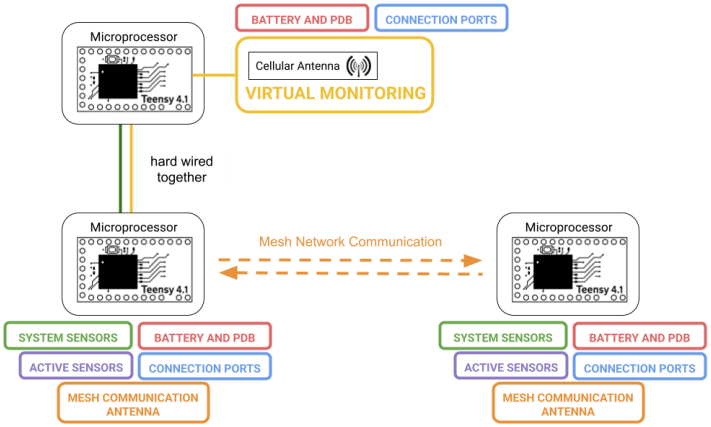
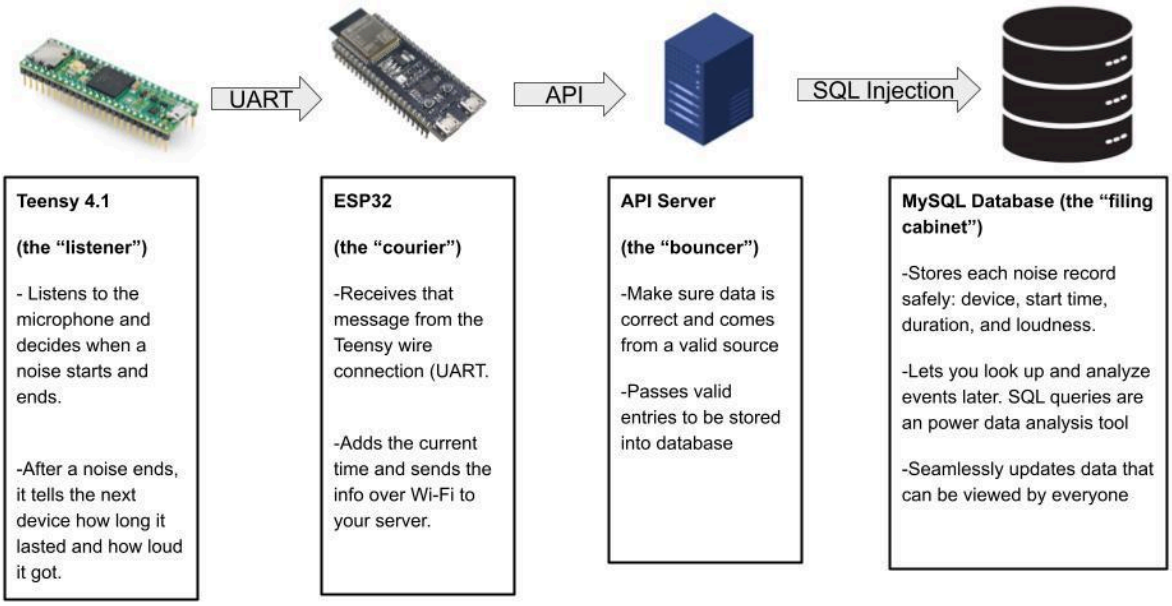
Shaman 1 : Test 2.0 :

Introduction

In the P0 stage of development an essential goal is to develop a field testing apparatus. This device is not fully representative of the final P2 version of the Shaman 1, but simulates much of its technology while also having additional features that are used solely for testing and the analysis of the sensors. This is the proposed schematic:



“noise_detector” is a project that is a simple representation of the field testing apparatus, in other words the simulation of the simulation for the Shaman. The goal of this project was to familiarize myself with the technology that will be used for the testing apparatus. “noise_detector logs any loud noise into a database, following this framework:



Prerequisites

To run “noise_detector”, ensure that you have:

- Micropython is successfully flashed onto the Teensy 4.1, follow Teensy 4.1 Firmware Setup Guide Instructions for more details
- record_on_button successfully runs on device, follow GUIDE: Shaman 1 : Tests 1.0 : record_on_button for more details
- jumper cables
- ESP32
- Arduino IDE installed
- mysql workbench installed (not necessary but easiest for development long term)
- USB C cable

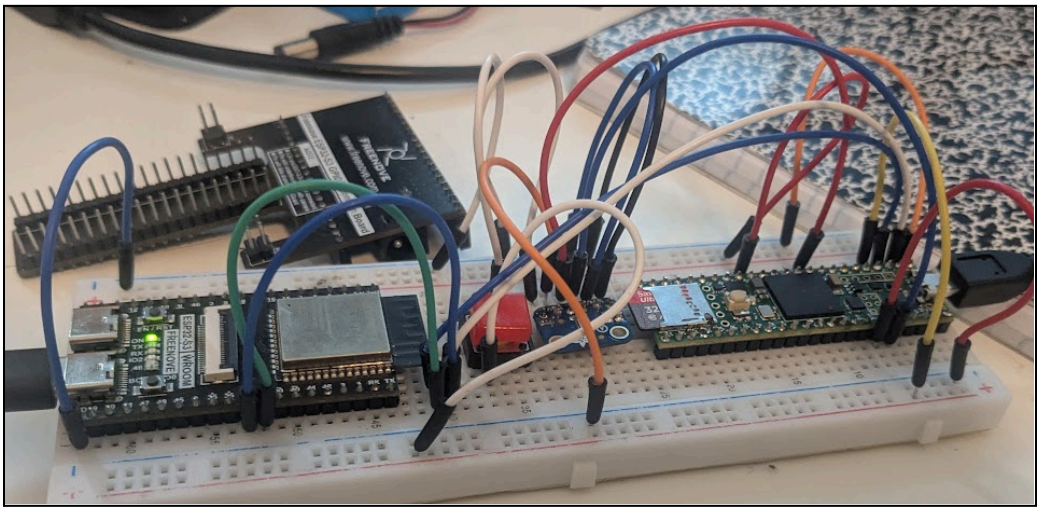
Wiring

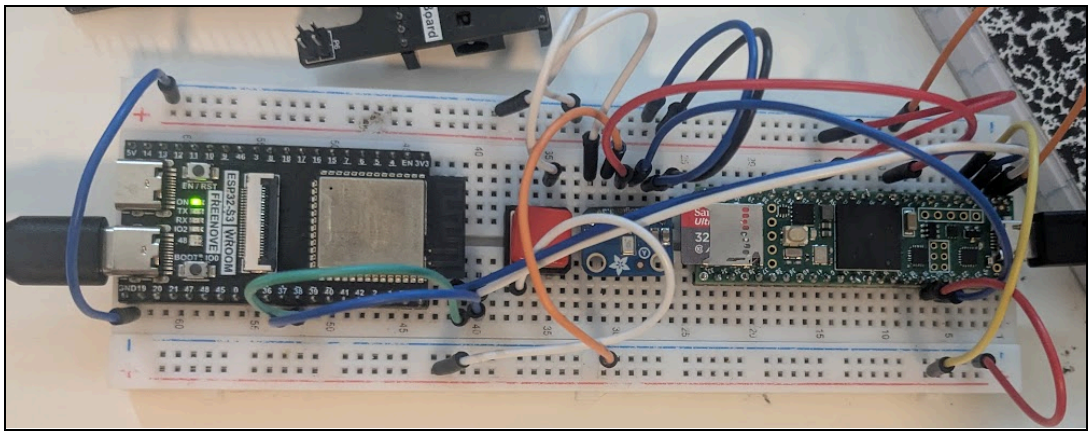
Follow this table for wiring (this assumes button is wired from Tests 1.0)

Teensy	ESP32
1	38
0	37

Note: The ESP32 and Arduino should be plugged in with their respective cables.

Photo of my setup:





Running test_connections

“noise_detector” is a complex program. “test_connections” is designed to test that your environment is set up properly. Ensure this program runs as follows:

When the button on Teensy 4.1 is pressed, it toggles the LED on ESP32-S3 WROOM and the ESP32 POSTs the event (LED_ON / LED_OFF) to a local FastAPI server that prints it. The goal of these programs is to establish that each component can successfully communicate with each other like this:

Teensy 4.1 -> ESP32 -> FastAPI Server

1. Install FastAPI + Uvicorm
In VS code, run these commands in your environment dedicated to running the api server:

python -m venv .venv
.venv\Scripts\Activate.ps1
pip install fastapi uvicorn pymysql

2. Run Server
Add “api.py” from the folder “test_connections” into the VS code environment:
Input this command into terminal:
uvicorn api:app --host 0.0.0.0 --port 8000 --reload

3. Flash Code to Teensy
Add “teensy_to_esp” from “test connections” into the VS code environment and upload code

4. Note IP Lan Address + note Wifi Information

In windows power shell run “ipconfig”:, and take note of the number listed as “IPv4 Address”. Also make sure you know your wifi name and password

5. Arduino + update WiFi information
Upload “esp_to_api.ino” into Arduino and at the top of the code edit wifi information. Upload code to ESP-32. Additionally enable serial monitor in “Tools” on the top left side

6. Analyze Logging
Pressing the button should create that logs for each program that look like this:

api.py:

```
INFO:     Will watch for changes in these directories: ['C:\\Users\\kylek\\OneDrive\\Desktop\\Shaman_1\\Tests_2.0']
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [19820] using StatReload
INFO:     Started server process [15900]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
LED_ON
INFO:     192.168.0.152:49477 - "POST /events HTTP/1.1" 204 No Content
```

teensy_to_esp.py:

```
PS C:\Users\kylek\OneDrive\Desktop\Shaman_1\Tests_2.0> mpremote connect COM9 run teensy_to_esp.py
Teensy ready (UART1 D1/D0, button D2)
Sent: b'LED_ON'
```

esp_to_api.py:

```
WiFi: connecting.....
WiFi: connected, IP=192.168.0.152
ESP32 ready.
API target: 192.168.0.179:8000/events
POST 204 () :: {"device_id":"esp32-s3-01","state":"LED_ON","esp_epoch":1760902805}
Received Teensy message 'LED_ON' — message to API successful
```

Running noise_detector
1. Extract and open “noise_detector”
2. Create database in mysql, make sure mySQL was installed in the same folder as [api.py](api.py)
3. Update the files from “test_connections” with the new code (file names are the same)
4. Use query to ensure device is logging properly like this:

id	device_id	event_start_utc	duration_ms	peak_dbfs	created_at_utc
2	esp32-01	2025-10-20 03:46:08.449000	5551	-12.33	2025-10-19 20:46:16.766251
1	esp32-01	2025-10-20 03:46:00.467000	1533	-16.73	2025-10-19 20:46:02.157997
NULL	NULL	NULL	NULL	NULL	NULL