Smart Helmet

Mounted Motorcycle Proximity Sensor with Helmet Display

Julian Bonnells, Jorge De Gouveia,
Jeremy Reimers, Blake Scherschel

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450, U.S.A.

***Abstract* — An original electrical and computer engineering senior design project, the Smart Helmet is a mounted motorcycle proximity sensor with helmet display. The Smart Helmet system is planned to be a solution for any motorcycle rider that wishes to increase his or her awareness and safety while driving on the road. The Smart Helmet system will be able to gather proximity information of surrounding objects, using proximity sensors, to scan an area behind the rider to determine if other vehicles are close by. The system will draw power from the motorcycle and interpret various electrical signals as well. This motorcycle data will then be sent wirelessly from the motorcycle-mounted module and received by the helmet module. The helmet display will alert the rider if a vehicle is too close to the motorcycle when the rider attempts to change lanes.**

***Index Terms* — Microcontrollers, Proximity Effects, Data Transfer, Bluetooth, Solar Energy**

I. Introduction

The thrill of riding a motorcycle is something many people marvel for. What usually get in the way of the enjoyment are the dangers that come with sharing the road with other vehicles at high speeds. Modern technology has improved the safety features of most regular vehicles, but motorcycles have seemingly not benefitted from the same technological revolution.

This project aims to give motorcyclists an easy and effective way of staying aware of the potential dangers lingering outside of their natural field of view. By incorporating a Heads-Up Display (HUD) on the front of the helmet, motorcyclists should not have to rotate their entire body and head away from their line of travel to know if there are vehicles to the side, back, or even their blind spots. The device will be divided into two parts: wireless attached helmet HUD, and a rear-mounted area detection sensor which will be connected to a motorcycle interface. The bike interface will send the critical information, wirelessly, to the helmet HUD.

Others have attempted to deliver similar functionality, but they are either too unreliable or too overpriced for the realistic consumer. Our device aims to be a dependable and simple solution at a fraction of the price. The planned consumers for our project will be vendors, seeking to sell next generation safety equipment to motorcyclists. The design goals listed will be used as guidelines when developing the hardware and software requirements. In summary, our project seeks to meet the following goals:

- Research, design, and implement a cost-efficient motorcycle proximity detection solution
- Incorporate distance sensing methodologies to detect objects from a far
- Successfully detect oncoming automobiles while in motion
- Integrate wireless communication between microcontrollers to transmit proximity data
- Implement the design to be easily applicable for other automobile systems
- Design the Smart Helmet to be operable by many types of users with little to no training required
- Follow industry standards to ensure safety, reliability and environmental care

II. System Concept

The smart helmet will include two wireless communicating systems: the Mounted Motorcycle Interface and the Helmet Heads Up Display (HUD). These two systems operate completely asynchronously and communicate via wireless transmissions. The Mounted Motorcycle Interface is further divided into two parts: the Front-Mounted Motorcycle Interface and Rear-Mounted Proximity Detection Module.

The Front-Mounted Motorcycle Interface is involved with hosting the MCU, wireless transmitter, and interface for reading the status and diagnostics from the bike. It will be mounted directly onto the front of the motorcycle, and will read rider's inputs (turn signal, etc.), listen for proximity information from the rear-mounted proximity detection module, and send uniform information wirelessly, via Bluetooth, to the Helmet Heads-up Display (HUD) module. Fig. 1 shows the setup of the Bike Mounted Module board block diagram.

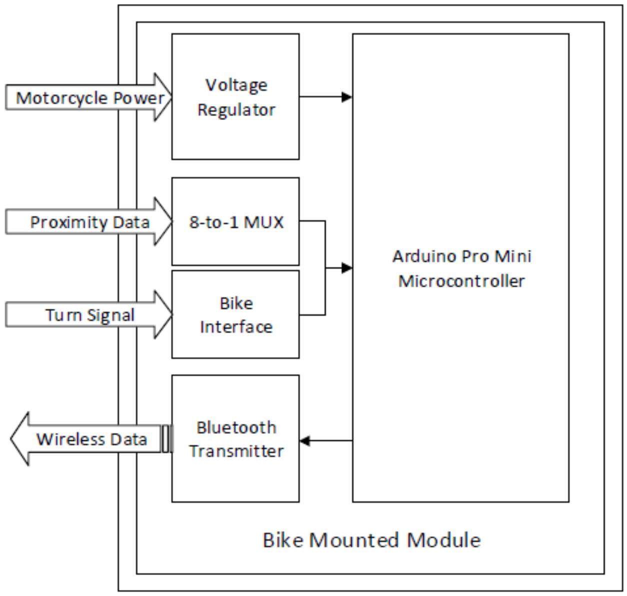

Fig. 1. Bike Mounted Module block diagram showing proximity detection processor, motorcycle interface, and wireless transmitter.

The Rear-Mounted Proximity detection module will consist of 6 proximity sensors, split between left and right sides, and mounted on the rear of the motorcycle to gauge a wide field behind the motorcycle. If the rider is signaling to merge, but an object is within 20 feet of the bike, an alert will be sent to the helmet Heads-Up Display (HUD) module. The proximity sensors that are available for this project are well enough advanced that measurement time should be close to instantaneous. The rear-mounted module is essentially just all of the proximity sensors in a daisy-chain configuration, and there are no applicable software requirements.

The Helmet Heads-Up Display (HUD) module will be located in the helmet of the user. It will listen for information from front-mounted module wirelessly, and display bike information to rider via a Heads-Up Display (HUD). The module will also alert the rider, visually, if an object is in a critical proximity to the bike. It will be powered by a 4.15V rechargeable battery pack, which will also be accessible by the user in case it needs a replacement. Solar panels will also be installed onto the helmet to extend battery life during the daytime. Fig. 2 shows the setup for the Helmet Mounted Module board block diagram.

The HUD will be designed to give a visual proximity warning to the rider, when the rider is signaling to merge, but is not clear to do so. An audible tone will also sound if the rider is not clear to merge, alerting the driver even more so, to help avoid an accident. The smart helmet will be lightweight, as to not over encumber the user, and communicate with the bike wirelessly, in order to give full freedom of motion. A minimum of a 2-hour battery life is expected from a full charge, with prolonged life from solar panels constantly charging the battery. USB charging will also be an option for when the Smart Helmet is not in use.

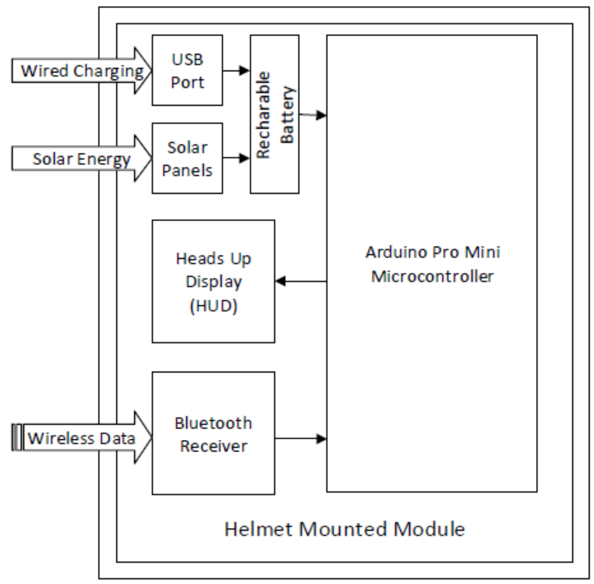

Fig. 2. Helmet Mounted Module board block diagram showing graphics processor, wireless receiver, HUD, and power regulation

III. System Components

The Smart Helmet system is best described as a collection singular modules that are interfaced together to create the final product. This section will introduce each module and give a brief overview of its function to the system as a whole.

A. Microcontroller

The Smart Helmet design uses two Arduino Pro Mini microcontrollers that act as relays between the helmet module and bike module. This microcontroller was selected because it has a large number of digital and analog input/output pins, which is perfect for interfacing with our peripherals. This microcontroller was also chosen due to the simplicity and familiarity with Arduino. This chip runs at 16 MHz, which is necessary to process the incoming proximity sensor data as well as transmitting and receiving wireless data.

B. Proximity Sensor

The LM-MaxSonar EZ series ultrasonic proximity sensor was the sensor chosen for the Smart Helmet design. Measuring at only 0.6" x 0.9" x 0.8, this tiny proximity sensor is able to sense objects up to 6.45 meters away while still remaining relatively inexpensive. Ultrasonic sensors use high frequency sound waves to detect and localize objects. These sensors will be used to collect proximity data around the rider. Multiple proximity sensors will send sensing data to the Arduino microcontroller to be consolidated and processed when the rider chooses to merge left or right via the motorcycle's turn signals.

C. Wireless Data Transceiver

Wireless communication will be required between the two microcontrollers to send and receive proximity sensing data. For the Smart Helmet design, we opted to use the HM-10 Bluetooth transceiver for both the bike module and helmet module. The Bluetooth device on the bike module will act as a master and will send proximity data to the helmet. The Bluetooth device on the helmet module will act as a slave and will receive the proximity data to display on the HUD. The reason we chose this Bluetooth module was because of the low power consumption and the reliability of the communication over long distances. Fig 3. Shows the communication between the two Bluetooth devices.

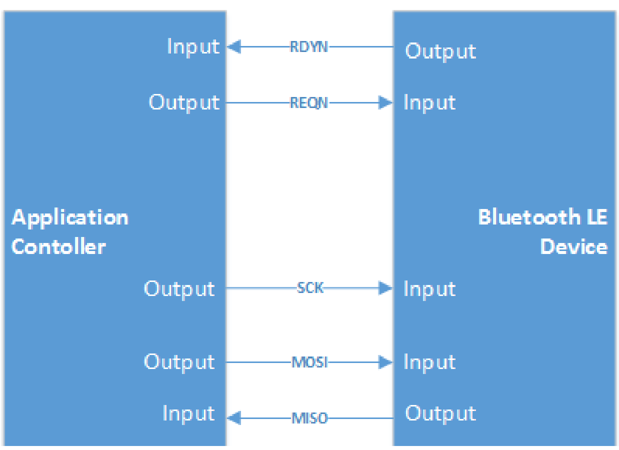

Fig 3. ACI Interface between application controller and BLE device

D. Visual Display

The HTDS-WI96 is a 0.96" OLED display used on the helmet module to act as a Heads-up Display for the rider. We chose this display because of the low power usage and the self-luminous properties of OLED. Because this display is to be used on the helmet module, we also needed to ensure that is did not cause any blind spots or distractions for the rider. This specific visual display was also considered because of its compatibility with the Arduino series.

E. Battery Charger

Lithium rechargeable batteries will be implemented into the Smart Helmet design, not only to keep costs down, but also to prolong the battery life for the helmet module. To charge these batteries, we have designed a battery charging circuit that utilizes a MCP73831 charge controller. This controller features adjustable current charging from 15mA to 500mA and takes roughly 165 minutes to achieve a full charge.

F. Solar Energy

Solar power will be implemented onto the helmet module in an effort to extend the battery life. The helmet module will use 40 mA solar panels and a voltage regulator to drop down the voltage to the proper levels. The panels will be small enough to be form-fitting, but large enough to provide enough power. The panels will be wired directly to the same charging circuit at the USB. The solar panels will charge the lithium batteries while the helmet is out in daylight, but it will be up to the rider to still use the battery charger for when there isn't sufficient sunlight.

G. Bike Interface

The Front-Mounted Motorcycle MCU has the task of continuously watching the status of the motorcycle's turn signal switch. This switch is controlled by the driver of the motorcycle and a signal of high would indicate that the driver is intending to turn or merge in that direction. The Smart Helmet project uses this information to warn and potentially stop the driver from accidentally merging or turning into a hazard on the road. This interface is accomplished by reading and measuring the voltage effected by the turn signal switch. The demo motorcycle used as a test for the interfacing is the Honda Supersport CB400F.

IV. HARDWARE DETAIL

In this section, each of the components introduced in the previous section will be talked about in technical detail.

A. Microcontroller

The MCUs utilized on both the Helmet and Motorcycle Mounted modules is the Atmega328p chip stations on an Arduino Pro Mini board. The Arduino Pro Mini can be closely related to the Arduino Uno and has the included ability to support enough digital and analog input/output connections for the project on both endpoints. The Motorcycle Mounted module MCU reads the sensor and proximity values and sends the information to the Helmet Mounted module where the data is received and displayed to the visual display. The two MCUs communicate via a Bluetooth connection which is connected over a serial connection.

The specifications for the Arduino Atmega328p microcontroller are listed below (1). These specifications fit the requirements set by the team during the design stage.

- Architecture: 8-bit AVR RISC
- Processor Speed: 20 MHz
- Operating Cores: 1
- Storage ROM: 32 KB
- SRAM: 2 KB
- I/O Pin Ports: 26
- Power Consumption: A+

The Helmet Mounted module has a simple logic flow since it acts as a straightforward client that only listens for information. The logic flow can be seen below.

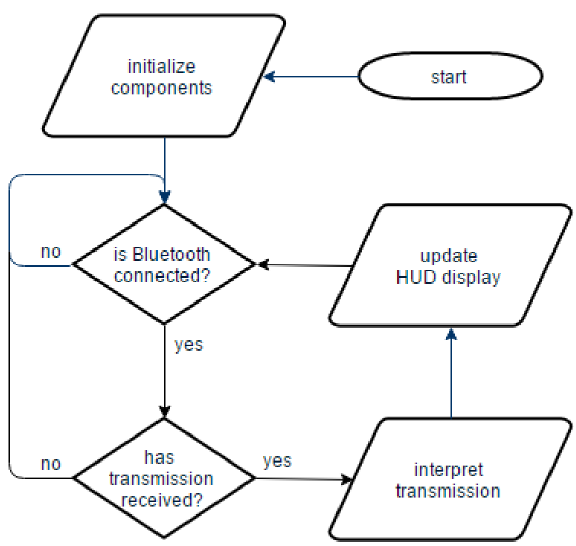

Fig. 4. Helmet Mounted Module logic flow

B. Proximity Sensor

The LM-MaxSonar EZ series proximity sensors are classified as ultrasonic transceivers. They are capable of both transmitting and receiving ultrasound waves, and converting those waves to electrical signals. The MaxSonar EZ series sensors evaluate the echoed waves from a target in order to detect distance from the sensor. The MaxSonar series produces vibrations anywhere at a frequency from 40 kHz to 250 kHz which is far out of reach of what the human ear can pick up. Distance of the foreign objects is measured using the Time of Flight of the sound vibrations transmitted as shown in Equation 1. The speed of the sound vibrations is a function of temperature which may negatively influence the readings if not properly calibrated for.

$$Distance = \frac{Speed\ of\ Particle \times Time\ of\ Flight}{2} \quad (2)$$

Each of the proximity sensors will be wired directly to an Arduino microcontroller. The MaxSonar EZ sensors will be wired independently to one another. Each sensor will have its own respective analog pin on the microcontroller. Each of the sensors will receive 5 volts from the microcontroller and will be grounded to the ground on the microcontroller.

The specifications of the MaxSonar EZ sensors are listed below (3). These specifications fit the requirements set by the team during the design stage.

- Size: 15.52 x 22.36 x 20.00 mm (0.61 x 0.88 x 0.79 in.)
- Weight: 4.23 g
- Power Needed: 2.5 to 5.5 V
- Range: 0 to 254 in. (6.45 meters)

When the sensors receive power from the microcontroller, they will capture sensor data and return a measurement in inches back to the microcontroller. From here the microcontroller can process the data, convert the measurement, and send it over to the other module before issuing a delay. During the delay, the sensors return to idle. Since each sensor is wired to their own unique analog pin, the microcontroller has the ability to trigger each sensor individually depending on which turn signal is turned on. Fig. 5 shows this sensor software logic.

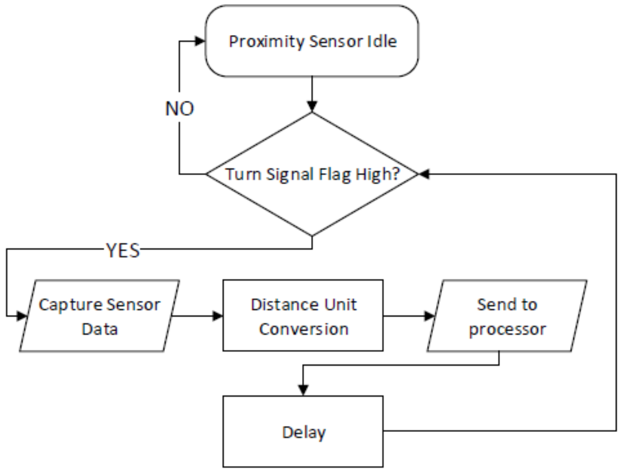

Fig. 5. Proximity sensor software logic loop

C. Visual Display

The helmet module will use two 0.96" OLED displays sold by DIYMall. It has a resolution of 128x64 which can be programmed fully using the Arduino IDE and libraries. These displays require minimal power, only 0.08W, when compared to other types of visual displays such as TFT and LCD. This display was also chosen because of its high contrast, which makes it easily viewable even when outside in sunlight. The display requires either 3.3V or 5V supply voltage to power on, which it receives from the Arduino MCU.

This visual display uses two input/output pins which act as a clock signal and data signal. The I2C-bus data signal (SDA) acts as a communication channel between the Arduino MCU and the display. Both the data and the acknowledgement are sent through this signal. The other IO pin is the clock signal (SCL). Transmission of information on the SDA line depends on the clock signal on this pin. Each transmission of data takes place during a single clock period of SCL. The clock is provided by the Arduino MCU, which acts as a master control. Internal to this display is a SSD1306 CMOS driver which controls the OLED dot-matrix display. The internal Graphic Display Data RAM (GDDRAM) on this chip is 128 x 64 bits wide and is divided into eight pages, from 0 to 7. When one data byte is written into the GDDRAM, the corresponding page of the current column is filled. By manipulating the GDDRAM buffer, we will be able to output our desired data to the visual display.

D. Wireless Data Transceiver

The Front-Mounted Motorcycle Interface MCU will be utilizing a Bluetooth module via a 2-pin serial connection. The module that the Smart Helmet will be using is the Cc2540 Bluetooth BLE 4.0 Serial module. The MCU can communicate with the module as if it were an input-output byte stream over the serial stream.

When information is sent over the wireless connection, each byte is received one at a time and there is no guarantee of delivery. This resulted in the design of a psuedo-UDP-style logic cycle where each byte sent over the connection can signify a certain piece of the overall packet of information. The flow of the logic is below.

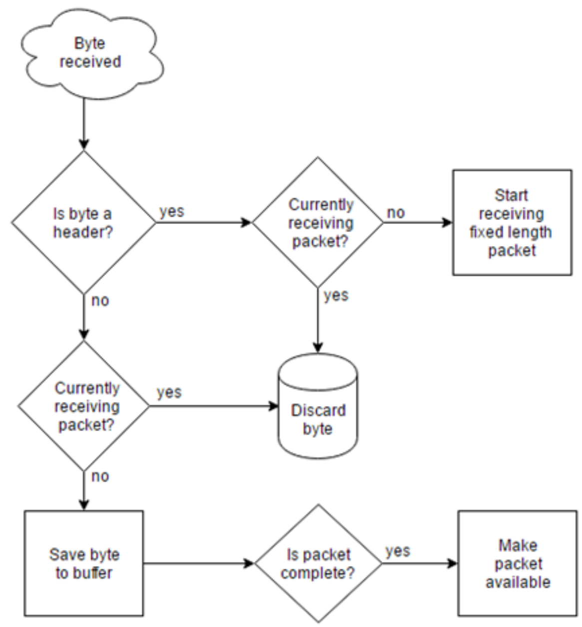

Fig. 6. Wireless communication packet processing loop

E. Battery Charger

The battery charging circuit shown in Fig. features adjustable current charging from 15mA to 500mA with a required power supply of at most 6V. Power can also be drawn from a USB port, which produces 5V and 150mA. When using USB for charging, the charge current should not be set higher than 150mA. The charging current is adjusted using the switchboard and varying output values can be attained by selecting which switches are high. For the Helmet Mounted module, we will use this voltage regulator circuit to provide a constant +5V to the system.

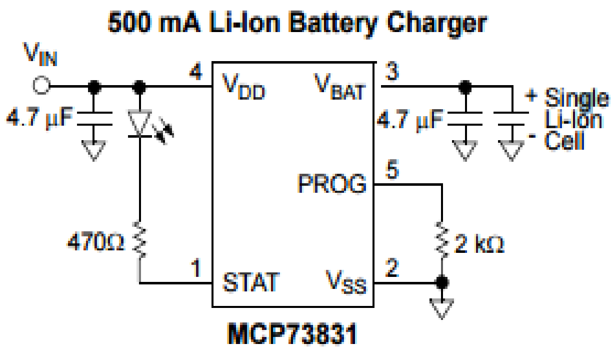

Fig. 7. MCP73831 battery charging circuit [4]

F. Solar Energy

The primary power source for the helmet module will be from the solar panel situated on top of the rider's helmet. The solar panel will can deliver +5V of operating power to the microcontroller and battery charging circuit. The battery charger will be responsible for charging the Lithium-Ion rechargeable battery, so a constant +5V from the solar panel is necessary. If adequate sunlight cannot be reached, the solar panel will stop supplying power and instead the battery will be the main source of power for the Helmet Module.

G. Bike Interface

The bike interface is accomplished by using two wires as the interface for determining the state of the turn signal by measuring the active voltage at that point in the motorcycle's circuitry. To safely accomplish this, a simple voltage divider is needed to step down the 12V source that typically is used to power the LEDs. The Atmega328p is very adaptable in its ability to read analog inputs, but the voltage must be within a specific range to avoid damaging the MCU. Any voltage below 5V is considered safe for the MCU, but a target of ~3V will be used for extra precaution against possible damage.

To divide a voltage in a circuit, a voltage divider is used to split the voltage over at least two resistors. Two resistors selected based on the following voltage divider formula.

$$V_{out} = V_{in} * \frac{R_2}{R_1 + R_2} \quad (5)$$

Using the voltage divider formula, a motorcycle turn signal module was created to intercept the turn signal from the motorcycle. The circuit is shown in the figure below.

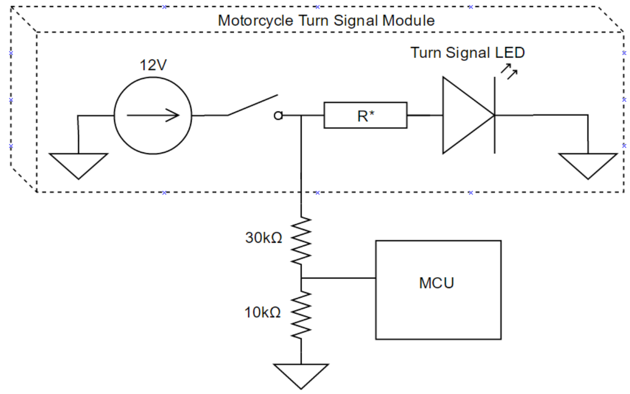

Fig. 8 Interface Circuit for Motorcycle Turn Signal Module

The above circuit allows the MCU to safely read for a high or low voltage from the motorcycle's turn signal module. While a more standard way of interfacing with the motorcycle was definitely preferred, this was the most universal way to support as many motorcycles as possible to use the Smart Helmet project.

V. Software Detail

The software for the Smart Helmet is split into two parts operating on two separate asynchronous MCUs. The two MCUs will be running completely independently of each other and will only communicate over wireless transmissions. Since the decision of which MCU to use was decided to be the Atmega328p, we only have 2KB of RAM to work it, thus requiring the software we deploy to be small and efficient.

The code is split into namespaces, or modules, where each has a very distinct set of tasks to manage. Each namespace encapsulates the underlying complexity of what the module is tasked to manage, thus making the top-level coding of the logic very straightforward and easy to debug. Fig. 9 shows these modules and what tasks they control. Please note that each module has submodules to control more specific tasks, and that these are only the top-level modules exposed.

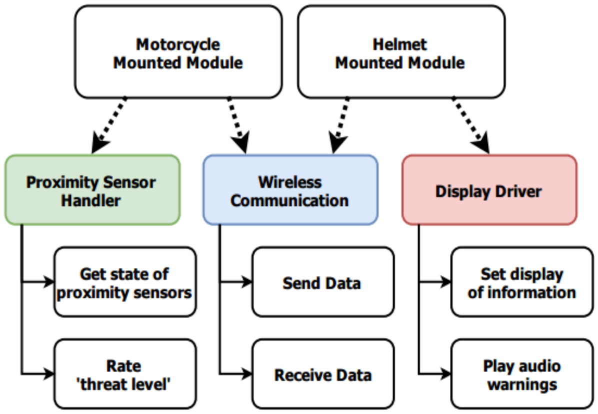

Fig. 9. Module controllers for each system subassembly

The "Wireless Communication" module is a wrapper around a serial connection to the wireless chip that is communicating between the Motorcycle and Helmet. This module is designed for easy top-level use to send and receive data in a straightforward manner.

In terms of the "Proximity Sensor Handler", the sensor readings depend on the distance from an oncoming vehicle where the proximity reading will be categorized into a 0-5 rating system. 0 is no threat and 5 is an immediate danger. This simple rating system will be presented to the user visually through the "Display Driver" when received on the Helmet Module. The low rating scale and minimal items being displayed not only correlate well with the type of data being displayed but also are a minimal memory load on the user. The user will not have to process the information given to them, they will be simply react to the information being displayed instantaneously.

VI. Board Design

The entirety of the Smart Helmet design is implemented on two separate Printed Circuit Boards (PCBs). The circuit boards contain two copper layers which hold all the connections between the components. Each component is interfaced to each other through soldering and wire-wrapping which was performed manually.

The PCB prototype was drawn out using the Eagle PCB design software. After testing all the applicable signals using breadboards, the design was replicated virtually and sent off to be printed. We opted to use the printing service, Elecrow to print our circuit boards.

VII. Administrative

We have decided that the working design prototype should be completed no later than April, to give us ample time to redesign, if necessary, and add additional features. We are confident that we can meet all of these milestones and create a working prototype on time.

To create a product that can provide for consumer needs, as well as being affordable to the consumer, design and manufacturing costs had to stay low. The only competing company right now has their product available for retail at around $500. Our goal is to create a simpler product for less, so that even the consumer on a budget can have more safety on the road.

In order to keep costs low, power and efficiency are high utilizing cheap, but still reliable, components. Proximity sensors and a helmet display are the two most expensive pieces of equipment in our design. The LV-MaxSonar sensors are being used for this project, and cost about $30 per unit. The helmet display had to be small enough to allow for maximum visibility by the rider, while also being visible enough to be usable. The .96 inch, 128x64, single color displays that are incorporated into this project cost about $10 a piece and provide enough clarity for the user, while still small enough as to not obstruct vision of the road.

By utilizing innovative techniques for power and system design, the total cost of this project is just under $250 for the system. If we were to market and sell this product at a high volume, mass production of this design can reduce this cost by at least 10%. Retail price for the system can be speculated at about $350; high enough to be profitable, but still low enough to be competitive.

VIII. Biography

Julian Bonnells
Currently 23 years old, Julian is pursuing a bachelor's degree in Computer Engineering. He hopes to obtain a master's degree once he has completed his undergraduate studies. Using the skills that he has gained from various projects and internships throughout his academic career, Julian hopes to contribute his knowledge of software and hardware to the overall Smart Helmet design.

Jorge De Gouveia
Currently 23 years of age and a senior at the UCF Jorge's goal is to be able to apply his programming knowledge in the real world and integrate it with hardware in order to make a product that could be of use to the general public. Jorge sees Smart Helmet not only as an

invaluable learning experience but also a gateway to either of his dream careers as a software engineer in the space industry or in the video game industry. Creating a videogame that is loved by thousands of people or being on a team that discovers the next big space discovery is what drives him to be the best computer engineer that he can be.

Jeremy Reimers

Currently 28 years old, Jeremy is a senior at the University of Central Florida in Electrical Engineering, continuing his passion for learning how things work, and learning how to improve them. He has experience with electronics design and assembly, programming, 3D Modelling, and IT. He hopes to use his skills to improve the way of life for few, or for many, but overall aims to make a positive impact on the world. For the Smart Helmet, he will be utilizing his skills to ensure proper power is provided to all necessary components of the system.

Blake Schereschel

Currently 23, he can be found spending a significant amount of his time programming and tinkering for fun. Much deliberation was had over the decision of which major to pursue: computer engineering or computer science. After taking a few introductory level classes in engineering, he knew that he wanted to be an engineer. To have a fuller understanding of the lowest level topics of computers only pushed his already deep excitement for programming even deeper. Currently, Blake has had internships at Lockheed Martin, Blizzard Entertainment, and Leidos Engineering, and hopes to utilize his skills to their potential on the Smart Helmet project. He hopes to benefit the team in any way he can, but primarily focusing on software architecture and low level systems design.

ACKNOWLEDGEMENT

We offer our thanks and gratitude to the professors who have agreed to review our final project that we have spent many hours working on and the professors that have helped and supported us along the way. With their help and guidance, we were able to see our design to fruition.

REFERENCES

[1] http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
[2] http://www.lidar-uk.com/how-lidar-works/
[3] http://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf
[4] http://www.microchip.com/wwwproducts/en/en024903
[5] https://learn.sparkfun.com/tutorials/voltage-dividers