

고급IoT프로젝트 01분반

Take Home Exam

20191312 배준재

구성은 app.py, index.html, bjjcode.ino로 구성되어있다.

app.py에 flask로 서버, index.html에 웹페이지, bjjcode.ino에 아두이노 코드가 있다.

Index.html은 templates폴더를 만들어 그 안에 생성하여 flask가 index.html을 못찾았을 때를 대비한다.

아두이노 코드는 컴파일 위치와 업로딩을 위해 다음과 같이 bjjcode아래 위치해있다.

따라서 프로젝트 tree는 다음과 같다

Project

|-app.py

|-templates

- |--index.html

|-bjjcode

-|--bjjcode.ino

app.py를 실행할 때 다음을 설치해야한다.

pip install flask, pip install serial, pip install picamera, pip install opencv-python, pip install pyserial,

Python app.py를 통해 실행한다. 다음 사진은 실행하여 웹페이지에 접속하면 보이는 화면이다.

IoT Control Interface

Capture	
Enter number	7-Segment
Enter text	LCD

기능

Capture를 누르면 사진이 찍힌다. 캡처가 되면 도레미도레미가 울린다.

Enter number에 숫자를 넣고 7-segment버튼을 누르면 표시가 된다.

Enter text에 문자열을 넣고 LCD버튼을 누르면 LCD에 문자열이 표시된다.

```
(myenv) bij@raspberrypi:~$ cd templates/
(myenv) bij@raspberrypi:~/templates$ vi index.html
(myenv) bij@raspberrypi:~/templates$ (myenv) bij@raspberrypi:~/templates$ cd ..
(myenv) bij@raspberrypi:~$ python app.py --host 0.0.0.0
 * Serving Flask app "app"
 * Debug mode: off
 WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://192.168.137.1:5000
 * Running on http://192.168.137.159:5000
Press CTRL+C to quit
192.168.137.1 - - [04/Nov/2024 22:08:02] "GET / HTTP/1.1" 200 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "POST /capture HTTP/1.1" 302 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "GET / HTTP/1.1" 200 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "POST /capture HTTP/1.1" 302 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "GET / HTTP/1.1" 200 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "POST /capture HTTP/1.1" 302 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "GET / HTTP/1.1" 200 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "POST /capture HTTP/1.1" 302 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "GET / HTTP/1.1" 200 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "POST /capture HTTP/1.1" 302 -
192.168.137.1 - - [04/Nov/2024 22:08:03] "GET / HTTP/1.1" 200 -
```

```
192.168.137.1 - - [05/Nov/2024 20:46:52] "POST /capture HTTP/1.1" 302 -
192.168.137.1 - - [05/Nov/2024 20:46:52] "GET / HTTP/1.1" 200 -
(myenv) bij@raspberrypi:~$ ls
app.py      Bookshelf  IoT-Take-Home-Exam  photo_20241105-204045.jpg  Public      templates
Arduino     Desktop    Music                photo_20241105-204140.jpg  templates   Videos
bin         Documents  myenv                photo_20241105-204640.jpg  Templates   wiringpi_3.10_arm64.deb
bijcode     Downloads  photo_20241105-204041.jpg  Pictures
(myenv) bij@raspberrypi:~$
```

사진은 현재 디렉토리에 저장된다.

각 버튼을 눌러 전송 결과는 다음과 같이 나온다.

통신방법은 post로 웹페이지 요청을 serial을 통해 usb로 연결한 아두이노에 /dev/ttyACM0에 직접 연결하여 통신한다.

플라스크 서버에서 html에 route를 통해 연결하고 post를 날리면서 Arduino.write를 통해 아두이노를 호출한다. 아래는 예시이다.

@app.route('/capture', methods=['POST']) <= capture했을 때 POST를 날림

def capture():

cap = cv2.VideoCapture(0)

ret,frame = cap.read()

if ret:

timestamp = time.strftime("%Y%m%d-%H%M%S")

image_path = f'./photo_{timestamp}.jpg' <= 현재 디렉토리에 사진저장

cv2.imwrite(image_path,frame)

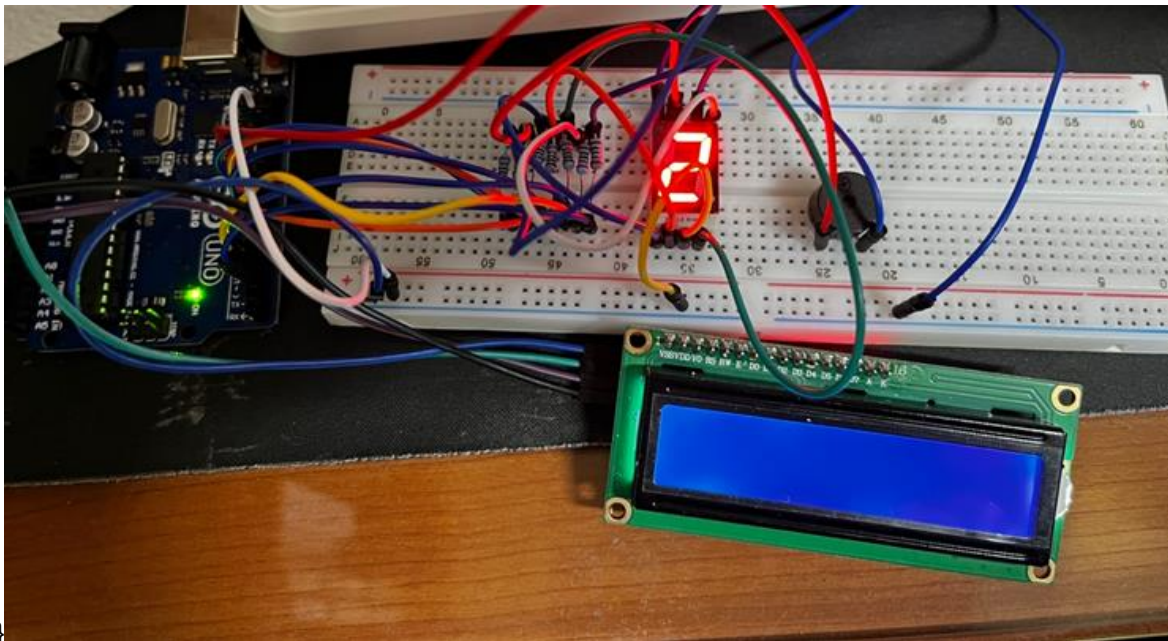
```
    arduino.write(b'MELODY\\n') <= 아두이노에서 MELODY를 실행  
    cap.release()  
    return redirect(url_for('index'))
```

index.html

```
<form action="/capture" method="post">  
<button type="submit">Capture</button>
```

bjjcode.ino

```
else if (input.startsWith("MELODY:")) { <= flask에서 요청을 받아서 함수 실행  
    playMelody();
```



아두이노 회로는 위 사진과 같다.

7segment는 2~8, 부저는9번핀, lcd는 A4,A5에 연결하면 된다.

실행시 주의 사항으로 아두이노 컴파일러 path를 잘 설정 해야 한다.