

IOT Based Smart Crop Protection System For Agriculture :

Question :

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

Solution:

```
// ARDUINO PINS (TRIGGER PIN, ECHO
PIN) 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
pinMode(TRIG_PIN, OUTPUT);
digitalWrite(TRIG_PIN, LOW);
//Set Echo pin as input to measure the time duration of pulse returning 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 (~340 m/s). cm = pulse_width / 58.0;
inches = pulse_width / 148.0
```

```

// Print out results if ( pulse_width > max_dist ) { Serial.println("Out of range");
}
else { Serial.println("*****"); Serial.print("Distance Measured 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:

The screenshot displays the Wokwi IDE interface. On the left, the 'sketch.ino' file is open, showing a C++ program for an Arduino Uno connected to an ultrasonic sensor. The code includes pin definitions, setup, and a loop that triggers an 'Alert !!' when the distance is less than 100 cm. On the right, the 'Simulation' window shows a 3D model of the Arduino Uno and the ultrasonic sensor connected by wires. A 'Distance' slider is set to 2cm. Below the simulation, the serial monitor output shows the following sequence of events:

```

*****
Distance Measured in cm : 2.07
Alert !!
*****
Distance Measured in cm : 2.00
Alert !!
*****

```

sketch.ino

diagram.json

Library Manager

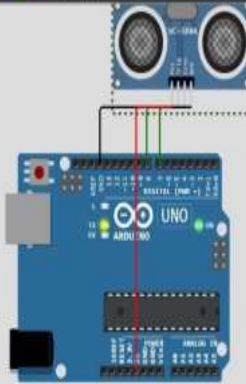
Simulation

00:47.106 100%

```
1
2
3
4
5 // ARDUINO PINS (TRIGGER PIN, ECHO PIN)
6 const int TRIG_PIN = 7;
7 const int ECHO_PIN = 8;
8
9 // Anything over 400 cm (23200 us pulse) is "out of range"
10 const unsigned int max_dist = 23200;
11
12 void setup() {
13
14   // The Trigger pin will tell the sensor to range find
15   pinMode(TRIG_PIN, OUTPUT);
16   digitalWrite(TRIG_PIN, LOW);
17
18   //Set Echo pin as input to measure the time duration of pulse returning back from the d
19   pinMode(ECHO_PIN, INPUT);
20
21   // We'll use the serial monitor to view the sensor output
22   Serial.begin(9600);
23 }
24
25 void loop() {
26
27   unsigned long t1;
28   unsigned long t2;
29   unsigned long pulse_width;
30   float cm;
31   float inches;
32
33   // Hold the trigger pin high for at least 10 us
34   digitalWrite(TRIG_PIN, HIGH);
```

Editing Ultrasonic Distance Sensor

Distance: 268cm



Distance Measured in cm : 271.79

Distance Measured in cm : 271.72

Distance Measured in cm : 271.72

Distance Measured in cm : 271.79

Activate Windows