Post Serp Launch Atttempt Testing Notes

This document contains testing notes from the issues identified during the launch attempt on 14/04/2024 at Serpentine.

**Issues Identified:**

* Connecting drogue or main ematch causes ASM330 to malfunction (red dot on UI) and the buzzer works intermittently.
* After indeterministic periods of time, the flight computer becomes unresponsive over radio.
* Battery retention did not protect battery sufficiently. Requires redesign.
* Battery had insufficient capacity.

**Recreating behaviour at home:**

Strelka was wired up to the power profiler and given 4.0V input power. The arming switch wires were run through wires that are tied together. Wire length is around 15-20cm. E-match is connected to drogue port.

Results:

Buzzer issue:

Multimeter shows 5V and 3.3V on respective rails. No information collected on timeseries profile of power lines.

When the ematch is not connected to either port, the buzzer worked correctly. When the ematch is connected, the buzzer did not work correctly. It either did not work at all or was stuttering. Rebooting Strelka did not cause the buzzer to start working.

Removing the ematch and cycling power did not allow the buzzer to start working again.

Buzzer issue appears to occur even when the ematches are not connected.

The buzzer operation also does not appear to be dependent on the connection of the gps antenna.

Operational issues:

Even though an ematch is connected to the drogue port, the ground station still shows open circuit.

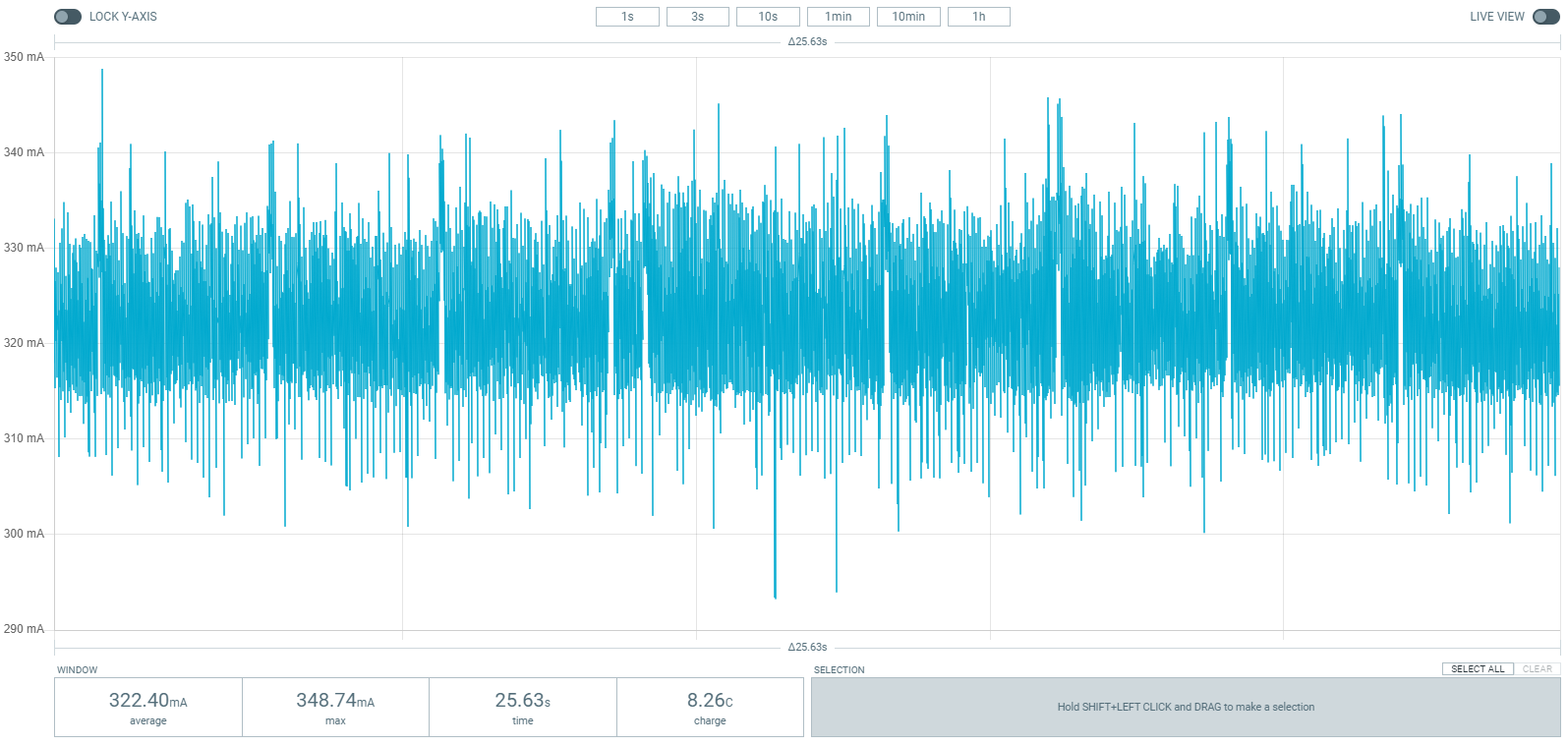
When an ematch is connected. The ASM330 sensor state moves to red. As soon as the ematch is connected, the ASM330 malfunctions.

Note: Can confirm that the streaming at 1Hz is actually occurring at 1Hz.

After ~4:33 mins operation (in this case), the radio appears to stop streaming. From this point, it is completely unresponsive. It also appeared to stop streaming half way through a transmit which shown on the power profiler output (current draw) as a cut in current half way through a transmit spike.

It appears that the RFM95 stops transmitting. All tasks appear to be working correctly and the LoRa\_transmit(&LoRa\_Handle, send\_pkt, len + 15, 1000); function is returning 0 (nominal). The current graph shows that no transmission occurred.

Current graph during transmission halt failure:



What is odd is that it seems to be undetectable from the software.

The register read of the RFM95W’s RegIrqFlags (0x12) register returned 0x8 which is a correct for a TxDone result.

It also does not receive packets that are sent. No breakpoint is triggered in the Rx callback in the firmware when a command is sent from the ground station.

Upon resetting the microcontroller (uploading firmware) without power cycling it, the issue appeared to fix itself meaning that I was able to communicate to it over LoRa. When resetting the microcontroller, it also toggles the reset pin of the RFM95W which may fix the issue.

In one case where the microcontroller halted the RFM95 stopped responding and the transmit function returned 1 (error).

When the radio dropped out, I ran LoRa\_init() and it caused the device to start drawing current in a way indicative of it working correctly. The only difference was that I was not seeing the data being received on the ground station. Further, Strelka did not respond to any messages sent from the ground station. Completely resetting Strelka (via reset pin), returned it to normal operation.

Running LoRa\_init() then LoRa\_startReceiving() returned the transmitter to its original state. There is something in the process the reinitialises the transmitter to work again.

It appears this is an internal issue specific to the rfm95. It may be a faulty chip or a duty cycle that’s too high.

The issue appears to be specific to streaming. When streaming is not enabled, the refresh function works for a long time (+7 mins) however, if streaming is enabled it only runs for a couple of minutes after that then stalls.

Significantly decreasing the stream frequency appears to increase the longevity of the data streaming. The transmitter was able to run for 24 minutes without issue at 1 message every 10 seconds.

**Minimum supply voltage:**

Microcontroller will not boot below 2.5V at the input.

**Requirements for semaphores:**

* Access the deployment pins. Implemented
  + If a chute deployment command is executed, osDelay used and in the meantime, test\_continuity() is called, it will change the pin states used by the deploy chute command.
* SPI bus access.
  + More than one sensor should not have access to the same SPI bus during a transmission sequence.
* Memory writes.
  + More than one thread should not have access to flash memory concurrently.