Status Update: Phase 2

Team: Sun Emojis
Members: Jake Mingolla, Melissa Blotner, Reema Al-Marzoog, Eliza Schreibman

Work completed:

- IOS app
  - Reema and Melissa
- Countdown module (bean sketch)
  - Eliza
- Screen (bean sketch)
  - Jake

How does the work you got done compare to the goals you outlined in your Project Proposal for this phase:

- Goals that were met
  - IOS app receives and displays UV information, gets notifications when the timer has run out, and can reset the timer at any point
  - Screen displays a “timer” (a bar that fills up as time goes on) and changes to an alert screen (dancing suns) when the timer fills up, indicating the user should reapply sunscreen
  - Hardware button also resets timer
  - Added haptic feedback when the user needs to reapply sunscreen
  - The app sends alert notifications to the user when it is time to reapply.
  - Disconnect from the iOS main graph screen to support interactions with multiple beans

Struggles:

We were struggling to get the button to work but once we added a 1 mega ohm resistor to the grounded side of the button helps the floating values read on the bean. Before the resistor was added the button was very temperamental and would reset under the slightest touch to the wires. Rarely the timer is still resetting without being pushed which we think has to do with the button. Hopefully once we shift to the PCB it will be solved, if not we can figure out in software how to account for the button.

We struggled with adding the Charts Cocoapod to the project during iOS development.

How does your code interface with your hardware? (brief summary so that we can have a sense of your overall architecture; what is different pieces of code is driving?

- The arduino sketch main loop accomplishes four things

  - Read the current UV data, convert it to UV index, and send every 100th reading to the app (heuristically set, can be changed in order to minimize throughput to the bean) to the iOS app to display on the graph
  - Refresh the screen with the appropriate timer bar (filling up as time progresses)
  - Check to see if the reset button (hardware or IOS) was pressed
  - Check if the time has been exceeded given the current UV reading (and so it is time for the user reapply sunscreen and reset the timer via the hardware button or IOS button). In this case, the bean sends a message to the iOS app to send an alert notification

Testing:

- We tested our arduino code without the need for the hardware or the IOS app by using the serial terminal (running Chris' sonicmac application) to read serial output and set input, simulating IOS app sending information to the arduino code.
- We tested the IOS's ability to read and graph UV data by simulating UV readings in a simple arduino program (basically just a loop that sent fake values to the app).
- We tested our timer with the app to make sure the logic of the reset was working before we got it working with the hardware as well.

Directions:

- To use the breadboard, just put a battery in the bean or connect a battery to the breadboard adapter. You can also open a serial terminal to see debugging output and the UV information.
- If you're using a Bean that doesn't already have the sketch loaded onto it, load the sketch to the Bean.
- To test the app, have it downloaded onto your phone and (within the app) bluetooth pair it with the bean.
- To test it, expose the breadboard to different levels of sunlight and notice changes in the UV values measured. The screen should also be displaying sample information.
- The screen timer can be reset by pressing the yellow button or through the app.
- You will need to download the U8GLib in order to use the screen.
- https://github.com/olikraus/u8glib