****

**ECE 387**

**Final Project Report**

By Alex Onkst

*College of Engineering and Computing, Miami University, Oxford, OH, 45056*

**Smart Bench Press**

**Dr Jamieson**

May 17, 2019

**Introduction**

For my final individual project, I decided that I wanted to design a “smart bench-press,” wear a user can bench while also receiving various information from the bench press before, during, and after their workout. The idea stems from a personal issue that I sometimes have with bench presses when I am working out by myself. Sometimes I forget how many reps I’ve done, sometimes I catch myself favoring my right side rather than performing a well-balanced, proper rep, and I wanted to design something that could remedy these issues.

Another reason I decided to take on this task was because it would allow me to explore a real-world application of my midterm chip, the Infrared Distance Sensor. I wanted to utilize what I had learned from the midterm chip and implement it into an actual complex embedded system, and the idea of using the IR sensor to detect reps on a bench press seemed like a perfect application. Throughout this report I will discuss my design process: what I wanted to do, what I was able to do, and what challenges I came across along the way.

**Project Proposal**

When brainstorming what I actually wanted to implement in this design, several different goals came to mind, which I discussed in my project proposal. The main goal was to have a bench press that could count my reps and tell the user when a rep was not properly balanced. Once these were successful, I wanted to see what other tasks I could have the bench press perform, which are listed below along with the original goals:

* Use 2 IR sensors (one above the other) to count reps
* Use force-sensitive resistors to detect that the bar has been removed from its resting position, and begin counting the reps
* Use other IR sensors to detect how straight the rep is
* Give immediate feedback to the user when an improper rep is observed
* Display the reps on a 7-segment display or an LCD.
* Have the user enter the weight of their rep on an LCD (i.e., if their reps are 10 or 100 lbs)
* Take this user input and the number of reps and converting it to some sort of score to display on the LCD
* Implement an “emergency spotter” system that utilizes 2 linear actuators to protrude from the bench press for the user to place the bar if the bar has been under the bottom IR sensor for too long.

As with most proposals, successfully implementing all of the things I wanted to do proved to be a little ambitious. Nevertheless, I wanted to see what aspects of this project I could accomplish with the time and resources that I had. The formal written proposal can be found in the [Project Proposal](https://github.com/AlexOnkst/ECE-387-Final/wiki/Project-Proposal) page on the GitHub wiki.

**Design Process**

The first thing I needed to do for my project was to design a basic frame of a bench press that could also hold all of the components and wiring of my embedded system. Therefore, I made a simple PVC frame of what could be considered a typical bench press, and then placed a pegboard into the apparatus. I chose PVC for the frame because I felt that it was cheap and sturdy enough as a prototype, and the pegboard allowed for wires to and components to be connected with little to no hassle. I chose pegboard because I felt that it could be used to easily hide the wiring of the components out of sight while also allowing easy access to the wiring of each component. This frame can be seen below:

Next I wanted to make sure that the force resistors could detect when the bar was removed from its resting position. I ended up placing a resistor on both of the bar rests, and then connected these to the Arduino. After some simple experimentation, I discovered that the resistors worked properly and could easily detect when the bar is removed. These resistors can be seen below:

Afterward, I wanted to add the two IR distance sensors used for counting the reps onto the pegboard. This was simply done by taping them onto the pegboard and feeding their wiring into the holes to the back of the board, where the Arduino and most of the wiring is found. The coding for these two sensors themselves did not prove too troublesome, but as discussed later in this report, there were some issues along the way incorporating this code with other components. Luckily, I was successful in getting them to properly count reps under a few conditions.

The first condition was that during a rep, the bar was held fairly close to the distance sensor when the bar moved up and down. This was because in my code I set a threshold reading for the distance sensors, since the Arduino is constantly reading the voltages from the sensors. Therefore, if the bar was held too far away from the pegboard, the Arduino would not count the reps.

The second condition was that the Arduino be restarted every time you wanted the machine to count the reps. Originally I wanted the loop to start over every time the bar was placed back in the resting position, but I was unsuccessful in getting the code to work. Proud