

# Assignment 4

Project Link: <https://wokwi.com/projects/290056311044833800>

Code:

```
#define ECHO_PIN 2

#define TRIG_PIN 3

#define organization "kv09p4"

#define device type "Groot"

#define deviceId="13"

#define authmethod="token"

#define authToken="123456789"

void setup() {

  Serial.begin(115200);

  pinMode(LED_BUILTIN, OUTPUT);

  pinMode(TRIG_PIN, OUTPUT);

  pinMode(ECHO_PIN, INPUT);

}

float readDistanceCM() {

  digitalWrite(TRIG_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG_PIN, LOW);

  int duration = pulseIn(ECHO_PIN, HIGH);

  return duration * 0.034 / 2;

}

void loop(){

  float distance=readDistanceCM();

  if(distance<=100)

  {

    Serial.println("person detected");

  }

  else{

    Serial.print("Measured distance:");
```

```

Serial.println(readDistanceCM());
}

delay(1000);
}

```

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area displays a table of devices. The first device is highlighted with ID 13, Status 'Disconnected', Device Type 'Groot', Class ID 'Device', and Date Added 'Oct 28, 2022 9:25 PM'. Below the table, a detailed view for device 13 is shown, including its Identity, Device Information, Recent Events, State, and Logs. The Device Information section lists: Device ID: 13, Device Type: Groot, Date Added: Oct 28, 2022 9:25 PM, Added By: 613519106030@smartinternz.com, and Connection Status: Disconnected. At the bottom, a status bar indicates '1 Simulation running'.

This screenshot shows the 'Recent Events' tab for device 13. The table lists several events, all named 'event\_1', with values like '{"distance":110}', '{"distance":91}', '{"distance":83}', '{"distance":110}', and '{"distance":38}'. The format for all events is 'json'. A modal window titled 'Device Type: Groot' is open, allowing the creation of a new event type. The modal includes fields for 'Event type name' (set to 'event\_1'), a 'Schedule' (set to 'Every Minute'), and a 'Payload' editor. The payload is a JSON object: 

```
{
  "distance": random(10, 120)
}
```

. The modal also has buttons for 'Send', 'Upload a CSV file', 'Cancel', and 'Save'.

IBM Watson IoT Platform

Ultrasonic

Line chart

15  
10  
5  
0

23:02:20 23:02:30 23:02:40 23:02:50 23:03 23:03:10

1 minute now

distance

Device Type: Groot

Events 1

New event type

Event type name event\_1 Send

Schedule 1 Every Minute

Payload

Specify the event payload in the editor window or by uploading a CSV file.

```
0 {  
1 "distance": random(10, 120)  
2 }  
3 }
```

Upload a CSV file

What functions can I apply?

Cancel Save

WOKWI

SAVE SHARE

hc-sr04.ino by urish

diagram.json Library Manager

```
6 #define ECHO_PIN 2  
7 #define TRIG_PIN 3  
8 #define organization ="kv09p4"  
9 #define device type ="Groot"  
10 #define deviceId="13"  
11 #define authmethod="token"  
12 #define authToken="123456789"  
13 void setup() {  
14   Serial.begin(115200);  
15   pinMode(LED_BUILTIN, OUTPUT);  
16   pinMode(TRIG_PIN, OUTPUT);  
17   pinMode(ECHO_PIN, INPUT);  
18 }  
19 float readDistanceCM() {  
20   digitalWrite(TRIG_PIN, LOW);  
21   delayMicroseconds(2);  
22   digitalWrite(TRIG_PIN, HIGH);  
23   delayMicroseconds(10);  
24   digitalWrite(TRIG_PIN, LOW);  
25   int duration = pulseIn(ECHO_PIN, HIGH);  
26   return duration * 0.034 / 2;  
27 }  
28 void loop(){  
29   float distance=readDistanceCM();  
30   if(distance<=100)  
31   {  
32     Serial.println("person detected");  
33   }  
34   else{  
35     Serial.print("Measured distance:");  
36     Serial.println(readDistanceCM());  
37   }  
38   delay(1000);  
39 }
```

Simulation

00:22.280 99%

Measured distance:177.34  
Measured distance:177.24  
Measured distance:177.40  
Measured distance:177.24  
Measured distance:177.33  
Measured distance:177.24  
Measured distance:177.33