SMART WATER FOUNDATION

Wokwi-project:

Downloaded from <https://wokwi.com/projects/378938059483297793>

Simulate this project on <http://wokwi.com>

**Sketch.ino**

**// Pin assignments**

**const int pirSensorPin = 23; // Pin connected to PIR motion sensor**

**const int ultrasonicTriggerPin = 33; // Pin connected to Ultrasonic sensor trigger**

**const int ultrasonicEchoPin = 12; // Pin connected to Ultrasonic sensor echo**

**const int switchRelayPin = 26;**

**int pirState = LOW;**

**int val = 0; // Pin connected to the switch relay**

**// Variables**

**bool isMotionDetected = false; // Flag to track motion detection**

**void setup() {**

**// Initialize pins**

**pinMode(pirSensorPin, INPUT);**

**pinMode(ultrasonicTriggerPin, OUTPUT);**

**pinMode(ultrasonicEchoPin, INPUT);**

**pinMode(switchRelayPin, OUTPUT);**

**digitalWrite(switchRelayPin, LOW); // Turn off the switch initially**

**// Other setup code for Ultrasonic sensor if needed**

**}**

**void loop() {**

**// Check PIR motion sensor**

**val = digitalRead(pirSensorPin); // read input value**

**if (val == HIGH) { // check if the input is HIGH**

**digitalWrite(switchRelayPin, HIGH); // turn LED ON**

**if (pirState == LOW) {**

**// we have just turned on**

**Serial.println("Motion detected!");**

**// We only want to print on the output change, not state**

**pirState = HIGH;**

**}**

**} else {**

**digitalWrite(switchRelayPin, LOW); // turn LED OFF**

**if (pirState == HIGH) {**

**// we have just turned of**

**Serial.println("Motion ended!");**

**// We only want to print on the output change, not state**

**pirState = LOW;**

**}**

**}**

**// Check Ultrasonic sensor**

**int distance = measureDistance();**

**// Control switch based on motion and distance**

**if (isMotionDetected || distance <= 100) {**

**digitalWrite(switchRelayPin, LOW); // Turn on the switch**

**} else {**

**digitalWrite(switchRelayPin, HIGH); // Turn off the switch**

**}**

**}**

**int measureDistance() {**

**// Send a pulse to the Ultrasonic sensor**

**digitalWrite(ultrasonicTriggerPin, LOW);**

**delayMicroseconds(2);**

**digitalWrite(ultrasonicTriggerPin, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(ultrasonicTriggerPin, LOW);**

**// Measure the duration of the pulse**

**long duration = pulseIn(ultrasonicEchoPin, HIGH);**

**// Calculate the distance based on the speed of sound**

**// (343 m/s or 0.0343 cm/µs)**

**int distance = duration \* 0.0343 / 2;**

**return distance;**

**}**

**Diagram.json:**

{

  "version": 1,

  "author": "Usha ",

  "editor": "wokwi",

  "parts": [

    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -110.5, "left": -14.6, "attrs": {} },

    {

      "type": "wokwi-pir-motion-sensor",

      "id": "pir1",

      "top": -216.8,

      "left": 194.22,

      "attrs": {}

    },

    {

      "type": "wokwi-led",

      "id": "led1",

      "top": 111.6,

      "left": -284.2,

      "attrs": { "color": "cyan" }

    },

    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -228.9, "left": -330.5, "attrs": {} },

    { "type": "wokwi-slide-switch", "id": "sw1", "top": 129.2, "left": 281.5, "attrs": {} }

  ],

  "connections": [

    [ "esp:TX0", "$serialMonitor:RX", "", [] ],

    [ "esp:RX0", "$serialMonitor:TX", "", [] ],

    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v0" ] ],

    [ "ultrasonic1:TRIG", "esp:D33", "green", [ "v0" ] ],

    [ "ultrasonic1:ECHO", "esp:D12", "green", [ "v0" ] ],

    [ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ],

    [ "led1:C", "esp:GND.2", "green", [ "v0" ] ],

    [ "led1:A", "esp:D26", "green", [ "v0" ] ],

    [ "pir1:VCC", "esp:3V3", "red", [ "v163.2", "h0", "v-86.4" ] ],

    [ "pir1:OUT", "esp:D23", "green", [ "v38.4", "h-153.74" ] ],

    [ "pir1:GND", "esp:GND.1", "black", [ "v0" ] ],

    [ "sw1:1", "esp:GND.1", "green", [ "v0" ] ],

    [ "sw1:2", "esp:D4", "green", [ "v0" ] ]

  ],

  "dependencies": {}

}

**Stimulation:**

