Secure, Anti-Theft, Flexible, Engineered Taillight (SAFE T)

Jonathan Diaz Jay Kurczy Brandon Therrien Paul Wilson

***Abstract*—Safety is one of the utmost concerns when discussing vehicular travel, it is important that any mode of transport possesses a high aptitude for end-user protection. While motor vehicles do come equipped with lighting and an array of safety features the more modern they are, traditional manual-powered vehicles such as bicycles do not and this is a leading cause of accidents and fatalities for their users. It is in light of this that SAFE-T was developed, not only as a full lighting system but also as a theft deterrent. This group chose to create SAFE-T as a multi-faceted bicycle add-on due to one of the members being a bicycle owner and the aforementioned safety concern of bicyclists in general.**

***Index Terms*—LEDs, Microcontroller, RFID, PIR Sensor, Mobile App**

I. Introduction

Approximately 870000 people commute via bicycle, with such a large number of users it becomes difficult to ignore safety and security concerns [1]. Developed as a more modern lighting and anti-theft system SAFE-T encompasses app-controlled systems along with analog to provide an ideal user experience. The user may toggle any of the lights they wish along with adjusting their brightness. With the advent modern technology we have seen an increase in motor vehicle related collisions and fatalities which have also included bicyclists.

Now more than ever it is important to increase average bicyclists' safety and visibility to prevent their participation in motor vehicle collisions. Luckily, the aforementioned means that there is a greater availability of technologies and viable features which may mitigate that risk, modern technology is indeed a double-edged sword. The project is implemented on an ESP32 micro controller which is charged with communicating between the mobile app and the electrical systems including but not limited to the lighting and alarm.

A. Goals and Objectives / Requirements

As aforementioned, this team seeks to rectify some safety issues associated with bicycling. Our product, SAFE-T is intended to provide clear turn-signal indication along with head and tail lights to indicate rider position to automobile drivers, plus some extra anti-theft features.

Many of the goals deal with the most important aspects of the product, namely the size, ease of use, and protection from exterior elements. It is very vital that the product be protected from weather as bicyclists traverse all manner of terrain, protected from tampering as otherwise this would defeat its anti-theft capabilities, be smaller and more portable so it is easily affixable, and be smartphone-enabled for further ease of use, multi-platform availability is also important for market reach. The requirements delve into the specifics for the main subsystems namely battery functionality, environment proofing, and the lights' lumen output.

TABLE I
Goals and Objectives

Interface with MCU wirelessly
Remote Control of Lights
Slider to Adjust Light Brightness
Compatible with Android and iOS
Lightweight, easy to install
Easy to use, very accessible
Small and portable
Waterproof and environment proof
Tamper proof
Front and Tail Light
Turn Signal Indicators
Anti-theft Alarm with Motion Detection

TABLE II
Functional Hardware Requirements

Headlight needs to output at least 400-600 lumens
Tail light needs to ouput at least 50 lumens
Signal lights need to output at least 50 lumens
2000mAh Rechargeable Battery
Recharges within 2-4 hours
IPX6 Waterproof
Alarm at least 100dB
MCU Architecture 16-bit or greater

II. Background

SAFE-T is poised against a number of competitors, a simple search on Amazon reveals a plethora of available products that fulfill similar roles. A key distinction to make is that while many of these competitors' functionalities include a combination of head, tail, and signal lights, and an alarm, they do not possess all of them and require the purchase of multiple products and this is where SAFE-T truly shines as an all-encompassing product with a near unheard-of mobile app control. A large number of competing products are either just a headlight, just the tail light, the tail light and alarm, the tail light and turn signals, or the tail light, alarm, and signals.

Some competitors offer complete product packages but these are always composed of multiple of their products. Also, while many use a wireless controller, they do not offer smartphone connectivity.

A. Meilan X5 Smart Bike Tail Light

This competitor product offers a remotely controlled rechargeable tail light and turn signal. This product has 30 LEDs and is advertised as outputting up to 85 lumens while being visible from 150ft away. The battery is 2000mAh and the runtime is advertised as 5 - 10 hours.

B. Ascher Ultra Bright USB Rechargeable Bike Light Set

This competitor product is a combination set which includes both a head and tail light with four light modes. It boasts easy installation and two separate rechargeable batteries, a 2Ah one for the headlight and a 330mAh one for the tail light. Each has a single button to switch between lighting modes. The headlight is advertised as having a lumen output of 300 and a runtime of five hours at full brightness while the tail light's runtime is 9.5 hours at full brightness. This product weighs in at about 0.46lbs.

C. Intelligent Bike Tail Light

This competitor product offers a rechargeable tail light with an alarm and both are remotely controllable via a keyfob. The alarm is advertised as 120dB, the battery as charging in two to three hours and has a capacity 700mAh. The light has three modes and the product is feather-light at 0.194 lbs.

III. Ease of Use

SAFE-T is designed with the end-user in mind, it includes a number of physical buttons and switches that allow for an intuitive control of its systems along with a modern twist in the form a smartphone-paired app which offers the same array of controls. The widespread use of and user familiarity with smartphones means that the end-user experience with the product will be a simple and easy one such that any person regardless of the aforementioned should be able to quickly integrate it into their bicycling routine.

IV. System Components

The best way to explain this product is by describing all of the metaphorical cogs in its system. There is a small number of parts which comprise its main functionality that are described in greater detail in the following subsections. Though the parts number few, each one is critical to core system functionality and thus may not be discounted.

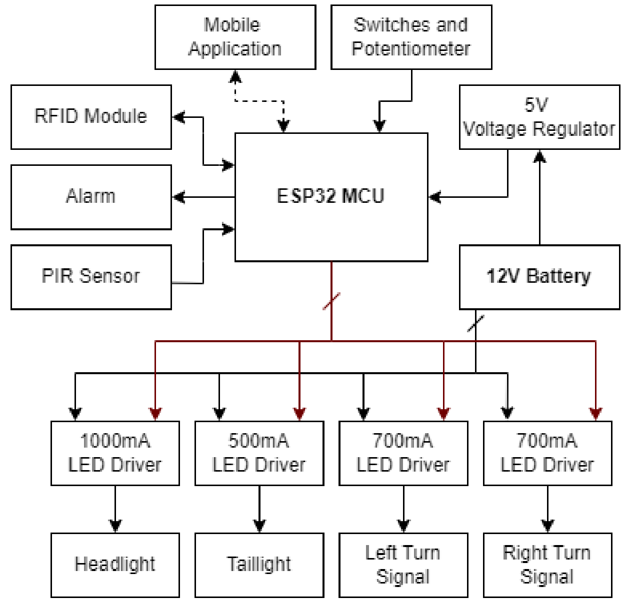

Fig. 1. Complete System Diagram

A. Radio-frequency Identification

The RFID module utilized for this project is the Mifare RC522 RF IC Card Sensor Module which is paired with an S50 card. This module is used to arm and disarm the alarm and acts as a sort of key. The module requires a voltage of 3.3V from the micro controller and uses an SPI interface to communicate. The connections made between the microcontroller and the RFID module required the following: SDA, MOSI, MISO, CLK, 3.3V, and GND.

Fig. 2. Mifare RC522 RF IC Card Sensor Module

B. Passive Infrared Sensor

The PIR Sensor is another important component in the anti-theft system as it allows the product to detect motion once the alarm has been armed via the RFID module. This sensor requires an input of 5V but may easily be secured anywhere on the product and integrates quite easily. The PIR sensor can be used to detect motion from humanoids up to 20 feet away. The delay and sensitivty can be adjusted. It also has a sensing range of about seven meters with a 120 degree cone detection.

The PIR sensor requires three connections to be made. These connections being the 5V, OUT, and GND. The 5V was connected to a 5V bus supplied by the 12V to 5V step down voltage regulator. The OUT connection was connected to a microcontroller pin. The out pin essentially is a simple high (3V) or low (0V) signal that operates when motion is detected. The microcontroller used this out signal as an input for the security system. The PIR sensor out signal once set to high would have the microcontroller output a signal to the audio amplifier. The audio amplifier then would provide power and the adjusted frequency to the alarm speaker to go off.

C. Battery

The battery utilized for this project is the ECI Power 12V 4Ah Lithium LiFePO4 which more than meets the specified requirement of a 2Ah battery. It is lightweight and easily integrates into the design, weighing a measly 1.7lbs it is lightweight and maintains the requirement of an easily portable device. Being a Lithium battery it is of course rechargeable, is very easy to charge due to it’s simplistic terminal access and overall design and has a net voltage of 12.8V.

Fig. 3. ECI 12V 4Ah Lithium LiFePO4 battery

The battery is connected to a power distribution board to connect the 12V to 5V voltage regulator and all four BuckPuck LED drivers. The battery can be recharged using the ExpertPower 12V 2A Smart Charger, which in terms of typical charging time can take from two hours to five hours.

Fig. 4. ExpertPower 12V 2A Smart Charger

In addition to the 12V battery the group also decided to opt into a battery monitor. The battery monitor is very useful to keep track of the battery life since without it there is no way of knowing whether the battery is about to die. As a bonus feature of the battery monitor it also comes with a temperature indicator for any worries of the electronics within the housing generating too much heat. The battery monitor itself is a simple LCD that displays battery capacity, voltage, and temperature in Fahrenheit. The monitor is also dust proof, waterproofed, and has input reverse connection protection for safety. It was cheap and was bought from Amazon for $12.

Fig. 5. 12V Battery Capacity Monitor

D. Step Down Converter

The step down converter the team decided to go with is called the DROK DCDC converter. It can handle voltages ranging from 4.5 to 40V. It can be adjusted to dial down voltage by using a mini screwdriver on an adjustable potentiometer. The voltage regulator would intake any voltage but for our purpose would realistically be anywhere from 1.5V or higher to approximately 12.8V which is the max voltage the battery can output. The voltage would then be stepped down to roughly 5V. The 5V is used to power the microcontroller board PCB, the PIR sensor, and the audio amplifier.

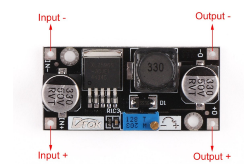

Fig. 6. DROK Step Down Voltage Regulator Module

E. Audio Amplifier

The speaker takes 3W of power and has an internal resistance of 4 Ω. To meet this specification, an audio amplifier was used. The audio amplifier takes various connections as input and they were connected to the 5V source and the output from the microcontroller that generated the required waveform for creating the alarm sound. The output from the audio amplifier is an analog wave that meets the specifications of the speaker. The audio amplifier we used contains the LM386 amplifier chip–a low voltage one–has a power range of 5-12V input that was easily accessed within the closed system we created. As mentioned before, we used 5V as it meets the specification and we could easily access it from the voltage regulator we used, which is connected to the 12V battery.

F. Alarm

The speaker utilized for the alarm on this product is a mono enclosed speaker which has a two terminal connection, it is compact and fit easily into the product. This speaker is a 3W 4 Ω speaker which is low power enough to be supported by our power subsystem and should be loud enough to meet our specified requirement of a 100dB alarm.

Fig. 7. Mono Enclosed Speaker

G. Lighting

The lighting portion of the product is the most important subsystem because it is what would be the face of the product, it is the main advertised feature. The requirement for this subsystem was that it be at least 400 - 600 lumens. There are four lights that are composed of multiple smaller LEDs. The headlight has its own heatsink while the tail and signal lights share one. The light subsystem is the main power draw and determines the battery life of the product. One of the key benefits of using LEDs is the high light output with a minimal footprint, although the utilization of heatsinks was necessary due to the extreme heat generated by them.

TABLE III
LIGHT SPECIFICATIONS

Purpose	***Power***	***Voltage***	***Current***	***Lumen Output***
Headlight	10W	9 - 11V	900 - 1100	900 lm - 1000 lm
Taillights	3W	2.0 - 2.2V	400 - 500mA	60 - 70 lm
Signal Lights	3W	2.0 - 2.2V	700mA	60 - 70 lm

The LED drivers selected for each lighting component will all be the LuxDrive BuckPuck LED drivers. The size is super compact, efficient, and easy to use. Since there are a total of four components for the LED system which include the headlight, tail lights, left turning signal, and right turning signal there will be four BuckPuck LED drivers. Each LED driver was specifically picked to output electrical characteristics of the LEDs such that the headlight requires 9 - 11V at 900ma - 1100mA meaning the compatible LED driver for it can be easily selected by picking a BuckPuck that has a constant current of the LED. We selected a 1000mA BuckPuck for the headlight, a 500mA BuckPuck for the taillight, and lastly for both turning signals we chose a 700mA BuckPuck. Each BuckPuck LED driver requires a minimum of 5V to power and can intake up to 32V. The control dimming works such that there is a CTRL pin and a REF pin, the built-in regulated 5V REF pin provides output to power logic circuitry or a microprocessor meaning it eliminates the need for an additional power supply. The CTRL pin is simply the input to provide external control.

Fig. 8. BuckPuck LED Driver

For the most part, the LED system worked such that a read input signal to the microcontroller will output another signal to the base of a transistor to control the LEDs via driver. A BJT transistor was used for controlling because the ESP32-S2-WROVER cannot output 5V. For taillight and turning signal control a momentary push button was used as a read signal to the microcontroller in which the microcontroller will output high (3.3V) or low (0V) to the base of a BJT transistor. Whereas the headlight control was different since the input is a potentiometer while the rest of the circuit logic remained the same.

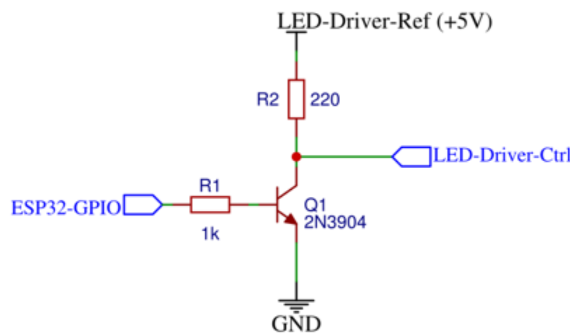

Fig. 9. Schema for connecting the MCU and LED Drivers

H. Other

A number of switches were used to create the physical controls that are affixed to the user's bicycle, these switches include push buttons along with a potentiometer. The potentiometer is the only control for the headlight, it controls the brightness and when turned completely in the counter-clockwise direction the headlight is deactivated. The push buttons toggle the signal and tail light. There are four LED drivers, one for each of the turn signals, one for the headlight, and one for the taillight. The drivers are important as they convert the battery voltage of 12V to that which is required by each respective headlight as detailed in III. Lastly a voltage regulator is required to output the voltage specified for the micro controller which is 3.3V.

I. Product Housing

The product housing is usually overlooked as it seems so simple but the truth is rather the opposite. As previously stated, the housing plays a passive role in the anti-theft subsystem as it must be tamper-proof, if easily opened the alarm system may be quickly disarmed which is a major security flaw. Further, the resistant casing insulates the vital circuitry that powers the product from rain, snow, and dirt. The controls are affixed to either side of the handlebar and are adjacent to the headlight as depicted in 10. The wiring is designed to be as minimal as possible, with the headlight and control connections running along the bicycle's frame as depicted in Fig. 11.

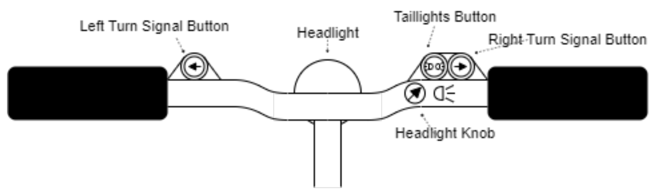

Fig. 10. Handlebar Layout

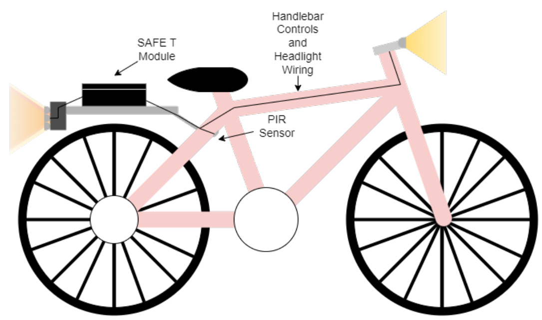

Fig. 11. Bike Wiring

V. Microcontroller Software

The microcontroller software uses the NodeMCU firmware which was originally made for the open-source NodeMCU boards but work with the ESP32 family of chipsets, in the same vein as Arduinos, except NodeMCU is specific to the Espressif chips.

One of the key reasons for choosing NodeMCU over the ESP-IDF SDK–which is the official SDK–is because of the lack of build times and compilations. NodeMCU uses an "interactive" model, in which the application code runs in a extremely small virtual machine–specifically the Lua virtual machine designed to fit the limitations of the ESP chips. This means that it allows us to effectively hotswap the code we wrote to easily debug at the cost of slight performance hit. This has reduced the microcontroller testing times significantly as we were able to iron out the minor mistakes in code we made quickly. The firmware also adds a bunch of libraries although they have to be decided upon before compiling the firmware to minimize the firmware size and to avoid bloating of the ROM table partitions the ESP chips use to execute instructions. The firmware was flashed on to the microcontroller using the esptool.py software provided by the manufacturer.

The code consists of different sections. We polled the data from the potentiometer with the ADC utilizing the library of the firmware and used this information and the data from the mobile application to output a specific voltage from the microcontroller that is required by the LED driver to control the intensity of the headlights. The mobile application can only change the brightness when the knob on the bicycle is set to off. The microcontroller outputs this specific voltage using the DAC, which is also accessed with the help of a library available in the firmware. This data is polled in the main loop written that does various other things.

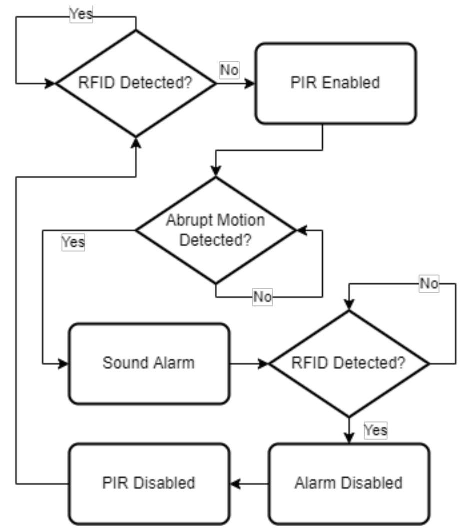

Fig. 12. Security Check Loop

The buttons for the turn signals and the taillights on the other hand only require simple triggers from the button which is why interrupts were used to get this information. Since they behave effectively like toggling buttons, the limitation imposed above need not be done here.

The logic for the turn signals involve an XOR-like logic when they are turned on, i.e. when one of them is on, and when the other one is turned on, the one that is on is turned off then the other one is turned on. To do so, internal states were used to keep track of what is on and what is off. The usage of states also helps when it is integrated with the mobile application. The timing for the turn signals to toggle their states to create the flashing effect is tracked by the tmr module of the firmware.

The security system consists of controlling the RFID module, which uses the SPI protocol, the speaker for which we use a PWM generator for the alarm sound and the PIR sensor which can be used with interrupts. The system is armed when the NFC tag specific to the device is removed and when the motion detector detects motion while it is armed, the speaker goes off, producing an alarm sound. The logic for this security loop works as shown Fig. 12. The mobile application provides the same functionality due to this reason. The main loop mentioned before also polls the RFID data from the module and matches it with the desired tag.

The mobile application integration is achieved by a publisher model, where the MCU gives out information only when the mobile application requests the MCU for the requests made, and only executes things when these subscriptions are made by the application. This allows for various things like seeing if the module is active, and also offloads the workload of keeping the application up to date as the responsibility of the application.

VI. Mobile Application

The mobile application is another important aspect of this product as it contributes heavily to the modernization of the product as a whole and promotes ease of use by allowing the user to utilize an intuitive control screen that mirrors all physical controls. An important aspect of this is that the mobile application is available on the most widely adopted operating systems which are iOS and Android.

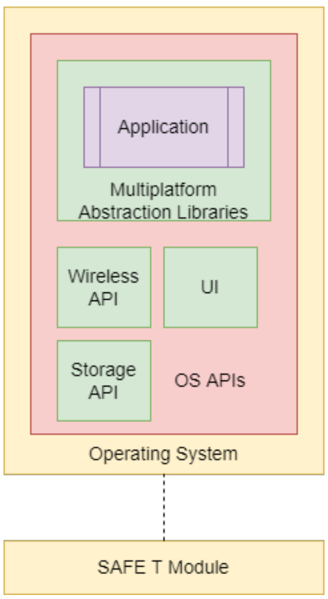

Fig. 13. Software Block Diagram

In order to support these operating systems the mobile application was developed with React Native which is a modern framework that runs natively on both. The framework uses the Node.js platform which acts as the JavaScript engine for compilation and execution of the application package. The team used TypeScript, a superset of JavaScript, to program the application. The application mainly sends the commands as requests to the microcontroller and the microcontroller executes the corresponding commands. A third-party library was used to make these requests to the microcontroller. Since we were working with React Native, we took advantage of the JavaScript promises for error handling as it cannot be guaranteed that the requests made are always going to be fulfilled by the microcontroller. A third-party library for the Native sliders in the application was used to mimic the functionality of the knob on the handle bar. Ultimately, our goal was to keep the application completely optional and the design approach we took has helped us achieve this goal.

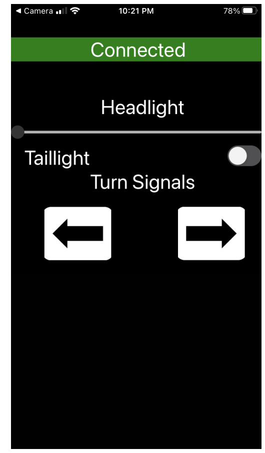

Fig. 14. Mobile Application UI

VII. Schematics/PCB or Board Design

For our project system to work properly we have to have a microcontroller. Thus, our essential PCB is the microcontroller board where it will hold the ESP32-S2-WROVER module, a voltage regulator, USB-to-SERIAL converter, USB connector, and headers to allow access to all pins to the ESP32. The PCB design was straightforward and the only problem we had with it was the impact of the global chip shortage. This affected some component selections such as the voltage regulator. Most voltage regulators, especially from well known brands like Texas Instruments, were out of stock or the lead time was too far ahead. Due to this challenge the team had to make sure to pick available alternative components for the PCB design. Research was done to properly ensure time wasn't wasted since PCB manufacturing typically takes more than a week when ordering from overseas. All essential components placed on the PCB will be discussed to further explain their purpose. We used JLCPCB, a Shanghai-based PCB manufacturer, to print our PCBs.

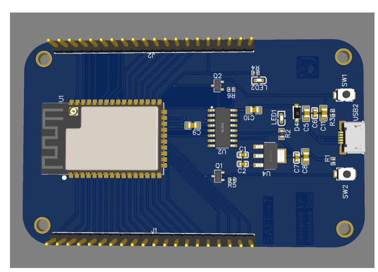

Fig. 15. SAFE T PCB

A. Microcontroller

The microcontroller utilized for this project is the ESP32 which is not only a very commonly manufactured micro controller but also just as commonly used in a variety of products and designs. The ESP32 has a 32-bit LX7 Microprocessor operating up to the 240 MHz range, it has 128 KB ROM, 320 KB SRAM, supports GPIO, SPI, UART, I2C, I2S, LED PWM,

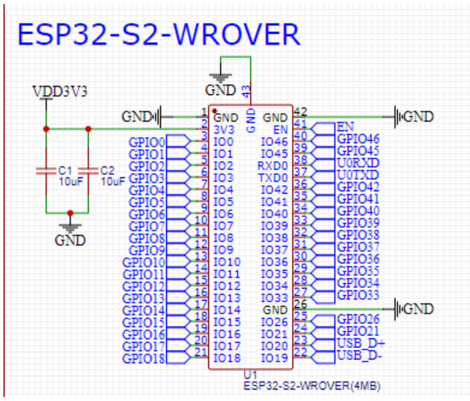

Fig. 16. ESP32-S2-WROVER Schematic

and many other features [2]. It also has an operating voltage supply of 3.0 $\sim$ 3.6V. It's also incredibly important to note that the main reason this microprocessor was selected out of many other options was primarily due to having an integrated WIFI module as well as Bluetooth Low-Energy capability.

This component is the brains of our product and is extremely vital to its operation. This micro controller is perfect for our project as it supports the interfaces we need for the RFID and wireless capability while having a very minimal form factor.

B. Power Supply Design

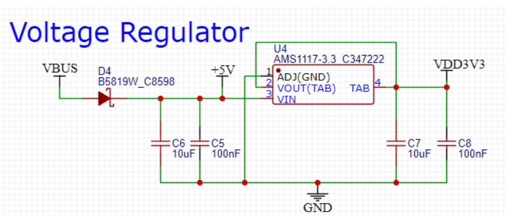

Fig. 17. Voltage Regulator Schematic

The power design is straightforward and simple. The input voltage, output voltage, output current, as well as small footprint is what went into consideration for the selection of the voltage regulator. Based on the ESP32-S2-WROVER datasheet it requires a regulated voltage of 3.3V to the power supply pins with a current of 0.5A. Since there already exists a step down converter module that accomplishes 12V to 5V all that needs to be done is to have a voltage regulator for the PCB that can intake 5V and step down to 3.3V. With that in mind, the AMS1117-3.3 is a great choice in that it has a small footprint that requires no complicated analog circuitry and is a simple 5V to 3.3V step down linear voltage regulator that can output up to 1A. Typical applications of the microcontroller will use 0.5A at most further making this voltage regulator a great choice for the PCB design. The capacitor values from the input and output are picked based on what the AMS1117-3.3 specifies which are just 10uF and 100nF in both sides. Lastly, The voltage regulator input has a schottky diode for safety precaution.

C. Serial-USB Converter

For the USB to serial chip the team chose the CH340. The CH340 is a USB bus converter chip that is compatible with serial application programs in the Windows operating system through emulating the standard serial interface. It can tolerate 5V and 3.3V source voltage. It can also support a communication baud rate up to 2Mbps. The connections for the chip for our design required that the RX, TX, D-, D+, 5V, DTR, and RTS are connected. Both DTR and RTS were connected to transistors to create a simple logic for auto programming. For example if the EN (Reset) and GPIO0 buttons are high then DTR and RTS would result in both being high.

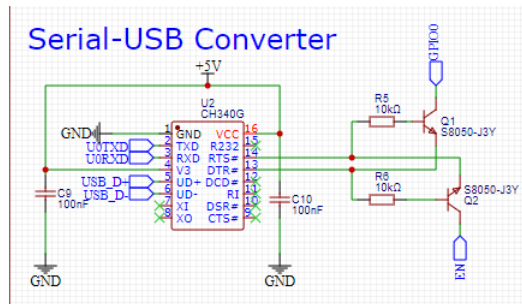

Fig. 18. USB-Serial Schematic

D. USB Connector

A debugger and programming connector is needed to be able to debug and program the ESP32-S2-WROVER. The ESP32-S2-WROVER has built in JTAG circuitry to allow a USB cable to connect to the D+ and D- pins. The micro USB type connector will simply be connected to the D+, D-, external voltage supply, and ground. The idea of this USB port is that a USB cable can directly connect to the ESP32-S2-WROVER to debug and program it, which is essential. The team did not need to be specific as to what model we chose for the USB connector since most work the same way. The only concern was conserving space so a small footprint USB connector was chosen.

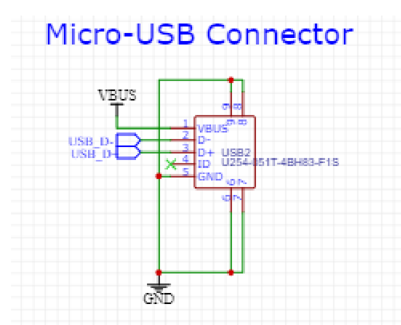

Fig. 19. Micro USB Connector Schematic

VIII. Cost Analysis

TABLE IV
BUDGET AND FINANCING

Item	Quantity	Price
Smartphone	1	$0.00 (owned)
Bicycle	1	$0.00 (owned)
3D Printed Material	X cubic inches	$15.00 per cubic inch
Buttons (with wire)	5	$7.99
MCU	1	$10.00
PCBs	x layers, x sq. inches	$20
Main Battery	1	$50
Alarm	1	$5
Headlights	1	$15
Brake Signals	10	$10
Turn Signal Lights	10	$10
RFID	1	$20
Misc. Hardware	x	$20
Sensor	2	$10
LED Drivers	4	$50

IX. Conclusion

The purpose of this project was to create a product that integrates modern technologies with solutions to a well-aged increasingly apparent problem, bicyclist safety. Although the world is moving away from manual-powered vehicles it is at the same time rekindling its use of green transportation. Now more than ever, especially due to the release of motor vehicles capable of achieving speeds at a faster rate, safety concerns for bicyclists are at an all time high. SAFE-T seeks to rectify these concerns by providing an all-in-one bicycle add-on providing lighting and anti-theft features. The product performed at least to expectation in every regard and in some places surpassed them.

The authors felt compelled to take on this challenge not only because of the personal stake but also because of the large market which while served, could be served better. This product hopes to serve as an example for accessibility and modernization for an aged and established market.

Acknowledgments

The authors wish to personally thank professors Dr. Samuel Richie and Dr. Lei Wei for their support and assistance through these tough times and throughout our project selection, design, and implementation. Further acknowledgment goes to the ECE department, especially our faculty reviewers.

References

[1] Bureau, U.S. Census. “May 17 Is National Bike to Work Day.” Census.gov, 8 Oct. 2021, https://www.census.gov/library/stories/2019/05/younger-workers-in-cities-more-likely-to-bike-to-work.html.

[2] Espressif Systems, ”ESP32-S2-WROVER” ESP32-S2-WROVER datasheet, Mar. 2010 [Revised Mar. 2022].

Biographies

Jonathan Diaz is a senior graduating in May 2022 with a bachelor’s degree in Electrical Engineering. He accepted an offer to work for Boeing at Washington state as an Electrical Cabin and Network System Design Engineer after graduating from UCF. He is also interested in pursuing a master’s in Electrical Engineering.

Jay Kurczy is a UCF senior graduating with a degree in electrical engineering. He is planning on pursuing a career in his field after graduation, with an interest in renewable and non-renewable power, as well as integrated circuit technology. He is considering pursuing a master’s degree in the near future.

Brandon Therrien is a Computer Engineering senior at UCF. He is planning to find work in software engineering after graduating and would like to find work where he’s making a difference in the world. He is considering a master’s in Computer Engineering or Computer Science.

Paul Wilson is a computer engineering senior at the University of Central Florida pursing a bachelor’s degree. He has worked with computers since a young age and intends to continue working with them. He has accepted a position at Deloitte, Lake Mary as a Solution Analyst and intends to start right after graduation. He also plans to obtain a master’s degree in computer engineering in the foreseeable future, with a focus on computer graphics and image processing.