5 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopted test strategy and instruments. In this overarching introduction, given an overview of the testing strategy and your team's overall testing philosophy. Emphasize any unique challenges to testing for your system/design.

In the sections below, describe specific methods for testing. You may include additional types of testing, if applicable to your design. If a particular type of testing is not applicable to your project, you must justify why you are not including it.

When writing your testing planning consider a few guidelines:

- Is our testing plan unique to our project? (It should be)
- Are you testing related to all requirements? For requirements you're not testing (e.g., cost related requirements) can you justify their exclusion?
- Is your testing plan comprehensive?
- When should you be testing? (In most cases, it's early and often, not at the end of the project)

5.1 Unit Testing

What units are being tested? How? Tools?

Our device is used for testing photosensitive devices. It will utilize different colored LEDs to test the capabilities of optoelectronic devices under different gas conditions. We will be using a Keithley 2400 to collect data from our device to be analyzed by a computer.

5.2 Interface Testing

What are the interfaces in your design? Discuss how the composition of two or more units (interfaces) are being tested. Tools?

Our design combines several hardware and software interfaces. It will have 2 to 3 separate hardware interfaces which each have their own microcontroller and communicate with each other via I2C (or another method if our plans change). The data collected by our device will be output to a computer to be analyzed and displayed on a digital interface that we design. Additionally we will be using an SMU to make measurements that will need to be connected to the host computer through a wired "GPIB" interface.

5.3 Integration Testing

What are the critical integration paths in your design? Justification for criticality may come from your requirements. How will they be tested? Tools?

As previously mentioned there are several subsystems that are being developed concurrently. Integrating these systems is one of the key challenges to making our design work. The most critical of these system integrations will be getting the Software user interface to communicate with the PCB board that handles LED flashing & servo movement. Secondly will be integrating the Software

interface with the Keithley SMU. If the software can interface with these other two components we can achieve nearly all core functionality our client needs. I.E. testing solar cell electrical response. Other features such as integrating the control of gas flow are core requirements for the final product but are not needed to begin systems level testing and making real measurements. Integration testing will be done by testing basic functionality. For example, on the software side we can develop basic commands such as "Move servo to Red LED" , "Make SMU voltage measurement" , "Make LED pulse at x Hz". When running these commands we should obviously see some sort of corresponding response from the hardware.

5.4 System Testing

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

In order to do system testing, we have to review each individual component, and compare the results to expected values, this would be done via the unit tests. For testing the interface, we would feed data with known results into the interface, and see if the outcome matches the expected results. If every component functions correctly we know that the entire system is functioning as expected.

5.5 Regression Testing

How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure they do not break? Is it driven by requirements? Tools?

We will implement a series of interface/unit tests to be run using a 'testing' DUT that will remain consistent throughout the testing process. Whenever we add a new component, if the tests fail, we'll know that the component we added broke old functionality and therefore needs to be changed.

5.6 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?
There are several functional design requirements that we could test for the device that we have built. First, We can test the LED and servo on the PCB, which will be placed above and under the DUI. Before placing the PCB into the device we are building, we could test the device by selecting the light frequency that we want from the computer, seeing how many angles the servo rotates, and checking if the LED we have chosen is blinking. Since there will be two servos and 2 LEDs, we need to ensure that both will rotate to the same angles and that both LEDs will blink simultaneously.Then, we can test the mass flow controller. The mass flow controller will have an input of plus and minus 15V, which will power up the gas device. Then, there will have a zero to 5V regulator, which adjusts the flow rate of the gas. We could test the device by looking at the LED on the PCB board. In designing the PCB for the device, we placed some LED on the PCB so that we could know that all of the circuits on the PCB were working. If a circuit is not working, the LED will not work.

5.7 Results

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.

We cannot do any testing on the device as no physical device has been built. However, from the circuit we have designed, we can know if the circuit will work or not by doing the tests we mentioned above. The diagram below shows an LED connected to the circuit, which will light up if the circuit is working.

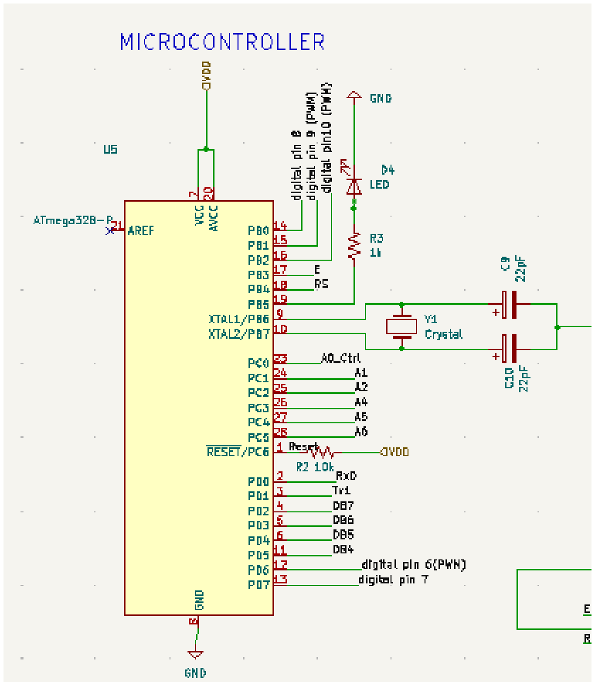

Figure 1: Circuit for MCU for Mass Flow Controller.

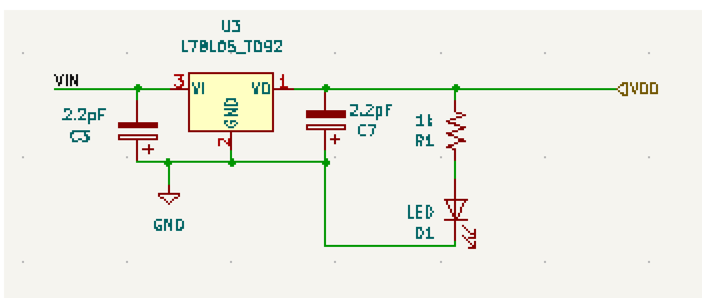

Figure 2: Circuit for 5V regulator for Mass Flow Controller.