The challenge given in METR2800 was to design and build a robot that would make use of a reaction wheel (flywheel) to rotate itself and fire a laser at active infrared targets. The robot had to first locate the position of the active target, then fire the laser at it for 3 seconds. Targets had different zones worth different points (10 for a bullseye, then a 5-point and a 1-point zone) and the points awarded would be the minimum of the zones the laser touched while it was on (i.e. you couldn’t strafe across the target).

The robot had to use a flywheel to turn left and right but was able to use a servo for up and down aiming of the laser. Any circuit boards present in the final product could not be prebuilt and the microcontrollers used had to be from the Atmel AVR family. This ruled out the use of boards such as the Arduino and all but necessitated that the circuit board for the project needed to be specially designed.

My Contributions

Early CAD design and keeping the CAD files updated

Early in the project, I designed the internal structure of the robot, leaving space for the electronics and mechanical drive system for the flywheel. I designed the flywheel, shaft coupler and motor mount for the system. I designed the flywheel to be as wide as possible to maximise its rotational inertia and therefore the systems control authority. The shaft coupler was designed to be 3D printed and connected to the shaft using grub screws threaded through heat set inserts in the plastic coupler. The motor mount was designed to be made from sheet metal aluminium but ultimately the 3D printed prototype I made to check the fit worked perfectly well so we stuck with that.

Throughout the project, I maintained the CAD file to ensure that the design on the computer matched the product we were creating in real life. This included adding and removing temporary supports, adding holes that were drilled for mounting, and keeping track of our cosmetic enhancements.

ECAD Design

I designed the circuit board we used for the project using Autodesk’s Eagle software. Due to the short time constraints on the project, we could only really afford to create one version of the PCB before the demonstration. This meant that it was critical that I included all the necessary functionality into the first version of the board. At this point, the group had decided on what we thought were the necessary components for the functionality required but we could not be sure as we hadn’t tested. The main concern was having the option of adding more sensors if we found it necessary. As such the board was designed to accommodate the minimum hardware with the option of adding an additional 8 infra-red sensors using a multiplexed input. Ultimately, we didn’t need this, and the functionality of the first design proved satisfactory for the project.

Software

One of the key points of the task given to us by the course instructors was that we could choose to have the IR targets emit either a constant signal or one pulsed at a frequency assigned to each group. It was favourable to use the PWM option since the room in which the testing would occur had a very large window that could easily give a false positive as a target. Given that we all hadn’t really been taught how to properly handle PWM signals it was a source of much confusion for the cohort. Several groups spent many weeks and made really good attempts and designing filter circuits that would process the signals but ultimately, I was unaware of any group who found success in this method.

Our group decided that it would be best to use software filtering to determine if a detected signal was PWM or not. The code I wrote for it store 30 readings of the sensor taken in quick succession and then found the average of that dataset. If there were peaks and troughs present that differed to the mean by a given threshold, this would indicate the presence of a target.