**PHASE 3: Smart Water Management Based on IOT**

**IOT Devices:**

HC-SR04 Ultrasonic Distance Sensor

|  |  |
| --- | --- |
| **VCC** | **Voltage supply (5V)** |
| **ECHO** | **Measure the high pulse length to get the distance** |
| **GND** | **Ground.** |

Operations:

To start a new distance measurement set the TRIG pin to high for 10us or more. Then wait until the ECHO pin goes high, and count the time it stays high. The length of the ECHO high pulse is proportional to the distance.

* Centimeters – Pulse Micros/58
* Inches –Pulse Micros/148

**Relay Module:**

**Pin names:**

VCC - Supply voltage

GND - Ground

IN - Control signal (e.g. from micro-controller)

NC - Normally closed

COM - Common pin

**OPERATIONS:**

The relay is an electronic switch.

When the IN pin is high / disconnected, COM is connected to NC (NC means normally closed).

When the IN pin is low, COM is connected to NO (NO means normally open).

Setting the "transistor" attribute to "pnp" inverts the logic: when IN is high, COM is connected to NO, and when IN is low / disconnected, COM is connected to NC.

**\*ESP32:**

The ESP32 is a popular Wi-Fi and Bluetooth-enabled microcontroller, widely used for IOT Project The ESP32 is a popular Wi-Fi and Bluetooth-enabled microcontroller, widely used for IOT Projects. Wok wi simulates the ESP32, ESP32-C3, ESP32-S2, ESP32-S3, ESP32-C6 (beta), and ESP32-H2 (alpha).cts. Wok wi simulates the ESP32, ESP32-C3, ESP32-S2, ESP32-S3, ESP32-C6 (beta), and ESP32-H2 (alpha).

**\*Resistors**

**\*LED bulbs**

**\*Gates**

**Program:**

#define PIN\_TRIG 26

#define PIN\_ECHO 25

#define LOWLED 18

#define MIDLED 19

#define HIGHLED 21

#define MOTOR 27

unsigned int level = 0;

void setup() {

pinMode(LOWLED, OUTPUT);

pinMode(MIDLED, OUTPUT);

pinMode(HIGHLED, OUTPUT);

pinMode(MOTOR, OUTPUT);

digitalWrite(LOWLED, HIGH);

digitalWrite(MIDLED, HIGH);

digitalWrite(HIGHLED, HIGH);

digitalWrite(MOTOR, LOW);

Serial.begin(115200);

pinMode(PIN\_TRIG, OUTPUT);

pinMode(PIN\_ECHO, INPUT);

}

void loop() {

// Start a new measurement:

digitalWrite(PIN\_TRIG, HIGH);

delayMicroseconds(10);

digitalWrite(PIN\_TRIG, LOW);

// Read the result:

int duration = pulseIn(PIN\_ECHO, HIGH);

Serial.print("Distance in CM: ");

Serial.println(duration / 58);

Serial.print("Distance in inches: ");

Serial.println(duration / 148);

level = (duration / 58);

if(level < 100)

{

digitalWrite(LOWLED, LOW);

digitalWrite(MOTOR, HIGH);

digitalWrite(HIGHLED, HIGH);

digitalWrite(MIDLED, HIGH);

}

else if ((level > 200 ) && (level < 400))

{

digitalWrite(LOWLED, HIGH);

digitalWrite(HIGHLED, HIGH);

digitalWrite(MIDLED, LOW);

}

else if (level >= 400 )

{

digitalWrite(HIGHLED, LOW);

digitalWrite(MIDLED, HIGH);

digitalWrite(LOWLED, HIGH);

digitalWrite(MOTOR, LOW);

}

delay(1000);

}

**OUTPUT:**

Distance in CM: 89

Distance in inches: 34