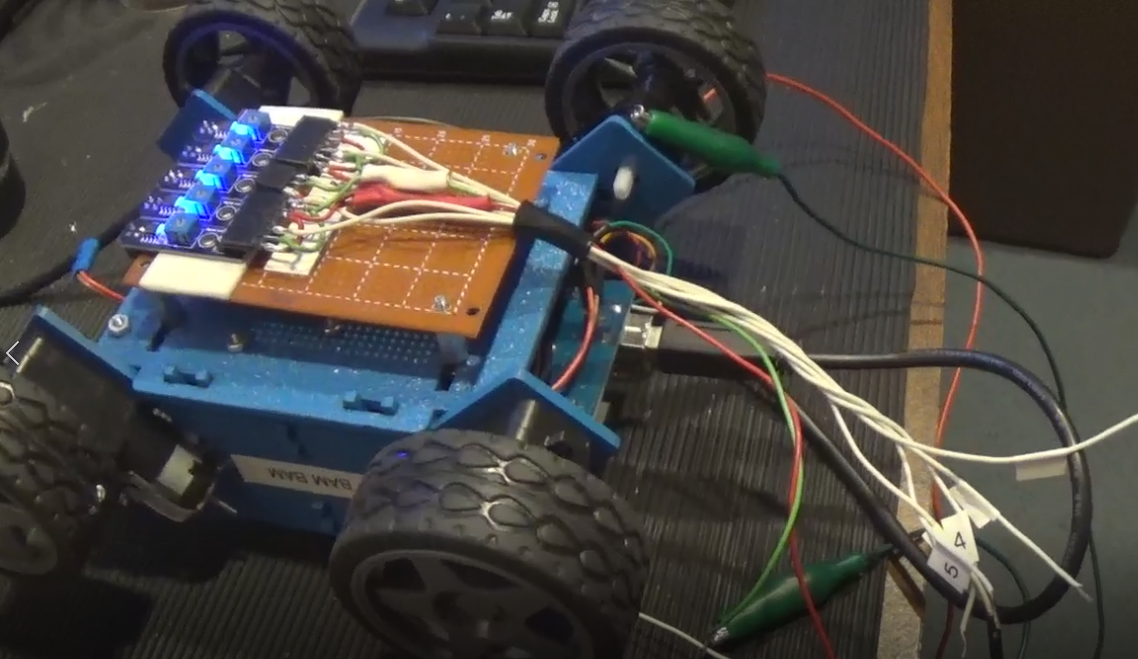
**Ed’s Robot Project - Week 3**



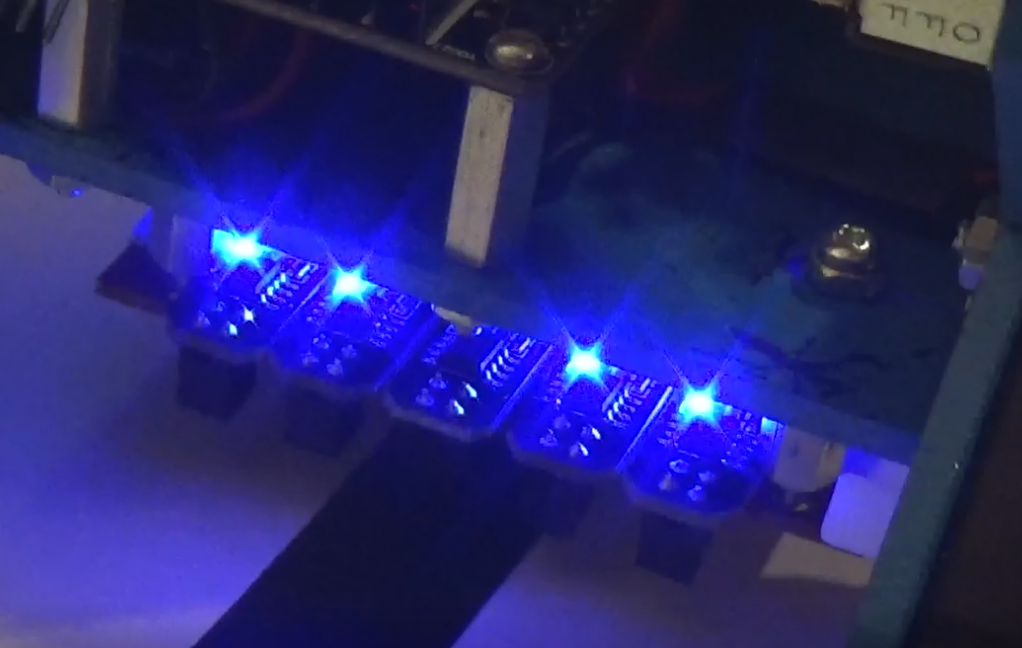
**It was a pretty quiet week. The line sensors didn’t arrive until Thursday.**

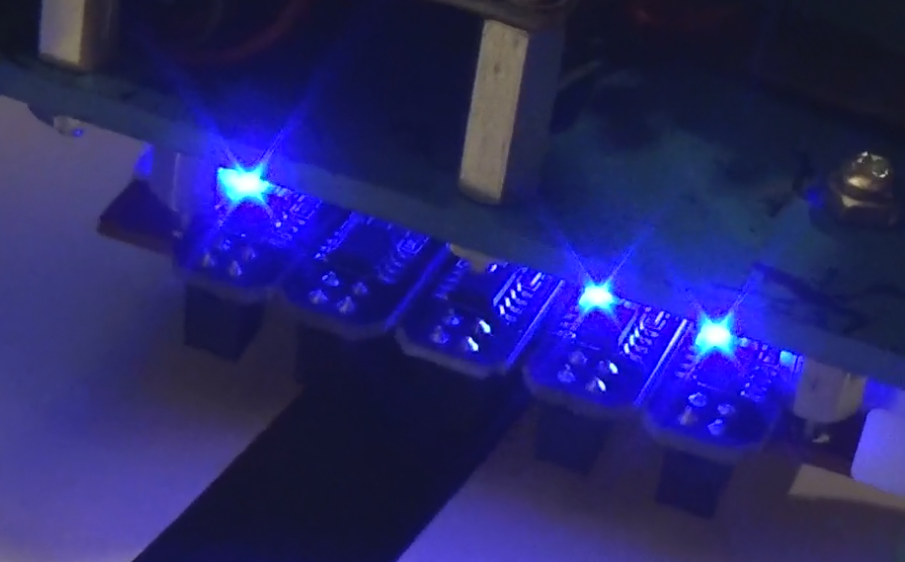
**Limited on digital inputs, but I had all six of my analog inputs available. I had already decided to use five line-sensors for the project AND to mount them in the center of the unit, logically reversing them for tracking the line backwards.**

****

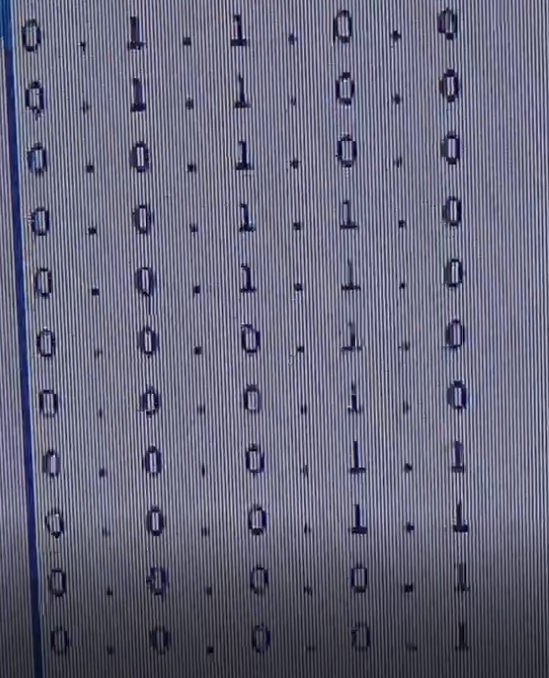
**I used double-sided tape to mount them for wiring, testing and adjusting them for the correct distance. **

**After adding some code to digitize the sensor inputs and display the values, I put a piece of black tape on a sheet of paper and dragged the sensors across it.**

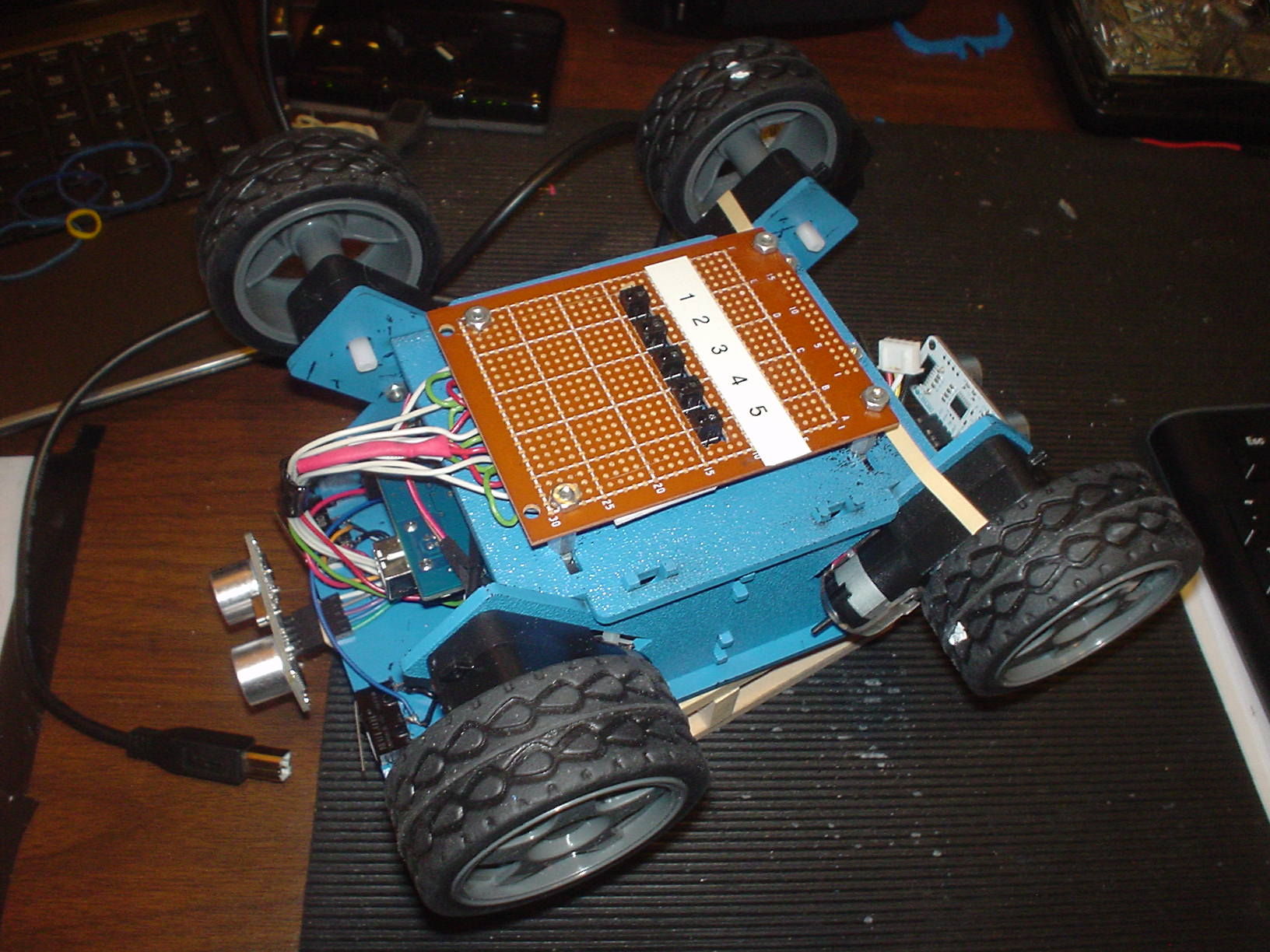
****

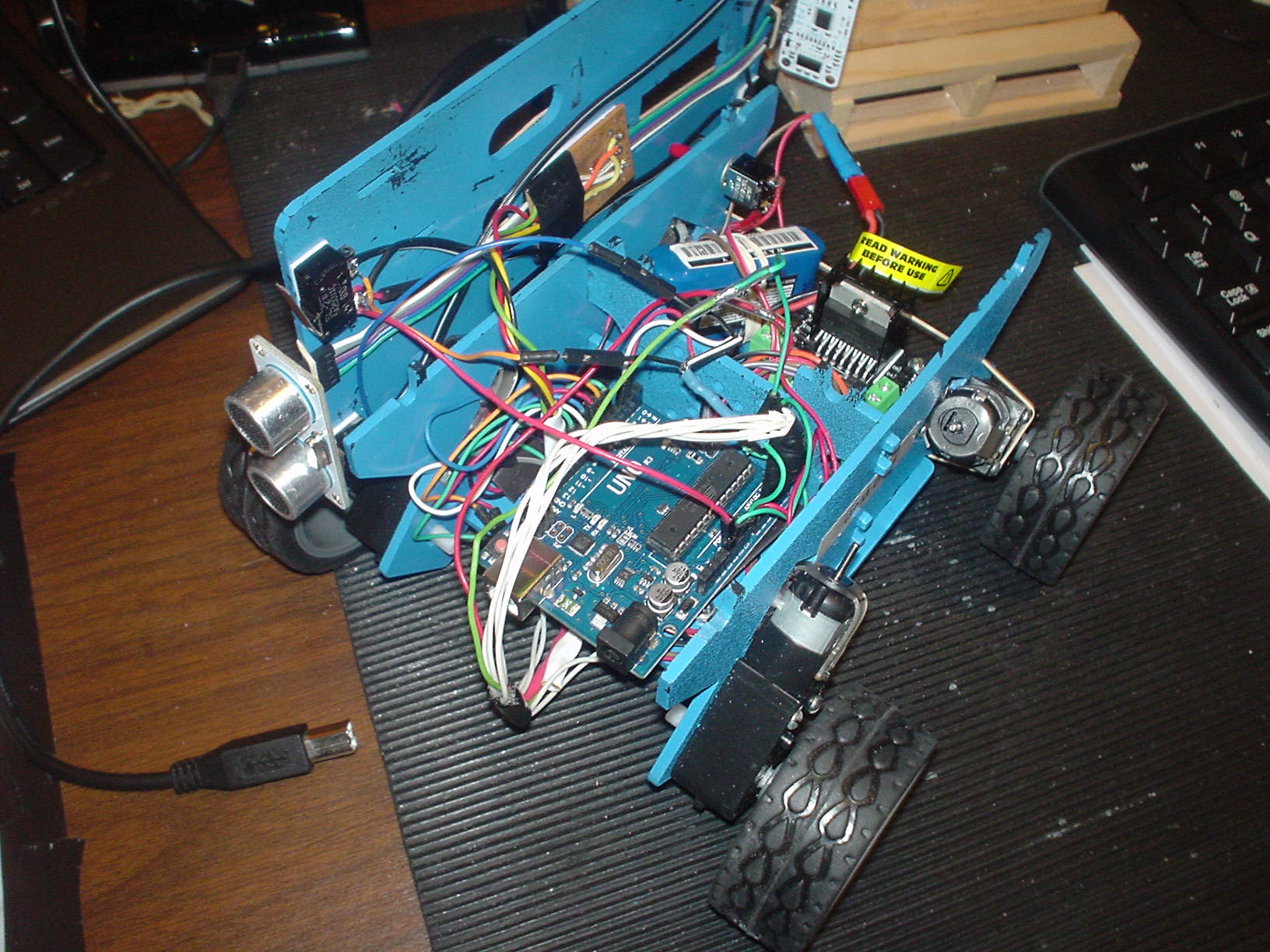


**The sensors are spaced to the width of the tape. First one sensor “sees” the tape, then that sensor and the next one both see it, then the only the second sensor and so on…**



**After calibrating, the sensors were mounted in the center of the chassis and connected to the analog pins.**





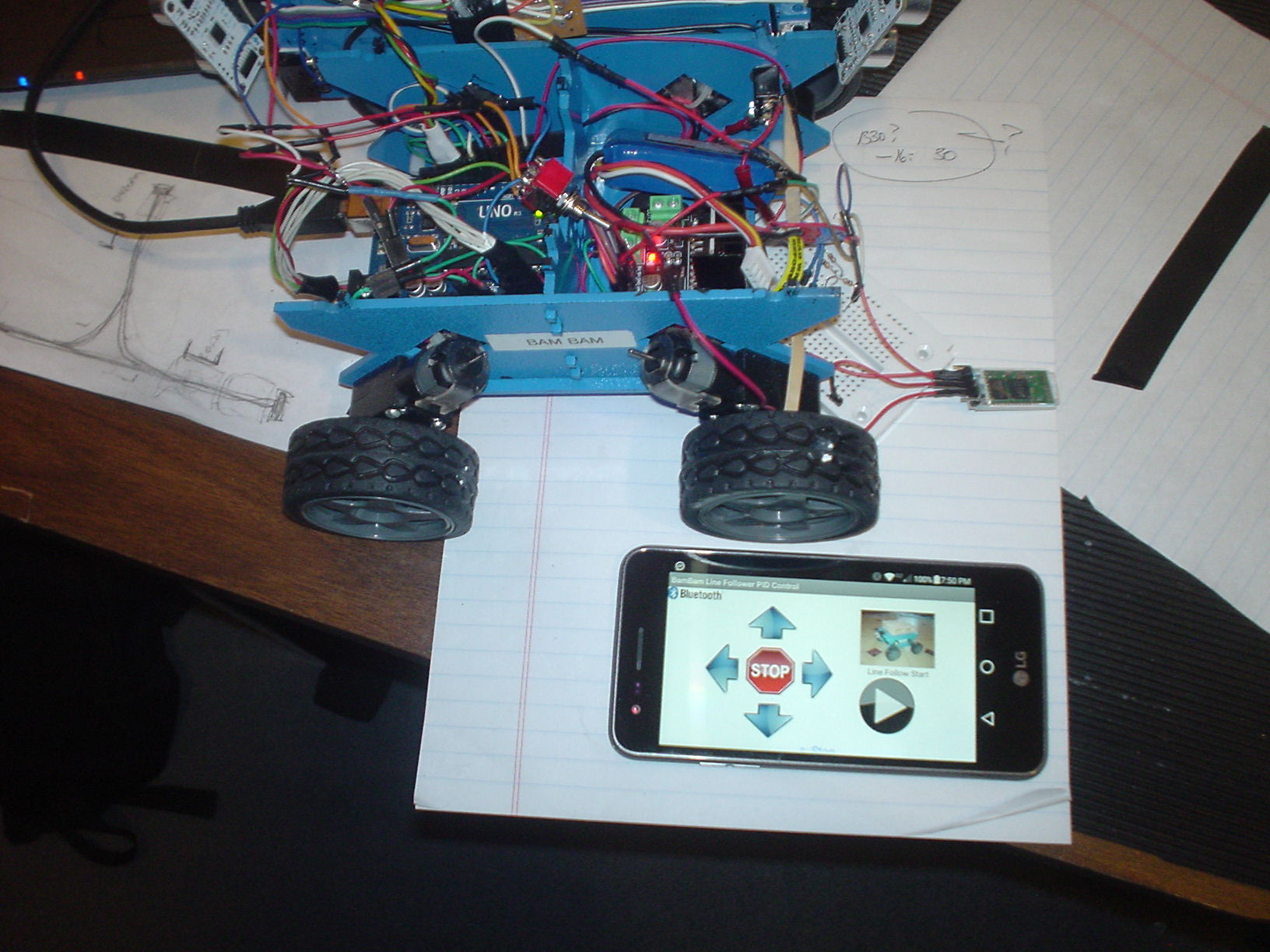
In the code, sensor 1 and 5 were turned off to tweak the code for tracking a straight line using only the center three sensors. Code for reversing the robot at the end of the line and tracking while traveling backwards was added but was not tweaked.

I’m thinking of using that method in conjunction with the right and left switches. Hitting the left switch would kill the right-most sensor (from that orientation) and hitting the right switch would turn off the left sensor – to blind the unit from reacting to a track in that direction. Maybe that will help - just a thought.

The ultrasonic rangefinder wiring was a do-over after a broken connection damaged one of the sensors. Fortunately, reflowing the solder joints brought the sending unit back to life. My motor controller also failed, probably from prolonged abuse during the forward/ reverse tracking sessions. My left motors were dead. Luckily there was a spare – I had purchased two of these thinking I had to control each of the four wheels independently.

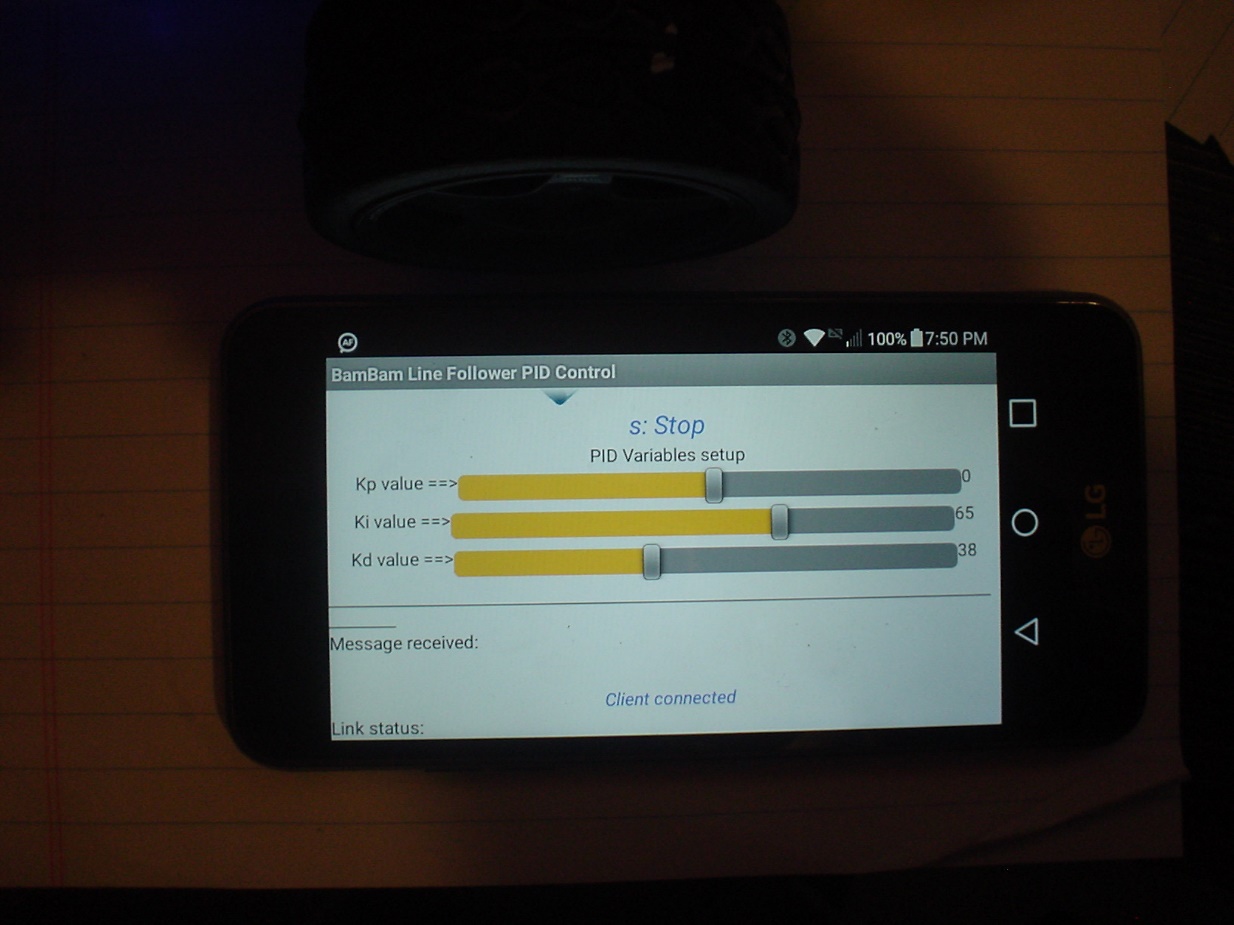
My thinking with getting the unit to track a straight line with just three sensors is I’d use the outside two sensors for turning. I implemented a poor man’s proportional control by saving my previous sensor reading before setting my turning parameters to correct my path. Manually tweaking and testing the straight-line code was a time killer - and it’s far from perfect. I did some searching and I found a robot project using PID code and an Android / Bluetooth interface. I picked up a Bluetooth card - which drove me nuts until I tested it and found it was bad! Ran out and got another one (after testing it in the store). I got the Bluetooth card hooked into the last two pins on the Arduino, the Bluetooth interface put into the robot code and the Android app communicating with Arduino!

I customized the Android controller app. I’ll need to wrap my head around the PID code and retrofit it from a servo-based robot to my four-motor project. A lot went together in the past few days and it looks like it:



**Some major housekeeping is in order before I’m back on the road!**

My Android phone interface will allow me to tweak the PID values “in the field”:



Except for rewiring, the hardware build should be finished.

I’m hoping that PID control will get it turning corners smoothly.

