Aspiration Assisted Motor-Driven Biopsy Needle

Report: 10-31-2024

1. **Work Planned to be Completed:**

This week my primary focus was the integration of an LCD and getting some input capabilities integrated into the current circuit. Ideally, I could hook up the LCD and have it read potentiometer values as analog inputs. These inputs could then be configured to represent actual values to drive the motor. In order to do this, I will likely have to modify the current state machine.

Additionally, I would like to look into alternative sources of power other than the 12V supply we currently have. Something smaller that could fit into the handheld frame would be ideal. Not only that but I want to look at getting some form of filesystem so that we can get the current data in a clear and organized format. This will require researching alternatives like some of the Raspberry Pi products which should be pretty straightforward given its my next lab in class. Obtaining or creating the remaining KiCAD components for the PCB is another important item of lower priority on my Todo list.

For future work, documentation, and version control I need to setup a repository. I will likely create the GitHub repository relatively quickly but I need to find a way to get it either synced to teams and plan to manually update it.

1. **Work Completed:**

My primary task for the past 2 weeks was getting some form of input so we can adjust the motor speeds. I decided to use a simple potentiometer circuit that sends a 0-5V value to the Arduino’s analog read pin. With this I could map the ADC output to something useful for the motor. The general operation of this can be seen in a video I have uploaded to the TEAMS.

A hand holding a small blue screen with wires

Description automatically generated

***Figure 1:*** *This was the text circuit for the LCD to see how viable potentiometer inputs could be. I previously thought 5% increments were about as stable as I could get but I now believe that we can get accuracy to 1%.*

This implementation required that I change how the states were handled. Before, each state was effectively executing the code of the future state so I have moved all of the code behind so it makes more sense logically. Additionally, I switched out the previous if-else structure for a simpler switch-case structure and changed how some of the safety features like minimum and maximum distances were checked. Before it was possible that the checks could “lock” the states and so transitions would require two button presses instead of one. That has been for the most part fixed aside from some button bouncing which I will have to look into.

A screenshot of a computer program

Description automatically generated

***Figure 2.1:*** *This shows some of the modifications to the state machine, particularly a call to a function that reads the desired input speed.*

A screenshot of a computer program

Description automatically generated

***Figure 2.2:*** *This shows the actual function that reads the desired input speed.*

Additionally, I was able to speak with Carsten about a few ideas regarding how to power the circuit and I am now narrowing down on several options for the potential battery. A 12 V 10 A battery is way too large to be practical but given the circuit only reaches slightly under 2 A at max load, there is more flexibility than previously expected. Right now I am considering several different 6V cells that I could put in series to generate the needed 12 V (we would only need two batteries). I will probably make my final selection within the week and order them.

I was also able to get the GitHub repository up so now all of the files I am working with are accessible and the changelogs can be viewed here:

[**https://github.com/redtrek/ntm\_needle\_biopsy**](https://github.com/redtrek/ntm_needle_biopsy)

A screenshot of a computer

Description automatically generated

***Figure 3:*** *New GitHub repository.*

Finally, this week I was able to begin my design lab with the Raspberry Pi Pico which is currently the most promising option for getting a functioning file system. Once I grow more familiar with working with the Pico I believe we could move all the code over to the Pico so that any recorded data can be more easily accessible and analyzed.

A screenshot of a computer

Description automatically generated

***Figure 4:*** *File system accessible via the Pico. Will need some work to make this more natively accessible to a Windows and will likely require some code for USB data transfer.*

1. **Future Work:**

Ideally, I would like to focus more on getting more ground on the PCB design. Next week I plan to focus more on that KiCad library I spoke about earlier. Code wise I am also looking into getting some proper velocity and rpm calculations. Additionally I need to decide whether migrating to the RP2040 is a good idea or if I should move to something like the Teensy. On the hardware side, I want to order those new batteries and debounce the pushbutton we have.

**Current CEN3907C Due Dates**

|  |  |
| --- | --- |
| *Assignment* | *Due Date* |
| Pre-Alpha Build | 10/25 |
| Driver Lab | 11/1 |
| Elected Lab | 11/14 |
| Design Plan Revision | 11/1 |
| Prototype Presentation | 11/19 |
| Design Prototype | 11/22 |
| Presentation Reviews | 12/3 |