

ASSIGNMENT 4

Assignment Date	1 NOV 2022
Student Name	MANOJ R
Student Roll Number	312319104089
Maximum Marks	2 Marks

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

QUESTION:

Write code and connections in wokwi for the ultrasonic sensor.

Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

Upload document with wokwi share link and images of IBM cloud

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 t1;
  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
  t1 = micros();
  while ( digitalRead(ECHO_PIN) == 1);
  t2 = micros();
  pulse_width = t2 - t1;

  // 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("Measured Distance is : ");
Serial.print(cm);
Serial.println(" cm");

if(cm<100){
  // while(true){
  Serial.println("Alert ON !!");
  // }
}

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

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

```

OUTPUT:

