Motor Control Progress Report #3

Skittle Sorter Project

Jaidon Lybbert, Motor Control Design Lead, EWU IEEE Student Chapter

Week of 07/19/20

Problem:

- As code grows, it is necessary to make adjustments to the structure to make it more maintainable, in this case, my code had outgrown its single stepper.c file, and needed to be split into sections that made sense and were portable
- Development thus far had been using a unipolar 4-phase 28byj-48 stepper motor, which is not the motor used for our project, small modifications to the code had to be made to adapt to the bipolar 2-phase NEMA-17 motor
- The NEMA-17 motor requires a large current, and a beefier driver circuit, for this we decided on the l293B dual H-bridge driver, which would need to be interfaced with the motor

Research:

- Splitting files required some brush up on how to link compiled c object files using make, as well as what should and should not be included in a header file
    - https://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/
    - https://www.tutorialspoint.com/cprogramming/c_header_files.htm
- Learning about how a bipolar stepper motor is driven compared to a unipolar motor was a necessity. The following document contains very detailed information about different methods of driving stepper motors
    - https://www.st.com/resource/en/application_note/cd00003774-stepper-motor-driving-stmicroelectronics.pdf
- It was necessary to refer to the L293B datasheet to interface the IC with the motor
    - https://www.st.com/resource/en/datasheet/l293b.pdf
- Flyback diodes were used in the driver circuit, and consequently researched
    - https://en.wikipedia.org/wiki/Flyback_diode

Action:

- The huge stepper.c file was split into init.c, umc.c, and stepper.c, with associated header files
- The code was modified to add support for driving the NEMA-17 motor
- A “home” switch input was added to an onboard button along with an associated debouncing function
- The driver circuit was built, and interfaced with the tm4c and NEMA-17 for testing

Value:

- Maintaining existing code improves readability and prevents future issues from arising
- Constructing the circuit for the NEMA-17 allows for testing of the software and hardware, as well as further development of functionality
- Basic inputs to the system allows for more robust testing