Professional Readiness for Innovation, Employability and Entrepreneurship

Smart Farmer -IoT Enabled Smart Farming Application

ASSIGNMENT – 4

NAME: Arun Kumar S

Write Code and connections in wokwi for ultrasonic sensor. Whatever distance is less than 100 cm send "Alert" to ibm cloud and display in devicerecent 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
 mint1= 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
measurementDelay(1000);
}
```

OUTPUT :

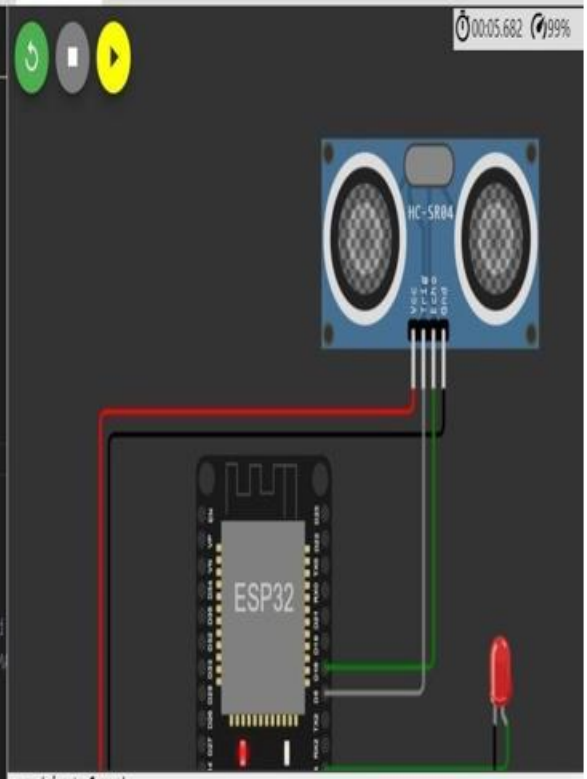
When object is not near to the ultrasonic sensor

sketch.ino | diagram.json | libraries.txt | Library Manager

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt


void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "f59trs"//IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonicsensor"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "distancedetection"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "AlGMGaaF01nawa1QA3"     //Token
String data3;
float dist;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMM
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wificlient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wificlient); //calling the predefined client id

int LED = 4;
int trig = 5;
int echo = 18;
void setup()
```

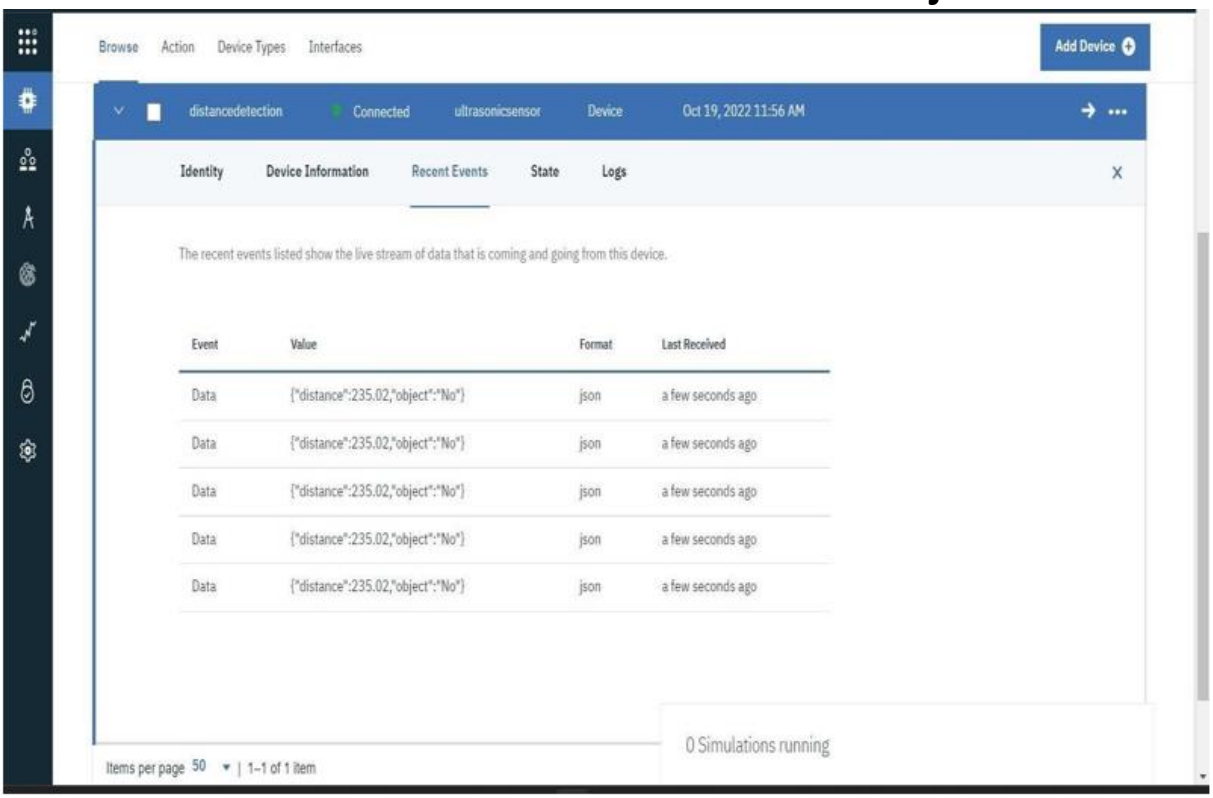
Simulation

00:05.682 99%

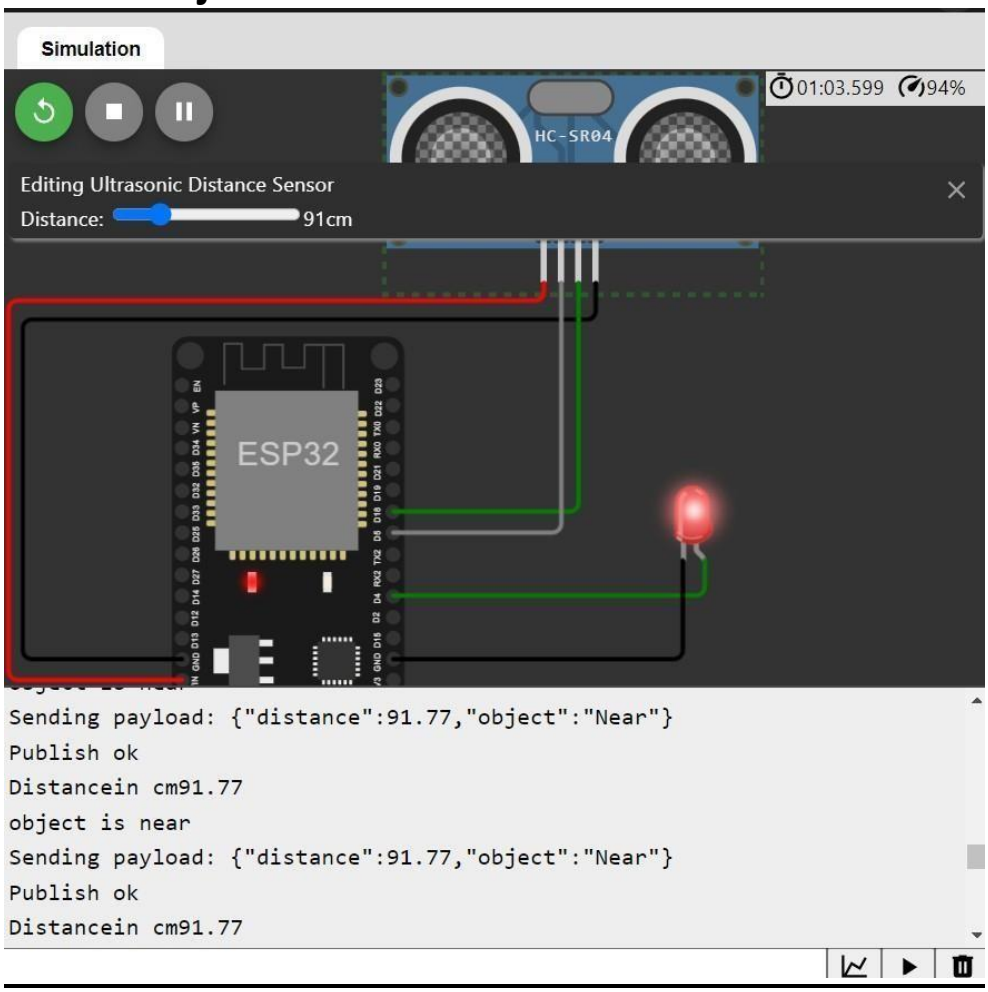



```
no object found
Sending payload: {"distance":403.45,"object":"No"}
Publish ok
Distancein cm233.00
no object found
Sending payload: {"distance":233.00,"object":"No"}
Publish ok
```

Data sent to the IBM cloud device when the object is far



When object is nearer to the ultrasonic sensor



Data sent to the IBM cloud device when the object is near