

CMPE 493 Semester Project

DotBox (May 2023)

*Joshua Ferrer, Alan Wong, Benjamin Chen,*

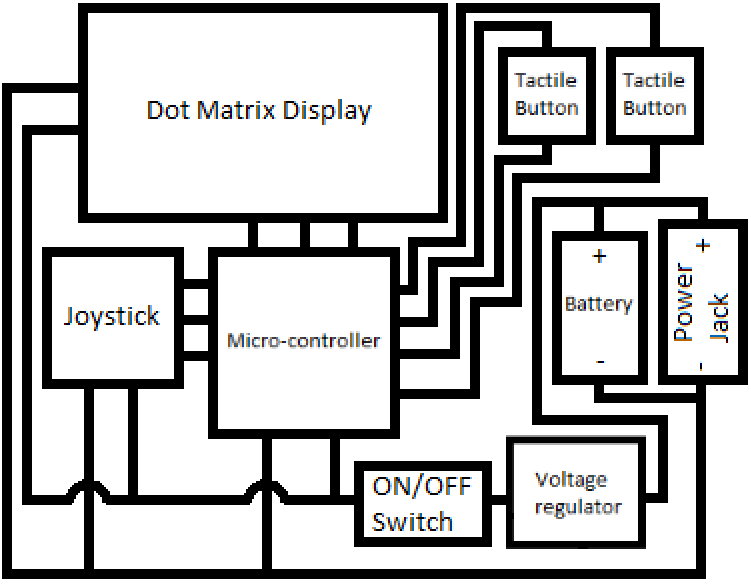
*Joshua Deanon, Anthony McDougle*

***Abstract* - This is a journal that will describe the functionalities and components of the DotBox, a handheld device for playing games.**

1. INTRODUCTION/Project Description

DotBox is a handheld gaming console that has a joystick and two buttons that interact with the menu and the games. The menu allows users to choose one of the two games to play.

The first game is called the Maze Game. In this game, the user has to find their way to the exit while facing an obstacle of not knowing where the walls are at first. The walls would appear once the user is in the area. From there, the user must find their way to the exit. The next game is called Arrow Crossing, where players are first given an amount of three for their health and zero points. The player is able to move a dot on the display. The main objective is to reach ten points by moving the dot to the same location as that of the dots that appear at a random location on the display. Only one of these dots appears on the display at a time and, if collected, will either increment health or points. Arrows will also cross the screen that, if they hit you, will decrease the player’s health by one. If the player has less than five health and they collect a dot, this will increment the player’s health by one, not their points. Once the player reaches five health and they collect a dot, it will now increment their points by one, rather than their health. If the player gets to zero health, they lose the game. If they reach 10 points, they win.



*Block Diagram*

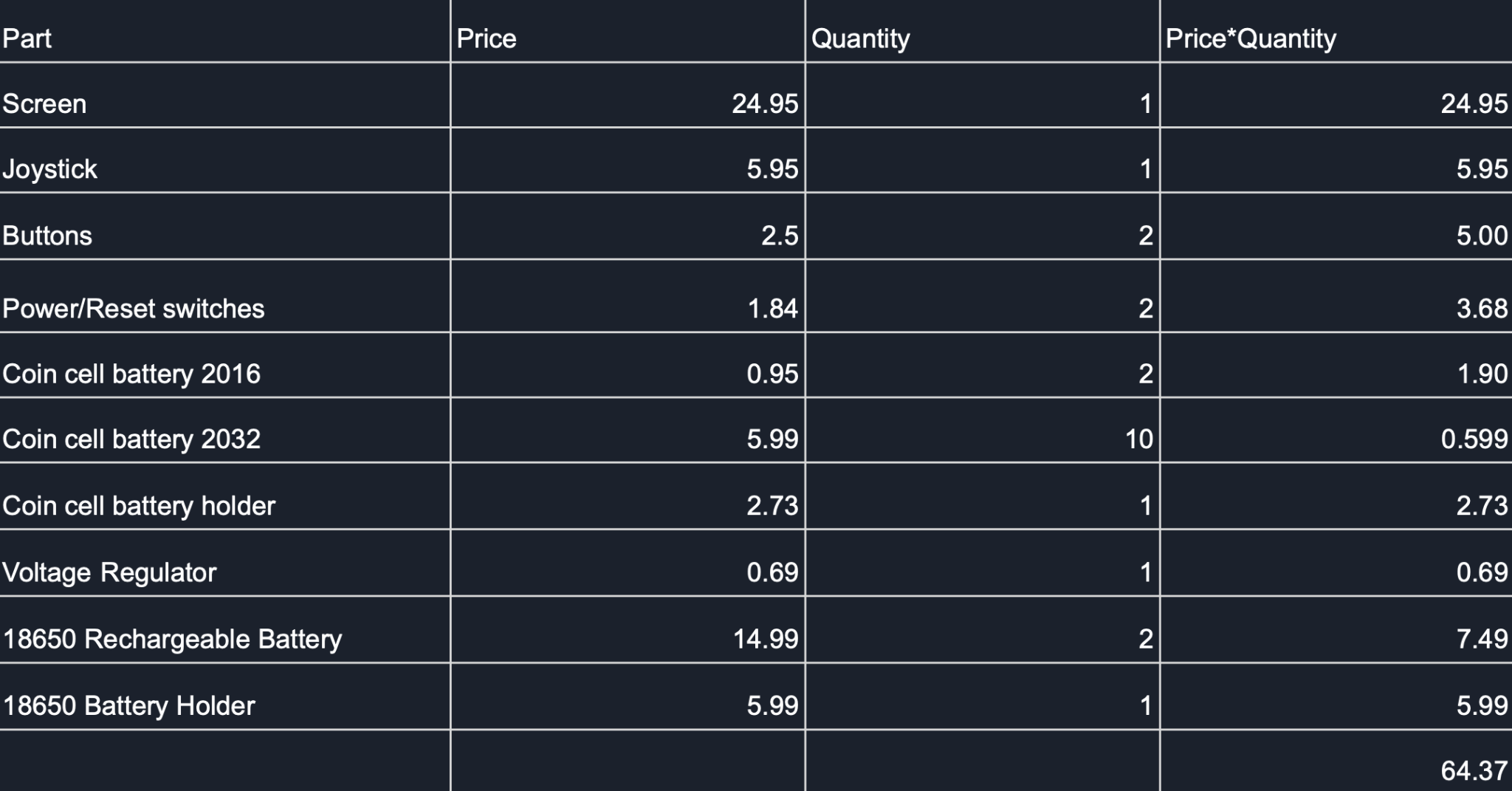
1. THEORY OF OPERATION
2. *Description of block functionality*

In the block diagram, power is supplied from the both power jack and battery. With the option to switch between the two power sources and the option to complete power off the device with two separate switches. The joystick will act as user input to switch between games in a menu which will display the two options of a game between the maze and the arrow crossing. One of the buttons will act as input to select which game to play in the menu and the other button will allow for the user to exit the games and return to the menu. Once in a game the screen will display a dot that represents the player and other objects according to which game the user chose to select and the joystick will allow the user to move the dot up to the edges of the screen.

1. *Constraints on design parameters*

One of the constraints in this project was the limited amount of time to have a working system. Also, having chosen a dot matrix display with only 16 by 24 “pixels”, the games and texts outputted to the screen had to be built and chosen taking this into account.

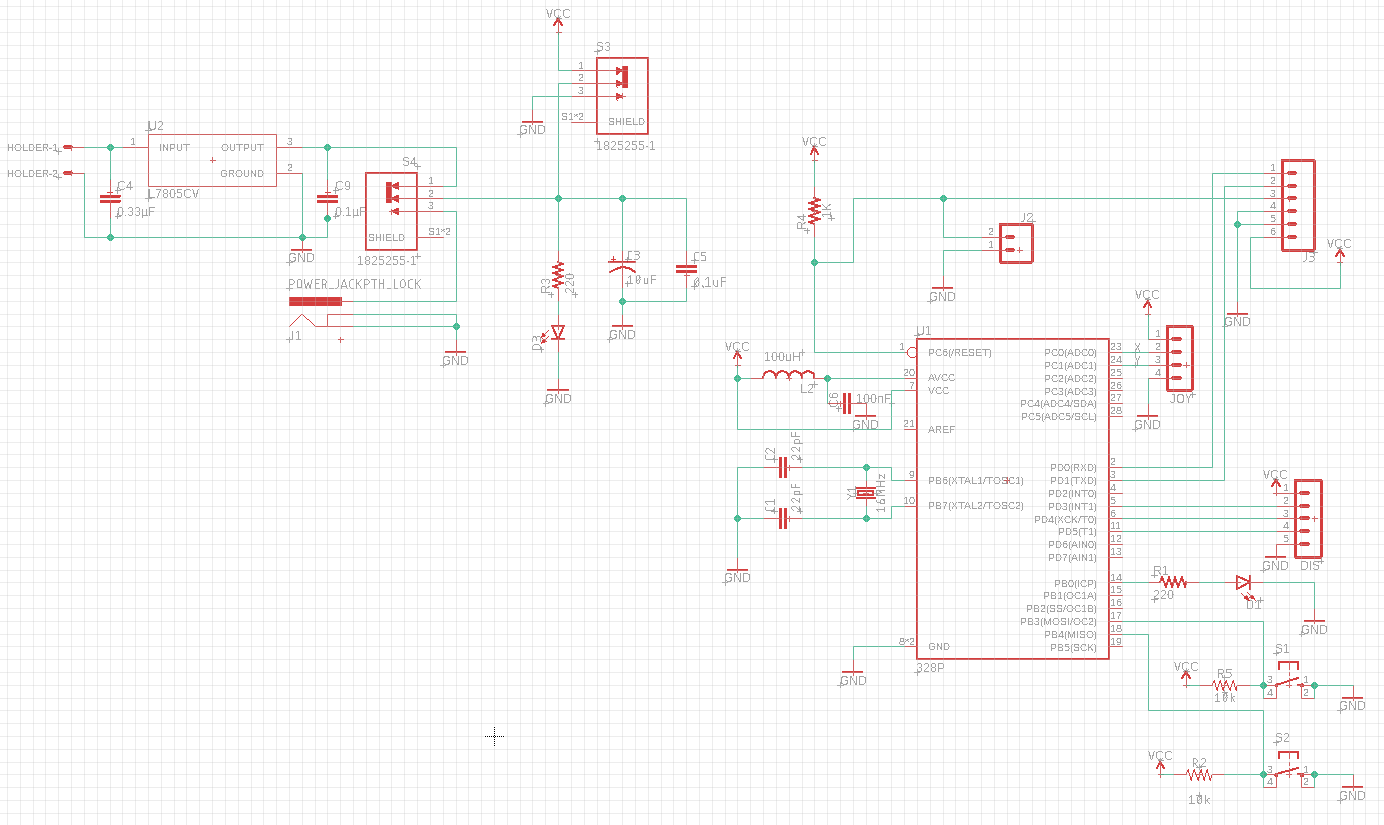
1. Implementation
2. *Bill of Materials*

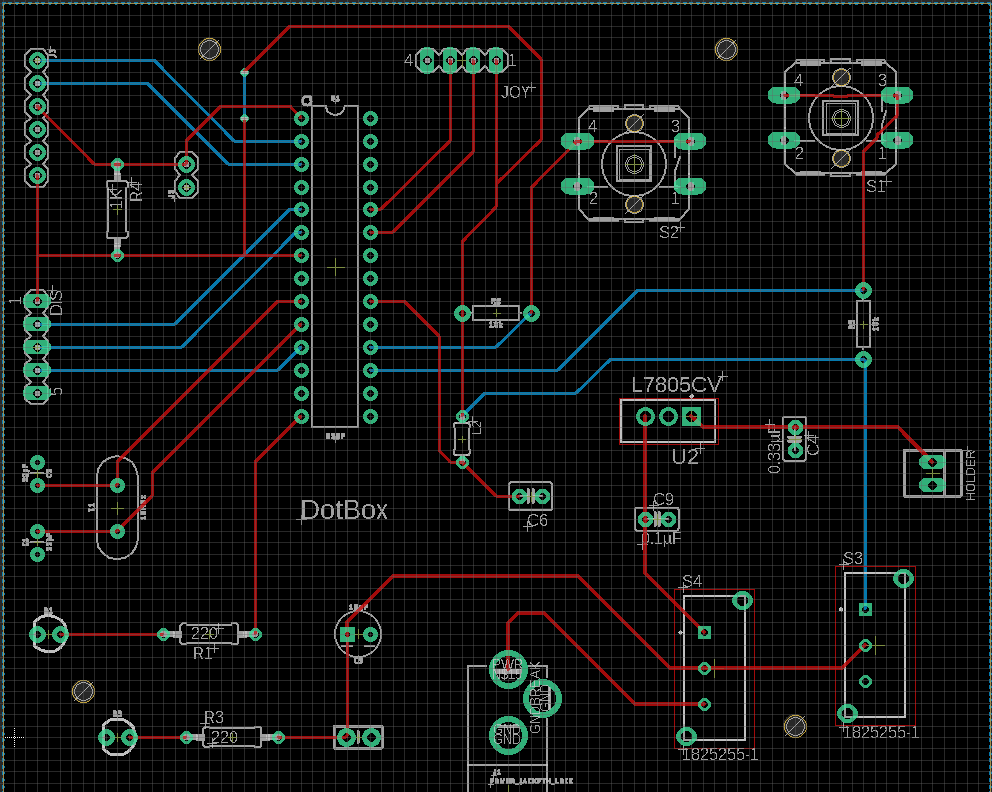


1. *Software tools needed for implementation*

The software tools that we used and implemented for our project was Eagle AutoDesk to develop and design our PCB. For coding the microcontroller we used Arduino IDE, where we coded in C++. The 16x24 screen chosen was powered by a HT1632C Driver, which came with the Adafruit\_HT1632LEDMatrix library that allowed for the easy display of objects and text on the screen. The code developed started from both code taken from Dr. Sumarsono and code provided online that was meant to help one know how to use the dot matrix library.

1. *Schematic images*

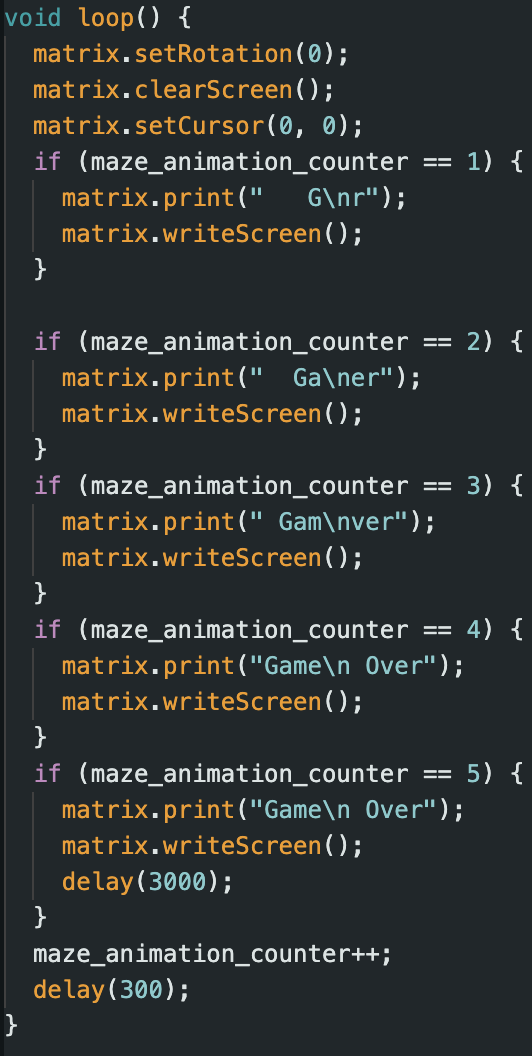
**

1. *Board layout*
2. EXPERIMENTAL RESULTS

Our group experimented with adding a battery pack to make the gaming console a true handheld console but we ran into some errors with being able to connect it to our board. One of the problems was placement inside our case in which we didn't plan out how we wanted the final problem to look so we ran into placement problems for the battery pack. As we hooked up the battery connecting to the board we realized the wires between the battery holder and the board could not stay, which would be a huge problem because if someone was to hold the game in hand and move around it would potentially disconnect. So at this point we realized that having a plug connected would be optimal for the functionality of the game.

We also experimented with the library Adafruit\_HT1632 and added different animations to create the title DotBox, You Win!, Game Over, Menu. We found a way to use the matrix functionality with 1s and 0s to display these animations. In the middle of doing that we ended up finding something that is more efficient and easier to make a cool animation. We created a looping algorithm that prints out the letters and slides the animations across the

screen.





1. CONCLUSION & FUTURE WORK

In conclusion, the creation of DotBox was inspired by the infamous Gameboy console as it sparked us to create something just as creative and fun. To add on, it was a great learning experience for our group as it taught us many valuable lessons for the future. For example, it is important to picture the final result when working on a design. To add on, as engineers we have to consider what the project would be like from the consumer’s perspective. If we were to revisit the current design, we would modify the placement of the power switches and what types of buttons we would use for the console. Potential 3D printed casing while making the console easier to hold for both adults and youth, and figuring out how to implement the battery in our console would also be our next steps to further advance our project. DotBox was still a successful project, as we finished the project as more experienced computer engineers.

References:

1. Ada, L. (n.d). 16x24 led Matrix. Adafruit Learning System. [Online]. <https://learn.adafruit.com/16x24-led-matrix>
2. adafruit (n.d). HT1632 [Online]. https://github.com/adafruit/HT1632
3. adafruit (n.d). matrixdemo.ino [Online]. https://github.com/adafruit/HT1632/blob/master/examples/matrixdemo/matrixdemo.ino
4. S. Mudgal. (2012, September).(No Title).[Online]. https://stackoverflow.com/questions/12657962/how-do-i-generate-a-random-number-between-two-variables-that-i-have-stored

