

# **CSE321\_Project2**

## **Security System**

### **Table of Contents**

1. <b><u>Cover Page</u></b>	<b>Pg:2</b>
2. <b><u>Specifications</u></b>	<b>Pg:3</b>
a. <b><u>Inputs</u></b>	
b. <b><u>Outputs</u></b>	
c. <b><u>Functions</u></b>	
3. <b><u>Applications</u></b>	
<b>Pg:3</b>	
4. <b><u>Features</u></b>	<b>Pg:4</b>
5. <b><u>Block Diagram</u></b>	<b>Pg:5</b>
6. <b><u>Functionality Diagram(Flow Chart)</u></b>	<b>Pg:6</b>
7. <b><u>BOM</u></b>	<b>Pg:7</b>
8. <b><u>Schematic</u></b>	<b>Pg:8</b>
9. <b><u>Test Plan</u></b>	<b>Pg:9-10</b>

# **CSE Project 2**

# **Security System**

**By: Patryk Kasza**

**Person #: 50275611**

**Email: [pskasza@buffalo.edu](mailto:pskasza@buffalo.edu)**

## **Specifications**

### **1. Inputs:**

- a. 4x4 matrix keypad

### **2. Outputs:**

- a. LED flash
- b. LCD display

### **3. Functions:**

- a. ISR, buttonHandler
- b. inputHandler
- c. checkCodes
- d. \_checkCodes
- e. eraseCode
- f. appendCode
- g. resetPassword
- h. LCD library(1802.h/cpp)

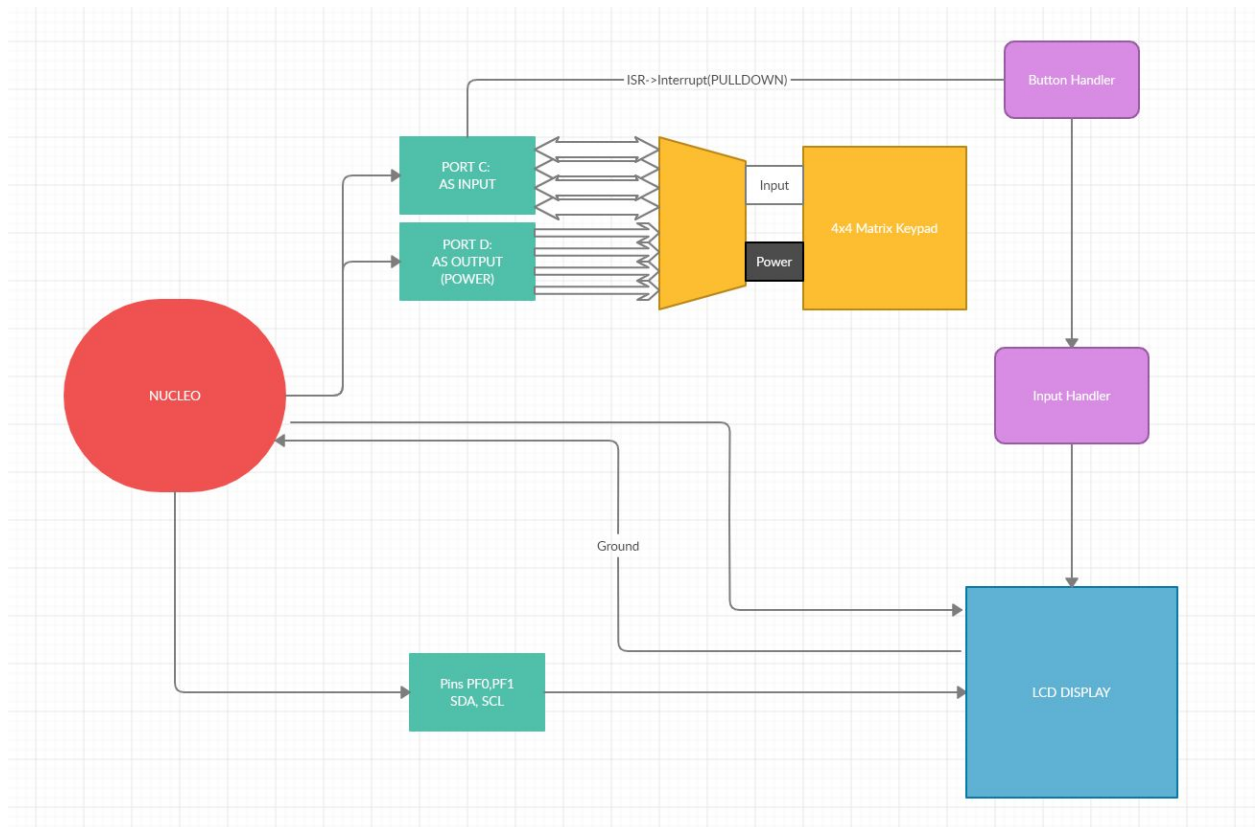
## **Applications**

- You can use this design as a basic security system. You can connect the software aspect to other software applications; using the mode(LOCKED, UNLOCKED, RESET) to determine allowed behavior within that software application.
- Connect it physically to a switch of sorts that will lock or unlock a door(or anything really)

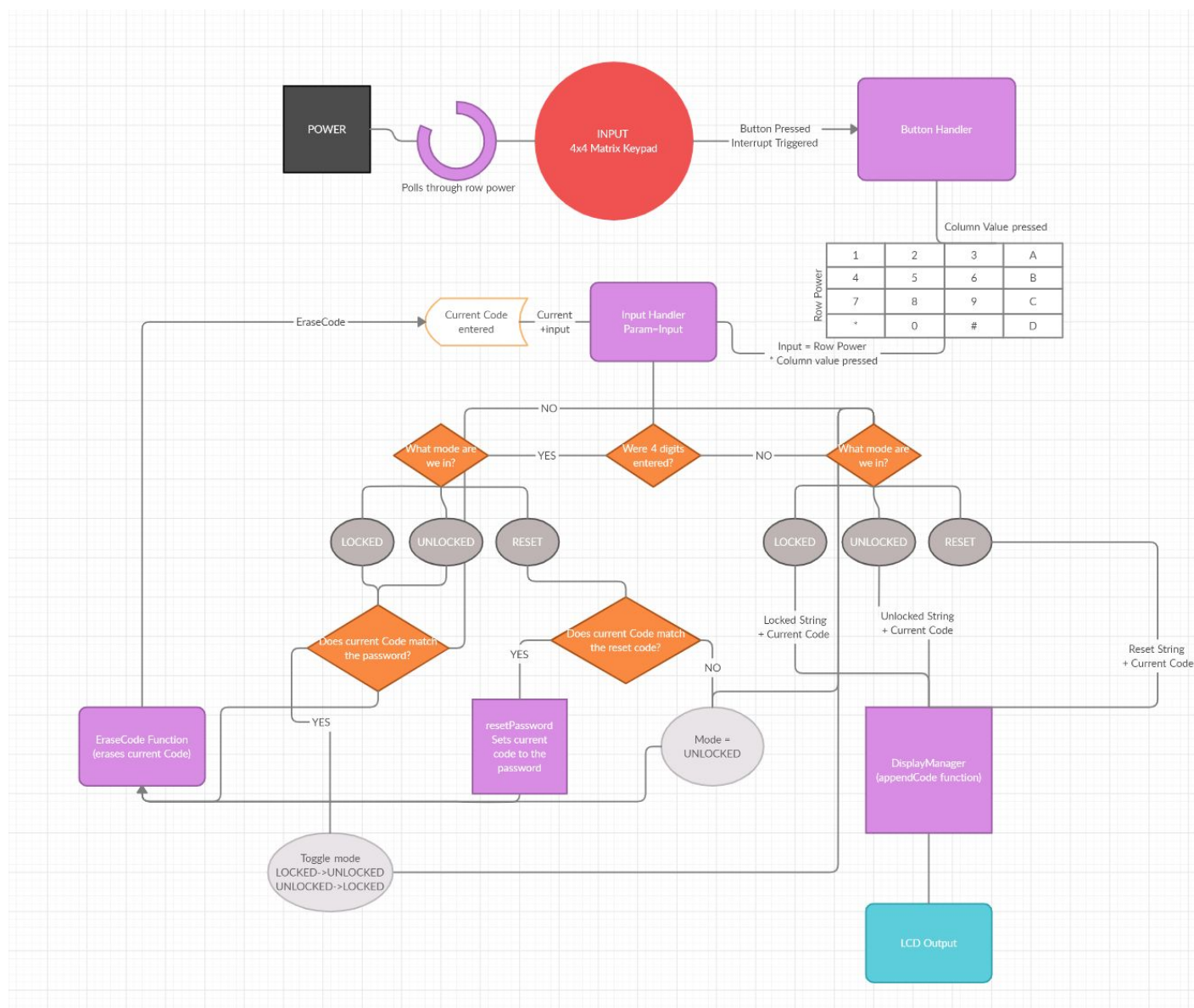
## **Functionality**

- LOCK, UNLOCKED, RESET modes
- Initial state is locked
- Initial password is 1111
- Password to reset password \*\*\*\*
  - Only works in UNLOCKED mode
- Screen when LOCKED-> Locked:\_\_\_\_ ( \_\_\_\_ = input)
- Screen when UNLOCKED-> Success!\_\_\_\_ ( \_\_\_\_ = input)
- Screen when RESET-> New Code:\_\_\_\_ ( \_\_\_\_ = input)
- Screen when incorrect password in LOCKED-> WrongCode!Locked
- Screen when incorrect password in UNLOCKED-> WrongCode!
- Blue LED flash when button press
- Red LED flash when incorrect password entered
- Password to LOCK or UNLOCK **CANNOT** be \*\*\*\*
  - Will need to change it, won't be able to LOCK

## Block Diagram:



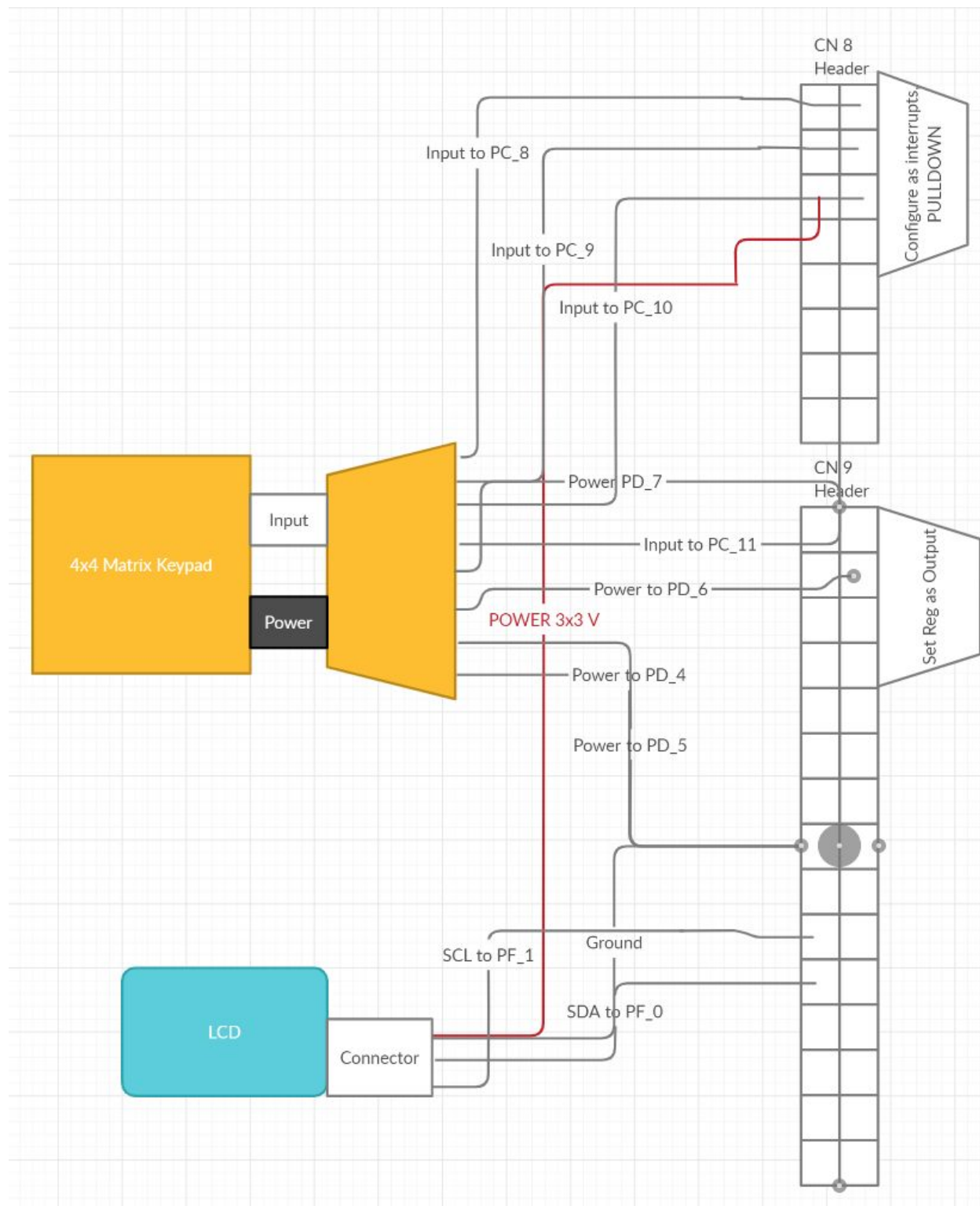
# Flow Chart



## **BOM**

1. 4x4 Matrix Keypad
2. Nucleo L4R5ZI
3. 1802 LCD Display
4. At Least 8 male to male jumper wires
5. 4 male to female jumper wires
6. Micro USB cable

## Schematic





## **Test Plan**

### Testing Matrix Keypad:

1. Set up so one row has power
  - a. Any input?
2. Associate correct IDR values with values on keypad
  - a. Correct values?
  - b. Any bounce?
3. Adjust bounce, one button press = one input registered
  - a. Set up printStatements in the ISR
    - i. See how many times something is printed on 1 button press
4. Include polling through rows
  - a. Print the ODR for PORT D
    - i. Any change?
  - b. Associate values for correct ODR value and IDR value
    - i. Correct values?

### Testing LCD:

1. Is the LCD displaying correctly?
  - a. lcd.print(hello), lcd displays hello?
2. Print a global string variable
  - a. Does lcd display correctly?
3. Update that global string variable with input
  - a. Does lcd display correctly clear and update?
4. Associate offsets with strings representing different modes/displays
  - a. Are the offsets correct?
  - b. Does it update with the input code continuously?
5. Does lcd correctly transition through modes?
  - a. Wrong password on locked, stay locked screen? Vice Versa
  - b. Correct password on locked, go to unlocked screen? Vice Versa
    - i. Does it register when the correct code is put in?
  - c. Does it go to reset mode only in unlocked
    - i. Wrong code displayed when reset code put in locked mode
    - ii. Prompt for new code displayed when reset code put in unlocked mode

- d. After resetting password is mode unlocked?
- 6. Functionality
  - a. Stress test it
  - b. Lock and unlock continuously
    - i. Change password
      - 1. Old password invalid?
      - 2. New password valid?
    - ii. Should continuously display correct screen for each transition mode