Senior Design 2019

Deliverable: Software Design

Date: January 29th, 2019

Team: Notorious EMG

Team Members: Vi Tran

Software Architecture Block Diagram

The purpose of our software is to indicate when the microcontroller is looking for a device to pair with and indicate when it has successfully paired with a device. It then reads in an analog signal, in this case, the EMG signal, and converts it to a digital signal for controller to analyze. Using a conversion factor, it scaled it to the corresponding voltage value. The initial value of the incoming signal is saved, when the software reasons in a signal value different from the initial value, i.e the muscle signal has changed, it writes over the muscle signal variables and saves the previous value continuously. This data is then written to as SD card for storage and average maximum value and the potential difference between the average maximum value and average value is calculated to report power and muscle fatigue.

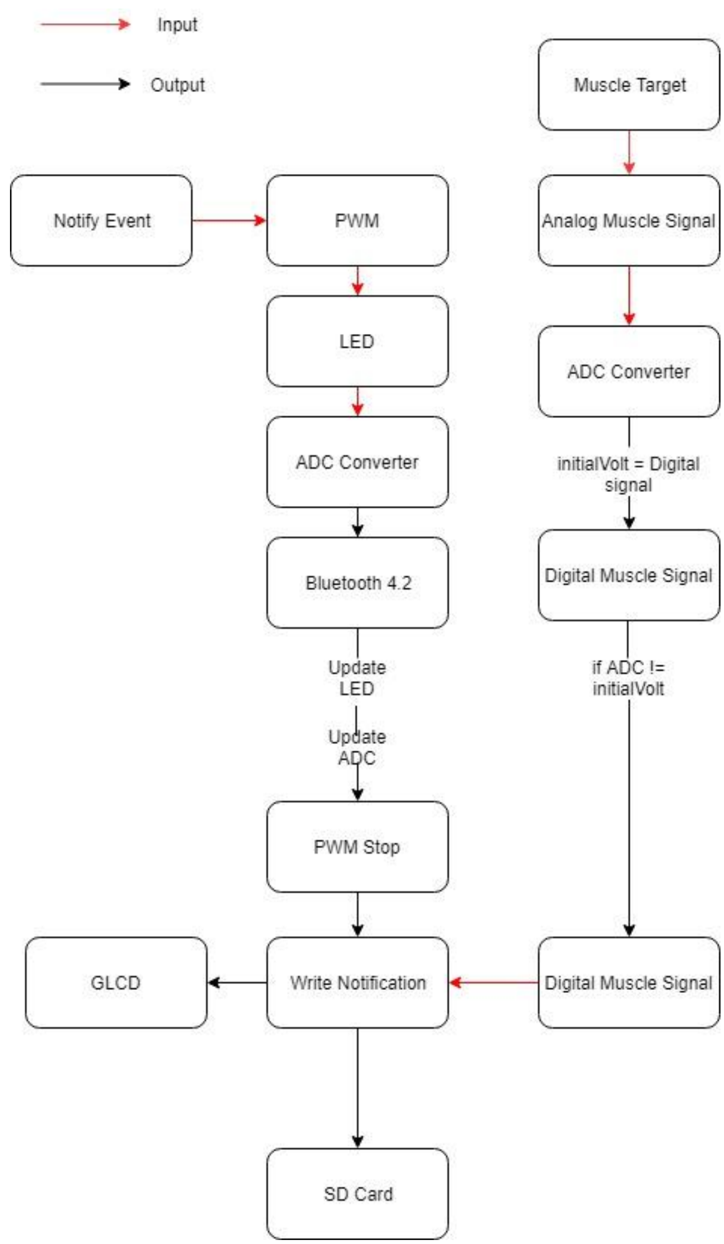

Figure 1 – Software Architecture Block Diagram

Software Structure Overview

The main software structure uses polling as its primary control. We are using polling because we want the BLE device to constantly advertise itself and check for surrounding devices until it is connected. GAP and GATT are the two most important concepts in understanding the Bluetooth system. GAP, or the Generic Access Profile, is how the devices connect, while GATT, or Generic Attribute Profile, is how the devices trade information. In this case, by using the GATT server, the database runs inside firmware and remembers the information for the client to read information. The GAP peripheral in this case will be the PSOC6 embedded BLE device and the GAP central will be the SD card we are writing to. GAP Central will talk to the GAP Peripheral. The GAP Peripheral will advertise that is it present and read to connect and the GAP Central will sense that it is around and make the connection. For this process, the software uses polling to check the new data, in this case a BLE connection, on a regular basis. The BLE device will be constantly advertising itself for a GAP central device to connect. The main program integrates the polls into the main loop to display a blinking LED when it is in advertising mode. Once the BLE device has successfully connect5ed to a GAP central device, the main software completes polling to indicate a connection.

Software Routine Diagram

The main software routines of the Muscle Guide are separated into two distinct systems. The first system is the EMG device which includes a pairing and ADC routine with LEDs to indicate the state of pairing of the device. The second system is the RTC box which includes again a pairing routine, LEDs to indicate the state of the paring device, a data write routine, and an additional LCD routine with an algorithm to calculate and display the EMG signal waveform, muscle fatigue, and power.

The pairing routine for the EMG device uses the Get State command to confirm a connection between the two devices. Once it has detected a connection, it enters the ADC routine to convert the analog EMG signal to a digital signal. Once connected, it also enters the LED routine that indicates the state of the EMG device. Initially the LED will be blinking using a PWM to indicate that it is looking for a device to pair with. Once the system has detected a connection, the PWM will be called to stop. This allows the LED to stop blinking to show the BLE device has been paired.

The pairing routine and LED routine is similar to the EMG device. The system of the RTC box has the addition on a write data routine which writes the Gatt data to an SD card once the devices are connected, while a GLCD routine is implemented to calculate and print muscle fatigue and power values, along with the EMG waveform. The algorithm to calculate the muscle fatigue and power values can be seen highlighted in pink, the “Results Algorithm”.

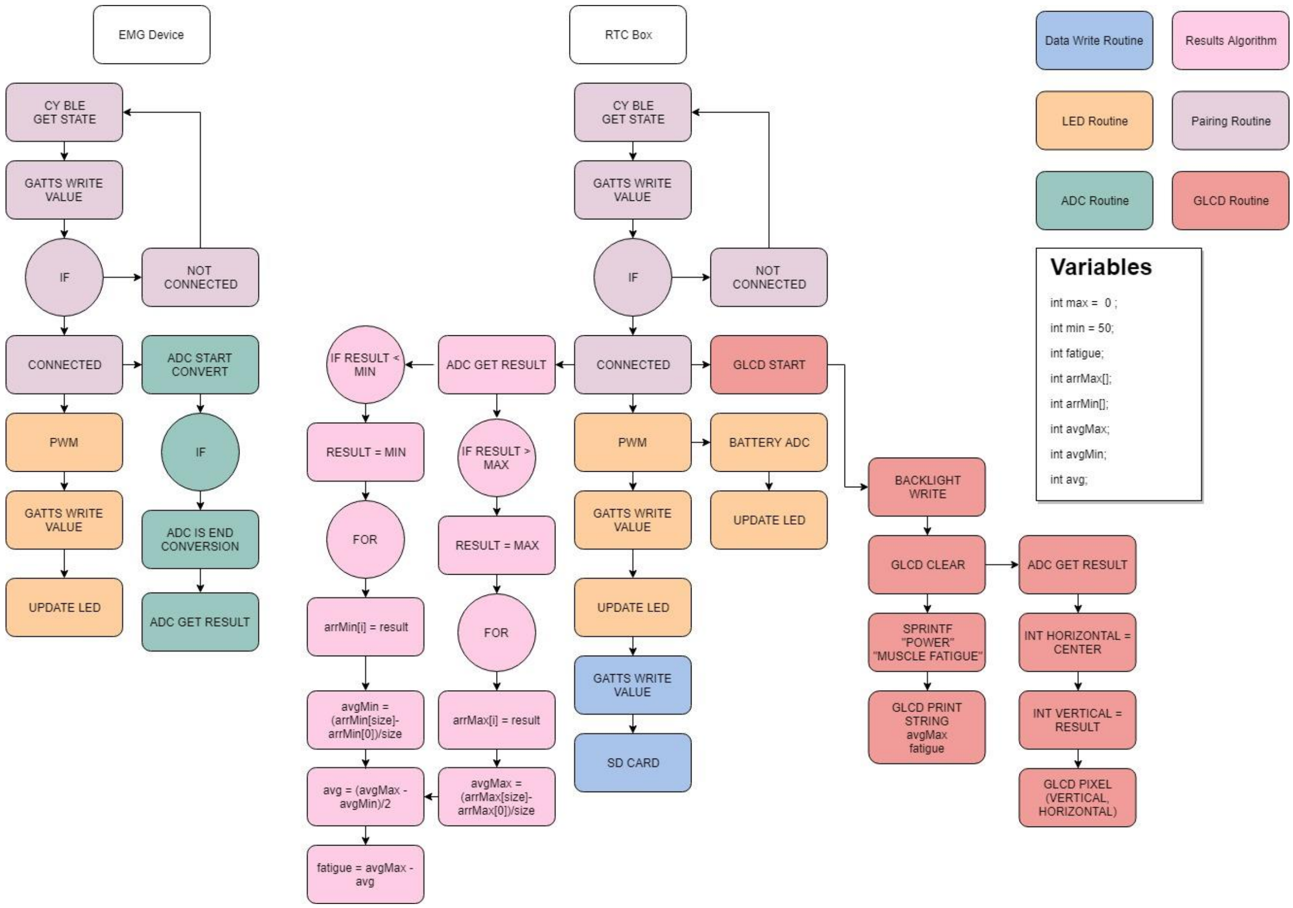

Figure 2 – Software Routine Flowchart