AUTOMATED IC CHIP TESTER

ECE 445 Design Document

Team 47: Michael Ruscito (ruscito2), Ryan Yoseph (ryosep2),
and Alison Shikada (shikada2)
TA: Anand Sunderrajan

Spring 2021

Contents

1. Introduction

1.1 Problem and Solution Overview

A common frustration in ECE 385 is when students find a given IC chip does not work as expected. Testing each chip manually is tedious and time consuming. Students are encouraged to test individual chips before using them, but this advice is rarely followed. By adding the time constraint of assignments, complexity of 385 labs, and potential to ruin chips while working on 385, neglecting to unit test the IC chips incorrectly influences students to think that their IC chips are in mint condition.

Our project is targeted toward students by providing a small, portable solution to unit test IC chips quickly and easily. Our goal is to automate the process of chip testing by using a database of TI datasheets and a streamlined UI for easy testing. The user would only need to select the chip model number and press "submit". The output of our device will state whether or not the chip is fully functioning. Internally, our device prepares the appropriate signals and sends them to our PCB. Our software logic would then analyze the output signal and send its output to a web application, accessible by phone or laptop. A bigger picture visual aid is depicted in Figure 1.

Figure 1: Visual Aid

1.2 Background

Manufacturers themselves are not necessarily to blame for faulty IC chips. There are several tests run during the IC chip fabrication process: such as Pre-Burn-in, Burnin and Final Test [1]. Burn-in testing occurs when a device is put under elevated temperatures to "ensure required high quality and reliability of the produced semiconductors before shipping them to final users."

In his research on "Reliability Challenges in 3D IC Packaging Technology", Tu [2] concludes that the most serious reliability concern is joule heating, or the passage of electric current through a conductor produces heat. In a class setting, this is especially prevalent. According to Ronen and Eliahu, most of the common difficulties students have in the study of simple electric circuits are due to an incomplete understanding of the concepts by which idealized models predict the behavior of a system [3]. By offering a portable IC tester, we provide a form of double checking the practice of “idealizing” the circuit model.

1.3 High-Level Requirements

- The device’s internal logic must be able to determine if a chip is working properly in under 250 milliseconds. The ESP32 has a data rate of 150 Mbps, but our main time constraint is iterating through the truth table in our internal logic, making 250 ms our goal for average return time.
- The interface between the ESP32 and WiFi device can select from the 18 types of IC chips provided in the ECE 385 standard lab kit and output the correct testing conditions for the chip.
- The power supply must contain rechargeable Lithium Polymer (LiPo) batteries complete with an integrated Battery Management System (BMS) that grants the user 3 hours of use time.

2. Design

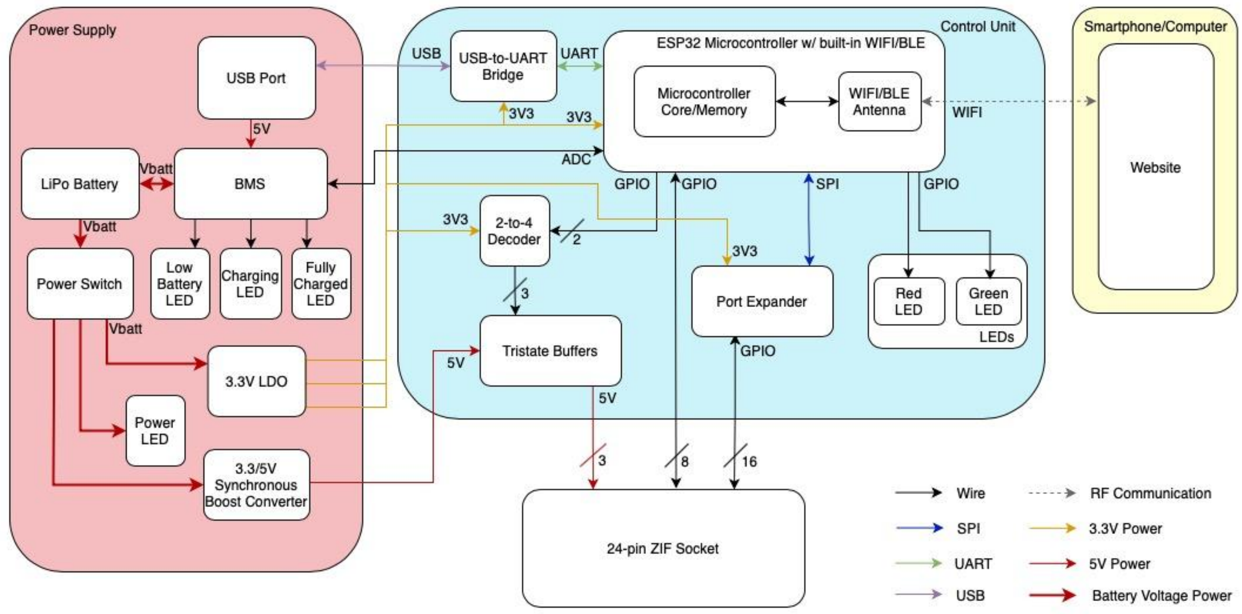

Figure 2: Automatic IC Chip Tester Block Diagram

This device requires three main subsystems to meet the high-level requirements: a regulated power supply, a dedicated control unit, and a smartphone or computer with WiFi capabilities.

The power supply utilizes a USB-rechargeable Lithium Polymer (LiPo) battery with two Low Dropout (LDO) voltage regulators to provide constant voltage/constant current to the device. The control unit is in charge of executing the test suite for the IC chip under test and does so by providing signals to the ZIF socket and communicating with the smartphone via WiFi. The ZIF socket provides an interface for the users to easily insert the IC chips they wish to test.

Lastly, the smartphone allows the user to select from a library of IC chips provided in ECE 385 and communicates this information with the control unit. The ESP32 microcontroller will act as an access point and produce an IP address. The user will merely need to type the IP address in a web browser to access the application. There, they will be immediately prompted with a password in order to see the information provided by our control unit.

Together, the three subsystems can quickly tell the users whether or not their IC chip is functioning properly streamlined through a clean user interface.

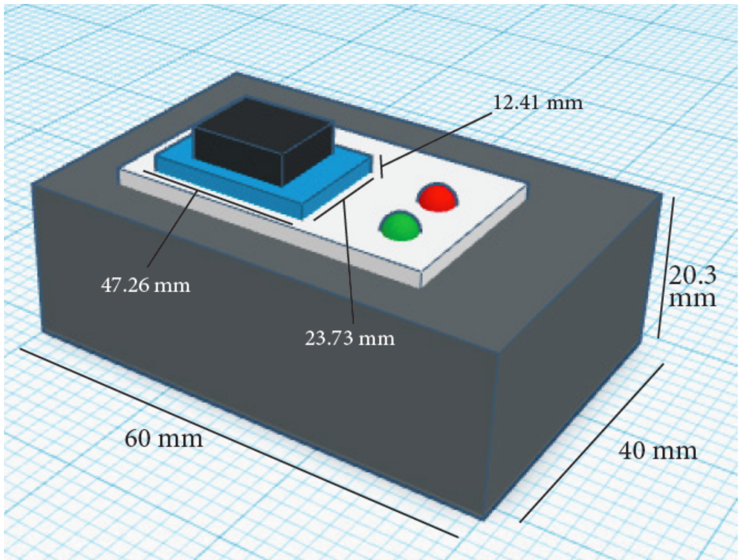

Figure 3: Device Physical Design

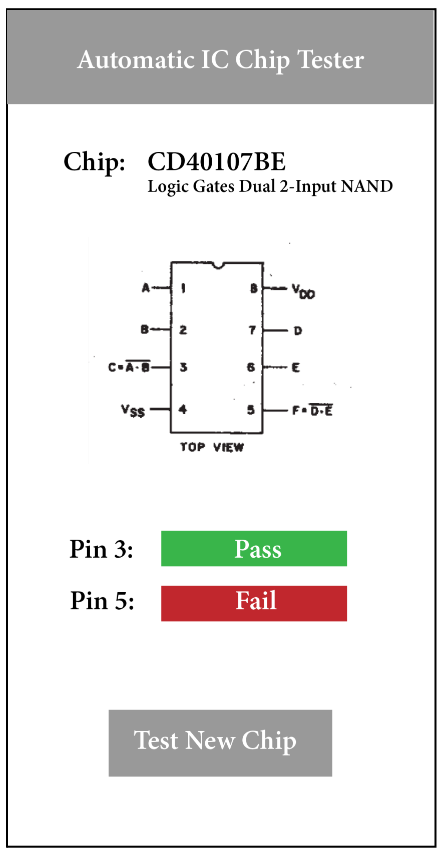

Figure 4: User Interface Design

2.1 Power Supply

The power supply subsystem will provide constant voltage and current to the control unit. While on, the power supply will provide 3.3 V to the microcontroller, USB-to-UART bridge, two-to-four line decoder, and the port expander, as well as 5 V to the level shifter. Furthermore, the power supply will have a USB port to safely charge the LiPo battery, even during use. Furthermore, the microcontroller will monitor the LiPo battery capacity to ensure it does not go above 90% nor below 10% of its total capacity. This is especially important since overcharging or fully discharging can cause internal shorts that will damage the battery and potentially harm the user [5].

2.1.1 Lithium Polymer Battery

A 3.7 V single cell lithium-polymer battery will power the device, and thus dictate device longevity (based on battery life) as well as charging and discharging parameters. The battery provides power when the switch is closed, but only when the battery's capacity is within a safe operating range. Our team concluded that the battery should only be operated when the device is between 10-90% of its total capacity because LiPo batteries should not be stored fully charged, nor should they ever be fully depleted due to safety hazards [5]. This will be regulated through our BMS chip which will feed the battery's voltage into an Analog-to-Digital Converter (ADC) on the ESP32. Thus, this allows the user to simultaneously charge the battery while also using the device.

To ensure that the user has 3 hours of runtime, we can calculate the minimum battery capacity to meet this requirement.

$$\textit{Capacity [Ah]} = \textit{Current [A]} * \textit{Runtime [h]} \tag{1}$$

$$\textit{Capacity [Ah]} = 0.564\ [A] * 3\ [h] = 1.692\ [Ah] \tag{2}$$

Based on the calculated minimum capacity of 1.692 Ah, our team chose a battery with 2.0 Ah of typical capacity to ensure our battery meets the high-level requirement (ESP32-WROOM: 500 mA, 2-to-4 decoder: ~24 mA, ~18.5 for USB-to-UART bridge, ~1 mA for port expander, and 20 mA for tristate buffer, the total output current necessary is 563.5 mA) [6].

2.1.2 USB Port

For user accessibility, we designated a USB 2.0/3.0 port to charge the battery. The USB is omnipresent and widely used in many different applications, therefore eliminating any unnecessary cable-finding hassle for the user. A standard USB 3.0 will provide 5 volts and upwards of 900 mA to the dedicated BMS, which will charge the battery [7].

2.1.3 Power Switch

The power switch controls the on/off state for the entire device. Internally, it protects the integrity of the device and preserves the battery life. Additionally, the power switch provides the user a convenient physical switch for quick on/off capability. The switch is rated to handle 650 mA of current and allows enough overhead to protect the integrity of the device.

2.1.4 Power LED

After the power switch, a dedicated power LED indicates when the battery is powering the device. This added feature is another user safety precaution to communicate when the device is turned on.

2.1.5 3.3 V Low-Dropout (LDO) Regulator

This voltage regulator is required to step down the battery voltage to a regulated 3.3 V that is sent to the majority of the components in the control unit subsystem. Furthermore, this regulator, the TPS777, has a built-in enable pin that allows an open circuit if the input voltage drops too low. Finally, the regulator supplies 542 mA of current to the devices that need 3.3 V. Therefore, to ensure enough overhead, our team chose a regulator that has 600 mA of output current.

2.1.6 3.3 V/5 V Synchronous Boost Converter

This switch-mode boost regulator takes the 3.8 V from our LiPo battery and maintains a 5 V nominal voltage to ensure adequate operation of the tristate buffer. The reason why we chose to include this type of voltage regulator is to ensure that our device is compatible with the V_{CC}'s for the IC chips we are testing.

2.1.7 Battery Charger/BMS

The battery charger and Battery Management System (BMS) are required to safely ensure that our LiPo battery is not overcharged nor fully depleted [5]. We decided to use the MCP7381 Li-Polymer Charge Management Controller from Microchip for its constant-current/constant voltage charge algorithm with selectable preconditioning and charge termination. As seen in Figure 5 below, LiPo batteries are charged at a constant current until the battery reaches its nominal voltage. At this point, the current decays asymptotically to ensure that the battery is not overcharged. This is the crucial role of the BMS to ensure the longevity of our LiPo battery.

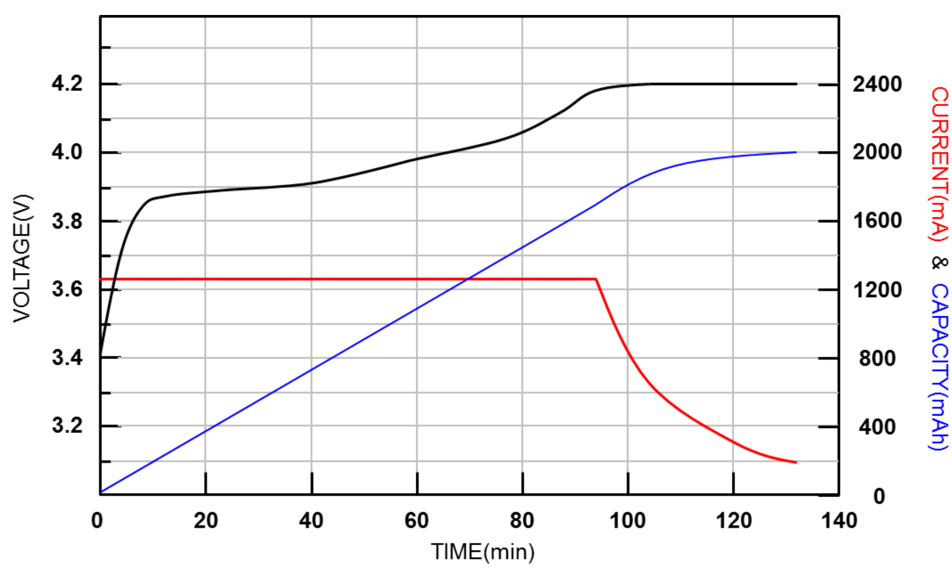

Figure 5: Charging Curve for Lithium Polymer Battery [5]

Table 2.1: Power Subsystem Requirement and Verification

Requirement	Verification
1. The 3.3 V LDO regulator must be capable of outputting 3.3 V ± 0.1 V at 600 mA.	1. Measure the output voltage with a voltmeter to ensure the regulator outputs 3.3 V ± 0.1 V. 2. To ensure the regulator can handle different input voltages, sweep the input voltage from the maximum of 2.5 V to 6.0 V in 0.5 V increments.
2. The power switch must be able to control the on or off state of the device.	1. Power the device and allow it to run for 10 minutes with the switch closed. Then, open the switch to stop the flow of power. Repeat this procedure 6 times, each time increasing the time that the device is on by 10 additional minutes (10, 20, 30, 40, 50, and finally one hour).
3. The 5 V voltage regulator must provide 5 V +/- 0.1V and be able to handle 60 mA sourcing to the tristate buffers.	1. To ensure the regulator can handle different input voltages, sweep the input voltage from the maximum of 12V to 3.5 V in 0.5 V increments.

2.2 Control Unit

This subsystem is responsible for proper handling of the WiFi signal as well as controlling the I/O for the chip under test through SPI protocol communicating with the port expander. Moreover, this subsystem includes the ESP32 microcontroller, SPI-based port expander, tristate buffers, 2-to-4 decoder, LEDs, and a USB-to-UART bridge. The ESP32 miscrocontroller will act as a Soft Access Point with a limit of five connected stations. In this AP Mode, the ESP32 will create an HTTP Server that users can access.

2.2.1 ESP32

Using the WiFi communication within the microcontroller in this subsystem, the web application can communicate with the dedicated server on our microcontroller so that users can select what chip they would like to test. Given the data transfer rate of WiFi protocol 802.11n is between 72 to 217 Mbps [9], we can easily satisfy our high level requirement of delivering a result in under 250 ms. Moreover, the ESP32 also communicates with the port expander using a SPI bus with a maximum frequency of 10 MHz [10]. Finally, we will use the 520 KB of SRAM on the ESP32 [11] to store the expected output of the IC chip test. If the digital output logic from the chip under test matches all of the expected logic for each test case, then the chip is working as expected. If not, then we know that the chip is faulty.

Table 2.2.1: Control Unit Subsystem - ESP32 Requirements and Verifications

Requirement	Verification
1. The device's internal logic must be able to determine if a chip is working properly in under 250 milliseconds. 2. The microcontroller must be able to accurately detect the digital logic from the outputs of the chip under test. 3. The microcontroller's firmware must contain a library of test suites for all 18 chips provided in the ECE 385 lab.	A. Use millis() or esp_log_timestamp() functions in the firmware of the ESP32 microcontroller to calculate the time difference between when the system receives data from the user specifying which chip to test and the time the chip is determined to be working or not. B. For a 3.3 V input voltage, an acceptable output voltage for a "low" logic state would be from -0.3 V to 0.8V and an acceptable "high" logic would be from 2.47V to 3.6V [11]. Logic 1 is high and logic 0 is low. C. We will compare the output of the logic gate with the truth table of a given IC chip to determine the "correct" testing conditions. D. Test chips through the user interface using both chips that have been manually tested and verified to be working as well as chips that are broken. Compare the device's diagnostic of the chips to the known state.

2.2.2 Port Expander

The port expander allows us to test all of the IC chips that are used in ECE385. This component is a necessity to our design because there is a limited number of GPIO pins on the ESP32 microcontroller. After accounting for all pins already being used, there are only 8 available GPIO pins to control the test I/O. Thus, we are using a port expander to ensure that we have the capability to test IC chips with 16 or potentially more pins. The design choice to use a SPI-based port expander rather than I2C-based is because the maximum frequency of SPI communication is 10 MHz whereas the I2C port expander operates at a maximum of 100 kHz. The faster maximum frequency helps ensure that the high level requirement is met to deliver results within 250 milliseconds.

2.2.3 2-to-4 Decoder

Similar to the port expander, the purpose of the decoder is to solve the issue of limited GPIO pins on the ESP32 microcontroller. The enable pins of the tristate buffers are required to have only one active at a time, but require three separate signals to differentiate between each enable pin. By utilizing a two-to-four decoder to control which enable pin is active on the tristate buffers, only two GPIO pins are required from the ESP32 rather than three without the use of the decoder. Additionally, the enable signals for the tristate buffers are inverted, and the outputs from the decoder are inverted. Since these match, this negates the need to invert the control signals in the firmware of the ESP32.

2.2.4 Tristate Buffers

The IC chips require 5 V to power them, but only 3.3 V for the logic signals. The V_{CC} pin has three possible locations on a DIP chip - top right pin, pin 4, or pin 5. In order to supply the correct pin with 5 V, the 3.3 V output from the ESP32 is increased while also allowing for 3.3 V logic signals when the corresponding pin is not V_{CC}. Using tristate buffers with regulated 5 V from the power supply as input and control signals from the ESP32 as enable, the output from the tristate buffer will be 5 V for the pin corresponding to V_{CC}, and high impedance for the other two pins. We chose to use tristate buffers rather than level shifters because the tristate buffers provide simpler circuitry as well as less noise. Both provide the same output, but the use of level shifters would also require a regulated 3.3 V from the power supply and the output signals would have slightly more variance than only using tristate buffers.

2.2.5 LEDs

The purpose of the red and green LED lights are simply for the user to have a quick visual of whether or not the chip is working properly. If the tested IC chip is functioning properly, the green LED will turn on and stay on until the user is ready to test another chip. Else, if the chip is faulty, the red LED will turn on.

2.2.6 USB-to-UART Bridge

A USB-to-UART connection is required in order to upload firmware to the ESP32 during development. The device can be plugged into a computer via USB where ESP32 is programmed, then through UART to the microcontroller. This is not intended to be used after development of the project, but does allow for easy programming during development as well as future updates to the firmware expanding the library of testable IC chips.

Table 2.2.2: Control Unit Subsystem Requirements and Verifications (Excluding ESP32)

Requirement	Verification
1. The port expander must be able to parallelize 16 bits from the SPI communication with the ESP32. 2. The SPI signal must be able to operate at 10 MHz.	A. Test firmware housed on the ESP32 writes and reads each of the 16 bits in the SPI communication and verifies that the corresponding output from the port expander matches with a multimeter. B. Using test firmware, write a single output value to toggle between logic high and low. Use an oscilloscope to measure the frequency of change in a single output of the port expander and divide by 32 since the SPI data packet contains 32 bits [8].
1. The decoder must be able to correctly translate two binary inputs into a single output corresponding to the binary number.	A. Measure the voltage out of the ZIF socket pin in the top right, compared to the voltage of pins 4 and 5. The voltage of the selected pin must be 5 V +/- 0.5 V.

Table 2.2.2 (continued)

1. The tristate buffer outputs 5 V to the selected pin and high impedance to other 2 pins.	A. Measure the voltage out of the ZIF socket pin in the top right, compared to the voltage of pins 4 and 5. The voltage of the selected pin must be 5 V +/- 0.5 V. The other 2 pins must match the ESP32's output within the tolerances described for it's logic values.
1. The ESP32 must be able to be programmed by a computer plugged into the device via USB-to-UART bridge.	A. Plug the device into a windows computer and verify that the ESP32 shows up as a serial device. B. Compile firmware for the ESP32 written in C and write to the device via USB. The output LEDs will respond if successful.

2.3 ZIF Socket

The ZIF Socket is the central testing dock for the user. When the user has selected the IC chip he or she would like to test, they will place it in a dedicated spot so that the proper signals are sent to the chip. The ZIF socket allows for easy insertion and removal of IC chips without the risk of the user seating the pins incorrectly into the device or potentially breaking pins.

Table 2.3: ZIF Socket Requirements and Verifications

Requirement	Verification
1. The user must be able to easily insert and remove IC chips in the ZIF socket without any struggle or risk of damaging pins. 2. The ESP32 must be able to send digital signals to the IC chips under test	A. Plug IC chips of different pin counts into the ZIF socket and remove them by pushing up the lever and confirming that zero force is required. B. Test chips through the user interface using both chips that have been manually tested and verified to be working as well as chips that are broken. Compare the device's diagnostic of the chips to the known state.

2.4 Smartphone

The smartphone subsystem will serve as the primary user interface while configuring the device. With the ESP32 chip's role as the soft Access Point, the smartphone will act as a station (a relationship similar to that of a router and its WiFi client).

2.4.1 WiFi Application

The application will be accessed from the IP address provided by the ESP32. It will offer a button to initiate testing of the IC chip in place and selection from a database of datasheets. Once the testing is complete, the application will display the original chip selection, the findings after testing its output, and an option to test another IC chip.

Table 2.4: Smartphone Subsystem Requirements and Verifications

Requirement	Verification
1. The interface between the WiFi device and ESP32 can select from the 18 types of IC chips provided in the ECE 385 standard lab kit.	A. The web application has a selection of 18 types of IC chips that corresponds to test vectors on the ESP32 microcontroller.
2. The Wifi app can output the correct testing conditions for the chip.	B. The web application correctly displays the same output as determined by the ESP32 microcontroller and corresponding LEDs.

2.5 Tolerance Analysis

An important requirement for our project is that it provides fast results to the user. In order to provide the correct logic values to the pins of the chip under test, we are using a port expander since the ESP32 microcontroller does not have enough GPIO pins to support all of the chips used in ECE 385. A bottleneck in the performance of our device is the speed at which the logic can be sent from the ESP32 microcontroller through the port expander using a communication protocol to the chip under test back to the port expander and re-serialized to send back to the ESP32 microcontroller for analysis. Our initial idea was to use a port expander based on I2C protocol, however the I2C communication is limited to a 400 kHz maximum frequency. By using a port expander based on SPI protocol instead, the maximum frequency is increased to 10 MHz. This increased maximum frequency should allow our device to handle the I/O at a fast enough rate to provide the user results within the high-level requirement of 250 milliseconds. A data packet in SPI protocol to the port expander consists of 32 bits [8], meaning that a single read or write action of the GPIO pins will take 32 clock cycles. For combinational chips, the number of read/write operations to the port expander for a complete test suite is equal to $2^{(N+1)}$ where N is the number of inputs into a single gate. The maximum value of N is 4, meaning the maximum read/write operations for combinational chips is 32. The non-combinational chips will require more operations with the port expander. We estimate that the 4-bit bidirectional shift register will require the most time to test extensively requiring 61 read/write operations to the port expander. Testing this IC chip at the maximum frequency of the SPI protocol means that a test suite can be executed for a chip in $32 * 10^{-7} * 61$ seconds which equals less than 196 milliseconds. Then add the additional time for the ESP32 to run which is negligible compared to the SPI communication as well as the time to communicate via wifi to receive the user's selected device which operates at 150 Mbps data rate. This leaves room for non-optimal SPI frequencies to still reach the high-level requirement of 250 milliseconds to return results.

3. Cost and Schedule

3.1 Cost Analysis

The average starting salary for a 2018-2019 UIUC ECE graduate is about $85,000 [12]. Assuming there are 52 working weeks per year and 40 hours per week, the average hourly wage is about $40.

Table 3.1.1: Labor Costs

Member	Hourly Wage	Hours per Week	Number of Weeks	Multiplier	Total
Alison Shikada	$40	15	16	2.5	$24,000
Michael Ruscito	$40	15	16	2.5	$24,000
Ryan Yoseph	$40	15	16	2.5	$24,000
				Total Labor Cost:	$72,000

Table 3.1.2: Parts Cost

Part Number	Description	Cost
ESP32-WROOM-32E	ESP32 Microcontroller	$2.68
MCP23S17	16-bit I/O Expander (SPI)	$1.32
24-6554-10	24-pin ZIF Socket	$11.04
CD74HCT125E	Quad tristate buffer	$0.85
MCP73831-2ATI/MC	Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers	$0.62
CP2104-F03-GMR	USB-to-UART Bridge	$1.72
SN74LVC1G139	2-to-4 Line Decoder	$0.68
LP605060JU + PCM + WIRES 70MM	BATT LITH POLY 1S1P 1900MAH 3.7V	$17.56
MMBT2222	NPN BJT	$1.58
MAX856	3.3 V/5 V or Adjustable-Output, Step-Up DC-DC Converters	$2.71
AP2111H-3.3TRG1	Linear Voltage Regulator IC Positive Fixed 1 Output 600 mA SOT-223	$0.49

Table 3.1.2 (continued)

MAX1725EUK+T	IC REG LIN POS ADJ 20MA SOT23-5	$2.00
USB 3.0 Cable	USB 3.0 Cable	$2.74
	Total Parts Cost:	$45.99

Based on our labor costs and material expenses, our total is as follows: $72,000 + $45.99 = **$72,045.99**

3.2 Schedule

Table 3.2: Group Schedule and Important Dates

Week	Important Due Dates	Alison Shikada	Michael Ruscito	Ryan Yoseph
3/1	3/4- Design Document Due	• ESP32 wifi app or in house app • backending Shibboleth or 2FA • RV block	• control unit design • part numbers • RV block	• power supply • RV block
3/8	Design Review	• Revise Design based on feedback	• Revise Design based on feedback	• Revise Design based on feedback
3/15	3/17- Teamwork Evaluation I 3/18- 1st Round PCB Orders 3/19- Simulation Assignment 3/19- Soldering Assignment	• Order parts	• Order parts	• Order parts
3/22	3/25- 2nd Round PCB Orders	• Set up simple ESP32 server	• Begin development on ESP32 firmware for test suites	• Begin development on ESP32 firmware for MCP7381 communication
3/29		• Test simple CMOS gate with ESP32 algorithm	• Solder components to PCB	• Solder components to PCB

Table 3.2 (continued)

4/5	4/5- Indiv. Progress Reports Due 4/6- 3rd Round PCB	● Add all 18 IC chips to database ● Start developing Web App	● Electrical testing on PCB	● Electrical testing on PCB
4/12		● Finish testing algorithms for correct output	● Continue development of ESP32 firmware for test suites	● Complete firmware development of ESP32 power delivery
4/19	Mock Demo	● Test ESP32 + Web App Integration ● Finish UI for Web App	● Test/debug device performance	● Test/debug device performance
4/26	Demonstration Mock Presentation	● Deliver demo ● Draft Final Paper	● Deliver demo ● Draft Final Paper	● Deliver demo ● Draft Final Paper
5/3	Presentation 5/5- Final Paper Due 5/6- Lab Notebook Due	● Deliver Presentation ● Finish final paper	● Deliver Presentation ● Finish final paper	● Deliver Presentation ● Finish final paper

4. Discussion of Ethics and Safety

In electing to design a Web application as opposed to a BLE application, we have a higher exposure to cyber attacks of malicious intent. IEEE's Code of Ethics, Section I, Policy 1 [13] plays a role in our attempt to protect the user from a breach of privacy. While we believe the benefits of a WiFi application outweigh that of a BLE alternative, it is our responsibility to not compromise the security of our users. Additionally, it is our responsibility to not abuse the trust of our users by not extracting any data from their device.

In the MVP, our ideal solution integrates a secure 2-factor authentication system using Shibboleth. This would prevent users not affiliated with the university from accessing private information. As a POC, we plan on integrating a user/password lookup table to mimic the authentication process in order to access the home screen of our application. An alternative solution would be to make the ESP32 a soft access point, similar to a hotspot. However, since a user/password implementation is more similar to our MVP, we will elect to go in that direction.

Otherwise, our project has several potential safety hazards. Batteries can be dangerous when used outside of the recommended operating conditions. If a battery is brought to extreme temperatures it can become a fire hazard [14]. We address this issue by only using our project within environments of -18°C to 55°C. Using tristate buffers to control the battery drain, our project will also ensure that the power drawn from the batteries is within the safe operating conditions.

References

[1] Y. H. Ng, Y. H. Low and S. Demidenko, "Improving Efficiency of IC Burn-In Testing," 2008 IEEE Instrumentation and Measurement Technology Conference, Victoria, BC, Canada, 2008, pp. 1685-1689. [Accessed March 1, 2021].

[2] K. N. Tu, “Reliability challenges in 3D IC packaging technology,” *Microelectronics Reliability*, vol. 51, no. 3, pp. 517–523, Sep. 2010.

[3] Ronen, M. and Eliahu, M. (2000), “Simulation — a bridge between theory and reality: the case of electric circuits,” *Journal of Computer Assisted Learning*. [Abstract]. Available: Wiley Online Library, https://doi.org/10.1046/j.1365-2729.2000.00112.x. [Accessed March 1, 2021].

[4] “Logic Signal Voltage Levels: Logic Gates: Electronics Textbook,” All About Circuits. [Online]. Available: https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/#:~:text=For%20a%20CMOS%20gate%20operating,a%20%E2%80%9Chigh%E2%80%9D%20logic%20state. [Accessed: 28-Feb-2021].

[5] DNK Power Company, “Complete Guide for Lithium Polymer (Lipo) Battery: History, Charging, Safety, Storage, and Care,” Lithium ion Battery Manufacturer and Supplier in China-DNK Power, March 11, 2019. [Online]. Available: https://www.dnkpower.com/lithium-polymer-battery-guide/. [Accessed: March 5, 2021].

[6] Texas Instruments, “High-Speed CMOS Logic Quad Buffer, Three-State”, 54HC125 datasheet, Nov. 1997 [Revised Aug. 2003].

[7] N. Stokes, “How to Charge Your Phone Faster,” *techlicious.com*, Jul. 2, 2019. [Online]. Available: https://www.techlicious.com/tip/charge-your-iphone-android-phone-faster/. [Accessed: March 28, 2021].

[8] Microchip Technology, “16-bit I/O Expander with Serial Interface”, MCP23017 datasheet, 2007.

[9] J. Jongerius, “Understand Wi-Fi 4/5/6/6E (802.11 n/ac/ax)”, January 15, 2018. [Online]. Available: https://www.duckware.com/tech/wifi-in-the-us.html#wifi4. [Accessed: February 19, 2021].

[10] NXP Semiconductors, *I2C-bus specification and user manual*, Rev. 6 - April 2014.

[11] Espressif Systems, “ESP32 Series Datasheet,” v3.5, Aug. 2016 [Revised Jan. 2021].

[12] Grainger Engineering Office of Marketing and Communications, “Salary Averages,” Electrical and Computer Engineering. [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages. [Accessed: March 1, 2021].

[13] IEEE, “IEEE Code of Ethics,” *IEEE*. [Online] Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: February 18, 2021].

[14] Energizer Brands, LLC., *Alkaline Manganese Dioxide Handbook and Application Manual*, Energizer, 2018.