University of Southern California
ITP 348 - Making Smart Devices: Introduction to Electronics/Wearables
Spring 2023

FINAL PROJECT:
DEVELOPER DOCUMENTATION

Menna Elamroussy
elamrous@usc.edu

TABLE OF CONTENTS

TABLE OF CONTENTS	**2**
OVERVIEW	**3**
WIRING DIAGRAM AND COMPONENTS	**4**
DEVICE SETUP INSTRUCTIONS	**5**
SUMMARY OF CODE	**6**
CLOUD VARIABLES AND DASHBOARD	**8**
PHOTOS	**9**

Author's Notes

Hello, thank you for your interest in Robot Whisk. We are always happy to hear different ideas on how to modify and improve our whisk. Please use the following guide to understand more about the code and setup of the whisk.

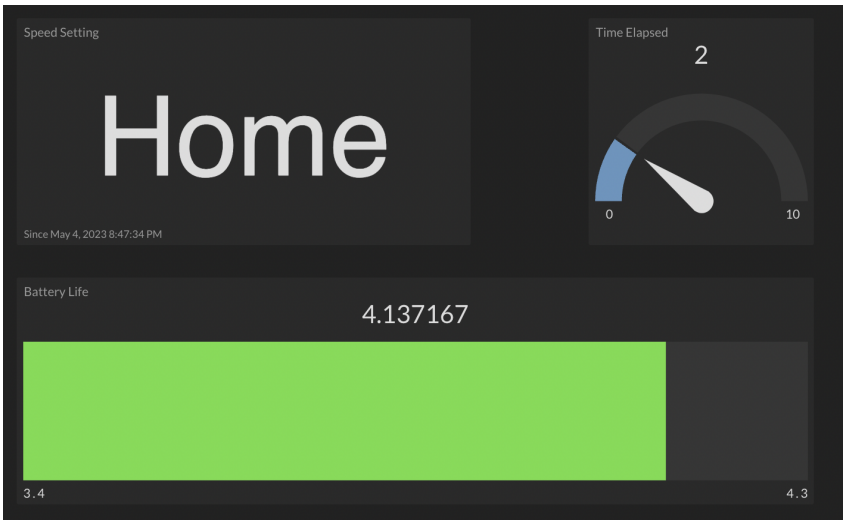
OVERVIEW

The Robot Whisk is a bluetooth controlled whisk that spins on different settings depending on what the user chooses. It uses a motor to spin the whisk and an OLED screen to display to the user what setting they have chosen and how much time has elapsed. It also uses a piezo buzzer to play a tune once the 10 seconds have elapsed and a potentiometer to allow the user to adjust the speed of the whisk, should they desire.

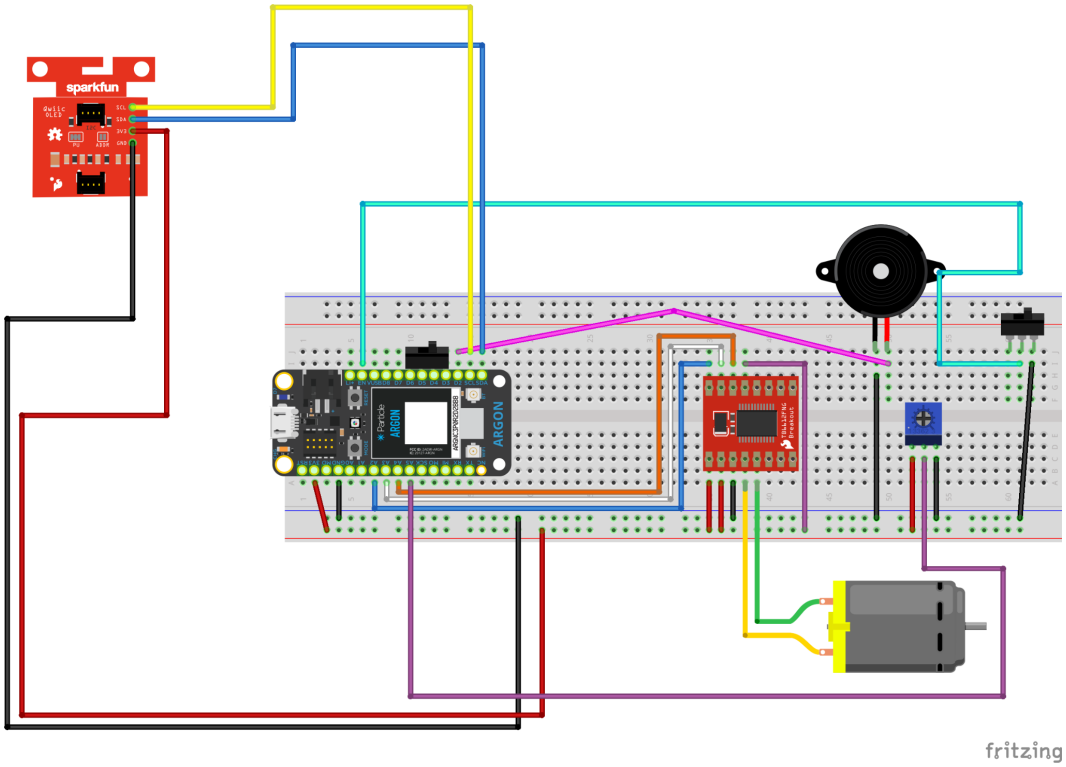
The **BlueFruit Connect** app is being used to allow the user to switch between the following settings: low, medium, high, pulse, or custom.

Link to BlueFruit Connect: https://apps.apple.com/us/app/bluefruit-connect/id830125974

The user can also view further information on the **initial state dashboard** such as current setting, time elapsed and battery life.



WIRING DIAGRAM AND COMPONENTS



The above wiring diagram was used as the basis for wiring the Robot Whisk however please note that in the implementation, different positions and different colored wires may have been used.

I also added a LiPo battery so that the whisk could run without needing to be connected to a laptop but that is optional.

Component	Purpose
DC Motor	Allow the whisk to spin
OLED screen	Display helpful information to the user
Potentiometer	Allow the user to change the speed of the whisk
Piezo Buzzer	Play a tune when the 10 seconds have elapsed
Switch	Turn the whole project on and off

DEVICE SETUP INSTRUCTIONS

- Wire all the components using the wiring diagram
- Open Visual Studio Code and using the command palette choose “configure project for device” and choose your device
    - I used version deviceOS@5.0.1
- Cloud flash the project using the command platte
- Once the project has flashed and is breathing cyan, you should see a whisk bitmap display on the OLED screen
- Open up the BlueFruit Connect app and choose the Argon device
- Open the control pad and use the following guide

Low speed

Medium Speed

High Speed

Pulse

Custom Speed

- Settings 1 to 4 are press once buttons meaning the user only has to press them once and the setting will turn on
- Depending on which setting the user chooses, the OLED screen will display the setting and the time elapsed.
- After 10 seconds, the piezo buzzer will play a tune to indicate that the 10 seconds have elapsed, a message on the OLED will pop up saying “food ready” before going back to the home screen.

SUMMARY OF CODE

The code is split up into the following sections (labeled in the code).

Section 1: Variables and Files

- Variable names are defined
- An enum was used to represent the different settings (home, low, medium, high, pulse, custom)

Section 2: Piezo Buzzer Tune

- Code was derived from https://musescore.com/user/16403456/scores/4984153
- The notes for the tune are written in an array called `int melody[]`
- The `playSong()` function plays the tune of the Mii channel theme

Section 3: Motor Functions

- These functions `digitalWrite()` the AIN1 and AIN2 wires of the motor depending on whether the motor should be on or off. They also `analogWrite()` different speeds to the PWMA motor pin depending on the setting chosen by the user
- They consist of:
    - `stopWhisk()`
    - `lowSpeed()`
    - `mediumSpeed()`
    - `highSpeed()`
    - `customSpeed()`
    - `pulse()`

Section 4: Setup

- Here is where all the initial code is setup such as `pinMode()` and initializing the OLED

Section 5: OLED Functions

- These are the functions that display different things on the screen.
- The consist of:
    - `displaySpeed(String Speed)`
    - `displayDone()`
    - `displayHomeScreen()`

Section 6: Receiving Bluetooth Input

- This is where the bluetooth input is received the the enum state is changed based on which button the user pressed

Section 7: Main Loop

- The main loop uses a `millis()` timer to keep track of how long the user has been in each setting.

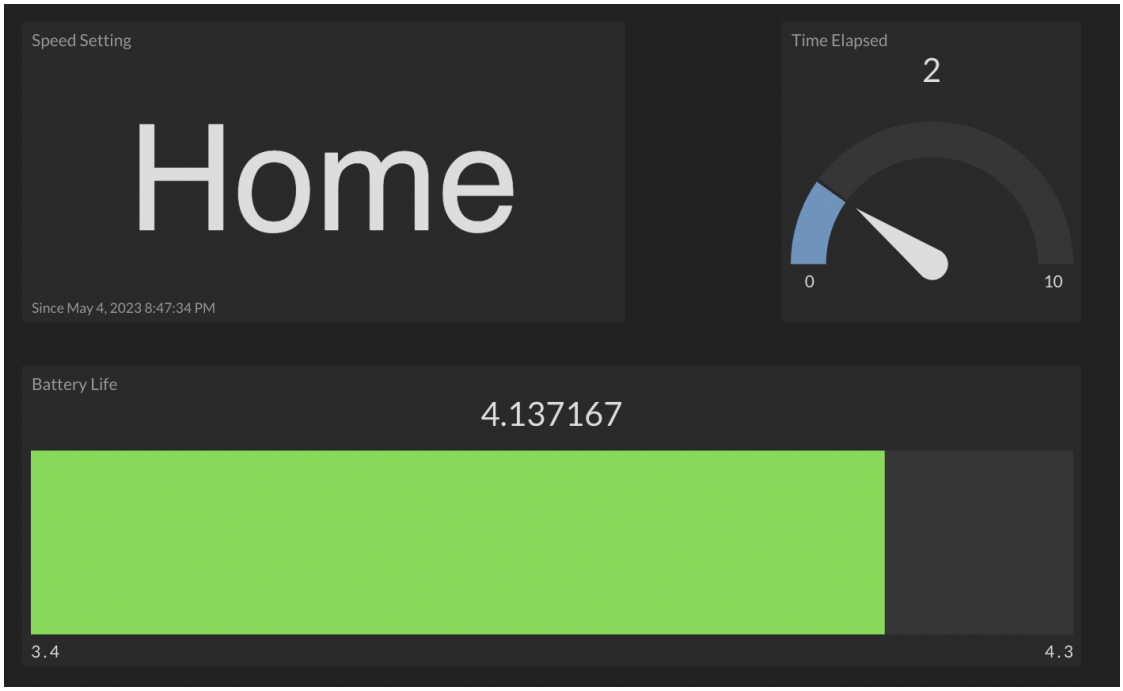
- It uses `switch` statements to switch between enum states and call the different motor and OLED functions depending on which setting was chosen.
- It also publishes the data to the InitialState dashboard using `Particle.publish()`.

CLOUD VARIABLES AND DASHBOARD

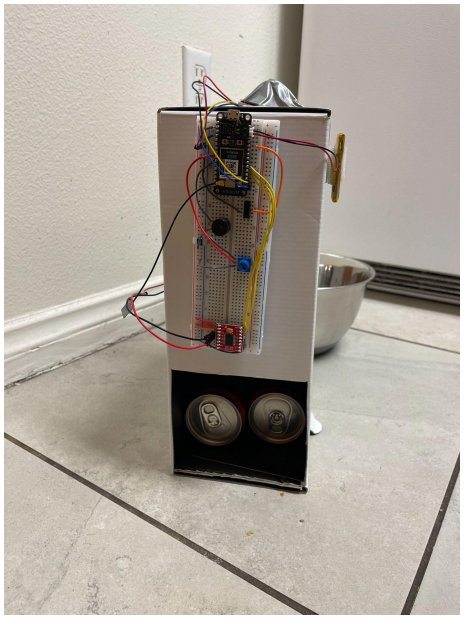
This project utilized webhooks to publish data to the cloud and to the InitialState dashboard. This code is written in the “Section 7: Main Loop” portion of the code. It uses a `millis()` timer to publish code every 1000 milliseconds or 1 second.

Link to InitialState:

Event Name	Details
FinalProjectSetting	Displays which setting the user has chosen from home, low, medium, high, pulse and custom.
FinalProjectTimer	Displays how much time has elapsed out of the 10 seconds if the user chooses settings 1 to 4.
FinalProjectBattery	Displays the voltage of the LiPo battery so that the user knows when it needs to be charged. (A description of voltage ranges is given in the user documentation)



PHOTOS



Thank you for reading our developer documentation. Please reach out with any questions.