

3조

김명진 송지호 신지환

목 차

배경과
동기

소개

구현방법

보완 및
학습효과

팀원
역할

배경과 동기

일반시계와 알람시계의 차이점: 알람기능 수행 여부의 차이

알람기능: 정해진 시간에 맞추어 사용자가 일어날 수 있도록 도와줌

기존의 알람시계: 알람음이 울려도 사용자가 무의식 중에 쉽게 알람을 끌 수 있어서 그 기능을 상실해 버리는 경우가 대부분



‘얼굴 인식 기능’

> 의식을 가진 상태에서만 알람을 끌 수 있게 알람을 구상함

소개

알람 기능은 총 4가지가 있다. 입력하기전, 사용자의 이름을 입력 받는다.

1.Set alarm 2.Set alarm tone 3.Set alarm size 4. Exit

- 1 (Set alarm)

알람 시간을 입력함. 입력범위를 초과하면 다시 입력하게 함.

입력하면 LED Matrix의 해당 패널에 알람 시간이 출력됨.

알람 시간이 되면 알람이 울림과 동시에 카메라가 켜지며, 얼굴이 인식될 때까지 알람이 울리게 됨.

ESC를 눌렀을 때 알람이 울리기 전에 입력해둔 사용자의 이름과 인식된 얼굴의 이름이 일치하면 알람이 꺼짐.

만약 일치하지않으면 ESC를 눌러도 카메라가 다시 작동되고 알람도 계속 울림.

- 2 (Set alarm tone)

Select the alarm sound(1~3): 알람음을 1~3 사이의 정수로 입력함

입력하면 LED Matrix의 해당 패널에 숫자가 출력됨.

- 3 (Set alarm size)

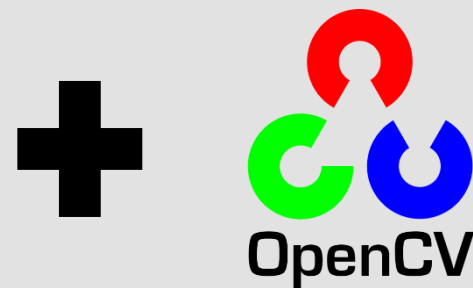
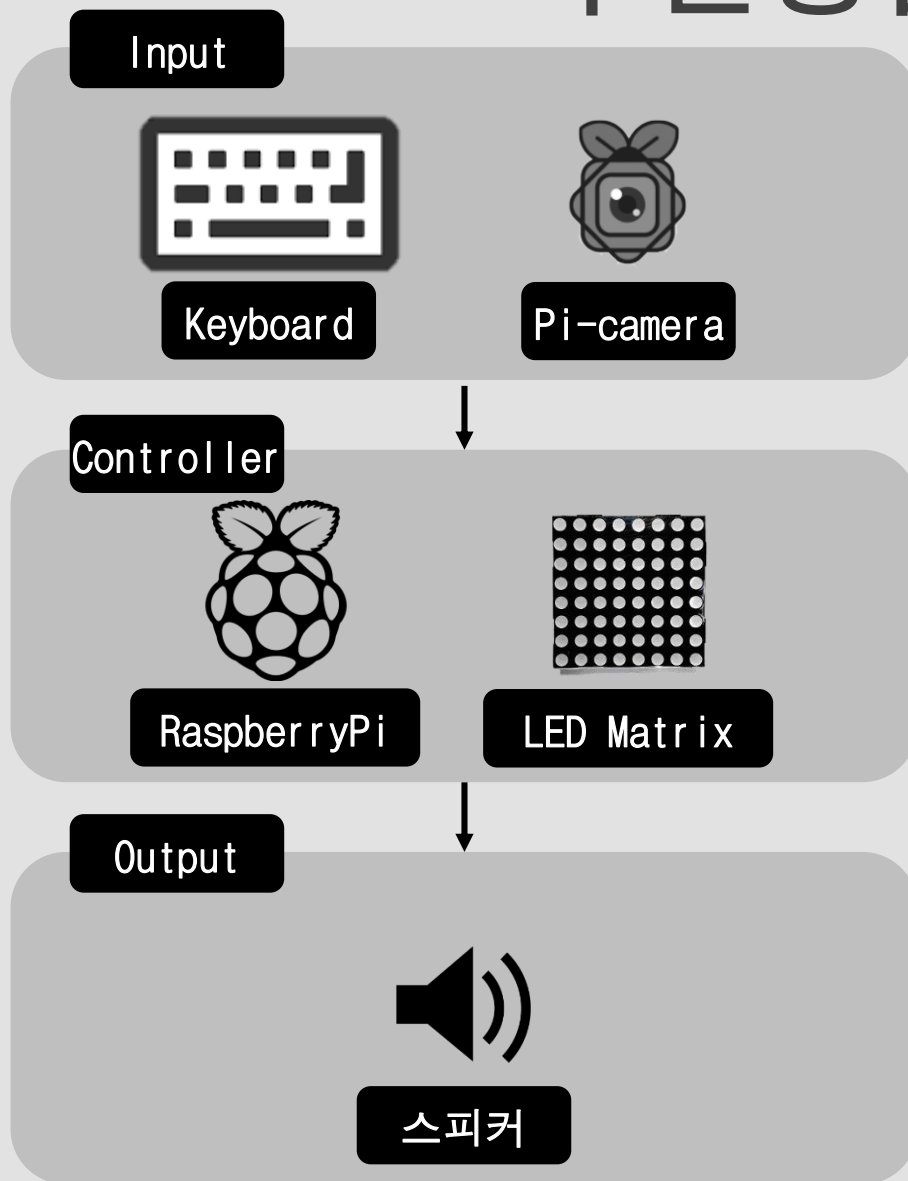
Please enter the size of the alarm: 알람음의 크기를 0~10사이의 정수로 입력함

입력하면 LED Matrix의 해당 패널에 숫자가 출력됨.

- 4 (Exit)

알람 프로그램을 종료시킴.

구현 방법



구현 방법

datetime에서 현재 날짜와 시간을 제공함.
Pygame에서는 mixer에서 sound 기능을 제공함
얼굴 인식 소스 코드를 모듈로 활용해 import함.

1 (Set alarm)

- input함수를 사용해 0~24 사이의 정수를 ahour로 받고 0~60 사이의 정수를 aminute로 받는다.
LED Matrix에 해당 알람 시각을 update해준다.

- datetime 모듈을 사용해 현재시각을 now라는 변수에 저장한후 now.hour과 ahour을 비교하여
alarmtime이라는 변수를 datetime(year, month, day, ahour, aminute, 0)의 형태로 저장한다.
time_gap이라는 변수를 선언해 (alarmtime-now).seconds로 저장하고
time.sleep(time_gap) 으로 프로세스를 일시 정지 시킨다.

- 다시 프로세스가 작동하면 무한 루프 안에서 sound.play(-1)을 만나 sound를 연속적으로 재생한다.
그리고 얼굴인식 소스코드에서 import 해온 함수의 return값(카메라에 인식된 사용자명)을 return_name
라는 변수에 넣는다. 그리고 맨처음에 입력한 user_name == return_name 이면 해당 루프를 탈출하면서
sound.stop()을 만나 알람음이 꺼진다. 만약 다르다면 다시 카메라가 켜진다.

구현 방법

실시간 얼굴 인식 by Pi-camera

지속적인 학습을 통해 사용자를 인식하게 하였고
카메라에 얼굴이 인식될 시 이름이 출력되게 구현함.

01_face_dataset.py, 02_face_training.py, recognition.py로 구성됨.

01_face_dataset.py를 통해 사용자의 얼굴은 인식하여 jpg 파일로 저장함.

02_face_training.py를 통해 앞서 저장한 jpg파일들을 불러와 yml확장자로 저장함.

1 (Set alarm) 에서 from recognition import face_recognition을 하여 recognition.py 내부의 face_recognition함수를 불러옴.

얼굴을 인식하여 학습된 결과를 통해 카메라 출력화면에 이름을 띄워줌.

ESC를 누를 시, 카메라가 종료되며, 인식된 이름을 return해줌.

구현 방법

2 (Set alarm tone)

알람음을 dictionary 형태로 사전에 정의하였다.

사용자로부터 input함수를 사용해 1~3 사이의 정수를 audio_sound로 받은 후 Pygame의 mixer.sound()를 사용해 해당 오디오를 알람음으로 설정 후 LED Matrix에 해당 오디오 번호를 update해준다.

3 (Set alarm size)

사용자로부터 input함수를 사용해 0~10 사이의 정수를 audio_volume로 받은 후 Pygame의 set_volume()를 사용하기 위해 10.0으로 나눠준 후 알람음 크기를 설정한다. (set_volume()의 인자값의 범위는 0~1이기 때문)
LED Matrix에 해당 알람음 크기를 update 해준다.

4 (Exit)

LED Matrix를 종료한다.

보완 및 학습효과

보완해야할 점

1. 얼굴 인식의 정확도가 생각보다 낮다는 점
-> 다양한 날에 사진을 찍어야 함
2. 얼굴 인식에서 ESC키를 눌러야 return_name가 return되므로 사용자가 번거로울 수 있다는 점
-> 내부 함수를 수정해 특정 시간이 지나면 자동으로 꺼지게 바뀌어야 함
3. 원래 세수를 한 얼굴을 인식하려 했으나 오픈소스 내부 알고리즘 상 불가능하다고 판단되어 그냥 얼굴을 인식 했다는 점 -> 활발할 때의 사진을 찍어 잠이 들면 상태에서는 인식이 어렵게 함
4. 새로운 사람을 추가하려할때, 함수 내부의 list를 직접 수정해야 한다는 점
-> 소스코드를 실행할 때, 이름을 입력하면 추가되는 방식으로 바뀌어야 함

학습 효과

1. 파이썬 언어를 통해 LED Matrix를 작동하는 법을 이해함
2. 오픈소스를 통해 구현하기 어려운 소스코드를 편리하게 사용할 수 있음

팀원 역할

김명진

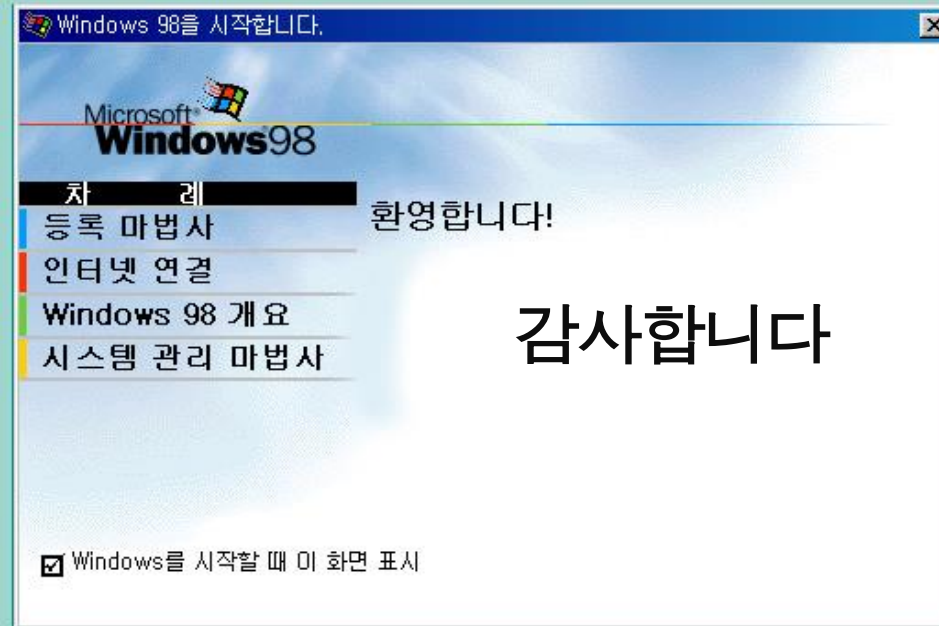
- 알람음 설정, 알람 볼륨 설정 소스코드 작성
- LED Matrix에 띄울 초기 UI구상 및 소스코드 구현
- 외장 3.5mm스피커 연결 작업
- 얼굴 인식 관련 소스코드와 알람 소스코드 병합
- 알람 시간 소스코드에서 알람 시간과 현재시간의 차이를 계속 반영하는 부분을 time모듈을 사용하여 한번만 현재시간을 불러오게 함으로 LED Matrix의 깜빡임을 없앴

송지호

- 알람시간 소스코드와 음악재생 소스코드를 병합하여 초기 알람 소스코드 작성
- LED matrix에 띄울 글자를 5 X 3 배열에 0과 1을 사용하여 리스트 형태로 작성
- LED Matrix에 띄울 초기 UI구상
- 알람 소스코드에서 시간을 비교할 때 입력값에 따른 실행 오류 디버깅

신지환

- Raspberry Pi와 Pi-camera 연결 및 드라이버 설치 및 테스트 소스코드 작성
- 실시간 얼굴 인식 소스코드 (face_dataset.py, face_trainer.py) 작성
- face_recognition.py 소스코드 작성 및 개편
- 음원 파일을 mp3->wav 형식으로 변환, 음원 파일 병합
- 알람 소스코드 디버깅 및 수정



YouTube URL: <https://youtu.be/3ChfQ7W7OEg>

Github URL: <https://github.com/sgh002400/osscap2020>