

GPI controller (Auto scheduler) 개발 - 1

Openpyxl과 APScheduler 라이브러리를 이용한 프로그램

GPI controller는 라디오 송출기술 업무의 편의성을 높이고자 개발하게 되었다. 이는 생방송이 아닌 경우의 모든 파일을 방송 시간에 맞춰 자동으로 재생시키는 프로그램이다. 그 방법으로 APC의 라디오 송출 스케줄을 받아 DASO(오디오 파일 시스템) 단말의 Play List를 스케줄에 따라 자동으로 재생시키도록 하였다. 이와 관련된 내용 및 개발 과정들을 소개하겠다.

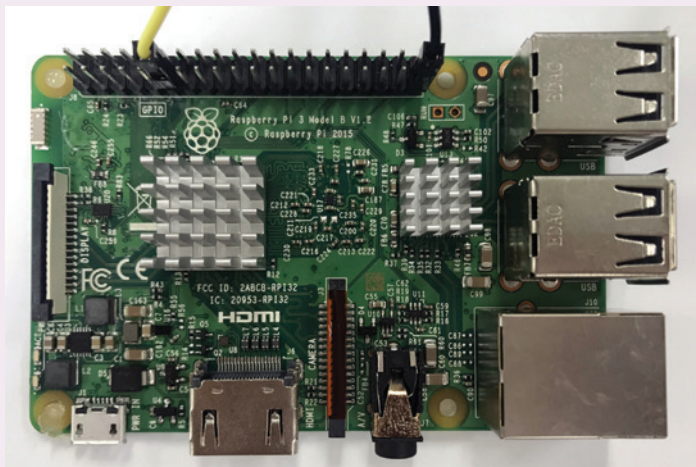
HW - Raspberry Pi 3 model B, Photo(opto) Coupler

SW - Python 3.6

Raspberry Pi (라즈베리 파이)



라즈베리 파이는 신용카드 크기의 싱글 보드 컴퓨터로 리눅스 커널 기반 운영 체제를 사용한다. GPU는 펌웨어 이미지를 통해 접근이 가능하며, 이 이미지는 SD 카드로부터 부팅할 때 GPU에 로드된다. 2016년에 출시를 발표한 세 번째 버전인 라즈베리 파이 3는 무선랜 및 블루투스 기능이 기본 내장되어 있으며, 이전 모델인 라즈베리 파이 2의 900MHz보다 개선된 1.2GHz의 속도를 구현한 브로드컴 BCM2837 64bit ARMv8 프로세서를 기반으로 한다. 또한, 전력관리 기능도 개선되어 최대 2.5A의 전력만을 사용하는 동시에 강력한 외부 USB기기를 더욱 많이 지원할 수 있다. 저전력 블루투스 (Bluetooth Low Energy)와 무선랜을 지원하게 된 라즈베리파이 3는 구입 즉시 IoT, 블루투스 헤드폰 또는 스피커, 와이파이 게이트웨이, 홈 클라우드 스토리지 등에 사용할 수 있다.



Raspberry Pi 3 model B

PYTHON language 및 Library (파이썬 언어 및 라이브러리)

[illegible]

Windows Defender SmartScreen prevented an unrecognized app from running because it might damage your PC. To help protect your privacy, Outlook prevented automatic download of some pictures in this message. To help protect your privacy, Outlook prevented automatic download of some pictures in this message.

PyCharm IDE window showing a Python script named 'go_controller_0501.py' with the following code:

```
def job_func(x):
    stat = "job start", str(round(x * 100))
    print(stat)
    for i in range(1, 10):
        l.append(i)
        start = "job start", str(round(x * 100))
        print(i)
        print(start)
    for i in range(1, 10):
        sched.add_job(job_func, "cron", hour=int(l[0][1]), minute=int(l[0][1]), second=int(l[0][1]) > 5, cron=D)
        print(i, l[1], l[4])
    else:
        if int(l[0][1]) < 10:
            sched.add_job(job_func, "cron", hour=int(l[0][1]), minute=int(l[0][1]) + 1, second=int(l[0][1]) > 5, cron=D)
            print(i, l[1], l[4])
        else:
            if int(l[0][1]) < 10:
                sched.add_job(job_func, "cron", hour=int(l[0][1]), minute=int(l[0][1]) + 1, second=int(l[0][1]) > 5, cron=D)
                print(0, l[1], l[4])
            else:
                sched.add_job(job_func, "cron", hour=int(l[0][1]) + 1, minute=int(l[0][1]) + 1, second=int(l[0][1]) > 5, cron=D)
                print(0, l[1], l[4])
    sched.start()

for i in range(1, 10):
```

Algorithm 소개

1. APC 송출 스케줄 파일(엑셀 파일) Load

D열이 생방송이 아닌 경우, A열의 시간 값과 C열의 프로그램명을 가져온다.

2. 시간 정보를 RAM memory에 저장

시간 값은 리스트 형식으로 t1에 저장하고, 프로그램명은 같은 형식으로 c1에 저장한다.

3. 스케줄러에 수행 작업 저장

리스트 형식의 시간 값을 스트링 형식으로 변환한 뒤 시, 분, 초로 나누어 스케줄러에 맞는 형식으로 작업을 걸어 준다. 정지 신호를 먼저 발생하므로 지정된 시간보다 5초 앞당겼다.

4. 프로그램 송출 시간에 맞춰 재생 및 정지 신호 발생

작업에는 정지신호를 먼저 발생하고 timer를 이용하여 5초 뒤에 시작 신호를 발생시켰다. 이때, 실제 신호 발생 시각을 함께 나타내어 제시간에 정확히 신호가 발생하였는지를 확인하였다.

PYTHON(파이썬) 언어

이번 개발은 파이썬 언어를 이용하여 프로그래밍을 했다. 파이썬 언어는 고급 프로그래밍 언어로, 2.0과 3.0 두 가지 버전이 존재한다. 파이썬 2.0은 2000년 10월 16일 배포되었고, 많은 기능이 추가되었다. 파이썬의 3.0 버전의 최종판이 긴 테스트를 거쳐 2008년 12월 3일자로 발표되었다. 2.x대 버전의 파이썬과 하위호환성이 없다는 것이 가장 큰 특징이다. 파이썬 3.0의 주요 기능 다수가 이전 버전과 호환되게 2.6과 2.7 버전에도 반영되기도 하였다. 현재 2.x대 버전은 2.7까지 개발되어 있고, 3.x대 버전은 3.8이 개발 중에 있다. 이번 개발에는 파이썬 3.6을 사용하였다.

파이썬은 다양한 플랫폼에서 쓸 수 있고, 라이브러리(모듈)가 풍부하다. 가장 낮은 수준의 라이브러리까지 포함하면 너무 많아서 감당하기 쉽지 않으므로, 최근 파이썬 패키지 인덱스, 곧 PyPI(Python Packages Index)로 불리는 라이브러리의 저장소(repository)를 관리하는 공식 기구를 새롭게 도입하게 되었다. 2018년 1월 기준으로 파이썬 패키지 인덱스는 125,762개의 다양한 기능을 가진 패키지를 관리하고 있다. 웹 사이트 서버를 구현하려고 하면 Python Web Framework로 Django, Flask, Bottle 등의 라이브러리가 있다. 기계학습 알고리즘을 쓰고 싶다면 python machine learning으로 scikit-learn이 있다. OpenCV를 이용하면 얼굴인식을 코드 몇 줄로 할 수도 있고, 기본적으로 설치되는 모듈인 tkinter 모듈을 이용하면 간단한 GUI 프로그래밍을 할 수 있다. PyGame으로는 게임도 만들 수 있다. 이번에 사용한 라이브러리는 Openpyxl, APScheduler 등이다.

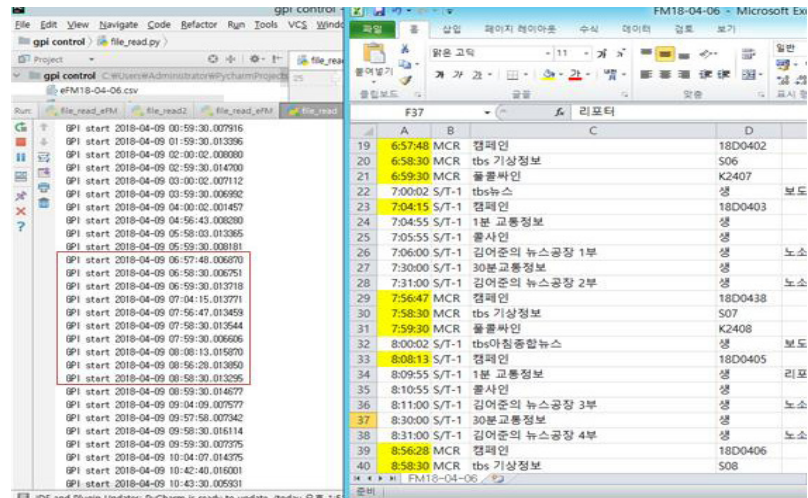
1. Openpyxl은 엑셀 2010의 xlsx / xlsm / xltm 등의 파일을 읽고 쓰는 데 사용하는 라이브러리이다. 엑셀 문서를 여는 데 필요한 메서드는 load_workbook()이다. 엑셀 시트에 있는 셀에 접근하려면 해당 시트를 검색한 다음 셀의 위치(좌표)를 지정하기만 하면 된다. 엑셀 문서에서 A열 2행, 즉 A2에 접근하고 싶다고 가정한다면 이를 다음과 같이 구현할 수 있다.

```
sheet = excel_document.get_sheet_by_name('Sheet1')
print(sheet['A2'].value)
```

이 라이브러리를 이용하여 엑셀 파일로 만들어진 방송 스케줄 정보에서 생방송을 제외한 스케줄의 시간 정보를 가져오도록 했다.

2. APScheduler는 작업을 자동실행 시키도록 해주는 라이브러리로 종류가 7가지 정도이며, 그중에서 BackgroundScheduler와 BlockingScheduler의 두 가지의 차이를 살펴보면, BackgroundScheduler는 다수 JOB 수행 시 사용을 하고, BlockingScheduler는 단일 JOB 수행 시 사용을 한다. 이번 개발에서는 BlockingScheduler를 사용하였고, 3가지 수행방식인 cron(cron 표현식으로

python code를 수행), date(특정 날짜에 python code를 수행), interval(일정 주기로 python code를 수행) 중 cron 방식으로 스케줄을 수행했다.

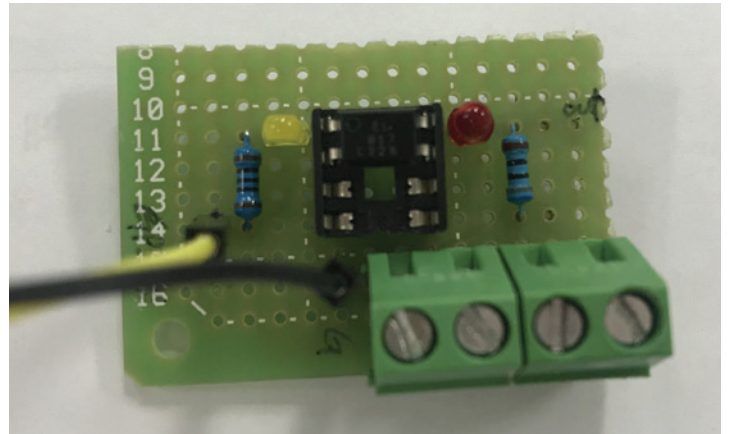


	A	B	C	D
19	6:57:48	MCR	김태연	18D0402
20	6:58:30	MCR	tbs 기상정보	S06
21	6:59:30	MCR	물류싸인	K2407
22	7:00:02	S/T-1	tbs뉴스	생
23	7:04:15	S/T-1	김태연	18D0403
24	7:04:55	S/T-1	1분 교통정보	생
25	7:05:55	S/T-1	물사인	생
26	7:06:00	S/T-1	김여준의 뉴스공장 1부	생
27	7:30:00	S/T-1	30분 교통정보	생
28	7:31:00	S/T-1	김여준의 뉴스공장 2부	생
29	7:56:47	MCR	김태연	18D0438
30	7:58:30	MCR	tbs 기상정보	S07
31	7:59:30	MCR	물류싸인	K2408
32	8:00:02	S/T-1	tbs야생동물뉴스	생
33	8:08:13	S/T-1	김태연	18D0405
34	8:09:55	S/T-1	1분 교통정보	생
35	8:10:55	S/T-1	물사인	리포
36	8:11:00	S/T-1	김여준의 뉴스공장 3부	생
37	8:30:00	S/T-1	30분 교통정보	생
38	8:31:00	S/T-1	김여준의 뉴스공장 4부	생
39	8:56:28	MCR	김태연	18D0406
40	8:58:30	MCR	tbs 기상정보	S08

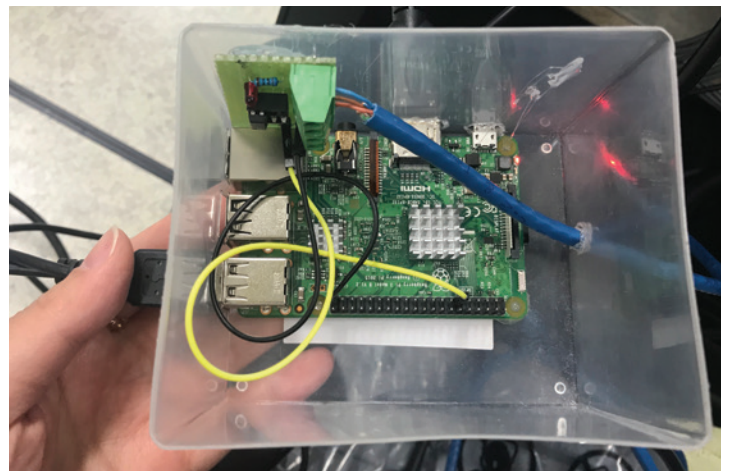
방송 스케줄 정보가 담긴 엑셀파일과 python code 실행 결과

Photo(opto) Coupler (포토커플러, 광 결합기)

포토커플러는 회로 간을 전기적으로 절연한 상태에서 전기 신호를 전달하는 목적으로 발광 소자와 수광 소자를 광학적으로 결합하여 하나의 패키지에 내장한 광 복합 소자를 말한다. 처음에 SW 개발 후 GPI 신호를 바로 DASO 단말에 연결했다. 그러나 이 경우 LOW(0) 값일 때, 미세하게 흐르는 전압도 HIGH(1)로 간주하여 계속 신호를 보내게 된다. 그리하여 GPI 신호를 Photo coupler(광 결합기) 회로에 연결하여 미세 전압을 차단하고, DASO 단말로 신호를 보냈다. 이 문제를 해결하느라 꽤 많은 시간이 걸렸다.



Photo(opto) Coupler 회로



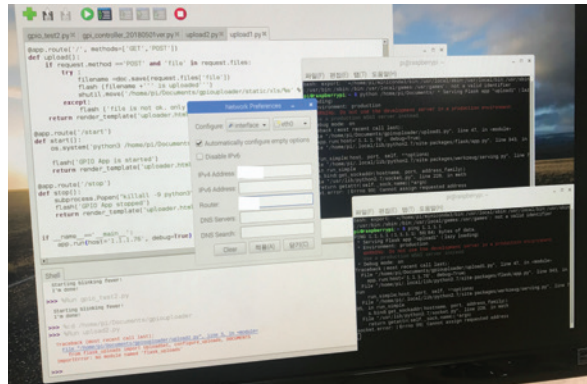
Raspberry Pi 3 모듈 및 Photo(opto) Coupler 회로 모듈

GPI controller 실제 설치

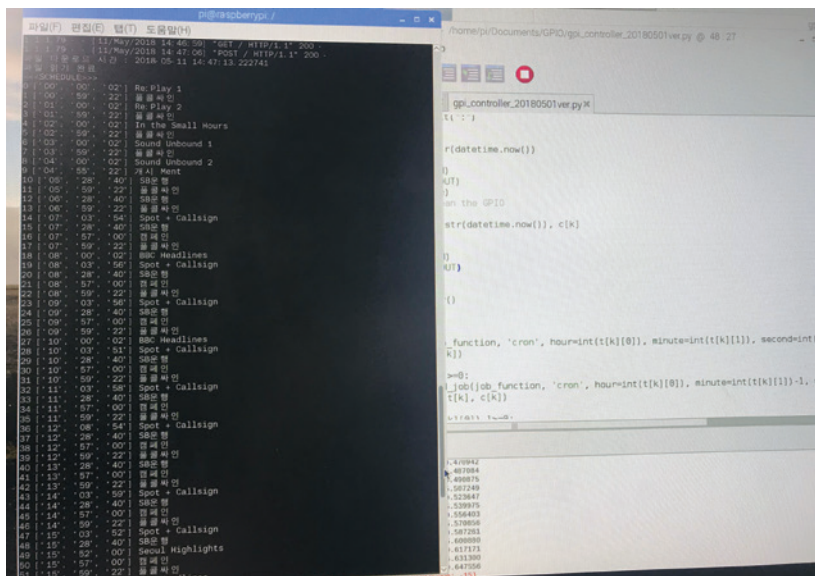
GPI 신호 HIGH(1) 값을 통해 PLAY와 LOW(0) 값을 통해 STOP을 한다. 라디오 주조정실에 GPI controller를 시범적으로 설치했다.



라디오 주조 Rack에 설치 모습



Raspberry Pi 네트워크 설정 화면



Raspberry Pi 3에서 실행한 화면 (방송 스케줄을 받아와 정해진 시간에 자동 GPI PLAY/STOP 신호 발생)

이번 호에서는 GPI controller 개발과정 및 관련 SW/HW에 대하여 살펴보았다. 다음 호의 주제 “Flask와 Pandas 라이브러리를 이용한 프로그램”에서는 이를 응용하여 web application 모듈을 만들어 프로그램 실행과 모니터링을 수월하게 하는 알고리즘에 대하여 살펴보겠다. ☺