GestureCap 2025 – Latency Measurement on A15

These steps assume no changes or updates are made to the A15 code/system after I leave. All instructions are specific to the scripts inside the gesturecap2025 project folder.

1. Software Setup (A15 side)

Boost GPU clocks before starting experiments:

```
sudo nvidia-smi -lgc=3000,3000 && sudo nvidia-smi -lmc 8000,8000
```

Check GPU clocks

Run this command to confirm GPU clocks:

```
nvidia-smi -q -d CLOCK
```

Reset GPU clocks

Reset GPU clocks after experiment is done:

```
sudo nvidia-smi --reset-memory-clocks && sudo nvidia-smi --reset-gpu-clocks
```

Run scripts with GPU

Use prime-run for all main Python scripts to utilise the GPU.

Example:

```
prime-run python latency_mp.py
```

Scripts Involved

record_flircam.py – camera positioning

calibration.py – set reference line and calibration distance

latency_mp.py – latency testing

latency_mp_2.py – latency testing + saving frames before taps

log_serial.py – serial logging from Teensy

join_tables.py – combine latency logs into a CSV

2. Audio Side Setup

Connect AUX cable from A15 to the speaker.

Place the microphone sensor close to the speaker membrane.

3. Teensy Setup – Microphone + Speaker Mode

Connect Teensy pins: GND → Ground, 3V → VCC, A0 → Pin 23 (or whichever analog pin you configure).

Confirm selected pin matches the Teensy code.

Upload the raw_data_plot code to the teensy to check the values in silent conditions and set the threshold in "latency.ino" comfortably above that level.

Test the setup by tapping the microphone and observing the readings.

4. Teensy Setup – Raw AUX Analog Mode

Use an open-ended AUX wire: Ground terminal → GND on Teensy, Positive terminal → Analog pin (currently Pin 23).

Repeat the steps for the Teensy code and raw_data_plot upload, set threshold in "latency.ino" comfortably above silent readings.

Threshold Limits in latency.ino

In the latency code, set the lower limit and upper limit thresholds:

- Lower limit: theoretical minimum latency minus a few ms for margin.

- Upper limit: theoretical maximum latency plus a few ms.

This prevents spurious readings outside expected bounds.

Current Threshold Values

For raw analog AUX values: threshold is set to 30.

For microphone + speaker setup: threshold is set to 80.

These values were chosen by observing silent readings with raw_data_plot and then setting thresholds comfortably above those baselines.If you notice false positives or missed taps, consider adjusting these thresholds.

5. Physical Setup – Aluminum Foil & Hand Tap Sensor

Paste aluminum foil at the edge of a flat surface.

Connect any Teensy GND pin to the foil using an alligator clip/wire.

Connect a wire to the buttonPin (currently set to Pin 2).

Attach this wire to the side of your left pinky finger, minimizing obstruction of the back of your hand.

Connect the Teensy to the A15 via USB.

6. Camera Setup

Connect the FLIRcam to A15.

Run record_flircam.py, adjust the camera so foil edge is parallel to the reference line, then press q to close.

```
prime-run python record_flircam.py
```

7. Calibration

Run calibration.py.

Click twice along the foil edge with as much precision as possible.

Keep left hand vertical on the surface with pinky resting sideways on the foil.

The script will measure distance between Avg(y-coord(17 to 20)) and reference line for 1 second.

Ensure right hand is not visible in the frame during that time.

```
prime-run python calibration.py
```

8. Verification of Calibration

Run latency_mp.py.

Hover test: taps only trigger when hand is very close to the foil (< few mm).

Rapid taps test: tap rapidly on the surface false positives should be rare (about 1 in 7 or better).

Still hand test: Rest your hand sideways on the surface. No taps should be registered when hand is resting still.

If tests fail, repeat calibration and adjust camera exposure in flircam.py or room lighting.

```
prime-run python latency_mp.py
```

9. PureData Setup

Open PureData and load beep.pd.

Set frequency to 200.

Go to Audio Settings: select AUX output device, set delay = 3 ms.

Press button in PureData to check if sound plays from speaker, if not, try changing the output device, restarting computer, etc.

10. Logging

Run latency_mp.py again.

Open a split terminal and run log_serial.py.

When prompted, enter experiment details: .

This script uses a configuration file named 'config.json' with the following keys: - device: The device on which the experiment is conducted. - baud_rate: The baud rate for the serial connection (e.g., 9600 or 115200). - method: A description of the audio detection method used in the experiment, such as raw mic values, delta max-min, etc. - frequency: The frequency of the audio signal in Hz. - threshold: The threshold value used in the experiment. - pd_delay: The delay value in milliseconds. - output_method: The method of output such as AUX with speaker-mic, AUX terminals connected directly, or a focusrite with speaker-mic.

Re-attach wire to pinky, tap a few times to confirm that latencies are logging.

Restart both scripts and begin experiment. Tap your hand about 250 times to obtain ~200 latency samples

```
prime-run python latency_mp.py

python log_serial.py
```

11. Data Processing

Two files saved: TableA.csv, TableB.csv.

Run join_tables.py with input and output paths. This script takes care of all the false positives/negatives.

Final CSV will contain total latency and breakdown of internal latencies corresponding to each tap.

```
prime-run python join_tables.py --tablea path/to/TableA.csv
      --tableb path/to/TableB.csv --out path/to/final.csv --tol_ms 50
```

12. Optional – Pre-Tap Frame Capture

Run latency_mp_2.py if you want to save frames before tap is detected.

Wait at least 500 ms between taps (saving takes 500 ms).

```
prime-run python latency_mp_2.py
```

13. Cleanup

Disconnect the FLIR camera from A15.

Disconnect the Teensy from the USB port.

Reset GPU clocks to default settings.