

NAME: SARAN.A
REGISTER NUMBER: 714019106096

A

S

S

I

G

N

M

E

N

T

4

Ultrasonic sensor simulation in Wokwi

Question :

Write a code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100cms send an "Alert" to IBM cloud and display in the device recent events.

Code:

```
#  
i  
n  
c  
l  
u  
d  
e
```

```
<  
W  
i  
F  
i  
.  
h  
>
```

```
#  
i  
n  
c  
l  
u  
d  
e
```

```
<  
P  
u  
b  
S  
u  
b  
C  
l  
i  
e
```

```

n
t
.
h
>
void callback(char*
subscribetopic, byte* payload,
unsigned intpayloadLength);
//-----credentials of IBM Accounts-----
#define ORG "kotoq5"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type
mentioned in ibm watson IOT Platform#define
DEVICE_ID "12345"//Device ID mentioned in ibm
watson IOT Platform #define TOKEN "12345678"
//Token
String data3;
char server[] = ORG
".messaging.internetofthings.ibm
cloud.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;
c
o
n
s
t

i
n
t

e

```

c
h
o
p
i
n

=

1
8
;

d
e
f
i
n
e

S
O
U
N
D

—
S
P
E
E
D

0
.
0
3
4

l
o
n
g

d

u
r
a
t
i
o
n
;
f
l
o
a
t

d
i
s
t
a
n
c
e
;

v
o
i
d

s
e
t
u
p
(
)

{
S
e
r
i
a
l
.

```
b  
e  
g  
i  
n  
(  
1  
1  
5  
2  
0  
0  
)  
;
```

```
p  
i  
n  
M  
o  
d  
e  
(  
t  
r  
i  
g  
p  
i  
n  
,
```

```
O  
U  
T  
P  
U  
T  
)  
;
```

```
p  
i  
n  
M
```

o
d
e
(
e
c
h
o
p
i
n
,

I
N
P
U
T
)
;

w
i
f
i
c
o
n
n
e
c
t
(
)
;

m
q
t
t
c
o
n
n
e


```
c
t
(
)
;
}
void loop()
{
  d
  i
  g
  i
  t
  a
  l
  W
  r
  i
  t
  e
  (
  t
  r
  i
  g
  g
  e
  r
  ,
  L
  O
  W
  )
  ;
  d
  e
  l
  a
  y
  M
  i
  c
  r
  o
  s
```

```
e
c
o
n
d
s
(
2
)
;
d
i
g
i
t
a
l
W
r
i
t
e
(
t
r
i
g
p
i
n
,
H
I
G
H
)
;
d
e
l
a
y
M
i
c
```

```

r
o
s
e
c
o
n
d
s
(
1
0
)
;
d
i
g
i
t
a
l
W
r
i
t
e
(
t
r
i
g
p
i
n
,
L
O
W
)
;
d
u
r
a
t

```

i
o
n
=
p
u
l
s
e
I
n
(
e
c
h
o
p
i
n
,
H
I
G
H
)
;
d
i
s
t
a
n
c
e
=
d
u
r
a
t
i
o
n
*
S

O
U
N
D
-
S
P
E
E
D
/
2
;
S
e
r
i
a
l
.
p
r
i
n
t
(
"
D
i
s
t
a
n
c
e
(
c
m
)
:
"
)
;
S
e

```

r
i
a
l
.
p
r
i
n
t
l
n
(
d
i
s
t
a
n
c
e
)
;
i
f
(
d
i
s
t
a
n
c
e
<
1
0
0
)
{
S
e
r
i
a

```

```
1
.
p
r
i
n
t
l
n
(
"
A
L
E
R
T
!
!
"
)
;

d
e
l
a
y
(
1
0
0
)
;

```

P
u
b
l
i
s
h
D
a

```
t  
a  
(  
d  
i  
s  
t  
a  
n  
c  
e  
)  
;
```

```
d  
e  
l  
a  
y  
(  
1  
0  
0  
0  
)  
;  
i  
f
```

```
(  
!  
c  
l  
i  
e  
n  
t  
.  
l  
o  
o  
p  
(  
)  
)
```



```
{  
  
m  
q  
t  
t  
c  
o  
n  
n  
e  
c  
t  
(  
)  
;  
}  
}  
delay(1000);  
}  
vo  
id  
Pu  
bl  
is  
hD  
at  
a(  
fl  
oa  
t  
di  
st  
) {  
mq  
tt  
co  
nn  
ec  
t(  
);  
St  
ri  
ng
```

```

pa
yl
oa
d =
"{
\"
Di
st
an
ce
\"
:
;
pa
yl
oa
d
+=
di
st
;
payload +=
\", \"ALERT!!\\\": \"\"Distance
less than 100cms\\\""; payload
+= "}";
Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");
} else {
Serial.println("Publish failed");
}
}
void mqttconnect() {
if
(!
cl
ie
nt
.c
on
ne
ct
ed

```

```
()  
)  
{  
    Serial  
    .p  
    ri  
    nt  
    ("Re  
    co  
    nn  
    ec  
    ti  
    ng  
    cl  
    ie  
    nt  
    to  
    ")  
    ;  
    Serial  
    .p  
    ri  
    nt  
    ln  
    (s  
    er  
    ve  
    r)  
    ;  
    while (!!!client.connect(clientId, authMethod, token)) {  
        S  
        e  
        r  
        i  
        a  
        l  
        .  
        p  
        r  
        i
```

```
n  
t  
(  
"  
.  
"  
)  
;  
  
d  
e  
l  
a  
y  
(  
5  
0  
0  
)  
;  
}
```

```
initManagedDevice();  
Serial.println();  
}  
}  
void wificonnect()  
{  
Serial.println();  
Serial.print("Conne  
cting to ");  
WiFi.begin("Wokwi-  
GUEST", "", 6);  
while (WiFi.status()  
!=WL_CONNECTED) {  
delay(500);  
Serial.print(".");  
}  
Serial.  
println  
("");  
Serial.  
println  
("WiFi  
connect
```

```

ed");
Serial.
println
("IP
address
: ");
Serial.
println
(WiFi.l
ocalIP(
));
}
void initManagedDevice() {
if
(client.subscribe(subscr
ibetopic)) {
Serial.println((subscrib
etopic));
Serial.println("subscrib
e tocmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
for (int i = 0; i < payloadLength; i++) {

```

```

/
/
S
e
r
i
a
l
.
p
r
i
n
t

```

```
(  
(  
c  
h  
a  
r  
)  
p  
a  
y  
l  
o  
a  
d  
[  
i  
]  
)  
;
```

```
d  
a  
t  
a  
3
```

```
+  
=
```

```
(  
c  
h  
a  
r  
)  
p  
a  
y  
l  
o  
a  
d  
[  
i
```

```
]
;
}
Serial
.println
("
data
:
"+
data
ta
3)
;
data
ta
3=
""
;
}
```

Diagram.json:

```
{
  "version": 1,
  "author":
    "
    :
    "
    S
    W
```

```
e
e
t
y
s
h
a
r
o
n
"
,
"
e
d
i
t
o
r
"
:
"
w
o
k
w
i
"
,
"
p
a
r
t
s
"
:
[
{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -4.67, "left": -
114.67, "attrs": {} },
```



```

    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": 15.96, "left":
89.17, "attrs": {} }
],
"connections": [
  [ "esp:TX0", "$serialMonitor:RX", "", [] ],
  [
    "esp:R
X0",
"$seri
alMoni
tor:TX
", "",
[] ],[
  "
  e
  s
  p
  :
  V
  I
  N
  "
  ,
  "
  u
  l
  t
  r
  a
  s
  o
  n
  i
  c
  1
  :
  V
  C
  C
  "
  ,
  "
  r

```

```

e
d
"
,
[ "h-37.16", "v-178.79", "h200", "v173.33", "h100.67" ]
],
[ "esp:GND.1", "ultrasonic1:GND", "black", [ "h39.87", "v44.04", "h170" ]
],
[ "esp:D5", "ultrasonic1:TRIG", "green", [ "h54.54", "v85.07", "h130.67"
] ],
[ "esp:D18", "ultrasonic1:ECHO", "green", [ "h77.87", "v80.01", "h110" ]
]

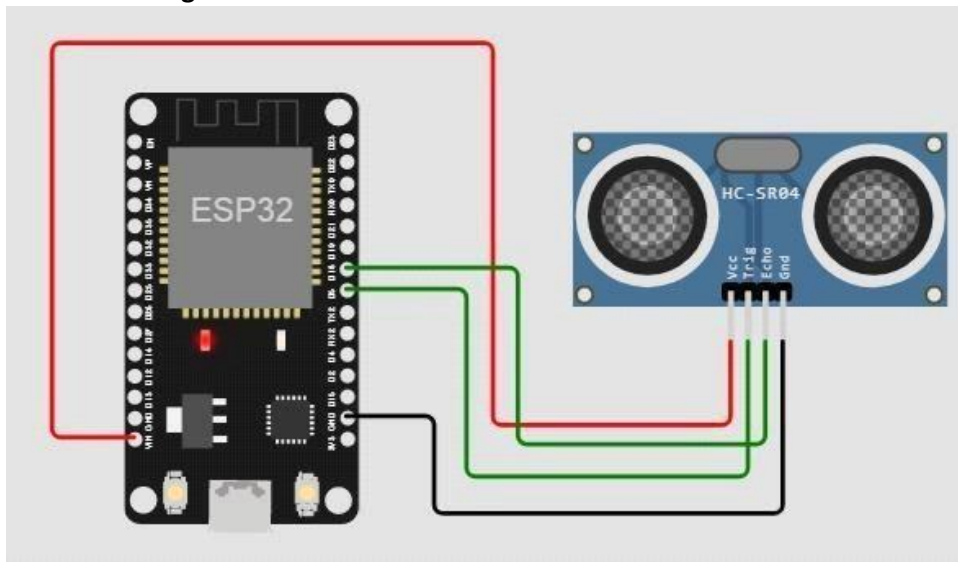
]
}

```

Wokwi simulation link:

<https://wokwi.com/projects/346508314441417298>

Circuit Diagram:




Output:

Wokwi output:

```
Connecting to ....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to ytluse.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK

Distance (cm): 399.92
Distance (cm): 399.96
Distance (cm): 399.94
Distance (cm): 399.98
Distance (cm): 399.94
Distance (cm): 399.92
Distance (cm): 399.94
```

IBM cloud output:

	Browse Action Device Types Interfaces				
	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	
	event_1	{"distance":7,"Alert":"Distance less than 10"}	json	a few seconds ago	
	event_1	{"distance":9,"Alert":"Distance less than 10"}	json	a few seconds ago	
	event_1	{"distance":8,"Alert":"Distance less than 10"}	json	a few seconds ago	
	event_1	{"distance":9,"Alert":"Distance less than 10"}	json	a few seconds ago	