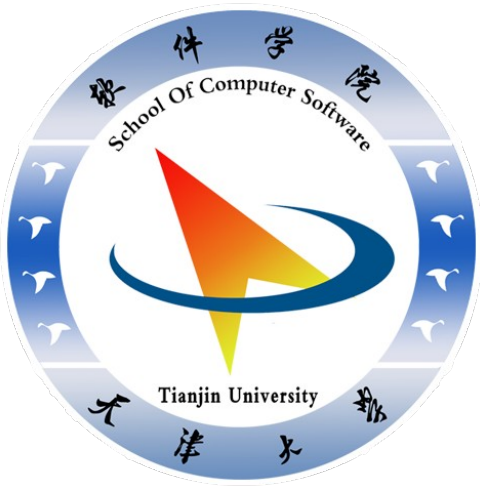
天津大學
Tianjin University

THIRD EYE

Smart shoe for Blind man with mobile application



Presenting to

Professor Zhang, Yikui

The School of Computer Software

Tianjin University

THIRD EYE

Smart shoe for Blind man with mobile application

Group Member

Mohammed Hasan mohammed algamali	Mansoor Saleh Mubarak Bamahel	Taiabur Rahman
PhD 6118000082	Master 6218000158	Master 6218000126

Problem Statement

Third Eye Project

According to the WHO, about 30 million people are estimated to be permanently blind worldwide. These people are totally dependent on others. We have created designed and built an “Ultrasonic Third Eye Project” device which will help blind people to walk with ease independently. As a simpler version, we have used only one ultrasonic sensor in this project. For better accuracy and assistance two or three sensors can be used.

Objective

To help blind people to walk with ease and to be warned whenever their walking path is obstructed with other objects, people or other similar odds. As a warning signal, a Bluetooth device is connected in the circuit, whose gives alarming sound according to the distance of object.

Real-life effect

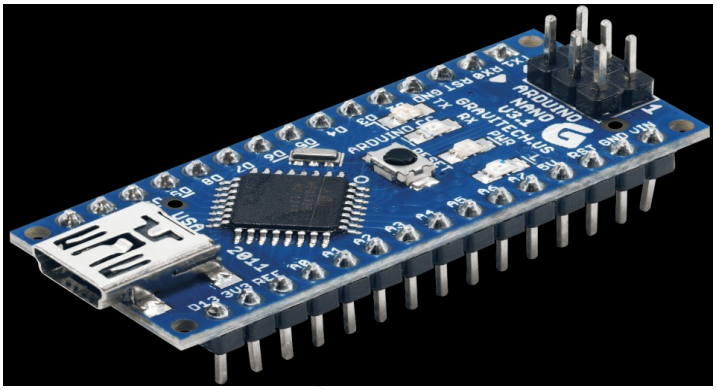
- This project is intended to be developed as tool or aid that will help blind people in moving or traveling.
- The dependency on others is reduced and these people can become more self-reliant.
- This project is intended to be developed as tool or aid that will help blind man in moving, by mean facilitates the life of the blind man.
- The dependency on others is reduced and these people can become more self-reliant.
- Can be used by people with poor eyesight.

Third eye project feature

- Direction Detect
- Speech Guide (Alert)
- Sound buzzer (Alert)
- Remote Tracking (Extension)

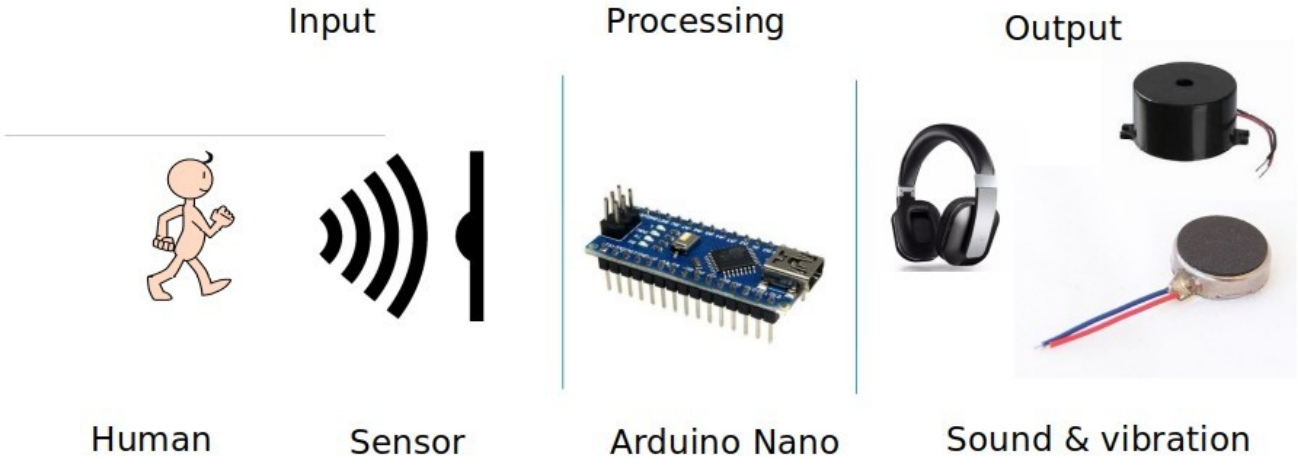
Require Component

- Arduino Nano Board
- Ultrasonic Sensor
- Acceleration Sensor
- Headphone/buzzer
- Vibration motor
- Jumper Wires

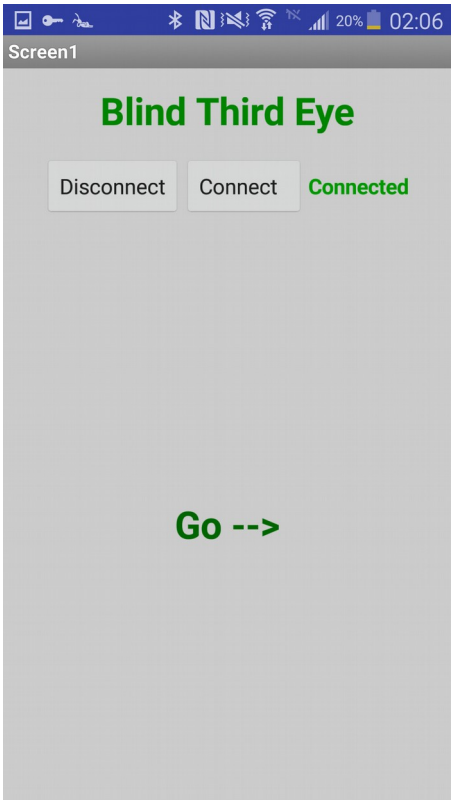


How to work it

we decide to do it something like this. After we are stating working we change to plan.

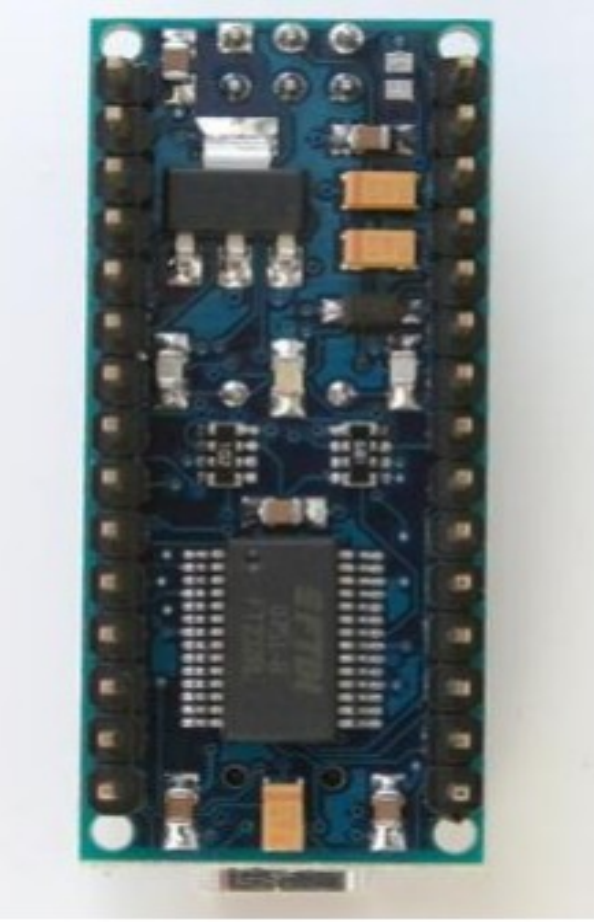


Output with mobile apps



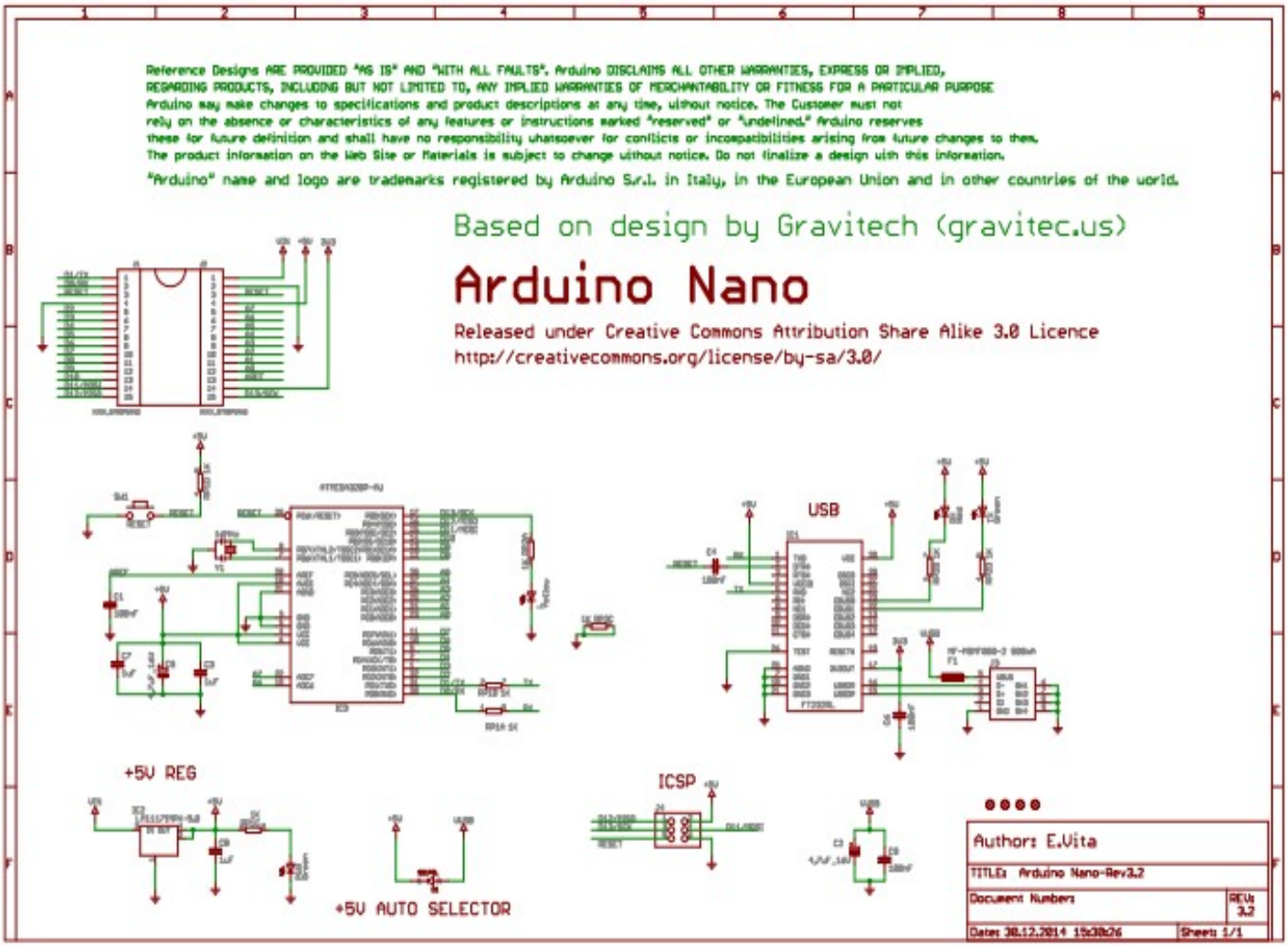
Arduino nano Board

Arduino Nano Front view



Arduino Nano back

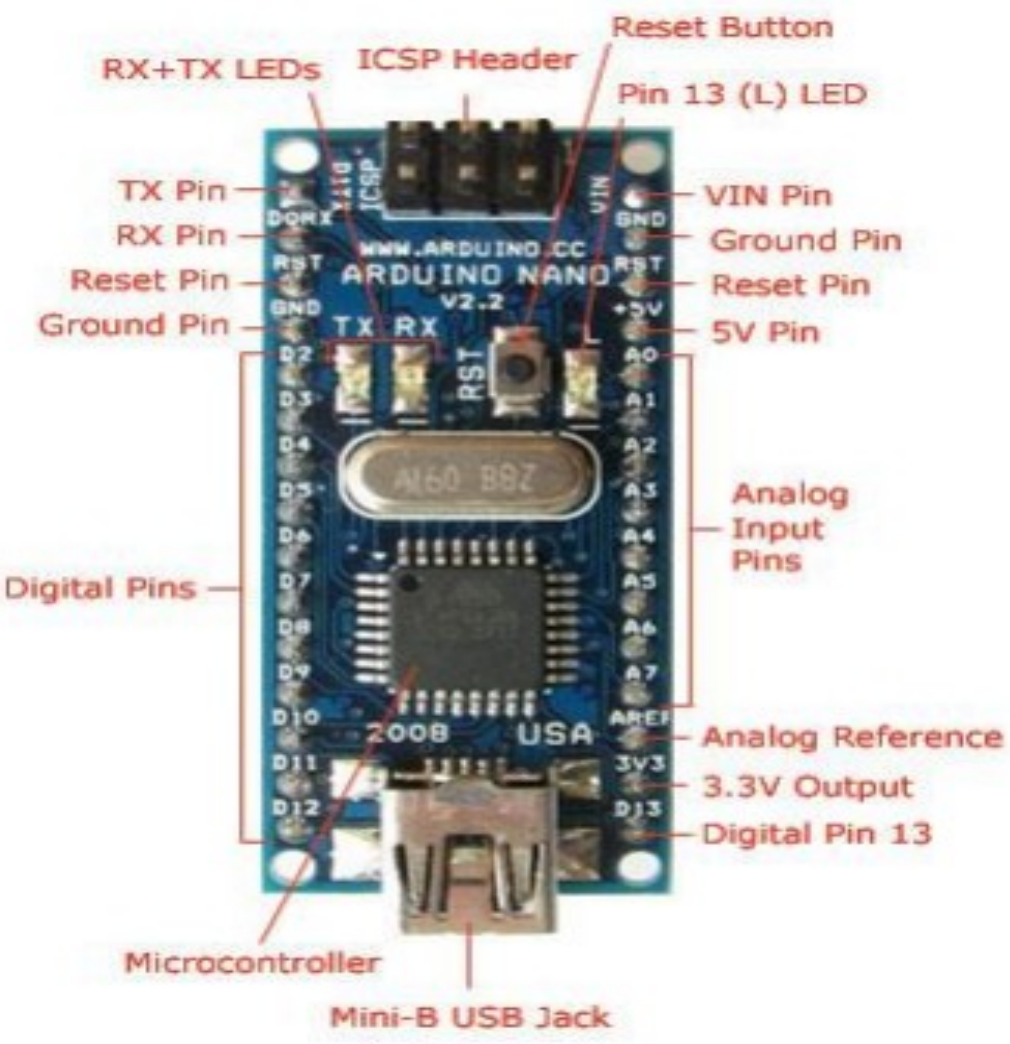
Arduino nano Board : Details views



Arduino nano Board

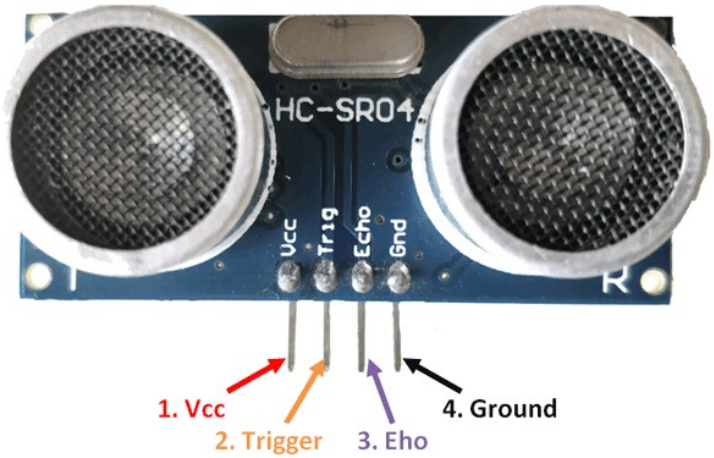
Specifications:

Microcontroller	: Atmel ATmega168 or ATmega328
Operating Voltage	: 5 V
Input Voltage	: 7-12 V
Digital I/O Pins	:14 (of which 6 provide PWM output)
Analog Input Pins	: 8
DC Current per I/O Pin	: 40 mA
Flash Memory	: 16 KB
Clock Speed	: 16 MHz



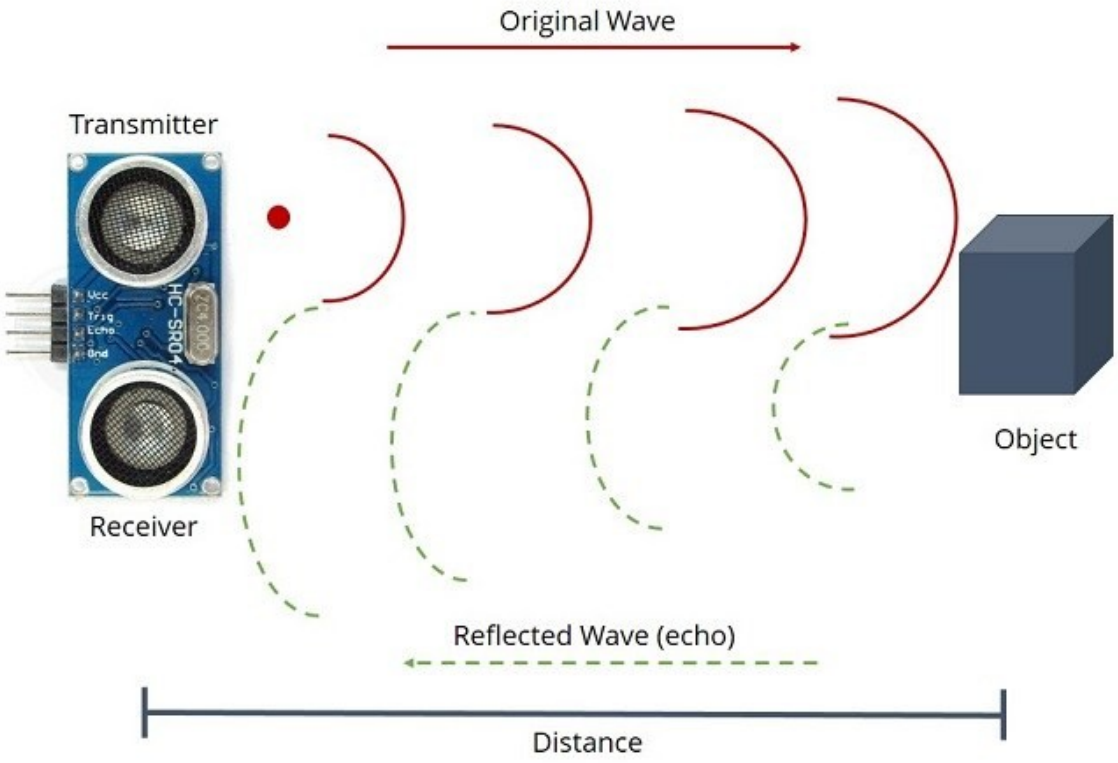
Ultrasonic Sensor

The main component used for this device is the ultrasonic sensor. The ultrasonic sensor transmits a high frequency sound pulse and then calculates the time to receive the signal of the sound echo to reflect back. The sensor has 2 circles. One of them acts as the transmitter and transmits the ultrasonic waves. The other one acts as a receiver (mostly a small microphone) and receives the echoed sound signal. The sensor is calibrated according to the speed of the sound in air. With this calibrated input, the time difference between the transmission and reception of sound pulse is determined to calculate the distance of the object.



Specifications:

- Ultrasonic Ranging Module HC - SR04
- Operating Voltage : 5 V
- Accuracy : 3mm
- Measuring angle covered : <15°
- Operating Current : <15mA
- Operating Frequency : 40Hz



Ultrasonic Sensor

Pin Configuration:

VCC:

The Vcc pin powers the sensor, typically with +5V

Tigger:

Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.

Echo:

Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.

Ground:

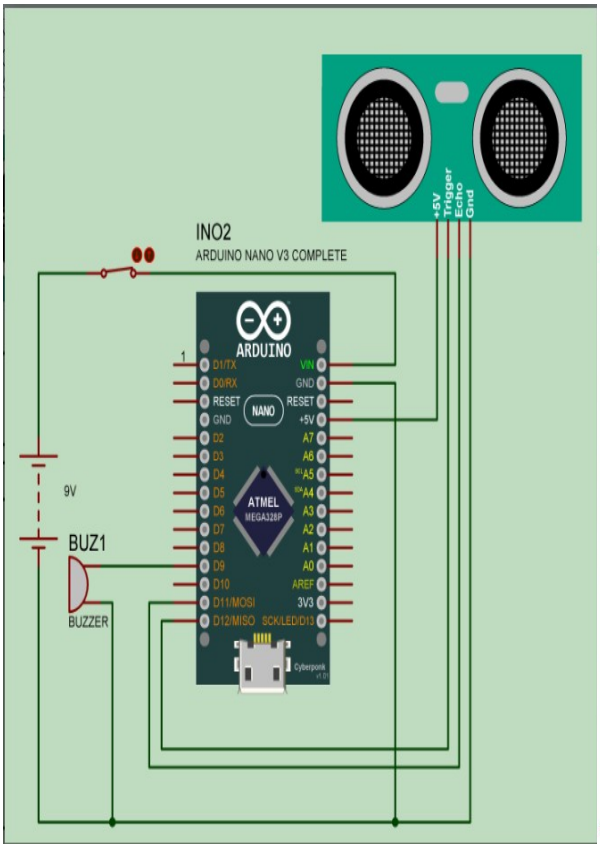
This pin is connected to the Ground of the system

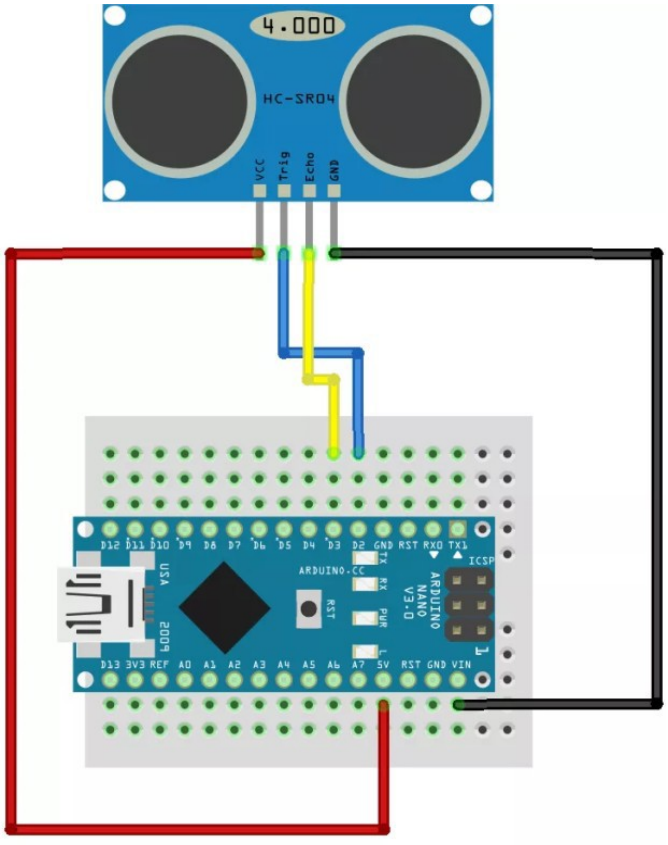
Ultrasonic sensor object detect

Arduino nano board connect with Ultrasonic Sensor

The HC-SR04 Ultrasonic Module has 4 pins, Ground , VCC, Trig and Echo. The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the trig and echo pins to any Digital I/O pin on the Arduino Board.

Arduino Nano Pin	Ultrasonic Sensor Pin
5v	VCC
GND	GND
I/O pin	GND
I/O pin	ECHO



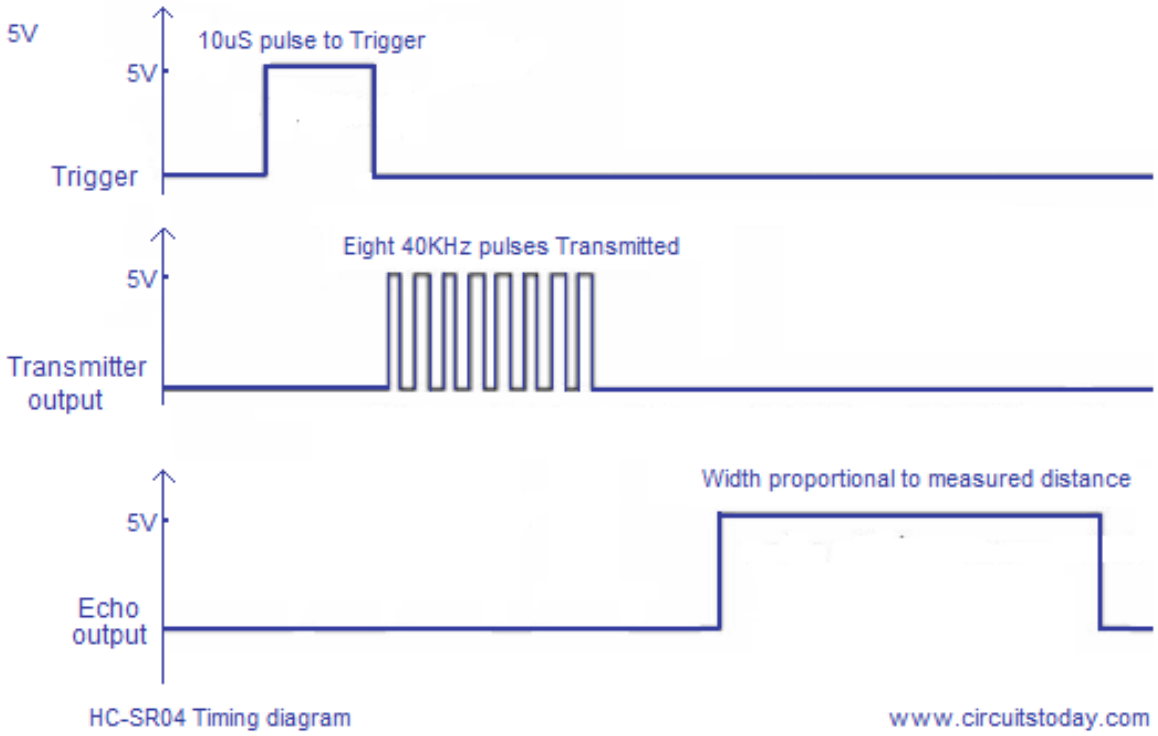


- The HC-SR04 sensor attach to the Breadboard
- The Sensor VCC connect to the Arduino Board +5V
- The Sensor GND connect to the Arduino Board GND
- The Sensor Trig connect to the Arduino Board Digital I/O 9
- The Sensor Echo connect to the Arduino Board Digital I/O 10

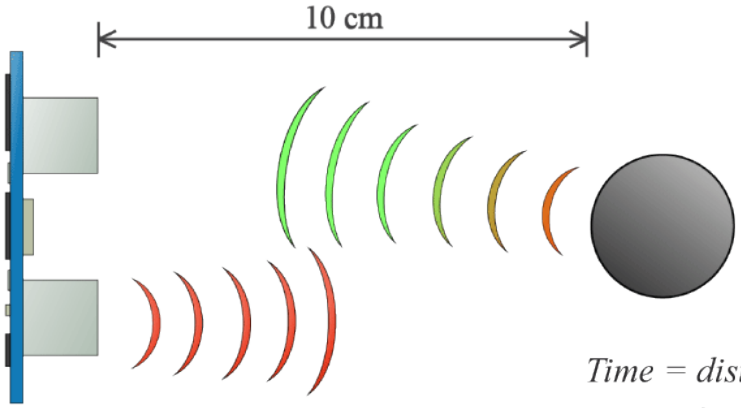
Buzzer and LED (for testing purpose)

- The Buzzer attach to the Breadboard
- The Buzzer long leg (+) connect to the Arduino Board Digital 11
- The Buzzer short leg (-) connect to the Arduino Board GND
- The LED attach to the Breadboard
- The Resistor connect to the LED long leg (+)
- The Resistor other leg (from LED's long leg) connect to the Arduino Board Digital 13
- The LED short leg (-) connect to the Arduino Board GND

The timing diagram of HC-SR04



Time calculation



speed of sound:

$$v = 340\ m/s$$

$$v = 0{,}034\ cm/\mu s$$

Time = distance / speed:

$$t = s / v = 10 / 0{,}034 = 294\ \mu s$$

Distance:

$$s = t \cdot 0{,}034 / 2$$

Vibrating Disk Motor

- Motor construction : 10- Flat coreless vibration motor
- Coil construction : Flat coreless core
- Rated Load : Vibration weight
- Rotation direction : CCW or CW
- Motor position : Any position available
- Voltage range for use : DC 2.5V - 3.5V
- Allowable temperature range: -20°C to 70°C

天津大學
Tianjin University

Project Software (interface)

Mac and Windows Interface

Using software to control the hardware device ,the speech guide and power button will be simulated with this Mac & Win application interface.

Mac interface program: Xamarin Cocoa C# & Xcode.

Windows interface program: Visual studio c#.

Connect Arduino with Mac c# programing

```
using System.IO.Ports;

myPort = new SerialPort();

myPort.BaudRate =9600;

myPort.PortName="/dev/cu.wchusbserial1410";

myPort.Open();
```

Connect Arduino with windows c# program

```
using System.IO.Ports;

myPort = new SerialPort();

myPort.BaudRate = 9600;

myPort.PortName = COM4";

myPort.Open();
```

Connect Arduino with Linux Ubuntu c# program

```
using System.IO.Ports;

myPort = new SerialPort();

myPort.BaudRate = 9600;

myPort.PortName =/dev/ttyUSB0";

myPort.Open();
```

Arduino Code for the Project

```
// Define pins for ultrasonic , buzzer and led
#define trigPin 9
#define echoPin 8
#define led 11
int const buzzPin = 9;
void setup()
{
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);// trig pin will have pulses output
  pinMode(echoPin, INPUT);// echo pin should be input to get pulse width
  pinMode(led, OUTPUT); // led pin is output to light on/off led
  pinMode(buzzPin, OUTPUT); // buzz pin is output to control buzzering
}
void loop()
{
  // Duration will be the input pulse width and distance will be the distance to the
obstacle in centimeters
  int duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
   // Distance is half the duration devided by 29.1 (from datasheet)
  distance = (duration/2) / 29.1;
  if (distance <20)
  {
    digitalWrite(led,HIGH);
    digitalWrite(buzzPin, HIGH);
    }
    else
    {
      digitalWrite(led,LOW);
      digitalWrite(buzzPin, LOW);
    }
 Serial.println(String (distance));
// Waiting 60 ms won't hurt any one
delay(60);
}
```

C# Programming interface for the Project

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Speech.Synthesis;
namespace Project_Aultra
{
    public partial class Form1 : Form
    {
        SerialPort _serialPort;
        SpeechSynthesizer reader;
        delegate void SetTextDelegate(string value);
        public Form1()
        {
            InitializeComponent();
            button2.Enabled = false;
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            _serialPort = new SerialPort();
            reader = new SpeechSynthesizer();
             _serialPort.BaudRate = 9600;
             _serialPort.Parity = Parity.None;
             _serialPort.DataBits = 8;
             _serialPort.ReadTimeout = 500;
             _serialPort.WriteTimeout = 500;
             _serialPort.DataReceived += new
SerialDataReceivedEventHandler(DataReceivedHandler);
              txtobj.Visible = false;
        }
        private void button1_Click(object sender, EventArgs e)
        {
            _serialPort.PortName = comboBox1.Text;
            try
            {
              //  _serialPort.PortName = comboBox1.Text;
                if (!_serialPort.IsOpen)
                {
                    _serialPort.Open();
                    pictureBox1.Enabled = true;
                    button2.Enabled = true;
                }
            }
            catch
            {
                MessageBox.Show("Error opening the serial port!");
            }
        }
        public void SetText(string value)
        {
            if (InvokeRequired)
```

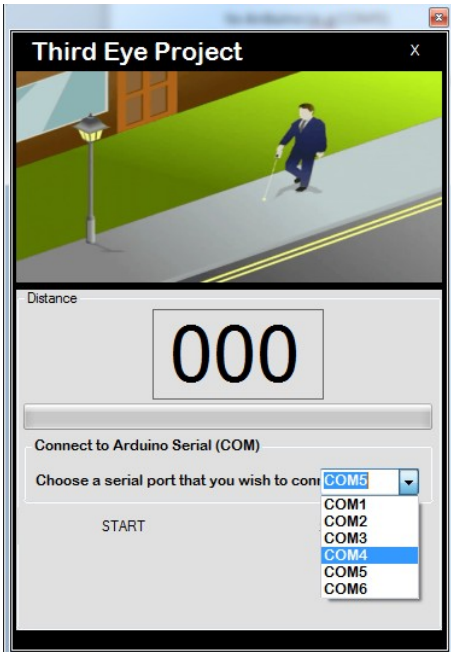
```
                try
                {
                    Invoke(new SetTextDelegate(SetText), value);
                }
                catch { }
            else
                label1.Text = value;
        }
        private void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
        {
            string indata = _serialPort.ReadLine();
            SetText(indata);
        }
        private void button2_Click(object sender, EventArgs e)
        {
                    _serialPort.Close();
                    pictureBox1.Enabled = false;
                     lblmessage.Text = "Application Paused";
                reader.Pause();

        }
        private void label1_TextChanged(object sender, EventArgs e)
        {
            if (int.Parse(label1.Text) <= 100)
            {
                reader.Resume();
                lblmessage.ForeColor = Color.Red;
                lblmessage.Text = "Be carefull an object in front of you";
                reader.SpeakAsync("Be carefull an object in front of you");
                txtobj.Visible = true;
                prgsCM.Value = (int)(int.Parse(label1.Text) /* * 0.25 */ );
              pictureBox1.Enabled = false;
            }
            else
            {
                lblmessage.ForeColor = Color.Green;
                lblmessage.Text = "Go a head";
                pictureBox1.Enabled = true;
                prgsCM.Value = 100;
                txtobj.Visible = false;
                reader.Pause();
            }
        }
        private void button3_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}
```

天津大學
Tianjin University

Device connect with computer port

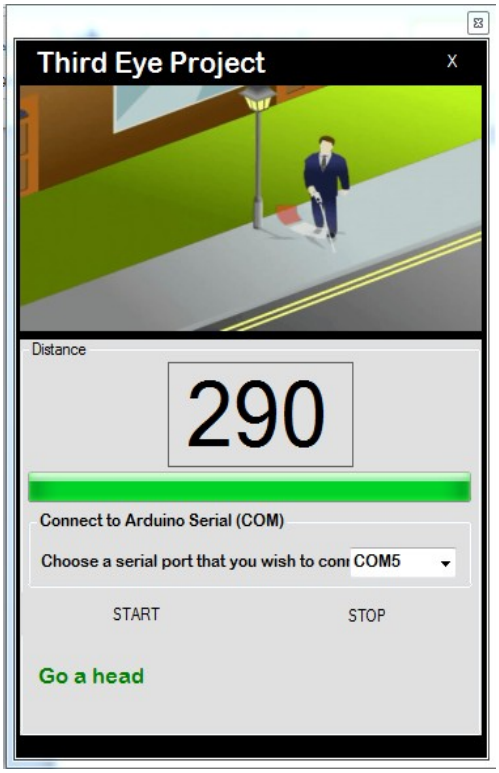
Users have to activate the sensor by choosing a port and then pressing start button in order to connect to Arduino (e.g COM5)

1. Initialized connection first

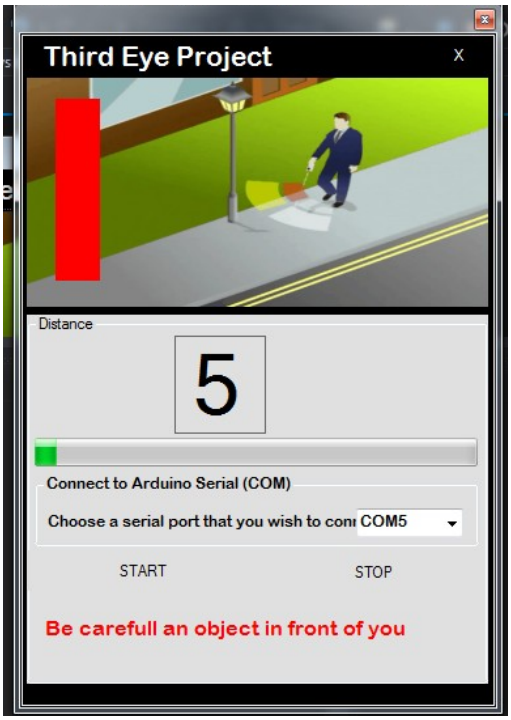


2. Press start button

3. Far distance no worries



4. Sound start within closed object



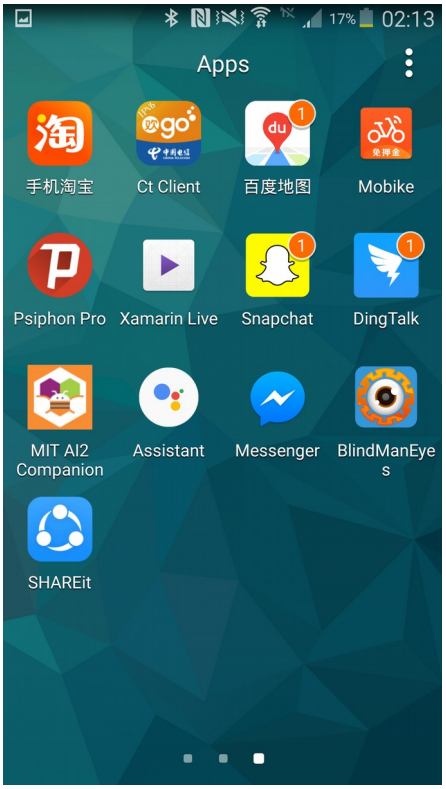
天津大學
Tianjin University

Project Mobile App

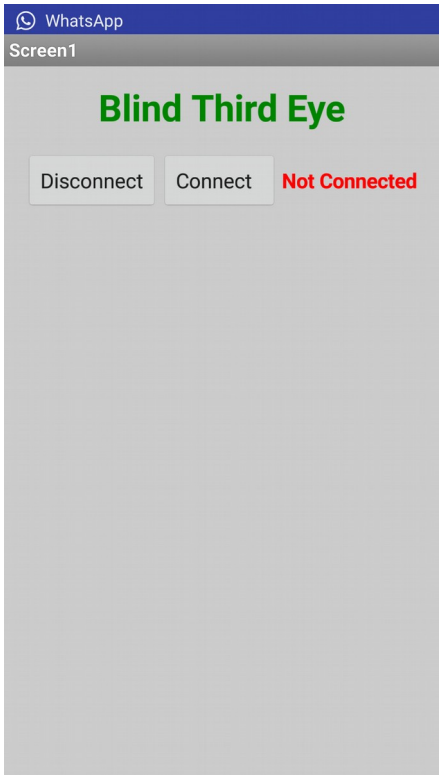
Third eye project App feature

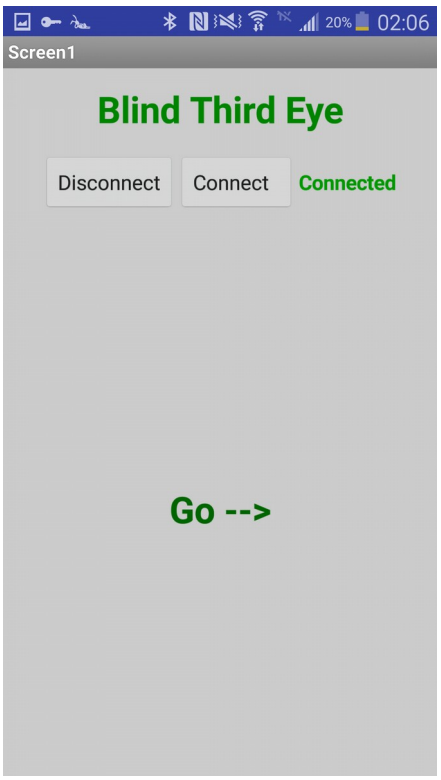
- Speech(Sound) Alarms instead of buzzer
- Sound and Vibration for 5 seconds (Alarms)
- Connected to Bluetooth (Speech- notification)
- Disconnected from Bluetooth (Speech- notification)

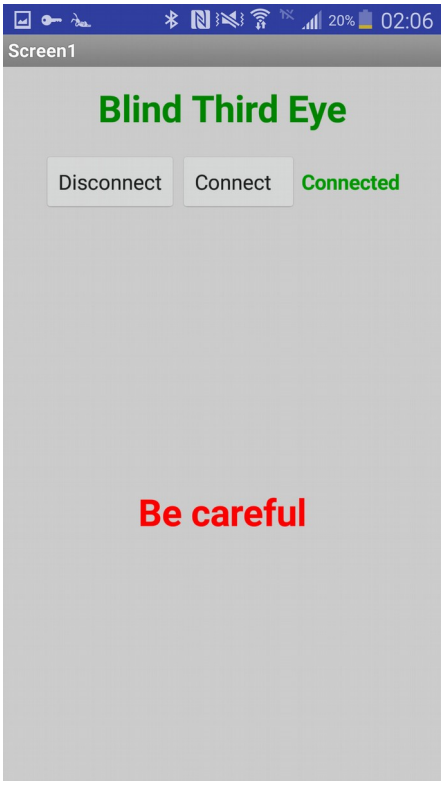
Third eye project icon



Third eye project App interface



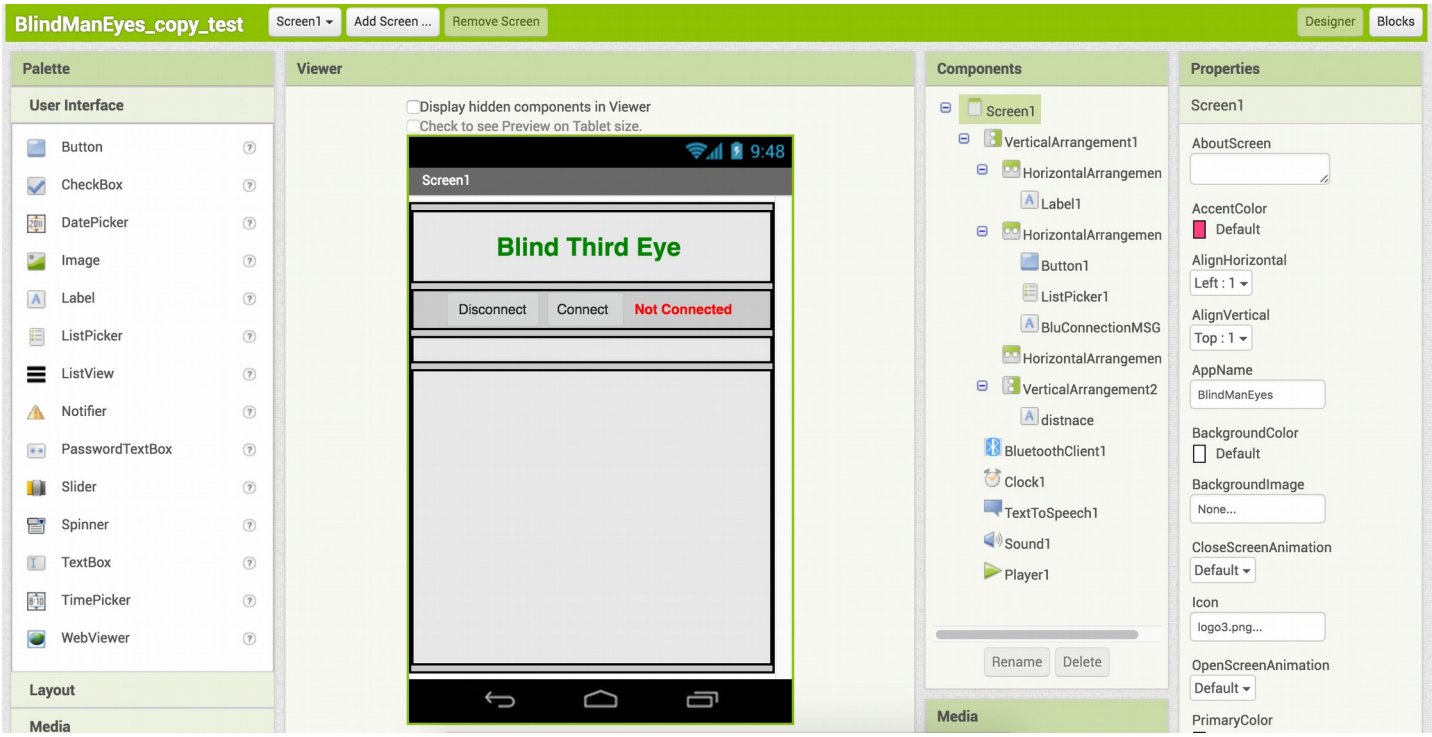




Third eye project Arduino programming

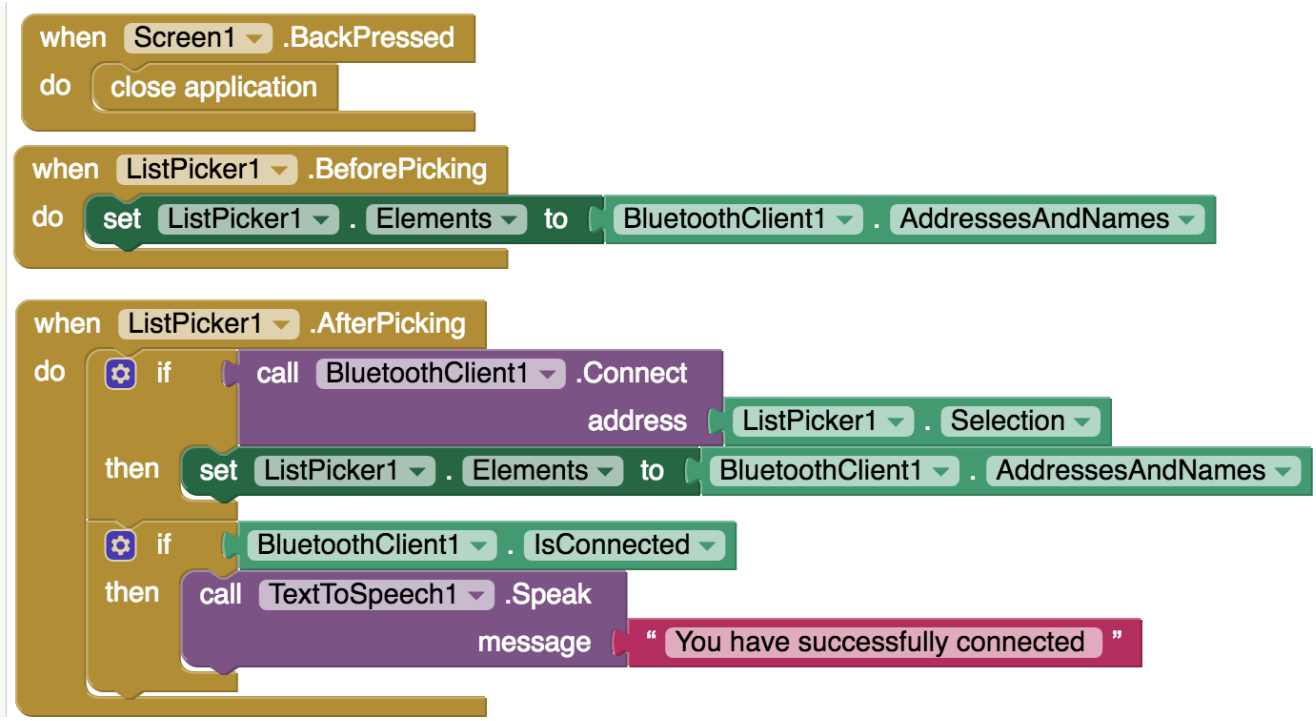
```
1
2  const int trigPin = 11;
3  const int echoPin = 12;
4
5  long duration;
6  int distance;
7  int safetyDistance;
8
9  #include <SoftwareSerial.h>
10 SoftwareSerial BTSerial(8,9); // RX | TX
11
12
13 void setup() {
14 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
15 pinMode(echoPin, INPUT); // Sets the echoPin as an Input
16
17
18 Serial.begin(9600); // Starts the serial communication
19  BTSerial.begin(115200);
20 }
21
23 void loop() {
24 // Clears the trigPin
25
26
27 digitalWrite(trigPin, LOW);
28 delayMicroseconds(2);
29
30 // Sets the trigPin on HIGH state for 10 micro seconds
31 digitalWrite(trigPin, HIGH);
32 delayMicroseconds(10);
33 digitalWrite(trigPin, LOW);
34
35 // Reads the echoPin, returns the sound wave travel time in microseconds
36 duration = pulseIn(echoPin, HIGH);
37
38 // Calculating the distance
39 distance= (duration/2)/29.1;
40
41 safetyDistance = distance;
42 if (safetyDistance < 20)
43
44 { //Enter the Distance
45   BTSerial.print("d");
46 }
47 else
48 {
49 BTSerial.print("n");
50 }
51
52
53 delay(1000);
54 }
```

App inventor programming-design interface

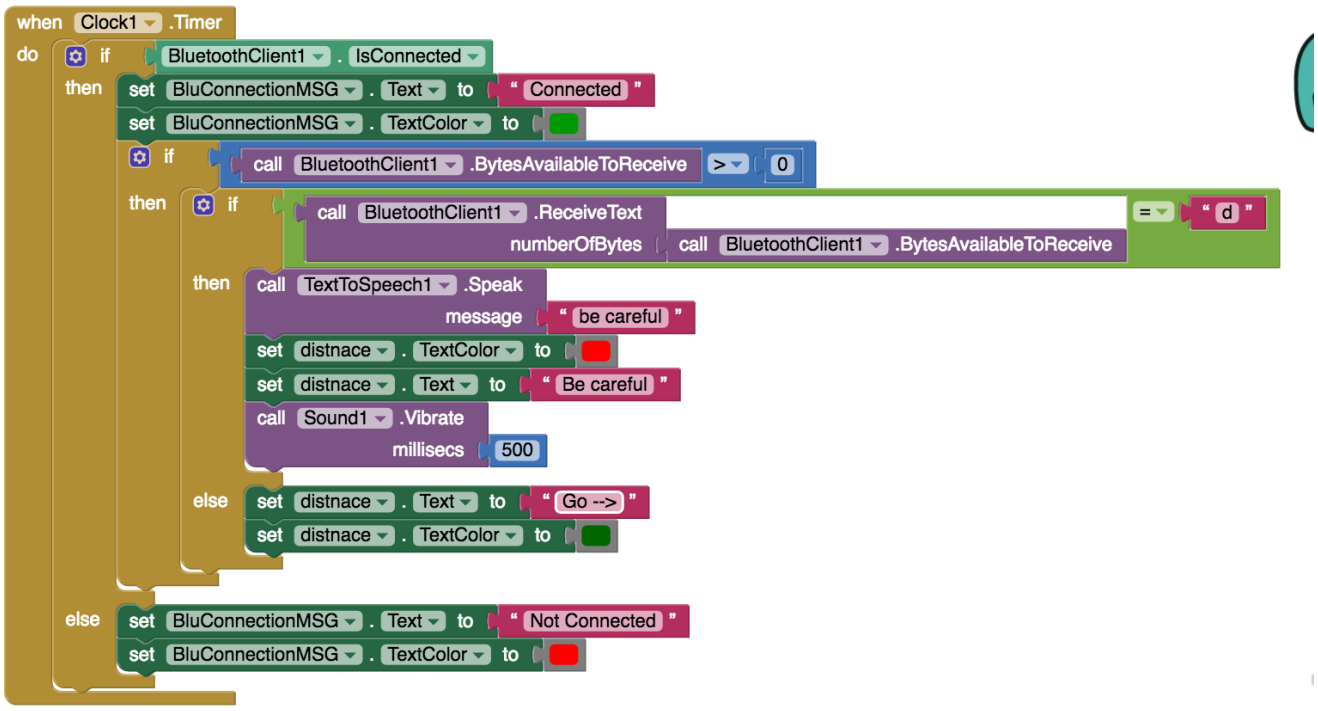


天津大学
Tianjin University

Third eye project App inventor programming

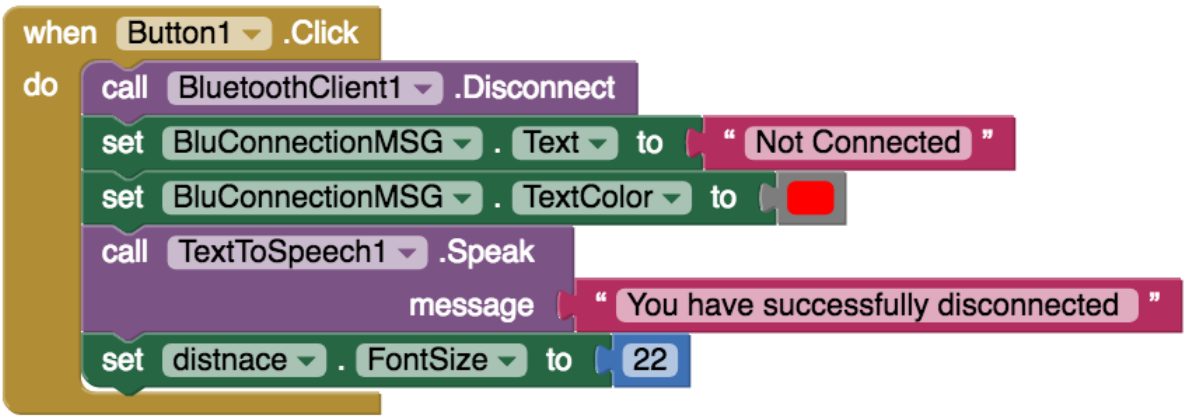


App inventor programming-blocks



0

App inventor programming-blocks



Future Work

This is a great work with ultrasonic sensor and Arduino. It is a great starting point for future. We can work with more than one ultrasonic sensors and make path for blind to take using sounds. We can implement more than one sensor and use Portable GPS Tracker (For monitoring the position of blind person).

Thank you