Programming Assignment 2 – Upgraded Weather Station (w/ Sensors)

Assigned/Due: See online for exact due date (11:59pm)

YOU MAY WORK WITH OR WRITE CODE TOGETHER WITH YOUR PARTNER ON THIS PROJECT.

Functional Requirements

In this project, you will push the boundaries of your knowledge to add new features based on some functionality we haven't quite covered in class (but should be within reach with a bit of research).

The below requirements must be met to receive full credit (see **Grading Breakdown** section at the end), but you are given considerable freedom and flexibility to implement these features however you see fit; that is, you don't need to write code in a certain way and are generally free to design your screens as you feel best, with minimal exceptions. Extend your Weather Station program **from Project 1** to include ALL of the following:

1) **SHT40/VCNL4040 Integration:** The ability to display sensor data for both Temperature and Humidity from the SHT40 (over the I2C interface).
   a) Both the SHT40 (Temperature/Humidity) and the VCNL4040 (Proximity/Light) must work simultaneously as they will both be used to drive functionality in the program.
2) **Temperature & Humidity Sensor Integration**: Create a new page that contains live readouts from the temperature and humidity sensors; your screen should have the following properties
   a) It should be its own screen (so, a third screen, in addition to the OpenWeather and Zipcode screens)
   b) It should indicate somewhere you are seeing live readouts
   c) Should update every 5s
      i) HINT: Only redraw if the temperature or humidity change from the last readout
      ii) NOTE: This is more frequent than you'd really want for battery reasons, but it'll help us gauge its functionality (e.g., if you hold your finger over the sensor or perhaps change rooms or go outside, it should likely change temperature readings)
   d) It should prominently display the local sensor reading for temperature
   e) It should prominently display the local sensor reading for humidity
   f) Temperature conversions and timestamps (added in P1) should also work for data on this screen)
3) **Proximity Sensor Integration:** Automatic control of LCD power based on proximity to sensor
   a) NOTE: This will simulate turning off a phone screen when it gets pressed to the cheek; use your discretion for what proximity threshold to enable/disable LCD
   b) Turn the LCD off when the VCNL4040 sensor is too close to an object
   c) Turn the LCD on when the VCNL4040 sensor gains distance from an object
4) **Light Sensor Integration**: Automatic dimming and brightening of LCD based on light sensor
   a) As ambient light increases, brighten the screen
   b) As ambient light decreases (i.e., it gets darker in the room), dim the screen

Grading Breakdown

Grading will be broken down as follows:

- **10% - Turned in all source code**
- **90% - Video demonstration**
    - **30% - SHT40/VCNL4040 Integration**
        - 10% out of 30% - Displayed sensor data using the Adafruit libraries
        - 30% out of 30% - Displayed sensor data using your own Arduino I2C library calls like we did for the VCNL4040 in class (i.e., **NO ADAFRUIT LIBRARIES**)
            - 10% penalty for using VCNL4040 Adafruit library
            - 10% penalty for using SHT40 Adafruit library
    - **30% - Local Weather Screen**
        - 10% - Screen shown and has:
            - clear language to indicate "Local" sensor/weather readings
            - placeholder for temperature
            - placeholder for humidity
        - 5% - Demo temperature display is reading live sensor data
        - 5% - Demo humidity display is reading live sensor data
        - 5% - Demo temperature conversion from F➔C➔F w/ the push of a button
        - 5% - Demo timestamp on screen showing time of last sensor reading
    - **10% - LCD Screen Power via Proximity Sensor**
        - 5% - Screen turns off when in close proximity to object
        - 5% - Screen turns/remains on when NOT in close proximity to object
    - **10% - LCD Screen Dimming via Ambient Light Sensor**
        - 5% - Screen dims (darkens) when ambient light decreases
        - 5% - Screen brightness increases when ambient light increases
    - **10% - Showed and discussed key code portions, explaining how you implemented each of the functional requirements**

NOTE: Points can and WILL be deducted at any point for performance issues (e.g., response is too slow, there is a noticeable lag between pressing the button and the screen updating with the result).

Submission Instructions

1.) Submit to Blackboard the following:
   a. **Video recording** (preferably YouTube link as a submission comment) demonstrating the functionality of your program and source code explanation:
      1. Make sure to attempt to show the local temperature/humidity sensor working (e.g., place your device in the fridge for a bit)
      ii. Make sure to cover ALL of the areas receiving points in the Grading Breakdown section above
      iii. Your video should be a 1-3 minute demo with 3-5 minutes of code highlights (no need to go into excessive detail, although you won't be penalized if you do)
      iv. Make sure there is sound
      v. Phone recordings are fine, just make sure the video clearly shows the LCD screen and what is happening with button presses
         1. If the instructor cannot see your screen clearly, you will not get any points for demoing features
   b. **Source files**, including:
      i. All **.cpp files**
      ii. All **.h files**
      iii. Your **platform.ini file**