

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



PROJECT REPORT – GROUP 03
EMBEDDED SYSTEMS

ARDUINO SAFE LOCK SYSTEM

Lecturer: TS. Ngô Lam Trung

Class: 147829

Group Members:

No	Name	ID
1	Nguyễn Thọ Đạt	20215193
2	Trần Tùng Dương	20215190
3	Cho Su Wai	2023T070

Hà Nội, July 2024

Nội dung

I. Introduction.....	Error! Bookmark not defined.
1.General function	Error! Bookmark not defined.
2.Work distribution	Error! Bookmark not defined.
II. Hardware design	Error! Bookmark not defined.
1.Hardware components.....	Error! Bookmark not defined.
2. Arduino Uno.....	Error! Bookmark not defined.
3.Locking Mechanism.....	Error! Bookmark not defined.
III.Software design	Error! Bookmark not defined.
1.Software architechture	Error! Bookmark not defined.
2.Software functions	Error! Bookmark not defined.
IV.Result	Error! Bookmark not defined.

I. Introduction

1. Project description:

The Safe Lock program is a secure access control system using an Arduino Uno, combining RFID (Radio-Frequency Identification) and keypad authentication to manage access. This system uses an RFID reader, a keypad, an LCD display, a servo motor, and EEPROM storage to provide security for various applications.

2. General functions:

- To unlock the user presses * and enters password or scans with a valid card. The door will remain locked if the password or card is invalid.
- To lock the user press D
- To change password, the user press #, enter old password then new password.

3. Work Distribution

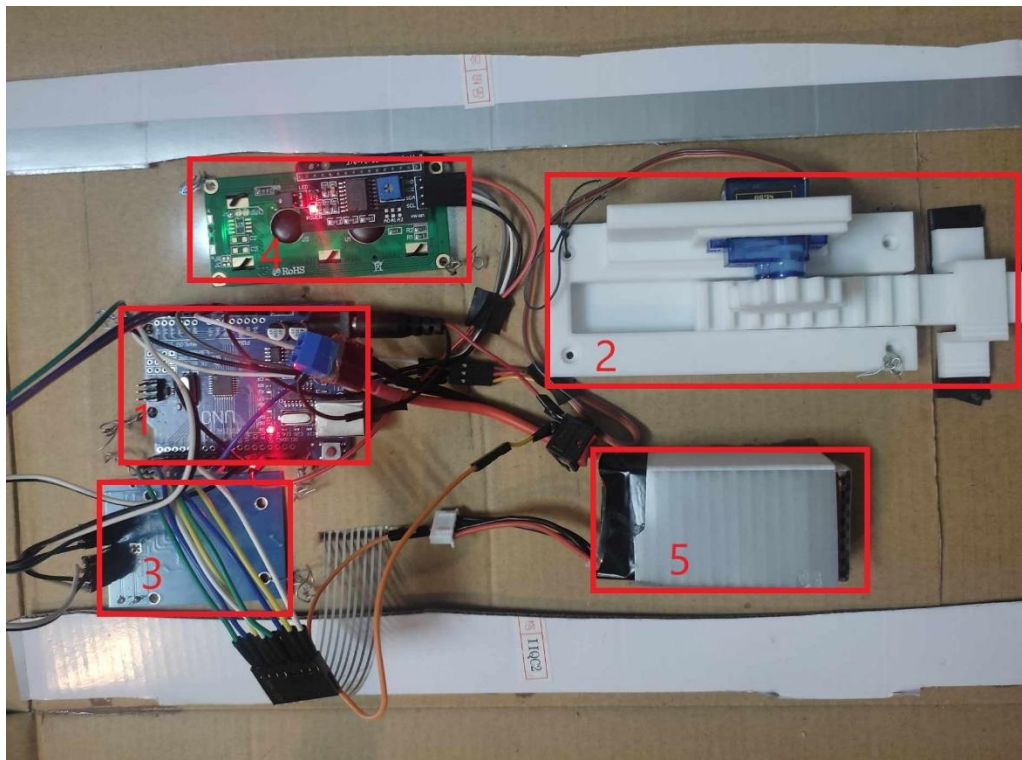
Member	Work
Nguyễn Thọ Đạt	Software + Hardware Design Report
Trần Tùng Dương	Software + Hardware Design Report
Cho Su Wai	Software + Hardware Design Report

II. Hardware Design

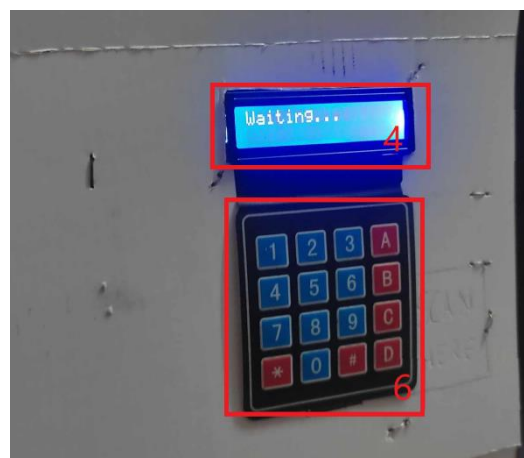
1. Hardware components

1. Arduino UNO R3 CH340

2. 3D-printed lock + Servo motor SG90
3. RFID RC522
4. LCD 1602 I2C
5. Battery
6. Keypad 4x4



Back side



Front side

Hardware Connections

- **RFID Reader:**

- SS_PIN to A2
- RST_PIN to A3
- Connect other pins according to the MFRC522 datasheet (MOSI to 11, MISO to 12, SCK to 13)

- **Keypad:**

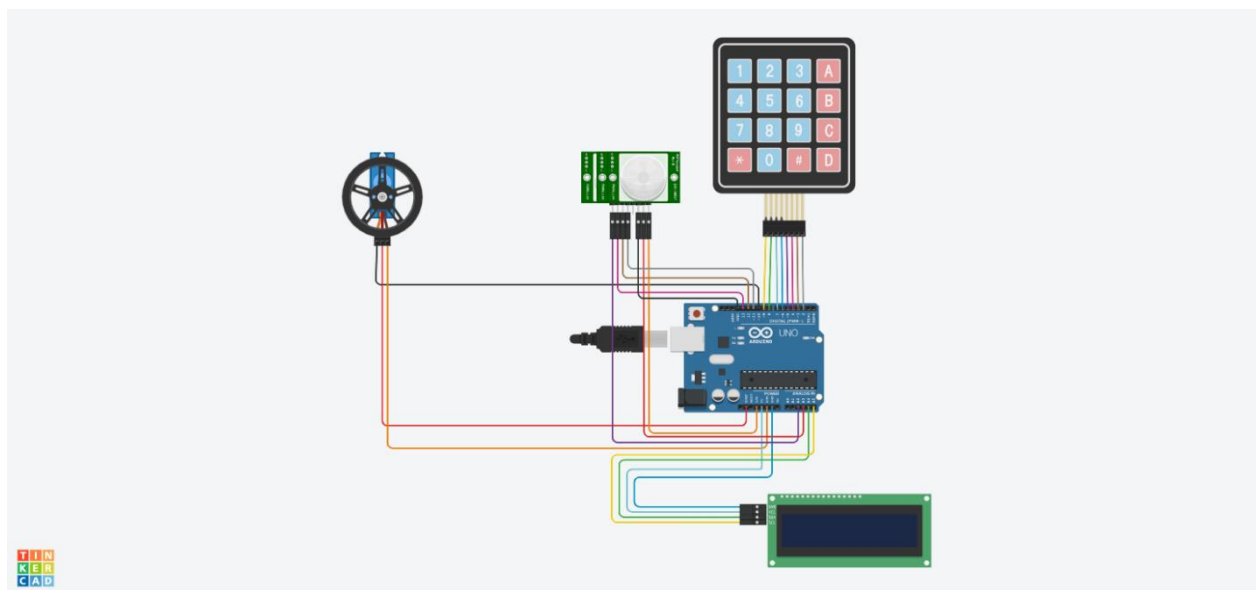
- Connect row pins to Arduino pins 9, 8, 7, 6
- Connect column pins to Arduino pins 5, 4, 3, 2

- **LCD Display:**

- Connect according to the LiquidCrystal_I2C library documentation (SDA to A4, SCL to A5)

- **Servo Motor:**

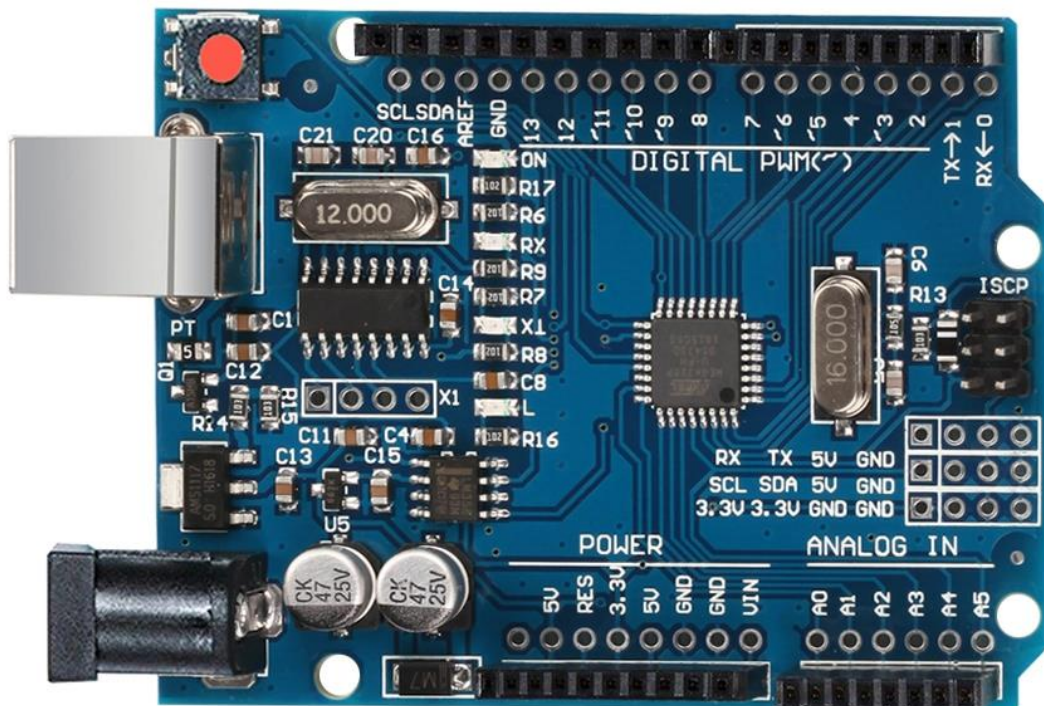
- Signal pin to Arduino pin 10



2. Arduino UNO

Arduino Uno:

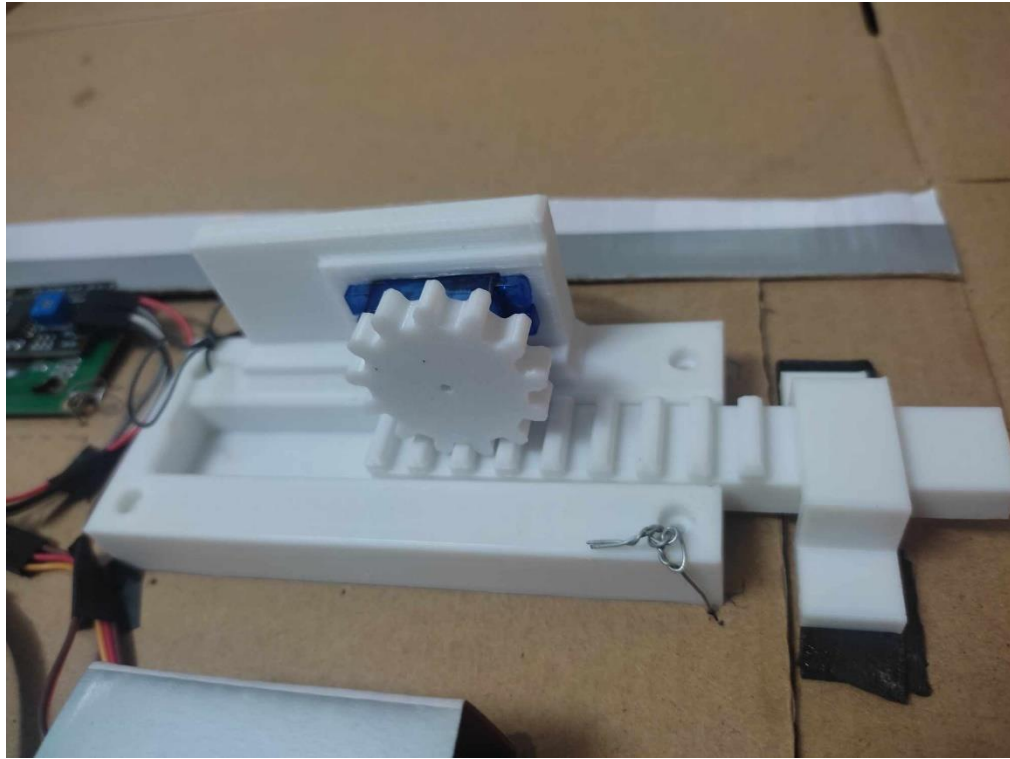
- This is the central microcontroller managing the entire system.
- It interfaces with all connected components and executes the main control logic



Arduino UNO Kit

3. Locking Mechanism

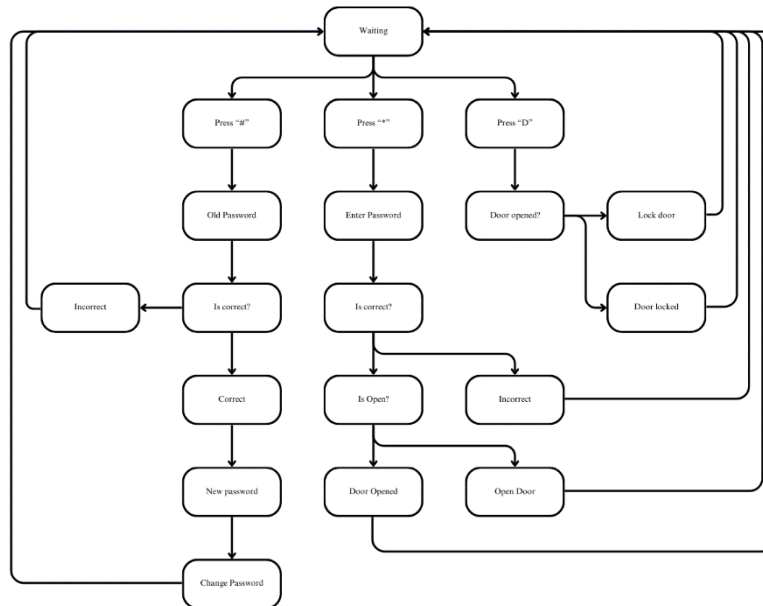
A simple locking mechanism is 3d-printed and then integrated with a Servo motor. The motor will spin to lock or unlock position when signaled.



3d-printed lock and Servo motor

III. Software Design

1. Software architecture



2. Software functions

2.1. Setup

This function is executed once when the microcontroller is powered on or reset. It initializes various components and settings required for the program to function correctly:

- It initializes various components and settings required for the program to function correctly.
- The SPI bus is initialized using `SPI.begin()`, which is essential for communication with SPI devices (the RFID reader in this case)
- The code then initializes an RFID reader.

- Then attach a specified pin to the servo motor, allowing the program to control the servo motor.
- The code reads the current master password and lock's state from EEPROM

```
void setup() {
  lcd.init();
  lcd.backlight();
  SPI.begin();           // SPI bus
  mfrc522.PCD_Init();
  myServo.attach(signalPin);
  lcd.setCursor(0, 0);
  lcd.print("Waiting...");
  // Initialize Master password from EEPROM
  for (int i = 0; i < Password_Length - 1; i++) {
    EEPROM.get(i, Master[i]);
  }

  // Initialize Lock state from EEPROM
  EEPROM.get(100, isOpen);
}
```

Setup function

2.2. Handle User Input

- The system will display a default waiting screen to wait for user input
- The program continuously scans for a card. If a valid card is detected and door is locked, the system will send signal to servo to unlock. If the card is invalid, the door stays unlocked.
- If the user chooses *, the system asks user to enter password. If password is correct and door is locked, the system will signal servo to unlock the door.
- If the user chooses D and door is unlocked, the system will signal servo to lock the door.
- If the user chooses #, the system asks user to enter password. If the password is correct, user can update new password.

```

void loop() {
  readID();
  if (tagID == tag_UID || tagID == tag_UID1) {
    DisplayText(0, 0, "Valid card");
    lcd.clear();
    if (isOpen) {
      DisplayText(0, 0, "Door is opened!");
      lcd.clear();
    } else {
      // Unlock();
      SetLock(0,ANGLE,true);
    }
    tagID = "";
    WaitKey();
  } else if (tagID != tag_UID && tagID != NULL){
    DisplayText(0, 0, "Invalid card");
    tagID = "";
    lcd.clear();
    WaitKey();
  }
  customKey = customKeypad.getKey();
  if (customKey == '*') {
    DisplayText(0, 0, "Enter Password:");
    GetCode();

    if (data_count == Password_Length - 1) {
      lcd.clear();

      if (!strcmp(Data, Master)) {
        DisplayText(0, 0, "Correct");
        if (isOpen) {
          DisplayText(0, 0, "Door is opened!");
          delay(1000);
        } else {
          // Unlock();
          SetLock(0,ANGLE,true);
        }
      } else {
        DisplayText(0, 0, "Incorrect");
      }
      WaitKey();
    }
  }

  if (customKey == '#') { // To change the code it
    DisplayText(0, 0, "Old Password:");
    GetCode();
    if (data_count == Password_Length - 1) {
      lcd.clear();

```

```

      if (!strcmp(Data, Master)) {
        DisplayText(0, 0, "New Password: ");
        ChangePassword();
      } else {
        DisplayText(0, 0, "Incorrect");
      }
      WaitKey();
    }
  }

  if (customKey == 'D') {
    if (isOpen) {
      // Lock();
      SetLock(180,ANGLE,false);
      DisplayText(0, 0, "Door locked");
    } else {
      DisplayText(0, 0, "Door locked!");
    }
    WaitKey();
  }
}

```

Main loop to read from card and keypad

- The GetCode() function is used to read user code entered from the keypad and store the code in a data array. It will display the entered code as * for additional security.

```

✓ void GetCode() { // Getting code sequence
✓   while (data_count < Password_Length - 1) {
     customKey = customKeypad.getKey();
✓   if (customKey != NULL) {
       Data[data_count] = customKey;
       lcd.setCursor(data_count, 1);
       // lcd.print(Data[data_count]);
       lcd.print("*");
       data_count++;
     }
     delay(100);
   }
}

```

GetCode Fucntion

- readID() function is used to read user's card. When an RFID tag is presented to the reader, the master (Arduino Uno) communicates with the MFRC522 RFID reader over the SPI bus to read the tag's unique identifier (UID).
- The master sends commands and receives data from the RFID reader through the MOSI, MISO, and SCK lines.

```

✓ boolean readID() {
  //Check if a new tag is detected or not. If not return.
  if (!mfrc522.PICC_IsNewCardPresent()) {
    return false;
  }
  //Check if a new tag is readable or not. If not return.
✓  if (!mfrc522.PICC_ReadCardSerial()) {
    return false;
  }
  tagID = "";
  // Read the 4 byte UID
✓  for (uint8_t i = 0; i < 4; i++) {
    //readCard[i] = mfrc522.uid.uidByte[i];
    tagID.concat(String(mfrc522.uid.uidByte[i], HEX)); // Convert the UID to a single String
  }
  tagID.toUpperCase();
  mfrc522.PICC_HaltA(); // Stop reading
  return true;
}

```

Read card id fucntion

2.3. Handle Lock

- SetLock is used to set lock position and lock state to lock or unlock
 - o To lock set x=0, y=ANGLE=200 , isOpen =False;

- To unlock set x=180, y= ANGLE=200, isOpen = True (Rotate servo the opposite direction of unlock)

```
✓ void SetLock(int x, int y, boolean z) {
    myServo.write(x); // Rotate servo
    delay(y);         // Wait
    myServo.write(91);
    isOpen = z;
    EEPROM.put(100, isOpen);
}
```

SetLock function

2.4. Change password

- This function is used to change password. It stores new password entered by user from keypad in a Data array then updates master password and saves new master password to EEPROM.

```
✓ void ChangePassword() {
    clearData(); // Clear any existing data

    while (data_count < Password_Length - 1) {
        ✓ char key = customKeypad.getKey();
        ✓ if (key) {
            Data[data_count] = key; // Store the new key press in Data array
            lcd.setCursor(data_count, 1);
            lcd.print(Data[data_count]);
            // lcd.print("*");
            data_count++;
        }
        delay(100); // Small delay to debounce keypad input
    }

    // Update the Master password
    ✓ for (int i = 0; i < Password_Length; i++) {
        Master[i] = Data[i];
        EEPROM.put(i, Master[i]);
    }

    DisplayText(0, 0, "Pwd Changed");
}
```

Change password function

IV. Result



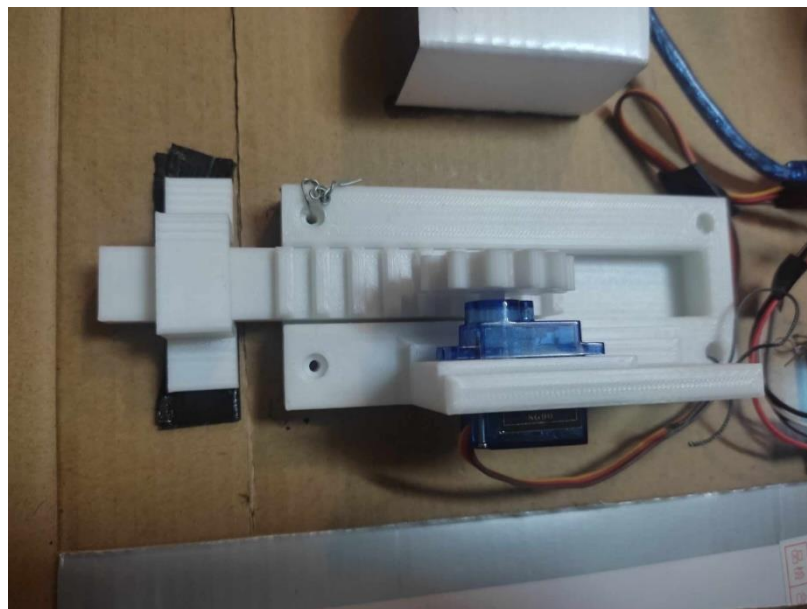
Correct Password Display



Valid Card Display



Unlock Position



Locked Position