

CyberTrade: Stock Bot

Abstract

CyberTrade allows stock traders to buy, sell, and trade stocks automatically using a bot that executes trades based on a specific trading strategy, with the goal of automating profits. Most investors don't have the time to dedicate to always getting the best trade—something that bots can do. The main goal of CyberTrade is to increase revenue and reduce losses and risks with as little work from the user as possible. There is a long list of advantages to having a bot monitor the markets—minimized emotions, improved order entry speed, and preserved discipline (Folger, 2).

Overall Description of Project Idea.

For the automated bot to function, it will require several inputs from the user: their card, the tick symbol of their stock interest, and the number of stocks they want to buy/sell. In addition, we will also need 2 Arduino Boards and Breadboards, where one Arduino will be used for user inputs and the second Arduino for output the result from the bot.

CyberTrade will start its analysis through the data extracted from Yahoo Finance contingent on the user's inputs. The bot will indicate to them when they can make the most profit in the time range input. It will then calculate the closing values per each time period to acquire the RSI value. Depending on this, it will signal the user to buy or sell the stock for maximized profit. For those individuals that are preoccupied, there is a feature that allows them to toggle the buzzer for when they are away from the system.

The purpose of CyberTrade is for stock traders or anyone with an interest in stocks to benefit from buying, selling, and trading stocks automatically using a bot that performs these trades based on a specific trading strategy, to increase revenue and reduce losses and risks with as little work from the user as possible. This is especially useful for those who are new to stock trading and don't have the knowledge to make a significant profit. And also for those investors who don't have time on their hands to get the best trade.

Description of each subsystem.

Automated Bot — RFID and LCD

The card will contain data regarding money, the stocks they acquire, and details regarding it, which are outputted on the LCD when scanned. The automation of the system will be based on the Relative Strength Index (RSI) of the stock of the user's choice. This is a technical indicator used in the analysis of financial markets to chart the current and historical strength and weakness of a stock based on the closing prices of a recent trading period (Fernando, 2022). Depending on this technical analysis' result, the bot will buy or sell the number of stocks the user intends, given that it is plausible in relation to the money that is stored in their card. Once this has occurred, it will output the activity on the LCD, which will be accompanied by the other notification subsystem.

Notification System — Red and Green LED Blink Lights, Buzzer, and Buttons

For this part of the project, two LED lights of different colors will light up to represent how the user's choice of stock's price is performing in real-time. Red LED will blink if the

price of the stock underperforms and decreases in value given the time period, and green LED inversely. This component's output completely relies on the user's input in the first part, Automated Bot. Adding on, at any moment that there is new activity (bearish -> bullish, buying/selling stocks), the buzzer should buzz. Since the user will not be constantly around the system, the buzzer will be utilized as a notification system. In addition, there will be two buttons. Because of the noise the buzzer excludes, the user is given an option to toggle the sound system on and off based on their preference. The other button's purpose is to turn start the analyzer with the same functionalities except the buzzer noise.

The purpose of the first Arduino is to gather all the inputs from the user and the purpose of the second Arduino is to output the result from the bot to the user.

List of Input and Output devices per each subsystem.

*Subsystem 1. **Input devices.***

| Device | Behaviour | Output |
|--------------------------|--|---|
| RFID Card + RFID Scanner | The Card acts as the stock portfolio for the user. | Stock portfolio, with amount of stocks and their ticker symbols |
| Keyboard | Serial Communication from the user | Ticker Symbol |
| Keyboard | Serial Communication from the user | Number of Stocks |
| Button 1 | Serial Communication from the user | Turns on the analyzer |
| Button 2 | Serial Communication from the user | Toggles the buzzer based on the user's noise preference |

*Subsystem 1. **Output devices.***

| Device | Behaviour | Input |
|-----------|--|--|
| Red LED | The Red LED indicates that there is a decline in profit. | Result from the Bot when there is a loss |
| Green LED | The Green LED indicates that there is a decline in profit. | Result from the Bot when there is a profit |

| | | |
|--------|--|--|
| LCD | On Scanning the card, which has the information stored about the user's stock portfolio. It output the result on the LCD. | Prints of the stock portfolio, with Ticker Symbol and amount of stocks |
| Buzzer | Since the user is not always monitoring the bot, the buzzer will be used to alarm the user when it is the best time to make a profit | Alarming Sound from the buzzer. |

Communication Mechanism.

The NFC Cards that will be used have an RFID Chip, which will be read by an RFID Scanner. This card contains detailed information regarding the user: the amount of money they have on the card, the stocks, and detailed information—stock name, ticker, time purchased, price purchased, time sold, price sold, and the percentage and actual gain or loss. Once it is scanned, the user is asked to input more information, such as the tick symbol of their stock interest, and the amount of stocks they want to buy/sell. This is done through a keyboard, and serial communication, and will then start the program. Given the tick symbol, Yahoo Finance API will be used to acquire data to calculate the RSI. Concurrent with the calculation, red and green LED lights will be blinking depending on the performance of the stock. Once the signal to whether buy or sell has occurred, a message will be displayed through a LED and the details regarding the undertaking. All of these functionalities will be done through Serial Communication. The python script that handles the analysis of the RSI and the performance of the stock market writes to the INO file that handles the outputs. When the python script reads a signal that a valid RFID card is scanned, it sends out data regarding the user's portfolio and sends those to the file that handles the outputs by writing to the serial. Similarly, the input file sends data to the output file whenever either of the buttons were pressed. Thus, the output file does all the checking from the data that is being written to the serial, and handles accordingly through outputting through the LCD, the 2 LED lights, and the buzzer.

Similar projects (State of the art)

A project that has a similar idea is the Stocks Market Analyzer. Both projects are similar in the way that it analyzes stocks and allows the user to easily determine the best time to buy and sell stocks and it uses the same Yahoo Finance API (1Sheeld, 2015). Unlike this project, we are not going to be sending an SMS notification to the user, but rather

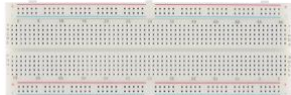


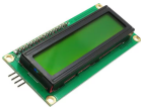
output the signal on the LCD concurrently with the buzzer. In addition, we are using technical analysis to recognize the signals, and this particular project does not.








Another project that touches upon the same topic, stocks, is the Stock Price Displayer (Chiu, 2021). Same as the project above, it has the same concept—stocks and uses the same Yahoo Finance API. In comparison to our project, the Stock Price Displayer is limited to outputting the ticker symbol and stock price on the LCD. In addition, the stock change throughout the day is implemented on a rocket ship that spins, depending on the stock's performance.

Statement of originality

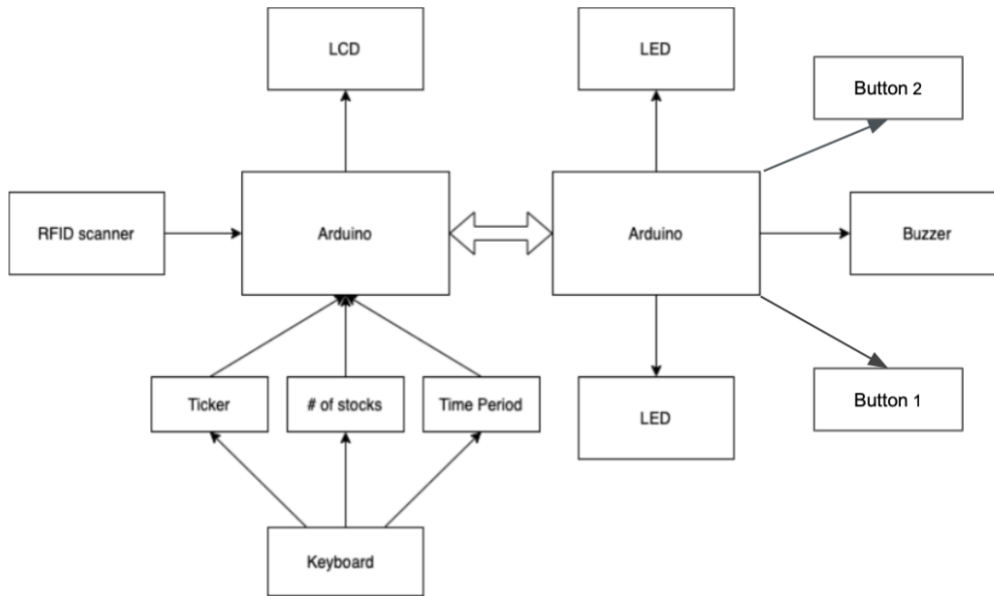
What differentiates this project from those found online is the simplified communication made possible with the use of an NFC Card to make any transactions. In addition, it does not only provide the current price of the stock but rather, additional features are included—time to best buy and sell a stock with an actual technical analyzer. In addition, we have a notification system that allows the user to toggle it on and off based on their preference to alert them about profitable Buy and Sell Timings depending on the RSI calculation.

List of Materials

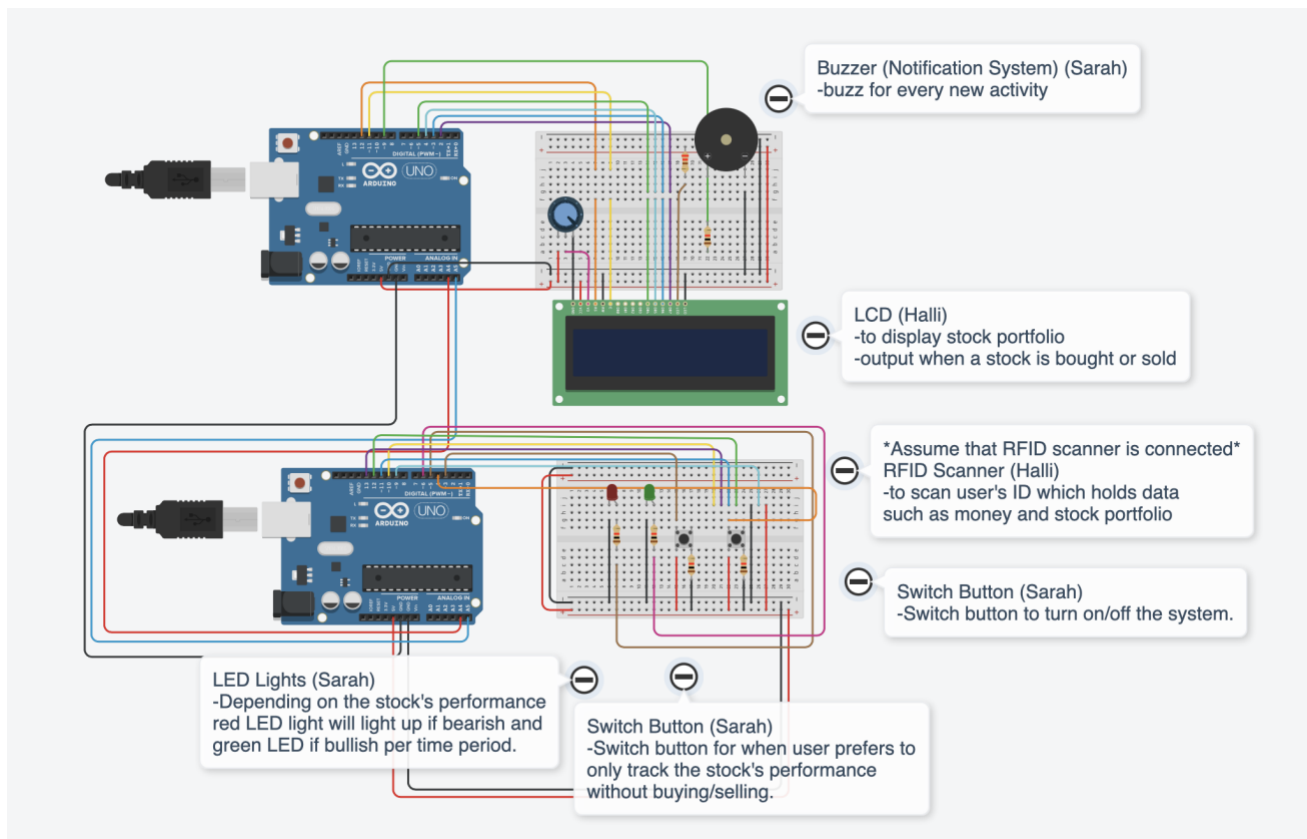
| | |
|---|---------------------|
|  | 2x Breadboard |
|  | 2x Arduino Uno |
|  | Green LED + Red LED |
|  | LCD Display |

| | |
|---|-----------------|
|  | Potentiometer |
|  | Buzzer |
|  | 2x Jumper Wires |
|  | 5x Resistors |
|  | RFID Card |
|  | RFID Scanner |
|  | 2x Buttons |

Block Diagram of the system (Big picture diagram).



Hardware diagrams



Steps to build your project

1. Extract data from Yahoo Finance using the YahooFinancials library based on a ticker, and for the time being work based on it.
2. Write a script that would automatically schedule tasks repeatedly at a specific time. In this case, run the code that extracts data for every 14 time interval
3. For every time that the program is ran and live market data is accessed, with a possibility of a second lag, the LEDs are lit up depending if it is bearish or bullish based on the market price from the previous time period.
4. Based on the performance, the previous data should be tracked. Contingent on this, it will send data to the output INO file to handle the LEDs, whether to set the green or red to high or low.
5. Analyze the data by using the panda library, specifically the exponentially weighted (EW) calculation using the exponential moving average. Calculation is as follows:

$$RS = \frac{Avg.Gain}{Avg.Loss}$$
$$RSI = 100 - \frac{100}{1 + RS}$$

Where:

- avg.gain is average gain over time window
 - avg.loss is average loss over time window
6. Based on the results, write to the serial based on the RSI if it is:
 - above 70, asset is considered overbought (overvalued) — generating a potential buy signal.
 - below 30, asset is considered oversold (undervalued) —generating a potential sell signal.
 7. When it is ensured that the program is correctly working based on the expected results (generates the correct signals), employ some of the user inputs: ticker, time period, and whether they'd want to buy or sell.
 8. Run and make certain that the program continues to serve its expected purpose, considering the possible edge cases.
 9. Utilize the RFID card and the scanner and implement it into the program— authenticating the card.
 10. Create a profile for the specific card. This will hold data—their money and stock portfolio.

11. When it is ensured that the card is authenticated, write a signal to the python script as it holds the stock portfolio and send that data to the arduino to be outputted on the LCD.
12. From the input INO file, send a signal to the output INO file whenever the yellow button is pressed. The output INO file should handle this, and ensure that the buzzer should buzz at any moment that there is a buy/sell signal. It should also handle the toggle of this button.
13. Now, similar to step 12, the output INO file should handle whenever the red button is pressed—it should start the analyzer without the buzz notification.
14. Lastly, once all objects are integrated, ensure that all possible edge cases are handled.

User guide

1. Given that the 2 Arduinos are connected to a Computer, the stock bot is ready to be used.
2. The user trying to use the stock bot has a Card that they choose to scan.
3. Upon scanning the Card the stock portfolio prints on the LCD, printing out the number of stocks they have and the price they bought it for each the ticker symbols. This can especially be helpful to make a decision when trading stocks.
4. The user is then prompted for input from the computer and enters their input using the keyboard.
5. Given a choice, they are then asked for the Ticker Symbol.
6. Upon this setup, the user is now ready to press the red button, which starts the stock bot analyzer.
7. Then, the bot then continues to perform the analysis and indicates to them how the stock they are tracking is performing using the red and green LED. If the green LED is lit, this implies that the stock price has gone up based on the previous time period, and otherwise for the red LED.
8. If the analyzer acquires data that affirms that it is the best time to buy and sell to make a profit, an alarm goes off by the buzzer and outputs on the LCD.
9. If anytime that the user would like to turn on the notification system incase they don't want to be constantly around the bot, they can push the Yellow button which will alert them by buzzing when it is the best time to Buy/Sell.

Youtube Link of Demo: <https://youtu.be/zXrwrnLCtmM>

Timeline

Proposed Timeline of Development

| Week | Objective for the Week | Milestone Deadlines |
|----------------|---|--|
| Week 8 | <ul style="list-style-type: none">-Get all the additional materials needed for the Project beforehand-Familiarize with the API to be used to execute the buy and selling-Scrap data from Yahoo Finance API given a period of time | Expo Registration Wed 3/2/22 |
| Week 9 | <ul style="list-style-type: none">-Red and Green LED Blink Lights with set stock-Analyze the RSI, and then execute the automated buy and sell | |
| Week 10 | <ul style="list-style-type: none">- Ensure that the Automated Bot is working with sample values | |
| Week 11 | Spring Break | Spring Break |
| Week 12 | <ul style="list-style-type: none">-Establish connection of the NFC cards and the RFID reader with the arduino-Reinforce the user-inputs aspect-Ensure the buttons are working as expected | Tue 3/29/22 |
| Week 13 | <ul style="list-style-type: none">-Finalize the functionality is obtained-Clean up and Error Checking-Prepare for Group | Fri 4/8/22 |

| | | |
|----------------|---|--------------------|
| | Presentation | |
| Week 14 | -Prepare for Expo and make sure project is ready to be demoed | |
| Week 15 | -Prepare for Expo and Finalize any details | Tue 4/19/22 |
| | -Prepare for Expo | Fri 4/22/22 |
| Week 16 | -Attend 3 Group Presentations and Evaluate -Get started on the Final Design Report | Mon 4/25/22 |
| | -Finish up Report to reflect all the details needed to recreate product | Wed 4/27/22 |
| | -Submit Teamwork Assessment | Thu 4/28/22 |
| | -Submit Final Design Document | |

References

- 1Sheeld. "Stocks market Analyzer." 4 October 2015,
<https://create.arduino.cc/projecthub/1sheeld/stocks-market-analyzer-a13900>
- Chiu, Sherwin. "Stock Price Displayer." 20 February, 2021,
<https://create.arduino.cc/projecthub/sherwinchiu89/stock-price-displayer-d2ccbe>
- Fernando, Jason. "Relative Strength Index (RSI)." *Investopedia*, Investopedia, 19 Feb. 2022, <https://www.investopedia.com/terms/r/rsi.asp>
- Folger, Jean. "Pros and Cons of Automated Trading Systems." *Investopedia*, Investopedia, 9 Feb. 2022,
<https://www.investopedia.com/articles/trading/11/automated-trading-systems.asp>.

Yo Ko, Chu. "Technical Analysis with R." 1 Jan 2020,
<https://bookdown.org/kochiuyu/technical-analysis-with-r-second-edition2/relative-strength-index-rsi.html>.