Senior Design I
Spring 2020

Safety Helmet

Department of Electrical Engineering and Computer Science
University of Central Florida
Dr. Lei Wei

Group 3
Alejandro Velasco: Electrical Engineer
Mariela Barragan: Electrical Engineer
Daniel Ram: Computer Engineer
Michael McCoy: Computer Engineer

Table of Content

List of Tables

1. Executive Summary

The reasoning behind our project is to implement important safety features to a current motorcycle helmet that will aid the driver while they are traveling on the road. We are aiming to create a low cost, easy to use and accurate safety helmet that will alert the driver when other vehicles on the road are getting too close to them or their blind spots. We also want to notify the driver if they appear to be displaying symptoms showing that they might possibly be going to sleep and lastly want to create a notification system that will quickly notify their emergency contact in their phones and the authorities if a crash occurs. Many helmets currently on the market just offer one of these features and do not tailor to the safety aspect of what we are trying to focus on.

In order to alert the driver of any incoming vehicles getting too close to them we will install a camera and ultrasound sensors on the helmet to notify the driver that an object is getting dangerously close. The notifications will take place through either speaker installed inside of the helmet or some sort of visual notification. A camera will also be used to detect eye movement and notify the driver if they are displaying signs of possibly going to sleep. Lastly, we will use an app and through camera and sensors if a collision has occurred and the driver does not respond to the app notifications a text/call will be made to the emergency contact in their phones as well as possibly to an emergency dispatch. If time and money permits, we would also like to add a heads-up display that will notify the driver of how fast they are going along with gps directions.

This document will demonstrate how our group of both electrical and computer engineers are able to work together on one project and incorporate all of our ideas, research and designs that we have amassed over our school career. This will be a well-documented paper showing the journey that it will take for us to complete our goal and make our senior design project a success.

2. Project Description

The helmet will provide safety features for drivers by monitoring driver conditions and environment conditions. The helmet will have ultrasonic sensors to detect vehicles in the driver's dead spot, and rear camera to provide a view of the back. Infrared camera to detect if a driver is falling asleep and a sensor to detect accidents. The helmet will be activated only when used therefore a sensor to detect if not in use will be applied. The helmet will have a display to display data such as speed, RPM, warnings and backward view will be projected to the side view of the driver inside the helmet so it’s not obscuring view. The main control and processing will be done with on board MCU and additional intense computation will be done on mobile devices via a mobile application. The MCU will collect data and transmit the data to the mobile device.

2.1 Project Motivation

Commuting the roads in today's world can be a very challenging and dangerous experience. Our highways and local roads are scattered with cars, trucks and motorcycles and we must all share the roads equally. One of these forms of transportation does not have the same safety and benefits than the others and is in a great disadvantage when traveling on the road. You probably might have guessed already that we are referring to 12motorcycle transportation. Data from the NHTSA shows that in 2017 per registered vehicle, the fatality rate for motorcyclists was 6 times the fatality rate for passenger car occupants. 5,172 motorcyclists were killed in motor vehicle traffic crashes that year. The NHTSA estimates that helmets are 37 percent effective in preventing fatal injuries. What if we are able to bump that percentage up and successfully be able to prevent more deaths? Our group believes that this is possible by adding technological advancements to the helmets that are currently in circulation.

From our research we noticed that many of the current safety helmets in the market seem to emphasize more on communication features and flashiness than actual safety features geared towards the rider and their well-being. Companies and projects that have started out with promise seem to fizzle out and never go into production due to high cost and other factors. Our motivation for this project is to be able to complete this project with the knowledge and skills that we have accumulated at the University of Central Florida and be able to produce a helmet that equals or is better than many of the current safety helmets that are in production or still in the developmental stages.

2.2 Objectives and Goals

The main objective that we are trying to accomplish for our project is driver safety and the ways that we as a group can apply engineering and technological concepts to a motorcycle helmet to make it safer for the driver wearing it while they are on the road. Ultrasonic sensors, rear facing camera, infrared camera and heads up display will all be incorporated onto the helmet to help with our objectives.

The ultrasonic sensors will be used in three different ways. It will detect vehicles that are in the driver's dead spot and alert the driver via a speaker inside of the helmet that an object is approaching. It will activate the helmet to turn on when it will be in use and when not in use the helmet will turn off. This will help us in maximizing the battery life. Last thing we will do is use the sensors in case of an accident, once it detects an accident it will communicate with an app and notify the I.C.E contact on the phone.

Two cameras will be used, a rear facing camera that will provide the driver with a view of the back via a heads-up display that they can easily view. This will help the driver by not having to turn around and see what is behind them when they are trying to maneuver through traffic and will allow them to focus on what is in front of them. The infrared camera will be installed inside of the helmet and will detect eye patterns to determine if the driver is not going to sleep. If the camera detects anything close to this, it will alert the driver through a speaker.

The heads-up display inside of the helmet will be able to display real time data that is being collected like the speed, RPM, warnings and the backward camera feed. It will be carefully positioned so that it is not in the view of the driver but that it is also easily accessible when it is needed.

The main control and processing will be done on board with an MCU and additional intense computation will be done on a mobile device via a mobile application. The MCU will be collecting all of the data necessary and transmitting it to the mobile device.

The following goals below demonstrate what we will be striving to achieve for our project in both hardware, software and as a team:

- Research, design and build a state-of-the-art safety helmet in a timely and efficient manner.
- Use a minimum number of sensors to detect when objects come into close proximity and be able to alert the driver when that happens.
- Be able to turn on or off depending on if the helmet is in use.
- Sensors are able to detect when a crash or accident has occurred and notify mobile applications which will notify I.C.E. contact.
- Successfully use infra-red thermal cameras along with eye detection technology to detect when a person is falling asleep at the wheel.
- Pair rear facing camera along with sensors to aid the driver in not having a “blind” spot when on the road.
- Use a microcontroller along with a Raspberry Pi to control the system. Raspberry Pi and microcontrollers will work with each other.
- Tie all things above and be able to communicate all the data through speakers in the helmet along with a heads-up display.
- Heads up display will not be in the way of the driver.
- Use a mobile application to store data and communicate with the helmet.
- Keep power consumption to a minimum.
- Have our design be user friendly and be able to be used worldwide.

- Adhere to all safety and road standards.

As a team we hope to be able to satisfy all of these goals put forth and demonstrate that we can succeed. If time permits and all goals are successfully completed, we would like to add more objects and features to our project.

2.3 Requirements Specifications

The requirement specifications that we took into consideration are shown in the table below. All requirements chosen are to keep the project as user and budget friendly as possible. We are trying to incorporate the most innovative technology that we can get our hands on but also use tried and true technology that has proven to be successful in the past. Some specifications may change as we start our project and dive deeper into our research but for the most part, we should stick to what we have and be able to make our project work.

Table 1: Requirement Specifications

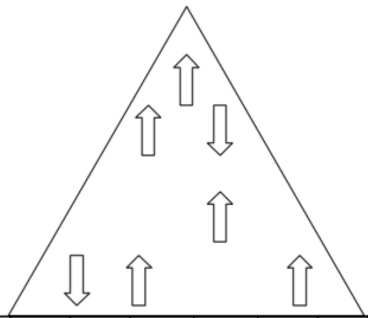
1	**Specification**	**Description**
2	**Size/weight**	No larger than **3 inches wide** and **2 inches tall.** No heavier than **4.8 ounces.**
3	**Power Usage**	Must be low enough so batteries can last long enough for a **full average ride** (if not using onboard 12-volt battery).
4	**Cost**	Must not be larger than **500 USD**.
5	**Usualibity**	Simple enough for the average person to operate. Set up time should be **less than 30 seconds.**
6	**Microphone/Speaker**	Able to give the driver the ability to send voice commands to a mobile device and wake up the driver when sleep is detected through the camera.
7	**Analog to Digital Converter** **Digital to Analog Converter** **DC to DC Converter**	Convert audio signal from a microphone at a resolution of **minimum 16 bits**, convert digital audio signals to analog so it can be played through speakers, handle **up to 15 volt maximum** so it can be connected to an onboard **12 volt battery** (prototype will not use 12 volt bike battery).

8	**Bluetooth 4.0 Module**	Communicate with mobile devices to transmit and receive data that require more processing power.
9	**Back View Camera/Sleep Detection Camera**	Minimum of **1280x720 pixel** camera for capturing back view, infra-red thermal camera with a minimum resolution of **320x240 pixels.**
10	**Sensors**	Minimum of 2 ultrasonic sensors,1 gyroscope, temperature sensor

2.4 Quality of House Analysis

The Quality of House shown below in figure 2 is meant to give us an understanding of how the marketing requirements will correlate alongside the engineering requirements. Marketing requirements are the voice of the customer, this will help us translate what they want and help us fill in our engineering requirements. It is meant to encompass the design, manufacturing, sales and marketing of the smart helmet. The main goal of our project is to be able to make our helmet cost effective to the consumer and close to 100% accuracy. As expected, we can see a lot of negative correlation between cost and our engineering requirements. Comparing and understanding how to balance this section will help us in achieving our goal and be able to balance the house. The house of quality demonstrates how we will achieve our goals and keep us on track to make the best choices whenever possible.

The three engineering specifications chosen for demonstration will be the proximity detection sensor efficiency, motion sensory efficiency and the eye detection software. Demonstrating all three of these will be the focus of the Smart Helmet design. Once these are complete, we will be able to add any other features and designs to the helmet.



Engineering Requirements		Power Usage	Size/Weight	Sensor Efficiency	Eye Detection	Element Endurance	Fault Tolerance
Marketing requirements		-	-	+	+	+	+
Cost	-	↑		↓	↓	↓	↓
Reliability	+	↑		↑	↑	↑	↑
Ease of use	+		↑				
Maintenance	+	↑				↑	↑
Battery life	+	↑		↑	↑		
Design	+		↑				
Engineering Requirement targets		12 V	3 in/4.8 ounce	>90%	>90%	>90%	>90%

+	Positive Polarity
-	Negative Polarity
↑	Positive Correlation
↓	Negative Correlation

Figure 1. House of Quality

2.5 Block Diagram

The Smart Helmet project can be broken down into a main block diagram. Figure x below demonstrates which tasks each group member will be responsible for completing. The Controller block will be the main focus of the project. The power supply, voltage regulation module, speaker, display, sensors, wireless communication module and mobile application will be controlled by the Controller and is the most important part of this design.

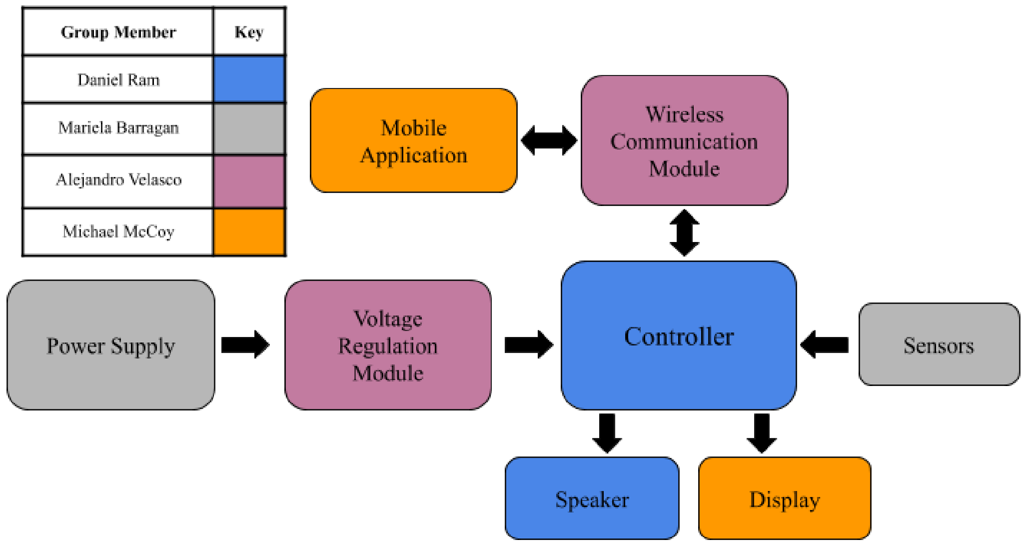


Figure 2. Project Block Diagram

3. Research Related to Product Definition

The safety helmet project is a mixture of modern-day technology and affordability that will be incorporated into a current modern-day helmet. The research section will allow us to demonstrate what is currently on the market and the relevant technology that is being used in today's helmets. Components, part selections and diagrams of what is all involved will be demonstrated, compared and ultimately will guide us in the direction to make our safety helmet stand out from the crowd.

3.1 Existing Similar Projects and Products

There are currently many similar projects and products out in the market that will allow us to compare and get ideas as to how we can model and replicate our own smart helmet. The range of models that we found will allow us to compare what works best and of course what does not work best when it comes to real world usage. We will narrow down what part selection and technology will be the most affordable and hit a perfect price point for consumers.

3.1.1 CrossHelmet

Borderless is a product design firm that originated in 2012 and focuses on research and development. They help many other companies make new products, brands and systems. The founder, Arata Oono is an industrial designer and worked as an automotive designer for 7 years in Tokyo. They are currently focused on development of their smart motorcycle helmet CrossHelmet X1. They bill it as the next generation motorcycle helmet with sound control and 360-degree visibility that will transform your riding experience. It will bring state of the art technology to a safety staple that has remained stagnant for years.

The CrossHelmet will integrate dual-monitor heads-up display with 360 range of vision, sound management, and Bluetooth features to riders. The rear-view camera is combined with a wide-view visor that allows more road-vision and eliminates blind spots. The dual monitor heads up display projects a crisp and clear image regardless of the lighting conditions. According to their website they claim that safety is their number one priority and those two features will achieve that. This helmet is currently not in production. It is currently available to reserve over on the Indiegogo website and the current estimated shipping is set for March 2020. The final price comes out to $1749.00

Figure 3. CrossHelmet X1

3.1.2 Skully Technologies

Skully technologies was one of the first companies to market a heads-up AR display helmet with its FENIX AR helmet. The company started out as a startup in Silicon Valley and got a huge backing from the get-go. The high-tech smart motorcycle helmet was supposed to change the game for rider safety and investors poured in millions and customers lined up to purchase the helmets.

Some of the features of the AR helmet are its heads-up display that is always in focus, transparent display and has a convenient position for the driver. It came equipped with an ultra-wide-angle rear-view camera, GPS navigation, and a high-speed microprocessor for situational awareness. It was supposed to signal the beginning of the motorcycle tech revolution. It had the potential to save lives and make our daily existence a little bit easier.

TECH SPECS

- Lightweight, aerodynamic polycarbonate shell
- Ultra-wide-angle rearview camera
- DOT/ECE safety certification
- SKULLY anti-fog, anti-scratch, anti-glare visor
- SKULLY Synapse (TM) Vision Enhancement - shows true-to-life imaging of rear-view panorama
- Audio/Visual GPS Navigation

- SKULLY Synapse (TM) Smart Heads up Display System with infinite focus
- SKULLY quick release chin strap and visor
- 3D laser-cut foam for a perfect fit
- Bluetooth connectivity to smartphone
- Internet connectivity via smartphone
- Over-the-air updates - download new features as they are released

Figure 4. SKULLY

Unfortunately, the company three years into its existence shut its doors and crashed with plenty of controversy. Cousins Ivan and Rafael Contreras purchased the assets of SKULLY technologies, finished development and even added to the capabilities of the helmet. The cousins could not complete what the original founders of the SKULLY brand had started. After a promising start it seems they as well could not finish, and their website and social media sites have been taken down. This company is an example of what can possibly go wrong, not once but twice to the same product.

3.1.3 Quintessential Design

Quintessential Design is an innovations lab with a strong mission to change the way consumers think about normal safety standards and provide game changing safety standards. Quin was established in 2017 by a diverse team whose backgrounds include award-winning industrial design, women’s safety advocacy, patent law, and more. The Quin motorcycle helmet focusses on being the only truly “smart” goal for helmets - Safety. This company is what our senior design team will be mainly focusing on recreating.

The Quin Helmets use technology to deliver essential safety solutions, focusing on three main areas. Bluetooth invisible integration which allows on the move maps, hands free calls, seamless music and Siri/Google assistant. “Intelliquin” crash detection system which accurately detects dangerous levels of crash force experienced by the rider and informs emergency contacts of the crash location. Lastly the Intelliquin SOS beacon protocols which helps riders reach out for help through a manually triggered emergency message to nearest responders. The power of Quin's innovation lies at the heart of the helmet: The Arc Chip, which is no bigger than a coin, The Arc Chip has a range of sensors that help detect any force from a crash to a slight bump. Through DFU (Device Firmware Upgrade), Quin helmets upgrade via system updates that are easily loaded by running the Quin Firmware Upgrade through the micro - USB connection. Riders can also upload personal audio preferences for calls and music.

Figure 5. Quinn

The company offers three different varieties of helmets. Ghost carbon edition, McQ and the Spitfire. The prices range from $299-$639. Out of all the companies researched for our project they offer the best solutions for the money and strongest technology for what we are trying to compare ourselves with.

3.1.4 JARVISH X-AR

JARVISH was founded in 2014 by the Advanced Technology Division (ATD) team of Foxconn. After two years of research the company launched the first smart helmet in the market in 2017. Their current mission is to make your motorcycle as smart as you and they intend on doing this with distraction-free smart features that help the rider focus attention on the road ahead. The cutting-edge technology that the helmet has are advanced augmented reality, voice activated retractable heads-up display, Amazon Alex Integrated, carbon fiber shell, 360-degree front and rear cameras and active noise cancellation.

The heads-up display leverages optical waveguide technology for riders who prefer visual instructions. Even in bright light, the projections are visible. It has the ability to display directions, notify the rider of traffic and weather conditions, indicators, speed, hazard road signs and incoming caller id. One of the really interesting features of the HUD is that it can be controlled and retracted easily via voice command. The helmet comes with a front and back built in 2K camera and can be activated on demand by voice and allows the rider to have 360-degree visibility. The rider can record video and store it on the 256GB removable memory card or broadcast in real-time to a rider's social media account by voice command.

Automated sensors turn the helmet on when you wear it and turn it off when taken off. Gyro, e-compass, accelerometer and ambient light sensors analyze weather and road conditions to provide real time alerts. The active noise cancellation tunes out noise to focus on the ride reducing wind noise and enhancing the human voice. It does this by featuring an individual audio processor and high-resolution sound regeneration quality. Even at speeds that get up to 100 kph/60 mph.

The Jarvish app allows seamless integration with the helmet. It includes Incoming call notifications, road slip warnings, weather notifications, gas station icon, dangerous section warnings, speed warning, traffic notification, GPS navigation, download videos and view system status. All of this is powered by a military grade solid state FLCB (Flexible type Lithium Ceramic Battery). It is ultra-thin and flexible giving the smart helmet endless possibilities.

Figure 6. Jarvish Heads Up Display

3.2 Relevant Technologies

The smart helmet project will be demonstrating the most relevant technology that we can get our hands on while keeping it as low cost as possible. We will be implementing and researching many different variations of what is needed to complete our project. This section will demonstrate every technology that has been used and a final choice made based on what fits our needs will be presented in section 3.4

3.2.1 Wireless Technology

The smart helmet project will be communicating data wirelessly from the Raspberry Pi to our smart application that will be installed on a mobile device. There are several wireless technologies available to us. Bluetooth, Wi Fi, LoRa, Zigbee to name a few. We decided that the best ways to achieve our goals would be through either the use of Wi-Fi and/or Bluetooth technology.

Bluetooth is useful when transferring information between two or more devices that are near each other when speed is not an issue. It is best suited to low- bandwidth applications like transferring sound data with telephones or byte data with handheld computers, keyboards and mice. Wi-Fi is better suited for operating full scale networks because it enables a faster connection, better age from a base station and better wireless security than

Bluetooth. The following subsections will demonstrate examples of both and demonstrate which form of wireless technology we went with for the project.

3.2.1.1 Bluetooth Technology

Bluetooth is a wireless technology standard that is used for exchanging data between fixed and mobile devices over short distances using short-wavelength UHF radio waves from 2.4 to 2.485 GHz and building personal area networks. Bluetooth is currently managed by Bluetooth Special Interest Group (SIG) which has more than 35,000 members. Bluetooth is a standard wire-replacement communications protocol primarily designed for low power consumption, with short range based on low cost transceiver microchips in each device.

To use Bluetooth wireless technology, a device has to be able to interpret certain Bluetooth profiles, which are just definitions of possible applications and specify general behaviors that Bluetooth-enabled devices use to communicate with other devices. SIG formalized specifications that everybody must adhere to when designing products equipped with Bluetooth.

3.2.1.2 Wi-Fi Technology

Wi-Fi is another form of wireless networking technology based on the IEEE 802.11 family of standards. Devices that can use Wi-Fi technologies include desktops, laptops, smartphones, tablets, smart TVs, and printers to name a few. All these compatible devices can network through wireless access points to each other as well as to wired devices and the internet. There are different versions of Wi-Fi which are specified by various IEEE 802.11 protocol standards, with different radio technologies determining radio bands, and the maximum ranges, and speeds that may be achieved.

Wi-Fi most commonly uses the 2.4 GHz (120 mm) UHF and 5 GHz (60 mm) SHF ISM radio bands, the bands are subdivided into multiple channels. The channels can be shared between networks but only one transmitter can locally transmit on a channel at any moment in time. Wi-Fi wavebands have relatively high absorption and work best for line of sight use. Some of the problems with Wi-Fi are that many common obstructions such as walls, pillars, home appliances etc. may greatly reduce range, but this also helps minimize interference between different networks in a crowded environment. Speed and spectral efficiency of Wi-Fi has increased and will continue to increase in the future. Speeds have been able to achieve over 1 Gbit/s

Table 2: Wireless Technology Comparison

	Bluetooth	**Wi-Fi**
Frequency	2.4 GHz	2.4, 3.6, 5 GHz
Cost	Low	High
Bandwidth	Low (800 Kbps)	High (11 Mbps)
Security	Less secure	More secure
Range	5-30 meters	32 - 95 meters
Power Consumption	Low	High
Latency	200 ms	150 ms
Bitrate	2.1 Mbps	600 Mbps

3.2.2 Sensor Technology

A sensor is a device, module, machine, or a sub system whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor. The smart helmet will be equipped with two different kinds of sensors, proximity and motion sensors.

3.2.2.1 Proximity Sensor

The smart helmet project will be using proximity sensor technology around the helmet to be able to detect any incoming objects that are getting too close to the rider. Proximity sensors include all sensors that perform non- contact detection in comparison to sensors, such as limit switches, that detect objects by physically contacting them. They convert information on the movement or presence of an object into an electrical signal. Different proximity sensor targets demand different sensors. Our project requires the sensor to detect a moving object with simple to complicated surfaces so that leaves us with only a few proximity sensor technologies to choose from.

- **Ultrasonic Proximity Sensors:** Detect the presence of objects through emitting high-frequency ultrasonic range. It performs through the conversion of electrical energy. The sonic transducer emits sonic waves which bounce off the objects. The bounced wave is then returned to the sensor and uses the waves to determine the distance and proximity of the object.

- **IR Proximity Sensor:** IR, short for infrared, detects the presence of an object by emitting a beam of infrared light. Similar to ultrasonic but instead of sonic waves, IR is transmitted. The infrared light is emitted from the IR LED emitter, the beam of light hits the object and gets reflected back in an angle. The reflected light will reach the light detector and the sensor in the light detector determines the position/distance of the object.

3.2.2.1 Motion Sensor

One of the features of the smart helmet is the ability to detect if the rider has been in a collision or accident. In order to make this happen we will be using motion sensor technology to be able to detect if and when this happens. Gyro and Tilt sensor technology will be discussed and through research and development we will be able to decide which one gives us the best data.

- **Gyro Sensor:** A gyroscope senses angular velocity, relative to itself, thus it measures its own rotation, using an inertial force called Coriolis effect. They oscillate at relative high frequency in order to measure this and are thus one of the most power-hungry motion sensors. They can also be affected easily by other vibrations, like motor or speaker on the same device.
- **Tilt Sensor:** Allow you to detect orientation or inclination. They are small, inexpensive, low power and easy to use. If used properly, they will not wear out. They are very simple to use which make them very popular. Usually made by a cavity of some sort and a conductive free mass inside, such as a blob of mercury or rolling ball. One end has two conductive poles so when the sensor is orientated that the end is downwards, the mass rolls onto the poles and shorts term, acting as a switch throw.

3.2.3 Camera Technology

Our Smart Helmet project will be depending on two different camera technologies. A back-view camera that will be either mounted onto the helmet or integrated into it will help capture the rear view and relay the information back to the rider through a display. The second camera will perform the sleep detection on the rider using infra-red thermal technology.

- **HD Wide Angle Camera:** In order to monitor the rear view of the road we will equip the helmet with a minimum 1280x720 pixel camera. The higher resolution and wide angle allow for a better view of the road.
- **Infra-Red Thermal Camera:** Convert thermal energy into visible light to enable the analysis of a specified object or area. The image generated is referred to as a thermogram and is reviewed through a method known as thermography. Infra-red cameras are sensitive to wavelengths from about 1,000 nm to about 14,000 nm.

Table 3: Camera Technology Comparison

Camera Technology	HD Wide Angle	Infra-Red Thermal
Megapixels	2,304,600	8,081,920
Resolution	1280	960
FPS	30	30/60/90
Cost	$13.99	$27.99
size	45mm x 45mm	25mm x 23mm
Weather rating	IPSS	none

3.2.4 Display Technology

One of the ways that we want the outside data to be able to come back to the driver is through a display or heads up display inside of the smart helmet so the rider can view this data. We have several display technologies to choose from and they all come with pros and cons. The display that we had to choose had to be small and compact in order to fit inside of the helmet but big enough so that the rider could see.

3.2.4.1 LCD

A liquid-crystal display is a flat panel display or other electronically modulated optical device that uses the light modulating properties of liquid crystals combined with polarizers. LCDs are available to display arbitrary images or fixed images with low information content. LCD screens do not use phosphors, they rarely suffer image burn in when a static image is displayed on a screen for a long time, they are however susceptible to image persistence. A great advantage that the LCD screen offers is its low electrical power consumption.

3.2.4.2 LED

A LED display is a flat panel display that uses an array of light emitting diodes as pixels for a video display. Their brightness allows them to be used outdoors where they are visible in the sun for store signs and billboards. They are capable of providing general illumination in addition to visual display and offer higher contrast rations than a projector and are thus an alternative to traditional projection screens.

3.2.4.3 OLED

An organic light-emitting diode (OLED) is a light-emitting diode in which the emissive electroluminescent layer is a film of organic compound that emits light in response to an electric current. They are used to create digital displays in devices such as television screens, computer monitors, and portable systems. An OLED display works without a backlight because it emits visible light. Thus, it can display deep black levels and can be thinner and lighter than a liquid crystal display.

Table 4: Display Technology Comparison

	LCD	**LED**	**OLED**
Power Consumption	Medium	Low	Low
Burn-in	No	Rare	Rare
Cost	Low	Low	Low
Viewing Angle	165 degree viewing angle	Shift noticeably	170 degree viewing angle
Mechanism	Liquid Crystal	Light emitting diodes	Organic light emitting diodes
Backlight	Yes	Yes	No
Picture Quality	Good	Good	Best

3.3 Strategic Components and Parts Selections

Deciding which components and parts that we would be using for the smart helmet project took plenty of research and development from all involved. The goal of this project is to maximize what we have available at our school and purchase the parts that will enable us to use the minimum amount of money while selecting the best technology. The following section compares all parts and components needed to successfully produce the smart helmet.

3.3.1 Power Supply

One of the most important technology selections made will be how our project will be powered. We have several options that we can power the microcontroller along with the Raspberry Pi. They can both have their own power supply or share one, and we will determine which one will give us the best use of our money and allow us to take full advantage of the technology needed to run our project. The microcontroller will be able to run off a rechargeable battery or along with the raspberry Pi run off the motorcycle power supply. Voltage regulators will be needed in order to maintain the optimum power needed for our helmet to work.

3.3.1.1 Battery Selection

For the past several years batteries have been a huge part of our normal daily life. From powering our toys that we played with as toddlers to currently powering everything from our laptops, cell phones, and pretty much everything around us. They have evolved greatly the last several years allowing us to take advantage of our technological achievements. The goal for our smart helmet is that we can use all of its features without having to recharge the battery until the rider has completed his or her daily trip around town or if possible, not have to recharge the battery at all. We need a battery that is lightweight as possible and can withstand a long shelf life as well as being affordable and low maintenance. The following subsections will describe all the battery types available to us and at the end be able to select the perfect battery for the Smart Helmet.

3.3.1.1.1 Alkaline Battery

The alkaline battery is considered a primary battery or non-rechargeable battery deriving its energy from the reaction between zinc metal and manganese dioxide. They account for over 80% of manufactured batteries in the US and over 10 billion individual units produced worldwide. Its capacity is strongly dependent on the load and the amount of current it can deliver is proportional to its physical size. The nominal voltage as established by the manufacturer is 1.5 V and if one needs more this can be achieved by connecting them in series of cells, for example three alkaline batteries can generate between 4.5 and 5.0 V. Alkaline batteries are used in many common household items around the world due to the low cost and how easily accessible they have become. Even though some alkaline batteries are described as being rechargeable the majority of the consumers shy always from doing it due to possible leaking of hazardous liquids and rupture.

Figure 7. Alkaline batteries

3.3.1.1.2 Aluminum-Ion Battery

Aluminum-ion batteries are considered a secondary cell or rechargeable battery. Aluminum ions provide energy by flowing from the negative electrode of the battery, the anode to the positive electrode, the cathode. This kind of battery offers the possibilities of low cost and low flammability and properties that lead to high capacity. The energy stored in aluminum batteries on a per volume basis is higher than that in other metal-based batteries.

One of the downsizes of this battery type is the relatively short shelf life. When you combine heat, rate of charge, and cycling it decreases the energy capacity dramatically. For our project we are going to be using a rechargeable battery. Although this type of battery checks off this box it is not a popular choice for people in the consumer world as it is still being worked on and developed. From a price point they are considerably more expensive than all the other choices available to us in the choices for rechargeable batteries.

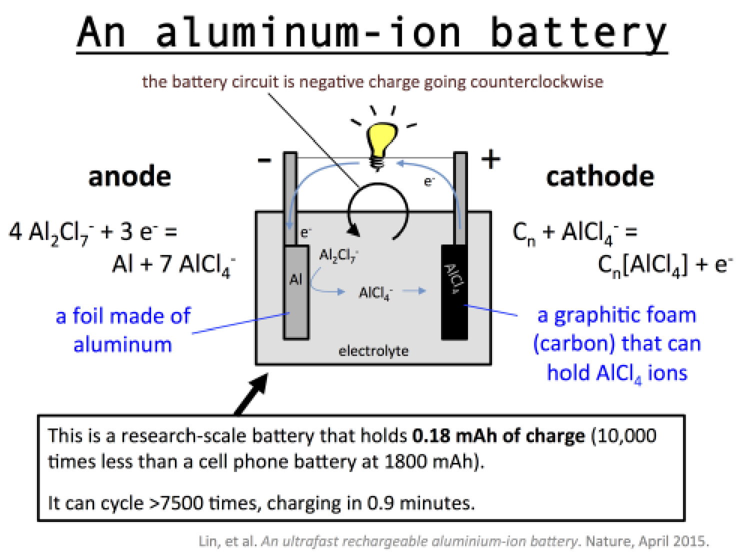


Figure 8. aluminum-ion batteries

3.3.1.1.3 Lithium-Ion Battery

Lithium-ion batteries are secondary cells or more commonly referred to as rechargeable batteries. They are the most commonly used for portable electronics and electric vehicles and are starting to grow in popularity for military and aerospace applications. Development for the technology started out in the 1970s-1980s, and then commercialized by Sony and Asahi Kasei in 1991. The batteries have a high energy density, no memory effect and low self-discharge making them one of the most popular secondary cell batteries available. Some of the downfalls to this kind of battery are the safety hazards that they pose just like any other battery type. They contain flammable electrolyte and if damaged or incorrectly charged can lead to explosions or fires. The major downfall of the battery is the increased vulnerability to thermal degradation and how prone to capacity fading they become over thousands of cycles.

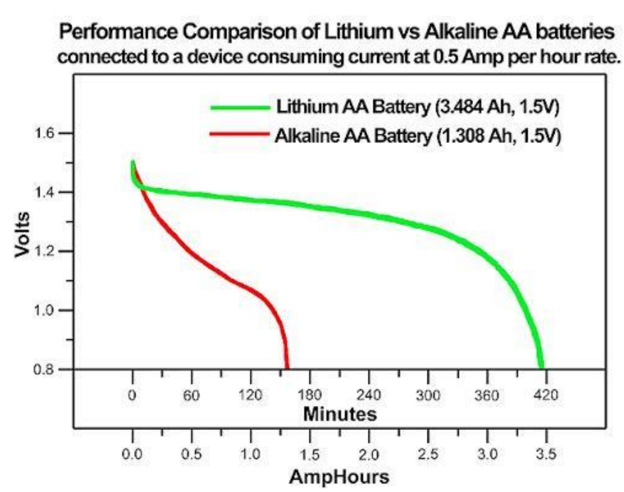


Figure 9. comparison batteries

Lithium-ion batteries have a nominal open-circuit voltage of 3.2 V and typical charging voltage of 3.6 V and can currently be fast charged in 45 minutes or less. When power options were being discussed we decided to go with a battery that would be rechargeable due to cost and convenience. The lithium-ion battery checked off all the requirements needed for our project.

3.3.1.1.3 Lead-Acid Battery

Lead-acid batteries are the earliest type of rechargeable battery. They are very attractive for use in motor vehicles due to their low cost and its ability to supply high surge currents meaning that the cells have a relatively large power-to-weight ratio. A lead acid battery's nominal voltage is 2 V for each cell. For a single cell the voltage can range from 1.8 V loaded to full discharge, to 2.10 V in an open circuit at full charge. One of the benefits of lead acid batteries is how environmentally recyclable they are compared to other batteries. They have one of the most successful recycling programs in the world. In the United States 99% of all battery lead was recycled between 2009 and 2013.

Table 5: Battery Technology Comparison

Battery Type	**Lead Acid**	**Alkaline**	**Aluminum Ion**	**Lithium Ion**
Rechargeable	Yes	No	Yes	Yes
Nominal Voltage (V)	2.1	1.5	2.65	3.2
Energy Density (Wh/kg)	30-40	85-190	406	1060
Shelf Life (years)	1	5-10	1	2-3
Cost	low	low	low	low
Availability	high	high	low	high

3.3.2 Proximity Detection Sensors

The Safety Helmet is going to have multiple sensors in order to help achieve the safety features that we want it to accomplish. These different sensors are going to detect, measure and record different inputs such as distance, motion, and angular velocity. We want the Safety Helmet to be able to detect if there are other vehicles or hazards in the blind spots. Currently there are numerous proximity distance sensors that are available in the market, ranging from ultrasonic ones that use sound waves to detect the distance of objects to laser ones and infrared ones that use light to detect distance. This section compares these different kinds of sensors and discusses the advantages and disadvantages of each. At the end, a selection will be made.

3.3.2.1 HC-SR04 Ultrasonic Range Finder

How does the HC -SR04 Ultrasonic Range Finder work? The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo. The sensor emits high-frequency sound waves from the transmitter (trig pin) towards the desired object and the object picks up the sound's waves. The sound waves bounce off the object and return to the transmitter (echo pin). Since we know how fast soundwaves travel, we use the time it took for the soundwaves to return to the object in order to calculate the distance of the desired object. For example, let's say an object is 10 cm away, we know that the speed of sound is 0.034 cm/µs. If we divide $\frac{10cm}{0.034\ cm/us}$, we obtain 294 us. So, the sound wave would have to travel for 294us to reach an object that is 10 cm away. But let's keep in mind that this number is double because the sound wave is traveling towards the object, and then it needs to bound back. In order to obtain the distance in centimeters we multiply our time 294us by 0.034cm/us and we obtain about 10cm.

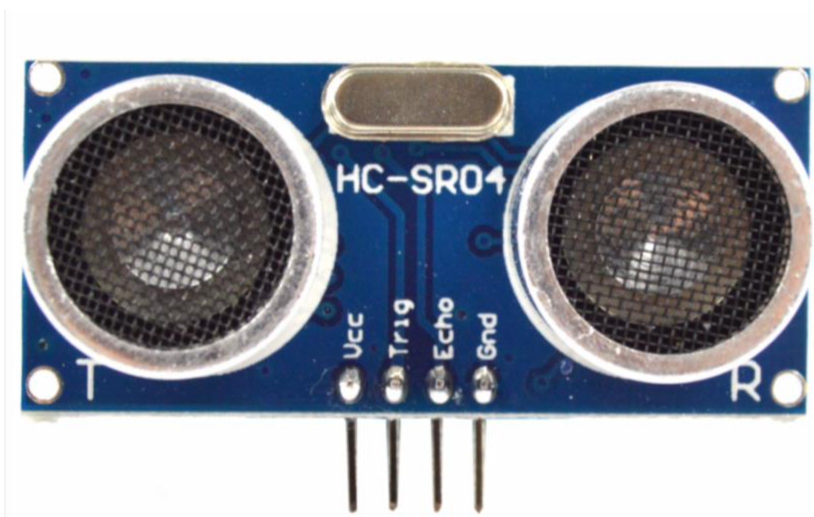


Figure 10: HC-SR04 Ultrasonic Range Finder
https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/

Specifications

- -Working Voltage (DC): 5V
- -Working Current: 15 mA
- -Working Frequency: 40Hz
- -Max/Min Range: 4 meters/2 centimeters

- -Measuring Angle: 15 degrees
- -Trigger Input Signal: 10uS TTL pulse
- -Echo Output Signal: Input TTL level signal and the range in proportion
- -Dimension: 45X20X15mm
- -Resolution: 0.3cm

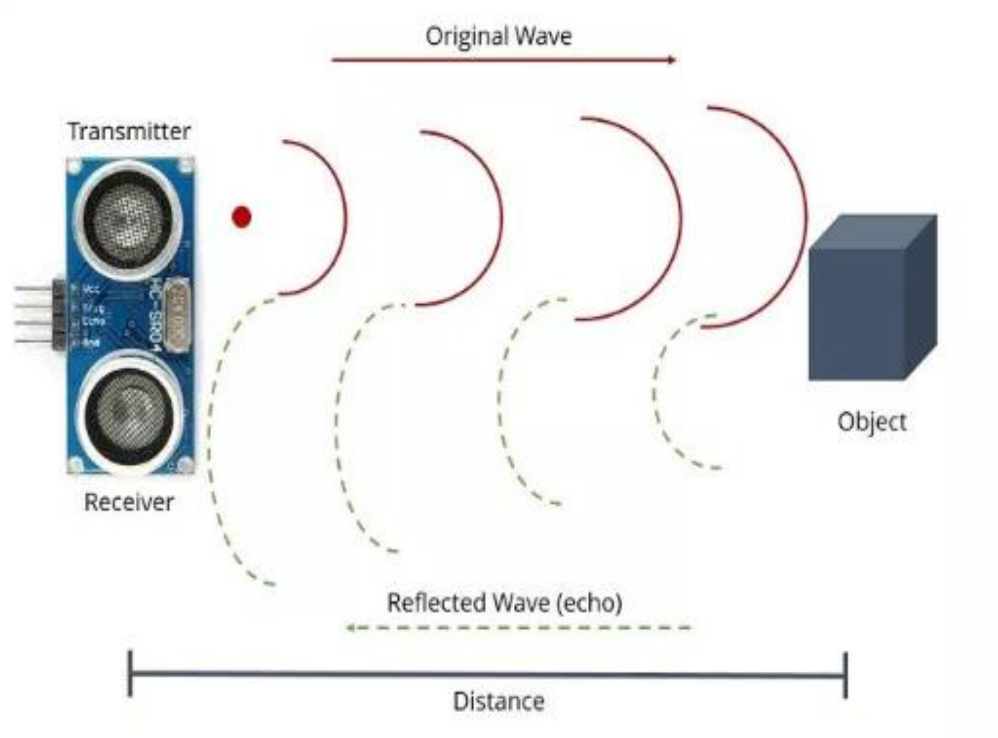


Figure 11 : HC-SR04 Ultrasonic Range Finder
https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/

3.3.2.2 HRLV-MaxSonar-EZ Ultrasonic Range Finder

Just like HC-SR04, the HRLV-MaxSonar-EZ uses ultrasonic waves to measure distance. The HRLV-MaxSonar-EZ uses high frequency sound waves in order to detect objects. It measures the time it takes for the high frequency sound waves to be transmitted and reflected back to the sensor. Based on the result that is obtained, the sensor can determine the distance. The same example and calculation that was used in section 3.2.1.1 with the HC-SR04 can be applied here because it works in a similar manner as the HC-SR04 by using ultrasonic sound waves. The HRLV-MaxSonar-EZ , although more expensive than the HC-SR04, is more accurate and portable.

Figure 12: HRLV-MaxSonar-EZ

Specifications

- -Working Voltage (DC): 2.5V-5.5V
- -Average Current draw: 1.65 mA at 5V
- -Max/Min Range: 6.45meters/0 meters
- -Dimension: < 1 cubic inch
- -Resolution: 1-cm

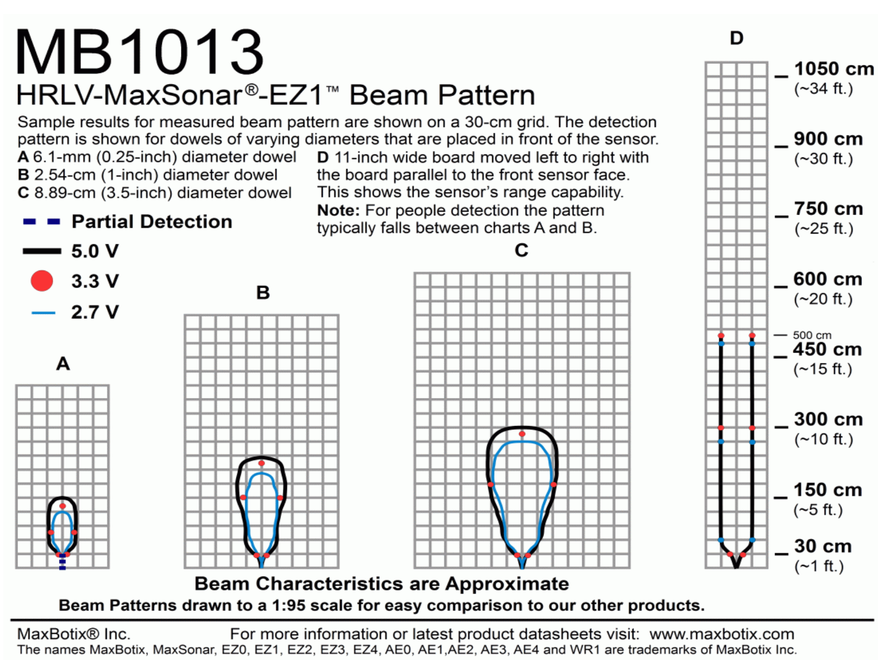


Figure 13. Beam Pattern

3.3.2.3 Sharp GP2Y0A21YK0F

Unlike the previous two sensors that used sound waves in order to calculate distances, the Sharp GP2Y0A21YK0F is an IR distance sensor that uses a beam of infrared light that reflects off an object in order to measure its distance. Since this sensor uses light instead of sound to measure distance, it is more accurate than ultrasonic ones. Also, this sensor is light weight and it has a low power consumption. Triangulation is the method that is used to measure the distance where the angle of a reflected IR beam is measured. Figure 6 illustrates how triangulation works. Basically, the emitter emits a pulse of IR light, if there is an object present, the light will reflect off the object and returns to the detector (PSD). This creates a triangle between the point of reflection, the emitter and the detector.

Figure 14: Sharp GP2Y0A21YK0F

Specifications

● -Operating Voltage:	4.5V to 5.5 V
● -Average Current Consumption:	30 mA
● -Min/Max Range:	10cm to 80cm
● -Dimensions:	44.5 mm × 18.9 mm × 13.5 mm
● -Output Type:	Analog Voltage
● -Update Period:	38 ± 10 ms

The Sharp GP2Y0A21YK0F sensor returns an analog output (voltage) that is used to determine the distance of an object. There is a relationship between the sensor's output voltage and the inverse of the measured distance which is approximately linear over the sensor's usable range. Figure 16 shows this voltage distance relationship. One of the limitations that Sharp GP2Y0A21YK0F sensor has is that objects must be at least 10 cm away in order to obtain an accurate reading. For example, if we obtain a voltage reading of let's say 2.3V, by looking at the graph in figure 16, that could be a reading of 3cm or 10cm. Therefore, the accurate readings would occur after the peak at a distance of 10 cm. If we do another example after the peak, let's say we obtain an analog reading of 0.5V, by looking at the graph, the object would be about 60 cm away. This sensor is accurate with the readings because compared to the other ones, this one uses light to determine distance. The update period of the measurements is 38 ± 10 ms which is fast.

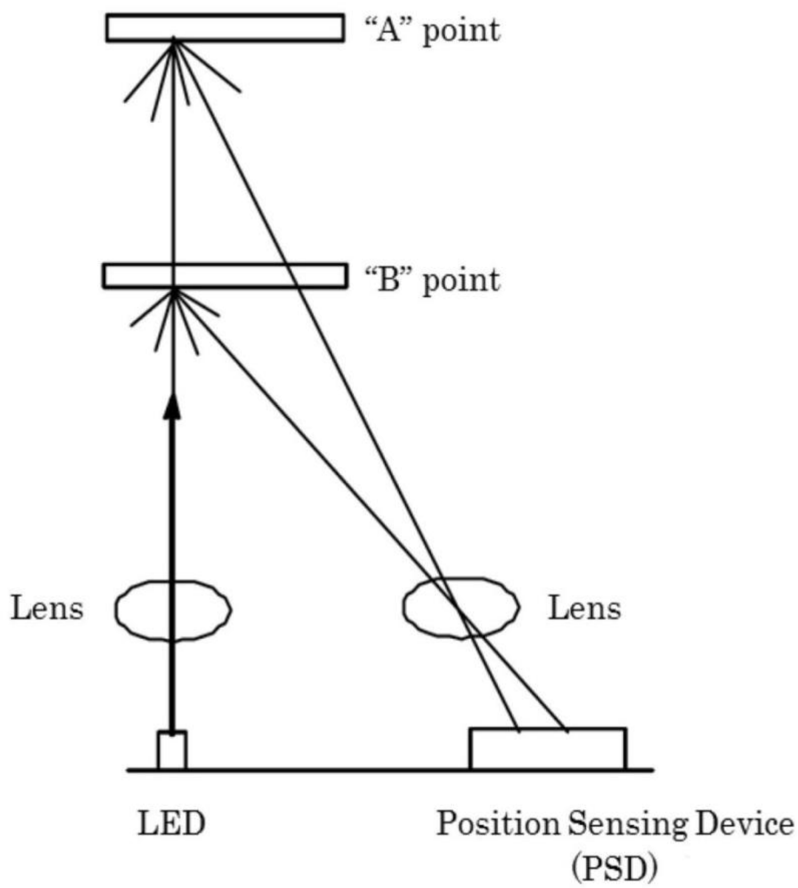


Figure 15: Sharp GP2Y0A21YK0F Triangulation

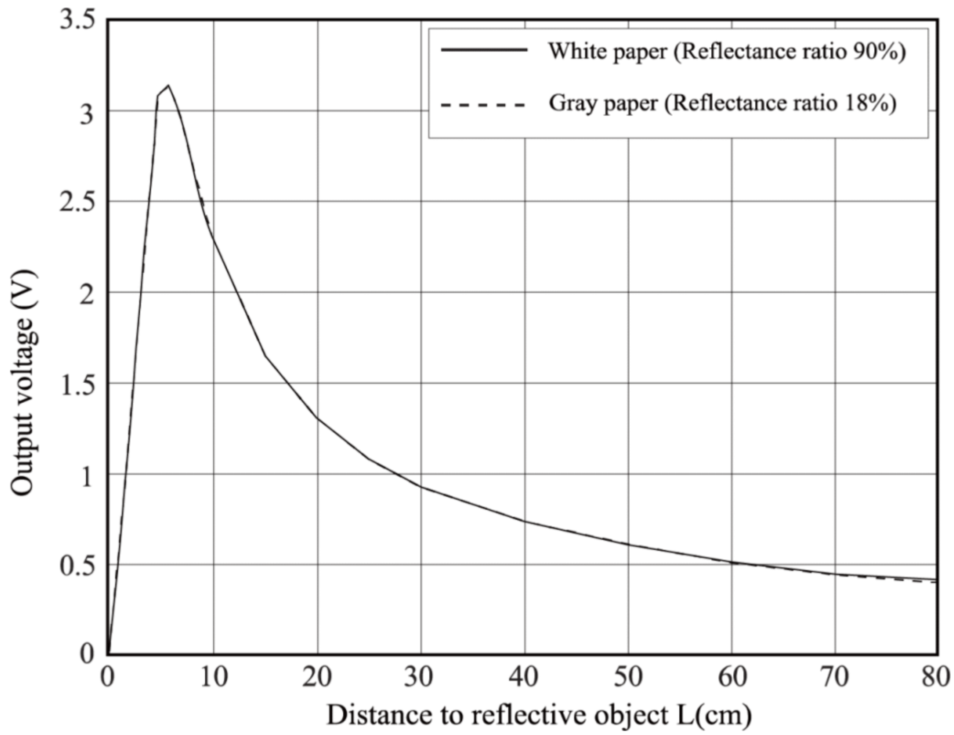


Figure 16: Output voltage vs distance to object

3.3.2.4 Comparing Distance Sensors

In this section, we will be comparing the different distance sensors that were discussed. Advantage and disadvantage of each of the distance sensors is going to be discussed as well as some of their key specifications. To conclude, a selection is going to be made, and that's going to be based on the sensor that best meets our requirements for this project.

Advantages vs Disadvantages

With the HC-SR04 sensor some of its advantages are that it's the lowest cost out of the three and also the reading is not affected by color, transparency or if the environment is dark. Some of its disadvantages are that it has difficulties in reading reflections of objects that have different textures as well as fast moving objects and it's sensitive to variation in the temperature.

With the HRLV-MaxSonar, some of its advantages are that it has high resolution and out of the three components it's the smallest one. Some of its disadvantages are that it's the most expensive out of the three and just like the HC-SR04, it's sensitive to variation in temperature.

With the Sharp GP2Y0A21YK0F, some of its advantages is that it's the most accurate one out of three because it uses light and also, it has less difficulty detecting moving objects. An advantage that it has is that since it has a narrow beam, it is likely to miss objects that fall outside of its range.

Table 6: Distance Sensor Comparison

	HC-SR04	**HRLV-Maxsonar-Ez**	**Sharp GP2Y0A21YK0F**
Price	$3.95	$29.95	$13.07
Size	43×20 ×15mm	22.1×19.9×15.5mm	29.5×13.0×13.5 mm
Manufacturer	Digi-Key	Digi-Key	Digi-Key
Operating Voltage	5 V	2.5 V - 5.5 V	4.5 V to 5.5 V
Min/Max Range	400 cm	645 cm	10 cm to 80 cm
Output Type	Echo	Analog Voltage	Analog Voltage
Current Consumption	15 mA	2.5 mA to 3.1 mA	30 mA

3.3.2.5 Distance Sensor Selection

The distance sensor that was chosen was the HC-SR04. It was decided to go with this sensor because it's a module that we have worked before with. Compared to the other ones, it would be easier to program and incorporate into the helmet. Cost plays a role in our part selection; we already have multiple HC-SR04 ultrasonic sensors so we wouldn't have to order it. We would need two HC-SR04 ultrasonic sensors, we would mount one in each side of the helmet,

3.3.3 Motion Sensors

We want the Safety Helmet to be able to detect if there is a collision or an accident. If a collision or accident is detected, the Safety Helmet will notify 911 or an emergency contact. In order to be able to detect if an accident or a collision has happened, we are going to be using motion sensors that can detect tilt. The concept is, that when the rider is riding the motorcycle, there should be little or no tilt. If an accident or collision were to happen, the motorcyclist would most likely be on the ground and a large tilt would be detected. The motorcyclist would be given about 10 seconds to respond if they are ok. If there is no response, 911 or an emergency contact will be notified.

There are various motion tilt sensors that are available in the market. The way that these sensors work is by measuring the tilt or angle in multiple axes of a reference plane. These types of sensors allow for easy detection of orientation and inclination. Two sensors that detect tilt are the Gyro Sensor SN-ENC03R0 and the Tilt Sensor- AT407.

3.3.3.1 Gyro Sensor (SN-ENC03R0)

Two main types of gyroscopes are vibrating structure gyroscopes and rotating gyroscopes. The SN-ENC03R is a single axis vibrating structure gyro sensor. It has a high sensitivity and it is low cost. It is reliable and easy to use. The increase of the sensitivity is due to a low-noise amplifier that is incorporated in the sensor. Since this is a vibrating structure gyroscope, it works by using a phenomenon called Coriolis Effect. Basically, incorporated in this sensor there is a vibrating structure, this vibrating structure continues to vibrate even if the support of it is exposed to some type of rotation. The vibrating object can exert a force on the support if there is some type of rotation and by measuring the force that is exerted, we can determine the rate of rotation.

Figure 17: SN-ENC03R0 Gyro-Sensor

Main Features:

- Single axis gyro sensor with an analog output
- Powered by +5V with 3.3V onboard voltage regulator
- Raw Sensitivity: 0.67mV/°/s
- Full Scale Range: 300°/s
- Onboard Amplifier Gain: 4.7x
- Dimensions: 24.13mm X 22.86 mm

This type of motion sensor would be a good choice to incorporate into our project. It's reliable and accurate because of its high sensitivity. We want the Safety Helmet to be as economical as possible. The SN-ENC03R Gyro-Sensor is economical and it is also small in size with dimensions of 24.13mm X 22.86 mm.

3.3.3.2 Tilt Sensor- AT407

The AT407 Tilt Sensor is basic and easy to use. It has a really low power consumption as well as a really low electrical rating. Compared to the SN-ENC03R0 Gyro-Sensor, it is way more economical and smaller in size. The way that it works is that inside the cylindrical enclosure case there is a small ball and some pins. When the cylindrical enclosure is upright the small ball makes contact with these pins but if the cylindrical enclosure is tilted then the ball will not make contact with the pins.

Main Features:

- Electrical Rating: <6mA 24VDC
- Electrical Life: > 50,000 Cycles
- Contact Resistance: 1 ohm
- Solder Temperature:250°C 3 Seconds
- Ambient Temperature: 0°C~100°C

Figure 18: Tilt Sensor- AT407

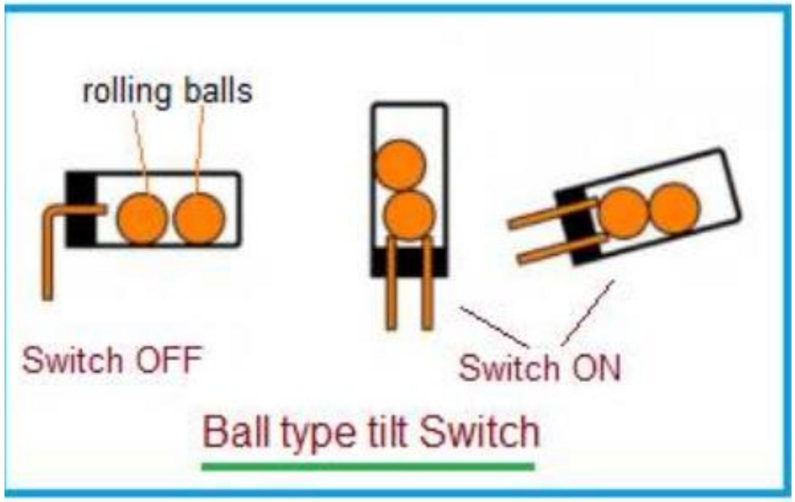


Figure 19: Tilt Sensor- AT407

In the figure above we can see that the sensor is on when the cylindrical component is upright or when it is tilted a little. We can see the little balls are making contact with the pins. We can also see when there is a large tilt, the little balls are not making contact with the pins and the switch is off. In the Safety Helmet, when the motorcyclist is riding, their head is straight and upright. If they are turning or moving their head, there's going to be some tilt, but it won't be severe enough to cause the little balls to stop making contact with the pins. On the other hand, if there is an accident or collision and the motorcyclist is on the ground, the tilt will be large enough to cause the little balls to stop making contact with the pins. When this happens, there is going to be a period of about 10 seconds where the motorcyclist can communicate if they are ok, if there is no response, 911 or an emergency contact will be notified.

Although the AT407 Tilt Sensor has many advantages like its price and its ease of use, there are some disadvantages. For example, if this tilt switch is exposed to currents above the ones that are given in the specifications, the contact sensors can get damaged. Another major concern with this type of switch is that as previously mentioned, the function of it is to measure if there is a great amount of tilt in case of an accident, if there is a sudden acceleration or deceleration, results can become unreliable.

3.3.3.3 MPU-6050 3 Axis Gyro

The MPU-6050 is a triple-axis gyroscope containing a 16-bit AD converter chip with 16 bit of output data. This sensor is very accurate, and it has a small footprint of 2 x 1.6 x 0.1 cm. It requires a power supply of 3-5V, and the current required for it to operate is 3.6mA. In order to communicate, it uses the I2C communication protocol. Because of its 16-bit

ADCs, we can use simultaneous sampling of the gyroscope. This device also requires less need for user calibration because of its enhanced biased and sensitivity temperature.

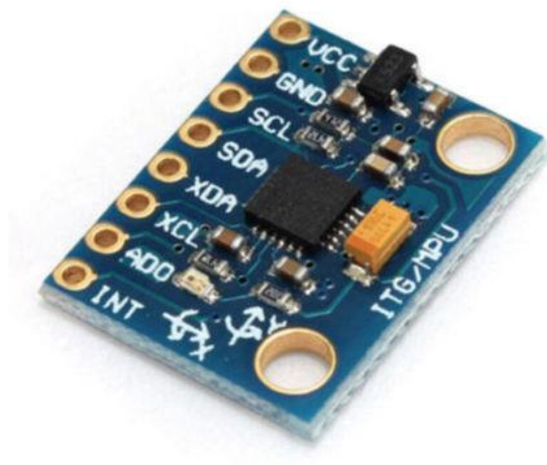


Figure 20: MPU-6050 3 Axis Gyro

Main Features:

- Power Supply: 3-5V(internal low dropout regulator)
- Communication: I2C communication protocol standard
- Gyro Range: +/-250 500 1000 2000 °/s
- Size: 2 x 1.6 x 0.1 cm
- Gyroscope operating current: 3.6 mA
- Standby current: 4uA

This sensor would be a good choice to include in our project. It is really reliable, and it has a high accuracy. If we compare it to the tilt sensor, we wouldn't have to worry with this one about sudden acceleration or deceleration where results can become unreliable. This is an economic sensor and it is small in size.

3.3.3.4 Comparing Motion Sensors

In this section, we will be comparing the different motion sensors that were discussed. Advantage and disadvantage of each of the motion sensors is going to be discussed as well as some of their key specifications. To conclude, a selection is going to be made, and that's going to be based on the sensor that best meets our requirements for this project.

Advantages vs Disadvantages

Some of the advantages of the Gyro Sensor SN-ENC03R0 is that it has high sensitivity at 0.67mV/°/s and it also has a higher accuracy compared to the Tilt Sensor- AT407. Some of the disadvantages are that the ease of use not as basic as with the Tilt Sensor- AT407 and compared to the Tilt Sensor- AT407, the Gyro Sensor SN-ENC03R0 is bigger in size and not as economic

Some of the advantages of Tilt Sensor- AT407 is that it's basic and easy to use and it has low power consumption and it's more economical than the Gyro Sensor SN-ENC03R0. Some of its disadvantages are that results can be unreliable if there is a sudden acceleration or deceleration and also there could be errors in the results if there are a lot of vibrations.

Some of the advantages of the MPU-6050 3 Axis Gyro is that it has a high sensitivity and out of the three sensors, it is the most accurate one. Some of the disadvantages, is that implementing the software could be complicated compared to the three other sensors. Down below is a table comparing all of the sensors that we researched to provide us with a better understanding of which to go with.

Table 7: Motion Sensor Comparison

	Gyro Sensor SN-ENC03R0	**Tilt Sensor- AT407**	**MPU-6050 3 Axis Gyro**
Price	$18.80	$1.95	$9.95
Size	22.86 x 22.86 x 5.08 mm	29X5.2mm	21.2x 16.4mm x 3.3mm
Manufacturer	Digi-Key	Digi-Key	Digi-Key
Measurement Range +/-	300 °/s	30 °/s	250 500 1000 2000 °/s
Current Consumption	3.5 mA	<6 mA	3.6 mA
Operating Power	3 V~ 5.25 V	Up to 24 V	3 V - 5 V

Axis	Single	None	Three

3.3.3.5 Motion/Tilt Sensor Decision

After comparing both motion sensors, having the Gyro Sensor SN-ENC03R0 would be the best decision. The main factor that contributed to this decision was that the Tilt Sensor-AT407 just seemed to be really unreliable if there is too much vibrations or if there is a sudden acceleration or deceleration. Although we want the helmet to be as economical as possible, we also want it to be as safe and as reliable as possible. Choosing the Gyro Sensor SN-ENC03R0 felt that it was the best choice out of the two.

3.3.4 Camera Module Comparison

In this section we will discuss the different types of camera modules that we have considered to take care of the drowsiness and rear collision detection tasks for the smart helmet system. Each module we have chosen to evaluate satisfies the basic requirements such as having a relatively low power and footprint. There are different requirements for each individual task. For example, the eye drowsiness detection task requires for a camera to be resilient to low lighting conditions due to its location within the helmet.

3.3.4.1 Infrared Cameras

This section will give a description of each camera evaluated for the eye drowsiness task. These cameras were selected for evaluation due to their ability to perform in low light conditions. Each camera has a different way for accounting for low light environments, such as removing the infrared filter that is usually included with cameras as well as emitting its infrared light to produce better quality images.

NoIR Camera Module V2

This camera module is created to capture images in environments that do not have optimal lighting conditions. Since lighting conditions may be less than optimal within a motorcycle helmet, the infrared camera will allow us to worry less about environmental conditions affecting the performance of our camera. The NoIR camera Module V2 was specially created to work with the Raspberry Pi (SBC), being fully compatible with the Raspbian operating system and physically connecting to the Raspberry Pi via the CSi (Camera Serial Interface) located on the board.

Figure 21: NoIR Camera

5MP IR-CUT Infrared Light Surveillance Camera Module

This camera module includes several features to remove lighting distortions and comes with infrared LED to enable image capture in low light conditions. This camera also maintains full support of the Raspberry Pi (SBC) and is fully configured to work with the Raspbian operating system.

Figure 22: 5MP IR-CUT

Table 8: Camera Comparison

	5MP IR-CUT	*NoIR*
Focal Length	4mm	3.04mm
Field of View	70° at 4mm	62.2°
Resolution	1080p	1080p
Footprint	(35x35x45)mm	(25x25x9)mm
Voltage Input	3.3V	3.3V
Cost	$13.99	$27.99

Arducam Mini Module Camera

This camera module can be configured to work with a variety of microcontroller platforms. Sensor settings can be set through an I2C interface, while camera commands and data output can be interfaced through SPI. The module also includes communication libraries for easy software development. It integrates 2MP CMOS image sensor OV2640 and provides miniature size as well as easy to use hardware interface and open source code library. In addition, the camera's relatively small footprint will allow some flexibility when integrating with the motorcycle helmet.

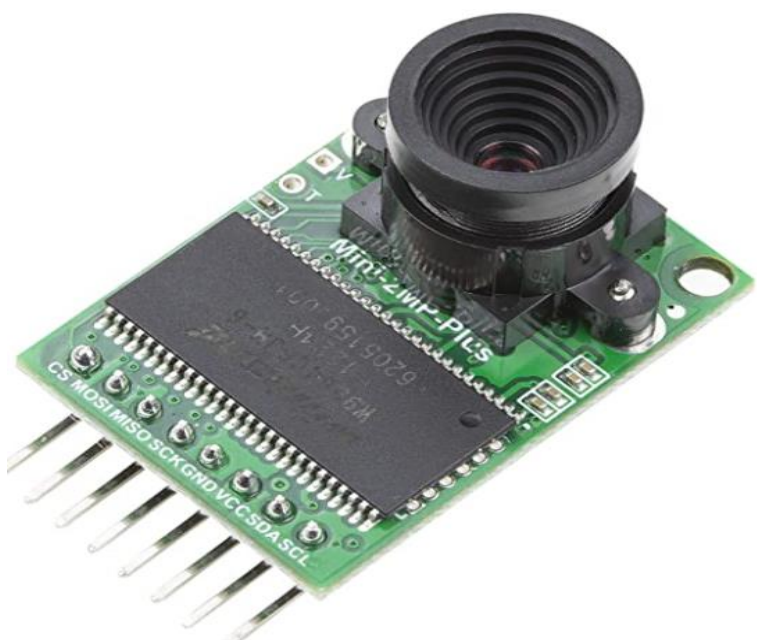


Figure 23: Arducam

Specifications / Features:

- 2-megapixel sensor
- Infrared sensitive
- 5V/3.3V IO port tolerant
- Footprint: 50mm x 38 mm x 36mm
- Weighs ~ 2.88 ounces

Camera Down select

This section describes the camera modules we have down selected to handle the image processing drowsiness detection and rear collision avoidance features in our system. We selected the Raspberry Pi NoIR Camera Module V2 to handle the image capture that will feed into our drowsiness detection algorithm. Although the price is much higher than the 5MP IR-CUT Infrared Light Surveillance Camera Module, the smaller footprint will give us some flexibility when fitting the camera within the helmet. The Raspberry Pi NoIR Camera Module is also recommended by the Raspberry Pi Foundation to ensure hardware compatibility. For the rear collision avoidance feature we selected the Arducam Mini Module Camera due to its established communication interface and resolution.

3.3.5 Transmission hardware

For this project, we will need hardware to transmit data between the MCU and a mobile device. The data transmission has to be wireless in order to allow easy usage for potential users. It doesn't have to be very fast because the data that is going to be transmitted is just basic user control such as navigation information and texting in case of emergency. Communication between the MCU and Raspberry Pi will be done with wires and not wireless. With that in mind we considered two types of wireless communications, Bluetooth and WIFI. We decided to look into these two because they are widely used and support most modern mobile devices. Also, it's relatively easy to set up a connection with a mobile device which is part of the goal of our project, to make user experience simple and easy for new users. Here we only discuss transmission between the helmet (or more precisely the MCU in the helmet) to a mobile device and not transmission between MCU and sensors or MCU and Raspberry Pi, this will be discussed in another section.

3.3.5.1 Bluetooth

We first consider the Bluetooth module for its simplicity for the user. Bluetooth is a wireless communication standard between two devices over a short distance (usually up to

10 meters) using UHF radio waves. Ultra-High Frequency (UHF) radio waves between 2.400 - 2.485 GHz are used to create a personal area network that allows for a transmission of data in both directions.

There are many Bluetooth modules available and could have been selected but we decided to look into HC-05 Bluetooth module which is popular for microcontroller applications. This module is easy to set up and use due to its popularity and many tutorials on the internet.

One of the benefits of this module is that it can be used in full-duplex mode which we need in our project. It can also easily pair to a mobile device which is what we need in order to allow the mobile app to contact an emergency contact when necessary. Also, its support for UART is a huge plus because our microcontroller supports UART, additionally the wide range of baud rate is giving us more flexibility with our microcontroller choices.

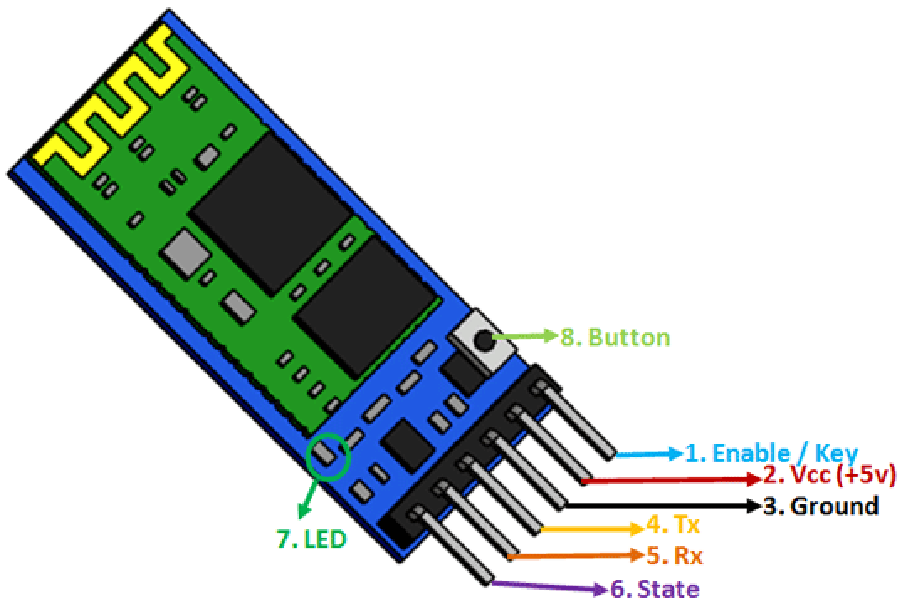


Figure 24: HC-05 Bluetooth Module

HC-05 Technical Specifications

- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol
- Can operate in Master, Slave or Master/Slave mode
- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800

3.3.5.2 WIFI

The second choice for a communication module is WiFi. WiFi is a wireless network technology based on the IEEE 802.11 standards. It's using short range wireless communication to create a local area network and create a connection between all the devices in the network. WiFi module is fabricated on an integrated circuit and is widely used in mobile devices, laptops and Internet of Things. Although WiFi is meant to be connected through a router it is possible to connect two devices directly without an external router however, it is not as simple as Bluetooth.

We decided to look into the ESP8266 WiFi module.

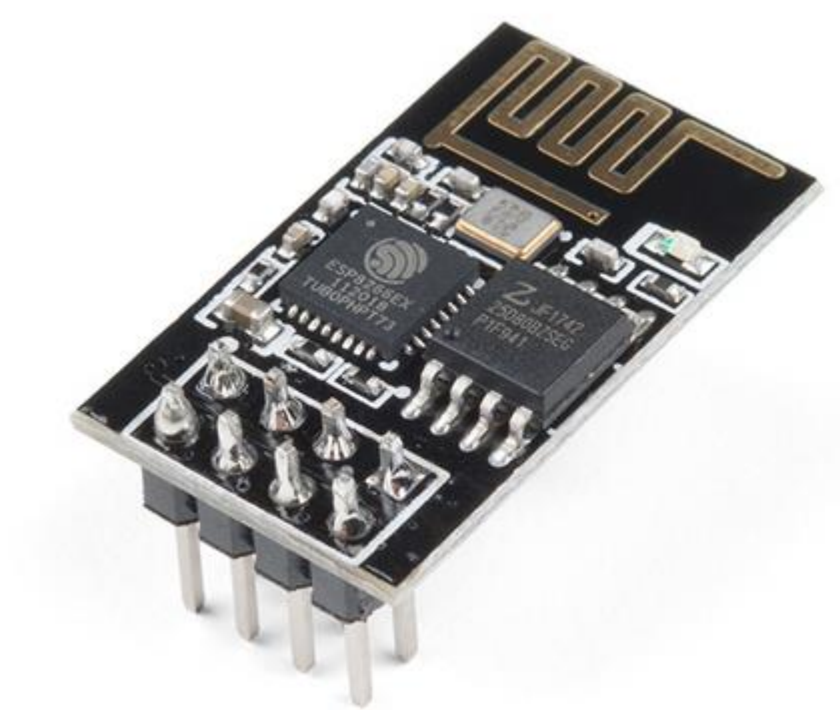


Figure 25: ESP8266 WiFi Module

Main Features

- Supply voltage (Vcc) 3.0V-3.6V
- Average power consumption of 80mA
- 802.11 b/g/n Standards with 2.4GHz frequency support
- WPA/WPA2 Encryption
- Wifi Direct support
- Support for TCP/IP protocols
- Clock frequency support up to 80MHz
- Only 3 pins (except ground and Vcc)

The main benefit of this module is that it supports WiFi Direct which means that we don’t need an external router in order to connect to a mobile device. Additionally, having a WiFi module in this project means that we can add more network features in the future such as connecting to a router to perform firmware updates.

Final Decision

After considering both modules we saw that the power requirements are not too different and since we only need a simple connection between the helmet and mobile device, Bluetooth module is our choice. Also, in order to make simple user experience Bluetooth is usually simpler to set up then WiFi direct. Some mobile devices don’t support WiFi direct and nearly all mobile devices support Bluetooth.

3.3.6 Control

The helmet will have two types of control computer, main control and secondary control. The main control will be done by a microcontroller unit (MCU) which will communicate directly with sensors and perform calculation required for data transmission from sensors. It will communicate directly with mobile applications and Raspberry Pi by sending and receiving data and data representation on LCD screens and indicators.

The secondary control is done by a Raspberry Pi and it is responsible for computer-intensive tasks. Since MCU is limited in that regard it will transmit images from a thermal camera, directly to the Raspberry Pi and the Raspberry Pi will perform analysis using computer vision techniques to extract data about the state of the user. If the state of the user is not awake enough to perform safe driving MCU will alert the user using the indicators and onboard speaker.The following diagram shows the main structure of the control system:

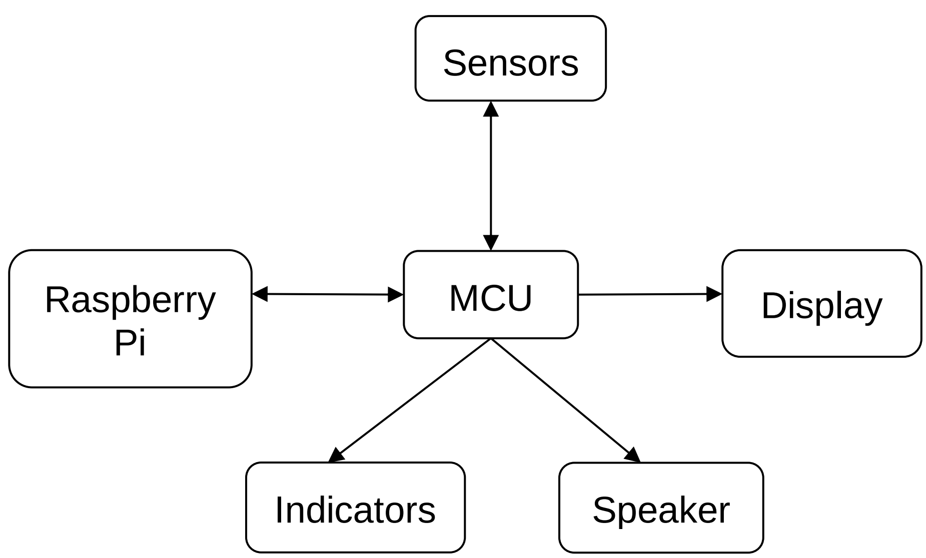


Figure 26: Control System Diagram

3.3.6.1 Microcontroller

The main control for this project is the microcontroller, it is a low power microcontroller which will communicate and extract data from cameras, buttons, indicators, and other sensors. It will also send data from the thermal camera to the Raspberry Pi for analysis using computer vision techniques. It will be responsible to display relevant information to the user and alert the user when necessary.

3.3.6.1.1 Microcontroller Choices

In order to decide which microcontroller to use we need to identify the components that are going to be attached and the capabilities required. Based on our initial plan we need an MCU with a large number of GPIO pins to support our wide range of sensors. Since we plan to incorporate a display inside the helmet and a display is usually require a large number of pins we'll have to use a controller circuit that will interface between the MCU and the display using one of the transmission protocols (I^2C, SPI, UART) our MCU then must support these protocols. Since we already used MSP430 MCU in embedded systems this has a priority since less learning curve will be required from us if we use that MCU. On the other hand, learning new MCU can enhance our knowledge in different technologies which will prove valuable in our self-improvement and skills. Since MSP430G2 doesn't have enough GPIO pins we might consider the MSP430FR6989 which has a larger number of pins. Since the helmet also contains a speaker, we'll need DAC in order to play audio on the speaker. The audio signal will be transmitted from the mobile device though the Bluetooth module and the MCU will transmit the signal to the speaker through the DAC circuit.

For some other choices of MCU we looked into Microchip selection which can be a good alternative to TI Launchpad with extensive documentation and popularity in industry. Microchip offer wide range of MCUs from low power 8-bit MCUs to 32-bit MCUs and after researching about several choices we decide to look into SAM C20 which is based on 32-Bit ARM Cortex M0+ and offer up to 48 I/O pins and slightly cheaper then MSP430FR6989 however, it will pose a greater learning curve.

Another microcontroller that we considered is the STM32 which is an Arm based MCU with Arm Cortex-M3 32-bit RISC processor and can run on up to 48MHz clock. This microcontroller has up to 129kb of flash memory and a wide range of features such as 6 timers and 51 I/O pins and support communication protocols I^2C, SPI, UART and USB.

TI Launchpad MSP430

The MSP430 microcontroller is simple, easy to use and since we used it in embedded systems, we lean more toward choosing this microctornoller. It supports all the features we need for this project and is well documented.

Main Features:

- 16-Bit RISC Architecture with up to 16MHz clock
- Supply voltage between 1.8V - 3.6V
- 3 Low power modes (LPMx)
- Integrated LCD driver
- 12-Bit analog to digital converter
- Serial communication protocols I²C, SPI, UART and IrDA
- Five 16-bit timers with up to 7 capture/compare register 16-bit each
- 83 General purpose I/O pins

The following diagram shows the pinout of the MSP430 we intend to use for this project.

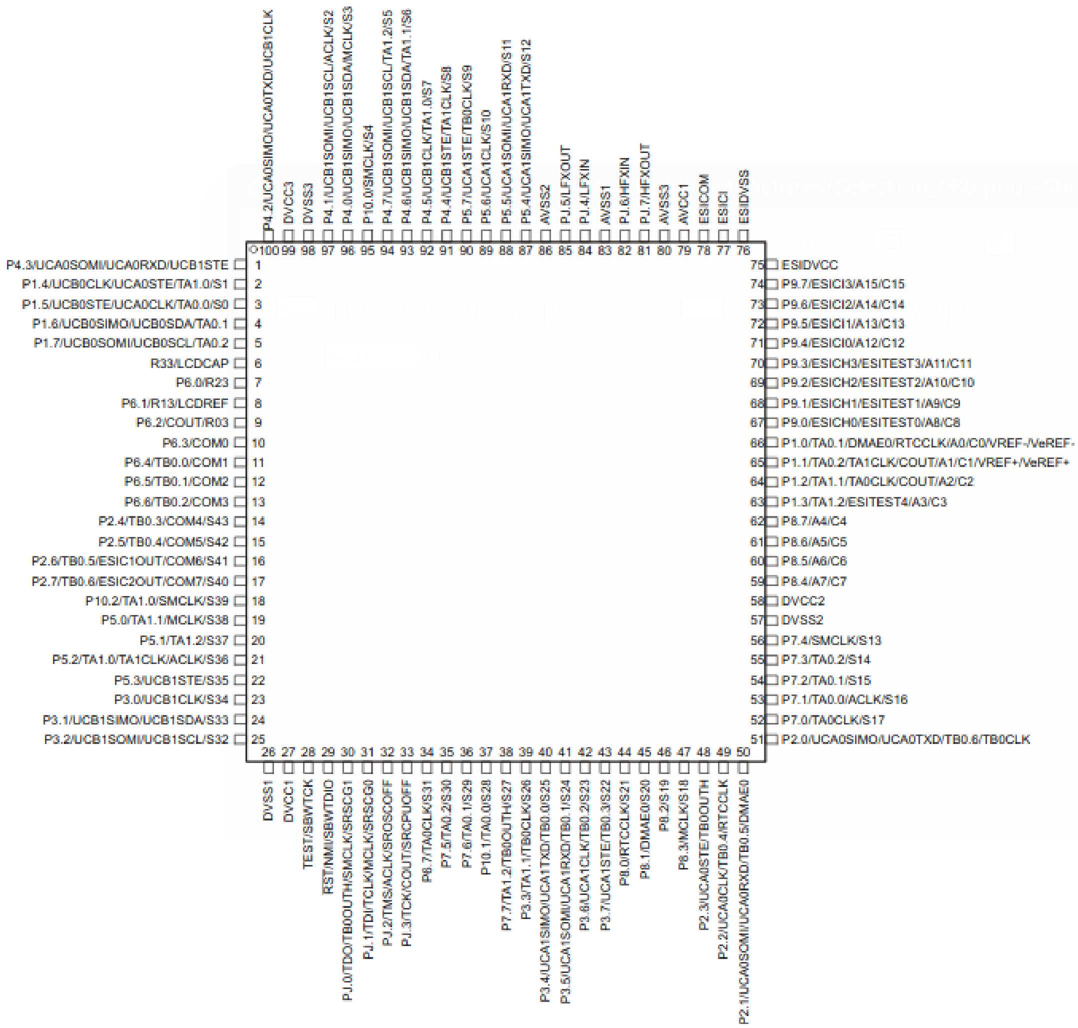


Figure 27: MSP430FR6989 pinout

Microchip SAM C20

Microchip microcontrollers are increasing in popularity recently and they offer a wide range of microcontrollers. The ATSAMC20G17A has a slightly stronger CPU with support of a clock up to 48MHz and is based on ARM Cortex M0+ which is a 32-bit RISC architecture similar to the MSP430, but slightly a smaller number of pins compared to TI. Microchip also offers several tools for development for free such as MPLAB X IDE and MPLAB Harmony which is a framework library for all Microchip devices.

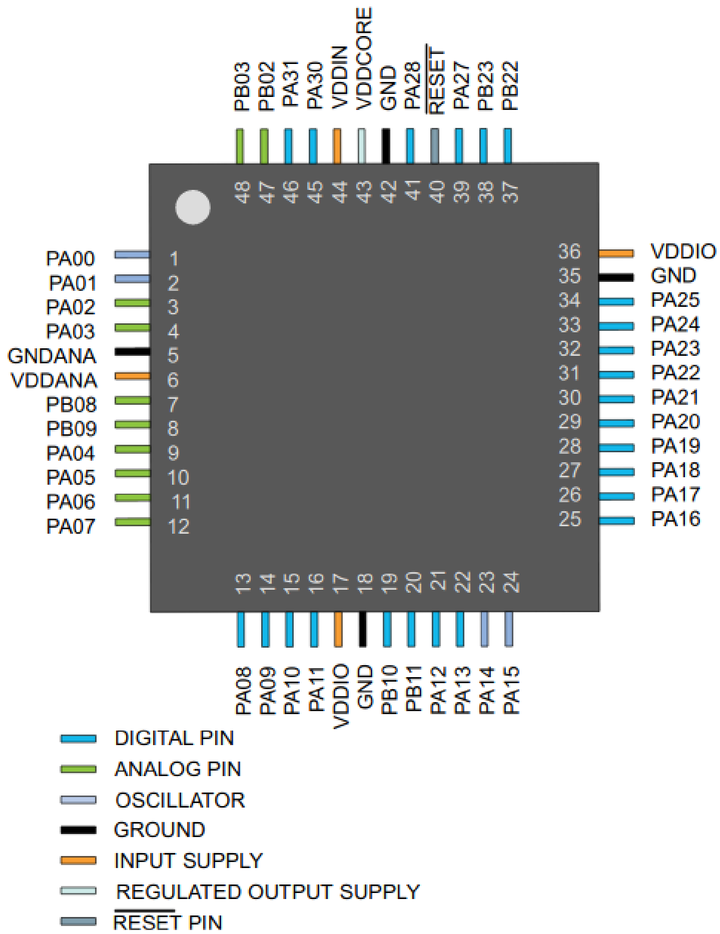


Figure 28: Pinout for ATSAMC20G17A

Main features:

- 32-Bit RISC architecture with up to 48MHz clock
- Supply voltage between 2.7V - 5.5V
- Idle, standby, and off sleep modes
- 12-Bit analog to digital converter
- 38 I/O pins
- Communication protocols: I²C, SPI, UART, LIN, RS-485
- Five 16-bit timers
- 128KB program flash memory

From first look we can see that the SAM C20 has less I/O pins, but higher main clock frequency compared to MSP430 which might be better for fast video stream to the Raspberry Pi.

STMicroelectronics STM32 Series

ST is well known in Microelectronics devices and after examining their line of MCUs we decided to look into their STM32 series or more specifically the STM32F102CB microcontroller which is based on ARM Cortex-M3 processor similar to the others discussed above. It is a 32-bit RISC architecture with a clock frequency up to 48MHz and a wide variety of features.

This microcontroller is very similar in features to the SAM C20 from Microchip however it has more I/O pins. Some of the I/O pins also have 5V tolerance which might be useful since some of the sensors operate on 5V.

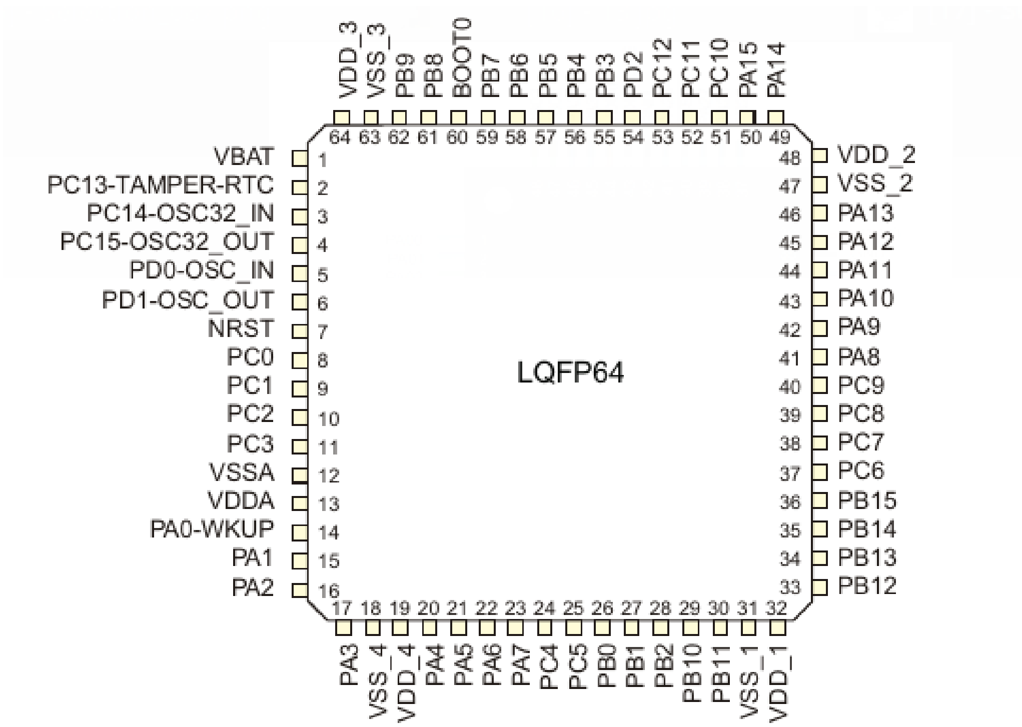


Figure 29: Pinout for STM32F102CB

Main Features:

- 32-Bit RISC architecture with up to 48MHz clock
- Supply voltage between 2.0V - 3.6V
- Sleep, stop, and standby modes
- 12-Bit analog to digital converter
- 51 I/O pins (with most of them 5V tolerant)
- Communication protocols: I^2C, SPI, UART, USB 2.0
- Three 16-bit timers and two 24-bit countdown timers
- Up to 128 KB flash memory

This microcontroller is very similar in features to the SAM C20 from Microchip however it has more I/O pins. Some of the I/O pins also have 5V tolerance which might be useful since some of the sensors operate on 5V.

3.3.6.1.2 Microcontroller Comparisons

In order to choose the right microcontroller, we need to address the needs for our project. We're planning on connecting two cameras which between 4 to 12 pins (depends if we are using I^2C), two ultrasonic sensors which require 2 pins each, gyro sensor which require 2 pins, Bluetooth module which require 4 pins and a few extra pins for interfacing with Raspberry Pi and additional sensors and buttons. From this analysis we will need at least 20 I/O pins for the current setup and a few more extra which mean at least 35-40 pins. Another important feature is fast data transmission, since we will have two cameras interfacing with the microcontroller, it needs to be able to transmit the images fast enough to the Raspberry Pi in order to perform analysis. The microcontroller should also be able to control a display so a microcontroller with display interfacing hardware is more plausible.

Power Consumption

Since the helmet is meant to be used for a full average ride it requires us to choose a microcontroller that has low power consumption while retaining important features such as number of I/O pins. Power consumption from data sheet (extreme cases) for STM32F102CB is $3.6V \times 150mA = 540mA$, for ATSAMC20G17A is $5V \times 92mA = 460mW$ and for MSP430FR6989 is $3V \times 1.875mA = 8mW$. As we can see, since the MSP430 has a smaller processor it requires significantly less power than the other two microcontrollers.

Memory Size

In our project the heavy-duty computation will be done with the Raspberry Pi therefore, memory size is not a big factor in our choices however, we still looked into the memory provided by each microcontroller. For STM32F102CB the memory size is up to 128KB flash memory and 16KB SRAM memory, for ATSAMC20G17A is up to 256KB flash memory and up to 16KB SRAM with additional 8KB independent self-programmable flash memory, for MSP430FR6989 is up to 128KB ferroelectric nonvolatile memory. Since memory is not of a major importance, MSP430 will be the right choice in this regard.

Cost

One of our goals is to keep the cost as low as possible, comparing the cost of each microcontroller: STM32F102CB cost $4.7, ATSAMC20G17A cost $1.84, and MSP430FR6989 cost $8.6. From this we can see that the Microchip microcontroller is the cheapest, but these prices are for the chip only. Since we are going to need the development board which we already have for MSP430 from embedded system class it will save us money and time. Since for the other two boards we'll either have to build a development board or order one.

Clock Frequency

Some of the features in this project require us to make intensive computation. First, we will need a microcontroller that can control a display. Second, we will have two cameras attached to the microcontroller which will require the microcontroller to read and transmit the images from the rear camera to the display and from the thermal camera to the Raspberry Pi. The data transmission needs to be fast enough in order to present smooth video from the rear camera and for the Raspberry Pi to perform analysis in a reasonable time. Clock frequency for STM32F102CB is up to 48MHz, for ATSAMC20G17A is up to 48MHz, and for MSP430FR6989 is up to 16MHz. As we can see the MSP430 has a slower clock, but we think that it will be enough to support our features. The following table summarize this comparison:

Table 9: Microcontroller Comparison

	STM32F102CB	ATSAMC20G17A	MSP430FR6989
Power Consumption	540mW	460mW	8mW
Memory Size	128KB	256KB	128KB
Cost	$4.70	$1.84	$8.60
Clock Frequency	48MHz	48MHz	16MHz
I/O Pins	51	84	100
Number of Timers	6	8	5

ADC Resolution	12-bit	16-bit	12-bit
Operating Voltage	2.0 to 3.6V	2.7 to 5.5V	1.8 to 3.6V
Number of Interrupts	16	16	16

Final Decision

Given the analysis above we decided to choose MSP430 for several reasons. First, the power consumption is significantly smaller compared to the other two microcontrollers and it is due to smaller and more power efficient processors. Second, is our familiarity with MSP430, since we already have some experience using MSP430 in embedded system labs and as well as some personal projects it will save is time which otherwise will go toward learning a new microcontroller. Third, is the cost, it's true that the cost of all the microcontrollers analyzed above are not expensive but since we already have the development board for MSP430 it will save us from buying a development board for another microcontroller or even designing a new development board if none is available to purchase. Fourth, is our familiarity with TI toolset such as Code Composer Studio that we used in the lab.

3.3.6.2 User Drowsiness Detection

According to the National Highway Traffic Safety Administration, driver drowsiness was responsible for 72,000 crashes and 800 deaths in 2013. With this staggering number in mind, we have incorporated a way to detect user drowsiness levels in order to make the roads a safer place. Innovations in computer vision allow us to accomplish this task. A camera will be embedded within the helmet, taking images of the user while they are driving. These frames will be processed by several facial localization algorithms to locate the eyes of the user. Once the algorithm has found the user's eyes, we can then run the drowsiness detection algorithm, which will output a scalar value indicating whether or not the user is on the verge of falling asleep.

Drowsiness Detection Approaches

Several considerations must be made when devising a method for drowsiness detection. Factors such as low lighting within the helmet may greatly affect the efficiency and performance of the image processing, which may result in deadly consequences. Below is an algorithm that can be used to perform drowsiness detection. The description of the algorithm below assume localization of the face and eyes has already been computed:

Aspect Ratio

In this approach we establish key points from localized features on the face, extracting key points that correspond to the eyes of the user. Once the key points are extracted, we can perform an aspect ratio calculation from the key points that make up the eye. If the eye is closing the aspect ratio will be a smaller value and vice versa. By being able to output a scalar value indicating the percentage of eye closure by a user, we are able to make a reasonable estimation that the user is falling asleep. The algorithm should take into account user blinks to not generate false positives.

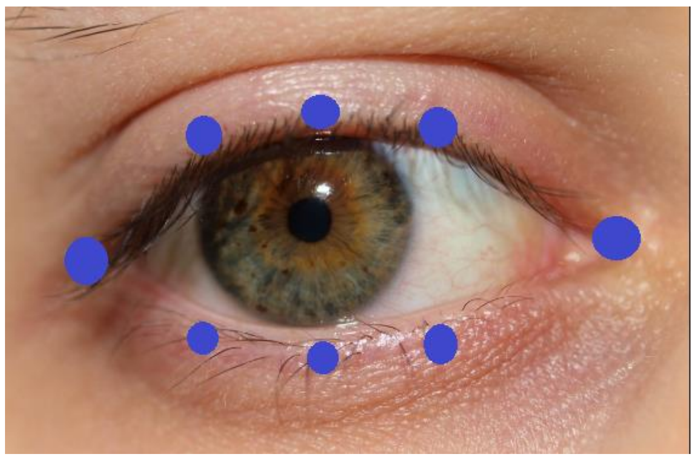


Figure 30: Aspect Ratio

OpenCV (Open Source Computer Vision Library)

OpenCV is an optimized computer vision library containing several modules to help with image processing, video analysis, multiple-view geometry, and much more. Originally developed by Intel, many leading researchers have contributed work to the library over the years making it one of the most popular computer vision resources in the field. Developers are able to rapidly integrate computer vision features within their applications due to the library's well-documented and established application interface.

3.3.6.3 Raspberry Pi

The Raspberry Pi is an open source single-board computer developed by the Raspberry Pi Foundation to promote the teaching of fundamental computer science concepts. The most recently released board (Raspberry Pi 4) features a Broadcom system on chip (SoC), with an integrated ARM-Processor and graphics processing unit. The Raspberry Pi 4 is essentially a small-footprint full-fledge computer that is capable of handling large computational tasks, such as image processing. The Raspberry Pi's Broadcom processor will be able to quickly and efficiently run the software that will help detect user drowsiness.

Figure 31. Raspberry Pi 4

Raspbian Operating System

The Raspberry Pi hardware can be managed by a number of Linux distributions, mainly supporting the Raspbian operating system. The Raspbian operating system already includes many of the drivers and dependencies needed to simulate a desktop computer.

Table 10: Raspberry Pi Specs

Specifications	
CPU	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
RAM	2GB LPDDR4-2400 SDRAM

WIFI	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Ethernet	Gigabit
USB	2 USB 3.0 ports; 2 USB 2.0 ports
GPIO Header	Raspberry Pi standard 40 pin
HDMI	2 × micro-HDMI ports (up to 4kp60 supported)
Display-Port	2-lane MIPI DSI
Camera Port	2-lane MIPI CSI
Power Consumption	3.4W
Cost (2GB RAM)	$45.00

3.3.6.4 Mobile Application

With a high percentage of people having mobile devices we felt that the smart helmet system would benefit from incorporating a mobile application. The system will include a mobile application that will interface with onboard modules through Bluetooth to display information to the user. Not only will a mobile application enhance the user's product experience, but also extend the functionality of our system. By leveraging hardware on the user's mobile device, we are able to extract user location information which can be calculated by the mobile device's GPS (Global Positioning System) module. This locational information will allow our system to provide users with proximity information from target destinations. The mobile application will also serve as a dashboard to view system diagnostic information, such as battery life and component function. The application can also be configured to contact certain individuals in case of an accident or system failure.

Cross-platform framework

The technologies used to develop the systems mobile application should allow for cross-platform use. React Native, is an open source cross-platform mobile framework that was developed by Facebook to create their Ads Manager application. Applications that use React Native are written entirely in JavaScript and are then rendered with the running platform's native code.

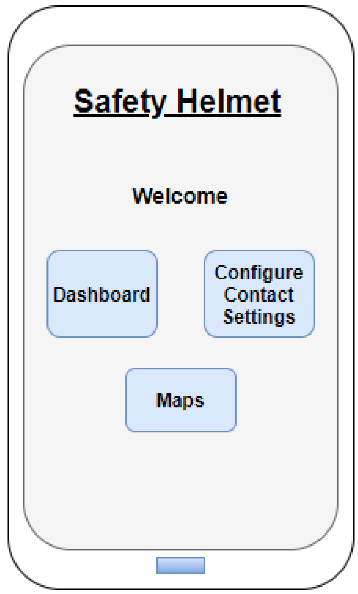


Figure 32. Mobile application wireframe

3.3.7 Display

Providing a visual interface for the user is a major feature that has been included as a requirement for the system. Streaming real-time data to users from a windshield has become commonplace in the automotive industry. In order to get all of the data from the Smart Helmet to the rider an electronic visual display or screen will be used. The electronic visual display for the Smart Helmet project that we will be choosing has to be small enough to fit inside of the helmet but big enough so that the rider can easily make out what is being shown. There are many types of displays available in the market. This section will show the various technologies that accomplish such a task and discuss the down select process from the components that we considered.

3.3.7.2 Heads-up display (HUD)

The heads-up display was mainly created as a safety feature, allowing users to maintain their usual view pose, while having access to other data simultaneously. Originally developed for military applications, the heads-up display has gained popularity in a variety of commercial applications as well. Many cars today incorporate the heads-up display to allow drivers to easily view their current speed while maintaining their eyes on the road. Since our system will be displaying information to the user during operation of a motor vehicle, we feel incorporating a heads-up display within our system will increase the safety of our users and add a futuristic aesthetic to our overall product design.

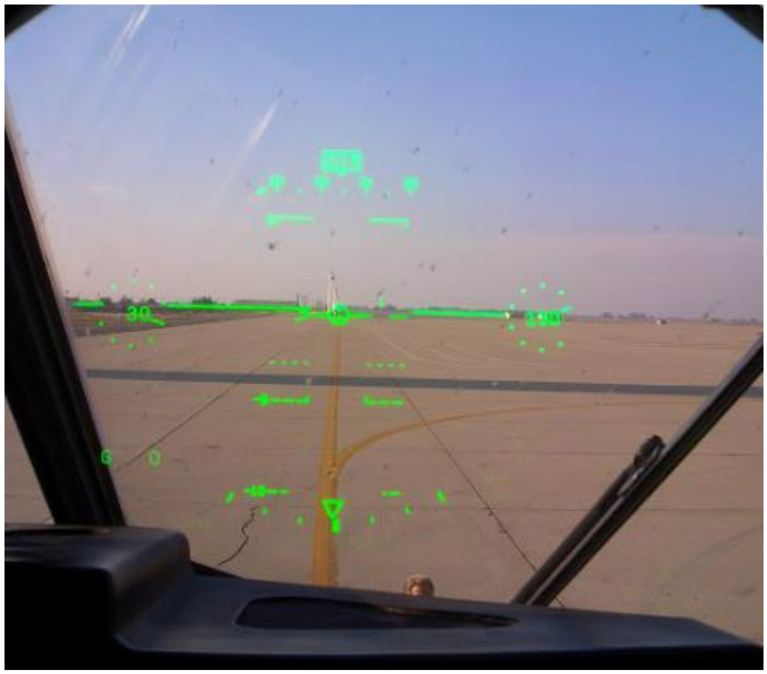


Figure 33. HUD application

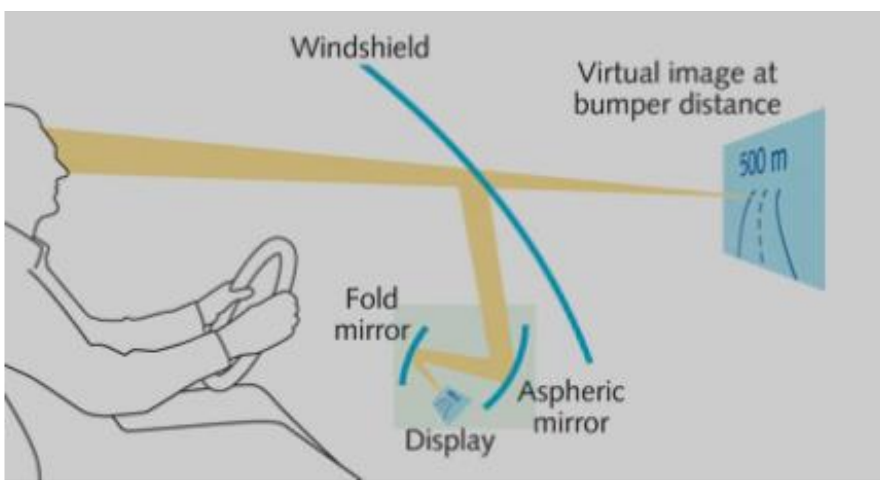


*Figure 34. HUD (How it work*s)

How it works

The heads-up display generates the virtual image that the user sees by first projecting a bundle of light to a fold mirror which is then reflected to a rotatable mirror. The function of the rotatable mirror is to magnify the original image to achieve a desired projection distance. The projection distance generated is the distance of the virtual image from the user's eyes.

3.3.8 Indicators

We want the helmet to be able to warn the driver if there is some type of danger in the blind spot. The distance sensors would if there is something in the blind spot of the motorcyclist, let's say another vehicle. If the distance sensor detects that the vehicle is too close to the motorcyclist, we want the helmet to warn the driver that there is danger and that they should not turn. We can do this in multiple ways. One way that we could do it is by adding two little displays into the helmet that would alert the rider if there is danger. Another way that we could do it, is by maybe using LED lights. We can put an LED in the helmet and if it lights up, it means danger, there is something in the blind spot and the rider should not turn.

3.3.8.1 Display Indicators

Displays can be used in order to give warnings to motorcyclists about danger in blinspots. We could possibly have two displays, one on the right side of the helmet and one on the left side. Let's say there is an incoming vehicle coming from the left side of the motorcyclist, the display on the display will then display a warning symbol in order to make the motorcyclist aware that they shouldn't turn because there is an incoming vehicle.

Figure 35. Display Indicator

Non-transparent OLED screen displays could be a good choice. Many of them come small in size and they are clear to see. The display will need to be able to communicate effectively important information to the motorcyclist. If there is danger in the blind spot of the motorcyclist, it would say so in the display. If we decide to go with the display, we could also incorporate other features such as turning signals. Other than effectiveness and different features that can be incorporated into the display, we also must keep in mind other factors such as price, voltage and power consumption.

3.3.8.2 LED Indicators

An alternative to having a display to warn the user about hazards, we can use LEDs instead. LED's can be placed in the left side and right side of the helmet. For example, let's say there is a vehicle coming from the left side of the motorcyclist, there would be an LED light on the left side of the helmet that would go off or start flashing, this would warn the driver that there is an incoming vehicle and that they should not turn.

Using LED indicators is much easier to incorporate than incorporating OLED screens into the helmet. By using the LED indicators, we would also be saving in cost because their price is a fraction of the cost of OLED displays. Also compared to OLED displays, the operating voltage and power consumption of the LEDs would be significantly less. Although these have many advantages, one of the main drawbacks is that other than warning the motorcyclist about hazards, we wouldn't be able to incorporate extra features such as turning signals.

Figure 36: LED Indicators

3.3.8.3 Indicators Decision

Upon evaluation, we decided that we would have both the OLED indicator and LEDs. We would mount one LED screen into the helmet, and it would display information such as turning signals or warnings such as incoming vehicles in the blind spots. We would also mount two LED lights, one the right side and one left side. Their main goal is to warn the motorcyclist about hazards in the blind spots.

3.3.9 Speaker and/or Buzzer

A speaker or a buzzer or both can be incorporated into the helmet. Having these components in the helmet will help better incorporate the safety features it has. One of the goals is for the helmet to wake up or alert the user if they are falling asleep. By having a speaker or buzzer we would be able to achieve that. Not only that but if time permits, we could also have the speaker or buzzer go off in order warn the motorcyclist that there are hazards in the blind spots.

3.3.9.1 FIT0449 - Digital Speaker Module

A speaker that can be incorporated into the safety helmet could be the FIT0449 - Digital Speaker Module. This speaker has many advantages, from the specifications, we can see that it's small in size, it also has a simple interface and this speaker can also be used as a buzzer. If we were to use the speaker, we would only need one, and it could be placed on the left or right side of the helmet next to the motorcyclist ear.

The helmet is going to have cameras that are going to be able to detect if the motorcyclist is falling asleep. If the motorcyclist is falling asleep, then the speaker would go off to help wake up the driver. We want the sound coming out of the speaker to start at a low volume and then get progressively louder. We want it to be like this because if the speaker goes off at a loud volume, it might startle the motorcyclist and that might cause an accident. If time permits, we could also have sound coming from the speaker to warn the motorcyclist that there is danger in the blinspots.

Main Features:

- Operating Voltage: 2.0V -5.5v
- Interference: Digital:
- Module Size: 40mm x 40mm
- Power rating (Speaker Component): 0.5W
- Impedance (Speaker Component): 8ohms

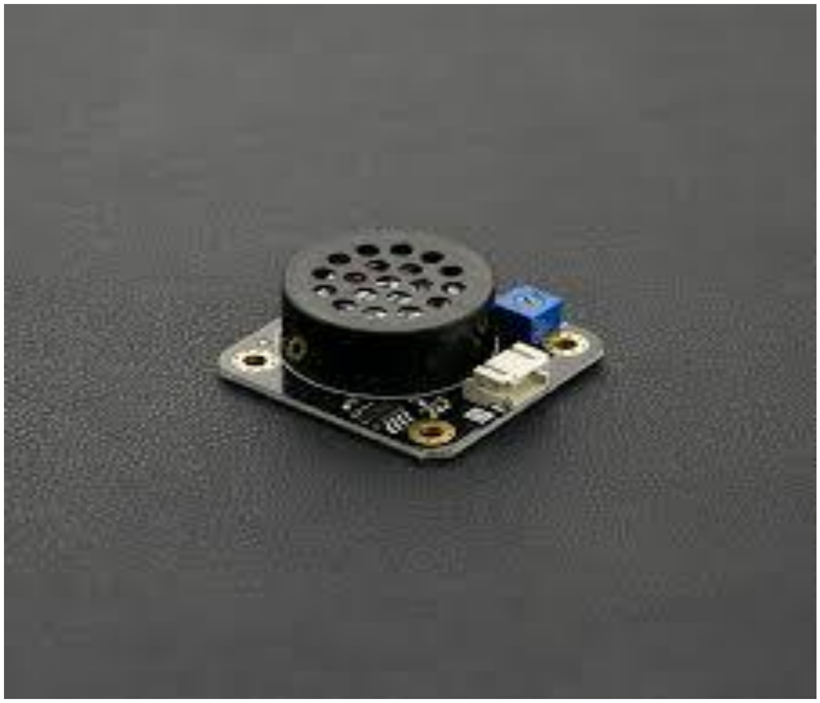


Figure 37: FIT0449 - Digital Speaker Module

3.3.9.2 1669 Stereo Enclosed Speaker Set

The 1669 Stereo Enclosed Speaker Set could also be another good choice that we could incorporate into the helmet. This module comes with two speakers compared to the FIT0449 - Digital Speaker Module which is only one. This speaker also requires a really low power consumption of 3W or less and it requires a low impedance of 4 ohm. Its interference is easy to use so therefore it would be easy to incorporate into the helmet. One of the drawbacks that it has is that upon researching this part, nothing could be found about whether or not this module could be used as a buzzer. Also, it's a little bit bigger in size and the cost a few dollars more compared to the FIT0449 - Digital Speaker Module.

Main Features:

- Weight: 25g per speaker
- Dimensions: 30mm x 70mm x 17mm
- Power: 3W or less
- Impedance Requirement: 4 ohms
- Cable length is 30’’
- 24mm x 64mm mounting rectangle

Figure 38: 1669 Stereo Enclosed Speaker Set

3.3.9.3 Wired Earbuds

The last option that we looked at are tried and true old-school wired earbuds. Apple EarPods rest comfortably inside and stay inside a variety of ear shapes and sizes. They have superior overall audio quality and even come with a built-in microphone and volume control.

Main Features:

- Weight: .2 pounds
- Ear Cushion Material: Silicone
- Cord Length: 3 feet

Table 11:Speaker Comparison

	FIT0449 - Digital Speaker Module	**669 Stereo Enclosed Speaker**	**Wired Earbuds**
Power Rating	0.5 W	3 W	1 W
Weight	30 g per speaker	25 g per speaker	90 g
Dimensions	40mmx40mm	30mmx70mmx17mm	7.5mmx7.5mm
Cable length	3.5 ft	4 ft	3 ft
Cost	$6.00	$7.50	$5.00

3.3.9.4 Playing Audio from The Microcontroller

There are many ways we could connect the headphones to play the audio. The headphones could be connected to the Raspberry Pi or they could be connected to the microcontroller. The best approach would be to connect the headphones to the microcontroller. The way that we are going to do this is by using a DAC converter that's going to be connected to the microcontroller and then we will be using an amplifier in order for the motorcyclist to be able to hear clearly. In the sections below we will be discussing the DAC that was chosen as well as the amplifier that was chosen.

Digital Analog Converter (DAC)

A Digital to Analog Converter (DAC) is a system that is used to convert digital signals to an analog one. The analog signal that the DAC will be outputting will be audio. Afterwards the DAC is going to be connected to an amplifier and connected to the amplifier, we are going to connect the headphones. The microcontroller doesn’t have a built in DAC and therefore we would need to incorporate one into the design. This DAC will be connected to one of the pins of the microcontroller and if drowsiness is detected, a signal will be sent in order to make sound to wake up the motorcyclist. We could design a DCA and incorporate it into our project, but the best way would be to buy a DAC that is already prebuilt. Currently in the market, there are many DAC modules that are already prebuilt and that would be easy to add to our project. The MCP4725 Breakout Board is a 12-bit DAC that would be ideal to include into our design. This module is controlled via the I2C interface and it requires a voltage supply of 3-5V.

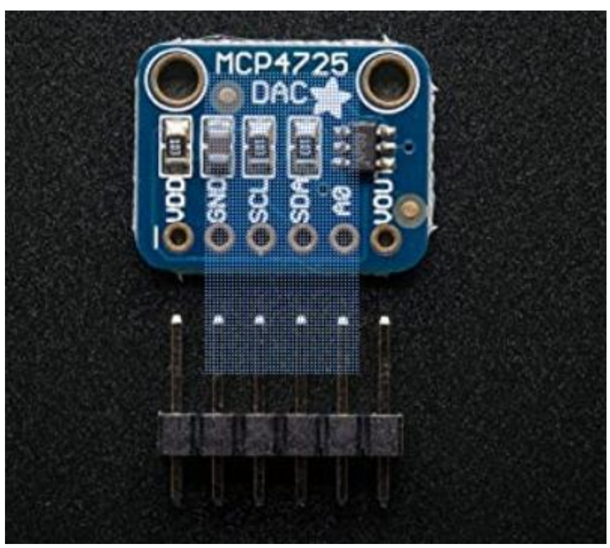


Figure 39: MCP4725 DAC

Audio Amplifier

As previously mentioned, after we connect the DAC into one of the pins of the microcontroller, we will also need an amplifier in order to be able to hear the sound. Just like with the DAC we could ourselves design the component and build it and make a PCB for it, but the best approach would be to get a prebuilt one. Currently in the market there are many ones that already come prebuilt and one of the most popular ones is the LM386 Audio Amplifier.

The LM386 chip is basically a low power audio frequency amplifier that requires a really low power. The microcontroller should be able to supply the chip with plenty of power in order for it to work. This specific module without any additional components can deliver an amplification gain of 20 which should be sufficient for the motorcyclist to hear.

3.3.9.5 Speaker/Buzzer Decision

After carefully examining the FIT0449 - Digital Speaker Module, the 1669 Stereo Enclosed Speaker, and the Wired Earbuds it was concluded that the Wired Earbuds are the best choice. We want the helmet to be as economical as possible, but safety should also be its main goal. Both the FIT0449-Digital speaker module and the 1669 Stereo enclosed speaker are really good choices and have better quality but the size and the difficulty of installing them was what made us go in different directions with them. The wired earbuds are the most affordable out of the three and fit the smart helmet the best.

3.4 Parts Selection Summary

The table below lists the different components and sensors that we decided to choose to include in our project. Our decisions based on the research that we conducted and based on what components will meet our requirements the best. We want the helmet to be as economical as possible, but we also want it to be reliable. Those two factors also played an important role in our part selection.

Since we got sent to an online form of classes while we started building the project, we have not been able to meet each other due to the COVID 19 stay at home orders. The four of us each have in our possession all the different components which are needed to build the Smart Helmet. The picture below shows what parts we each have and some that are still missing. The missing parts are still in transit and the proof will be shown to you in the appendix section.

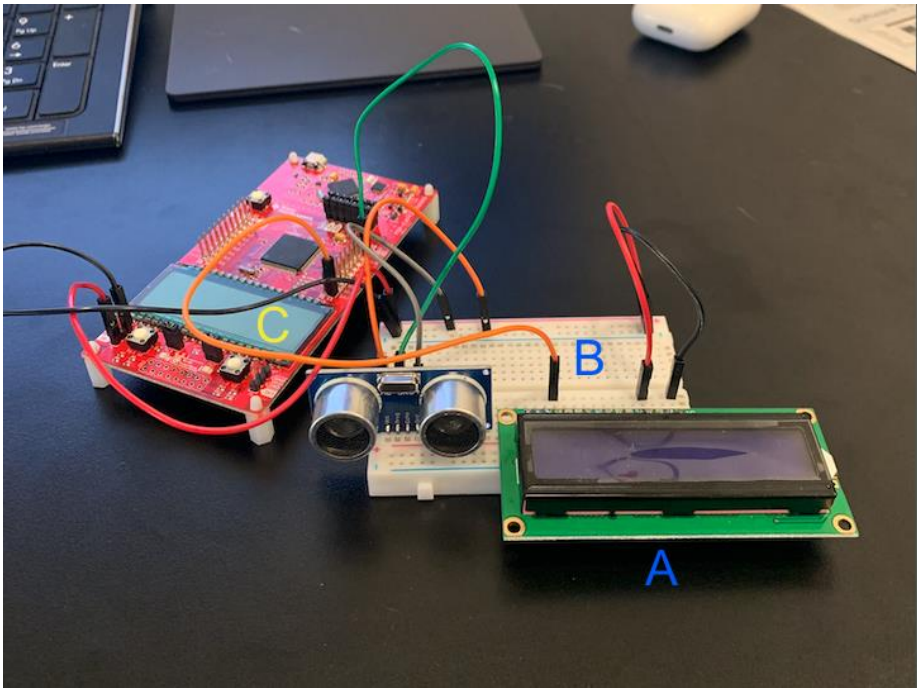


Figure 40: Components A,B, and C.

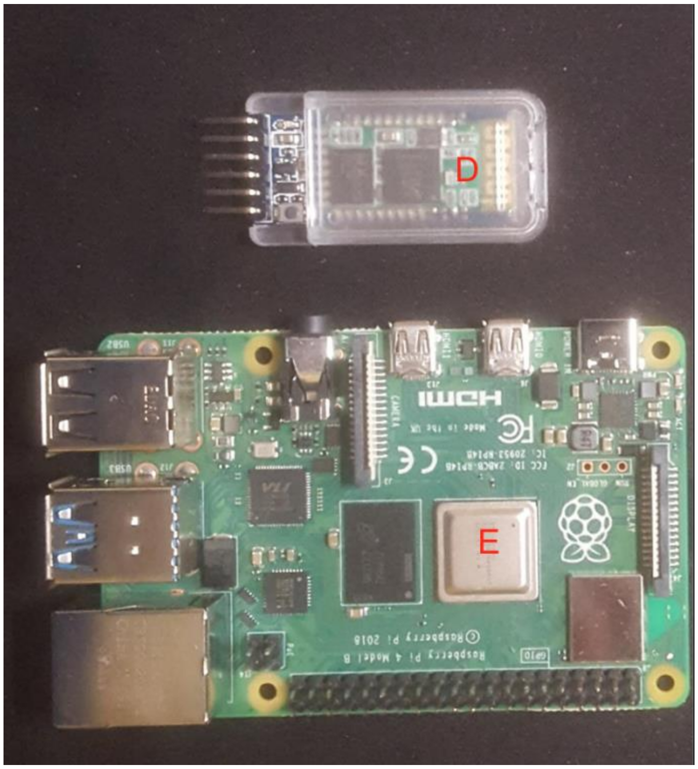


Figure 41: Components D, E.

Table 12: Parts Selection

	Sensor/Component Chosen
Proximity Sensor - B	HC-SR04 Ultrasonic Range Finder
Motion Sensor	MPU-6050 3 Axis Gyro
Transmission Hardware - D	HC-05 Bluetooth Module
Infrared Camera	Pi NoIR Camera V2 Module
Microcontroller - C	MSP430FR6989
Indicators - A	OLED display module and LED lights
Speaker/Audio	Earbuds
Raspberry Pi -E	Raspberry Pi

4. Standards and Constraints

In modern days when building a new product, it's important to follow standards. For this project, we decided to follow popular and widely used standards in communication, hardware realization and software. Following standards will ensure more modular design with possible improvements and adding features in the future. In this section we will describe the standards that will be used for this project.

4.1 Related Standards

In this section we discuss standards that are related to our projects and we plan to comply with them. We first cover networking standards used for our Bluetooth module and after that we'll discuss standards related to Android development in Java.

4.1.1 IEEE Project 802.15.1 Wireless Personal Area Network Standard

What is IEEE?

The institution of electrical and electronic engineers is a professional association for electronic engineers and electrical engineers and associated disciplines. It is the largest technical professional society in the discipline, and it is designed to connect between technical professionals to perform conferences and technology standards.

IEEE 802.15.1 Current Status

IEEE Project 802.15.1 is derived from the Bluetooth 1.1v foundation specification, the IEEE Std 802.15.1™-2002 standard was published on June 14th, 2002. All IEEE standards are updated every five years on the date of publications. If the standard is not revised it will be submitted to the Standards Board for administrative withdrawal.

IEEE 802.15.1 Overview

The new IEEE Std 802.15.1™-2002 standard is a resource for those who implement Bluetooth devices. In the standard, the lower transport layer is defined. Bluetooth is an industry specification for a short-range RF based wireless communication for portable personal devices. The IEEE provided a standard adaptation for the Bluetooth Specification 1.1v Media Access Control (MAC).

IEEE 802.15.1 Standard Security

This standard is made for Wireless Personal Area Network (WPAN) which is used to transmit information over short distances. One of the main risks when using wireless communication is security. Since data packets are transmitted using RF it can be intercepted and stolen. To combat this vulnerability IEEE defines security measures to be taken when implementing Bluetooth hardware. The definition is covering all layers of the WPAN network providing a secure encrypted data transmission. Since it is done in a peer-to-peer network, security is embedded into the application layer and as well as the link layer. When pairing is requested there is an authentication procedure that is responsible to authenticate between two devices. The generic authentication procedure describes how the Link Manager Protocol (LMP) authentication and pairing procedures initiate a link between one Bluetooth device to another. The procedure depends whether a link already exists or not or if pairing is allowed or not.

The procedure is provided in the following flow chart:

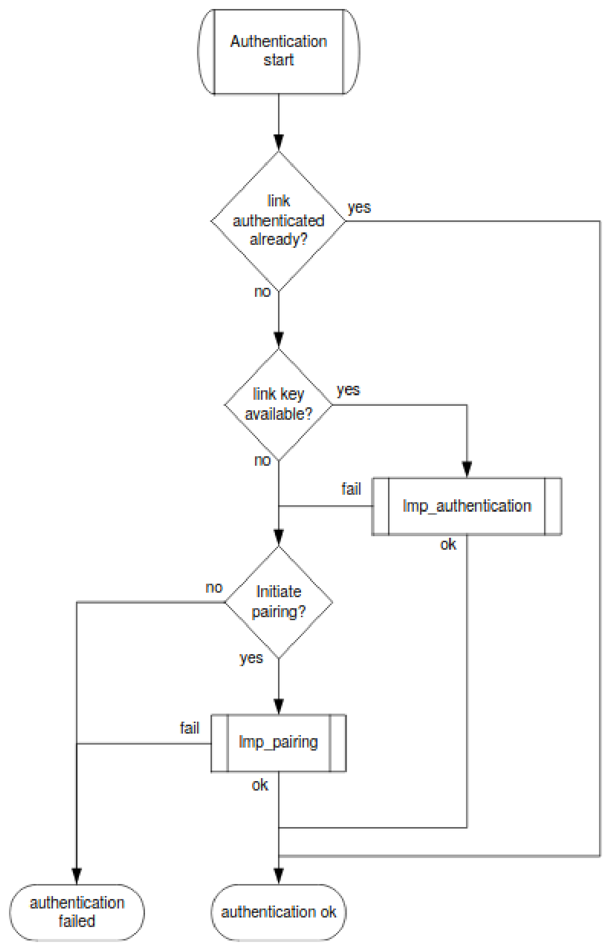


Figure 42. Definition of generic authentication procedure

In order to create a link, there are three security modes used.

Security mode 1 (non secure): When a device is in security modes one it should never initiate a security procedure.

Security mode 2 (service level enforced security): When a Bluetooth device is in security mode 2, it shall not initiate any security procedure before a channel establishment request (L2CAP_ConnectReq) has been received or a channel establishment procedure has been initiated by itself. Whether a security procedure is initiated or not depends on the security requirements of the requested channel or service.

Security mode 3 (link level enforced security): When a Bluetooth device is in security mode 3, it shall initiate security procedures before it sends LMP_link_setup_complete. A Bluetooth device in security mode 3 may reject the host connection request based on settings in the host (e.g., only communication with pre-paired devices allowed).

4.1.2 Android Standard

Android is an operating system funded by Google originally developed by Android Inc from Palo Alto, California. Android is a mobile operating system based on a modified version of the Linux kernel. It is primarily designed for touchscreen mobile devices such as smartphones and tablets. Android is an open source software (except proprietary software developed by third parties such as Google) and it is developed by a consortium known as Open Handset Alliance, with the main contributor and commercial marketer being Google. Developing software for Android is officially done with Java some rules and guidelines to ensure that Java coding criteria are met. There are rules and guidelines for Java development and as well as development for the Android ecosystem.

The Android company has described the rules and guidelines for Android Java developers which are important to follow in order to keep the Android experience consistent and easy to the user. It is also important for the Android ecosystem and the stability of applications developed for it. The guidelines are described with great details and examples of good and bad code so a developer can easily implement them in their project.

1. Don't ignore exceptions

When developing in Java we often need to deal with exceptions, and it is very tempting to ignore them.

```
void setServerPort(String value) throws NumberFormatException {
    serverPort = Integer.parseInt(value);
}
```

In the code above, we throw the exception to the caller of the method.

If an access to I/O device is performed it’s important to use, try-catch block in order to catch the exception in case the I/O device is not available. Avoiding exceptions can cause a program to crash or unpredictable behavior.

2. Don't catch generic exceptions

There are several different types of exceptions among them are generic exceptions and error exceptions. When using try-catch block to catch an exception it is not recommended to catch both types in one try block.

```
try {
   someComplicatedIOFunction();        // may throw IOException
    someComplicatedParsingFunction();   // may throw ParsingException
    someComplicatedSecurityFunction();  // may throw SecurityException
    // phew, made it all the way
} catch (Exception e) {                 // I'll just catch all exceptions
    handleError();                      // with one generic handler!
}
```

In the example above we can potentially have several types of exceptions thrown and we try to catch all types with one generic exception. This is not recommended since different types of exceptions might require different code to handle the exception.
So, the alternative way to catch different types of exceptions is to use several separate blocks such as this.

```
try {
    ...
} catch (ClassNotFoundException | NoSuchMethodException e) {
    ...
}
```

Each block handles a specific type of exception and avoids generic exceptions.

3. Don't use finalizers

Finalizers are a way to have a chunk of code executed when an object is garbage collected. While finalizers can be handy for cleanup (particularly of external resources), there are no guarantees as to when a finalizer will be called (or even that it will be called at all).

Android doesn't use finalizers. In most cases, you can use good exception handling instead. If you absolutely need a finalizer, define a close() method (or the like) and document exactly when that method needs to be called (see InputStream for an example). In this case, it's appropriate but not required to print a short log message from the finalizer, as long as it's not expected to flood the logs.

3. Fully qualify imports

In programming for Android, we normally use libraries and classes from the Android framework. The correct way to import classes is to use the full name of the package and class we want to use. For example, if we want to import 'Bar' class from the package 'foo' we can import in two ways:

- `import foo.*;`

This includes everything under the package 'foo'.

- `import foo.Bar;`

This only includes the class Bar which makes the code more readable. An explicit exception is done for Java libraries such as (`java.util.*`, `java.io.*`, etc.) since they're in use frequently.

4.1.3 Java library rules

There are conventions for using Android's Java libraries and tools. In some cases, the convention has changed in important ways and older code might use a deprecated pattern or library. When working with such code, it's okay to continue the existing style. When creating new components, however, never use deprecated libraries.

4.1.4 Java style rules

1. Use Javadoc standard comments

Every file should have a copyright statement at the top, followed by package and import statements (each block separated by a blank line), and finally the class or interface declaration. In the Javadoc comments, describe what the class or interface does.

```
/*
* Copyright 2019 The Android Open Source Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package com.android.internal.foo;

import android.os.Blah;
```

```java
import android.view.Yada;

import java.sql.ResultSet;
import java.sql.SQLException;

/**
 * Does X and Y and provides an abstraction for Z.
 */

public class Foo {
        ...
}
```

The example above shows the format of the comment each file must have.

2. Write short methods

When possible it’s important to keep methods small and focused. Since sometimes there is a need for a long method there isn’t any hard limit, but it’s recommended to break up the method if it doesn’t hard the main structure of the program.

3. Define fields in standard places

Fields should be defined at the top of the file or immediately before the method that uses them.

4. Limit variable scope

It’s important to limit variables scope to a minimum as possible. Doing that increases readability and reduces the chance of making errors. Declare local variables as close as where they are being used and as soon as declared initialize. If it’s not possible to initialize (maybe because we don’t know what for yet) we should declare the variable later when we do and then initialize.

One exception is when using try-catch blocks. If a variable is initialized with the return value of a method that throws a checked exception, it must be initialized inside a try block. If the value must be used outside of the try block, then it must be declared before the try block, where it can't yet be sensibly initialized.

5. Order import statements

The ordering of import statements is as follow:

1. Android imports
2. Imports from third parties (com, junit, net, org)

3. java and javax

The reason for this order is that since people want to first look at Android imports and the imports people want to look at least are Java imports. It's easier for people to follow this style and also easier for IDEs to organize imports in this way.

6. Use spaces for indentation

We use four (4) space indents for blocks and never tabs. When in doubt, be consistent with the surrounding code. We use eight (8) space indents for line wraps, including function calls and assignments

7. Follow field naming conventions

- Non-public, non-static field names start with m.
- Static field names start with s.
- Other fields start with a lowercase letter.
- Public static final fields (constants) are ALL_CAPS_WITH_UNDERSCORES.

8. Use standard brace style

Braces should be at the same line as the code before them and not on their own line.

Example:

```
class MyClass {
      int func() {
     if (something) {
             // ...
      } else if (somethingElse) {
          // ...
      } else {
             // ...
       }
      }
}
```

Also, there should be braces for conditions even when there is only one line inside the condition unless it is possible to fit the entire condition with its inside in one line.

9. Limit line length

The length of each line in the code should be at most 100 characters long. If a comment is longer than 100 characters and contains a URL it is allowed to have a longer than 100 characters line to allow easy copy and paste. Also import statements are allowed to be longer than 100 characters since humans rarely read them.

10. Use standard Java annotations

It's important to use standard java annotations for language elements. Annotations should precede other modifiers for the same language elements.

Standard Java annotations:

- Use the `@Deprecated` annotations whenever a section of code is older the current version of Java used for this program.
- Use the `@Override` whenever a method is overriding another method from superclass.
- Use the `@SuppressWarnings` whenever it's not possible to eliminate a warning so it's easier to recognize other warnings.

11. Treat acronyms as words

Treat acronyms and abbreviations as words in naming variables, methods, and classes to make names more readable. Since SDK and Java acronyms are inconsistent it's impossible to be consistent with the surrounding code therefore, we need to treat acronyms as words.

12. Use TODO comments

When working on a project we often need to break down our work to smaller steps therefore adding a TODO comment is a good way to break up the work and assign different parts to the team members.

13. Log sparingly

When debugging and testing the code we often need to print a log or stack trace. There are several types of logs we can print:

- ERROR: Is used when something fatal has happened, that is, something that causes the app to crash or something that is visible to the user and can only be recoverable by a loss of data.
- WARNING: When something serious and unexpected happens, for example, if it has a user visible consequence but it will not cause a loss of data.
- INFORMATIVE: Is used when something that has a great impact on the program is happening.
- DEBUG: Is used when important actions are happening that can be useful to identify errors.
- VERBOSE: Is used for anything else that is not one of the above.

4.2 Realistic Design Constraints

Every project has designed limitations, these limitations will be explored here. While we are not experienced engineers, we're learning how to manage our project while staying within time and cost constraints. Our main goal is to keep cost as low as possible while realizing all the features we intend to implement in our project. For our project we have some additional difficulty while developing our project. In December 2019 an COVID-19 epidemic spread across the world and around the beginning of March 2020 university transformed to distance learning which required us to reconsider some of our design choices and implementations. In the following few pages we'll discuss design constraints that greatly affect our project. The constraints are limitations that have to be applied to our project. These constraints include but are not limited to the time duration in which the project had to be done, Environmental limitation which limited us in acquiring parts and ordering custom parts such as PCB. There are several main types of constraints that have to be considered when designing a product:

- Health
- Time
- Safety
- Social
- Political
- Environmental
- Economic
- Ethical
- Sustainability

4.2.1 Economic and Time Constraints

Since we didn't have any sponsorship for our project, we had to fund this project out of our own packets which required us to minimize cost of material and features in the final project. Many of us are currently not working and school is our full-time occupation, so we did not have much money to spend on the project. We chose to use some of the hardware that some of us already have from previous classes such as the microcontroller used in previous classes. For the rest of the hardware that we did not have to equally split the cost between all the members. We also chose cheaper parts such as ultrasonic sensors instead of IR based distance sensors and had to compromise accuracy.

Since we all are still full-time students and have responsibility for other classes, we had to find time to work on our project. We kept regular meetings with the group members and made sure we are still on track and fulfilling deadlines we decided on. Our project span over spring and summer semester and since summer semester is usually shorter it is shortening our project to seven months. In order to finish our project in time we decided that all the group members have to keep deadlines and finish their parts according to the plan. We kept in contact using several modes of multimedia communications such as Google Hangout, WhatsApp and Discord.

4.2.2 Environmental and Social Constraints

This specific constraint played an important role in our project. In December 2019 the COVID-19 epidemic started to spread across the world from China. On January 21st the first confirmed case in the United States soon after during Spring break schools and universities around the world changed to distance learning which impacted our project greatly. First, we couldn't access the lab which prevented us from properly testing our design choices and components we ordered. Second, is the impact of the epidemic on the economy. Many job places stopped or reduced their workload significantly in order to prevent the virus from spreading. As a result, our parts took a long time to arrive and, in some cases, we had to cancel our order and reorder from other places. Despite these limitations we're confident that we can finish this project, however, we might need to remove some features that we decided on initially.

Another constraint that can be included in this section would be the political constraints which fall a little in the social side. Some of the issues that political constraints cover would be products that are going to be used against our government or banned by the use of other governments. Quotas, taxes or subsidies on the ability of specific suppliers or technologies to change their capacity in response to change in demand. Our Project does not fall into the political constraint and therefore we did not talk about it much here.

4.2.3 Health and Safety Constraint

Another side effect of the COVID-19 epidemic forces us to test hardware at home with limited equipment and some guesswork. Dealing with electronic circuits can pose danger of electrocution or violent exothermic chemical reaction in li-ion batteries. Our end goal of the project is aimed to use lead-acid batteries built in every motorcycle with operating nominal voltage of 12 volt.

In order to simulate that we might need to use Li-ion batteries since one of our group members already have them and ordering a new lead acid battery is against our goal of keeping the cost as low as possible. Even though our end goal is to use the built-in battery on a motorcycle we might be forced to make our project standalone product with its own battery. If we are forced to do that, we might consider placing the Li-ion battery outside the helmet or attached to the motorcycle's body far from the head. Since an explosive reaction of a Li-ion battery inside the helmet can be lethal.

There are several important properties of batteries that need to be considered:

- Continues discharge current
- Over voltage and low voltage
- Internal temperature

The main concern with batteries and especially Li-ion batteries is continuous discharge current. This limitation is a very important constraint which might require us to have more batteries to distribute the discharge current. The main cause of explosion of Li-ion batteries is high discharge current. Since the chemistry inside these batteries is a reaction of highly unstable metals such as lithium, discharge current measures have to be taken into account such as: reverse-polarity protection (using diodes), short circuit protection and proper heat dissipation. The Raspberry Pi is a power-hungry device which requires up to 3A of current. Therefore, a battery that can deliver more than 3A of continuous discharge current is required.

The second reason which can cause batteries to leak or react violently is overvoltage or very low voltage. For example, 18650 Li-ion batteries have a nominal voltage of 3.7V, most manufacturers recommend that a maximum voltage (voltage of battery fully charged) not to exceed 4.2V fail to meet this requirement can cause the battery to explode. Also, the battery should not be discharged to a voltage lower than 2.9V which can seriously damage the battery or cause it to react violently.

When discharging a high current from a battery, heat is generated as a result of the reaction which in turn can accelerate the reaction and cause damage or explosion. In order to overcome that a proper heatsink will be used to dissipate the heat properly and safely. Additionally, Li-ion batteries should not be left under the sun which can cause it to heat and cause damage.

4.2.4 Manufacturability Constraints

Manufacturability constraint is the availability of making or ordering hardware in a project. For us, as discussed before, we had trouble getting some parts on time or finding certain parts. We were required sometimes to order parts with a higher price so they could arrive faster for testing. Despite that, we still received some parts late which caused some delays in the development of our project, but we are still on track and expect to complete the project on time. As of completion of this Senior Design I report all of the remaining parts are currently in transit and have documentation showing that they will be arriving to us before the start of Senior Design 2 next semester.

Not only do we have to take into consideration the availability of the parts during the current states, but we also have to make sure that we have availability for any parts in the following summer semester in Senior Design 2. This semester we only had the chance to do minimum testing due to the COVID-19 pandemic and we plan on doing the rest of the testing in the coming months. We had to make sure to order extra components parts of each one so that if in the future something breaks or is not working correctly, we are able to replace the component with no problem. Some of the parts can become discontinued, or just are no longer in the market. That would pose a pretty big problem since we did our research and testing with all of the parts that we currently have and cannot afford to wait several weeks for new parts due to the summer semester being shorter than normal.

5. Project Hardware and Software Design Details

In this section, we would be outlining how we are going to design and test the different hardware and software components for the Smart Helmet respectively. For hardware, we will detail how we intend to implement our power supply and related circuits in order to supply power to the different components. We will also detail how we intend to do the connections for the PCB design. For software, we will outline how we intend to implement the programming for the microcontroller and the Raspberry pi, and also the design for the application.

5.1 Power Supply

A power supply is used to supply power to electrical loads. The power supply converts the electric current from a certain source to a voltage in order to power electric loads. The source power of the power supply varies. Examples include electrical outlets or energy storage devices such as batteries. Two main types of power supply are DC power supply and AC power supply.

5.1.1 DC vs AC

AC or alternating current power is that standard form of electricity that comes out of outlets. The composition of the AC wave comes from the flow of electrons. As the electrons move, they can move in a positive or negative flow depending on the sinusoidal wave created by the current. The waves come from alternators at electrical power plants creating the AC power. The spinning inside of the alternators create waves of alternating current when the wire moves into areas of different magnetic polarity.

In a DC power supply, current only flows in one way whereas in an AC power supply, current changes direction. Direct current uses electrons that move in a straight line. This linear movement , in contrast to AC wave motion gives the current its name. DC current comes from batteries, solar cells, fuel cells, and alternators. Most electronics require this power type and is why most electronics have a DC power source that is in the form of batteries or need to convert the AC power from outlets to DC power through a rectifier. It is preferred for electronics because ACs highs and lows of alternating current can damage the insides of them.

If a DC voltage is desired, and an AC power source is used for the DC power supply, an AC-to-DC would be required in order to obtain a DC output voltage. With an AC power supply, the power comes from power plants and the power is available in the wall sockets. In the USA, the power available in the wall socket is 120-volt, 60-cycle AC power.

For the Smart Helmet, we would be using a DC power supply.

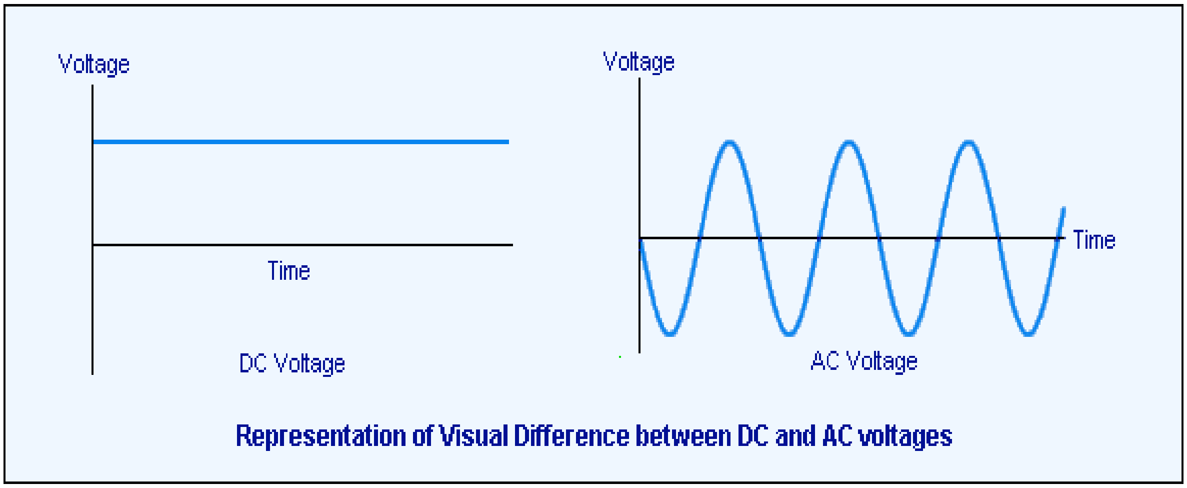


Figure 43: DC vs AC voltages

5.2 First Subsystem: Design of the Power Supply

The goal for our design is to have the battery of the motorcycle power up all the components for our design. The reason for this design is so that the Smart helmet user does not have to constantly charge the helmet after each time that they use it. At first, we thought about incorporating a portable battery, but we changed our mind when a lot of issues we saw would come about due to this.

Many motorcycle batteries are 12-volt batteries. The components for the Smart Helmet require way less than the 12V. Therefore, we would have to design a voltage regulator. Some of the components require different voltages to operate so the voltage regulator would have different outputs.

Since we would be working with a DC power supply, we don't have to worry about making an AC to DC converter circuit. A DC to DC circuit might be required in order to convert the 12V to a lower voltage for the components. A DC power supply such as the ones that are available in the engineering labs could be used to test our design and to simulate the 12V motorcycle battery.

Figure 44: Motorcycle Battery

Figure 45: Triple Output DC Power Supply

5.2.1 Voltage regulators

As it was stated in the previous section, our power supply is of 12 V but the various components that are in our project require well below the 12V. Voltage regulators such as the one in the figure below would need to be included in our design in order for our project to be successful. A voltage regulator is designed to output a certain voltage level.

This section will outline the different power, current and voltage requirements of the different components and also the design of the voltage regulator circuit will be discussed. There are 2 main types of voltage regulators, there are linear voltage regulators and switching voltage regulators. Each of them have a different design and they have their advantages and disadvantages.

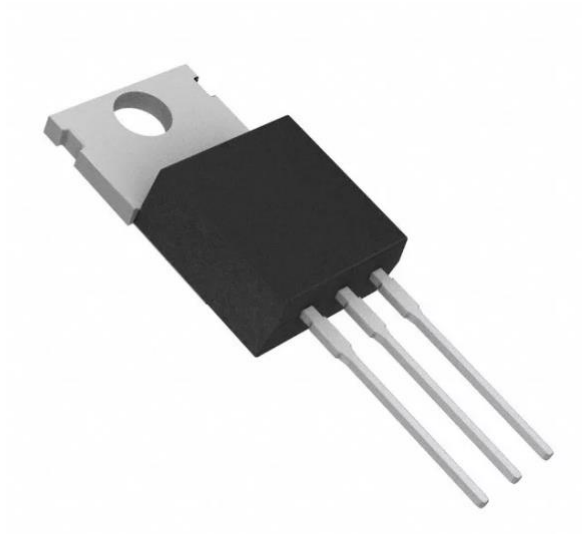


Figure 46: Example of voltage regulator

Linear Voltage Regulator

Linear voltage regulators are mostly used to power devices that are very low power devices where the difference between the input voltage and output voltage is low. This kind of regulator tends to be inefficient because that difference between the input voltage and output voltage is decapitated by heat. Since our input voltage is going to come from the 12V battery and some of our components require 5V, other components required 3.3V and others around 2V, a linear regulator would not be a good choice. The difference between the output voltage and input voltage would be too much and too much energy would be dissipated as heat which could cause problems to the PCB.

Some advantages of the linear regulator are its simple circuit configurations, few external parts and its low amount of noise that it produces. Some disadvantages are its relatively poor efficiency, considerable heat generation and that it only step down (buck) operation.

Switching Voltage Regulator

Switching regulators are way more efficient than linear voltage regulators. This kind of regulator works more effectively when working where the difference between the input and output voltages is high. This type of regulator would make more sense to include in or project.

The advantages of the switching voltage regulator are its high efficiency, low heat generation and boost, buck, and negative voltage operation. Some disadvantages will be more external parts are required, having a complicated design and the increase of noise that it produces.

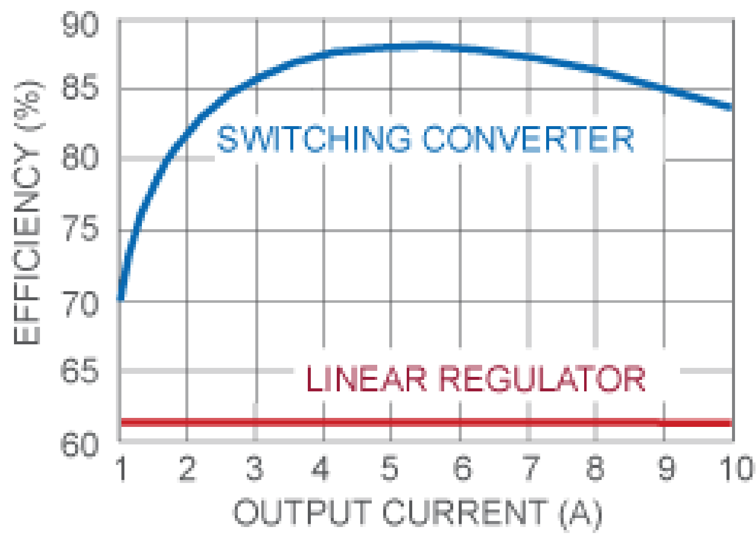


Figure 47: Efficiency of Voltage Regulators

This figure above is a visual representation of the efficiency vs the current of the linear voltage regulator and the switching voltage regulator. Upon closer inspection we can see that the efficiency is better with the switching voltage regulator compared to the linear regulator.

Comparison Between the Voltage Regulators

This table goes more into detail about the two kinds of voltage regulators. It describes their different specifications and based on that, we can see the advantages and disadvantages.

Table 11: Voltage Regulator Comparison

	Linear Regulator	**Switching Regulator**
Design Flexibility	Buck	Buck, Boos, Buck-Boost
Efficiency	Normally low to medium-high for low difference between Vin-Vout	High

Complexity	Low	Medium to High
Size	Small to medium, larger at higher power	Small at similar higher power(depending on the switching frequency)
Total Cost	Low	Medium To high-external components
Ripple/Noise/EMI	Low	Medium to high
Vin Range	Narrow(Depending on power dissipation)	Wide

5.3 Second Subsystem: Microcontroller, Sensors, and indicators

In this part, it's going to be discuss how the microcontroller and the different sensors such as the ultrasonic sensors and the gyroscope is going to interact with the microcontroller and it’s also going to be discuss the power requirements that would be needed in order of the design to be successful.

The microcontroller is going to be in charge of controlling the ultrasonic sensor, the gyroscope, and the visual and audio indicators. The ultrasonic sensor is going to detect objects in the blindpots and if the objects are in close proximity, on the OLED display, it's going to warn the motorcyclist that there is an incoming vehicle and there are going to be some LEDs that are going to go off inside of the helmet. The gyroscope is going to be used for accident detection. If there is an accident, 911 or an emergency contact will be detected. It will do this by detecting if the vehicle has come off its axis for a period of time that will be set as a cut off. The microcontroller requires an operating voltage of 3.3 or 5 V as do the other components that it will be controlling.

5.3.1 Microcontroller

The Smart Helmet will be controlled via two different forms. The first will be the MSP430FR6989 MCU. We will use the development kit board that Texas Instruments has available to program and physically test out the MCU along with the rest of the components that will be specifically tasked to control the proximity sensor, motion sensor, OLED display, LED lights and audio.

In order to get the MSP430FR6989 up and running we will use Code Composer Studio which is an integrated development environment that supports TI's microcontrollers. It provides a C compiler, source code editor, project build environment, debugger, profiler and many other features. Texas Instruments offers a great amount of resources on their website to make sure the microcontroller being used is working correctly and if you come into any problems, they will be able to assist you.

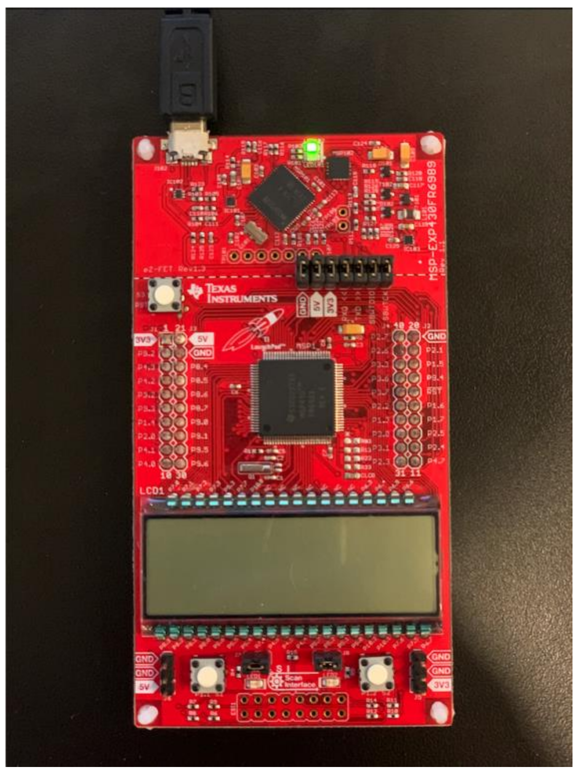


Figure 48. MSP430FR6989

Once the code is finalized, we will proceed by testing out each of the components that the microcontroller will be tasked to control. Each component will be connected to the microcontroller via a breadboard and tested indoors and outdoors. This will be detailed more in the following sections below specifically for each component under the software testing.

5.3.1.1 Microcontroller Power Requirements

The table shows the different recommended operation conditions such as voltages that would be needed in order for the microcontroller to be functional.

Typical data are based on V_{CC} = 3.0 V, T_A = 25°C unless otherwise noted.

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage range applied at all DVCC, AVCC, and ESIDVCC pins (1) (2) (3)		1.8 (4)		3.6	V
V_{SS}	Supply voltage applied at all DVSS, AVSS, and ESIDVSS pins			0		V
T_A	Operating free-air temperature		–40		85	°C
T_J	Operating junction temperature		–40		85	°C
C_{DVCC}	Capacitor value at DVCC and ESIDVCC (5)		$1_{-20\%}$			µF
f_{SYSTEM}	Processor frequency (maximum MCLK frequency) (6)	No FRAM wait states (NWAITSx = 0)	0		8 (7)	MHz
		With FRAM wait states (NWAITSx = 1) (8)	0		16 (9)	
f_{ACLK}	Maximum ACLK frequency				50	kHz
f_{SMCLK}	Maximum SMCLK frequency				16 (9)	MHz

Figure 49: Microcontroller power specifications

By studying the figure above, we can see that the supply voltage has to be between 1.8V and 3.6 V. We are going to be using a supply voltage of 3.3V. As stated in the previous section, a voltage regulator will be necessary in order to convert the 12V DC voltage from the battery supply to a voltage of 3.3V We need also enough power to power up the microcontroller so therefore 3mA is going to be used in order to make sure that we have enough power for the microcontroller to be functional.

5.3.2 Sensor Requirements

The two sensors that we are going to be using are the ultrasonic sensor and the gyro sensor. Both of these are going to be connected to the microcontroller and they are going to provide us with important measurements. Each of these sensors require a different voltage and power specifications in order for them to work properly. In this section we are going to discuss those specifications.

5.3.2.1 Ultrasonic Proximity Sensor

The proximity sensors will be tested indoors and outdoors to be able to get the most accurate measurement possible. Indoors testing will happen at the open senior design lab at the University of Central Florida. In order to test out the proximity sensors they will be attached to a breadboard. The breadboard will allow us to connect jumper wires to the proximity sensors and the microcontroller board. We must make sure we have the sensors configured correctly with the MSP430FR6989 which involves the correct input and output pins and mode of operation. We will verify that the sensors power on and off. Once that is done the sensors will be tested out at five different ranges and multiple rounds so we can make sure that we are obtaining the best possible and accurate readings. That will be discussed more in detail in the software design testing section.

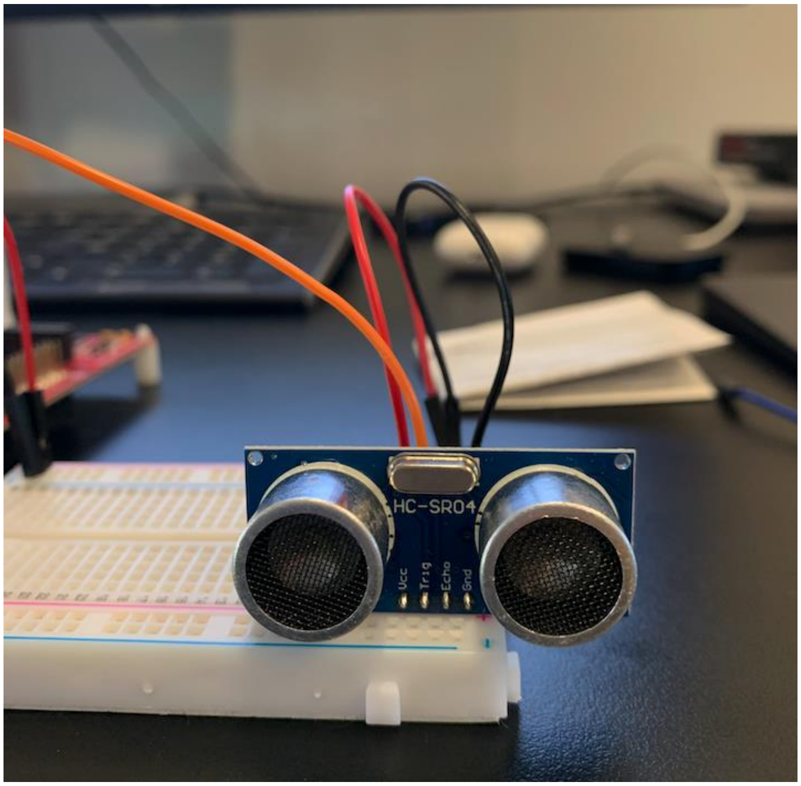


Figure 50. Proximity Sensor testing

HC-SR04 Ultrasonic Range Finder Power

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Figure 51: HC-SR04 Ultrasonic Range Finder power specifications

The figure above, discusses the power requirements in order for the ultrasonic range finder to work. We can see that the working voltage is DC 5V and the working current requires 15mA. The motorcycle battery provides a 12V DC voltage. Because the microcontroller's working voltage is 3.3V, and the working voltage for the sensor is DC 5V, the voltage regulator is going to need an output of 3.3V and 5V. Also, we make sure that the power regulator outputs a current of 15mA in order to make sure that we have the required specifications.

The ultrasonic range finder has 4 pins. The Vccpin must be connected to the 5V power supply, the Trigger pin must be connected to a trigger pulse in put pin in the microcontroller, the Echo pin must be connected to an echo pulse output pin in the microcontroller, and finally the ground pin must be connected to a 0V ground pin in the microcontroller.

5.3.2.2 Gyro Motion Sensor

The Motion sensor testing will be performed indoors and outdoors in the same manner that the proximity sensors will be tested. The indoor testing will take place in the Senior design open lab and we will be using jumper cables, breadboards and our MSP430FR6968 microcontroller board. First thing we must test out is the communication between the microcontroller and the motion sensor. Just like the proximity sensor we must make sure we have the correct output and input pins connected along with the appropriate interface. Once the microcontroller and motion sensor are communicating with each other and functioning correctly the actual testing of data will start. The motion sensors job is to detect and measure when a crash has happened. Two different forms of testing will be conducted on the motion sensor. Axis rotation and G force impact on the Smart Helmet.

MPU-6050 3 Axis Gyro Power

The figure below details the specifications for the gyroscope that we will be using. From the specifications we can see that the operating voltage range is 2.375V-3.46V. It will also require a 3.6mA operating current. Because of the range of the voltage, 3.3V is the voltage that we are going to design for the sensor. Since the microcontroller also requires 3.3V, the sensor can use the same output of the voltage regulator that the microcontroller will be using.

6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0		±250		°/s	
	FS_SEL=1		±500		°/s	
	FS_SEL=2		±1000		°/s	
	FS_SEL=3		±2000		°/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)	
	FS_SEL=1		65.5		LSB/(°/s)	
	FS_SEL=2		32.8		LSB/(°/s)	
	FS_SEL=3		16.4		LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±20		°/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		°/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		°/s	
Linear Acceleration Sensitivity	Static		0.1		°/s/g	
SELF-TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	1
GYROSCOPE NOISE PERFORMANCE	**FS_SEL=0**					
Total RMS Noise	DLPFCFG=2 (100Hz)		0.05		°/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to10Hz		0.033		°/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		°/s/ √Hz	
GYROSCOPE MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
LOW PASS FILTER RESPONSE	Programmable Range	5		256	Hz	
OUTPUT DATA RATE	Programmable	4		8,000	Hz	
GYROSCOPE START-UP TIME	**DLPFCFG=0**					
ZRO Settling (from power-on)	to ±1°/s of Final		30		ms	

Figure 52: MPU-6050 3 Axis Gyro Specifications

5.3.3 Indicators Power Requirements

As discussed previously, we want the helmet to warn the user if there are any hazards in the blindpots. In order to do that, we are going to be using indications. We are going to have an OLED screen that is going to display if there is a hazard in the blind spot. Also, we are going to have some LED lights that are going to go off if there is a hazard. In section the power requirements of these components are going to be discussed. The LED lights will be tested by attaching them to the breadboard and using the jumper wires to attach them to the microcontroller. The first test we must do is make sure that the power on and off. After this is confirmed we make sure that they receive data from the microcontroller. We do this by sending a command for it to turn on and off through the microcontroller.

LED Power Requirements

The LED lights are going to serve as a warning that there is a hazard in the blind spot. We intend to put one on the left side and one on the right side of the helmet. If there is a hazard, then the LED light is going to light up and it's going to warn the motorcycle that there is a hazard in the blind spot. We order LED's of different colors. The table below summarizes the specification of each colored LED.

Specification:

Color	Size	Wavelength(nm)	Voltage(V)	MCD	Lens color
Red	5mm	620-630	1.8-2.3	800-1000	Red diffused
Yellow	5mm	580-590	1.8-2.3	800-1000	Yellow diffused
Green	5mm	520-530	2.8-3.6	800-1000	Green diffused
White	5mm	6000-6500k	2.8-3.6	14000-16000	Clear
Blue	5mm	460-470	2.8-3.6	800-1000	Blue diffused

Figure 53: LED light specifications

If we look at the chart, we can see that some colors have different voltage ranges. For example, the red color LED light has a voltage range of 1.8V -2.3V compared to the blue LED light that has a voltage range of 2.8V-3.6V. To fit our power supply specifications better, it would be best to choose a green, white or blue LED light since the voltage range is 2.8V-3.6V. If chosen, then it can be connected to the voltage regulator with the output of 3.3V. That would be the same output that the microcontroller will be connected to. We must also be conscious that the current won't be too high, otherwise it might damage the LED light. The same 3.0mA current that the microcontroller will be receiving should be enough to power the LED.

OLED Screen Power Requirements

The OLED screen that we are going to be using is going to be mounted in front of the helmet, so it is easy for the user to see. This screen has 4 pins. It has a GND pin that's going to be connected to the power ground, it has a VCC pin that's going to be connected to a 3V-5V power supply, it has a SCL pin that's going to be connected to the CLK clock and it has a SDA pin that's going to be connected to the MOSI data.

Since the range of the voltage supply is from 3V-5V, it'll be fine to use the 3.3V output in the voltage regulator. As previously mentioned, the gyro and the microcontroller also require a voltage of 3.3V. All these components can be connected to the same 3.3V output in the voltage regulator.

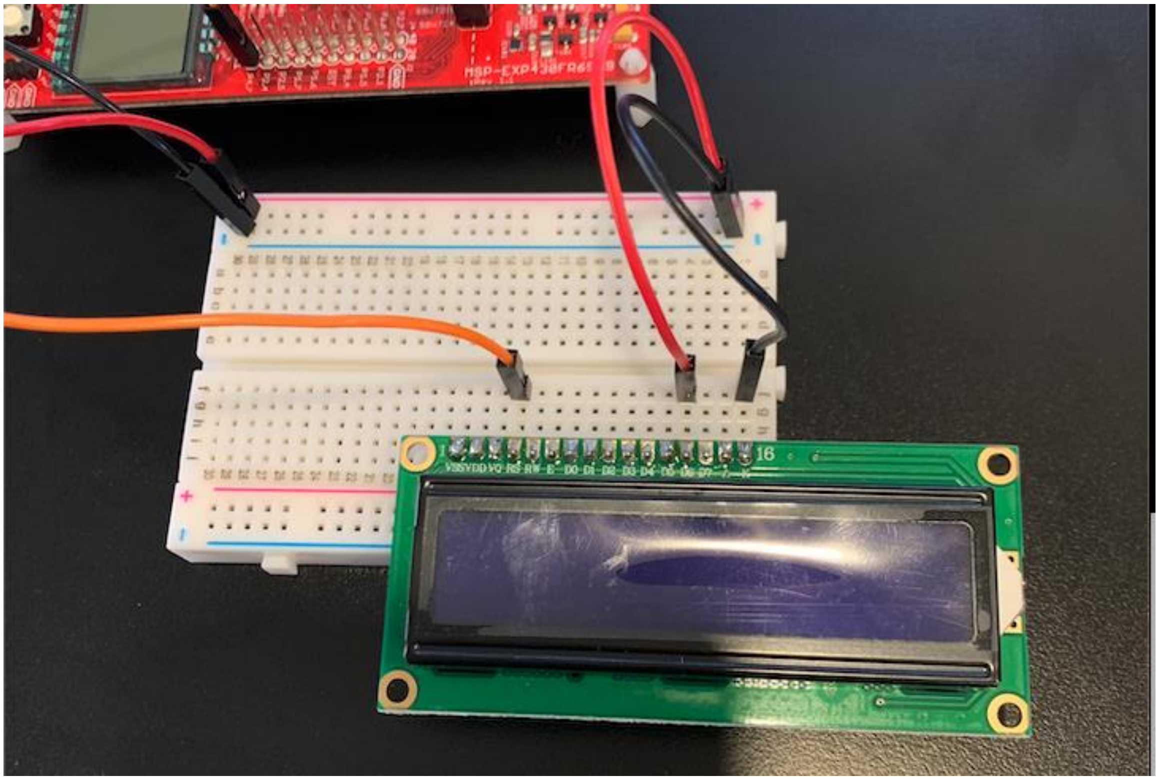


Figure 54. OLED Testing

5.3.4 Bluetooth Module Power Requirements and Connections

The Bluetooth module pays a critical role in communication. It is going to be used to connect the helmet to the user's phone. In order for it to operate, it's going to require a voltage range of 3.3-5V. This means that the Bluetooth module can be connected to the 3.3V output or the 5V output of the voltage regulator. For our design, the Bluetooth module is going to be connected to the 3.3V output. We also must make sure that the current is not too high or too low in order for the module to receive the appropriate power. If we connect it to the same output as the microcontroller, 3mA should be enough current to power it.

The figure below shows how the Bluetooth module would be connected to the microcontroller.

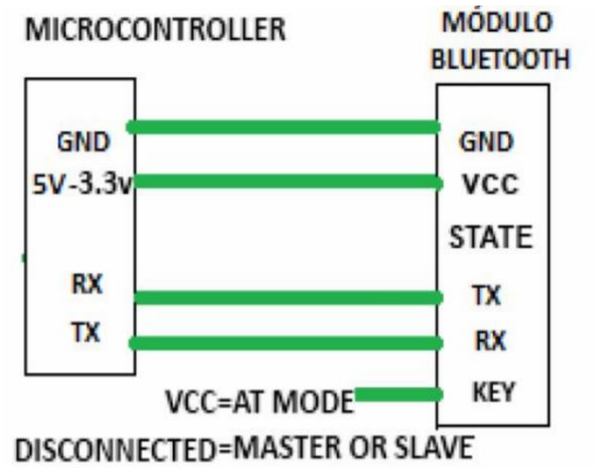


Figure 55: Application Circuit of Bluetooth Module

Wireless transmission testing will be performed for the Bluetooth module. The testing will ensure proper communication between the Smart Helmet and the mobile device that will have the Smart Helmet application installed. For the moment only the motion sensor will communicate if any G Forces have been detected on the helmet. When this happens, the data will be transmitted onto the mobile application notifying it of what has just happened.

In order to complete this test, we must first ensure that the motion sensor is fully functional and working correctly. Once we ensure that the sensor is functioning properly, we test both endpoints at different ranges. The biggest range difference should be no more than 3-4 ft between the microcontroller and mobile application. This being the case we want to test out at least 6 ft just to have enough wiggle room. Once both transmitters indicate through their LEDs that transmission is successful, we can stop testing.

5.4 Third Subsystem: Raspberry pi and Camera

The Raspberry Pi plays an important role in detecting drowsiness. Since the Raspberry pi is a computer, it requires a lot of more power compared to the other sensors, components, and the microcontroller.

5.4.1 Raspberry Pi

The second form of managing and controlling the Smart Helmet components will be with the aid of a Raspberry Pi. It differs from the MSP430FR6989 in that it is not a microcontroller but an actual tiny, dual-display, desktop computer. The power that it provides will be needed in order to be able to put the eye sensor or image processing feature of the Smart Helmet into play. Testing for the Raspberry Pi will be done indoors and with the use of a monitor and a camera attached to the Pi. The software needed to run the computer vision will be installed first. After that testing will be done to make sure the Raspberry Pi runs and operates correctly.

5.4.2 Infrared Camera

Testing of the infrared camera is pretty basic and straightforward. Testing will start out by making sure it powers on and off correctly. Once powered on we will connect it to a monitor and make sure the appropriate display is showing. The infrared camera main focus is to capture the eyes of the rider and detect once the eyes appear to be closing. We will test out the camera by taking video of different sets of eyes and making sure the eyes are recognized by the camera and give a good enough image. The final test is with us putting on the motorcycle helmet and taking pictures of each of our eyes to make sure the camera is able to take the best possible images of our eyes.

Figure 56. Camera Testing

5.4.3 Raspberry Pi and Camera Module Power Requirements

According it's specifications, the recommended input voltage is of 5V and the recommended put current is of 2A. The Raspberry pi is going to be connected to the 5V output of the voltage regulator.

If the Raspberry Pi is going to be powered via the GPIO, we only need to worry about two pins as shown in the figure above. We would connect the 5V source to pin #2 and we would have to connect the ground of that source topin #6.

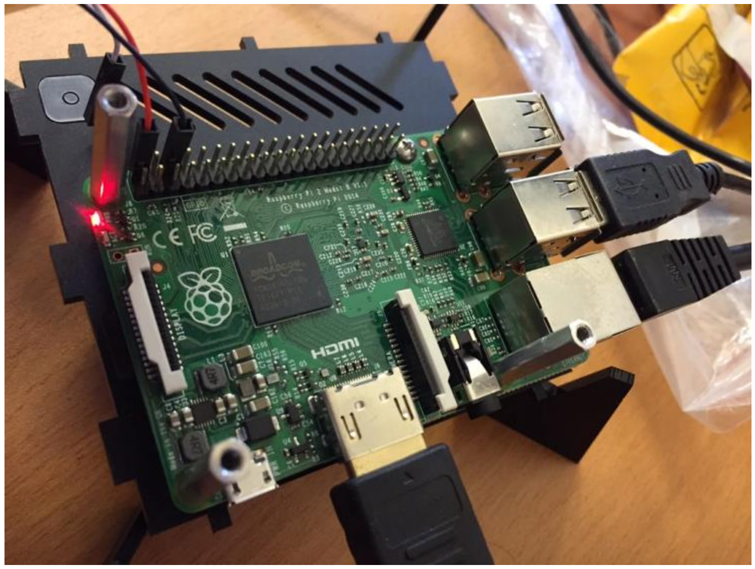


Figure 57: Raspberry Pi

The camera module is going to be connected to the Raspberry Pi and it is going to be used to detect drowsiness. The camera will receive the necessary power from the Raspberry Pi, the camera doesn't need to be connected to a voltage regulator, it just needs to be connected to the Raspberry Pi.

5.5 Related Schematics

This section will touch on all the component schematics that will be used for the design of the Smart Helmet. All of our schematics will be designed with the student version of Autodesk EAGLE and we will also be using NI Multisim to design our voltage regulator schematic. Both of this software are very user friendly and we have used them in previous semesters during our lab and for other projects. Most of the components that we will be using already include schematics that are available online. We use the library function that EAGLE has which allows us to easily transfer them over and be able to complete the design.

5.5.1 NI Multisim Schematic Design

The Multisim software integrates industry-standard SPICE simulation with an interactive schematic environment that enables us to visualize and analyze electronic circuit behavior. It will allow us to reinforce and improve the voltage regulator design needed for the Smart Helmet.

The figures below show the voltage regulator that was designed. There are going to be 3 outputs. The 5V, 2A current, that's where the Raspberry Pi is going to receive the power, the 5V, 15mA output, that's where the ultrasonic sensor is going to receive the power. The

3.3V, 3mA output that's where the microcontroller, LED display, and the indicators are going to receive the power.

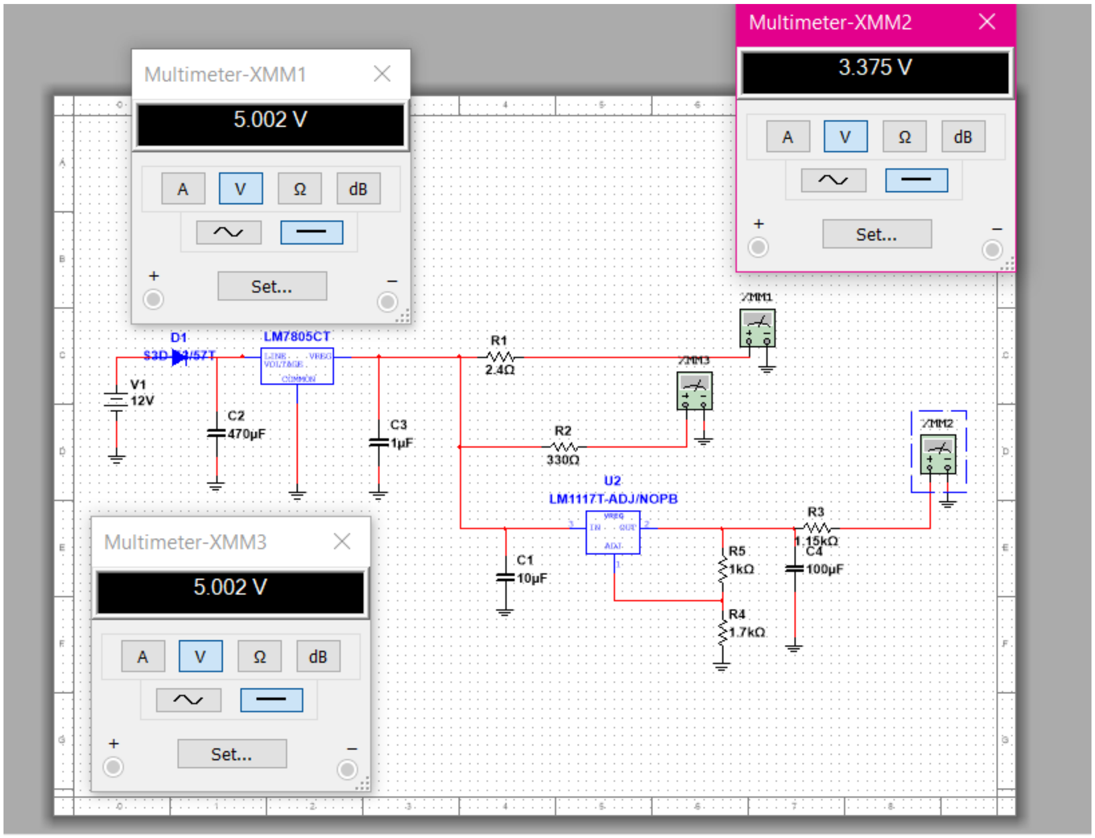


Figure 58: Voltage Regulator Design

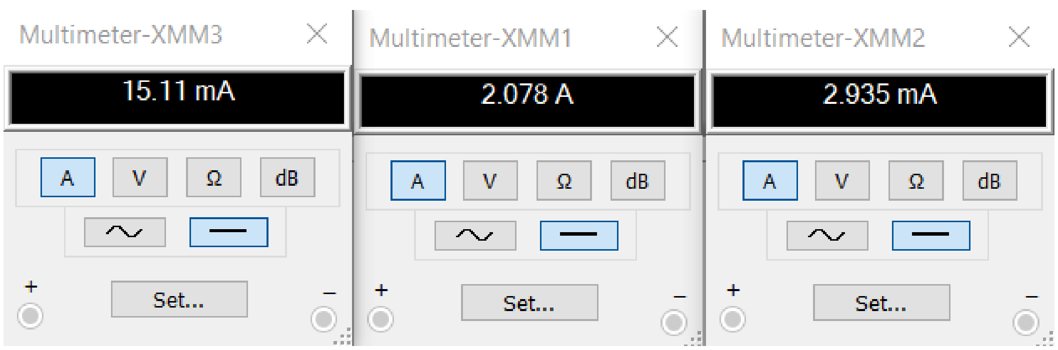


Figure 59: Voltage Regulator Output Currents

5.5.2 Autodesk EAGLE Design

Eagle is a scriptable electronic design automation application with schematic capture, printed circuit board layout, auto router and computer aided manufacturing. EAGLEs schematic editor will allow us to design and complete our circuit diagrams. Each part component will be downloaded from its respective device libraries with .LBR extensions. Once our schematics are completed and tested out with the use of our prototype physical components EAGLE allows us to complete the PCB routing to all components and complete our board. The Smart Helmet will have a dedicated module that will include the MSP430FR6989 microcontroller and finalized PCB design which will be mounted on the helmet. Our Raspberry Pi will have a separate module mounted on the motorcycle. In the following subsections we will be demonstrating each schematic design in the EAGLE software along with its pin selection and wiring.

5.5.2.1 MSP430FR6989

The MSP430FR6989 was chosen to control the Smart Helmet due to all of its advanced features that will enable us to take full advantage of all of the components that will be managed by its embedded microcontroller. The micro controller has a 16 bit up to 16 MHz clock and wide supply voltage range from 5 V down to 1.6 V that will enable use of its holistic ultra-low power system and FRAM technology that will combine speed, flexibility and endurance to our project.

The pin selection for the MSP430FR6989 was one of the more complicated parts of the schematic design and involved a lot of research from the Texas Instruments MSP30FR6989 documentation that is provided on their website. The datasheet, family user guide and errata guided pdf files aided us tremendously in figuring out the pin selections. Along with those pdf files Texas Instruments has a great support and training community online that we can lean on when in need of help.

To power the microcontroller we will be utilizing the 12 V power supply that is located on the motorcycle and will be inserting a voltage regulator between our Smart Helmet PCB module and power supply to decrease the voltage to properly power both the PCB module and Raspberry Pi which both will be set to operate at 5 V.

Connected to the microcontroller, we are going to have various sensors that will provide us with important measurements and data. As previously mentioned, those sensors include the ultrasonic rangefinder and the gyroscope. An OLED display as well as a speaker and LED indicators will also be connected to the microcontroller. The Raspberry Pi will be connected to one of the microcontroller’s pins in order to communicate information between them. Since the microcontroller doesn’t come with a built in DAC, we will include one as well an audio amplifier in order to connect the headset and be able to hear sound from the microcontroller. Sound will be played if drowsiness is detected in order to alert the motorcyclist.

Figure 60: MSP430FR6989 Schematic

5.5.2.2 HC-05 Bluetooth Module

The HC-05 will be adding two way (full duplex) wireless functionality to our project. The HC-05 Bluetooth schematic is very popular and easily accessible to get transferred over to the AUTODESK Eagle software. It will be very easy to configure because it operates using the Serial port Protocol (SPP). It will be powered with +5V and will be connecting the Rx pin of the module to the Tx of the microcontroller and Tx pin of module to the Rx of the microcontroller. Pin configuration for the HC-05 is demonstrated below along with the actual schematic built in Eagle.

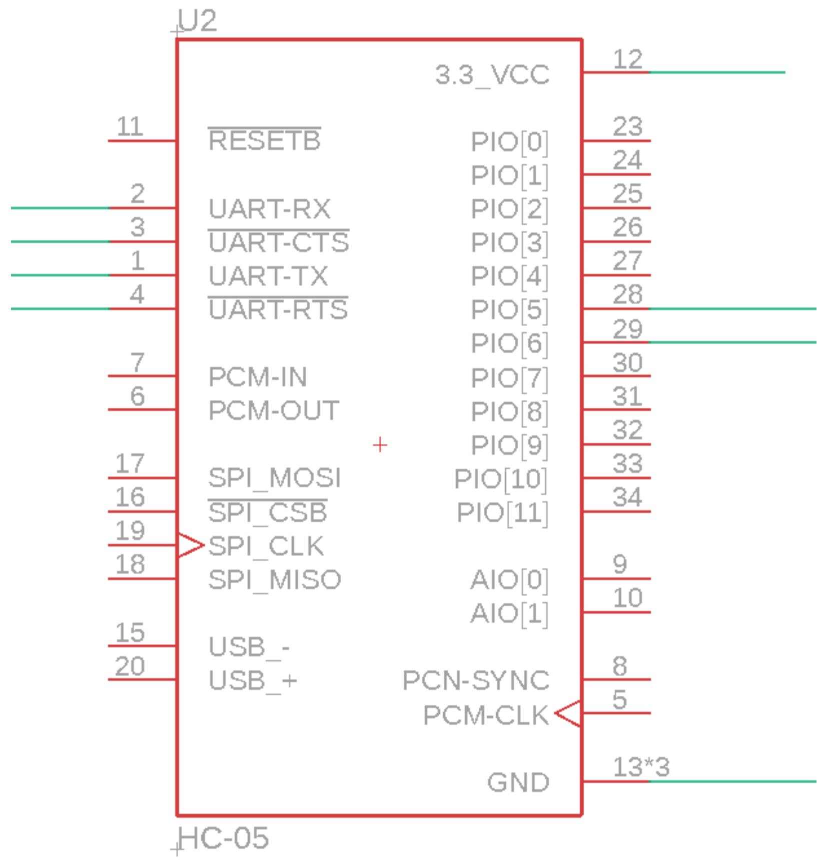


Figure 61: HC-05 Schematic

5.5.2.3 MPU 6050 3 Axis Gyro with Accelerometer Sensor

The MPU 6050 is another popular sensor that has easy access to its schematic and pin configurations. It has well documented and revised libraries available hence why it is very easy to use. The MPU6050 module allows us to read data from it through the IC bus, any change in motion will be reflected on the mechanical system which will in turn vary the voltage. It has a 16-bit ADC which will accurately read the changes in voltage and store it in the FIFO buffer and make the interrupt pin go to high. The data will then be ready to be read with the use of our MSP430FR6989 MCU through IIC communication.

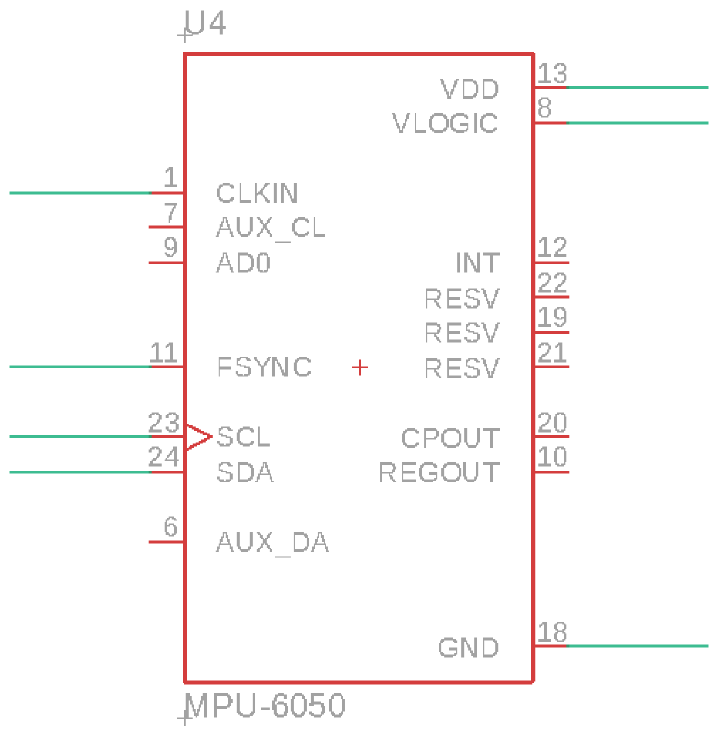


Figure 62: MPU 6050 3 Axis Gyro Schematic

5.5.2.4 Power Supply

The microcontroller and Raspberry Pi will be powered by the rider's motorcycle 12 V battery. For schematic purposes we designed an efficient DC-DC conversion circuit that will be in the range of 10-13 volts and an output should be 12 V.

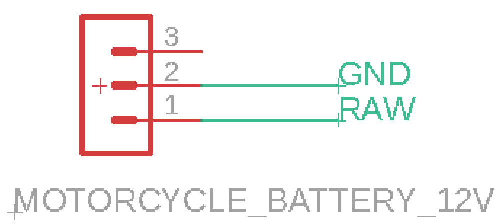


Figure 63: 12V Power Supply Schematic

5.5.2.5 HC-SR04 Ultrasonic Module

The HC-SR04 schematic was obtained from Ultra Librarian and then the library was downloaded and exported to Autodesk Eagle. The pin assignment for this specific module was easy to configure. Basically, the Vcc is going to be connected to a 3.3V source and the GND pin is going to be connected to ground. Data is going to be obtained from the TRIG and from the ECHO pins which will be connected to the MSP430FR6989 MCU. Pin configuration for this HC-SR04 ultrasonic module is demonstrated below along with the actual schematic built in Eagle.

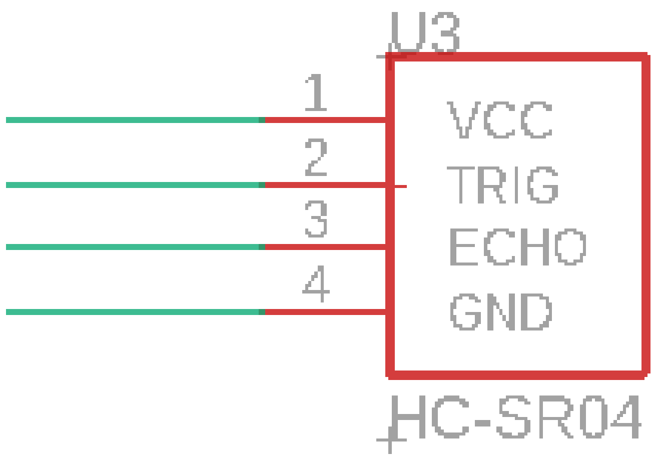


Figure 64: HC-SR04 Ultrasonic Module Schematic

5.5.2.7 LED Lights

LED lights are going to be required in our design in order to warn the motorcyclist about dangers in the blidspots. These are going to be connected to the MCU, the MCU will send a signal if the ultrasonic sensor detects that there are vehicles in close proximity.
Below is a schematic of the LEDs. LEDs are a type of diode that converts electrical energy into light. A resistor must be included in order to limit the incoming current to LED otherwise, it will get damaged.

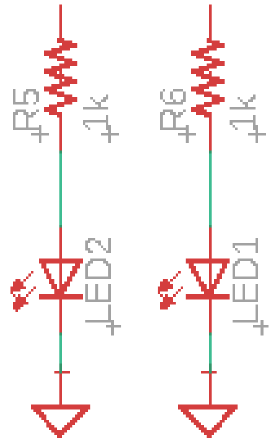


Figure 65: LED Schematic

5.5.2.8 Power Supply for Raspberry Pi

Demonstrated below is the power supply for the Raspberry Pi. The Raspberry Pi requires 5V and 2A in order to function. The 12V from the battery would be too much voltage for the Pi. Using TI's Power Designer feature, we were able to design a voltage regulator that will be able to provide it with sufficient power. Power Designer provided us with libraries for the different components of the design. These libraries were then uploaded to Eagle where the schematic was created. The 5V output of the power supply is going to be connected to the one of the 5V pins in the Raspberry Pi.

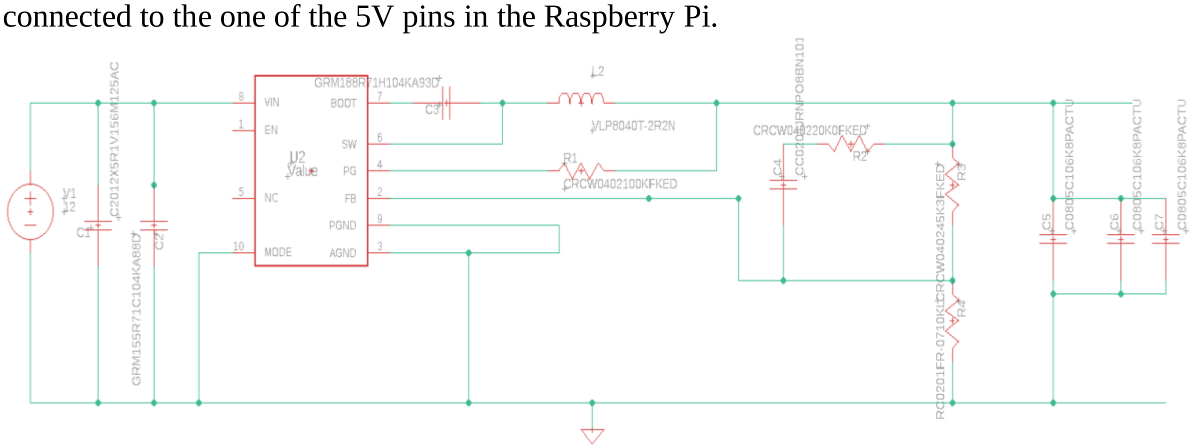


Figure 66:Power Supply for Raspberry Pi

5.5.2.9 Raspberry Pi

The schematic for the Raspberry Pi was already included in the EAGLE library. Below shows what this schematic looks like as well as the pin configuration . Basically, the 5V pin is going to be connected to the 5V power supply that was designed in the previous section. The microcontroller and the Raspberry Pi are going to be connected, we would be using one of the GPIO pins of the Raspberry Pi to connect the microcontroller to. Data and information would be communicated through that GPIO pin. For example, when the camera that is connected to the Raspberry Pi detects drowsiness, it will send a signal to the microcontroller, and when the microcontroller receives that signal, there will be sound coming from the headphones to wake up the motorcyclist. The camera module would just need to be connected to the Pi's CSI port.

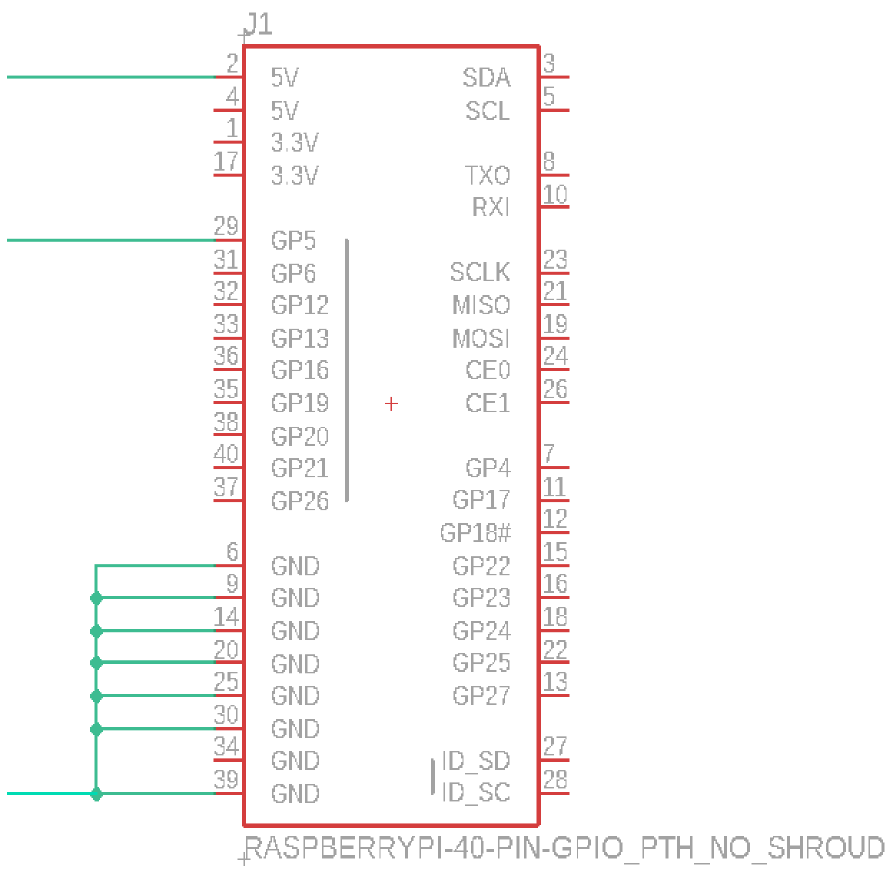


Figure 67: Raspberry Pi Schematic

5.5.2.10 OLED Display

The OLED Display is going to be connected to the microcontroller. Important information will be shown in the OLED display. The display is going to be mounted in front of the helmet where it won't block the motorcyclist field of view but where it would be easy for them to see. The ultrasonic sensors will be detected if there are vehicles in close proximity in the blind spot of the motorcyclists. If they are vehicles in close proximity, a warning will be shown in the OLED display. Other data such as time and Bluetooth connection could also be shown in the display. Information and data would be transmitted through the I2C interface. So basically, this module has 4 pins. It has the VCC which will be connected to 2.2V-5.4V source, GND pin will be connected to ground. The two other pins are the SDA and SCL pins. Information would be transmitted through these pins. The SDA is the data signal and SCL is the clock signal. The schematic for the OLED as well as the different pin configurations are shown below.

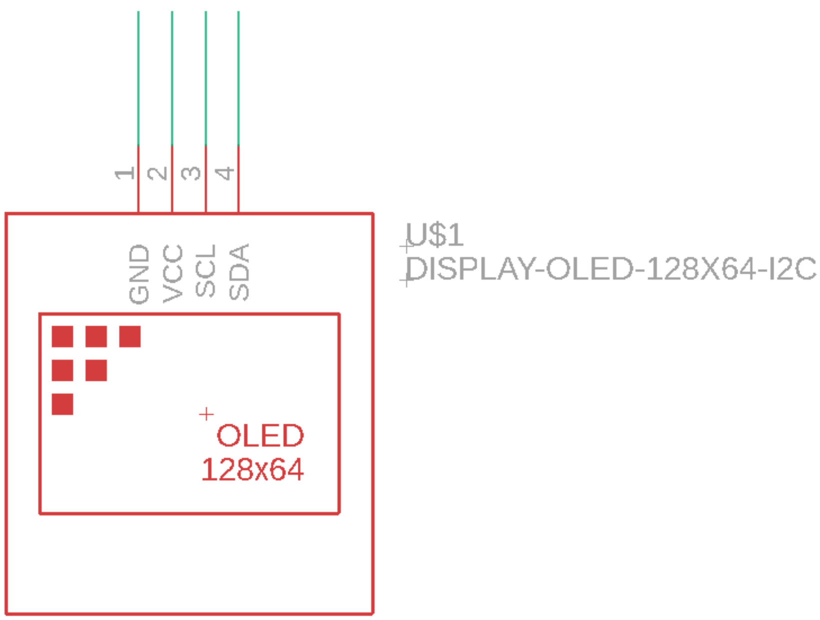


Figure 68: OLED Display Schematic

5.5.2.10 Headphone Schematic

For the drowsiness detection, we want the helmet to be able to alert the motorcyclist. For the helmet, we are going to be using a headset in order to alert the motorcyclist. If the camera and the Raspberry Pi detect that the rider is falling asleep, there is going to be sound coming out of the speaker to try to wake up the motorcyclist. The way that this is going to work is that a signal will be sent to the microcontroller from the Pi if drowsiness is detected.

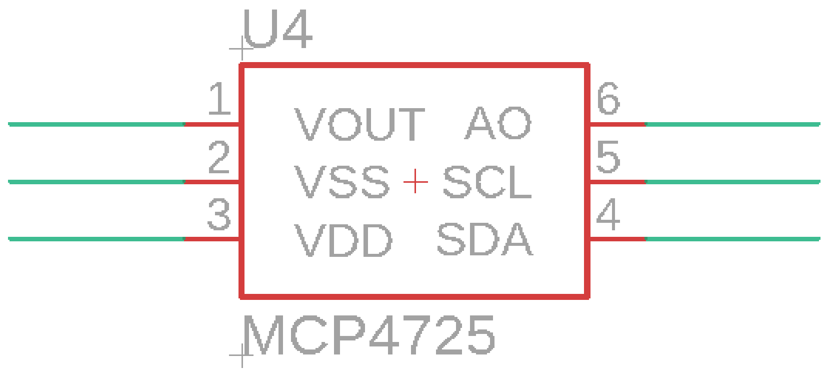


Figure 69: DAC converter

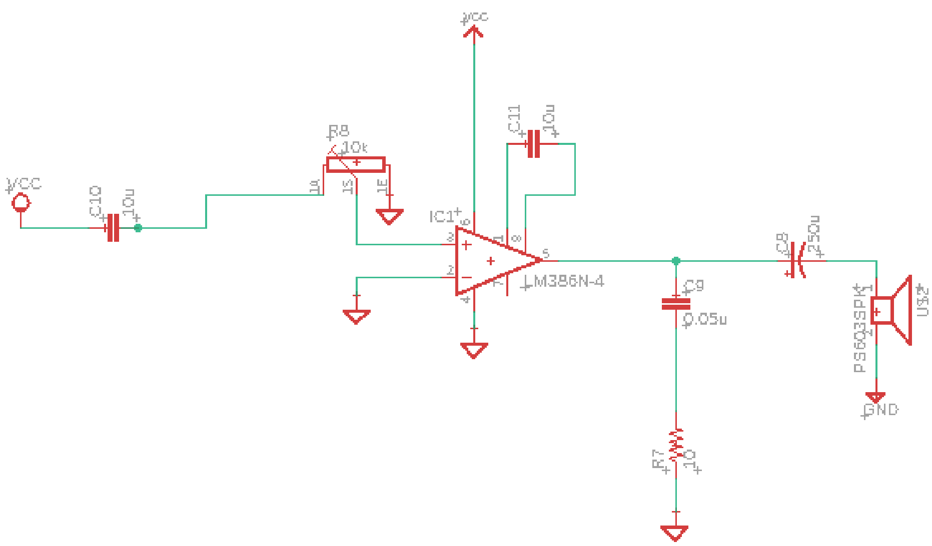


Figure 70: LM386 Audio Amplifier

There are many ways to implement the way that the headphones could be connected. The way that we decided to go about this, is that A DAC is going to be connected to one of the pins of the microcontroller in order to convert the digital signal into an Analog one. The microcontroller doesn't come with a DAC built in and therefore a DAC module will be necessary. The LM386 Audio Amplifier is going to be connected to the DAC and this will help to amplify the sound so that the motorcyclist can hear it loud and clear. Finally, the headphones are going to be connected to the LM386 Audio Amplifier. Demonstrated below is the schematic for both the amplier and the DAC converter.

5.5.2.11 Final Schematic

The figure below demonstrates all of our schematics assembled together in the AUTOCAD Eagle software.

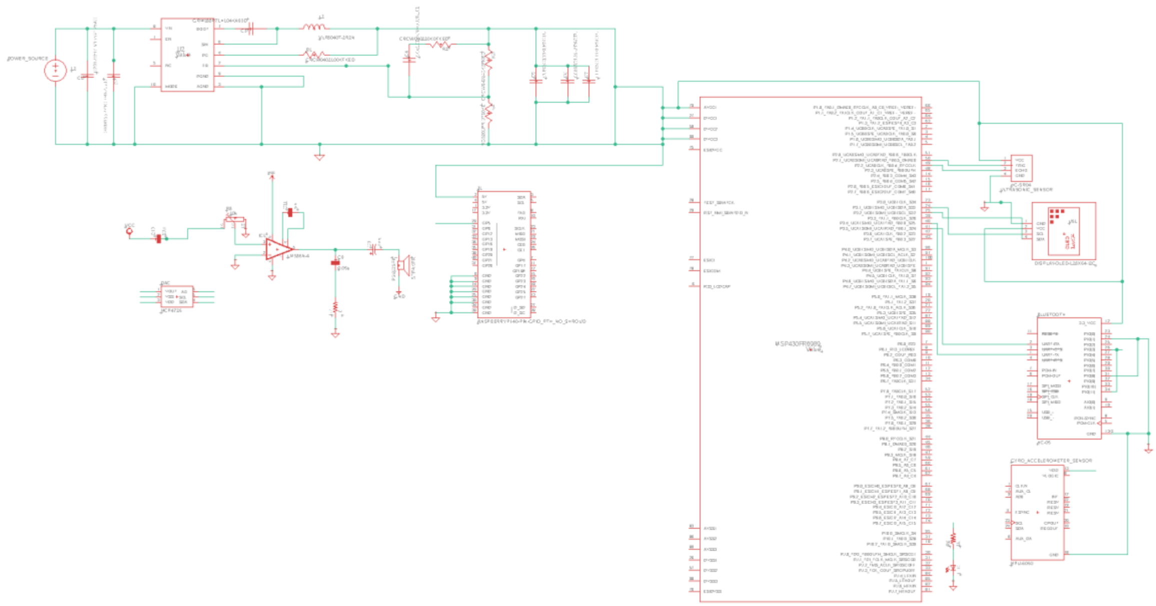


Figure 71: Final Schematic

5.6 Software Design

There are two main parts of software for this project and a third part which is additional functionality. The main software for this project is the code that is running inside the microcontroller. The second main software is the Raspberry Pi which is responsible for the sleep detection using a thermal camera.

5.6.1 Image Processing Software Design

This section takes a detailed look at the software design for the system's eye detection feature. Much of the image processing will be handled by the Raspberry Pi, which will interface with a camera to stream images of the user's eye region. The main functionality of this feature is to alert and prevent users from falling asleep while operating their vehicle. Below is a high-level flowchart of how each individual component works together to accomplish this task.

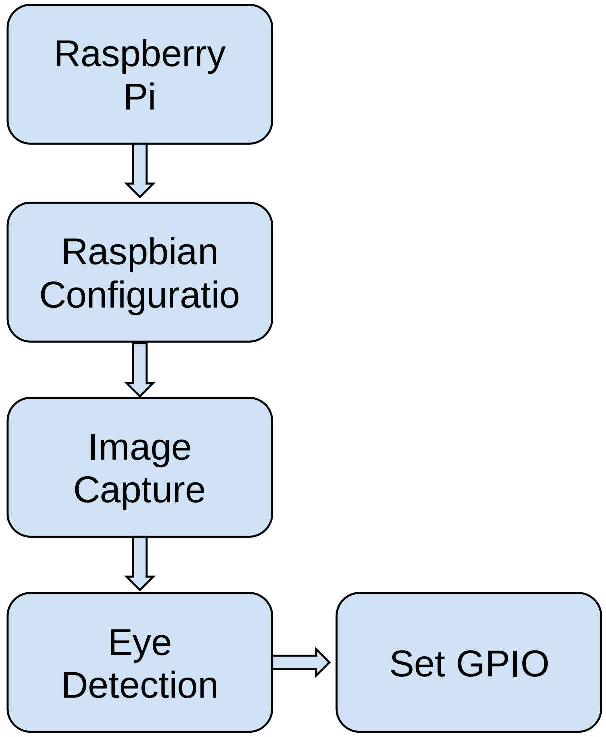


Figure 72. High-level Software Flow Diagram

The image processing software will incorporate two libraries: Opencv, a powerful open-source computer vision library and dlib, which is a machine learning library containing powerful object and facial detectors.

The drowsiness detection software will be implemented in Python, which is a powerful scripting programming language that allows for rapid prototyping and development. Python's weakly typed nature and garbage collection allows the programmer to focus more on algorithm development rather than syntax and memory management. Python is also well-documented and supported by many libraries and frameworks such as the ones mentioned above.

Since the camera we will be using has direct compatibility with the Raspberry Pi, all that needs to be done is to connect the camera to the CSI (Camera Serial Interface) port on the board. We will be using the OpenCV library to take image output from the camera and feed those into our feature detection algorithm. Given an image, the feature detector returns an array of points outlining major facial landmarks. By indexing a subset of the array, we are able to collect all the points associated with the eye region. These points will feed into an aspect ratio calculation which will be used to determine if the user is getting sleepy.

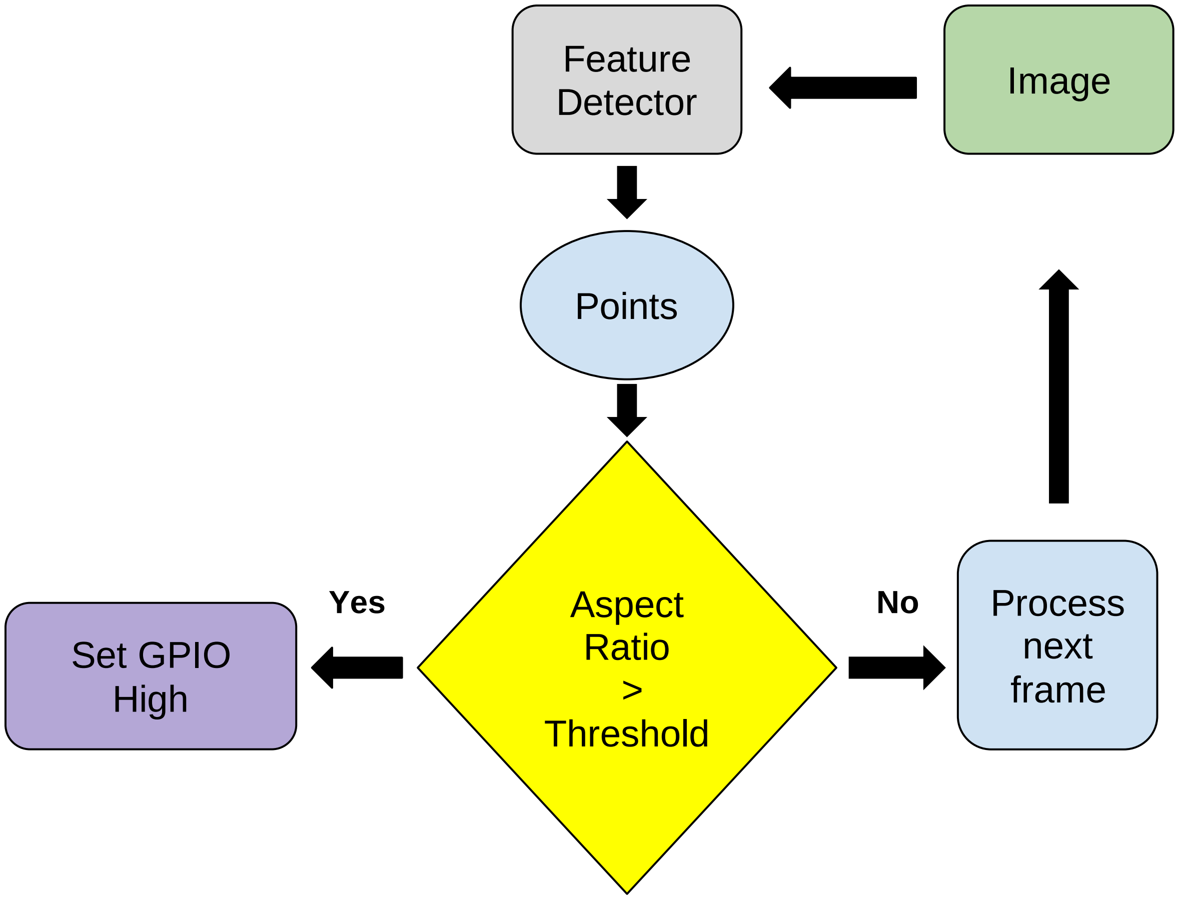


Figure 73. Drowsiness Detection Algorithm Flow Figure

Once the algorithm outputs a value below our established aspect ratio threshold, the program will set one of the GPIO pins high to cause an interrupt in the microcontroller. This interrupt will then allow the microcontroller to run a process to wake the user.

5.6.2 Android Application

The main purpose of the mobile application is to connect to the microcontroller and send an emergency message in dangerous situations. In this section we describe the main methodology we followed to develop the Android application. We will also lay out a description of the way the microcontroller is connected to the Android application. A description of the Android app itself and its general design and the tools used to develop and test its software.

5.6.2.4 Android Development Tools

As discussed in the standard section Android is built on top of a Linux kernel which is running Java Virtual Machine (JVM). Therefore, the official language that Android is supporting is Java. There are other alternatives such as React Native which is an open source mobile application framework that is built on top of JavaScript using Node.js. Since we are already familiar with Java as an officially supported language, we decided to develop our app using Java.

Integrated Development Environment (IDE)

Some of our team members already have experience developing for Android using Android Studio IDE, therefore, we decided to go with Android Studio. Android Studio officially supports Java and Kotlin with a very good integration of Java into the IDE. It also automatically applies Android and Java standards while coding which can speed up development time. Android Studio contains several tools for development which can be helpful with the design and coding. One of these tools is the simulation where an app can be fully simulated inside the computer for quick testing. It also supports debugging logs which is essential when testing code and fixing bugs. Another reason we chose Android Studio is because it is backed up by Google with comprehensive documentation and a large community which makes it easier to learn the development process.

5.6.2.5 Mobile Application design Methodology

In this stage of our project development we are still deciding on features that might be added to the project which will require us to change or add to the mobile app. Due to that, we decided to follow the Agile methodology. Agile is a flexible software development which allows the project to quickly react to changes in requirements and constraints. It does that by applying an iterative collaboration between self-organizing cross-functional teams. Agile methods or Agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals.

5.6.2.6 Connection Between Mobile Application and Microcontroller

One of the most important features of this project is the ability to send a message in case of emergency. The microcontroller can communicate with several sensors and communication modules. The microcontroller will constantly monitor the state of the driver through the sensors, when an emergency situation is detected an interrupt routine will send a signal to the mobile application and send a preset emergency message to the contact selected ahead of time by the user. Other functionality between microcontroller and mobile device is the ability to use navigation software on mobile devices. The instructions audio will be transmitted to the microcontroller and the microcontroller will play the audio to the speakers. Additionally, the mobile device will transmit visual signals such as right turn or left turn arrows that will be displayed on driver's built in LCD inside the helmet.

The transmission will be composed of simple signals such as in emergency situations. Visual data, such as visual navigation arrows, and audio from common navigation apps like Google Maps.

5.6.2.7 Mobile Application Initial Design

The mobile app will have the main screen where the user can access settings there, they can set emergency contact a custom message which can include GPS location. In the figure below (Figure 5.6.2.7.1) we can see the main design of the app. The Dashboard button will allow the user to access information from the microcontroller about sensory data such as a car in a dead spot. The Maps button will open a favorite navigation app that can interface with our microcontroller. The settings button will allow the user to set a custom message, fill up personal information and preferences, and emergency information. Users will be able to set up Bluetooth through the settings also.

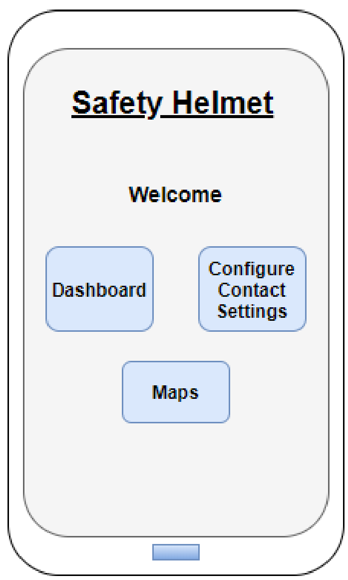


Figure 74. Initial mobile application prototype

5.6.3 Microcontroller

The microcontroller is the main control for the safety helmet. It will connect to all the sensors and constantly monitor them, once an action needs to be performed it will react quickly and perform the required task. It will react via interrupts and signal the driver via LCD and built in speaker and LEDs.

5.6.3.1 Microcontroller Safety System

The microcontroller will be directly connected to several safety sensors which are able to raise an interrupt in certain conditions:

Detection of a Car in the Dead Spot

Two distance sensors are connected to the microcontroller and placed on the outside of the helmet each covering about 45 degrees of the rear left and rear right of the driver. In case a vehicle is detected an interrupt will be raised and the microcontroller will initiate a service routine that will signal to the driver through the appropriate indicator and on the LCD. The signal will last for a few seconds to allow the driver to recognize the signal.

Accident Detection

Tilt or Gyro sensor will be placed inside the helmet and if the sensor is detecting a tilt of more than 70 degrees relative to the vertical axis it will raise an interrupt which initiates a timer. If the driver stays tilted for longer than 10 seconds a service routine will send a signal through the Bluetooth module to the mobile device. The signal will be interpreted through the mobile app and will send an emergency message to a preset contact saved in app memory. We decided to raise an interrupt only when tilted more than 70 degrees in order to be sure that the driver is having an accident. The 10 seconds timer is there to prevent false alarm in case the driver is tilted for a short time which indicates that an accident did not occur.

System Activation Detection

Since several features like accident detection depend on the orientation of the helmet and additionally in order to save energy a mechanism that detects if the helmet is in use is required. A small button inside the helmet will be pressed once the user wears the helmet which will automatically set the microcontroller to ON state. Once the microcontroller is in ON state all safety features are activated and ready. During OFF state the microcontroller will go into low power mode in order to save energy.

Sleep Detection

Sleep detection is done using computer vision techniques (more on that in Raspberry Pi sections). The microcontroller will be connected to the Raspberry Pi, when the Raspberry Pi will detect sleep state it will signal the microcontroller which will raise an interrupt and a service routine will start execution. The service routine will send an audio signal to the speakers in order to signal the driver from the danger of falling asleep.

Display and Indicators

A small display will be placed inside the helmet in driver eyesight but on the side, so the driver is not distracted during driving. The display will present information about navigation and indicators such as a car in the dead spot and current time. In addition to the display several LED indicators on the sides will be placed in driver eyesight but on the bottom left and right to not distract the driver. These two indicators will indicate a car in the dead spot.

5.6.3.2 Microcontroller Development Tools

We chose the MSP430FR6989 microcontroller from Ti and our development is done in Code Composer Studio provided on their website. One of the reasons we chose this microcontroller is because we are familiar with it from embedded system class and we have some experience working with CCS in the lab. Since we can't use the lab on campus due to the epidemic (more on that in section 4.2) we have to use a simple affordable multimeter in order to debug code and check ports' logic. We also use Konsole terminal to send data from the microcontroller to the computer through UART which helps when debugging.

5.7 Design Summary

The main features of our project are the safety features which include all discussed above. The helmet will have a built in LCD which will present some information about navigation and sensory data as additional features. The android app is an important part of this project and it will be responsible for sending an emergency message which can save life. The Raspberry Pi is responsible for detecting if the driver is too tired and should send a signal to alert the driver. The speaker is another important part of that, and it will be used to alert the driver first and maybe add the option to listen to music if time permits.

6. Project Prototype Construction and Coding

Designing our software is a very demanding task and requires us to learn datasheets and perform rigorous testing in order to ensure a proper behavior of all the components. There are three main programs that are required in order to implement every part of the project and are composed of: Microcontroller program, Raspberry Pi and Android application.

6.1 Final Coding Plan

First, we are going to discuss the software structure and design of the microcontroller then we will cover the software for the Raspberry Pi and finally the android application. The microcontroller is expected to communicate with both Raspberry Pi and Android applications.

6.1.1 Microcontroller Software

The code for the microcontroller will start by initializing all the components connected starting from all the sensors, then Bluetooth module and I/O pins and finally LCD display and indicators. Each sensor will be set up with an interrupt and its interrupt service routine including the pins from the Raspberry Pi. Each interrupt will perform its part by executing the code in its ISR and maybe calling another function to perform further operations such as Bluetooth communication if needed. The pin from the Raspberry Pi will be set up with an interrupt similar to the sensors. When the pin is set too high it means that the driver is detected as falling asleep and a sound will be played through the speaker. Once all interrupts are handled and the interrupt queue empty the microcontroller will enable low power mode.

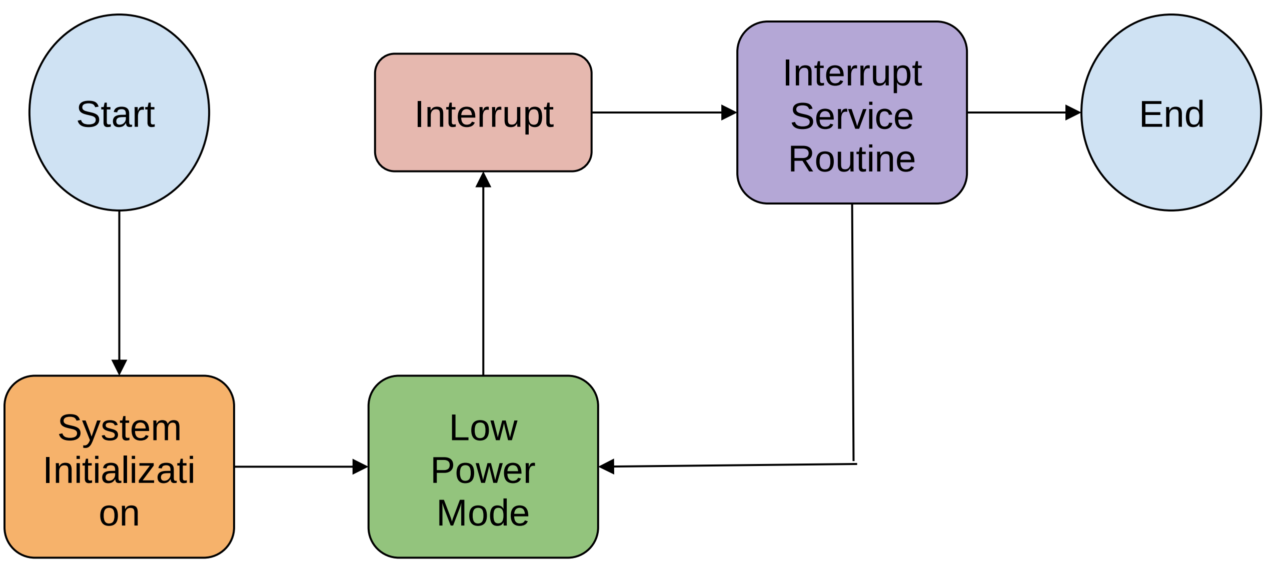


Figure 75.General flow of microcontroller software

6.1.2 Image Processing Software

The image processing component of the system is designed to be modular, meaning that it can be treated as a black box that takes in inputs and outputs information that other parts of the system can use. This separation allows development to be independent of other components of the system. Below are step-by-step instructions on how to setup and configure the Raspberry Pi to start developing the image processing application.

System Setup:

1. Purchase Raspberry Pi
   Link listing current suppliers: https://www.raspberrypi.org/products/
2. Download Linux distribution for Raspberry Pi
   - For this project we used the recommended *Raspbian* Distribution
   - Download link: https://www.raspberrypi.org/downloads/raspbian/
3. Flash installed Linux distribution onto a memory card to insert into Raspberry Pi
   Once the Raspbian operating system is downloaded we need to write its image to the Pi's flash storage. To do this we can use a tool called *balenaEtcher*.
   Link to download: https://www.balena.io/etcher/
   After copying the image to the sd card, power up the Raspberry Pi and insert the card. You should see a blinking light to indicate that the Pi is booting.
4. Install external libraries and dependencies
   In order to install the dependencies needed to run the software it is advised that the Raspberry Pi has an internet connection to take advantage of the *pip installer*.
   Libraries:
   - *OpenCV*
     To install OpenCV you can run the following in the command line:
     python -m pip install opencv-python
     You can check whether the installation was successful by running python in the terminal and running: **import cv2**
     If no errors come up, then the library was ported successfully.
   - *Dlib*
     We will be using Dlib to handle the feature extraction of the face and eyes.
     For step to download Dlib and configuring the camera:
     Link: http://dlib.net/compile.html

7. Project Prototype Testing Plan

Once the Hardware and software design details along with the completion of construction and coding of our project are all done it will be time to go to our last step of the prototype Smart Helmet project. Our testing plan for the prototype will be separated into two sections. Hardware and software testing will take place in both an environmental setting along with the specific scenarios that we will need our project to function like it is supposed to.

7.1 Hardware Test Environment

In order for us to get a complete picture of how our hardware will respond we must test it out in both a “vacuum” environment and in the actual environment that it will be put to use. The “vacuum” environment will take place on the University of Central Florida campus. UCF has several open labs that are accessible to us to complete our testing like the senior design lab and TI lab to name a few.

The Smart helmet is made so that the rider can use it in many weather conditions. We live in the state of Florida and it offers us many weather conditions to test our helmet components. Once we have completed all of our indoor “vacuum” testing we will take our testing outside and depending on the weather conditions we will test our hardware out on as many possible scenarios.

As of the moment that this report is being made, we are in a special circumstance around the world with the COVID-19 pandemic affecting us all. We are very limited on what as this moment we can test out so we will try to provide as much possible testing data.

7.1.1 Vehicle Testing

Ideally, we would have liked to have a motorcycle to test out the Smart Helmet. Unfortunately, this was not possible due to none of us owning any motorcycles and due to the COVID-19 pandemic we could borrow one from anybody. Since this was not possible, we will be using the closest thing possible to replace a motorcycle, a bicycle. Using a bicycle will allow us to simulate as close as possible to a motorcycle. What we can't capture or test out with a motorcycle, another option that we can use is a vehicle. All of us have access to one and will be the last resort if we have to come to that.

7.2 Software Test Environment

The Smart Helmet will utilize three software programs that will allow us to build our project. Each software is unique, and each will be tested out with their respective components. Testing will be done indoors at the senior design lab in order to simulate a perfect environment. Once testing indoors is complete we will perform testing outside in as many possible environmentally different scenarios.

The Texas Instruments MSP43FR6989 launchpad utilizes its own software development tool; Code Composer Studio Integrated Development Environment for MSP Microcontrollers. This tool includes its own compiler, source code editor, project build environment, debugger, and profiler. The Raspberry Pi runs off the Raspbian operating system which allows it to simulate a desktop computer. It will allow us to run Python and implement the drowsiness detection software. Lastly testing of our mobile application will be performed using Android Studio, it contains several tools for development including a simulation where the application can be fully tested inside of a computer for quick and accurate testing.

The Smart Helmet Project software testing environment is meant to succeed in both normal and harsh environments. Once both of these testing requirements are performed and completed, we will go onto the final software specific testing.

7.3 Software Specific Testing

Software specific testing will be the last step in the Smart Helmet design. The software programming is what makes the Smart Helmet unique and allows it to stand out from the rest of the helmets currently in production. The MSP430FR6989, Raspberry Pi and Android application will each be tested alongside the specific components that they will be controlling and analyzing.

7.3.1 MSP430FR6989

The MSP430FR6989 software is going to be primarily designed to control the proximity sensor, motion sensor, oled display, LED lights and audio. This information will be transmitted wirelessly through the Bluetooth module and received onto the mobile application which will be installed on a mobile phone. The software will be tested out individually for each component and once that is successful everything will be put together and tested out. The following subsection is a description of each component and how the software was tested on each one.

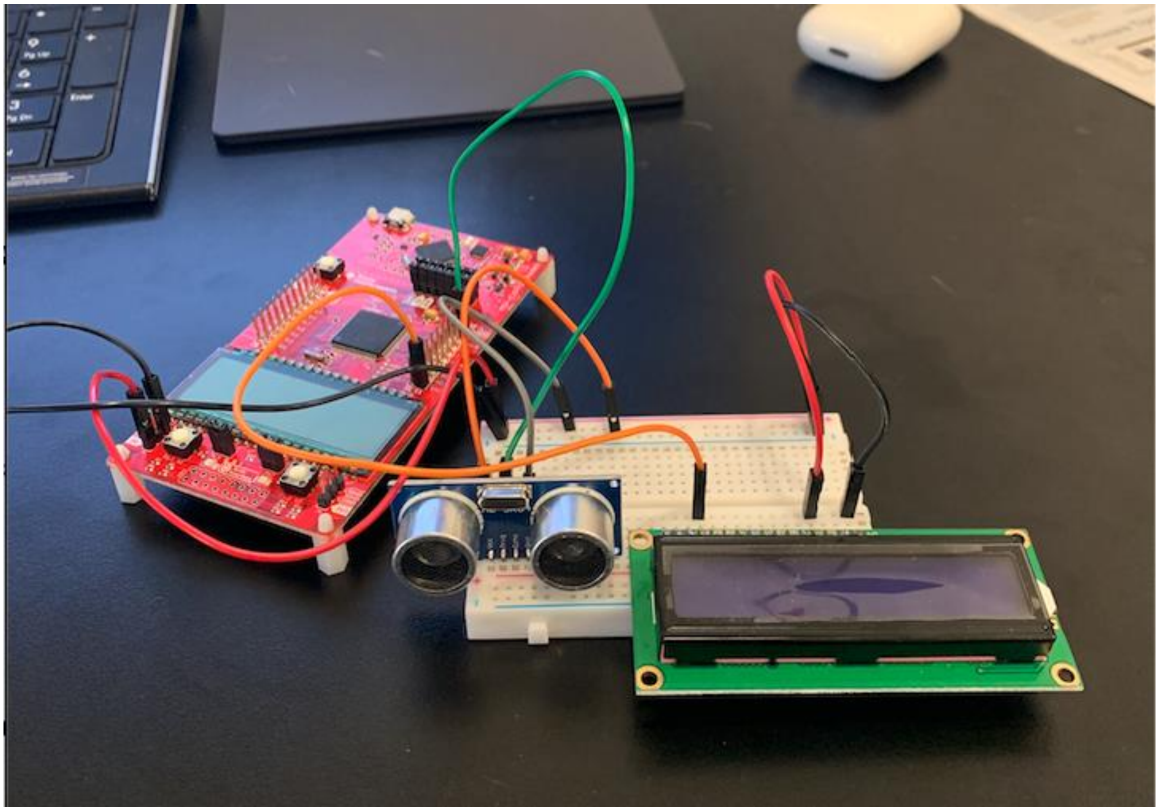


Figure 76. MSP430FR6989 Testing

7.3.1.1 Proximity Sensor

Indoor and outdoor testing of the proximity sensor will mimic each as much as possible to maintain consistency in our measurements. The only difference is in the outdoor testing we will have the sensors aligned the same way as how we will have them on the bike helmet. If possible, we would like to have them mounted on the motorcycle helmet itself. Due to the current COVID 19 situation that is not possible at the moment though. Two different environmental testing's will be performed, GOOD and EXTREME ratings so that we can see how big of an impact each environment will have on the measurements.

Table 13: Indoor Testing

Distance (ft)	
5	✓
10	✓

15	✓
20	✓
25	✓

Table 14: Outdoor Testing - Good Weather

Distance (ft)	
5	✓
10	✓
15	✓
20	✓
25	✓

Table 15: Outdoor Testing - Extreme Weather

Distance (ft)	
5	✓
10	✓
15	✓
20	✓
25	✓

7.3.1.2 Motion Sensor

Axis rotation will be performed with the motion sensor installed on the helmet. The purpose of the test is to have the microcontroller be notified once a certain angle has been reached or crossed. Several different angles will be tested, and the optimum angle threshold will be used once that is determined.

The G force testing will be performed the same as the axis rotation. Testing will determine once a certain G force has been attained on the helmet. The Smart Helmet along with the sensor will be dropped from certain distances and measurements of G forces will be performed until accurate enough G force data is extracted.

The same testing will be performed outside with the exact same parameter to ensure dependability and consistency. Testing will be successful once the data from the sensor is accurately detected and transferred over to the microcontroller. Once this is done, we can go on to testing the wireless Bluetooth transmission.

7.3.1.3 OLED Display

Software testing involving the OLED display will involve proper visibility of what is being displayed on the screen. We must make sure we use easy to read fonts along with a character size that the rider won't have to struggle to look at while they are focusing on driving their motorcycle. Testing on how long the font will stay displayed on the screen will also take place along with the brightness of the screen.

The Smart Helmet is going to be designed to be able to be used day or night, so it is important to test out the displays in both daytime and nighttime scenarios. Testing of the software will take place indoors first to make sure it is transferring data correctly from the microcontroller. Once testing indoors is complete we will take testing outside to be able to more accurately test out the visibility factor.

7.3.1.4 LED Lights

Testing of the LED lights will take place with a breadboard and the microcontroller attached to them via jumper cables. The LED lights are meant to be a warning for the riders when a vehicle is approaching. The proximity sensors will alert the microcontroller if any object is getting too close. Once that is recognized by the microcontroller it will be programmed to send a notification to the Led lights to turn on and off. Testing will compromise making sure this happens and we will test out several ways of what will be best, so the driver does not get affected by the lights in a negative way. Once testing is complete, we can go on to testing the final component.

7.3.1.5 Audio

Our final component that needs to be tested with the microcontroller is the audio connection that will notify the user of the Smart Helmet. The Smart Helmet will be equipped with headphones that the rider can put into their ears when putting the helmet on. This will verbally notify them of any hazards or objects getting close to them. In order to connect the headset to the microcontroller we will attach a 3.5 audio jack to the microcontroller along with an audio amplifier circuit. The headphones will work alongside the LED lights and will get a response via the microcontroller once it is notified via the proximity sensors. The code will be tested out to make sure it is all working and will be connected using a breadboard.

Once we can verify through testing that this will work, we will try and have both the Raspberry Pi and microcontroller communicate with each other and share the audio. The Raspberry Pi main purpose is to monitor the eye movement of the driver that has the Smart Helmet on. Once they notice that they are possible going to sleep we not only want a visual notification but want an audio notification as well. We believe having both will elicit a better response from the driver that has the Smart Helmet on. This is just an extra feature that we would like once we can get the audio to perform correctly with the microcontroller and the proximity sensor feature.

7.3.2 Raspberry Pi

In order to test the Raspberry Pi, we must first download an operating system on its flash memory card. We then will insert the card into the Pi and wait for it to configure itself with the newly installed operating system. Once we see that the Pi has booted properly, we will then test each of its ports to make sure they are operational. Since we will be using the GPIO off the Raspberry Pi to trigger interrupts in the microcontroller when a user displays drowsiness, we will set each of the pins to a logic high (3.3V) and use a multimeter to validate that each pin is outputting the proper voltage. Once we install the appropriate libraries and modules on the Raspberry Pi, we can run a unit test on each component of the program to make sure that there are no errors in the code. We must also test whether the interface between the infrared camera and the Raspberry Pi to make sure that they are able to communicate.

7.3.3 Eye detection Application

This section describes the testing procedure used for the eye detection software. In order to test the application properly it would be helpful to think of the associated inputs and outputs of the program. The image processing software takes in images from a camera and processes each frame, performing an aspect ratio calculation on the eye region. In order to test this functionality, we ran the program and outputted the aspect ratio calculations to the console.

Figure 77. Localization of eyes

```
WAKE UP
9
WAKE UP
6
8
WAKE UP
8
WAKE UP
7
WAKE UP
6
7
WAKE UP
8
```

Figure 78. Aspect Ratio output with alert

Once the points we were using for the calculation were found to be stable and accurate, we then began to move forward to trigger an external device when the user was found to be falling asleep. In order to do this, we configured a msp430 microcontroller to communicate with the machine that is hosting the image processing software using UART. We then added a script that sends predetermined commands to the serial port of the host machine that will trigger the microcontroller to turn an LED on or off, depending on whether or not the user is found to be falling asleep. Although this is not the only way to test the image processing software, we felt that it was a pretty quick and easy way to test all it’s expected functionality.

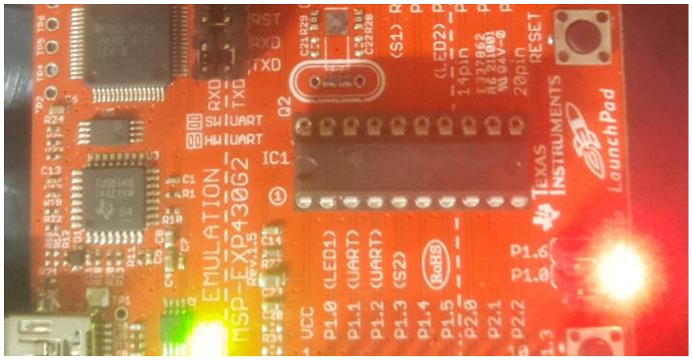


Figure 79. Microcontroller led output high when eyes closed

7.3.4 Android Application

The reason for choosing the Android application is because it is one of the most popular operating mobile systems being used today in a variety of smart phones, tablets and computers. This application is more comfortable and advanced for the average user. It is an open source operating system which means that it is free, and anyone has access to use it. Communication with the Android application will involve either touching or inputting text onto the device that the driver chooses to use. Some of the many features available for it are head set layout, storage, messaging, multilanguage support, multi touch, video calling, screen capture, external storage, streaming media support and optimized graphics.

Software testing is an integral part in the development and release of applications. Without proper testing an application may fail on its users, which may result in catastrophic consequences. Our systems ability to be fault tolerant relies on strict testing to make sure our users are safe. This section discusses the main stages of software testing as well as some of the tools that are available to perform these tasks quickly and efficiently.

In order to validate the functionality of the mobile application, we will perform a series of software tests to check whether the actual results are the ones expected. We will first begin to perform a unit test of the application, where we break the application's functionality into individual components to validate that they are working properly. Next we will perform a system integration test, where we group components of similar functionality and test them together. The system integration testing will allow us to see how well each individual component interfaces with each other. Once the system integration tests are completed, we can now test the entire application as a whole. We will have volunteers use the application to see if they can find any bugs that we may have missed during earlier testing phases. We will also need to check the applications ability to interface with the sensors that make up the system.

There are several testing suites that we can use to run the software tests such as Kobiton, which was created to automate software testing on mobile applications. These testing tools will allow us to debug our application a lot quicker than testing manually. Since the application will be mainly handling user localization and contact alerts, those features will be the priority of our testing.

8. Administrative Content

The sections below outline the general planning and finances for the project. We break the project into three phases which will allow us to keep track of what needs to be completed in a specific timeframe. Within each phase are several milestones and tasks that are delegated to each member of the group. Project expenses are broken down into categories with a comparison of what was actually spent vs. the projected amount. The goals of this project is to present a system that keeps motorcyclists safe as well as being cost-effective.

8.1 Milestones

This section describes the several goals that need to be accomplished in order to successfully deploy the system. Each team member will be assigned tasks based on interest and experience level. The goal is to develop each portion of the system in parallel in order to begin system integration much earlier to provide sufficient time for testing.

Due to the COVID-19 outbreak, much of development will happen individually with weekly status meetings to keep on schedule. Although we are planning to integrate all the proposed features into the actual helmet, we may show the capabilities of each component separately as a proof of concept. Many of the deadlines and milestones we set earlier in the semester may be changed due to the inability to receive parts and use required lab bench equipment.

We have divided the project into *three* main phases:

Research & Planning

This section includes the planning steps needed to be taken in order to prepare for system design. The capabilities of the system must be evaluated to generate the proper requirements. Once the requirements are generated, different technologies and constraints must be researched in order to perform a justified component down selection.

1. System requirement generation
2. Component evaluation and down selection
3. Component procurement

System Design

This section discusses the second phase of the development cycle which includes the system design. This phase will share some overlap with the research and planning phase. The requirements that are generated during the first phase of development have a great influence on the design and structure of the system. Some requirements are subject to change if the technologies available cannot provide the proper capabilities to have a certain feature. Once the technologies are filtered and requirements are solidified, designs for the system can move on to the next phase of development.

1. System architecture design
2. Sub-system design
3. Establishing concrete system capabilities

System Development & Integration

This section discusses the final stage of the development cycle which includes system integration and testing. Once components are procured, they will be tested and evaluated to be proved functional. After testing is completed, each component will be integrated into their corresponding sub-system. A final testing stage will be performed to show that all parts of the system are interfaced correctly. Once the system is fully debugged and tested, the product is ready for demonstration.

1. Develop system components
2. Component integration
3. Testing and evaluation
4. System delivery

TASK	ASSIGNED TO	PROGRESS
Phase 1: Research and Planning		
Requirement Generation	Team	100%
Component Evaluation	Team	100%
Component Downselect	Team	95%
Phase 2: System Design		
Procure system components	Team	20%
Create electrical schematics	Mariela/Alex	50%
Design communication between sensors and MCU	Daniel	10%
Power system design	Mariela/Alex	25%
PCB Design	Alex	0%
Drowsiness detection algorithm design	Michael	40%
Phase 3: System Integration and Development		
Create mobile application	Daniel/Michael	0%
Develop and integrate power system	Mariela/Alex	0%
Interface mobile application with MCU	Daniel/Michael	0%
MCU communication between sensors	Daniel	0%
Process images and send output to MCU	Michael	0%
Integrate system components onto helmet	Team	0%
Testing	Team	0%

Figure 80. Milestones and task assignments

8.2 Budget and Finance

Proper budgeting and financial planning are essential to the success of any project. While generating the requirements for the system, it is important to target the capabilities that may require more funding. If proper evaluation of these areas are not completed, vital components of the system may not receive the appropriate resources, which can ultimately result in a failure to deliver the product. This section gives a brief outline of the project's budget and pending expenses. Since this project is not being funded externally, each member has agreed to split the expenses evenly.

Below is the current part expense breakdown:

Table 16: Expense Breakdown

Parts	Quantity	Cost
Gyro Sensor SN-ENC03R0 *	1	$13.65
HC-SR04 Ultrasonic Range Finder	2	$7.90
SSD1306 0.96-inch I2C OLED display *	2	$8.00
1669 Stereo Enclosed Speaker Set *	1	$7.50
Motorcycle Full Face Helmet HJM A110 Matt Black *	1	$50.00
Raspberry Pi 4 (2GB)	1	$41.80
MSP430FR6989	1	$4.47
Raspberry Pi NoIR Camera Module V2 - 8MP *	1	$27.48
Total	10	$160.80

Key: * - Parts not yet acquired

Table 17: Overall Expense Breakdown

Category	**Estimated**	**Actual (4/2/20)**
Parts	$240.00	$160.80
Misc. Components	$60.00	$0.00
Software	$40.00	$0.00
Total	$300.00	$160.80

9. Conclusion

When the four of us first got together as a group to discuss what we wanted to accomplish for our Senior Design project, we decided that we wanted to not only make a project that would satisfy the requirements of the University of Central Florida so that we could graduate but to also have something proud of to be able to display future potential companies that we are trying to get hired at. After researching several previous senior design projects from all over the globe we landed on a project that we knew would make a difference, but also put all of our engineering knowledge on display. The main goal of our project was to fix a problem that was affecting thousands of people all over the world and do it with a very attainable budget. After many hours of research and development we were able to mostly finish and create the Smart Helmet with all of the features that we had in mind when we first got together.

A pretty big event happened while we were in the middle of our senior design 1 class, the sudden pandemic COVID-19 that affected all of the globe. This event put all of us in situations that nobody expected. Not only did we have the stress of trying to complete this project, but we had more stress brought upon us from the COVID-19 situation. We could not get together in person and we were left speaking with each other over phone or video chatting. Even though all of these things were mounted against us we managed to finish and put together an almost complete Senior Design 1 Report. This showed how hard that we worked and put our teamwork and communication skills to good use. We will continue to do the online route going into Senior Design 2 and look forward to completing our project.

Appendices

Appendix A - Copyright Permissions

services@jarvish.com

Image Copyright Permission

Hello,

my name is Alejandro Velasco an Electrical engineer student at the University of Central Florida. I would like to request permission to use your image below that I will be using for my Senior Design project

SHOP FEATURES QUINTEMEN ROAD SHOW 0

CONTACT US

Get in touch for press, careers, and product queries - we'd love to hear from you!

Name *

alejandro

First Name

velasco

Last Name

Email Address *

bajavelasco@gmail.com

Subject *

image copyright permission

Message *

Hello,

my name is Alejandro Velasco an Electrical engineer student at the University of Central Florida. I would like to request permission to use your images for my senior design project.

SUBMIT

SUBSCRIBE

Sign up with your email address to receive news and updates.

First Name

Last Name

Email Address

SIGN UP

We respect your privacy. You'll only get the coolest, rarest updates from us.

info@lmgpr.com

Image copyright permission

Hello,

my name is Alejandro Velasco an Electrical engineer student at the University of Central Florida. I would like to request permission to use your image below that I will be using for my Senior Design project

hello@crosshelmet.com

Image Copyright Permission

Hello,

my name is Alejandro Velasco an Electrical engineer student at the University of Central Florida. I would like to request permission to use your image below that I will be using for my Senior Design project

Figures 10 & 11 (Pending approval)

Name (required)

Mariela Barragan

Email Address (required)

marbarragan@live.com

Confirm Your Email Address (required)

marbarragan@live.com

Subject (required)

Copywright Permission

Describe Your Question Briefly (required)

Hello, Im currently an electrical engineering student at the University of Central Florida and i'm working on my senior design group project. I'm emailing you in order to request permission to use some of the diagrams/figures found on your website. We will provide all neccesary citations and referances in the Final Report. It will only be used for academinc purposes

Thank you in advance!
Mariela Barragan

☑ I'm not submitting a technical question about my project (required)

Figures 12, 14, 17, 18, 20, 35, 36, 37, 38, 39, 46 (Pending approval)

Hello,

I'm currently an electrical engineering student at the University of Central Florida and I'm working on my senior design group project. I'm emailing you in order to request permission to use some of the diagrams/figures found on your website. We will provide all necessary citations and references in the Final Report. It will only be used for academic purposes.

Thanks!
Mariela Barragan

Sent from Mail for Windows 10

Figures 15 & 16 (Pending approval)

Name *

Mariela	Barragan
First	Last

Email *

marbarragan@live.com

Comment or Message *

Hello,

I'm currently an electrical engineering student at the University of Central Florida and I'm working on my senior design group project. I'm emailing you in order to request permission to use some of the diagrams/figures found on your website. We will provide all necessary citations and references in the Final Report. It will only be used for academic purposes.

Thanks!
Mariela Barragan

Figure. 19 (Pending approval)

From: marbarragan@live.com

To: contact@rfwireless-world.com; Cc & Bcc

Copyright Permission

Hello,

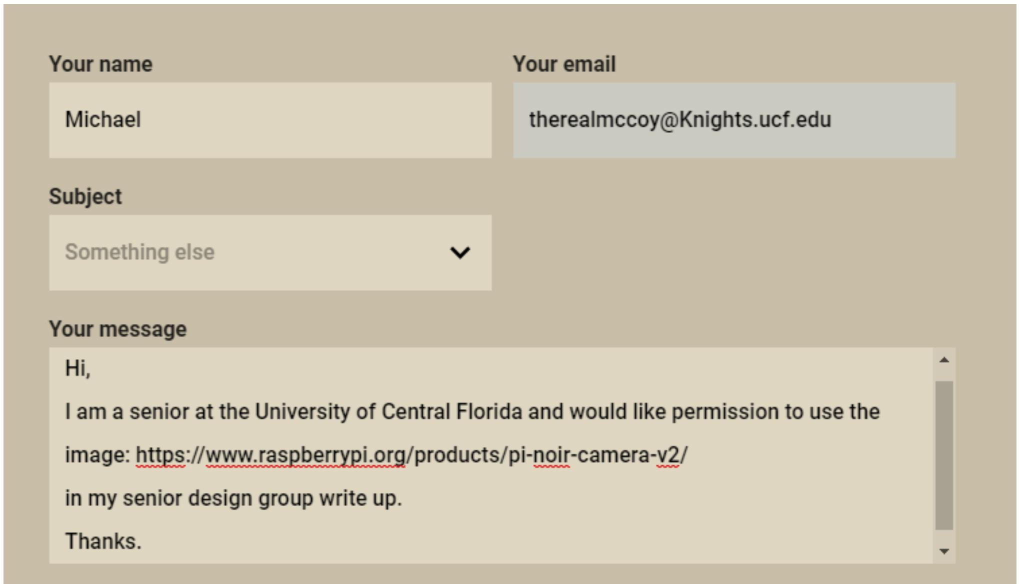
I'm currently an electrical engineering student at the University of Central Florida and I'm working on my senior design group project. I'm emailing you in order to request permission to use some of the diagrams/figures found in your website. We will provide all necessary citations and references in the Final Report. It will only be used for academic purposes.

Thanks!
Mariela Barragan

Sent from Mail for Windows 10

Figure. 43
https://blog.gogreensolar.com/2015/02/ac-vs-dc-breakers.html

Figure. 21 (Pending approval)



Your name

Michael

Your email

therealmccoy@Knights.ucf.edu

Subject

Something else

Your message

Hi,

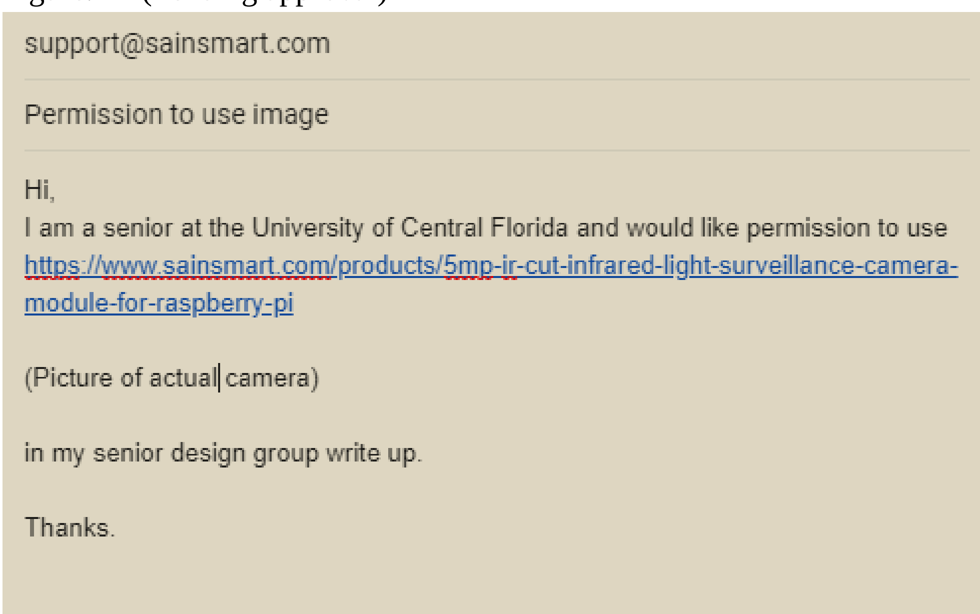
I am a senior at the University of Central Florida and would like permission to use the image: https://www.raspberrypi.org/products/pi-noir-camera-v2/

in my senior design group write up.

Thanks.

https://www.raspberrypi.org/products/pi-noir-camera-v2/

Figure. 22 (Pending approval)



support@sainsmart.com

Permission to use image

Hi,

I am a senior at the University of Central Florida and would like permission to use https://www.sainsmart.com/products/5mp-ir-cut-infrared-light-surveillance-camera-module-for-raspberry-pi

(Picture of actual camera)

in my senior design group write up.

Thanks.

https://www.sainsmart.com/products/5mp-ir-cut-infrared-light-surveillance-camera-module-for-raspberry-pi

Figure. 30 (Labeled for reuse)

https://www.flickr.com/photos/msvg/5197694152/

Figure. 31 (Pending approval)

Your name

Michael

Your email

therealmccoy@Knights.ucf.edu

Subject

Something else

Your message

Hi,

I am a senior at the University of Central Florida and would like permission to use the image: https://www.raspberrypi.org/products (Image of Raspberry Pi 4)

in my senior design group write up.

Thanks.

https://www.raspberrypi.org/products/

Figure. 32
Created using: https://wireframe.cc/

Figure. 33 (Labeled for noncommercial use)
https://commons.wikimedia.org/wiki/File:C-130J_Co_Pilot%27s_Head-up_display.jpg

Figure. 34 (Pending approval)

cholton@endeavorb2b.com

Permission to use image

Hi,
I am a senior at the University of Central Florida and would like permission to use

https://www.laserfocusworld.com/optics/article/16550165/novel-optics-plastic-microoptics-cater-to-automotive-hud-design

(Image of HUD how it works visual)

in my senior design group write up.

Thanks.

https://www.laserfocusworld.com/optics/article/16550165/novel-optics-plastic-microoptics-cater-to-automotive-hud-design

Appendix B - Datasheets

HC-SR04 Ultrasonic Range Finder
https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

MPU-6095 3 Axis Gyro
https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

MSP430FR6989
http://www.ti.com/lit/ds/symlink/msp430fr6989.pdf

Raspberry Pi
https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf

HC-05 Bluetooth Module
http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf

Raspberry Pi NoIR Camera Module V2
http://www.farnell.com/datasheets/2056180.pdf

MCP4725 DAC
https://www.sparkfun.com/datasheets/BreakoutBoards/MCP4725.pdf

LM386 Amplifier
http://www.ti.com/lit/ds/symlink/lm386.pdf

Appendix C - References

Benne Bakker-Anthony Carson-Anthony Carson-Benne Bakker- Tushar- Tushar-Arduino- Arduino- Arduino - https://www.makerguides.com/sharp-gp2y0a21yk0f-ir-distance-sensor-arduino-tutorial/

Distance Sensor Comparison Guide. (n.d.). Retrieved from https://www.sparkfun.com/distance_sensor_comparison_guide

RF Wireless World. (n.d.). Retrieved from https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Tilt-Switch-or-Tilt-Sensor.html

AC vs DC Breakers. (2015, February 24). Retrieved from https://blog.gogreensolar.com/2015/02/ac-vs-dc-breakers.html

OLED Technology. (n.d.). Retrieved from https://www.usmicroproducts.com/learn/oled-technology

What is a DAC (Digital-to-Analog Converter) and How Does it Work? (2019, June 9). Retrieved from https://www.audioadvice.com/videos-reviews/what-is-a-dac/

Build a Great Sounding Audio Amplifier (with Bass Boost) from the LM386. (2018, June 21). Retrieved from https://www.circuitbasics.com/build-a-great-sounding-audio-amplifier-with-bass-boost-from-the-lm386/

Hubner, J. (2019, August 6). Power Supply Design Tutorial. Retrieved from https://www.powerelectronicsnews.com/power-supply-design-tutorial/

Gyro sensors - How they work and what's ahead: about Gyro sensor: Technical Information: other Information. (n.d.). Retrieved from https://www5.epsondevice.com/en/information/technical_info/gyro/

CrossHelmet - the smart motorcycle helmet. (2020, March 30). Retrieved from https://www.kickstarter.com/projects/491835187/crosshelmet-the-smart-motorcycle-helmet

802.15.1-2005 - Ieee Standard For Information Technology-- Local and Metropolitan Area Networks-- Specific Requirements-- Part 15.1a: Wireless Medium Access Control (mac) and Physical Layer (phy) Specifications For Wireless Personal Area Networks (wpan)
https://standards.ieee.org/standard/802_15_1-2005.html

Aosp Java Code Style For Contributors : Android Open Source Project
https://source.android.com/setup/contribute/code-style

A Global Team. (n.d.). Retrieved from https://crosshelmet.com/contact/index.html

SKULLY Technologies Becomes First to Market With Its FENIX AR Smart Helmet. (2018, September 25). Retrieved from https://www.businesswire.com/news/home/20180925005436/en/SKULLY-Technologies-Market-FENIX-AR-Smart-Helmet

STM32F102CB. (n.d.). Retrieved April 20, 2020, from https://www.st.com/en/microcontrollers-microprocessors/stm32f102cb.html

ATSAMC20G17A. (n.d.). Retrieved April 20, 2020, from https://www.microchip.com/wwwproducts/en/ATSAMC20G17A

(n.d.). Retrieved from http://www.ic0nstrux.com/index.php?route=information/articles&article_id=6

Best smart motorcycle helmets to smarten up your ride. (2019, October 30). Retrieved from https://www.androidauthority.com/best-smart-motorcycle-helmets-1014050/

JARVISH: The Smartest Motorcycle Helmet Ever Made. (2020, April 17). Retrieved from https://www.kickstarter.com/projects/308263621/jarvish-the-smartest-motorcycle-helmet-ever-made

Quin Design Helmets. (n.d.). Retrieved from https://quintessential.design/

Kendall, M. (2016, August 24). Skully: How hot start-up became latest Silicon Valley darling to flame out. Retrieved from https://www.eastbaytimes.com/2016/08/22/skully-how-hot-start-up-became-latest-silicon-valley-darling-to-flame-out/

List of battery types. (2020, March 31). Retrieved from https://en.wikipedia.org/wiki/List_of_battery_types

OLED. (2020, April 8). Retrieved from https://en.wikipedia.org/wiki/OLED

Wi-Fi. (2020, April 19). Retrieved from https://en.wikipedia.org/wiki/Wi-Fi
Shawn, Shawn, & Veepee Controls. (1969, September 1). All about Proximity Sensors: Which type to use? Retrieved from https://www.seeedstudio.com/blog/2019/12/19/all-about-proximity-sensors-which-type-to-use/

Clearscience. (2015, April 14). clearscience. Retrieved from https://clearscience.tumblr.com/post/116385215853/in-their-paper-an-ultrafast-rechargeable

Bluetooth. (2020, April 19). Retrieved from https://en.wikipedia.org/wiki/Bluetooth

Thermographic camera. (2020, April 1). Retrieved from https://en.wikipedia.org/wiki/Thermographic_camera

Drowsy Driving: Asleep at the Wheel. (2019, November 7). Retrieved from https://www.cdc.gov/features/dsdrowsydriving/index.html

Gite, V., Gite, V., & Gite, V. (2019, June 24). Raspberry PI 4 Released - Complete specs and pricing. Retrieved from https://www.cyberciti.biz/linux-news/raspberry-pi-4-released-complete-specs-and-pricing

Buy a Pi NoIR Camera V2 – Raspberry Pi. (n.d.). Retrieved from https://www.raspberrypi.org/products/pi-noir-camera-v2/

5MP IR-CUT Infrared Light Surveillance Camera Module for Raspberry Pi. (n.d.). Retrieved from https://www.sainsmart.com/products/5mp-ir-cut-infrared-light-surveillance-camera-module-for-raspberry-pi

Software Testing Life Cycle: Different stages of Software Testing. (2019, May 22). Retrieved from https://www.edureka.co/blog/software-testing-life-cycle/

Appendix D - Explanation of Missing Components

During Spring break UCF campus was closed due to COVID-19, shortly after postal services worked on a low staff and packages took longer to arrive to us. Unfortunately, we could only test the Ultrasonic sensor which we already had from Junior design.
The rest of the components should arrive somewhere during the break and we're planning to test them immediately as they arrive.

Below are receipts and invoices of all the components we order and are still waiting for their arrival: