Smart Secure Doors System

CS122A: Fall 2017

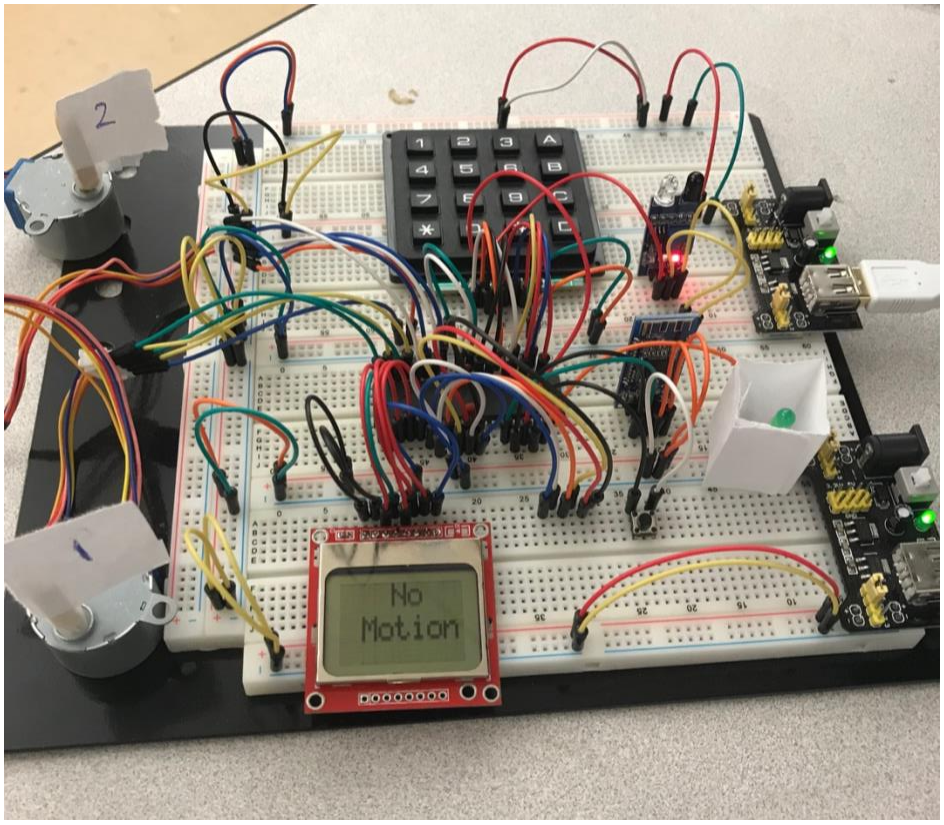
Srisri Gokanapudy

Table of Contents

Introduction

Security is a constantly growing field because as new methods of protection are devised, so are new methods of breaking the protection. I would like to contribute to the field of home security by building a Bluetooth operated door. Opening a door with an app on my phone is something I'd like to implement in my house one day.

The project is a door security system. When someone approaches the door, an IR sensor detects motion and asks for the passcode. Once the correct passcode is entered, the stepper motor rotates opening the door. The security system can be implemented on any door anywhere to monitor entry. Someone enters a passcode on their phone and via Bluetooth, the door will open.



Hardware

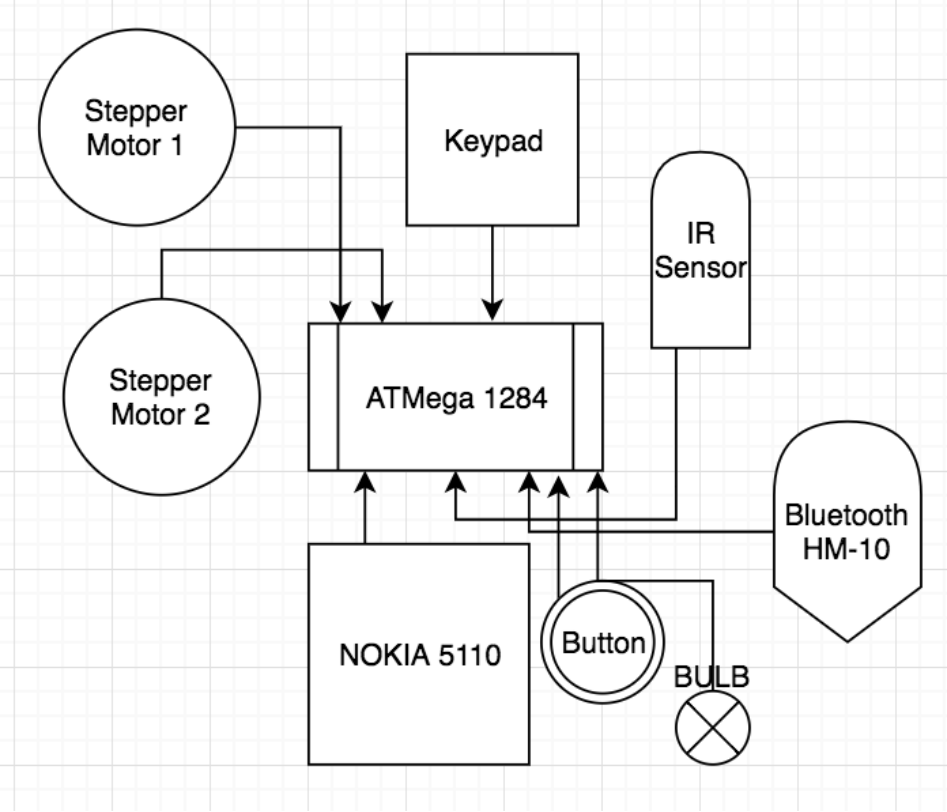
Parts List

The hardware that was **used** in this project is listed below. The equipment that was not taught in this course has been bolded.

- ATmega1284
- **IR Sensor**
- **Bluetooth HM-10 with "Serial" iPhone application**

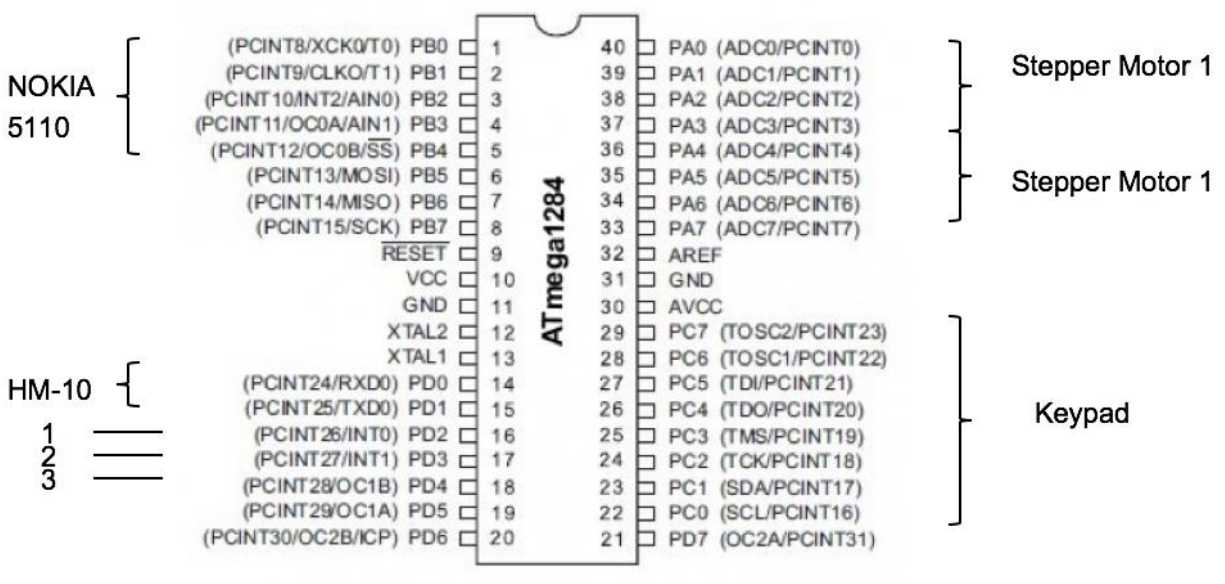
- **NOKIA 5110 LCD**
- Keypad
- Button
- 2 Stepper Motors
- Light bulb to represent light in a room

Block Diagram



(USART)

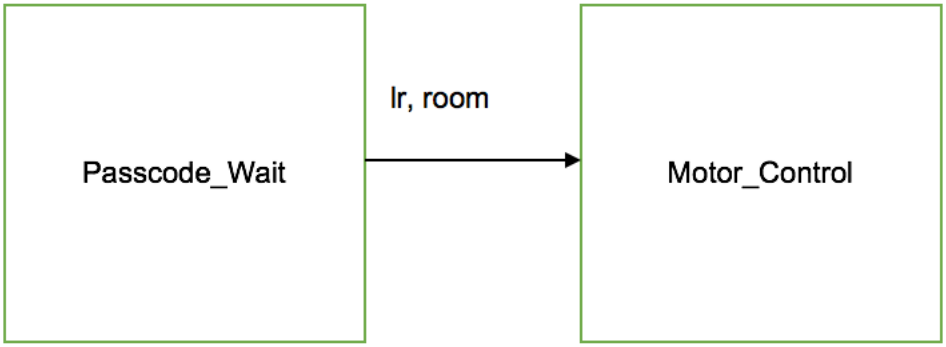
Pinout (For each microcontroller/processor)



1 – IR Sensor, 2 – Button, 3 – Light Bulb

Software

The software designed for this project was implemented using the PES standard. The overall design as a task diagram is included below.



Here is a short description of the tasks in the project. *Note: The appendix will include images of all SM's designed*.

Passcode_Wait: Upon motion detection, the menu state allows the user to either check the passcode or enter it. Upon correct entry (123) on the keypad or on the iPhone via Bluetooth, the lr variable is set to 1. This initiates the next SM.

Motor_Control: If the lr variable is set to 1 by the Passcode_Wait SM, the Motor_Control SM turns the motor to the right for 200 ms representing unlocking the door, then turns left for 200 ms representing locking the door. If the button is pressed, the 2nd motor is opened instead of the 1st.

Implementation Reflection

I enjoyed developing the Security Door Systems project because I learnt how to use new tools such as the stepper motors, IR sensors, NOKIA 5110 display, and the Bluetooth HM-10. I enjoyed learning about and implementing the Free RTOS scheduler compared to the task scheduler that I used in previous projects. I chose this scheduler because my state transitions were more input based.

If I were to redo this project, I would divide the project into more SM's so that the process would be smoother. Having the IR sensor and keypad check for input at a much slower period than the NOKIA displays required me to introduce a count variable to wait for the count. Since the period is smaller, this causes the SM to write to the NOKIA multiple times causing a waste in power resources.

Milestone

For my milestone I aimed to complete my 70-80 point project. A 70-80 point project correctly implements the stepper motors, keypad, NOKIA 5110 display, and IR Sensor. The system must read the IR sensor and ask for the passcode on the NOKIA 5110 display upon motion. Upon correct entry of a previously set passcode (123) on the keypad, the stepper motors, representing doors, must rotate allowing entry. Buttons are used to toggle the light bulb which represents light in a garage.

I completed the milestone by November 9th, the end of the 2nd week. I was able to complete it on time because I started putting in many hours to work on the project early. I was having trouble implementing the PIR sensor because it was either too sensitive too light or it would not pick up any motion at all. I rewrote my proposal, changing the PIR sensor to a regular IR sensor. The IR sensor was easy to implement and allowed me to complete the milestone on time.

Completed components

I completed all 3 levels of the project: 70-80, 80-90, 90-100. I changed my proposal to implement the IR sensor instead of the PIR sensor. This allowed me to complete all 3 levels on time. Here are the point projects again for reference:

A 70-80 point project correctly implements the stepper motors, keypad, NOKIA 5110 display, and IR Sensor. The system must read the IR sensor and ask for the passcode on the NOKIA 5110 display upon motion. Upon correct entry of a previously set passcode (123) on the

keypad, the stepper motors, representing doors, must rotate allowing entry. Buttons are used to toggle the light bulb which represents light in a garage.

A 80-90 project correctly implements everything described in the 70-80 project but in addition, the lock can be opened by entering a value on an iPhone app that connects to the Bluetooth HM-10 such as Serial. The Bluetooth HM-10 module is used to communicate with the ATmega1284 instead of the keypad, adding a “wireless” element to the project.

A 90-100 project correctly implements everything described in the 80-90 project but in addition, the stepper motors are attached to visuals such as a paper door and paper garage door to better visually demonstrate the security system. The light bulb will be in a cardboard box so lighting the bulb is shown as lighting a room. In addition, the user will have a menu on the NOKIA 5110 from which they can choose to either see the passcode or enter the passcode, on motion detection.

Incomplete components

No incomplete components.

YouTube Links

- Short video: https://www.youtube.com/watch?v=GVA2EdrXpKQ
- Longer video: https://www.youtube.com/watch?v=HFIyPQfFuIE

Testing

I tested my code everyday as I found errors often. Getting each tool to work such as the IR sensor, keypad, NOKIA 5110 display, and Bluetooth module required testing to get working as intended.

- 70 - 80 point project
    - Tested by my lab partner and friends from CS122A labs.
    - IR SENSOR:
        - I set a light bulb to light up upon motion detection. If it lit, the IR sensor was detecting motion and working properly.
    - NOKIA 5110 LCD Display:
        - I sent test strings such as “Hello World” to display. Then I displayed the string “motion detected” when the IR sensor detected motion. This allowed me to ensure both components worked together.
    - KEYPAD:
        - I tested the keypad by entering input and outputting it on the NOKIA 5110. If whatever I entered appeared it meant the keypad works.
    - STEPPER MOTORS:
        - I tested the stepper motors by setting them to turn whenever a key was pressed on the keypad.

- 80 - 90 point project
    - Tested by my friend who is a freshman EE major.
    - BLUETOOTH HM-10:
        - I set the light bulb to light if the microcontroller received anything via USART from the HM-10. Then I sent strings from the iPhone 7 to be output on the NOKIA 5110 display. Then I sent a code on the iPhone 7 and caused it to turn the stepper motors.
- 90 - 100 point project
    - Tested by my roommate who is a business major.
    - To test that the paper representing doors don't interfere with the IR sensor, I tested to find the required distance between the cardboard and motion detection.

Known Bugs

- I was having trouble implementing the PIR sensor because it was either too sensitive too light or it would not pick up any motion at all. I rewrote my proposal, changing the PIR sensor to a regular IR sensor. The IR sensor was easy to implement and allowed me to complete the project on time.

Resume/Curriculum Vitae (CV) Blurb

Smart Doors System:

- Created a Bluetooth door security system, in C using embedded systems and the ATMEGA1284 microprocessor. An IR sensor detects motion and asks for the passcode. Upon correct input on a keypad or on the "Serial" app, the stepper motors rotate representing doors unlocking and locking.
- Hardware: Bluetooth HM-10 module, NOKIA 5110 LED Display, Stepper motors, keypad, IR Sensor, buttons and bulbs
- Software: C programming: state machines, Free RTOS scheduler, pointers and structs
- GITHUB project link: https://github.com/sgoka001/BTSecurityDoor

Future work

- I will change the Bluetooth password feature. Currently, if the device pairs with the HM-10 and sends any entry the doors unlock. I will change it so that a certain password is required to be input on the mobile device to unlock.
- I will remove the "Check Passcode" option on the menu. I included it for the project for demo and testing purposes. I will change it to a "Change Passcode" option which will allow the user to enter Admin mode and change the passcode. I will implement EEPROM to store the new passcodes so the passcode doesn't change when the system is restarted.
- Once I have implemented the above features, I will use the project to apply for a job at the company, Security Door Controls. This company, upon which I completed a Company Research Report, specializes in creating Bluetooth security doors, similar to my project.

- If I deployed the Smart Doors System and implemented them in my own home, I would need bigger stepper motors as the wooden doors are much heavier and require more power to turn. I would encase my NOKIA 5110 display, IR sensor and Bluetooth HM-10 module in a glass case to protect them from being meddled with. I would stick the keypad to the wall so it's easier to see, in case the Bluetooth module isn't working or the user doesn't have an iPhone to connect.

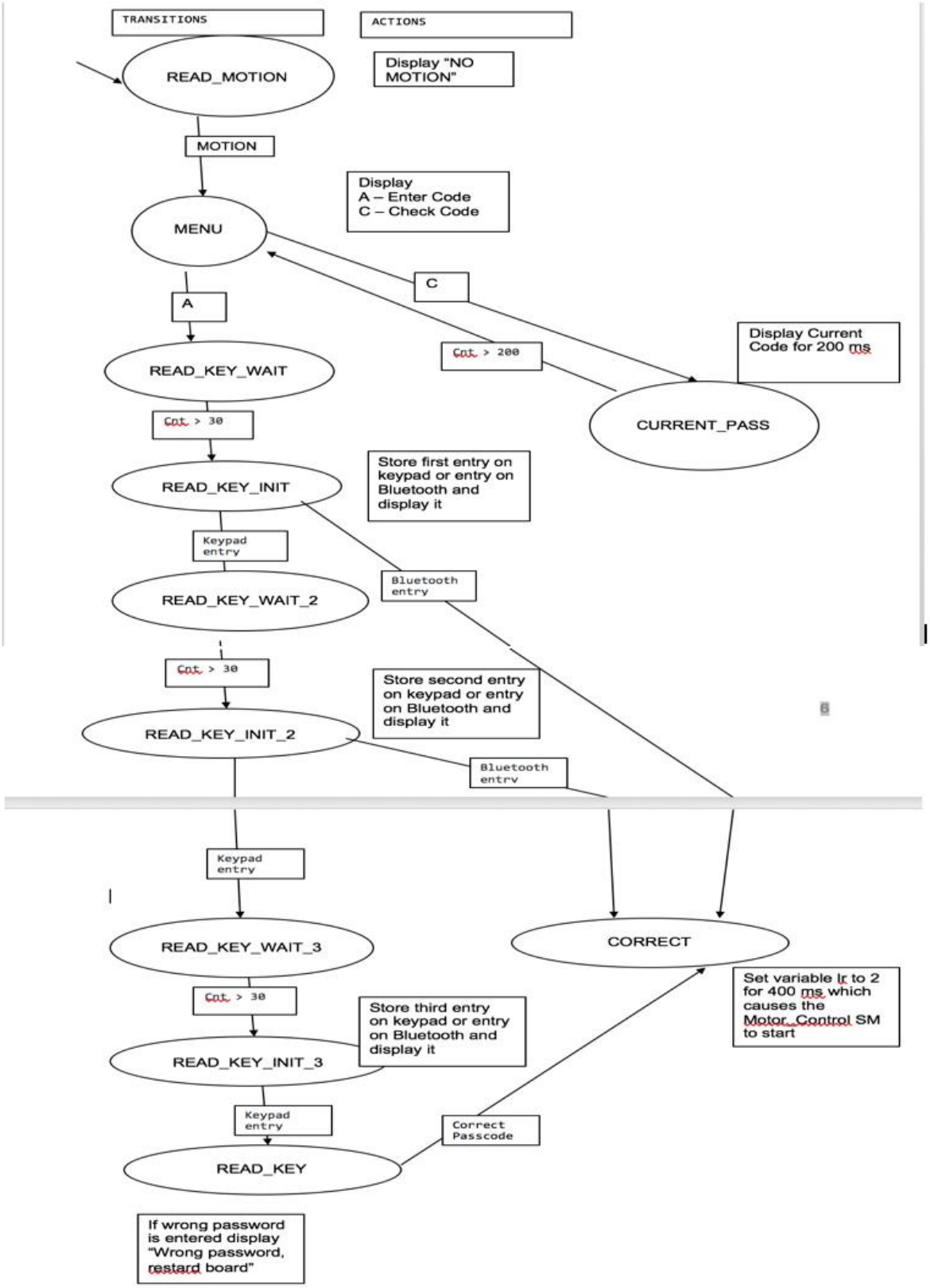
References

- NOKIA_5110.H and NOKIA_5110.C were acquired from: https://github.com/LittleBuster/avr-nokia5110
  I used the functions from these files to write messages to my NOKIA 5110 LCD Display.

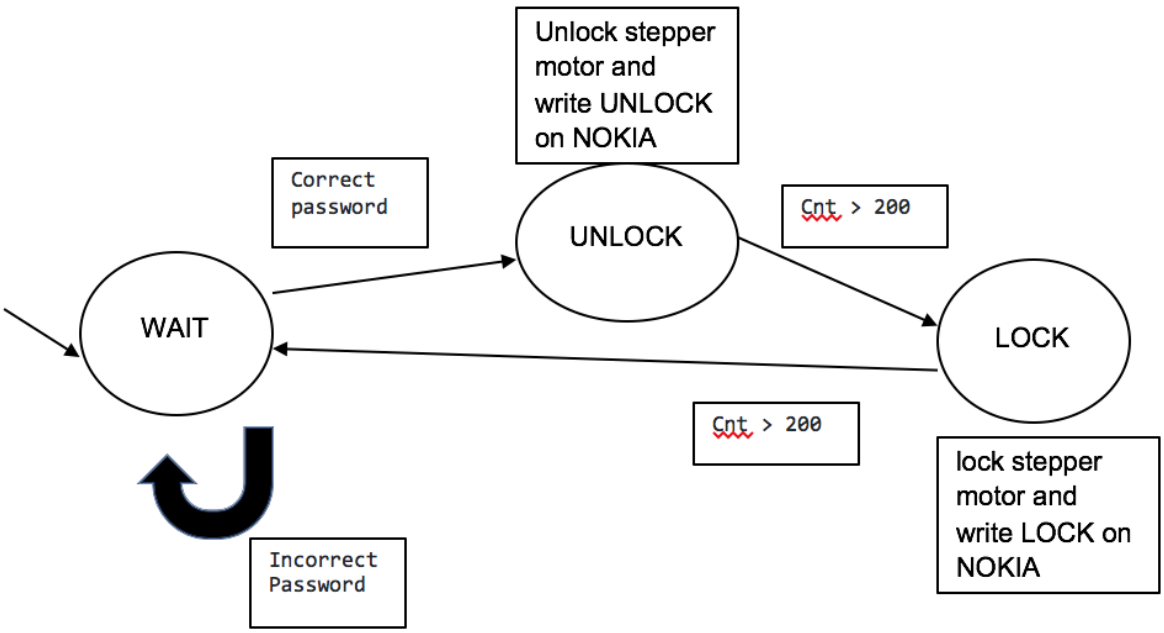
Appendix

Passcode_Wait: Upon motion detection, the menu state allows the user to either check the passcode or enter it. Upon correct entry (123) on the keypad or on the iPhone via Bluetooth, the lr variable is set to 1. This initiates the next SM.

(image on next page)


TRANSITIONS
ACTIONS
READ_MOTION
Display "NO MOTION"
MOTION
Display
A – Enter Code
C – Check Code
MENU
C
A
Cnt > 200
Display Current Code for 200 ms
READ_KEY_WAIT
CURRENT_PASS
Cnt > 30
READ_KEY_INIT
Store first entry on keypad or entry on Bluetooth and display it
Keypad entry
Bluetooth entry
READ_KEY_WAIT_2
Cnt > 30
Store second entry on keypad or entry on Bluetooth and display it
READ_KEY_INIT_2
Bluetooth entry
Keypad entry
READ_KEY_WAIT_3
CORRECT
Cnt > 30
Store third entry on keypad or entry on Bluetooth and display it
Set variable lr to 2 for 400 ms which causes the Motor_Control SM to start
READ_KEY_INIT_3
Keypad entry
Correct Passcode
READ_KEY
If wrong password is entered display "Wrong password, restard board"

Motor_Control: If the lr variable is set to 1 by the Passcode_Wait SM, the Motor_Control SM turns the motor to the right for 200 ms representing unlocking the door, then turns left for 200 ms representing locking the door. If the button is pressed, the 2nd motor is opened instead of the 1st.



link to github: https://github.com/sgoka001/BTSecurityDoor