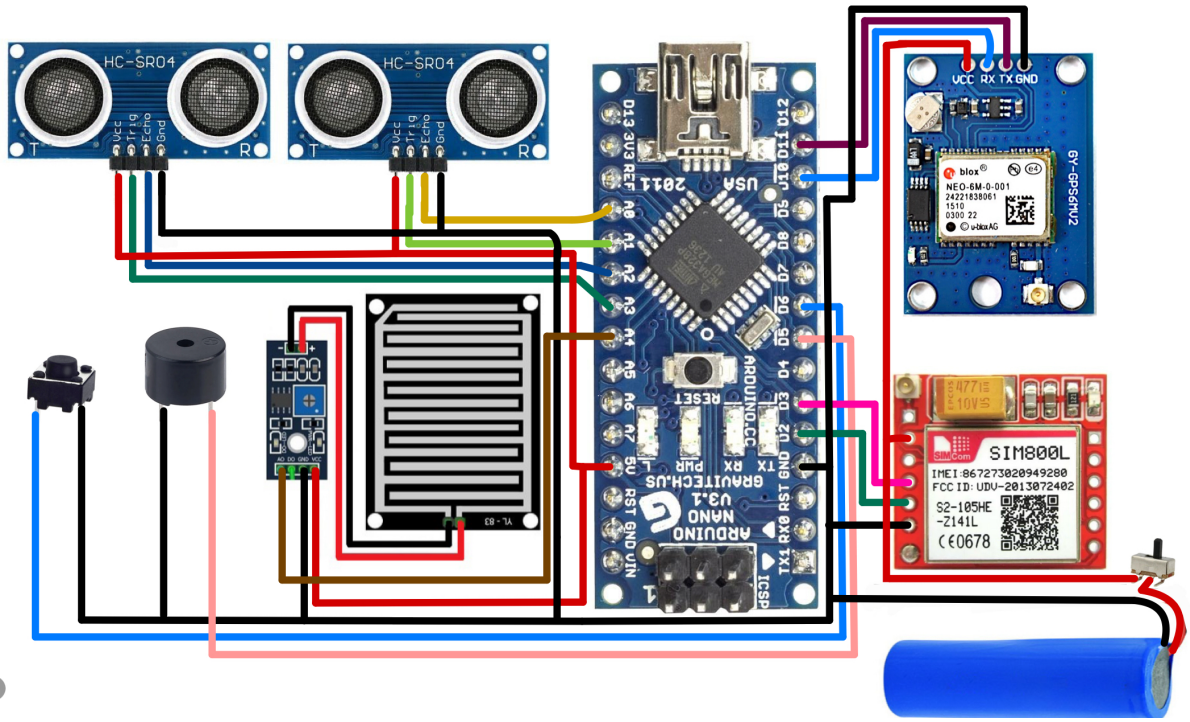
Blind Stick with GPS and GSM Using Arduino Nano

This project focuses on creating a blind stick that integrates multiple sensors and communication modules to enhance the mobility and safety of visually impaired individuals. The stick provides real-time distance measurement, water detection, GPS tracking, and emergency alerts through a GSM module. Below are the components, connections, and features described based on the schematic diagram provided.

Components Required

1. **Arduino Nano**: The central microcontroller that coordinates the data from sensors and communication modules.
2. **HC-SR04 Ultrasonic Sensors** (x2): These sensors measure the distance to objects in front of the stick, warning the user of obstacles.
3. **Rain Sensor**: Detects the presence of water or moisture, alerting the user to potential hazards like puddles or wet surfaces.
4. **Buzzer**: Provides audible feedback when an obstacle or water is detected.
5. **Push Button**: Can be used as an emergency button or to manually activate/deactivate certain features.
6. **NEO-6M GPS Module**: Captures the geographic location of the user for tracking purposes.
7. **SIM800L GSM Module**: Sends SMS alerts or location data via a GSM network.
8. **Battery**: Powers the circuit with 3.7V, typically provided by a lithium-ion battery.

Circuit Diagram



Wiring and Connections

- **HC-SR04 Sensors**:
    - **VCC** connected to the 5V pin of the Arduino Nano.
    - **GND** connected to the GND pin of the Arduino Nano.
    - **Echo** pins connected to Analog pins A0 and A1 of the Arduino for receiving distance data.
    - **Trig** pins connected to digital pins D8 and D9 for sending ultrasonic pulses.
- **Rain Sensor**:
    - **VCC** connected to 5V.
    - **GND** connected to GND.
    - **A4(Analog Output)** connected to the A4pin on the Arduino Nano for analog moisture detection.
- **Buzzer**:
    - **Positive terminal** connected to a digital pin (D5) on the Arduino for sound activation.
    - **Negative terminal** connected to GND.
- **Push Button**:
    - One side connected to GND.
    - Other side connected to digital pin D6 for detecting the button press.
- **NEO-6M GPS Module**:
    - **VCC** connected to 5V.
    - **GND** connected to GND.
    - **RX (Receiver)** connected to TX pin (D10) of the Arduino Nano for communication.
    - **TX (Transmitter)** connected to RX pin (D11) of the Arduino Nano.
- **SIM800L GSM Module**:
    - **VCC** connected to a regulated 4.2V supply (direct connection to a lithium-ion battery).
    - **GND** connected to GND.
    - **TX (Transmitter)** connected to D2 of the Arduino Nano.
    - **RX (Receiver)** connected to D3 for receiving commands from the Arduino.
- **Battery**:
    - **Positive terminal** connected to a switch to power the circuit.
    - **Negative terminal** connected to GND.

Functionality Overview

1. **Obstacle Detection**: The two ultrasonic sensors continuously measure the distance to nearby objects. If an object is detected within a predefined distance (e.g., less than 1 meter), the buzzer will sound to alert the user.

2. **Water Detection**: The rain sensor will trigger the buzzer when water is detected, warning the user to avoid wet surfaces or puddles.

3. **GPS Location**: The NEO-6M GPS module acquires the real-time location of the user, which can be accessed or transmitted via GSM.

4. **Emergency Alerts**: The GSM module can send an emergency SMS with the user's location in case of emergencies. The push button can be used to trigger this alert manually.

Additional Features

- **Low Power Consumption**: The circuit is designed to operate efficiently using a 3.7V lithium-ion battery, allowing the stick to function for long periods without frequent recharging.

- **Compact Design**: Using an Arduino Nano ensures a compact form factor, suitable for embedding into a walking stick.

Conclusion

This blind stick project combines multiple technologies (ultrasonic, rain sensors, GPS, and GSM) to assist visually impaired individuals by providing real-time obstacle detection, location tracking, and emergency communication. It can be further enhanced by adding vibration motors for tactile feedback or integrating a solar charging system for the battery.

CODE

```
#include <SoftwareSerial.h>
#include <TinyGPS++.h>

int buttonpin = 6;
float lattitude, longitude;
float a[2];
float *p;
SoftwareSerial gpsSerial(2, 3);
SoftwareSerial gsmSerial(10, 11);
TinyGPSPlus gps;

const int trigPin1 = A1;
const int echoPin1 = A0;
long duration1;
int distance1;
const int trigPin2 = A2;
const int echoPin2 = A3;
long duration2;
int distance2;

void setup() {
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(5, OUTPUT);
  pinMode(A4, INPUT_PULLUP);
  Serial.begin(9600);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  pinMode(buttonpin, INPUT_PULLUP);
  Serial.begin(9600);
  delay(1000);
  gpsSerial.begin(9600);
  delay(1000);
  gsmSerial.begin(9600);
  delay(1000);

  Serial.print("-Tracking-");
  Serial.print("***Location***");
  gsmSerial.println("AT+CNMI=2,2,0,0,0");
  delay(3000);
  Serial.print("Initializing……");
```

```
  delay(2000);
  Serial.print("System Ready  ");
  delay(1000);
}

void loop() {

  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = duration1 * 0.034 / 2;
  Serial.print("Distance1: ");
  Serial.println(distance1);
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);
  duration2 = pulseIn(echoPin2, HIGH);
  distance2 = duration2 * 0.034 / 2;
  Serial.print("Distance2: ");
  Serial.println(distance2);

  if (distance1 <= 20 || distance2 <= 20) {

    digitalWrite(5, HIGH);
    delay(1000);
    digitalWrite(5, LOW);
    digitalWrite(5, HIGH);
    delay(1000);
    digitalWrite(5, LOW);
    digitalWrite(5, HIGH);
    delay(1000);
    digitalWrite(5, LOW);

  }

  else if (distance1 <= 15 || distance2 <= 15) {

    digitalWrite(5, HIGH);
```

```
    delay(500);
    digitalWrite(5, LOW);
    digitalWrite(5, HIGH);
    delay(500);
    digitalWrite(5, LOW);
    digitalWrite(5, HIGH);
    delay(500);
    digitalWrite(5, LOW);
  }

  else
    digitalWrite(5, LOW);
  int sensorValue = digitalRead(A4);

  if (sensorValue == 1) {

    digitalWrite(5, HIGH);
    Serial.println("Rain Alert");
    delay(1500);
    digitalWrite(5, LOW);
    digitalWrite(5, HIGH);
    delay(1500);
  }

  else
    digitalWrite(5, LOW);
  if (digitalRead(buttonpin) == LOW) {
    Serial.println("button pressed");
    delay(2000);
    SendMessage();
  }
  if (gsmSerial.available() > 0)
    Serial.write(gsmSerial.read());
  while (gsmSerial.available()) {
    gsmSerial.read();
  }

  while (Serial.available()) {
    Serial.read();
  }
  get_gsm();
}
float *get_gps() {
```

```
  gpsSerial.listen();
  Serial.println("INSIDE get_gps");
  while (1) {
    while (gpsSerial.available() > 0) {
      gps.encode(gpsSerial.read());
    }
    if (gps.location.isUpdated()) {

      Serial.print("LAT=");
      Serial.println(gps.location.lat(), 6);
      Serial.print("LONG=");
      Serial.println(gps.location.lng(), 6);
      lattitude = gps.location.lat();
      longitude = gps.location.lng();
      break;
    }
  }
  a[0] = lattitude;
  a[1] = longitude;
  return a;
}

void get_gsm() {
  gsmSerial.listen();
  while (gsmSerial.available() > 0)

  {
    Serial.println("INSIDE gsmSerial.available");
    if (gsmSerial.find("Track")) {
      Serial.println("INSIDE track");
      gsmSerial.println("AT+CMGF=1");
      delay(1000);
      gsmSerial.println("AT+CMGS=\"+9188305848xx\"\r");
      delay(1000);
      p = get_gps();
      gsmSerial.listen();
      Serial.print("Your Car Location: ");
      gsmSerial.print("Your Car Location: ");

      Serial.print("LATTITUDE=");
      Serial.print(*p, 6);

      gsmSerial.print("LATTITUDE=");
```

```
      gsmSerial.print(*p, 6);
      gsmSerial.print(",");

      Serial.print("LONGITUDE=");
      Serial.print(*(p + 1), 6);

      gsmSerial.print("LONGITUDE=");
      gsmSerial.print(*(p + 1), 6);
      delay(100);
      gsmSerial.println((char)26);
      delay(1000);
    }
  }
}

void SendMessage() {
  gsmSerial.println("AT+CMGF=1");
  delay(1000);
  gsmSerial.println("AT+CMGS=\"+9188305848xx\"\r");
  delay(1000);
  gsmSerial.println("I Am In Problem Plz Help Me");
  delay(1000);
  p = get_gps();
  gsmSerial.listen();
  Serial.print("Your Position is : ");
  gsmSerial.print("position is : ");
  Serial.print("LATTITUDE=");
  Serial.print(*p, 6);
  gsmSerial.print("LATTITUDE=");
  gsmSerial.print(*p, 6);
  gsmSerial.print(",");
  Serial.print("LONGITUDE=");
  Serial.print(*(p + 1), 6);
  gsmSerial.print("LONGITUDE=");
  gsmSerial.print(*(p + 1), 6);
  delay(100);

  gsmSerial.println((char)26);
}
```