SMART WATER MANAGEMENT

INTRODUCTION:

In this IoT project, we will create a water level monitoring web server using an HC-SR04 Ultrasonic sensor and ESP32. It will be a contactless water management system. First, we will learn to interface HC-SR04 with ESP32. After that, we will see program our ESP32 board with the ultrasonic sensor to build our water monitor web server. The web server will show the current water level measured as the distance in cm by HC-SR04. So let’s begin!

Now, first let’s discuss various components of IoT base contactless water level monitoring system such as ultrasonic sensor, connection diagram with ESP32, and Arduino sketch. In the end, we will see a video demo.

MACHINE LEARNING ALGORITHM:

1.Explanation of the specific machine learning algorithm used for water consumption analysis

2.Data Collection: sources of data, including IOT devices sensors and data streams. Data quality and security considerations.

INTRODUCTION TO MATERIALS:

HC-SR04 Ultrasonic Sensor Introduction

To interface the HC-SR04 ultrasonic sensor with ESP32, we should know the functionality of each pin of the ultrasonic sensor. By knowing the functionality of input and output pins, we will be able to identify which GPIO pins of ESP32 should be used to interface with HC-SR04.

HC-SR04 Pinout

The pin configuration of an ultrasonic sensor consists of four pins namely VCC, Ground, Trigger, and Echo pin.

VCC and Ground are used to power sensor. We should supply 5 volts to the VCC pin and connect the GND pin with the ground terminal of the power supply.

Trigger: It is an input pin. Trigger pin is used to initiate the ultrasonic sensor to start distance measurement or distance ranging. When users want to get distance measurements from the sensor, we apply a 10μs pulse to this pin.

Echo: This is a pulse output pin. The echo pin produces a pulse as an output. The width of pulse or on-time of the pulse depends on the distance between the ultrasonic sensor and the obstacle which is placed in front of the HC-SR04 sensor. In idle conditions, this pin remains at an active low level.

Further details on ultrasonic sensor working are provided in the next section.

How HC-SR04 Sensor Works?

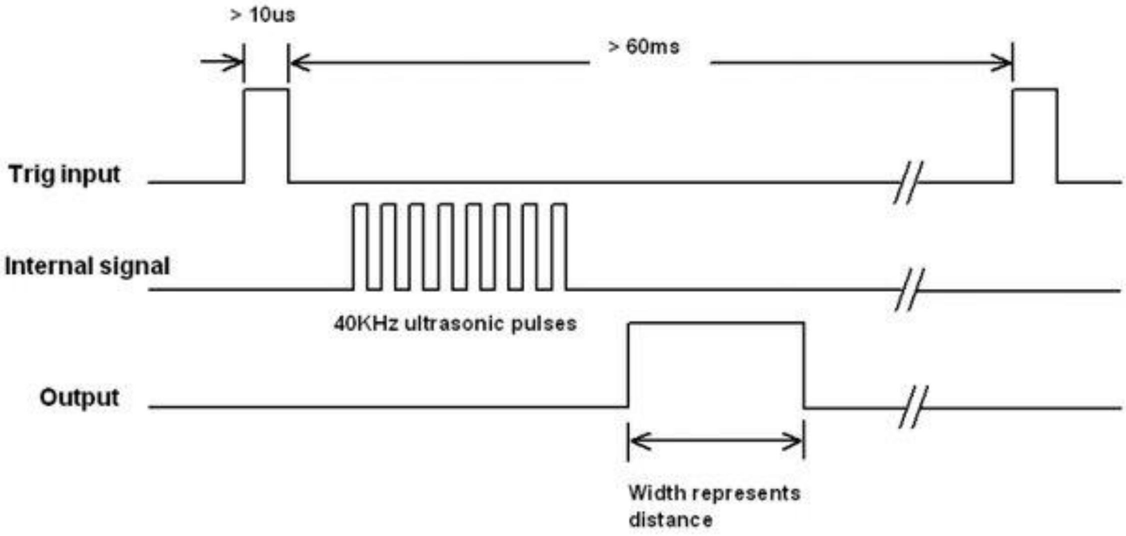
HC-SR04 ultrasonic sensor measures distance by using inaudible ultrasonic sound waves of 40KHz frequency. Like sound waves, ultrasonic waves travel through the air and if there is any obstacle in front of them, they reflect according to their angle of incidence. Moreover, if an object is placed parallel to an ultrasonic transmitter, ultrasonic waves reflect exactly at an angle of 180 degrees. Therefore, for distance measurement with HC-SR05 sensor, we place the object under test exactly in a parallel position with an ultrasonic sensor.

HC-SR05 ultrasonic sensor consists of two basic modules such as ultrasonic transmitter and ultrasonic receiver module. The transmitter circuit converts an electrical signal into a 40KHz burst of 8 sonar wave pulses. The input electrical signal to the transmitter circuit is 10μs pulse input to the trigger pin of the HC-SR04 sensor. As we mentioned earlier, we apply this trigger input signal through ESP32 or any microcontroller. On the other hand, the ultrasonic receiver circuit listens to these ultrasonic waves which are produced by the transmitter circuit.

Measure HC-SR04 Echo Pulse Time with ESP32

- To start ranging with HC-SR04, first, we apply 10μs pulse to the trigger pin of the HC-SR04 sensor from the ESP32 digital output pin.
- As soon as 10μs input trigger signal becomes active low, the transmitter circuit produces a burst of 8 ultrasonic sonar pulses. At the same time, the Echo pin also makes a transition from a logic low level to a logic high level.
- When the Echo pin goes high, we start to measure time with the ESP32 duration measurement function.
- These waves travel through the air and if there is any object placed in parallel to the sensor, these waves reflect back after a collision with the object.
- As soon as the ultrasonic waves received by the receiver circuit after striking with an object, the echo pin goes low. ESP32 detects this transition of echo output signal from active high to an active low level and stops the measurement.

In short, by measuring the on-time of the Echo output pulse signal, we can measure the distance.



The duration for which the echo output signal remains high depends on the distance between the ultrasonic sensor and the object which we place in front of the sensor. Higher is the distance, the higher the time sonar waves will take to reach back to the ultrasonic receiver circuit. Because ultrasonic waves travel through the air with the speed of sound and speed remains constant.

How to Convert Time Duration into Distance

In the next section, we will see how to measure pulse duration using ESP32. Let's assume that we have measured the output pulse on time (t) with ESP32. Now the question is how to convert this measured time into distance.

Well, this is the most obvious part of this tutorial. we can convert the pulse duration (t) into the distance (S) using this equation. Distance (S) = Speed (v) * t

Here v is the speed of ultrasonic waves in air. The speed of ultrasonic waves in air is equal to the speed of sound which is 340 m/s (meter per second).

The above equation will give distance output in units of meter. But, if you want the distance in centimetre units, multiply 340 with 100. Hence, the above equation becomes, s = 34000 * t

The time given in the above formula should also be divided by two. Because ultrasonic waves travel from the transmitter to the obstacle and then reflect back to the receiver circuit by traveling the same distance. We want to find the distance between HC-SR04 and the object only. Therefore, the formula to calculate distance Becomes, $s= 17000 * t$

How to Interface HC-SR04 Ultrasonic sensor with ESP32

Until now we have seen the working of the ultrasonic sensor and the pin details. Now we know that to interface an HC-SR04 sensor with ESP32, we need four pins out of which two are power supply pins and two are digital input output pins. One GPIO pin of the ESP32 will be used as a digital output pin to provide a trigger signal to the ultrasonic sensor. Similarly, one GPIO pin will be used as a digital input pin to capture echo output signal of output sensor.

Now make the connection of the ESP32 with the HC-SR04 sensor according to this connection diagram. In this schematic diagram, we use the GPIO5 pin of ESP32 to provide a trigger signal and GPIO18 to capture the echo output pulse.

HC-SR04	ESP32
VCC	Vin
GND	GND
Trigger	GPIO5
Echo	GPIO18

We will set up the ultrasonic sensor on top of a container that will carry the water. This way we will be able to determine the water level by obtaining the distance (cm) from the sensor sending it over to the ESP32 water level monitor web server.

CONCLUSION:

Summarize the key points of the document and emphasize the importance of smart water management.