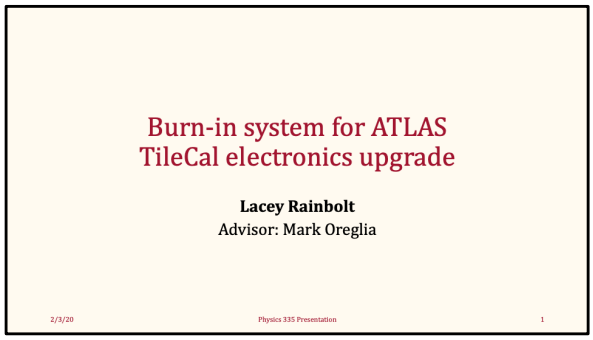
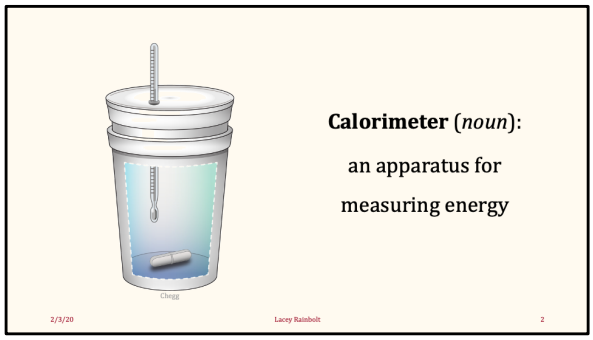


Hi, I'm Lacey Rainbolt

My 335 advisor is Mark Oreglia, and the project I'm working on is a burn-in system for the ATLAS TileCal electronics upgrade.



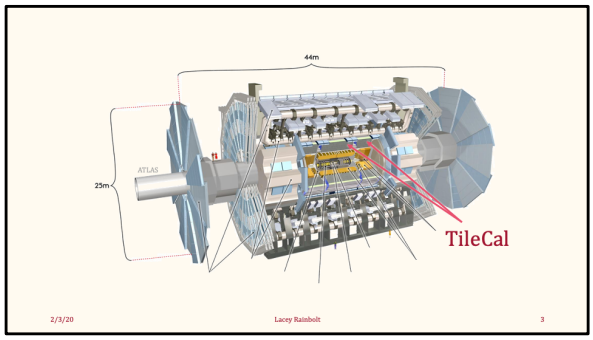
So first of all, what is the TileCal?

Well, the “tile” stands for “tile,” and the “cal” stands for “calorimeter”

Maybe you remember, from high school chemistry, using a coffee cup calorimeter to measure the heat--and therefore the energy--released in a chemical reaction

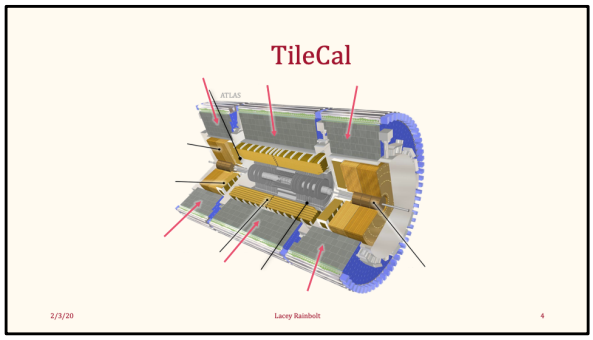
And in the ATLAS detector, the tile calorimeter is used to do pretty much the same thing. We collide a couple protons, and tons and tons of other particles come flying out, and we need to measure the energy carried by each and every one of those particles individually. With incredible accuracy, and on a huge scale.

So, to make a gross oversimplification, the ATLAS TileCal is like a massive, high-tech coffee cup.



So, to refresh our memory, here's a picture of the entire ATLAS detector, which is huge, with some tiny people standing on it for scale

We're going to zoom in on that middle core region, where there are some pink arrows pointing to the TileCal,



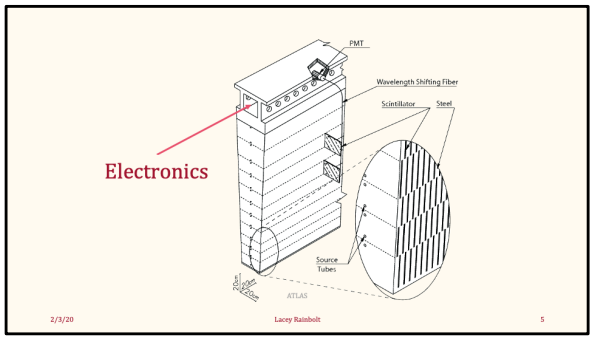
Which here we can see a little better.

The areas the pink arrows are pointing to are the visible areas of the TileCal

You can see that it has this tiled pattern, and that's where the "tile" comes from in "the name

The entire TileCal wraps around this whole cylinder, and it's separated into a bunch of wedges, 4 across by 64 around

And if we zoom in on one of those wedges…

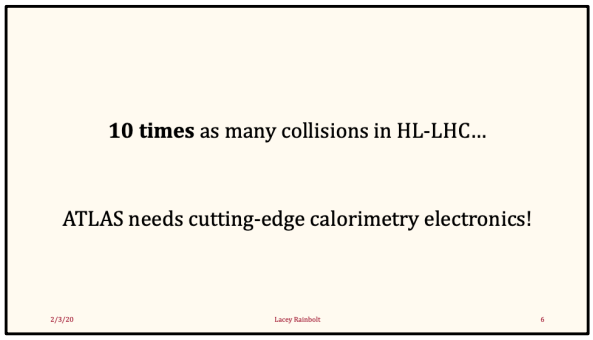


We can see where the electronic components go, which is in this slot on the top, so on the outside of the cylinder

In this bubble on the right, we can also see the tiled pattern of scintillator and steel

The steel absorbs energy from the particles produced in the collision.
A particle comes up from the bottom and collides with one of the steel nuclei,
which creates a cascade or “shower” of decays containing all sorts of particles.

When these decay products hit the plastic scintillator, some of their energy is converted to photons, which we can then count with our electronics to measure the energy.
Interspersing the steel and scintillator plates in this “tiled” fashion makes it possible to get a good idea of where exactly in the calorimeter these decays are occurring, and associate them with one incoming particle.



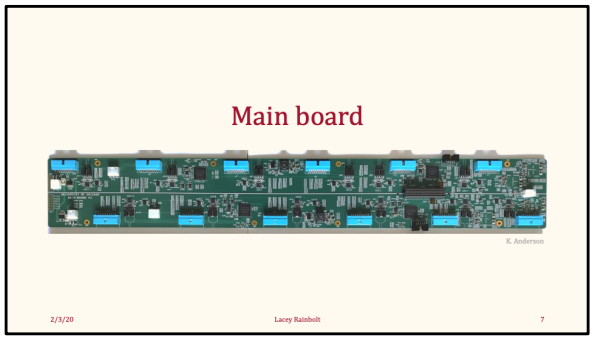
The current TileCal system has been working quite well for the last ten years or so that ATLAS has been collecting data

But, in order to keep up with the times, all of the LHC experiments are getting a major upgrade

This new era will be known as the HL or “high luminosity” LHC, where “luminosity” basically means “number of collisions”

The collision rate will be five times what it is now, which is 1000 collisions per second per square centimeter, so five times *that* and data will be collected for twice as long, which makes for 10 times as many collisions overall

The HL-LHC turns on at the end of 2027
So the TileCal electronics need to be upgraded in order to handle all of the radiation…
and to make the most of advancements in technology that have been made since the original system was designed about 20 years ago.

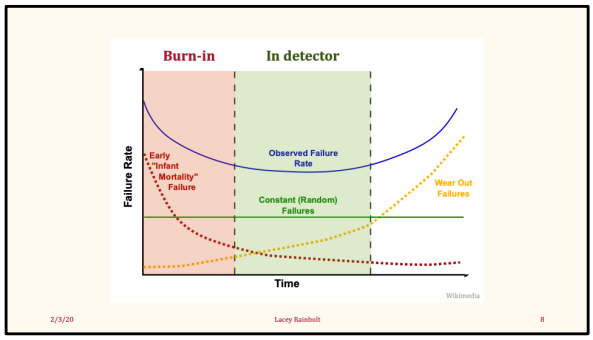


So here's a picture of one what is called a "main board" for the TileCal

This is the component I'm *also* testing for my project, so you can tell it's pretty important

Each one of these is just over two feet long, and there will be nearly 1000 of them in the detector

All of which will be tested here at UChicago



So for the testing, there's this phenomenon in electronics engineering known as the "bathtub curve," which is shown here

Basically, electronics are most prone to failure very early and very late in their lifetime – so, maybe a little like humans?

So what we're concerned with right now for the main boards this early "infant mortality" period, where this red dotted line indicates a high failure rate early on

If this is going to happen, we want it to happen now, in order to greatly reduce the possibility that any boards fail after we've put them in the detector

So we speed this first section of the process along by doing what's called a "burn in."

The burn-in will take place in this room shown here, which is in the accelerator building

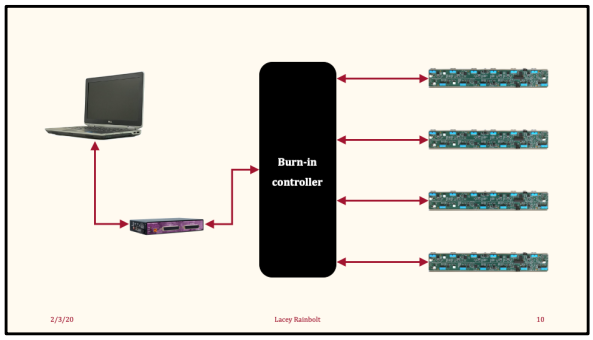
A lot of it has already been set up and ready to go

Here we have five stations, each which has two of these wood boxes
Each box will hold two main boards—in the middle you can see one board in there
Inside each of these boxes, under the panel the boards sit on, is a heater.

By keeping the boards at a high temperature as we run tests, we can make them age about one year in only one week.

We'll need to burn in about 1000 boards, so it will be some undergrads' job to change out four boards every day and keep an eye on the laptops, which will show output from the tests

My job is to work on the firmware that runs the tests and to design a GUI that makes the output undergrad friendly.



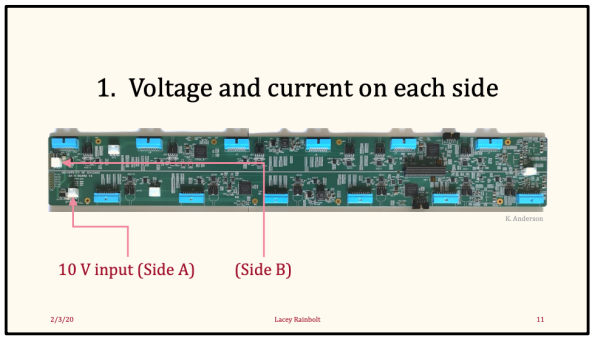
So here's basically what tone of those stations is composed of

We have a laptop running RedHat Linux, which is used as a user interface to monitor and control everything

The laptop connects via USB to this purple I/O box, which is used to translate commands from the box and read back data

This big black box is a custom controller we'll use to run tests on the boards during the burn-in. During normal operation the daughterboards are used to communicate with the main boards, but we can't expose the daughterboards to burn-in conditions because they're too delicate and expensive.

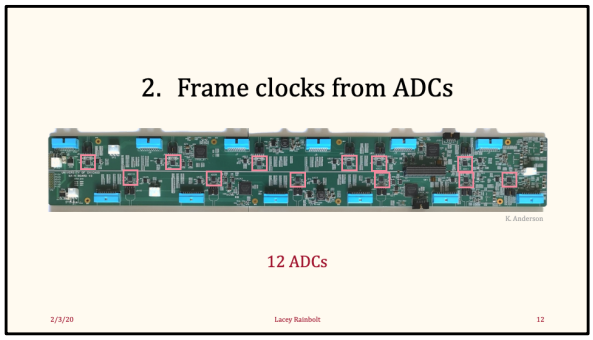
So instead we program this controller to monitor some important main board functions. The reason I drew it as a black box is because I actually haven't seen what it looks like yet, since some other people are using it right now, but hopefully that will change very soon.



First is the voltage and current from each side of the board

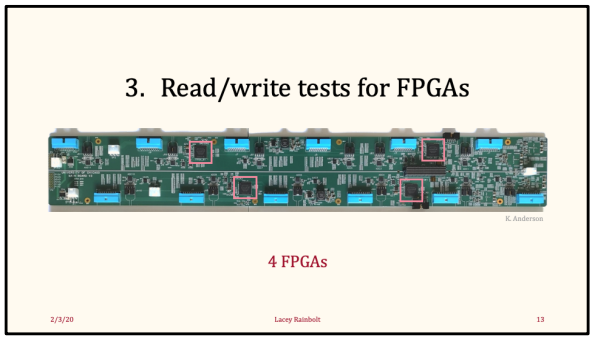
Each side of the board has its own power supply to minimize failure, and is designed to run at 10 volts

We'll run the boards at five different voltages *below* 10 V and monitor the voltage and current from each side.

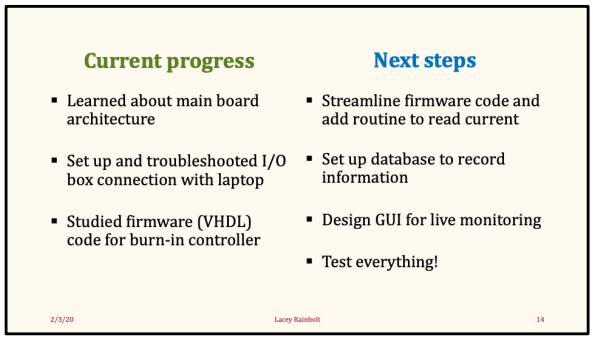


Second are the frame clocks from the board's analog-to-digital converters, which digitize the data coming in from the FENICs cards

There are twelve ADCs in total, and we need to make sure that their timing is synchronized across the board so everything works correctly



Finally, we need to make sure the FPGAs on the board are working properly.
The board is divided into four sections and each is controlled by an FPGA (field programmable gate array),
So in order to make sure they're working correctly, we'll send in random binary numbers and make sure they read the same number back to us without corruption or loss



So here's an overview of what has been done

Firstly, I spent some time learning about how the boards are set up

Next, I figured out how to correctly link up the purple I/O box with the laptop. Since I don't have the controller right now, I connected it to another FPGA controller we had in the lab, so I had to learn a little bit about how to program *that* and make sure that was working, and reconnect some things I had plugged in backwards, but it's all good now (thumbs up)

What I've been doing most recently is studying the current version of the "firmware" for the burn-in controller
This was written by Kelby Anderson, who is research faculty who does a ton of stuff for the TileCal, and who's been helping me a lot with figuring all of this stuff out

Anyway, the code he wrote for some earlier tests is written in VHDL, which is a hardware description language and I'd never seen anything like it before!
So I've been trying to understand how everything works, and what part of the code does what.

Now for what remains to be done

What I'll do next is start streamlining the code with some custom headers to make it easier to read and maintain
And I'll also need to add the functionality to read in the current, since that's something it was decided to monitor more recently

I'll also need to set up a database to record some subset of the information that we obtain during each test,

Set up a GUI for the live monitoring that will later be performed by one or more undergrads,

And finally, test all of this and make sure it works…
Just like the main boards will work when we put them in the detector :)