Pocket Pal

By

Jack Dai (jackdai2)

Arkajit Dutta (arkajit2)

Alexandria Valdez (avaldez4)

Final Report for ECE 445, Senior Design, Spring 2021

TA: AJ Schroeder

5 May 2021

Team 70

Abstract

For our project, we decided to build an automatic coin dispenser that will distribute a certain combination of coins based on an amount entered by the user. Pocket Pal handles four coin types—penny, nickel, dime, and quarter—that can be inserted into one of the four coin compartments by the user. Each coin compartment acts as a storage unit and has a solenoid attached to it. When commanded, the solenoid pushes coins into Pocket Pal's dispensing tray for the user to collect and make a purchase. In this report, we outline the functionality of each module in Pocket Pal and discuss the design considerations that were made during the assembly of our device.

Contents

1. Introduction

1.1 Objective

Cash payments can be a hassle, especially in crowded businesses like coffee shops or fast-food restaurants. People want to get in and out of these places as quickly as possible, so the additional time that people spend calculating and accumulating the correct combination of coins can be inconvenient and tedious.

With Pocket Pal, this is a problem of the past. Instead of frantically fishing around for the perfect amount of change, users can input the coin amount of their purchase into the device and Pocket Pal will dispense. This compact, high-tech wallet keeps track of how many coins it has at any given time, and it will automatically update these values whenever the user inserts more coins into the device. Pocket Pal is designed to calculate and dispense the correct combination of coins needed for a purchase, overall making cash purchases quicker.

1.2 Background

Although credit and debit usage is a solution for some people, cash purchases are still prevalent in both the United States and overseas. The 2019 Diary of Consumer Payment Choice found that cash purchases in the United States are common for small payments; cash is used for about half of all purchases under $10 and 42% of all payments under $25 [1]. Additionally, the German Association of Money and Bond Services found that 75% of Germany's purchases in 2020 were made through cash, even with the rise in card payments attributed to the pandemic [2].

Pocket Pal is a solution for those who prefer to use cash for small payments. Scrambling for the right coins at a cash register is often time-consuming, but users can hasten this process with Pocket Pal. Our innovative wallet features the capability to determine the exact combination of coins to match the amount input by the user. It will then dispense the coins, sparing users from a frantic look through their wallets. The user can also drop coins into Pocket Pal's coin compartments to reload the device, which increments the coin counts through the use of IR sensors. Our primary focus is the tracking and dispensing of coins, as this is the most time-consuming aspect of cash transactions when compared to bills. Even though there are similar products on the market that can sort coins [3], most cannot do this automatically. Instead, coins must be placed into their respective compartments by the user. Only a few wallets can automatically sort coins [4], but the user must still calculate and grab the correct combination of coins to make a cash purchase. Unlike these other products, Pocket Pal's unique design handles both the sorting and dispensing of coins so that the user does not have to.

2. Design Outline

2.1 Introduction

As seen in Figure 2.1, our device consists of four modules—the power supply, control unit, coin loader module, and coin dispenser module.

Our power supply module consists of a 3.7 V Lithium-ion battery and a 12 V Alkaline battery. The initial plan was to use a singular 12 V supply with a step voltage regulator supplying 5V to the other loads. These loads included two ATmega328PU chips, one OLED screen, one keypad, four IR Sensors, and four IR Emitters. However, we made a design consideration to use two different power supplies for the two different sets of loads.

Our control unit module consists of two ATmega chips and the user interface components which consist of one keypad and one OLED screen. The OLED communicates with the keypad input via our primary microcontroller pins and displays the user input. Our control unit module originally had only one microcontroller. However, we did not have enough pins for our original design and had to add another microcontroller to Pocket Pal.

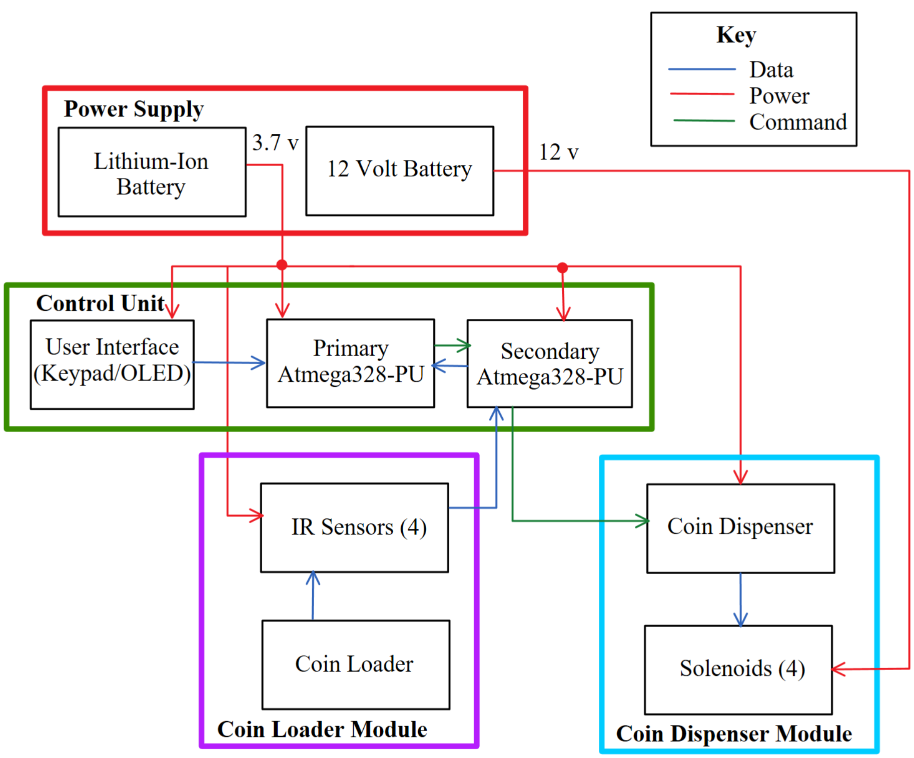


Figure 2.1 Pocket Pal Block Diagram

2.2 High-Level Requirements

- The coin loader must sort a mix of pennies, nickels, dimes, and quarters into their proper compartments with 95% accuracy.
- The coin dispenser must dispense the correct coin amount in a maximum of 15 seconds.
- The overall design of Pocket Pal must be compact, with a maximum size of 6" x 4" x 1.5".

2.3 Coin Loader Module

Our coin loader module has four cylindrical coin compartments to store each coin type separately. An IR sensor and IR emitter are mounted on opposite sides of each compartment to act as a motion sensor and detect when a coin is inserted into the device. One alternative to this approach was to use an ultrasonic sensor. However, ultrasonic sensors have a slower refresh rate than an IR sensor [5], making them less reliable for detecting fast-moving coins.

2.3.1 Coin Loader

The coin loader is purely mechanical and consists of four cylinders, or compartments, meant for storing each coin type. They are sorted by diameter, and the user can place a coin into its respective compartment to load Pocket Pal. We originally wanted to create a mechanical coin loader that would sort coins by type automatically, but through talks with the machine shop we decided on omitting this functionality since it had no electronic components.

2.3.2 IR Sensors

Our IR sensor schematic is pictured in Figure 2.2, where the IR sensors are the components labelled U1, U2, U3, and U4, and the IR emitters are the components labelled D1, D2, D3, and D4. To power the IR emitters, a 38 kHz PWM signal is generated by the control unit (labelled FROM_CONTROL_UNIT in Figure 2.2). We also needed a 20 ohm resistor to satisfy the current requirements of having four emitters in parallel. Refer to Figure 2.3 for these calculations [6].

When an IR sensor detects its corresponding emitter signal, it outputs a low signal to our control unit (labelled TO_CONTROL_UNIT_P/D/N/Q in Figure 2.2). Once the user inserts a coin into its compartment, the coin will temporarily block the IR signal from reaching its sensor and a high signal will be sent to our control unit.

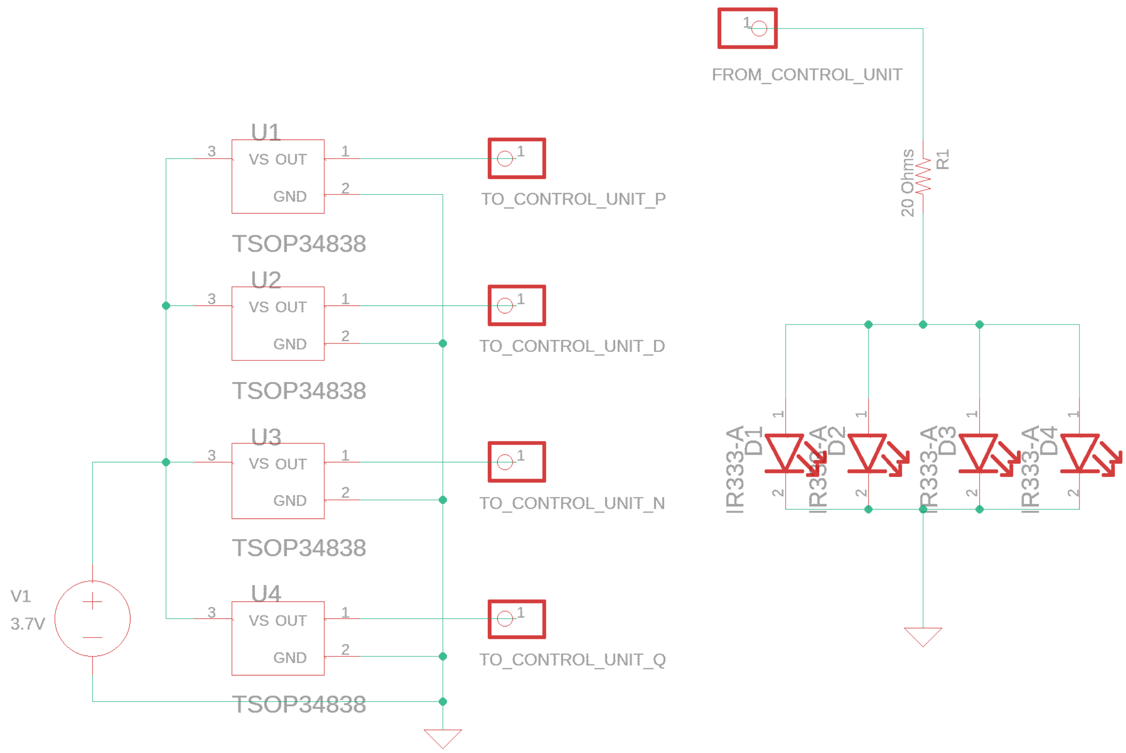
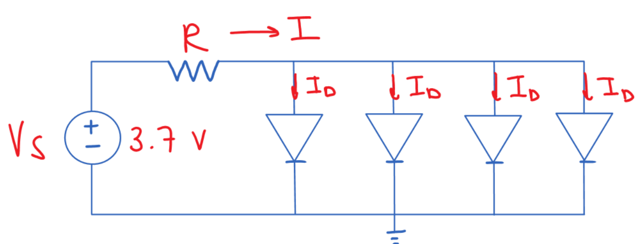


Figure 2.2 IR Sensor and Emitter Schematic



Part: Everlight Emitter IR333-A
Minimum Current = 20 mA = I_{DMin}
Maximum Current = 100 mA = I_{DMax}
(taken from Datasheet)

Minimum Current Case:
$I = (4 \text{ Emitters})(I_{DMin}) = 80$ mA
$R = V_S/I = 3.7$ V/80 mA = 46.25 Ohms -> Max resistance for 20 mA emitters in parallel at 3.7 V

Maximum Current Case:
$I = (4 \text{ Emitters})(I_{DMax}) = 400$ mA
$R = V_S/I = 3.7$ V/400 mA = 10.925 Ohms -> Min resistance for 100 mA emitters in parallel at 3.7 V

Figure 2.3 IR Emitter Calculations

2.4 Coin Dispenser Module

The coin dispensing module releases coins using solenoids. Each coin will be pushed one at a time into the dispensing tray. The signal for dispensing coins comes from the secondary ATmega microcontroller.

2.4.1 Coin Dispenser

The coin dispenser requires the solenoids to receive exactly one signal for each coin that needs to be released. The proper number of signals are given by our combinations algorithm. This algorithm calculates increasing combinations of coins until it finds a total value that is greater than or equal to the target number. It will then exit the algorithm right after the target is reached. In its worst case scenario, the user would input a monetary value close to the amount of coins in the wallet, leading to a runtime of $O(2^n)$, where n is the number of coins in the wallet. In its best case scenario, the user would enter an invalid amount or zero and the algorithm would exit immediately.

There is an alternative algorithm that takes the largest coin that can be inserted until there are either not enough coins or the target has been surpassed. This algorithm runs in $O(n)$, where n is the number of coins. However, this algorithm does not always find the most accurate value. Our team decided to opt for accuracy over efficiency in our coin dispenser, so the former algorithm was chosen.

2.4.2 Solenoids

We used 35 mm stroke solenoids to provide accurate dispensing of each coin. Table 2.1 shows the dimensions of each coin type. Observe that the quarter is approximately 25 mm in diameter. The stroke of the solenoid must be greater than 25 mm or the coins will not be pushed out of the coin dispenser. Our team experimented with 10 mm stroke solenoids, but could not get them to dispense reliably.

We initially considered using motor based dispensing to distribute coins since motors are easier to use compared to solenoids. However, they were less reliable and we could not guarantee that only a single coin would be released at a time. Our team also considered changing the coin loader to have a vertical insertion of coins rather than a horizontal insertion. This would eliminate the need for long stroke solenoids since we would only need a solenoid stroke larger than the thickness of each coin type. Our final design featured a horizontal insertion of coins with solenoids because it had the largest storage capacity and was the most reliable at dispensing.

Table 2.1 Coin Dimensions

Coins	Pennies	Nickels	Dimes	Quarters
Radius (m)	0.009525	0.010605	0.008955	0.01213
Width (m)	0.00152	0.00195	0.00135	0.00175
Surface Area (m^2)	0.00028502	0.0003533	0.0002519	0.0004622

2.5 Control Unit Module

The control unit consists of two microcontrollers and the user interface. The microcontrollers are either in the loading state or dispensing state. During the loading state, the ATmega328-PU microcontrollers work together to detect and increment coin counts. During the dispensing state, the microcontrollers will calculate the necessary coin combination to release and signal the dispenser module.

2.5.1 User Interface

The user interface consists of the keypad and OLED display. The user types in a desired amount then presses enter or backspace. The enter button starts the coin dispensing process and backspace allows the editing of the desired monetary amount. The OLED displays the number of coins in the wallet and the user input.

The original OLED that our team purchased had too many pins and could not be mounted onto our device by the machine shop. We swapped it out for an SSD1306 OLED with 7 pins.

2.5.2 Primary Atmega328-PU

The primary microcontroller (PM) is responsible for keeping track of Pocket Pal's current coin counts, running the coin dispensing algorithm, and switching between loading and dispensing mode. Even though the use of one microcontroller would have been more straightforward to implement, we had to add the secondary microcontroller (SM) to have enough pins for all of our components. We decided to keep all of the calculations, logic, and control signals centralized on the PM so that we only needed to send high and low signals, and not more complicated data, between the two Atmega chips.

The PM receives the keypad's input and sends this information to the OLED display. To designate whether the circuit is currently in loading or dispensing mode, the PM has two flags that it sends to the SM (LOAD_FLAG and DISPENSE_FLAG in Figure 2.4). It also has four additional signals (SECONDARY_Q/N/P/D) between the two Atmega chips. In loading mode, these four signals travel from secondary to primary and tell the primary which coin counts to increment. In dispensing mode, these four signals change directions so that the PM can tell the SM which of the four solenoids to turn on.

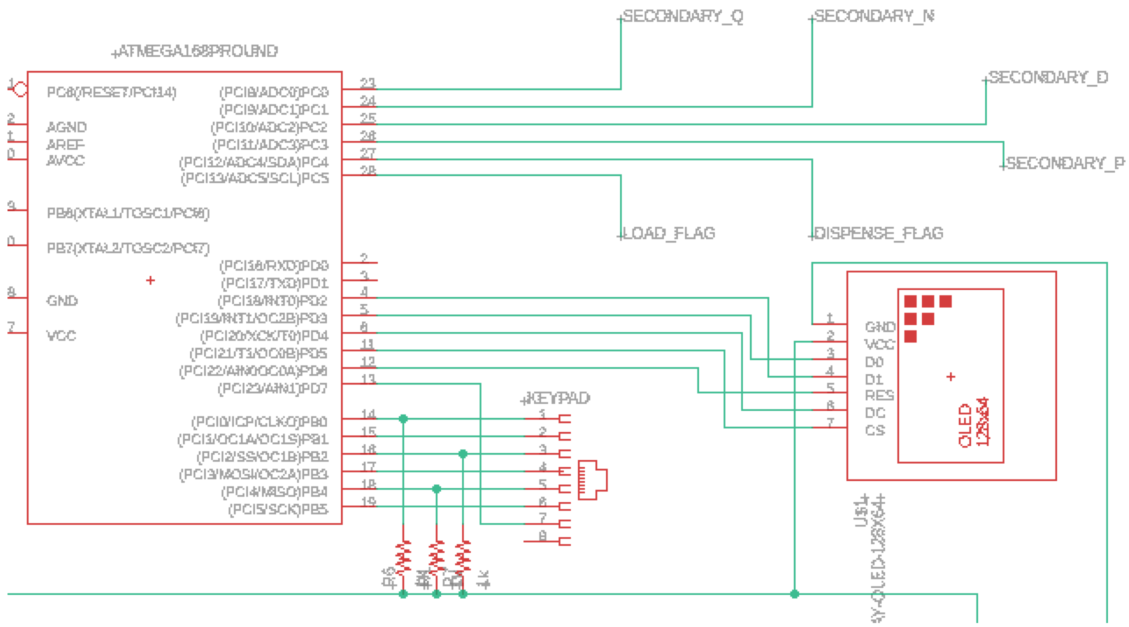


Figure 2.4 Primary Microcontroller Schematic

2.5.2 Secondary Atmega328-PU

The secondary microcontroller (SM) acts differently depending on the loading and dispensing flags sent by the primary microcontroller (PM). When in loading mode, the SM sends the four output signals of our IR sensors to the PM so that it knows which coin counts to increment. When in dispensing mode, the SM receives a high signal from the PM corresponding to which of the four coin types must be dispensed. In return, the SM sends a high signal to the correct MOSFET which turns on the solenoid it is connected to.

2.6 Power Supply Module

The power supply module consists of two batteries. The Lithium-ion 3.7 V battery powers the control unit and loader module. The 12 V alkaline batteries power the solenoids.

Our original design with motors only required a single 5 V battery and a few voltage regulators. We switched to the Lithium-ion battery since it was rechargeable and more compact. During our project, we considered swapping over to small 10 mm stroke solenoids that run on 5 V. However, due to hardware difficulties explained earlier, 10 mm stroke solenoids were never used.

2.6.1 Lithium-Ion Battery

We diverted from our original design of AA batteries, and instead opted for a 3.7 v Lithium-Ion battery due to its portability and rechargeability. Due to time constraints, we were never able to prototype or incorporate the battery into our final product.

2.6.2 12 Volt Alkaline Battery

The 12 V batteries consist of eight AA duracell batteries in a battery pack. This change was made close towards the end of our project. We could not find any cheaper 12 V battery alternatives that were still within our budget.

3. Design Verification

3.1 Coin Loader Module

3.1.1 Coin Loader

The coin loaders do not have any verifications as the requirements were removed. Initially, we planned on sorting coins based on their size for the coin loader module. However, after recommendation from the Machine Shop we decided to scrap that and have a more intuitive and simpler coin loader module where the user manually inserts coins in the respective coin compartments.

3.1.2 IR Sensors

Our team verified the IR sensors through LED simulations. Figure 3.1 depicts the setup used to verify the IR sensors. When an object obstructs the sensor, the LED turns off. This signifies that the IR emitter's signal cannot reach its corresponding IR sensor, which simulates a coin being inserted into the device.

Figure 3.2 shows the testing of the IR sensors with aluminum foil. The corresponding LED turned off to indicate a detected coin. Figure 3.3 shows the final state of all the coins after a cycle of verification.

The IR sensors were verified through the foil test. However, we realized that the sensors have difficulty detecting actual coin drops during the demonstration. We followed up with another test with the coins dropping horizontally and vertically, as shown in Table 3.1. The long drop is when the coin is dropped vertically in front of the IR sensor. The flat drop is when the coin is dropped horizontally in front of the IR sensor. As shown in Table 3.2., the flat coin drop has a lower detection rate than the long drop. Unfortunately, Pocket Pal has a flat drop orientation so if we had the opportunity we would redesign our coin compartments to support the long drop.

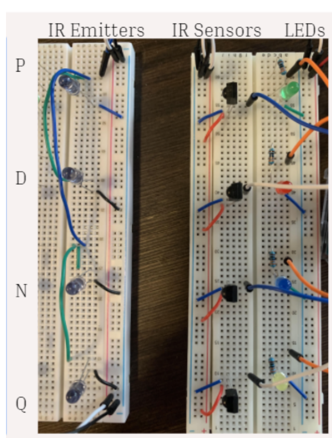
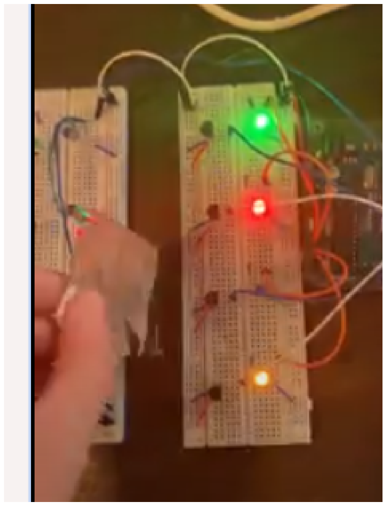


Figure 3.1 IR Sensor Simulation Setup



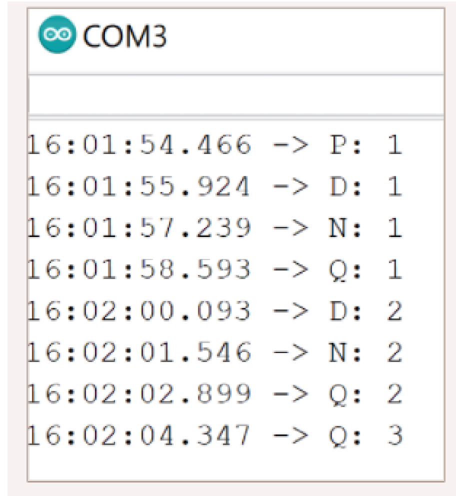


Figure 3.2 IR Sensor Verification *Figure 3.3 IR Sensor Detection*

Table 3.1 Long and Flat Coin Drop Results

Trial	Dime Detected	Penny Detected	Nickel Detected	Quarter Detected	Accuracy
Long 1	2	2	3	3	0.83
Long 2	3	2	3	3	0.92
Long 3	2	2	3	3	0.83
Flat 1	1	1	1	3	0.50
Flat 2	1	2	1	3	0.58
Flat 3	0	2	2	3	0.58

Table 3.2 Long and Flat Coin Drop Average Accuracy

Long Drop Accuracy	Flat Drop Accuracy
32/36 = 0.89	20/36 = 0.56

3.2 Coin Dispenser Module

3.2.1 Dispenser

Our team verified the coin dispenser's requirements through LED simulations. Figure 3.4 shows the setup used to test the dispenser. The green LED on the left indicates that the circuit is in dispensing mode. Each LED from left to right corresponds to the quarter, nickel, dime, and penny compartments. When a coin LED lights up, it indicates the dispensing of a coin. This can be seen in Figure 3.5.

As seen in Figure 3.6., we were able to successfully verify the requirements for our dispenser. The high level requirement requires the complete dispensing cycle to complete within 15 seconds. It takes about 0.5 seconds for a coin to signal. Thus, if the device dispenses less than 30 coins, this high level requirement will be satisfied.

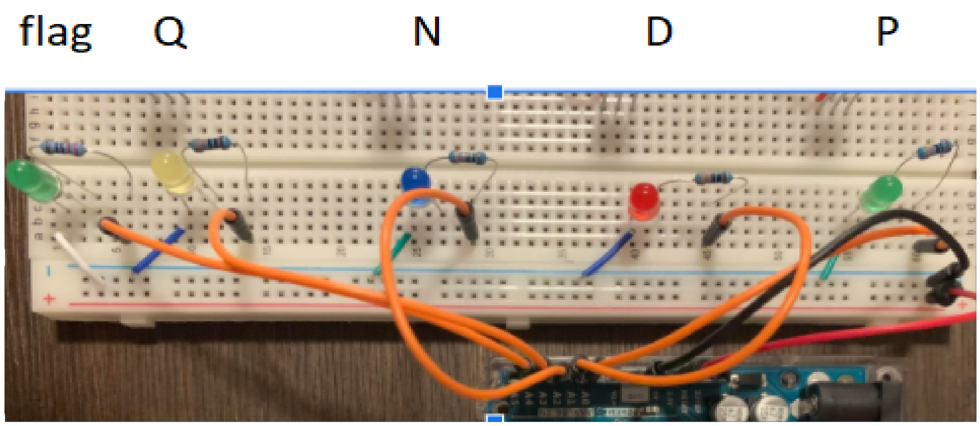


Figure 3.4 Coin Dispenser Simulation Setup

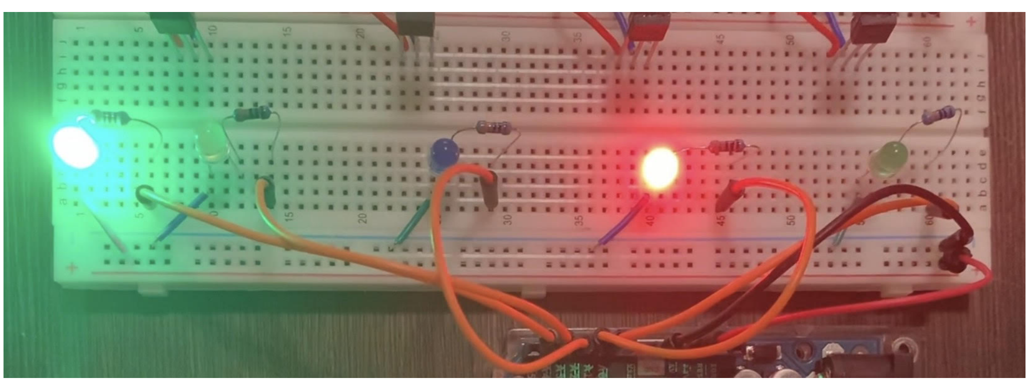


Figure 3.5 Dime LED Dispense Signal

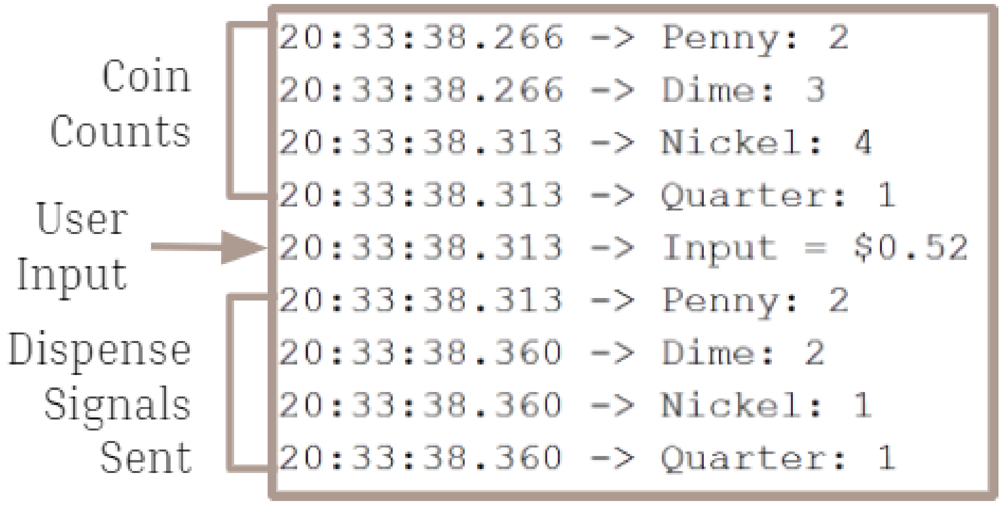


Figure 3.6 Arduino Output of Dispensing Signals

3.2.2 Solenoids

Our requirements and verification table in Appendix A specifies that the solenoid must have a minimum stroke length of 25 mm. We tested this by simply dispensing a few coins with the device. Figure 3.7 showcases our solenoids in action. They have quite a bit of force to them!

Figure 3.7.1: Solenoids in Idle State

Figure 3.7.2: Solenoids in Dispensing State

3.3 Control Unit Module

3.3.1 User Interface

The verification of the user interface was done using the instructions in our requirements and verification table in Appendix A. The OLED will show the number of coins in the wallet. The user will also be able to see their input on the OLED screen.

Our requirements and verification table specifies that the user interface must communicate with the microcontroller within three seconds. Figure 3.8 shows our team verifying the functionality of the user interface. The OLED screen updates almost instantly. All the numbers, backspace, and enter all work as intended. See Figure 3.9 for the placement of our backspace and enter buttons.

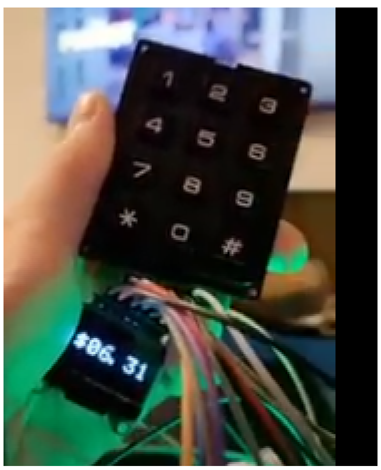


Figure 3.8 User Interface Verification

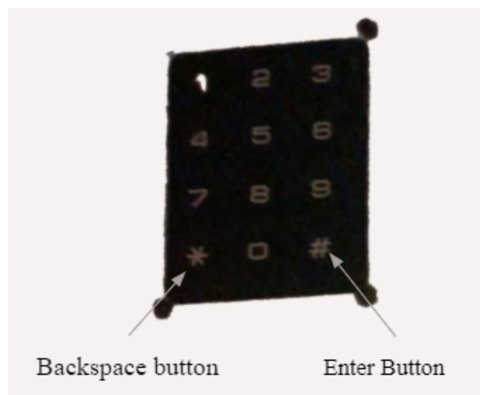


Figure 3.9 Keypad Instructions

3.3.2 Primary Atmega328-PU

The primary microcontroller was verified with the control unit, IR sensor, and dispensing module. To verify our program, our team started by loading in coins. Then, we entered a valid amount on the keypad and pressed the enter button. Lastly, the program decremented the coins in the wallet.

Figure 3.10.1 shows the coins being loaded and the coin counts being updated on the OLED display. The user then inputted a desired amount from the wallet as shown in Figure 3.10.2. Lastly, Figure 3.10.3 shows the coin counts decreasing on the OLED display to signal coins being dispensed. The coin counter decreases quarter, nickel, and penny counts from four to three, one to zero, and one to zero respectively, indicating the dispensing of one coin of each type to match the user input (0.25 + 0.05 + 0.01 = 0.31).

Figure 3.10.1 OLED Loading

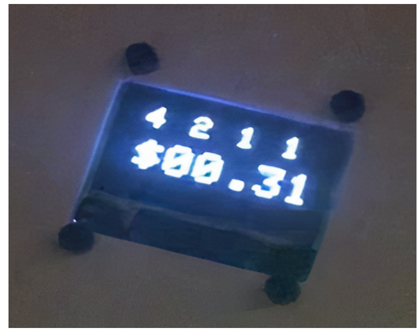


Figure 3.10.2 OLED with User Input

Figure 3.10.3 OLED Dispensing

3.3.3 Secondary Atmega328-PU

The secondary microcontroller was verified through the incrementing and decrementing of our coin counts. Section 3.2.2 will showcase our full functionality of both microcontrollers.

3.4 Power Supply Module

Due to time constraints, we were unable to verify both the lithium-ion and alkaline batteries.

4. Costs

4.1 Parts

Table 4.1 depicts our parts and costs table for Pocket Pal. The total amount of money we spent on parts amounted to $127.51.

Table 4.1 Parts and Costs

Part	Manufacturer	Quantity	Unit Cost
Microcontroller	Microchip Technology/ATMEGA328P-PU	2	$5.95
IR sensors	Vishay Semiconductor Opto Division/TSOP34838	4	$1.59
IR emitters	Everlight Electronics Co Ltd/IR333-A	4	$0.32
Solenoids	Abletop Push Pull Solenoid DC 12V 35mm Long	4	$19.95
OLED screen	AOM12864A0-0.96WW-ANO	1	$7.50
12 Button Keypad	Sparkfun Electronics/COM14662	2	$4.50
Portable battery (Duracell - CopperTop 12V Alkaline Batteries)	Duracell Inc	8	$0.50
Lithium-Ion Polymer Battery	TinyCircuits/ASR00011	2	$3.49
Voltage Regulator	STMicroelectronics/L78S05CV	1	$0.69
Total			$127.51

4.2 Labor

Each person in the group worked at least ten hours a week. We are using $20/hour to estimate our hourly wage. The development time of our project was approximately 12 weeks. The total labor cost for our team is shown in Eq. 1.

$$2.5 * (3 \text{ students} * \$50/\text{hour} * 10\text{hour} / \text{week} * 12 \text{ weeks}) = \$45{,}267 \quad (1)$$

The machine shop spent about three hours on constructing our design, with an estimated $50/hour wage. Their labor cost is shown in Eq. 2 below.

$$\$50/\text{hour} * 3 \text{ hours} = \$150 \tag{2}$$

4.3 Total Costs

The total cost is the sum of student labor costs, machine shop labour costs, and parts costs, as shown in Eq. 3. The total cost of Pocket Pal amounted to $45, 544.51.

$$\$45{,}267 + \$150 + \$127.51 = \$45{,}544.51 \tag{3}$$

5. Conclusion

5.1 Accomplishments

Our IR Loader Module, Dispensing Module, and Control Unit Modules all work as intended.

5.2 Uncertainties

Our team encountered a lot of uncertainties and challenges during the project. The original OLED that we ordered had 32 pins and far exceeded the number of available pins on the microcontrollers. The machine shop also had difficulties mounting the original OLED. We bought a SSD1306 OLED to reduce our overall pin usage.

The second uncertainty that we encountered was the purchasing of the solenoids. The 35 mm stroke solenoids were very expensive and difficult to obtain. Our first and second order for these four solenoids were canceled due to shipping cycles and delays. The machine shop had to purchase the necessary solenoids for our project.

Due to the nature of our project, we needed solenoids with a minimum of 25 mm stroke. This resulted in buying very specific solenoids, with a power rating of 12 V 8 A. Our 12V battery pack was not suitable for use as the Alkaline batteries were not able to sustain 8A current for extended periods.

5.3 Ethical considerations

Our primary safety concern with Pocket Pal is the overheating of the batteries. Aside from the solenoids, all of the other components have a low power consumption in an attempt to reduce the risk of a fire hazard.

Portability of the wallet is important to us. The batteries used for the solenoid were dry cell alkaline batteries, which are safe to carry on flights [11]. The smaller Lithium-ion batteries can also be safely carried on-board when installed in the device [11].

As stated in the IEEE code of ethics, we have a responsibility to “improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems” [12]. Through Pocket Pal, we will be able to showcase these capabilities by increasing the flexibility of coin usage for smaller payments.

Pocket Pal will not distribute any collected information as per IEEE Code of Ethics II.9., stating that we have the responsibility “to avoid injuring others, their property, reputation, or employment” [13]. We take privacy very seriously and will never disclose the amount of coins that a user is currently carrying in their wallet.

5.4 Future work

Our team believes that Pocket Pal can be a commercially viable product.

The size of the wallet can be optimized to satisfy the original high level requirements of 6" x 4" x 1.5". The coins can be inserted vertically to eliminate the need for a long stroke solenoid. Pushing out vertically facing coins will only require solenoids with a stroke longer than the thickness of the coins. Therefore, we can use 5mm stroke solenoids which can operate on 5 V and will be operating at the same voltage as the other components in the device.

The power unit hampered us a lot because of the high power consumption. Smaller stroke solenoids will consume less power. This will help reduce the risk of overheating. For example, our team looked at 10 mm stroke solenoids that consume 5 V [14].

For the allotted time in the course, our team only implemented coin loading and dispensing. With more time, we would try and incorporate bill dispensing with gears into Pocket Pal's design. Having both bill and coin dispensing will increase the usability and flexibility of Pocket Pal.

References

[1] R. K. and S. O’Brien, “2019 Findings from the Diary of Consumer Payment Choice,” Federal Reserve Bank of San Francisco, 26-Jun-2019. [Online]. Available: https://www.frbsf.org/cash/publications/fed-notes/2019/june/2019-findings-from-the-diary-of-consumer-payment-choice/. [Accessed: 19-Feb-2021].

[2] E. Nicholson, “Pandemic Or Not, Germans Still Prefer Cash,” NPR, 11-Aug-2020. [Online].Available:https://www.npr.org/2020/08/11/899343670/pandemic-or-not-germans-still-prefer-cash#:~:text=%22With%20cash%2C%20it's%20easier%20to,stores%20and%20restaurants%20do%20not. [Accessed: 19-Feb-2021].

[3] Coin Sorter Wallet. [Online]. Available: http://coinsorterwallet.com/. [Accessed: 03-Mar-2021].

[4] Kickstarter. [Online]. Available: https://www.kickstarter.com/projects/marlin/radical-wallets-that-sort-your-coins-and-notes-ins/comments. [Accessed: 03-Mar-2021].

[5] “Types of Distance Sensor and How to select one?,” Latest open tech from seeed studio, 05-Jan-2021. [Online]. Available: https://www.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/. [Accessed: 05-May-2021].

[6] “IR333-A Datasheet by Everlight Electronics Co Ltd,” Digi. [Online]. Available: https://www.digikey.com/htmldatasheets/production/852966/0/0/1/ir333-a.html. [Accessed: 05-May-2021].

[9] “Coin Specifications,” *United States Mint*. [Online]. Available: https://www.usmint.gov/learn/coin-and-medal-programs/coin-specifications. [Accessed: 05-May-2021].

[10] “Abletop Push Pull Solenoid DC 12V 35mm Long Stroke Small Electromagnetic Electric Magnet For Automobiles”,Amazon Electronics. Available: https://www.amazon.com/Abletop-Solenoid-Electromagnetic-Electric-Automobiles/dp/B07G15X91N [Accessed: 05-May-2021].

[11] “Travel Tips Tuesday: Safely Packing Batteries for Your Trip,” Travel Tips Tuesday: Safely Packing Batteries for Your Trip | Transportation Security Administration, 11-Jun-2013. [Online]. Available: https://www.tsa.gov/blog/2013/06/11/travel-tips-tuesday-safely-packing-batteries-your-trip. [Accessed: 04-Mar-2021].

[12] “IEEE Code of Ethics,” IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 19-Feb-2021].

[13] Fact Sheet – Lithium Batteries in Baggage, 20-Mar-2020. [Online]. Available: https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=23054#:~:text=Spare%20(uninstalled)%20lithium%20metal%20batteries,passenger%20in%20carry%2Don%20baggage.&text=Even%20in%20carry%2Don%20baggage,accidental%20activation%20and%20short%20circuits. [Accessed: 05-May-2021].

[14] A. Industries, “Medium Push-Pull Solenoid - 5V or 6V,” adafruit industries blog RSS. [Online]. Available: https://www.adafruit.com/product/3992?gclid=CjwKCAjwhMmEBhBwEiwAXwFoEcR2lyZJcIClg6UypGzBwGS-Ak_TF-_1V6zr_fLK7qvxUNDxcACNuxoCy3kQAvD_BwE. [Accessed: 05-May-2021].

Appendix A Requirements and Verification Table

Power Supply A. Charging the battery at maximum current and voltage can be sustained below 85°C to ensure the user's safety. B. Battery must be able to last for eight hours, which is the typical length of a work day. C. Battery must be able to supply 8 A of current to the required solenoid at all times.	A. Connect the battery to all relevant Pocket Pal components, as specified in our block diagram (Figure 2.1). Power the battery at full capacity (12 V) without limiting current. With a voltmeter, measure the voltage across the battery and ensure it is running at 12 volts. Remove the voltmeter. Without modifying the current setup, leave the battery running for 8 hours without human intervention. Using an IR thermometer, check that the battery temperature after eight hours is less than 85°C. B. Connect the battery to all relevant Pocket Pal components, as specified in our block diagram (Figure 2.1). Power the battery at full capacity (12 v) without limiting current. With a voltmeter, measure the voltage across the battery and ensure it is running at 12 volts. Leave the battery running for 8 hours without any human intervention. After eight hours, use a voltmeter to measure the voltage across the battery and make sure it is still running at 12 volts. C. Connect a solenoid parallel to the diode and n-channel MOSFET powered by a 12 V battery. Measure current around the terminals of solenoid using an ammeter/oscilloscope for 10 minutes. Current should be >8 A at all instances.
Control Unit (User Interface)	A. Using the keypad, enter a cash value into the user input and start a timer.

A. The OLED screen must communicate with the microcontroller and begin the dispensing process within 3 seconds after the user inputs the desired cash amount.	Have your partner write down the time in which they hear the first coin drop into the collecting tray. Stop the timer when the last coin has been dispensed. Subtract the time recorded in step two from the time recorded in step three, as this is the time it takes for the OLED screen to communicate with microcontroller and begin the dispensing process.(Figure 3.8)
Control Unit (Microcontroller) A. Microcontroller (16Hz internal) should consume less than 22mW with maximum current supplied. B. Microcontroller should supply current to the required solenoid for dispensing when the user inputs an amount.	A. Remove loads and connect the 12 V power supply to the microcontroller. Using an ammeter, measure the current across the microcontroller terminals and observe whether current is < 0.2 mA for 8 V supply. B. Enter some monetary value ($0.01, $0.05, $0.10, $0.25 and other combinations) into the keypad. Wait until all coins are dispensed, and verify that the correct number of coins were released by writing calculations on paper. If the exact monetary value was released from Pocket Pal, we know that the microcontroller correctly supplied current to the required solenoids.
Dispenser Module (Coin dispenser) A. Dispenser releases exactly one coin in one push and pull motion.	A. For one coin compartment, check that it has two or more coins inside of it. Enter the value of one coin (from the same coin compartment in step one) into the keypad (ie: $0.05 for a single Nickel) After Pocket Pal dispenses, ensure that only one coin of that type was released from the device when there were multiple coins in the compartment. If there is more than one coin in this compartment, enter the value of the excess coins so that only one coin will be left in the compartment. After Pocket Pal dispenses, ensure that

	only one coin of that type was released from the device when only a single coin was in the compartment. Repeat steps one through 5 for each coin compartment. (Figure 3.5)
Coin Dispenser Module (Solenoids) A. Push-pull solenoids' plunger should "push," at minimum, the diameter of the coin for each compartment.	A. Supply minimum current (to a solenoid ($F \propto 1/x^2$) to determine maximum stroke for each solenoid. Maximum stroke needs to be greater than the maximum diameter of coins (24.5mm - Quarters). (Figure 3.7)
Coin Loader Module (IR Sensors) A. IR receivers output 5.3 **[changed to 3.7]** volts whenever a coin passes over the IR emitter. For the IR receiver datasheet, see [7]. The IR emitter is found at [8]. B. IR receivers output -0.3 **[changed to LOW signal]** volts at all times, unless a coin is passing over the IR emitter. The IR receiver never outputs 5.3 **[changed to 3.7]** volts unintentionally due to the push-pull mechanism of the solenoid moving coins up and down.	A. With a voltmeter, measure the voltage of the IR emitter for each compartment, and ensure it is five **[changed to 3.7]** volts and powered on. With a voltmeter, measure the voltage of the IR receiver for each compartment, and make sure it is -0.3 **[changed to LOW signal]** volts and powered on. For one coin type, insert a single coin. Using a voltmeter, measure the voltage output of the IR receiver for that coin type's compartment, and make sure the receiver outputs 5.3 **[changed to 3.7]** volts when the coin passes over the IR emitter. Repeat steps three and four for each coin type.(Figure 3.2) B. Using a voltmeter, measure the voltage of the IR emitter for each compartment, and ensure it is five **[changed to 3.7]** volts and powered on. With a voltmeter, measure the voltage of the IR receiver for each compartment, and make sure it is -0.3 **[changed to LOW signal]** volts and powered on. For one coin compartment, supply maximum power to its solenoid at maximum voltage (12 V). Measure the voltage output of the IR

	receiver for the same coin compartment, and ensure the receiver always outputs -0.3 **[changed to LOW signal]** volts. Repeat steps three and four for each coin type.
Coin Loader Module (Coin Loader) [**No longer applicable]** A. Loader sorts a mix including all four types of coins (quarter, nickel, dime, penny) with 95% accuracy.	A. Collect four coins of each coin type (16 total) and put them in a pile. Grab a single coin at random, and insert it through the coin loader slot. Repeat step 2 for the other 15 coins. Count how many coins were successfully sorted into the correct coin compartment, and record the ratio of correct coins to total coins (e.g. 14 coins were sorted correctly, so the ratio is 14/16). Remove all 16 coins from the device and put them in a pile. Repeat steps two through four two more times. Calculate the average percentage of all three runs, which is the value we will use to determine our coin loader's accuracy.