# APPROACH

This document details the team’s approach to the design. The design will consist of the golf glove and wrist-mounted microcontroller subsystems. The glove subsystem will include linear stretch sensors and an accelerometer. Another accelerometer will be mounted on the controller. The controller will also contain a rechargeable battery. An electrical connector will pair the two subsystems. A desktop application will receive sensor data from the controller over Bluetooth and provide feedback to the user.

## Hardware

The Golf Glove is composed of several subsystems outlined in the Design Constraints section of this document. In this section, critical components of each subsystem are compared and selected.

* + 1. **Glove**

The glove subsystem consists of the garment itself, linear stretch sensors, flexible pressure sensors and electrical connector components. The garment will provide mounting for the sensors. The sensors will receive power and transfer data to the controller through the electrical connector mounted on the glove.

**3.1.1.1. Glove Garment**

The glove garment is the mounting point for all of the linear stretch sensors, an accelerometer, and pressure sensors. A connector will also be mounted on the glove to connect to the controller, making the glove and controller exchangeable. For the initial prototype, two sizes of gloves will be created to fit different golfer’s hand sizes comfortably. The material used will be standard leather.

**3.1.1.2. Linear Stretch Sensors**

Crucial to the design of the Golf Glove is the choice of a linear stretch sensor. These sensors vary linearly in resistance or capacitance as the sensor stretches. They typically consist of some elastic material fused with a flexible resistive material or capacitive shape. The conductive material is applied to the elastic material such that the electrical characteristic varies linearly with lateral extension.

Two classes of linear stretch sensors were considered: LiquidWire and StretchSense. The team chose to use the LiquidWire class of sensors because of their resistive variability, low cost, and the existing relationship between the team and the manufacturer. StretchSense sensors vary linearly in capacitance as the sensor is stretched, while LiquidWire sensors vary in resistance, which requires a simpler circuit to measure with a microcontroller.

**3.1.1.3. Flexible Pressure Sensors**

The force between the user’s hands and the gold glove must be recorded accurately. Therefore, the team has decided the employ the use of flexible sensors to detect the force exerted by the golfer’s hand on the golf club. In Table 3.1 you will see the comparisons between different flexible.

Table 3.1 Flexible Pressure Sensor Comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Diameter**  **(mm)** | **Price**  **(USD)** | **Resistance**  **Range**  **(Ohms)** | **Interface/Communication** |
| Interlink 402 | 15 | 7 | Infinite-250 | Analog |
| Velostat Conductive  Sheet | Custom | 3.95 | Variable | Analog |

The team decided on selecting the interlink 402 resistive touch sensor because of its ability to be quickly and easily integrated into the system. This flexible sensor can be bent in a multitude of different shapes and is made with tinned probes that allow for easy and seamless connection to the main system. At 15mm in diameter, the sensor can be easily wrapped around the thumb pad of the glove and be measured through a simple resistor divider circuit. In comparison the the velostat conductive sheet, the interlink 402 sensor does not need to be cut or measured for each glove and its resistance value can be programmed into the firmware as a constant. While this sensor may be more expensive than the other option, it requires little work to integrate and is why the team selected it as our sensor.

**3.1.1.4. Electrical Connector**

The electrical connection between the glove mounted sensors and the wrist mounted unit must be both robust, and a standard that allows for multiple data inputs and outputs. Rather than create a new proprietary connector, the team has decided to use available standards because of their ease of use and overall support. In Table 3.2 you see the comparisons of different possible standards of connectors.

Table 3.2. Standard Connection Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| **Standard** | **Number of**  **Connections** | **Reversible** | **Max Power**  **(A)** |
| USB-C | 12 | Yes | 5 |
| USB-Micro | 5 | No | 5 |
| Cat-5 | 8 | No | .63 |

The team has decided to utilize a standard USB-C connection between the glove mounted sensors and the wrist mounted unit. By utilizing USB-C the glove will have more than enough power throughput and enough pins to transfer data into the wrist mounted microcontroller. The reversibility was the deciding factor in the team’s choice because a reversible connector removes possible confusion and frustration that a user may experience with other standards of connectors.

* + 1. **Wrist-Mounted Controller**

The wrist mounted controller containsseveral components critical to collecting and transmitting data to the user’s mobile device. This section compares options the team considered for each of these components.

**3.1.2.1. Inertial Measurement Unit**

The Golf Glove requires two inertial measurement units (IMUs)—one in the wrist-mounted controller and one on the back of the hand itself. These IMUs will measure acceleration of the hand and the rotation of the wrist. The LSM9DS1 IMU by STMicroelectronics was chosen by the team. The details that drove this decision are described in Table 3.3 and in the following paragraph below.

Table 3.3. Inertial Measurement Unit Feature Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **IMU** | **Cost (USD)** | **Power Draw (mA)** | **Accelerometer/Gyroscope** | **Magnetometer** | **Interface** |
| STM LSM9DS1 | $ 6.14 | 4.6 | ±16g linear acceleration,  ±2000dps angular rate | ±16 gauss | I2C / SPI |
| Bosch BMX055 | $ 7.02 | 0.13 mA | ±16g linear acceleration,  ±2000dps angular rate | ±2500 uT | I2C / SPI |

The team chose to use the LSM9DS1 because of its availability with North American electronic components vendors and its use on the Cypress development board for the CYW20719. This IMU is also sold through the electronics hobby site Adafruit, thus C libraries for the device are already widely tested and available.

**3.1.2.2. Microprocessor**

This section compares several microprocessors considered for the Golf Glove wrist-mounted controller. The Cypress CYW20719 was selected due to its integrated Bluetooth Low Energy capabilities and mature development community. Table 3.4 overviews the devices considered for use in the Golf Glove.

Table 3.4. Microprocessor Feature Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Microprocessor** | **Cost (USD)** | **Wireless Connectivity** | **Power Draw (mA)** | **Flash Memory (kB)** | **ADC** |
| Cypress CYW20719 | $ 9.63 | BR/EDR/BLE Bluetooth 5.0 @ 2 Mbps | 10 mA | 1024 | 1 x28-channel, 10-bit, 3.6 ksps |
| Microchip IS1871 | $ 2.98 | BLE Bluetooth 5.0 | 8 mA | 256 | 1 x6-channel |

The Cypress CYW20719 was chosen for the Golf Glove wrist-mounted controller because it contains a Bluetooth Low Energy-capable radio and an adequate number of ADCs on-chip. The Microchip processor is not available in North America during the time of this project and does not have a readily available development environment. Cypress provides WICED, a featured, free integrated development environment (IDE), for use with the microprocessor. The CYW20719 comes with a complete Bluetooth stack/driver on-board and Cypress has released several development and Bluetooth debugging tools specifically for use with this line of products. The team decided that the ease of use of the Cypress development environment best fit the development timeline of this project.

**3.1.2.3. Radio**

The Golf Glove will utilize the built-in Bluetooth 5.0 radio from the chosen microcontroller. The reasoning behind the team’s choice is grounded in ease of use. By using the build in radio system, the device will no longer need to interface with another, possibly inefficient, radio system. Another reason behind the choice is the size of the system, utilizing a built in radio decreases the overall device size and makes economic sense removing the need to buy and integrate another external radio system.

It was important to the team that the internal radio system utilized a low power system with moderate to high data output. The integrated 5.0 Bluetooth system includes many benefits over other systems. Bluetooth version 5.0 came alone with updates to data throughput and range. Compared to Bluetooth 4.2, version 5.0 doubled the data transfer from 1 Mbps to 2 Mbps, and quadrupled the maximum range from 200 feet to 800 feet, all while maintaining low energy usage.

**3.1.2.4. Temperature Sensor**

The Golf Glove will utilize a contact sensor embedded in the fabric of the glove. Table 3.5 shows the comparisons made when choosing a temperature sensor.

Table 3.5. Temperature Sensor Choices

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Component** | **Price**  **(USD)** | **Communication Interface** | **Sensor Type** | **Accuracy**  **best/worst**  **(℃)** | **Resolution**  **(bits)** | **Max Voltage**  **(V)** |
| Microchip  MCP9700-E | 0.28 | Serial | Contact | ±4/6 | 8 | 5.5 |
| Microchip  MCP9808 | 1.16 | I2C/SMBus | Contact | .5/1 | 12 | 5.5 |
| Microchip  MCP9805 | 0.88 | I2C/SMBus | Contact | ±1/±3 | 10 | 3.6 |

The temperature sensor will be used to calibrate the other sensors and adjust any values that are slightly temperature dependent. The Microchip MCP9700-E is our selected temperature sensor for the Golf Glove. The resolution and price of the sensor was one of the main deciding factors. It is unnecessary to select a temperature sensor with high resolution because of the application. The sensor must be able to detect the overall temperature of the glove and the MCP9700-E fits this purpose. The sensor can operate on a voltage as low as 2.3 V and an operating current of 0.006 mA. By running at such a low power level, the device can continue operating at a longer time. The team decided to focus on a low price sensor because the accuracy of the temperature readings is not mission-critical. †

† Temperatures around the glove may vary greatly due to friction, user heat, and other environmental variables. Temperature readings are used to provide a small amount of error correction that may be caused by temperature changes in the multiple sensors. Therefore, an expensive temperature sensor with high accuracy is unnecessary for the Golf Glove.

**3.1.2.5. Power Requirements**

The Golf Glove is a battery powered system that must remain operational for a minimum of 5 hours. The team calculated max power usages of each high power internal/external device to create an estimated minimum capacity required. The main devices the team considered are the I/O pins, the internal Bluetooth transmitter/receiver, and the dual IMUs that will be used to gather swing data. Each one of these devices’ max power draw must be taken into consideration when selecting battery capacity. The maximum power usage calculations over a 5 hour period are found in Table 3.6. By adding up the power usage of each device and adding another ~150 mAh as a buffer, we determined that a 500 mAh battery would be sufficient to consistently supply the Golf Glove with power for a 5 hour period. The most important considerations of battery selection for the Golf Glove are its ability to recharge and be a slim form factor.

Table 3.6. Power Usage Calculations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Device/Subsystem** | **Max Power**  **(mAh)** | **Quantity Needed** | **Hours of Use** | **Power Usage**  **(mAh)** |
| I/O Pins | 16 | 4 | 5 | 320 |
| Radio RX/TX | 11.5 | 1 | 5 | 57.5 |
| IMU | 4.6 | 2 | 5 | 46 |
| Temp. Sensor | 0.006 | 1 | 5 | .03 |

**Total Power Usage:** (I/O pins) + (Radio) + (IMUs**)** =320mAh + 57.5mAh + 46mAh + .03mAh = **423.53mAh**

Given the above requirements and allowing some margin for error, the battery should have a minimum capacity of 500 mAh. Table 3.7 compares a few options that meet this minimum capacity.

Table 3.7. Battery Choices

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Component**  **(Battery)** | **Price**  **(USD)** | **Capacity**  **(mAh)** | **Output Voltage**  **(V)** | **Form**  **Factor** | **Chemical Composition** | **Rechargeable** |
| LP503035 | 1.00 | 500 | 3.7 | Pouch Cell | Li-Ion Polymer | Yes |
| UxCell  AAA | 1.64 | 500 | 1.2 | AAA  Cylinder | Ni-MH | Yes |
| Duracell  AAA | 0.55 | 541 | 1.5 | AAA  Cylinder | Alkaline | No |
| Panasonic  CR3032 | 2.45 | 500 | 3.0 | Coin Cell | Lithium | No |

The LP503035 was selected as the power source for the Golf Glove. The battery incorporates recharging capabilities and contains a capacity sufficient to run the Golf Glove for an extended period of time. This battery is sufficient to drive the internal circuitry with an average output voltage of 3.7 V at a cost-effective price. Another important feature of the LP503035 is its thin form factor allowing for sleek containment inside the main wrist housing.

## Software

The microcontroller will collect each swing’s data from the four stretch sensors on the glove, the two IMUs, and the two pressure sensors. Following, the microcontroller packages this data and sends it over Bluetooth to a coaching application that can display it as either raw data or as a readable data graphic. The coaching application will store recorded swings in a local database, thereby allowing the user to review their previous swings at a later time.

* + 1. **Microcontroller Firmware**

Microcontroller firmware drives the sensor logic for the design, handling swing detection as well as sensor data acquisition. The main components of the firmware are described below. The firmware will poll over I2C for the IMUs and an ADC for linear stretch sensors. As for measurements, the two IMUs will each produce three gyrometer measurements and three accelerometer measurement; the two pairs of stretch sensors will measure two wrist angles; and the microcontroller will measure a single temperature reading. These sensor readings will have a resolution of 16 bits for each value.

The firmware will have two main modes of recording with the first mode detecting a swing using a set of threshold sensor values. The data will be recorded at a rate of 120 Hz to produce a high resolution reading of swing movement. A swing is completed once the accelerometer stops reading velocity changes and the hand pressure releases. While recording, data will be stored in the microcontroller flash memory. Once completed, a pointer to the swing memory will be added to a FIFO queue and then sent to the coaching application on the next event loop.

The second mode is real data transmission mode where sensor data is sent to the application as it is measured in real time. This mode will be useful for the user to see sensor data while practicing swing positioning, without the detection interfering with readings. The coaching software will receive a stream of sensor data as it is measured, and will be rate limited by having the microprocessor sleep for the duration required to achieve a target rate of 120hz.

**3.2.1.1. Implementation**

Microcontroller firmware will be developed in C. The software will consist of a main event loop that will sleep the microcontroller until the conditions for a swing are met. During mode one Swings will be identified by a sudden change in pressure as well as accompanying change in accelerometer velocity reading. Once a swing is detected, the microcontroller begins reading the sensor data into the device memory. While recording, the microcontroller is also checking for the swing end condition. Once the stop condition is met, the swing data will be sent to the coaching software.

In mode two, the swing detection logic will be disabled and data will be communicated to the coaching software immediately while the controller is in the “Detect a Swing” state. The controller software will poll for received Bluetooth data and check for a flag from the controller software. The flag will either disable or enable real-time reading, which will change the state logic for “Detect a Swing”.

**3.2.1.2. Software Flow Diagram**

The state flow diagram in Figure 3.1 models the three states of our microcontroller.

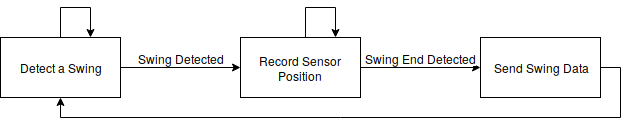


Figure 3.1. Firmware Stateflow Diagram

**3.2.1.3. Interfacing**

The controller will interface with the coaching software over Bluetooth. Once the controller receives power, it will begin broadcasting for a Bluetooth connection. Next, the coaching software will initiate a paired Bluetooth connection to the controller so that data transfer can begin. The connection status will be displayed on the coaching software to the user.

* + 1. **Coaching Software**

This section details the software approach for the visualization and coaching software application. The visualization software is responsible for receiving, storing, and processing the sensor measurements from the wrist-mounted microcontroller for display to the user. The application stores the swing data in a local database and queries the database to display the requested swing. Figure 3.2 provides an overview of this backend system.

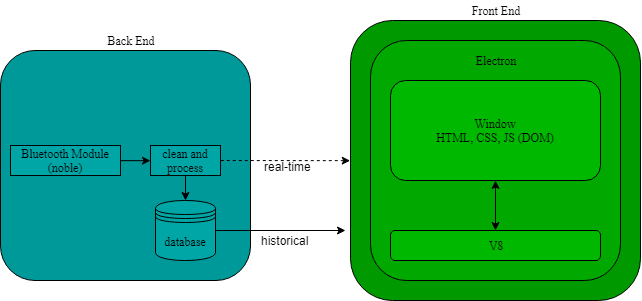


Figure 3.2. Computer Application Overview

**3.2.2.1. Implementation**

The coaching software will receive data from the microcontroller over BLE communication. The interfacing will rely on Node.js’s BLE library, noble [1]. Noble will handle all radio transmission overhead such as message length, error checking, and acknowledgements. Data will be received as a stream into a buffer. Once a new timestamp is read from the buffer, the buffer will be saved into a SQLite database as a frame and sent to the display. The display will be powered by Electron, an open-source, cross-platform, desktop application framework [2] that combines the Chromium browser rendering engine and Node.js. Using the React framework [3] inside Electron for front end rendering, the application will respond to both real-time display frames and whole swing display frames. Real-time display frames will be instant measurements containing one data point, while whole swing display frames will be a coagulation of measurements containing a snapshot of the users swing.

#### 3.2.2.2. Software Flow Diagram

Figure 3.3 shows how the desktop application will function.

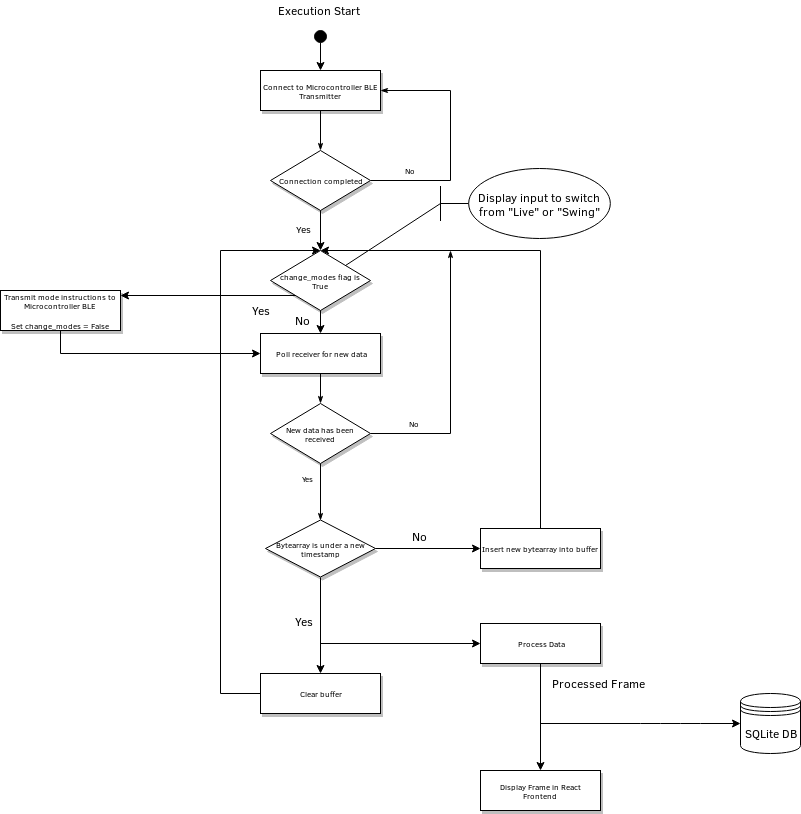


Figure 3.3. Desktop Application Data Flow

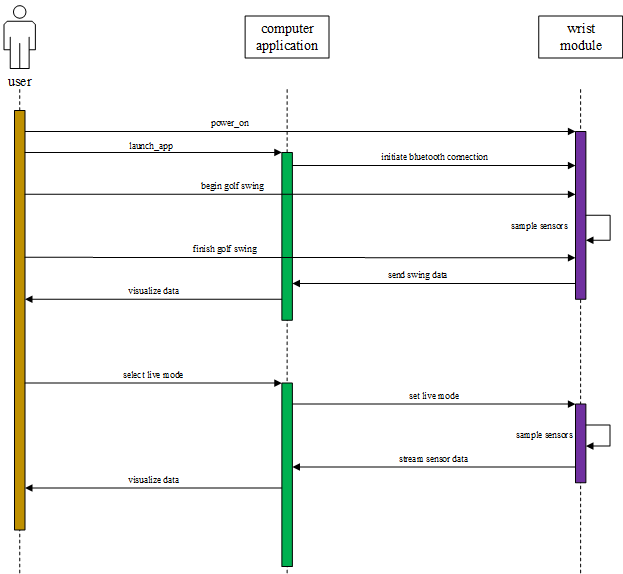
**3.2.2.3. Interfacing**

Node.js on the desktop will utilize the noble BLE library for interfacing to the microcontroller. It will first pair with the client, then always listen for incoming data. If the display mode is changed from “Live” to “Swing” or vice-versa, the Node.js application will transmit a message to the microcontroller before listening for new messages.

Assuming there are 20 bytes in a Bluetooth payload, the first byte will be the timestamp followed by 19 bytes of sensor data.

### 3.2.3. Use Cases

Figures 3.4 and 3.5 describe the sunny and rainy day use cases for Golf Glove, respectively.

Figure 3.4. Sunny Day Use Case

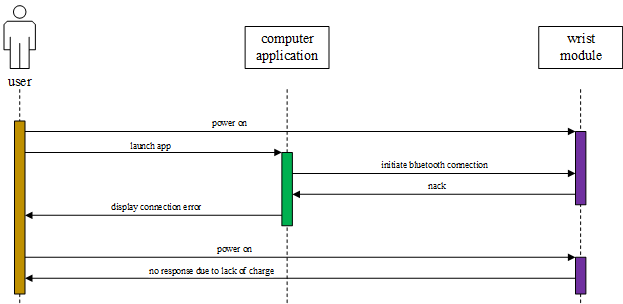


Figure 3.5. Rainy Day Use Case

## 

## References

[1] “noble/noble,” *GitHub*, 08-Jun-2018. [Online]. Available: https://github.com/noble/noble. [Accessed: 26-Sep-2018].

[2] “Electron,” *Electron*. [Online]. Available: https://electronjs.org/. [Accessed: 26-Sep-2018].

[3] “React – A JavaScript library for building user interfaces,” *React*. [Online]. Available: https://reactjs.org/. [Accessed: 26-Sep-2018].