

# **Tabletop Fantasy Football**

ECE 445 Design Document - Fall 2026

Project # 54

Parthiv Adimulam, Nicholas Goo, Raahil Parikh

Professor: Cunjiang Yu

TA: Xiyue Zhu

October 2nd, 2026

# Contents

|                                                       |           |
|-------------------------------------------------------|-----------|
| <b>1 Introduction</b>                                 | <b>4</b>  |
| 1.1 Problem                                           | 4         |
| 1.2 Solution                                          | 4         |
| 1.3 Visual Aid                                        | 5         |
| 1.4 High-Level Requirements                           | 5         |
| <b>2 Design</b>                                       | <b>6</b>  |
| 2.1 Physical Design                                   | 6         |
| 2.2 Block Diagram                                     | 8         |
| 2.3 Functional Overview & Block Diagram Requirements  | 8         |
| 2.3.1 Main Control Subsystem                          | 8         |
| 2.3.2 NFC Reader Subsystem                            | 10        |
| 2.3.3 User Interface Subsystem                        | 11        |
| 2.3.4 Display Subsystem                               | 12        |
| 2.3.5 Visual Feedback Subsystem                       | 14        |
| 2.3.6 Audio Feedback Subsystem                        | 15        |
| 2.3.7 Power Management Subsystem                      | 18        |
| 2.3.8 External Interface Subsystem                    | 19        |
| 2.4 Hardware Design                                   | 21        |
| 2.4.1 Power Architecture, Battery Sizing & Regulation | 21        |
| 2.4.2 Battery Charging & Protection                   | 22        |
| 2.4.3 NFC Reader Hardware                             | 22        |
| 2.4.4 Display, LED, Audio, and Input Hardware         | 22        |
| 2.5 Software Design                                   | 23        |
| 2.5.1 System State Machine                            | 23        |
| 2.5.2 Card Scanning & Lineup Management               | 23        |
| 2.5.3 Live Statistics, Scoring & Event Detection      | 24        |
| 2.5.4 Feedback and User Input                         | 24        |
| 2.5.5 Connectivity and Security                       | 25        |
| 2.5.6 Data Interfaces                                 | 25        |
| 2.6 Commercial Component Selection                    | 25        |
| 2.7 Tolerance Analysis                                | 26        |
| 2.7.1 NFC Read Range and Isolation                    | 27        |
| 2.7.2 Card Scan Time                                  | 27        |
| 2.7.3 Score Update Latency                            | 27        |
| 2.7.4 Power Rail Capability at Peak Load              | 28        |
| <b>3 Cost Analysis</b>                                | <b>28</b> |
| 3.1 Cost for Project Group Labor                      | 29        |
| 3.2 Cost for Parts                                    | 29        |
| 3.3 Cost for Equipment, Facilities, and Shop Services | 31        |

|                                               |           |
|-----------------------------------------------|-----------|
| 3.4 Grand Total, Sum of Costs                 | 31        |
| <b>3.5 Schedule</b>                           | <b>31</b> |
| <b>3.6 Risk Analysis</b>                      | <b>32</b> |
| <b>4 Ethics and Safety Considerations</b>     | <b>33</b> |
| 4.1 Societal Impact                           | 33        |
| 4.2 Applicable Engineering Standards          | 34        |
| 4.3 Ethical Considerations                    | 34        |
| 4.4 Electrical and Mechanical Safety Concerns | 34        |
| 4.5 Safety Procedures and Risk Mitigation     | 35        |
| <b>References</b>                             | <b>36</b> |

# 1 Introduction

## 1.1 Problem

Fantasy football is a popular way for sports fans to compete with friends while following professional football. The Fantasy Sports and Gaming Association reports that more than 53 million people participate in fantasy sports annually [1]. However, the traditional experience is designed around digital platforms and long-term leagues. Players typically create or join a league, coordinate a draft, manage a roster, and stay engaged for an entire NFL season with commitments as high as multiple times per week. This works well for dedicated players, but introduces friction for people who simply want to compete with a friend while watching football for a single afternoon.

For example, two friends watching NFL games together may want to pick players and compete on how those players perform that day. With existing platforms, doing so requires a website or mobile application and a league or contest to be configured before play begins. Much of the experience also happens individually on phones or computers even when the participants are in the same room. Physical scoreboards can display sports information and traditional board games that are sports-themed provide face-to-face interaction and engagement, but neither combines physical gameplay with automatically updated statistics from live NFL games. Our project addresses this gap with a more immersive and tangible form of fantasy football. Its societal relevance is primarily social and cultural: it offers another way for people to interact while taking part in a form of sports entertainment already enjoyed by millions.

## 1.2 Solution

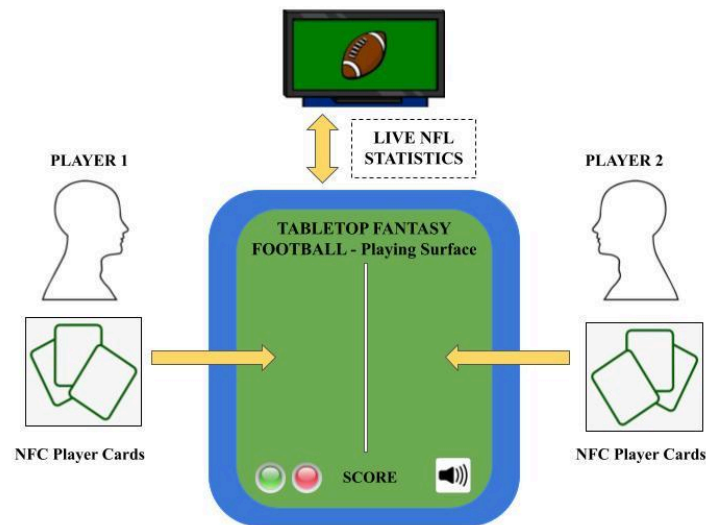
Our solution is a battery-powered Tabletop Fantasy Football system that lets two users play a head-to-head matchup using physical player cards while watching live NFL games. Each physical card contains a passive NFC tag identifying an NFL player, and the board has one NFC reader for each team. To build a lineup, a user scans a card on their team's scan pad and then places it in the labeled slot for that position. A lineup has five required positions: one quarterback (QB), one running back (RB), one wide receiver (WR), one tight end (TE), and one defense (DEF). The system identifies each scanned card, and the backend software assigns the player to the correct lineup position automatically. If a user scans a second card for a position that is already filled, the newest scan replaces the earlier one. When a team is satisfied with its lineup, the user presses that team's Select button to lock it. After both teams have locked valid lineups, the Start button begins the matchup. This keeps the interaction physical, like a tabletop game, instead of requiring a phone or computer.

An ESP32 microcontroller with Wi-Fi serves as the primary controller. Live NFL player statistics are obtained from the SportRadar NFL API [2], which provides player and team-level game statistics in real time. The system converts these statistics into a fantasy score for each user using configurable scoring rules and updates the matchup automatically throughout the games. A 16x2 LCD shows the current scores and setup state, ten RGB LEDs provide player-specific visual feedback, and a speaker provides audio for events such as touchdowns and successful or rejected scans. The board uses three primary controls - Team 1 Select, Start, and Team 2 Select - so normal gameplay does not require a phone or companion application.

The novelty of our solution comes from combining physical tabletop interaction with automatically updated fantasy-football scoring in a single self-contained device. Compared with the conventional digital

workflow described above, users interact with physical NFC-enabled player cards rather than repeatedly using a phone or computer to manage a lineup and view scores. At the same time, unlike a static sports-themed board game, the board remains connected to live NFL data and automatically converts player statistics into fantasy points. The combination of NFC-based lineup entry, live statistics, player-specific RGB feedback, audio feedback, and an onboard display is intended to create a single-session fantasy-football experience that remains primarily face-to-face and physical during normal gameplay.

### 1.3 Visual Aid



*Figure 1: Intended use of the tabletop fantasy football system, where two players use NFC-enabled player cards to compete in a matchup that updates automatically using live NFL statistics.*

### 1.4 High-Level Requirements

Because the primary problem is the friction of traditional fantasy football leagues, the system must provide a convenient, responsive, and gamified experience. The high-level requirements therefore emphasize latency, setup time, and detection accuracy. To consider the project successful, the system must fulfill the following:

- The system shall detect and identify an NFC player card scanned at a team's scan pad within 5 seconds, with a success rate of at least 95%, and assign the player to the correct lineup position for that team.
- The system shall update the displayed fantasy score within 15 seconds of the corresponding player-statistic update becoming available from the live NFL data source, for at least 90% of tested scoring events.
- The system shall operate continuously for at least 90 minutes of active gameplay without requiring a manual reset or losing the current lineup, fantasy score, or matchup state.

## 2 Design

### 2.1 Physical Design

The board is a flat tabletop surface measuring approximately 500 mm wide by 420 mm deep. It is divided into a 110 mm-deep upper control section and a 310 mm-deep lower play area. The upper section contains a 100 mm x 110 mm NFC region for Team 1, a 300 mm x 110 mm center control region, and a 100 mm x 110 mm NFC region for Team 2. Each NFC region contains a marked 70 mm x 70 mm scan zone with an NFC reader and antenna mounted beneath it. The center control region contains the 16x2 LCD and the Team 1 Select, Start, and Team 2 Select buttons.

The lower play area is divided into two 240 mm-wide team areas separated by a 20 mm center divider. Each team area contains five standard playing-card slots arranged in two rows: QB, WR, and RB on the upper row, and TE and DEF on the lower row. The cards are standard 63.5 mm x 88.9 mm playing-card size and rest in shallow holders. Each slot has one addressable RGB LED located below it. The speaker is centered near the bottom of the board with a nominal 45 mm grille opening. The Li-Po battery, power electronics, ESP32, NFC electronics, and wiring mount beneath the board. The card slots are placement areas only and do not contain NFC readers; a user scans a card at the team's scan zone and then places it in the matching slot.

**Table 1: Board dimensions**

| Dimension                | Nominal Value                                            |
|--------------------------|----------------------------------------------------------|
| Overall board            | 500 mm x 420 mm                                          |
| Upper control section    | 500 mm x 110 mm; 100 / 300 / 100 mm split                |
| Lower play area          | 500 mm x 310 mm; 240 / 20 / 240 mm split                 |
| NFC marked scan zone     | 70 mm x 70 mm on each side                               |
| Player card / slot basis | 63.5 mm x 88.9 mm standard card                          |
| LCD / speaker opening    | LCD approx. 80 mm x 36 mm;<br>speaker approx. 45 mm dia. |

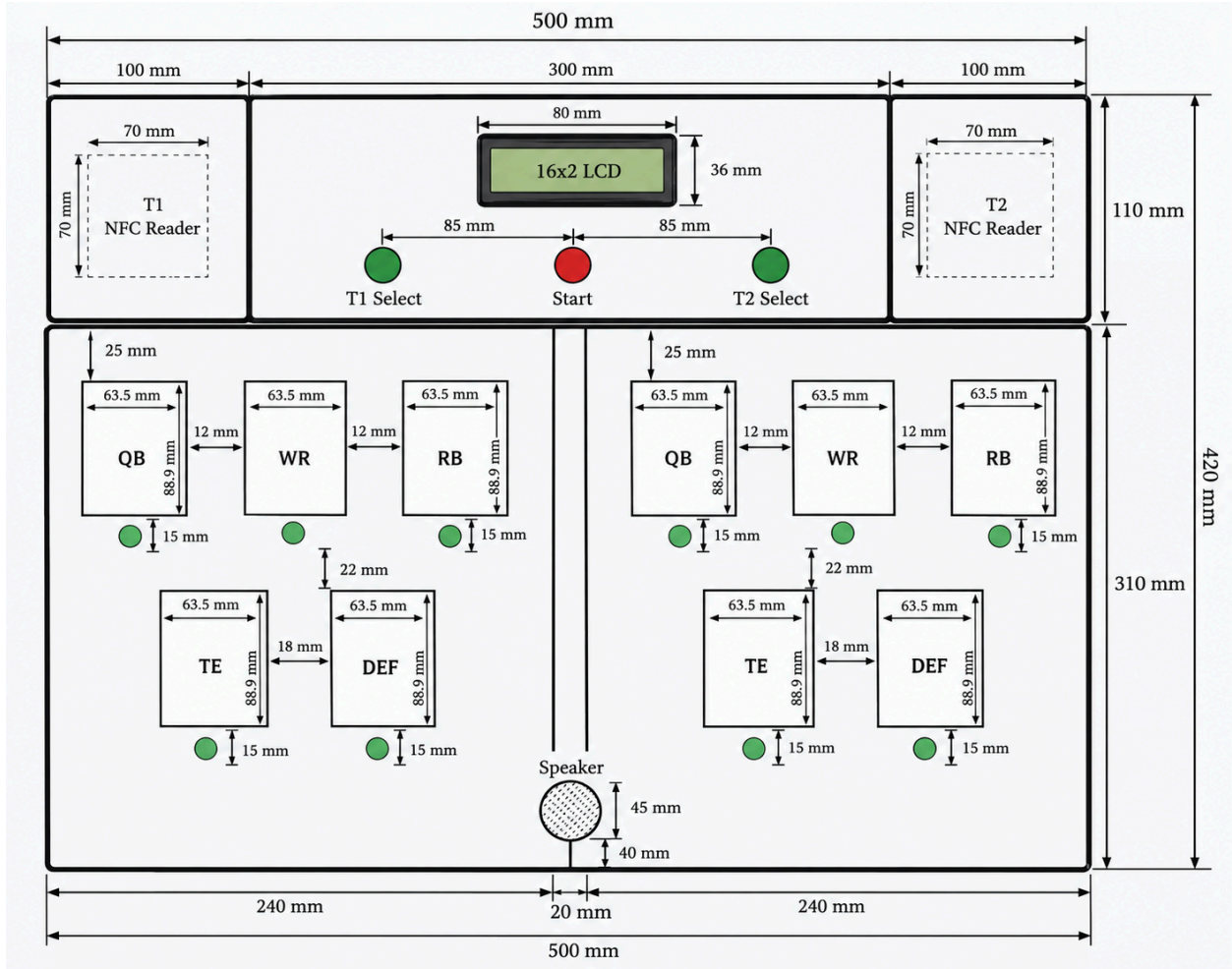


Figure 2: Top view of the planned board layout.

## 2.2 Block Diagram

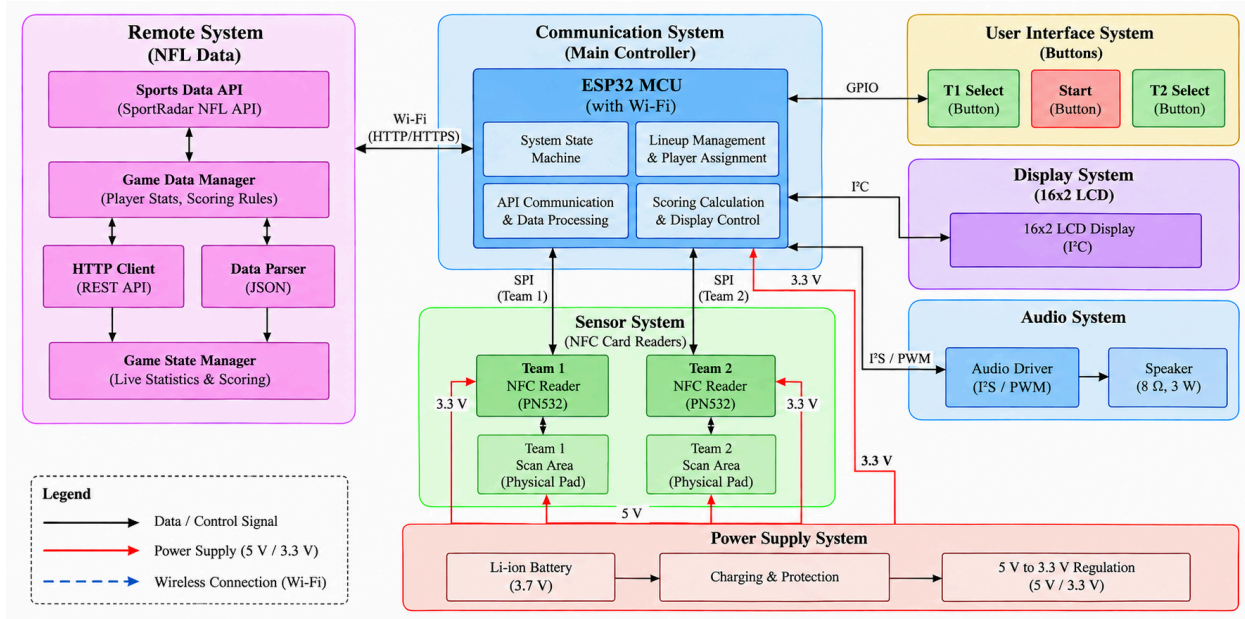


Figure 4: Tabletop Fantasy Football block diagram, showing the major subsystems, power connections, and communication interfaces.

The system is organized into modular subsystems centered around the Main Control Subsystem. The Power Management Subsystem provides regulated 3.3 V and 5 V rails to the electronics. The Main Control Subsystem communicates with the two NFC readers over SPI, communicates with the Display Subsystem over I<sup>2</sup>C, reads the Team 1 Select, Start, and Team 2 Select buttons over GPIO, drives the Visual Feedback Subsystem using a level-shifted LED data signal, and generates a PWM signal for the Audio Feedback Subsystem. The two external information sources are the passive NFC player cards, which are read at 13.56 MHz, and the live sports-data backend, which is accessed by the ESP32 over 2.4 GHz Wi-Fi using HTTPS. Separating these functions into independent subsystems allows the NFC, user-interface, display, visual-feedback, audio, power, and software functions to be developed and verified independently before full-system integration.

## 2.3 Functional Overview & Block Diagram Requirements

### 2.3.1 Main Control Subsystem

The Main Control Subsystem is the central processing unit of the system and is implemented with an ESP32 microcontroller with integrated Wi-Fi. It receives player identifiers from the two NFC readers following card scans, reads the Team 1 Select, Start, and Team 2 Select buttons, retrieves live statistics from the sports-data backend over Wi-Fi, and controls the Display, Visual Feedback, and Audio Feedback Subsystems. Its lineup-building software looks up each scanned card in a configured database in memory, assigns the player to the correct position in the scanning team's lineup, and replaces an earlier card when a second card for the same position is scanned. A team's Select button locks that lineup only after all five required positions are filled. After both lineups are locked, the Start button transitions the system into live gameplay. The controller then calculates fantasy scores according to the selected scoring configuration

and decides when events such as touchdowns, catches, fumbles, or field goals should trigger feedback. Software design is detailed in Section 2.5.

**Contribution:** Provides the logic backbone that turns scans, button presses, and live statistics into lineups, scores, and feedback for players.

**Interfaces:**

- SPI to the two PN532 NFC readers using 3.3 V logic, with the readers sharing clock and data lines and using separate chip-select signals. The SPI clock will be limited to 5 MHz or less in integration.
- I<sup>2</sup>C communication to the 16×2 LCD subsystem using 3.3 V logic.
- 2.4 GHz Wi-Fi using HTTPS to communicate with the sports-data API, with an approximately 5 second request timeout.
- Three active-low GPIO inputs from the Team 1 Select, Start, and Team 2 Select buttons. Each input is externally pulled up to 3.3 V and is debounced in firmware.
- One GPIO data output to the addressable RGB LED chain through a 3.3 V-to-5 V level shifter.
- One PWM audio output from the ESP32 to the audio filtering and amplification stage.
- After a valid NFC UID is received, lineup processing and the corresponding LCD, LED, and audio feedback shall begin within 1 second.

**Table 2: Main Control Subsystem - Requirements & Verification**

| Requirements                                                                                                                                                                                                                                                                  | Verification                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• The ESP32 must communicate with the NFC readers, LCD, LEDs, audio system, buttons, and Wi-Fi backend simultaneously without resets or loss of state.</li> </ul>                                                                      | <ul style="list-style-type: none"> <li>• Run all peripherals concurrently for 10 minutes with the serial log (115200 baud) recording the reset reason and a once-per-second response from every peripheral.</li> <li>• <b>Pass if:</b> there are zero watchdog or brown-out resets and every peripheral responds every second.</li> </ul>                                                                                                                                                                                                                            |
| <ul style="list-style-type: none"> <li>• A newly received sports-data update must be processed and reflected on the board within 15 seconds of receipt.</li> </ul>                                                                                                            | <ul style="list-style-type: none"> <li>• Send 100 simulated updates from a laptop test server containing known player events.</li> <li>• Toggle one GPIO when the update is received and another when the LCD and LED update completes; measure the interval with a logic analyzer or oscilloscope.</li> <li>• <b>Pass if:</b> all 100 intervals are 15 s or less (design target: 1 s).</li> </ul>                                                                                                                                                                   |
| <ul style="list-style-type: none"> <li>• The controller must assign each scanned card to the correct position in the scanning team's lineup, replace an earlier card when a second card for the same position is scanned, and refuse to lock an incomplete lineup.</li> </ul> | <ul style="list-style-type: none"> <li>• Scan 25 cards covering all positions for each team and read the resulting lineup from the serial log.</li> <li>• Scan a second card for an already-filled position in 10 trials.</li> <li>• Press each team's Select button with a position empty in 10 trials and verify that the lineup remains unlocked.</li> <li>• <b>Pass if:</b> all 50 cards are assigned to the correct position, the newest scan replaces the earlier card in 10 of 10 trials, and the incomplete lineup is refused in 10 of 10 trials.</li> </ul> |

| Requirements                                                                                                                                                                                     | Verification                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>The controller must associate each incoming game event with the correct player and lineup position before activating visual or audio feedback.</li> </ul> | <ul style="list-style-type: none"> <li>Lock known lineups for both teams and inject 30 events (touchdown, reception, fumble) for known players.</li> <li><b>Pass if:</b> the score changes for the correct team in 30 of 30 events and only the LED at that player's slot responds in at least 27 of 30 events (90%).</li> </ul> |

### 2.3.2 NFC Reader Subsystem

The NFC Reader Subsystem provides the physical player-identification mechanism for the game. The board has two readers, intended for one to be used per team. Each reader and its antenna sit beneath a marked scan pad and operate at 13.56 MHz. Each player card holds a passive NFC tag corresponding to one NFL player through its unique identifier (UID). When a card is held on a pad, the reader reads the tag identifier and sends it, together with the team (reader) number, via SPI to the Main Control Subsystem containing the ESP32. Cards resting in the slots are not actively tracked once they are scanned into the line-up and finalized.

For this subsystem, there were several design decisions and considerations that had to be made. We originally proposed use of 10-12 discrete NFC readers, but discovered that this approach would greatly increase the chance of RFID interference and therefore inaccuracy in player detection, alongside a significantly larger cost overhead for the additional readers. This led to our improved design that reduces cost and mitigates RF interference, which utilizes two independent PN532-based NFC reader modules with one “scan station” assigned to each user. Rather than continuously determining the position of each card, user lineups are maintained as a software data structure; this can also streamline power consumption as the NFC readers may remain idle after lineups are set rather than constantly processing modifications.

After consideration of different NFC reader implementations, we determined that the NXP PN532 NFC controller would best fit our needs for its ability to perform RF generation, antenna matching, tag modulation, and NFC communication internally, as well as the feasible cost of materials for 2 units. This option reduces the RF risk associated with utilizing a bare NFC controller that would potentially require a clock, tuned antenna, filtering, and more.

This subsystem contributes to major design considerations by reading a card reliably when it is held on the pad and maintaining that a reader does not pick up cards already resting on the board or on the other team's pad (see Section 2.7.1).

**Contribution:** Reads physical NFC-equipped player cards and communicates with Main Control Subsystem for software processing.

#### Interfaces:

- 13.56 MHz, ISO 14443A tags; read range of 15-35 mm (Section 2.7.1; read range differs by model).
- SPI to the ESP32 at 3.3 V logic, with one chip-select line per reader. SPI was selected over I<sup>2</sup>C so each module can share clock / data signals while having independent chip-select lines.
- Supply: 3.3 V  $\pm$  5% (3.135-3.465 V), at most 200 mA for both readers at peak.

- Output: tag UID and team number; if the reader reports an ambiguous / multiple-scan target, it should be rejected.

**Table 3: NFC Reader Subsystem - Requirements & Verification**

| Requirements                                                                                                                                                                                                                                           | Verification                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• The board must have two NFC readers/antennas operating at 13.56 MHz, one for each team, and each scan must be reported to the Main Control Subsystem with the correct team and player information.</li> </ul> | <ul style="list-style-type: none"> <li>• Inspect the assembled board for two scan pads. Scan a known test tag on each pad 10 times and read the tag ID and team from the serial log.</li> <li>• <b>Pass if:</b> both readers report the correct tag ID and team in 10 of 10 scans each.</li> </ul>                                                                                                                                                                                                                                      |
| <ul style="list-style-type: none"> <li>• A card held on a scan pad must be identified in more than 95% of scans, within 5 seconds.</li> </ul>                                                                                                          | <ul style="list-style-type: none"> <li>• Perform 100 scans (50 per pad) at varied positions and orientations on the pad. Mark the moment the card is presented with a pushbutton tied to an ESP32 GPIO and log the time the tag ID is reported.</li> <li>• <b>Pass if:</b> at least 95 of 100 scans are correctly identified, with each taking <math>\leq 5</math> seconds</li> </ul>                                                                                                                                                   |
| <ul style="list-style-type: none"> <li>• A reader must not read cards resting in the board slots or on the other team's pad, and its read range must be within the 15-35 mm design window.</li> </ul>                                                  | <ul style="list-style-type: none"> <li>• Place cards in all ten slots and scan 50 cards on one pad. Hold a card on one pad and monitor the other pad's reader.</li> <li>• Move a card away from a pad in 5 mm steps with a ruler and record the largest distance with a reliable read (10 of 10 reads).</li> <li>• <b>Pass if:</b> each scan produces exactly one event for the correct team, the other reader reports nothing in 50 scans, and the maximum reliable read range is between 15 mm and 35 mm for both readers.</li> </ul> |
| <ul style="list-style-type: none"> <li>• If two cards are held on a pad together, the scan must be rejected rather than assigning a wrong or corrupted player.</li> </ul>                                                                              | <ul style="list-style-type: none"> <li>• Hold two stacked cards on a pad 20 times.</li> <li>• <b>Pass if:</b> the scan is rejected with the error indication in at least 19 of 20 trials, and no lineup position is changed by a rejected scan.</li> </ul>                                                                                                                                                                                                                                                                              |

### 2.3.3 User Interface Subsystem

The User Interface Subsystem consists of three momentary pushbuttons corresponding to Team 1, Start, and Team 2. Each button is connected directly to an ESP32-S3 GPIO input and uses a 10 k $\Omega$  external pull-up resistor to the 3.3 V rail. When a button is released, the corresponding GPIO is held at logic high; pressing the button connects the input to ground, producing an active-low signal. The ESP32 firmware monitors these inputs and applies software debouncing so that mechanical contact bounce does not generate multiple commands from a single press. A button input is accepted only after remaining continuously low for at least 20 ms, and a second command cannot be generated until the button has first returned to the released state. During lineup setup, the Team 1 and Team 2 buttons are used to confirm their respective lineups, while the Start button begins the matchup only after both teams have completed and confirmed valid lineups. If a button is pressed when its corresponding action is not valid, the command is ignored and the LCD provides feedback indicating the required action.

**Contribution:** Provides simple physical controls for confirming each team’s lineup and starting the matchup without requiring a phone or computer.

#### Interfaces:

- Three active-low GPIO inputs to the ESP32: Team 1 Select, Start, and Team 2 Select.
- Pull-up resistors hold each input at 3.3 V when the button is released; pressing the button connects the input to ground.
- Software debounce interval of 20 ms, with exactly one logical event generated per physical press.
- Team Select inputs are accepted during lineup setup, while the Start input is accepted only after both teams have valid, locked lineups.

**Table 4: User Interface Subsystem - Requirements & Verification**

| Requirements                                                                                                                                                                          | Verification                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• A valid button press must produce exactly one logical command after debouncing.</li> </ul>                                                   | <ul style="list-style-type: none"> <li>• Press each of the three buttons 50 times, including fast repeated presses, while logging events over serial.</li> <li>• <b>Pass if:</b> every press produces exactly one event.</li> </ul>                                                                                                                                             |
| <ul style="list-style-type: none"> <li>• A Team Select button must lock only a complete lineup. Start must be ignored until both teams are locked, then begin the matchup.</li> </ul> | <ul style="list-style-type: none"> <li>• For each team, test Select with a complete lineup and with one position empty in 10 trials each. Attempt Start before both teams lock and after both lock in 5 trials.</li> <li>• <b>Pass if:</b> incomplete lineups never lock, complete lineups lock, early Start presses do nothing, and valid Start presses enter LIVE.</li> </ul> |
| <ul style="list-style-type: none"> <li>• Inputs must be compatible with ESP32 logic levels and never float when buttons are inactive.</li> </ul>                                      | <ul style="list-style-type: none"> <li>• Measure each input pin released and pressed.</li> <li>• <b>Pass if:</b> released pins read 3.3 V +/- 5% and pressed pins read 0.8 V or less.</li> </ul>                                                                                                                                                                                |

### 2.3.4 Display Subsystem

The Display Subsystem provides the primary persistent visual interface for both users. During lineup setup, the display communicates which team is currently entering players, which positions have been filled or remain empty, whether a lineup has been locked, and whether the system is connected to the live-data source. During the LIVE state, the display primarily shows the current fantasy score for both teams. Unlike the RGB LEDs and speaker, which provide temporary event feedback, the LCD provides persistent information about the current game state.

The design uses a standard 16-character by 2-line LCD1602 together with a PCF8574T I<sup>2</sup>C I/O expander, commonly referred to as an LCD “backpack.” The LCD itself normally requires several parallel control and data signals. Connecting these signals directly to the ESP32 would consume multiple GPIO pins and require a larger wiring harness between the main PCB and the display mounted in the upper control panel. The PCF8574T instead receives commands from the ESP32 using only the two I<sup>2</sup>C signals, SDA and SCL, and converts them into the parallel control/data signals required by the LCD.

In the current schematic, SDA and SCL each use a 4.7 k $\Omega$  pull-up resistor to the 3.3 V rail. The PCF8574T is operated from the 3.3 V supply and its parallel I/O pins connect to the LCD control and data pins. The backpack drives the LCD register-select, read/write, enable, and data connections required for the display interface. A 10 k $\Omega$  potentiometer provides an adjustable voltage for the LCD contrast input so that text visibility can be tuned after assembly. The LCD backlight is also current limited by the associated series resistance shown in the display circuit.

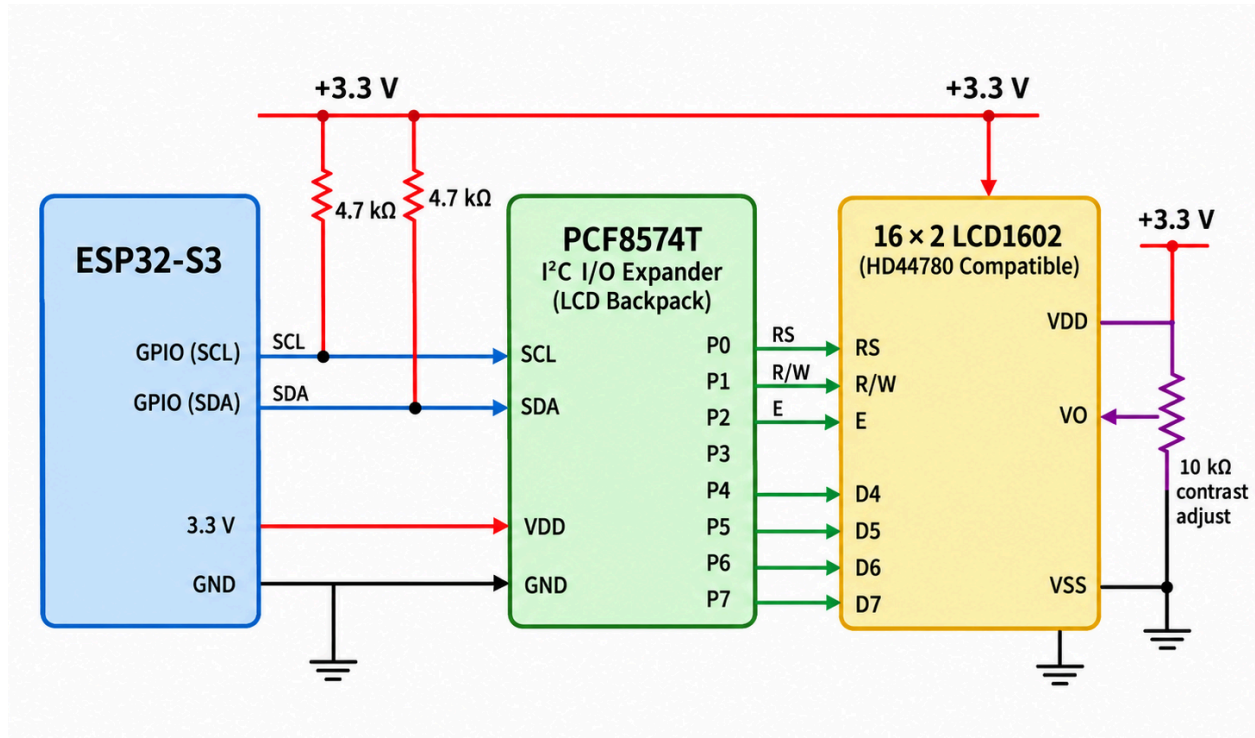


Figure 5: LCD backpack interface between the ESP32-S3 and the LCD1602 display

The I<sup>2</sup>C bus will operate at a standard-mode rate of approximately 100 kHz. This is far more bandwidth than required because only small amounts of text are transferred and the display is refreshed only when a score or system state changes. The PCF8574 family supports I<sup>2</sup>C operation at up to 100 kHz and operates over a supply range that includes the 3.3 V rail used in the current design. Texas Instruments

The display selection is therefore primarily motivated by low interface complexity, low GPIO usage, sufficient character capacity, and easy physical mounting. With five players per lineup, expected fantasy scores can be represented using three or four characters including one decimal place, allowing both teams' scores to be displayed simultaneously on a 16x2 screen. Status messages can use the same screen during lineup entry, network connection, or error states.

**Contribution:** Shows matchup state, real-time fantasy scores for each player, prompts for easy setup, and other high-level information to communicate the game / board's internal state to the user. Provides persistent system-state and scoring information so that users can build lineups, understand system status, and follow the matchup without requiring a phone or computer.

#### Interfaces:

- I<sup>2</sup>C communication from the ESP32 using SDA and SCL at approximately 100 kHz.

- 4.7 kΩ pull-up resistors on SDA and SCL to 3.3 V.
- PCF8574T I/O expander converts the two-wire I<sup>2</sup>C interface into the parallel control/data interface required by the LCD.
- 10 kΩ potentiometer provides LCD contrast adjustment.
- Regulated 3.3 V supply; peak display/backlight current remains within the power budget defined in Section 2.4.1.
- The LCD is refreshed only when a displayed score or system state changes.
- A changed display value shall appear within 500 ms after the ESP32 issues the corresponding update command.

**Table 5: Display Subsystem - Requirements & Verification**

| Requirements                                                                                                                                                                                        | Verification                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• The LCD must simultaneously show both users' scores from 0.0 through 999.9 points and update when the ESP32 sends a changed value within 500ms.</li> </ul> | <ul style="list-style-type: none"> <li>• Inject 20 known score changes from the ESP32 and compare the transmitted value in the serial log with the value shown on the screen as well as the time differential from transmission to display.</li> <li>• <b>Pass if:</b> the displayed value matches the transmitted value within the designated timeframe in 20 of 20 trials.</li> </ul> |
| <ul style="list-style-type: none"> <li>• The LCD must operate from its regulated supply without pulling that rail outside its tolerance.</li> </ul>                                                 | <ul style="list-style-type: none"> <li>• Probe the 3.3 V rail with an oscilloscope while the backlight is at maximum and the screen is refreshing.</li> <li>• <b>Pass if:</b> the rail stays between 3.135 V and 3.465 V.</li> </ul>                                                                                                                                                    |

### 2.3.5 Visual Feedback Subsystem

The Visual Feedback Subsystem provides temporary visual feedback for fantasy football events associated with players in the active lineup of either user. The subsystem consists of RGB LEDs at each player-card roster slot location on the physical game board. The ESP32 controls them so that individual LEDs can change color, brightness, or pulse pattern. When the Main Control Subsystem detects an event such as a touchdown, catch, or fumble, it identifies the lineup slot belonging to that player and activates only the LEDs at that slot. The LED's animation duration will be limited generally to approximately 2-3 seconds so feedback is noticeable but not persistent as LED's are meant to display individual events. This scoring event reflection is also to fall under the high-level latency requirements of updating the physical board within 15 seconds of the corresponding live-data update.

During lineup building, the LED at a slot also lights up when a card for that position is scanned, showing the user where to place it on the board for homogeneity. This turns changes in live NFL statistics into visible physical reactions, and it is the main visual feature that distinguishes the project from a conventional fantasy application. This subsystem therefore contributes to the overall goal of creating an interactive alternative to a conventional digital fantasy-football application.

The board uses ten individually addressable RGB LEDs, one below each player-card slot: five for Team 1 and five for Team 2. The LED subsystem uses a buffer between the ESP32 and LED data line because the ESP32 operates at 3.3 V logic while the LEDs operate from the 5 V rail.

**Contribution:** Shows each event at the affected player's card.

**Interfaces:**

- Single data line from the ESP32 through a 3.3 V-to-5 V level shifter.
- Supply: 5 V +/- 5%, at most 600 mA at peak for ten installed LEDs (firmware brightness limited).
- Timing: the LED responds within 1 second of the Main Control Subsystem alerting to an event.

**Table 6: Visual Feedback Subsystem - Requirements & Verification**

| Requirements                                                                                                                                                                                                                                | Verification                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Each slot's LEDs must be independently controllable, and the correct slot must respond to at least 90% of injected test events.</li></ul>                                                           | <ul style="list-style-type: none"><li>• Inject 30 events (15 “positive”, 15 “negative”) for known players in known slots and record which LEDs respond.</li><li>• <b>Pass if:</b> the correct slot responds in at least 27 of 30 events and no other slot lights in any trial.</li><li>• Positive: Catch, Touchdown...</li><li>• Negative: Fumble, Interception...</li><li>• Dependent on position / offense / defense.</li></ul> |
| <ul style="list-style-type: none"><li>• Touchdowns, catches, fumbles, and other fantasy-football related events must be visually distinguishable by color, brightness, or pattern, as well as from user interaction during setup.</li></ul> | <ul style="list-style-type: none"><li>• Trigger each event type 10 times in random order while three observers who cannot see the controller write down the event type.</li><li>• <b>Pass if:</b> at least 90% of the 90 observer identifications (81 of 90) are correct.</li></ul>                                                                                                                                               |

### 2.3.6 Audio Feedback Subsystem

The Audio Feedback Subsystem produces short audible cues for major gameplay events, such as touchdowns, as well as user-interaction events such as successful and rejected NFC scans. The subsystem consists of a PWM audio output generated by the ESP32, a filtering and amplification stage, and a speaker mounted in the lower portion of the board.

PWM audio was selected because the required sounds are short tones and simple event cues rather than high-fidelity audio. The ESP32 can generate the required waveform directly without a separate digital-audio interface. The PWM output is filtered to reduce the high-frequency switching component and is then amplified before driving the speaker. Tone frequency, pattern, and duration are software configurable. Typical cues are expected to last approximately 0.1- 0.3 seconds so that they are noticeable without continuously interrupting gameplay.

The subsystem shall begin an audible response within 500 ms after receiving an event command from the Main Control Subsystem. Audio playback is handled without blocking NFC scanning, live-statistics polling, score calculation, or display updates.

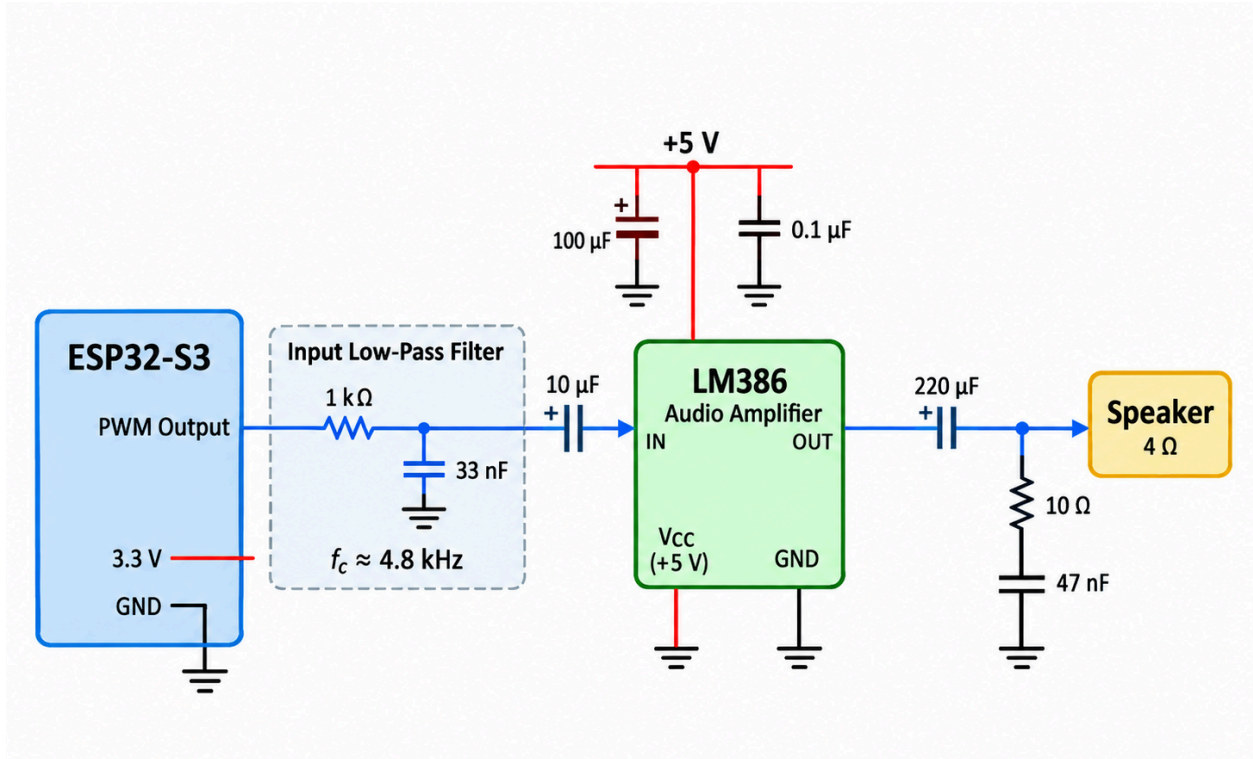


Figure 6: Simplified audio feedback subsystem using PWM generation, filtering, LM386 amplification, and speaker output.

**Contribution:** Provides audible confirmation of major fantasy-football events and user interactions, increasing the physical and interactive nature of the tabletop system.

#### Interfaces:

- PWM output generated by the ESP32 using 3.3 V logic.
- PWM signal passed through the audio filtering and amplification stage before driving the speaker.
- Amplifier supplied from the regulated 5 V rail.
- Peak current budget of up to 500 mA for the audio subsystem.
- Audio response begins within 500 ms of a valid event command from the Main Control Subsystem.

Table 7: Audio Feedback Subsystem - Requirements & Verification

| Requirements                                                                                                                                                                                                            | Verification                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• The speaker must produce the designated audio cue for specific fantasy-football related events and user interactions when commanded by the Main Control Subsystems.</li> </ul> | <ul style="list-style-type: none"> <li>• Inject each football-related event 10 times and have an observer note which cue plays, and inject user-interaction events 10 times as well and monitor the output.</li> <li>• <b>Pass if:</b> the correct cue plays in 20 of 20 trials.</li> </ul> |

| Requirements                                                                                                                              | Verification                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>Audio playback must not cause ESP32 resets, Wi-Fi failures, or loss of rail regulation.</li> </ul> | <ul style="list-style-type: none"> <li>Run Wi-Fi polling and NFC scanning continuously for 30 minutes while playing the loudest cue every 10 s. Log resets and failed HTTP requests and probe the 5 V rail with an oscilloscope.</li> <li><b>Pass if:</b> there are zero resets, zero failed requests caused by playback, and the 5 V rail stays between 4.75 V and 5.25 V.</li> </ul> |

### 2.3.7 Power Management Subsystem

The Power Management Subsystem supplies all electronics on the board and allows the system to operate without a continuous external power connection. The design uses a single-cell (1S) lithium-polymer battery with a nominal voltage of 3.7 V and a fully charged voltage of 4.2 V. A BQ24074 charger and power-path controller accepts 5 V from the USB-C connection, safely charges the battery, and allows the system load to be powered from USB when external power is available [12] . A DW01A protection IC and two external AON2406 N-channel MOSFETs provide battery overvoltage, undervoltage, overcurrent, and short-circuit protection [15], [16] .

The protected battery/system voltage is converted to a regulated 5 V rail using a TPS61236P synchronous boost converter [13] . The 5 V rail directly supplies the RGB LED and audio subsystems and also feeds a TPS62162 synchronous buck converter that generates the regulated 3.3 V rail used by the ESP32, NFC electronics, and display [14] . Hardware design and power calculations are detailed in Section 2.4.

**Contribution:** The Power Management Subsystem provides the stable regulated supplies required by every electronic subsystem and provides safe charging and protection for the rechargeable battery. Loss of rail regulation could reset the ESP32, interrupt NFC communication, or cause incorrect display, LED, or audio behavior.

#### Interfaces:

- Battery: 1S Li-Po, 3.7 V nominal and 4.2 V fully charged, with a selected capacity of 3000 mAh and a continuous discharge capability of at least 5 A.
- Charging input: 5 V USB-C input to the BQ24074. The charger is configured for approximately 500 mA charging/input operation and regulates the cell to approximately 4.2 V.
- Battery protection: DW01A with two AON2406 back-to-back protection MOSFETs for overvoltage, undervoltage, overcurrent, and short-circuit protection.
- Output 1: regulated 5 V  $\pm$  5% (4.75-5.25 V) from the TPS61236P, designed for up to 2 A output.
- Output 2: regulated 3.3 V  $\pm$  5% (3.135-3.465 V) from the TPS62162, rated for up to 1 A output.

**Table 8: Power Management Subsystem - Requirements & Verification**

| Requirements                                                                                                                                                                                                                                                                                                                                    | Verification                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• The regulated 3.3 V and 5 V rails must remain within <math>\pm 5\%</math> of nominal voltage under the maximum expected system load. The 3.3 V rail must support at least 850 mA and the 5 V power stage must support the combined 5 V and downstream 3.3 V loads without loss of regulation.</li></ul> | <ul style="list-style-type: none"><li>• Operate the 3.3 V rail at an 850 mA load while simultaneously applying a 1.10 A direct load to the 5 V rail.</li><li>• Measure both rails and their ripple using an oscilloscope with a 20 MHz bandwidth limit.</li><li>• Pass if: the 3.3 V rail remains between 3.135 V and 3.465 V, the 5 V rail remains between 4.75 V and 5.25 V, ripple is no greater than 100 mV peak-to-peak, and no regulator or battery-protection shutdown occurs.</li></ul> |

| Requirements                                                                                                                                                                                                                                                                                                                                                                                        | Verification                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>The battery and power-conversion circuitry must support at least 90 minutes of continuous active gameplay without loss of system state or rail regulation.</li> </ul>                                                                                                                                                                                        | <ul style="list-style-type: none"> <li>Fully charge the 3000 mAh battery and operate the complete system for 90 minutes with Wi-Fi active, NFC polling enabled, the LCD operating, and periodic LED and audio events.</li> <li>Record battery voltage, 3.3 V rail voltage, and 5 V rail voltage every 5 minutes and monitor power-component temperatures.</li> <li>Pass if: there are zero resets, both regulated rails remain within <math>\pm 5\%</math> for the full 90 minutes, the battery remains above 3.0 V at the end of the test, and no power component exceeds 60 °C.</li> </ul>                                                                                                                                                                                                                                                                                               |
| <ul style="list-style-type: none"> <li>The charging circuit must charge the 1S Li-Po battery to approximately 4.2 V and terminate charging appropriately. The battery-protection circuit must prevent overcharge, excessive discharge, overcurrent, and short-circuit operation. The protection path must support at least 3.5 A without nuisance shutdown during a transient peak load.</li> </ul> | <ul style="list-style-type: none"> <li>Charge a discharged test battery with the system load disabled while monitoring battery voltage and charge current.</li> <li>Pass the charging test if the battery approaches 4.2 V and charge current decreases during charge termination.</li> <li>Test the protection circuitry using a current-limited programmable supply in place of the battery. Slowly reduce the simulated cell voltage and verify that the DW01A disconnects the output near its specified 2.3-2.5 V overdischarge threshold.</li> <li>Increase the electronic load from 0 A through 3.5 A and then toward the protection threshold.</li> <li>Pass if: the protection circuit remains connected through at least 3.5 A, disconnects under an excessive-current or current-limited short-circuit condition, and reconnects normally after the fault is removed.</li> </ul> |

### 2.3.8 External Interface Subsystem

The External Interface Subsystem provides the two external sources of information for the system's full integration and desired functionality: live NFL statistics and the NFC player cards. The sports-data backend delivers player statistics to the ESP32 over Wi-Fi, and each card carries a passive NFC tag with a unique identifier that the NFC Reader Subsystem reads at 13.56 MHz.

Physical player cards are handled by the NFC Reader Subsystem, which converts each NFC tag's unique identifier into a specific player identity (See section 2.3.2). The network-facing portion of this subsystem, namely the ability to pull live NFL data, leverages the ESP32's integrated Wi-Fi interface (2.4 GHz) that allows for connection to an available wireless network. This is a large reason we chose this component to be the heart of our Main Control subsystem and project as a whole.

SportRadar's NFL API is a REST API that is accessible through HTTPS. Requests are authenticated using an API key which will be stored in the ESP32 and responses are returned as a JSON as this is a native format that the ESP32 can read. While the SportRadar API permits requests more frequently than we intend to poll, the intention is to avoid creating an experience where there is potential for constant

overlap in auditory and visual feedback which is why the intended update-rate is approximately 15 seconds.

Required statistics vary based on player position. The API allows us to track passing / rushing yards, passing / rushing touchdowns, fumbles, interceptions, and more, all of which are pivotal in determining total fantasy points.

One of the main considerations for this subsystem is fault handling. For example, if Wi-Fi becomes unavailable, the system should ensure that current lineup and fantasy scores remain stored by the ESP32 and the system attempts to reconnect to Wi-Fi in order to resume API polling as soon as possible.

**Contribution:** Integrates the two primary external information sources: card identities and live statistics.

**Interfaces:**

- HTTPS over 2.4 GHz Wi-Fi to the SportRadar NFL API [2], with JSON responses as defined by the selected API endpoints.
- Request rate limited to the limit of our API access tier (Section 2.7.3).
- Passive 13.56 MHz NFC tags, one unique UID per card.

**Table 9: External Interface Subsystem - Requirements & Verification**

| Requirements                                                                                                                                                                                                                                                                                                 | Verification                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Every player card must be uniquely identifiable, and identified accurately more than 95% of the time within 5 seconds of being scanned.</li> </ul>                                                                                                                  | <ul style="list-style-type: none"> <li>• Scan each of the 30 prototype cards at least 3 times on each pad and record the UID and identification time.</li> <li>• <b>Pass if:</b> every card always returns the same UID, no two cards share a UID, and at least 95% of scans are identified within 5 s.</li> </ul>                                                                                                                                                                   |
| <ul style="list-style-type: none"> <li>• Loss of Wi-Fi connection or an unsuccessful API request should not modify the current game state, including the stored lineup and fantasy score. The system should resume API polling within 30 seconds after a reliable network connection is restored.</li> </ul> | <ul style="list-style-type: none"> <li>• Begin a test matchup and record the current lineup for each user as well as their respective scores. Disable Wi-Fi access for approximately 60 seconds. Repeat 5 times.</li> <li>• <b>Pass if:</b> all stored lineup and scores remain unchanged, access is restored to API and polling continues within the designated time frame.</li> </ul>                                                                                              |
| <ul style="list-style-type: none"> <li>• The backend must provide enough data to calculate scores for all positions (QB, RB, WR, TE, DEF) with a valid Wi-Fi network connection.</li> </ul>                                                                                                                  | <ul style="list-style-type: none"> <li>• Inject 10 recorded statistic updates per position and compare the ESP32 score with a hand calculation. Then operate the system during a real NFL game and compare the time of each scoring event on the board with the time it appeared in the data feed.</li> <li>• <b>Pass if:</b> all calculated scores match the hand calculation within 0.1 points, and at least 90% of scoring events in the live game appear within 15 s.</li> </ul> |

## 2.4 Hardware Design

### 2.4.1 Power Architecture, Battery Sizing & Regulation

The board requires a regulated 3.3 V rail for the ESP32, NFC readers, and display, as well as a regulated 5 V rail for the RGB LEDs and audio amplifier. The final design uses a single-cell Li-Po battery rather than a 2S pack. A 1S battery has a nominal voltage of approximately 3.7 V and reaches approximately 4.2 V when fully charged, so the 5 V rail cannot be produced using a buck converter. The protected battery/system voltage is therefore boosted to 5 V using a TPS61236P synchronous boost converter. The TPS61236P supports a 2.3–4.4 V input range in its 5 V, 2 A reference application, which covers the intended operating range of the single-cell battery [13].

The regulated 5 V rail is then stepped down to 3.3 V using a TPS62162 synchronous buck converter. The TPS62162 provides a fixed 3.3 V output and is rated for up to 1 A, which exceeds the 850 mA peak 3.3 V current estimated in Table 10 [14]. The ESP32 alone may require a supply capable of at least 500 mA during high-current activity, so the 1 A regulator provides additional current margin for the ESP32, NFC readers, and display.

**Table 10: Preliminary power budget (estimates)**

| Load                                          | Rail  | Average current              | Peak current                 |
|-----------------------------------------------|-------|------------------------------|------------------------------|
| ESP32 with Wi-Fi active                       | 3.3 V | 150 mA                       | 500 mA                       |
| NFC readers (two)                             | 3.3 V | 110 mA                       | 200 mA                       |
| LCD and backlight                             | 3.3 V | 100 mA                       | 150 mA                       |
| RGB LEDs (10, brightness limited in firmware) | 5 V   | 200 mA                       | 600 mA                       |
| Audio amplifier and speaker                   | 5 V   | 150 mA                       | 500 mA                       |
| Total                                         |       | 3.3 V = 360 mA; 5 V = 350 mA | 3.3 V = 850 mA; 5 V = 1.10 A |

Using the average-current estimates in Table 10, the 3.3 V loads consume approximately  $0.360\text{ A} \times 3.3\text{ V} = 1.188\text{ W}$ , while the direct 5 V loads consume  $0.350\text{ A} \times 5\text{ V} = 1.75\text{ W}$ . Because the 3.3 V rail is generated from the 5 V rail through the TPS62162, assuming 90% conversion efficiency, the 3.3 V subsystem requires approximately  $1.188\text{ W}/0.90 = 1.32\text{ W}$  from the 5 V rail. The total average load on the TPS61236P boost converter is therefore approximately  $1.75\text{ W} + 1.32\text{ W} = 3.07\text{ W}$ . Assuming 90% efficiency for the boost converter, the average battery power is approximately  $3.07\text{ W}/0.90 = 3.41\text{ W}$ . For a 90-minute game, the required battery energy is therefore approximately  $3.41\text{ W} \times 1.5\text{ h} = 5.12\text{ Wh}$ .

To provide additional design margin, only 80% of the battery's nominal energy is assumed usable and a 25% safety margin is applied. This gives a required nominal battery energy of approximately  $5.12\text{ Wh} \times 1.25/0.80 = 8.0\text{ Wh}$ . At the nominal 3.7 V voltage of a single-cell Li-Po battery, this corresponds to a minimum required capacity of approximately  $8.0\text{ Wh}/3.7\text{ V} = 2.16\text{ Ah}$ . A 1S 3000 mAh Li-Po battery provides approximately  $3.7\text{ V} \times 3.0\text{ Ah} = 11.1\text{ Wh}$  of nominal energy, exceeding the

calculated requirement. Assuming 80% usable capacity, the estimated operating time is approximately  $(11.1Wh \times 0.80)/3.41W = 2.6$  hours, providing substantial margin over the required 90-minute continuous gameplay duration.

### 2.4.2 Battery Charging & Protection

The battery charging and protection circuitry is designed for a single-cell lithium-polymer battery. A TI BQ24074 charger and power-path controller receives 5 V from the USB-C input and performs controlled single-cell charging with a nominal 4.2 V battery regulation voltage. The charger is configured for approximately 500 mA operation using the ISET programming resistor and USB500 input-current configuration. The integrated power-path functionality also allows the system circuitry to be powered from USB while external power is connected, with available input current prioritized between the system load and battery charging.

Independent battery fault protection is provided by a DW01A single-cell protection IC. The DW01A monitors the cell for overcharge, excessive discharge, overcurrent, and short-circuit conditions. Its OD and OC outputs control two back-to-back AON2406 N-channel MOSFETs placed between the raw battery-negative terminal and system ground. The AON2406 was selected because its low on-resistance reduces voltage drop and power loss while providing sufficient current capability for the approximately 3.2 A worst-case battery current calculated for the power system.

The BQ24074 TS input is held in its valid operating range with a fixed 10 k $\Omega$  resistor rather than an external battery thermistor. Therefore, the prototype does not claim active battery-temperature monitoring. Likewise, the PGOOD and CHG status outputs are not required for normal gameplay and are left unused in the current design. Battery charging during development will be supervised and the battery will be inspected for swelling, damage, or abnormal heating as described in Section 3.5.

### 2.4.3 NFC Reader Hardware

The board has two NFC readers, one under each team's scan pad. A PN532-class reader is the reference design point used in our analysis (typical operating distance of about 50 mm) [7]. Antennas are PCB loop antennas tuned to 13.56 MHz with a matching network, following the manufacturer's antenna design guidance [8]. Because the user holds the card directly on the pad, only a short read range is needed. The antenna is therefore kept small and the reader output power is set so the maximum read range is about 55 mm, which prevents a reader from detecting cards resting in the slots (Section 2.7.1). Both readers share the ESP32's SPI bus with separate chip-select lines, so no multiplexer or GPIO expander is needed.

### 2.4.4 Display, LED, Audio, and Input Hardware

- **Display:** A 16 $\times$ 2 LCD1602 with an I<sup>2</sup>C backpack is mounted in the 300 mm-wide center control region. The nominal display-module footprint is approximately 80 mm  $\times$  36 mm. The I<sup>2</sup>C interface minimizes wiring between the display and the main PCB.
- **RGB LEDs:** Ten individually addressable 5 V RGB LEDs are installed, with one LED located below each player-card slot. The ESP32 drives the LED chain through a 3.3 V-to-5 V logic-level shifter. Firmware limits maximum brightness to reduce peak current demand on the 5 V rail.

- **Audio:** The ESP32 generates a PWM audio signal that is passed through a filtering and amplification stage before driving the speaker. The amplifier operates from the 5 V rail and includes local decoupling/bulk capacitance to prevent audio-current transients from disturbing the supply rails.
- **Buttons:** Three momentary pushbuttons are connected to ESP32 GPIO inputs: Team 1 Select, Start, and Team 2 Select. Each input uses a 10 k $\Omega$  external pull-up resistor to 3.3 V, making the buttons active-low. Firmware applies a 20 ms debounce interval and generates only one logical event for each physical press.

## 2.5 Software Design

### 2.5.1 System State Machine

The ESP32 firmware is organized as a state machine so that each phase of a matchup has clearly defined behavior. Table 11 lists the states.

**Table 11: Firmware states**

| State  | Behavior                                                                                                                                                                                                                                                         |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BOOT   | Initialize peripherals, load settings and the player-ID table, and connect to Wi-Fi. Show status on the LCD.                                                                                                                                                     |
| LINEUP | Both teams build lineups. Each scan is identified and assigned to the scanning team's lineup position (a repeat scan for a filled position replaces the earlier card). A team presses Set Lineup to lock its lineup once all five required positions are filled. |
| LIVE   | Entered when Start is pressed while both teams are locked. Poll live statistics, update scores, and trigger LED/audio feedback. Further lineup scans are ignored.                                                                                                |
| END    | When all lineup players' games are finished, show the final score and winner. Return to LINEUP on button press.                                                                                                                                                  |
| ERROR  | Handle Wi-Fi disconnection or unsuccessful API calls: keep the last score on screen, indicate the problem, and retry with back-off. Never loses the lineup or score.                                                                                             |

### 2.5.2 Card Scanning & Lineup Management

The two readers are polled continuously, alternating between them. A card counts as a scan when its tag is read on a certain number of consecutive polls; the intention is to test with this number being 2. The same card is then ignored until it has left the reader's field for roughly one second, so holding a card on the pad does not produce repeated scans. The tag UID is looked up in a table mapping UIDs to NFL players (player ID, name, and position). Unknown tags are rejected with a red LED blink and an error tone.

Lineup assignment is handled in software by the backend. The position of the scanned player determines which slot of the scanning team's lineup is filled, so users never select a position manually. If that slot is already filled, the new scan replaces the earlier player (for example, scanning a second QB overrides the first). The LCD updates to show the new player, and the slot LED lights to show where the card should be placed. Because only scans are read, the physical placement of cards on the board is a visual aid: the lineup is defined by what was scanned, not by where cards are placed.

When a team presses its Select button, the firmware checks that all five required positions are filled and that no player appears on both teams. If valid, that team's lineup is locked and further scans from that

team are ignored; otherwise the press is rejected and the missing position is indicated. Once both teams are locked, the system remains in LINEUP but enables the Start button. Pressing Start then transitions the system into the LIVE state.

The reader is configured to report up to two tags in its field. If more than one tag is present (for example, two cards held on the pad together), the scan is rejected with the error cue and the user is asked to scan one card at a time. No lineup position is changed by a rejected scan.

### 2.5.3 Live Statistics, Scoring & Event Detection

While in the LIVE state, the ESP32 periodically requests statistics for the games containing lineup players from the SportRadar NFL API [2], only polling games that are in progress and contain a rostered player. Responses are parsed (with a JSON filter to limit memory use) into a per-player statistics snapshot. Comparing each snapshot with the previous one yields event outcomes, such as a touchdown, reception, or lost fumble, which drive both the score calculation and the LED/audio feedback. Polling is limited to the API's rate limits to comply with the provider's terms (Section 3). Table 12 gives the default configurable scoring rules.

**Table 12: Scoring rules**

| Offensive players                                                                                                                                                                                                                                                                                                                                                    | Defense (DEF)                                                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• Passing yards: 0.04 per yard</li> <li>• Passing touchdown: 4</li> <li>• Interception thrown: -2</li> <li>• Rushing/receiving yards: 0.1 per yard</li> <li>• Rushing/receiving touchdown: 6</li> <li>• Reception: 1 (PPR; configurable to 0 or 0.5)</li> <li>• Fumble lost: -2</li> <li>• Two-point conversion: 2</li> </ul> | <ul style="list-style-type: none"> <li>• Sack: 1</li> <li>• Interception: 2</li> <li>• Fumble recovery: 2</li> <li>• Defensive/special-teams touchdown: 6</li> <li>• Safety: 2</li> <li>• Points allowed: 0 → 10; 1-6 → 7; 7-13 → 4; 14-20 → 1; 21-27 → 0; 28-34 → -1; 35+ → -4</li> </ul> |

### 2.5.4 Feedback and User Input

Each detected event type maps to an LED pattern at the affected player's slot and, for major events, an audio cue (Table 13). Scan results also give feedback so the user knows whether a scan worked. Events are queued so overlapping events are played in order without blocking the polling and NFC tasks, which run as separate tasks. Buttons are debounced in software: a press is accepted after the input is stable for 20 ms, and a single event is generated per press for Team 1 Select, Start, and Team 2 Select. The LCD is refreshed only when a score or status changes.

**Table 13: Event feedback mapping**

| Event                 | LED response (at player's position) | Audio                |
|-----------------------|-------------------------------------|----------------------|
| Touchdown             | Green pulse pattern                 | Touchdown fanfare    |
| Reception             | Blue flash                          | None                 |
| Fumble / interception | Red flash                           | None (optional buzz) |

| Event                    | LED response (at player's position)       | Audio             |
|--------------------------|-------------------------------------------|-------------------|
| Card scanned             | White pulse at the slot for that position | Short chime       |
| Unknown or rejected card | Red blink on that team's slot LEDs        | Low error tone    |
| Lineup locked            | That team's slot LEDs turn solid          | Confirmation tone |

### 2.5.5 Connectivity and Security

The API key and Wi-Fi credentials are stored in non-volatile memory and are never shown on the display or transmitted anywhere except to the intended service. Requests use HTTPS. Wi-Fi credentials and the SportRadar API key will be provisioned through the USB serial interface during initial system setup and stored in ESP32 non-volatile memory. Normal gameplay will therefore not require a phone or companion application. For the prototype, the ESP32 will communicate directly with the SportRadar API over HTTPS rather than using an intermediate proxy server. Credentials will not be displayed to users or transmitted to services other than the configured Wi-Fi network and SportRadar endpoint.

### 2.5.6 Data Interfaces

This section defines the data the firmware exchanges between its parts and with the sports-data backend.

- **Scan event (internal):** team (1 or 2), tag UID (up to 7 bytes, hex), and a millisecond timestamp, created by the NFC task and consumed by the lineup logic.
- **Player table entry (stored in flash):** UID, player ID, player name, position (QB, RB, WR, TE, or DEF), NFL team, and game ID. About 20-30 entries for the prototype, loaded at BOOT.
- **Lineup (per team):** five slots (QB, RB, WR, TE, DEF), each holding a player ID, plus a locked flag. A scan overwrites the slot for its position while the lineup is unlocked.
- **Statistics snapshot (per player):** passing yards, passing touchdowns, interceptions thrown, rushing yards and touchdowns, receptions, receiving yards and touchdowns, fumbles lost, and two-point conversions; for a defense, sacks, interceptions, fumble recoveries, defensive touchdowns, safeties, and points allowed. Each new snapshot is compared with the previous one to produce event deltas.
- **API request:** an HTTPS GET with the game ID and API key to the provider's per-game statistics resource. The response JSON is filtered to the fields above to limit memory use. For system design and latency analysis, firmware request scheduling is limited to at most one API request per second. The exact SportRadar endpoint names and JSON field mappings will be defined as constants in firmware after confirming the endpoints available under the team's API access tier.
- **Error handling:** a 5 s request timeout, retries after 1, 2, and 4 s then every 10 s, and the last good snapshot kept on screen while the ERROR state is active.

## 2.6 Commercial Component Selection

Table 14 lists the commercial components selected or under consideration. More information in the Cost Analysis section (3.2 Cost for Parts), which details more of our component selection.

**Table 14: Component selection**

| Function           | Candidate                                           | Rationale                                                                                                                                                    |
|--------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Microcontroller    | ESP32-S3-WROOM-1                                    | Integrated 2.4 GHz Wi-Fi and sufficient GPIO, SPI, I <sup>2</sup> C, PWM, and processing resources for the required subsystems.                              |
| NFC reader IC      | PN532-class (x2)                                    | One reader is assigned to each team; it supports 13.56 MHz NFC and the required card-identification interface.                                               |
| NFC tags/cards     | 13.56 MHz ISO 14443A tags (NTAG213-class)           | Inexpensive passive tags with unique UIDs, small enough for cards.                                                                                           |
| Battery            | 1S Li-Po, 3000 mAh, $\geq 5$ A continuous discharge | Provides 11.1 Wh nominal energy, exceeding the calculated 90-minute runtime requirement with design margin.                                                  |
| Charger/Power Path | BQ24074                                             | Provides single-cell Li-Po charging from the 5 V USB-C input and allows the system to operate from external USB power while the battery is connected.        |
| 5 V Regulation     | TPS61236P synchronous boost converter               | Boosts the protected single-cell battery/system voltage to the regulated 5 V rail required by the RGB LEDs, audio subsystem, and downstream 3.3 V regulator. |
| 3.3 V Regulation   | TPS62162 synchronous buck converter                 | Steps the regulated 5 V rail down to 3.3 V and provides up to 1 A for the ESP32, NFC readers, and display.                                                   |
| Battery Protection | DW01A + two AON2406 N-channel MOSFETs               | Provides overvoltage, undervoltage, overcurrent, and short-circuit protection for the 1S Li-Po battery.                                                      |
| Display            | 16x2 LCD1602 with I <sup>2</sup> C backpack         | Fits the center control region and provides both scores plus setup/status text using a four-wire I <sup>2</sup> C connection.                                |
| LEDs               | Addressable 5 V RGB LEDs                            | One data line controls many LEDs with individual color and brightness.                                                                                       |
| Audio              | LM386 audio amplifier and 4 $\Omega$ speaker        | Amplifies the filtered PWM audio signal generated by the ESP32 to drive the speaker for gameplay and user-interface audio cues.                              |

## 2.7 Tolerance Analysis

The NFC reader-to-card interface was identified as the most critical tolerance challenge because the reader must reliably detect a card intentionally presented at a scan pad while avoiding unintended detection of cards resting in the nearby player slots or on the opposing team's scan pad. Failure of this interface would directly violate High-Level Requirement 1 and could result in an incorrect lineup even if

every other subsystem operates correctly. The following analysis therefore evaluates the allowable NFC read-range window and the geometric separation between intentionally scanned and non-target cards.

### 2.7.1 NFC Read Range and Isolation

The main technical risk in the NFC subsystem is the read range of each reader. It must be long enough to reliably read a card held on the marked scan zone, but short enough that a reader does not detect cards already resting in the player slots or a card on the other team's scan zone. NFC operates at 13.56 MHz, and the actual detection distance depends strongly on antenna size, tuning, tag geometry, and supply. The PN532 reference operating distance is therefore treated only as a preliminary reference; final range will be measured on the assembled board.

**Lower bound.** A card held on the scan zone sits about 5-10 mm above the antenna due to the top panel and enclosure and may be tilted or slightly off-center. We therefore retain a required reliable read range of at least 15 mm.

**Upper bound.** In the 500 mm-wide layout, the two NFC reader centers are approximately 400 mm apart. The nearest resting card is the QB slot on the same side. With the scan-zone center about 55 mm from the top edge and the top-row card center about 179.5 mm from the top edge, the vertical separation is about 124.5 mm. The horizontal offset is about 5.5 mm. Assuming a 10 mm height offset between antenna and tag:

$$d = \sqrt{(124.5 \text{ mm})^2 + (5.5 \text{ mm})^2 + (10 \text{ mm})^2} \approx 125 \text{ mm}.$$

If the NFC tag is displaced by 20 mm toward the scan pad within the card, the nearest-tag distance is still approximately 105 mm. Limiting the maximum reliable read range to about 35 mm therefore leaves roughly 70 mm of geometric margin to the nearest resting card, while the opposite team's reader remains much farther away.

**Design target.** The read-range window remains 15-35 mm. The antenna size, matching network, and reader output power will be tuned to land in this window, and the range will be measured on the assembled board (Table 3). If the range is too long, the antenna size, matching network, or reader power will be reduced; if too short, they will be increased. The two readers are also polled alternately so only one RF field is active at a time.

### 2.7.2 Card Scan Time

High-level requirement 1 allows 5 seconds to identify a scanned card. Assuming each reader poll takes about 100 ms (to be measured), alternating between the two readers gives a 200 ms polling loop. Because a card must be read on two consecutive polls of the same reader, the worst case from the card entering the field is about  $2 \times 200 \text{ ms} + 100 \text{ ms} \approx 0.5 \text{ s}$ . Adding under 0.1 s for the table lookup, LCD update, and LED/chime response gives a worst case of about 0.6 s, far below the 5 s budget and leaving room for occasional failed reads that require the user to scan again. Failed reads are not necessarily independent (a card held at a poor orientation can fail repeatedly), so we do not rely on a modeled probability of success. The 95% success rate is verified by measurement over 100 scans (Table 3).

### 2.7.3 Score Update Latency

High-level requirement 2 allows 15 seconds from a statistic becoming available to the score being displayed. The worst-case time consists of the time for a polling cycle to reach the affected game, plus the

request and processing time. If a lineup spans  $N$  in-progress games and the API allows  $r$  requests per second, one full polling cycle takes about  $N / r$  seconds. The firmware is designed around a maximum request rate of  $r = 1$  request/s. Using  $N = 10$ , representing the worst case in which all ten lineup players are participating in different games, one complete polling cycle requires approximately  $N/r = 10$  s. Adding approximately 1.5 s for the HTTPS request and response, 0.2 s for JSON parsing, and less than 0.2 s for score calculation, LCD updating, and initiation of LED/audio feedback gives a worst-case system-side latency of approximately 12 s. This remains below the 15 s high-level requirement with approximately 3 s of margin. Typical lineups span fewer games (for example  $N = 4$  gives about 6 s). Polling only games that are in progress and contain a rostered player, and using a higher request limit if available, increases the margin. This time excludes the provider's own delay in publishing the statistic.

### 2.7.4 Power Rail Capability at Peak Load

The most demanding power condition occurs when the ESP32, NFC readers, display, LEDs, and audio amplifier simultaneously approach their estimated peak currents. From Table 10, the peak 3.3 V load is 0.85 A and the peak direct 5 V load is 1.10 A, corresponding to  $0.85\text{ A} \times 3.3\text{ V} = 2.805\text{ W}$  and  $1.10\text{ A} \times 5\text{ V} = 5.50\text{ W}$ , respectively. Because the 3.3 V rail is generated from the 5 V rail by the TPS62162, assuming 90% conversion efficiency, the 3.3 V stage requires approximately  $2.805\text{ W}/0.90 = 3.12\text{ W}$  from the 5 V rail. The TPS61236P must therefore supply a total peak output power of approximately  $5.50\text{ W} + 3.12\text{ W} = 8.62\text{ W}$ , corresponding to a 5 V output current of  $8.62\text{ W}/5\text{ V} \approx 1.72\text{ A}$ . Since the TPS61236P is designed to provide up to 2 A at 5 V, this leaves approximately 0.28 A, or 14%, of current margin above the calculated simultaneous peak load [13].

The battery-side current is highest when the cell approaches the lower end of its operating-voltage range. At a battery voltage of 3.0 V and an assumed 90% boost-converter efficiency, the peak battery current is approximately  $8.62\text{ W}/(3.0\text{ V} \times 0.90) = 3.19\text{ A}$ . The selected 3000 mAh Li-Po battery will therefore have a continuous discharge capability of at least 5 A, providing substantial margin above the calculated peak current. The battery-protection MOSFETs are also selected to support this current without nuisance shutdown. Similarly, the TPS62162 is rated for 1 A while the estimated maximum 3.3 V load is 0.85 A, leaving approximately 0.15 A, or 18%, of regulator-current margin [14].

Short-duration LED and audio load changes can occur faster than the regulator control loop can respond, so local bulk capacitance is included on the 5 V rail. For a 0.5 A current transient lasting approximately 100  $\mu\text{s}$ , limiting the rail deviation to 0.25 V requires  $C = \Delta I \Delta t / \Delta V = (0.5\text{ A})(100\mu\text{s})/(0.25\text{ V}) = 200\mu\text{F}$ . The design therefore uses at least 440  $\mu\text{F}$  of local bulk capacitance, implemented as two 220  $\mu\text{F}$  capacitors near the audio and other high-current 5 V loads. This is more than twice the calculated minimum and provides additional tolerance for converter response time and component variation. Final rail stability will be verified experimentally using the simultaneous-load procedure defined in Table 8.

## 3 Cost Analysis

Costs for this project primarily include parts, labor, and equipment. Due to the facilities available for students in senior design, much of the equipment and labor from non-group members (such as machine

shop personnel, course staff, and peer reviewers) necessary for this project is provided at no cost. However, these costs are identified below because they are required to complete the project.

### 3.1 Cost for Project Group Labor

Based on available Salary Averages information provided by the University of Illinois [10], the average starting salary for Electrical Engineering majors from Illinois is \$87,429, and the average starting salary for Computer Engineering majors from Illinois is \$119,064. For estimation purposes (since the group consists of both Electrical and Computer Engineering majors), the hourly rate calculation will be done using the midpoint of the average starting salary across these two majors at Illinois.

$(\$87,429 + \$119,064) / 2 = \$103,246.50$  *average starting salary for Illinois ECE graduate*

$\$103,246.50 / (52 \text{ weeks} \times 40 \text{ hours}) = \$49.64$  *average hourly wage for Illinois ECE graduate*

Based on the above calculation, the average starting hourly wage for an Illinois ECE graduate is approximately \$49.64. For simplicity and providing a slight buffer, this value will be rounded to \$50/hr.

Using the provided equation for labor for an estimated ~12 week project timeline with each group member contributing, on average, 5 hours per week, the estimated cost per partner is:

$\$50 / \text{hr} \times 2.5 \times 60 \text{ hrs} = \$7,500$

This brings the total cost of labor for the project group to:

$\$7,500 \times 3 \text{ members} = \$22,500$

### 3.2 Cost for Parts

The table below lists the major components required for the complete prototype. Components will be sourced through a combination of online orders through verified suppliers like DigiKey / Adafruit / TI, ECE shop orders, and personal inventory from past ECE Lab Kits, Arduino Kits, etc.

**Table 15: Estimated Parts and Prototype Hardware Cost**

| Description                               | Manufacturer                         | Part Number      | Quantity | Unit Cost | Extended Cost |
|-------------------------------------------|--------------------------------------|------------------|----------|-----------|---------------|
| ESP32 development board for prototyping   | Espressif Systems                    | ESP32-DevKitC    | 1        | \$10.00   | \$10.00       |
| PN532 NFC breakout module for prototyping | Various / NXP-based                  | PN532 breakout   | 2        | \$11.50   | \$23.00       |
| Breadboard and jumper-wire set            | Various (provided in ECE445 Lab Kit) | —                | 1        | \$10.00   | \$10.00       |
| ESP32-S3 module for final PCB             | Espressif Systems                    | ESP32-S3-WROOM-1 | 1        | \$6.50    | \$6.50        |
| NFC reader IC                             | NXP Semiconductors                   | PN532            | 2        | \$4.50    | \$9.00        |
| NFC antenna and matching components       | Various                              | TBD              | 1        | \$8.00    | \$8.00        |

| Description                                                       | Manufacturer                | Part Number        | Quantity | Unit Cost | Extended Cost   |
|-------------------------------------------------------------------|-----------------------------|--------------------|----------|-----------|-----------------|
| NFC tags for player cards                                         | NXP Semiconductors          | NTAG213            | 30       | \$0.50    | \$15.00         |
| Player card stock and printing                                    | Various                     | —                  | 1 lot    | \$15.00   | \$15.00         |
| 1S Li-Po battery, 3000 mAh, $\geq 5$ A continuous discharge       | TBD                         | TBD                | 1        | \$25.00   | \$25.00         |
| Charger / power-path controller                                   | Texas Instruments           | BQ24074            | 1        | \$2.50    | \$2.50          |
| Battery protection IC                                             | Fortune Semiconductor       | DW01A              | 1        | \$0.25    | \$0.25          |
| Battery protection N-channel MOSFET                               | Alpha & Omega Semiconductor | AON2406            | 2        | \$0.50    | \$1.00          |
| 5 V synchronous boost converter                                   | Texas Instruments           | TPS61236P          | 1        | \$3.50    | \$3.50          |
| 3.3 V synchronous buck converter                                  | Texas Instruments           | TPS62162           | 1        | \$2.50    | \$2.50          |
| 16×2 LCD with I <sup>2</sup> C backpack                           | Arduino Kit                 | LCD1602 + PCF8574T | 1        | \$18.00   | \$18.00         |
| Addressable 5 V RGB LEDs                                          | Worldsemi                   | WS2812B            | 10       | \$0.30    | \$3.00          |
| 3.3 V-to-5 V logic-level shifter                                  | Texas Instruments           | SN74AHCT125        | 1        | \$1.00    | \$1.00          |
| Audio amplifier IC                                                | Texas Instruments           | LM386              | 1        | \$1.50    | \$1.50          |
| 4 $\Omega$ speaker                                                | TBD                         | TBD                | 1        | \$4.00    | \$4.00          |
| Momentary pushbuttons                                             | Amazon                      | TBD                | 3        | \$0.30    | \$0.90          |
| USB-C connector, ESD protection, and CC components                | Various                     | TBD                | 1        | \$4.00    | \$4.00          |
| Resistors, capacitors, inductors, connectors, headers, and wiring | ECE Shop, Personal          | —                  | 1        | \$20.00   | \$20.00         |
| Board deck, card holders, and enclosure materials                 | ECE Machine Shop            | —                  | 1        | \$30.00   | \$30.00         |
| PCB fabrication, two revisions                                    | PCBWay                      | —                  | 2        | \$20.00   | \$40.00         |
| <b>Total Estimated Parts Cost</b>                                 |                             |                    |          |           | <b>\$253.65</b> |

However, components like Enclosure Materials and PCB orders are expensed by the ECE department and will not count against our provided \$150 per team budget.

### 3.3 Cost for Equipment, Facilities, and Shop Services

The project requires standard electronics-laboratory equipment including a current-limited DC bench supply, digital multimeter, oscilloscope, soldering/rework station, and logic analyzer. These resources are available through the ECE 445 laboratory and therefore do not create an additional direct equipment cost to the team. PCB assembly tools and normal shop resources are also provided by the course. No outside machine-shop labor is currently planned. Mechanical material costs for the game-board structure, card holders, and enclosure are included in Table 15. Similarly, they have no cost to our group directly.

**Table 16: Equipment, Facilities, and Shop Services**

| Resource                               | Source                                                         | Quantity | Direct Project Cost |
|----------------------------------------|----------------------------------------------------------------|----------|---------------------|
| Current-limited DC bench supply        | ECE laboratory                                                 | 1        | \$0                 |
| Digital multimeter                     | ECE laboratory                                                 | 1        | \$0                 |
| Oscilloscope                           | ECE laboratory                                                 | 1        | \$0                 |
| Logic analyzer                         | ECE laboratory                                                 | 1        | \$0                 |
| Soldering / rework station             | ECE laboratory                                                 | 1        | \$0                 |
| Standard ECE shop / assembly tools     | ECE facilities                                                 | —        | \$0                 |
| Outside machine-shop labor             | Not currently required; Will utilize in-house ECE Machine shop | 0 hr     | \$0                 |
| <b>Total Equipment / Facility Cost</b> |                                                                |          | <b>\$0</b>          |

### 3.4 Grand Total, Sum of Costs

Based on the calculations above in sections 2.8.1, 2.8.2, and 2.8.3, the preliminary engineering cost of this senior design project is (adjusting parts list for expensed costs):

$$\$22,500 + (\$253.65 - \$70) = \$22683.65$$

### 3.5 Schedule

**Table 17: Project schedule**

| Week           | Task                                                                                                                                                                                          | Person                                    |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Oct 5 - Oct 11 | Finalize BOM, Order parts, PCB Design, Breadboard Design<br>Wi-Fi and SportRadar API prototype<br>Single-reader NFC scan test on breakout board<br>Finalize Power Budget Calculation & Design | Everyone<br>Raahil<br>Parthiv<br>Nicholas |

| Week            | Task                                                                                                                                                                           | Person                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| Oct 12 - Oct 18 | NFC Pad testing, Power System Testing, Finalize Machine Shop<br>Scoring engine and lineup-assignment logic                                                                     | Everyone<br>Raahil              |
| Oct 19 - Oct 25 | Receive PCB, begin PCB assembly<br>Further testing of component integration                                                                                                    | Everyone                        |
| Oct 26 - Nov 1  | PCB order (if necessary again), Progress Demo<br>Build board deck, scan pads, and card holders<br>Design, print, and program cards                                             | Everyone<br>Nicholas<br>Parthiv |
| Nov 2 - Nov 8   | Continue PCB assembly<br>Power rail + NFC tests<br>Firmware state machine integration                                                                                          | Parthiv<br>Nicholas<br>Raahil   |
| Nov 9 - Nov 15  | Live-Game Testing, Mock Demo + Presentation Preparation<br>Integration of NFC scanning, lineup building, scoring, and feedback<br>Requirement 1 verification tests (card scan) | Everyone<br>Nicholas<br>Parthiv |
| Nov 16 - Nov 22 | 90-Minute Endurance Testing + Live-Game Latency Testing                                                                                                                        | Everyone                        |
| Nov 23 - Nov 29 | Fix Outstanding Bugs<br>Work on Presentation / Demo Material                                                                                                                   | Everyone                        |
| Nov 30 - Dec 6  | Final Assembly, Demo Preparation                                                                                                                                               | Everyone                        |
| Dec 7 - Dec 9   | Final report                                                                                                                                                                   | Everyone                        |

### 3.6 Risk Analysis

The subsystem most critical to a successful demonstration is the scan-and-assign flow, because every other feature depends on the system knowing which players are in each lineup. Table 18 summarizes the main risks and how the design mitigates them.

**Table 18: Risk analysis**

| Risk                                                                                      | Impact                                                  | Mitigation                                                                                                                                    |
|-------------------------------------------------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| NFC read range is too short (missed scans) or too long (reads cards resting on the board) | Missed or wrong scans; fails requirement 1              | Target read-range window of 15-35 mm; tune antenna size and reader power; measure on the assembled board (Section 2.7.1).                     |
| Two cards are held on a scan pad together, or one card is scanned twice                   | Wrong or duplicate player assigned; fails requirement 1 | Reader reports up to two tags and a multi-tag scan is rejected with an error cue; a card must leave the field before it can be scanned again. |

| Risk                                                                               | Impact                                          | Mitigation                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User scans a card but places it in the wrong slot, or swaps cards without scanning | Physical board does not match the scored lineup | Lineup is defined by scans, not placement; LCD shows the lineup by name; slot LED shows where to place each card; Set Lineup locks it.                                                                                                    |
| Live-data API rate limits, outages, or slow updates                                | Score updates late; fails requirement 2         | Poll only relevant in-progress games; cache last score; ERROR state retains score and retries; latency analysis in Section 2.7.3.                                                                                                         |
| Wi-Fi or audio current peaks cause brownouts on the ESP32                          | Resets, lost matchup state; fails requirement 3 | Converters rated for peak loads; bulk capacitance; monitor rails during test; save state to non-volatile memory.                                                                                                                          |
| Li-Po battery misuse (overcharge, short, overheating)                              | Fire or injury                                  | BQ24074 charge termination and power-path control; DW01A overvoltage, undervoltage, overcurrent, and short-circuit protection; protected battery enclosure; supervised charging during laboratory testing; battery inspection before use. |
| LEDs draw more current than the 5 V rail can supply                                | Voltage sag, audio noise                        | Limit LED brightness in firmware; size the 5 V converter for peaks.                                                                                                                                                                       |
| PCB errors delay the schedule                                                      | Delayed integration                             | Breakout-board prototyping first; design audit; two planned PCB revisions.                                                                                                                                                                |

## 4 Ethics and Safety Considerations

### 4.1 Societal Impact

Tabletop Fantasy Football is intended to make fantasy sports a more social and face-to-face activity. Traditional fantasy-football participation primarily occurs through individual phones or computers, even when users are physically watching a game together. Our system moves lineup selection, score viewing, and event feedback onto a shared physical device. The primary societal benefit is therefore social and cultural: the design encourages direct interaction between users while they participate in a form of sports entertainment already used by millions of people.

Economically, the prototype requires dedicated electronics and physical player cards, making it more expensive than a software-only application. However, the board and cards are designed to be reusable over many games rather than disposable after a single matchup. The same architecture could also be adapted to other sports if an appropriate statistics source is available, reducing the need to redesign the entire hardware platform for each sport.

Environmental impact is primarily associated with the PCB, battery, enclosure, and electronic components. To reduce unnecessary waste, the design uses rechargeable rather than disposable batteries

and reusable NFC cards. Replaceable or serviceable components will be preferred where practical, and damaged lithium batteries and electronic assemblies will be disposed of through appropriate electronic-waste or battery-recycling channels rather than normal trash. From a global perspective, the underlying architecture is not inherently restricted to the NFL, although availability, licensing, and format of live sports data may differ between leagues and regions.

## 4.2 Applicable Engineering Standards

Several engineering standards and regulations are relevant to the design. Because the system contains a 2.4 GHz Wi-Fi transmitter and a 13.56 MHz NFC interface, radio-frequency emissions are subject to applicable FCC Part 15 requirements [5]. A pre-certified ESP32 module is used for Wi-Fi to reduce RF implementation and compliance risk. The NFC cards and reader interface use ISO/IEC 14443-compatible proximity-card communication; ISO/IEC 14443-3 defines initialization and anticollision behavior for proximity objects [11]. [ISO](#)

The project also uses a rechargeable lithium-polymer battery. IEC 62133-2 provides safety requirements and tests for portable sealed secondary lithium cells and batteries under intended use and reasonably foreseeable misuse [9]. [IEC Webstore](#) These standards guide the team's battery-protection, charging, enclosure, and testing decisions, although the course prototype itself is not being represented as a formally certified commercial product.

## 4.3 Ethical Considerations

The project follows the IEEE Code of Ethics [3] and ACM Code of Ethics and Professional Conduct [4]. The design emphasizes protection of user safety, responsible use of computing resources, honesty in reporting test results, and respect for third-party services.

Live NFL statistics are obtained only through authorized API access. The software will comply with the selected data provider's access terms and rate limits rather than attempting to bypass usage restrictions. API credentials and Wi-Fi credentials will not be displayed to users or intentionally transmitted to unrelated services. Sportradar's NFL API uses authenticated REST requests and supports JSON responses, which matches the planned ESP32 data interface. [Getting Started](#)

The NFC player cards contain identifiers associated with NFL players and do not store personal information about the users of the tabletop system. The project also does not implement monetary wagering or gambling functionality. Any testing results included in the final report will report failures as well as successful trials rather than selectively omitting data that does not meet the desired result.

## 4.4 Electrical and Mechanical Safety Concerns

The primary electrical safety concern is the lithium-polymer battery. Incorrect charging, overdischarge, excessive current, physical damage, or short circuits can cause overheating or thermal runaway. The power subsystem therefore includes charging and protection circuitry for overvoltage, undervoltage, overcurrent, and short-circuit conditions. The battery is mechanically secured inside the board so it cannot move into sharp hardware or conductive surfaces during normal use.

During development, accidental shorts or incorrect component installation can also overheat regulators, wiring, PCB traces, or the battery. Initial power-up of new PCB revisions will therefore be performed from a current-limited bench supply before the battery is connected. The 3.3 V and 5 V rails will be verified before all subsystems are populated or enabled simultaneously.

Mechanical risks are comparatively low but include sharp enclosure edges, exposed fasteners, loose wiring, and components becoming detached inside the enclosure. The completed board will use rounded or finished edges, secure mounting hardware, strain relief where wires connect to movable or external components, and an enclosure that prevents users from contacting conductive circuitry during normal operation.

## **4.5 Safety Procedures and Risk Mitigation**

Before charging or installing the Li-Po battery, the team will inspect it for swelling, punctures, damaged insulation, exposed conductors, or unusual odor. A visibly damaged or swollen battery will not be used. Charging will only be performed through the designated charging/protection circuit and within the battery manufacturer's specified voltage and current limits. Battery charging will not be left unattended during laboratory testing.

New power circuitry will first be tested using a current-limited laboratory supply. The 3.3 V and 5 V outputs will be measured before sensitive components are connected, and supply current will be monitored for unexpected increases. During the 90-minute system endurance test, battery voltage, regulated rails, and component temperatures will be periodically monitored. Testing will be stopped immediately if the battery swells, emits an unusual odor, becomes abnormally hot, or if the power electronics exceed the established temperature limit.

The battery terminals and internal wiring will be insulated against accidental shorts, and the battery will be secured within the enclosure. The completed board will not expose energized conductors during normal user interaction. These procedures, together with the charging/protection hardware and low-voltage system architecture, are intended to protect both users and developers from the primary foreseeable hazards associated with the project. These battery-handling and power-up procedures will be maintained as the team's laboratory battery-safety procedure and followed whenever the Li-Po battery is charged, connected, tested, transported, or used during the final demonstration.

## References

- [1] Fantasy Sports & Gaming Association, “Fantasy Sports & Gaming Association,” reporting more than 53 million fantasy-sports participants, accessed Oct. 2026.
- [2] Sportradar, “NFL API Documentation,” NFL REST API documentation, accessed Oct. 2026.
- [3] IEEE, “IEEE Code of Ethics,” IEEE, Piscataway, NJ, USA, 2020.
- [4] Association for Computing Machinery, “ACM Code of Ethics and Professional Conduct,” ACM, New York, NY, USA, 2018.
- [5] Federal Communications Commission, “Radio Frequency Devices,” 47 C.F.R. Part 15.
- [6] Espressif Systems, “ESP32-S3 Hardware Design Guidelines,” Espressif Systems, accessed Oct. 2026.
- [7] NXP Semiconductors, “PN532 User Manual,” Rev. 02.
- [8] NXP Semiconductors, “AN1445 Antenna Design Guide for MFRC52x, PN51x, and PN53x / AN1444 RF Design Guide,” application note.
- [9] International Electrotechnical Commission, *IEC 62133-2:2017, Secondary Cells and Batteries Containing Alkaline or Other Non-Acid Electrolytes—Safety Requirements for Portable Sealed Secondary Lithium Cells, and for Batteries Made From Them, for Use in Portable Applications—Part 2: Lithium Systems*, IEC, 2017, including Amendment 1:2021.
- [10] University of Illinois Urbana-Champaign, Department of Electrical and Computer Engineering, “Salary Averages,” AY 2024–25 data, accessed Oct. 2026.
- [11] ISO/IEC 14443-3:2018, *Cards and Security Devices for Personal Identification—Contactless Proximity Objects—Part 3: Initialization and Anticollision*, International Organization for Standardization, 2018.
- [12] Texas Instruments, “BQ2407x Standalone 1-Cell 1.5-A Linear Battery Chargers with Power Path,” BQ24074 Datasheet.
- [13] Texas Instruments, “TPS61235P, TPS61236P 8-A Valley Current Synchronous Boost Converters,” TPS61236P Datasheet.
- [14] Texas Instruments, “TPS6216x 3-V to 17-V, 1-A Step-Down Converters with DCS-Control,” TPS62162 Datasheet.
- [15] Fortune Semiconductor Corp., “DW01A One Cell Lithium-ion/Polymer Battery Protection IC,” DW01A Datasheet.
- [16] Alpha & Omega Semiconductor, “AON2406 20 V N-Channel MOSFET,” AON2406 Datasheet.