ENGR3499B: Real World Microcontrollers
Assignment-2: User Experience Design

Note: Assignments 1 and 2 are to be done concurrently. You have _**3 weeks**_ to complete both. Feel free to finish faster if you want more time for your project. Grading will be done once you have completed all steps, however be prepared to discuss your progress on both assignments, during our weekly sessions.

_____

An area where engineers have a bad reputation is in designing products used by non-engineers. Good product design should be driven at every step by considering the user experience for everyone. Often we're accused of focusing on just the technical or cost factors. Let's not be that kind of engineer!

> The goal of UX design is to "improve customer satisfaction through the utility, ease of use, and pleasure provided in the interaction with a product."

Bad UX is a bit of a pet peeve of mine. For example, I have owned many BMWs and found their console's user interfaces terrible. I have since switched to Tesla which is a pleasure.

In this class you are developing a "Product". Some of the UX decisions have already been made (by the physical package you received). All the other decisions, including the software and user interface, will be designed by you. The UX for the *Cat Feeder* is pretty simple, but as with any product, you should consider this work a priority!

Assignment:

> Before designing any hardware, choosing any components or planning the software, always begin by thinking about the user and what your project needs to do to make them happy. You will also want to think carefully about scheduling your tasks, ensuring your project is completed on time. In this class, *Failure is not an option.*
>
> Create two Google docs and share them with me (stanley.reifel@olin.edu):
>
> Document 1 – UX notes:
> - Name the document: *Cat Feeder - UX Notes - <you name>*
> - In the document, create these lists, and create them in this order:
>   1. A bullet point list of anything you can think of, that makes this product better for the cat (our primary user).
>   2. A bullet point list of anything you can think of, that makes this product better for the cat's human.
>
>   *Note:* The Cat Feeder doesn't do very much, so the two lists above will likely be pretty short. But there are a few important ideas to capture.
>
>   3. Start thinking about the product you are making. Make a bullet list of everything it needs to do. For example, if you were making a household thermostat you might include:
>   - Measure the indoor temperature.
>   - Control the heater and AC.
>   - Turn the heater and AC on / off based on the temperature.
>   - Have a display and buttons for showing the current temperature and for setting the turn on / turn off temperatures.
>   - Perhaps having different temperatures for different parts of the day (morning temp, daytime temp, evening temp, sleep time temp)…
>   4. Make a list of all the hardware functions the Cat Feeder Controller needs, such that it's capable of doing the tasks defined in your list above. Note that the UI hardware and motor have already been chosen by the Cat Feeder's mechanism. For this you get a small display; Up, Down, and Settings buttons; plus a stepper motor and optical homing sensor.

Note: An alternative to making a bullet list of your hardware, you might consider drawing a block diagram of your controller instead. This also has the advantage of being great content for you portfolio page due at the semester's end. No need for fancy graphics. A neatish hand drawn sketch is fine.

5. Now think about what the software needs to do, and make that list. I often think in terms of *modes* the software will have. For example, a thermostat might have these modes:
   - Normal operation: Show current temp, Cycle heater & AC on / off
   - Set the heater temperature
   - Set the AC temperature…

Document 2 – Project Schedule:

- Name the document: *Cat Feeder – Project Schedule - <you name>*

- Make a numbered list of everything that you need to do, to complete your project. You will likely be building two complete versions of your controller board. Your first design should include all the functionality you expect your project will ever need. The second version is just to fix mistakes. In this list, be sure to think about: Overall design; Choosing components; Designing the schematic and PCB; Ordering parts; Programming; Testing…

- Now create a calendar that includes everything on your list above. Be sure to reference the *Schedule* shown on the class website for due-dates and such. Feel free to add anything on your calendar that might interfere with your progress (such as Thanksgiving or assignments in other classes).

Note: Consider both of these documents "Living Documents". We will be reviewing them throughout the semester, and you will likely be updating them as you go.