensor

Ultrasonido HC-SR04

Definición breve:

El HC-SR04 es un sensor ultrasónico que mide distancias mediante ondas sonoras. Emite un pulso y mide el tiempo que tarda en reflejarse para calcular la distancia.

Datasheet:

Alcance: 2 cm a 4 metros

Precisión: 3 mm

Voltaje de funcionamiento: 5V

Corriente de funcionamiento: 15 mA

Frecuencia de operación: 40 kHz



Modulo sensor de nivel de agua

[Visitar >](#)

Sensor

de Nivel de Agua

Definición breve:

Este sensor detecta el nivel de agua en un recipiente. Al estar sumergido, altera la salida de voltaje, lo que permite detectar la presencia o ausencia de agua.

Datasheet:

Tipo: Sensor capacitivo

Voltaje: 5V

Corriente: 5-10 mA

Rango de medición: De 0 a 100%

Salida: Analógica



**Modulo Relay Rele De 1 Canal 5v 10a  
Arduino Ky-019 Robotica**

[Visitar >](#)

Módulo Relé 5V KY-019

Definición breve:

El módulo relé es un interruptor controlado electrónicamente que permite encender o apagar dispositivos de mayor voltaje, como la bomba de agua, al ser activado por el ESP32.

Datasheet:

Voltaje de operación: 5V

Corriente máxima: 10A

Tipo de relé: SPDT (Single Pole Double Throw)

Tensión máxima: 250V AC / 30V DC



Bomba de Agua 5V

Definición breve:

La bomba de agua de 5V es un dispositivo que se utiliza para mover agua de un lugar a otro. Se activa al recibir una señal del relé y permite rellenar el recipiente cuando el nivel de agua es bajo.

Datasheet:

Voltaje de operación: 5V

Corriente de funcionamiento: 150 mA

Capacidad: 120-150 litros por hora

Presión máxima: 0.5 bar

Funcionamiento de los Componentes en Conjunto

El funcionamiento de este sistema se basa en la interacción de varios componentes que trabajan juntos para lograr la automatización del llenado de agua.

1. Sensor de nivel de agua: Detecta si el agua ha bajado por debajo del umbral y envía una señal al ESP32.

2. ESP32: Recibe la señal del sensor de nivel de agua y procesa la información, activando el relé si es necesario. Además, realiza la medición de la distancia del agua mediante el sensor ultrasónico.



3. Sensor ultrasónico HC-SR04: Mide la distancia entre el sensor y el nivel del agua en el recipiente. Con esta información, el ESP32 calcula el porcentaje de agua presente en el recipiente.

4. Módulo Relé: Al recibir la señal del ESP32, activa o desactiva la bomba de agua. Si el nivel de agua es insuficiente, el relé activa la bomba para rellenar el recipiente.

5. Bomba de agua: Se enciende cuando el relé lo indica y comienza a bombear agua al recipiente para que el nivel alcance el valor deseado.

Además, si el sistema detecta que el nivel de agua es bajo, el ESP32 se conecta a Wi-Fi y envía un mensaje de alerta a través de WhatsApp, asegurando que el usuario esté informado en tiempo real.

Código final:

```
int relePin = 7;    // Pin conectado a IN del relé  
int sensorPin = A0; // Pin conectado a la salida del sensor de agua
```

```
int umbral = 300;    // Valor umbral para detectar la presencia de agua (ajusta este
valor según tu sensor)
```

```
void setup() {
  pinMode(relePin, OUTPUT);
  digitalWrite(relePin, LOW); // Inicializa el relé en estado apagado
  Serial.begin(9600);        // Inicializa la comunicación serial para depuración
}
```

```
void loop() {
  int lecturaSensor = analogRead(sensorPin); // Lee el valor del sensor de agua
  Serial.print("Nivel de agua: ");
  Serial.println(lecturaSensor);

  if (lecturaSensor < umbral) {    // Si el valor es menor al umbral, significa que no hay
agua
    digitalWrite(relePin, HIGH);    // Activa el relé (enciende la bomba)
    Serial.println("No hay agua, encendiendo bomba.");
  } else {
    digitalWrite(relePin, LOW);    // Apaga el relé (apaga la bomba)
    Serial.println("Hay suficiente agua, apagando bomba.");
  }

  delay(1000); // Espera 1 segundo antes de verificar nuevamente
}
```

Código del ESP32 con conexión a Wi-Fi y mensaje de alerta por WhatsApp:

```
#include <WiFi.h>

#include <Callmebot_ESP32.h>

#include <HTTPClient.h>


int relePin = 23;    // Pin conectado a IN del relé

int sensorPin = 32;  // Pin conectado a la salida del sensor de agua

int umbral = 300;    // Valor umbral para detectar la presencia de agua (ajusta este
                    // valor según tu sensor)


const int echo = 18;

const int trigger = 5;

long tiempo;

float distancia;

int porc = 0;


// Configuración WiFi

const char* ssid = "Juli";

const char* password = "jorgeroberto";


// Configuración de CallMeBot para enviar WhatsApp

String phoneNumber = "54221*****"; // Número de teléfono con código de país

String apiKey = "54*****";         // API Key de CallMeBot

String mensaje = "bebedero sin agua"; // Mensaje de alerta


void setup() {

    pinMode(relePin, OUTPUT);

    digitalWrite(relePin, LOW); // Inicializa el relé en estado apagado
```

```

Serial.begin(9600);      // Inicializa la comunicación serial para depuración

pinMode(echo, INPUT);
pinMode(trigger, OUTPUT);

// Conexión WiFi
WiFi.begin(ssid, password);
Serial.print("Conectando a WiFi...");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("\nConectado a WiFi");
}

void loop() {
    medir();      // Realiza la medición de distancia
    porcentaje(); // Calcula el porcentaje de distancia
    int lecturaSensor = analogRead(sensorPin); // Lee el valor del sensor de agua

    Serial.print("Nivel de agua: ");
    Serial.println(lecturaSensor);
    Serial.print("Distancia: ");
    Serial.print(distancia);
    Serial.print(" cm, Porcentaje: ");
    Serial.print(porc);
    Serial.println(" %");
}

```

```

// Control del relé según el umbral y el porcentaje del sensor ultrasónico

if (lecturaSensor < umbral && porc > 25) { // Solo activa el relé si el porcentaje es
mayor al 60%

    digitalWrite(relePin, HIGH);          // Activa el relé (enciende la bomba)

    Serial.println("No hay agua suficiente, encendiendo bomba.");

} else {

    digitalWrite(relePin, LOW);          // Apaga el relé (apaga la bomba)

    Serial.println("Suficiente agua o nivel bajo detectado, apagando bomba.");

}

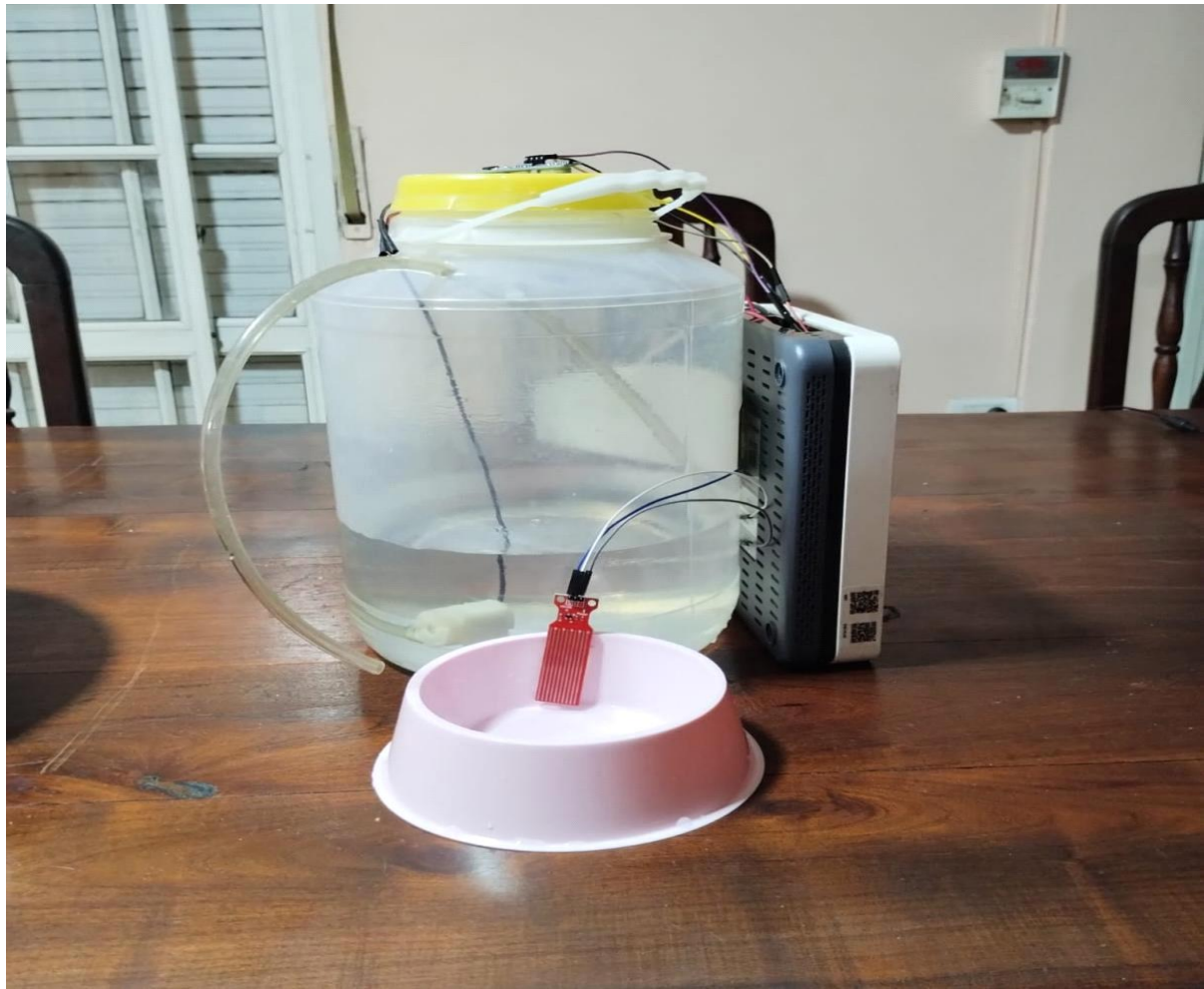
// Enviar mensaje de alerta si el porcentaje es menor o igual al 50%

if (porc <=

```

### Resultado Final del Proyecto:

El proyecto de “Beberero Automático” fue ensamblado y montado con éxito, cumpliendo con los objetivos propuestos. El sistema automatiza de manera eficiente el suministro de agua, activando la bomba de agua cuando el nivel es bajo y enviando alertas a través de WhatsApp cuando se requiere una acción del usuario. Los sensores y el ESP32 trabajaron de forma coordinada, garantizando un funcionamiento estable y fiable. Esto demuestra la viabilidad de la solución como un mecanismo práctico para mejorar la gestión de agua en recipientes, ofreciendo comodidad y automatización. Agradecemos a Juan Rosendo, Fabián Sciammaro, José Luis Maldonado por la orientación y apoyo en el desarrollo de este proyecto.



bebedero automático finalizado: