Overview

This document outlines areas of testing which would be valuable for the Merlot project. The communication layer would be the highest priority as we need to ensure it is stable for the success of the product.

SOC <-> MCU communication layer

This communication layer is critical to the success of our product and as such we want to ensure we have a strong suite of tests to ensure the correct functionality. Some sample test conditions would be:

- Ensure correct operation during normal conditions
- Ensure connection is rebuilt successfully after an MCU reset
- Ensure connection is rebuilt successfully after individual features restart (command router, bluebox, stp)
- Testing various operations of the communication channel (different payload sizes, throughput, etc)
- Testing correct handling of batching/waking during SOC doze
- Jes Sorensen would be a good person to contact for additional details of further testing to be performed.

MCU Flashing

Currently we don't not have exhaustive tests to validate flashing of our MCU. Some things we would like to cover are:

- Normal flashing scenario with success
- Recovering from a failed flash
- Ensuring correct zephyr image and assets are flashed
- Ensuring correct boot of the MCU after flashing
- Daniel Orogchock would be a good person to sync with in regards to this testing.

Sensor driver operation

We currently have a number of unit tests for the business logic around sensor usage, but nothing to ensure the drivers are behaving correctly. A good addition to the test framework would be ensuring the drivers we have for our sensors are operating correctly.

Audio

For existing audio tests, it only really tests if audio samples get sent over i2s to the MCU. Nothing is validating that the proper sound came out of the speaker. If we could capture the echo reference samples that we get back from the smart amp, and do some sort of processing to validate that we played the actual audio, it'd make the test much more robust.