ASSIGNMENT -4

|  |  |
| --- | --- |
| **ASSIGNMENT DATE** | **20 OCTOBER 2022** |
| **PROJECT NAME** | **IOT Enabled smart farming** |
| **TEAM ID** | **PNT2022TMID48510** |
| **MAXIMUM MARKS** | **2 marks** |

**WRITE A CODE AND CONNECTION IN WOWKI FOR ULTASONIC SENSOR.WHENEVER DISTANCE IS**

**LESS THAN 100 CMS SEND “ALERT” TO IBM CLOUD AND DISPLAY IN DEVICE RECENT EVENTS CODE**

// Pins const int TRIG\_PIN = 7; const int ECHO\_PIN = 8;

// Anything over 400 cm (23200 us pulse) is "out of range" const unsigned int MAX\_DIST = 23200;

void setup() {

// The Trigger pin will tell the sensor to range find

pinMode(TRIG\_PIN, OUTPUT); digitalWrite(TRIG\_PIN, LOW);

//Set Echo pin as input to measure the duration of //pulses coming back from the distance sensor pinMode(ECHO\_PIN, INPUT);

// We'll use the serial monitor to view the sensor output

Serial.begin(9600);

}

void loop() { unsigned long tl; unsigned long t2; unsigned long pulse\_width; float cm;

float inches;

// Hold the trigger pin high for at least 10 us digitalWrite(TRIG\_PIN, HIGH); delayMicroseconds(10); digitalWrite(TRIG\_PIN, LOW); // Wait for pulse on echo pin while (digitalRead(ECHO\_PIN) == 0);

// Measure how long the echo pin was held high (pulse width)

// Note: the micros() counter will overflow after ~70 min

tl = micros();

while (digitalRead(ECHO\_PIN) == 1); t2 = micros(); pulse\_width = t2- tl;

// Calculate distance in centimeters and inches. The constants // are found in the datasheet, and calculated from the assumed speed

//of sound in air at sea level (~340 m/s).

cm = pulse\_width/58.0; inches = pulse\_width/148.0; // Print out results if (pulse\_width > MAX\_DIST) { Serial.println("Out of range");

} else {

Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”);

Serial.print("The Measured Distance in cm: "); Serial.println(cm); if(cm<100){

// while(true)

{

Serial.println("Alert!!");

// }

}

Serial.print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

// Wait at least 1000ms before next measurement delay(1000);

}



