

ASSIGNMENT-4

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.

```
/*
 * Ultrasonic.cpp
 *
 * Library for Ultrasonic Ranging Module in a minimalist way
 *
 */

#if ARDUINO >= 100
  #include <Arduino.h>
#else
  #include <WProgram.h>
#endif

#include "Ultrasonic.h"

Ultrasonic::Ultrasonic(uint8_t trigPin, uint8_t echoPin, unsigned long timeOut) {
  trig = trigPin;
  echo = echoPin;
  threePins = trig == echo ? true : false;
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  timeout = timeOut;
}

unsigned int Ultrasonic::timing() {
  if (threePins)
    pinMode(trig, OUTPUT);

  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  if (threePins)
    pinMode(trig, INPUT);
```

```

        previousMicros = micros();
        while(!digitalRead(echo) && (micros() - previousMicros) <= timeout); // wait for
the echo pin HIGH or timeout
        previousMicros = micros();
        while(digitalRead(echo) && (micros() - previousMicros) <= timeout); // wait for
the echo pin LOW or timeout

    return micros() - previousMicros; // duration
}

/*
 * If the unit of measure is not passed as a parameter,
 * sby default, it will return the distance in centimeters.
 * To change the default, replace CM by INC.
 */
unsigned int Ultrasonic::read(uint8_t und) {
    return timing() / und / 2; //distance by divisor
}

/*
 * This method is too verbal, so, it's deprecated.
 * Use read() instead.
 */
unsigned int Ultrasonic::distanceRead(uint8_t und) {
    return read(und);
}

/*
 * Ultrasonic.h
 *
 * Library for Ultrasonic Ranging Module in a minimalist way
 *
 */

#ifndef Ultrasonic_h
#define Ultrasonic_h

/*
 * Values of divisors
 */
#define CM 28
#define INC 71

```

```

class Ultrasonic {
public:
    Ultrasonic(uint8_t sigPin) : Ultrasonic(sigPin, sigPin) {};
    Ultrasonic(uint8_t trigPin, uint8_t echoPin, unsigned long timeOut = 20000UL);
    unsigned int read(uint8_t und = CM);
    unsigned int distanceRead(uint8_t und = CM) __attribute__((deprecated ("This
method is deprecated, use read() instead.")));
    void setTimeout(unsigned long timeOut) {timeout = timeOut;}
    void setMaxDistance(unsigned long dist) {timeout = dist*CM*2;}

private:
    uint8_t trig;
    uint8_t echo;
    boolean threePins = false;
    unsigned long previousMicros;
    unsigned long timeout;
    unsigned int timing();
};

#endif // Ultrasonic_h

{
    "version": 1,
    "author": "Rozen Berg",
    "editor": "wokwi",
    "parts": [
        {
            "type": "wokwi-arduino-uno",
            "id": "uno",
            "top": 259.31,
            "left": 31.06,
            "rotate": 0,
            "hide": false,
            "attrs": {}
        },
        {
            "type": "wokwi-hc-sr04",
            "id": "ultrasonic",
            "top": 86.99,
            "left": 109.89,
            "rotate": 0,
            "hide": false,
            "attrs": { "distance": "100" }
        }
    ]
}

```

```

    ],
    "connections": [
      [ "uno:GND.1", "ultrasonic:GND", "black", [ "v-8", "*", "v8" ] ],
      [ "uno:13", "ultrasonic:ECHO", "green", [] ],
      [ "uno:12", "ultrasonic:TRIG", "purple", [ "*", "v4" ] ],
      [ "uno:5V", "ultrasonic:VCC", "red", [ "v16", "h-96", "*", "v12" ] ]
    ]
  }
}

```

/*

Ultrasonic Simple

Prints the distance read by an ultrasonic sensor in centimeters. They are supported to four pins ultrasound sensors (like HC-SC04) and three pins (like PING)) and Seeed Studio sensors).

The circuit:

* * Module HC-SC04 (four pins) or PING))) (and other with three pins), attached to digital pins as follows:

```

-----
| HC-SC04 | Arduino |   | 3 pins | Arduino |
-----
| Vcc | 5V |   | Vcc | 5V |
| Trig | 12 | OR | SIG | 13 |
| Echo | 13 |   | Gnd | GND |
| Gnd | GND |   | -----
-----

```

*/

```
#include "Ultrasonic.h"
```

/*

Pass as a parameter the trigger and echo pin, respectively, or only the signal pin (for sensors 3 pins), like:

```
Ultrasonic ultrasonic(13);
```

*/

```
Ultrasonic ultrasonic(12, 13);
```

```
int distance;
```

```
void setup() {
  Serial.begin(9600);
}
```

```
void loop() {
```

```
// Pass INC as a parameter to get the distance in inches
```

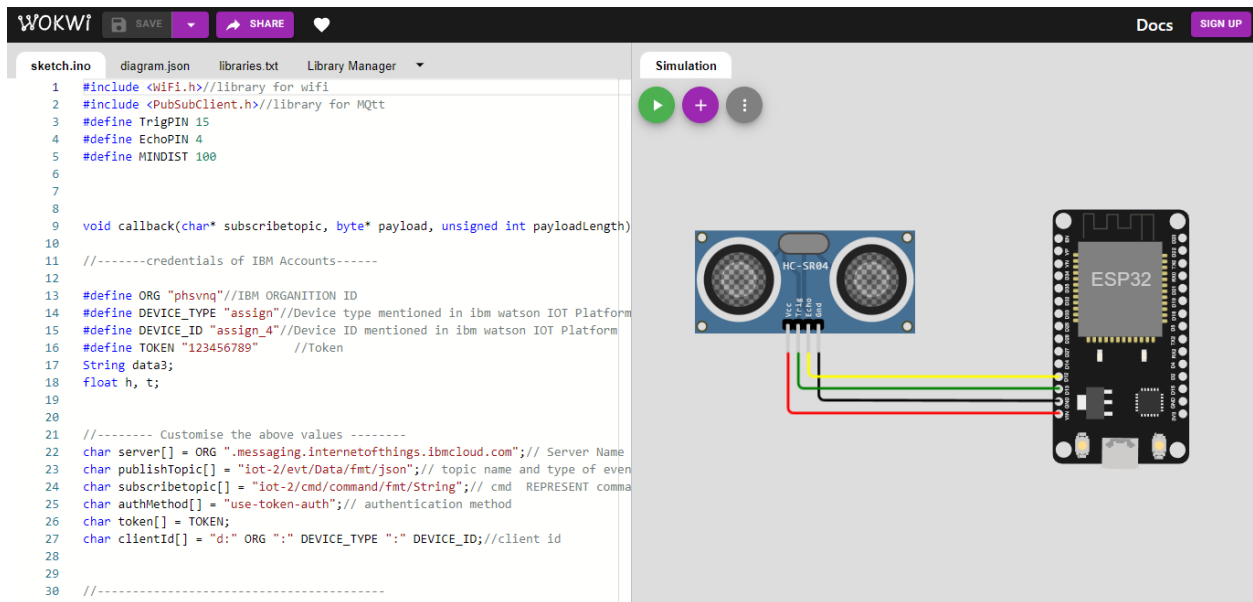
```
distance = ultrasonic.read(CM);
```

```
Serial.print("Distance in CM: ");  
Serial.println(distance);
```

```
distance = ultrasonic.read(INC);
```

```
Serial.print("Distance in Inches: ");  
Serial.println(distance);
```

```
delay(1000);  
}
```



Browse Action Device Types Interfaces

assign_4 R Disconnected assign Device Oct 26, 2022 8:31 PM → ...

Identity **Device Information** Recent Events State Logs ✕

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"MESSAGE":"ALERT!"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT!"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT!"}	json	3 minutes ago
Data	{"MESSAGE":"ALERT!"}	json	3 minutes ago
Data	{"MESSAGE":"ALERT!"}	json	4 minutes ago