The working principle of Arduino-Bluetooth Module

The Ultra Sonic HC-SR04 emits ultrasound at 40,000Hz that travels in the air. If there is an object or obstacle in its path, then it collides and bounces back to the Ultra Sonic module.

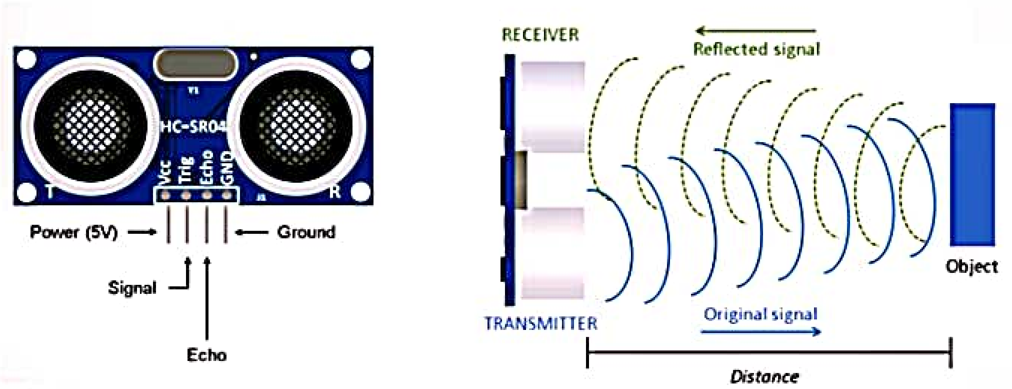
The formula **distance = speed*time** is used to calculate the distance.

calculate the distance.

Suppose, an object is placed at a distance of 10 cm away from the sensor, the speed of sound in air is 340 m/s or 0.034 cm/µs. It means the sound wave needs to travel in 294 µs. But the Echo pin double the distance (forward and bounce backward distance). So, to get the distance in cm multiply the received travel time value with echo pin by 0.034 and divide it by 2.

The distance between Ultra Sonic HC-SR04 and an object is:

$$\text{distance} = \frac{\text{speed} * \text{time}}{2}$$

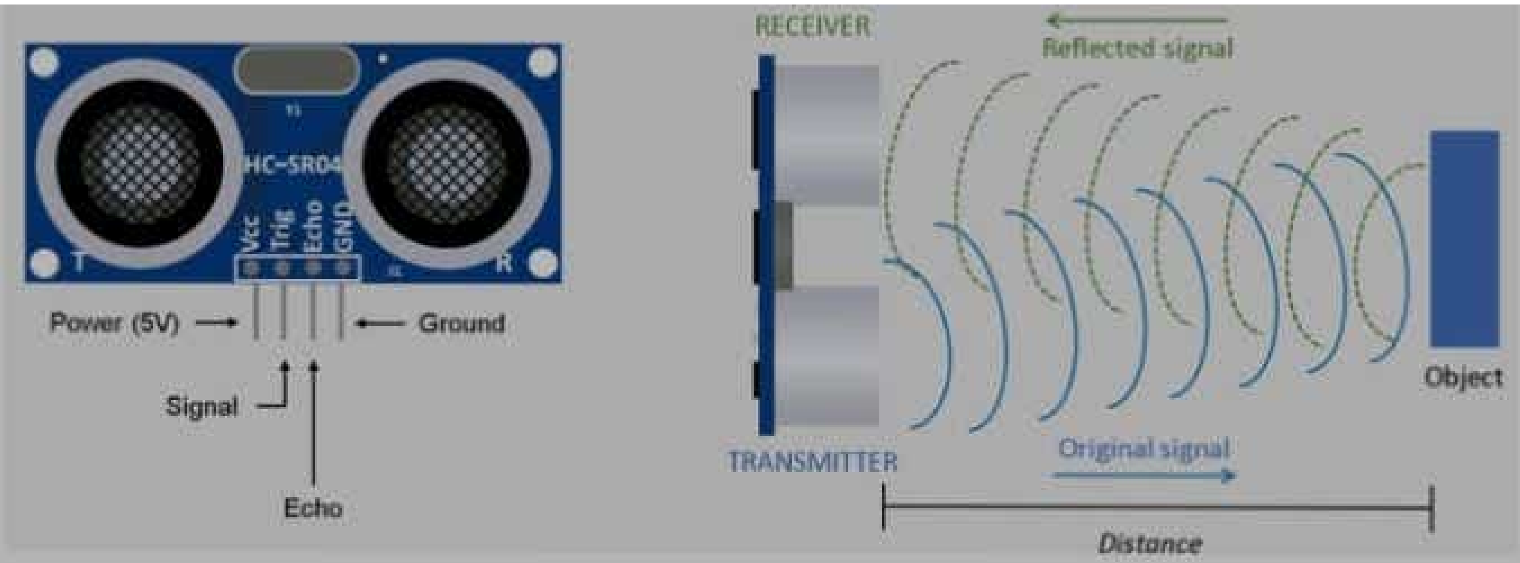


Speed of sound:

$$speed = 340\ m/s = 0.034\ cm/\mu s$$

$$time = distance / speed$$

10



Speed of sound:

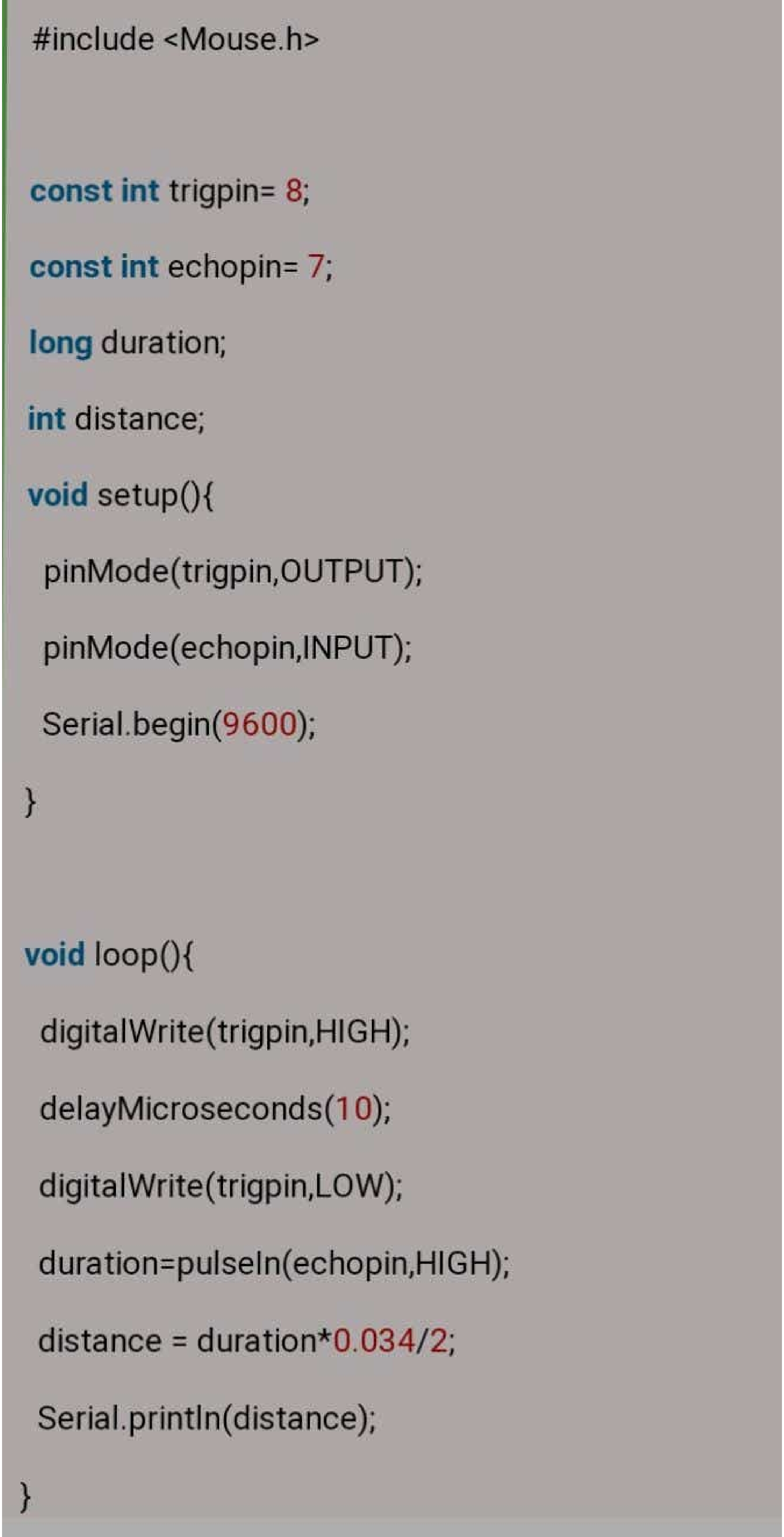
$$speed = 340\ m/s = 0.034\ cm/\mu s$$

$$time = distance/speed$$

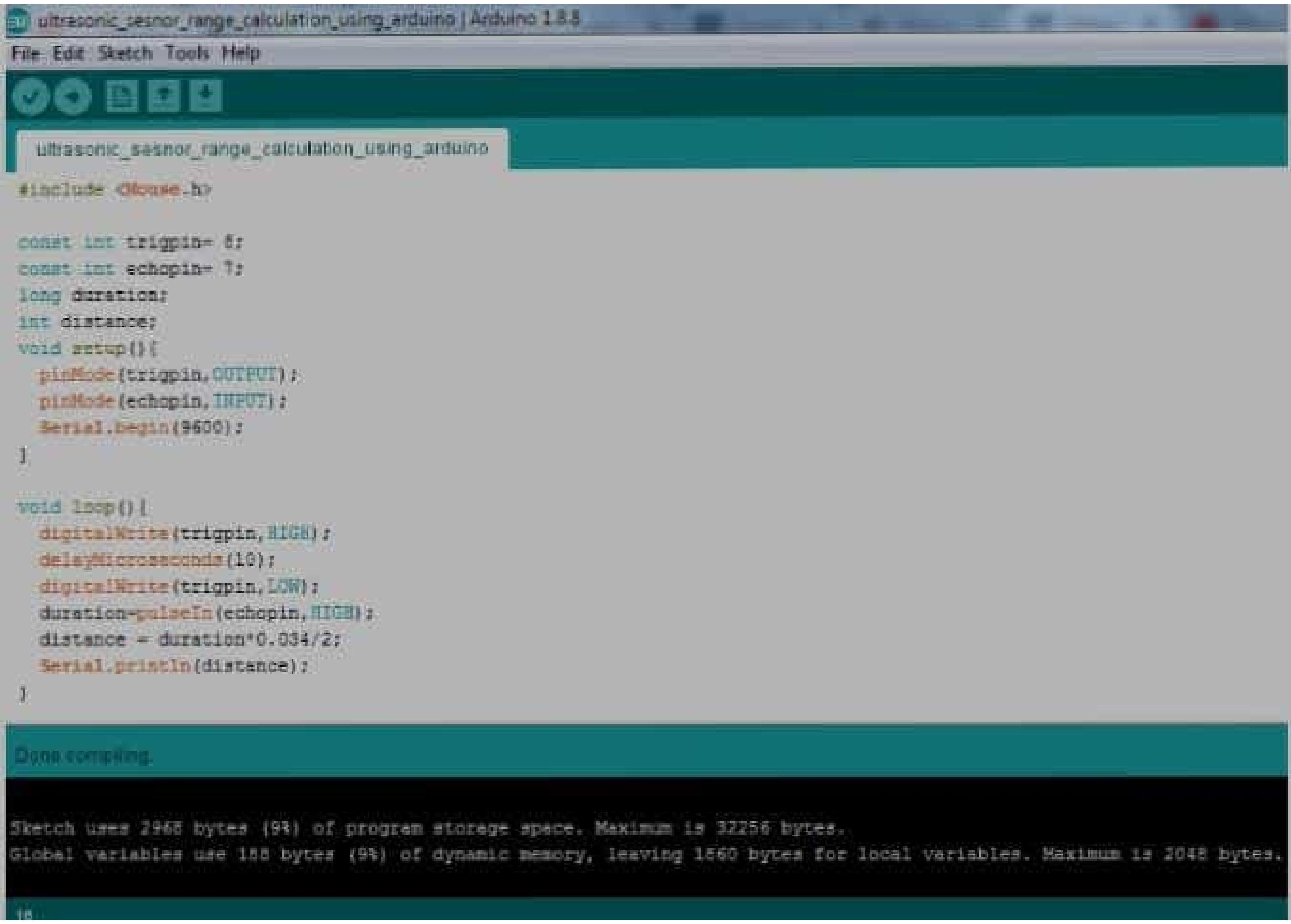
$$time = \frac{10}{0.034}\mu s = 294\ \mu s$$

$$distance = \frac{speed * time}{2}$$

$$distance = \frac{0.034 * 294}{2}$$



```
#include <Mouse.h>


const int trigpin= 8;
const int echopin= 7;
long duration;
int distance;
void setup(){
  pinMode(trigpin,OUTPUT);
  pinMode(echopin,INPUT);
  Serial.begin(9600);
}


void loop(){
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  distance = duration*0.034/2;
  Serial.println(distance);
}
```

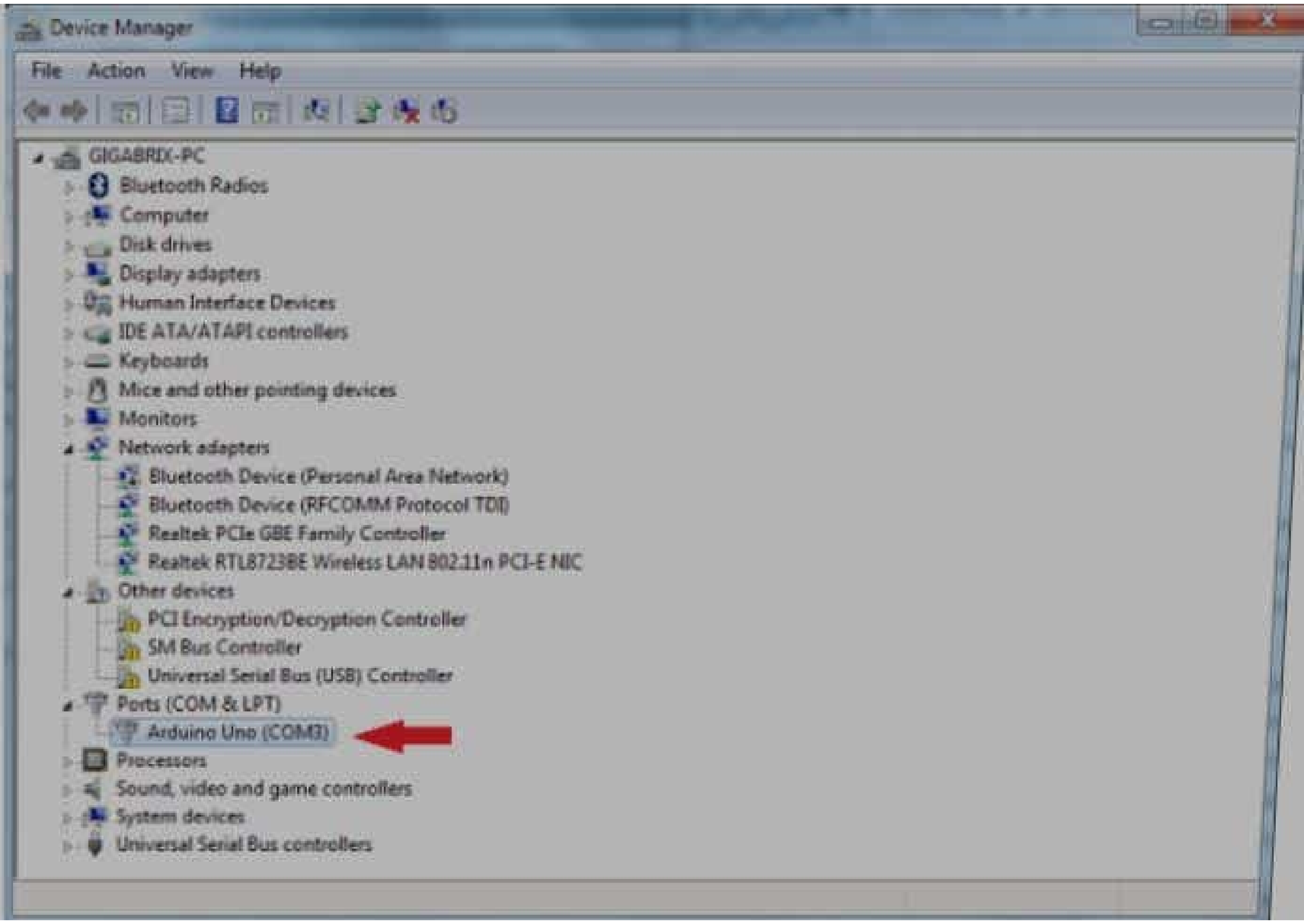


ultrasonic_sesnor_range_calculation_using_arduino | Arduino 1.8.8
File Edit Sketch Tools Help
ultrasonic_sasnor_range_calculation_using_arduino
18

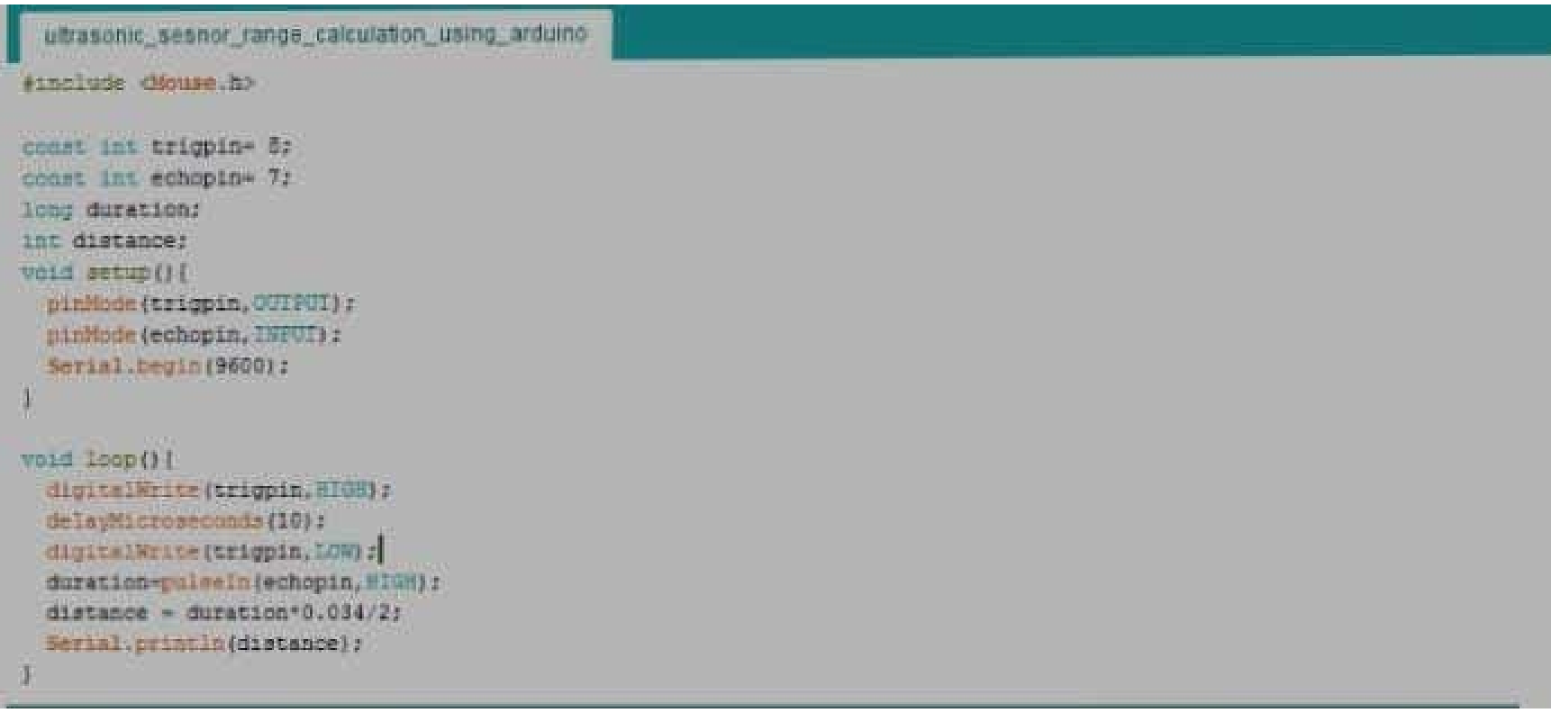
```
#include <Mouse.h>

const int trigpin= 6;
const int echopin= 7;
long duration;
int distance;
void setup(){
  pinMode(trigpin,OUTPUT);
  pinMode(echopin,INPUT);
  Serial.begin(9600);
}

void loop(){
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  distance = duration*0.034/2;
  Serial.println(distance);
}
```

Done compiling.

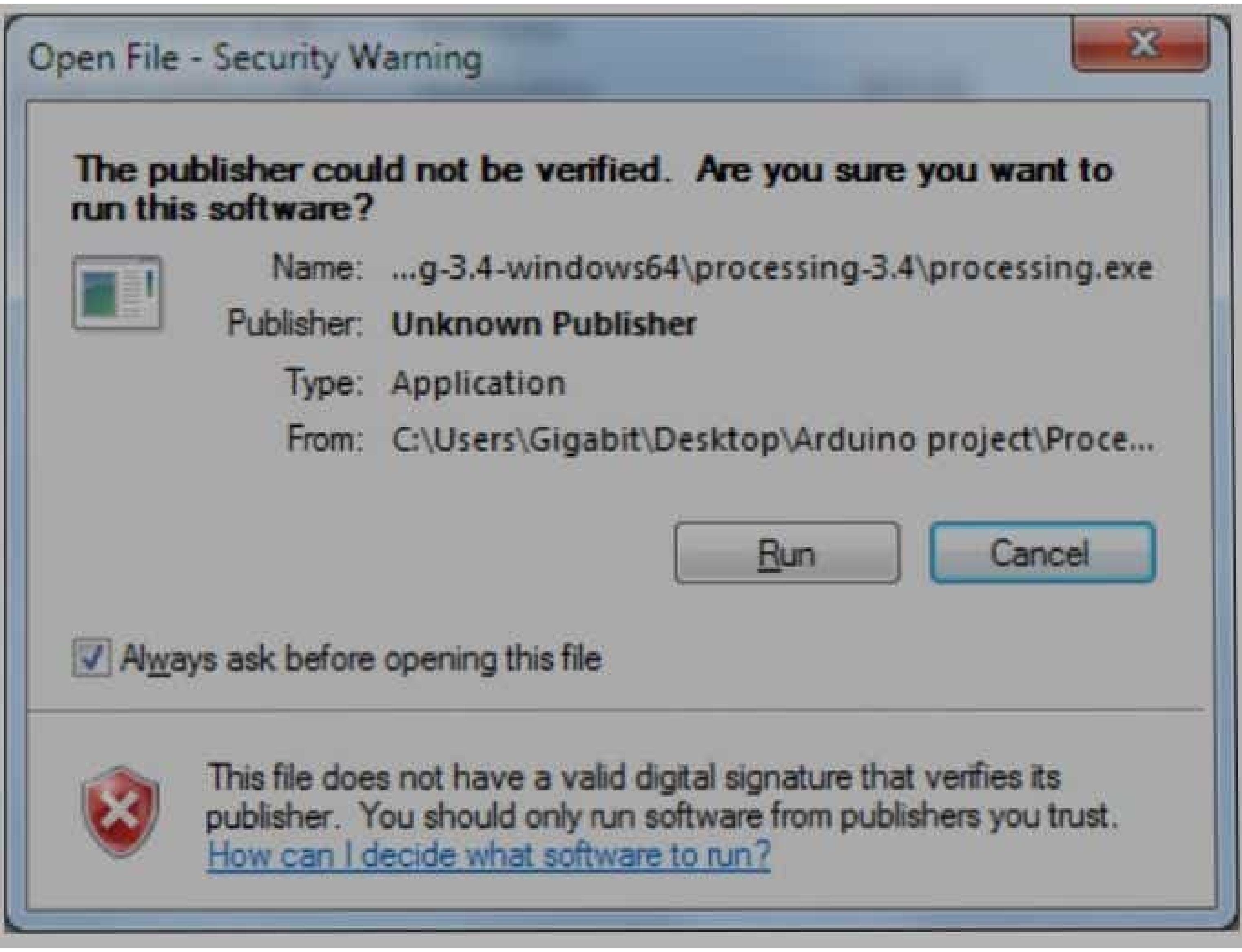
```
Sketch uses 2968 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 188 bytes (9%) of dynamic memory, leaving 1860 bytes for local variables. Maximum is 2048 bytes.
```



- GIGABRIX-PC
  - Bluetooth Radios
  - Computer
  - Disk drives
  - Display adapters
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
    - Bluetooth Device (Personal Area Network)
    - Bluetooth Device (RFCOMM Protocol TDI)
    - Realtek PCIe GBE Family Controller
    - Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC
  - Other devices
    - PCI Encryption/Decryption Controller
    - SM Bus Controller
    - Universal Serial Bus (USB) Controller
  - Ports (COM & LPT)
    - Arduino Uno (COM3)
  - Processors
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers



```
#include <Mouse.h>

const int trigpin= 5;
const int echopin= 7;
long duration;
int distance;
void setup(){
  pinMode(trigpin,OUTPUT);
  pinMode(echopin,INPUT);
  Serial.begin(9600);
}

void loop(){
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  distance = duration*0.034/2;
  Serial.println(distance);
}
```



Open File - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

Name: ...g-3.4-windows64\processing-3.4\processing.exe

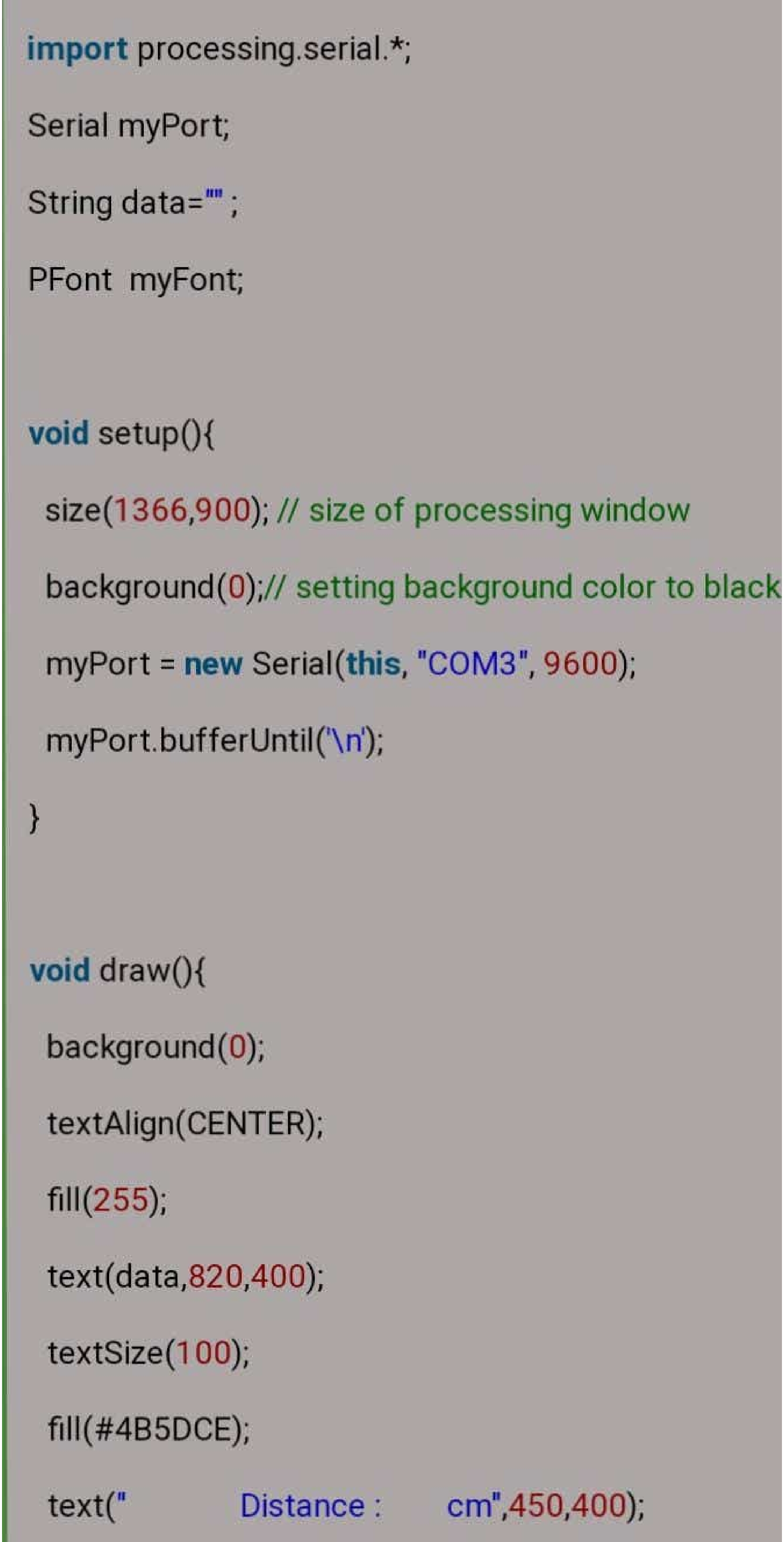
Publisher: **Unknown Publisher**

Type: Application

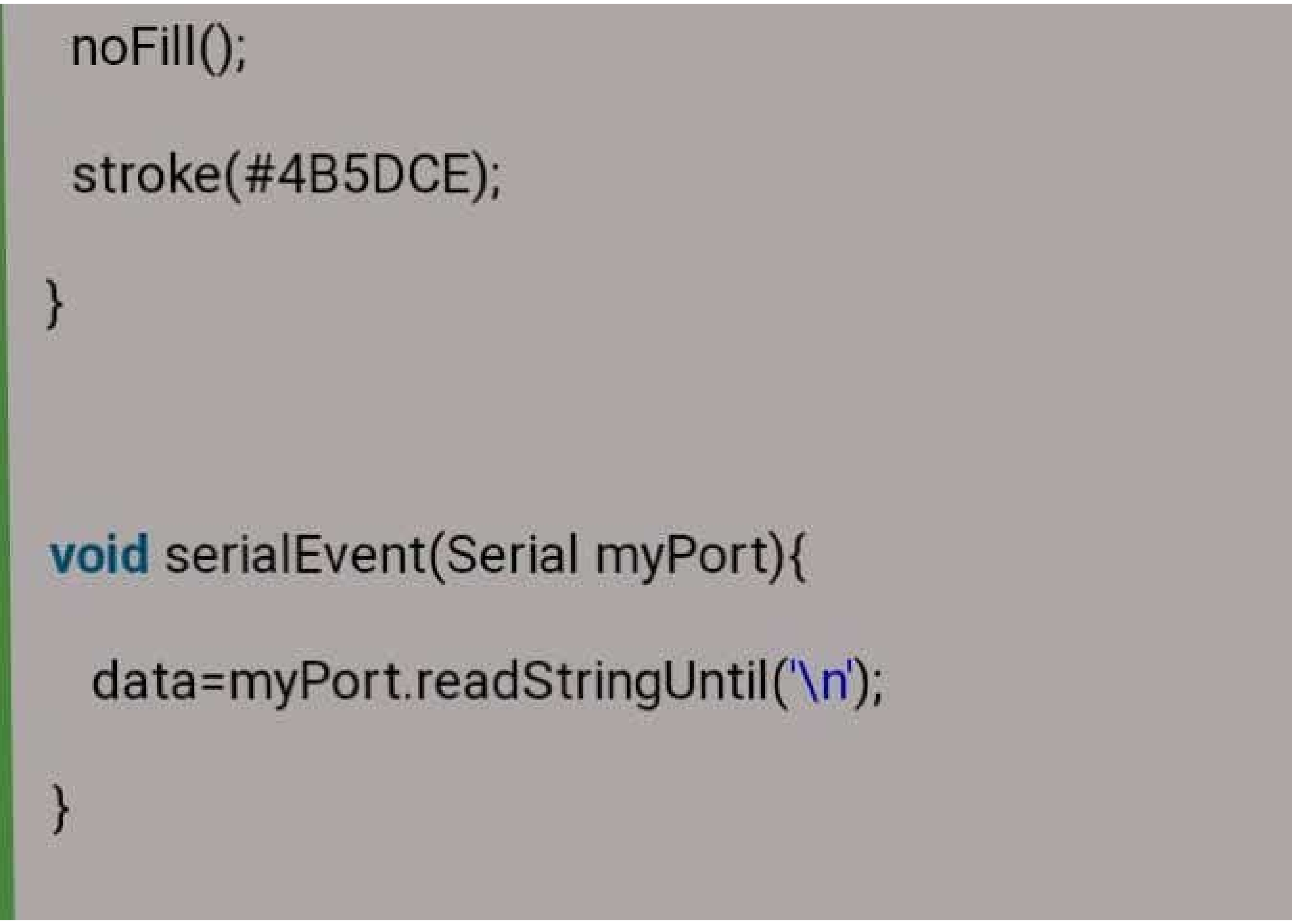
From: C:\Users\Gigabit\Desktop\Arduino project\Proce...

☑ Always ask before opening this file

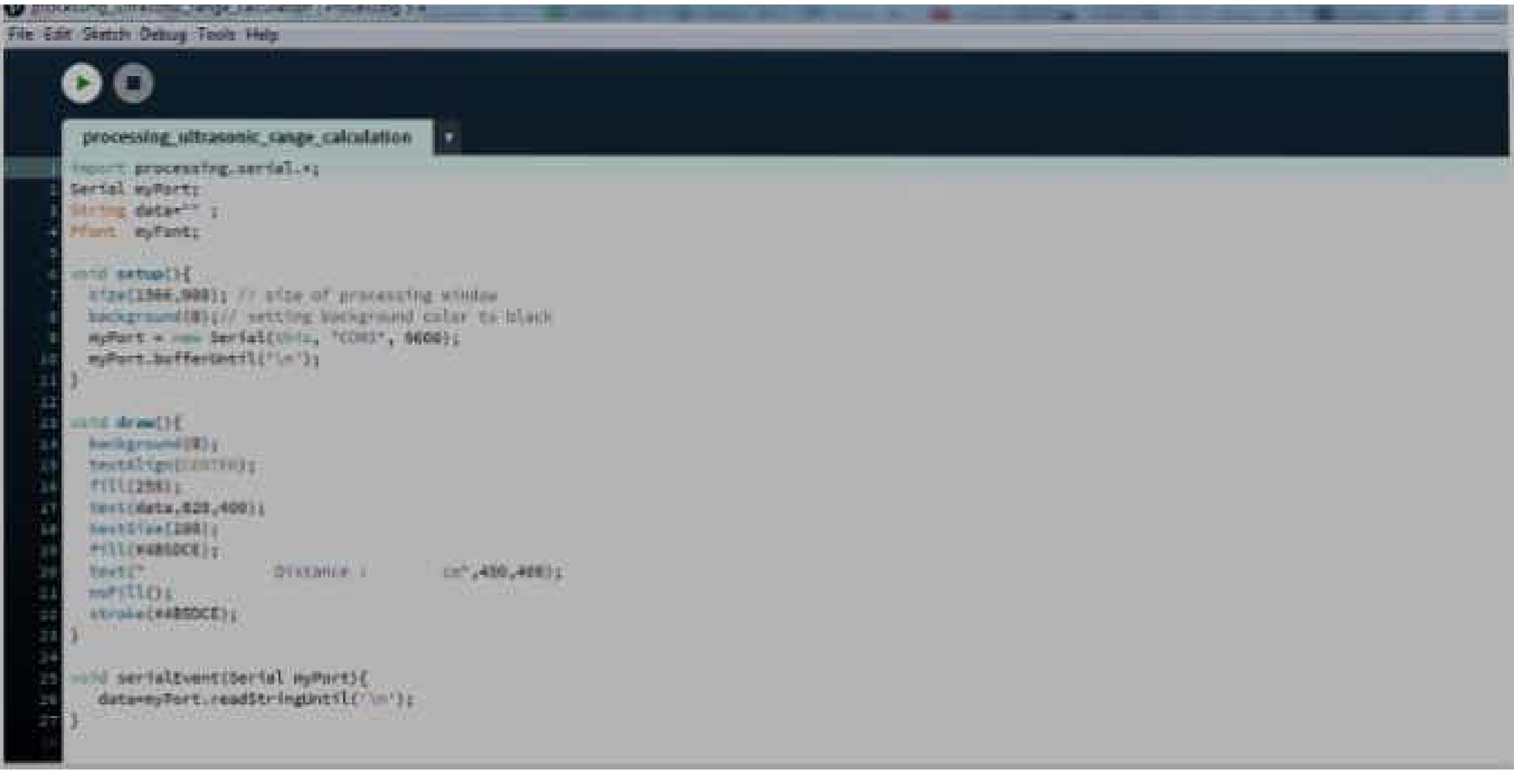
This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust. How can I decide what software to run?



```
import processing.serial.*;

Serial myPort;

String data="" ;

PFont  myFont;


void setup(){

 size(1366,900); // size of processing window

 background(0);// setting background color to black

 myPort = new Serial(this, "COM3", 9600);

 myPort.bufferUntil('\n');

}


void draw(){

 background(0);

 textAlign(CENTER);

 fill(255);

 text(data,820,400);

 textSize(100);

 fill(#4B5DCE);

 text("          Distance :       cm",450,400);
```

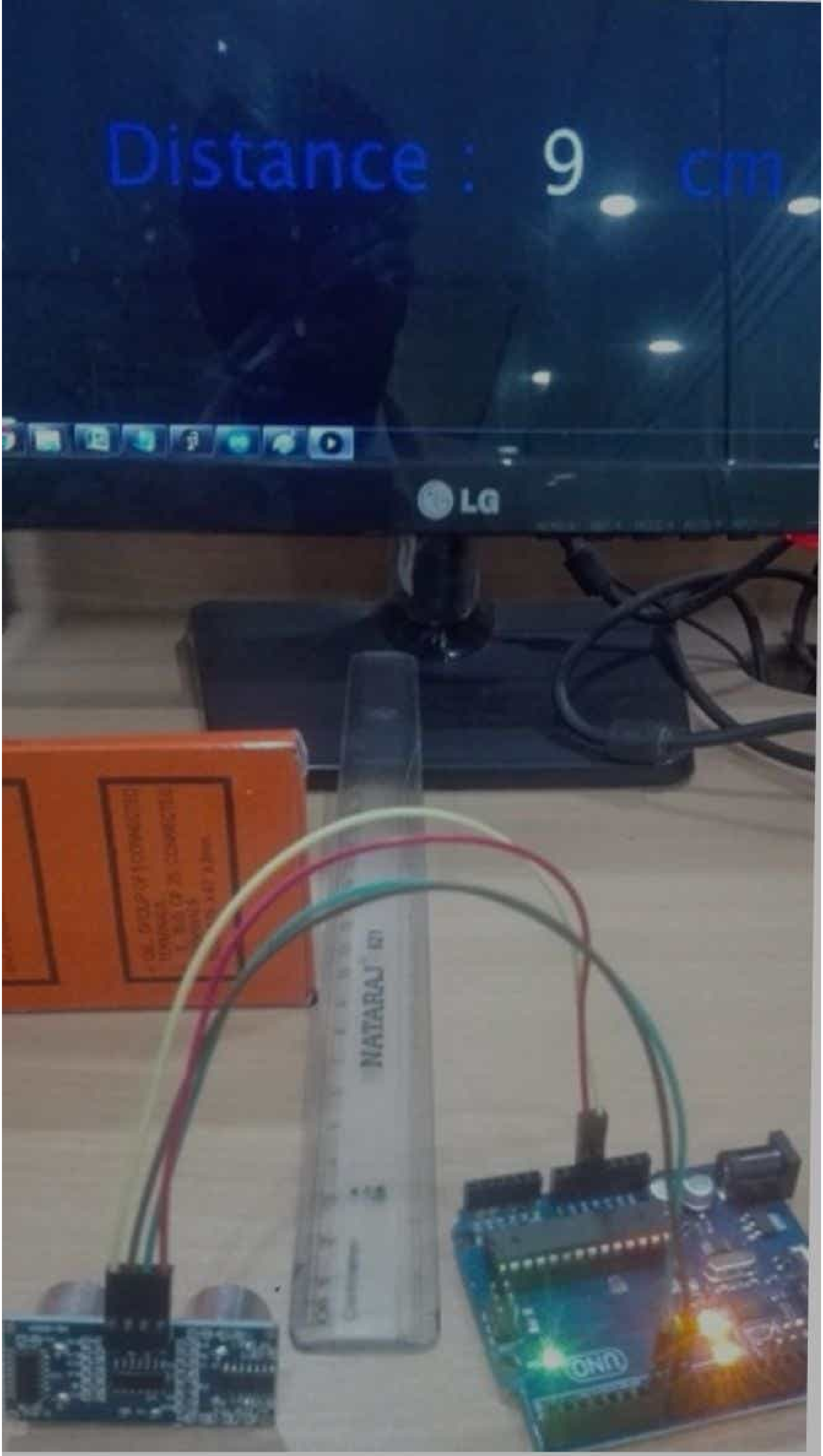


```
 noFill();

 stroke(#4B5DCE);

}


void serialEvent(Serial myPort){

  data=myPort.readStringUntil('\n');

}
```



File Edit Sketch Debug Tools Help
processing_ultrasonic_cange_calculation

```
import processing.serial.*;
Serial myPort;
String data="" ;
PFont  myFont;

void setup(){
  size(1366,900); // size of processing window
  background(0);// setting background color to black
  myPort = new Serial(this, "COM3", 9600);
  myPort.bufferUntil('\n');
}

void draw(){
  background(0);
  textAlign(CENTER);
  fill(250);
  text(data,820,400);
  textSize(100);
  fill(#4B5DCE);
  text("              Distance :        cm",450,400);
  noFill();
  stroke(#4B5DCE);
}

void serialEvent(Serial myPort){
   data=myPort.readStringUntil('\n');
}
```


Distance : 9 cm
LG
NATARAJ 621
UNO