**Detailed Design**

## **Introduction:**

This document will focus entirely on the design of our product beginning with our proof of concept which covers the interface between the Arduino Pro Mini microcontroller and the motion tracking device as well as the Bluetooth module. Next, we will dive into the overall design of our product which covers the smartphone application, data processing, as well as the hardware integration of the earbud and clipper units into the rest of our project.

## **Proof of Concept:**

Hardware Components Interface:

The Arduino Pro Mini microcontroller interfaces with the motion tracking device over I2C and the Bluetooth module through UART communication. The microcontroller was programmed with Arduino IDE so that it can run a script that constantly reads digital values from the motion tracking device using I2C communication. It then communicates over serial communication from the master bluetooth device off of the computers COM. Essentially, we uploaded a script to the microcontroller which is always running and always collecting data from the motion tracking device which then sends the data over to your computer through the Bluetooth module. Ultimately, our goal would be to transfer the data over to a robotic arm that could execute the clipper’s movements instead of needing a person to do it. This would greatly reduce the margin of error stemming from natural human errors.

Hardware Components in Figure 1:

* 1 x Arduino Pro Mini microcontroller
* 1x HC-06 Bluetooth module
  + Wireless serial communication
* 1 x MPU-9250 motion tracking device
  + Equipped with an accelerometer, a gyroscope, and a magnetometer
* 1 x 9 Volt battery
* 1 x 9 Volt battery clip connector
* 14 x Male-to-male jumper wires
* 1 x Breadboard

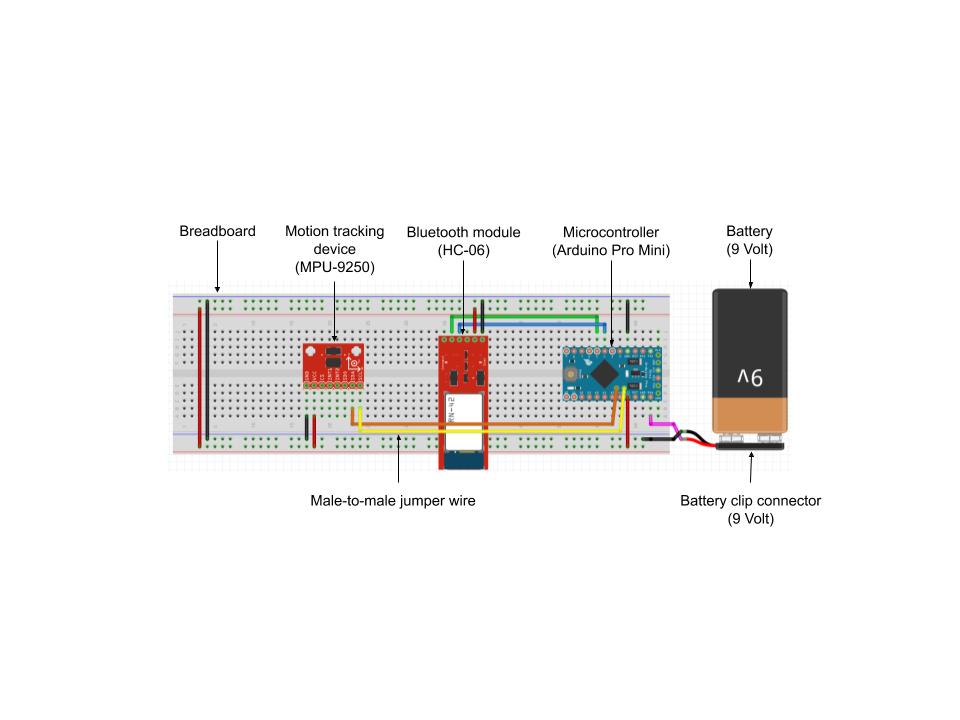


Figure 1: Proof of Concept Simulated Model Circuit Diagram

Wires Description:

* Black wires: ground
* Red wires: 5 Volt off Arduino Pro mini to power motion tracking device and BT module
* Pink wire: 2 x 3.7 Volt 18650 batteries connected in series (7.4 Volt) feeding to raw port on Arduino Pro mini
* Green wire: RX line for UART communication between Arduino Pro Mini and HC-06 Bluetooth module
* Blue wire: TX line for UART communication between Arduino Pro Mini and HC-06 Bluetooth module
* Orange wire: SDA line for I2C communication to collect sensor data
* Yellow wire: SCL line for I2C communication to collect sensor data

Software:

* Programming languages:
  + Arduino IDE: C,C++, and Java (for Arduino interface with hardware components)
  + PyCharm IDE: Python and Java (for data processing)
* Post processing software:
  + Turning a(t) linear acceleration values obtained from the accelerometer within the MPU-9250 motion tracking device into x(t) position values

Platform:

* Arduino Pro Mini microcontroller board based on ATmega328 which is a Microchip 8-bit AVR RISC-based microcontroller

**Overall Design:**

### Hardware:

## The hardware block diagram below in Figure 2 demonstrates the hardware interface between both the hairclipping and earbud unit including their respective subcomponents, the user’s smartphone, and the cloud server. It is important to note that for this project to work, two separate tracking units are required; the first tracking unit is the hairclipping unit used to track the clipper’s movements, and the other being the earbud unit used to simultaneously track the user’s head movements.

## 

## Figure 2: Hardware Block Diagram

## 

## Software:

The software design diagram below in Figure 3 demonstrates how our team is planning on designing the software flowchart for the smartphone application, the software for data processing, and the collection of data from both motion tracking devices.

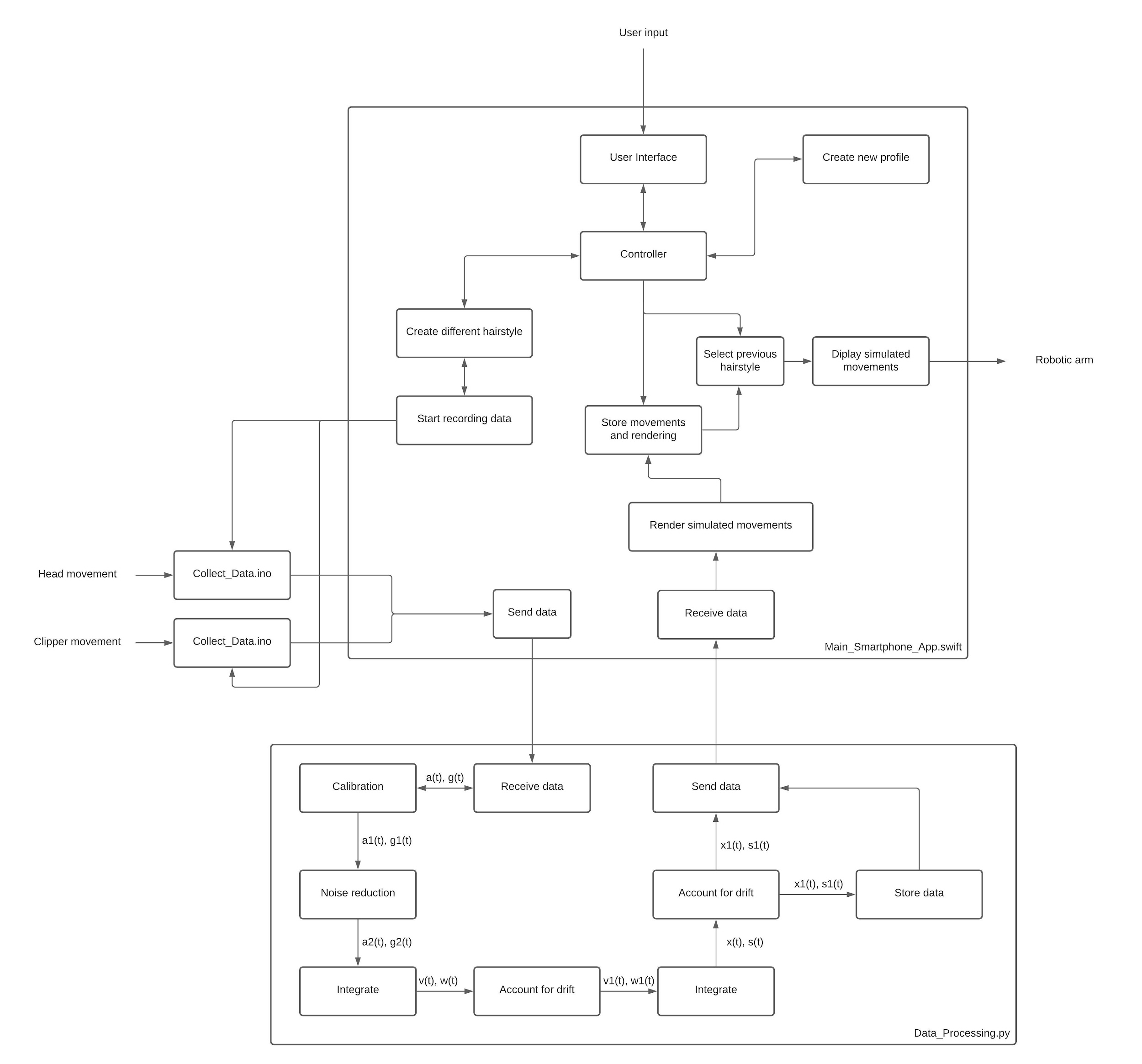


Figure 3: Software Design Diagram

Main\_Smartphone\_App.swift:

Along with our product, we are planning on designing a mobile application so that users can interface with the clippers through their smartphone. Users will be able to create a profile and save different hairstyles directly on the mobile application. Furthermore, the electric clipper would be able to measure its position in 3D space over time and send the data to the user’s smartphone via Bluetooth. The user would be able to press start on the app, power up the electric clipper, and obtain the movement data obtained from both their head and the hair clipper. From there, the recorded data will be sent to a cloud server to undergo post processing, save it, and send the data back to the user to be displayed.

Data\_Processing.py

Once the motion tracking device has gathered acceleration data, a(t), using its accelerometer sensor, we want to turn these values into positional data, x(t), while accounting for the data it over collected through noise. Thus, we have to perform noise reduction to smooth out our data as much as possible as there is always some noise that appears in a form of jitters which appears randomly throughout the signal. To do this, we apply a moving average which basically goes through the signal from left to right and sets each data point equal to the average of its nearby data points. We also need to pass the signal through a low pass filter that eliminates most high frequencies which generally represent the noise.