ECE 445: ODDS BOOSTER FINAL REPORT

By

Jack Arndt

Tim Green

Marco Rojas

Final Report for ECE 445, Senior Design, Spring 2022

TA: Uma Lath

04 May 2022

Project No. 37

Abstract

The Odds Booster is an ECE445 senior design project featuring the work of Jack Arndt, Tim Green, and Marco Rojas. The Odds Booster is meant to train players into emulating a casino experience on a physical casino table with information on what are the best decisions to make in a given situation. The dealer will scan RFID-embedded cards over our device, which will read the card and send that information to an app and an LCD monitor, all assisting in the end of goal of boosting your odds in casino games. The phone app will be used by the player, which will calculate the optimal decision and read it back to the player for him to execute. Throughout the project, we have found most of everything to function properly except the RFID card-reading module; we have provided simulations of the ideal Odds Booster experience in our app.

Contents

1 Introduction

1.1 Background

Card and casino games, such as Blackjack and Texas Hold'em, have been played for a long time and are still popular today. Many people are intimidated by these games, especially when first being introduced, because they are afraid to lose money while playing. A large source of this intimidation and fear is the lack of knowledge and experience players may have for the game. Currently, people often only learn these games by repetition and playing with their peers. We hope to help people learn to lose less money in these games and become more skilled players by creating an accessible teaching device that will provide the user with the best move to make in different situations in the context of a physical card game.

1.2 Usage and Objective

The method of providing this information to the user starts with the reading of playing cards. An RFID tag with a unique card ID is affixed to each of the 52 cards in standard deck. The card ID of card is read by the device as the dealer is passing the card to its specified location within the game. The device provides the instructions to the dealer of where to place each card within the dealing process through an LCD display. The card ID that has been scanned is sent to an external device via Bluetooth. This external device runs an app that will receive and store these card IDs to determine the state of the game. The app will then provide suggestions to the user as to what move to make given the current state of the game. A diagram showing the operation of the device is shown in Figure 1. This project focuses on Blackjack to show its implementation with one popular card game.

With this usage concept in mind, there are a set of main goals our project must meet to be fully successful. As the primary purpose of the project is user-facing, these high-level goals are primarily qualitative in nature and are as follows:

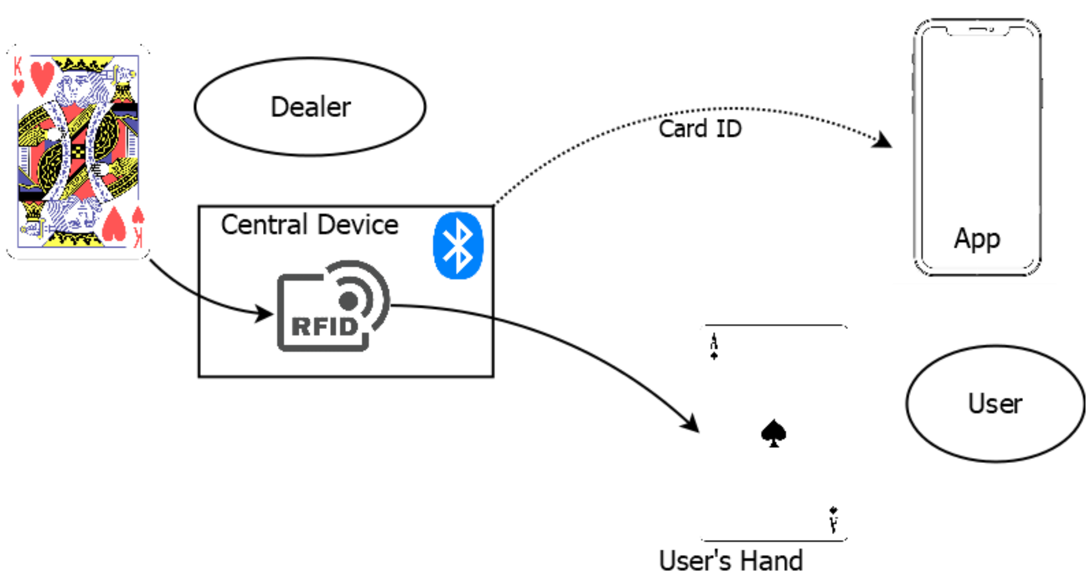

Figure 1: Device usage diagram showing flow of card being scanned and dealt to the user, then card ID being transmitted to the user's app via Bluetooth.

- The device must be able to scan and distinguish the 52 different RFID tags associated with each playing card.
- The device must be able to display the card and game information for all players to see as well as send the information to a phone app via Bluetooth.
- The phone app must be able to receive game information via Bluetooth from the device and use this information to determine the best possible move for the user.

1.3 Subsystem Overview

The system is split into five major subsystems. Four of these subsystems exist within the main device while the fifth is software written on an external device to communicate with the main device. The high level view of the components included within each subsystem as well as the interconnections between these subsystems is shown in the full block diagram in Figure 2. The center component shown is the MCU within the system. This component is utilized by and controls the card reader subsystem, the display subsystem, and the communications subsystem. For organization purposes, all discussion regarding the MCU will be within the communications subsystem.

The first subsystem is the power subsystem. It provides a stable voltage source to each of the hardware components. The major voltage lines used in the system are the 3.3 V source provided by the voltage regulator and the 9.6 V source provided by the Boost converter. The device is also designed to be rechargeable and thus has a micro-USB charging circuit with a Buck converter to generate a 4.2 V source. The card reader subsystem is the first step in the usage flow and flow of information. The subsystem includes an RFID Reader IC and a corresponding RF antenna circuit to be able to transmit and receive RF signals. The passive RFID tags can then be held above the antenna within this subsystem for

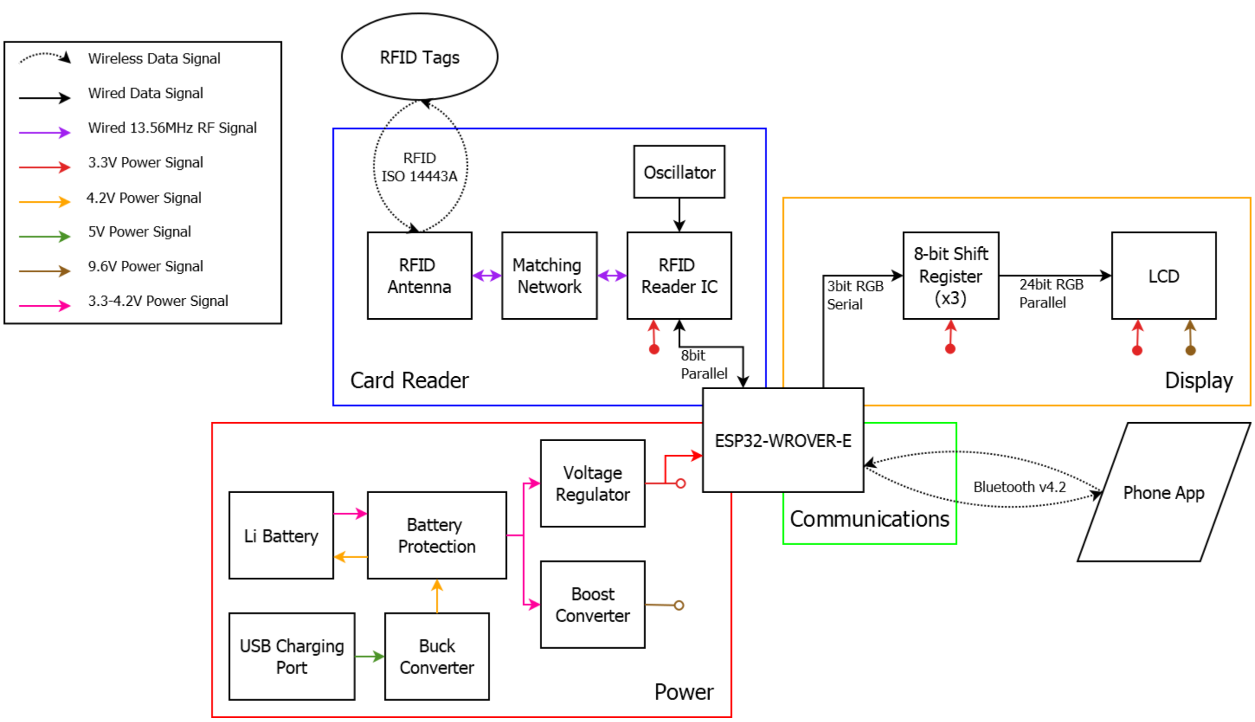

Figure 2: Full block diagram of Odds Booster device.

detection and scanning of the card ID. This subsystem receives a 3.3 V input from the power subsystem and communicates with the MCU in the communications subsystem through an 8-bit parallel protocol to transfer the scanned card IDs.

The display subsystem provides the user feedback of the operation of the central device. Dealing instructions as well as the most recent scanned card are displayed on the LCD within this subsystem to ensure correct operation of the device and provide help if the dealer is inexperienced. This subsystem receives the 3.3 V source from the power subsystem for digital components as well as the 9.6 V source from the power subsystem for powering the backlight of the LCD. Additionally, all control signals are provided from the MCU in the form of 3 serial lines used to encode 24-bit color data.

The communications subsystem includes the MCU and sends the card ID from the device to the external app. As this subsystem includes the MCU, it sends and receives wired data signals to and from the card reader and display subsystems. It also sends and receives 2.4 GHz RF signals using Bluetooth protocol to and from the app subsystem. The communication subsystem uses the 3.3 V source for power.

The app subsystem is the final step in the usage flow and is responsible for using the card IDs to generate suggestions for the user. The app subsystem exists on an external device and thus does not utilize the designed power tree. The external device is assumed to have a working power source and Bluetooth capabilities to interface with the communications subsystem.

1.4 Project Results

Through the development of this project, the system showed strong promise but the full functionality we had planned for was not met. The power provided to the device worked as expected allowing for stable sources for each of the other components. The Bluetooth aspects of the device worked perfectly, allowing for the request of data from the app and the transmission of card information from the MCU back to the app. Additionally, the app functioned properly providing proper suggestions to the user based on the status of the game in the form of a text based GUI. The main problems with the functionality of the project stemmed from the usage of the RFID Reader IC within the card reader subsystem, and the operation of the LCD within the display subsystem. These results will be discussed in further detail throughout this report.

This report will begin with the original design of the device, and the changes to this design made along the development process, broken down by the five subsystems previously discussed and shown within the full block diagram. Next the test results of each system are explained providing evidence of successful components of the project and reasoning for why any component did not function as expected. The total cost of the development of the project is examined in the context of the physical components required as well as the work hours done for the project. Finally, the end results of the project are discussed and potential improvements to the functionality of the project are explored.

2 Design

This section will discuss the considerations and planning of each subsystem, as well as the specifics used for the circuit schematics. All circuit schematics are included in Appendix B. An important piece along with the design, the requirements for success for each subsystem are included within Appendix A. Each component chosen within this design had a specific reasoning and the important components will be discussed in detail. One over-arching requirement for all components chosen is the need to be hand-solderable. Many modern components come in small with small packaging such as QFN (Quad Flat No-Lead) or BGA (Ball Grid Array). As our main construction method would be soldering by hand, we avoided all parts with these smaller sizes. The completed design of this device is shown in the board layout in Figure 15 in Appendix C and the physical implementation in Figure 16 in Appendix C.

2.1 Power Subsystem

The power subsystem provides stable voltage sources to each of the other hardware subsystems. The necessary voltage sources identified for the system were a 3.3 V source for digital CMOS components and a 9.6 V source to power the backlight of the display subsystem. A linear voltage regulator and boost converter were chosen to provide these voltages.

The components within our design that draw the most current are the RFID reader IC, the MCU, and the LCD backlight. The maximum current on each of these devices was used to determine an overall maximum current draw of

$$I_{MCU} + I_{RFID} + \frac{V_{boost}}{V_{bat}\eta_{boost}} I_{LCD} \approx 0.6 + 0.15 + \frac{9.6}{3.7 * 0.7} 0.05 \approx 0.96\ A \tag{1}$$

where I_{MCU} is the maximum current from the MCU, I_{RFID} is the maximum current from the RFID reader IC, η_{boost} is the estimated worst-case boost efficiency, V_{boost} is the boost output or LCD backlight voltage, V_{bat} is the nominal battery voltage, and I_{LCD} is the LCD backlight maximum current. This current requirement was used to find a battery capable of this continuous current.

The chosen battery is a Lithium Ion battery with nominal voltage of 3.7 V and can be recharged with a voltage of 4.2 V. It has a capacity of 3000 mAh leading to a worst-case battery life of 2.78 hrs. The battery has a maximum continuous discharge current of 1.5 A. A charging circuit is implemented with a buck converter to move from a 5 V micro-USB input voltage to the needed 4.2 V source. The micro-USB input was chosen as this is a common connector that many people own and use for charging small devices. The battery also includes a battery protection IC to protect from overdischarge and overcharge voltages and currents. This subsection will discuss the design of these modules in further detail.

2.1.1 Linear Voltage Regulator

A low dropout linear voltage regulator (LDO) is used to provide the 3.3 V source for digital components. The linear voltage regulator is chosen instead of a switching regulator because the change in voltage from the battery to the desired source voltage is very small, the linear regulator can match the input voltage at the output if the input falls below the desired output, and the linear regulator does not use high switching frequencies which could interfere with other high frequency signals on the board. The main disadvantage of linear regulators is the low efficiency for large voltage drops and subsequent heat generation. With a change of 0.4 V from the nominal battery voltage to the output voltage, the efficiency of this device will still be large and comparable to a switching device.

The LDO module implemented consists of the LDO along with external passive components for biasing and maintaining a stable output voltage. The LDO is chosen to have a maximum current of 1.5 A to support the full range of safe discharge currents from the battery. The LDO has an adjustable output voltage through feedback provided by a resistor voltage divider. This allows for fine tuning of the output voltage using a potentiometer. The R1 resistance in the divider is set to 10 kΩ and the R2 resistance used for the divider is calculated to be

$$R2 = R1\left(\frac{V_{LDO}}{1.216} - 1\right) = 10\left(\frac{3.3}{1.216} - 1\right) = 17.138\ k\Omega \tag{2}$$

where V_{LDO} is the desired output voltage. The resistance equation as well as the input and output capacitor capacitance and material are suggested by the LDO datasheet [2]. The schematic showing the LDO implementation is in Figure 9 in Appendix B.

2.1.2 Boost Converter

A boost converter is used to provide the 9.6 V source to the LCD backlight. A boost converter is chosen over other step-up options such as a Flyback converter because of its simplicity and common use in low power applications. The boost module implemented in this design is constructed using a boost converter gate driver IC. This component controls the PWM signals provided to the gates of the switching MOSFETs. The inductor, MOSFET, and other passive components are external to this IC. Fully implemented buck converter components are available for purchase, but we were urged to implement the module in a more complex way and thus chose the gate driver method. One benefit of the gate driver is the current does not flow through the IC allowing the external components to be chosen to better match the needs of the application.

The chosen gate driver suggests an inductor of value 10 µH. The diode voltage drop was chosen to be a low and common value of $V_D = 0.4\ V$. The PWM gate signal is chosen to be 500 kHz as an approximate center point to the available frequencies of the gate driver. The oscillations caused by the switching causes variations in the current through the boost components. The peak current can be calculated to be

$$I_{pk} = \frac{V_{out} + V_D}{V_{in}} I_{out} + \frac{(V_{out} + V_D - V_{in})V_{in}}{2(V_{out} + V_D) f_{osc} L} = \frac{9.6 + 0.4}{3.3} 0.05 + \frac{(9.6 + 0.4 - 3.3)3.3}{2(9.6 + 0.4)(500 * 10^3)(10 * 10^{-6})} = 0.38A \tag{3}$$

where V_{out} is the output voltage of the boost, V_D is the diode voltage drop, V_{in} is the minimum input voltage, I_{out} is the maximum output current, f_{osc} is the oscillation frequency, and L is the inductance. This current tells the maximum current to be supported by all components within the system, particularly the inductor to avoid magnetic saturation. To generate an oscillation frequency of 500 kHz, an oscillation bias resistor of 280 kΩ is used. To allow for the maximum range of voltage outputs, the maximum duty cycle is set by the duty bias resistor to the gate driver maximum of 85% using a resistance of 240 kΩ. The feedback to the system is provided through a voltage divider like the LDO. The output voltage is made adjustable by the inclusion of a potentiometer in the resistor voltage divider. R1 of the divider is chosen to be 82kΩ and R2 is then calculated to be

$$R2 = \frac{R1}{V_{out} - 1} = \frac{82}{9.6 - 1} = 9.5\ k\Omega \tag{4}$$

where V_{out} is the output voltage of the boost. The bias decisions, peak current calculations, and input and output capacitor capacitance and material is suggested by the boost converter gate driver data sheet [1]. The boost converter implementation schematic is shown in Figure 10 in Appendix B.

2.1.3 Buck Converter

A buck converter is used to provide the 4.2 V source to the battery for charging purposes. Like the boost converter, the buck converter was chosen because of its simplicity and common application in low power applications. A linear voltage regulator is not used in this application because the voltage drop is large enough to cause a more significant decrease in efficiency. Additionally, the switching converter is not used to power any components that may be affected by higher frequency switching interference. The buck module is constructed with a gate driver IC like the boost converter for complexity and to better match our application.

The chosen gate driver has a set oscillation frequency of 500 kHz. The maximum output current for the module is 2 A to accommodate the charging current of the battery of 1.5 A. The desired inductance of the buck converter inductor can be calculated to be

$$L = \frac{V_{out}(V_{in} - V_{out})}{0.2 V_{in} f_{osc} I_{out}} = \frac{4.2(5 - 4.2)}{0.2 * 5 * (500 * 10^3) * 2} = 3.3\ \mu H \tag{5}$$

where V_{in} is the input voltage to the buck, V_{out} is the output voltage to the buck, f_{osc} is the oscillation frequency of the gate driver, and I_{out} is the maximum output voltage of the buck. The peak-to-peak current ripple from the oscillations and the suggested maximum current limit of the system can be calculated as shown in Equation (6) and Equation (7) respectively.

$$I_{ripple} = \frac{V_{out}(V_{in} - V_{out})}{V_{in} f_{osc} L} = \frac{4.2(5 - 4.2)}{5 * (500 * 10^3) * (3.3 * 10^{-6})} = 0.4\ A \tag{6}$$

$$I_L = I_{load} + \frac{1}{2 I_{ripple}} = 2 + \frac{1}{2(0.4)} = 3.25\ A \tag{7}$$

The maximum current limit is used for component requirements and for the current limiting feature of the buck converter. The battery can support up to a 3 A current for fast charging. This current limit thus is in a good range to provide an effective method to prevent the output current of the buck from being too high by then lowering the output voltage. The current sensing for the limiting behavior is done using a current sensing resistor and the on resistance of the upper MOSFET. The current sensing resistance is

$$R_{cs} = \frac{R_{ds(on)} I_L}{200 \mu A} = \frac{35 m\Omega * 3.25 A}{200 \mu A} \approx 560\ \Omega \tag{8}$$

where $R_{ds(on)}$ is the on resistance of the MOSFET. The output voltage is made to be adjustable by the inclusion of a resistor voltage divider with a potentiometer. The feedback bias resistor R1 is chosen to be 8.2 kΩ. R2 can be calculated as

$$R2 = \frac{0.8R1}{V_{out} - 0.8} = \frac{0.8(8.2)}{4.2 - 0.8} = 1.93\ k\Omega \tag{9}$$

where V_{out} is the output voltage of the buck. The bias decisions, peak current calculations, current limiting biasing, and input and output capacitor capacitance and material is suggested by the buck converter gate driver data sheet [12]. The buck converter implementation schematic is shown in Figure 11 in Appendix B.

2.2 Card Reader Subsystem

This subsystem will discuss the design considerations and decisions that went into the detection of the card being scanned and communicating that information to the MCU. A high-level understanding of the way this subsystem is set up is shown in Figure 2 in the blue box.

2.2.1 RFID Tags

The very first step in the user's use of The Odds Booster is grabbing a card and scanning it over our central device that will identify the card. There were multiple ways the team thought of how the card will be identified. The most important parameters were ease of access and reliability, as this is a part of the project that users should hardly notice as they are playing. The Odds Booster is meant to be an emulation of the casino experience and having to lag the dealing of cards due to faulty detection is not desirable. One possibility was the use of QR codes printed on the cards, which would be scanned by an external camera. Another was putting light sensors on the cards, since the only time light is shined on the card is when they're being dealt. The former wasn't considered since having a camera as part of the system was deemed too extensive and potentially expensive. The latter wasn't considered because it'd be difficult to establish communication between the light sensors going off and having a microcontroller pick up on that and process what is going on in the game (as well as unintended readings).

The Odds Booster ended up using RFID tags on each card that can be scanned over an antenna connected to an RFID reader IC, which is then connected to the MCU; this is how casinos already do card detection for televised poker tournaments. Once the tags were decided, the next consideration is how they will be attached to each card. Casinos embed the tags into the cards since they shouldn't be able to be detached from the card under any circumstance (without obvious defacement). However, this process was found to be quite expensive and requires a lot of custom work, which could take a long time to complete. In the interest of time and simplicity, the Odds Booster uses small RFID tags with a wet-inlay (able to stick to surfaces). This is shown in Figure 3, inside the red constraints on the center of the card. The tags are put in the center of the card for the dealers to always know where the tags should be scanned over the antenna. The tags are encoded with a UTF-8 text file identifying the card.

2.2.2 RFID Reader Technology

After the decision was made to include RFID technology for card identification, we needed something that can pick up signals from the RFID tags and can process them to be sent to the MCU. The RFID tags operate at a frequency of 13.56 MHz, an RFID standard, and contain encoded text files that

Figure 3: Odds Booster cards with RFID tags stuck on them

take up 11 bytes of information. They also operate under the ISO 14443A protocol, which is standard for "proximity cards". The power tree also only permits a 3.3 V input for the RFID reader IC.

The antenna was a relatively easy search, since we simply needed an antenna that operates at a frequency of 13.56 MHz and has a solderable package. It also measures at an internal impedance of 50 Ω, a standard for any RF terminal. Originally, the team thought it would be necessary to include bidirectional couplers in our design, however this was decided against as these are for multi-port devices that need to bring clarity to signals going in multiple directions, where our design simply has one way it needs to go. That leaves the RFID reader IC, which needs to fulfill the above requirements. The MFRC522 is very commonly used since it's an entire module with an antenna already built in, but this increases the price tenfold (as well as having too much integration for the scope of this project). The MFRC500 is a good choice but could only take 5 V inputs so that is also off the table. Luckily, the MFRC531 is pin-compatible with the MFRC500 and can take either 3.3 V or 5 V inputs and fulfills the ISO 14443A protocol with SPI connection for the 11-byte files. The package is also an SO32, which can fairly easily get soldered onto a PCB. With a very extensive datasheet (117 pages), information was available on how to draw up the matching network and external oscillator to achieve proper integration, and that is what we followed up with to scope out the rest of our subsystem. A 13.56MHz quartz crystal oscillator is required to be added to the system, as shown in the Figure 10 of the MFRC531 datasheet [10, pg.29].

For the pins that relate to MCU interfacing, that is shown in Table 5 of the MFRC531 datasheet [10, p.8]. The Odds Booster utilizes the separated read/write strobe with a multiplexed address bus since it is most clear how to connect the pins and having a multiplexed address bus will make it easier to see the address and data information being sent between the two devices. This nullifies the usage of the three address pins, and requires direct connection of the ALE, NRD, NCS, and NWR pins to the MCU. A parallel interface was chosen over a SPI since the MFRC wouldn't have the automatic detection capabilities and would have further complicated the process.

As previously mentioned, there is documentation available on how to properly connect an active antenna to this device, as shown in Figure 6-1 of the MFRC500 Application Note [11, p.11]. This is what was implemented in the final schematic as shown in Figure 12 in Appendix B.

2.3 Communications Subsystem

The communications subsystem is responsible for sending the card information via Bluetooth to an external device. The subsystem is designed using a single MCU module with an internal 2.4 GHz antenna used for Bluetooth or Wi-Fi. This module was chosen to avoid the high frequency traces necessary for including an external 2.4 GHz antenna. These high frequency traces require small tolerances for proper functionality which would greatly increase the cost of the PCB.

The chosen module is the ESP32-WROVER-E. This MCU has a high internal clock useful for controlling the LCD which requires high frequency signals. The MCU has 8 MB of RAM, more than components used for similar applications which is useful for the higher memory requirements for holding Bluetooth communication queues and buffers as well as the frame buffer for the LCD. The module also has extensive documentation about how to use and program the module as well as the commonly used ESP API for making certain operations with the MCU simpler. One such operation is UART communication which makes the loading of programs to the MCU easier.

The schematic showing the implementation of the MCU module is shown in Figure 13 in Appendix B. GPIO pins are used to connect this module to the card reader subsystem through an 8-bit parallel interface with multiplexed address and data pins along with read and write signals. GPIO pins are used to connect this module to the display subsystem through three color encoding pins and synchronization or control signals such as the horizontal sync, vertical sync, and data enable pins. External features are included to allow for programming the MCU through UART header pins and controlling the boot loading method. The boot loading method is controlled by the slide switch allowing the choice of loading a new program through the UART pins on start-up or running the existing program stored in flash memory. The pushbutton is used to reset the MCU upon a bug or load a new program to the MCU. This pushbutton has a simple debouncing circuit with a time constant of $\tau = RC = 1\ ms$. This filters out high frequency parts of the press that could cause the system to restart multiple times with a single push and cause errors in the loading of the UART program. A simulation of the effect of this filter is shown in Figure 4 with the signal before and after the debouncing filter.

The firmware loaded to the MCU is simple. The code has three main actions: receive card ID, display card ID, and send card ID via Bluetooth. The received card ID is signaled by the IRQ interrupt pin from the card reader subsystem. Upon receiving a card, the second two actions are both performed. The simplicity of the firmware is due to the setup of the system. The Bluetooth capabilities are established to receive a packet on an interrupt from the system and run the receive action in an alternate thread. This

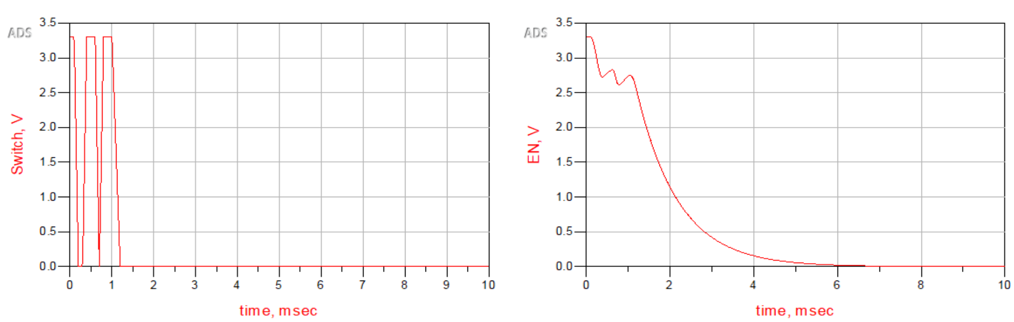

Figure 4: MCU Before and After Debouncing Filter Simulation

frees up the main loop to sit idle until a new card is scanned. Additionally, the LCD is set to read from direct memory access (DMA) memory allowing the GPIO pins to output the memory of the frame buffer without using the CPU. This means actions only need to be performed by the CPU to update the memory stored in the frame buffer. Upon receiving an interrupt from the card reader subsystem, the main loop simply sends the card ID to the external device in a Bluetooth packet and updates the frame buffer to show the card ID in text.

2.4 Display Subsystem

The display subsystem is used to provide the user feedback about the card that has just been scanned. The display component chosen for the system was selected to be visually appealing with a high pixel count and extensive color depth. Liquid Crystal Displays (LCDs) are cost effective and show more appealing picture qualities than monochrome or seven segment style displays. Many of these displays use a VGA style interface to control the display contents. Devices with a simpler communication method such as an SPI interface required an additional controller component or were more expensive. Our group felt comfortable working with a VGA controlled display because of experience with the protocol in previous work.

The selected display has a 24-bit color depth (8 bits for each RGB color). Using 24 GPIO pins from the MCU for this control is not possible for the chosen MCU. To reduce this, three data pins are used. One data pin is used for each color signal. These data pins will then output the 8 bits of color serially to a shift register for each pixel. The shift register will then hold the 8-bit color parallelly for use by the LCD. This requires a clock eight times the pixel clock to output the color data and control the shift registers.

The chosen LCD has a 320x240 pixel resolution. This resolution requires a 6.5 MHz pixel clock to maintain a 60 Hz refresh rate. The bit data clock mentioned previously at eight times the pixel clock requires a clock rate of 52 MHz for the data output and for the shift registers. During the construction phase, a shortcoming of the MCU was discovered. The master clock of the MCU is 160 MHz and peripheral clocks are based on this master using a decimal clock divider. The minimum divider value is four meaning the maximum peripheral clock rate for the MCU is 40 MHz. This causes the shift register method to fail due to system not being able to generate a fast enough clock signal. As a backup plan, each color output pin from the MCU is tied to all eight color input pins of the LCD. This reduces the color depth to 8 possible colors but leads to a functional device only needing the 6.5 MHz clock signal and not the 52 MHz clock signal.

The final consideration for the display is the LED backlight allowing the displayed image to be seen in all lighting settings. The display requires a 9.6 V source for this backlight and this source is provided by the boost converter. The schematic showing the display connector and shift register connections is in Figure 14 in Appendix B.

2.5 App Subsystem

The app subsystem runs on the user’s phone and is responsible for receiving card information from the MCU and sending the necessary control information back to the device through Bluetooth. Further, this is a particularly critical subsystem in that it is liable for calculating the optimal decisions or relevant game statistics for a given situation in each game setting and transmitting this information back to the user in the form of text. The flowchart of the application is as shown in Figure 5.

One main requirement of the external device running the app is it must have Bluetooth capabilities. The two main candidates for mobile app development were Android and iOS. While there exists a large user base for both platforms, each had their respective difficulties. We had initially proposed to develop the application using the Android Studio IDE and Android SDK because of Android's open-source nature and application development process promoting compatibility across most, if not all Android devices. However, we were unable to obtain the necessary Android hardware in a tolerable time frame to develop, test, and debug the software. Further, Apple has many strict guidelines and constraints in place regarding their development process for iOS-based applications that prevented us from adopting this approach. Taking this into account, we ultimately decided that our best approach would be to emulate a phone-based application by writing the software in a computer program using C++. This originally served as our back-up plan if we encountered issues with the phone application development process, as outlined above. However, we still were able to successfully interface with the MCU as almost all modern computers also have Bluetooth capabilities, so this approach was still well-within our defined requirements of the application. C++ was our language of choice to write the application due to our familiarity with the programming language and its ability to potentially be integrated into an Android app.

The choice of implementing Blackjack as our primary game selection was motivated by the fact that the user is prompted to take a definitive action (i.e. hit, stand, double), which greatly simplifies the overall experience for new users. On the other hand, games such as Texas Hold'em poker will output the "strength" of the user's hand in the form of percentage of "winning" a hand, which could cause initial confusion to players unfamiliar with such decisions. The algorithm shown in Figure 5 takes inspiration from a "Super-Easy Simplified Blackjack Basic Strategy Chart" [13] that is commonly utilized by many novice Blackjack players. For this project, we chose to only implement one, very basic strategy to appeal to a larger pool of players. However, future considerations and improvement of the application suggest extending user customizability by allowing the players to select their own strategy they would like to employ. This would only require minor modifications to the existing code.

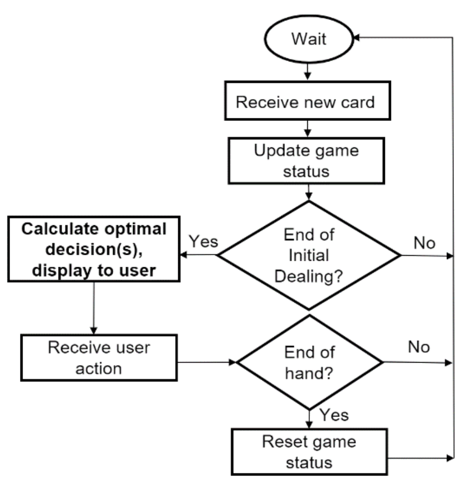

Figure 5: App flowchart diagram

3. Design Verification

The verifications corresponding to each subsystem are included along with the subsystem requirements in Appendix A. These tables show if the requirement was met or not during our time with this project. This section will discuss the results of the verifications and provide reasoning for why a verification was unsuccessful.

3.1 Power Subsystem

The power verifications discussed in this subsection are outlined in detail in Table 5 in Appendix A, along with their overall results. The LDO and boost converter are successful which indicates an operational device The failure of the buck converter only affects the charging capabilities. Finally, the battery protection IC included with our chosen Lithium-Ion battery was successfully unit tested and verified to ensure proper behavior of the battery prior to powering the system.

3.1.1 Linear Voltage Regulator

The LDO is tested by connecting a DC power source to the LDO input or battery pins. The input voltage is varied between 2.5 V and 3.7 V, and the output voltage produced by the LDO is measured. These measurements are taken at 0.2 V increments and are plotted in the graph shown in Figure 6. The output voltage matches what is expected by matching the input voltage up to 3.3 V then maintaining a constant 3.3 V output. This meets the needs specified in the requirements and verifications.

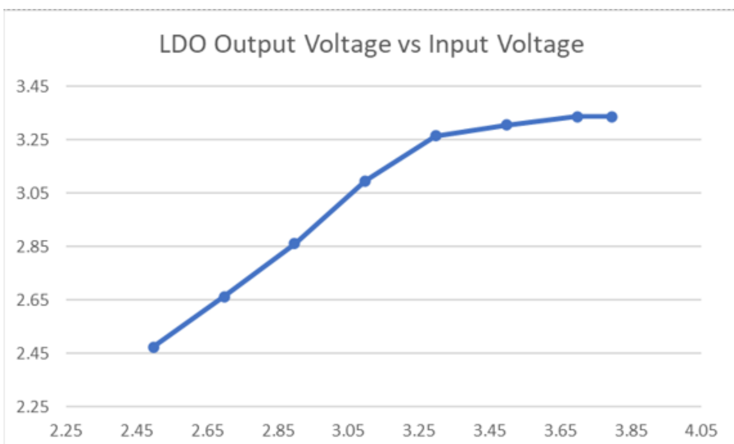

Figure 6: LDO stability testing results

3.1.2 Boost Converter

The boost converter testing method as specified in Table 5 of Appendix A does not specify a range of input voltages, but instead specifies the output voltage and tolerance for a given input voltage. This baseline requirement was met as well as additional testing with input voltages beyond the nominal battery voltage showcasing stable output voltages for the entire voltage range of the battery. This requirement was tested with a DC power source providing the input voltages. The various input voltages and recorded output voltages of the boost converter are shown in Table 1.

Table 1 Boost Converter Output Testing Measurements

Input Voltage (V)	2.99	3.29	3.59	3.89	4.19
Output Voltage (V)	9.59	9.59	9.59	9.59	9.60

3.1.3 Buck Converter

The buck converter does not meet the step down requirements. The testing method used is as stated in Table 5 of Appendix A. A DC voltage source is used at the charging pin input to the system and the output voltage of the buck converter is recorded. In the application of the device, this input voltage is provided by a USB input which does not vary significantly. The testing is thus performed using only the 5 V input voltage. The recorded output voltage of the buck converter is around 2 mV.

Through the debugging and investigation of the problems that may be causing this error, the gate signals generated by the buck controller IC are examined. Figure 7 shows the top MOSFET gate signal. When this signal is high, the output is connected to the input voltage. The output voltage is regulated by controlling the duty cycle and removing the high frequency changes through passive components. This gate signal shows a duty cycle very close to 0% leading to an output voltage very close to 0 V. The reasoning for why the regulation is not functioning properly is likely due to the feedback provided to the controller. A feedback voltage higher than expected at the controller IC will lead to a reduction in the duty cycle of the top MOSFET and a lower output voltage. The feedback provided to the controller may be tied high due to soldering problems. The buck gate driver IC has a much smaller packaging than expected and was very difficult to solder without unintentionally connecting pins.

3.2 Card Reader Subsystem

The testing method used for the card reader subsystem is stated in Table 6 of Appendix A

3.2.1 RFID Tags

The NXP TagWriter iPhone app is designed to be able to read RFID tags, which was downloaded for the use of verifying the tags' functionality. Each and every card was encoded also using this app, which has a Write feature with a Plaintext option to encode the tags. After encoding, the tags were tested using the same app and they were all able to be read, thus verifying its functionality.

3.2.2 RFID Reader Technology

Before discussing the shortcomings of this part of the project, one part of this that was able to be successfully resolved was the "0 net" from the PCB layout. During construction of the schematic, different partners worked on each subsystem, thus causing the issue of having different ground symbols

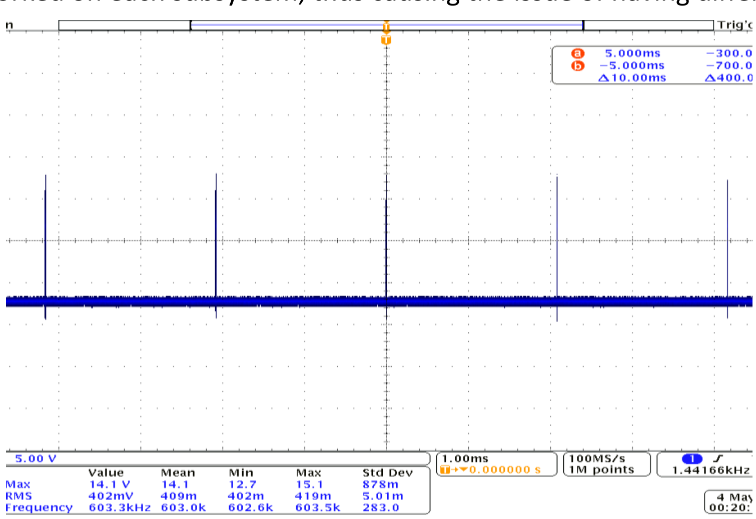

Figure 7: Buck device's duty cycle gate signals

used. When transferred to the layout, this translated the different ground symbol for *this* subsystem to not actually be ground, but rather a floating zero. This was solved via jumper wires to real ground pins on the physical board, thus connecting the floating pins to ground.

The MFRC531 was not able to successfully identify the microprocessor parallel interface type during the StartUp phase. In order to have proper communication to and from the MCU and MFRC, this automatic detection needs to be successful, in which the connection scheme is identified. Per section 9.1.2 of the datasheet, "The MFRC531 identifies the microprocessor interface using the logic levels on the control pins. This is performed using a combination of fixed pin connections and the dedicated Initialization routine" [10, p.7]. This points the conflict directly at the control pins for the connection scheme chosen for this project, which was Separated read/write strobe with a multiplexed address bus. These pins are NRD (not read input), NWR (not write input), NCS (not chip-select input), and ALE (address latch enable). Unfortunately, since there was no more GPIO pins available on the MCU as they were taken up by the display and MFRC combined, and as an oversight on how to properly set up the parallel interface, assumptions were made that the NCS pin was not necessary as there was only one chip being used and the pin isn't required. Thus, the NCS pin was connected to ground in the final implementation. Upon further inspection of the datasheet, Figure 20 [10, p.94] gives the timing diagram for the separate read/write strobe, in which it clearly shows the NCS pin acts as a latch to allow read/writes from the address bus. Therefore, it is the likely scenario that this is the prime suspect as to why the MFRC isn't operating properly and would be considered in a redesign.

One thing that certainly is not the issue is the external oscillator, which is connected to the MFRC operating at 13.56 MHz acting "as a time basis for the synchronous system encoder and decoder" [10, p.29]. The MCU was still able to read commands from the Command Register of the MFRC, and upon the StartUp phase it is evident that the MFRC can reach the end of the initializing phase after confirming "the oscillator attains clock frequency stability at an amplitude of >90% of the nominal 13.56 MHz clock frequency" [10, p.28]. This is evident via a command value "bf" in Command Register which indicates this exact part of the phase. The reason it is not "3f", like the datasheet says the Command Register should reach, is due to the MSB being "1", which indicates the "interface detection [is] ongoing" [10, p.48]. The Command Register would never leave this value, as the interface detection never finished successfully. Due to these complications, no requirements were able to be verified regarding proper communication to and from the MCU, as well as antenna readings triggered by the RFID tags since there is no RFID reader IC to pick up the signal.

3.3 Communications Subsystem

The communications subsystem is tested as explained in Table 7 in Appendix A by placing the device at a distance of about three feet from the paired Bluetooth device. A 5 kB file is generated on the paired external device containing the sequence of bites '12345' repeated a thousand times and sent to the MCU. The MCU receives this data in a sequence of packets with maximum length 989 bytes and responds with an acknowledgement recording the time of the sent acknowledgement. The external device repeats this upon receiving the acknowledgement. The total time of the 5 kB transfer is thus approximated with the difference between two acknowledge sends. The speed of the data transfer is measured by dividing the number of bits in the message (40,000) by the time between messages. This time approximation is larger than the actual time and thus the speed measured is a lower bound approximation. The actual throughput may be greater. The recorded average throughput with this method was 17.1 kbps. The speed measurements for five 5 kB file tests are shown in Table 2.

Table 2 Communications Subsystem Speed Test Measurements

Trial #	1	2	3	4	5	Average
Speed (kbps)	19.48	12.97	19.09	12.54	21.60	17.14

3.4 Display Subsystem

The display subsystem is tested as explained in Table 8 of Appendix A. The first requirement specifies the clock signals generated by the MCU. As discussed within the design section, the MCU is not able to output clock signals greater than 40 MHz. Because of this, the first requirement was not met. The second requirement specifies that the data of the output pins be updated synchronously with the clock signal. As there is no 52 MHz clock in the system using the back-up plan discussed previously, the test is performed updating the data signals with the pixel clock. This is seen as columns on the screen and switches the data pin from high to low on each rising edge of the clock cycle effectively behaving as a clock divider. Figure 17 and 18 in Appendix D show the oscilloscope readings from this test indicating the successful generation of the 6.66 MHz clock signal and the data signal with a frequency of 3.33 MHz, exactly half the clock frequency as expected. This test is considered successful because given the new considerations, the purpose of the test of controlling data pins at the necessary clock rate is achieved. The final test is updating the frame buffer stored in PSRAM during operation of the device. This test is not measured quantitatively but qualitatively through the changing of the screen while active. This test was seen to be successful before the backlight was functional and was seen by the changing of the screen from all black pixels to all white pixels when the frame buffer is updated by the MCU.

When originally constructed, the layout had an error where the V_{DD} pin and the GND pin used for the backlight of the LCD were swapped. Because of this, the initial tests with the LCD showed the expected functionality demonstrated in the previously mentioned tests and verifications. Through efforts to modify the board and fix the backlight issue, the functionality shown in previous tests was no longer seen. Instead, the backlight was seen to be functional, but the screen was not displaying the expected image. The clock and data signals measured at the MCU match what are shown in Figure 17 and 18. The synchronization signals (H-Sync and V-Sync) indicate when to move to the next line or when to refresh the screen are shown in Figures 19 and 20 respectively and are the expected frequencies of 16.34 kHz and 62 Hz. If any of these signals were not properly connected to the LCD due to soldering issues arising from the soldering difficulties and backlight issues, the screen would not be functional.

The implementation of the LCD in this system required several compromises such as the reduced color depth. Additionally, storing the frame buffer for the LCD screen uses a large majority of the available RAM of the MCU leaving little for the Bluetooth and card reading applications. If redesigning this system, a dedicated controller for the LCD would have better suited the device.

Figure 8: GUI suggesting user to "double" in given blackjack hand

3.5 App Subsystem

The app subsystem tests are shown in Table 9 in Appendix A and are all qualitative. The Bluetooth communication capabilities are confirmed through the communications subsystem tests explained previously whose results are shown in Table 2. The text based suggestion output is confirmed through the testing of the app. A sample image of the text-based GUI produced by the app is shown in Figure 8. The strength measure test for the case of Blackjack is instead a suggestion provided to the user as can also be seen in Figure 8. These suggestions were compared to the move suggestion table [13] for several sample hands. The cards and suggested move for several sample hands are shown below in Table 3. Each of these hands and all other suggested moves produced by the app over the course of testing and demonstrating have been compared to the reference table and match as expected showing accurate suggestions provided to the user.

Table 3 App Subsystem Suggested Move Sample Hands

Sample Hand #	1	2	3	4
User Card 1	7 Hearts	4 Diamonds	8 Clubs	Ace Diamonds
User Card 2	Jack Hearts	5 Hearts	5 Spades	6 Spades
Dealer Card	8 Diamonds	6 Clubs	9 Spades	10 Diamonds
Suggested Action	Stand	Double	Hit	Hit

The app subsystem passed each of its requirements, but as discussed earlier, the card reader subsystem is not fully operational. To enable the demonstration of the functional components of the device excluding the card reader, a simple simulation is made. The simulation operates by instead of reading a new card when it is scanned by the card reader subsystem and sending this information to the app, the app sends a request via Bluetooth to the communications subsystem. The communications subsystem generates a pseudo-random card from the deck without replacement and responds with this card ID. This allows the app to function as would be expected with a functional card reader. The simulation deck within the communications subsystem is reset to contain all 52 cards when the app subsystem requests that a new hand be started.

4. Costs & Schedule

This section examines the total cost of the project in terms of hardware costs and labor costs. The total cost of the project can be seen to be $31,632.98. This section also includes a breakdown of the work done for the project by each team member for each week. This schedule is shown in Table 4.

4.1 Parts

The parts cost was a small portion of the total cost and reached a total of $132.98 as shown in Table 10 in Appendix E. Note that the quantity of parts shown for some components is more than what is needed in our physical design. Spare components were purchased for small or difficult to solder parts.

4.2 Labor

The typical ECE graduate at the University of Illinois Urbana-Champaign will make around $35/hour. For 12 weeks at around 10 hours per week, with the 2.5x factor mandatory to this analysis, we arrive at a total of $10,500 per person. For three group members, the total labor cost is **$31,500**.

4.3 Schedule

Table 4 Schedule overview by week

Week	Jack Arndt	Tim Green	Marco Rojas
3/7	• Finalize PCB layout • Teamwork evaluation	• Finalize machine shop plans • Teamwork evaluation	• App pseudocode development • Teamwork evaluation
3/14	Spring Break	Spring Break	Spring Break
3/21	• MCU and Phone App development	• MCU Firmware Devkit Testing	• MCU and Phone App development
3/28	• PCB construction: *Power* subsystem • Individual Progress Report	• PCB construction: *Comms/LCD* • Individual Progress Report • MCU Bluetooth development	• PCB construction: *Card Reader* subsystem • Individual Progress Report
4/4	• PCB Construction • Power subsystem testing/debugging	• PCB construction • Comms hardware testing/debugging	• PCB Construction • Card Reader MCU interface development
4/11	• App pseudocode development	• MCU Bluetooth debugging • MCU Display development	• Card Reader MCU interface development • App pseudocode development
4/18	• Display subsystem testing/debugging • Phone app simulation development	• Display subsystem testing/debugging • Phone app simulation development	• Card Reader subsystem testing/debugging • Phone app simulation development
4/25	• Full system debugging + demo prep • Presentation prep	• Full system debugging + demo prep • Presentation prep	• Full system debugging + demo prep • Presentation prep
5/2	• Complete final paper • Finalize lab notebook • Teamwork Evaluation II	• Complete final paper • Finalize lab notebook • Teamwork Evaluation II	• Complete final paper • Finalize lab notebook • Teamwork Evaluation II

5. Conclusion

5.1 Accomplishments

The Odds Booster ended up with many accomplishments to be proud of, starting off with the hardware/physical features. The power tree is a success, with the right voltages being produced from the LDO and Boost for the MCU, LCD, and Card Reader subsystems. A physical playing card deck is available for use with RFID tags stuck on the exterior of the card, encoded with text files identifying the card in which the tag is on. The proper clock, data, and synchronization signals are produced in the display subsystem. The ESP32, the MCU for the Odds Booster, has successful UART programming set up to download programs to flash storage and execute them at the next reset or power up.

In terms of software, everything was a hit. The Odds Booster establishes functional Bluetooth communication via the MCU's communication subsystem, which was connected to a groupmate's laptop for the demo. The code written for the blackjack game is accurate and successful, giving the correct optimal decisions when prompted the cards; furthermore, a text-based GUI is present in this app. This GUI allows the user to easily follow the suggested actions from the calculations of the backend code.

5.2 Uncertainties

Despite the accomplishments, there were uncertainties and issues that were reached throughout the course of this project. To start with the hardware issues, the display subsystem ended up pushing the MCU to, and in some cases beyond, its limits. A controller for the LCD would've been preferrable removing the load on the MCU of generating the strict timing requirements and storage of the large frame buffer which took up almost 90% of the available data RAM. The buck converter would have provided the recharging functionality to the device, but the output of the buck converter was measured around 2 mV due to gate signal duty cycle regulation issues. The biggest pitfall for this project was the resulting sever in the project flowchart, the Card Reader module. The MFRC531 is an extraordinarily complex device with MCU interfacing options that are both plentiful and pinpoint precise. Of course, interfacing the RFID reader IC with the MCU is essential to the functionality of this project, and without that there is no full demo that can be presented. The NCS pin likely caused these issues and unfortunately wasn't caught before further close examination of the datasheet. Lastly, at the behest of ECE staff, no handheld cellular devices were given to the team for experimentation purposes regarding a native phone app. Thus, development of an Apple or Android app for this project simply was not possible, so testing the app on a laptop instead was the only effective course of action.

Throughout the construction of board itself, there were some user errors that may or may not have caused problems in the circuitry. Namely, the buck device intended for battery recharging had extremely small soldering pads and thus could've created a short or ineffective solder. Finally, spare parts here and there would've been nice; the pin connector for the LCD ended up having pin damage due to constant reheating and re-soldering on the board due to issues stated previously.

5.3 Ethical considerations

Regarding cheating, this is not a project that can be used anywhere but the home for personal use. There is no hope of users exploiting the casino directly by bringing this project on casino grounds and using it to their advantage in a game. This project demands the usage of a custom deck of cards with RFID tags stuck to each card, each embedded with its own identifying text file. The project is dysfunctional without this custom set of cards. Phone usage at a casino table is typically not allowed anyway, so using the phone app would also not be allowed. Straying from the usage of the project and rather focusing on the material that the user is being trained on, the knowledge obtained by training with the Odds Booster is all public knowledge and in no way, shape, or form is playing an optimal form of a casino game considered cheating. The intrinsic usage of this project does not constitute cheating once the knowledge is brought to the casino table.

Concerning gambling, this project only assists players who truly desire to play these games. As stated before, all knowledge intended to be obtained via the usage of this project is public and readily available. What makes this different is the physical usage and emulation of a real-time casino game on a table. This project does not give off the impression that you must go gamble otherwise your efforts are fruitless. On top of that, no gambling is required to play games with the Odds Booster. There are also no dollar values that are encouraged in the app's suggested actions, which would encourage users to gamble their money.

To apply all of this to the IEEE Code of Ethics [4], first, we must acknowledge what are the common laws that are broken when cheating in the casino. NRS 465.083 [5], according to Nevada law, prohibits players from cheating at a casino game, which is defined as the manipulation of the outcome of the game, or the payments made. Essentially, this is a fraud charge specifically citing it happening in the casino. The legal definition is stated as "alter[ing] the elements of chance, method of selection, or criteria which determine" the results, amount of payment, or frequency of payment in a game [5]; This relates to IEEE Code of Ethics I.4 [4]. As previously mentioned, this device cannot be used in a casino without the knowledge of the entire table, the dealer, and the casino staff.

Next, we examine the potential harm that this project can deliver to the players and public. This relates to IEEE Code of Ethics I.1 [4], and due to our planned usage of RFID and Bluetooth technologies, we are not transmitting or receiving any harmful signals that are outside the typical broadband spectrum of frequencies. There will be no cause of concern for the wireless communications occurring throughout the usage of this project. According to the FDA, there is no evidence of any adverse effects associated with the usage of passive RFID [6].

Furthermore, IEEE Code of Ethics I.3 [4] references conflicts of interest, which could potentially pertain to the casinos and their disliking to players increasing their odds of winning. Since this is purely an educational tool, there should be no issues with the casinos and the usage of this in the casino proper is virtually impossible thanks to our design of the project and tight security measures that are relevant to every casino. Since our project does not in any way put others down or engage in any sort of harassment or discrimination, the IEEE Code of Ethics II [4] is fulfilled as well. Lastly, taking a quick look at the Bluetooth Code of Conduct [7] will show that we are following each item and staying true to the ethics laid out. We are not violating any law, transmitting harmful content, or tampering, or sending excessively high-volume data transfers or bandwidth consumption.

5.4 Future work

Card counting is a strategy in blackjack that is *not* cheating but will often give casino staff a good excuse to exercise their right to expel players from the floor at any time for any reason. With that being said, as long as players don't get caught, what's the harm in *actually* beating the house odds? A very real improvement that could be made is integrating card counting into the code and showing the running count to the player on the app. There are also different rules for different tables, such as where the dealer will stand or different actions that users can take (EG surrender/insurance). These could be added into the code as a different version of the game. With that being said, there are even *more* edge cases that could be written into the code for suggested actions, which could also go into the code. Of course, Blackjack is not the only casino game that uses statistical probability. Texas Hold'em has intrinsic hand values and ranges with each possible iteration of a 7-card hand. Adding the game of Texas Hold'em would be a great way to give users an idea of how strong a given hand is, depending on the cards in the center and the two given to the user. For a real stretch, machine learning could be implemented to come up with its own suggested poker actions based on hundreds of millions of simulated hands.

Packaging is always something that can be cleaned up and repurposed for mass production. A 3D-printed box from the machine shop is not necessarily what consumers would want to pick up from the shelves of a store. Including proper packaging for the cards, LCD, battery, and PCB would be a great way to flirt with the idea of releasing this project to consumers. Another part of the project that could use some custom work is the antenna, which reads the RFID tags on the cards. There are always stronger readings that can be achieved, and with a more consistent antenna that can read from a further distance, that would be far less frustrating for the consumer to deal with when working with the Odds Booster in real time. This would require our own antenna design that more closely matches the Application Note for the MFRC531 [11].

References

[1] “S-8337/8338 Series Switching Regulator,” ABLIC, 2019.

[2] “LP3962/LP3965 1.5A Fast Ultra Low Dropout Linear Regulators”, *Texas Instruments*, April 2013.

[3] “IEEE Standard for Information Technology - Local and Metropolitan Area Networks - Specific Requirements - Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPAN),” *IEEE 802.15.1*, 2005.

[4] “IEEE Code of Ethics,” *IEEE Policies, Section 7 - Professional Activities*, 2020.

[5] “NRS 465.083 - Cheating at Gambling,” *Las Vegas Nevada Law*, 2014.

[6] “Radio Frequency Identification (RFID),” *U.S. Food and Drug Administration: Radiation Safety*, 2018.

[7] “Code of Conduct,” *Bluetooth SIG*, 2016.

[8] ECE 445 Course Staff, “ECE 445 Safety Guidelines,” 2022.

[9] ECE 445 Course Staff, “Safe Practice for Lead Acid and Lithium Batteries,” 2016.

[10] NXP Semiconductors, “MFRC531 Standard ISO/IEC 14443 A/B reader solution,” *nxp.com*, Jun. 30, 2015. https://www.nxp.com/docs/en/data-sheet/MFRC531.pdf.

[11] Philips Semiconductors, “Application Note: MF RC500 Active Antenna Concept,” *nxp.com*, Mar. 2002. https://www.nxp.com/docs/en/application-note/AN073811.pdf.

[12] “MIC2169 500kHz PWM Synchronous Buck Control IC”, *MICREL*, March 2009.

[13] G. Chang, “Super-Easy Simplified Blackjack Basic Strategy Chart,” Blackjack Calculation, 2014. http://blackjackcalculation.com/blackjack-super-easy-basic-strat.html (accessed Apr. 01, 2022).

Appendix A Requirement and Verification Tables

Table 5 Power Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. When the voltage of the lithium ion battery cell exceeds the overcharge protection voltage of 4.25 V (within a tolerance of ±0.05 V), the battery protection circuit must be able to disconnect the connected components and inhibit charging by turning off the charge control MOSFET.	1. To verify the correction functionality of the overcharge battery protection circuit, we can replace the lithium ion battery cell with a ramp DC voltage source, starting from the nominal 3.7 V battery voltage and slowly "ramping up" the voltage by 0.1 V increments to the rated 4.25 V overcharge protection voltage. During this test, we can probe the gate (voltage measurement), drain and source (current measurement) terminals using an oscilloscope to confirm that the charge control MOSFET enters the cutoff region, as intended. Further, we must also probe the load of this circuit and take a resistance measurement between the two output terminals to confirm there is a very high impedance (i.e. the load is disconnected from the circuitry).	Y
2. When the voltage of the lithium ion battery cell falls below the overdischarge protection voltage of 2.40 V (within a tolerance of ±0.05 V), the battery protection circuit must be able to disconnect the connected components and inhibit discharging by turning off the discharge control MOSFET.	2. To verify the correction functionality of the overdischarge battery protection circuit, we can replace the lithium ion battery cell with a ramp DC voltage source, starting from the nominal 3.7 V battery voltage and slowly "ramping down" the voltage by 0.1 V increments to the rated 2.40 V overdischarge protection voltage. During this test, we can probe the gate (voltage measurement), drain and source (current measurement) terminals using an oscilloscope to confirm that the discharge control MOSFET enters the cutoff region, as intended. Further, we must also probe the load of this circuit and take a resistance measurement between the two output terminals to confirm there is a very high impedance (i.e. the load is disconnected from the circuitry).	Y
3. The voltage regulator circuit must be able to maintain a voltage within a safe operating range of 3.3 V ± 0.1 V (safe operating range for average CMOS	3. Using a waveform generator, we can apply a step-wave that varies between 3 V and 3.7 V, with a DC offset of 3.3 V. Then, we can simply probe the output voltage of the LDO regulator to confirm that when the input voltage signal falls below 3.3 V, the output will follow the	Y

device [2]). If the input voltage falls below this range, the voltage regulator circuit must then match (follow) the input voltage signal to the LDO regulator. Otherwise, the LDO regulator must output a constant voltage 3.3 V.	input voltage signal. Any voltage ≥ 3.3 V, must output 3.3 V.	
4. The boost converter circuit must step up the lithium ion battery voltage of 3.7 V to a DC voltage of 9.6 V (± 0.2 V) as to provide enough power to the LED backlighting of the LCD display.	4. To verify the correction functionality of the boost converter circuit, we can probe the VIN and VOUT terminals using 2 channels on an oscilloscope. Then, we can confirm that the input and output voltages are at the intended values of 3.7 V and 9.6 V, respectively.	Y
5. The buck converter circuit must step down the USB charging port DC voltage from 5 V to 4.2 V ± (0.1 V).	5. To verify the correction functionality of the buck converter circuit, we can probe the VIN and VOUT terminals using 2 channels on an oscilloscope. Then, we can confirm that the input and output voltages are at the intended values of 5 V and 4.2 V, respectively.	N

Table 6 Card Reader Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
6. Tags must reliably be able to be read quickly and accurately, independent of the RFID Reader IC/Antenna portion of the module.	6. a. Either embed the tags in the middle of the card or stick the tags on the outside without severely affecting the overall thickness and feel of the cards. We have chosen tags that will be thin and flexible enough to be almost unnoticeable from a different deck's dimensions. Each deck will be stress-tested with multiple shuffles and leafings. b. Phone apps are able to pick up these RFID tags, courtesy of Avery-Dennison (the company selling the RFID tags) and we will ensure that these tags work properly upon purchase and delivery.	Y

7. Reliably pick up 13.56 MHz signals from 17 mm distance with little to no reflection.	7. Measure the antenna VSWR using a spectrum analyzer and signal generator to find out how much energy sent to the antenna is reflected back. The s-parameter simulation will give an idea of how well the antenna is impedance matched to the transmission line. A VSWR under 2 will be suitable for this application.	N
8. Ensure the internal impedances of the RFID Reader IC (which can be set between 1 and 100 Ω) and the RF Antenna (50 or 80 Ω) will have matched loads along the transmission line.	8. Run simulations using ADS plugging in terminals at both ends representing the antenna and IC and with the lumped components in between. Tolerances of 10% and 5% will be tested to determine the quality of components we will need.	Y
9. Properly transmit RFID tag info to MCU at Pin 5 at 13.56 MHz.	9. Probe pin 5 to an oscilloscope and scan RFID tag for the reader to pick up. Note the tag scanned and recognize its reading value on the oscilloscope at 13.56 MHz.	N

Table 7 Communications Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
10. The communication subsystem must use Bluetooth v4.2 to transmit and receive information to and from the phone app at a distance of 3 ft with an average throughput of 5 kbps and a bit error rate of less than 1%.	10. a. The MCU Firmware will be loaded onto the assembled device using the UART connection. A basic function will be implemented to create a 5 kb binary file holding pseudo-random bytes in a deterministic manner. The MCU will use the ESP32 Bluetooth API to transmit packets and receive acknowledgements b. The central PCB will be placed 3 ft away from a device containing the phone app. The phone app functionality is not important in this test, it simply must be capable of receiving Bluetooth packets, saving the information, and sending acknowledgements. The same data generation function will be provided to the phone app to be able to check for errors. c. The full 5 kb file will be transmitted and the total time will be recorded to obtain an average throughput measure.	Y

		The Bluetooth protocol often accounts for many bit errors through correction schemes, but the generated file on the phone app will be used to check bit error rate	

Table 8 Display Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
11. The microcontroller must generate both 6.5 MHz (±0.2 MHz) and 52 MHz (±2 MHz) clock signals simultaneously through its clock output pins.	11. a. Connect the ESP32-WROVER-E module to a 3.3 V power supply. b. Program the MCU in download mode using the UART interface and set the GPIO control registers to have CLK OUT2 at pin 35 and CLK OUT3 at pin 34 with frequencies 52 MHz and 6.5 MHz respectively. c. Connect these pins to separate channels of an oscilloscope. Use the frequency analysis features of the oscilloscope to record the frequency of each output signal.	N
12. The microcontroller must output the screen data serially to the three specified GPIO pins with a clock rate of 52 MHz (±2 MHz).	12. a. Connect the ESP32-WROVER-E module to a 3.3 V power supply. b. Program the MCU in download mode using the UART interface and set the GPIO control registers so pins 6,7, and 8 are the RGB output pins respectively. Provide a sample 8-bit single channel color value (xAA) to be sent serially to the output pins at a rate of 52 MHz. c. Connect the output pins to separate channels of an oscilloscope. The selected sample value requires the changing of pin value at each clock period. Use the frequency analysis features of the oscilloscope to record the frequency of each output signal to ensure 52 MHz operations.	Y
13. The microcontroller must write to external PSRAM to update displayed screen data while display is active.	13. a. With device fully constructed (battery connected, display connected). Program the MCU in download mode using the UART interface to contain a	Y

	single buffer in PSRAM full of a single background tone. Create a function to move a contrasting square to different positions in the buffer. b. Run the device using the PSRAM as the memory input to the display. Call the function repeatedly during operation. Observe the square on the display to ensure intended output matching the buffer location in the PSRAM.	

Table 9 App Subsystem Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
14. The app will utilize the Bluetooth capabilities of the device to send and receive data with the central device.	14. A sample "Hello World!" packet will be created and sent to the central device. The central device will respond with an acknowledgement. The phone app must be able to receive this acknowledgement ensuring both transmit and receive operations are functional.	Y
15. The app will use card information to accurately calculate the "strength" measure or the percent of possible alternate hands against which the user's hand will win.	15. A random game setup will be created with two cards provided to the user and a total of five cards available to all players. The algorithm will be tested by providing it this information and the output strength measure will be generated using the first three commonly available cards, then the first four, and finally all the commonly available cards (similar to how a Texas Hold'em game would progress). The algorithm will log all available hands at each stage and the percentile of the user's hand. These outputs will be verified manually by the testers to ensure proper operation.	Y
16. The app will display the measures and suggestions in the form of text for the user.	16. The software will combine the Bluetooth and strength measure calculations into a working app. The app will be tested in a game situation and card IDs will be sent from the central device. The app must display this information in an understandable manner to an average user (someone not familiar with the design of the device).	Y

Appendix B Design Schematics

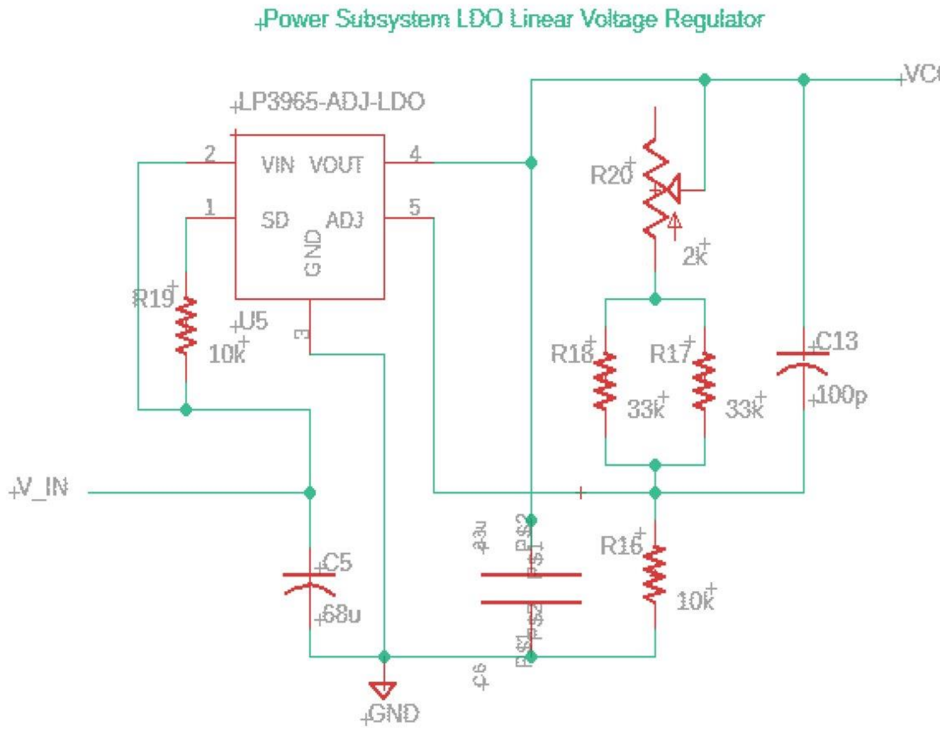

Figure 9: Power Subsystem Linear Voltage Regulator Schematic

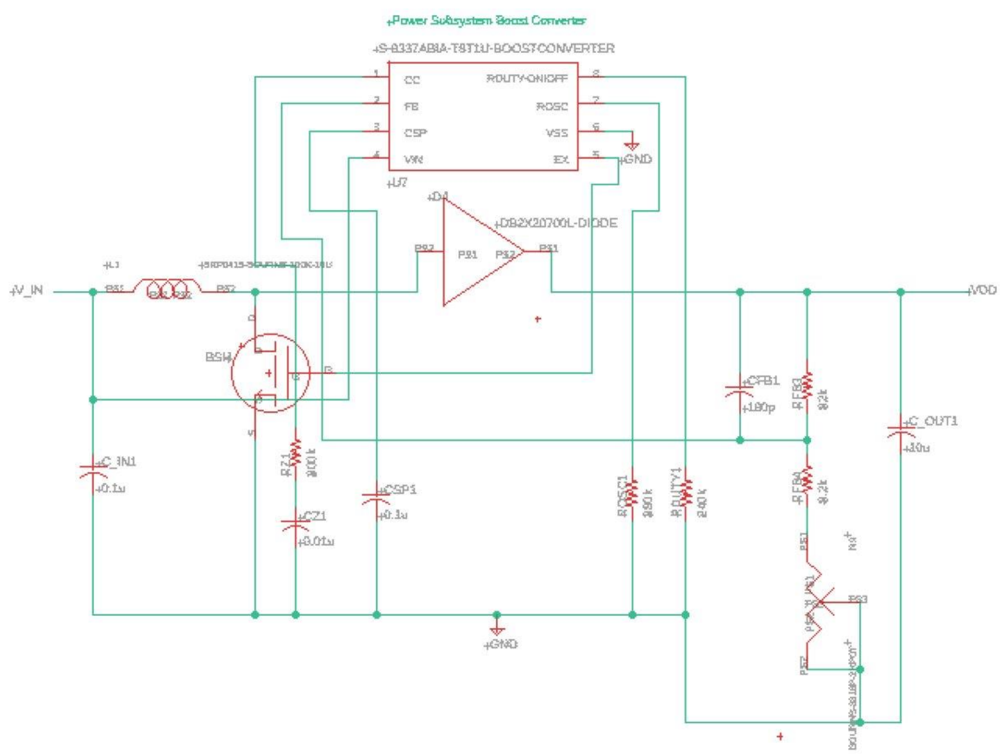

Figure 10: Power Subsystem Boost Converter Schematic

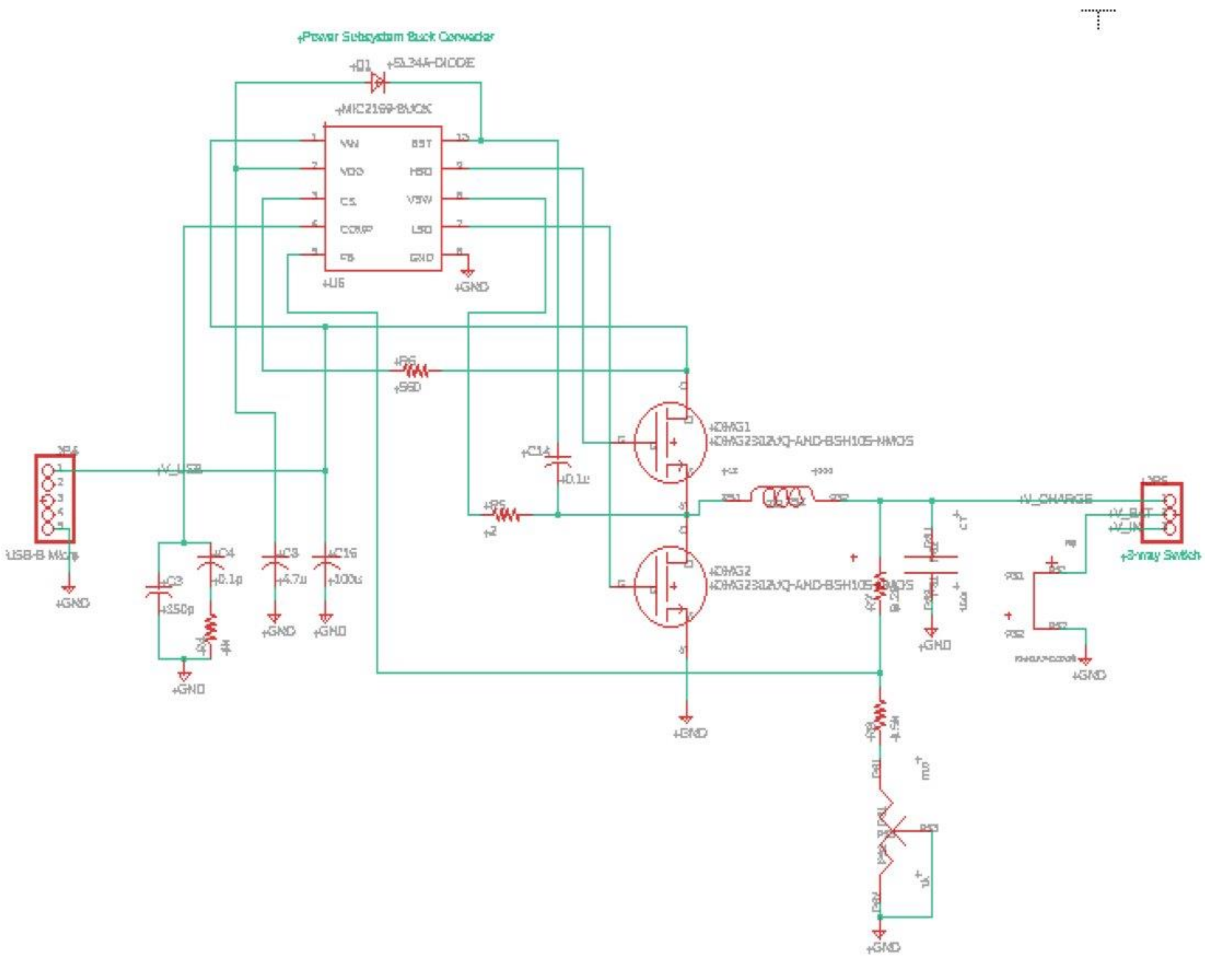

Figure 11: Power Subsystem Buck Converter Schematic

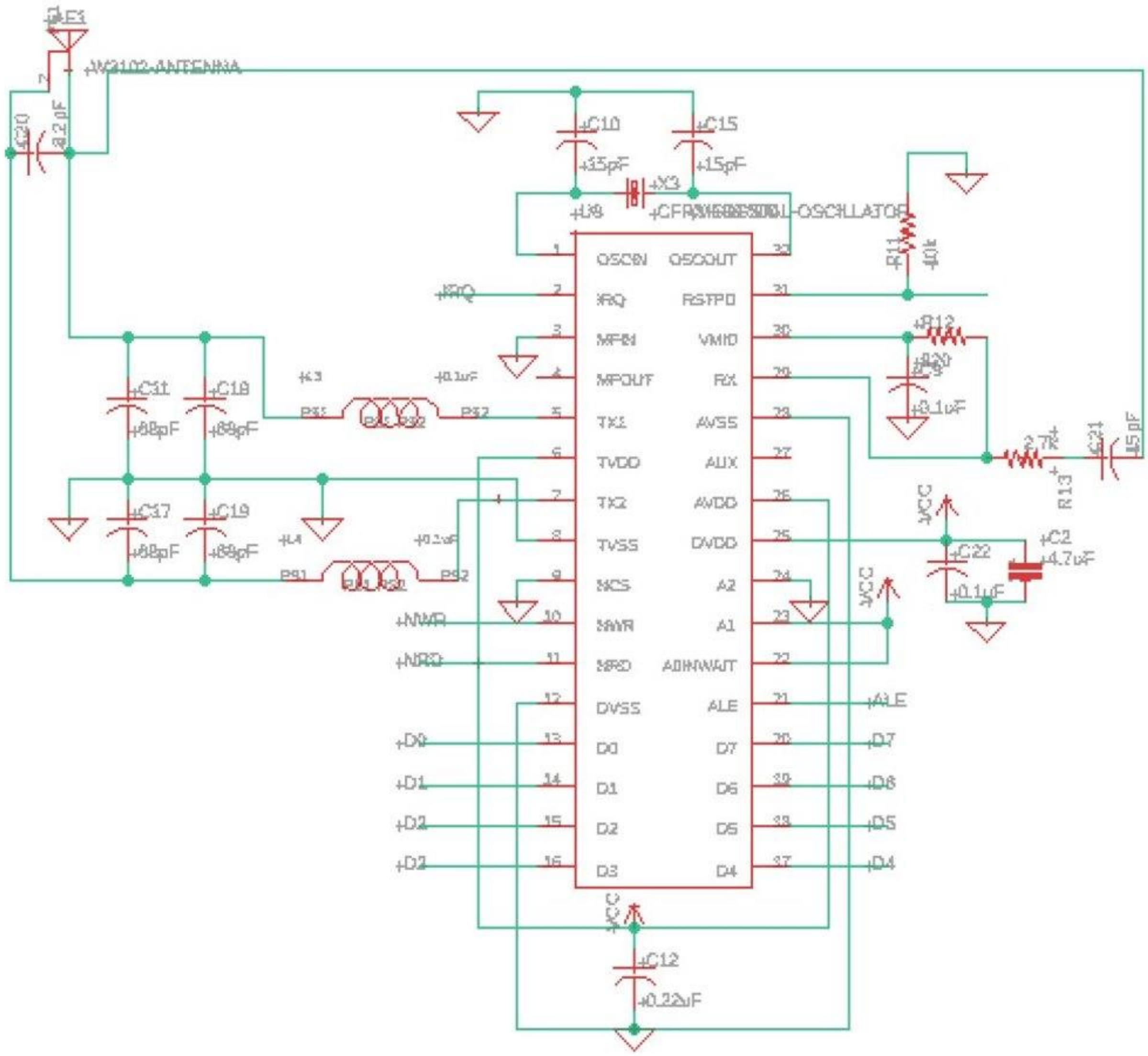

Figure 12: Card Reader Subsystem Schematic

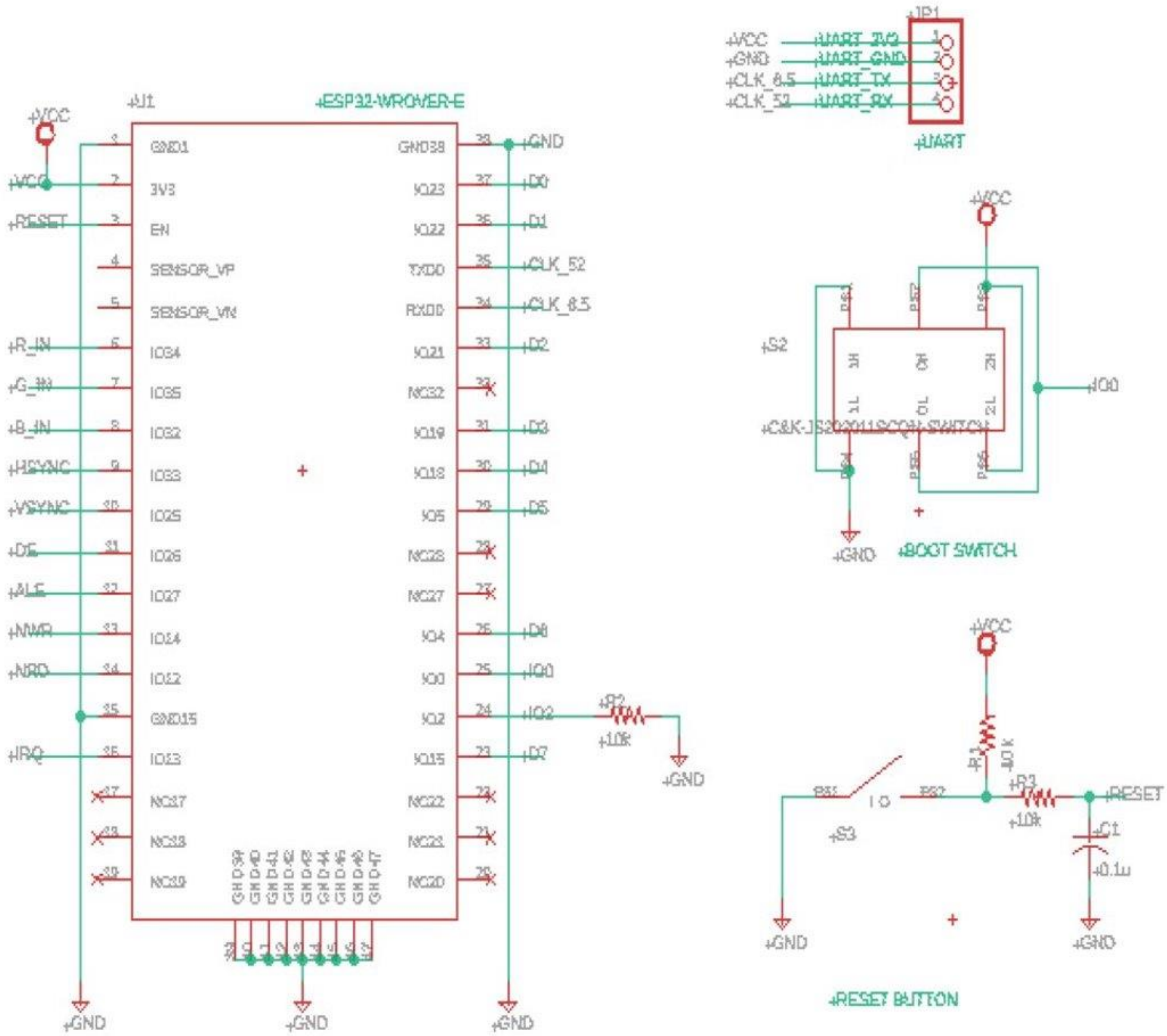

Figure 13: Communications Subsystem Schematic

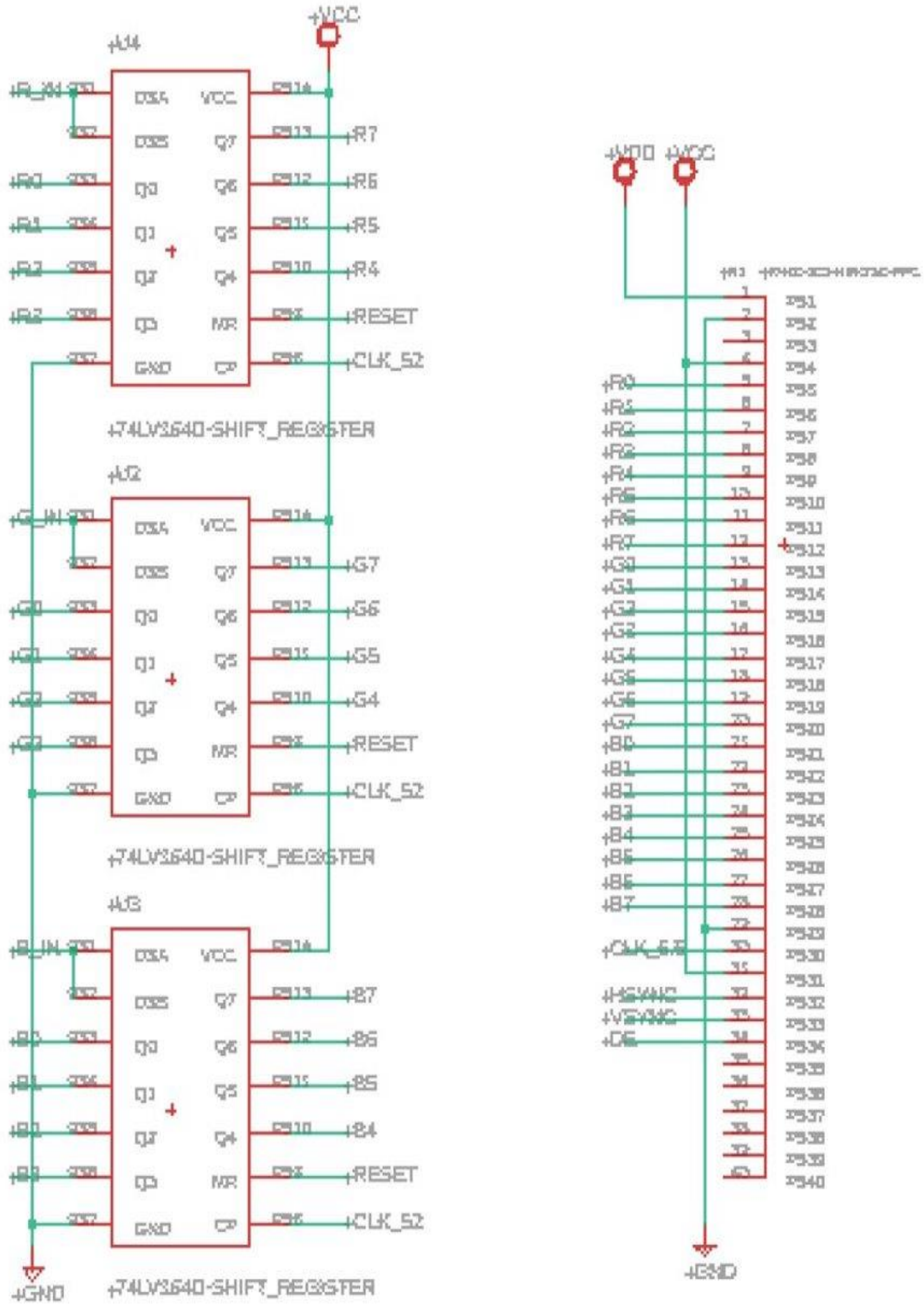

Figure 14: Display Subsystem Schematic

Appendix C Odds Booster Final Implementation

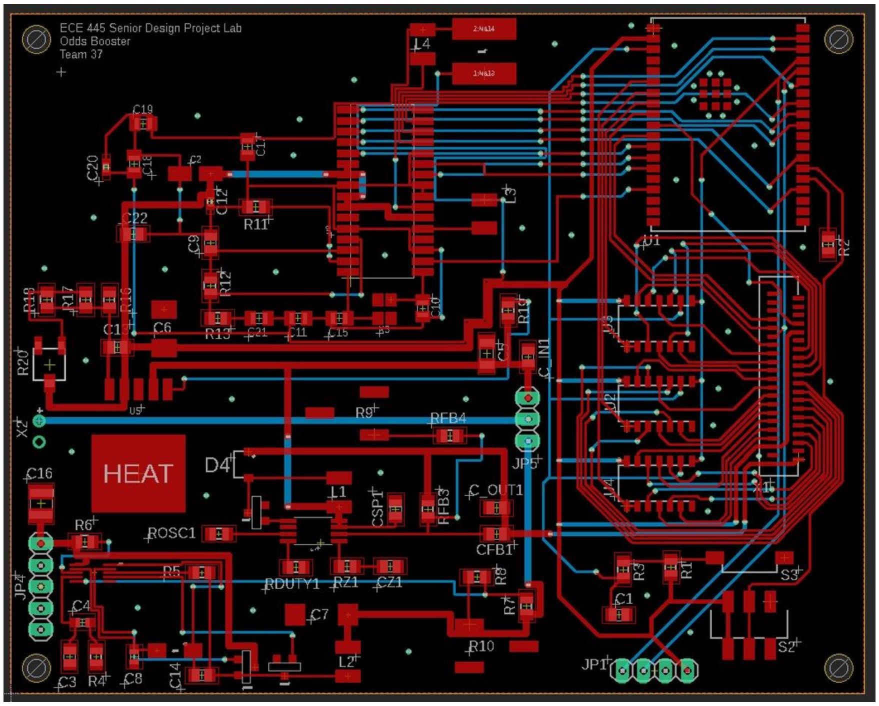

Figure 15: Final PCB layout

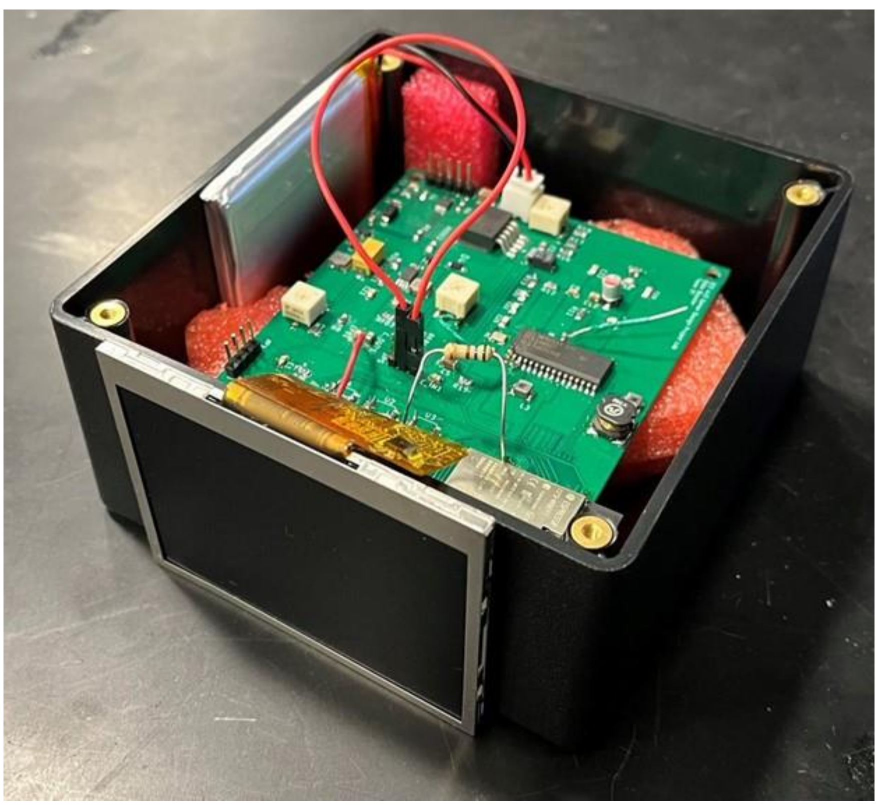

Figure 16: Odds Booster picture of final implementation

Appendix D LCD Signal Oscilloscope Readings

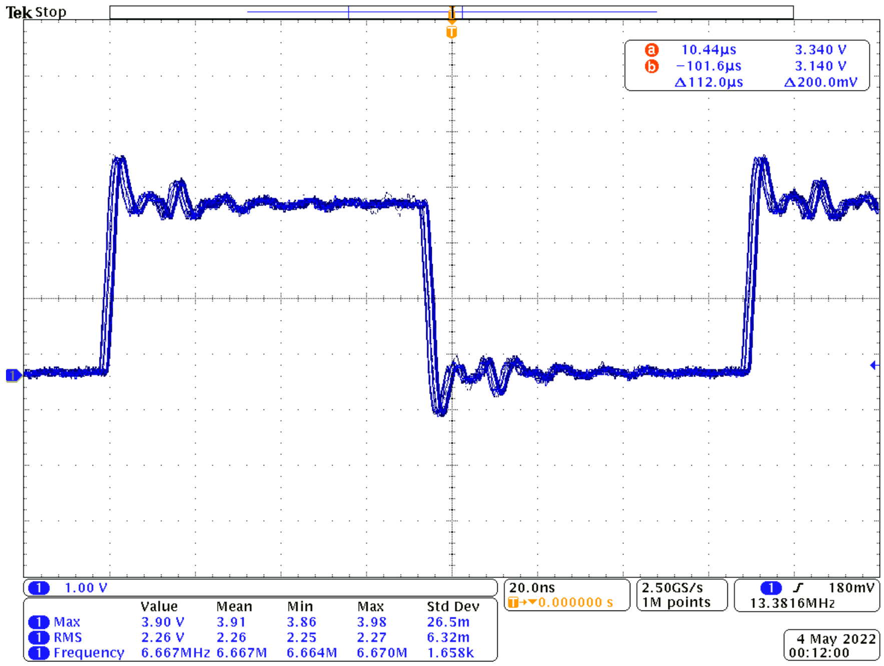

Figure 17: LCD Pixel Clock Output Signal

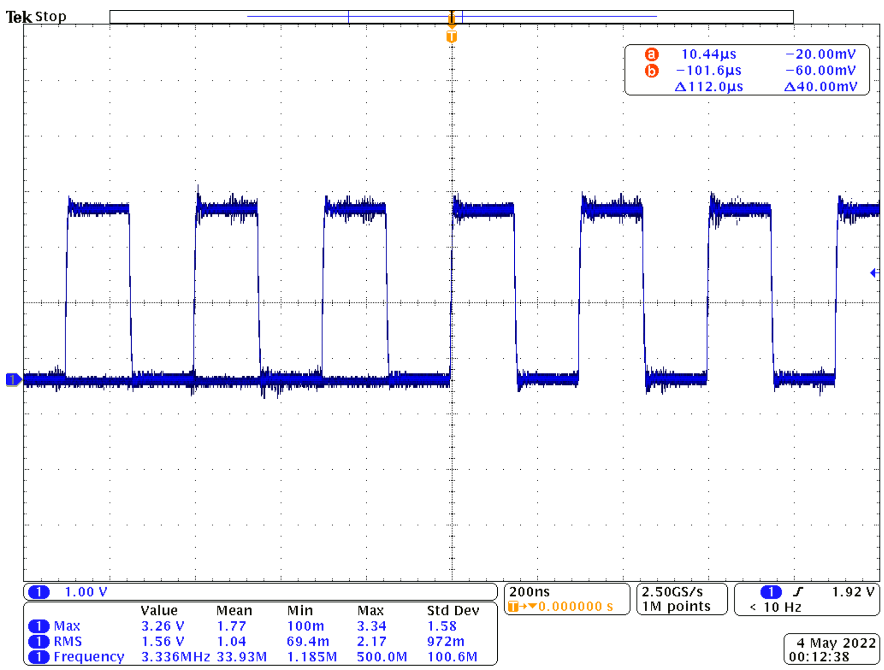

Figure 18: LCD Alternating Data Output Signal

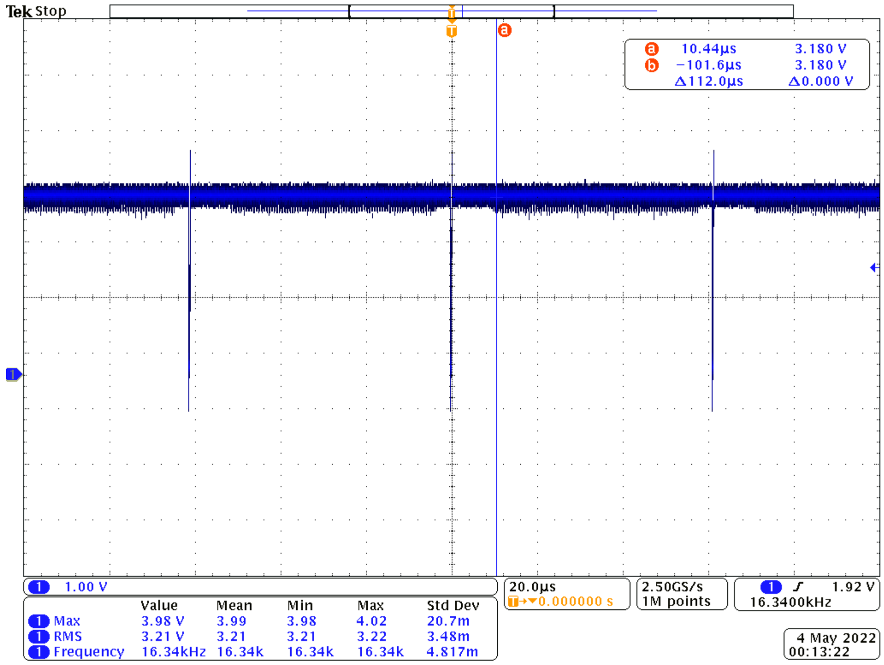

Figure 19: LCD H-Sync Output Signal

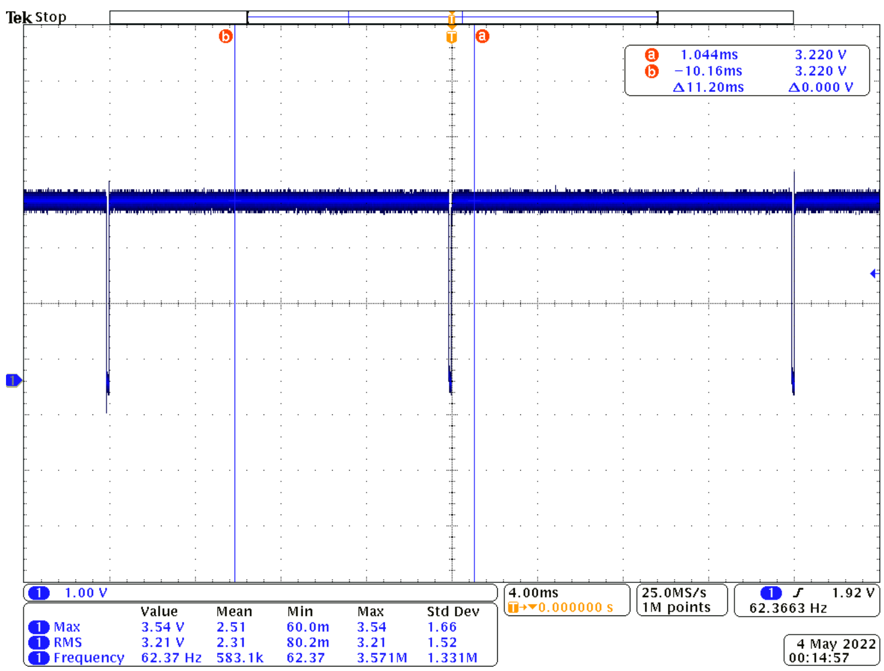

Figure 20: LCD V-Sync Output Signal

Appendix E Parts Cost Breakdown

Table 10 Cost Breakdown of Purchased Components

Part	Part Number	Retail Cost ($)	Qty	Actual Cost ($)
Li Battery	Mikroe-4474	14.50	1	14.50
Battery Connector	JST B2B-XH-A(LF)(SN)	0.15	2	0.30
USB Micro-B Port	GCT USB3505-30-A-KIT	2.81	1	2.81
On-Off-On Switch	Nidec Copal 8SS1022-Z	4.17	1	4.17
Header Pins	Wurth 61300511121	0.26	1	0.26
Boost Gate Driver	ABLIC S-8337ABIA-T8T1U	1.52	1	1.52
10µH Inductor	Bourns SRP0415-100K	1.12	1	1.12
Diode	Panasonic DB2X20700L	0.25	2	0.50
N-MOS MOSFET	Infineon BSS214NH6327XTSA1	0.38	2	0.76
2kΩ Potentiometer	Bourns 3361P-1-202GLF	1.27	2	2.54
Passive Boost Components	N/A	-	18	2.32
Linear Voltage Regulator	TI LP3965ESX-ADJ/NOPB	4.62	1	4.62
68µF Capacitor	TDK C3216X5R0J686M160AB	0.85	1	0.85
33µF Tantalum Cap.	Nemco PCT33/10CK	0.26	1	0.26
Passive Voltage Regulator Components	N/A	-	8	0.98
Buck Gate Driver	Microchip MIC2169YMM	2.52	2	5.04
N-MOS MOSFET	Diodes DMG2302UQ-13	0.44	5	2.20
3.3µH Inductor	Bourns SRP0415-3R3K	1.12	1	1.12
100µF Capacitor	Samsung CL32A107MPVNNNE	1.14	1	1.14
150µF Tantalum Cap.	KEMET T491X157K020AT	2.53	1	2.53
1kΩ Potentiometer	Bourns 3361P-1-202GLF	1.27	1	1.27
Diode	Surge SL34A	0.34	2	0.68
Passive Buck Components	N/A	-	9	1.88
Microcontroller	ESP32-WROVER-E	3.60	1	3.60
Development Board	ESP32-DEVKITC-VE	11.00	1	11.00
USB to UART	Paialu CP2102	8.76	1	8.76
On-On Switch	C&K JS202011SCQN	0.55	1	0.55
Pushbutton	C&K PTS636 SL43 SMTR LFS	0.12	2	0.24
Header Pins	Wurth 61300411121	0.19	3	0.57
Passive Communications Components	N/A	-	8	0.98
RFID Reader IC	NXP MFRC53101T/0FE,112	24.28	1	24.28
13.56MHz Antenna	PulseLarsen Antennas W3102	1.77	2	3.54
13.56MHz Xtal	Wurth 830082070	0.70	2	1.40
0.1µH Inductor	Bourns SRP0415-0R1K	1.12	3	3.36
4.7u Electrolytic Cap.	Wurth 865080640004	0.18	2	0.36
Passive Card Reader Components	N/A	-	20	2.22

LCD	Orient AFY320240A0-3.5N6NTNR	11.53	1	11.53
FCP Connector	Hirose FH40-40S-0.5SV	2.75	2	5.50
Shift Register	Nexperia 74LV164D	0.43	4	1.72
		Total	119	132.98