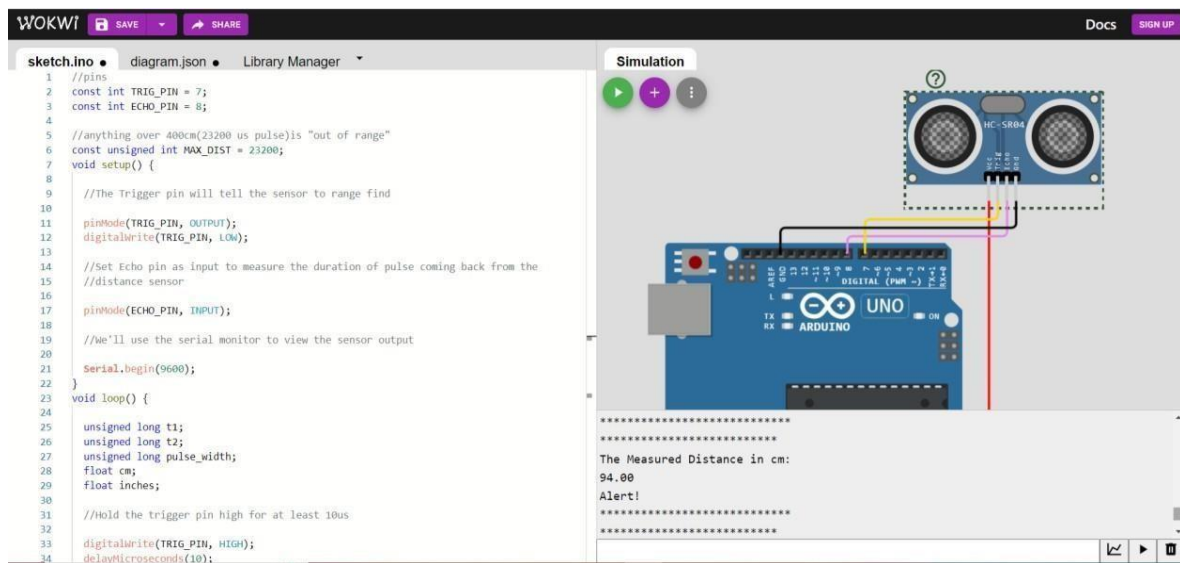


ASSIGNMENT-4

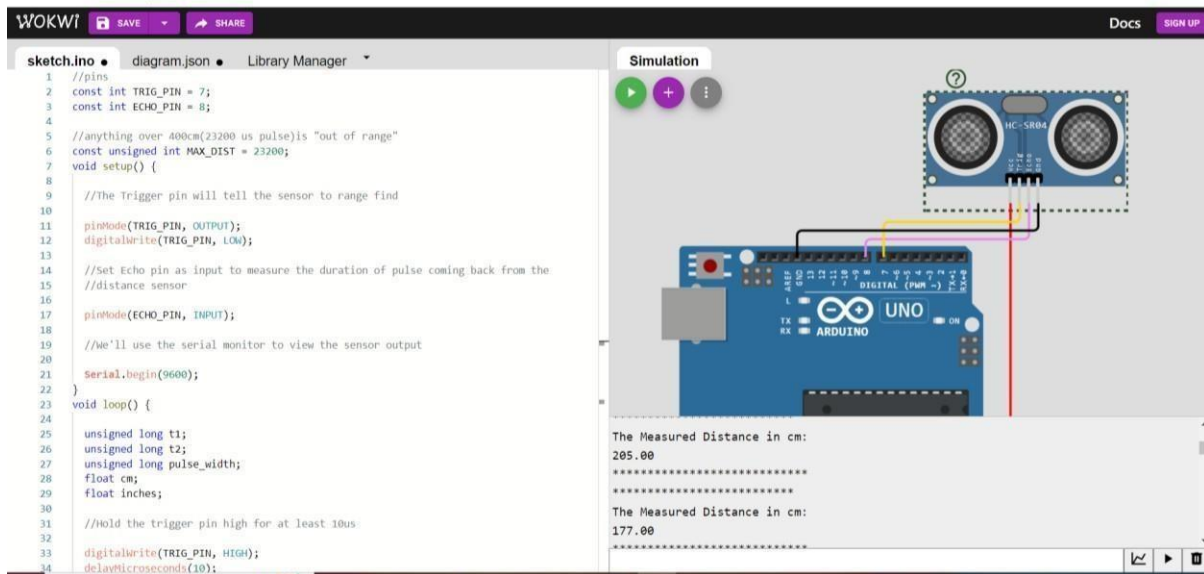
QUESTION:

Write Code and connections in wokwi for ultrasonic sensor. whatever distance is less than 100 cm send "Alert" to IBM cloud and display in device recent events.

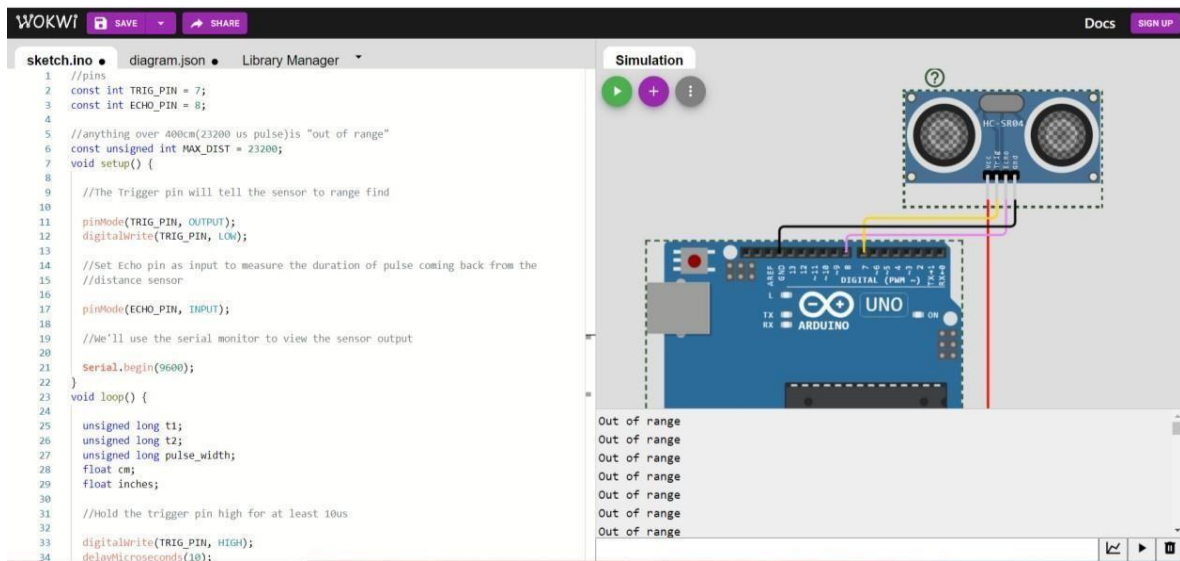
CASE 1: Distance less than 100cm → It Alerts



CASE 2: Distance more than 100cm → It won't Alert



CASE 3: Beyond limits → Out of Range



CODING:

```

//pins      const    int
TRIG_PIN = 7; const int
ECHO_PIN = 8;

```

```

//anything over 400cm(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 pulse 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 10us

    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 ~70min
    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(~340m/s)
    cm = pulse_width / 58; inches =
    pulse_width / 148.0;

    //print out results

    if (pulse_width > MAX_DIST) {

        Serial.println("Out of range");
    } else

```

```

{
  Serial.println("*****");
  Serial.println("The Measured Distance in cm:"); Serial.println(cm);
if (cm < 100)
{
  //while (true)
  {
    Serial.println("Alert!");
  }
}
  Serial.println("*****");
}

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

```

CIRCUIT:

