Group - 09

NAVIGATIONAL AID FOR VISUALLY CHALLENGED PEOPLE

INTRODUCTION

Our project delivers a Navigational tool (Smart Cane) for Visually Challenged people!!

Functionalities we have implemented are:

- Navigation on junctions (using RC522 RFID Module)
- Path Tracker (Using IR sensor and -> 8 inches path)
- Obstacle Detection (using Ultrasonic Sensor and Buzzer)
- Instructions via Voice command (using speaker module)

Implementation

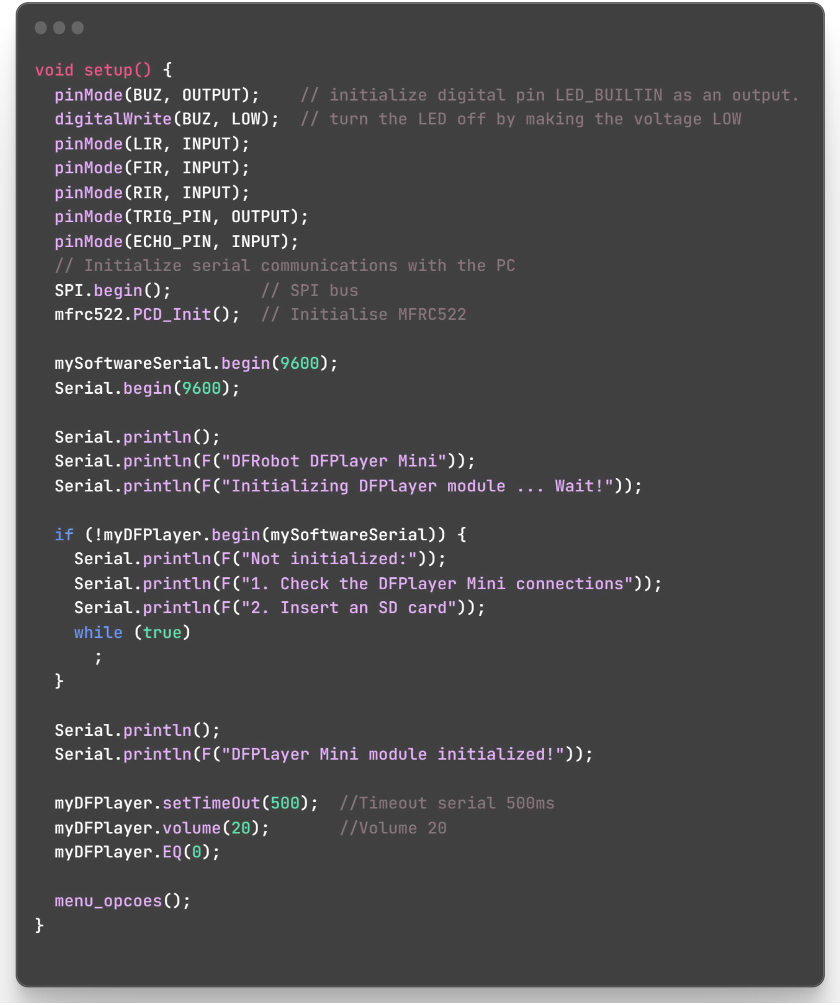
Hardware used:

- Arduino UNO
- Memory unit
- UltraSonic Sensor (HC-SR04)
- IR Sensors -3 sensors
- Buzzer
- Rfid Sensor and Passive Tags
- Speaker (audio module)

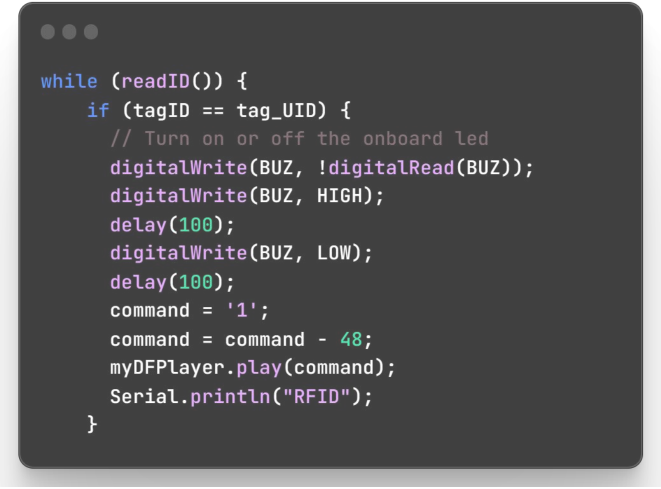
Code

includes several libraries such as

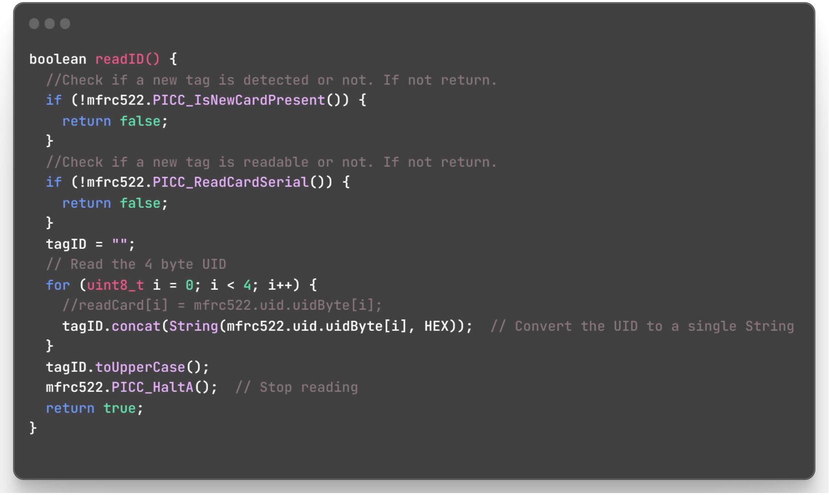
- SPI - for communication between MFRC522 RFID module and Arduino
- MFRC522 - module used to read RFID tags.
- SoftwareSerial - For communication between Audio module and Arduino
- DFRobot DFPlayer Mini - Audio module to play
- and NewPing. - Used for IR Sensor



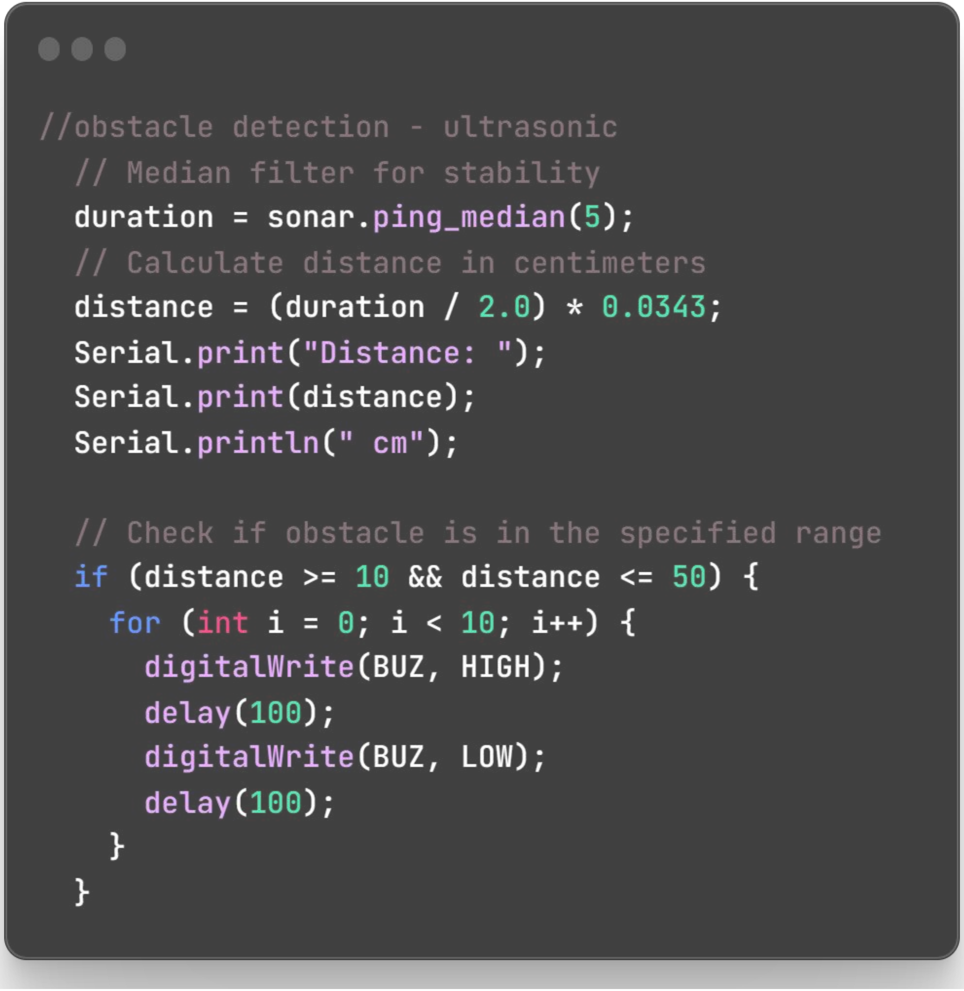
```
void setup() {
  pinMode(BUZ, OUTPUT);    // initialize digital pin LED_BUILTIN as an output.
  digitalWrite(BUZ, LOW);  // turn the LED off by making the voltage LOW
  pinMode(LIR, INPUT);
  pinMode(FIR, INPUT);
  pinMode(RIR, INPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  // Initialize serial communications with the PC
  SPI.begin();         // SPI bus
  mfrc522.PCD_Init();  // Initialise MFRC522

  mySoftwareSerial.begin(9600);
  Serial.begin(9600);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini"));
  Serial.println(F("Initializing DFPlayer module ... Wait!"));

  if (!myDFPlayer.begin(mySoftwareSerial)) {
    Serial.println(F("Not initialized:"));
    Serial.println(F("1. Check the DFPlayer Mini connections"));
    Serial.println(F("2. Insert an SD card"));
    while (true)
      ;
  }

  Serial.println();
  Serial.println(F("DFPlayer Mini module initialized!"));

  myDFPlayer.setTimeOut(500);  //Timeout serial 500ms
  myDFPlayer.volume(20);       //Volume 20
  myDFPlayer.EQ(0);

  menu_opcoes();
}
```



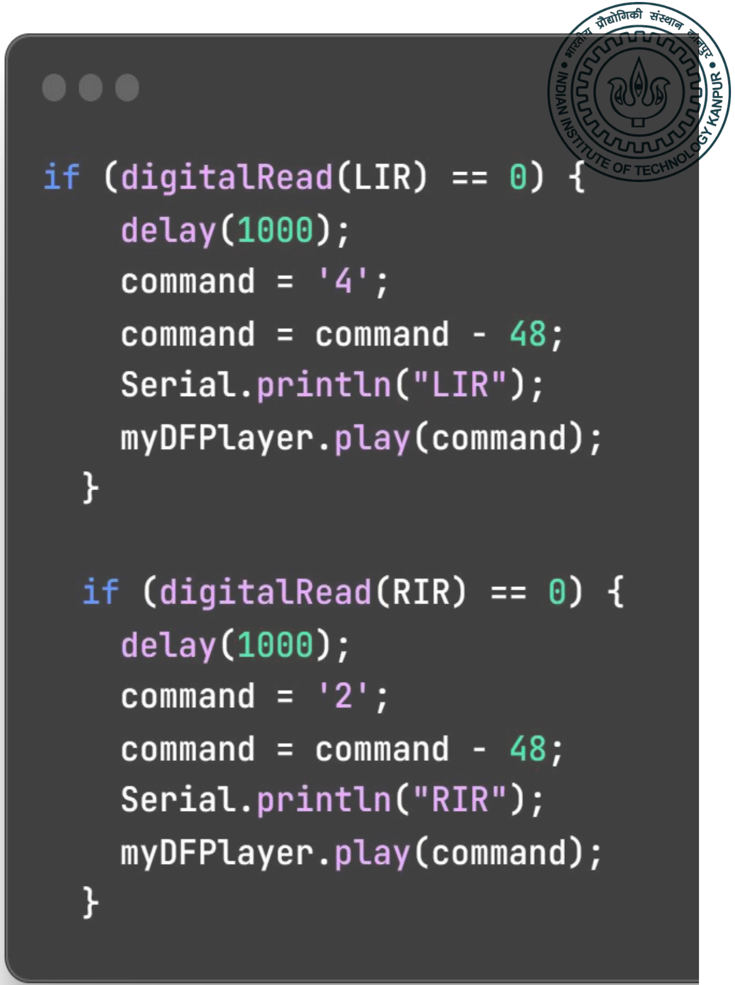
```
while (readID()) {
    if (tagID == tag_UID) {
      // Turn on or off the onboard led
      digitalWrite(BUZ, !digitalRead(BUZ));
      digitalWrite(BUZ, HIGH);
      delay(100);
      digitalWrite(BUZ, LOW);
      delay(100);
      command = '1';
      command = command - 48;
      myDFPlayer.play(command);
      Serial.println("RFID");
    }
```



```
boolean readID() {
  //Check if a new tag is detected or not. If not return.
  if (!mfrc522.PICC_IsNewCardPresent()) {
    return false;
  }
  //Check if a new tag is readable or not. If not return.
  if (!mfrc522.PICC_ReadCardSerial()) {
    return false;
  }
  tagID = "";
  // Read the 4 byte UID
  for (uint8_t i = 0; i < 4; i++) {
    //readCard[i] = mfrc522.uid.uidByte[i];
    tagID.concat(String(mfrc522.uid.uidByte[i], HEX));  // Convert the UID to a single String
  }
  tagID.toUpperCase();
  mfrc522.PICC_HaltA();  // Stop reading
  return true;
}
```



```
//obstacle detection - ultrasonic
  // Median filter for stability
  duration = sonar.ping_median(5);
  // Calculate distance in centimeters
  distance = (duration / 2.0) * 0.0343;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  // Check if obstacle is in the specified range
  if (distance >= 10 && distance <= 50) {
    for (int i = 0; i < 10; i++) {
      digitalWrite(BUZ, HIGH);
      delay(100);
      digitalWrite(BUZ, LOW);
      delay(100);
    }
  }
```



```
if (digitalRead(LIR) == 0) {
    delay(1000);
    command = '4';
    command = command - 48;
    Serial.println("LIR");
    myDFPlayer.play(command);
  }

  if (digitalRead(RIR) == 0) {
    delay(1000);
    command = '2';
    command = command - 48;
    Serial.println("RIR");
    myDFPlayer.play(command);
  }
```

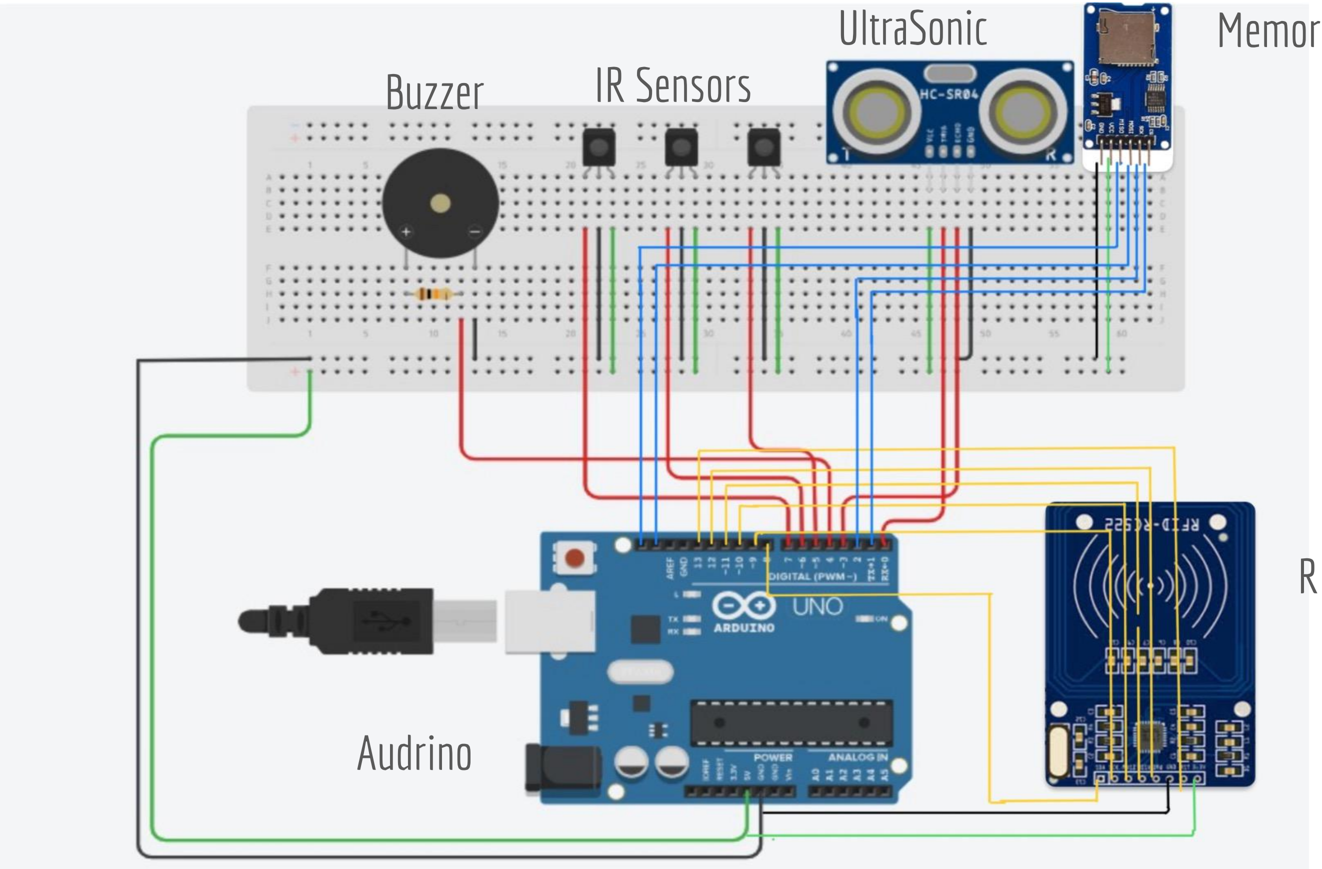
Mid-Project Alterations

- Unsuccessful implementation of Bluetooth Sensor.
- Initially aimed at executing audio responses through handheld device
- Shifted strategy to relay commands stored in memory unit through external speaker
- Commands correlated with sensor inputs to trigger corresponding sounds from the external speaker.
- Buzzer mapped to inputs from the ultrasonic sensor.

Obstacle Detection

- Obstacles detection in range of 10 - 50 cm. With the help of ULtraSonic Sensor .
- Will Usually fit at bottom part of Stick.
    - But a second Ultrasonic sensor could be used to detect the overhead Obstacles.
    - With the detection range of 40 cm to 100cm.
- On Receiving any obstacle within range, Feedback provided by Buzzer.
- Buzzer Set to beez 10 times within 2 seconds.
- To differentiate between buzz of RFID tag and ULtrasonic, it is implemented like this.

Circuit Diagram



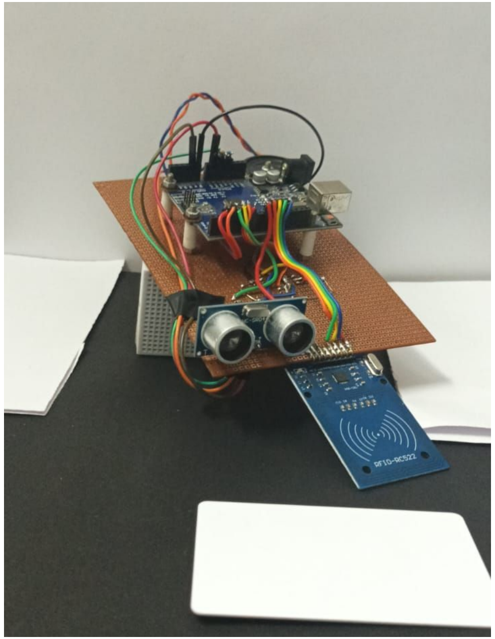
Path Tracking

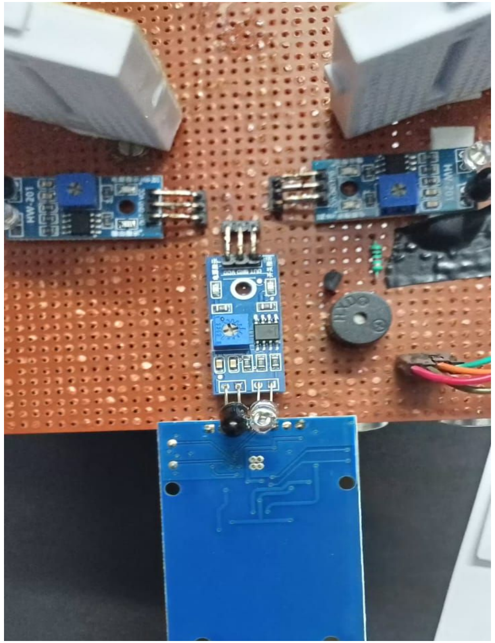
- For the path tracking we are using 3 IR Sensors
- Two IR Sensors are mounted on either side and one at the front.
- Sensors on the sides helps provide feedback for any alterations either on left or right
- Forward IR Sensor helps keep the stick aligned on the direction and path end detection.
- Pre-recorded audio feedbacks are provided to the user on any deviation.

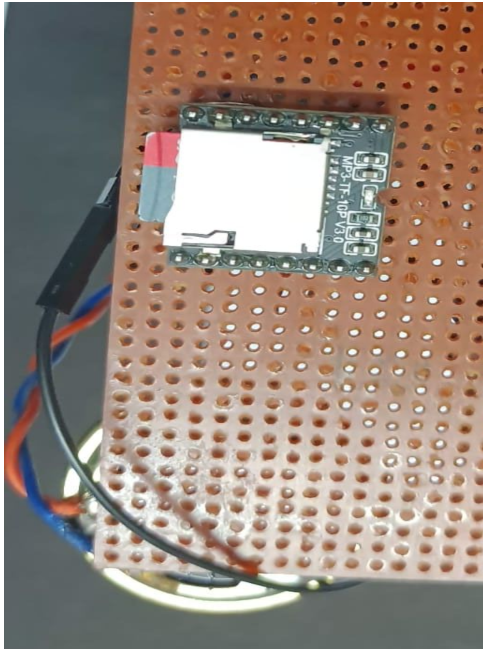
Junction Nodes with RFID Tags

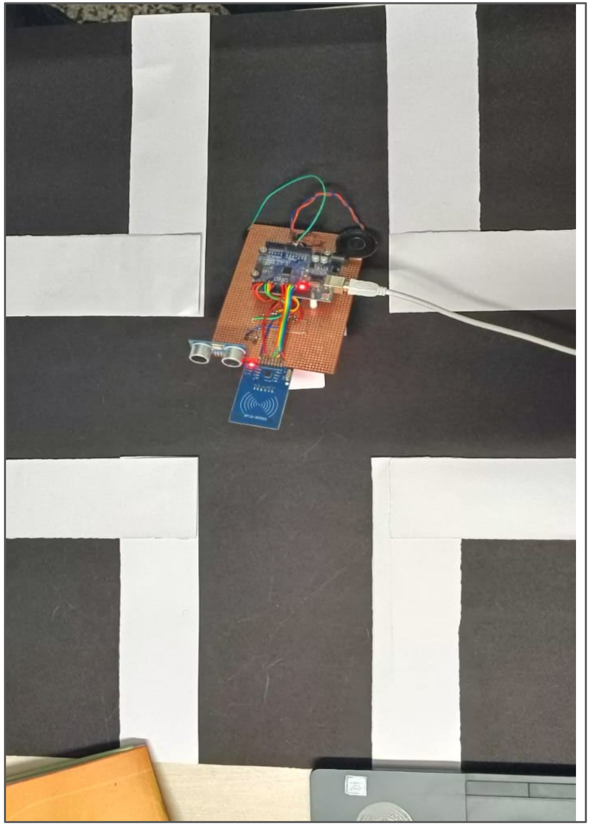
- For Junctions we are using RFID Tags and Receivers.
- For demonstration we have used only one Junction.
- RFID Sensor mounted on the front will detect any RFID Tags placed on the path.
- Upon detecting it will identify the particular junction and play the pre-recorded message accordingly.
- Multiple RFID Tags required for the entire bi-directional guidance system.

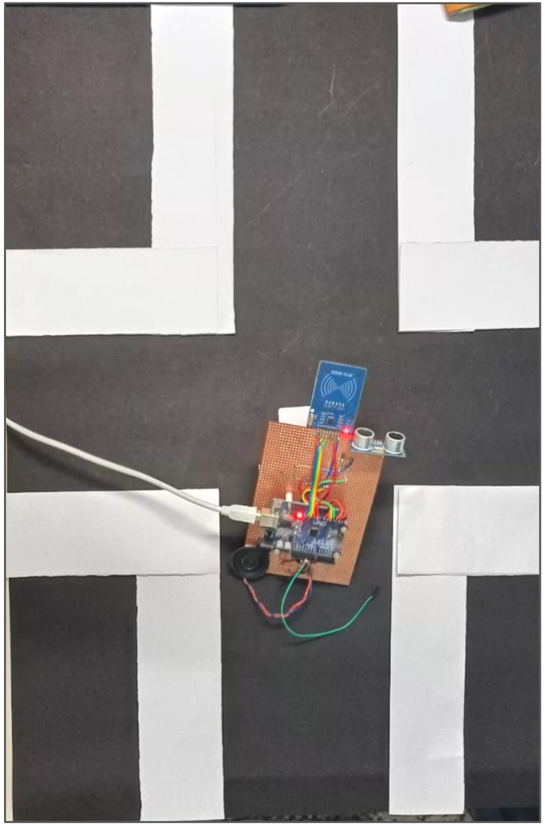
Project Hardware Snapshots



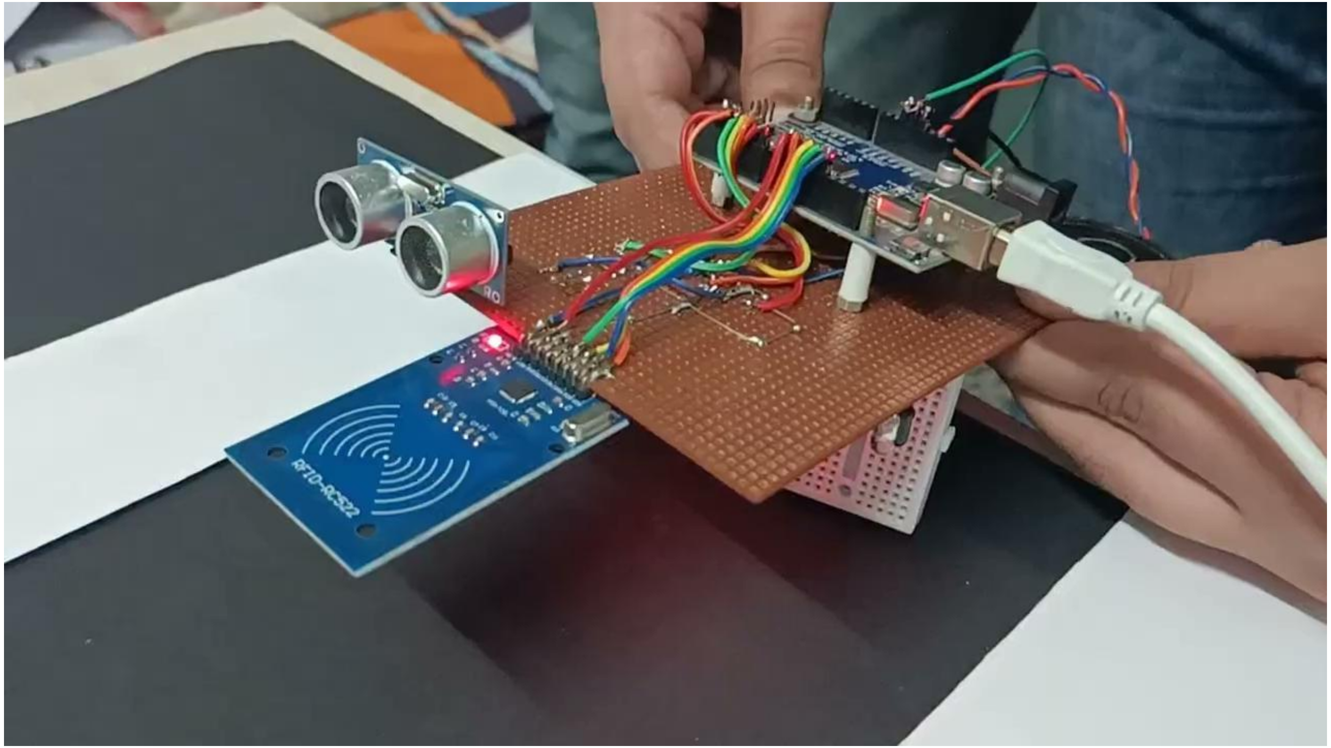








Training Demo Video



Precautions based on Observations

- RFID Module scanner fails to scan unless correctly aligned.
- The modules should be operated slowly, Infrared sensor prone to errors while detecting colors.
- Project working under optimal conditions.

Future Scope

The following points outline potential avenues for expansion, leveraging advanced technologies and thoughtful design considerations to further enhance the functionality and accessibility of our navigational aid.

- Machine Learning for Object Recognition:
    - Integrate machine learning algorithms for real-time object recognition using a camera.
    - Provide feedback about the types of objects in the user's path.
- Emergency Assistance Features:
    - Integrate features for emergency assistance, such as an SOS button that notifies predefined contacts or emergency services.
- Crowdsourced Navigation Data:
    - Explore the possibility of crowdsourcing navigation data to create a comprehensive and up-to-date database of user-generated insights

Acknowledgements

We would like to extend our sincere thanks to the course instructor **Dr. Amitangshu Pal.**

Also, we would like to sincerely thank all our team members for their immense efforts and commitment to this project.

Special thanks to Group 10 for helping us out and sharing ideas.

Thanks!

Akashdeep Singh

Abhishek Dubey

Darshan Jain

Nitish Kumar

Shaurya Agarwal

Shubham Srivastava