PokerBuddy: Chipless Poker Companion

ECE 445 Design Document - Spring 2025

Project # 85

Austin Abraham, Lorenzo Dumitrescu, Vishal Ramvelu

Professors: Arne Fliflet, Viktor Gruev, Rakesh Kumar, Michael Oelze, Yang Zhao

TA: Eric Tang

Contents

1 Introduction

1.1 Problem

Traditional poker games rely heavily on physical chips for betting, which can be cumbersome, error-prone, and subject to mismanagement or theft. Managing chip counts, handling physical money, and tracking bet amounts often slow down the game and can lead to disputes among players. In addition, determining whose turn it is during fast-paced games can be confusing and cause a lot of frustration between players. With the growing demand for digital integration in gaming, there is an opportunity to streamline the poker experience by eliminating physical chips and automating bet tracking and game flow. This is different from online poker because we want to maintain the in person experience of playing against your friends face to face, but without the inefficiencies of standard chips and markers that represent blinds.

1.2 Solution

We propose a modular device that removes the need for physical chips while enhancing the poker-playing experience. Each player will use a dedicated device that features LED displays to show both their current balance and the money in the pot, along with a built-in turn indicator light that activates when it is their turn. We will use a force sensitive touch sensor to interpret different gestures—one tap for fold, two taps for check, and a long hold for call—eliminating the need for manual chip handling to signal actions. Additionally, five colored buttons correspond to different chip denominations for quick and easy betting. While we could use some type of sensor for these buttons, we want to maintain the tactile feel and choose to use buttons for our design. These devices will wirelessly connect to a centralized mobile/web application that manages buy-ins, tracks all player balances, and synchronizes game status in real time, ensuring an efficient and error-free gaming experience. Although these devices will not track cards, they must handle the real-time logic of betting, maintaining balances, and managing turn order without relying on a computer. The game logic is distributed and managed by the PCBs in each PokerBuddy. This means that each PokerBuddy keeps track of:

- Whose turn it is to bet (reflected by the turn signal LED).
- The current bet amounts and how they contribute to the pot(reflected by LCD display).
- The players' individual balances(reflected by another LCD display).
- The outcome of each hand (i.e., when a player wins, the entire pot is automatically credited to their balance).

These devices communicate wirelessly with each other and can optionally sync with a centralized mobile application for overall game monitoring and account management by the host of the game (this will just be used for buying in chips and determining payouts at the end). The system is designed to be portable and is powered by disposable batteries, ensuring flexibility and ease-of-use in various settings.

1.3 Visual Aid

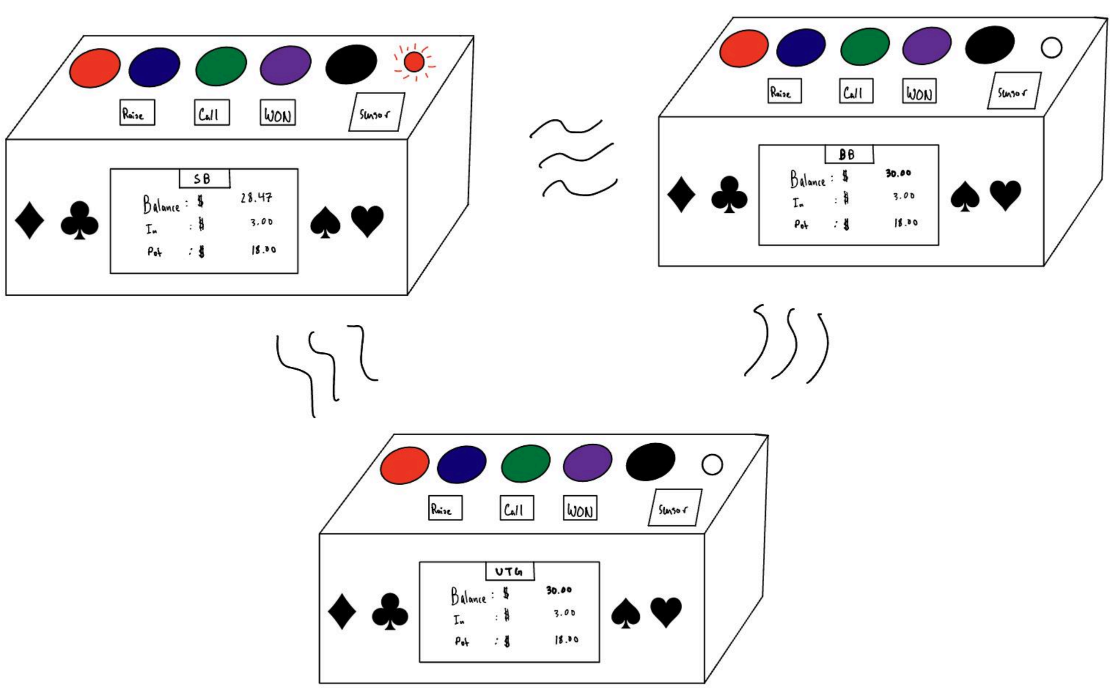

Figure 1: PokerBuddy Visual Aid

This visual aid shows 3 individual PokerBuddy devices playing together. Each device will consist of 8 buttons: CHIP_BTN_1, CHIP_BTN_2, CHIP_BTN_3, CHIP_BTN_4, CHIP_BTN_5, CALL_BTN, RAISE_BTN, and WON_BTN. Each chip button will be used in place of the chips that are used in poker. The call, raise, and won buttons will be used in place of the actions to continue the progression of the game from person to person. The device will also have 2 displays and a diode used for a turn indicator. The displays will show the player who's device it is and the other players important game statistics like their balance and the pot commission.

1.4 High Level Requirements

In order for us to consider our project to be a success, we must complete the following:

1. The game is mathematically correct. This means that 100% of the time the machine will keep track of all the players current balances, each individual game's pot and player contributions, along with the final calculations after each game has been completed.
2. The status of the game will update immediately and accurately after the user interacts with the device. This means that 98% of the time the user's inputs will be recognized correctly, while 100% of the time the machine will do the proper command from the read input.
3. The machine will have reliable wireless communication as long as all the devices are within an 8 foot radius of the table. This will ensure that all players know how close to the device to maintain proper gameplay and communication.

2 Design

2.1 Physical Design

The physical design of the PokerBuddy will consist of 8 buttons, an accelerometer, 2 displays, 1 diode, an ESP32, USB Adapter, voltage regulator, and a 3-D printed enclosure. The 3-D printed enclosure will be able to hold the PCB which is 64.5160 x 81.9150 mm with mounting holes 2.2450 inches and 2.9450 inches apart. It will also be created to enclose the buttons, displays, diode, and accelerometer. The 8 buttons will be using Cherry-MX 1.00u keyboard switches to give the user a satisfying experience. Each button has a different use case defined by the following: CHIP_BTN_1, CHIP_BTN_2, CHIP_BTN_3, CHIP_BTN_4, CHIP_BTN_5, CALL_BTN, RAISE_BTN, and WON_BTN. The accelerometer will be a LIS3DH which is able to detect simple hand motions and taps. The 2 displays will be 0.96 Inch OLED I2C IIC Display Module 12864 128x64 Pixel SSD1306 which will display stats for both the device's user and the other players. The diode will indicate the user whose turn it is currently. For the microcontroller, we will be using an ESP32-WROOM-32 because it has built in bluetooth and GPIO ports to connect to all the other devices. The USB Adapter will be a 6 pin connector on the board that can be connected to the department's existing device to program a microcontroller. The voltage regulator is an LM1117DT-3.3 which linearly converts our 5V input to 3.3V for the microcontroller. All the parts together will create a coherent device that is user friendly.

2.2 Block Diagram

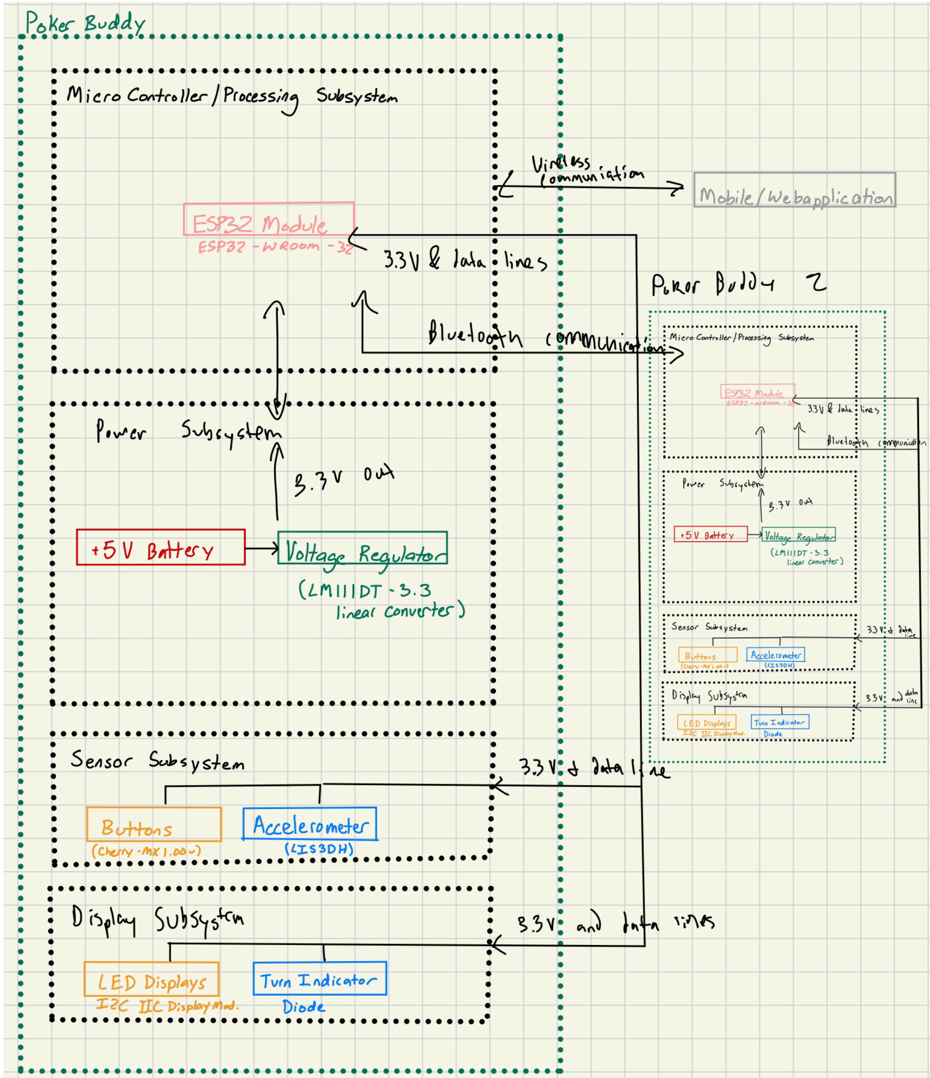

Figure 2: PokerBuddy Block Diagram

This block diagram consists of 4 separate subsystems including: Microcontroller and Processing Subsystem, Power Subsystem, Sensor Subsystem, and Display Subsystem. The Microcontroller and Processing Subsystem consists of the microcontroller and programming adapter. This subsystem will be used to control the other subsystems, read and write data, and do computations. The Power Subsystem is used to take in the power from the source and convert it to 3.3V to be used by the microcontroller. The microcontroller has a power line through it which can help power the other devices. The Sensor Subsystem consists of all the buttons and the accelerometer. These devices will be used to interact with the device's user. The Display subsystem will consist of the 2 displays for user visuals and the turn indicator which is a diode to display the user whose turn it is currently.

2.3 Subsystems and Block Descriptions

2.3.1 Microcontroller and Processing Subsystem

This is the core of the PokerBuddy design. It houses the ESP32-WROOM-32 module. This subsystem performs all the critical game logic which includes tracking bets, pot sizes, player balances, and turn order. It does this while orchestrating communication between the various sensors and displays. It sends and receives data between the Sensor Subsystem for user inputs, then updates the Display Subsystem in real time, and then relies on the Power Subsystem for stable operation. Additionally, this subsystem's wireless capabilities allow optional synchronization with a mobile or web application. The ESP32-WROOM-32 is connected to a USB adapter which will be used to initially program this device so that it is able to read and write data, do computations, and deal with the other devices. It interfaces with the Sensor Subsystem via 3.3V logic-level GPIO, and ADC, with sensor data required to be processed within 100 ms, and outputs to LED displays and a dedicated turn indicator diode, which must update within 5 seconds of a game event. The ESP32 provides built-in WiFi/bluetooth connectivity to ensure a wireless communication range of at least 16 feet, synchronizing real-time data with an optional Mobile/Web Application. This subsystem must operate at a stable 3.3V ± 0.1V and handle a minimum processing throughput that guarantees game logic execution in under 100 ms.

2.3.2 Power Subsystem

This subsystem supplies and regulates the voltage necessary for all the components to function reliably. It comprises a 5V battery feeding into a voltage regulator (LM1117DT-3.3 linear voltage converter) to ensure a steady 3.3 V supply for the microcontroller and other electronics. By providing clean, stable power, this subsystem is essential for maintaining the performance and longevity of the device. It directly interfaces with the Microcontroller and Processing Subsystem to prevent voltage fluctuations or power loss during play. It maintains the stability of the Microcontroller, Sensor, and Display Subsystems by guaranteeing a minimum of 500 mA of constant current with a low voltage ripple (<50 mV). Critical functions like wireless communication, sensor detection, or display updates would be jeopardized if the subsystem did not fulfill these requirements, for example, by supplying insufficient current or large voltage dips, which would cause instability across the system.

2.3.3 Sensor Subsystem

Within the Sensor Subsystem, accelerometer (LIS3DH) together with button (Cherry-MX 1.00u Keyboard switches) presses will control readings for fold, check, call gestures are recorded, andbetting amounts. The Microcontroller and Processing Subsystem receives the data from these sensors, processes it, and uses it to update the game's state and activate any required wireless events or displays. It must send legitimate input signals to the Microcontroller Subsystem in less than 50 milliseconds and runs at 3.3 volts logic levels. Hardware/software debouncing is utilized to minimize false triggers because accuracy is crucial—at least 98% of identified movements must be accurate. By resulting in missed or invalid betting inputs, removing any of these requirements—such as an adequate response time or high detection accuracy—would compromise the integrity of the game.

2.3.4 Display Subsystem

The Display Subsystem uses LED displays (0.96 Inch OLED I2C IIC Display Module 12864 128x64 Pixel SSD1306) that show each player's balance and the quantity of pot in use, as well as a turn indicator LED (diode) to indicate whose turn it is. It also gives the players visual feedback in real time. When betting actions or the turn order change, it instantly updates the displays based on data and control signals received from the Microcontroller and Processing Subsystem. Players are guaranteed to witness real-time changes in pot size and turn order as it updates within five seconds of valid inputs. Furthermore, in typical indoor lighting conditions, it maintains a legible brightness level while drawing power from the controlled 3.3 V rail. The system would not be able to accurately tell players if certain requirements were not met, which would erode players' confidence in the betting calculations made by the gadget.

2.4 Tolerance Analysis

One aspect of our project that could pose a risk is if our force-sensitive touch sensor does not consistently detect gestures with the required 98% accuracy. This would ultimately lead to misinterpreted inputs resulting in incorrect game actions causing arguments and poor user experience. To best address this, we need to make sure our project stays very accurate.

Our overall accuracy can be seen by this equation A = w1*p1 + w2*p2 + w3 * p3 where w1,w2,w3 is the action we take and p1,p2,p3 is probability for single tap, double tap, and long hold respectively. Assuming we take every action equally, all w is ⅓. Even without 100% accuracy on every action, with up to 4% error on any 2 probabilities we can still achieve the given target of 98%. This can be seen if we did statistical validation for 1000 trials with observed accuracies of p1 = 0.96, p2 = 0.96, and p3 = 0.99. With this we can calculate the 95% confidence interval using the formula CI = A (+,-) z * sqrt((A*(1-A))/n)). We use n = 1000 and z-score = 1.96 to get a value of 0.975 (+,-) 0.0106 meaning the range is [0.9644,0.9856]. The 95% confidence interval here includes the target of 98% demonstrating how even with inflated error margins our project will still reach needed accuracy to allow for smooth user experience.

To best handle this, we will use threshold based algorithms for our force sensor to clearly distinguish between each action. One sample way is single tap (force exceeds threshold for short duration < 200 ms), double tap (two force peaks within specific time window, 500 ms), and long hold (force exceeds threshold for extended duration > 1 second). We will also make sure our force resistor can detect light taps and long holds with high precision as well as have strong response time to meet subsystem requirements.

Overall, by closely testing and creating detailed ways to guarantee actions match up with expectations, we can make sure that our project stays within tolerance limits and creates a strong user experience.

3 Cost and Schedule

3.1 Cost Analysis

The physical cost of designing the PokerBuddy is perhaps the cheapest part of the whole project. The display that we are using costs about $3 and we will be using two per device. The switches will come out to about 4$ per device and the Accelerometer will be 1$. We will be using a 9V battery which we will approximate to be around $5. This brings the estimated cost for the physical components to around $20. If we factor in the cost of resistors, the esp32, and other miscellaneous components we can be conservative and estimate that it will be $30. We have about 6 legitimate weeks to put the project together and we estimate that we will each be putting in about 15 hours a week each, this comes out to 270 hours of work in total. If we value each group member at an average cost of $50/hr then the total cost of the project would come out to **$13550**.

Part	# Quantity	Link	Datasheet	Prices
Capacitor (1uF)	4	Link	Link	$0.00
0.96 Inch OLED I2C IIC Display Module 12864 128x64 Pixel SSD1306	2	Link	Link	$5.99
Cherry MX Hyperglide PCB Mount Switches	8	Link	Link	$6.66
Diode	1	Link	Link	$0.00
USB Adapter	1	Link	Link	$0.00
9V Battery	1	Link	Link	$4.46
Resistor (10k)	19	Link	Link	$0.00
Resistor (1k)	1	Link	Link	$0.00
ESP32-WROOM-32	1	Link	Link	$3.15
Linear Voltage Regualtor (LM1117DT)	1	Link	Link	$0.68
Accelerometer (LIS3DH)	1	Link	Link	$0.89
			Total	$21.83

Figure 3: Parts List

3.2 Schedule

We expect to have all of our parts in by the end of next week, which is the **week of 3/10**. Our goal **by the end of 3/10** is to test out all the parts and to assemble the basic circuitry for interfacing with the game. **The week of 3/17** we want to work on implementing the logic for the game and making sure we have established a communication protocol that works for smooth communication between our ESP32 devices. This means being able to pass turns via pressing buttons, having a responsive accelerometer that recognizes input signals, having buttons that work, and being able to output to our display. By **the end of 03/24** we want to have the game logic polished and everything working on the breadboard. The **week of 3/31** we again want to make sure the logic for the game is polished but also design the 3d printing part of the project, which is the actual housing for the PokerBuddy. We will have two people working on the 3d printing and one person working on starting to solder the pcb. **The week of 4/7** we want to have our PokerBuddy 3d printed and the PCB getting polished. **The week of 4/14** we want to start putting everything together and really making sure we don't have any bugs and are able to simulate a game. We want to be ready to go and focused on preparing for the mock demo. The rest of the time will really be just perfecting the design for the final demo.

4 Ethics and Safety

With our project, there are various ethical and safety issues that have to be addressed both while creating and using our project. To start, we have to make sure that our project has privacy and data security. The mobile app we create collects user data (player balances, game history, etc) raising concerns about personal information. From the IEEE code, we have to protect user privacy and ensure data security so we would do this by allowing for tokenized authentication so only respective users can log in and have the option to clear out data after every game. Another aspect is addiction and responsible use. Our project could facilitate excessive gambling. The IEEE Code of Ethics says that engineers must consider the societal impact of their work and avoid harm. To this effect, we would include features to set betting limits or session timers as well as provide consistent warnings/reminders to take breaks during extended gameplay.

In addition we should prevent safety issues which include electrical safety when developing. We will need to use AA batteries and wireless communication in our project and in doing this we have to make sure not to create electrical shocks or fires. In order to prevent this, we will develop our project incrementally testing smaller sub-components and creating clean block diagrams with voltages and electrical components clearly defined before moving forward to ensure robust design. One other area which could cause problems is the device's small components (buttons, sensors) which could pose choking hazards or injury if mishandled. We would try to mitigate this by developing the device with rounded edges and more durable materials as well including warnings in user manual about preventing smaller children from using this device.

By addressing these ethical and safety issues, our Poker Buddy project can make sure of responsible development and use.