ENGR3499B: Real World Microcontrollers
Assignment2 - Designing a Cat Feeder - Step 1

Note: Assignments 1 and 2 are to be done concurrently. This will not be graded, but it will be difficult to do well on the main project without doing a great job here. Also, 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 for 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, rather than the human 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've owned many BMWs and found the 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 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 the project needs to make them happy. You will also want to think carefully about scheduling your time, ensuring the project gets completed with perfection.

It's helpful to start by thinking about "why would anyone want a cat feeder". Well, this cat feeder is intended for the family that takes three day weekends away from home. Its job is to solve the problem "who will feed Fluffy while were are away".

Document 1 – UX notes:

- Create a Google doc and share it with me (stanley.reifel@olin.edu). Name the document: *Cat Feeder - UX Notes - <your name>*
- In the document, create these lists and create them in this order. The idea here is to get you to think about a product's design using a step-by-step procedure, beginning with the most basic concepts, ending with the engineering details.
    1. A bullet point list of anything you can think of that makes this product better for the cat (our primary user). Call it *What The Cat Wants*.
    2. A bullet point list of anything you can think of, that makes this product better for the cat's human: *What the User Wants*.

       Note: The Cat Feeder doesn't do very much, so these two lists will likely be pretty short. But there are a few important ideas to capture.
    3. Start thinking about the product that you are designing. Make a bullet list of *What The Cat Feeder Needs To Do*. For example, if you were making a household thermostat, you might include:
        - Turn the heater on when it's too cold.
        - Turn the AC on when it's too hot.
        - Have a way for the user to input what's too hot and what's too cold.
        - Let the user know the current temperature.

- Perhaps allow the user to set different temperatures for different parts of the day (morning temp, daytime temp, evening temp, sleep time temp)…

4. Taking your ideas from above, make a new bullet list of *How The Cat Feeder Is Going To Do It*. For our thermostat example you might include:
   - Measure the indoor temperature.
   - Be able to turn the heater / AC on and off.
   - Control the heater & AC based on the temperature.
   - Have a display showing the current temperature.
   - Have a display and buttons for setting the turn-on / turn-off temperatures.
   - Perhaps having a UI for more complicated programming (morning temp, daytime temp, evening temp, sleep time temp)…

5. Now make another list that includes all the *Hardware Blocks* needed by your controller to accomplish those takes. Our thermostat would need:
   - Source of power
   - Microprocessor with memory to store user settings
   - Display and buttons
   - A temperature sensor
   - FETs or relays to turn the heater / AC on and off

   In your list, be as detailed as possible. For example for the *Source of Power*, include where the power is coming from (i.e. plugs into the wall *or* batteries *or* USB cable *or* …), and what voltages and maximum currents are needed. Then try to select specific components.

   Note that the UI hardware and motor have already been chosen by the Cat Feeder's mechanical designer. For this you get a small display; three buttons; plus a stepper motor and optical homing sensor.

6. Now that you have given your design a lot of thought, you could optionally draw a block diagram of your controller. This is fabulous content for your portfolio page due at the semester's end. No need for fancy graphics. A neatish hand-drawn sketch is fine. Potential employers love seeing this kind of stuff as it gives insight into your thought process.

7. Now think about what the software needs to do. Start by reviewing your lists above to ensure everything's included. I tend to break the software into two groups: What needs to happen under-the-hood and the UI. For Under-the-hood I might outline the low-level functions that interface with the hardware. For our Thermostat, it could be something like this:
   - GetCurrentTemperature()
   - TurnHeaterOnOff()
   - TurnAcOnOff()
   - SaveTempSetting()
   - LoadTempSetting()
   - DisplayCurrentTemp()
   - ReadModeButton()
   - ReadUpButton()
   - ReadDownButton()
   - MainLoopToControlTemp()

   For the UI, I give a lot of thought to exactly how the buttons and display will be used. Doing a good job here is the difference between Apple and Microsoft. Rather than making a bullet list, I prefer something more like a Story Board, thinking in terms of *modes* the software will have. The 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:

- Between now and the end of the semester, you have a ton of work to do. You're much more likely to get everything done if you make a detailed schedule showing all the work, and when each part should be finished.
- Start by making a list of everything that you need to do. This list is just for you, you will not be turning it in. Be sure to think about: Overall design; Choosing components; Designing the schematic and PCB; Ordering parts; Programming; Assembling; Testing; Revising; Reordering…

  You will likely be building two versions of your controller board. The first one should be your best effort at completing the entire project perfectly. The second version is just to fix your little mistakes. Keep in mind that it takes about a week and a half to get your bare PCBs made and delivered to you.
- Now create a calendar showing the start and completion dates for everything on your list. Make it in the form of a traditional calendar using a table, showing *Sunday, Monday, Tuesday*… along the top; and one box for each day between now and the semester's end. Your project and portfolio content are due on the last "class day" that you and I meet.
- Feel free to add anything to your calendar that might interfere with your progress (such as Thanksgiving or assignments in other classes).
- To be successful, you will need to be working on many things, all at the same time. If you plan on finishing one task before starting the next, you will likely not get everything done.
- Save your calendar as a PDF, name it: *Cat Feeder – Project Schedule - <your name>*. Lastly, email it to me (stanley.reifel@olin.edu).