Amrita Viswa Vidyapeetham
Centre for Excellence in Computational Engineering Networking
Coimbatore - 641112 (India)

Smart Blind Stick Using Arduino Uno

Prepared By:
P.S.S.Sai Keerthana (CB.EN.U4AIE21038)
P.Komal Sai Anurag (CB.EN.U4AIE21039)
Udayagiri Varun (CB.EN.U4AIE21071)
Sejal Singh (CB.EN.U4AIE21061)

Supervised By:
Dr. Ambika P S
Asst. Professor

A 2nd Semester Project submitted to the Department of
Centre for Excellence in Computational and Networking
partial fulfillment of the requirements for final project of Principles of
Measurements and Sensors (PMS)

July, 2022

Amrita School of Engineering
Amrita Viswa Vidyapeetham
Coimbatore - 641-112 (India)

Declaration

We declare that the submitted report is our original work and no values and context of it has been copy pasted from anywhere else. We take full responsibility, that if in future, the report is found invalid or copied, the final decision will be of concerned faculty. Any form of plagiarism will lead to disqualification of the report.

Place: Ettimadai, Coimbatore
Date: 17.07.2022

Acknowledgement

We would like to express our special thanks of gratitude to our professor (Dr. Ambika P S) who gave this golden opportunity to do this wonderful project of Smart blind Stick using Arduino UNO. Her teaching skills gave us the path and driving force to complete this Project.

Also, we would to thank Professor Vinod C K for providing all the necessary equipment for completing the project and clearing doubts regarding the project every time we ask.

Table of Contents

Abstract

Blindness is a visual imparity where the victim has a lack of sight, to overcome such a handicap, devices like smart blind stick are made using a microcontroller (Arduino UNO), this is an innovative device which uses sensors and such initiatives are taken to help the visually impaired. Although walking sticks of many producers are already available, we like to implement an inexpensive upgrade to provide improved surrounding awareness which is essential for blind people especially. In this our plan of action is to use the five ultrasonic sensors in all the exposed directions the blind man is exposed to. This helps to determine the location of the of objects within close proximity that will avoid the threat of the collision and alert the user by the buzzer by the buzzer sound.

Introduction

About blind walking stick

Blind walking stick is also known as white cane, it is used by many people those are blind or visually impaired. It allows its user to scan their surrounding and help them in smoothening the navigation. It’s another use is to analyze the person’s situation and take care of him at all the time. There are different types blind stick used by different persons based on their requirement.

Aim of the Project

In this project we are using Arduino UNO to add additional sensing elements to make the blind stick more helpful for visually impaired. The features we are trying add UDS (Ultrasonic Distance Sensor) to the stick in all the possible direction, also we are including the LDR (Light dependent resistor) for detecting the amount of the light, moisture sensor& fire sensor to detect water& fire in the sensor’s range and finally the tilt sensor for the orientation of the stick.

Components and tools:

Components:

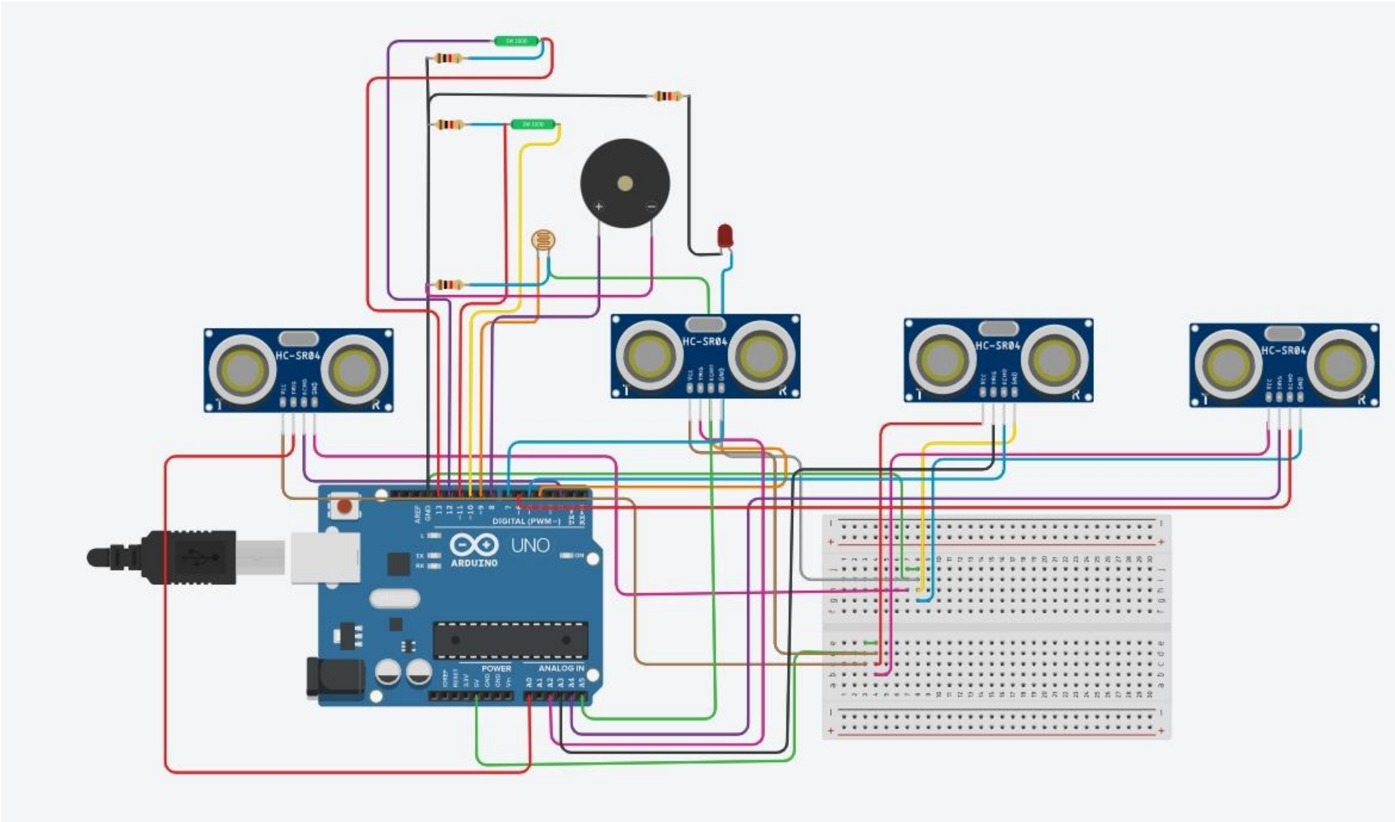
S.No	Name of the Component	Quantity
1.	Arduino UNO R3	1
2.	Ultrasonic Distance sensor	5
3.	Resistor	5
4.	LED	1
5.	Buzzer	1
6.	Photoresistor	1
7.	Tilt Sensor	2
8.	Jumper wires	1
9.	Speaker	1
10.	Flame Sensor	1
11.	Moisture sensor	1

Tools:

Tinker CAD

Implementation

3.1 Circuit Diagram



3.2 Arduino Code

sketch_jul17a

```
#include <NewPing.h>
#include <NewTone.h>
#include "Talkie.h"
#include "Vocab_US_Large.h"
#define maxDistance 100

Talkie voice;

const int trg1 = A0,ech1 = 2;
const int trg2 = A1,ech2 = 3;
const int trg3 = A2,ech3 = 4;
const int trg4 = A3,ech4 = 5;

int flameInPin = 2;
int msensor = A1;
int msvalue = 0;
boolean flag = false;

const int led = 7, buz = 8;
const int ldri = 9, ldro = A5;
const int ltilti = 10, ltilto = 11;
const int rtilti = 12, rtilto = 13;
const uint8_t scale[] PROGMEM = {239,226,213,201,190,179,169,160,151,142,134,127};

const int sensorMin = 0;
const int sensorMax = 1024;
```

```
NewPing sonar1(trg1, ech1, 30);
NewPing sonar2(trg2, ech2, 30);
NewPing sonar3(trg3, ech3, 30);
NewPing sonar4(trg4, ech4, 30);


long dist1;
long dist2;
long dist3;
long dist4;

long light;
bool  ltilt,rtilt;

void setup() {
  // put your setup code here, to run once:

  pinMode(trg1, OUTPUT);
  pinMode(trg2, OUTPUT);
  pinMode(trg3, OUTPUT);
  pinMode(trg4, OUTPUT);

  pinMode(ech1, INPUT);
  pinMode(ech2, INPUT);
  pinMode(ech3, INPUT);
  pinMode(ech4, INPUT);


  pinMode(buz, OUTPUT);
```

```
  pinMode(ldri, OUTPUT);
  pinMode(ldro, INPUT);
  pinMode(ltilti, OUTPUT);
  pinMode(ltilto, INPUT);
  pinMode(rtilti, OUTPUT);
  pinMode(rtilto, INPUT);
  pinMode(led,OUTPUT);

  pinMode(flameInPin, INPUT);

  pinMode(msensor, INPUT);

  Serial.begin(9600);
}
```

```
void loop() {

  //  digitalWrite(ldri,HIGH);

  msvalue = analogRead(msensor);

  if((msvalue<=980 ) && ( flag == false )){
    digitalWrite(led, HIGH);
    flag = true;
    voice.say(sp4_ALERT);
    voice.say(sp4_CAUTION);
    voice.say(sp4_CAUTION);
    voice.say(sp4_CAUTION);

  delay(1000);
  }

  if(digitalRead(flameInPin) == HIGH){
      voice.say(sp4_ALERT);
      voice.say(sp2_FIRE);
      voice.say(sp2_FIRE);
      voice.say(sp2_FIRE);
    }
```

```
  light = analogRead(ldro);

  if(light<20 ||light > 600){
    Serial.println("buz light");
    digitalWrite(buz,HIGH);
    delay(1000);
  }

  digitalWrite(ltilti, HIGH);
  ltilt = digitalRead(ltilto);
  digitalWrite(rtilti,HIGH);
  rtilt = digitalRead(rtilto);

  if(ltilt==1){
    digitalWrite(led,HIGH);
    delay(1000);
  }

  else if(rtilt==1){
    digitalWrite(led,HIGH);
    delay(1000);
  }

  else{
    digitalWrite(led,LOW);
    delay(1000);
  }
```

```
  dist1 = sonar1.ping_cm();
  dist2 = sonar2.ping_cm();
  dist3 = sonar3.ping_cm();
  dist4 = sonar4.ping_cm();


  if(dist1<20 && dist1!=0){
    NewTone(buz, 441);
    delay(1000);
    noNewTone(buz);
    voice.say(sp4_WARNING);
    voice.say(sp4_DANGER);
    voice.say(sp4_DANGER);
    voice.say(sp4_DANGER);
}

else if(dist2<20 && dist2!=0){
NewTone(buz, 441);
   delay(1000);
   noNewTone(buz);
   voice.say(sp4_WARNING);
   voice.say(sp4_DANGER);
   voice.say(sp4_DANGER);
   voice.say(sp4_DANGER);
}
```

```
else if(dist3<20 && dist3!=0){
    NewTone(buz, 441);
    delay(1000);
    noNewTone(buz);

    voice.say(sp4_WARNING);
    voice.say(sp4_DANGER);
    voice.say(sp4_DANGER);
    voice.say(sp4_DANGER);
}
 if(dist4<20 && dist4!=0){
    NewTone(buz, 441);
    delay(1000);
    noNewTone(buz);

    voice.say(sp4_WARNING);
    voice.say(sp4_DANGER);
    voice.say(sp4_DANGER);
    voice.say(sp4_DANGER);
}

else{
  digitalWrite(buz,LOW);
}
```

3.3 Working Logic

LDR:

The working principle of an LDR is photoconductivity, which is nothing but an optical phenomenon. When the light is absorbed by the material then the conductivity of the material enhances. When the light falls on the LDR, then the electrons in the valence band of the material are eager to jump to the conduction band.

The Light dependent Resistor (LDR) continuously scans the surrounding area for the quantity of the light. During scanning the light if it found the intensity of light very low or very high. It passes the information to the Arduino. Arduino reads that information and based upon that the buzzer buzzes for 1sec if the light intensity is low or high.

UDS:

Ultrasonic sensors work by emitting sound waves at a frequency too high for humans to hear. They then wait for the sound to be reflected back, calculating distance based on the time required. This is similar to how radar measures the time it takes a radio wave to return after hitting an object.

Together 4 Ultrasonic sonic sensors in for different directions continuously scan the area infront of him that is in the effective range. This information is feed as an input to the Arduino. Here as we are using 4 Ultrasonic sensors, if it detects any object in its range the distance will be given to the Arduino.

The Arduino decides to notify the user by buzzing the buzzer in such a way that based upon the distance the buzzer will buzz at different frequencies and also alerts the user the saying ‘DANGER’.

Tilt Sensors:

A tilt sensor is an instrument that is used for measuring the tilt in multiple axes of a reference plane. Tilt sensors measure the tilting position with reference to gravity and are used in numerous applications. They enable the easy detection of orientation or inclination. The two tilt sensors are calibrated to the cane's verticality and transmit a Boolean result to the Arduino. When the cane tilts more than a certain amount, the Arduino decides to notify the user's caregivers via the LED for 1 second until he has recovered.

Here in the stick, we use the tilt sensor to see whether there is a change in the angle of the stick. So, whenever the angle of the stick changes the mercury in the module tilts and the LED on it glows.

Flame Sensors:

This sensor/detector can be built with an electronic circuit using a receiver like electromagnetic radiation. This sensor uses the infrared flame flash method, which allows the sensor to work through a coating of oil, dust, water vapor, otherwise ice.

The flame sensor has a photodiode to detect the light and an op-amp to control the sensitivity. When it detects fire, it will provide a high signal upon the detection. Arduino reads the signal and provides the alert by through the speakers saying “FIRE!!”.

Moisture Sensors:

The Moisture sensors measure the water content in the soil and can be used to estimate the amount of stored water in the soil horizon. Soil moisture sensors do not measure water in the soil directly. Instead, they measure changes in some other soil property that is related to water content in a predictable way.

In this blind stick, the moisture sensor detects if water is contact with it and feed the information to the Arduino. It reads the information and alter user by saying ‘ALERT’ through the speaker.

4. Error Analysis:

All sensors are susceptible to errors, and it is crucial that we take these errors into account in order to be able to account for them and create a high-quality standard product. Due to the fact that some sensors do not distinguish between actual and expected readings, we have only calculated errors for UDS.

The error of the UDS is given by the formula:

$$\delta = \left| \frac{v_A - v_E}{v_E} \right| \cdot 100\%$$

δ = percent error

v_A = actual value observed

v_E = expected value

We have considered the set distance as expected value and the output from the Serial monitor by measurement of the UDS as actual value. This is done is MATLAB using the following code

CODE:

```
Expected_Value = [ 22.9, 18.1, 13.8, 19.7, 10.1];
Actual_Value = [ 22, 17, 13, 19, 9];
Error_Percentage =  (abs (Actual_Value-Expected_Value)./Expected_Value)*100
Average_Error_Percent = mean (Error_Percentage)
Accuracy = 100-Average_Error_Percent
```

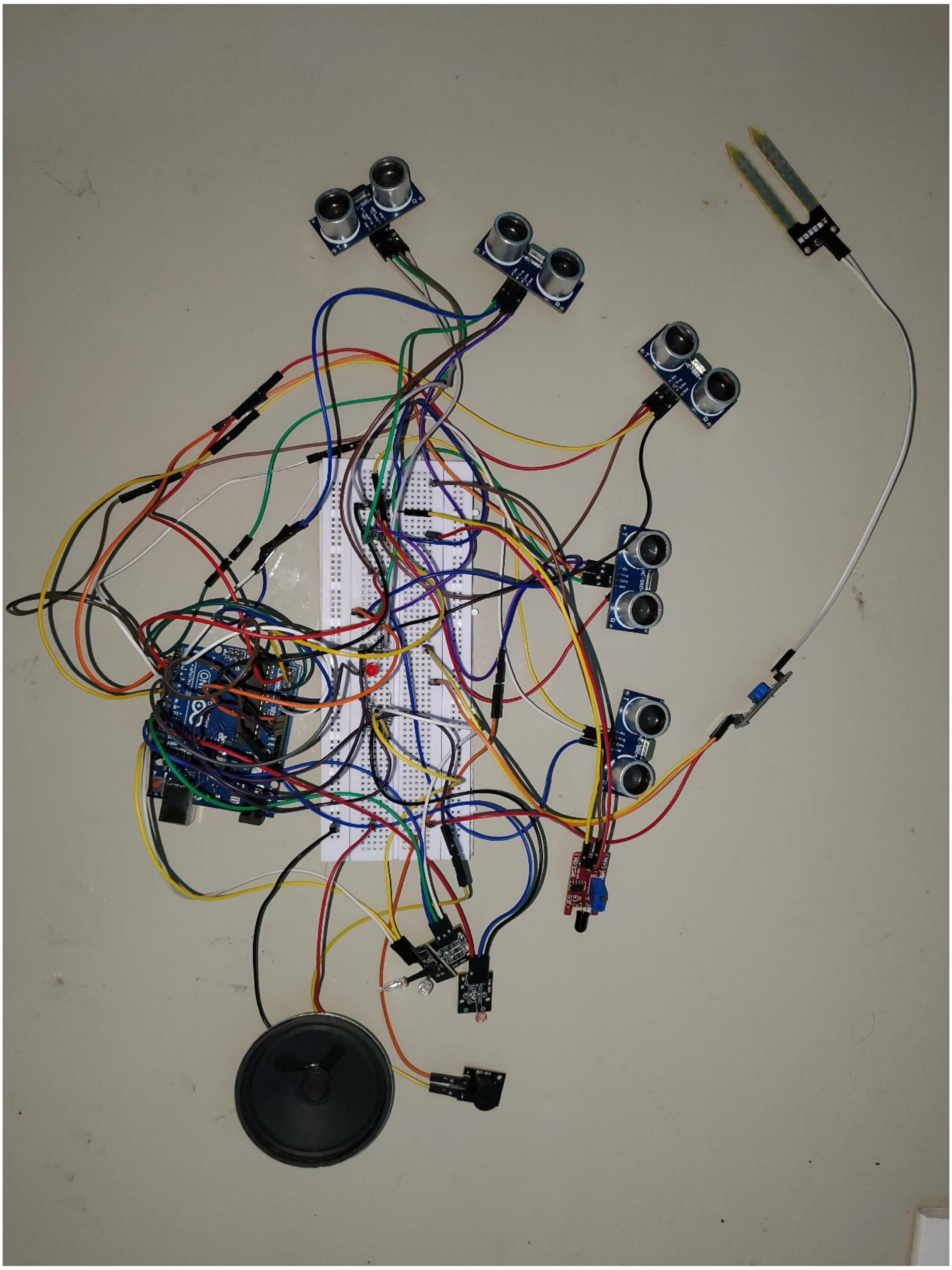
Error:

```
Error_Percentage = 1×5
     3.9301    6.0773    5.7971    3.5533   10.8911

Average_Error_Percent = 6.0498
Accuracy = 93.9502
```

We can see that the average error percentage is 6 %. The probability that this value will result in a false trigger is slim to none because we are only using it to calculate distance, and because it is a walking stick, it will serve its intended function in the case of fast-moving objects that would evade detection. Simply put, we can disregard this minor error in light of our project.

5. Model



6. Future work

It is possible to further extend this project, we can slightly modify the cane properties/features for example it could be used to support old people without vision problems, also helpful to correct bad human walking pattern, it could be also used by injured military veterans and also for it could be used in hospitals for temporary healing on some injuries to see how people recover and collect data etc.

We would also like to add the GSM and GPS modules to this project. But due to some signal issue i.e., the GSM module only works on 2G networks, here in our campus there is no strong 2G signals available and also GPS is not receiving the signal correctly. So, in future, if possible, we would like to add these things.

Bibliography

https://create.arduino.cc/projecthub/blackpanda856/play-music-using-arduino-uno-and-a-speaker-b94e4a

https://www.arduino.cc/reference/en/libraries/newping/

https://www.arduino.cc/reference/en/libraries/talkie/

https://github.com/ArminJo/Talkie/blob/master/src/Vocab_US_Large.h