Project Report - Programming Embedded Systems

FALL 2023 – CS4111

Engineering College – COMPUTER SCIENCE DEPARTMENT

INSTRUCTORS NAME: Dr. Zain Balfagih		
COURSE NAME: Programming Embedded Systems	COURSE NUMBER: CS4111	SECTION: 1

Names: Haddel Balahmar, Deema AHmidah, Shumokh Abdullah, Wejdan Alshateri

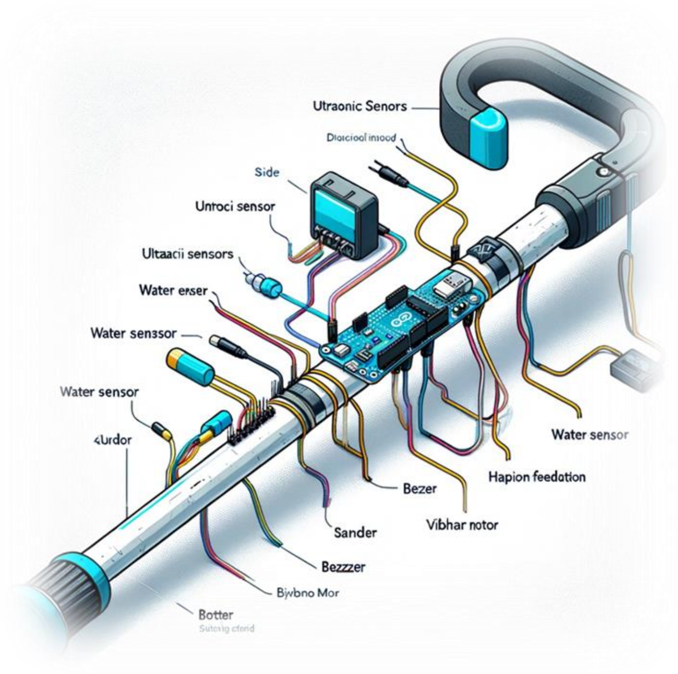
SMART BLIND STICK USING AURDINO

1. Introduction:

   Our advanced smart walking stick helps visually impaired people to perform navigation and to do their work independently and comfortably. Visually impaired people often face challenges to orient themselves in unfamiliar situations without relying on over. Another concern for them is their personal safety because they might not be able to see moving items behind them, which could cause them to trip and fall or hurt themselves. With approximately 285 million visually impaired people worldwide, coming up with these innovative solutions like our advanced smart stick are essential to overcome these challenges.

   In our design, we used 3 ultrasoni sensors one on the right, one on the left, and one in the front. Alongside these sensors, we have used buzzer so that it tells the visually empaired that there is an obstacle in front, providing an auditory cue to aid in safe navigation. Understanding that some users may have hearing difficulties or be in noisy environments, we have also added a vibration feature to the stick. Moreover, we have integrated a water sensor into the design. This sensor is specifically aimed at detecting wet or slippery surfaces, which are crucial for preventing slips and falls. In our implementation, all the envisioned features of the smart stick, including the three ultrasonic sensors, buzzer, vibration mechanism, and water sensor, are fully operational and functioning as intended.

2. Hardware:



The major components used are:

- Arduino Uno: Acts as the central processing unit of the device.
- Ultrasonic sensor: Serves as the 'eyes' of the system, providing spatial awareness.
- Water Sensor: Detects the presence of water on surfaces.
- Buzzer: Functions as an auditory signaling device.
- Vibration Motor: Delivers haptic feedback.
- 9V Battery: Supplies power to the system, ensuring portability.
- Connecting wires: Facilitate the electrical connections between components.

(a) Embedded Board Description:

The core of the system is an Arduino Uno, a versatile microcontroller board based on the ATmega328P. It provides a robust platform for interfacing with various sensors and output devices.
The Uno forms a robust platform, ensuring seamless interaction with the numerous components required for our project, and is especially suited for applications such as a smart blind stick, where reliability and ease of use are paramount.
it has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, a reset button, and more.

(b) Input device description:

- Ultrasonic Sensors

We use three HC-SR04 ultrasonic sensors that measure distances by emitting ultrasonic waves and detecting the reflection off objects. The sensors connect to the Arduino's digital pins and are powered by the board's 5V supply.

- Water Sensor

A water sensor detects the conductivity changes caused by moisture and outputs an analog signal to the Arduino's analog input pin.

(c) Output Device description:

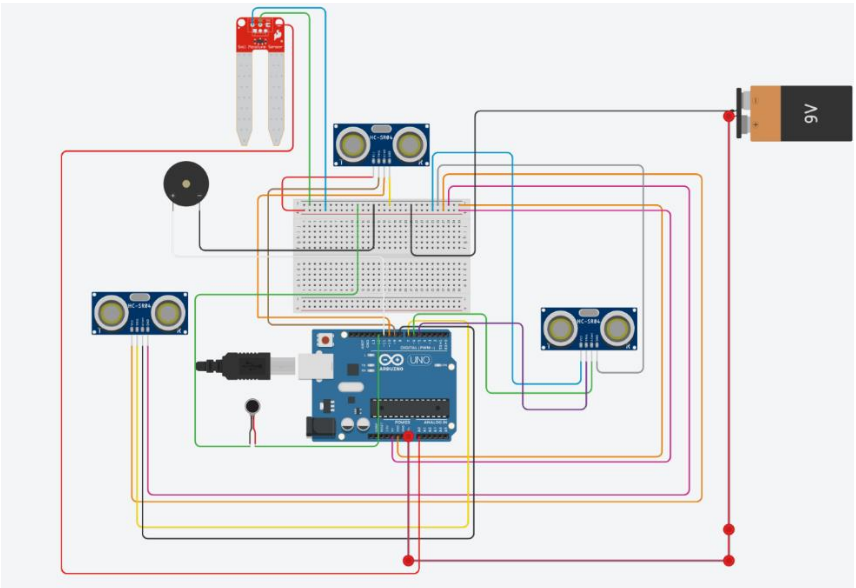
- Buzzer

A buzzer provides auditory feedback when obstacles are detected within a predefined safety range. It's connected to a digital pin on the Arduino.

- Vibration Motor

A vibration motor offers tactile feedback and is triggered under the same conditions as the buzzer. It's also connected to a digital pin and controlled programmatically.

(d) Connections:



(Note: In our actual project, we employ a water sensor. However, in Tinkercad, we have substituted the water sensor with a Soil Moisture Sensor simply because a water sensor was not available within the Tinkercad platform.)

Each Ultrasonic Sensor is connected to the Arduino with:
- VCC to 5V for power
- GND to ground
- TRIG to digital pins 9, 5, and 7 for sensors 1, 2, and 3, respectively
- ECHO to digital pins 10, 6, and 8 for sensors 1, 2, and 3, respectively

The Buzzer has its positive leg connected to digital pin 11 and its negative leg to ground, allowing it to be controlled via software to generate sound.

The Vibration Motor is connected to digital pin 12 for activation and to ground, providing tactile feedback in synchronization with the buzzer.

The 9V Battery connects with its positive side to the VIN pin, providing necessary operating voltage, and the negative side to the GND pin to complete the circuit.

Lastly, the Water Sensor outputs an analog signal to the Arduino's analog input pin, allowing the system to respond to changes in moisture levels detected.

3. Software:

(a) Sketch:

BlindStickProject.ino

```
//Haddel Balahmar,Deema AHmidah,Shumokh Abdullah,Wejdan Alshateri
// Define pins for the ultrasonic sensors
const int trigPin1 = 9;
const int echoPin1 = 10;
const int trigPin2 = 5;
const int echoPin2 = 6;
const int trigPin3 = 7;
const int echoPin3 = 8;

// Define pins for the buzzer and motor
const int buzzer = 11;
const int motorPin = 12;

// Define the pin for the water sensor
const int waterLevelPin = A0;

// Variables to store the duration and distance
long duration1, duration2, duration3;
int distance1, distance2, distance3;
int safetyDistance;

// Variable to store the water level
int waterLevel;

// Threshold for water level to detect water
const int waterThreshold = 300; // Adjust this value based on your calibration

// Melody for water detection - notes in the melody:
int melodyWater[] = {262, 294, 330, 349, 392, 440, 494, 523};
int waterNoteDurations[] = {4, 4, 4, 4, 4, 4, 4, 4}; // Duration for each note

// Melody for obstacle detection - notes in the melody:
int melodyObstacle[] = {523, 494, 440, 392, 349, 330, 294, 262};
int obstacleNoteDurations[] = {4, 4, 4, 4, 4, 4, 4, 4}; // Duration for each note

// Function to play melody
void playMelody(int melody[], int noteDurations[], int size) {
  for (int thisNote = 0; thisNote < size; thisNote++) {
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(buzzer, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(buzzer);
  }
}

```

Project Report - Programming Embedded Systems

FALL 2023 – CS4111

BlindStickProject.ino

```
void setup() {
  // Initialize the ultrasonic sensors
  pinMode(trigPin1, OUTPUT); pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT); pinMode(echoPin2, INPUT);
  pinMode(trigPin3, OUTPUT); pinMode(echoPin3, INPUT);

  // Initialize the buzzer and motor pin as output
  pinMode(buzzer, OUTPUT); pinMode(motorPin, OUTPUT);

  // Initialize the water sensor pin as input
  pinMode(waterLevelPin, INPUT);

  // Begin serial communication at a baud rate of 9600
  Serial.begin(9600);
}

void loop() {
  // Measure distance from each ultrasonic sensor
  distance1 = calculateDistance(trigPin1, echoPin1);
  distance2 = calculateDistance(trigPin2, echoPin2);
  distance3 = calculateDistance(trigPin3, echoPin3);

  // Find the smallest distance
  safetyDistance = min(distance1, min(distance2, distance3));

  // Read the water level from the sensor
  waterLevel = analogRead(waterLevelPin);  // Change this line to match your actual pin assignment
```

BlindStickProject.ino

```
  waterLevel = analogRead(waterLevelPin);  // Change this line to match your actual pin assignment

// Check if the water level is above the threshold
if (waterLevel > waterThreshold) {  // Change this line to match your calibration
  playMelody(melodyWater, waterNoteDurations, 8); // Play water detection melody
  digitalWrite(motorPin, HIGH); // Activate the motor

  delay(300);  // 7 seconds in milliseconds

  noTone(buzzer); // Stop the buzzer sound
  digitalWrite(motorPin, LOW); // Deactivate the motor
}
// Else, if an obstacle is detected within the safety distance
else if (safetyDistance <= 30) {
  digitalWrite(buzzer, HIGH);
  digitalWrite(motorPin, HIGH); // Activate if an object is close
}
// If no water or obstacle is detected
else {
  digitalWrite(buzzer, LOW);
  digitalWrite(motorPin, LOW); // Deactivate if no object is close
}

  // Print distance values to serial monitor
  Serial.print("Distance1: "); Serial.println(distance1);
  Serial.print("Distance2: "); Serial.println(distance2);
  Serial.print("Distance3: "); Serial.println(distance3);
  // Print moisture level to serial monitor
  Serial.print("Water Level: "); Serial.println(waterLevel);

  // Short delay to avoid tone overlap
  delay(100);
}


// Function to calculate distance based on ultrasonic sensor readings
int calculateDistance(int trigPin, int echoPin) {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    long duration = pulseIn(echoPin, HIGH);  // Declare the local variable 'duration' here
    int distance = duration * 0.034 / 2;
    return distance;
} // <-- Added missing closing brace here
```

(b) Explain:

- Code Functionality:

1. Ultrasonic Sensor Integration:
- The code uses three ultrasonic sensors (pins 9, 5, 7 for TRIG and 10, 6, 8 for ECHO) to measure distances by emitting ultrasonic waves and detecting their reflections. This helps in identifying obstacles in the path of the user.
2. Water Sensor Functionality:
- A water sensor is included in the system and connected to the analog pin A0. It detects changes in conductivity caused by moisture, alerting the user to the presence of water.
3. Feedback System:
- The buzzer (connected to pin 11) and vibration motor (connected to pin 12) serve as feedback mechanisms. They are activated based on the proximity of obstacles or the detection of water.
4. Melody Implementation for Alerts:
- Two different melodies are coded for water detection and obstacle detection, played through the buzzer. This provides an auditory signal to the user about different types of hazards.
5. Water Level Monitoring:
- The water level is constantly monitored through the water sensor. If it exceeds a certain threshold (set at 300), a specific melody is played, and the motor is activated.
6. Obstacle Detection:
- When the shortest distance measured by any of the ultrasonic sensors is within the safety range (30 cm), both the buzzer and motor are activated, providing immediate feedback.

- Code Structure and Originality:

- Most of the code, including the integration of sensors and output devices, was written by our team. This involved writing custom functions like `calculateDistance` for distance measurement and `playMelody` for playing melodies.
- Standard Arduino functions like `tone` and `noTone` are used for buzzer control. These are part of the Arduino's built-in library and not externally sourced.
- The uniqueness of our code lies in how we have orchestrated the operation of various components to work in unison, providing real-time feedback to the user.

4. Related Work:

Smart Blind Stick has been developed several times previously with attempts to help blind people and enhance their quality of life. Our exploration into existing solutions revealed a range of smart sticks, but none quite like ours. Many of these sticks employ technology like ultrasonic sensors or GPS, yet our design stands out due to its unique combination of features.

Our smart stick has three ultrasonic sensors, not just one. These sensors give a much wider detection range, so users get a better idea of what's around them. We also put in a buzzer and a vibration feature. Together, they alert the user about things like obstacles or changes nearby.

The really unique thing about our stick is that it can detect water. This isn't something you usually find in other sticks. It's really handy because it can warn the user about wet and slippery surfaces.

Comparing our stick to others, there are some similarities in using tech to help with navigation, but our smart stick takes it a step further with all these sensors. We focused on making something safe, easy to use, and that helps people move around independently.

5. Conclusion:

- Team Contributions:

Our team, comprised of four dedicated members, displayed remarkable teamwork and commitment. We managed to effectively collaborate during activity times and breaks between classes, making the most of our available time.

- The first member was instrumental in programming the Arduino, ensuring the ultrasonic sensors functioned optimally for precise distance measurement.
- The second member focused on the water sensor, skillfully integrating it with the Arduino and fine-tuning it for accurate moisture detection.
- A third member concentrated on the feedback system, optimizing both the buzzer and the vibration motor to provide intuitive and discernible alerts.
- The fourth member was pivotal in assembling and testing the hardware, ensuring seamless connectivity and functionality of all components.

- Challenges Encountered:

- We faced several challenges during the project. Calibration of the ultrasonic sensors was initially problematic, leading to inconsistent distance measurements.
- Integrating the water sensor with the existing setup required fine-tuning to avoid false alarms.
- Ensuring that the vibration motor provided an appropriate level of feedback without overwhelming the user was also a hurdle.

- Future Work and Improvements:

Looking ahead, there are numerous opportunities to enhance and expand upon our project:
- Implementing GPS functionality to provide real-time location tracking, aiding in navigation and orientation for the user.
- Redesigning the stick for improved ergonomics and user comfort, making it more user-friendly and practical for everyday use.
- Incorporating Bluetooth technology with a headset to verbally alert the user about nearby objects or obstacles. This feature would provide a more intuitive and interactive experience, offering precise guidance based on the stick's sensor readings.
- Further development could also involve experimenting with more advanced sensor technologies for enhanced detection and feedback capabilities, ensuring a safer and more reliable navigation tool for the visually impaired.

- Final Thoughts:

This project not only served as an academic endeavor but also as a venture into creating a meaningful tool for the visually impaired community.
Our Smart Blind Stick, though in its initial phase, holds the promise of evolving into an invaluable aid, offering independence and confidence to its users.
We are committed to continuing our work, enhancing the stick's capabilities, and striving to make such technology widely accessible, transforming it from a mere concept into a practical solution for everyday challenges.

6. References:

[1] SMART BLIND STICK USING ARDUINO. (2021). International Journal of Advanced Research in Computer Science Engineering and Information Technology, 6, 2321–3337. https://www.isrjournals.org/journals/computerscience_information_technology_journals/smartblindstickusingarduino1619617702.pdf

[2] Grover, S., Hassan, A., Yashaswi, K., & Shinde, N. K. (2020). Smart Blind Stick. International Journal of Electronics and Communication Engineering, 7(5), 19–23. https://doi.org/10.14445/23488549/ijece-v7i5p104

[3] How to make Smart Blind Stick with Arduino | Arduino Project. (n.d.). Www.youtube.com. https://www.youtube.com/watch?v=_RpSaj9j-GY

[4] cpb. (2012, April 18). WHO releases new global estimates on visual impairment. World Health Organization - Regional Office for the Eastern Mediterranean. https://www.emro.who.int/control-and-preventions-of-blindness-and-deafness/announcements/global-estimates-on-visual-impairment.html