

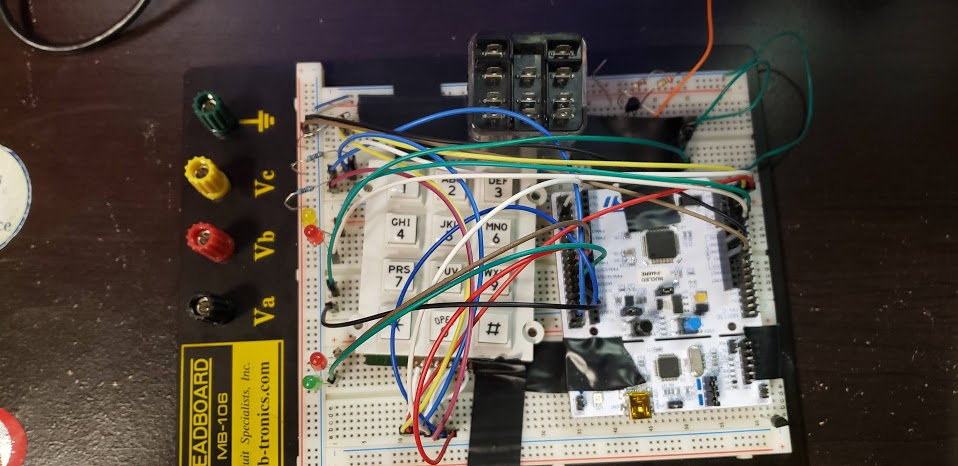
Micro-Controlled Button Lock

Kunal Mukherjee



February 19, 2019

University of evansville



Contents

[Problem Design 2](#_Toc1496455)

[Design 2](#_Toc1496456)

[Theoretical Results 4](#_Toc1496457)

[Simulation Results 5](#_Toc1496458)

[Experimental Design 5](#_Toc1496459)

[Measure Results 5](#_Toc1496460)

[Conslusion 5](#_Toc1496461)

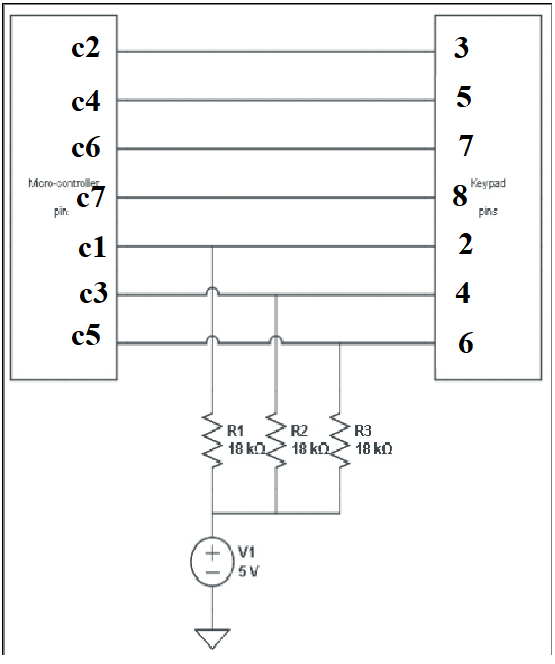
# Problem Design

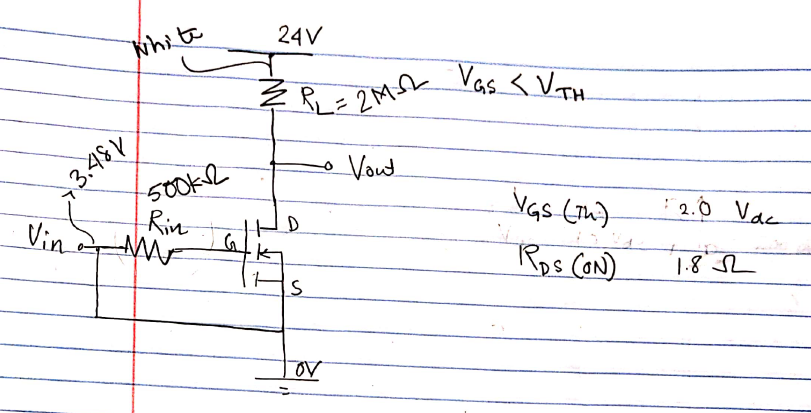
The problem we are trying to solve to is how to build a secure door lock using a microcontroller. The door lock will be simulated by solenoid. Once a correct answer is given, then a green led is supposed to blink 5 times and unlock the door for 10 seconds. If a wrong answer is given, then a red led should blink 5 times. Once a correct or wrong answer is inputted and the appropriate answer is taken the microcontroller should wait for the next set of code input. The design will be done using a keypad from the stockroom, a solenoid, red and green led, and BS170 with resistors. The Nucleo STM32F446 is the microcontroller that will be used.

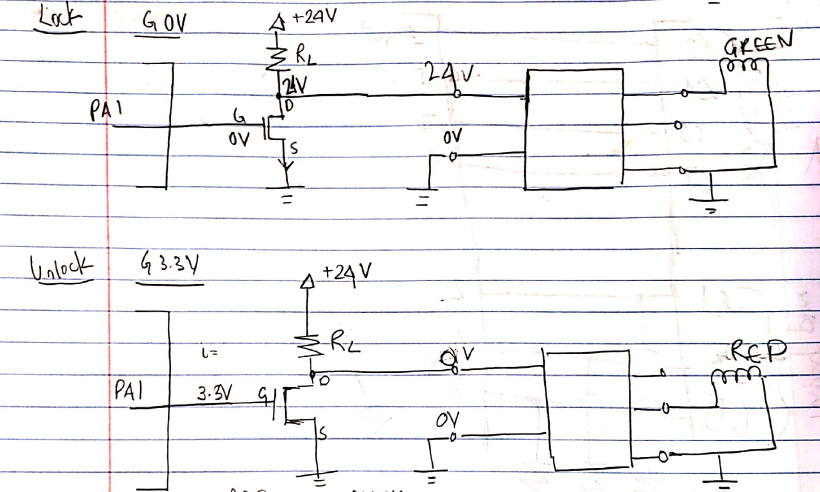
# Design

One of the major aspects of the design was how to get input from a Keypad. Therefore, I made a matrix of the pins that are connected to the buttons on the keypad. First the STM32F446 pins that are connected to the rows would be set to high output pins, and the column pins would be set as input pins. Then in order to detect if a pin is pushed the microcontroller would set one row output to be low and then try to detect if it receives a low input from the column of the keypad. Then, it would set that row back to high and set another row to low, that would happen until it detects the low signal and convert that signal into a key character. This design had lot of while statements, a better design was suggested by Mr. Randall to do the interrupt driven I/O, so that we are not polling the I/0 pins all the time, rather we are now interrupt driven.





The next major design hurdle was how to turn on and off a solenoid just using 3.3V from pin A5. For that I used BS170 N-channel mosfet as a switch. The VGS = 2.0V for BS170, so a 3.3V from a pin out can be used to turn/off the BS170 mosfet. Therefore, I connected the Vout from the mosfet to the positive end of the solenoid and connected the negative end of the solenoid to common ground. The drain end of the BS170 was connected to 24V, the turn off volts of the solenoid and the negative end was connected to common ground. Therefore, when PA5 was on, the solenoid would turn on and when the PA5 was off, the solenoid would turn off. 

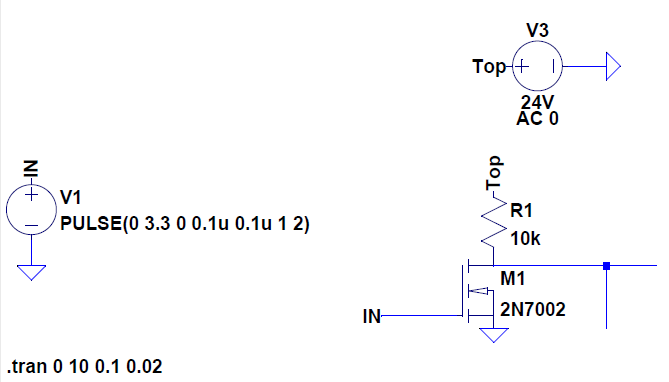


# Theoretical Results

The project should be working as per the following specifications mentioned above.

# Simulation Results

I used LTSPice to test the use of BS170. The simulation result matched with the BS170 device specification.



# Experimental Design

The locking mechanism will be tested by the use of helpful LEDs. Whenever a input will be entered for the code, a red LED will be lit up, a input for the unlocking code will lit up a yellow led, and depending weather the input code is correct or not, a green or red led will be blinked 5 times.

The working mechanism of BS170 was physically built at first, using a voltage sources for the gate and source voltage.

# Measure Results

The project did not need much measurement, except I used the DMM to measure the Vout of BS170, to see of Vout is going from 24V to 0V, as I need the voltage change for the solenoid to lock and unlock. I also, added an extra feature that will let the code to be reset back, if the user enters:000000.

# Conslusion

The project met the specifications of creating a micro-controlled button lock, using a BS170 mosfet, solenoid and STM32F446.

# Code

