# COMP6733 Project preliminary proposal

By: Jerome Hernandez (z5419632), Haopeng Pi (z5390194), Chen Yang (z5412354) and Terry Bai (z5430739)

# Topic Definition

## Project Outline

Traditional interaction methods with smart home devices, such as touchscreens and voice commands, can sometimes be limiting or inaccessible. This has presented the opportunity for innovative smart wearables that can detect and respond to gestures to seamlessly control smart home systems, such as lighting, temperature, entertainment units, etc.

## Statement of purpose

The purpose of this project is to develop a simple, working proof-of-concept for such a device described above, using the Arduino Nano 33 BLE Sense Rev 2 and its onboard capabilities for gesture recognition and control of Bluetooth-enabled smart home devices.

# Background

The research of Xu et al [1] demonstrates that a smartwatch can recognise a user's arm, hand and finger gestures through its built-in accelerometer and gyroscope sensors. They also demonstrated gesture recognition of characters written with an index finger with close to 95% accuracy. The equipment used was a Shimmer wristband, however, the Arduino Nano is equipped with similar IMU sensors to recreate such functionality. Other research has [3] found that an Arduino fixed on gloves can be used to capture features such as motion energy, gesture and shape to improve model recognition.

And, while machine-vision based gesture recognition methods are well established [2], they are not as common and plausible, due to relatively high costs.

Additionally, commercial products also demonstrate the feasibility of our project. For example; the Control4 [4] offers advanced smart home control with optional gesture control. And, the Myo Armband [5] - as a wearable that can recognise hand gestures using electrical signals and motion sensors - could potentially be integrated with smart home systems.

As recognised in similar concepts, such a device proposes two primary challenges: achieving functionality with a minimal amount of devices, and distinguishing between unintentional hand movements and intentional control gestures.

# Project plan

## Resources

- **Arduino Nano 33 BLE Sense Rev2**: A small, versatile microcontroller board designed for prototyping and development. It will serve as the primary hardware platform for the gesture-detecting wearable, handling sensor data acquisition and BLE communication.
- **BMI270:** A low power inertial measurement unit (IMU) with precise acceleration and angular rate (gyroscopic) measurement and intelligent on-chip motion-triggered interrupt features. It will be used as the primary sensor for collecting motion data.
- **APDS9960:** A digital proximity, ambient light, RGB, and gesture sensor on the Arduino Nano. May additionally be used to detect specific gestures to complement the BMI270.
- **BLE (Bluetooth Low Energy)**: An open standard for low-power wireless communication. Enables the wearable device to communicate and send control commands to similarly Bluetooth-enabled smart-home devices.
- **Machine Learning**: Lightweight machine learning will be used to process sensor data and build an algorithm to classify sensor data as different gestures in real-time.
- **MicroPython**: A lean and efficient implementation of Python 3 for microcontrollers. Will be used as the primary programming language for developing and deploying code.

## Method

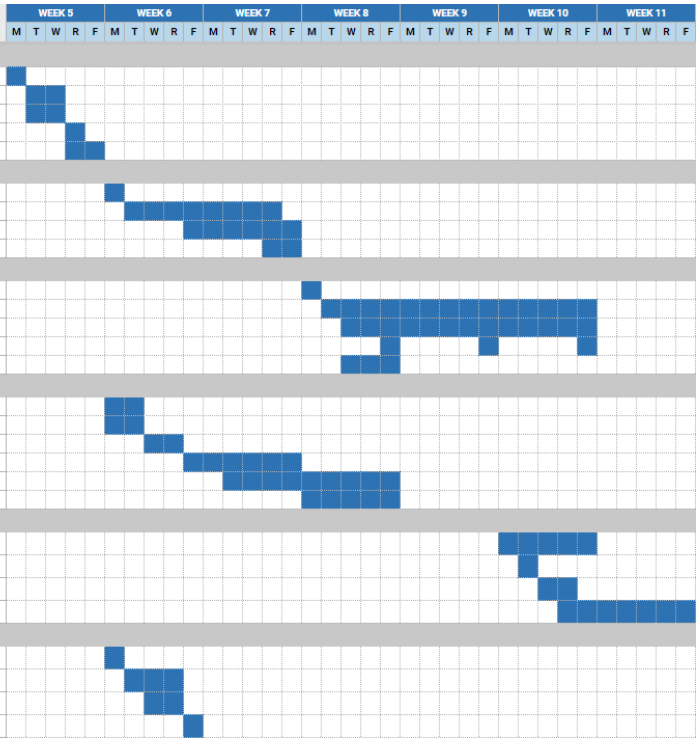
The following steps outlines the ideal method of operation for our proposed wearable:

1. **Sensor Data Collection**: Using sensors (accelerometer and gyroscope) on the Arduino Nano to collect data in real-time about hand and arm movements.
2. **Gesture recognition**: Using an algorithm trained using machine learning techniques to identify specific movement patterns and classify them as certain gestures, based on previously collected data.
3. **Control mapping**: When a gesture is recognised, the Arduino Nano is able to map gestures to specific commands to control a connected smart home device. These gestures are predetermined according to a defined mapping table.
4. **Communication of control**: Control commands are sent from the Arduino Nano to the smart home device via BLE.

## Milestones and Gantt chart

1. Define gesture set and control mapping
2. Collect sensor data for different gestures
3. Train a machine-learning model to identify gestures
4. Control third-party smart-home devices with Bluetooth
5. Integrating gesture detection and Bluetooth commands
6. Create the housing for the components

| WBS NUMBER | TASK TITLE | TASK OWNER | START DATE | DUE DATE | DURATION |
|---|---|---|---|---|---|
| 1 | **Define gesture set and control mapping** | | | | |
| 1.1 | Research common gestures | Jerome, Chen | Jun 24 | Jun 24 | 1 |
| 1.2 | List common smart-home devices | | Jun 25 | Jun 26 | 2 |
| 1.3 | Research common smart-home controls | | Jun 25 | Jun 26 | 2 |
| 1.4 | Select gestures for implementation | | Jun 27 | Jun 27 | 1 |
| 1.5 | Map gestures to controls | | Jun 27 | Jun 28 | 2 |
| 2 | **Collect sensor data for different gestures** | | | | |
| 2.1 | Create data collection plan | Chen, Terry | Jul 1 | Jul 1 | 1 |
| 2.2 | Develop data collection program | | Jul 2 | Jul 11 | 10 |
| 2.3 | Conduct data collection | | Jul 5 | Jul 12 | 8 |
| 2.4 | Verify overall data accuracy | | Jul 11 | Jul 12 | 2 |
| 3 | **Train a machine-learning model to identify gestures** | | | | |
| 3.1 | Pre-process sensor data | All | Jul 15 | Jul 15 | 1 |
| 3.2 | Select and implement ML algorithm | | Jul 16 | Aug 2 | 17 |
| 3.3 | Train model | | Jul 17 | Aug 2 | 16 |
| 3.4 | Test model accuracy | | Jul 19 | Aug 2 | 14 |
| 3.5 | Integrate model into wearable | | Jul 17 | Jul 19 | 3 |
| 4 | **Control third-party smart-home devices with Bluetooth** | | | | |
| 4.1 | Research BLE protocols of devices | Jerome, Haopeng | Jul 1 | Jul 2 | 2 |
| 4.2 | Research BLE communication libraries | | Jul 1 | Jul 2 | 2 |
| 4.3 | Implement BLE connection | | Jul 3 | Jul 4 | 2 |
| 4.4 | Implement BLE control command | | Jul 5 | Jul 12 | 8 |
| 4.5 | Test BLE control accuracy and reliability | | Jul 9 | Jul 19 | 11 |
| 4.6 | Implement BLE control through wearable | | Jul 15 | Jul 19 | 5 |
| 5 | **Integrating gesture detection and Bluetooth commands** | | | | |
| 5.1 | Optimise wearable code efficiency | All | Jul 29 | Aug 2 | 5 |
| 5.2 | Develop gesture-detection and BLE code | | Jul 30 | Jul 30 | 1 |
| 5.3 | Test integrated code programatically | | Jul 31 | Aug 1 | 2 |
| 5.4 | Test integrated code on wearable | | Aug 1 | Aug 9 | 9 |
| 6 | **Create the housing for the components** | | | | |
| 6.1 | Design enclosure | Haopeng, Terry | Jul 1 | Jul 1 | 1 |
| 6.2 | Create enclosure | | Jul 2 | Jul 4 | 3 |
| 6.3 | Assemble prototype | | Jul 3 | Jul 4 | 2 |
| 6.4 | Test wearability | | Jul 5 | Jul 5 | 1 |




# Risk assessment

| Risk | Likelihood | Impact |
|---|---|---|
| Team member leaves | Rare | Major |
| Team member sick | Moderate | Moderate |
| Sensor data inaccurate | Moderate | Major |
| Gesture recognition model inaccurate | Unlikely | Major |
| BLE communication unreliable | Unlikely | Moderate |
| Third-party device integration issues | Moderate | Major |
| High power consumption | Likely | Moderate |
| Hardware malfunction | Unlikely | Major |
| Budget issues | Unlikely | Major |
| Project scope creep | Moderate | Moderate |
| Intellectual property issues | Rare | Major |

# References

[1] Xu, C., Pathak, P.H. and Mohapatra, P., 2015, February. Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch. In Proceedings of the 16th international workshop on mobile computing systems and applications (pp. 9-14).
[2] Gangrade, J. and Bharti, J., 2023. Vision-based hand gesture recognition for Indian sign language using convolution neural network. IETE Journal of Research, 69(2), pp.723-732.
[3] Muneeb, M., Rustam, H. and Jalal, A., 2023, February. Automate appliances via gestures recognition for elderly living assistance. In 2023 4th International Conference on Advancements in Computational Sciences (ICACS) (pp. 1-6). IEEE.
[4] https://www.control4.com/os3/
[5] https://docwiki.embarcadero.com/IoT/en/Myo_Armband