Smart Blind Stick

Soumyajyoti Kumar

Abstract— The objective of this research paper is to come up with a solution using our knowledge to help the blind and visually impaired society people. The Smart Blind stick is based on Internet of Things (IOT) with Arduino board module. The proposed system detects obstacles in particular direction (left, right, front or down) with the help of ultrasonic sensor.[2]

Index Terms— Arduino, Arduino IDE, DC buzzer, IOT, LED light, Ultrasonic sensor.

1. INTRODUCTION

THE Blindness is a very common disability among the peoples throughout the world. According to the World Health Organization (WHO) 285 million people are visually impaired worldwide, 39 million are blind and 246 have low vision. About 90% of the world's visually impaired live in developing countries. For the indigents blindness is a curse. They need help to walk outside and all other daily essential works. So the paper glows a system that tries to remove the curse of blindness and make them self- dependent to do their daily chores. It is a walking stick, normally used by the blinds. But it is fully automated as well as manually operated, easy to maintain, cheap and it is very comfortable to use. The power consumption is low and can be operated easily. Above all the stick is very economic over the conventional one. The walking stick mentioned above is a stick that consists of a circuit board that contains a PIC micro controller, different sensors, and buzzer.

2. APPARATUS AND THERE USES

2.1 Hardware Part

1. **9V Battery Supply-**To give power supply to the model. (Quantity-1)
2. **PVC Pipe-** For making the stick. (Quantity-1)
3. **Arduino Uno R3-**To read inputs from the sensor and turn it out to an output. (Quantity-1)
4. **HC-SR04 Ultrasonic Sensor-**The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. (Quantity-1)
5. **Battery Snap Connector-**T This is Battery Snap
6. **Buzzer-**An active buzzer will generate a tone using an internal oscillator, so all that is needed is a DC voltage. (Quantity-1)
7. **Resistor**-Used to reduce current flow. (Quantity-1)
8. **LED Light-**Used to emit light in a specific command. (Quantity-1)
9. **Breadboard-**To make quick electrical connections between components- like resistors, LEDs. (Quantity-1)
10. **Jumper Wires-**An electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard. (Quantity-1)

2.2 Software Used

Arduino IDE

3. THEORY

The Blind stick is an innovative stick designed for visually disabled people for improved navigation. We here propose an advanced blind stick that allows visually challenged people to navigate with ease using advanced technology. The blind stick is integrated with ultrasonic sensor. My proposed project first uses ultrasonic sensors to detect obstacles ahead using ultrasonic waves. On sensing obstacles the sensor passes this data to the microcontroller. The microcontroller then processes this data and calculates if the obstacle is close enough. If the obstacle is not that close the circuit does nothing. If the obstacle is close the microcontroller sends a signal to sound a buzzer.

Initial paragraphs after the section title are not indented. Only the initial, introductory paragraph has a drop cap.

4. CIRCUIT DIAGRAM

1. Connect the echo and trigger terminal of the sensor to 10^{th} and 9^{th} digital pin of Arduino respectively.

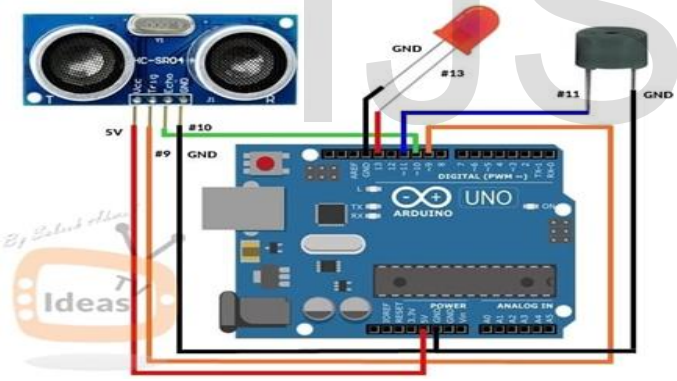
2 Connect the three terminals of buzzer to the breadboard.

3 Connect the Vcc and ground of the buzzer to the 5V and ground of power in Arduino respectively.

4 Connect the Vcc and ground of sensor in series with the Vcc and ground of the buzzer.

• *Soumyajyoti Kumar is currently pursuing bachelor degree program in electrical engineering in MAKAUT, India, PH-8335987516. E-mail: soumyajyotikumarofficial@gmail.com*

5 Connect the I/O pin of buzzer to the 8th digital pin of Arduino.
6 Connect the anode of LED to the 13th digital pin of Arduino and cathode to the digita
7 l ground of Arduino.
8 Connect the resistance in series with the cathode of LED.
9 Connect the 9V battery to Arduino by snap connector.



5. Code With Explanation

5.1 Codes

```
const int speakerPin = 8;
const int trigPin = 9;
const int echoPin = 10;
const int ledPin = 13;

//variables
long duration;
int distance;
int safetyDistance;

void setup () {
 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
 pinMode (speakerPin, OUTPUT);
 Serial.begin(9600);
}
void loop () {
 digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
safetyDistance = distance;
if (safetyDistance <= 30){
 digitalWrite(speakerPin, LOW);
 digitalWrite(ledPin, HIGH);
}
else{
 digitalWrite(speakerPin, HIGH);
 digitalWrite(ledPin, LOW);
}
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

5.2 Explanation

Code snippet 1:

```
const int speakerPin = 8;
const int trigPin = 9;
const int echoPin = 10;
```

Explanation:
At first we defined the port of pins of HC-SR04 (i.e **trigPin, echoPin)** in the Arduino IDE. Another pin is **speakerPin** which defined the I/O port of 5 V active alarm buzzer module. They are connected in the digital ports 9,10,8 respectively in the Arduino UNO R3.

Code snippet 2:

```
long duration;
int distance;
int safetyDistance;
```

Explanation:
We defined the variables i.e **duration, distance, safetyDistance.**

Code snippet 3:

```
void setup () {
```

```
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
pinMode (speakerPin, OUTPUT);
 Serial.begin(9600);
}
```

Explanation:

In this setup() method, we used the **pinMode** command to configure the pin as an input and output. We took the trigPin of the sensor as output ,echoPin as input and speakerPin as output.

Serial.begin(9600) command is used to start serial communication, so that the Arduino can send out commands through the USB connection. 9600 is called the **"baud rate".** It means then data rate is 9600 bps.

Code snippet 4:

```
void loop () {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
safetyDistance = distance;
if (safetyDistance <= 30){
 digitalWrite(speakerPin, LOW);
}
else{
digitalWrite(speakerPin, HIGH);
}
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

Explanation:

The **digitalWrite()** function is used to write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode(), its voltage will be set to the corresponding value: 5V for HIGH(here), 0V (ground) for LOW. If the pin is configured as an INPUT, digitalWrite() will enable (HIGH) or disable (LOW) the internal pullup on the input pin.

delayMicroseconds() pauses the program for the amount of time (in microseconds) specified by the parameter.

We used the trigPin as output and set in HIGH value for 10 microseconds and set in LOW value for 2 microseconds.

Now we used the **duration** variable and took input using **pulsePin** command from the echopin port connected to the sensor.

To measure the distance the sound has travelled we use the formula: Distance = (Time x SpeedOfSound) / 2. The "2" is in the formula because the sound has to travel back and forth. First the sound travels away from the sensor, and then it bounces off of a surface and returns back

Here, distance = duration * 0.034/2

We used 0.034 because at 20°C, the speed of sound is roughly 343 m/s or 0.034 cm/µs.

Then , we used an if and else loop to restrict a distance range that the sensor will follow every time(Here, 30 cm).We used the safetyDistance variable which is predefined. If the object is in between the range of 30cm , the speakerPin of buzzer will set to LOW

and buzzer will create a tone which will indicate the object is in the range of the sensor and if the object is out of the range, then speakerPin will set to HIGH value and buzzer will not create any tone.

At last, we print the distance of the object indicated by the sensor and we can see the distance in the serial monitor of Arduino.

.

Analog Ground pin-Grounding pin is **used** almost exclusively as a safety measure.

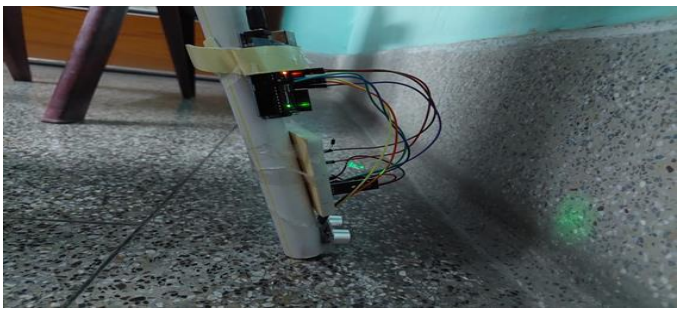
Analog 5V pin-Regulated power supply used to power microcontroller and other components on the board.

Digital Ground pin- Grounding pin is **used** almost exclusively as a safety measure.

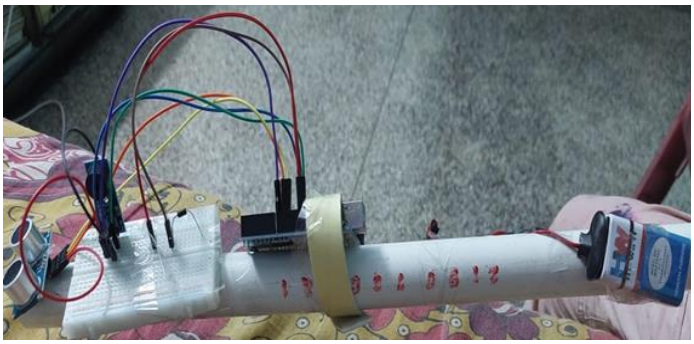
Digital 13th pin-*Digital pin 13* an OUTPUT and toggles it by alternating between HIGH and LOW at one second pace.

6. RESULTS

6.1 Figure-1



6.2 Figure-2



7. Conclusion

Smart blind stick is a modern and smart solution of problems faced by blind people in their life. This stick is made with sensor and if it is constructed with given architecture then it will help blind people to move without any human assistance. It is an attempt to make life of blind people easier and independent.[2]

Acknowledgment

The authors wish to thank my parents and my elder brother for their support. This work was supported in part by one of my institution friend.

References

[1]Smart walking stick - An electronic approach to assist visually disabled persons Mohammad Hazzaz Mahmud, RanaSaha, Sayemul Islam.
[2] SMART BLIND HELPING STICK USING IOT AND ANDROID Srishti Agrawal , Sakshi Vaval , Varsha Chawla , Ms Nupur Agrawal, Ms Kavita Namdev.