



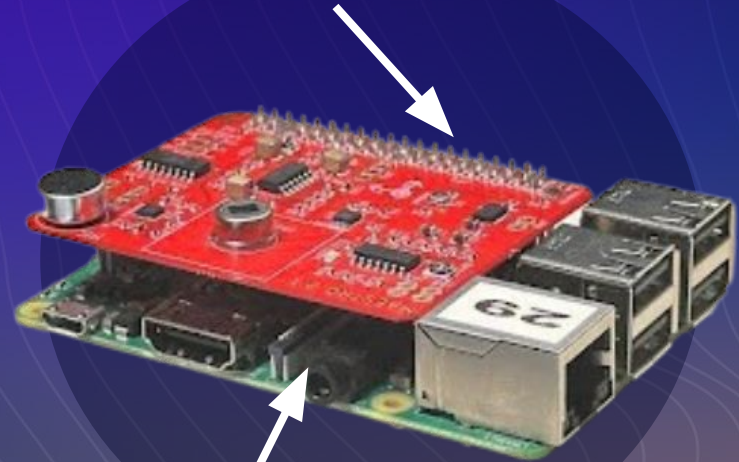
# SmartSpace LLMs

| Vedant Talati | Sathvik Samant |  
| Jose Rubio | Jonathan Duran |  
| Advisor: Dr. Ortiz |

# The Purpose

- Creating a **Large Language Model** (LLM)-powered “SmartSpace”
- LLM integration with the **physical** space
- Creating **robust** data collection infrastructure

Maestro (Custom Multi-Modal Sensor)



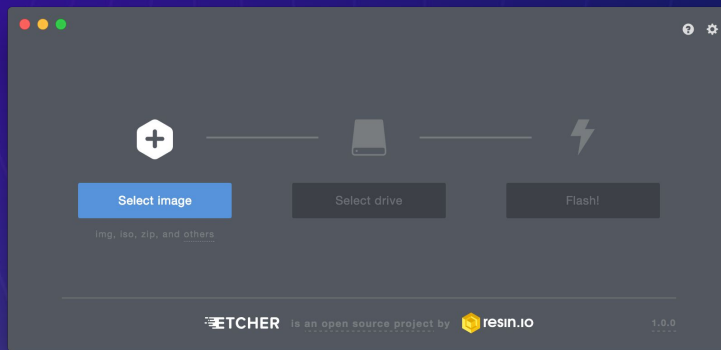
Raspberry Pi

# System Architecture



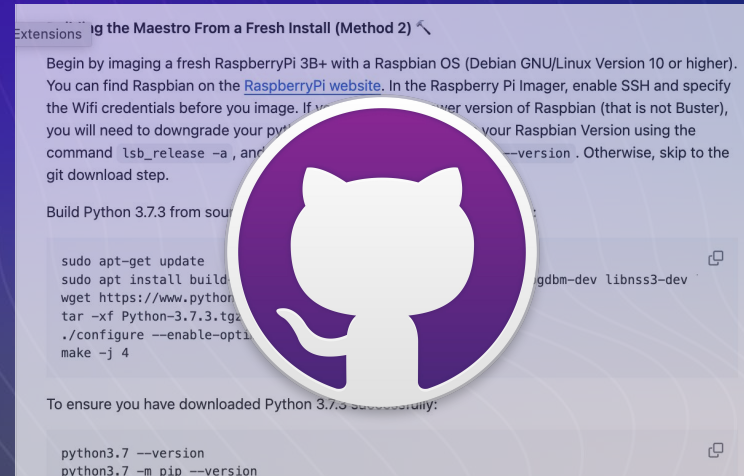
# SETUP 1

- Transferred all files from functioning Maestros to non-functioning Maestros by imaging
- Image download takes around 15 minutes
- Documented in Github



# SETUP 2

- Building the Maestro from a Fresh Install
- No physical microSD reader
- Greater customization
- Process was documented on Github



```
Extensions Building the Maestro From a Fresh Install (Method 2) 🔍  
  
Begin by imaging a fresh RaspberryPi 3B+ with a Raspbian OS (Debian GNU/Linux Version 10 or higher). You can find Raspbian on the RaspberryPi website. In the Raspberry Pi Imager, enable SSH and specify the Wifi credentials before you image. If you need to use a lower version of Raspbian (that is not Buster), you will need to downgrade your python version using the command lsb_release -a, and then use python3 --version. Otherwise, skip to the git download step.  
  
Build Python 3.7.3 from source  
  
sudo apt-get update  
sudo apt install build-essential python3-dev libssl-dev libffi-dev libncurses5-dev libncursesw5-dev libreadline-dev libsqlite3-dev libbz2-dev liblzma-dev libzstd-dev libfreetype6 libfreetype-dev libjpeg-dev libpng-dev libtiff-dev libtiff5 libtiff5-dev liblzma-dev libzstd-dev  
wget https://www.python.org/ftp/python/3.7.3/Python-3.7.3.tar.xz  
tar -xJf Python-3.7.3.tar.xz  
cd Python-3.7.3  
./configure --enable-optimizations  
make -j 4  
  
To ensure you have downloaded Python 3.7.3 successfully:  
  
python3.7 --version  
python3.7 -m pip --version
```

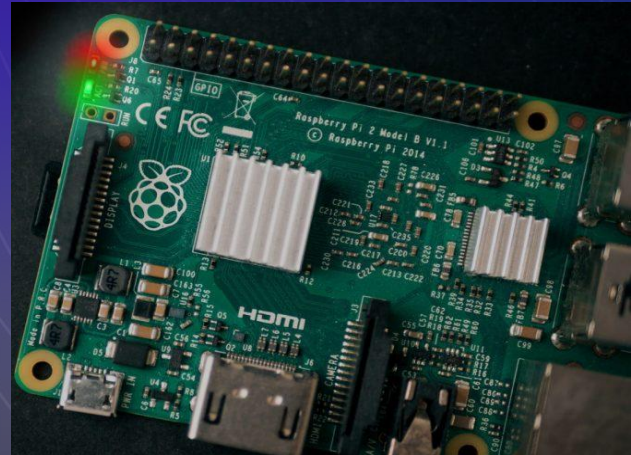
# Features: Start & Termination Control

- **Previously:** All or None :(
- **Now:** Maestros to start and terminate on demand → Significant efficiency gains + versatility
- Frontend Implementation through API Endpoint

```
for pi_num in rpis:  
    i = rpis[str(pi_num)]  
    status = end_maestro1(i)  
    print(f"Maestro {i}: {status}")
```

# Feature: Visual LED lights

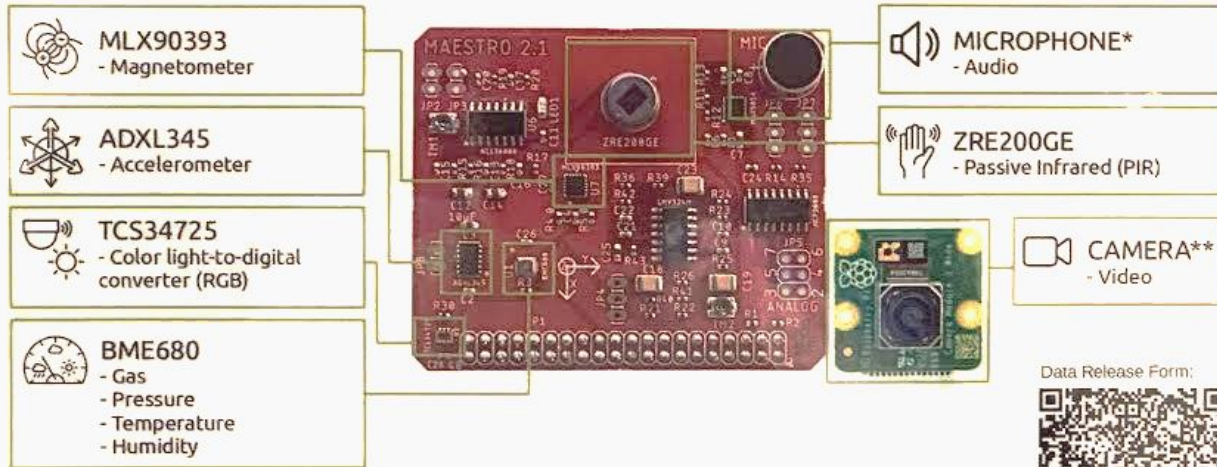
- Visualization of streaming data
  - Can't turn on/off LED lights
  - Disrupts internal hardware
  - Documented in Github
  - Graphical appeal



# Sensor Testing and Calibration

**!!! NOTICE !!!**

**CAMERA AND SENSOR DATA COLLECTION IN PROGRESS**



\*audio cannot be reconstructed to recover original signals  
\*\*camera data is used only for model training

Any questions?

[hedaya.walter@rutgers.edu](mailto:hedaya.walter@rutgers.edu)

[jorge.ortiz@rutgers.edu](mailto:jorge.ortiz@rutgers.edu)

Data Release Form:





# Loguru Logging

- Loguru library for logging script output
- Organized, timestamped way to log and debug
  - `logger.error('error message')`
- Function to download and clear old logs when they aren't necessary anymore
  - `.truncate(0)`



```
2022-09-30 14:14:37.636 | DEBUG | __main__:<module>:5 - A debug message.
2022-09-30 14:14:37.636 | INFO | __main__:<module>:6 - An info message.
2022-09-30 14:14:37.637 | SUCCESS | __main__:<module>:7 - A success message.
2022-09-30 14:14:37.637 | WARNING | __main__:<module>:8 - A warning message.
2022-09-30 14:14:37.637 | ERROR | __main__:<module>:9 - An error message.
2022-09-30 14:14:37.637 | CRITICAL | __main__:<module>:10 - A critical message.
~/dev/betterstack/community/demo/loguru-demo via 🐘 v3.10.6
```

# Frontend development

- 8 different easy to use features
- Chart feature used to display sensor data
- Coordinate grid to show Maestro activity and location

## Shutdown Selected Pis

Enter PI numbers to shut down (comma-separated):

## Shutdown All Pis

## Status of Maestros

## Start/Stop experiments

## Schedule appointment for later

Appointment time:

## Data Visualization:

## Data Download:

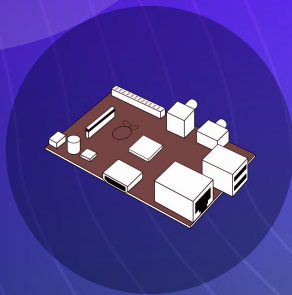
## Experiment Logs:

# ML & LLM Integration

- ML tagging and identification:
  - **YOLOv8**: real time object detection model
  - Motion detection w/ PIR
- LLM integration:
  - Creating a smart-space infrastructure enabled by conversational LLMs
    - OpenAI API

# Future Work

- Making the testbed completely remote
- Improved sensor measurements
- Better sampling rate/frequency
- **WARM (Weighted Average Reward Models):**
  - Mitigates reward hacking
  - Human aligned response



**Thank you**



## **Acknowledgments:**

---

We would like to thank **Professor Jorge Ortiz** for being our mentor, as well as **Jenny Shane** and **Ivan Seskar** for suggesting advice for a few of our challenges. We would also like to thank our undergraduate students, **Matthew Grimalovsky, Emma Lin, and Sanchita Agrawal**, for guiding us through the entire research project process.