# ***Table of Contents:***

[***Components*** 2](#_Toc25745334)

[***Report Body*** 4](#_Toc25745335)

[**Algorithms:** 5](#_Toc25745336)

[**Coding:** 6](#_Toc25745337)

[***Coding Explaining and Review:*** 19](#_Toc25745338)

[***References:*** 20](#_Toc25745339)

[***Conclusions and Recommendation:*** 20](#_Toc25745340)

# ***Introduction:***

Hello, fellow reader. Ever want something to lock your variables with a lock? Using a simple pad just like your smart phone you millennial. Of course you do. Cause I am forcing this upon you. As a desperate salesman. My code lock has it all. A servo motor to open and close your safe. A screen to display your inputs in UWU. I got you! And yes you ask for it. Want a piezo buzzer to annoyed the intruder for trying to get in your safe. We got you.

Hello, fellow reader. This code lock I’ve create has a four components. Piezo Buzzer, Key Pad, Servo Motor, and LCD Screen. The key pad used for all the input for this project. While the output depending on the input you put in the Arduino. If you type the right code, you can unlock or change the password. If incorrect, you could trigger the alarm. The LCD screen will display your results letting the user know where he or she is.

# Image result for keypad arduino***Components***

1 key pad:



1 servo motor:



1 Piezo Buzzer:



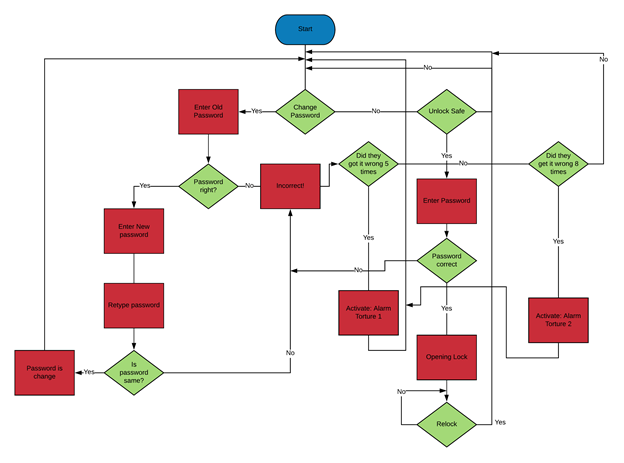
1 LCD screen:

Arduino Freenove Uno R3:



# ***Report Body***

# **Algorithms:**

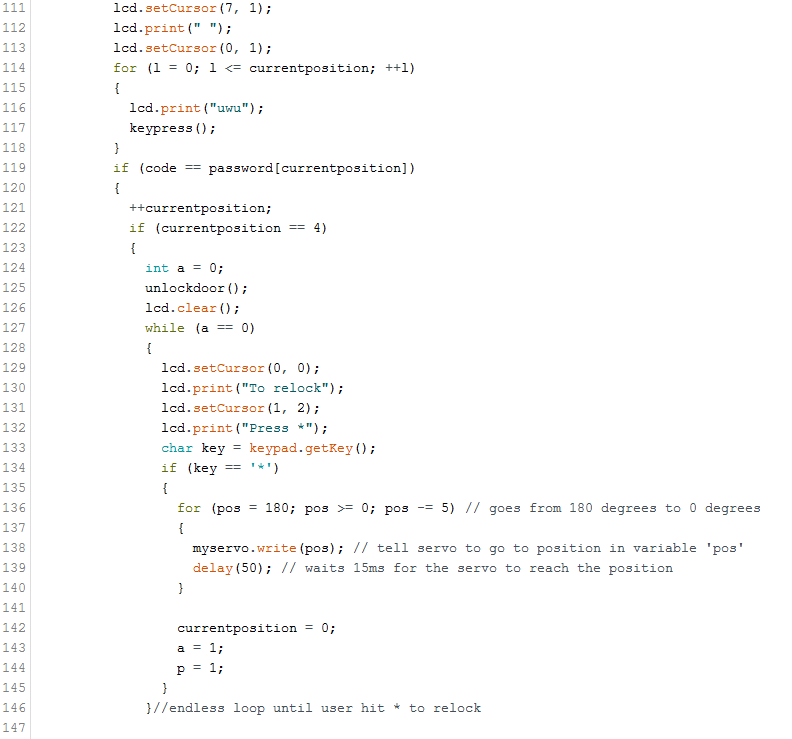


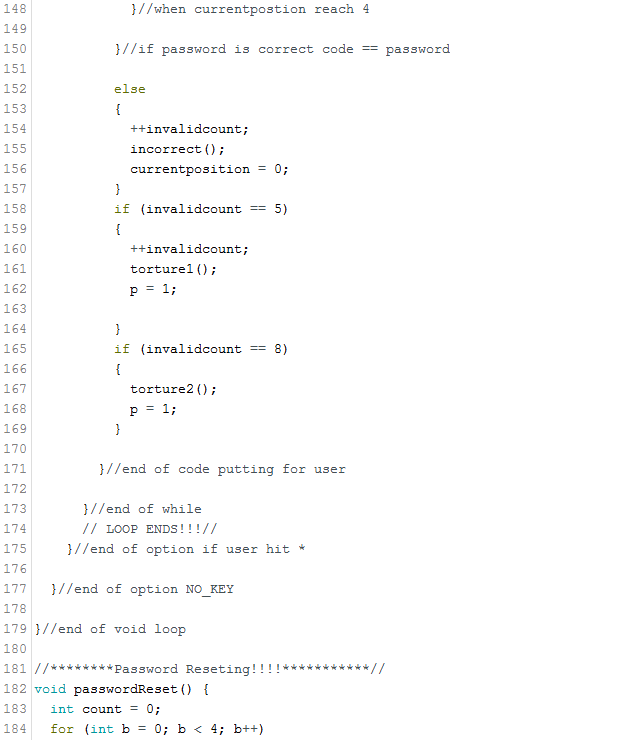
# **Coding:**



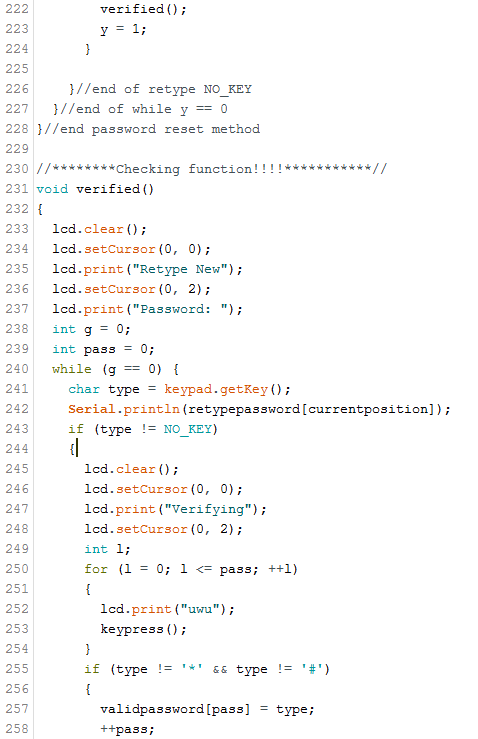






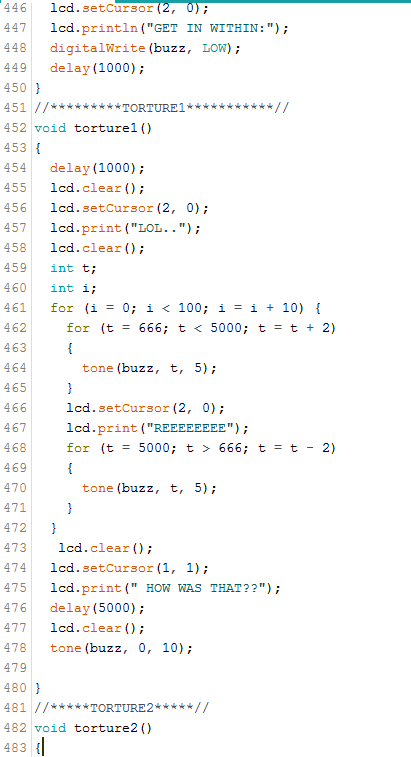
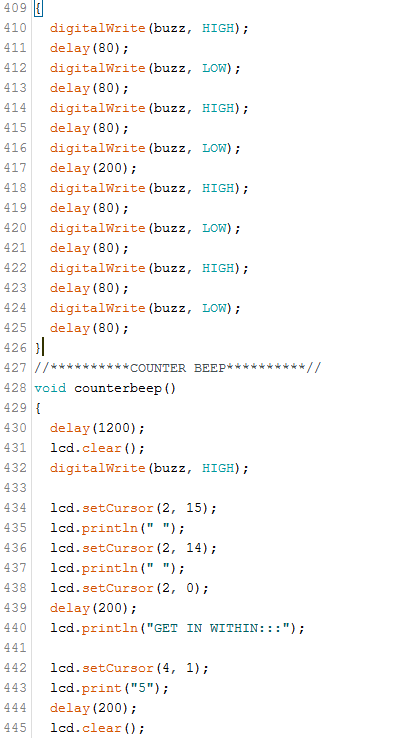














# ***Coding Explaining and Review:***

At the beginning of my code it includes some library requiring me to full implement these components. The LCD or as it’s known as Liquid Crystal Display. Keypad 4 by 4 and Servo motor. I will be going through parts of my code and used the number on the left side if you get lost. At 5-6 I’ve declare and set up my Servo motor variable. Then through lines 8-25 is the set up for my keypad. You can see the keypad requires 8 pins to set up. Each for a column and row. I’ve used Arduino pin 8 through 1. We have int currentposition which is the screen serves a purpose for position of 4 number you input. It will display UwU for every number you put into when inputting your password. The buzz is just a set up for my alarm and int invalidcount is for the alarm system. In void setup, line 31 we can see that I’ve setting up my LCD, piezo buzzer at 9, and servo motor at 10. LCD before set was using pins A1 – A5 on the Arduino.

I created many char array for the password. The char password is the default and can be change. The save password if for when changing the password, the original password you had never get lost. For example, retype and valid password purpose is for the user to change their password. If they don’t match to the old one. Then the save password or old password is not change.

The option can be seen in my code at 56 when the user use # to reset their password and cause the program go through the first check. See if you can type the old password. If it’s invalid, then invalid count goes up. This might cause the alarm to go off; if you’re not careful. Then you’ll be back to the start where it asks you to reset or enter password. Let’s say you got password right. Then the Arduino will go to the method called passwordReset(); This method is located at 182. Where the old password will be save if you failed to valid your new password. Then the method will use your old password as the main. If you pass the valid your new password, then the code will rewrite the char password. The verifying process is located at line 231. You can see the valid and not valid statement. Valid statement is at 263 and not valid is 290.

You can notice the while loop I’ve made in my code. This was due to the fact of having troubling error. With the void loop. So that the statement the user input wouldn’t be lost when they type their password. Programmer note –***Improvements***: I should have made more method when the user type int the input. Maybe then I wouldn’t have to used loop to lock the loop in a certain if statement. Next time used method called UserUnlocking and UserReset. When they put in # hashtag or \* asterisk key.

The program has many other methods like open door function at line 319 which moves the servo motor from 0 to 180. When incorrect by a single number then the buzzer will scream by using a method called incorrect() at line 337. Clearscreen() locate at 363 is used for clearing the key you put in. But I don’t used that method anymore due to the fact I’ve used UwU instead of asterisk. Keypress() at line 375 purpose is used for making noise when you hit a key on the keypad. Displayscreen() at line 382 is to let the user know to input their code when unlocking their door.

Let me go over function I don’t used in my code you know my clearscreen() is one and armservo(). They were used in the old code I put in reference. But I decided to improve on the code and rewrite as my own. Counterbeep() at line 428 and torture2() at line 482 are used for last attempt the user types the wrong password 8 times. Torture 1 is my favorite type of coding. I used an old project from making a smart bag. Making the alarm frequency going up and down. By changing the power in tone you have the pin, frequency, and duration in that order.

# ***References:***

<https://gist.github.com/gannybuzz/02bf3ad4c3db1268b29c>

<https://create.arduino.cc/projecthub/SurtrTech/keypad-door-lock-with-changeable-code-468b15>

# ***Conclusions and Recommendation:***

This was a frustrating new type of library that I’ve never mess with it was a challenge researching on this subject. With a lot of trial and error, I’ve gotten the code to work and I was glad to make this code lock. Gave me a challenge in my programming skill and problem solving. To find the solution to my final project and see the final product. I’ve still hate it. . . Just kidding I still see so many improvements to make on this projects. This code is already used in everyday life. With safe and vaults but more advance and high tech. With powerful gear and motors.

For my coding I would recommend removing the methods I didn’t used. Clean up my code by creating separates method instead having method for if user type # hashtag or \*asterisk key. Making it easier to find and rewrite for others. Could also fix the while loop I’ve implemented into the method. The alarm system could be fixed and improved on with security. Need to improve on the Unlocking part of my if statement. I’ve notice that my buzzer interrupts you if any the 4 numbers in the array are not correct. Example like if you type the first input is correct then second input is wrong. Then my code lock screams at you for typing the wrong password. This is not practical due to the fact that a robber can easily guess your password.

Programmer notes ***Improvements physical:***

* Having a tilt or accelerometer to detect if the person is trying to mess with the box or safe. For any extreme movement when it’s power one. Letting the user know someone is trying to break into the vault.
* We could a RFID reader to add another layer of security. For the program to allow you to unlock or even type code. The card need to be read or place down while accessing the vault.
* Used a speaker to input funny noise like someone reeing as an alarm sound
* Having a motor to make the box shake in angry when someone type the wrong code and turn itself off.