

ANKAYA NİVERSİTESİ TEKNOLOJİ TRANSFER OFİSİ RFID KARTLI GEÇİŞ KİLİT SİSTEMİ



Proje Raporu (TR)

Geliřtirenler:

Arda YILDIZ - ardayildiz029@gmail.com

Barkın SARIKARTAL - barkinsarikartal51@gmail.com

Destekleyenler:

Burin TUNA - btuna@cankaya.edu.tr

H. Hakan MARAŞ - hhmaras@cankaya.edu.tr

19/03/2024

İçerik:

1. Giriş.....	3
2. Proje Tanımı.....	3
3. Kullanılan Malzeme Listesi.....	3
a) Devrede Kullanılan Malzemeler.....	3
b) Kilit İçin Kullanılan Malzemeler	3
4. Donanım ve Yazılım.....	4
a) Devre Şeması	4
b) Sistem Yazılımı.....	6
5. Kurulum ve Bağlantılar.....	10
6. Değerlendirme ve Öneriler.....	12
7. Sonuç.....	12

1. Giriş

Bu rapor Çankaya Üniversitesi Teknoloji Transfer Ofisi Kuluçka Merkezi'nin kapısında kullanılmak üzere geliştirilmiş bir kartlı geçiş kilit sistemini detaylarıyla anlatır. Bu sistem, tanımlı kart sahiplerinin bahsi geçen alana erişmelerini sağlar.

2. Proje Tanımı

Bu projenin amacı, basit bir kartlı geçiş sistemi oluşturmaktır. RFID kart okuyucu, uzatılan kartın kart numarasını okur ve bu kart numarasını Arduino Nano'ya gönderir. Arduino Nano ise bu kart numarasını hafızasında bulunan kart numaraları ile karşılaştırır ve tanımlı bir kart olup olmadığına karar verir. Eğer okutulan kart, tanımlı bir kart ise röleyi tetikler ve kilidin açılmasını sağlar. Aksi hâlde, okutulan kart, tanımlı bir kart değil ise buzzer ile uyarı sesi çıkarır. Tanımlı kartlara sahip kişiler, bahsi geçen alana girdikten sonra temassız butona ellerini yaklaştırarak kilidi açabilirler.

3. Kullanılan Malzeme Listesi

a) Devre İçin Kullanılan Malzemeler

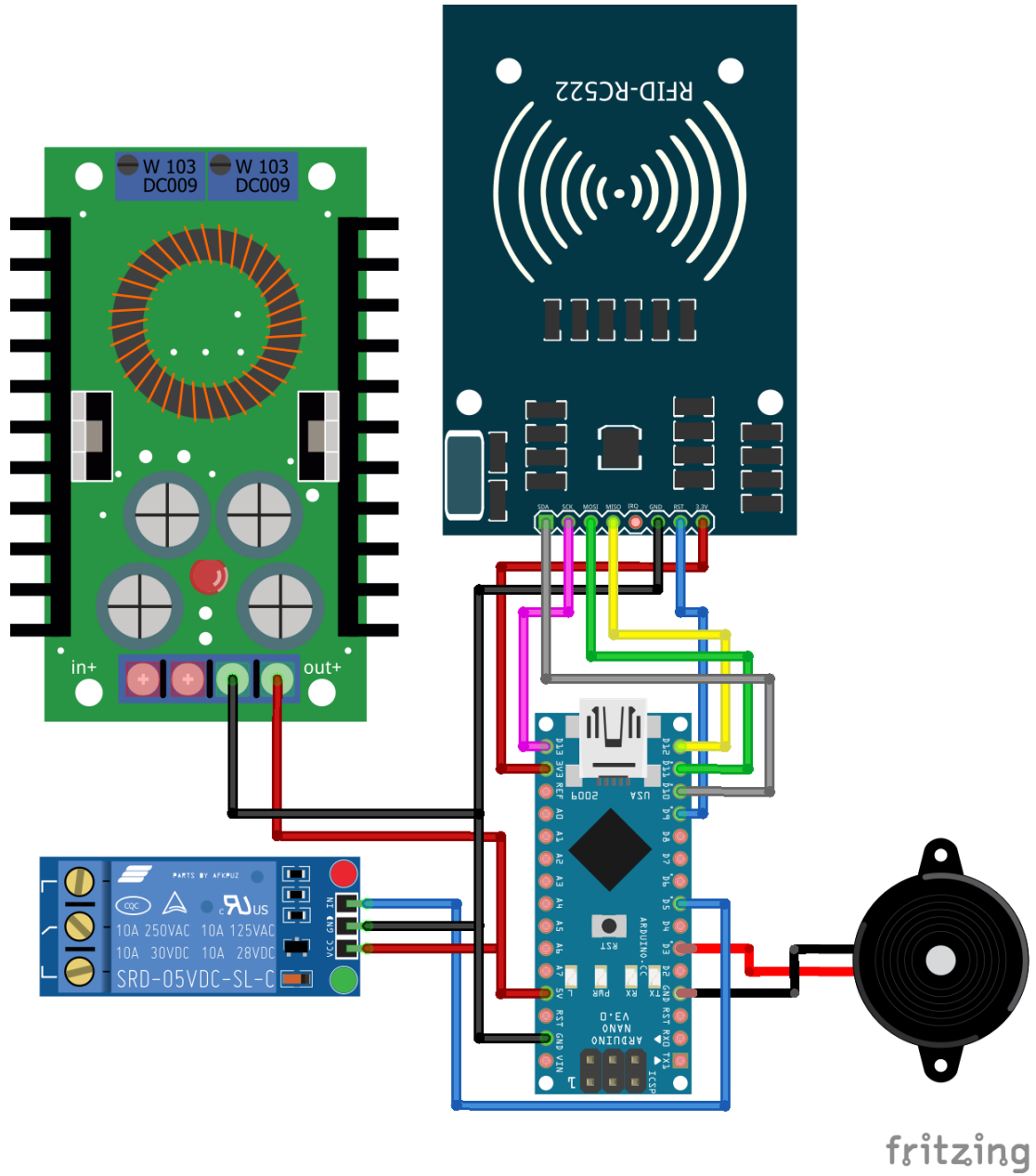
- Arduino Nano (1 adet)
- RC522 RFID Modülü (1 adet)
- 5V Tek Kanal Röle Kartı (1 adet)
- XL4046 DC-DC Step Düşürücü Voltaj Regülatörü (1 adet)
- 5V Aktif Buzzer (1 adet)
- 5x5 cm Delikli Pertinaks (1 adet)
- Altınkaya SE-210 IP-67 Contalı Kutu (1 adet)
- Jumper Kablolar
- Dişi Header'lar

b) Kilit İçin Kullanılan Malzemeler

- Merter MAK3 Akıllı Kilit
- Merter 12W Akülü Güç Kaynağı
- Merter MB02 Temassız Buton

4. Donanım ve Yazılım

a) Devre Şeması



Açıklama:

RC522 -> Arduino Nano:

- SDA -> D10
- SCK -> D13
- MOSI -> D11
- MISO -> D12
- IRQ -> Herhangi Bir Bağlantı Yapılmıyor
- GND -> GND
- RST -> D9
- 3.3V -> 3.3V

5V Tek Kanal Röle Kartı -> Arduino Nano:

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

5V Aktif Buzzer -> Arduino Nano:

- IN -> D3
- GND -> GND

XL4016 DC-DC Step Düşürücü Voltaj Regülatörü -> Arduino Nano:

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

b) Sistem Yazılımı

Sistemde kullanılan tüm kodlara <https://github.com/barkinsarikartal/DoorLockSystemWithRFID> linkinden ulaşılabilir.

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("ACILİYOR");
*/
    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

```

Sisteme kodları yükleyebilmeniz için UID_Okuma.ino ve ValidCards.h dosyalarının aynı klasör içerisinde bulunmaları gerekmektedir. Ayrıca, ValidCards.h dosyasında bulunan validCardIDs dizisi içerisindeki kart numaralarının sistemde kullanılacağı tanımlı kart numaralarıyla değiştirilmesi ve ek olarak NUM_VALID_CARD_IDS değişkeninin değerinin diziye eklenen kart sayısı ile değiştirilmesi gerekmektedir.

5. Kurulum ve Bağlantılar

Öncelikle, ValidCards.h dosyasına ekleyeceğimiz tanımlı kartların kart numaraları bilinmiyorsa, öğrenmek için aşağıdaki kod aynı devre şemasını kullanarak çalıştırılabilir ve Seri Monitör'den kart numaraları öğrenilebilir.

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()) {

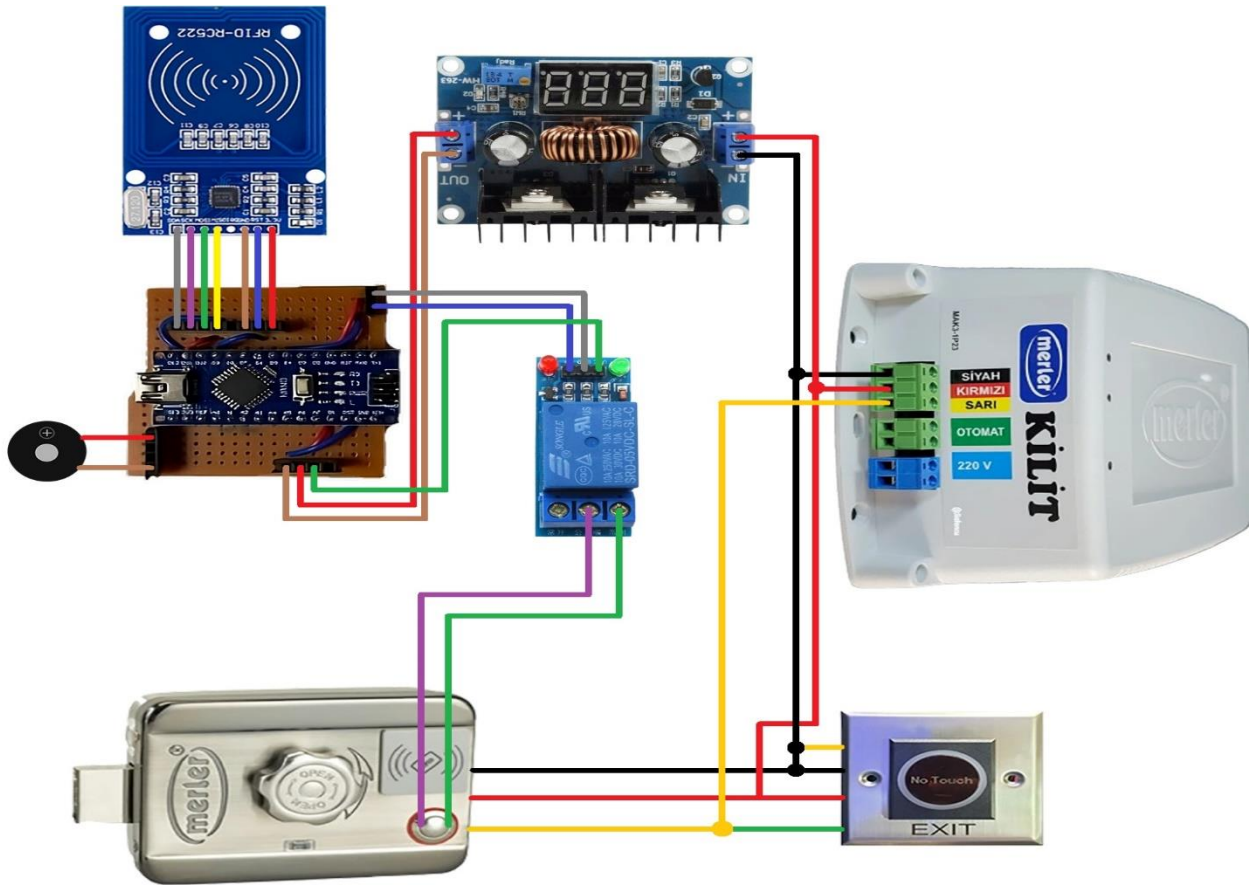
    readCardID = "";

    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);
  }
}

```

Bu adımla kart numaralarını öğrenip ValidCards.h dosyasını ihtiyaçlara göre düzelttikten sonra aşağıdaki bağlantılar yapılarak sistem kullanıma hazır hâle getirilebilir.



6. Değerlendirme ve Öneriler

Proje, basit bir kartlı geçiş sistemi oluşturmak için başlangıç seviyesi bir prototip olarak başarılıdır. Ancak, güvenlik ve kullanılabilirlik açısından daha gelişmiş özellikler eklenebilir. Örneğin, internet üzerinden bir admin panelinden tanımlı kartların eklenmesi veya silinmesi ve giriş çıkış zamanlarının bir yerde kayıt altında tutulması özellikleri eklenebilir. Ayrıca, farklı iletişim protokolleri veya entegrasyon seçenekleri de sisteme entegre edilebilir. Böylece kullanım alanı daha da genişletilebilir. Bu esneklik, çeşitli endüstrilerdeki farklı ihtiyaçları karşılamak için önemli bir avantaj sağlar.

7. Sonuç

Bu sistem, Arduino Nano kullanarak, okunan kart bilgilerini işleyerek doğrulama ve Çankaya Üniversitesi Teknoloji Transfer Ofisi Kuluçka Merkezi'ne erişim kontrolü için gerekli işlemleri gerçekleştirir. Bu temel kombinasyon, daha karmaşık sistemler için bir temel sunar ve ihtiyaçlar doğrultusunda genişletilebilir veya özelleştirilebilir. Sonuç olarak, bu proje, basit ancak güçlü bir temel oluşturarak kartlı geçiş sistemlerinin daha karmaşık ve özelleştirilmiş uygulamalarına kapı açmaktadır.