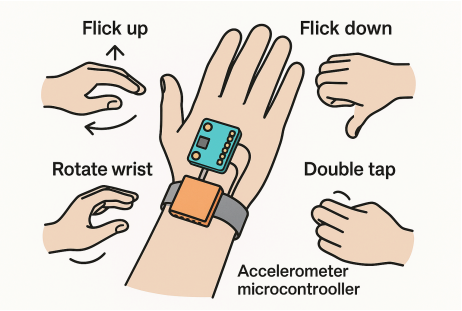
Project03 Hand gesture recognizer

Overview:
It is now time to apply what we have learned to a real project. We will create a device that measures linear acceleration and rotation with a 6-axis IMU. With this data you will identify 5 motions of your choosing. This will result in 5 categories one of which is a starting position. I will provide all the technical support and you are responsible for collecting the data and training your model.

The data:

The data from the microcontroller will be similar to this

Timestamp	AccelX	AccelY	AccelZ	GyroX	GyroY	GyroZ	TempC
2025-10-11T15:25:00	0.009272	-0.986248	0.270352	1.400000	4.830000	0.350000	32.15
2025-10-11T15:25:00	0.004392	-0.989664	0.264984	0.280000	4.760000	-0.140000	32.14
2025-10-11T15:25:00	0.002440	-0.981856	0.260592	0.840000	6.440000	0.000000	32.19
2025-10-11T15:25:00	0.004392	-0.976976	0.261568	0.490000	6.580000	-0.070000	32.11
2025-10-11T15:25:00	-0.000488	-0.991616	0.266448	0.700000	5.110000	-0.140000	32.18
2025-10-11T15:25:00	0.002928	-0.978440	0.264984	0.280000	6.580000	-0.070000	32.11
2025-10-11T15:25:00	0.003416	-0.985272	0.267912	-0.210000	5.600000	0.210000	32.14
2025-10-11T15:25:00	0.002928	-0.983808	0.266448	0.490000	5.740000	0.070000	32.13
2025-10-11T15:25:00	0.003904	-0.984296	0.264984	0.490000	5.530000	0.420000	32.14
2025-10-11T15:25:00	0.002928	-0.988200	0.263520	0.630000	5.740000	0.000000	32.16
2025-10-11T15:25:00	0.002928	-0.986736	0.260592	0.490000	5.950000	0.350000	32.14
2025-10-11T15:25:00	0.005368	-0.978928	0.266936	0.420000	6.020000	0.280000	32.17
2025-10-11T15:25:00	0.000000	-0.991128	0.253272	0.910000	6.160000	-0.280000	32.14
2025-10-11T15:25:01	0.001952	-0.982832	0.259616	1.400000	6.860000	-0.140000	32.18
2025-10-11T15:25:01	0.001464	-0.984296	0.259128	1.190000	4.480000	0.420000	32.14
2025-10-11T15:25:01	0.002440	-0.985272	0.267912	1.260000	4.690000	-0.280000	32.16
2025-10-11T15:25:01	0.003904	-0.989664	0.262544	1.260000	4.970000	0.140000	32.14
2025-10-11T15:25:01	0.003416	-0.980392	0.262056	1.400000	5.110000	-0.070000	32.14
2025-10-11T15:25:01	0.007320	-0.991616	0.269376	0.490000	5.110000	0.000000	32.16
2025-10-11T15:25:01	-0.003904	-0.983320	0.264496	1.680000	5.040000	-0.420000	32.14
2025-10-11T15:25:01	0.003904	-0.986248	0.269376	1.330000	5.530000	-0.350000	32.11



Determining the appropriate sampling rate for motion will be part of your project.

Stage 1 (exploratory stage):

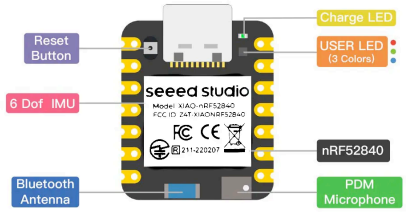
The first part of the project will have you make continuous gestures that you are interested in. Using the raw 6-axis IMU data (accelerometer and gyroscope), you will apply PCA to reduce the dimensionality of your motion data and visualize the underlying structure of your movements. Then, you will use DBSCAN, to automatically group similar motion patterns without using predefined labels. Your goal is to analyze the resulting clusters in PCA space and interpret them as five distinct gesture categories, each representing a characteristic pattern of motion discovered from your continuous data.

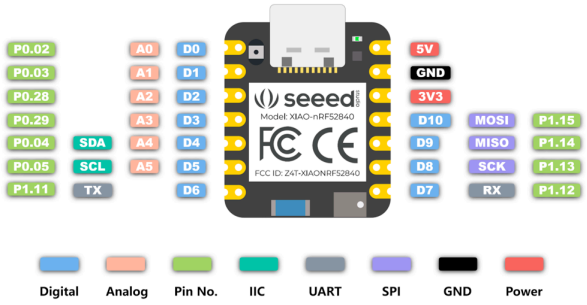
Stage 2:

With your five gesture categories identified in PCA space, you must now train a supervised classifier to recognize each gesture in real time. It can be any classification model you choose, such as a SVM or Random Forest (with boosting if needed). Your task is to extract representative samples from each of the five gesture clusters discovered in Stage 1, label them accordingly, and split the dataset into training and testing sets. The classifier should learn to map new IMU input data to one of the five categories. Once trained, your model should take live sensor readings or replayed data streams, apply the same preprocessing and PCA transformation, and output the predicted gesture label. The goal of this stage is to move from unsupervised discovery to real-time recognition, demonstrating that your system can accurately classify gestures as they occur.

The device:

For this project, we will be using the Seeed Studio XIAO BLE Sense, a compact yet powerful microcontroller that integrates all the features we need in a single board. It includes a 6-axis inertial measurement unit (IMU) capable of capturing both acceleration and rotational motion, a built-in microphone for audio sensing, and Bluetooth Low Energy (BLE) for wireless data transmission. Unlike simple sensor modules, this device contains a full-fledged nRF52840 microcontroller with onboard processing power, enabling you to both collect and analyze motion data directly on the board or stream it to your computer. Its small form factor makes it ideal for mounting on the back of the hand, where it can record continuous motion data for gesture recognition tasks in real time.





Deliverables:

In the process of this project you will generate source code, trained models, and figures. Assemble all work output into a single, well-organized repository (or shared folder) that includes: (1) preprocessing/PCA/DBSCAN notebooks or scripts, (2) classifier training and inference code, (3) all saved artifacts (scaler/PCA weights, model parameters), and (4) graphics —PCA plots, cluster visualizations, confusion matrices, and any latency/throughput charts. You will schedule a 30-minute meeting with me to demonstrate your finished system, and I will request these artifacts for review during our discussion.

Evaluation:
Your grade for this project will be based on both technical achievement and understanding. You will be evaluated on the quality and organization of your data collection, the correctness and creativity of your use of PCA and DBSCAN for unsupervised gesture discovery, and the performance and clarity of your trained classifier in recognizing gestures. Additional credit will be given for strong documentation, thoughtful interpretation of results, and a smooth end-to-end demonstration. In short, we are looking for both a working system and evidence that you understand how and why it works.

Commentary:

This is a substantial project—doable, and dare I say, fun. We'll hit a few snags along the way, and we'll solve them together. I'm assembling a mockup for you to review by next Tuesday (chips are ordered), and I'll handle the software stack: firmware for the Seeed Studio XIAO BLE Sense and a Python data reader. Your role is to design and perform the gestures, generate the data, and train a model that runs on your device.