ECE4806 Testing Documentation
F25-13

Purpose
This document outlines and records the testing process for team F25-13.

Testing Process and Subsections
Our testable systems include:

Hardware
- Voltage Verification Testing: assuring correct DC voltages at certain points in the circuit
- Waveform Testing w/ ADC and Pre-Amp:
- Switching Verification Testing:

Software
- GUI Operation Testing:
- Pin Clocking Observation Testing:
- Data Store/Receive Testing
- Ethernet Testing:

Full System Testing
Full system testing includes verification of all sent and received commands through the, as well as a full run-through of the image capture sequence. This sequence is:

1. Host computer sends capture signal to PCB through ethernet
2. Board completes image capture and stores into Flash Memory
3. Board idles and waits for next command
4. When commanded, PCB sends image to host computer through ethernet

The end goal is for all commands to be run through the GUI.

Testing First PCB Design
Our first board limited subsystem testing opportunities because of how poorly it worked, and from our lack of knowledge of flashing from the STMCube IDE. Our first main issue was voltage verification, which failed because the footprints of our opamps on the PCB were different from our actual parts. This oversight led to us not being able to finish the power regulation section of the board, meaning most of the parts that were soldered on didn't have the correct voltage. The opamps were meant to counteract the loading effect on the ends of our voltage dividers, and without them, our voltages were spread too thin. The only consistent voltage in our system was ~3.3V, which was being supplied from outside the board through the STLink USB device. With a stable 3.3V, we could power the MCU, Ethernet, flash memory, and temperature sensor sections of our board, but we wouldn't be able to test about half of our subsystems without the other voltages working. Our other tests were also cut short because of our inexperience with STMCube IDE, and the poor layout and part selection of our PCB design.

Because we lacked the training and resources at the time, our first board contained all through-hole resistors and capacitors, which increased the necessary size of the board significantly. Next, our layout didn't follow the design requirements for the ICs we were using. This led to our MCU (seemingly) browning out whenever we tried to flash it with our code. We'd get messages that printed to the console that would initially connect, then shut down the MCU after a few seconds. We didn't know if this was a hardware or software issue, but with everything wrong with the board already, and now that we had SMD soldering training, we decided to design a 2nd board. At best, our MCU could flash occasionally, but we didn't know this until after testing with our 2nd board.

We were told by an AMP lab student that the MCU issue was likely caused by poor capacitor choice and placement, which was corrected in the 2nd board. We also redesigned the power system. We reduced the number of bias voltage requirements by combining some together. For example, we were biasing both 17V and 15V, but the 15V pin had a tolerance between 13V and 20V, so we decided to just power it with 17V. The new design uses 2 buck converters and 3 linear regulators, which is much more efficient than the original voltage divider and op-amp combos before.

Testing Second PCB Design
The first thing we noticed after fully soldering the second board is a high pitched whine coming from our buck converters when connected to a power supply. We suspect the whine is coming from the inductors, specifically that they're not big enough. It's not an immediate issue, but something that needs to be corrected in future designs.

When conducting voltage verification testing, we found that the 17V output from our buck converter was actually stable at around 21V. We double checked anything that could've caused this mistake, but we couldn't find anything. What's even stranger is that every other voltage is correct. The way the power system works on our 2nd board is:

- 28V supply is linear regulated down to 25V
- 25V is bucked down to 17V
- 17V is bucked down to 10V
- 10V is linearly regulated down to 5V and 3.3V

With a system that cascades voltages like, an incorrect voltage at one stage of the process should change every voltage in the sequence, but this wasn't the case. We read correct voltage values for 25V, 10V, 5V, and 3.3V. We're not sure what's causing this, but it's another non-critical issue. The 17V bias serves as a MOSFET power source for some of the switching pins on the CCD. We're slightly worried that some of our MOSFETs might be damaged by this voltage increase, because they're rated to a Vds of 20V, but we have yet to test if this is an issue.

11/6/2025 - MOSFET Switching Test 1
Initial MOSFET Switching testing didn't have the results we were looking for. We expected each pin of the CCD that's clocked by the MCU to be at 0V until the GPIO pin that clocked it was turned on. Instead, each pin was constant at the voltage they were to be clocked to. We think this might have happened for a few reasons:

1) Incorrect resistors at the input of the MOSFET gate allowed 10V to charge the Vgs of the EM6K7 mosfets. These MOSFET gates are rated to +/-8V, so we think we might have damaged the MOSFETs.
2) Incorrect design and/or soldering. We might be measuring our outputs at the incorrect terminal. I'd say this is unlikely, since the design is so simple.
3) We don't fundamentally understand the inner workings of the MOSFET and gate driver pair.

We have yet to do in-depth probing of each terminal of each gate driver and MOSFET, but we'll start that tomorrow (11/7/2025).

11/7/2025 MOSFET Switching Test 2

The second round of MOSFET Switching Testing gave us more of the results we were looking for. We swapped out each of the MOSFETs with brand new components because we were unsure if the reason they weren't working was because they were damaged. When applying a test PWM pulse to each of the MOSFETs, we found that only one pair was working properly, but that it was properly switching the voltage applied to it. We also found that there were some incorrect connections on the PCB. There is a mismatch between the labeled GPIO pins and what CCD pin they connect to, but it's an easy fix of simply relabeling the pins in the MCU configuration.