tell me the connection of each port and each part #include <Wire.h>

```
#include <MPU6050.h>
#include <AltSoftSerial.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <math.h>  // Required for trigonometric functions

const String EMERGENCY_PHONE = "+917488090721";

#define SIM_RX 2
#define SIM_TX 3
#define GPS_RX 8
#define GPS_TX 9
#define BUZZER 5

SoftwareSerial sim800(SIM_RX, SIM_TX);
AltSoftSerial gpsSerial;
TinyGPSPlus gps;
MPU6050 mpu;

String latitude, longitude;
bool impactDetected = false;

int16_t ax, ay, az;
int threshold = 800;

float roll, pitch;
float initialRoll = 0, initialPitch = 0;
bool baselineSet = false;
float tiltThreshold = 20.0;

unsigned long lastImpactTime = 0;
unsigned long alertDelay = 30000;
unsigned long lastCheckTime = 0;
unsigned long checkInterval = 100;
unsigned long lastTiltTime = 0;
unsigned long resetTiltDelay = 20000;  // 20s cooldown before resetting tilt baseline

void setup() {
  Serial.begin(9600);
  sim800.begin(9600);
  gpsSerial.begin(9600);

  pinMode(BUZZER, OUTPUT);
```

```
  Wire.begin();
  mpu.initialize();
  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed!");
    while (1);
  }
  Serial.println("MPU6050 initialized");

  mpu.setSleepEnabled(false);

  sim800.println("AT");
  delay(1000);
  sim800.println("ATE1");
  delay(1000);
  sim800.println("AT+CPIN?");
  delay(1000);
  sim800.println("AT+CMGF=1");
  delay(1000);
  sim800.println("AT+CNMI=1,1,0,0,0");
  delay(1000);
}

void loop() {
  if (millis() - lastCheckTime > checkInterval) {
    checkImpact();
    checkTilt();
    lastCheckTime = millis();
  }

  if (impactDetected) {
    Serial.println("⚠ Emergency Detected! Sending Alert...");
    getGps();
    digitalWrite(BUZZER, HIGH);
    delay(5000);
    digitalWrite(BUZZER, LOW);

    sendAlert();
    makeCall();

    impactDetected = false;
    lastTiltTime = millis();  // Store time of last tilt-based alert
  }
```

```
  // Reset tilt baseline after cooldown to allow future tilt detection
  if (baselineSet && (millis() - lastTiltTime > resetTiltDelay)) {
    resetBaseline();
  }
}

void checkImpact() {
  mpu.getAcceleration(&ax, &ay, &az);

  int diffX = abs(ax);
  int diffY = abs(ay);
  int diffZ = abs(az);

  if ((diffX > threshold || diffY > threshold || diffZ > threshold) &&
      (millis() - lastImpactTime > alertDelay)) {
    Serial.println("⚡ Impact Detected!");
    impactDetected = true;
    lastImpactTime = millis();
  }
}

void checkTilt() {
  mpu.getAcceleration(&ax, &ay, &az);

  roll = atan2(ay, az) * 180.0 / M_PI;
  pitch = atan2(-ax, sqrt(ay * ay + az * az)) * 180.0 / M_PI;

  if (!baselineSet) {
    initialRoll = roll;
    initialPitch = pitch;
    baselineSet = true;
  }

  float rollDeviation = abs(roll - initialRoll);
  float pitchDeviation = abs(pitch - initialPitch);

  Serial.print("Roll: "); Serial.print(roll);
  Serial.print(" | Pitch: "); Serial.print(pitch);
  Serial.print(" | ΔRoll: "); Serial.print(rollDeviation);
  Serial.print(" | ΔPitch: "); Serial.println(pitchDeviation);

  if (rollDeviation > tiltThreshold || pitchDeviation > tiltThreshold) {
    Serial.println("⚠ Tilt Detected (Deviation Over 20°)! Activating System...");
    impactDetected = true;
```

```
  }
}

// Reset the initial baseline after a delay
void resetBaseline() {
  Serial.println("🔄 Resetting baseline orientation...");
  initialRoll = roll;
  initialPitch = pitch;
  baselineSet = true;
  lastTiltTime = millis();
}

void getGps() {
  boolean newData = false;
  for (unsigned long start = millis(); millis() - start < 2000;) {
    while (gpsSerial.available()) {
      if (gps.encode(gpsSerial.read())) {
        newData = true;
        break;
      }
    }
  }

  if (newData) {
    latitude = String(gps.location.lat(), 6);
    longitude = String(gps.location.lng(), 6);
  } else {
    Serial.println("No GPS data available!");
    latitude = "0.0";
    longitude = "0.0";
  }

  Serial.print("Latitude: ");
  Serial.println(latitude);
  Serial.print("Longitude: ");
  Serial.println(longitude);
}

void sendAlert() {
  String sms_data = "🚨 Emergency Alert!!\r";
  sms_data += "Location: http://maps.google.com/maps?q=loc:";
  sms_data += latitude + "," + longitude;

  sendSms(sms_data);
```

```
}

void makeCall() {
  Serial.println("📞 Calling emergency number...");
  sim800.println("ATD" + EMERGENCY_PHONE + ";");
  delay(20000);
  sim800.println("ATH");
  delay(1000);
}

void sendSms(String text) {
  sim800.print("AT+CMGF=1\r");
  delay(1000);
  sim800.print("AT+CMGS=\"" + EMERGENCY_PHONE + "\"\r");
  delay(1000);
  sim800.print(text);
  delay(100);
  sim800.write(0x1A);
  delay(1000);
  Serial.println("📩 SMS Sent Successfully.");
}
```