Group 4 - (TBD)

Systems Specifications

Solenoid Control Using Fingerprint Sensor and NFC Reader

California State University, San Bernardino

November 11, 2024 (Fall 2024)

CSE 5208-01

Shunsuke Akiya, Christopher Cao, Rafael Ceja, Miguel Rodriguez,
Matthew Uy

1. **The Concept**

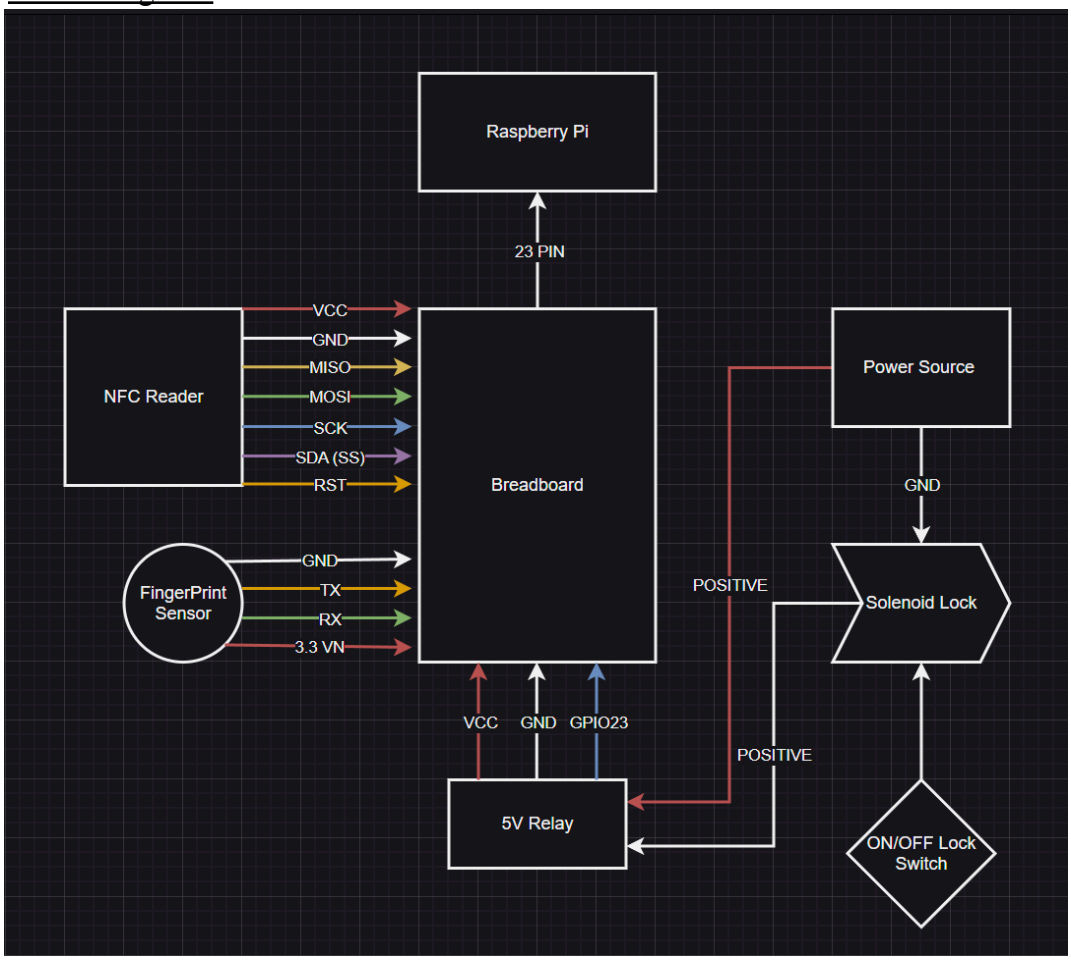
The iOpen smart lock can be unlocked using methods such as a fingerprint sensor and NFC reader. Both methods communicate with the Raspberry Pi and a web-based application. A solenoid acts as a door lock and unlocks when one of the above methods is used. It will then lock by itself after a short period of inactivity. The problem iOpen solves is users having to manually lock and unlock doors with a traditional key. With this smart lock, users will have multiple methods to unlock their doors as well as a traditional key which acts as a failsafe should the RFID card or fingerprint sensor unexpectedly stop working. iOpen's primary function is to improve users' quality of life and overall security. Users will no longer have to be concerned with the wear and tear of physical keys due to the lower maintenance of the fingerprint sensor and NFC reader.

Changes from requirements specification:

- Power source will now be a 12V battery rather than a 9V
    - 9V not powerful enough
    - Battery casing holding AA batteries that add up to 12V
- Arduino used for sensor testing
- Web-based application will be used instead of smartphone app to interact with smart lock
- Addition of missing components from requirements specification (solenoid/relay)
- Relay used to activate/deactivate solenoid
- Near Field Communication replacing RFID to have 2-way communication
- Purchase list modified to now only contain the following components:
    - 5V Fingerprint Sensor
    - NFC Reader with card and tag
    - 12V Solenoid
    - 3 AA batteries with casings/housings to hold batteries
    - Relay
    - On-off key lock switch

2. Inputs/Outputs and System Block Design

Block Diagram:



Inputs/Outputs:

- Raspberry Pi:
  - Inputs: Obtains data from fingerprint sensor, NFC reader, physical keyhole, and HTTP data from web app
  - Outputs: Signal sent to relay to unlock solenoid; feedback given to web app
- Fingerprint Sensor:
  - Inputs: Fingerprint from user, power from Raspberry Pi
  - Outputs: Unique ID assigned to fingerprint to unlock solenoid, sent to Raspberry Pi
- NFC Reader:
  - Inputs: Tag and card, power from Raspberry Pi
  - Outputs: Unique IDs assigned to tag and card to unlock solenoid, sent to Raspberry Pi
- Physical Keyhole:
  - Inputs: Physical key
  - Outputs: Solenoid unlocked manually by user with physical key

- Web App:
    - Inputs: HTTP data status codes on whether sent data successful or not
    - Outputs: Commands sent to Raspberry Pi to accomplish various tasks such as adding a new Wi-Fi connection to Raspberry Pi, pressing a button to unlock and lock the solenoid, etc
- Solenoid Lock:
    - Inputs: Power from relay to function to unlock/lock as needed
    - Outputs: Unlocking and locking
- 3 AA Batteries:
    - Inputs: N/A
    - Outputs: Provides power for solenoid and Raspberry Pi as needed

3. **Specification of the Blocks (Shun Akiya)**

Functionality:

- Raspberry Pi:
    - The Raspberry Pi is the brain of the entire project. All of the programming is done on it through the Thonny IDE and utilizes the Raspberry Pi OS (based on Debian Buster). All of the circuitry and wiring is also done on it, specifically by wiring the components to the 40 Pin GPIO header. To access
- AA Batteries:
    - The solenoid will utilize about 9 AA batteries, which is used to power the solenoid. The solenoid and batteries will be connected to the Raspberry Pi and a relay, allowing for unlocking and locking capabilities through enabling and disabling the relay.

- Relay:
    - As mentioned in the previous point, the relay's primary use is to allow arbitrary components to be enabled and disabled, in this case the 12V solenoid. By sending an input to the relay through programming (the corresponding GPIO pin that it's plugged into), it enables the solenoid to contract. To this, we can add some base conditions such as "if the fingerprint or NFC reader is true, then send an input to port 5. Else do nothing."
- 12V Solenoid:
    - Acts as a door lock that is unlocked using either the fingerprint sensor, NFC reader or with a physical key hole. It will be controlled using a relay.
- NFC Reader:
    - One of the three ways that will unlock the door. Within the NRC reader, there will be 3 ways to unlock it:
        - Using your phone
        - Pre Attached NFC Tag
        - Pre Attached NFC Card
    - The UID of the tag and card will already be installed in the lock, so the user will not have to set it up. The only setup that is required is for the phone.
- 5V Fingerprint Sensor:

    - The 2nd of the three ways that will unlock the door. The fingerprint sensor can register multiple fingerprints, and will allow the user to unlock the door. The user's thumb will be used as it is the most comfortable and logical way for the reader. If a registered user scans their fingerprint, the LED around the sensor will light up green, indicating a successful unlock. If not, it will light up red.
- On-Off Key Lock Switch:
    - Failsafe used to unlock solenoid should the NFC reader and/or fingerprint sensor fail.

Technical Specifications:

- Power requirement:
    - As of now, we discovered that a 9V battery was not enough to power the solenoid, so we switched to using AA batteries. After watching some videos with a very similar solenoid setup, we agreed on using 3 AA batteries. Though, we're currently struggling with being able to power the solenoid with 9 AA batteries.
- Communication Protocol:
    - The physical circuitry will be using the NFC protocol for the reader, and for the backend side, we will be using the HTTP protocol for fetching and requesting data.
        - In JavaScript, the fetch() function is used to send and request data. In this case, we will be sending data from the web app to the Raspberry Pi for setups such as adding and removing a fingerprint so that the sensor can store it in memory, pressing a button to lock and unlock the solenoid, adding a new Wi-fi connection for the Raspberry Pi, and adding a new phone for the NFC reader.

Inputs/Outputs:

- Raspberry Pi:
    - Inputs:
        - Data from the Fingerprint sensor
        - Data from the NFC reader
        - Input from the Physical keyhole
        - HTTP data sent from the Web app
    - Outputs:
        - Control signal to the relay for enabling/disabling the solenoid lock
        - Feedback to the Web app
    - Interactions:
        - Receives data from the Fingerprint sensor and NFC reader to validate access
        - Sends signal to the relay to unlock the solenoid
        - Communication with the Web app for remote control and other setups
- Fingerprint Sensor:
    - Inputs:
        - Power from the Raspberry Pi
        - Finger placed on sensor

    - Outputs:
        - Fingerprint data sent to the Raspberry Pi
    - Interactions:
        - Sends data to the Raspberry Pi to determine if the fingerprint is recognized or not
- NFC Reader:
    - Inputs:
        - Power from the Raspberry Pi
        - NFC card/tag placed near the reader
    - Outputs:
        - NFC data sent to the Raspberry Pi
    - Interactions:
        - Sends NFC tag data to the Raspberry Pi, which determines if the tag/card is recognized or not
- Physical Keyhole
    - Inputs:
        - Physical key
    - Outputs:
        - Unlocks the solenoid
- Web App
    - Inputs:
        - HTTP status codes on whether the sent data was successful or unsuccessful.
    - Outputs:
        - Sent fetch() commands to the Raspberry Pi for tasks such as adding and removing a fingerprint so that the sensor can store it in memory, pressing a button to lock and unlock the solenoid, adding a new Wi-fi connection for the Raspberry Pi, and adding a new phone for the NFC reader.
    - Interactions:
        - Allows remote management and access of the the door lock
- Solenoid Lock:
    - Inputs:
        - Power from the relay to enable/disable the lock
    - Outputs:
        - Physical locking/unlocking
    - Interactions:
        - Enabling or disabling of the solenoid when relay is activated by the Raspberry Pi
- 3 AA Batteries:
    - Outputs:
        - Power for the system
    - Interactions:
        - Powers the solenoid and/or the Raspberry Pi

4. **System Description**

Block Interaction:

- Raspberry Pi:
  - Communicates with fingerprint sensor and NFC reader to confirm access
  - Once access confirmed, signal sent to relay to unlock solenoid
  - Remote control by communicating with web app
- Fingerprint Sensor:
  - Fingerprint sent to Raspberry Pi whether received fingerprint is recognized
  - 1 of 3 methods to unlock solenoid
- NFC Reader:
  - Tag and/or card data sent to Raspberry Pi to determine if they are recognized
  - 1 of 3 methods to unlock solenoid
- Physical Keyhole:
  - Unlocks solenoid through physical key
  - 1 of 3 methods to unlock solenoid
- Web App:
  - Connects with solenoid for remote access
- Solenoid Lock:
  - Unlocks/locks when relay activated by Raspberry Pi
  - Powered by 3 AA batteries
- 3 AA Batteries:
  - Powers solenoid lock

5. **System Analysis**

Testing and Results (Figures, Tables, Graphs):

Testing with Solenoid Lock	**Result**
Voltage: 1 - 9v Battery	Fail
Voltage: 9 - 1.5v Battery (AA Battery)	Success
Multiple Testing for Fingerprint/NFC	Fail →Success
Solenoid Activation with Code	Fail

Analysis of Key Metrics:

- Performance
  - System performs as expected according to voltage. Solenoid receives power from the battery which then allows unlocking/locking. However, there were problems in achieving the power needed for solenoid.

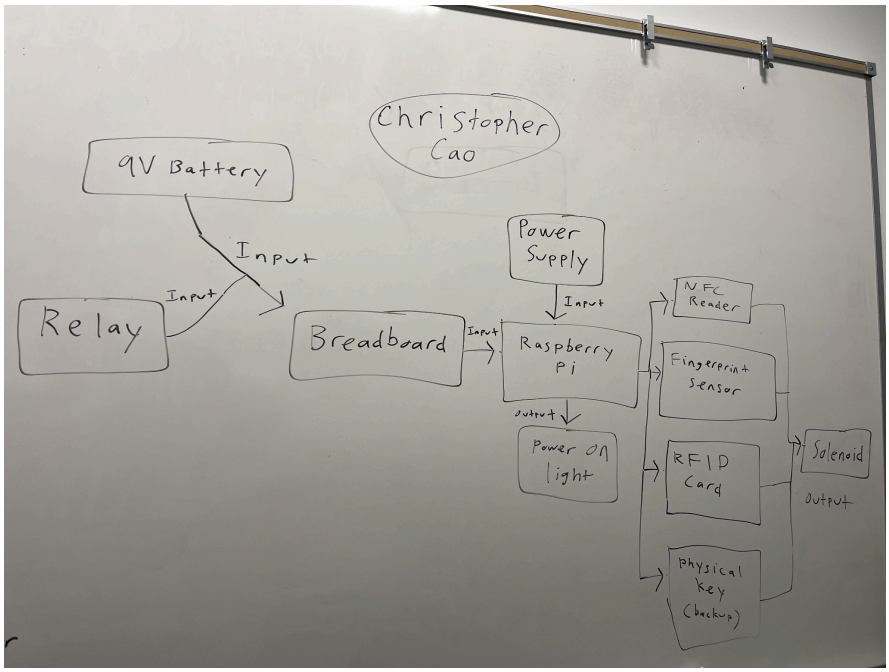
- Accuracy
    - Device and commands are accurate, Raspberry Pi is able to communicate with the breadboard which then can read the NFC reader or Fingerprint Sensor that can activate the solenoid lock.
- Efficiency
    - Device is efficient, uses as minimal power necessary to activate solenoid lock. Although lower power/power source may deteriorate the system because it will draw a higher current, the cost of equipment like batteries are easily replaceable and items are cost efficient.

Problem Areas:

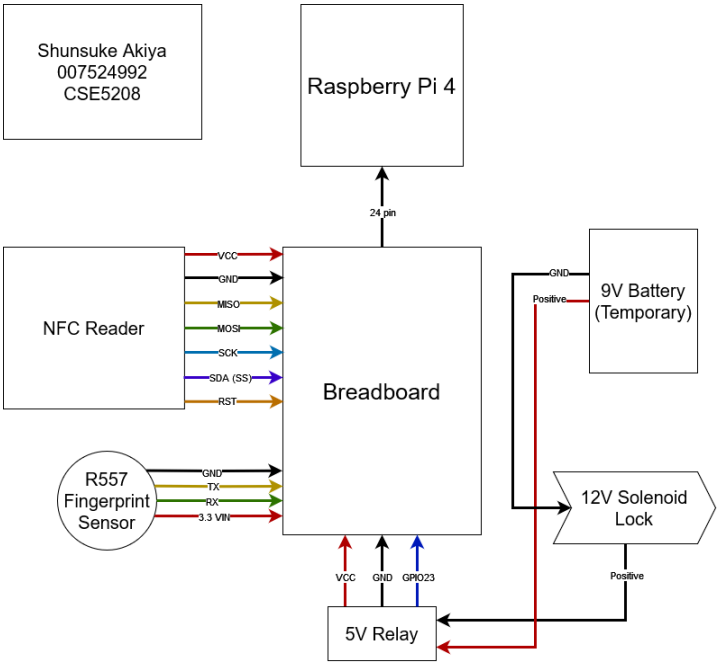
- Singular 9V battery not powerful enough to power solenoid
    - Replaced with 3 AA batteries housed in casing
- One AA battery exploded during testing
    - Immediately replaced

6. Individual Block Diagrams

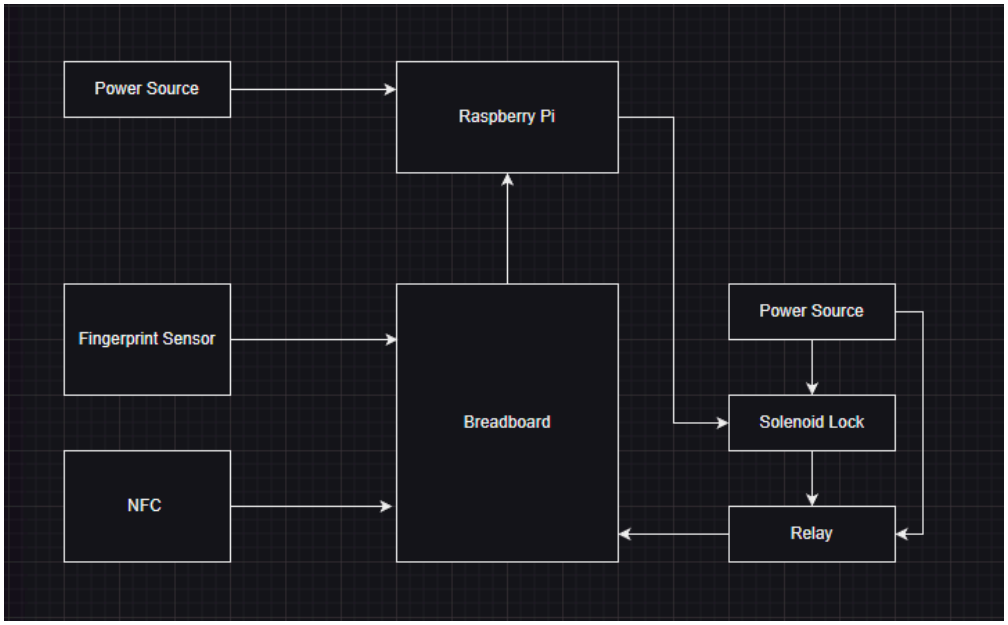
Christopher Cao:



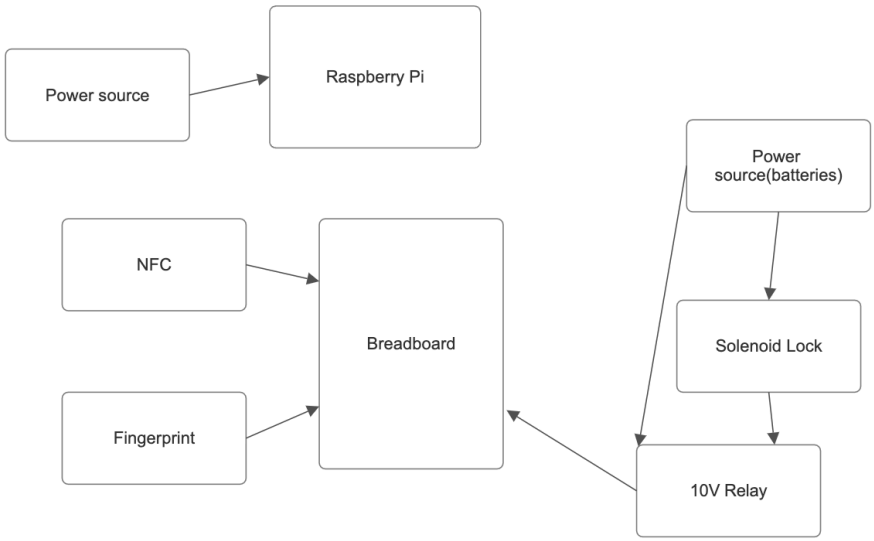
Shun Akiya:



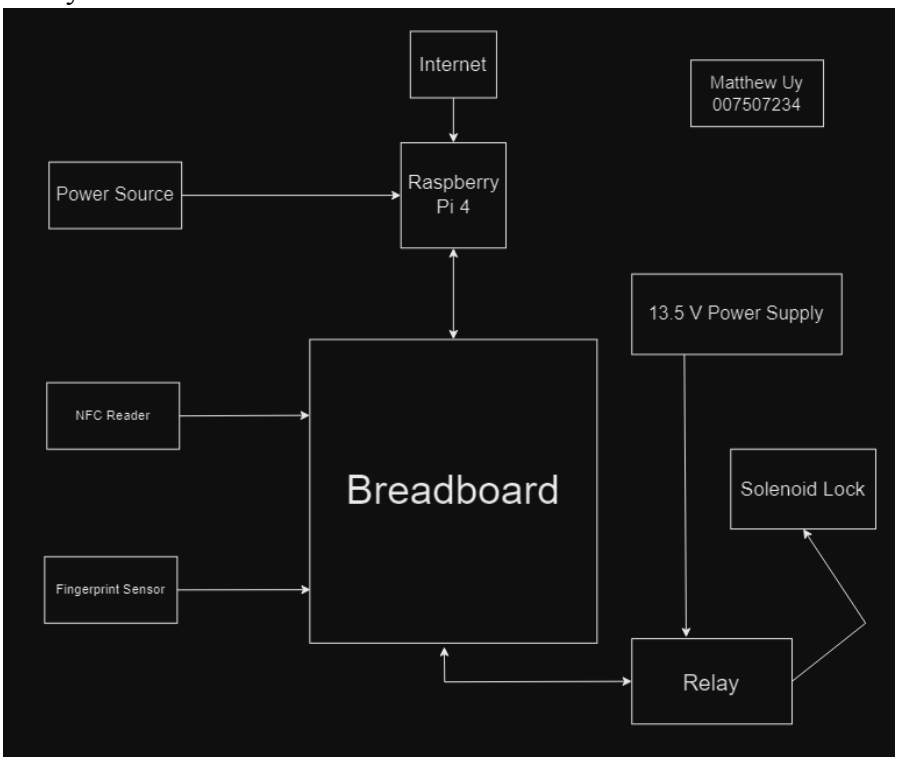
Miguel Rodriguez:



Rafael Ceja


Power source
Raspberry Pi
Power source(batteries)
NFC
Breadboard
Solenoid Lock
Fingerprint
10V Relay

Matthew Uy:


Internet
Matthew Uy
007507234
Power Source
Raspberry Pi 4
13.5 V Power Supply
NFC Reader
Breadboard
Solenoid Lock
Fingerprint Sensor
Relay