Calibration and Measurement

When the microcontroller is going through the calibration phase, a couple of pre-coded checks are put in place to ensure that the bricks were assembled in a way that will allow the sensor to succeed.

Rainbow Phase

First, the microcontroller begins with a quick rainbow animation to show that the microcontroller has turned on!

Yellow Phase

Next, the microcontroller reads the light sensor value before and after turning the white LED on. If the two measured values are the same, the purple error lights are shown to tell the builder that there might be an issue with their design!

Cyan Phase

If the yellow phase succeeds, the microcontroller will turn the LED brick on, and read a bunch of sensor readings over two seconds. The microcontroller will remember the largest and smallest values recorded, which will be used in the next phase.

Red Phase

When the cyan phase succeeds, the microcontroller will begin frequently reading the light sensor, and determining an air pollution value. The air pollution sensor works as follows:

1. First, read the light sensor
2. If the light sensor detects a new minimum or maximum light sensor value, it will be remembered for later.
3. Next, the microcontroller will take the region of numbers between the minimum remembered value and maximum remembered value, and divide it into eight different sections, one for each LED on the display brick. Depending on the region that the current measurement falls into, the corresponding amount of red LEDs on the display will light up!
4. This will loop back to step 1) and repeat forever.

This four-step process also explains why at the beginning right after calibration, the air quality will be measured much higher than normal. The microcontroller has no other measurements to compare to, so the eight divided regions are *very* small. Extremely small changes in the light sensor reading are enough to change the eight red LED measurements. But, when you introduce the particulate matter, a new maximum is introduced to the microcontroller, and the sensor becomes more accurate.

Individual Component Overview

Work in progress, to be completed 8/11.

**Microcontroller Overview**

So the “brains of the air quality sensor is a microcontroller, a very small and programmable computer! The microcontroller we are using is called the Raspberry Pi Pico, which is only 4 USD and programmable in a computer science language called Python. The Raspberry Pi Pico has 40 pins that you can turn on or off. To these pins, you can connect wires and control sensors, lights, and many other components.

**White LED Brick Overview**

The white brick uses a white LED that produces white light when connected to the microcontroller. The bricks were custom designed in a Computer-Aided Design (CAD) software called Solidworks and printed on a 3D Printer. 3D printers melt plastic and draw 2d pictures of plastic on top of each other to create a 3D shape.

**Sensor Brick Overview**

The teal sensor brick uses a photoresistor to detect light. A photoresistor is an electrical component that will change how much it wants to resist the flow of electricity based on the amount of light shining on its surface. Similar to the white LED, the teal bricks were designed in Solidworks and 3D printed.

**Display Lights Overview**

The long, 6x1 brick holds a printed circuit board with 8 Neopixels on it. A Neopixel is an RGB LED, which has one red, one green, and one blue led inside of the larger package. Neopixels are special in that you can control specifically the color of each individual light, all from three wires!