

않으면, 콘솔에 ‘no match’라고 표시한다.

pi 디렉터리에 들어간 상태에서 명령 프롬프트에 아래와 같이 입력하여 프로그램을 실행하자.

```
sudo python extract_data.py
```

콘솔 화면의 커서가 줄바꾸기를 하고, 리더 주위에 태그가 감지될 때까지 대기할 것이다. 태그를 감지하면 ‘match’ 혹은 ‘no match’를 표시하고, 일치하는 경우에는 LED를 10초 동안 표시할 것이다.

목표 태그의 ID가 실행 전에 하드 코딩되어 있기 때문에 이 프로그램은 제한적으로 작동하게 된다. 하지만 앞서 언급했듯, 이 프로그램은 이 장에서 진행하는 프로젝트의 기반이 되는 프로그램이므로 시스템의 확장을 통해 유연성과 유용성을 얼마든지 확보할 수 있다. 이어지는 내용을 참고한다.

```
extract_data.py

import time
import os
import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT)
nfc_id = None
while (nfc_id == None):
    nfc_id = os.system('sudo libnfc/libnfc-1.6.0-rc1/examples/nfc-poll > nfc_data.txt')
inFile = open('nfc_data.txt')
lines = inFile.readlines()
inFile.close()
buffer = []
for line in lines:
    line_content = line.split()
    if(not line_content[0] == 'UID'):
        pass
    else:
        buffer.append(line_content)
#print(buffer)
```

```
str = buffer[0]
id_str = str[2] + str[3] + str[4] + str[5]
#print(id_str)
if(id_str == 'bd0eb923'):
    print('match')
    GPIO.output(23, GPIO.HIGH)
    time.sleep(10)
    GPIO.output(23, GPIO.LOW)
else:
    print('no match')
```

확장하기

승인된 태그를 감지했을 때 라즈베리 파이가 전자식으로 자물쇠를 풀게 하기 위해서는, 위의 시스템을 확장해야 할 것이다. 대부분의 경우, 보안 구역이나 건물에 출입하기 위해 승인된 태그는 둘 이상이다. 각각의 태그는 고유 ID를 가지기 때문에 승인된 태그 ID 목록을 가지고 태그를 확인해야 한다. 파이썬에서 이 기능을 구현하는 방법에는 몇 가지가 있다. 데이터베이스에 태그 ID를 넣고 프로그램이 쿼리를 던지게 할 수도 있다. 실제 상용 시스템에서도 이런 방법을 사용하지만, 비용이 많이 든다.

특정한 태그를 감지하였을 때 프로그램이 내장된 이메일을 보내는 방식으로 확장할 수도 있다. 8장에서 소개했던 이메일 프로그램을 이 프로젝트의 프로그램(extract_data.py)과 쉽게 합칠 수 있다. 태그 ID를 감지하면 자동으로 이메일을 생성하고 보낼 수 있다. 아이, 혹은 장애인이 도움을 구하기 위해 이메일을 보내는 상황을 가정하면 쓸모가 있을 것이다. 물론, 긴급 상황시에는 이보다는 119가 적절할 것이다. 사용자는 리더 근처에 태그를 갖다 대기만 하면 되고, 그러면 자동으로 LED가 점등되며 이메일이 송신될 것이다.

여러 가지로 응용할 수도 있다. 태그를 찍어서 홈 오토메이션을 활성화시킬 수도 있고, 관개 시설이나 풀장 세척, 차고 개폐, 홈 스파 작동 등도 가능하다.