

Projekt Elektronika dla informatyków

System drzwi sterowany modułem Raspberry Pi

Arkadiusz Cios – 177055 - EFDI

Wstęp

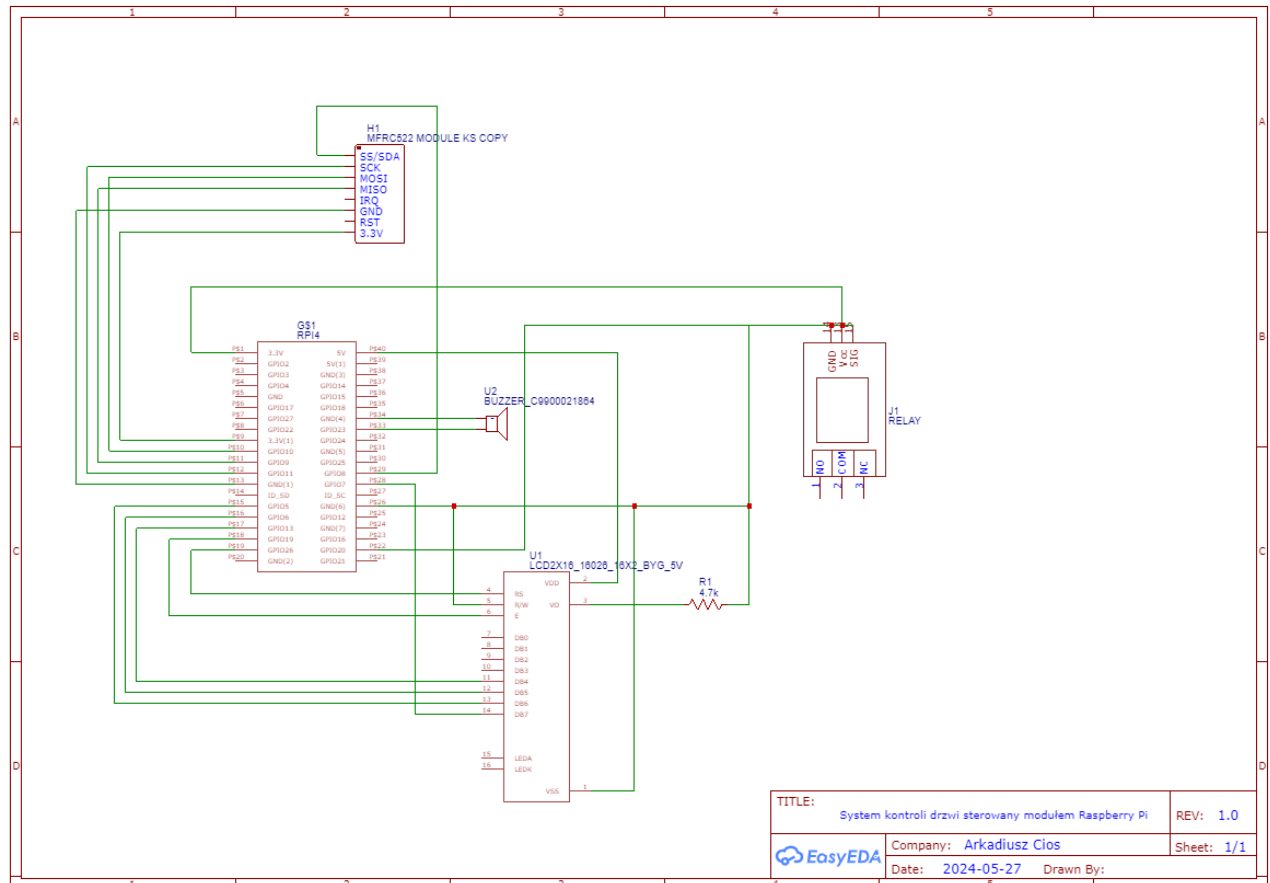
Istnieją sytuacje, w których zachodzi potrzeba ograniczenia dostępu do jakiegoś miejsca tylko dla uprawnionych osób. Problem ten towarzyszy ludzkości od bardzo dawna, już 4 tysiące lat temu starożytni Grecy wymyślili drewniane urządzenie służące do zamykania drzwi, odnalezione przez archeologów w pałacu w Chorsabad koło Niniwy. Jednak z czasem rozwój technologiczny postępował, w konsekwencji czego dzisiaj jesteśmy w stanie zaprojektować bardziej nowoczesne rozwiązania, służące temu celu, dające dodatkowe możliwości. Przedmiotem projektu jest automatyczny system kontroli drzwi wejściowych sterowany modułem raspberry pi. Wyobraźmy sobie przykładową sytuację, w której mamy sale operacyjną w szpitalu. Chcemy pozwolić wejść do niej tylko członkom upoważnionego personelu, przydatna by była również funkcja logowania wejść, aby wiedzieć kto i kiedy skorzystał z drzwi. Wymagane jest również, aby proces otwierania zamka był szybki i prosty, co umożliwiają karty RFID, potrafiące przechować informacje wielkości nawet jednego kilobajta i po przyłożeniu udostępnić systemowi informację o osobie próbującej otworzyć zamek.

Założenia projektowe

Projekt opiera się o Raspberry Pi, które wykorzystując moduł RFID RC522, zbiera dane z przyłożonych do modułu kart RFID, następnie wysyła zebrane dane do systemu API, który łączy się z bazą danych, zbierając dane o upoważnionych użytkownikach finalnie zwracając do urządzenia informację czy pozwolić użytkownikowi na otwarcie zamka. System w międzyczasie zapisuje dane o skorzystaniu z zamka w bazie danych, pozwalając na późniejszą analizę ruchu w drzwiach. W założeniach projektu korzystamy z drzwi sterowanych elektromagnesem, który zamyka oraz otwiera zamek, dlatego do modułu podłączony jest również przekaźnik, który w rzeczywistej sytuacji sterowałby elektromagnesem. Jest to jednak system, który przewiduje prostą modyfikację na np.

Serwomechanizm dla drzwi otwieranych w inny sposób. Za komunikację systemu z użytkownikiem odpowiada ekran LCD oraz buzzer sygnalizujący otwarcie zamka.

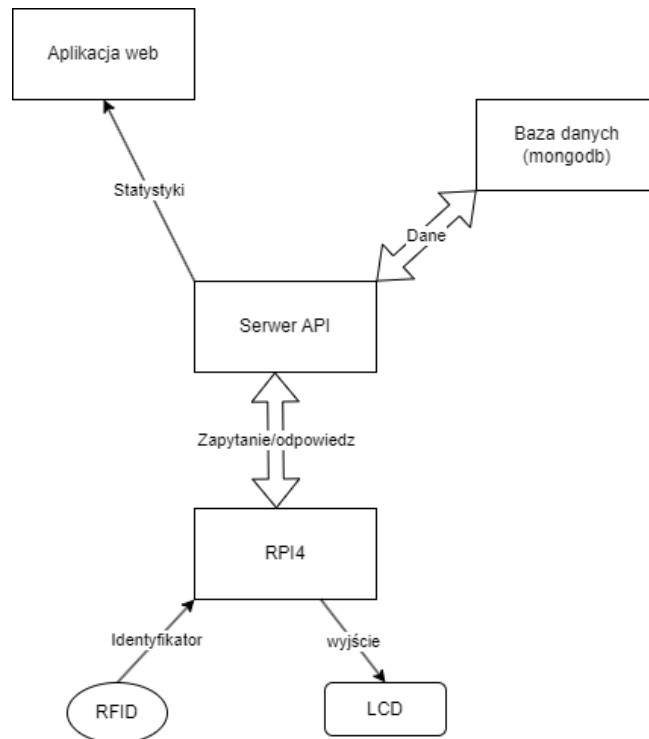
Schemat



Architektura systemu

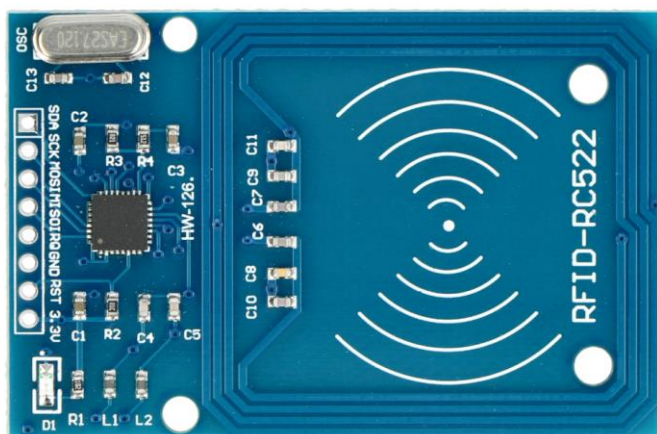
Sercem projektu jest Raspberry Pi, które obsługuje moduł RFID, wysyła dane zapisane na przyłożonej karcie na serwer, który wyszukuje w bazie danych informacji o zapisanym na karcie użytkowniku. Jeśli karta jest zapisana w bazie, system wysyła informację do RPi, że można otworzyć drzwi oraz wyświetla na ekranie LCD imię i nazwisko zautoryzowanego użytkownika. Dodatkowo prosta aplikacja web napisana w next.js pozwala na pobranie informacji z serwera i wyświetla je w sposób bardziej przystępny dla zwykłego użytkownika. Schemat działania widoczny jest na diagramie poniżej.

Diagram



Hardware

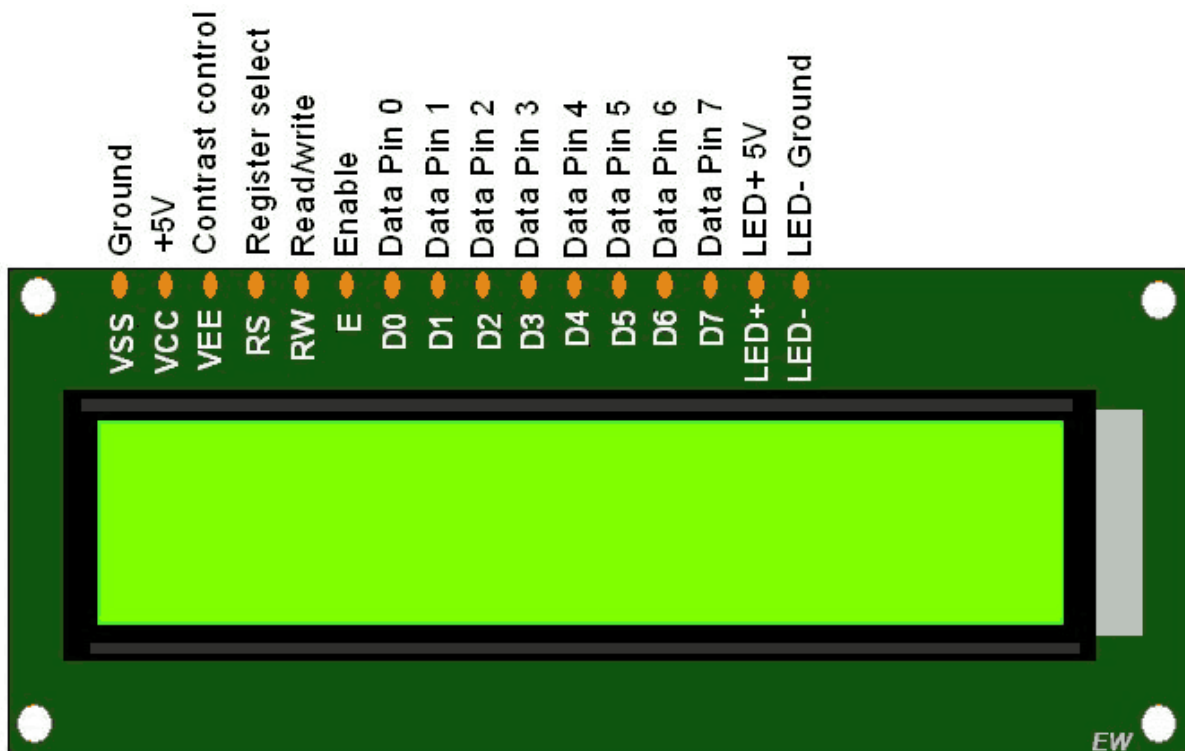
Moduł RFID MFRC522



Płytką umożliwia odczyt i zapis danych z urządzeń RFID na częstotliwości 13,56 MHz. Moduł oparty na układzie RC522, zasilany jest napięciem 3,3 V.

RFID (*radio-frequency identification*) jest technologią, która wykorzystuje fale radiowe do przesyłania danych. Działanie systemu jest następujące: czytnik wytwarza falę elektromagnetyczną, która jest odbierana przez antenę, a następnie filtrowana i dekodowana, aby odczytać odpowiedzi znaczników. Moduł komunikuje się przez SPI (*Serial Peripheral Interface*), który jest jednym z najczęściej używanych interfejsów komunikacyjnych pomiędzy systemami mikroprocesorowymi a układami peryferyjnymi, który jest również wspierany przez RPi.

Wyświetlacz LCD 16x2



Wyświetlacz LCD 16x2 to popularny moduł wyświetlacza, który może wyświetlać 16 znaków w 2 wierszach. Jest łatwy w obsłudze dzięki sterownikowi HD44780, a także szeroko stosowany w projektach edukacyjnych i przemysłowych.

Pin#	Name	Direction	Description
1	Vss	Power	Power Supply Ground
2	Vcc	Power	Power Supply (+3.3 or +5 V depending on module)
3	Vee	Input	Contrast Adjustment (analog input)
4	RS	Input	Register Select (0 = command, 1 = data)
5	R/W	Input	Read/Write (0 = write to display module, 1 = read from display module)
6	E	Input	Clock Enable (falling-edge triggered)
7	DB0	I/O	Data Bit 0 (not used in 4-bit operation)
8	DB1	I/O	Data Bit 1 (not used in 4-bit operation)
9	DB2	I/O	Data Bit 2 (not used in 4-bit operation)
10	DB3	I/O	Data Bit 3 (not used in 4-bit operation)
11	DB4	I/O	Data Bit 4
12	DB5	I/O	Data Bit 5
13	DB6	I/O	Data Bit 6
14	DB7	I/O	Data Bit 7
15	LED+	Power	Backlight Anode (+) (if applicable)
16	LED-	Power	Backlight Cathode (-) (if applicable)

Pin 3 pozwala na dostosowanie kontrastu ekranu do swoich potrzeb, dlatego w układzie pomiędzy pinem i masą zamontowany został rezystor 4,7K, który ustawia kontrast na czytelny.

Moduł przekaźnika



Przełącznik jest komponentem działającym na zasadzie elektromagnesu: prąd płynący w zwojnicy przełącznika wywołuje pole magnetyczne, które przyciąga stalową zworę (zwaną też kotwicą) zamykającą lub otwierającą odpowiedni styk lub ich zestaw.

Urządzenie pozwala na kontrole przepływu prądu za pomocą wejścia (stan wysoki 5V lub niski).

Buzzer

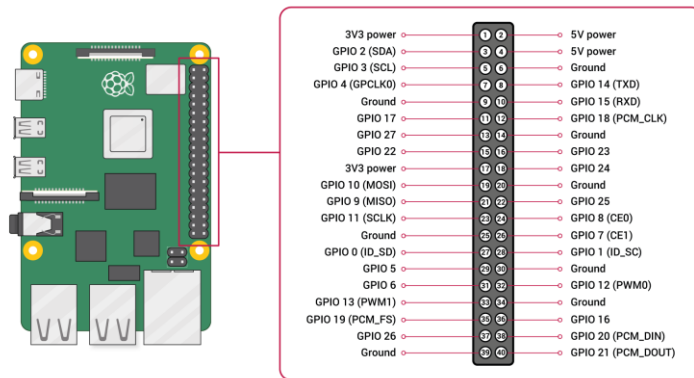


Komponent jest malutkim głośniczkiem, pozwalającym na generowanie sygnałów dźwiękowych. W projekcie generuje dźwięk sygnalizujący otwarcie drzwi, który trwa przez cały okres ich otwarcia. Wyzwalany jest przez sygnał PWM generowany przez oprogramowanie.

Raspberry Pi 4 model B



Raspberry Pi 4 Model B to minikomputer z czterordzeniowym procesorem ARM Cortex-A72 1.5 GHz, dostępny w wersjach z 2GB, 4GB, lub 8GB RAM. Ma dwa porty micro-HDMI, cztery porty USB (dwa 3.0, dwa 2.0), gigabitowy Ethernet, Wi-Fi 802.11ac, Bluetooth 5.0, i 40-pinowe GPIO.



GPIO (General Purpose Input/Output) to uniwersalne piny w mikrokontrolerach i minikomputerach, takich jak Raspberry Pi, które mogą być programowane do odczytu i wysyłania sygnałów cyfrowych. Te piny umożliwiają interakcję z różnymi urządzeniami peryferyjnymi.

Oprogramowanie

System obsługiwany jest przez oprogramowanie sterujące i wyświetlające dane.

Podzielone jest na trzy części:

- A) Kod Python na RPi, obsługujący komponenty układu i komunikujący się z systemem w chmurze
- B) Serwer z interfejsem API, który jest odpowiedzialny za autoryzację dostępu do otwarcia drzwi dla poszczególnych użytkowników, zapisywanie oraz pobieranie danych z bazy danych (MongoDB).
- C) Prosta aplikacja web, pozwalająca na przegląd statystyk, historii autoryzacji, w sposób przystępny dla użytkownika.

Kod Raspberry Pi

Kod na RPi odpowiada za obsługę podłączonych elementów oraz komunikację z serwerem.

```

import rfid
import requests
import time
import os
import RPi.GPIO as GPIO
from RPLCD.gpio import CharLCD
import buzzer
import relay

API_KEY = os.environ['API_KEY']
BASE_URL = "https://rfid-door-api-db3514e2b4df.herokuapp.com/"

lcd = CharLCD(pin_rs=37, pin_e=35, pins_data=[33, 31, 29, 36],
              numbering_mode=GPIO.BOARD,
              cols=16, rows=2, dotsize=8,
              charmap='A02',
              auto_linebreaks=True)

def logEntrance(id):
    json = {
        "userId": id,
        "API_KEY": API_KEY
    }

    res = requests.post(BASE_URL+'door/create', json=json, headers={"Content-Type": "application/json"})

    if res.status_code == 200:
        print("entrance has been logged into the system")
        print(res.json())

while(True):
    id = rfid.read()
    time.sleep(1)

    res = requests.get(BASE_URL + 'user/' + id)
    if res.text == "Not found":
        print(f"user with given id of {id} doesn't exist")
        continue
    user = res.json()
    username = user['name']
    print("logged in as " + username)
    lcd.write_string(username)
    relay.high()
    buzzer.beep()
    relay.low()
    logEntrance(int(id))
    time.sleep(3)
    lcd.clear()

```

Głównym skryptem na urządzeniu jest doorController.py, który steruje elementami podłączonymi do RPi, za pomocą innych napisanych skryptów oraz pobiera z serwera informacje o autoryzacji użytkowników i loguje udane próby otwarcia zamka.

 arekminajj	Add relay and buzzer controllers	7e7444c · 5 minutes ago	 2 Commits
 .gitignore	init	3 weeks ago	
 buzzer.py	Add relay and buzzer controllers	5 minutes ago	
 doorController.py	Add relay and buzzer controllers	5 minutes ago	
 lcd.py	init	3 weeks ago	
 read.py	init	3 weeks ago	
 relay.py	Add relay and buzzer controllers	5 minutes ago	
 rfid.py	init	3 weeks ago	
 write.py	init	3 weeks ago	

Większość modułów podłączonych do komputera mają własne skrypty sterujące. Cały kod źródłowy można znaleźć pod: <https://github.com/arekminajj/rfid-door-pi>

Logi aplikacji po przyłożeniu prawidłowej karty

```
place card on top of the reader module to read id  
card id: 334824260712  
written id: 1  
logged in as Arkadiusz Cios  
entrance has been logged into the system  
{'acknowledged': True, 'insertedId': '665cd6e5a6d5fe533cc44e74'}
```

Serwer

Jest to prosta aplikacja napisana w express.js, pozwalająca na odczyt i zapis w bazie danych, autoryzację dostępu oraz udostępniająca endpointy API z podstawowymi funkcjami:

1. GET /user/all: Pobiera wszystkich użytkowników.
2. GET /user/:id: Pobiera użytkownika po ID.
3. GET /door/all: Pobiera wszystkie logi otwarcia drzwi
4. GET /door/:userId: Pobiera logi dla określonego użytkownika.
5. POST /user/create: Tworzy nowego użytkownika.
6. POST /door/create: Rejestruje nowy log otwarcia drzwi.

Kod źródłowy: <https://github.com/arekminajj/rfid-door-api>

Aplikacja Web

Prosty system wyświetlania informacji o użytkownikach i logach otwierania przez nich drzwi. Jest wygodniejszym sposobem sprawdzenia tych informacji niż pobieranie surowych danych z serwera głównego lub bazy danych.

Przykładowa lista użytkowników

[Arkadiusz Cios](#)
[Jan Kowalski](#)
[Andrzej Kmicic](#)
[Babinicz](#)
[Jacek Soplica](#)
[Cezary Baryka](#)

Przykładowe zapisane dane

Imię i nazwisko: Arkadiusz Cios

Liczba wejść: 10

30 kwietnia 2024 12:07:05 CEST

30 kwietnia 2024 12:07:38 CEST

30 kwietnia 2024 15:27:30 CEST

30 kwietnia 2024 15:28:03 CEST

30 kwietnia 2024 15:28:22 CEST

30 kwietnia 2024 15:29:03 CEST

30 kwietnia 2024 15:29:23 CEST

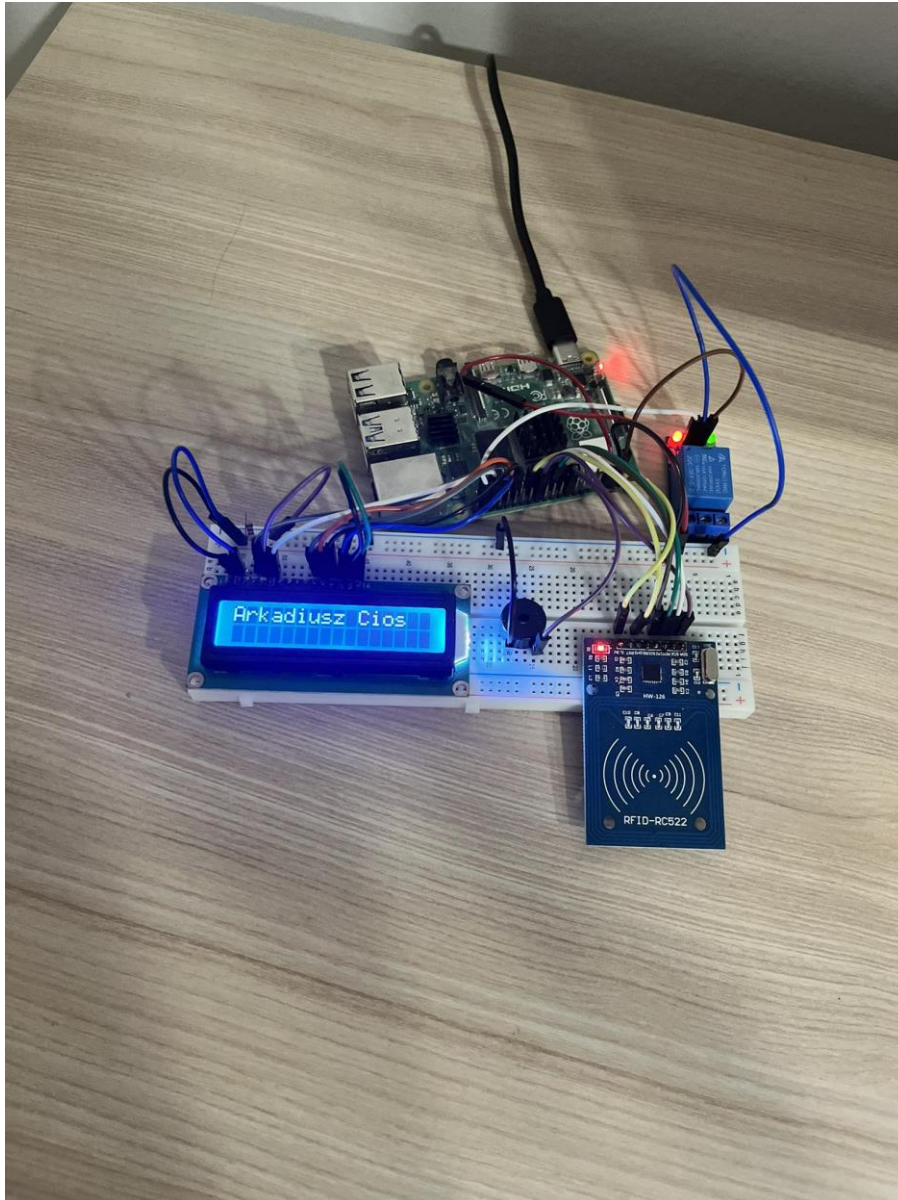
30 kwietnia 2024 15:29:38 CEST

1 maja 2024 15:33:26 CEST

9 maja 2024 11:53:16 CEST

Kod źródłowy: <https://github.com/arekminajj/rfid-door-next>

Działanie



Po przyłożeniu prawidłowej karty przypisanej do użytkownika przekaźnik został otwarty na dwie sekundy, buzzer generował dźwięk sygnalizujący, a na ekranie LCD pojawiło się imię i nazwisko przypisane do karty.

```
place card on top of the reader module to read id
card id: 675912008575
written id: 7
user with given id of 7                                doesnt exit
```

W przypadku przyłożenia nieautoryzowanej karty, system zgłosi błąd oraz nie otworzy drzwi.

Działanie projektu w wersji wideo można zobaczyć pod:

<https://www.youtube.com/shorts/ZsL3zmL47eY>

Wszystkie kody źródłowe, materiały i inne pliki są również dołączone w pliku ZIP.