ÇANKAYA UNIVERSITY TECHNOLOGY TRANSFER OFFICE DOOR LOCK SYSTEM WITH RFID

Project Report (EN)

Developers:

Arda YILDIZ - ardayildiz029@gmail.com
Barkın SARIKARTAL - barkinsarikartal51@gmail.com

Supporters:

Burçin TUNA - btuna@cankaya.edu.tr
H. Hakan MARAŞ - hhmaras@cankaya.edu.tr

19/03/2024

Contents:

1. Introduction

This report explains the door lock system with RFID developed for Çankaya University Technology Transfer Office Incubation Center’s door in details. This system allows people who has valid cards to enter the area in question.

2. Definition of Project

This projects aim is to create a basic door lock system with RFID. The RFID card reader reads the card ID of the card which held close to it and sends it to the Arduino Nano. Arduino Nano compares this card ID with the card IDs in its memory and decides whether it is a valid card or not. If the read card is a valid card, it triggers the relay and unlocks the lock. Otherwise, if the card read is not a valid card, the buzzer sounds a warning. Once inside the incubation center, people with valid cards can unlock the lock by placing their hand close to the contactless button.

3. List of Materials Used

a) Materials Used in the Circuit

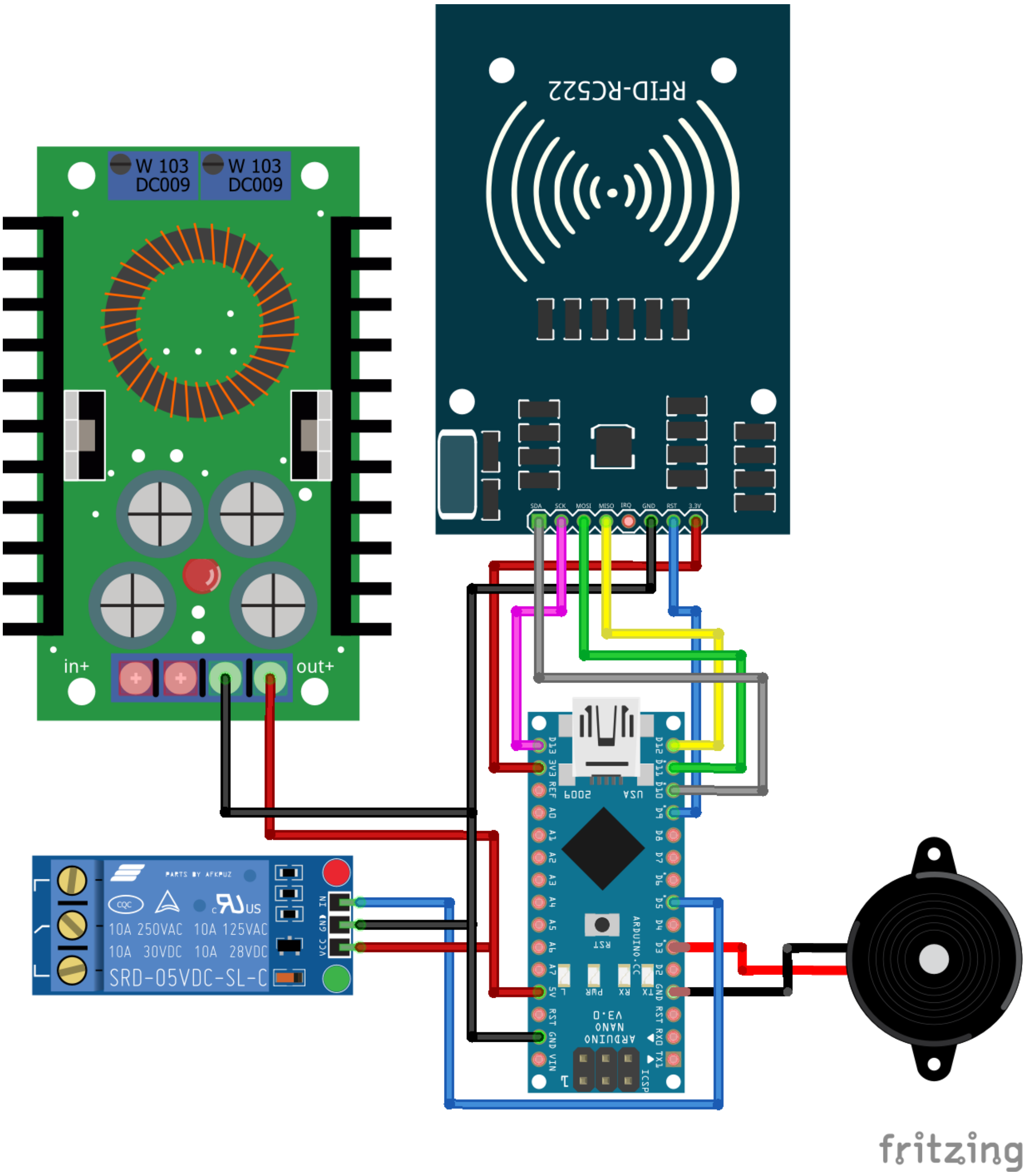
- Arduino Nano (1 piece)
- RC522 RFID Module (1 piece)
- 5V Single-Channel Relay (1 piece)
- XL4046 DC-DC Step Down Voltage Regulator (1 piece)
- 5V Active Buzzer (1 piece)
- 5x5 cm Perforated Pertinax (1 piece)
- Altınkaya SE-210 IP-67 Sealed Box (1 piece)
- Jumper Cables
- Female Headers

b) Materials Used for Lock

- Merter MAK3 Smart Lock
- Merter 12W Power Supply with Battery
- Merter MB02 Contactless Button

4. Hardware and Software

a) Circuit Diagram



Descriptions:

RC522 -> Arduino Nano:

- SDA -> D10
- SCK -> D13
- MOSI -> D11
- MISO -> D12
- IRQ -> No Connections are Made
- GND -> GND
- RST -> D9
- 3.3V -> 3.3V

5V Single-Channel Relay -> Arduino Nano:

- Vcc -> 5V
- GND -> GND
- IN -> D5

5V Active Buzzer -> Arduino Nano:

- IN -> D3
- GND -> GND

XL4016 DC-DC Step Down Voltage Regulator -> Arduino Nano:

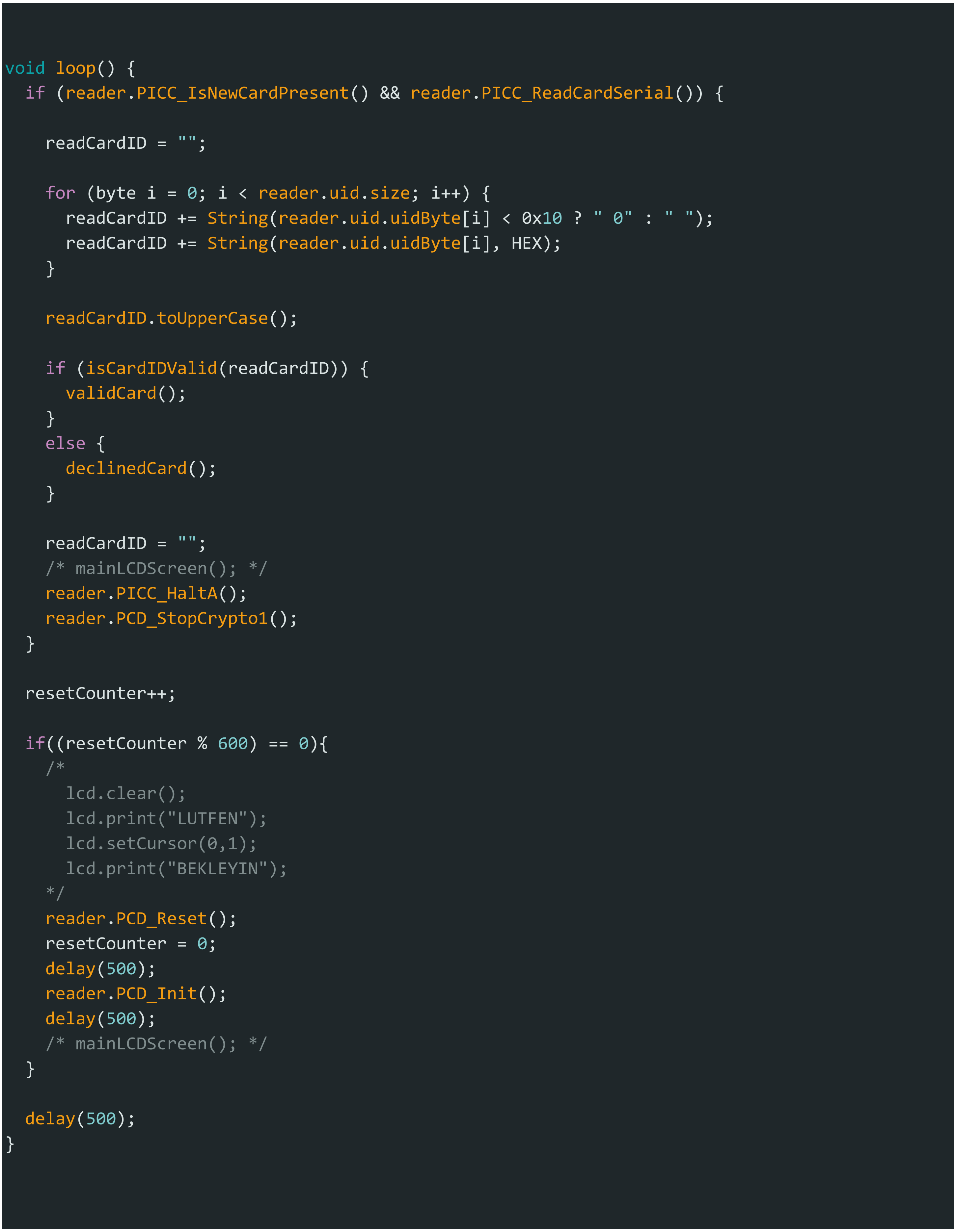
- OUT+ -> 5V
- OUT- -> GND

b) Software of System

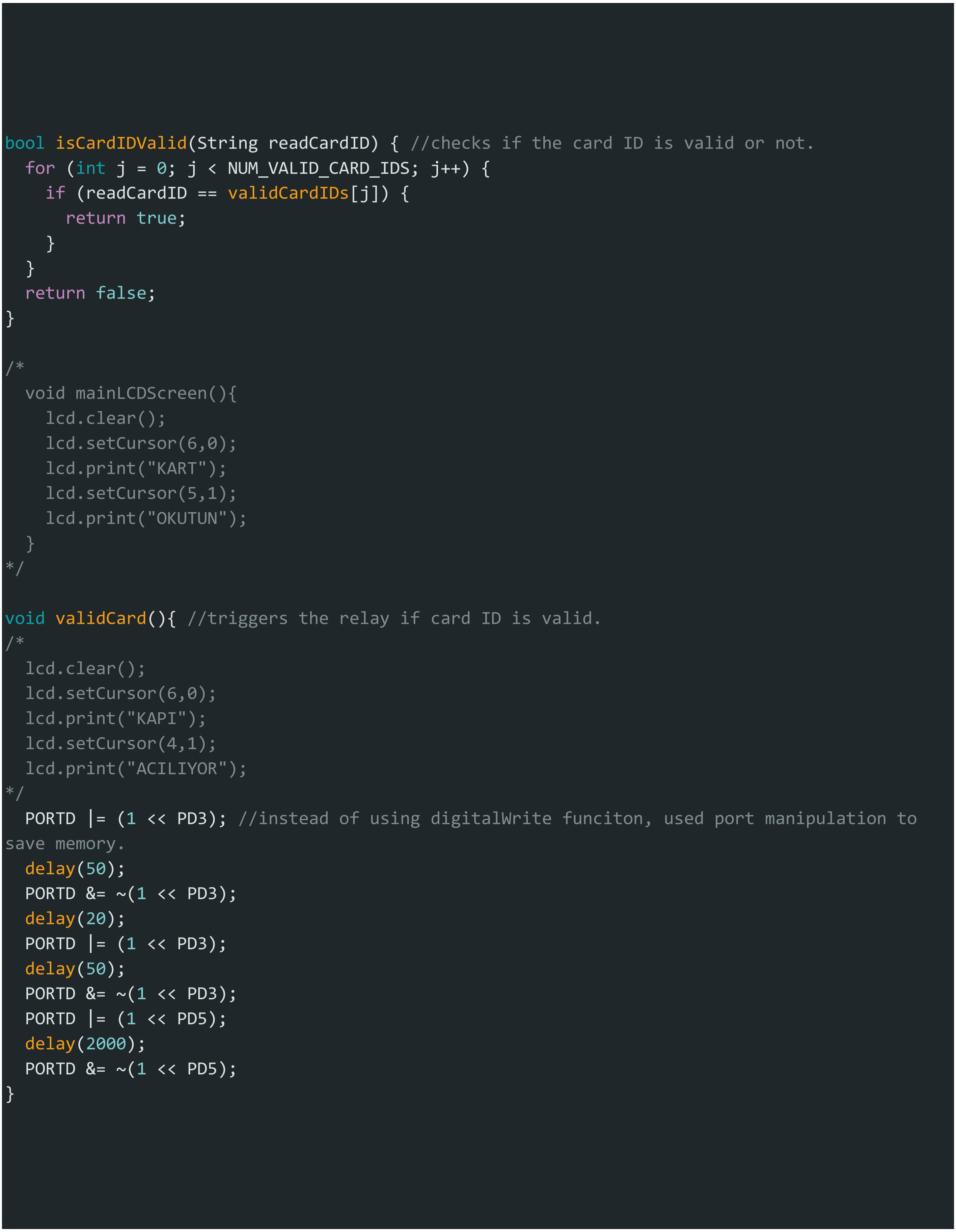
All of the codes can be found at https://github.com/barkinsarikartal/DoorLockSystemWithRFID.

b.1) UID_Okuma.ino

```
/*
  This code sequence is written to read RFID cards and trigger an electronic door lock if
card ID is valid.

  8658 bytes (28%) of program storage space, 478 bytes (23%) of dynamic memory.
  "LiquidCrystal functions are commented because they were decided not to be necessary by the
project provider."

  Contributors:
    Arda YILDIZ ardayildiz029@gmail.com
    Barkın SARIKARTAL barkinsarikartal51@gmail.com
  Supporters:
    Burçin TUNA btuna@cankaya.edu.tr
    H. Hakan MARAŞ hhmaras@cankaya.edu.tr
*/

/* #include <LiquidCrystal_I2C.h> //https://github.com/johnrickman/LiquidCrystal_I2C */
#include <MFRC522v2.h> //using version: 2.0.4
#include <MFRC522DriverSPI.h> //comes with MFRC522v2.h version: 2.0.4
#include <MFRC522DriverPinSimple.h> //comes with MFRC522v2.h version: 2.0.4
#include <MFRC522Debug.h> //comes with MFRC522v2.h version: 2.0.4
#include <Wire.h> //comes installed with Arduino IDE
#include "ValidCards.h" //can be found in this project's github repository.

/* LiquidCrystal_I2C lcd(0x27, 16, 2); */
MFRC522DriverPinSimple ss_pin(10);
MFRC522DriverSPI driver{ss_pin};
MFRC522 reader{driver};
String readCardID = "";
int resetCounter = 0;

void setup() {
  DDRD |= (1 << PD3); //defined digital 3 and 5 as output by port manipulation to save
memory.
  DDRD |= (1 << PD5);
  Serial.begin(115200);
  while(!Serial);
  reader.PCD_Init();
  /*
    lcd.init();
    lcd.backlight();
    mainLCDScreen();
  */
}
```



```
void loop() {
  if (reader.PICC_IsNewCardPresent() && reader.PICC_ReadCardSerial()) {

    readCardID = "";

    for (byte i = 0; i < reader.uid.size; i++) {
      readCardID += String(reader.uid.uidByte[i] < 0x10 ? " 0" : " ");
      readCardID += String(reader.uid.uidByte[i], HEX);
    }

    readCardID.toUpperCase();

    if (isCardIDValid(readCardID)) {
      validCard();
    }
    else {
      declinedCard();
    }

    readCardID = "";
    /* mainLCDScreen(); */
    reader.PICC_HaltA();
    reader.PCD_StopCrypto1();
  }

  resetCounter++;

  if((resetCounter % 600) == 0){
    /*
      lcd.clear();
      lcd.print("LUTFEN");
      lcd.setCursor(0,1);
      lcd.print("BEKLEYIN");
    */
    reader.PCD_Reset();
    resetCounter = 0;
    delay(500);
    reader.PCD_Init();
    delay(500);
    /* mainLCDScreen(); */
  }

  delay(500);
}
```



```
bool isCardIDValid(String readCardID) { //checks if the card ID is valid or not.
  for (int j = 0; j < NUM_VALID_CARD_IDS; j++) {
    if (readCardID == validCardIDs[j]) {
      return true;
    }
  }
  return false;
}

/*
  void mainLCDScreen(){
    lcd.clear();
    lcd.setCursor(6,0);
    lcd.print("KART");
    lcd.setCursor(5,1);
    lcd.print("OKUTUN");
  }
*/

void validCard(){ //triggers the relay if card ID is valid.
/*
  lcd.clear();
  lcd.setCursor(6,0);
  lcd.print("KAPI");
  lcd.setCursor(4,1);
  lcd.print("ACILIYOR");
*/
  PORTD |= (1 << PD3); //instead of using digitalWrite funciton, used port manipulation to
save memory.
  delay(50);
  PORTD &= ~(1 << PD3);
  delay(20);
  PORTD |= (1 << PD3);
  delay(50);
  PORTD &= ~(1 << PD3);
  PORTD |= (1 << PD5);
  delay(2000);
  PORTD &= ~(1 << PD5);
}
```

```
void declinedCard(){ //triggers the buzzer with a warning sound if card ID is not valid.
/*
  lcd.clear();
  lcd.setCursor(4,0);
  lcd.print("YETKISIZ");
  lcd.setCursor(6,1);
  lcd.print("KART");
*/
  PORTD |= (1 << PD3);
  delay(600);
  PORTD &= ~(1 << PD3);
  delay(100);
  PORTD |= (1 << PD3);
  delay(600);
  PORTD &= ~(1 << PD3);
  delay(100);
  PORTD |= (1 << PD3);
  delay(600);
  PORTD &= ~(1 << PD3);
}
```

b.2) ValidCards.h

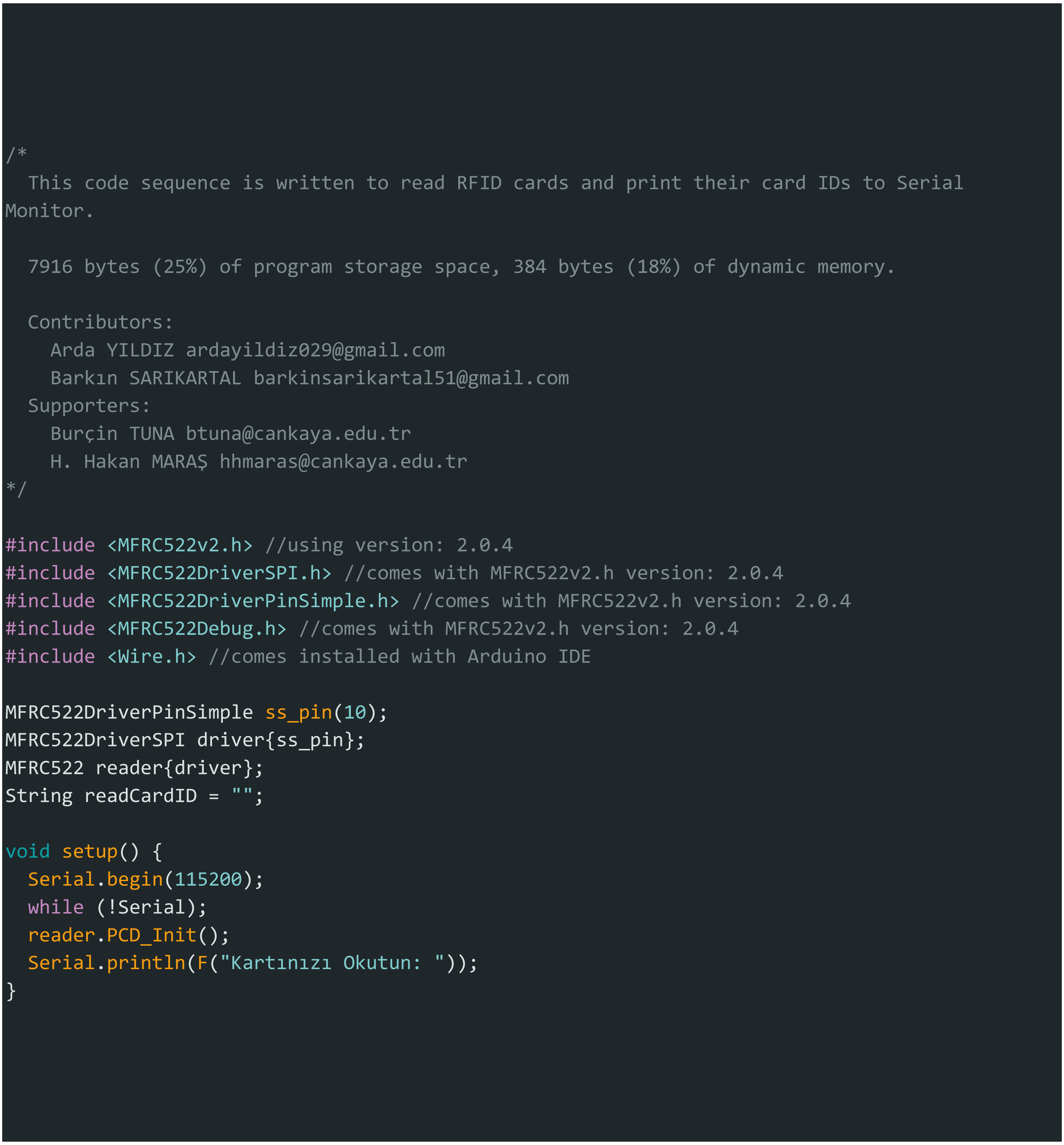
```
#ifndef VALIDCARDS_H
#define VALIDCARDS_H

#define NUM_VALID_CARD_IDS 5 //replace this number with your valid RFID card number.

const String validCardIDs[NUM_VALID_CARD_IDS] = { //replace these RFID card IDs with your valid RFID card IDs with a space at the beginning.
  " FA 31 3C F9",
  " A3 15 4A E3",
  " CB 48 C9 03",
  " 0B 75 C3 03",
  " 6B 3C B5 03"
};

#endif
```

To be able to upload this software to Arduino Nano, UID_Okuma.ino and ValidCards.h files must be in the same directory. Furthermore, the array in ValidCards.h file should be updated with the IDs that are going to be used in the system and in additionally, NUM_VALID_CARD_IDS variable’s value must be equal to number of card IDs added to the array.

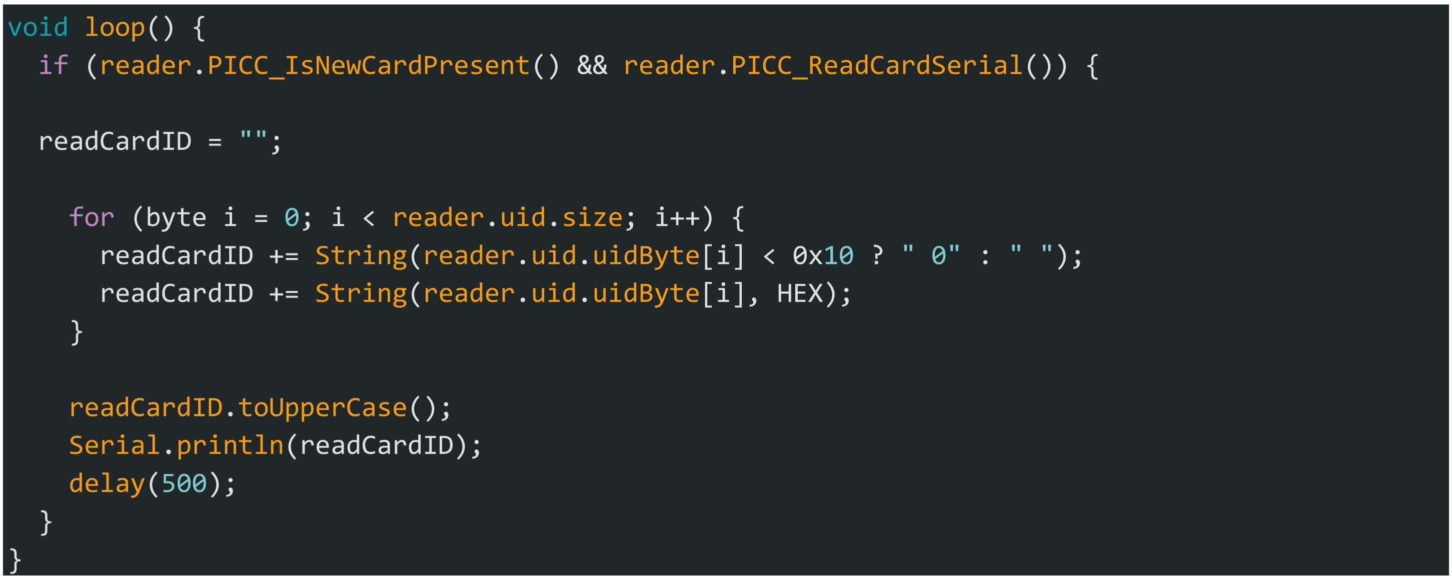
5. Installation and Connections

First of all, if card IDs which are going to be added to ValidCards.h file are not known, the code below can be executed with the same circuit diagram and card IDs can be read from Serial Monitor.

PrintCardID.ino

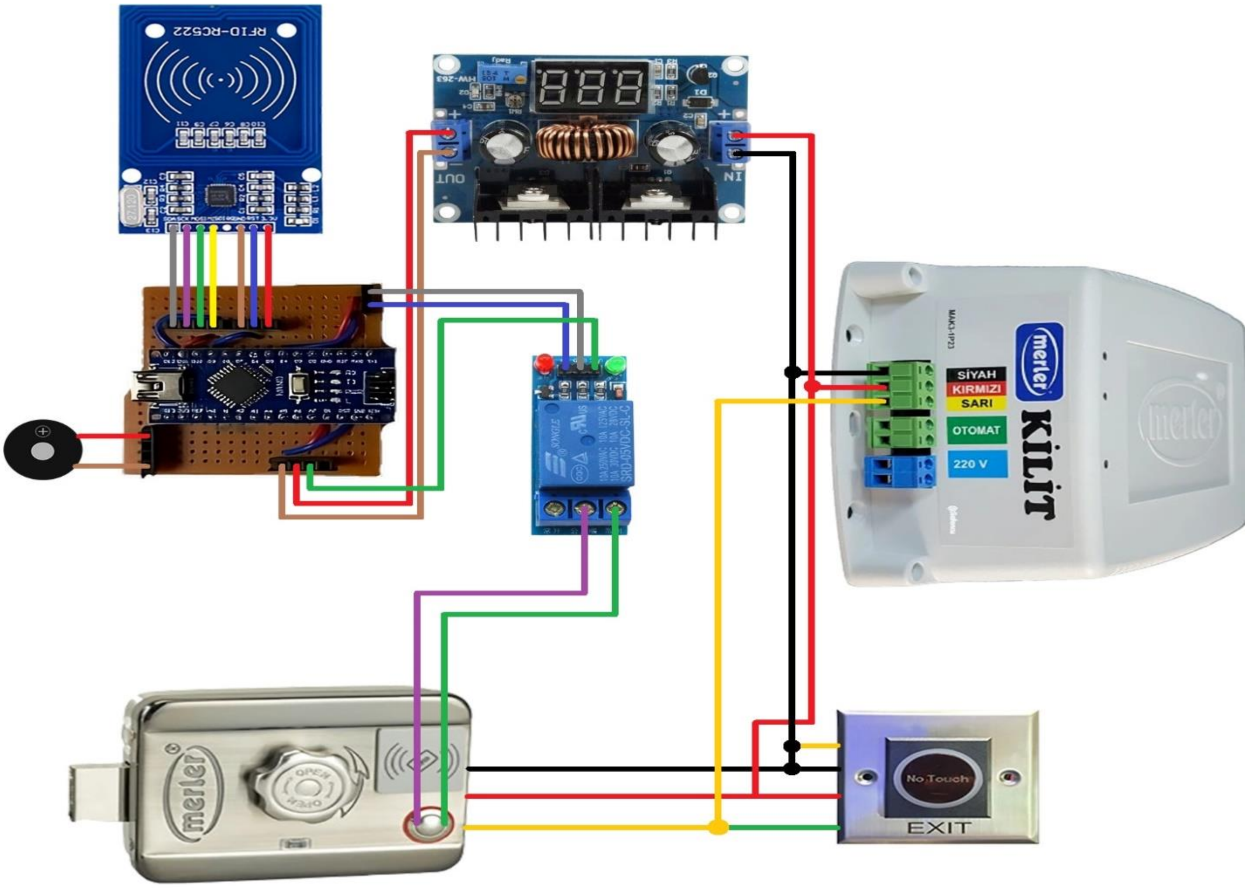


```
/*
  This code sequence is written to read RFID cards and print their card IDs to Serial
Monitor.

  7916 bytes (25%) of program storage space, 384 bytes (18%) of dynamic memory.

  Contributors:
    Arda YILDIZ ardayildiz029@gmail.com
    Barkın SARIKARTAL barkinsarikartal51@gmail.com
  Supporters:
    Burçin TUNA btuna@cankaya.edu.tr
    H. Hakan MARAŞ hhmaras@cankaya.edu.tr
*/

#include <MFRC522v2.h> //using version: 2.0.4
#include <MFRC522DriverSPI.h> //comes with MFRC522v2.h version: 2.0.4
#include <MFRC522DriverPinSimple.h> //comes with MFRC522v2.h version: 2.0.4
#include <MFRC522Debug.h> //comes with MFRC522v2.h version: 2.0.4
#include <Wire.h> //comes installed with Arduino IDE

MFRC522DriverPinSimple ss_pin(10);
MFRC522DriverSPI driver{ss_pin};
MFRC522 reader{driver};
String readCardID = "";

void setup() {
  Serial.begin(115200);
  while (!Serial);
  reader.PCD_Init();
  Serial.println(F("Kartınızı Okutun: "));
}
```



```
void loop() {
  if (reader.PICC_IsNewCardPresent() && reader.PICC_ReadCardSerial()) {

  readCardID = "";

    for (byte i = 0; i < reader.uid.size; i++) {
      readCardID += String(reader.uid.uidByte[i] < 0x10 ? " 0" : " ");
      readCardID += String(reader.uid.uidByte[i], HEX);
    }

    readCardID.toUpperCase();
    Serial.println(readCardID);
    delay(500);
  }
}
```

In this step, after learning card IDs and editing ValidCards.h file, the system can be made ready for use by making the following connections below.



6. Reviews and Recommendations

The project is successful for creating a basic door lock system with RFID as a prototype. However, more advanced features can be added in terms of security and usability. For example, adding or deleting valid card IDs from an admin panel over the internet and logging the entry and exit times in a file can be added. Additionally, the system can integrate various communication protocols and integration options, expanding its field of use. This flexibility provides a significant advantage in meeting diverse needs across industries.

7. Conclusion

This system performs the necessary operations for verification and access control to Çankaya University Technology Transfer Office Incubation Center by processing the read card information using Arduino Nano. This simple combination provides a framework for complex systems and can be expanded or customized as needed.

As a conclusion, this project creates a powerful but straightforward framework that leads to more complex and customized card access system implementations.