GYMplement

By

Srinija Kakumanu

Justin Naal

Daniel Rymut

Final Report for ECE 445, Senior Design, Spring 2021

TA: Alex Sirakides

05 May 2021

Project No. 26

Abstract

This report describes in detail our senior design project, GYMplement. This product is a pressure sensitive mat and companion app that allows users to characterize, count, and receive feedback on various bodyweight exercises. Our design is unique because it replaces the need for an individual to rely on another person for assessing their workouts. This allows for less experienced users to exercise individually in the comfort and safety of their own home, a need that arose due to COVID-19. Throughout the semester, we carried out several iterations of designing, implementing, and testing, and were able to deliver a successful and fully integrated project by the final demo date. This product was developed and tested with the guidance of the University of Illinois ECE 445 course staff and can be further developed for commercial applications.

Contents

1. Introduction

1.1 Objective

It was estimated in November 2020 that 25% of health and fitness clubs closed due to COVID-19 [1]. This forced 20% of Americans to pivot to at-home workouts, which in turn meant them losing access to both the expert knowledge of trainers and class instructors and the community knowledge and support of other people working out"[2].

While at-home workouts are an excellent way to stay healthy, displaced gym goers do not have access to trainers or spotters. This leads to performing exercises poorly and over time, injury or strain. Additionally, working out at home means it is harder to stay consistent and motivated.

The way to address these problems is to give people a way to collect and examine data from their workouts. Our proposed solution is to build a smart exercise mat and companion app to help augment at-home workouts. The pressure-sensitive mat would send data while the user is performing sets of exercise and the app would be used to analyze performance data.

1.2 Background

We contend there is a strong need for a product such as ours because the number of people who had to pivot to at-home workouts skyrocketed due to COVID-19. But the need for GYMplement goes beyond use during quarantine. There is a huge portion of Americans who want to exercise but cannot afford trainers or gym memberships. Our product would address both displaced gym-goers and novice exercisers by giving them a standalone tool to improve their performance without high recurring cost.

GYMplement does not require users to have anything besides their computer. Users would only be performing bodyweight exercises. By creating an ecosystem where people can collect and study their data as well as build custom workouts, we are addressing the issue of ineffective and potentially strenuous exercises.

1.3 High-Level Requirements

- The mat is able to distinguish between the phases and types of the following exercises: push-up, lunges, squat, sit-ups, planks.
- The mat is able to count reps of the aforementioned exercises within 90% accuracy.
- The mat is able to differentiate between two different qualities of exercises (a well-performed push-up and poorly-performed push-up.

2. Design

The GYMplement mat requires four modules shown in Figure 1 to operate successfully: a power supply module, a pressure sensing module, a control unit module, and a software module. The battery pack and buck-boost converter in the power supply module ensure that the sensors on the mat as well as the microcontroller are continuously powered while turned on. The pressure sensing module consists of 3,872 sensors, each covering a ¼ square inch area of the mat. The control unit individually reads each of the pressure sensors and communicates them to the software application through a bluetooth chip. Finally, the software application provides visual feedback and personalized critiques of their form during their workout.

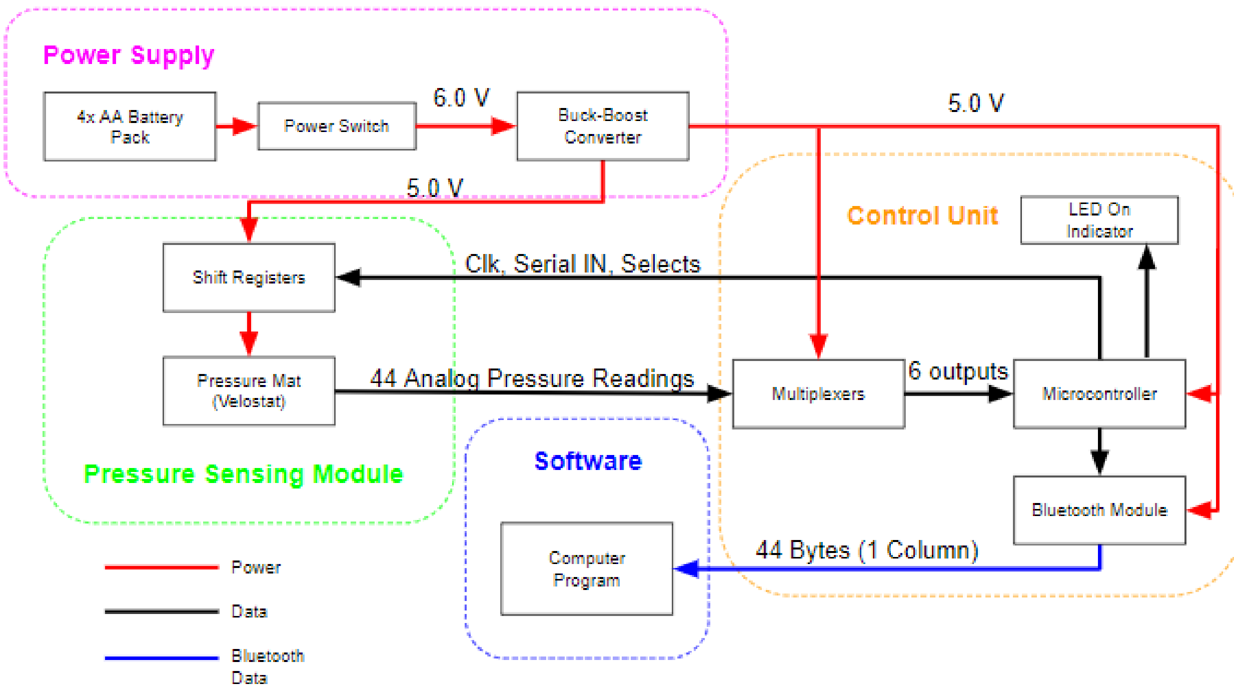

Figure 1: Block diagram

The flow sequence of the hardware is consistent, as the conditional states of our project are all implemented in the software application. Once the mat is powered on, the rows of the copper grid immediately begin receiving power. The pressure sensing module provides power to only one row at a time and promptly returns the 44 readings associated with that row. Once the entire mat has been read and saved, the pressure sensing module is cleared and a new mat reading begins without delay.

2.1 Pressure Sensing Module

The pressure sensing module is the most unique module of our design. Utilizing the velostat material is an extremely cost effective method to create a detailed pressure mat which contains 3,872 sensor readings within an area of 968 in^2, which equates to 4 sensors/in^2. Copper tape is used as the conductor to supply a voltage across the velostat in order to determine its resistance, and therefore the amount of pressure applied to it.

2.1.1 Pressure Mat (Velostat)

The pressure mat is composed of five different layers. The outermost layers of the mat are composed of an adhesive vinyl wrap. This was chosen because it is a non-conductive, waterproof material that also has an adhesive side. These qualities are desirable for our application where we will be running current through copper tape on one side of the material, and the other side will experience moisture from the user performing their workout. The next two outermost layers are both made of ¼" copper tape. One layer will be comprised of 88 rows, while the other will have 44 columns. The final layer which separates the rows and columns of copper is the velostat. Figure 2 illustrates the layers of the mat.

Figure 2: Side view of the layers of the pressure mat (not to scale)

By using velostat to separate an array of copper tape, at each intersection of the rows and columns, we are able to determine the relative pressure being applied to that specific location. Figure 3 helps to describe where the pressure readings are taken with respect to the copper array.

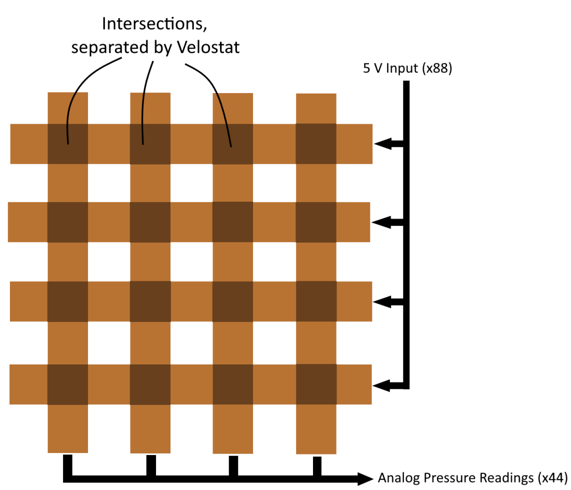

Figure 3: Top view of the copper tape array

2.1.2 Shift Registers

The shift registers are used to supply 5 V to each of the 88 rows of copper. Serial shift registers proved to be the optimal device for this task because only one row could be powered at a time. By connecting the serial input pin of one shift register to the output pin of the previous chip, we could ensure that only one high signal would be sent through the line of 11 eight-bit shift registers at a time (totaling 88 shift register outputs). The reason for wanting only one high row at a time is because if more than one were to be powered, then the analog value on a column of copper tape would be representative of the average pressure being applied to the points where that column intersects each row being supplied with 5 V.

Additionally, diodes would need to be used on each of the shift register outputs as shown in Figure 4. These diodes are used to make sure that current only flows out of the shift register pins and not into them. If the diodes were not included then when the output of a shift register pin is low, it would be grounded and therefore draw current and cause invalid voltage readings on the columns.

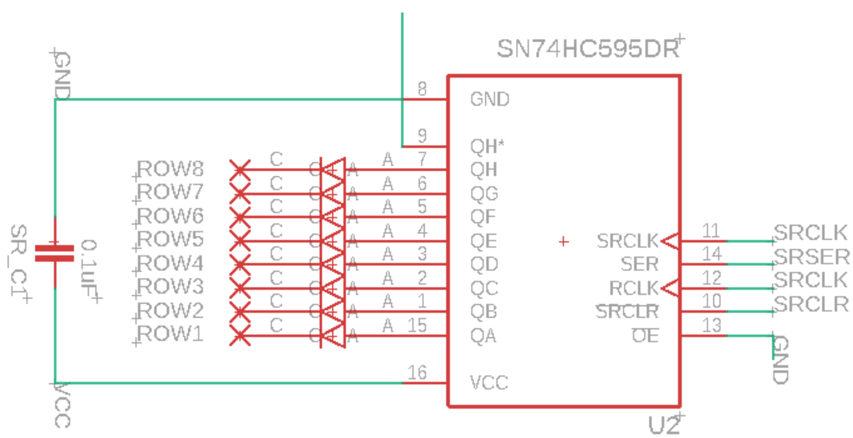

Figure 4: Schematic of the first serialy connected shift register

2.2 Power Module

The power module is responsible for supplying power to the pressure sensing module and the control unit. All components that must be provided with power require a 5 V DC supply. This was an intentional design choice made when choosing the individual components that would be used throughout the project. It is also equipped with a switch between the battery pack and buck-boost converter to provide the user a way to disconnect power from the entire product.

2.2.1 4x AA Battery Pack

Four AA batteries are used in series as the primary power supply. By selecting a power source that utilizes common batteries found in most households, the user is able to conveniently replace the batteries should they lose their charge. An additional advantage of using this battery pack is that it allows the mat to be portable and it can therefore be used on any durable, flat surface without requiring an electrical outlet to be located nearby. However, using four AA batteries in this configuration will provide an output of 6 V, which will need to be regulated to the required 5 V.

2.2.2 Buck-Boost Converter

In order to step-down the 6 V supply provided via the battery pack, we implemented a buck-boost converter into our design. This was found to be the ideal method of voltage regulation because of the small difference between the provided and desired voltage levels. Due to the 1 V difference between the two, many step-down converters would not have been able to reliably provide a 5 V output. Moreover, by using a buck-boost converter, which has the ability to step the voltage up or down, it is still able to provide a 5 V output when the batteries near the end of their charge begin to output a voltage that may fall below 5 V.

2.3 Control Module

The control module is in charge of collecting the data from each of the 3,872 sensor locations and sending the results to a computer for processing to be done. Within the control unit, the microcontroller is responsible for all control and clock signals of the shift registers and multiplexers. It also must convert the analog pressure readings to digital values and then send this data to the bluetooth module where it is sent to and processed by the computer program.

2.3.1 Multiplexers

As stated previously, there are 44 total copper columns that will provide the analog voltage pressure readings. Our microcontroller only has six pins capable of analog to digital conversions; therefore, we must use multiplexers (MUXs) to reduce the 44 analog inputs, to just 6. This can be easily accomplished by using six 8-to-1 MUXs. The final MUX will only need to use four of its eight available inputs. MUXs are fairly simple in operation as they only require power, three select bit signals, and the desired inputs. The three select bits are controlled by the microcontroller, and the inputs are provided from copper 44 copper columns which are connected via ribbon cables. The final important design consideration when implementing the MUX submodule was ensuring that each of the inputs had a 1 kΩ resistor so that the copper columns were connected to ground, or else current would not be able to flow. This means that we are actually reading the voltage across the 1 kΩ resistors; however, this can still be used to find the velostat resistance, as shown in Equation (1)

$$R_{velostat} = (R_1 V_{cc} / V_{analog}) - R_1 = 5\,k\Omega \cdot V / V_{analog} - 1k\Omega \quad (1)$$

where $R_{velostat}$ is the resistance of the velostat, R_1 is the resistance of the chosen pull-down resistor (1 kΩ), V_{cc} is the voltage supply (5 V), and V_{analog} is the voltage at the input of the MUX. Figure 5 shows a basic representation of the schematic for collecting the voltage across one sensor.

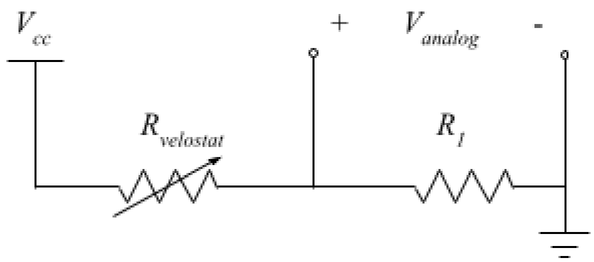

Figure 5: Basic schematic of a single pressure sensor voltage reading

2.3.2 Microcontroller Unit

The microcontroller unit (MCU) we chose to implement our design is the ATmega328P. This is a commonly used MCU and is most well known for its use on the Arduino Uno boards. Due to its popularity and sufficient amount of input and output (I/O) pins, we found it to be a good choice. In order to control the MUXs, the MCU used three digital output pins to control the select signals. We then connected its six analog to digital converter pins to the outputs of each of the MUXs. Operation of the shift registers requires three total output pins from the MCU. One output controls the clock signal of the shift registers which causes the shift register outputs to shift on each of the falling edges of the clock. The next MCU output is able to clear all shift registers, this is done on each startup of the mat. Finally, the first of the eleven shift registers has its serial input pin connected to the MCU because there has to be some way to initialize the first column of the copper array to be 5 V at the beginning of each data collection.

The final function of the MCU is to communicate serially with the HC-05 bluetooth module. Due to the limited dynamic memory of the MCU, we were unable to collect an entire array of values and send it all at once, which would have been the fastest way to send the data. Instead, we had to collect 44 eight-bit digital values for one row of the mat and send it before collecting the values for the next row.

2.3.3 Bluetooth Module

For the purposes of communicating with a computer from our PCB via bluetooth, we chose the basic HC-05 breakout board. It is a simple module that can be programmed to change the device name and baud rate using AT commands. Besides power, it only requires two connections to operate: transmit (Tx) and receive (Rx). The Rx pin of the HC-05 expects a 3.3 V signal to be used; however, our MCU operates at 5 V so we used a voltage divider to reduce the voltage being sent to the Rx pin.

2.4 Software

While the hardware is responsible for gathering data, the software is responsible for transmitting data, analyzing data, and giving the user feedback. Once a mat reading has been sent via Bluetooth to the software application, image processing must be performed to provide the user with feedback such as rep counting and quality of exercise. The app's responsibilities can be distilled into the following steps: Characterize, Count, and Critique.

2.4.1 Transmitting Data

In order for our application to receive new data from the mat, the microcontroller needed to be programmed to interface with the Bluetooth chip. To do so, we uploaded a bootloader to the microcontroller that allowed us to program it using the Arduino IDE instead of a more bare-bones approach using C language. Although using Arduino and its libraries would inevitably result in a slower processing time, the ease in setting clock and select pins as well as the useful "AnalogRead()" function made it a more beneficial choice for the project.

The microcontroller program accomplished two major goals: operate the IC chips on our PCB and write data to the Bluetooth serial bus for transmission. Operating the IC chips simply consisted of flipping the multiplexer select pins and making many AnalogRead() calls inside a set of loops. After enough loops were iterated such that one full row had been read, these bytes were sent to the Bluetooth chip to be received by the computer application. Although initially a long process of roughly two seconds to read all rows of the mat, we were able to speed this up to nearly 0.4 seconds by increasing the clock divisor for the internal ADC. This was possible because our design only called for eight bits of precision, and since the internal ADC gave ten bits, we could ignore the less accurate, sped up reading by bit shifting the two least significant bits out. After all 88 rows of the mat are sent, a "new mat" signal is sent to notify the computer application so that it can distinguish an old set from a new set of data.

2.4.2 Receiving and Visualizing Data

Once the mat and the laptop are paired via Bluetooth and the program is started, the pressure values generated by the user on the mat are sent row by row. Speed is a vital characteristic when trying to distinguish small changes in pressure over time, so an issue that would surface when making an application is a slower readout time. If it takes too long to analyze and plot the data, it will miss the "new mat" signal and end up skipping every other mat reading. This doubles the readout time and is unacceptable.

To counteract this, our Python application contains threads. Threads are simply processes that can run simultaneously in one program. Typically, programs execute in order line by line, but using threads, our program can be receiving data while also analyzing and plotting it. This means the required analysis time will never impede the frequency of incoming Bluetooth data. The analysis thread waits until the Bluetooth thread has acquired a full mat reading (i.e. 88 row readings). A live heatmap is then generated from this data which gives the user a look at their movements. This data is also used to characterize, count, and critique exercises as discussed in the following sections.

2.4.3 Image Processing

Before the data can be distilled into types, counts, and qualities of exercises, the array of pressure values must be searched for the highest areas of pressure. These areas correspond to where the user is in contact with the mat. In order to do this, we used convolution. To generate the kernels needed for convolution, we had one of our group members do a squat, push-up, and plank on the mat. Then, using manual visual inspection we extrapolated a kernel for each of the following: left and right arm, left and right foot, left and right hand. These were the golden standards used during normal operation.

2.4.4 Distinguishing Type of Exercise (Characterize)

For each full array of pressure values received from the mat, or snapshot, we convolved the golden standards over the current matrix to get the highest areas of pressure, also known as the regions of interest (ROI's). This means we had six ROI's in total that corresponded to a pair of hands, a pair of feet, and a pair of arms. The correct pair of ROI's was chosen using a mean squared error (MSE) function. The MSE function calculated the mean squared error of a given pair of ROI's and its golden standard, returning a value corresponding to the sum of all the pixel errors between the two. For example, if the golden standard was fed to the MSE function, it would return a zero since they are identical. Each pair of ROI's is given to the MSE function, and the pair with the lowest return value is the chosen type of exercise. If the ROI matched a hand, the snapshot was deemed a push-up. An arm resulted in a plank, and a foot resulted in a squat.

A confirm feature was later implemented into the software design. Before beginning an exercise, the user would first take the stance of the exercise they wish to perform. The program would then wait for ten consecutive characterizations to tell the user to begin. A different characterization would reset the counter to zero, making the program wait for a stable reading. After an exercise was confirmed, that exercise is indefinitely assumed until the user finishes the reps of that exercise. This means only the ROI's pertaining to that exercise are found and used, allowing the program to run faster and present more accurate pressure readings during the workout.

2.4.5 Count Reps (Count)

To count the reps, the total pressure inside the ROIs was calculated for each snapshot. If there was a large enough difference between the ROI total pressures of the previous and current, it meant a rep was completed. Some exercises exert more pressure on the ground than others. For example, a squat puts all weight on the feet, while a pushup distributes weight over both the hands and feet. This means push-ups and squats required different pressure difference thresholds, namely, a greater threshold for squats than push-ups. Therefore, a list of threshold values was required for each type of exercise that were dependent on the user's weight. Planks do not have reps, so a timer was implemented.

In addition to using pressure difference to count reps, we also incorporated a "momentum factor" into our equation. When doing one rep of an exercise, there are actually two pressure changes: a dip in pressure when you let yourself accelerate downward and a spike in pressure when you accelerate yourself back upwards. Since two pressure changes could correspond to two reps per exercise another parameter was needed. The momentum factor, in essence, makes it difficult for more reps to be counted in a shorter amount of time. After a rep is counted, the next mat reading requires a pressure change three times larger than normal to result in another rep. The following requires a change twice as large, and it returns to normal on the third reading after a rep. This prevents consecutive dips and spikes from counting as two reps in our program.

2.4.6 Recognize Quality of Rep (Critique)

We defined a "good" plank, push-up, and squat as a rep where the distance between hands, feet or arms and the difference between pressure applied on both sides is within a certain bound. For the distance, we needed to account for when a user could be performing a squat or push-up that required the placement to be wider or narrower than shoulder width. After referring to several different pieces of fitness literature [3] [4] [5] [6] [7], we decided that a distance greater than 1.5 times a user's shoulder width or less than 0.5 times a user's shoulder width would be considered low quality. In these cases, the program would tell the user if they were too wide or narrow and adjust accordingly. For the weight, if the ratio of left to right or right to left was greater than a factor of 1.2, the user was notified of which side was being used more and how to adjust. We also had checks for if the hands, feet, or arms were misaligned. If they were not on the same plane, the user was notified which side was higher and to adjust accordingly.

3. Design Verification

3.1 Pressure Sensing Module

3.1.1 Pressure Mat (Velostat)

There were two steps taken to sufficiently test the pressure mat portion of our product. The first step in making the pressure mat was to apply the rows and columns of copper tape to the two sheets of vinyl. Next, the ribbon cables were soldered to every column and row of tape. Once this was done, we verified that the resistance from one end of the ribbon cable to the opposite end of the copper tape was low enough to be considered negligible. The maximum resistance measured was 0.6 Ω and this is small enough to not have a noticeable effect on the voltage throughout the mat. The final step of constructing the mat was to apply the layer of velostat to one of the copper tape and vinyl layers and then carefully place the other layer of copper tape and vinyl on top of it. Now that the whole mat was physically constructed, we were able to test applying pressure to the mat. In order to this, we simply connected a multimeter to a row and column of the mat through the ribbon cables. Next we located where that row and column intersect and applied pressure to it while observing the change in resistance. It was easy to see that even a small amount of applied pressure led to a noticeable change in resistance.

3.1.2 Shift Registers

In order to test the shift registers, we soldered them to our custom designed printed circuit board (PCB). Then we programmed the MCU to set the first serial bit high at the beginning of every array collection. Because the MCU also controlled the clock signal, we knew that this occurred after 88 clock pulses. An oscilloscope was connected to one of the shift register outputs to verify that it would output a single pulse at a constant frequency. Figure 6 shows the results from this test which prove that the shift registers work as expected.

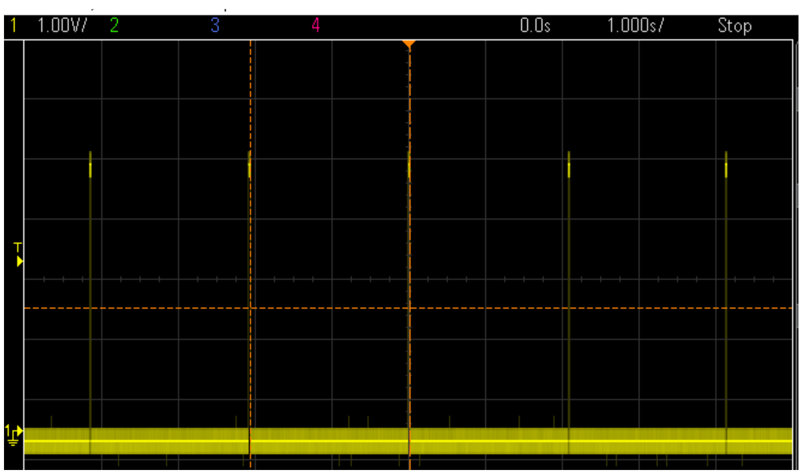

Figure 6: Oscilloscope screen capture showing operation of single shift register output

3.2 Power Module

3.2.1 4x AA Battery Pack

Verifying correct functionality of the battery pack was very simple. The batteries were loaded into the pack and we used a multimeter to measure the voltage across the output wires, this was verified to be 6 V.

3.2.2 Buck-Boost Converter

All dc-dc voltage converters will have some sort of voltage and/or current ripple because of the components that it is made of and the switching that occurs. Therefore, we tested our buck-boost converter by connecting it to an electronic current load. The load drew a constant 1 A, and we observed the resulting waveform on an oscilloscope, shown in Figure 7.

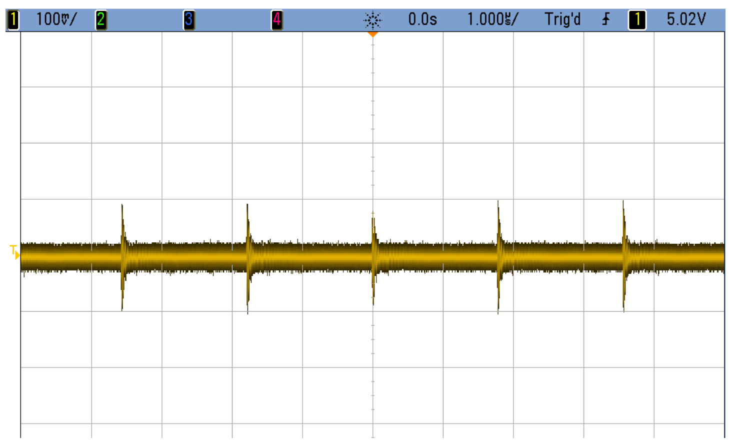

Figure 7: Buck-boost converter voltage ringing at load of 1 A

As seen from this image, the peak-to-peak ripple was only 200 mV. Our original goal was to achieve an output of 5 V ± 3 %. This equates to a peak-to-peak value of 300 mV, which means our converter operates well within the desired range.

Another important characteristic of the power supply is the length of its battery life. The average energy contained in a AA battery is approximately 2.5 A-h [8]. Our product drew only 70 mA during operation. This means that our device will operate for around 36 hours before batteries need to be replaced. That amount of time equates to 36 one-hour workouts, which means someone who works out daily would only need to replace the batteries once a month.

3.3 Control Module

3.3.1 Multiplexers

The multiplexers were tested after they were soldered to the PCB. We had uploaded the MCU with the program that it would use to carry out all of its functions. There was then an oscilloscope connected to the output of one of the MUXs so that we were able to observe the waveform of all eight puts being cycled through. We then applied pressure to one of the intersections of the copper column that it is connected to in order to make the periodicity of the output more noticeable. Figure 8 shows the result and verifies that the MUXs operate correctly.

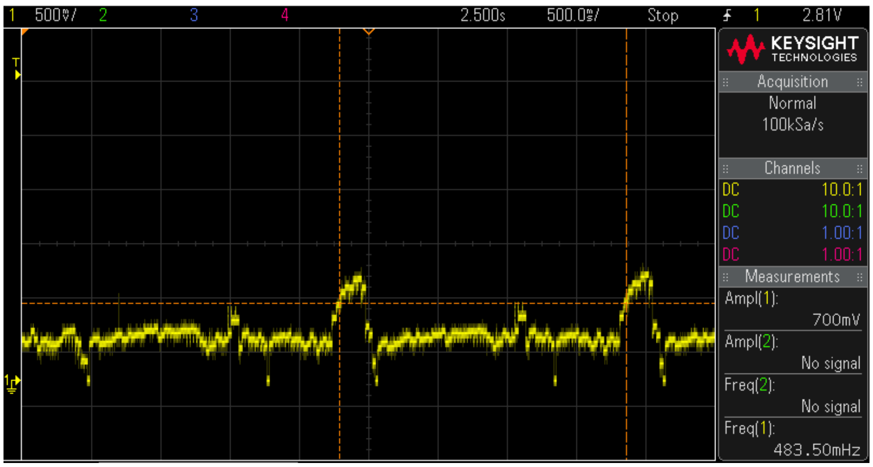

Figure 8: Oscilloscope capture of MUX output

3.3.2 Microcontroller

The first aspect of the microcontroller (MCU) which we verified was that we could properly upload a simple program to it. This was done using an FTDI programmer, which acts as a USB to serial converter so we could program the MCU with our computers, and the test program we uploaded was designed to make an LED blink. Once we accomplished this, the actual program was uploaded to the chip and it was tested by putting it into our PCB and seeing if our project would send accurate data to the computer that it was paired with. It proved to be successful in controlling all other IC chips, converting analog voltages to digital, and communication with the HC-05.

3.3.3 Bluetooth Module

Before integrating the HC-05 bluetooth module into our project, we tested it using an Arduino UNO. Once we were able to communicate with a laptop via the HC-05 we knew that it would be able to work with our MCU because the Arduino Uno uses the same exact chip. Once we combined it with the rest of our project we saw that it successfully paired with any laptop with bluetooth capabilities and was able to send the pressure readings one column at a time.

3.4 Software

3.4.1 Transmitting Data

To verify data transmission was functional, a short Python program was written to open a serial port and write incoming data to the terminal. To ensure everything was correct, two steps were performed. The first step was to ensure data is coming in periodically. The incoming readings should be the same length and take the same amount of time. If the mat is not being used, then the values should also be very similar since no change is. If one of these statements is not reflected in the terminal, then there is an error with how data is being written or read from the serial port. The second step was to ensure the data was meaningful, i.e. not all zeros or ones. This often occurred due to incorrect data conversion between strings, integers, and bytes. Once all these requirements were met, we had a valid stream of incoming readings to begin testing with.

3.4.2 Visualizing Data

Now that a steady stream of data has been obtained, there are no longer any hardware issues at hand. All remaining issues would be due to software alone. Verifying data visualization only requires pressure on the mat being recognized in the visualization. Orientation and refresh rate issues were immediately apparent upon start up.

3.4.3 Image Processing

After data transmission was verified, a groupmate completed a push-up, squat, and plank on the mat. We saved a snapshot for each exercise, generated a heatmap from them and cut the array down to contain only the high pressure areas using manual visual inspection.

3.4.4 Distinguishing Type of Exercise (Characterize)

To test and verify the characterization capabilities of the application, we performed several iterations of planks, squats, and push-ups. While one member did the exercises, another member watched the terminal to see if the program was characterizing the correct exercise. Once we were able to recognize each exercise about thirty times each, we considered this requirement achieved.

3.4.5 Count Reps (Count)

The same method described in 3.4.4 was used to verify rep counting. Implementing the momentum factor described in 2.4.5 increased the accuracy to over 90%.

3.4.6 Recognize Quality of Rep (Critique)

The same method described in 3.4.4 was used to verify quality critique. Every aspect of a poor exercise our mat was designed to detect was performed and successfully identified.

4. Costs

A breakdown of the cost for parts and labor are shown in the two tables below. The total cost of our project, including parts and labor, ended up being $45,203.93.

4.1 Parts

Table 1: Detailed Parts Cost List

Part Number	Description	Quantity	Unit Price [USD]
Adafruit; 1361	11" x 11" Velostat sheet	8	4.95
Amazon; OFC00006	1/4" x 21.9 yards copper tape	5	5.98
Amazon; B00B2IHCJ6	10 ft. 50-wire ribbon cable	1	10.95
Amazon; B01L1W65AE	11.5" x 60" vinyl wrap	4	10.9
Amazon; ADA830	4x AA battery holder	1	8.99
Amazon; B01G9KSAF6	HC-05 bluetooth module	1	9.99
Amazon; ADA284	FTDI programmer	1	14.9
DigiKey; ATmega328P-PU	Microcontroller	1	2.52
Mouser; 1-2199298-9	28 pin IC socket	1	0.47
Mouser; ECS-160-18-4X-CKM	16MHz crystal oscillator	1	0.53
Mouser; C0805C220J4GACTU	22pF capacitor	2	0.24
Mouser; CMP0805AFX-1002ELF	10kΩ resistor	1	0.14
Mouser; 150080AS75000	SMD LED	1	0.23
Mouser; RN732ATTD5000B25	500Ω resistor	1	0.71
Mouser; SN74HC595DR	8-bit shift registers	11	0.353
Mouser; 1N4148WS	diode	88	0.15
Mouser; CD74HCT4051M96	8-to-1 MUX	6	0.46
Mouser; CRCW08051K00FKEAC	1kΩ resistor	50	0.059
Mouser; CRCW08052K00FKEBC	2kΩ resistor	1	0.16
Mouser; EEE-FN1K100XL	10μF electrolytic capacitor	1	0.54
Mouser; C0805C104K5RAC7411	0.1μF ceramic capacitor	20	0.126
Pololu; 2574	5V buck-boost converter	1	14.95
		Total Parts Cost	**$203.93**

4.2 Labor

For labor calculations, we assumed an hourly salary of ~$40/hour and that we each worked 15 hours a week for 10 weeks.

Table 2: Breakdown of Labor Costs

Team Member	Weekly Hours	Number of Weeks	Multiplier	Cost Per Member
Daniel Rymut	15	10	2.5	$15,000
Justin Naal	15	10	2.5	$15,000
Srinija Kakumanu	15	10	2.5	$15,000
			Labor Cost	**$45,000**

5. Conclusion

5.1 Accomplishments

By the end of the semester, we delivered a fully functional and integrated product. Our pressure-sensitive mat generated data. The data was successfully sent to a laptop. The heatmap updated every time a user moved on the mat which confirmed that the mat was functioning and transmitting data. The characterization, count, and critique of reps provided by our algorithm matched what a human user would say for planks, push-ups, and squats.

5.2 Uncertainties

One bug that we were not able to address by the final demo was that when the user stepped onto the mat, the entire row on which there was any pressure would light up. However, when viewing the heatmap it was still clear what the user was doing (hands, feet, or arms) because the areas where the user was actually placed would be lit up the highest. Also, the data sent to the algorithm was still distinct enough for the analysis to be completed.

5.3 Ethical Considerations

The primary ethical concern of GYMplement is privacy. For the version of the product presented at the final demo, the only body measurement saved was the shoulder width of the group member from whom we based the golden standards on. However, in future iterations of the product we would be asking the users to enter their own personal body measurements so they calibrate the golden standards to themselves.

For many people, especially those trying to get in shape, this may be sensitive information. Our users assume that the private data they enter is safe, so we must ensure that the data they put in their smartphone will not be used or distributed in any way that breaches privacy. To accomplish this, we guarantee that user data will not be stored on a server and that the app will not connect to the internet. This allows us to assure users that their data is secure as long as the phone itself is not broken into or compromised. In other words, we must ensure that we adhere to Principle 1.6 of the ACM Code of Ethics, "Respect Privacy".

Furthermore, the GYMplement mat must adhere to Principle 1.2 of the ACM Code of Ethics, "Avoid Harm". The mat contains conductive copper tape and is battery powered, so users should avoid water while using the GYMplement mat. Any contact between electrical components and water may cause the circuit to short and overheating to occur. Thus, users should not use our product outdoors or near any source of water. Users will unavoidably sweat during a workout, so the yoga mat material will be water resistant and fully encompass all electronic components such that none are exposed.

Lastly, as with all pieces of workout equipment, users must exercise individual caution when on the mat. If a user is feeling strain or pain, no amount of computer analysis is better than stopping a workout and resting. Therefore we caution all users of GYMplement to prioritize their health and safety while working out. If something does not feel right, step off the mat and take a break.

5.4 Future Work

To further develop GYMplement, we would make several additions. First, we would expand the library of exercises the app can analyze. First, this would entail generating golden standards for any other body parts or orientations these new exercises would need. For example, if we wanted to support a contralateral limb raise, we would need a standard for the user laying on the mat on their torso. Second, we would need to define what makes a good rep of the exercise and analyze the ROIs to see if the data generated falls within the “good” or “bad” bounds.

Another enhancement we would make is, as mentioned in the previous section, adding a calibration process for users so that they generate their golden standards according to their own measurements. The last most immediate enhancement we would work on is creating a mobile app where users would see the heatmap and feedback as well as create circuits and past workout data. This would make GYMplement truly be portable since people can much more easily travel with their mobile devices rather than their laptops.

References

[1] “One in Four Health & Fitness Clubs Could Close by Year-End,” *IHRSA*. [Online]. Available: https://www.ihrsa.org/about/media-center/press-releases/15-of-health-fitness-clubs-in-the-u-s-permanently-closed/. [Accessed: 17-Feb-2021].

[2] K. Eschner, “COVID-19 has changed how people exercise, but that doesn't mean gyms are going away,” *Fortune*, 12-Jun-2020. [Online]. Available: https://fortune.com/2020/06/11/coronavirus-gyms-workouts-fitness-apps-reopening/. [Accessed: 17-Feb-2021].

[3] David Cruz, DC. “Squat Form – What Does It Tell Us? Part 1.” NASM, 2020, blog.nasm.org/newletter/squat-form.

[4] Koh, Da Hyun, et al. “Plantar Pressures in Individuals with Normal and Pronated Feet According to Static Squat Depths.” Journal of Physical Therapy Science, The Society of Physical Therapy Science, Sept. 2015, www.ncbi.nlm.nih.gov/pmc/articles/PMC4616105/.

[5] Chuckpaiwong, B, and T Harnroongroj. “Http://Www.smj.org.sg/Sites/Default/Files/5007/5007a6.Pdf.” Republic of Singapore, Singapore, 2009.

[6] Dorman, Deanna, and Deanna Dorman. “How to Do The Perfect Lunge.” Paleo Blog, 12 Jan. 2018, blog.paleohacks.com/how-to-do-the-perfect-lunge/#:~:text=How%20to%20Correct%20It%3A%20When,toes%20in%20the%20front%20foot.

[7] Finn, Christian. “Should I Do Sit-Ups?” Men's Health, Men's Health, 10 Dec. 2018, www.menshealth.com/uk/building-muscle/a750314/should-i-do-sit-ups/.

[8] “Pololu - Understanding battery capacity: Ah is not A,” *Pololu Robotics & Electronics*. [Online]. Available: https://www.pololu.com/blog/2/understanding-battery-capacity-ah-is-not-a#:~:text=AA%20batteries.,to%204%20Wh%20per%20cell. [Accessed: 28-Apr-2021].

Appendix A Requirement & Verification Table

Requirements	Verification	Results
	4x AA Battery Pack	
Provide constant power for at least thirty, one-hour workout sessions.	a. Once the circuit has been implemented on the PCB, measure how much current is drawn during operation b. Four AA batteries in series have ~2.5A*h of energy while providing 6 volts. c. Therefore, the following must be true: 30 hours $\leq$ 2.5A*h / I_{on}	$I_{on} \cong 0.07$A therefore the mat can operate for approximately 36 hours
	Buck-Boost Converter	
Convert input voltage of 6V to an output of 5V ± 3% (max ripple = 300 mV_{p-p})	a. Connect the battery pack to the input terminal of the buck-boost converter b. Connect a resistive load that causes the output current to be equal to I_{on} c. Use an oscilloscope voltage probe connected across the resistance to measure the voltage and voltage ripple d. Verify that they are within specified limits	Max ripple was found to be 200 mV_{p-p}
	Ribbon Cable - Copper tape connection	
Ensure that solder joints that connect the ribbon cable to the copper tape are sufficiently connected for current flow	a. Adhere the copper rows and columns to their respective outer vinyl layers b. Solder the ribbon cables to the rows and columns of tape c. Use a multimeter to	Maximum resistance measured was 0.6Ω

	measure the resistance from the end of the ribbon cable to the end of the copper tape d. All resistances should be less than 1Ω	
Velostat - Pressure Dependent Resistance		
Applying pressure to the mat with an applied layer of velostat will cause the resistance of said layer to decrease	a. Connect one of the 88 copper rows to 5V of power and connect the copper columns to ground in series with 1kΩ resistor. b. Use an oscilloscope to measure the voltage across the 1kΩ resistor. c. Apply pressure to the intersection of the copper column and row, there should be an observable increase in the voltage across the resistor	Verified
Microcontroller/HC-05 Programming		
Must be able to program the microcontroller to send data to a computer via the bluetooth module	a. Upload bootloader to the ATmega328P in order to enable use of Arduino IDE b. Upload test code that prints text to the terminal which the HC-05 is connected to c. Connect HC-05 Rx/Tx pins to the designated pins on microchip d. Connect computer to HC-05 using terminal emulator and observe whether or not text is printed to screen	Verified
Multiplexor operation		

The MUXs must quickly output each of the eight inputs to the ADC in order to be read	a. Connect the exercise mat to the soldered PCB and observe voltage across the output of a MUX with an oscilloscope b. With a constant pressure applied to the mat, there should be a periodic signal showing the voltages across the 1kΩ resistors	Verified
Software		
Must be able to read in data from mat	a. Heatmap accurately reflects the mat and updates when a user moves on the mat	Verified
Must be able to characterize the type of rep	a. Program characterizes a rep as a human would	Verified
Must be able to count number of reps completed	a. Count generated by program matches how many reps a human would count	Verified
Must be able to critique reps	a. Reps deemed high-quality by a human are also deemed high-quality by the program. b. Reps deemed low-quality by a human are also deemed low-quality by the program.	Verified