**Helmet Software Design**

|  |  |
| --- | --- |
| 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](http://Poconnell20055@icloud.com/) |
| Team members | Brian Bergmann | Documentation | 412-735-9704  [Brian.bergmann144@gmail.com](http://Brian.bergmann144@gmail.com/) |
| Advisor / Instructor | Sharon Perry | Facilitate project progress; advise on project planning and management. | 770-329-3895 |

**Table of Contents**

**1.** **Introduction** **3**

1.1. Document Outline 3

1.2. Document Description 3

[*1.2.1.* *Introduction* *4*](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?new=1&ui=en-us&rs=en-us&wopisrc=https%3A%2F%2Fkennesawedu.sharepoint.com%2Fsites%2FTeam-4850-02Spr24-Indy--5Helmet%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F71cba8518116439186b79c81d190337a&wdenableroaming=1&mscc=1&hid=c918b1fe-6776-420f-af81-38f003166cb7.0&uih=teams&uiembed=1&wdlcid=en-us&jsapi=1&jsapiver=v2&corrid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&usid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&newsession=1&sftc=1&uihit=UnifiedUiHostTeams&muv=v1&accloop=1&sdr=6&scnd=1&sat=1&rat=1&sams=1&mtf=1&sfp=1&halh=1&hch=1&hmh=1&hwfh=1&hsth=1&sih=1&unh=1&onw=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fwww.microsoft365.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Normal&wdorigin=TEAMS-ELECTRON.teamsSdk.openFilePreview&wdhostclicktime=1707848653171&wdredirectionreason=Unified_SingleFlush#_Toc166398923)

*1.2.2.* *System Overview* *4*

**2.** **Design Considerations** **6**

2.1. Assumptions and Dependencies 6

2.2. General Constraints 6

[2.3. Goals and Guidelines 7](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?new=1&ui=en-us&rs=en-us&wopisrc=https%3A%2F%2Fkennesawedu.sharepoint.com%2Fsites%2FTeam-4850-02Spr24-Indy--5Helmet%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F71cba8518116439186b79c81d190337a&wdenableroaming=1&mscc=1&hid=c918b1fe-6776-420f-af81-38f003166cb7.0&uih=teams&uiembed=1&wdlcid=en-us&jsapi=1&jsapiver=v2&corrid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&usid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&newsession=1&sftc=1&uihit=UnifiedUiHostTeams&muv=v1&accloop=1&sdr=6&scnd=1&sat=1&rat=1&sams=1&mtf=1&sfp=1&halh=1&hch=1&hmh=1&hwfh=1&hsth=1&sih=1&unh=1&onw=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fwww.microsoft365.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Normal&wdorigin=TEAMS-ELECTRON.teamsSdk.openFilePreview&wdhostclicktime=1707848653171&wdredirectionreason=Unified_SingleFlush#_Toc166398928)

[**3.** **Architectural Strategies** **8**](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?new=1&ui=en-us&rs=en-us&wopisrc=https%3A%2F%2Fkennesawedu.sharepoint.com%2Fsites%2FTeam-4850-02Spr24-Indy--5Helmet%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F71cba8518116439186b79c81d190337a&wdenableroaming=1&mscc=1&hid=c918b1fe-6776-420f-af81-38f003166cb7.0&uih=teams&uiembed=1&wdlcid=en-us&jsapi=1&jsapiver=v2&corrid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&usid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&newsession=1&sftc=1&uihit=UnifiedUiHostTeams&muv=v1&accloop=1&sdr=6&scnd=1&sat=1&rat=1&sams=1&mtf=1&sfp=1&halh=1&hch=1&hmh=1&hwfh=1&hsth=1&sih=1&unh=1&onw=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fwww.microsoft365.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Normal&wdorigin=TEAMS-ELECTRON.teamsSdk.openFilePreview&wdhostclicktime=1707848653171&wdredirectionreason=Unified_SingleFlush#_Toc166398930)

[**4.** **System Architecture** **10**](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?new=1&ui=en-us&rs=en-us&wopisrc=https%3A%2F%2Fkennesawedu.sharepoint.com%2Fsites%2FTeam-4850-02Spr24-Indy--5Helmet%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F71cba8518116439186b79c81d190337a&wdenableroaming=1&mscc=1&hid=c918b1fe-6776-420f-af81-38f003166cb7.0&uih=teams&uiembed=1&wdlcid=en-us&jsapi=1&jsapiver=v2&corrid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&usid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&newsession=1&sftc=1&uihit=UnifiedUiHostTeams&muv=v1&accloop=1&sdr=6&scnd=1&sat=1&rat=1&sams=1&mtf=1&sfp=1&halh=1&hch=1&hmh=1&hwfh=1&hsth=1&sih=1&unh=1&onw=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fwww.microsoft365.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Normal&wdorigin=TEAMS-ELECTRON.teamsSdk.openFilePreview&wdhostclicktime=1707848653171&wdredirectionreason=Unified_SingleFlush#_Toc166398931)

[4.1. Subsystem Architecture 10](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?new=1&ui=en-us&rs=en-us&wopisrc=https%3A%2F%2Fkennesawedu.sharepoint.com%2Fsites%2FTeam-4850-02Spr24-Indy--5Helmet%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F71cba8518116439186b79c81d190337a&wdenableroaming=1&mscc=1&hid=c918b1fe-6776-420f-af81-38f003166cb7.0&uih=teams&uiembed=1&wdlcid=en-us&jsapi=1&jsapiver=v2&corrid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&usid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&newsession=1&sftc=1&uihit=UnifiedUiHostTeams&muv=v1&accloop=1&sdr=6&scnd=1&sat=1&rat=1&sams=1&mtf=1&sfp=1&halh=1&hch=1&hmh=1&hwfh=1&hsth=1&sih=1&unh=1&onw=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fwww.microsoft365.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Normal&wdorigin=TEAMS-ELECTRON.teamsSdk.openFilePreview&wdhostclicktime=1707848653171&wdredirectionreason=Unified_SingleFlush#_Toc166398932)

[**5.** **Policies and Tactics** **11**](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?new=1&ui=en-us&rs=en-us&wopisrc=https%3A%2F%2Fkennesawedu.sharepoint.com%2Fsites%2FTeam-4850-02Spr24-Indy--5Helmet%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F71cba8518116439186b79c81d190337a&wdenableroaming=1&mscc=1&hid=c918b1fe-6776-420f-af81-38f003166cb7.0&uih=teams&uiembed=1&wdlcid=en-us&jsapi=1&jsapiver=v2&corrid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&usid=ded94b4c-da75-4e4e-ba31-4e5eb76deadc&newsession=1&sftc=1&uihit=UnifiedUiHostTeams&muv=v1&accloop=1&sdr=6&scnd=1&sat=1&rat=1&sams=1&mtf=1&sfp=1&halh=1&hch=1&hmh=1&hwfh=1&hsth=1&sih=1&unh=1&onw=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fwww.microsoft365.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Normal&wdorigin=TEAMS-ELECTRON.teamsSdk.openFilePreview&wdhostclicktime=1707848653171&wdredirectionreason=Unified_SingleFlush#_Toc166398933)

**6.** **Detailed System Design** **12**

6.1. Classification 13

6.2. Definition 13

6.3. Responsibilities 13

6.4. Constraints 13

6.5. Composition 13

6.6. Uses/Interactions 13

6.7. Resources 13

6.8. Processing 14

6.9. Detailed Subsystem Design 14

**7.** **Glossary** **14**

**8.** **Bibliography** **15**

# 1. Introduction

This document is used to outline the overall goals for the software designs for our project. The SDS is broken down into multiple sections based upon the Design Considerations, Architectural Strategies, Policies and Tactics, and Detailed System Design.

## *1.1.* *Document Outline*

* Introduction
* System Overview
* Design Considerations
  + Assumptions and Dependencies
  + General Constraints
  + Goals and Guidelines
  + Development Methods
* Architectural Strategies
* System Architecture
* Policies and Tactics
* Detailed System Design
* Glossary
* Bibliography

## *1.2.* *Document Description*

Here is the description of the contents (by section and subsection) of the proposed template for software design specifications:

### 1.2.1. Introduction

This document serves as a comprehensive guide to the software design for the Indy5 Helmet. Its primary purpose is to outline the architectural and detailed design decisions made throughout the development process, offering valuable insights into the structure and functionality of the software. Intended for a diverse audience including software developers, architects, project managers, and other stakeholders involved in the Indy5 Helmet project, this document identifies the system/product using applicable names or version numbers. References to related documents, both companion and prerequisite, are provided, offering background and context for a deeper understanding. Additionally, it sets the stage for future documents resulting from this design, such as a test plan or development plan. To enhance clarity, important terms, acronyms, and abbreviations are defined in the glossary. The document's contents encompass design considerations, architectural strategies, system architecture, policies, tactics, and detailed system design, providing a structured outline to grasp the Indy5 Helmet project.

### 1.2.2. System Overview

The Helmet project is dedicated to integrating cutting-edge technologies into a smart helmet system crafted for monitoring and assessing the force experienced by football players during gameplay. A pivotal component of this system is the accelerometer embedded within the helmet, capturing real-time impact data and precisely calculating the force exerted on the player's head. The Raspberry Pi serves as the central processing unit, managing information and employing Twilio, a programmable communication tool, to send and receive text messages.

## Functionality:

The core functionality revolves around the embedded accelerometer, capturing instantaneous impact data during football gameplay. This data is crucial for calculating the force exerted on the player's head, providing real-time insights into potential concussive events.

## Design Approach:

Adopting a modular design approach, the system comprises key components, including the Raspberry Pi-based Processing Module and the Communication Platform utilizing Twilio. This design strategy ensures adaptability, scalability, and seamless integration of features without the need for a separate data collection module.

## Organizational Structure:

The system is organized around vital components:

- Processing Module (Raspberry Pi):

- Responsible for processing impact data to determine the force exerted on the player's head.

- Communication Platform (Twilio):

- Utilizes Twilio for sending immediate text message alerts to the coach when a potential concussion-indicative force threshold is exceeded.

## Interactions Between Components:

Seamless interaction occurs between the Processing Module and the Communication Platform (Twilio), ensuring accurate and timely transfer of impact data. The Communication Platform acts promptly, leveraging Twilio to send text message alerts to the coach when the calculated force surpasses the predefined threshold.

Scalability and Extensibility:

The modular design facilitates scalability, allowing the addition of features or components to adapt to evolving player safety requirements and technological advancements, while Twilio ensures a reliable and versatile communication channel.

*Usability Considerations:*

Prioritizing user experience, the Communication Platform (Twilio) provides instantaneous, actionable information to coaching staff, enhancing player safety through immediate notifications. The Helmet project establishes itself as a valuable tool for long-term monitoring and analysis, contributing to advanced concussion prevention strategies in sports with the secure and efficient text messaging capabilities of Twilio.

# 2. Design Considerations

This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution.

## *2.1.* *Assumptions and Dependencies*

For the helmet to function and send messages as intended it needs to be connected to a Wi-Fi source constantly, and since we don't want to wear the helmet and then hit it hard enough to give us concussions, we will be lowering the threshold for the code to register a large hit and send a text message, however with just changing a few numbers in the code, it will work for a concussion level hit.

## *2.2.* *General Constraints*

## Time Constraints:

The project is subject to a predefined timeline, necessitating efficient and focused design efforts to meet milestones and deadlines. Iterative design processes should be streamlined to ensure timely completion, with periodic reviews to assess progress against the project schedule.

*Budgetary Constraints:*

Financial considerations play a pivotal role in the design, development, and implementation phases. The project must adhere to a predefined budget, necessitating careful resource allocation and cost-effective decision-making throughout the design process.

## Hardware and Software Constraints:

The design must operate within the constraints of available hardware and software resources. This includes considerations for the capabilities and limitations of the Raspberry Pi and Twilio integration, ensuring compatibility and optimal performance.

Environmental Factors:

External environmental factors, such as the potential impact of weather conditions during gameplay, pose challenges. Design considerations must account for the robustness and resilience of the smart helmet system in varying environmental conditions.

## *2.3.* *Goals and Guidelines*

The Helmet project adheres to the Keep It Simple, Stupid (KISS) strategy, guiding the design process with a focus on simplicity, clarity, and streamlined functionality. The goals and guidelines for the project are outlined as follows:

## Simplicity in Design:

The primary goal is to achieve a straightforward and uncomplicated design for the smart helmet system. Emphasizing simplicity ensures ease of understanding, reduces potential points of failure, and enhances overall system reliability.

*User-Friendly Operation:*

Guided by the KISS strategy, the design prioritizes a user-friendly interface and operation. The smart helmet system should be intuitive, allowing both players and coaching staff to interact with the technology effortlessly.

## Minimized Complexity in Communication:

The communication between the Raspberry Pi and Twilio is designed to be clear, concise, and efficient. Avoiding unnecessary complexities in the communication protocol ensures reliable and timely transmission of text message alerts to coaching staff.

## Optimized for Quick Deployment:

Following the KISS strategy, the design prioritizes a system that can be swiftly deployed and integrated into existing football helmets. Minimizing complexity expedites the implementation process, allowing for timely testing and usage on the field.

## Scalability with Simplicity:

While accommodating potential future enhancements, the design maintains a commitment to simplicity. The system should be scalable without introducing unnecessary intricacies, enabling seamless integration of additional features in line with evolving safety requirements.

By adhering to these goals and guidelines, the Helmet project ensures that the smart helmet system is not only effective in monitoring and assessing player safety but is also approachable, user-friendly, and adaptable for future advancements. The KISS strategy acts as a guiding principle, emphasizing simplicity without compromising functionality.

# 3. Architectural Strategies

The Helmet project employs specific design decisions and strategies that significantly influence the overall organization of the system and its higher-level structures. These strategies are aligned with key abstractions and mechanisms to achieve a coherent and effective system architecture.

3.1. Modular Design Approach:

Decision: The project adopts a modular design approach, breaking down the system into distinct components such as the Processing Module (Raspberry Pi) and the Communication Platform (Twilio).

Reasoning: Modularity enhances maintainability, scalability, and ease of future upgrades. Each module serves a specific function, allowing for independent development and testing. This aligns with the design goal of efficient maintenance and system adaptability.

## 3.2. Integration of Twilio for Communication:

Decision: Twilio is selected as the communication platform for sending text message alerts to the coaching staff.

Reasoning: Twilio's programmable communication tools align with the KISS strategy, offering a straightforward and reliable method for sending immediate alerts. The decision balances simplicity with effectiveness, ensuring a clear and efficient communication channel in line with the project's goals.

## 3.3. Real-Time Data Processing with Raspberry Pi:

Decision: The Raspberry Pi serves as the central processing unit for real-time data processing, including impact calculations and decision-making.

Reasoning: Leveraging the computing power of the Raspberry Pi enables quick and accurate processing of impact data. This decision aligns with the project's goal of user-friendly operation and efficient deployment by ensuring timely notifications to coaching staff.

## 3.4. Prioritizing User Interface Simplicity:

Decision: The user interface is designed to be simple and intuitive, reflecting the KISS strategy. Once the Pi turns on, we will have a program run from boot.

Reasoning: A straightforward user interface enhances usability, allowing players and coaching staff to interact seamlessly with the smart helmet system. This design decision prioritizes user experience without compromising system functionality.

## 3.5. Emphasis on Scalability without Complexity:

Decision: The system is designed to be scalable for potential future enhancements while maintaining simplicity.

Reasoning: Balancing scalability with simplicity ensures that the system can evolve without introducing unnecessary complexities. This decision aligns with the project's commitment to adaptability and streamlined functionality.

3.6. Clear Communication Protocol:

Decision: The communication protocol between the Raspberry Pi and Twilio is designed to be clear and concise.

Reasoning: Clarity in communication minimizes the risk of errors and ensures the reliable transmission of text message alerts. This decision supports the project's goal of efficient communication, especially during critical moments in the field.

By incorporating these architectural strategies, the Helmet project achieves a cohesive and well-organized system architecture. Each decision is guided by the project's design goals, principles, and the need for a robust, user-friendly, and scalable smart helmet system. The trade-offs made during the decision-making process prioritize simplicity, efficiency, and adaptability to meet the project's overall objectives.

# 4. System Architecture

The system architecture for the Helmet project is straightforward, reflecting the simplicity and effectiveness required for impact detection and immediate alerting. The primary components include:

## 4.1. Subsystem Architecture:

**- Impact Detection Module:**

- Responsible for capturing real-time data on impacts using the Adafruit Accelerometer within the football helmet.

**- Raspberry Pi Processing Module:**

- Serves as the central processing unit, analyzing impact data received from the Impact Detection Module using a Python program.

**- Communication Platform (Twilio Integration):**

- Facilitates immediate text message alerts using Twilio in response to impact data surpassing the concussion threshold.

4.2. Rationale for System Decomposition:

The system is intentionally kept minimal, aligning with the KISS (Keep It Simple, Stupid) strategy. The Impact Detection Module focuses solely on capturing impact data, allowing the Raspberry Pi Processing Module to handle basic computations internally. The Communication Platform acts independently to swiftly trigger text message alerts when needed.

## 4.3. Collaboration and Interaction:

The collaboration is straightforward:

- The Impact Detection Module communicates real-time impact data to the Raspberry Pi Processing Module.

- The Raspberry Pi Processing Module, upon detecting a significant impact, triggers the Communication Platform to send an immediate text message alert using Twilio.

4.4. Design Patterns:

Given the simplicity of the system, design patterns are kept to a minimum. The Observer pattern is subtly employed between the Impact Detection Module and Raspberry Pi Processing Module, allowing efficient communication without unnecessary complexity.

This minimalist system architecture encapsulates the essence of the Helmet project — an accelerometer integrated with a Python program. It efficiently captures impact data, processes it locally, and triggers immediate alerts using Twilio. The design emphasizes clarity, directness, and a streamlined approach to achieve the project's fundamental goal of enhancing player safety through timely concussion detection.

# 5. Policies and Tactics

This section outlines key policies and tactics adopted in the Helmet project, focusing on ensuring reliable impact detection, efficient processing, and prompt communication of potential concussions.

5.1. Threshold Policy:

To effectively identify potential concussions, the project implements a threshold policy within the Raspberry Pi Processing Module. This policy defines the specific force level that, when exceeded, triggers the Communication Platform to send an immediate text message alert. The threshold is carefully determined to balance sensitivity to potential concussions with the avoidance of false positives.

5.2. Real-time Processing Tactic:

A real-time processing tactic is employed within the Raspberry Pi Processing Module to swiftly analyze impact data upon detection. This tactic ensures minimal latency in evaluating the force exerted on the player's head, allowing for immediate decision-making regarding concussion alerts. Real-time processing aligns with the project's goal of providing instant notifications to coaching staff.

5.3. Asynchronous Communication Strategy:

The Communication Platform utilizes an asynchronous communication strategy with Twilio. This tactic ensures that the impact detection process is not hindered by the time-consuming task of sending text message alerts. Asynchronous communication guarantees that the system remains responsive to ongoing gameplay while concurrently notifying coaching staff of potential concussions.

5.4. Simplicity Principle:

A guiding principle throughout the project is the adherence to simplicity. This tactic ensures that the Helmet system remains user-friendly, easy to maintain, and robust in operation. The simplicity principle influences design decisions, favoring straightforward components and processes, with the goal of enhancing user acceptance and system reliability.

5.5. Trade-off: No User Interface for Autonomy:

A deliberate trade-off is made by opting not to include a user interface. This tactic aligns with the autonomy principle, allowing the Helmet system to initiate impact detection immediately upon power-up without relying on user interaction. This trade-off streamlines the system's operation, reducing potential points of failure and contributing to overall system robustness.

These policies and tactics collectively define the operational framework of the Helmet project, emphasizing responsiveness, reliability, and simplicity in achieving the goal of enhancing player safety through effective concussion detection.

# 6. Detailed System Design

The system will have three main components: the hit detection system, the raspberry PI processing system and the communication platform. These components will work together to achieve the product's goal to detect when there is a substantial hit to the head.

## *6.2.* *Definition*

The hit detection system will be an accelormeter that is attached to the helmet. The raspberry PI will be attached to the helmet to read data from the accelerometer and determine if a high impact hit occurred. The communication platform will be used by using Twilio integration to send a message to the necessary people on the sideline.

## *6.3.* *Responsibilities*

The accelerometer is responsible for detecting a large force impact. The responsibility of the raspberry PI is to interpret the data from the accelerometer and determine how severe the impact was. The communication platform's responsibility is to inform people on the sideline of the hit.

## *6.4.* *Constraints*

Some design constraints include how sometimes during a game, severe weather can occur, and water damage will compromise the Raspberry Pi. Also, this helmet is only going to track the force of one large hit, however this is not the only way players can suffer a concussion, they can also occur after a bunch of smaller hits add up and then cause the concussion

## *6.6.* *Uses/Interactions*

The interaction between all the components is crucial because the raspberry PI relies upon the accelerometer to produce accurate information for detection. The Raspberry PI also relies on the communication platform to send out necessary messages.

## *6.7.* *Resources*

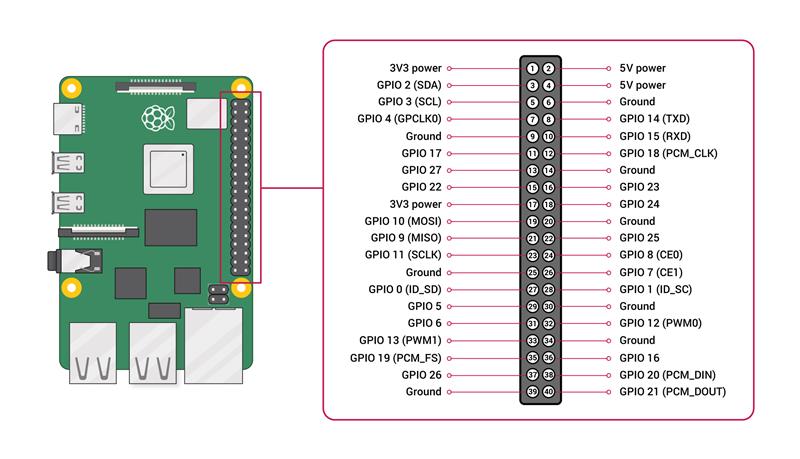
The main resources come from the software and hardware capabilities of the raspberry PI. The computer system has limited computing power that must be optimized. It also requires a power source which must used to operate the machine, this has a limited power supply due to size.

## *6.8.* *Processing*

The majority of the processing power will be done through the raspberry PI, it will be responsible for analyzing the data as well as communicating with Twilio to send the messages. If the raspberry PI were to fail it would cause the entire system to not operate.

## *6.9.* *Detailed Subsystem Design*

The system will consist of the accelerometer connected to the raspberry PI, the raspberry PI will need to be connected to an external power source to operate. The raspberry PI will also connect to Twilio’s servers in order to send out messages about the data it is receiving. The connections to the rasberry PI will be done by using wires like in the diagram below.



# 7. Glossary

Raspberry Pi – A microcontroller/ minicomputer that we will be using to run the Raspbian OS.

Adafruit Accelerometer – The accelerometer we will be using to track the force on the helmet

Twilio - provides programmable communication tools for making and receiving phone calls, sending and receiving text messages, and performing other communication functions using its web service APIs.

# 8. Bibliography

Raspberry Pi Forums: This was the site used to figure out the wiring for the raspberry pi wiring as well as the source of the image.