Wirelessly Connected Laser Shooting Gallery

Thomas Stoeckert, Rachel Goodman, Anna Malaj, Jamauri Balzourt

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — In this paper, we propose a new design for a laser shooting gallery that is meant to improve upon traditional gallery systems. The purpose of this project is to create a wireless laser gallery shooting system that is reconfigurable, expandable, and convenient to use, with the goal of creating a better user experience than traditional systems, as well as a fun portable game for people of all ages to enjoy.**

***Index Terms* — Battery management systems, laser diodes, mesh networks, microcontrollers, phototransistors, three-dimensional printing, wireless devices.**

I. Introduction

Typical proprietary laser shooting galleries have limited reconfigurability outside of their core structure. Most existing systems rely on fixed, hard-wired points that can typically only be set up in expensive, permanent installations, limiting both configuration possibilities and market reach. However, with the recent explosion in low-power and low-cost wireless computing, we believe that a solution involving multiple discrete, self-contained, low-cost wireless targets alongside a custom-designed "gun" controller and software suite would allow for increased user convenience, more interesting user experiences, and additional features that traditional systems are unable to achieve.

II. System Overview

From a top-level view, our system consists of a main controller and at least one, but possibly multiple, targets. When the controller's trigger is pulled, it emits a laser beam that can be sensed by phototransistors on the targets, allowing the user to play an arcade-style shooting game with four different game mode options. To register hits and exchange state information, the controller and the targets communicate wirelessly with each other over a mesh network. This allows the system to be portable and ensures that target locations can be reconfigured. The system is also designed to be expandable, as users can effectively connect as many or as few targets as they desire.

In designing our system, our goal was to provide users with an enhanced gameplay experience. With this goal in mind, we developed a series of requirements for our system that will ensure gameplay is as convenient as possible. These requirements, which define features like battery life, response time, and weight, are listed in Table 1.

TABLE 1
SYSTEM REQUIREMENTS

Number	Purpose	Description	Requirement
1.	Performance	The controller should have a high active uptime	≥ 3 hours
2.	Performance	Each target should have a high active uptime	≥ 6 hours
3.	Performance	The time between pulling the trigger and the target's visual response should be small	≤ 0.1 seconds
4.	Performance	The system should have a quick startup time	≤ 25 seconds
5.	Performance	Target pairing should be completed in minimal clicks	≤ 4 clicks
6.	Energy	The controller should be energy efficient, consuming a small amount of current	≤ 3000 mA/4 hours
7.	Usability	The controller should achieve a low maximum weight	≤ 10 lbs
8.	Usability	The target should achieve a low maximum weight	≤ 5 lbs
9.	Usability	The system should be in "ready to play" state within a short period after startup	≤ 2 minutes

III. Hardware Overview

This section gives an overview of the hardware design for this project.

A. The "Gun" Controller

The main controller for the system is a single device, styled after a traditional laser tag gun or pistol. Its core functionality is two-fold; it serves as the master controller for the entire game system, organizing and managing game data and device behavior for the various targets in the system, and it also serves as the main user interface for the system, with a traditional trigger input activating its laser diode. In addition, a simple status screen on the controller is used to inform the user about the game state or device status. Other sensory functionality is also included in the form of aural feedback through an onboard speaker.

The ESP32, a relatively modest microcontroller with built-in wireless capabilities, is used for this role – while managing game and device statuses over a wireless network might be complicated, it does not require a massive amount of computing power. It is mostly event-based, and as such is designed to be energy and computationally efficient.

All these components are powered via an onboard battery with a built-in BMS chip to manage its status, as well as to provide some utilities for charging and discharging it. The battery is also able to be easily removed from the device for recharging purposes.

B. Multiple Target Devices

Targets are a critical component of any shooting gallery setup. The core functionality of the targets in this system is to communicate with the main controller and wirelessly signal a successful hit whenever its sensor receives a hit from the laser diode. This functionality is expanded with a lighting display in the form of an LED array and a speaker in order to provide the user with feedback indicating that a hit has been registered, or to convey information about the current game state. These are all managed by an on-board ESP32 microcontroller.

The ESP32 microcontroller also requires wireless capabilities to signal to the main controller that a hit has been registered. While it is not managing game state, but rather acting as a follower to the controller device, it still might require some more advanced computing resources in order to drive the onboard display or optional audio component.

Much like the controller, each target device contains a battery with a built-in battery management circuit and is easily removable for charging.

C. Component Considerations and Selections

In order to simplify the process of designing two unique hardware systems (controller and target) as much as possible, we made component selections for both devices in parallel. Both devices make use of the same model of microcontroller, battery, power system, and audio system. The only parts chosen separately were parts unique to each device, such as the laser diode for the controller and the LED strip lights for the target.

The first component selected was the microcontroller. For this selection, we prioritized the features of wireless communication ability, low power consumption, sufficient memory and storage, and software development support. With all of these considerations in mind, we selected the ESP32 microcontroller, WROOM package.

The next component selected was the battery. The goal for the battery was to maximize active playing time for the system before it needs to be recharged. We used an Efest 18650 4S1P 14.8V 2600mAh battery pack, which was provided to us for free by Smart Charging Technologies LLC. This battery pack has a built-in BMS chip, so no additional battery management systems were needed.

Once the batteries had been selected, corresponding voltage regulators could be chosen. All components on the controller required only a 3.3V input, and the target had components requiring both 3.3V and 5V inputs. To power the 3.3V components, we selected the UA78M33CDCYR linear voltage regulator from Texas Instruments, which has a fixed output of 3.3V, and up to 500mA. For the 5V components, we selected the MC7805CDTRKG linear voltage regulator from onsemi that has a fixed output of 5V, and up to 1A.

For the audio system on both devices, we selected the MAX98357A audio amplifier chip, and then selected MakerHawk 4 Ohm 3 Watt speakers to go with it.

As for selecting the components unique to each device, the laser diode for the controller and the corresponding sensor for the target required the most testing to select. A number of different laser diodes were tested, and a number of different types and models of photosensors were tested. In the end, we chose the laser diode and photosensor that worked best together. The laser diode chosen was a 5mW 650nm Red laser from Adafruit, and the photosensor chosen was a 630nm radial phototransistor from Everlight Electronics.

Lastly, any other unique components such as the OLED display for the controller and the LED strip lighting for the target were chosen easily from Adafruit to suit the exact needs of the project, and therefore did not require extensive testing.

IV. SOFTWARE OVERVIEW

This section overviews the software involved in this project.

A. Design and Implementation Approach

The target and the controller each have their own software package to manage their functionality. To simplify development, we created a core package for functionality that is shared between the target and the controller. Platform-specific code was written using the core package as a base.

The software in our project can further be broken down into two categories: Hardware Managers and Logic Modules. Hardware Managers include code for managing peripheral devices like the speakers, LEDs, or the display, among other devices. Logic Modules differ between both platforms. They are used to define how each platform behaves depending on the state that the platform is in. We make use of callbacks and fixed frequency update loops to perform the required logic.

For our development environment, we make use of platformIO [1] and Visual Studio Code [2]. platformIO is a development platform for embedded C and C++ that can be downloaded directly into the Visual Studio Code IDE. It provides easy access to tools and support for the ESP32. Libraries can also be searched and easily integrated using this tool. To track our development, our group used Git, specifically through GitHub [3]. We organized our code into a main branch for the core functionality, and individual branches for platform-specific code. We further documented our code using the Markdown tools directly available on GitHub.

B. Architecture Overview

Our system architecture is designed to be event-driven, where tasks will be executed using a real-time scheduler. From startup, both the controller and the targets will transition between different states depending on user input and various events that occur in the game. The controller begins in the initialization state, where the system prepares its hardware modules for gameplay. Once the device is ready, the player can pair any number of wireless targets to their controller for use during gameplay. From there, the player chooses a game mode and begins to play. The controller keeps track of when the game is complete, and the target lights ignite and extinguish as they are hit. Once a game is complete, the player sees the results on the controller. The controller's possible states and transitions are shown in Fig. 1.

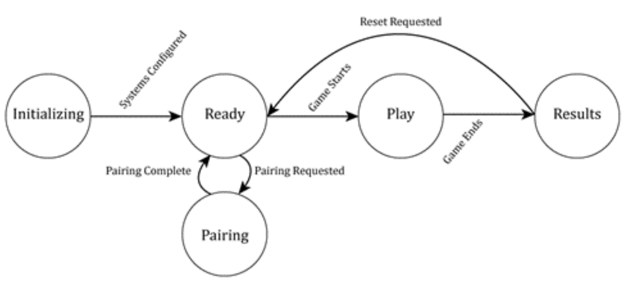

Fig. 1. State diagram for the main controller.

The diagram for the target is similar, as shown in Fig. 2, but excludes the results state, as that state is only needed by the controller.

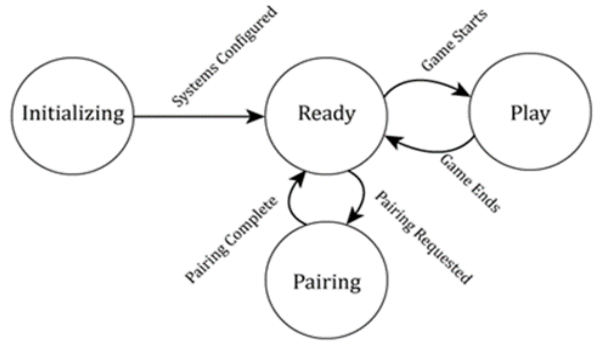

Fig. 2. State diagram for each target.

Transitions between states can be triggered by user input, which includes a left button tap, a right button tap, or a trigger pull. Outside factors, such as a game timing out, may also trigger a state transition. The possible events that the controller may encounter are summarized in Table 2. The events that the targets may encounter are summarized in Table 3.

TABLE 2
CONTROLLER EVENTS

Event	Channel	In/Out	Relevant States
INPUT_LEFT	Hardware	Incoming	All
INPUT_RIGHT	Hardware	Incoming	All
INPUT_TRIGGER	Hardware	Incoming	All
PAIR_REQUEST	Wireless	Incoming	Pairing
PAIR_REJECT	Wireless	Outgoing	Pairing
PAIR_ACCEPT	Wireless	Outgoing	Pairing
PAIR_COMPLETE	Wireless	Outgoing	Pairing
TIMER_ELAPSED	Software	Outgoing	Play
TARGET_HIT	Wireless	Incoming	Play
TARGET_IGNITE	Wireless	Outgoing	Play
TARGET_EXTINGUISH	Wireless	Outgoing	Play
GAME_START	Wireless	Outgoing	Ready, Play
GAME_END	Wireless	Outgoing	Play

TABLE 3
TARGET EVENTS

Event	Channel	In/Out	Relevant States
INPUT_PAIRING	Hardware	Incoming	Ready
INPUT_LASER	Hardware	Incoming	Pairing, Play
PAIR_REQUEST	Wireless	Outgoing	Pairing
PAIR_REJECT	Wireless	Incoming	Pairing
PAIR_ACCEPT	Wireless	Incoming	Pairing
PAIR_COMPLETE	Wireless	Incoming	Pairing
TARGET_HIT	Wireless	Outgoing	Play
TARGET_IGNITE	Wireless	Incoming	Play
TARGET_EXTINGUISH	Wireless	Incoming	Play
GAME_START	Wireless	Incoming	Play
GAME_END	Wireless	Incoming	Play

C. Networking and Scheduling Considerations

To simplify the networking aspect of this project, we make use of an ESP32-compatible library called painlessMesh [4]. We use painlessMesh to create a dynamic network of nodes, which allows us to wirelessly send messages between two or more devices. Each device has a 32-bit unsigned integer called a chipID to identify its address. When a user pairs a controller to a target, each device learns the other's chipID. Once learned, the devices can send messages, formatted as a string, by addressing the correct chipID. When a message is sent, it will be in the following format:

<timestamp>;<tag>|<data> (1)

The timestamp is the time the message was created, the tag is a string describing what the message contains, and the data is the information that the platform wished to send. Using this custom protocol, the controller and target can send each other event information in real time.

For scheduling considerations, we use another library called TaskScheduler [5]. TaskScheduler is a FreeRTOS-based library for scheduling real-time tasks and ensuring everything is completed by its deadline.

D. Lighting Effects

Lighting effects are determined by the controller and sent to the targets during gameplay. When the controller determines that a target's lighting should change, it will send a message containing an effect code directly to that target. This effect code is sent in a single string using the following format:

LightingPattern|Loop|Clear|StartTime|Timeout
|Frequency|PrimaryColor|SecondaryColor (2)

Lighting Pattern is a hex-coded single character describing the pattern the lights should make. Options include a static color, blinking of every light in the strip, or a marching blinking pattern. Loop is a Boolean for determining if the pattern should be played indefinitely, ignoring the given Timeout value. Clear is a Boolean for determining if the target's light strip should be cleared or remain as it was when the current lighting pattern has been stopped. StartTime is a 32-bit unsigned integer indicating the time in milliseconds that the effect began. Timeout is an unsigned long representing the amount of time the pattern should run. Frequency is an unsigned long indicating the frequency that the effect should play. Primary and Secondary Color are both 32-bit unsigned integers containing the RGB values of the colors being displayed on the light strip.

E. Game Modes

Our system includes four game modes for the user to play: Time Trial, Whack-A-Mole, Horde, and One-Shot. Depending on the chosen game mode, the controller will send specific lighting and timing information to the targets.

Time Trial allows the user one minute to shoot as many targets as possible. At the start of the game, all available targets are lit and active. When a target is hit, it is deactivated. The user must shoot every active target before the targets are ignited again.

In Whack-A-Mole, targets are randomly activated, and deactivate after one second. The user has one minute to shoot as many targets as possible before they deactivate.

Horde is similar to Whack-A-Mole, except there is no time limit. The game ends when a user has failed to hit an active target before it has deactivated.

In One-Shot, all targets are active. The user has an unlimited amount of time, but only three attempts to strike a target. If the user succeeds, the number of attempts resets. If the user misses all three attempts, the game is over.

V. 3D Printed Device Enclosures

Each device (target and controller) requires an enclosure to protect its electronics from environmental factors such as dirt, dust, and any outside damage that may cause accidental shorting of circuit components. This physical enclosure also gives each device its form and structure, providing the user something to either comfortably hold or to hang on a wall.

In most traditional consumer electronics, the body of a device tends to be composed of plastic or metal, either machined or injection-molded, in order to construct a regular, structurally sound, and economically produced component of the device. However, the costs and challenges associated with a traditionally manufactured body are outside both the skillset and budget of our project, leaving us with a common manufacturing technique for prototyping physical components: additive manufacturing, also known as 3D printing. Additive manufacturing is used frequently for developing prototypes, and is a cheap, easy, and straightforward process.

Because the controller and target devices have drastically different form factors from one another, each enclosure was designed independently. Both enclosures, however, were designed with the ability to be taken apart easily in order to access all critical components, allowing us to make any changes necessary, and allowing the user to access the battery for charging. This section provides an overview of the 3D printed enclosure designs for both the controller and the targets.

A. Controller Enclosure

The design of the controller's enclosure proved to be the more challenging of the two, not only requiring a smaller form factor for more components, but also consideration for ergonomics. The controller is designed to rest in the palm of a user's hand, much like a traditional weapon, but also contains some larger, heavy components (primarily the battery) which can affect the usability of the "gun" if the "gun" is poorly balanced, too heavy, or just cumbersome to use.

The notable hardware components of this device that had to be worked into the shape of the controller enclosure include the laser diode, speaker, OLED screen, control buttons, and the trigger. The laser diode was the easiest to place – it points out of the front of the controller, laid along the center line of the controller's barrel. The speaker is large and thick and needs to be pointed towards the outside of the device as much as possible in order to get the best effect. The OLED screen had to be easy for a user to see, and the control buttons were placed within reach of the position where the user will rest their hands. The trigger was placed in its traditional position, within reach of the user's pointer finger.

Designing an enclosure to rest comfortably in the user's hand was an iterative process, one where we made repeated small adjustments to get the weight, balance, and grip of the device as correct as possible. Fig. 3 shows the 3D model of the controller enclosure design, created in Fusion 360.

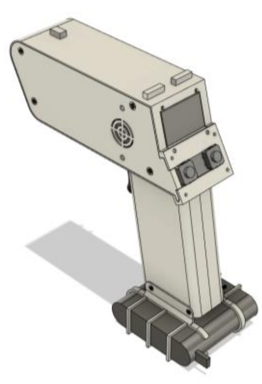

Fig. 3. Design of the controller enclosure.

B. Target Enclosure

The Target has the simpler enclosure of the two devices, and was therefore prototyped and printed first, which provided the team an opportunity to bolster our experience with case design and 3D Printing.

The Target enclosure must, at a minimum, enclose the custom PCB that holds the critical components of the system, as well as the system's battery.

Much like with the controller, component placement is the most critical aspect of the target enclosure design. The light-sensing phototransistors are placed in the center of the target, optically exposed to the environment to allow for a laser to hit them without obstructions. The speaker chosen for this system, too, is facing forward out of the front of the target, with a speaker grille to allow air to carry soundwaves from the target to the player. The array of WS2182 lights is also facing the user in order to signal to the user the state of the device, show lighting effects, and serve as indicators for where to aim.

Power and Pairing controls are located on the rear of the device, as they are elements which will not be used during active play sessions.

For maintenance access, the rear panel of the device is able to be removed entirely and is affixed to the front using a small count of common screws. An additional hardware requirement is a standard wall-mount hanging hole, like those found commonly on picture frames, in order to facilitate the device's mounting onto walls and other surfaces. Fig. 4 shows the 3D model of the target enclosure design, created in Fusion 360.

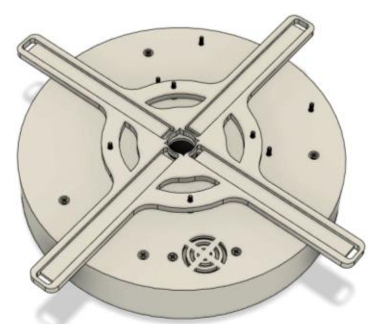

Fig. 4. Design of the target enclosure.

VI. Significant PCB Design

One of the major requirements for this project was a significant PCB design, which was defined as a PCB designed by the group containing its own onboard microcontroller and power components, as well as any sensors and other peripheral input/output components.

We did experience a few issues with our design along the way, and our PCBs were finalized in their third revision. In their first revision, the chosen ESP32 package was too small to work with, so we selected a larger one for the second revision. In the second revision, the switching voltage regulator design that we had originally intended to use proved to be too unreliable, so we placed all switching voltage regulators in parallel with linear voltage regulators for the third revision, intending to use whichever was more reliable in testing, and leave the other option off the board as an open circuit. This took up slightly more space on the PCB but ensured that our power design would work reliably.

In the end, the third revision of both boards performed as expected in testing. For power, we decided to use the linear voltage regulators (which were more reliable) and left the switching regulators off the board.

With the third revisions assembled and working, our goal of significant PCB design on both our controller and target PCBs was accomplished. The design of both PCBs will be detailed in this section.

A. Controller PCB Design

The controller PCB was designed with size and shape in mind. In order to fit into the “barrel” of its handgun-shaped enclosure, it was designed to be as small, long, and thin as possible.

The end of the controller PCB that is furthest away from the user contains the two 3.3V linear voltage regulators and their associated capacitors, in order to keep the warmest components as far from the user as possible. The screw terminals that connect to the laser diode, speaker, and power switch also lie on this end of the board. The audio amplifier chip and its associated resistors and capacitors are located conveniently in between the output to the speaker and the microcontroller. The end of the board closest to the user contains the ESP32 microcontroller, the programming buttons and pins, and connections to the trigger, OLED display, and battery. All screw terminals and other connectors are situated as close as possible to the location of the component they connect to within the enclosure in order to simplify the assembly process and make the most efficient use of connecting wires. Fig. 5 shows a picture of the assembled final controller PCB.

Fig. 5. Assembled controller PCB.

B. Target PCB Design

The target PCB was much easier to lay out, as the target enclosure is physically much larger than the controller. Therefore, the target PCB could be any size and shape needed to fit all components, and the components could be laid out almost anywhere on the board..

The target contains many of the same major components as the controller, including the ESP32 microcontroller, the 3.3V linear regulators, the audio amplifier chip, and all necessary programming pins and buttons. Items unique to the target include the five phototransistors, the 5V linear regulator, the 3-position screw terminal for connecting to the LED strip, and the onboard pairing button and power switch (located on the back of the board for easy access once assembled in the enclosure). Fig. 6 shows a picture of the assembled final target PCB.

Fig. 6. Assembled target PCB.

VII. Testing and Results

Our finished system is pictured in Fig. 7, including the gun controller and one example target. We ran a series of tests on this completed system to ensure all requirements listed in Table 1 were met.

Fig. 7. Completed system prototype.

To measure power draw, we plugged each device into a 16V power supply, then continuously measured the current draw for each device over two hours. The devices were tested in parallel, with a program on the controller pulsing the laser every second. The target would send a message back to the controller. This ensured that the power impact of gameplay, including network messages and lighting changes, was accounted for. We measured the average current draw for the controller to be 167.88 mA. We measured the average current draw for the target to be 452.67 mA. As we are using a 2600 mAh battery pack, we calculated the expected battery lifespan to be 15.49 hours for the controller and 5.74 hours for the target. Given these measurements, Requirements 1, 2, and 6 are met.

For Requirement 3, response time was measured as the time between when the user pulls the trigger and when the target detects a laser hit. This was measured by software running on the controller, which tracks the network time of the most recent trigger pull. The controller listens for a network message from the target containing the time of the associated laser hit, which is emitted when a target detects the laser. When the target hit message is received by the controller, the response time is calculated and displayed on the device. A summary of the average response time at various shooting distances is shown in Table 4. The average response times at all tested distances hover exceptionally close to our 0.1 second requirement, leading us to conclude Requirement 3 is sufficiently met.

TABLE 4
SYSTEM RESPONSE TIME

Test Iteration	Time at 6in Distance (µs)	Time at 1ft Distance (µs)	Time at 6ft Distance (µs)
1	101756	108069	109829
2	101831	100109	109360
3	102875	100144	107306
4	104228	100209	101893
5	102084	106058	103798
6	101641	106588	92182
7	100289	106648	93474
8	102058	105840	117592
9	100870	106693	107076
10	102176	102727	104489
Variance (s)	**1.132079733**	**10.020969611**	**57.569254544**
Average (µs)	**101980.8**	**104308.5**	**104699.9**
Average (s)	**0.1019808**	**0.1043085**	**0.1046999**

For Requirement 4, startup time was measured as the time elapsed between the power switch being turned on and the system being shifted into the Ready state. On the controller, the Ready state is shown by the display becoming the "Main Menu" interface, while an audio clip plays. On the target, the Ready state is shown by the lights turning green, alongside an audio clip. A summary of the average startup time for the controller and the target is shown in Table 5. Timing measurements were taken by hand using a stopwatch and should therefore be taken with the understanding of some human error. As shown, the combined startup time of both components of our system met our 25 second requirement.

TABLE 5
SYSTEM STARTUP TIME

Test Iteration	Startup Time (s)	
	Controller	Target
1	2.14	1.8
2	2.11	1.73
3	2.07	1.66
4	2.09	1.8
5	1.87	1.67
6	2.09	1.74
7	1.98	1.6
8	2.09	1.67
9	2.17	1.45
10	1.94	1.74
Variance (s)	**0.008939**	**0.010893**
Average (s)	**2.055**	**1.686**

Requirement 5 is included to aid in the expandability of the system. Less effort required by the user to pair targets means more targets can be reasonably paired to the system. To pair targets, the user must navigate menu options to reach the "Pair Targets" option. The user is then required to successfully fire the laser at the target he would like to pair, which may take more than one click if the user does not hit the sensor on his first try. Assuming pairing on the first attempt, pairing can be successfully completed in a minimum of three clicks. As a user can reasonably pair a target within four clicks, we determine this requirement has been met.

Requirements 7 and 8 were tested using a common kitchen scale. The weight of the gun was measured to be 427 grams, or approximately 0.94 pounds. The weight of the target was measured to be 441 grams, or approximately 0.97 pounds. As such, these measurements meet our specifications.

For Requirement 9, the "ready to play" state was defined as the time taken from when the system is turned on to when the user can begin playing a game mode. On average, testing showed this time to be between 5 and 10 seconds. As such, a user can reasonably begin playing the game within 2 minutes of startup, meeting our requirement.

VIII. Conclusion

In sum, the system we designed and built passed our proposed standards for performance, energy consumption, and user convenience. Each of the requirements laid out prior to the prototyping of this project were met and, in many cases, exceeded. Though our system is just a prototype and is not ready for commercialization in its current form, we believe it shows promise as a proof of concept for future work.

As such, we consider this two-semester period of designing and prototyping a success. Though we experienced some setbacks - notably in our multiple redesigns of our PCB - our team worked together well. We enjoyed and appreciated the opportunity to design a project from the ground up with a group of multidisciplinary engineers, and we hope to carry the lessons we learned into our future professional endeavors.

Acknowledgements

We wish to thank Dr. Samuel Richie for his advisement during the planning and execution of this system. We would also like to thank Dr. Zhishan Guo, Dr. Qun Zhou Sun, and Dr. Reza Abdolvand for serving as faculty reviewers for our project. Finally, we thank Smart Charging Technologies for donating battery packs to our group free of cost, as well as providing their lab space for our group to develop our project.

Biographies

Thomas Stoeckert is an Honors Computer Engineering undergraduate student at the University of Central Florida. He has previously worked at Universal Creative as an intern, working as a tools developer with the Ride and Show Performance team, but works in his free time on a wide breadth of software and hardware projects. After graduation, he will be working with the Modeling, Simulation, and Training team as a Software Engineering intern at Walt Disney Parks, Experiences, and Products in Orlando.

Rachel Goodman is an Honors Electrical Engineering senior at the University of Central Florida. After graduation, she will begin her career at Burns & McDonnell in Maitland, FL in the Substation department. After spending a few years in industry, she hopes to become a licensed Professional Engineer and attain a Masters in Engineering Management, with the eventual goal of becoming a project manager. In her free time, she enjoys running, reading, and visiting all of Orlando's many theme parks.

Anna Malaj is an Honors Computer Engineering student at the University of Central Florida. During her time at UCF, she has conducted research in real time and intelligent systems, as well as interned as a software developer. After graduation, she plans to work as a software engineer at General Dynamics Mission Systems in Orlando. She hopes to attain a Masters in Computer Science after spending some time in industry.

Jamauri Balzourt is an Electrical Engineer undergrad at the University of Central Florida. After working at Smart Charging Technologies for three years, he is currently working on improving his low voltage battery health algorithms and obtaining additional industry experience before pursuing a Masters degree.

References

[1] PlatformIO Labs, "PlatformIO," 2022. [Online]. Available: https://platformio.org/. [Accessed 28 March 2022].

[2] Microsoft, "Visual Studio," 2022. [Online]. Available: https://visualstudio.microsoft.com/. [Accessed 28 March 2022].

[3] GitHub, Inc, "GitHub," 2021. [Online]. Available: https://github.com/. [Accessed 28 March 2022].

[4] S. Franzyshen, "painlessMesh," 2022. [Online]. Available: https://gitlab.com/painlessMesh/painlessMesh. [Accessed 28 March 2022].

[5] A. Arkhipenko, "TaskScheduler," 2022. [Online]. Available: https://github.com/arkhipenko/TaskScheduler. [Accessed 28 March 2022].