ASSIGNMENT-4

NAME : SANTHOSH KUMAR.M.

REG.NO : 110719106030

OBJECTIVE

Write Code and connections in wokwi for ultrasonic sensor. whatever distance is less than 100 cms send “Alert” to ibm cloud aand display indevicerecent events.

SOLUTION

```
//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
 Pin Mode(TRIG_PIN, OUTPUT);
 digital Write(TRIG_PIN, LOW);
```

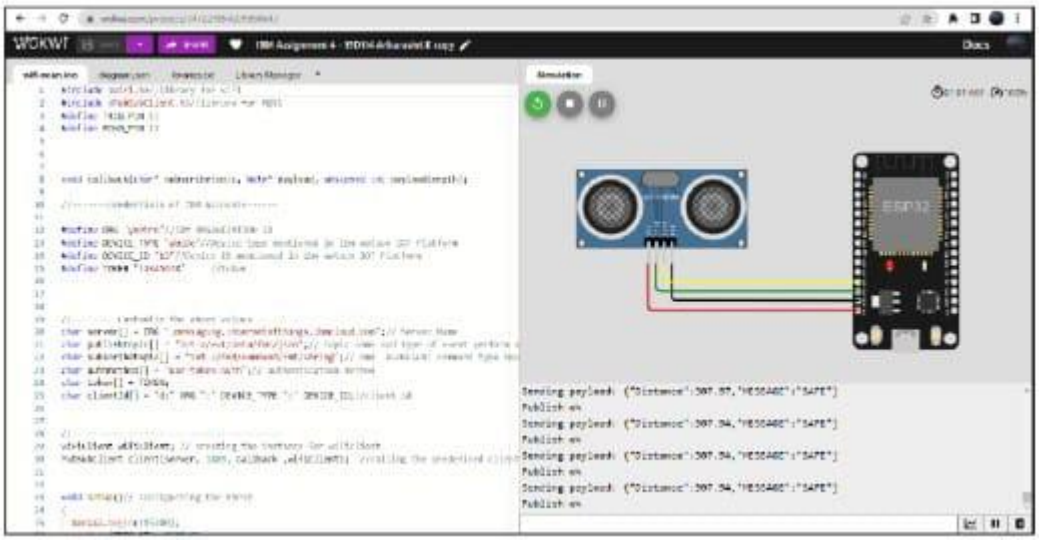
```
//Set Echo pin as input to measure the duration of
//pulses coming back from the distance sensor
pinMode(ECHO_PIN, INPUT ) ;
 // We'll use the serial monitor to view the sensor output
Serial.begin(9600);
 }
void loop()
{ unsigned long
 t1;
unsigned long t2;
unsigned long pulse_width;
float cm;
 float 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 (- 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.print("The Measured Distance in cm: ");
Serial.println(cm);
 if( cm < 100 )
{ //while(true)
{ Serial.println(“Alert!!”); //
}
```

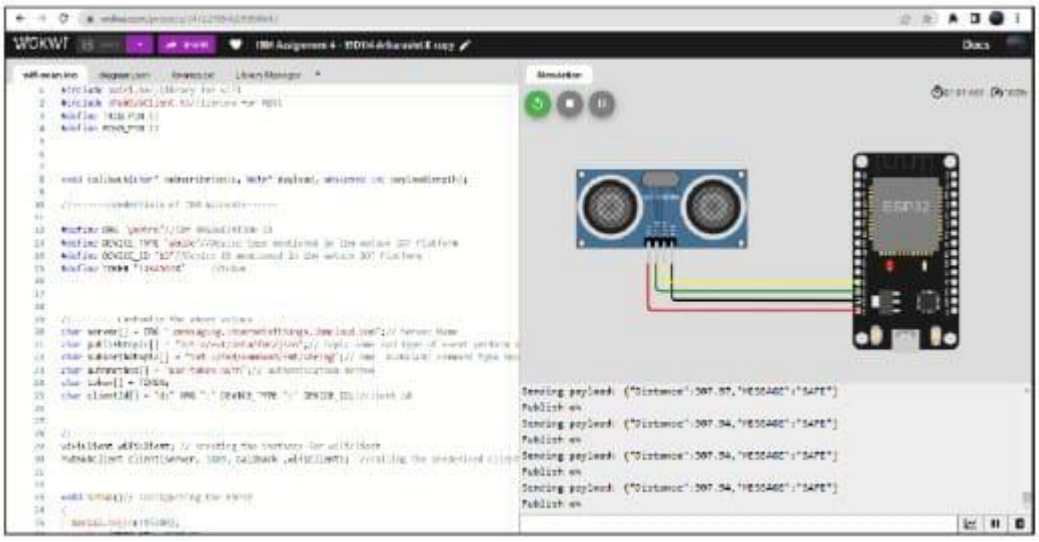
```
 }
Serial.print("**********************************
*"); }
 //wait at least 1000ms before next measurement
Delay(1000);
}
```

OUTPUT

1.If the distance is less than 100cm, it alerts.

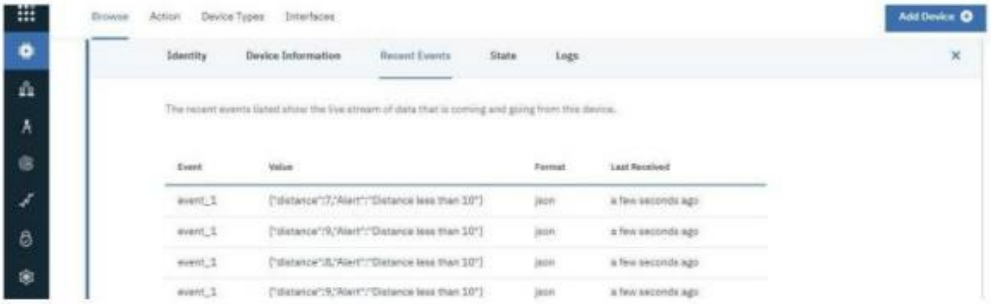


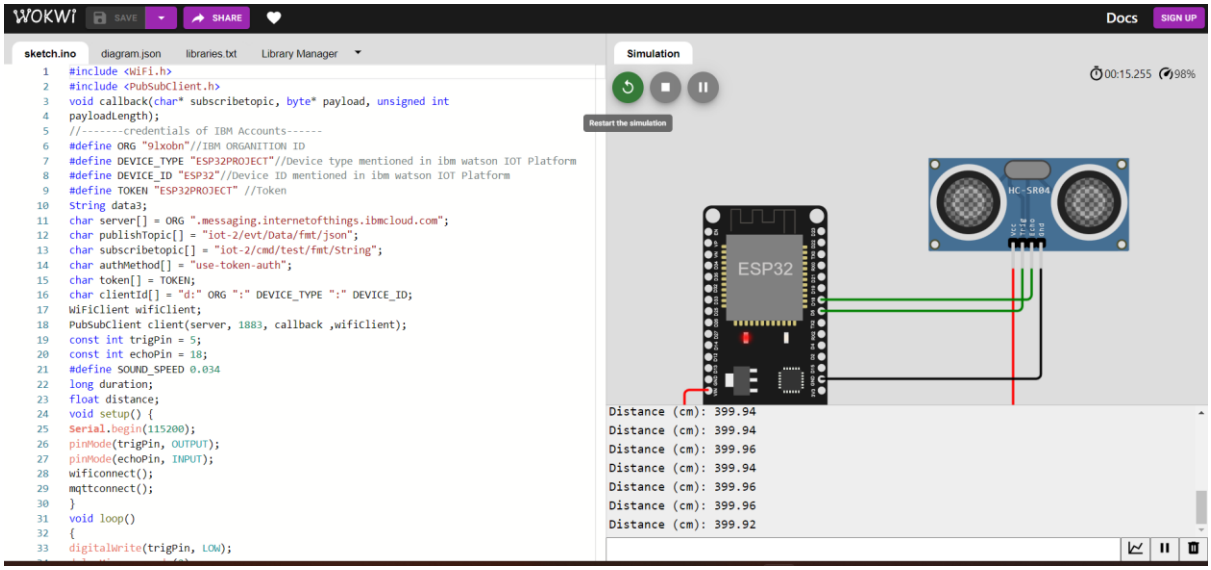
2. If the distance is more than 100 cm, it won't alert.



3. Simulation and code execution

IBM cloud output:






```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "9lxobn"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32PROJECT"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "ESP32"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "ESP32PROJECT" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wificlient;
PubSubClient client(server, 1883, callback ,wificlient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
```