Cryptocurrency Bot On Raspberry Pi 4

Samuel White Computer Science

Colorado State University

Fort Collins, United States of America

[Samuel.White@colostate.edu](mailto:Samuel.White@colostate.edu)

Samuel McKay Computer Science

Colorado State University

Fort Collins, United States of America

[Sam.McKay@colostate.edu](mailto:Sam.McKay@colostate.edu)

Ernest Duckworth

Computer Science

Colorado State University

Fort Collins, United States of America

[Ernest.Duckworth@colostate.edu](mailto:Ernest.Duckworth@colostate.edu)

***Abstract*—Throughout most of the world, many people live paycheck to paycheck. Some people are not even that privileged and have to make a paycheck last as long as it can before the money runs out. However, there are ways to make passive income. One of these methods is trading cryptocurrencies like BitCoin, Ethereum, and Dogecoin to name a few. With a few simple steps and a little bit of money, the average user can get a Cryptocurrency automated trader up and running and, hopefully, it will help alleviate some financial trouble.**

***Keywords—Raspberry Pi, Cryptocurrency, Bitcoin, Trading, Python, CS370***

I. INTRODUCTION

Cryptocurrency has been around for over a decade but only recently has it gained any attraction with its use. In summary, instead of having a centralized government deciding the price of a dollar or coin, groups of people can decide how much one unit of currency is worth. Getting into the nitty-gritty details of how this works is quite complicated, but for basic use cases, Cryptocurrency can be used by everyday users. Another potential benefit of using Cryptocurrency compared to, typical money like the U.S. Dollar or Rubles, is transactions are all anonymous. There is still a log of everything transaction and with enough technical knowledge someone could find who used a Cryptocurrency, but this anonymity provides another security factor for the users. Cryptocurrencies also have another benefit, they can be traded like stocks. The goal of this development project is to use a Raspberry Pi 4 to run a program, or “bot”, to trade these currencies for the user without any human interference.

II. Problem Characterization

*A. Identifying the Problem*

One of the challenges with Cryptocurrencies is they are very volatile. Unlike the stock market opening and closing on specific days of the week and on certain holidays, these digital currencies never stop changing prices. This is because there is no governing body to stop trading or changing the price. Anyone at any time can invest and change the price.

This can create some challenges if someone is trying to make money off of trading digital currencies or assets. Overall they are unpredictable and can change on a whim. In order to have a chance of making money, someone or something has to constantly be monitoring the price to determine when to buy or sell. Giant hedge funds spend millions of dollars to develop algorithms and hire people to try and create new and effective ways to better predict the future price of stocks. However, not everybody can spend that type of money, but people can still get involved in trading stock. Some people buy a couple of shares of certain companies and hold onto them for a couple of years in hopes they go up. These are long-term investments. Others become day-traders where they actively monitor the market, every minute of every hour when they get the chance. The hope is to make small amounts of money every so often and in the long run, have it add up to a net positive amount.

This is a great strategy but overall it is very time-consuming. The majority of the population does not have the time to sit around and watch the market all day. They especially do not have time to watch Cryptocurrencies which are even more unpredictable than stocks. These currencies are changing every day of the year, regardless of which holiday it is. It is almost impossible for one person to accurately predict the future prices of these currencies. The question that is asked is, is it possible for someone to utilize cryptocurrencies without having to constantly monitor the market? Moving forward, we analyze how a Raspberry Pi 4 can help answer this question.

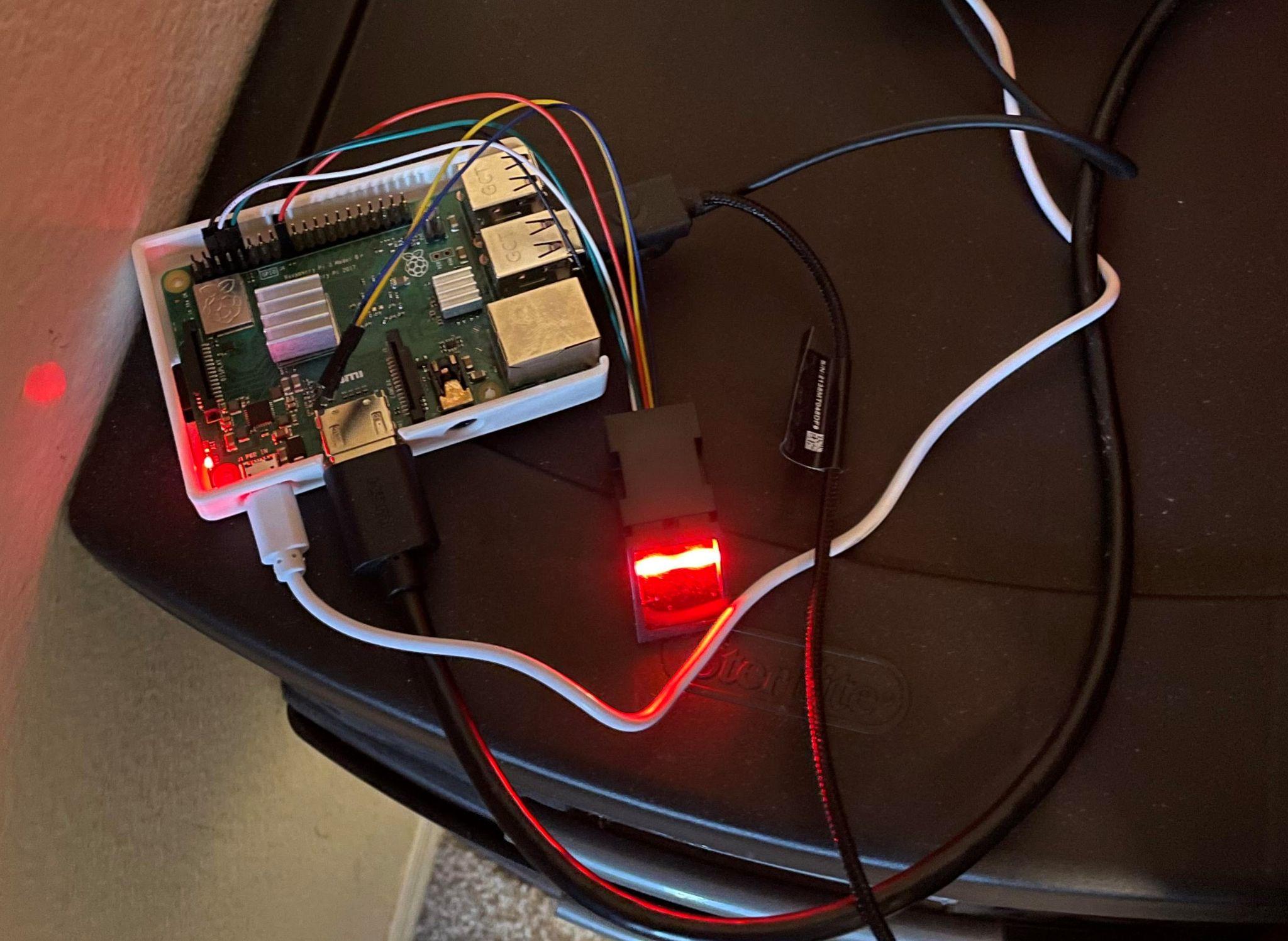


Fig. 1. Raspberry Pi 4 with Finger Print Scanner

III. PROPOSED SOLUTION AND IMPLEMENTATION STRATEGY

*A. Automation of Trading*

In order to circumvent the shortcomings of humans, we decided to write some software that will use an API to get Cryptocurrency data every so often. Based on this data the software will determine whether or not it should buy or sell the currency. The hardware that will be running this software is a Raspberry Pi 4. This machine is a small computer that can fit into the palm of a hand. Even though it is so small, it still has four CPU cores and eight gigabytes of memory. This is comparable to a modern computer, but it overall draws less power. In comparison to typical computers, it is also much cheaper at around $50. This software could be run on any computer but we chose the cheaper and more portable Raspberry Pi.

*B. Data Collection*

In order to access the current digital coin prices, we will be using CCXT, which is an API that can be used with Python. We can get a lot of data from this API but some of the ones that are used in this project are Cryptocurrency symbols/tickers, orderbooks, account information and most importantly candles.

A candle is one of the most important representations of data for a coin that is used. They are used in the stock market as well as in Cryptocurrencies. A candle is a chart that represents the high, low, open, close and volume values of a specific stock or coin over a specified period of time. These values are heavily used in data analysis. They are the data that is analyzed to create indicators relevant to determining specific buy/sell entry/exit points. They give the analyzer a brief, but important, summary of the coin and how it is behaving. Based on this previous data, the bot will analyze indicator(s) to better determine if it is in an entry, exit, or neutral point. It will also periodically check to make sure that we have not hit our stop loss on any specific investments.

*C. Programming Approach*

The Raspberry Pi 4 has four cores which means it is possible to run four threads simultaneously. The four major components of the bot are the GUI, Coin Scanner, Coin Analyzer, and Coin Trader. Each of these components will be run on a separate thread.

The Graphical User Interface plays an important part in how the user will interact with the bot. Since there are several buttons and input boxes the library of choice was Tkinter. Tkinter gives the programmer the option to create widgets that intertact with the bot and can be rendered independently of one another. The bot will do the trading on its own, but the user will still need to specify which coins to trade and how much money to give the bot. Once the bot has been started the graph will update once every thirty seconds, tracking the liquidated assets of the user. Once the GUI is stopped, through the stop button or by closing out of the window, all remaining stocks held by the user will be sold and all threads will be killed.

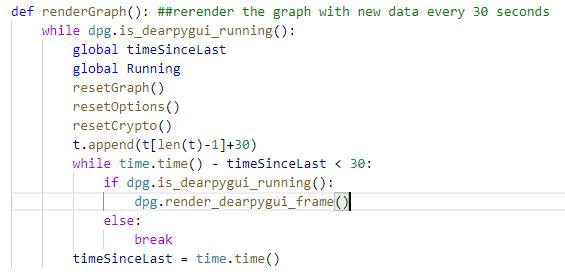


Fig. 2. Part of Tkinter GUI code

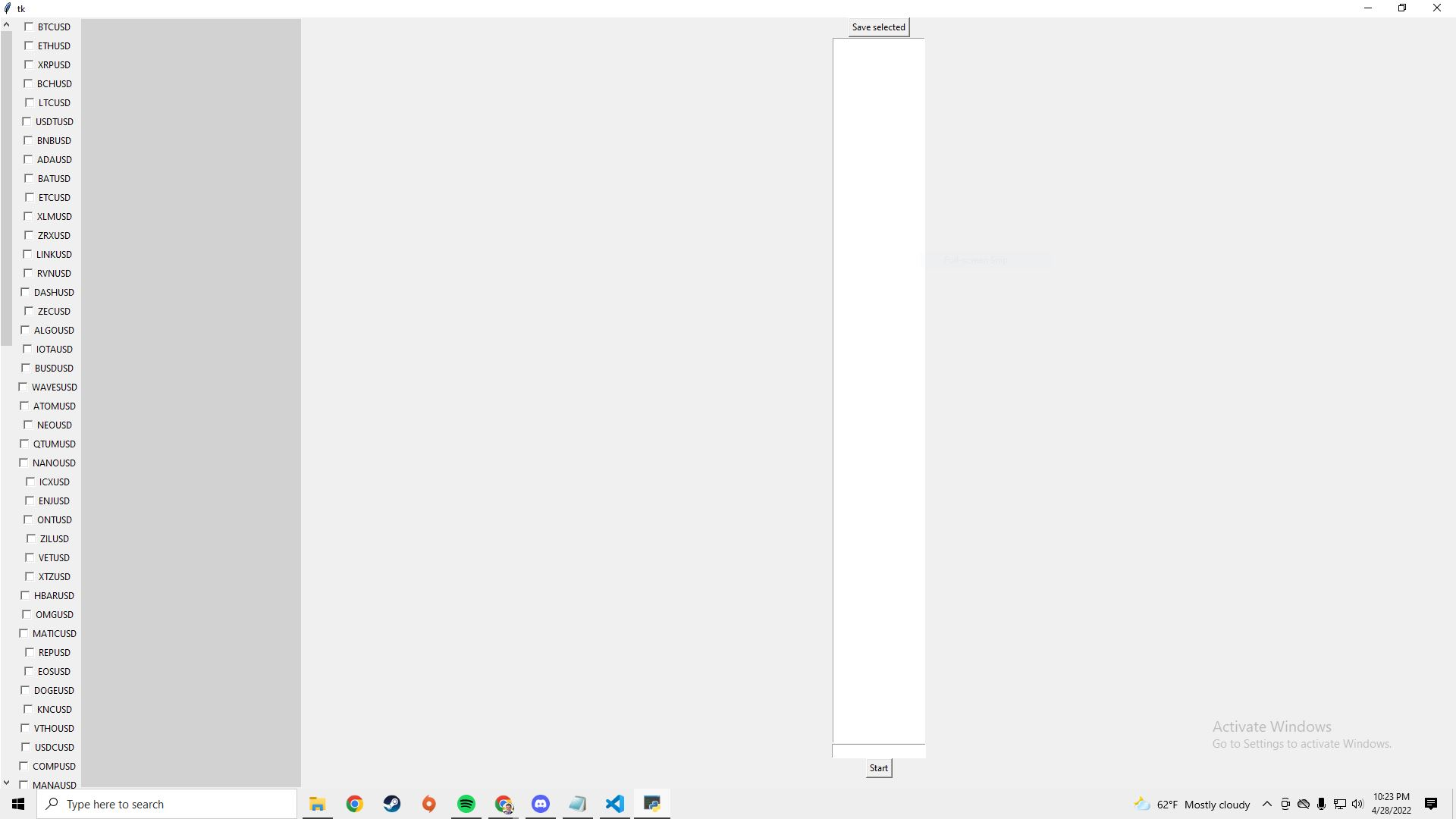


Fig. 3. Current GUI Implementation

Fig. 1. Current Graphical User Interphase being Used

The Coin Scanner's sole purpose is to scan the current BinanceUS market for the coin data. The data is stored in the model ‘CoinData’. CoinData is specific to a single coin/usd pair. It is responsible for holding all information about the coins candles aswell as the indicators for the corresponding candles. BinanceUS is the current crypto trading platform of choice. There are others that are available such as CoinBase and Kraken but BinanceUS has a simplistic API key generation and setup. Scanner will gather the candles and insert them into a Python Pandas data frame. Then it will use that candle data to generate its corresponding indicator data for numerous indicators such as the RSI. From this moment, it will add the current coin’s ticker to a queue. With four processes running at the same time and in order to ensure synchronization, a custom queue was implemented. This will generate a typical consumer and producer problem where the scanner is the producer and the analyzer will be the consumer.

The queue simulates a modified version of the First in First out algorithm. The producer puts the ticker at the back of the queue; however, if the producer tries to insert another ticker that is already prevalent in the queue, that ticker will jump to the front and the second copy will not be added. Since it has been seen twice, it is prioritized, and the analyzer needs to catch up with two new data sets for the coin.

The analyzer will consume the current/most prevalent ticker from the queue. Based on the ticker, the analyzer will access the shared memory model ‘CoinData’ between the scanner to create a deep copy of its Candles and Indicators for the given coin. From this point on the analyzers purpose is to determine if the coin needs to be bought or sold. The most prevalent way this can be done is through the RSI, relative strength index. This is one of the most common momentum indicators used in trading. It is using a straight forward strategy if the RSI is low it is likely in an oversold position thus we want to enter. If the RSI is high it is in an overbought position thus if we have a position in the coin we would sell. If its in a neutral zone then wait for it to enter a high/low position. Once the analyzer has made its decision, it passes this information onto the shared queue between trader. In the form of an ‘order’. The ‘order’ is compromised of 3 things: coin ticker(ex: BTCUSD), type (buy, sell or wait) and weight (confidence in trade calculated from the level of the RSI).

The trader thread’s purpose is to buy, sell, or wait on the current coin in question. It accesses the CCXT API and executes these trades instructed by the analyzer. As it reads orders from the shared queue it determines if the order is valid. This can mean a number of things. If the order is a buy, it checks to see if there is any free USD to make any trades with currently. It also checks to see if we are already in a position for the coin, as we don’t want to have multiple entries for the same coin. If the order is a sell, it checks to make sure that we have a valid amount of the coin. If we are waiting, it still checks to see if we are in a position. If we are in a position on the coin we are waiting on, we double check to make sure we haven’t hit our stop loss. If we have hit our stop loss we exit the position. After the trader has executed a buy and sell pair on the same coin it records its success rate on trades and it calculates its average gain as a percentage.

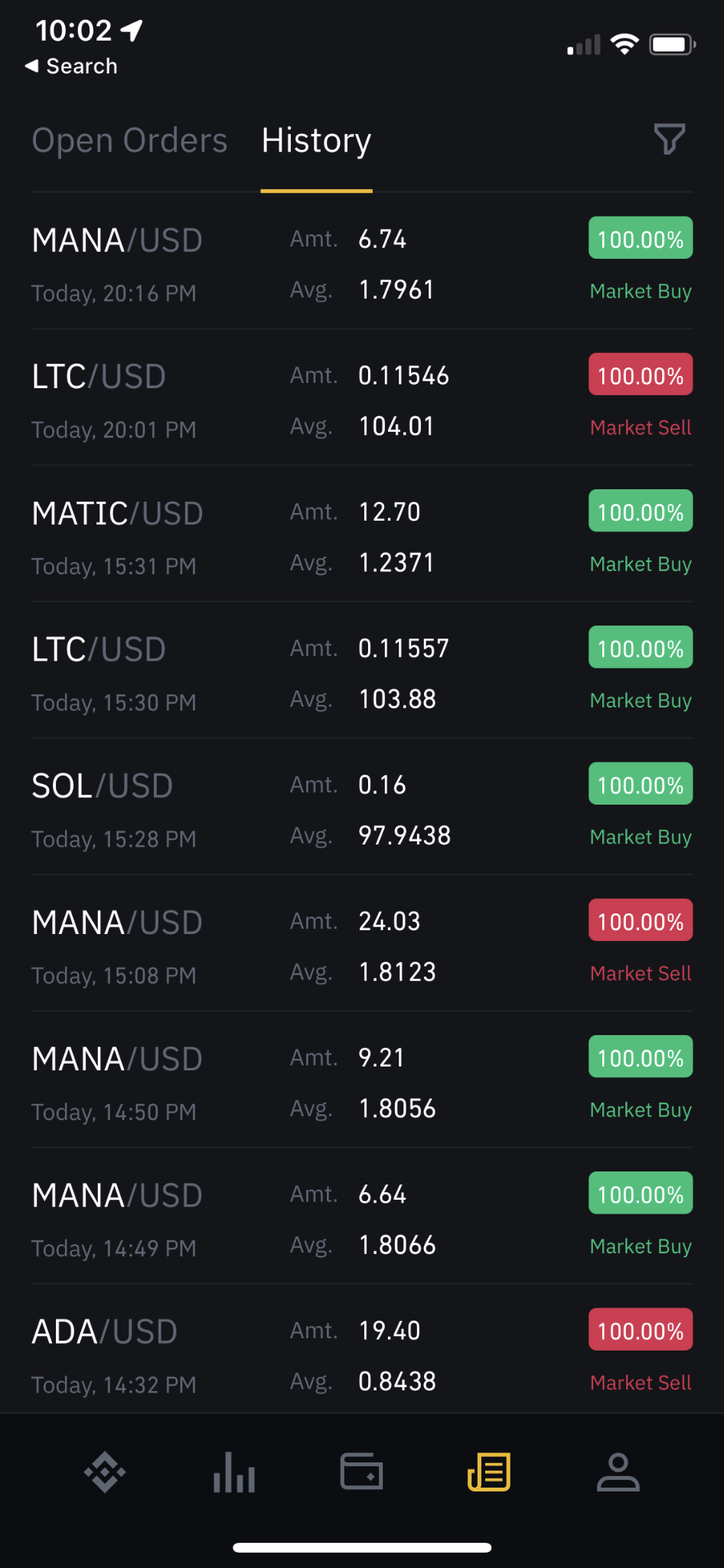


Fig. 4. Bot’s Trading History

*D. Security*

With the Raspberry Pi 4 having access to personal data and money from the user, there needs to be security. The Pi itself has a username and password but in order to add an extra layer of security, a fingerprint scanner was implemented through the use of the python-fingerprint library. Due to the constraints of serial to USB communication the fingerprint scanner is wired directly to the board through the use of pins. The fingerprint sensor will be able to recognize more than one user with different profile settings. To log in to any specific account, the user will need to scan their fingerprint. The fingerprint scanner itself stores an image file that has been converted to key characteristics. The program then searches the enrolled templates to see if there is a match between the new finger and previous fingers. If a match does occur, the code starts the GUI using that user’s config file. If there is no matching fingerprint the user is enrolled in the sensor, then is prompted to create a config file to connect to Binance.

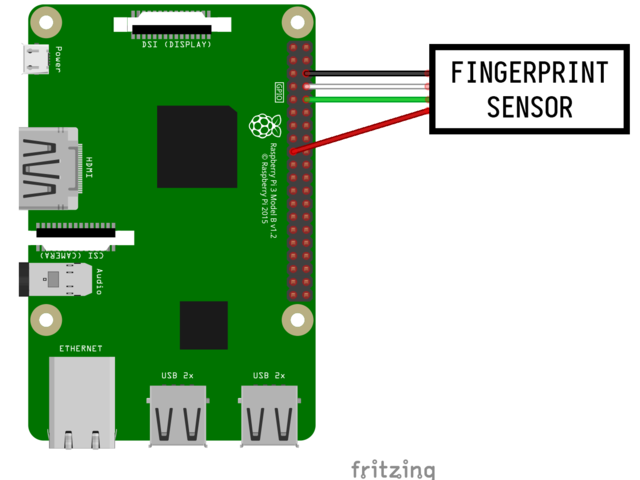


Fig. 5. Finger Print Scanner Pins

IV. Conclusion

Most people on Earth live paycheck to paycheck. Having the opportunity to make passive income could greatly benefit many people. Having a cheap, automated solution is what we tried to create. The volatility of most cryptocurrencies leads many people to the conclusion that the only way to make a profit trading cryptos is to carefully watch the performance of each. In a trading system, this leaves crypto currencies undesirable to the general population, unless they have the time to buy and sell manually, or they have a mining rig to generate these cryptos for a long-term investment that has large dividends. By implementing multithreading and using an API that communicates with BinanceUS we are able to solve a modified version of the producer-consumer problem. Since the threads perform their jobs without input from the user we hope to create a service that can trade cryptos for an hour, day, or even years.

REFERENCES

[1] “Tkinter - Python interface to TCL/TK¶,” *tkinter - Python interface to Tcl/Tk - Python 3.10.4 documentation*. [Online]. Available: https://docs.python.org/3/library/tkinter.html. [Accessed: 28-Apr-2022].

[2] “How to use a Raspberry Pi fingerprint sensor for Authentication,” *How to use a Raspberry Pi Fingerprint Sensor for Authentication*, 17-Aug-2020. [Online]. Available: https://tutorials-raspberrypi.com/how-to-use-raspberry-pi-fingerprint-sensor-authentication/. [Accessed: 27-Apr-2022].

[3] L. Ada, “Adafruit optical fingerprint sensor,” *Adafruit Learning System*, 31-Oct-2019. [Online]. Available: https://learn.adafruit.com/adafruit-optical-fingerprint-sensor/circuitpython. [Accessed: 27-Apr-2022].

[4] “Exchanges,” *Exchanges - ccxt 1.80.93 documentation*. [Online]. Available: https://docs.ccxt.com/en/latest/manual.html. [Accessed: 28-Apr-2022].

[5] “Pandas.dataframe¶,” *pandas.DataFrame - pandas 1.4.2 documentation*. [Online]. Available: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html. [Accessed: 28-Apr-2022].

[6] “Threading - thread-based parallelism¶,” *threading - Thread-based parallelism - Python 3.10.4 documentation*. [Online]. Available: https://docs.python.org/3/library/threading.html. [Accessed: 28-Apr-2022].

[7] “Dear Pygui's documentation,” *Dear PyGui's Documentation - Dear PyGui documentation*, 26-Oct-2021. [Online]. Available: https://dearpygui.readthedocs.io/en/latest/index.html. [Accessed: 27-Apr-2022].