Threshold

Owen Duggan (oduggan@g.clemson.edu)

Ben J. Shinoski (bshinos@g.clemson.edu)

Randy Reed (rreed@g.clemson.edu)

Project Summary:

Threshold is an innovative wearable device designed to enhance workout performance by detecting muscle failure in real time using an EMG sensor. Equipped with LED indicators and a rest timer, Threshold provides instant feedback to help users optimize their training. By delivering precise, data-driven insights, the device notifies users when they reach their threshold, ensuring they maximize each set and fully utilize their muscle capacity. This results in more effective workouts, improved performance, and safer training sessions.

Project Description and Implementation

Changes:

Throughout the semester we realized we bit more than we could chew, so we had to change a few things. Originally we planned for a web application in order to see real-time tracking but due to our HM-10 BLE device, it made it difficult to connect to a port onto our laptop and we also had time constraints. It still does send a bluetooth signal out that is possible to read through an app called nrfConnect.

Another change we made was getting rid of the LilyPad button. We did this as it was another unnecessary component that we thought could be done through the button directly on the msp430. This led us to use just a simple button system instead of a button with multiple controls as well.

Software:

Our software is split into small easy to adjust drivers so that we could easily test each part of the board. The main.c file is the main structural component of our code, controlling the buttons responsible for measuring muscular failure and adjusting rest time. We use falling edges to properly monitor these buttons and call the ADC every 50ms for a fresh EMG sample.

We process signals captured by the myoware device through the P3.0 Pin using the built-in 12-bit mode, allowing us to read the voltage provided from the muscle on a scale from 0-4096. The signals provided by the myoware device are gathered to accurately measure the peak of each repetition in terms of electrical activity. Once a reading from the myoware reaches a certain threshold for a specified period of time, it is identified that we are in repetition, calculating the max output of this rep, and indicating when the rep is finished. We use these maxes to calculate an overall max which is used to compare local maxes to determine when muscular failure is achieved. Additionally, we attempt to track reptivie low voltage repetition which also indicates failure.

We utilize our setGreenLED() function and setRedLED() function to flip on and off the leds to indicate when in activity versus when in failure. These leds also determine the status of the LCD, which begins its countdown when the red led is activated.

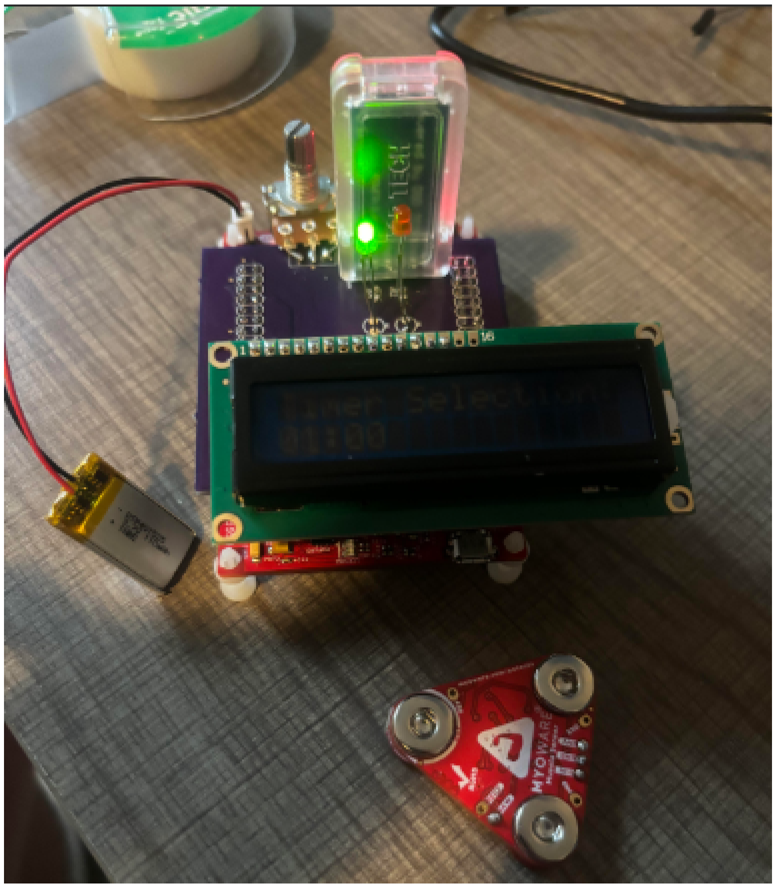
The HM-10 BLE module talks over UCA3. *bluetooth.c* brings up the clock at 8 MHz, sets 9600 baud, and exposes a uart_send_string() that main uses whenever failure hits. Any phone running nRF Connect can watch the packets.

We do delays two ways: a quick busy-wait delayMs() for setup calls that run at 1 MHz, and delayMsTimer() that scales with the active clock so our one-second rest countdown stays accurate even when we switch to 8 MHz for BLE. Both are in *timer.c*.

Hardware:

The hardware for our project stayed mainly the same throughout the semester as we worked towards making each individual component work. The MyoWare 2.0 EMG Sensor is the main piece of our project and reads electrical muscle activity. This is where the data that runs our embedded system comes from. The MSP430FR5994 Microcontroller handles all the signal processing, logic, and I/O. There are Red and Green LEDs that give the user visual feedback for muscle contraction/failure. It is green when you should be working out and red when you hit failure and the rest timer starts. The 16x2 LCD Screen outputs rest timer and failure status. After each failure it starts a countdown timer which is hardcoded for one minute. If time prevailed we would allow for the user to change the amount of time. We have a 3.7V LiPo Battery that powers the system, which is then stepped down to 3.6V in order to maintain constant voltage on our board. Another component is our custom PCB from OSH Park which is our consolidated circuit for sensor, power, and output connections. The HM-10 BLE is a lower energy bluetooth device that allows our device to send radio signals when failure is reached. Another component our device has is a 50K potentiometer to change the brightness of the backlight of the LCD. Without this component the LCD would be useless.

Prototype Pictures:



Challenges:

There were a lot of challenges that came with this project. The most challenging part of the project was getting the LCD to function properly. It took a lot of testing and it was hard to learn the library that controlled it. We originally did not also have the correct potentiometer to properly change the brightness of the backlight.

Another challenge we encountered was on how to actually connect our battery to our system. We planned to have a battery mount on our board which it would plug into, but after testing it seemed to not work. It could be that our soldering job was not great or our step down regulator was not allowing enough current through. It ended up not being a big problem as we can directly connect it to a msp430 pin.

The last main challenge we faced was the design of our board. We ended up waiting a long time to order our board because we kept stumbling upon new problems with each of our components. I am glad we did do that though because after we knew how to make sure each piece worked we were able to breadboard it then fully design our board with the traces going to the correct places. It took many revisions but our board almost fully works.

Team Coordination:

Our team coordinated pretty well as we all did a bit of everything. We communicated through a text group chat which we were all pretty active in. There were lots of times planning trades to give each other the hardware so we could work on it. In the end we ended up working it out pretty well. A github was used in order to push each of our changes and also let others review it. That is also where we stored all of our other documentation and notes as well.

Resources

One resource we used was the Unity C testing framework library. This framework allowed us to create and run C unit and integration tests with ease. We put the library within our tests folder and called upon necessary files when certain functions were needed. The library also came with an example file that we modified in order to fit our needs.

Other resources we used included the msp430 library which allowed us to easily call upon registers, pins and bits.

Class Feedback

1. What was the most challenging part of this class?

The most challenging part of the class for me, Ben, was getting fully caught up with everything that was going on. I did not have much experience with MCUs and did not have you as my OS professor. I was overwhelmed at first as everyone else seemed to know what they were doing. It was a topic I was so unfamiliar with but I am glad I took the time to truly learn more about it. Eventually I caught up and was able to understand everything along with the rest of the class but it definitely took some time.

The most challenging part of this class for me, Owen, was the learning curve specifically on the electrical engineering side. I had had no experience with any electrical work prior to this class, but I am definitely happy with what I was able to learn. I do wish there were materials posted onto the canvas as I am someone who learns better from reviewing something like slides as compared to lectures in class.

2. What would you like to see added or changed in future semesters?

A great addition to the class would be another makerspace work day earlier in the semester when all the teams had their individual components in. It would be great to play around with each component in the makerspace with the rest of our team and ask you questions easily if needed. It definitely would have saved us a lot of time and headaches.

3. What general comments do you have about the class?

This class was truly one I enjoyed a lot and definitely places within the top 3 classes I have taken at Clemson. It was a great class to offer as it gives computer science students the opportunity to learn more of the electrical engineering/hardware side. The hands-on learning in the class was also something I really enjoyed as we got to see how things worked first hand and mess around with stuff. I wish all classes were taught like this.