# BENCH PRESS SMART HELPER

By

Alejandro del Rosal

Eduardo Quintana

Carlos Suberviola

Final Report for ECE 445, Senior Design, Spring 2022

TA: Uma Lath

4 May 2022

Team 35

## Abstract

The Bench Press Smart Helper aims to minimize injuries related to the bench press by providing necessary help when performing the exercise. Our solution integrated a computer vision program with an electric hoist which is activated and helps the user lift the barbell during previously specified repetitions or at failure. The device can count repetitions, detect failure, and assist the user in precise moments. The amount of help the hoist provides is a function of how much force the lifter is exerting on the barbell. Finally, assisted repetitions were on average 1 second longer than non-assisted repetitions.

# Contents

# 1. Introduction

## 1.1 Problem

The bench press is by default the best compound exercise to build the chest, triceps, and shoulders. It is also the powerlifting exercise to which more injuries are related to. A recent study by BMJ Open Sport & Exercise Medicine conducted over sub-elite to elite powerlifters [1] showed that up to 46 % of their injuries were caused by the bench press, accounting for almost half in some athletes. To avoid these injuries, one tends to either lower the intensity of the bench press or choose other exercises that are not as prone to causing injuries. However, this is far from ideal, as other movements do not encompass the benefits of the bench press –the two largest electromyography (EMG) studies [1] for chest exercises hold the barbell bench press as the clear winner for best exercise for chest activation– and performing this exercise at half intensity does not optimize growth and strength in any way.

Aiming to perform the bench press with a good intensity, we end up seeking external help, which generally comes from other people at the gym, and is usually deficient and very inconsistent. Furthermore, even if there are individuals who can provide good and consistent aid, it is still not ideal to rely on people to have a good workout at all. Finally, when performing the bench press, going to failure does not necessarily mean not being able to complete more repetitions with the allocated weight. A few more repetitions can always be squeezed out with some help, and that extra effort is essential in maximizing muscle use. Our design guarantees the possibility of putting that extra effort safely with reliable and consistent help.

## 2.2 Solution

Our design solves three problems: the first one, the dependence on another person to perform the bench press with good intensity and minimized risk of injury; the second one, the inconsistency and unreliability of human help; the third one, the inability of going to failure without it posing a serious health risk.

We created a smart machine that solves all these problems simultaneously. Our solution simulates the role of the ‘spotter’ in a more sophisticated and complete way. The spotter is the person at the gym who helps another person perform an exercise; their task is to actively follow the motion of the lifter, correct any mid-exercise mis forms, and ultimately, provide an extra hand to the person lifting the weights. However, unlike a real spotter, our system can provide quantitative feedback and accurately control the amount of help it delivers, and when it delivers it.

Integrating computer science, electrical engineering and mechanical engineering, our project aims to help lifters minimize the possibility of getting injured and maximize chest activation by providing consistent, reliable help.

## 2. Design

Our design is as follows: placed above the bench, right above where the lifter would move the barbell vertically, is an electric hoist. The electric hoist is controlled by a printed circuit board (PCB), which in turn is connected to a Raspberry Pi that runs a Python computer vision program. Depending on what the computer vision program detects, the electric hoist moves either up or down in precise moments, with the goal of helping the lifter move the barbell when necessary.

Figure 1: Final Design Front Picture

Figure **2**: Final Design Back Picture

The design consists of five subsystems: mechanical, power, sensing, control, and user interface. Each plays a crucial role in the successful operation of the Bench Press Smart Helper, and all need to always communicate effectively.

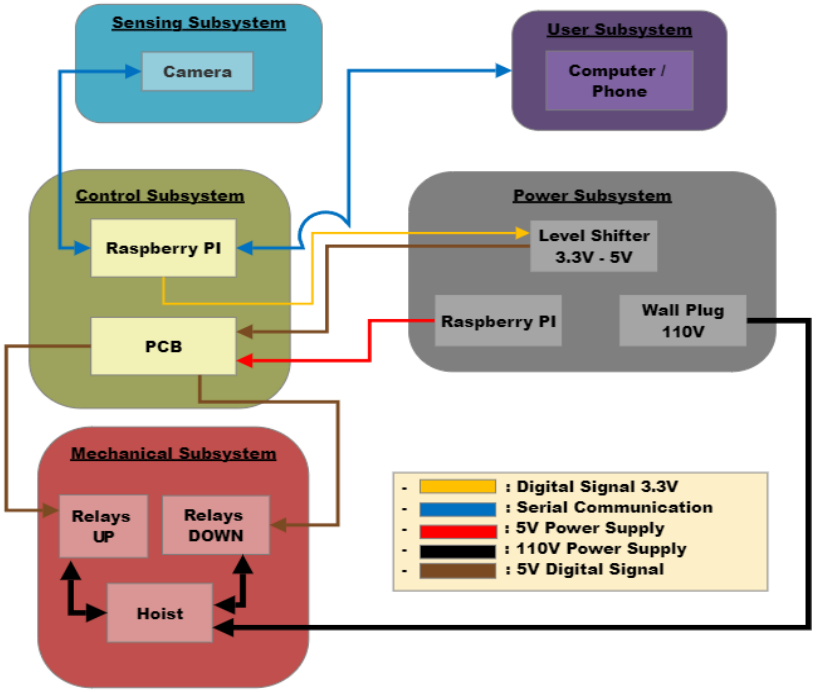


Figure 3: Block Diagram

### 2.1 Mechanical subsystem

The ones related to this subsystem were some of the most important decisions of the project. We needed a device that could lift a good amount of weight vertically without interfering with the user's

performance. Besides, we needed a mountain frame that was not too big so demonstrations could be done in the Lab.

Our first thought was to use a pneumatic system. We wanted to use two pneumatic cylinders to lift the barbell. However, this system included a compressor, a valve and two pneumatic cylinders. Building a system like this one was out of the budget by far.

After ruling out this idea, we thought that the better way to accomplish this was to use a motor that had the power to lift enough weight. We investigated and we found that an electric hoist could fulfil the requirements needed to achieve our goal.

Finally, the Mechanical subsystem consists of the hoist, the mounting frame, and a system of 4 relays. Originally the Hoist was controlled by a switch that we managed to remove and replace for the relay system. These four relays do the function of the switch.

### 2.1.1 Electric Hoist

An electric hoist is a type of motor that has a built-in cable and is used to move heavy objects vertically as opposed to a winch which similarly to a hoist is used to move heavy objects but in this case in the horizontal direction.

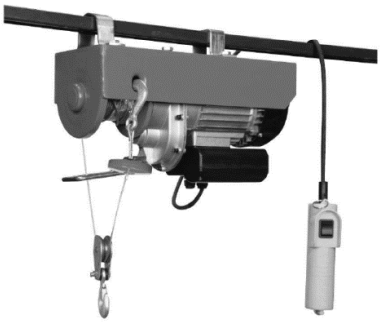

Figure 4: **Electric Hoist Pittsburgh Automotive**

| | |
|---|---|
| Electrical Rating | 120VAC / 60Hz / 4.5A |
| Rated Capacity | 440 lb. double line<br>220 lb. single line |
| Lifting Speed | 26 FPM (single); 13 FPM (double) |
| Cable Length | 39' |
| Cable Diameter | 0.12" (3mm) |
| Duty Cycle | 20%* |

Figure 5: Hoist Specifications

### 2.1.2 Mounting frame

We needed a structure with enough height to hold the hoist. Besides, we wanted it to be portable to try it in a real gym after testing it in the Lab. It needed to have the same width that the metal brackets of the hoist, so we could attach them. The Machine Shop help us with the design idea of the mounting frame. Finally, our design uses the mounting frame shown in Figure 10.

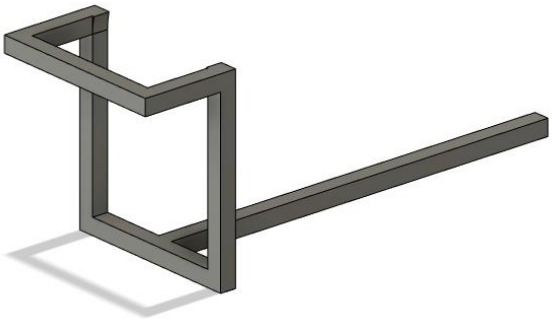

Figure 6: Mountain Frame CAD design

### 2.1.3 Relays

The electric hoist had a manual switched that activated the hoist up or down manually. The connections of the hoist were as follow:

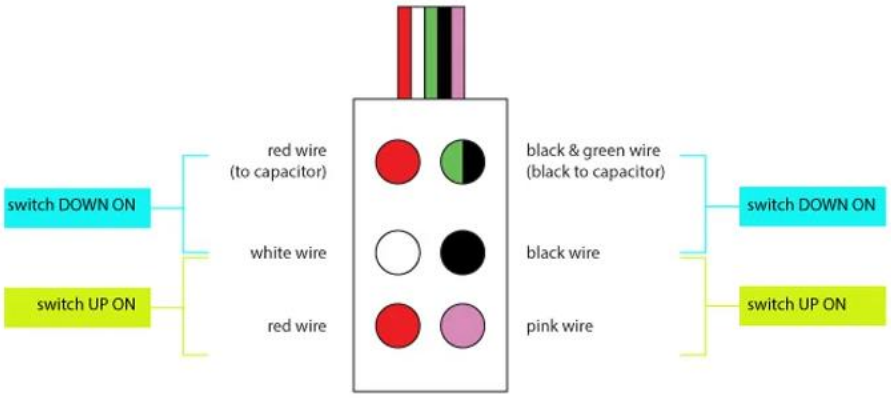


Figure 7: Manual Switch connections

To solve this problem, we decide to use four Solid State Relays that make the connections between the different wires of the hoist, so it is activated to go down or up by sending a signal to the low voltage part of the relays.

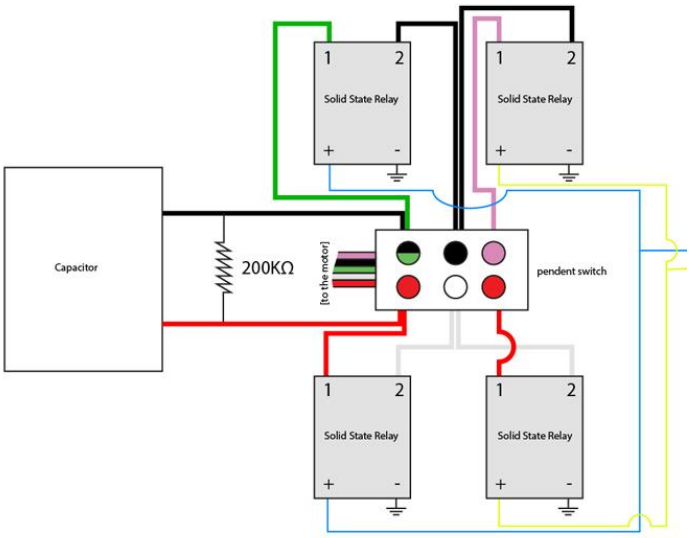


Figure 8: Electric Hoist and Solid State Relay connections [2]

The SSR40DA Solid State Relays that have a DC input between 3V and 32V, and an AC output between 24V and 380V allows us to make the connection between hoist and control subsystem.

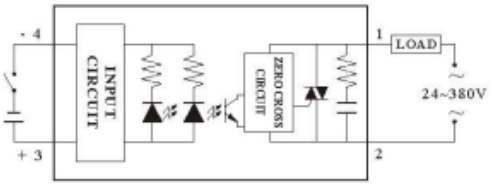


Figure 9: Solid State Relays Schematics

## 2.2 Control Subsystem

Consisting mainly of the programming of our Raspberry Pi 4B and the PCB, the control subsystem may arguably be the most important one out of the five subsystems when considering it works as the brain of our design. It has the task of managing and processing the input and output signals from our sensing subsystem and controlling the hoist, accordingly, thus taking on the biggest responsibility of the project's success.

For our control subsystem to succeed in its task the most important requirement is to establish effective communication between the Raspberry Pi and the PCB and between the PCB and the hoist so that the video image captured by our camera is processed by our Raspberry Pi thus producing the input digital signals that the PCB will receive indicating a change in the hoist action.

### 2.2.1 Hardware

The focus when tackling the hardware component of our control subsystem was understanding the pin layout of both devices and the I/O signals they can handle.

#### *2.2.1.1 Raspberry Pi 4B*

The Raspberry Pi is a single board computer that runs Python, while also providing a set of GPIO (general purpose input/output) pins, allowing the user to control electronic components for multiple purposes. In our project, we use the Raspberry Pi to run the computer vision program and activate certain digital pins to signal the PCB in precise moments (depending on what the program detects).

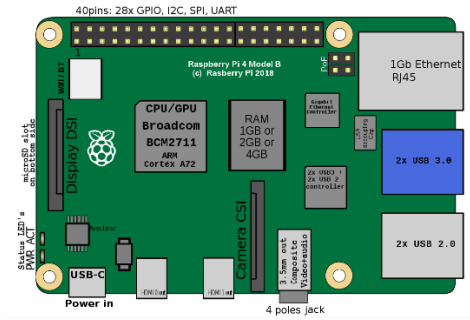


Figure 10: Raspberry Pi

#### *2.2.1.2 Printed Circuit Board (PCB)*

The printed circuit board (PCB) is the interface between the computer vision program and the electric hoist. It consists of resistors, capacitors, a crystal oscillator, a voltage regulator, and most importantly, the microcontroller.

For our project, we chose the **ATMega 328**. The reason for this is that it is a very well documented microcontroller and has various built input/output pins that fit our purposes. The ATMEGA328 runs the hoist activation logic according to the signals it receives from the Raspberry Pi.

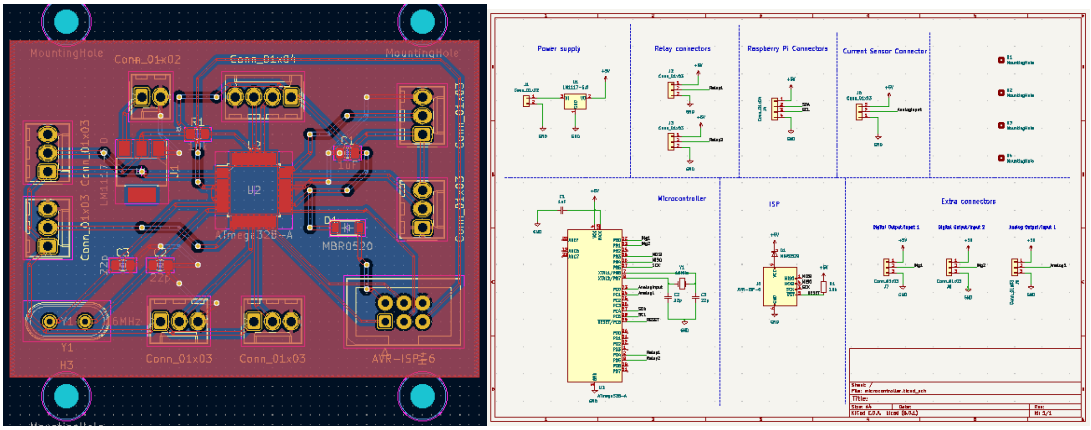


Figure 11: PCB Schematics in KiCad

## 2.2.2 Software

### *2.2.2.1 Computer Vision Python Program*

Our project incorporates an extensive computer vision program written in Python. It has several functions: controlling user interface, tracking the position of the barbell, tracing its motion, counting repetitions, detecting failure, and generating visual feedback.

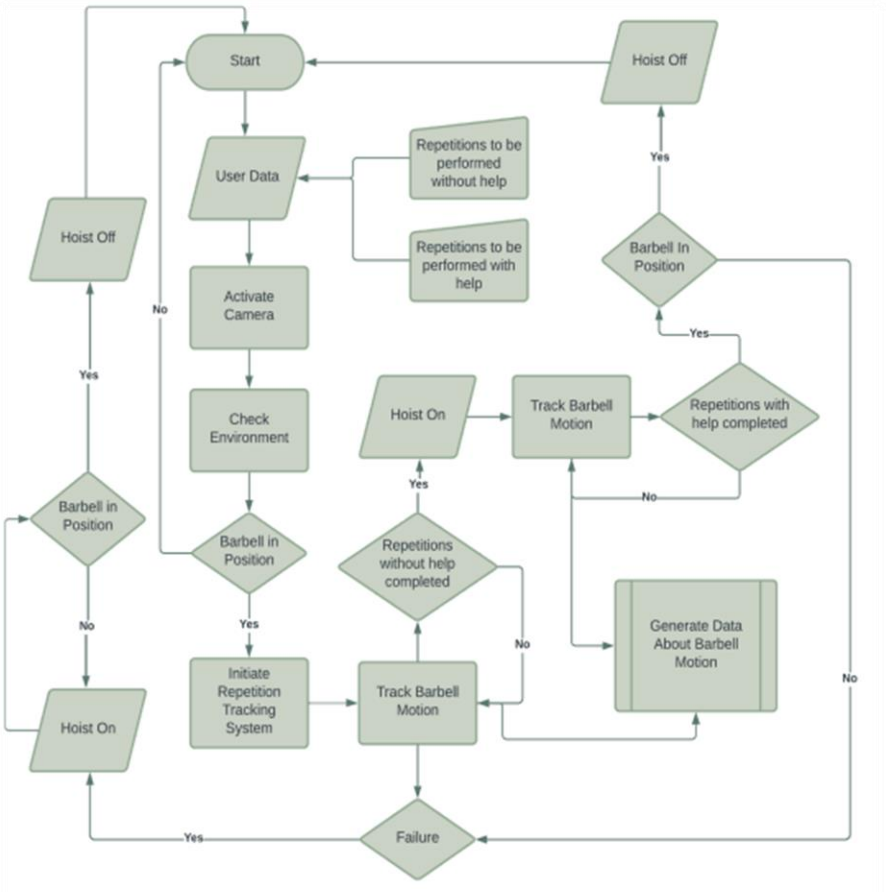


Figure 12: Project Flowchart

#### *2.2.2.2 Arduino IDE C Program*

This program retrieves and interprets the outputs of the Raspberry Pi and generates the inputs for the electric hoist. The outputs of this program are the direct commands to move the hoist up or down and are closely related to what the Raspberry Pi outputs.

## 2.3 Sensing Subsystem

The sensing subsystem is the simplest out of the five. It consists of a camera module that is compatible with the Raspberry Pi, and its only real function is to always capture the entire barbell.

### 2.3.1 Webcam

As our webcam, we used the Arducam 5MP Camera for Raspberry Pi, 1080P HD OV5647 Camera Module V1, which is sold separately as the default Raspberry Pi 4 camera module.

Figure 13: Camera Raspberry Pi

## 2.4 Power Subsystem

The power subsystem oversees the supplying the voltage required for all other subsystems to work. The voltages used in this project range from 3.3V, 5V, 12V to 110V. To deliver these voltages to their respective components, we have used the following resources:

### 2.4.1 Wall Outlet

Starting point of our power supply; will provide around 1800W and 110V from the grid which will then be on one hand, fed directly to our hoist and on the other one led to our AC-DC converter to obtain the desired 5V.

### 2.4.2 AC-DC Converter

For our PCB to work, we need an effective input supply voltage of 5V in efforts not to burn our microcontroller. To achieve this, we are first downscaling the 110V drawn from our regular wall plug to 12V through a transformer with the following specifications [Figure 14]. Then the reduced AC voltage will go through a full wave rectifier obtaining then 12V DC. Finally, the 12V DC will serve as input for a 78L05 voltage regulator that will further reduce the voltage to the 5V DC required by our PCB.

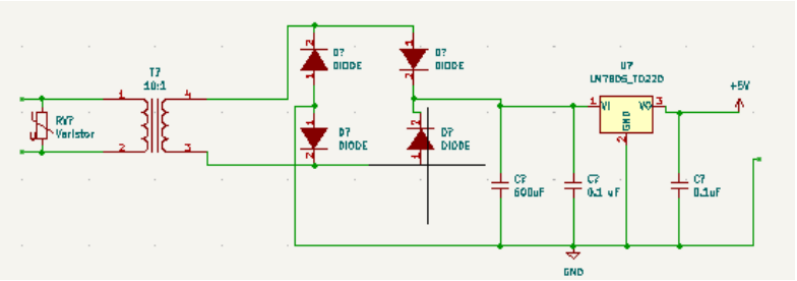


Figure 14: Schematics of the AC-DC converter

Before working with such high voltage and risking damaging both ourselves and our components we decided to simulate the functionality of our AC-DC Converter through LTSpice, and the results are within the expected, input voltage converts effectively from 110V to 5V [Figure 15].

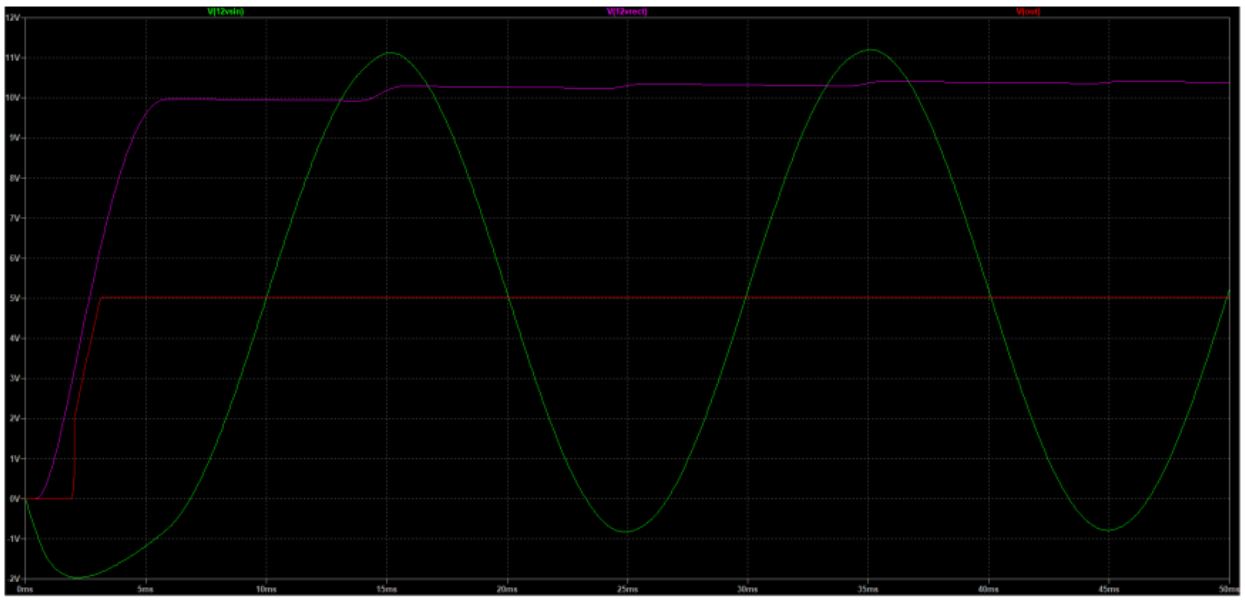

Figure 15: AC to DC Converter simulation

### 2.4.3 Level Shifter

One of the most important aspects of the project is the system integration, specially between the PCB and the Raspberry Pi 4B, since in the case that communication is unable to be established between both, it will be unfeasible for the PCB to know when to send the digital signal to activate the hoist.

The Raspberry Pi has a digital output of 3.3V and our microcontroller has a digital input ranging from 3.5V to 5.5V which makes it impossible for them to work without the intervention of a third party like the level shifter. In addition, the relays that will serve as the switch to activate the hoist have a lower threshold of 3V which our theoretically our 3.3V digital signal should meet however, when attempted to make the connection the 3.3V proved to be insufficient to activate the relays.

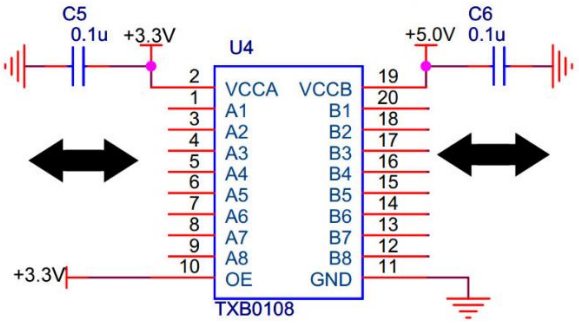


Figure 16: Level Shifter Schematics

## 2.5 User Interface

Our user interface allows the user to: set the desired hoist position, input the number of repetitions to be performed without and with help, and see the visual feedback the program generates about the performance of the exercise.

### 2.5.1 Number of Repetitions

The user is asked to input the number of repetitions he or she will attempt to perform without and with help. If a person inputs '3' in 'repetitions without help' and '2' in 'repetitions with help', the hoist will start moving at the beginning of the fourth repetition and will continue to do so until the end of the fifth repetition.

```
pi@raspberrypi:~/Documents/gorilift $ python3 gorilift.py
How many repetitions will you attempt to perform without help? 3
How many repetitions will you perform with help? 1
```

Figure 17: User Input for Repetitions

### 2.5.2 Electric Hoist Adjustment

The user is free to adjust the starting position of the hoist prior to performing the exercise. There is a built-in function where the lifter is asked in what direction he or she wants to move the hoist, and for how many seconds.

```
pi@raspberrypi:~/Documents/gorilift $ python3 move_hoist.py
In which direction do you want the hoist to move? down
For how many seconds? 2
```

Figure 18: User Input for Adjustment

### 2.5.3 Feedback

After completing all the repetitions, the user can see a chart with the time it took to complete each repetition and a graph with the inclination of the barbell through time. Additionally, the user can see the average time per repetition and average barbell inclination.

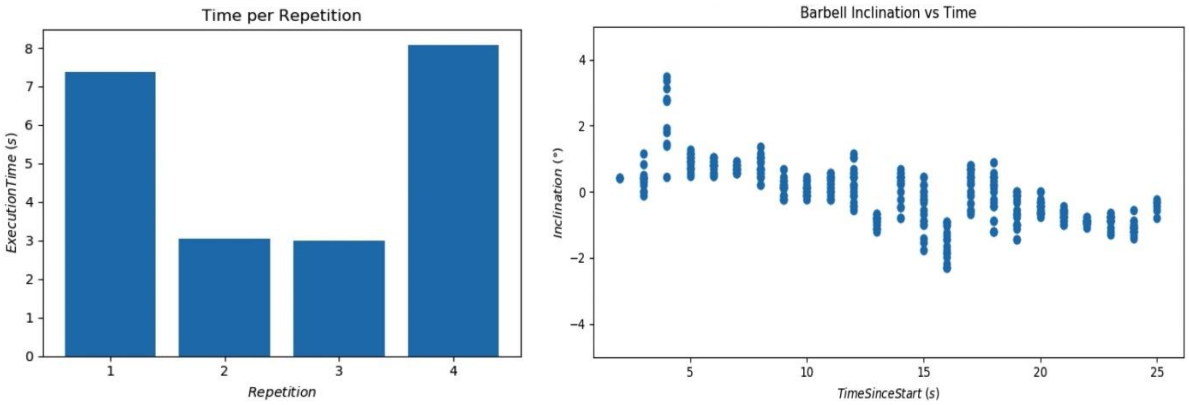


Figure 19: Feedback

# 3. Design Verification

In the following section we will dive deeper into the design development, issues encountered along the way and the final state of each of the subsystems after the resolution of such issues. One of the most important things considered when approaching these issues has been to optimize as much as possible the dimensions and space of both circuitry and wiring to make our device as portable and subtle as possible.

Refer to Appendix A to see all the requirements and verifications.

## 3.1 Mechanical subsystem

The mechanical subsystem, as mentioned earlier, is composed of the hoist, the mounting frame, and the relays. Once the hoist was purchased, the Machine Shop started working on the frame and get the hoist and the mounting frame assembled. The wires were disconnected from the manual switch, and they were connected to the relays. To do this, some wires needed to be soldier. We had to make sure that they were able to carry enough current and that they supported high voltages. Wires that supported up to 600mV were soldier to the wires that used to go to the switch, so they became longer and easier to work with.

Once the connections between the high voltage terminals of the relays and the hoist were done, we had to ensure the 60uF was discharging properly. The manual switch discharged the capacitor, and this enabled the hoist to go up and down when needed. Nevertheless, without the manual switch we needed to find a way to discharge it. We added a 200Kohm resistor in parallel with the capacitor, but it did not discharge fast enough. The hoist would let us change the direction every 8 seconds, which was impractical for our purposes. To solve this, we added 4 resistors of 100Kohm in parallel. After this, we were able to change the direction of the hoist in 1 second, which was enough.

Finally, the mechanical subsystem of the project with all the wires properly connected look like the following.

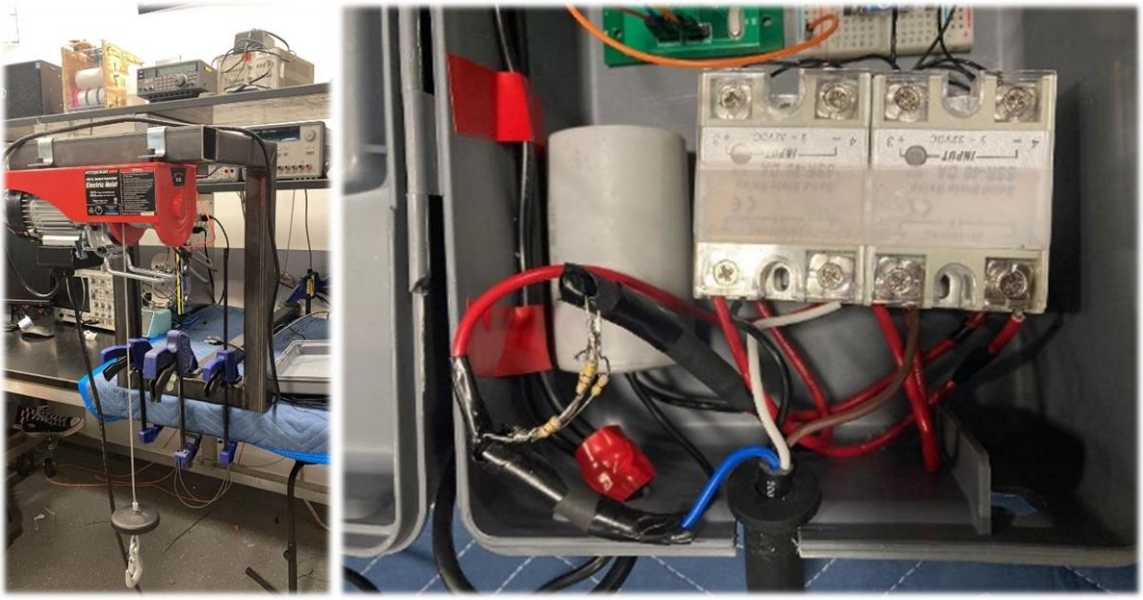

Figure 20: Mechanical Subsystem

To verify the connections were done properly, we set an input of 5V in the 2 Relays that activate the hoist to go down, and after that we did the same to activate the hoist to go up. We verified that the hoist was effectively moving, and the change of its direction was done in one second, so the mechanical subsystem was working correctly.

## 3.2 Control Subsystem

We decided to clearly divide in two parts the control subsystem, so verifications could be made simultaneously.

### 3.2.1 Hardware

The PCB design, soldering a debugging was a big part of the hardware. Once we had the PCB soldered with the ATMega328, we burned the bootloader and we run an easy program with two LEDs to verify it was working as expected. This two LEDs turning on when specified simulated the activation of hoist up and down respectively.

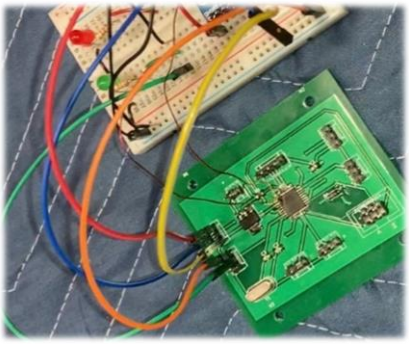

Figure 21: PCB Verification

### 3.2.2 Software

The first thing that should happen once the program starts running is a live feed from the camera. If the user is unable to see the video on the phone or computer screen, something is not right. Once the video is running, there should be a rectangle on each end of the barbell and a horizontal line across the rack, which means that the barbell was rightfully detected.

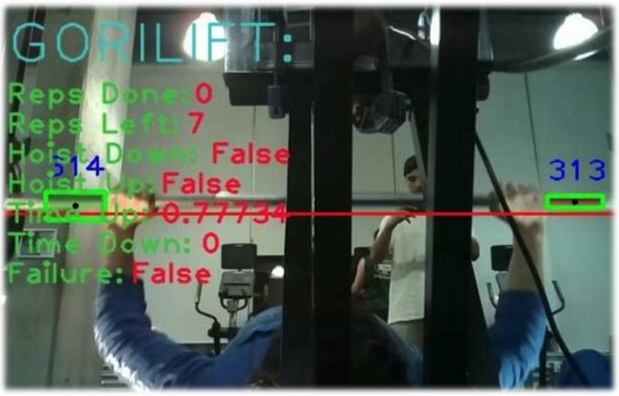


Figure 22: Repetition Tracking

If the hoist moves down 3 seconds after the un-rack, the lifter is ready to go.

Once the program is running, it is constantly retrieving the real-time coordinates of each end of the barbell. It uses the actual and previous positions of the barbell to count the repetitions and detect failure. In the following equation, redhistory[-1] and greenhistory[-1] represent the current *y* position of each end of the barbell:

$$position = \frac{redhistory[-1] - greenhistory[-1]}{2}$$

By subtracting values to the arguments of the two variables, we can have access to all the previous coordinates of the barbell. For example, if we want to get the coordinates of the barbell ten frames before, we should simply replace the previous equation with:

$$position\ ten\ frames\ before = \frac{red\square istory[-11] + greenhistory[-11]}{2}$$

By taking advantage of real-time coordinates and previous ones, we were able to devise logic for failure detection and repetition counting.

Finally, if the barbell sits still below the starting horizontal line for more than 3 seconds, the program detects failure and automatically activates the hoist to help the lifter perform the last repetitions.

## 3.3 Sensing Subsystem

Our webcam must be able to capture the barbell from end to end horizontally and follow along the vertical motion of the exercise. Different camara resolutions may present an issue as it did when we first tested our software developed by using the laptop's webcam and then we switched to the Raspberry Pi Cam module which had a higher resolution, other issues that required some coding and tweaks on our code to fix included adjusting for different lightning conditions throughout the day and ignoring background noise in the frame.

## 3.4 Power Subsystem

Regarding the AC-DC converter, we decided not to build it because of its cost and time consumption. Moreover, we were able to power the PCB with the Raspberry Pi which has some pins that provide 5V.

The level shifter was firstly connected to a fixed input of 3.3V and we verified that the output was 5V (giving a 5V reference voltage on that side). After verifying the level shifter was functioning as expected, we changed the fixed input of the 3.3V side of it for the Raspberry Pi. We checked if the other side of the

level shifter was providing 5V and it was. So finally, we integrated the Raspberry Pi and the PCB through the level shifter. We checked the voltages in the input and output, and they were the expected ones.

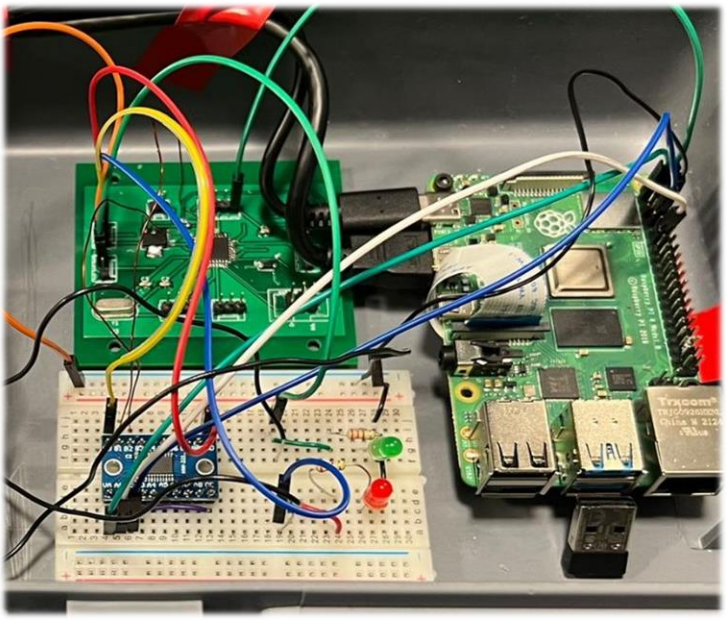

Figure 23: Raspberry Pi and PCB Connection

## 3.5 User Interface

As far as user interface is concerned, we found that the optimal solution was to enable any electronic device with Wi-Fi connection to control our Raspberry Pi through a VNC Server, which works with an app called VNC Viewer and is compatible with Windows, MacOS and Linux. With VNC Viewer, the user can input the repetitions, control the hoist, and view the feedback.

### 3.5.1 Number of Repetitions

**The user can only input integers. If not, the program throws an error.**

### 3.5.2 Electric Hoist Adjustment

The **user can only write either 'up' or 'down' when the program asks for hoist motion direction and can enter any positive float number when it asks for how many seconds.**

### 3.5.2 Feedback

**The Python program utilizes the *datetime* library to track time while the program is running. It records the time per repetition and the amount of time the program runs. It also calculates the angle of the barbell in every frame with simple trigonometry functions.**

The program **can run for indefinite time and can determine a barbell inclination within the range of [-90, 90] degrees.**

# 4. Costs

## 4.1 Parts

All the information and cost of the different parts used are gathered in Table 1

**Table 1 Parts Costs**

| Part | Manufacturer | Cost/unit ($) | Quantity | Total Cost ($) |
|---|---|---|---|---|
| Electric Hoist | Pittsburgh Automotive | 119.99 | 1 | 119.99 |
| Solid State Relays | Omega | 9.95 | 4 | 39.80 |
| ATMega328P | Atmel | 10.12 | 1 | 10.12 |
| Raspberry Pi Model 4B 2GB | Raspberry Pi Foundation | 50 | 1 | 50 |
| 22pF Capacitor | Kemet | 0.36 | 2 | 0.72 |
| Raspberry Pi Camera Arducam5 | CanaKit | 13.49 | 1 | 13.49 |
| SD Card 16GB | Raspberry Pi Foundation | 9.28 | 1 | 9.28 |
| Breadboard | Pololu | 5 | 1 | 5 |
| Logic Level Shifter | Texas Instruments | 2.5 | 1 | 2.5 |
| RPI USB-C Power Supply | CanaKit | 8.80 | 1 | 8.80 |
| **Total** | | | | **259.7** |

## 4.2 Labor

In this part we display the labor cost in Table 2.

**Table 2 Labor Cost**

| Employee | Weekly hours | Hourly pay | Weeks | Cost (USD) |
|---|---|---|---|---|
| Alejandro | 15 | 35 | 12 | 6300 |
| Carlos | 15 | 35 | 12 | 6300 |
| Eduardo | 15 | 35 | 12 | 6300 |
| **Total** | | | | 18900 |

## 4.3 Total cost

In Table 3, the total cost of the project is shown.

**Table 3 Total Cost**

| | |
|---|---|
| **Labor Cost** | 18900$ |
| **Parts Cost** | 259.7$ |
| **Total Cost** | 19159.7$ |

# 5. Conclusion

This project encompassed a successful integration of different academic fields. On the one hand, as electrical engineers, it was our task to make sure that all the physical components of our design operated within their recommended electrical requirements and that the circuitry was as simple and space efficient as possible. As far as mechanical engineering is concerned, our team did a fantastic job in making sure that the mounting frame allowed for full range of motion and that the electric hoist was stable and able to withstand the weight of the barbell. Finally, as programmers, we designed an effective algorithm that could track the barbell, count repetitions, and detect exercise failure.

## 5.1 Accomplishments

Aside from the final product, our project was accompanied by a long list of accomplishments, both on the technical and on the personal side.

Our first big accomplishment was conceiving the idea, which many would consider the hardest part; it was a result of hours of thinking, brainstorming, and good criticism. Each member of the team had something important to bring and intelligent observations about the other teammates thoughts. As a team, we slowly built a concept that was compatible with the individual ideas, and that made up for an exciting project. We conceived an idea that we were all very excited to work on since the first day.

As far as engineering accomplishments are concerned, there is a very long list. Given the nature of our project, requiring multiple physical components and testing, the first success that came to our hands was on the software side; since we did not really have to wait on any deliveries or test anything in the lab for our code, we were able to get started on the barbell tracking program very early on. It was challenging at first considering that none of us had experience with computer vision; however, with some research and practice, our team could develop a program which successfully tracks the motion of the barbell, counts repetitions, detects failure, and provides visual feedback.

Our accomplishments in the lab were a different story. We struggled a lot before we were able to get anything working. At first, our PCB design was incorrect, so we had to redesign and reorder it multiple times. On top of that, we burned a few microcontrollers in the process trying to figure out the right connections so that every single component of the circuit board was operating within the desired voltage range; the list goes on. Nonetheless, we managed to get everything working. Our first accomplishment in the lab was getting the electric hoist to behave according to commands sent from the computer (instead of the remote control that it comes with). Then, we managed to pass the desired signals from the Raspberry Pi onto the circuit board, but not from the circuit board to the hoist. Finally, after countless failures, we got the last two parts to communicate by simply reevaluating which pins from the microcontroller to use as outputs.

We accomplished a system that can track the barbell, count repetitions, detect failure, and activate the hoist in precise moments to help lifters minimize injuries and maximize results from the bench press.

## 5.2 Uncertainties

One of our greatest uncertainties concerned the PCB design and implementation. We spent a long time trying to fix a PCB that had no bugs. We realized soon enough, though, that there was a very specific way to load C++ programs onto the microcontroller, and that we had included all the necessary connections; still, we did investigate possible worst-case scenarios and considered choosing other alternatives given that our PCB was not responding.

At the beginning of the project, we also had multiple uncertainties concerning the high-level functionality of our project. Did we want our product to be capable of counting repetitions and detecting failure? Did we want to limit it to just one? Thankfully, with early software development, we realized that offering both options was possible, and we sure did.

Finally, now that we have everything working, we are not sure about how we would implement our product in real life gyms. We have been putting a lot of thought about the way in which we would sell our idea: a separate component compatible with any bench? The entire bench-helper combo? Both have their pros and cons, but we will know for sure once we look more into what best fits our interests and the gyms'.

## 5.3 Ethical considerations

Based on the Code of Ethics [5], our one true ethical consideration has to do with the fact that our product's purpose is, ultimately, to prevent injuries. That is a very delicate goal given that injuries can occur in countless ways, and we cannot guarantee that the users of the Bench Press Smart Helper will never get injured while using it. However, we can and have to guarantee that the mounting frame is stable enough, when performing the exercise within the allowed range of weight by the weight, so that it does not pose a health risk and a concerning factor for potential accidents if they were to occur.

## 5.4 Future work

As mentioned before, our goal is to implement our assessment and protection systems in fitness centers. For that, we will need to optimize the spatial design, and decide on whether we want to sell it as a separate unit that can be incorporated into any bench, or if we want to create a product that includes a bench among its features. The first one would probably be a lot more cost efficient, but the second one could probably have more selling features. That is something we will investigate in the future.

Furthermore, we do not want to limit ourselves to providing help for the bench press exclusively. We want to extend our design and make it compatible with squats and other compound barbell movements. We believe that given our discoveries and accomplishments throughout this project, that it will not pose a serious challenge.

# References


[1] Victor Bengtsson, Lars Berglund and Ulkira Aasa: "*Narrative review of injuries in powerlifting with special reference to their association to the squat, bench press and deadlift.*" Us National Library of Medicine National Institutes of Health, 2018. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6059276/#:~:text=Earlier%20studies%20that%20have%20reported,12%25%E2%80%9331%25%20to%20the

[2] Chucky, Amelie. "*How to Hack an Electric Hoist (AC Motor).*" Instructables. Instructables, October 25, 2017. https://www.instructables.com/How-to-hack-an-Electric-Hoist-AC-motor/.

[3] Bengtsson, Victor, Lars Berglund, and Ulrika Aasa. 2022. " Injuries In Powerlifting With Special Reference To Their Association To The Squat, Bench Press And Deadlift". BMJ Journals.

https://bmjopensem.bmj.com/content/4/1/e000382.

[4] Higley, Samantha. 2022. "*Campus Recreation | The Pros And Cons Of Smith Machines*". Campusrecreation.Wvu.Edu.

https://campusrecreation.wvu.edu/news/health-and-wellbeing/2021/03/23/the-pros-and-cons-of-smith-machines

[5] "*ACM Code of Ethics and Professional Conduct*." Developed by the ACM Code 2018 Task Force

https://www.acm.org/code-of-ethics

[6] ''*Image Processing in Opencv*." OpenCV. Accessed May 4, 2022. https://docs.opencv.org/3.4/d2/d96/tutorial_py_table_of_contents_imgproc.html.

## Appendix A Requirement and Verification Table

**REQUIREMENTS AND VERIFICATIONS:**

Control subsystem (hardware)

| REQUIREMENT | VERIFICATION |
|---|---|
| 1. The 3.3V to 5V and 5V to 3.3V level shifter will have a tolerance of 15%. The output of the RaspberryPi pins must be between 2.97V and 3.63V. At the same time, the output voltage of the level shifter must be between 4.25 V and 5.75V. | 1.a. Place probes in the output signal of the Raspberry Pi and ground to check that 3.3 V are coming out.<br>1.b Check the other side of the level shifter while having it connected properly to a 5V source and the ground. There should be 5V+15%. |
| 2. The voltage at PCB microcontrollerpins should be 5V +0.3V. | 2.a Place probes on PCB ground while providing voltage to the input of the PCB and connecting it to ground as well. Voltage coming in the Microcontroller must be 5V+15%.<br>2.b Create an easy programming test that sets one of the pins used as an output digital high.<br>2.c Make sure that the output is 5V+15% by using the probes between the header pin of the output pin and the header pin of the ground of the PCB |
| 3. Relays should be capable of moving the electric hoist wire up and down according to the signalshoist ‘up’ and ‘down’. | 3.a Create a program that sets two output digital pins of the microcontroller. One will be the one that activates the hoist down and the other one activates up.<br>3.b By using a delay of 2 seconds turn the down signal off and on consequently (so it does not hit the floor)<br>3.c Connect the two signals to two relays each and connect the hoist to the wall plug. |
| 4. The microcontroller is able to establish a direct connection with the hoist and activate it when desiredand for as long as the user wants. | 4.a. We will run a simple code setting the pin connected to the microcontroller to high (up and down directions voltage).<br>4.b. The electric hoist should start working. |

Control subsystem (software)

| REQUIREMENT | VERIFICATION |
|---|---|
| 1. The first thing to show up in the terminal after running the program should be a message asking for the amount of repetitions to be attempted without help, and right after another message asking for the repetitions to be performed with help. | 1.a. Clicking enter on the command python3 gorilift.py, and seeing no errors.<br><br>1.b. If the user does not input an integer, the program will throw an error and the previous command should be entered again. |
| 2. After inputting the total number of repetitions to be performed, a live stream from the webcam should be projected onto the screen. | 2. The camera shows live video, and some fixed (Repetitions, Hoist Up, Hoist Down, etc.) text appears on the screen as well |
| 3. The program detects the two ends of the barbell and draws a horizontal line at the starting point. | 3.a. There are two rectangles on the screen, one surrounding the green end of the barbell, the other one surrounding the red end of the barbell.<br><br>3.b. There is a red horizontal line located in the average of the first y coordinates the program detects for the red and green ends of the barbell. |
| 4. After moving the barbell above the starting point and leaving it still for 3 seconds, the motor should turn on and the wire should move downwards. | 4. Once the program starts, move the barbell up and leave it still for 3 seconds. The Hoist Down signal should become true. |
| 5. If the barbell sits below the horizontal line at the same position for 3 seconds, the failure signal will activate and turn the motor on in the upward direction until it reaches maximum height. | 5. Once the user is not able to perform any more repetitions, leave the barbell below the red line in roughly the same place for 3 seconds, and the program will turn the failure signal on. |

Sensing subsystem

| REQUIREMENT | VERIFICATION |
|---|---|
| 1. The webcam should be able to capture the barbell from end to end horizontally and follow along the movement vertically. | 1. Run the code and check the color is tracked correctly by moving the barbell. |

Mechanical subsystem

| REQUIREMENT | VERIFICATION |
|---|---|
| 1. The Mounting Frame is able to hold a 20-pound barbell without tilting nor obstructing the lifter's performance. Calculated with a 1.5 security coefficient. | 1. Holding a 20-pound barbell in the hoist hook and running a code that moves up and down the barbell |
| 2. The electric hoist must be able to lift the weight accurately without disturbing the user's performance. | 2. Prepare for doing the exercise |