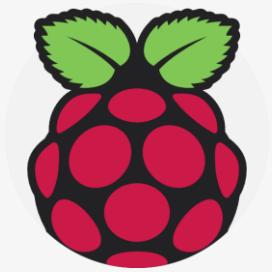




RFID를 이용한 출퇴근 관리 시스템

부경대학교 202213309 여은하

01 사용 기술



Raspberry 4



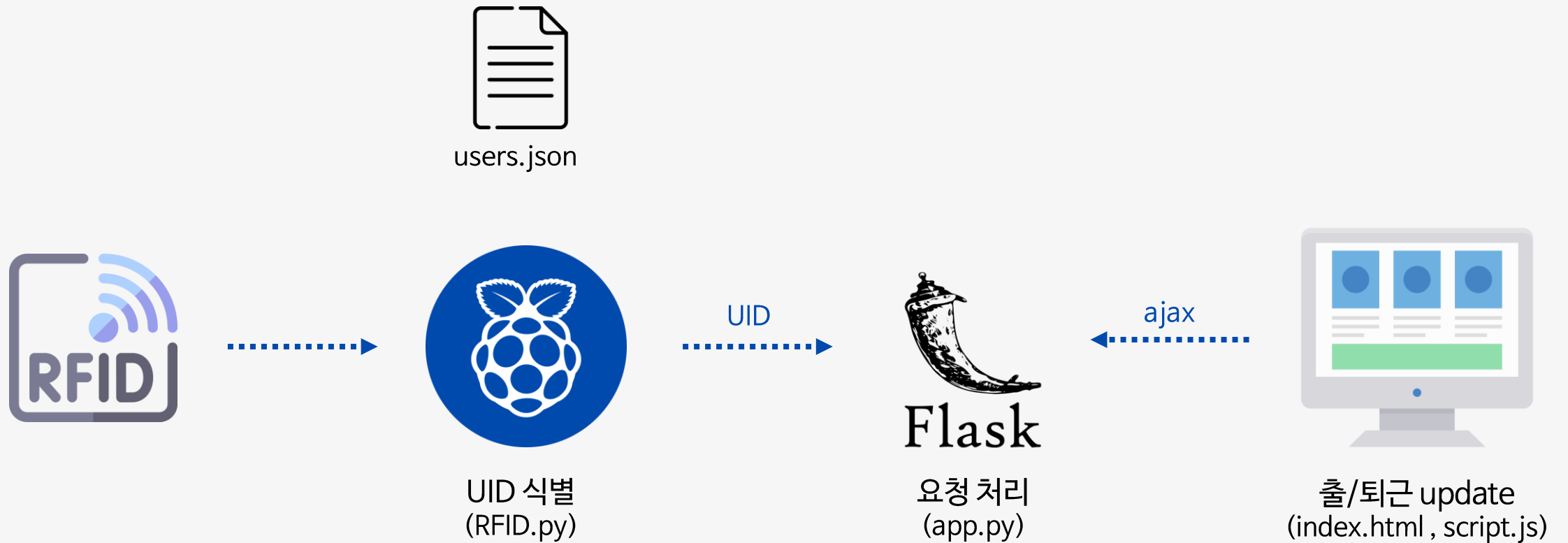
MFRC-522 RFID



2색 LED



02 로직 설명



02 로직 설명

RFID Attendance System

출근 (online 효과)



EUNHA

go to work:16:18

퇴근



PKNU

go to work:16:18

leave work:16:32

미출근 (offline 효과)



STARLINK

03 주요 소스코드 - users.json

```
[
  {
    "uid": 693214255894,
    "username": "EUNHA",
    "image": "/static/eunha.png",
    "checked_in": false,
    "start_time": "00:00",
    "end_time": "00:00"
  },
  {
    "uid": 97817702172,
    "username": "PKNU",
    "image": "/static/PKNU.png",
    "checked_in": false,
    "start_time": "00:00",
    "end_time": "00:00"
  },
  {
    "uid": 460768275501,
    "username": "STARLINK",
    "image": "/static/starlink.png",
    "checked_in": false,
    "start_time": "00:00",
    "end_time": "00:00"
  }
  ...
]
```

유저 정보가 포함된 json 파일

uid : RFID 식별 넘버

username : 사용자 이름

image : 프로필 이미지 경로

checked_in : 출근여부

start_time : 출근시간

end_time : 퇴근시간

03 주요 소스코드 - RFID.py

Python에서 제공하는 RFID 리더 라이브러리인
MFRC-522 사용

```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
import requests
import json
import time

reader = SimpleMFRC522()

#users.json
with open('users.json') as f:
    valid_uids = {user['uid']: user for user in json.load(f)}

#Flask server.
FLASK_SERVER_URL = 'http://127.0.0.1:5000/update_uid'

def send_uid_to_server(uid):
    data = {'uid': uid}

    try:
        requests.post(FLASK_SERVER_URL, json=data) #POST요청 전송
    except Exception as e:
        print(f"Error sending UID to server: {e}")

#RFID
try:
    while True:
        id, text = reader.read() #RFID 식별

        if id in valid_uids: #led code
            send_uid_to_server(id)
            time.sleep(1)
    ...
```

2. /update_uid 엔드포인트에 UID를 POST 형식으로 전송

1. RFID 인식

RFID가 인식되면 해당 넘버가 users.json안에 있는지 확인
유효하면 send_uid_to_server 호출

03 주요 소스코드 - app.py

```
from flask import Flask, request, jsonify, render_template
import json
from datetime import datetime

app = Flask(__name__)

with open('users.json') as f:
    valid_uids = {user['uid']: user for user in json.load(f)}

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/get_users', methods=['GET'])
def get_users():
    return jsonify(list(valid_uids.values()))

@app.route('/update_uid', methods=['POST']) #POST 요청 받음
def update_uid():
    data = request.get_json()
    uid = data.get('uid')

    user = valid_uids.get(uid)

    if user:
        current_time = datetime.now(kr_tz).strftime('%H:%M')

        if user['checked_in']: #이미 출근했다면 => 퇴근 기록
            user['end_time'] = current_time
            user['checked_in'] = False #출근 기록
        else:
            user['start_time'] = current_time
            user['checked_in'] = True

        return jsonify({'status': 'success', 'uid': uid, 'username': user['username'],
                        'time': current_time, 'message': 'success'})
    else:
        return jsonify({'status': 'error', 'message': 'Invalid UID'}), 400

if __name__ == '__main__':
    app.run(debug=True)
```

유저 리스트를 웹 사이트에 띄우는 함수

1. RFID.py 에서 보낸 POST 요청 받으면 해당 UID의
상태(checked_in) 확인

출근 => 현재 시간을 “퇴근 ” 에 기록

미출근 => 현재 시간을 “출근 ” 에 기록

Json 형식으로 리턴

03 주요 소스코드 - index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>RFID Attendance System</title>

  <link rel="stylesheet" href="/static/style.css">
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script src="/static/script.js"></script>

</head>
<body>
  <h1>RFID Attendance System</h1>
  <div class="wrap">
    <div class="box-zone flex" id="user-container">

      <!-- <div class="box online" data-uid="{user.uid}">
        
        <h2>eunha</h2>
        <div>go to work:<span>09:00</span><br>
        </div>
      </div> -->

    </div>
  </div>

  <footer>Embedded System Programming 2 final project
    - 202213309 Yeoeunha</footer>

</body>
</html>
```

CSS, jQuery 로드

유저 리스트가 보여지는 공간

03 주요 소스코드 - script.js (1)

```
$(function(){
  function loadUsers() {
    $.ajax({
      url: '/get_users',
      type: 'GET',
      success: function(users) {
        const userContainer = $('#user-container');
        userContainer.empty();

        users.forEach(user => {
          let userBox;
          if(user.start_time !== "00:00"){ //출근을 한 상태라면
            if(user.end_time !== "00:00"){
              userBox = `
                <div class="box">
                  
                  <h2>${user.username}</h2>
                  <div>go to work:<span>${user.start_time}</span><br>
                    leave work:<span>${user.end_time}</span>
                  </div>
                </div>
              `;
            }else{
              userBox = `
                <div class="box online">
                  
                  <h2>${user.username}</h2>
                  <div>go to work:<span>${user.start_time}</span><br></div>
                </div>
              `;
            }
          }
        });
      }
    });
  }
  loadUsers();
}
```

1. /get_users 엔드포인트에 ajax로 데이터 요청

2. 성공하면 받은 데이터를 바탕으로 RFID를 태그한
유저의 정보 업데이트

출근+퇴근

=> 출/퇴근 시간이 기록된 div 박스 업데이트

출근 (업무 중)

=> online 효과와 출근 시간만 기록된 div 박스
업데이트

03 주요 소스코드 - script.js(2)

```
}else{ //출근안함
    userBox = `
        <div class="box offline">
            
            <h2>${user.username}</h2>
        </div>
    `;
}

userContainer.append(userBox);
});
error: function(error) {
    console.error('Error loading users:', error);
}
});
}

loadUsers();

setInterval(loadUsers, 1000); // 1초마다 업데이트
})
```

(출근 전)

=> offline 효과가 포함된 div 박스 추가

변경사항 index.html에 반영

3. 1초마다 ajax 요청 전송

```
GPIO.setmode(GPIO.BCM)
reader = SimpleMFRC522()
```

```
GPIO.setwarnings(False)
r=14
g=15
GPIO.setup(r, GPIO.OUT)
GPIO.setup(g, GPIO.OUT)
red=GPIO.PWM(r,100)
green=GPIO.PWM(g,100)
```

```
def redControl():
    red.ChangeDutyCycle(100)
    time.sleep(2)
    red.ChangeDutyCycle(0)
```

```
def greenControl():
    green.ChangeDutyCycle(100)
    time.sleep(2)
    green.ChangeDutyCycle(0)
```

... (RFID 코드)

#RFID

```
try:
    red.start(0)
    green.start(0)

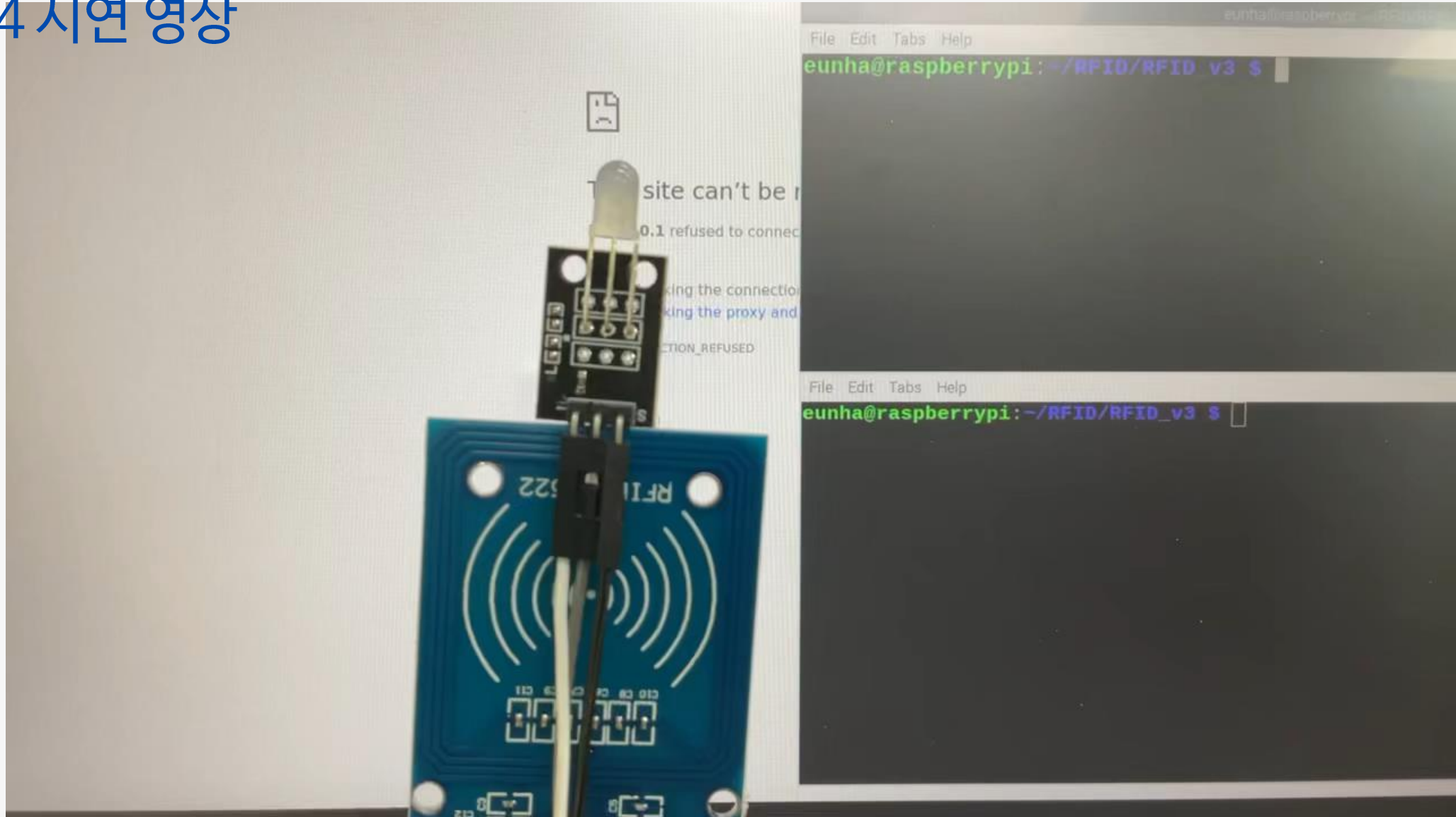
    while True:
        id, text = reader.read() #RFID 식별
        print(id)
        if id in valid_uids: #led code
            send_uid_to_server(id)
            greenControl()
        else:
            redControl()
```

03 주요 소스코드 - RFID.py+led

유효한 UID => 초록LED

없는 UID => 빨강LED

04 시연 영상



QnA

부경대학교 202213309 여은하