

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

---

KIERUNEK:                   Informatyka (INF)  
SPECJALNOŚĆ:               Inżynieria Internetowa (INT)

ZASTOSOWANIA SYSTEMÓW  
WBUDOWANYCH

Badanie czasu pracy

Work time guardian

AUTORZY:  
Mikołaj Jermakowicz, Kamil Machnicki

PROWADZĄCY PRACE:  
dr inż. Marek Woda

OCENA PRACY:



# Spis treści

|  |           |
|--|-----------|
| <b>1 Wstęp</b>   | <b>3</b>  |
| 1.1 Cel i zakres pracy . . . . .                         | 3         |
| 1.2 Funkcjonalności . . . . .                            | 3         |
| 1.3 Moduły . . . . .                                     | 4         |
| <b>2 Sterownik</b>                                       | <b>5</b>  |
| 2.1 Wykorzystany sprzęt . . . . .                        | 5         |
| 2.2 Podłączenie . . . . .                                | 7         |
| 2.3 Konfiguracja . . . . .                               | 9         |
| 2.4 Oprogramowanie . . . . .                             | 10        |
| <b>3 Aplikacja</b>                                       | <b>11</b> |
| 3.1 Program prezentacji danych . . . . .                 | 11        |
| 3.1.1 Podgląd obecnych pracowników . . . . .             | 12        |
| 3.1.2 Podgląd prepracowanego czasu pracowników . . . . . | 12        |
| 3.1.3 Dodawanie pracowników . . . . .                    | 13        |
| 3.2 Technologie . . . . .                                | 13        |
| 3.3 Przygotowanie środowiska . . . . .                   | 13        |
| 3.4 Baza Danych . . . . .                                | 13        |
| 3.4.1 Baza danych na maszynie wirtualnej . . . . .       | 14        |
| 3.4.2 Baza dnych na Raspberry Pi . . . . .               | 14        |
| <b>4 Podsumowanie</b>                                    | <b>15</b> |
| 4.1 Ocena krytyczna . . . . .                            | 15        |
| 4.2 Możliwości rozwoju . . . . .                         | 15        |
| <b>Bibliografia</b>                                      | <b>16</b> |
| <b>Spis rysunków</b>                                     | <b>18</b> |
| <b>Spis tabel</b>  | <b>20</b> |



# Rozdział 1

## Wstęp

Jako zaliczenie kursu Zastosowania Systemów Wbudowanych należało zrealizować wybrany przez grupę projekt, dotyczący układów wbudowanych. Pracę wykonać można było wykorzystując platformy Raspberry Pi, Arduino, bądź Beagleboard. Przez naszą grupę wybrany został temat *Badanie czasu pracy*, który zrealizowany został na płytce Raspberry Pi.

### 1.1 Cel i zakres pracy

Celem wybranego tematu było stworzenie projektu, który badałby godziny pracy pracowników danej placówki. Każdy pracownik przy wejściu oraz wyjściu odbijałby się przypisaną do siebie kartą, a system zliczałby godziny spędzone tego dnia i przedstawiał je w wygodnej formie dla administratora. Administrator mógłby dodawać pracowników oraz przypisać kartę do pracownika. W przypadku, gdy uprawniony użytkownik (taki, który znajduje się w bazie) odbije się w systemie, system poinformuje go o tym w sposób wizualny, a administrator będzie mógł zobaczyć w panelu dane pracownika i godzinę odbicia. W przypadku, gdy odbijana karta nie jest przypisana do żadnego pracownika, system w sposób wizualny zaalarmuje o tym.

### 1.2 Funkcjonalności

Lista funkcjonalności wymaganych do zrealizowania projektu:

- Rozpoznawanie pracownika na podstawie odczytu z karty.
- Zapis godziny wejścia i wyjścia pracownika.
- Prezentacja danych zebranych w bazie danych.
- Sygnalizacja przepuszczenia pracownika i błędnej karty.
- Panel wyświetlania pracowników i godziny ich pracy.
- Panel dodawania nowych pracowników i przypisywania do nich karty.

## 1.3 Moduły

Aby zrealizować projekt, został on podzielony na dwa moduły:

1. **Sterownik** - moduł obsługujący wczytywanie karty i zapisywanie danych do bazy, a także informowanie użytkownika o pomyślnej bądź niepomyślnej autoryzacji.
2. **Aplikacja** - panel administratora, gdzie wyświetlane są godziny pracy pracowników oraz gdzie można zarządzać pracownikami i przypisywać do nich karty.

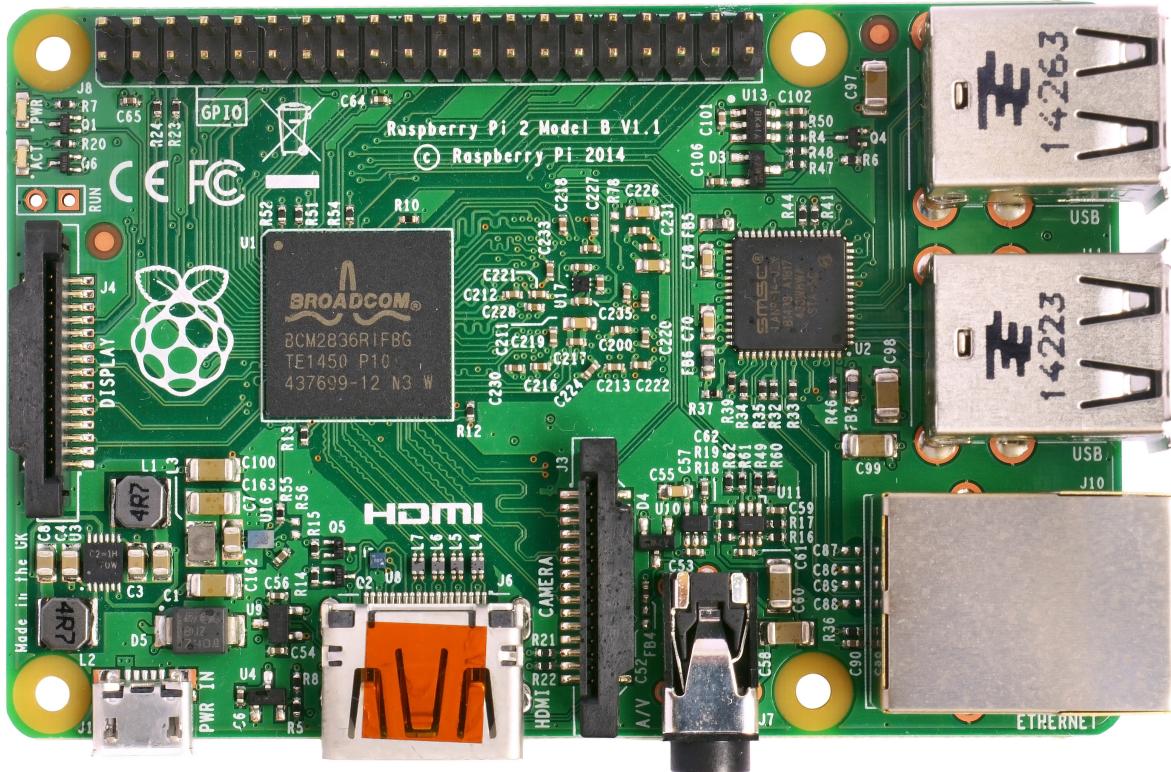
# Rozdział 2

## Sterownik

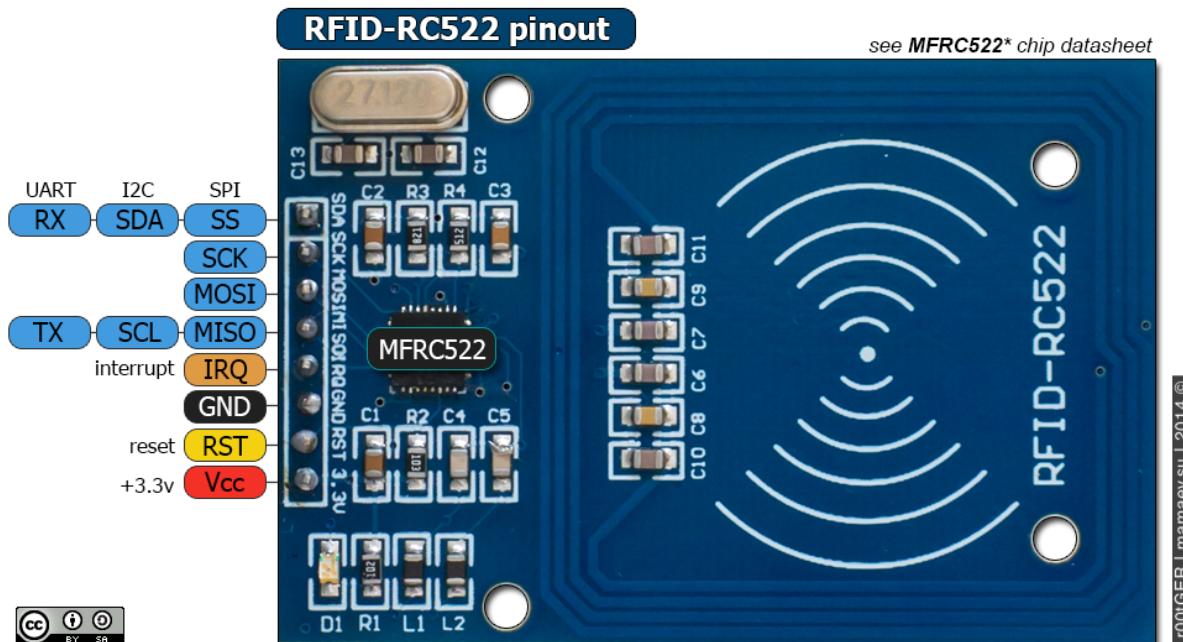
Aby zrealizować zamierzony cel, wybrana została technologia zbliżeniowa RFID, gdyż jest ona obecnie bardzo popularna i można znaleźć wiele sprzętu w niej operujących.

### 2.1 Wykorzystany sprzęt

Jako system bazowy został wykorzystany Raspberry Pi w wersji 2 B v1.1 (Rysunek 2.1). Różni się on od poprzednich wersji płytka ilością pinów oraz wielkością pamięci operacyjnej i szybszym mikroprocesorem.

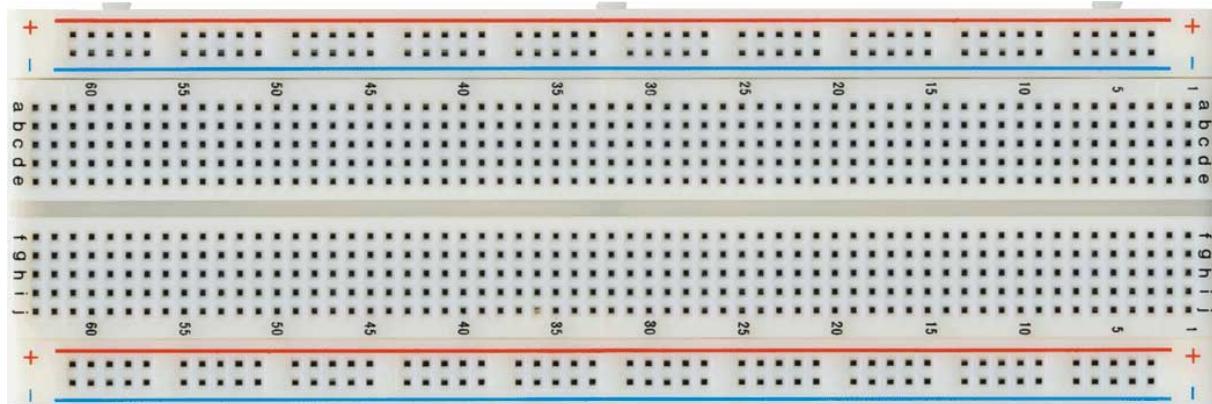


Aby obsłużyć technologię RFID użyto modułu z anteną MF RC522 (Rysunek 2.1). Pracuje on z częstotliwością 13,56 MHz, czyli najpopularniej występującą częstotliwością RFID. Moduł ten łączy się z Raspberry po interfejsie SPI.



Rysunek 2.2 Czytnik RFID z anteną MF RC522<sup>2</sup>

Aby złączyć moduły ze sobą wykorzystana została płytka stykowa, zwana inaczej płytą prototypową (Rysunek 2.1).



Rysunek 2.3 Płytkę prototypowa na 830 styków<sup>3</sup>

Do sygnalizacji poprawnej bądź niepoprawnej autoryzacji, użyte zostały dwie zwykłe diody - czerwona do sygnalizacji błędu oraz zielona do sygnalizacji poprawnej autoryzacji. Do połączenia ich z płytą wykorzystano dwa rezystory, o oporności odpowiednio 220 Ohm oraz 150 Ohm.

<sup>2</sup><https://github.com/r00tGER/RFID-RC522>

<sup>3</sup><http://technovade.pl/arduino/arduino-akcesoria/plytka-stykowa-prototypowa-830-otworow.html>

## 2.2 Podłączenie

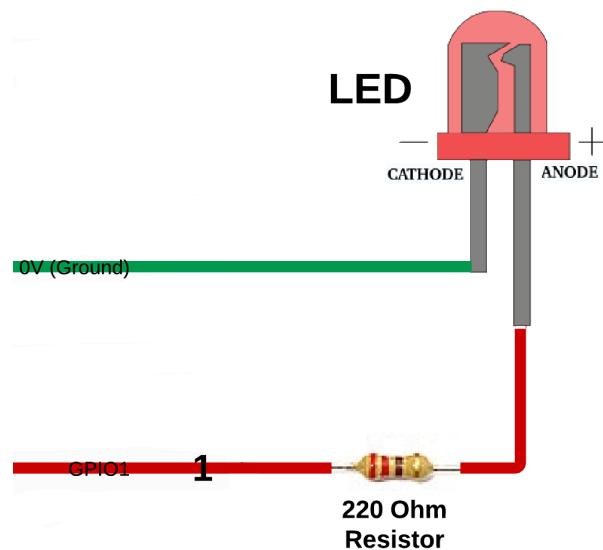
Przed przystąpieniem do realizacji należało najpierw przylutować czytnik RFID do goldpinów, gdyż w przeciwnym wypadku siła sygnału byłaby za słaba, aby móc przesyłać dane do oraz z powrotem z płytki. Po zlutowaniu podłączono wszystkie styki czytnika do pinów SPI, zgodnie z dokumentacją modułu MFRC522 (Pozycja [1]) oraz schematem rozłożenia wyjść na płytce (Pozycja [5]).

MFRC522 podłączony został do poniższych portów (pin IRQ nie jest wymagany do podłączenia, więc pozostał luźny):

Tabela. 2.1 Połączenia pinów czytnika RFID

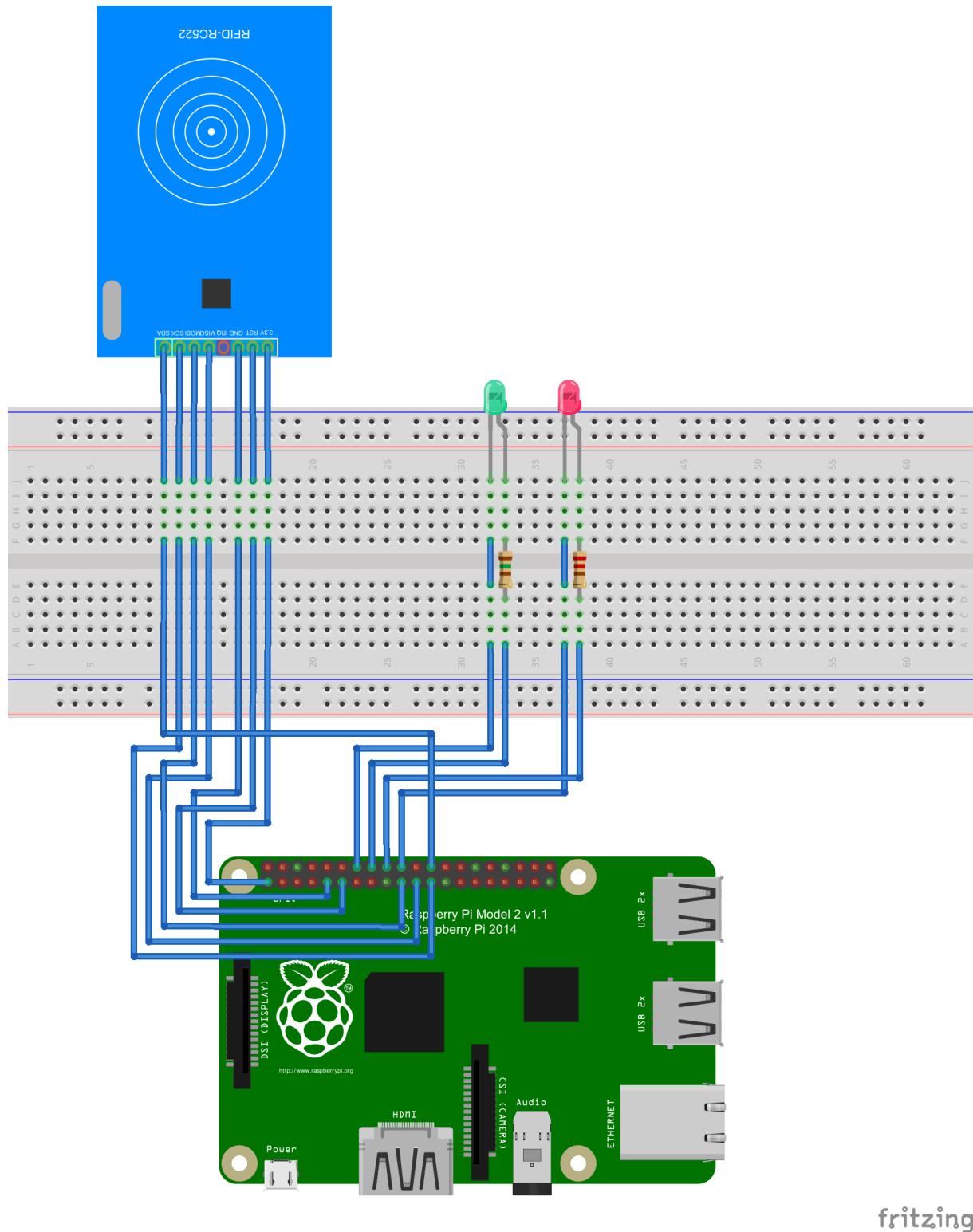
| MFRC522 | Numer pinu Raspberry | Nazwa pinu |
|---------|----------------------|------------|
| SDA     | 24                   | GPIO8      |
| SCK     | 23                   | GPIO11     |
| MOSI    | 19                   | GPIO10     |
| MISO    | 21                   | GPIO9      |
| IRQ     | x                    | x          |
| GND     | 9                    | GND        |
| RST     | 22                   | GPIO25     |
| 3.3V    | 1                    | 3V3        |

Diody podłączone zostały jak na Rysunku 2.2, czyli katodę diody czerwonej podłączono do masy, zaś anodę poprzez 220-Ohmowy rezystor do pinu GPIO24. Dioda zielona podłączona została bardzo podobnie, z tą różnicą, że spięto ją z pinem GPIO23 oraz ze względu na większe natężenie prądu potrzebne do jej zaświecenia, zastosowano opornik 150-Ohmowy.



Rysunek 2.4 Podłączenie czerwonej diody

Poniższy schemat z programu Fritzing<sup>4</sup> przedstawia finalny układ połączeń.



Rysunek 2.5 Finalny wygląd połączeń

<sup>4</sup><http://fritzing.org/home/>

## 2.3 Konfiguracja

Aby zmusić kod do działania, należy skonfigurować potrzebne narzędzia i biblioteki. Należy najpierw uruchomić do działania porty SPI. Aby tego dokonać wpisujemy w terminalu:

```
$ sudo raspi-config
```

Następnie przejść do Advanced Options i przestawić SPI na Enabled. Na starszych wersjach Raspberry należało również SPI z listy zablokowanych, poprzez uruchomienie edytora:

```
$ sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

I zakomentowanie linii dodając przed nią znak #:

```
blacklist spi-bcm2708
```

Kolejnym krokiem jest zrestartowanie urządzenia, i po restarcie wpisanie:

```
$ ls /dev/spidev0.*
```

Powinniśmy otrzymać wynik podobny do następującego, co świadczy o tym, że SPI zostało uruchomione poprawnie:

```
/dev/spidev0.0 /dev/spidev0.1
```

Kolejnym krokiem jest włączenie drzewa urządzeń. Należy wpisać:

```
$ sudo nano /boot/config.txt
```

I dodać na końcu pliku następującą linijkę:

```
device_tree=on
```

Następnie należy jeszcze zaktualizować sterownik dla mikrokontrolera Broadcom BCM 2835. Pobieramy jego najnowszą wersję<sup>5</sup> i wpisujemy kolejno:

```
$ tar zxvf bcm2835-1.xx.tar.gz
$ cd bcm2835-1.xx
$ ./configure
$ make
$ sudo make check
$ sudo make install
```

Po tych krokach urządzenie jest już skonfigurowane do korzystania z SPI oraz modułu RFID.

---

<sup>5</sup><http://www.airspayce.com/mikem/bcm2835/>

## 2.4 Oprogramowanie

Oprogramowanie napisane zostało w języku Python i wykorzystuje do działania dwie zewnętrzne biblioteki. Pierwszą jest SPI-Py<sup>6</sup>, która umożliwia korzystanie z interfejsu SPI z poziomu interpretera Python. Drugą jest MFRC522-python<sup>7</sup>, dzięki której można korzystać z modułu czytnika kart RFID.

W aplikacji wyróżnić można kilka modułów:

1. `main.py` - główny moduł systemu, odpowiada za pętlę programu.
2. `logger.py` - moduł odpowiedzialny za obsługę logów aplikacji.
3. `database.py` - łączenie się z bazą danych oraz dodawanie do niej rekordów.
4. `nfc.py` - obsługa odczytywania danych z karty.

Program uruchamia się wpisując w konsoli (wymagana jest wersja Pythona co najmniej 3.0):

```
python3 main.py
```

Aplikacja spróbuje teraz połączyć się z bazą danych i w wypadku pomyślnego połączenia uruchomiona zostanie główna pętla programu. Pętla ta uruchamia mniejszą pętlę w module `nfc.py`, który to w czasie ciągłym (aż do momentu wcisnięcia kombinacji klawiszy Ctrl + C) uruchamia cewkę na module MFRC522 i sprawdza czy w pobliżu nie znajduje się karta RFID. Jeśli tak, to odczytywany jest identyfikator karty, który następnie porównywany jest z identyfikatorami wszystkich przypisanych do systemu użytkownikami. Jeśli Id zostanie odnaleziony, do bazy zostaje dodany pod Id pracownika timestamp z aktualną datą i godziną odbicia się. System sygnalizuje też poprawną autoryzację miganiem zielonej diody. Jeśli identyfikator nie zostanie odnaleziony w bazie, system sygnalizuje to miganiem diody czerwonej.

Aplikacja kończy swe działanie w momencie wykrycia kombinacji klawiszy Ctrl + C, kończąc działanie modułu MFRC522 i zamykając połączenie do bazy danych.

---

<sup>6</sup><https://github.com/lthiery/SPI-Py>

<sup>7</sup><https://github.com/mgxw/MFRC522-python>

# Rozdział 3

## Aplikacja

### 3.1 Program prezentacji danych

Aplikacja została wykonana z myślą o pracodawcy, który chciałby mieć podgląd obecnych pracowników w biurze, oraz podsumowanie przepracowanego czasu dla poszczególnego pracownika.

Aplikacja składa się z kilku modułów:

1. `main.py` - główny moduł systemu, odpowiada za uruchomienie aplikacji.
2. `workingTimeDialog.py` - obsługa bazy danych i zapytań użytkownika.
3. `uiWorkingTime.py` - moduł konfigurujący graficzny interfejs.

Program uruchamia się wpisując w konsoli:

```
python3 main.py
```

### 3.1.1 Podgląd obecnych pracowników

W tej części programu można podejrzeć obecnych pracowników ich imię, nazwisko oraz czas przybycia. Widok odświeża się do pół sekundy.

| Present Employees |            |           |
|-------------------|------------|-----------|
|                   | First Name | Last Name |
| 1                 | Mateusz    | Nowak     |
| 2                 | Marcin     | Jankowski |

Rysunek 3.1 Obecni pracownicy

### 3.1.2 Podgląd prepracowanego czasu pracowników

Podgląd umożliwia wybrania przedziału czasu. Następnie wyświetli się imię, nazwisko oraz przepracowany czas.

| Time Worked |                  |                      |
|-------------|------------------|----------------------|
|             | First Name       | Last Name            |
| From:       | 2015-12-01 00:00 | To: 2015-12-02 18:47 |
| 1           | Jan              | Kowalski             |
| 2           | Mateusz          | Nowak                |
| 3           | Anna             | Kowalczyk            |
| 4           | Marcin           | Jankowski            |

Rysunek 3.2 Przepracowany czas

### 3.1.3 Dodawanie pracowników

Ta część programu pozwala na dodawanie użytkowników. Wymagane są wszystkie pola do wpisania. Należy podać: Imię, Nazwisko, id taga rfid. email oraz hasło do systemu.

The form consists of five text input fields arranged in two rows. The first row contains 'First Name:' with an input field, 'Email:' with an input field, and 'TagId:' with an input field. The second row contains 'Last Name:' with an input field and 'Password:' with an input field. Below these fields is a single 'Add User' button.

Rysunek 3.3 Dodaj pracownika

## 3.2 Technologie

Wykorzystane technologie:

- Python3.4 - Język programowania, w którym została wykonana aplikacja.
- PyQt5 - Biblioteka języka python do interfejsu graficznego.
- psycopg2 - Biblioteka języka python do komunikacji z bazą danych PostgreSQL
- PostgreSQL - Baza danych.
- Vagrant - Środowisko do przygotowania maszyn wirtualnych.
- Raspbian - Dystrybujca linuxa przygotowanego dla RPi na podstawie dystrybucji Debian.

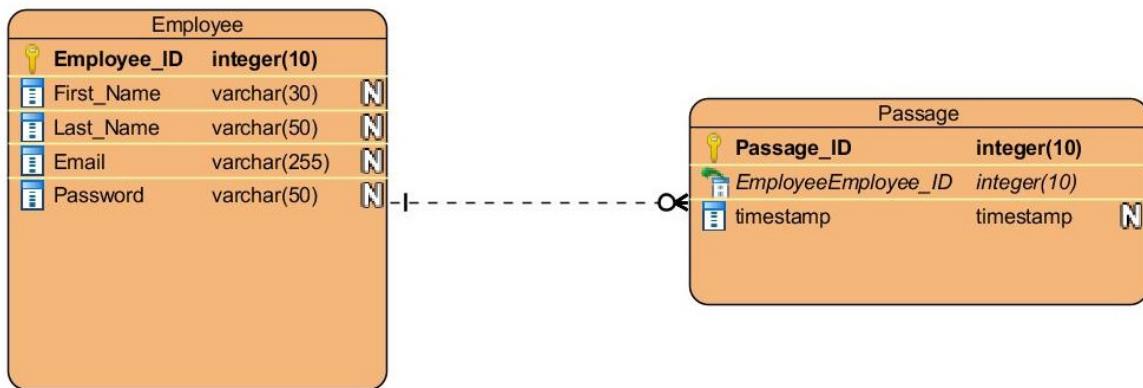
## 3.3 Przygotowanie środowiska

Aby uruchomić program prezentacji danych potrzeba zainstalować interpreter Python3.4, oraz biblioteki PyQt5, psycopg2. Program używa danych przechowywanych w bazie danych PostgreSQL.

## 3.4 Baza Danych

Baza danych składa się z dwóch tabel, Pracowników oraz przejść. Do każdego pracownika może być przypisane wiele przejść.

Możliwe jest ustawienie bazy danych na dowolnym urządzeniu. Takie rozwiązanie było niezbedne ze względu na to, że posiadaliśmy tylko jeden komputer RPi. Aplikacja prezentacji danych, komunikuje się wyłącznie z bazą danych, więc możliwe było tworzenie aplikacji bez komputera RPi.



Rysunek 3.4 Diagram ERD

### 3.4.1 Baza danych na maszynie wirtualnej

W ramach projektu został przygotowany plik konfiguracyjny maszyny ([github:/database/Vagrantfile](https://github.com/JanuszKwiatkowski/Database/blob/main/Vagrantfile)). Aby wygenerować maszynę wirtualną potrzebny jest program Vagrant. Uruchomienie skonfigurowanej maszyny wirtualnej, polega na wejściu do katalogu z plikiem konfiguracyjnym i wywołaniem komendy "vagrant up". Następnie trzeba przekazać port 5432 bazy danych z maszyny wirtualnej do systemu operacyjnego na ip 127.0.0.1, w projekcie wykorzystywany był do tego program putty. Dane logowania do maszyny wirtualnej to: użytkownik - vagrant, hasło - vagrant.

### 3.4.2 Baza dnych na Raspberry Pi

Do pracy na RPi Wybrałem dystrybucje linuxa Raspbian. Aby skonfigurować bazę danych na Raspberry Pi należy wykonać następujące komendy:

```

apt-get update
apt-get install postgresql postgresql-contrib postgis gpsbabel
git libsqlite3-dev libreadline-dev libpq-dev libbz2-dev
 zlib1g-dev libpqxx-dev libzip-dev -y
echo -ne "alamakota\alamakota" | su - postgres -c 'createuser -P -e wbudowane'
su - postgres -c 'createdb -e -O wbudowane wbudowane'
su - postgres -c 'psql wbudowane < /vagrant/czaspracyBD.sql'
su - postgres -c 'psql wbudowane -c "GRANT_ALL_ON_TABLE_Employee_TO_wbudowane;"'
su - postgres -c 'psql wbudowane -c "GRANT_ALL_ON_TABLE_Passage_TO_wbudowane;"'
su - postgres -c 'psql wbudowane -c "GRANT_USAGE_SELECT_ON_ALL_SEQUENCES_IN_SCHEMA_public_to_wbudowane;"'

```

# Rozdział 4

## Podsumowanie

W ramach projektu Zostały zaimplementowane wszystkie funkcjonalności podstawowe jakaś zakładaliśmy:

- Rozpoznanie pracownika na podstawie odczytu z karty
- Zapis godziny wejścia i wyjścia pracownika
- Przentacja danych zebranych w bazie danych
- Sygnalizacja przepuszczenia pracownika i błędnej karty

### 4.1 Ocena krytyczna

Nie udało się zrealizować funkcjonalności opcjonalnych:

- Hasło wpisywane z klawiatury numerycznej
- Analiza zebranych danych
- Obsługa niepoprawnego zachowania użytkownika

Aplikacja prezentacji przedstawia jedynie surowe dane, oraz nie została zaimplementowana obsługa błędów

### 4.2 Możliwości rozwoju

System można rozszerzyć między innymi o funkcjonalności opcjonalne. Przydatne mogło by być generowanie raportów z analizą danych. Innym przydatnym rozszerzeniem mogłoby być wpisywanie urlopów oraz uwzględnianie ich w raportach.



# Bibliografia

- [1] Dokumentacja modułu mfrc522.  
<http://www.elecrow.com/download/MFRC522%20Datasheet.pdf>. Dostępne na dzień: 2015-11-30.
- [2] Dokumentacja sterownika psycopg. <http://initd.org/psycopg/docs/usage.html>. Dostępne na dzień: 2015-11-30.
- [3] Poradnik dotyczący łączenia modułu mfrc522 z raspberry 2.  
<http://helloraspberrypi.blogspot.com/2015/10/raspberry-pi-2-mfrc522-python-to-read.html>. Dostępne na dzień: 2015-11-30.
- [4] Poradnik na temat zdalnego łączenia się z systemem raspberry.  
<https://anwaarullah.wordpress.com/2013/07/16/direct-access-raspberry-pi-shell-and-desktop/>. Dostępne na dzień: 2015-11-30.
- [5] Rozłożenie pinów na płytce raspberry pi 2.  
<http://www.element14.com/community/docs/DOC-73950/1/raspberry-pi-2-model-b-gpio-40-pin-block-pinout>. Dostępne na dzień: 2015-11-30.



# Spis rysunków

|     |                                  |    |
|-----|----------------------------------|----|
| 2.1 | Raspberry Pi w wersji 2 B v1.1   | 5  |
| 2.2 | Czytnik RFID z anteną MF RC522   | 6  |
| 2.3 | Płytką prototypową na 830 styków | 6  |
| 2.4 | Podłączenie czerwonej diody      | 7  |
| 2.5 | Finalny wygląd połączeń          | 8  |
| 3.1 | Obecni pracownicy                | 12 |
| 3.2 | Przepracowany czas               | 12 |
| 3.3 | Dodaj pracownika                 | 13 |
| 3.4 | Diagram ERD                      | 14 |



# Spis tabel

|  |   |
|--|---|
| 2.1 Połączenia pinów czytnika RFID . . . . . | 7 |
|--|---|