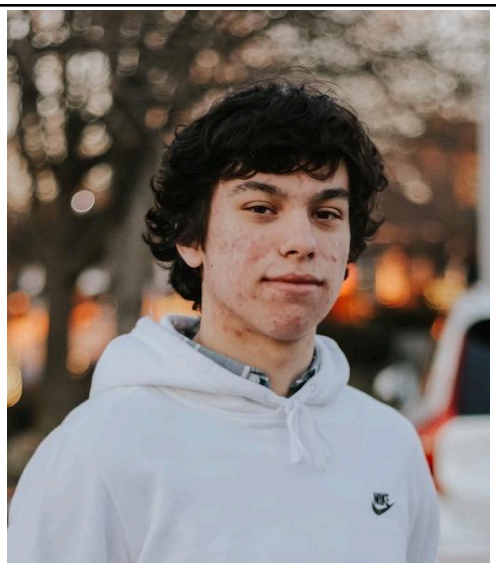
Final Report: INDY5 Helmet Project

CS4850, Spring 2024
Professor Perry
April 27th, 2024

Patrick O'Connell



Brian Bergmann

Project Team

Roles	Name	Major responsibilities	Cell Phone / Alt Email
Team leader	Patrick O'Connell	Developer	240-812-2237 Poconnell20055@icloud.com
Team members	Brian Bergmann	Documentation	412-735-9704 Brian.bergmann144@gmail.com
Advisor / Instructor	Sharon Perry	Facilitate project progress; advise on project planning and management.	770-329-3895

Team Website Link	Team INDY5 - Helmet - Home (indy5-2024.github.io)
Final Report Presentation Video	TO DO
Source Code Link	https://github.com/INDY5-2024/INDY5

Total Lines of Code: Total Number of Components:

1. Introduction

The Indy5 Helmet project aimed to develop a sensor integrated into sports helmets, primarily football helmets, to track high-stress impacts and promptly notify coaches or medical personnel on the sideline. This final report encapsulates the journey of conceptualization, design, development, and implementation of the Indy5 Helmet system.

1.1 Overview

The system utilizes Raspberry Pi as the central processing unit along with an Adafruit Accelerometer to detect impacts. The original plan was for Twilio to be employed for real-time communication, allowing the system to send SMS alerts to the sideline upon detecting a significant impact. However, due to variables out of our control, we were not approved to use Twilio, so we had to shift gears to use SMTP in a Python file.

1.2 Project Goals

The primary goal was to create a device capable of quickly and accurately alerting the sideline when a major hit occurs, potentially indicating a concussion. The device had to work wirelessly and send alerts to the appropriate personnel immediately after impact.

1.3 Definitions and Acronyms

- Raspberry Pi: A microcontroller/minicomputer used to run the Raspbian OS.
- Adafruit Accelerometer: The accelerometer employed to track force on the helmet.
- Twilio: Provides programmable communication tools for sending and receiving text messages.
- SMTP: Simple Mail Transfer Protocol

1.4 Assumptions

Assumptions included the continuous connection to Wi-Fi for proper functioning and the adjustment of impact thresholds to differentiate between regular hits and concussion-indicative impacts. It is also assumed that this is just a proof of concept. Our prototype could in no way be used in a real football game.

2. Design Constraints

Environmental factors such as severe weather and water damage posed constraints. The system was designed to track single significant hits, acknowledging that concussions could also result from cumulative impacts. We also had physical constraints for the physical prototype. We had to have some sort of battery to power the raspberry pi. We used a basic Portable Charger that would sit in our pockets during the demonstration.

2.1 Environment

Originally, the system's environment involved a Raspberry Pi with an accelerometer attached to the helmet, utilizing Twilio for communication. Stable Wi-Fi connection was necessary for message transmission. Since switching to an email, the same still applies in terms of a Wi-Fi

Connection. However, now it is necessary to also have a sender email address, target email address, and the password and Secret ID for the sender email.

2.2 User Characteristics

The target users included football players susceptible to concussions, particularly offensive/defensive linemen, quarterbacks, centers, wide receivers, and running backs.

2.3 System

The system's processing capability was limited by the Raspberry Pi's computing potential, but sufficient for the intended purpose.

3. Functional Requirements

The system was required to detect hits, calculate the force received, determine injury potential, and send messages to the sideline accordingly.

4. Non-Functional Requirements

4.1 Security:

As the system relied on Wi-Fi for communication, the risk of data corruption during transmission was acknowledged. However, no private information was stored on the device, minimizing security risks.

4.2 Capacity:

The device must handle repetitive message transmission and detect multiple hits effectively.

4.3 Usability:

The device was designed to be user-friendly, catering to users with varying levels of technological expertise.

4.4 Other:

Durability was crucial, considering the device's integration with the helmet and exposure to high-impact forces during gameplay. We purchased an aluminum case for the raspberry pi to ensure no damage would be done to it during demonstrations.

5. External Interface Requirements

5.1 User Interface Requirements:

Messages sent to the sideline were designed to be readable and understandable, providing essential information about the impact.

5.2 Hardware Interface Requirements:

The system required seamless communication between the accelerometer and Raspberry Pi for accurate impact detection.

5.3 Software Interface Requirements:

Originally, the Raspberry Pi must interface with Twilio to send SMS alerts in a readable format. Now using SMTP, we must have the Raspberry Pi communicate with SMTP to send an email in a readable format to users.

5.4 Communication Interface Requirements:

Originally, Twilio served as the communication interface for transmitting alerts to the sideline. We switched to SMTP which served as our replacement interface for transmitting messages.

6. Narrative

In order to design a helmet that had the ability to accomplish the goals that we were looking for we needed to start with a base design and plan. The first thing that we did was source a helmet that would be used as the base for all the designing of the components. We were able to find an old replica NFL helmet that would be useful as an example prototype for our project. The next task we had was to develop a plan for what system we were going to use for our project. Due to prior knowledge within our group we decided to go with the base component of our system being run off of a raspberry PI. The reason we chose the PI was due to the ease of use of the system and because of the small size it would allow for us to easily conceal the computer within the helmet.

After sourcing the raspberry PI we needed a way for the computer to actually detect the impact received from the hit. To accomplish this we were able to source and locate an adafruit accelerometer. An accelerometer is a small instrument that can be used to detect acceleration. We are able to use this within the helmet to calculate the force that is applied to the helmet based upon the acceleration that is received. To simplify this process we were able to use the is tapped function that is built into the adafruit library. This allows for us to detect when the helmet is tapped or hit and gives us control over how much force is required to alert the system of the impact.

Additionally we were able to source additional parts for the raspberry PI such as a batter pack that was used in order to make the system more portable and not require a power outlet. Originally we planned to used a small lithium ion battery that would be concealable within the helmet, but due to pinout restrictions on the raspberry PI we were unable to use that, instead we opted to use a larger battery pack that would not be able to be concealed within the helmet but rather be attached via cable. Although this was not ideal it did allow for the system to have a larger battery life due to the larger size of the battery. We also sourced a small case with a built in fan for the raspberry PI to allow for it to be more protected within the helmet and to stay cool after running for long periods of time.

The next task in the design process for our group was creating a program that would be able to run automatically on the raspberry PI on bootup that would allow it to receive data from impacts and connect to the necessary personnel on the sideline. The first component of that was to create a python script on the raspberry PI. We created a script that when started would run on an infinite loop, every second it would check to see if there was an impact or tapped received by the accelerometer. Upon impact it would send a message to the sideline but still keep running to see if another impact was

received. We also attempted to have the script run on startup by using crontab however we were unable to successfully implement it.

After getting the python script to run we had to determine a way to get the message received by the personnel on the sidelines. Although we did have the option of creating our own app, we opted to simply communicate with the necessary people via communication tools they already have to reduce the complexity of the system as well as make it more user friendly for the client. Our original plan was to communicate to the sideline via text. Our group has had prior experience using twilio as an automated text messaging service, however as of January 2024 they changed the process required to get a number used to text from. Instead of easily being able to pay for and acquire a number, now they can only be sent out to registered businesses, not individuals. We attempted to get approved and were able to successfully implement Twilio's API into our project, however were ultimately denied for not being a legitimate business.

Since we were not able to use Twilio's text messaging features we had to pivot late into our project development cycle. We opted to switch instead of communicating via text to communicate via email since there were a lot less hoops we had to jump through. In order to effectively communicate via email was a simple and easy process. In our python script we were able to set up a secure connection to a SMTP server via gmail. Then using a dedicated gmail account we were able to send an email to whatever address we would like and whatever message we would like. We set our script up to send us an email informing us that a player has been hit on the field and may have received a concussion.

The final step of our project was combining all of the pieces together and creating a full prototype design helmet. We were able to modify the helmet and secure the raspberry PI and accelerator inside the helmet. We then were able to connect the PI to our power source and run the python script. After running the code we were able to "hit" the helmet hard enough to trigger the accelerometer to determine a hit. After the hit was received a message was then sent to our email account informing us that the hit took place.

7. Challenges and Risk Assessments

7.1 Challenges:

Throughout our development cycle we encountered multiple problems that were unexpected but we were able to figure most of them out to create a functioning product at the end. The first problem we ran into was using a battery. We attempted to use a smaller battery attracted to the

raspberry PI via pinouts to reduce space and make the overall computer require less parts. However after installing we realized that this would require the same pinouts to be used as the accelerometer and they would both not work at the same time so we had to scrap that idea.

Another problem that we ran into during our development was the attempt for the python script to run on the bootup of the raspberry PI so that peripherals were not required. We followed multiple different guides and techniques to get the commands to run such as crontab and a bash script, however they did not seem to cooperate well with the adafruit accelerometer so ultimately we were unable to get the scripts to run on startup and instead will require them to be ran via a console command.

The final and most impactful challenge that we ran into was the change in Twilio's policies. We were expecting to use Twilio as our SMS messaging provider to send a text message to the sideline, we had their API fully encoded into our code and had it working in a test environment. However they had recently changed their requirements to get a number to text from so instead of just paying for the number they could only be given to authorized business. Due to this we had to pivot the way that we communicated to the sidelines from the raspberry PI. We opted to use an SMTP server to send an email from a dedicated email account. This is not the ideal way to quickly contact someone on the sideline but could still be used to receive communication from the computer.

7.2 Risk Assessments:

With our device there are multiple risks that are involved that we must be aware of. First of all since the raspberry PI must send an email to inform a coach of the hit it must require a stable internet connection, on football fields this is not a given and if the device were to lose internet it would no longer function. Another risk we must consider would be the physical risks involved. Since the computer is housed inside of a helmet that is often struck and hit hard, there is a chance that the computer could be hit as well resulting in it breaking and no longer functioning. Additionally, since the computer is outside we must be mindful of risk from the weather or other elements that could contaminate and break the system.

8. Test Plan and Test Report

8.1 Test Plan:

Our test plan is pretty straight forward, we simply run the python script on the computer and determine if the helmet can detect an impact. We want to make sure that the system only sends a message after a hit has occurred. By testing, it also allows us to determine a good threshold for the amount of acceleration needed by force in order to register a hit that could provide a concussion. For testing purposes we turned the threshold down to ensure that the messages were being sent after an impact.

8.2 Test Report:

After conducting our tests we determined that the helmet is working as intended. Upon impact the helmet is able to detect the hit and successfully send a email to the intended recipient. We have found that the time the email takes to send is quite slow ~30 seconds, but the information sent is accurate.

9. Conclusion

Overall throughout this process we certainly encountered a lot of hurdles to get through, the end product was similar to the original plan we had however, there were quite a few changes we had to make along the way. We, however, were able to adapt to these changes and create an end product that shows off the potential that this technology has. There is certainly a lot of room for improvement with our project and I'm sure with more time and people we would be able to take the project further.

Appendices

- Reference: Raspberry Pi Forums (Wiring Information)
- INDY5 - Project Plan
- INDY5 - SDS Document
- INDY5 - Requirements Document