

# Fejlesztői dokumentáció

## Témaválasztás

Projektünknek témájaként azért egy RFID beléptetőrendszer elkészítését választottuk, ugyanis számunka a Mikroelektromechanikai rendszerek tantárgyig ismeretlen volt a Raspberry Pi világa, így érdekesnek és lehetőségekkel telinek találtuk egy beléptetőrendszer fejlesztését.

## Tervezési fázis

A tervezés első lépéseként összegyűjtöttük és beszereztük azokat az eszközöket, amelyeket a projekt során használtunk. A folyamat során több eszközt is megvizsgáltunk és kiválasztottuk az egymással kompatibiliseket és a számunkra legmegfelelőbbeket.

Ezután megvizsgáltuk a projekt megvalósításához szükséges elérhető internetes forrásokat, útmutatókat. A programkód megírásához a Python nyelvet választottuk, különböző könyvtárak segítségével minden, a feladathoz szükséges problémát meg tudtunk oldani.

Végezetül a megvalósításhoz szükséges kapcsolási rajzot, a rendszer működését, illetve a kód felépítésére fektettünk hangsúlyt.

## Hardver és Szoftver specifikációk

### Tápegység

A raspberry folyamatos működéséhez szükség volt egy tápra, melynek micro usb az egyik végpontja, ezzel zavartalanul tud futni a beléptetőrendszer.

### Kamera modul - Raspberry cam v1.2

Mivel a beléptetőrendszernek, a belépni nem jogosult felhasználókról egy képet szükséges készítenie, ezért elengedhetetlen egy kamera modul használata. A választás a Raspberry cam v.1.2-re esett. Ennek a beüzemelését egy későbbi fejezetben részletezem.

### Vezeték nélküli hangszóró

Mivel nincs szükség nagy teljesítményű hangszóróra azért az olcsó és könnyen hordozható JBL GO-ra esett a választás.

## Tok

Az alapgép vásárlásakor járt hozzá egy tok is, azonban ehelyett egy másikat kellett beszereznünk, amit később a "felmerülő problémák" címszó alatt kifejtünk.

## Mini HDMI to HDMI átalakító

A Raspberry Pi Zero W-n egy Mini HDMI port található, így a monitorhoz való csatlakoztatás végett szükségünk volt egy átalakítóra.

## USB HUB - Gembird UHB-OTG-02 OTG 3 portos USB2.0 HUB

A fejlesztés és a bemutatás megkönnyítése végett egy USB hub használata mellett döntöttünk, így egyszerre használható egy egér és egy billentyűzet is.

## Freemail fiók létrehozása és beállítása

Mivel a Gmail megszüntette eddigi lehetőségét, a harmadik féltől származó alkalmazások már nem használhatják email fiókunkat biztonsági okokból. Ezen probléma elhárítására a freemail.hu levelezőszolgáltatót választottuk, amely számunkra kellő biztonságú és kifejezetten gyors. A regisztráció után a beállításokban itt is engedélyezni kellett a külső kliensek (esetünkben a Raspberryn futó Python script) számára való hozzáférést, majd mentés után már élesítve is volt az email szolgáltatásunk.

Kliensek engedélyezése Engedélyezve

Amennyiben használ sz. pl. Outlook, Thunderbird, Apple Mail vagy Android Mail alkalmazásokat, engedélyezned kell az IMAP/POP3 elérést



## Raspberry Pi OS

Maga az operációs rendszer telepítése egyszerű, csupán egy SD kártya olvasására alkalmas laptop szükséges hozzá. Az OS letölthető ingyenesen a <https://www.raspberrypi.com/software/operating-systems/> oldalról, a telepítést pedig az alábbi guide szerint végeztük: <https://namerre.hu/raspberry-pi-os-operacios-rendszer-telepitese-a-raspberry-pi-mini-pc-re/>

## Hangfileok generálása

A hangfájlokat, amik egy Bluetooth hangszórón keresztül játsszuk le, amelyek visszajelzésként funkcionálnak.

<https://play.ht/text-to-speech-voices/hungarian/> → Ezen weboldal segítségével generáltuk a hangfájlokat, ez egyfajta OpenAi projekt, vagy ahhoz hasonló kezdeményezés. A hangfájlokat pedig a Camtasia Studio 9 videóvágó programban véglegesítettük.

- Amennyiben a felhasználó RFID-ja még nem szerepel az adatbázisban, úgy a(z) „uj\_rfid\_regx.wav” hangfájl kerül lejátszásra

- A regisztrációs folyamat végeztével a(z) „reg\_keszx.wav” hangfájl kerül lejátszásra.
- Ha a felhasználó RFID-ja már szerepel az adatbázisban, úgy a(z) „belepes\_engedelyezvex.wav” hangfájl kerül lejátszásra.

## Felmerülő problémák

1. Chip hiány végett nehezen találtunk egy megfelelő specifikációkkal bíró és árkategóriában lévő raspberry-t.
2. Az eredeti tokba a kamerát nem tudtuk volna beépíteni, így egy másik, direkt erre a célra lévő tokot választottunk.
3. A Mini HDMI porthoz, hogy tudjunk csatlakozni a monitorunkhoz, szükségünk volt egy Mini HDMI to HDMI átalakítóra. Továbbá az átalakítónk és az új tokunk nem volt kompatibilis, így abból le kellett faragnunk, hogy működés közben többé ne csúszhasson ki.
4. Az egy USB-C port kevésnek bizonyult, az egér, billentyűzet, pendrive csatlakoztatása végett szükségünk volt egy USB HUB-ra.
5. A beléptetőrendszert megpróbáltuk automatizálni, de ez akkora teljesítményvesztéssel járt, hogy elvetettük az ötletet, egy erősebb Raspberry PI-al lenne megoldható.

Ezt a megoldást próbáltuk ki az automatizálás elérése érdekében:

```
sudo crontab -e
@reboot python3 /home/pi/edkb-rfid/read.py
```

## Rendszerkövetelmények

Az email küldése miatt internetkapcsolat szükséges a beléptetőrendszer működéshez, illetve aktív bluetooth kapcsolat tartása szükséges a vezeték nélküli hangfallal.

## Összeszerelés/Áramköri rajzok

### Tok

A Raspberryt könnyen beillesztettük a tokba, csak bele kellett pattintani.

### Kamera

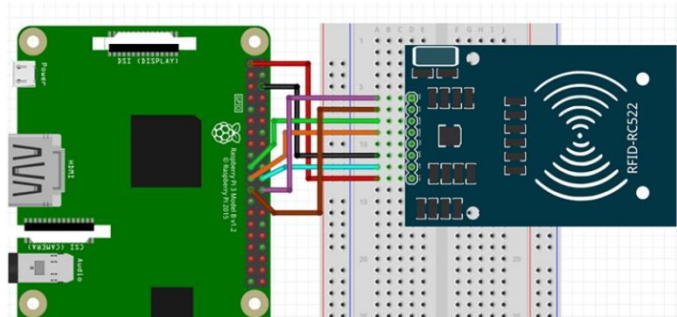
A CSI portba való illesztése, és a Raspberry bekapcsolása után a "Raspberry Pi Configuration utility" menüben engedélyezni kellett a kamera hozzáférést, mivel az alapértelmezetten letiltásra került, majd a konzol megnyitása után a `raspistill -o image.jpg` paranccsal használható a beszerelt kamera.

## RFID-RC522

Az RFID olvasót az összekötés után a kódban részletezett könyvtárral és parancsokkal, bármilyen beállítása nélkül használni lehetett.

A bekötés során az alábbi két rajzot vettünk figyelembe:

- **SDA** connects to **Pin 24**.
- **SCK** connects to **Pin 23**.
- **MOSI** connects to **Pin 19**.
- **MISO** connects to **Pin 21**.
- **GND** connects to **Pin 6**.
- **RST** connects to **Pin 22**.
- **3.3v** connects to **Pin 1**.



## Programkód részletezése

A beléptetőrendszer első feladata beolvasni az RFID kártyának az RFID-ját. Ehhez a z alábbi két könyvtár használata szükséges:

```
from mfrc522 import SimpleMFRC522
import RPi.GPIO as GPIO
```

A könyvtárak segítségével egy változóba könnyedén beolvasható az RFID, ami a szemléltetés kedvéért a konzolon jelenik meg az alább látható kódrészlet alapján

```
reader = SimpleMFRC522()

try:
    id = reader.read()
    print(id)
```

Ha az RFID beolvasásra került utána megvizsgáljuk, hogy az adatbázisban (rfid.csv) megtalálható-e a beolvasott RFID. Hogyha az adatbázisban szerepel az RFID akkor a belepes\_engedelyzve.wav hangfile kerül lejátszásra. Ha pedig nincsen akkor az adatbázis hozzáadja az adatbázishoz az RFID-t, küld egy emailt a beállított email címre az új felhasználó felviteléről, és készít egy képet a felhasználóról. Végül lejátssza a reg\_kesz.wav hangfilet. Az alábbi könyvtárak szükségesek a feladat megvalósításához:

```
import csv
import time
import smtplib
from subprocess import call
from email.mime.text import MIMEText
```

```
from email.mime.multipart import MIMEMultipart
from pydub import AudioSegment
from pydub.playback import play
```

A .csv file vizsgálatát egy függvénybe szerveztük ki az alábbi megoldással:

```
def number_in_file(number, filepath):
    with open(filepath, 'r') as f:
        for line in f:
            if str(number) in line:
                return True
    return False
```

Az elágazás, ami a két hangfájl lejátszásáért felel:

```
if number_in_file(number, filepath):
    print(f'{number} szerepel az adatbázisban: {filepath}')
    song=AudioSegment.from_wav('/home/pi/edkb-rfid/belepes_engedelyezvex.wav')
    play(song)
else:
    print(f'{number} nem szerepel az adatbázisban: {filepath}')
    song=AudioSegment.from_wav('/home/pi/edkb-rfid/uj_rfid_regx.wav')
    play(song)
```

A fénykép készítéséért felelős kódrészlet:

```
fenykep_nev = str(id) + '.jpg'
cmd = 'raspistill -t 500 -w 1024 -h 768 -o /home/pi/edkb-rfid/pics/' +
fenykep_nev
call ([cmd], shell=True)
print('Fenykep kesz,elmentve')
```

Az adatbázisban nem található RFID hozzáadása az adatbázishoz:

```
with open(filepath, 'a') as f:
    f.write(str(id)+"\n")
```

Az email küldéséért, felelős kódrészlet, itt minden szükséges maraméter megadása megtörténik, ami elengedhetetlen az email elküldése érdekében:

```
email_sender = 'edkbrfid@freemail.hu'
email_receiver = 'eredit.daniel@gmail.com'
```

```
subject = 'RfidAdmin'
msg = MIMEMultipart()
msg['From'] = email_sender
msg['To'] = email_receiver
msg['Subject']= subject
body = 'Egy uj rfid lett hozzaadva az adatbazishoz. Id: '+str(id)
msg.attach(MIMEText(body, 'plain'))
text = msg.as_string()
connection = smtplib.SMTP('smtp.freemail.hu', 587)
connection.starttls()
connection.login(email_sender, 'Mikroelektro2023')
connection.sendmail(email_sender, email_receiver, text )
connection.quit()
print('Email elkuldve')
```

Ezután a reg\_keszx.wav hangfile lejátszása történik:

```
song=AudioSegment.from_wav('/home/pi/edkb-rfid/reg_keszx.wav')
play(song)
```

Végül a GPIO könyvtár memória szemétét tüntetjük el:

```
finally:
    GPIO.cleanup()
```

## A read.py teljes kódja

```
#!/usr/bin/env python

import csv
from mfrc522 import SimpleMFRC522
import RPi.GPIO as GPIO
import time
import smtplib
from subprocess import call
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from pydub import AudioSegment
from pydub.playback import play

def number_in_file(number, filepath):
    with open(filepath, 'r') as f:
        for line in f:
            if str(number) in line:
                return True
    return False

reader = SimpleMFRC522()

try:
    id = reader.read()
    print(id)

    filepath = 'rfid.csv'
    number = id

    if number_in_file(number, filepath):
        print(f'{number} szerepel az adatbázisban: {filepath}')
        song=AudioSegment.from_wav('/home/pi/edkb-rfid/belepes_engedelyezvex.wav')
        play(song)
    else:
        print(f'{number} nem szerepel az adatbázisban: {filepath}')
        song=AudioSegment.from_wav('/home/pi/edkb-rfid/uj_rfid_regx.wav')
        play(song)
        fenykep_nev = str(id) + '.jpg'
        cmd = 'raspistill -t 500 -w 1024 -h 768 -o /home/pi/edkb-rfid/pics/' +
fenykep_nev
        call ([cmd], shell=True)
        print('Fenykep kesz,elmentve')
        with open(filepath, 'a') as f:
            f.write(str(id)+"\n")
        email_sender = 'edkbrfid@freemail.hu'
```

```
email_receiver = 'ereditcs.daniel@gmail.com'
subject = 'RfidAdmin'
msg = MIMEMultipart()
msg['From'] = email_sender
msg['To'] = email_receiver
msg['Subject'] = subject
body = 'Egy uj rfid lett hozzaadva az adatbazishoz. Id: '+str(id)
msg.attach(MIMEText(body, 'plain'))
text = msg.as_string()
connection = smtplib.SMTP('smtp.freemail.hu', 587)
connection.starttls()
connection.login(email_sender, 'Mikroelektro2023')
connection.sendmail(email_sender, email_receiver, text )
connection.quit()
print('Email elkuldve')
song=AudioSegment.from_wav('/home/pi/edkb-rfid/reg_keszx.wav')
play(song)
finally:
    GPIO.cleanup()
```

## Fényképek a kész projektről

