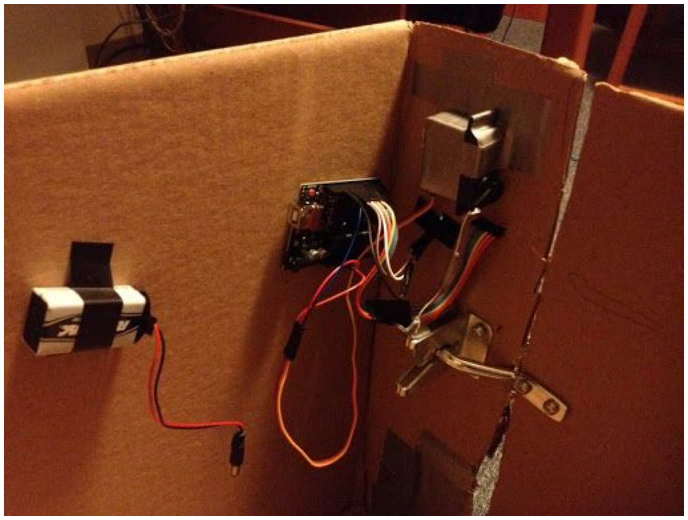
David Weber, Lane Barnes
Lynn Schmidt (Mentor)

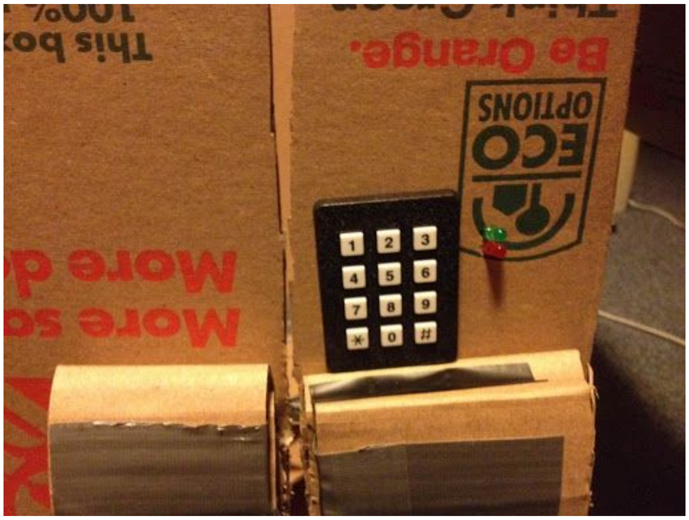
Keypad Lock

We have built a lock that uses a keypad and passcode to open and close a door. We do not know much about circuits and programming, but were willing to try and learn to make something cool. We will be using Arduino Uno to program the functionality of the lock, as making our own circuit boards is out of our knowledge. We needed to design a 0-9 keypad that sends signals to a device that stores and reads passwords, and sends a signal to a servo to open and close the lock. The exact kind of door the lock will be used for is unknown, but perhaps we can make it so you can put it on many kinds of doors, such as normal doors, cabinets, drawers, refrigerator doors, etc. Making it hold onto a normal door will be our plan as of now. We are using a 9V battery to power the arduino on its own, but it can only last a day or two before dying. The servo sucks up our battery life, and we need to look into a better power solution. As of right now, the whole arduino, latch, and wiring will be exposed, but only on the locked side, while the unlocked side only reveals keypad and LEDs. For demo purposes, we are using a big cardboard box with a side cut into a door and a surrounding frame to mount our gate latch. Time permitting, we plan on adding more features, such as a LED display/speaker (for button press confirmation and right/wrong password notification), multiple password saves and management, or mounting the lock to a real door.

Inside



Outside



Background

Our inspiration for choosing this project was an idea for a “secret knock” lock (Of course, after researching online, we discovered that it has been done many times). It would record and detect a specific rhythm of knock of your choosing, and would open the door if the secret knock was used. We found this to be too ambitious, as we started our OOP late in the semester, so we decided to create a simpler keypad lock.

Our lock is ran using an Arduino Uno that can be programed. Connected to the Arduino is a keypad, a servo, one red and green LED lights, and 9v battery (See Basic Design Schematic). The keypad has 7 pins that selects a key by the combination of the column (out of 3) and the row (out of 4) in a 2D array (see 3x4 Keypad PIN Layout). The servo rotates 90° to pull on a string which pulls open a gate latch (see Servo Gate Latch Mechanism). The green LED blinks once when a button is pressed and three times when a password is correct or saved successfully. The red LED blinks three times whenever a password is entered incorrectly, time runs out after a pressed key, or when a password larger than the set password maximum is attempted. To wrap all the components together, our program allows the keypad to communicate with the arduino to signal LED blinks, open the lock, and save new passwords (see Pseudo Code).

We have tested for many different scenarios and are confident in its functionality. There are some final things we need to finish, such as 3D print casings for our components, wire management, and mounting our lock to an actual wooden door and frame. We can attempt to add a speaker or an LED 7-digit display for improved user interface. We need to search for better power saving solutions, as out 9v battery cannot power the lock for more than a day or two. We are happy with what we have accomplished so far, and feel that we have succeeded in our original goal.

What We Learned

During this project we learned some of the basics of wiring a simple circuit. Such as, how a LED light needs to receive current from the 5 volt pin and be connected (in series) to a resistor that runs to ground. Additionally we had to frequently visit the sites such as Stack Overflow in order to gain an understanding of how C++ works. We had to exercise much problem solving and troubleshooting between the programming of the Arduino and wiring the keypad, LED’s, and servo. We also had to learn how to work in a partnership by dividing the work between us fairly and to suit our strengths and weakness and to fulfill the deadlines given to each other.

Pseudo Code

Saved password (array) initialized to empty password and must be filled by user

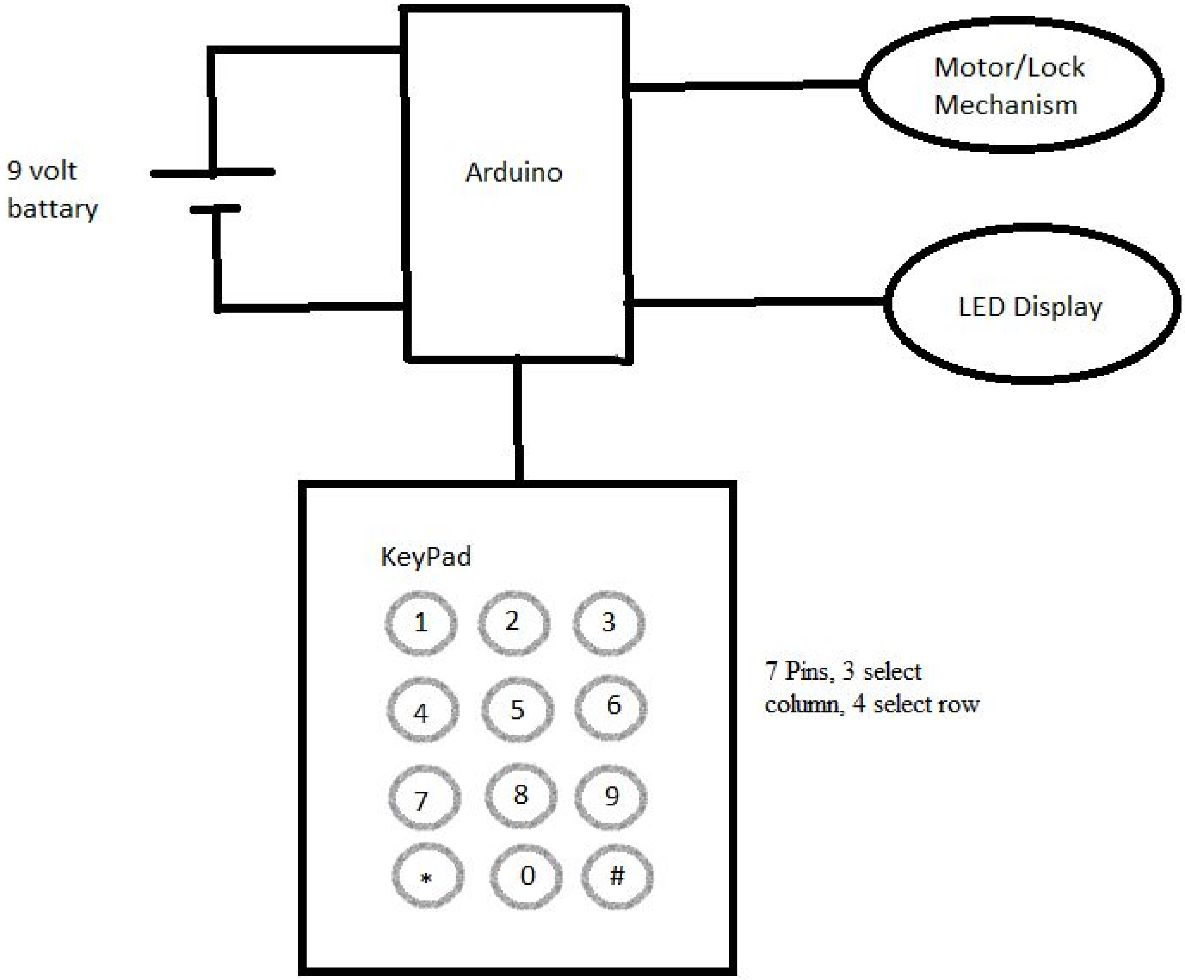
```
EnterPassword (Loop)
    ● Standby for user input
    ● If following key presses are not entered within 5 seconds of eachother
        ○ Return to EnterPassword
    ● If more presses than defined password limit (8 digits)
        ○ Return to EnterPassword
    ● If following key presses are entered with '#'
        ○ If CheckPassword returns True
            ■ Go to OpenLock
        ○ If CheckPassword returns False
            ■ Return to EnterPassword
    ● If following key presses are entered with '*'
        ○ If CheckPassword returns True
            ■ Go to SavePassword
        ○ If CheckPassword returns False
            ■ Return to EnterPassword

CheckPassword
    ● Compares each element of entered password array to each of saved password array
        ○ If an element pair do not match
            ■ Return False
        ○ Otherwise return True

SavePassword (Loop)
    ● If following key presses are not entered within 5 seconds of eachother
        ○ Return to EnterPassword
    ● If '*' is pressed
        ○ Go back and retry SavePassword (in case of accidental press)
    ● If following key presses are entered with '#'
        ○ Save as the new password and return to EnterPassword

OpenLock
    ● Open lock, wait 5 seconds, close lock
    ● Go to EnterPassword
```

Basic Design Schematic

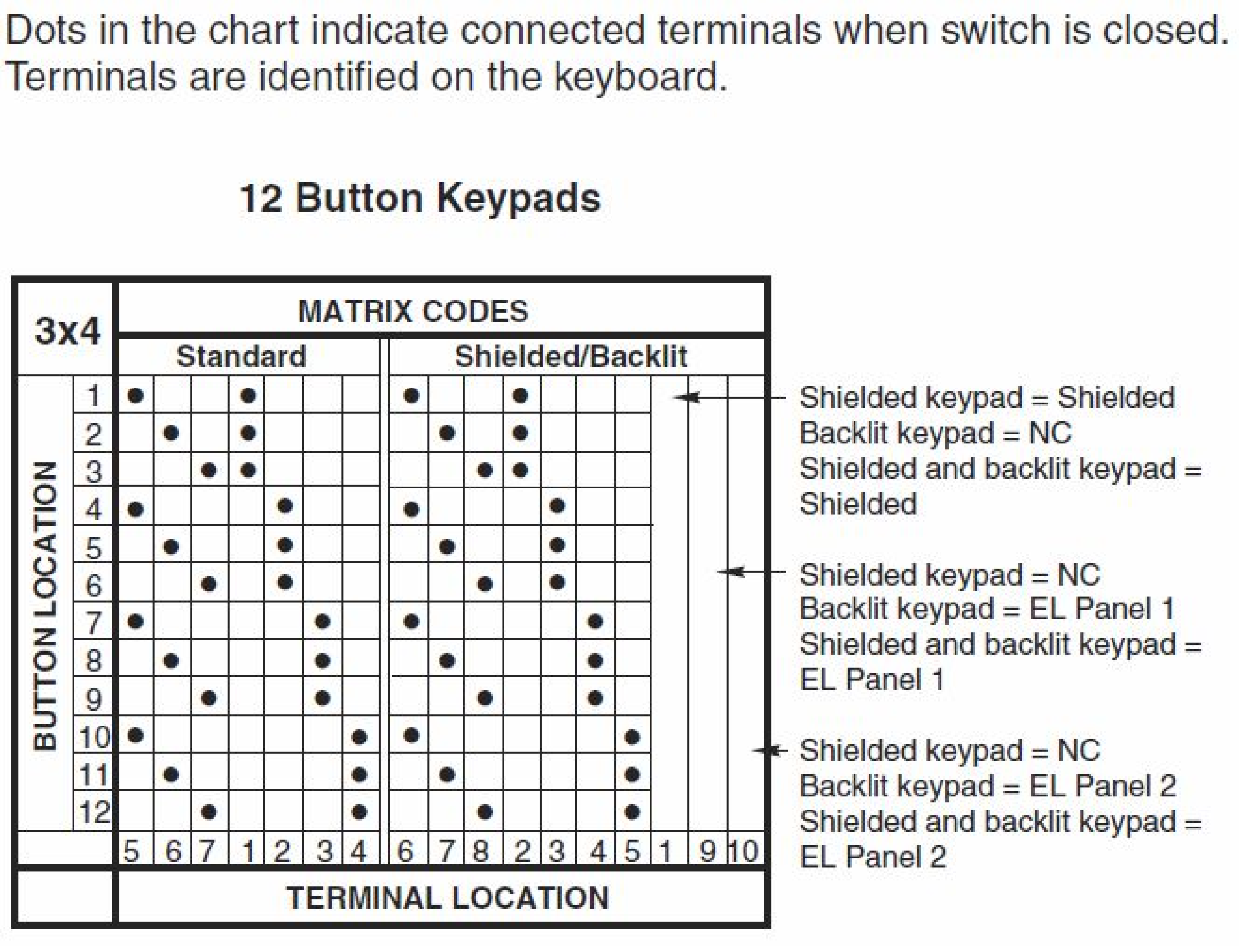


We will be using a servo and a gate latch for our lock mechanism.

There will be a green and red LED that indicate wrong or right passwords

7 Pins, 3 select column, 4 select row

3x4 Keypad PIN Layout



Dots in the chart indicate connected terminals when switch is closed. Terminals are identified on the keyboard.

12 Button Keypads

3x4		MATRIX CODES																
		Standard							Shielded/Backlit									
BUTTON LOCATION	1	●			●				●			●						
	2		●		●					●		●						
	3			●	●						●	●						
	4	●				●			●				●					
	5		●			●				●			●					
	6			●		●					●		●					
	7	●					●		●					●				
	8		●				●			●				●				
	9			●			●				●			●				
	10	●						●	●						●			
	11		●					●		●					●			
	12			●				●			●				●			
		5	6	7	1	2	3	4	6	7	8	2	3	4	5	1	9	10
		TERMINAL LOCATION																

Shielded keypad = Shielded
Backlit keypad = NC
Shielded and backlit keypad = Shielded

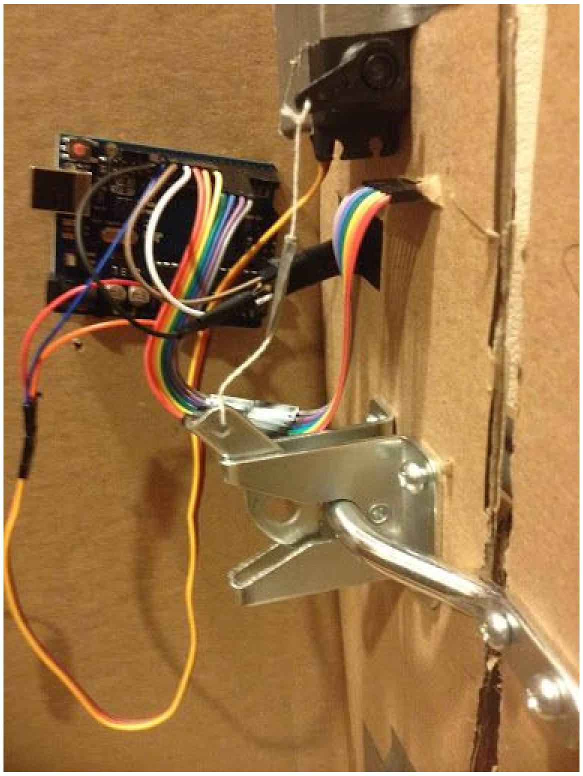
Shielded keypad = NC
Backlit keypad = EL Panel 1
Shielded and backlit keypad = EL Panel 1

Shielded keypad = NC
Backlit keypad = EL Panel 2
Shielded and backlit keypad = EL Panel 2

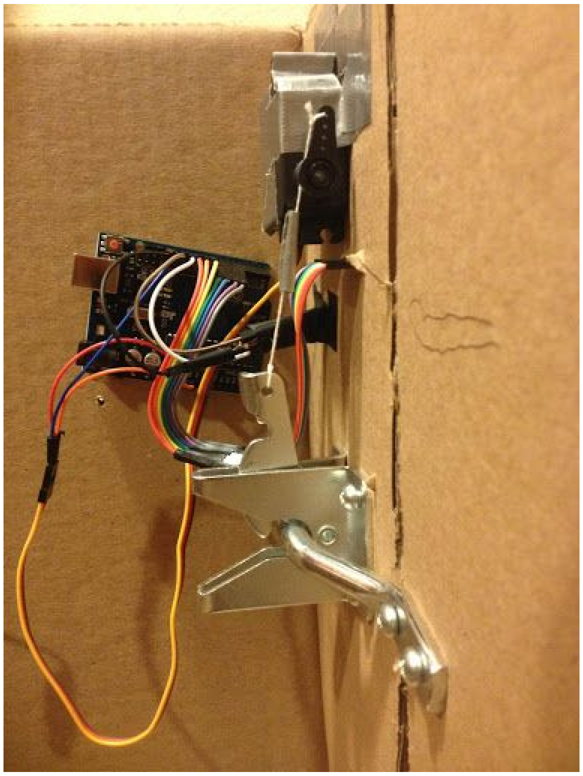
Image taken from http://www.mouser.com/ who sells the 12 button keypad we are using

Servo Gate Latch Mechanism

Servo at 0°, Latch is closed



Servo rotates 90°, Opens Latch



Appendix A

- ECE 103~In this course we learned the basics behind how LED's function with respect to the direction of current and how resistors operate in a system. (*This is the only engineering course David has taken*).
- CS160 taught us the basics of programing in Java. From there, we had to transfer what we knew into C++ syntax to program our Arduino using the websites such as Stack Overflow.
- We used two Arduino libraries, KeyPad (by Mark Stanley and Alexander Brevig) and Servo (by Michael Margolis), off of the Arduino website complete this project
- We have received help from our mentor, Lynn Schmidt, he helped us by sharing the keypad library, giving us the idea to use the servo, and wiring the keypad to the correct pins of the Arduino.