# Slide 2

Hello and welcome to the Introduction to Autodesk Eagle training session. In the first of this two parter I will show you how to design a schematic in eagle. The Avionics set up we will be designing is based of the one that is going to be used in both of the rockets that will be launching at Spaceport America Cup this year.

Now before you get to work in Eagle, you’re going to have had to do some avionics design. You’ll need to have picked your main processor like an Arduino nano or a Teensy 3.6 like we use in VESNA. You’ll also needed to have selected your IMU and any other sensors you want, plus you’ll need to have a general idea of the capabilities of your rocket.

How will you start your rocket?

If you have an independent payload how will you deploy it?

How will you launch your parachute?

# Slide 3

Once you’ve settled on these components, you’ll need their part libraries so you can actually use them in Eagle. Many electronic component stores include links to relevant part files like CAD models and Eagle files on a store page, but sometimes you will need to browse around to find the correct version for your component.

If you download the library for a personal project on a single PC, simply paste the file into the libraries folder of you Eagle install. But if you need to use the project on multiple PCs or are working in a group, uploading the file to Autodesk Library.io website will allow you to share a project’s parts will multiple Autodesk account. Plus, this will then auto-sync you libraries into eagle.

# Slide 4

This is the Control Panel for Autodesk Eagle. Its where you can easily adjust your program settings and it is where you will need to begin to start a new schematic.

First, you’ll need to open a new project to store your schematic in, this keeps things contained and is what will eventually mean we can easily transfer a schematic into a board for the final design.

The first thing you will need is a place down a frame, this will keep your workspace neat and organised. I’ve selected a landscape C sized frame so that there will be enough space for all the components.

I’m also going to divide this frame into four boxes. VESNA avionics platform is designed to be modular to ensure that it can easily be ported over to another rocket without needing a ground up re-design.

# Slide 5

We are now going to want to start adding in our components, but to do that we need to make sure they are included in the schematic. First, you’ll want to open your library manager, this is what syncs with both your Autodesk library.io and the library install folder on your PC. To include a library in your project simply find it in the available tab and hit use.

# Slide 6

These four sections will exist as sperate PCB plates, that way only a single plate will need to be swapped out. In our new Control unit section, we are going to add in the part of our Teensy 3.6, our main processor. We will also add in 4 LEDs with resistors in series to the UI section, these will act as a basic output for error codes so that we know if anything is wrong during boot up.

# Slide 7

We need to connect the LEDs to their resistors and to output pins on the Teensy this is done using the net tool. We are including wiring the LEDs’ cathode to a ground symbol, these work universally across our schematic, so as long as we add the same ground signal each time, they will all connect up to the same point.

# Slide 8

VESNA uses two IMUs, the DFROBOT is our low-G sensor and will be used when precision is required. This is most critical at apogee when we need an exact reading to know when to open the fairing and release the payload and drogue chute.

Due to Eagle’s search engine being exact we need to know the precise name of each component to find it. This doesn’t work like google, even the case must be correct.

# Slide 9

The ADXL 377 is then the high-G sensor used for the initial take off. This has a substantially higher range of +-200g but a lower Voltage per G output making it less precise. There is no library part for this sensor, but there is one for the ADXL 335 which is the same size and has the same pin positions. All we need to do is rename them to be in the correct order.

# Slide 10

Our Teensy also needs a power input. It can take anywhere from 3.6 to 6V, we have chosen an input of 5V for safety.

# Slide 11

We also need to have a master switch that controls the power in to the Teensy. As this will not be mounted to the PCB directly but instead to the skin of the rocket to it can easily be reached, we need to then wire it externally to the PCB. For this we use 1 by 8 header to represent each of the button’s pins. From looking at the schematic we can see how each pin need to be wired up and where the connection is to include the switch in series.

# Slide 12

The Teensy 3.6 has various pins that must be connected for it to work properly. These are mainly various extra ground pins dotted around the board. Always make sure you’ve consulted the datasheet, so you know what pins need connecting, you don’t want to blow a central component just because you missed one pin.

# Slide 13

If you need access to some pins but you’ve left nets in the way, just use the select tool to group and move them. If you move them along and axis Eagle will shift the nets cleanly.

# Slide 14

Now we have access to the pin we want we can start to build our set up for the solenoid connection. This solenoid controls the opening and closing of our fairing. As the solenoid requires a 12V supply and is being switched on and off by a 3.3V signal we use an opto-isolator as a non-mechanical light gate, this way there is not signal interference and mechanical vibrations cannot turn on or off the solenoid. We also need to include a Darlington Transistor to make sure the current is stepped up to what the solenoid requires.

# Slide 15

The whole avionics set up is powered by a 14.4V liPo battery which is stored off the board. As such we just need to include a 1 by 2 header to connect the battery’s cables to.

# Slide 16

Then we can insert our voltage regulator, we are going to need 3 of these. One for 12V, one for 5V and one for 3.3V. The circuit set up of the Voltage regulator can be found inside its datasheet.

# Slide 17

As the battery is a LiPo we don’t have to worry about substantial current loss from splitting the circuit into parallel for the 3 voltage regulators.

# Slide 18

The polarised capacitors in the Voltage regulator setup deal with high frequency noise suppression. We still need a solution for low frequency. This can be done by putting another high value polarised capacitor across the battery supply.

# Slide 19

Having copied and pasted the Voltage regulator setups we will now need to change their outputs to 5V and 3.3V so that all of our components can receive the right power.

# Slide 20

The Voltage regulators can be set to output a certain voltage by setting two resistors to certain calculated values. The equations that define these resistors can be find in the voltage regulator’s data sheet. Various key bits of info can also be found here; a larger ratio of R1 to R2 means a higher Vout, R2 has a minimum of 10k Ohms and the voltage adjustment input is 1.24V.

With this info you write down your equations. In order to keep R1 at the minimum possible R2 must be set to 10K Ohms. Then you grab a calculator and find R2. Standard resistors don’t come in every value, so you’ll need to use another calculator so find the nearest possible option. Make sure you know if your initial value is the minimum or max allowed.

For this the value is a minimum so R1 can be set to 91K Ohms and R2 to 10k Ohms. Repeating the Vout equation gives us the final Vout value of 12.524V. it doesn’t matter that this is slightly over has some voltage will be lost in components. To set a resistor find the Value button on the left hand side that looks like a fraction with 10k in bold on the denominator.

# Slide 21

The process should then be repeated for the other two voltage regulators. As you can see this gives us final Vouts of 5.07V and 3.44, both close enough to the desired output.

# Slide 22

Our LED’s also need to have their resistor values set. When it comes to these sorts of small 5mm basic LEDs there are only two resistor values I would suggest. 330 Ohms for safety and 220 Ohms for brightness. As the avionics is mounted in the transition section of the rocket it may be hard to see the LEDs, so 220 Ohms is the best option.

# Slide 23

The Solenoid setup also requires calculated resistor values, the equations for which can be found in the datasheets of the components. The opto-isolator can only take 1.4V otherwise the IR LED will blow, and the high 12 volts signal will never pass through.

The next problem is that the opto-isolator only outputs 50mA which is nowhere near enough for the solenoid. To solve this, we step up the current using a Darlington transistor. Like the voltage regulators the transistor’s current output is controlled by the value of a series resistor.

After finding the transistor gain, we can complete the equations and find the required resistor value.

# Slide 24

Then just like before we can easily edit the values of the resistors and type in the value we want.

# Slide 25

And that is us all done. All the components are correctly wired up and the resistor can capacitor values are set to make sure the circuit wont blow.