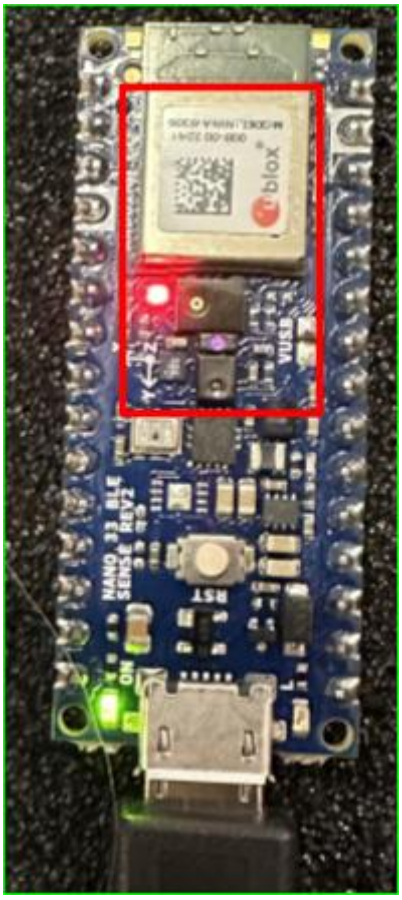
Arduino Nano 33 BLE Project

Proximity and Hand Gesture Sensor

Our project deals with using the gesture and proximity sensor, and accelerometer sensor. The Arduino board represents the car a person is driving. My hand represents another car on the road. The idea is when my hand gets too close to the proximity sensor around below 50, the light turns from green to red to signal that a car is about to crash into you. In the Arduino code, updateProximity function is where the value is being updated. In that function, you are reading in the data from the sensor and outputting it.

After the proximity value is below 50, it will start detecting hand gestures to signal which direction the car should steer. Hypothetically this would only work with an autonomous vehicle. Swipe left means to brake and steer left, swipe right means to brake and steer right, and swipe down means the car will just brake and not steer in any specified direction. When you swipe left, the LED lights up blue for a second to signal that you are steering left. When you swipe right, the LED lights up green for a second to signal that you are steering to the right. Digital write is where you are setting the voltage and controlling where the light should be on or off and what color.



Accelerometer and Gyroscope Sensor

So after the proximity sensor detects that a car is about to crash into you. You can choose whether to simulate that you were going at a fast speed by making a fast motion that signals your car was at a fast velocity at the time somebody is about to hit you. We will be using the x-axis to show that. So if the person holding the Arduino board makes a quick short

motion on the x-axis, it will print in the console that you are going too fast, and the car is about to tilt and has a risk of rollover. To prevent the car from being flipped over or upturned, the person holding the board will have to tilt the board in a fast motion. For example, if you swipe left to brake and steer left, the car will tilt to the right. To counteract that, the person holding the Arduino board will have to do in a quick motion tilt the board to the left. Same thing for the other direction. If you swipe right to brake suddenly and steer right at a fast speed, the car will be violently tilting toward the left and you have to tilt the board to the right at a fast motion to stabilize the car. If you swipe down, since you are just braking and the car could tilt upward, there is nothing in this case you can do to control the car.

```
Proximity: 232
Proximity: 231
Proximity: 227
Proximity: 212
Proximity: 198
Proximity: 168
Proximity: 156
Proximity: 121
Proximity: 100
Proximity: 85
Proximity: 59
Proximity: 53
Gesture: Detected RIGHT gesture: Brake and Steer Right
Accelerometer: Car is tilting too left!
Accelerometer: Tilt car right!
Gyroscope: Car stabilized
Gesture: Detected LEFT gesture: Brake and Steer Left
Accelerometer: Car is tilting too right!
Accelerometer: Tilt car left!
Accelerometer: Car is tilting too right!
Accelerometer: Tilt car left!
Gyroscope: Car stabilized
Gesture: Detected DOWN gesture: Brake and No Steer
Gesture: Detected UP gesture: Not used
```

Bluetooth Integration/Python Script

We integrated BLE communication by using the ArduinoBLE library. The whole point of the bluetooth is to have it connect with the python script so the data can be shown through the python console. So we added a service to the BLE. Inside the service, we have five characteristics to represent the five features it will have. This includes the proximity, gesture, accelerometer, gyroscope, and mic. Once we added that service to the BLE, we started advertising it. In the Python script, it contains the UUID for each characteristic as well as the one service. It also contains the device name too. In the Arduino code, we write a value to the characteristic which means sending the data through bluetooth to the python script, so that when you run the python script it prints the data.

```
Scanning for BLE devices...
Found device: None, Address: D7:2F:F8:9D:D1:FB
Found device: None, Address: 1C:AF:4A:34:5A:CD
Found device: None, Address: 51:E3:5A:7A:B1:71
Found device: None, Address: A0:D7:F3:55:C4:BB
Found device: None, Address: 45:23:42:4D:A2:3E
Found device: ProximitySensor, Address: 10:83:62:74:4D:F5
Connecting to ProximitySensor at address 10:83:62:74:4D:F5...
Connected to ProximitySensor!
Discovering services and characteristics...
Service: 00001800-0000-1000-8000-00805f9b34fb
Service: 00001801-0000-1000-8000-00805f9b34fb
Service: 00000000-5ec4-4083-81cd-a10b8d5cf6ec
  Characteristic: 00000001-5ec4-4083-81cd-a10b8d5cf6ec - Properties: ['read', 'notify']
Subscribing to proximity notifications...
  Characteristic: 00000002-5ec4-4083-81cd-a10b8d5cf6ec - Properties: ['read', 'notify']
Subscribing to gesture notifications...
  Characteristic: 00000003-5ec4-4083-81cd-a10b8d5cf6ec - Properties: ['read', 'notify']
Subscribing to gyroscope notifications...
  Characteristic: 00000004-5ec4-4083-81cd-a10b8d5cf6ec - Properties: ['read', 'notify']
Subscribing to accelerometer notifications...
  Characteristic: 00000005-5ec4-4083-81cd-a10b8d5cf6ec - Properties: ['read', 'notify']
Subscribing to microphone notifications...
Receiving data... Press Ctrl+C to exit.
Proximity: 248
Proximity: 246
Proximity: 248
```