GYMTRON: THE AUTOMATIC WEIGHT TRAINING TRACKER

Mark Rumpel & Kyle Shaw

University of Regina: Electronic Systems Engineering

Abstract

Gymtron allows gym goers to seamlessly log their weight training activity by automatically recording metrics such as sets, reps, weight, and duration. In addition, the user is provided with real time feedback to ensure they are completing the exercise correctly. Once their workout is complete, users can view their data from any internet connected device by logging into a web interface.

Keywords: Fitness tracking; Weight training.

1. INTRODUCTION

In today's world, technology is integrated into almost every area of one's life, whether it is a smart phone, robots used in assembly lines, or other forms of everyday automation, there's no denying that technology is here to stay. However, one area significantly lacking automation is the health and fitness industry, specifically in workout tracking. That being said, there are many fitness apps and devices available that are able to track one's progress, but they all require some sort of user input to gather the necessary data. This is still rather cumbersome and time consuming so we saw a need for a way to automatically track ones exercises and make the results of those workouts viewable later. This need was the driver behind our initial goal for our design project, which was to automatically track a user's exercise with as little input from the user as possible. Through further research, we found there were more applications for a seamless tracking device than just personal fitness. When we spoke to Barclay Dahlstrom from the Kinesiology department, he recommended that such a device could be used to track rehabilitation patients as well as track individual participants in large research studies (personal communication, 2015). Thus, we found that there was indeed a need for a way to automatically track exercises.

To accomplish our goal of automatically tracking a user's exercise, there were several requirements we needed to meet. Firstly, we had to track the movement of the user which can be done by measuring the position of the weight stack on an exercise machine. Next, to ensure that each user is able to view their personal data, we needed to have a way for them to log into the machine using a unique identifier. Additionally, we also wanted to provide some visual feedback in the form of lights to guide the user's form as well as ensure that only reps completed using proper form are logged. Finally, we need to store all of the logged data and present it to the user via a personalized webpage.

2. METHODS AND MATERIALS

2.1 Design Process

With the project defined, we started on the process of designing a physical system that could meet our requirements. First, we had to define the exercise that we were going to track. There are many methods of weight training so in the interest of time we chose a specific method and exercise to focus on. We chose to use a weight stack style machine for performing tricep pushdowns. These machines are designed for the user to complete a specific exercise by lifting a weight stack using a system of pulleys. The movement of the weight stack always follows the same vertical path, and therefore, is easier to track than other weight training methods such as free weights.

Next we broke the project down into several independent functions that could be integrated together to form the desired weight training tracking system. This broke the problem up into smaller pieces, making the project easier to manage. We then brainstormed different methods to implement each function. With a number of options for implementing each function, we then evaluated each option based on its difficulty to implement, cost, ability to perform the function, and the ease in which it could be integrated with the rest of the system. During this process, some options had to be further researched to make an informed decision. After evaluating each alternative, we put the best solutions together into an integrated system.

We chose to use a Raspberry Pi to provide the function of a microcontroller to read sensors and control user feedback as well as a server to allow users to view information about their workout via a web interface. To track the user's reps we used an ultrasonic distance sensor. This sensor allowed us to track the position of the weight stack from which we inferred the user's movements. To track the weight the user has selected, we used a Bluetooth Low Energy module called a LightBlue Bean. This module includes a microcontroller that was used to read a unique resistance assigned to each weight in the stack. The user simply has to plug an audio cable into a receiver containing the unique resistance while they insert the weight selection pin. Their selection is then communicated to the Raspberry Pi via Bluetooth. To provide feedback to the user we used an RGB LED light strip that communicated the state of the machine using colour and brightness. To allow the user to quickly log in we used a Near Field Communication (NFC) module to read a unique identifier on a Mifare card. These cards are the same size and shape of standard ID cards such as a driver's license and enable the user to sign into the machine by simply swiping it past the NFC module. Finally, we built a user interface panel that included buttons for incrementing the current set and logging out of the machine. A timeout function for both of these buttons was also implemented in case a user forgets to logout or wants their set to be incremented automatically. This panel also included two LEDs to indicate whether or not the machine is currently in use.

2.2 System Integration

After choosing all of the individual components, we needed to come up with a plan to integrate them. This involved two main efforts: connecting the hardware together and writing the software to control the hardware. Throughout this process, we built each section of the system and tested it to ensure it worked as expected. Then we started piecing each portion of the system together by adding small sections to the system, testing the system to ensure it still operated correctly, fixing any problems that arose, and then adding in another portion of the system until it was complete. Following this process ensured that any problems that we ran into were easier to solve as we dealt with integration problems one by one. A block diagram of our final system is shown in Figure 1.

2.2.1 Hardware Integration

With all of the components selected for our system, we had to develop a strategy to physically connect them all together. The main component in the system is the Raspberry Pi, so this process consisted mainly of developing the necessary hardware to connect external devices to the Raspberry Pi. This process was complicated by varying operating voltages of our sensors, and the output current capabilities of the Raspberry Pi GPIO pins. Our general strategy for connecting these external devices to the Raspberry Pi was to design a circuit that would allow the Raspberry Pi to communicate/control the external element and then house all of these circuits on a single protoboard that served as an external interface to all sensors. This board was then connected to the Raspberry Pi via a ribbon cable to simplify the connection process. Before building any circuits on the protoboard we tested them in the lab using a breadboard to make sure they worked as expected. We chose to use a protoboard over designing a printed circuit board, as all of the circuits we required were simple to build. Based on our previous experience we felt confident that we could build them reliably on a protoboard. In addition, choosing to go with a protoboard simplified our schedule for the project, as we did not have to account for the time it takes for manufacturers to receive, build, and ship printed circuit boards.

2.2.2 Software Integration

We chose to use the C programming language to implement a program that could collect and store data into a database. We chose C because it is the same language that the external libraries we planned to use were written in, and it was a language that we both had previous experience using. To allow both of us to work on the code simultaneously and easily integrate it once complete, we decided to write functions for each external device that we had to interface to, and then call these functions from a main control program. We defined the required inputs and outputs of each function before starting so we could test our respective portions of the software without using our partner's complete code.

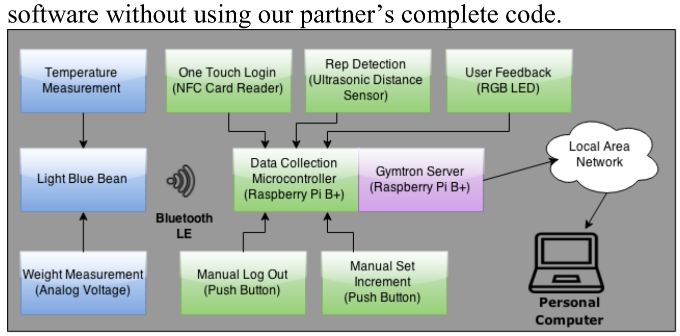

Figure 1: Gymtron Block Diagram

3. RESULTS

Overall, our design for the Gymtron did end up performing as expected and meeting all requirements specified. The following sections will elaborate on the specific results of each device and how they performed versus our expectations and overall in our system.

3.1 NFC Login

The NFC device was extremely reliable over the course of this project and never gave an erroneous reading. However, there were two occasions on which the NFC device was unable to be found and output an error message. This was a very rare occurrence though and was easily fixed by restarting the executable so we didn't consider it to be a significant problem. As far as outputs, the results of the NFC device were binary, in that it either read the correct unique identifier (UID) or it read an incorrect UID so the measure was essentially the ability to read the correct UID. Thus, since it read the correct UID almost every iteration, our results were as expected and excellent overall in terms of reliability.

3.2 Web Interface

For the webpage, it was required that the users were able to login, view their past workouts, and then select a workout and view the details for that workout. For the scope of this project it was decided that a very basic webpage would be built. With this in mind, the visual effects and design of the page were left very basic and more focus was put on the actual function of the page. The output of the website is again binary in that it will either work or not. In this case, we can measure the website by whether or not it performed all the necessary specifications; which were to allow the user to login, then view basic details for all workouts and then view details for any particular workout. We did this by creating a standard test case that can be performed and if all steps perform as expected, the original design requirements have been met. By this measure, the webpage performed as expected as met all requirements including some extra features such as only being able to access the login page if one is not logged in.

3.3 User Interface Panel

Originally, we did not have any user interface and relied on timeout periods to increment the set and log the user out. However, we found that relying on each user to have perfect timing was unreliable given that each user's workout can differ in tempo. Additionally, utilizing timeouts left the user with little control over the progress of their workout and resulted in user's blindly trusting the device's set timeout periods. Adding the buttons and LED's may have sacrificed some speed from the overall process but it greatly increased the accuracy of the results and made the experience much more intuitive for the user. The performance of the interface is once again measured in a standard test case, and both the buttons and the LED's performed as expected in testing. Furthermore, the timeout functionality of both the logout and increment set buttons performed as expected as a button event was simulated once the time limit was reached.

3.4 User Feedback

The LED strip worked as expected, and we received positive feedback from visitors during project day as to its ability to clearly communicate the status of the machine to the user. We also received a lot of feedback from interested parties that they would like to see a more detailed display to communicate information such as your current set/rep or the weight that you used last time. These comments suggest that we may have over simplified our communication with the user, and also highlighted that as engineers we have to ensure we take the needs of our clients into consideration instead of just what we think is best. In conclusion, the RGB light strip was effective in providing a simple feedback system to the user, but could have benefited from a supplementary display for more detailed information.

3.5 Rep Measurement

Our strategy for rep measurement met the requirements we set out to meet, and allowed us to successfully complete our standard test. We had some problems with the quality of the data that we received from the ultrasonic sensor, but after filtering the data, we were able to get reliable results. The particulars of the problems we experienced with the ultrasonic sensor are described in section 4.2. With the filter in place, we were able to measure every successful rep that we completed and only measured a rep after the user had completed the necessary actions.

3.6 Weight Measurement

The weight measurement system we designed worked better than expected. The only communication errors occurred when the LightBlue Bean ran out of battery, and our method of calculating the weight from a resistance worked every time. The resistors that we used had a tolerance of 10 percent so our method of resistance selection and weight calculation did not depend on exact resistance values, making it more robust and easy to manufacture. During battery failure, we implemented code that threw an error message, and then used default a temperature and set the weight to negative one (indication it was unavailable). This allowed the rest of the machine to continue operating normally. We also programed the LightBlue Bean in such a way as to limit battery consumption. This was successful as in four months of testing we only had to replace the battery once. To conclude, the weight measurement system was robust and supplied accurate information as required by our standard test case.

4. DISCUSSION

Overall, we can see from the results that we were indeed successful in meeting the requirements of our design. We were able to login into the machine and track the user's workout accurately while providing feedback to the user. We were also able to store the data in a database and then display each user's data back to them in a personalized fashion via the Gymtron webpage.

4.1 What Worked Well

One of the strengths of using the Raspberry Pi and C for our platform is the abundance of libraries and support available for devices. Having pre-built libraries for things such as GPIO, Bluetooth communication, MySQL interaction, and the NFC device were a great help during the design process and saved us a lot of time over the course of the project. This was especially true for the NFC device which was able to utilize libnfc, which is an open source library for various NFC applications. Libnfc provides basic start up code to help users get their device running initially; we used this code as a base for the login function we implemented for Gymtron. This helped the NFC device function very reliably as the base code being used has already been tested by other users and optimized. Overall, the libraries were very helpful in that they let us focus on our design rather than having to program all of the back end functions we would need to implement our design.

4.2 What Did Not Work Well

The portion of our design that we had the most problems with was the quality of the ultrasonic sensor data. The sensor occasionally provided erroneous data that had the effect of causing the RGB LED light strip to unpredictably change colour as well as produce the occasional phantom rep. In order to ensure reliable operation of the machine we had to remove this data using a filter. We tried both an averaging filter and a median filter. We found that the median filter worked the best. We had to limit the filter size to 3 samples for both the averaging and median filters, as larger sample sizes caused the program to unpredictably freeze. This occurred with a variety of delay times between samples. The program froze because the Raspberry Pi would suddenly use 100 percent of the system resources. We observed this using the task manager provided by the Raspberry Pi operating system. We did not have enough time to dig deeper into this problem, but suspect it had something to do with our management of the processor's time and the nature of the delay function in a multitasking system.

5. CONCLUSIONS

Overall, this project has been a great learning experience for us in both engineering design as well working in team environment. We have learned how important planning and communication are in major projects as well the importance of the testing process. Even though we were successful in meeting our specifications for this project, there are still some things we would do differently, as well as a few things we would like to implement in the future if we had more time.

5.1 What Would We Do Differently?

An aspect we would have changed for this design project is the formality of our testing process. Originally, we tested individual components as they were implemented and we really had no formal process for testing each part other than it performed as we were expecting in normal conditions, this disregarded special operation cases and left us exposed to bugs in uncommon cases. To test the entire system, we essentially just tested the machine as if we were an average user; this again left us exposed to bugs in the corner cases. We found this during our demonstration to the faculty, as we encountered a couple bugs we had never seen before. Afterwards, we decided to implement a formal testing case that would cover all cases, and allow us to document performance in each case. Doing this allowed us to find and fix several bugs to improve Gymtron's performance overall. In hindsight, we would have done this much earlier in the design process, which likely would have saved us a lot of time in the end.

5.2 What Would We Do if We Continue?

Lastly, we will discuss a few things that we would like to add to our project if we were to continue working on Gymtron after graduation. Firstly, we would implement a dedicated offsite server to store the data rather than using the Raspberry Pi as the webserver for each device. Doing this would be a much safer and scalable option as all the data could be stored in a single place, and backed up easily in addition to being housed in a safer environment for electronics (i.e. in a datacenter rather than a gym). Furthermore, we would also switch the method of data transfer to the database to Wi-Fi simply to reduce the amount of wiring needed. Next, we would start to implement different types of exercises and machines to make our product more marketable to a wider audience, rather than just having a single exercise available. In addition to more exercises, we would also add more analysis to the webpage. Examples of deeper analysis could be anything from using graphs and charts to display the user's data more visually, to benchmarking users to the population, or adding personalized machine settings to each profile to ensure the most accurate data possible is being collected by accounting for physical differences between users. Finally, we would also change the way we track the movement of the weight stack either by using a rotary encoder, which offers more accuracy than an ultrasonic sensor, or by using an accelerometer, which allows for more complex parameters to be measured but is also much more complex than the ultrasonic sensor. In conclusion, there are many future improvements that could be made to Gymtron, and we would have more than enough work ahead of us if we decide to continue working on the project.

Acknowledgements

We would like thank our project advisor Dr. Abdul Bais as well as Doug Wagner, Dr. Paul Laforge, and Dave Duguid for their guidance during the development of our project. In addition, we would also like to thank Dr. John Barden for sharing his knowledge on tracking athletes with accelerometers and Barclay Dahlstrom for lending his expertise in the area of weight training and providing some interesting applications for Gymtron. Finally, we would like to thank our classmates for their assistance in brainstorming solutions to some of the problems we ran into during the development of our project.