

# Physics 335 project status report:

## Design and construction of a system to test ATLAS TileCal electronics

**Student:** Lacey Rainbolt, [jlrainbolt@uchicago.edu](mailto:jlrainbolt@uchicago.edu)

**Advisor:** Mark Oreglia, [m-oreglia@uchicago.edu](mailto:m-oreglia@uchicago.edu)

March 5, 2020

## 1 Summary

The burn-in monitoring system for the ATLAS TileCal Main Board (MB) electronics is underway. The exploratory, information-gathering phase of the project is complete. Work has begun on the project deliverable, which is a system that implements the low-level firmware of a controller card (which communicates with the MBs) to perform a series of quality control tests, interpret the data and display human-readable results to a user, and store the results for future reference.

## 2 Information gathering

I spent the first few weeks of the winter quarter educating myself about the MBs, their many components, the functionality of those components, and the tests that will to be performed during burn-in. I also learned how the burn-in process will proceed and the timescale over which it will take place. This education was necessary in order to create a complete and well-rounded picture of the work that lie ahead.

The firmware for the controller card, which controls the functionality of the MBs at the hardware level, has already been drafted by another member of the lab, Kelby Anderson. However, in order to interface this firmware to software written in Python, I need to have a working knowledge of its functionality. I was unfamiliar not only with the hardware description language, but also the entire concept of digital design. I spent a couple of weeks carefully reading through the code, as well as some manuals and resources describing the language and the basics of digital design.

Finally, I spent about one week exploring possible options for the GUI interface I will write to make it possible for an undergraduate to run the tests. I concluded that the best option is to use a native Python GUI builder, `tkinter`. I went through some tutorials on how to create an interface using Tk, which gave me a general idea of how the Python code for the tests will need to be structured.

## 3 Hardware familiarity and testing

The FPGA controller card that we will use to test the MBs during burn-in is in the process of being fabricated. However, the other hardware components of the burn-in system have been available since the beginning of the quarter, so I have been familiarizing myself with them. We will use a Linux laptop as an interface, on which I have installed software necessary to compile and test the firmware code.

The controller card will interface with the laptop via an I/O box, which has a protocol for sending and receiving data and commands via USB to and from the controller card. In lieu of the controller card, I have tested the I/O box using another FPGA controller, and in the process was able to get a bit of experience with firmware.

I have also gained some familiarity with the probe that we will use to monitor the temperature during burn-in, which connects to the laptop via USB.

## **4 Flowchart and software design**

As a project deliverable, I will produce a flowchart that describes the burn-in monitoring program. The flowchart will begin with the initialization of the MBs; cycle through each of the four tests that will be performed; describe the data that will be collected from each test; and explain the storage, analysis, and display of the data. I have created a first draft of the flowchart and discussed it with my advisor. I am in the process of implementing the changes we discussed, and will continue to iterate on the flowchart as I begin work on the software.

## **5 Remaining tasks and prognosis**

My immediate next step is to begin writing the Python modules that will perform the tests, and a script that cycles through them. I will begin with the temperature test, since that one does not require the controller card. The controller card is scheduled to arrive early in the spring quarter. In the meantime I will learn about the protocol to communicate with it via the I/O box, so I will be ready to implement the remaining tests once it arrives.

Upon the arrival of the card, we will also need to test the firmware and make changes as necessary. Once the firmware is functional, we will be in a position to test the software modules that have been written at that time. At this stage we will determine how frequently we will run the tests, as well as how many times we will cycle through them.

Once the software has been written and these decisions have been made, it will be possible to create a GUI. I am not optimistic that we will make it to this stage before the deadline on May 22. At any rate, I will have completed a detailed flowchart of the functionality of the GUI and written the modules to perform the tests. Having these items completed will put me or someone else in a good position to build a GUI at a later date.