

5-6주차 실습 보고서

DHT11 데이터 Node-red 통신

과목명: 웹기반임베디드시스템
교수님: 류대현 교수님
작성날짜: 2020년 10월 12일
작성자: 전자소프트웨어학과 201910063 정윤미
목차
1. 실습방향 ----- 2
2. 실습
(1) NodeJS + Node-RED 설치 및 플로우 작성 ----- 2
(2) python코드와 Node-RED 통신 ----- 3
(3) Node-RED dashboard를 통한 센서값 UI 설정 ----- 4
(4) 실습결과 ----- 6
3. 문제점 및 해결
(1) Node-RED 실행 오류 ----- 7
(2) Node-RED 센서값 디버깅 예러 ----- 7
4. 소감 ----- 8

1. 실습방향

- 4주차의 진행사항을 바탕으로 Node-RED를 설치하고, 온습도 센서값을 출력해주겠습니다.

2. 실습

(1) NodeJs+ Node-RED 설치 및 플로우 작성

```
pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
curl: (3) Failed to convert -sL to ACE; string contains a disallowed character

% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           0         0     0           0          0      0      0     0
*****
*****
```

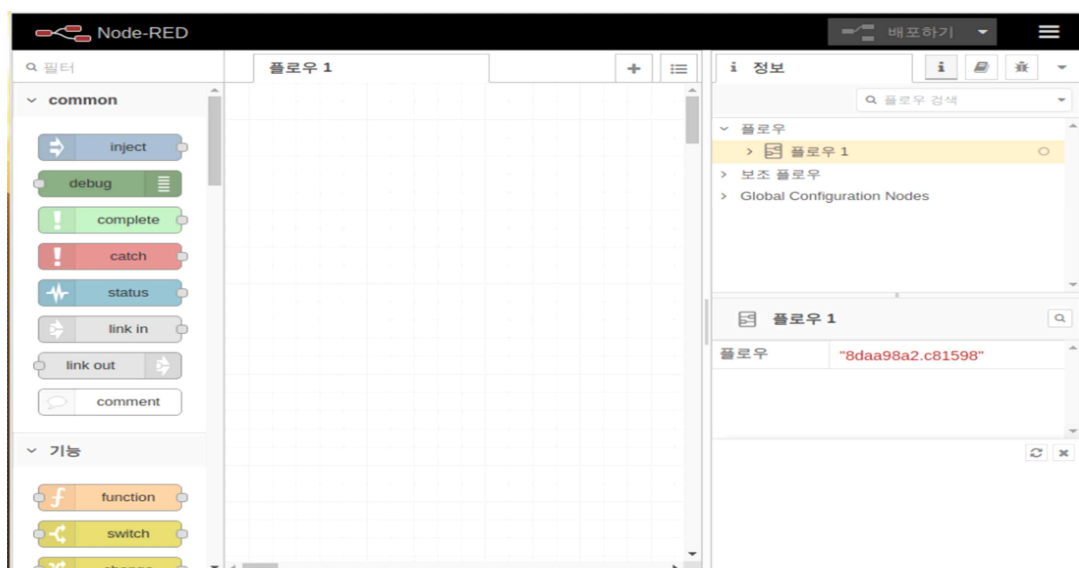
bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered 명령어를 사용하여 NodeJS와 Node-RED를 설치하였습니다.

```
pi@raspberrypi:~ $ node-red
12 Oct 12:59:58 - [info]

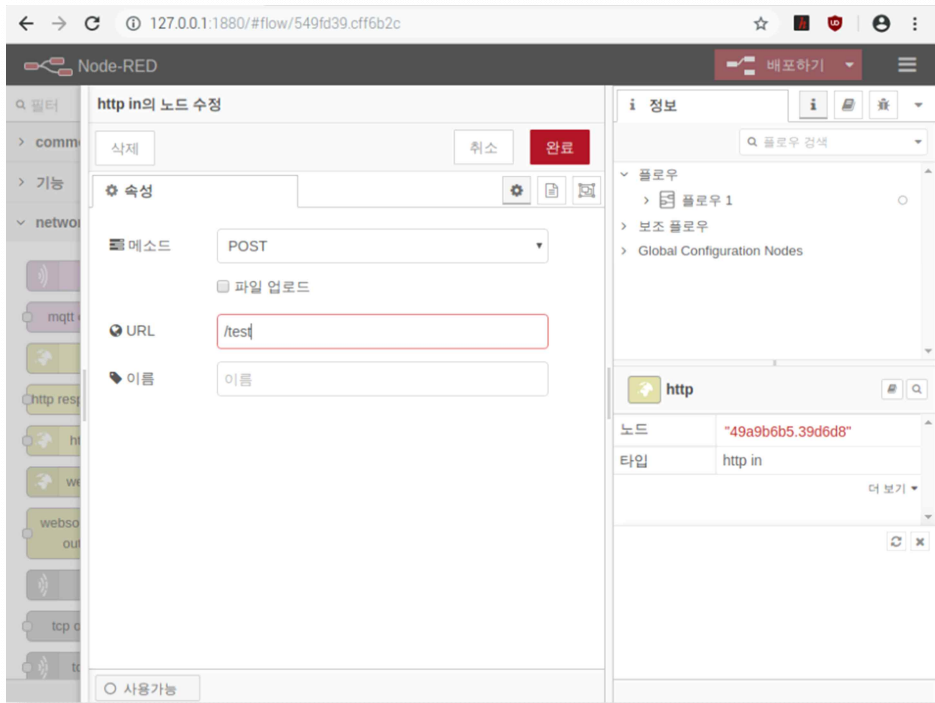
Node-RED에 오신것을 환영합니다.
=====

12 Oct 12:59:58 - [info] Node-RED 버전: v1.1.3
12 Oct 12:59:58 - [info] Node.js 버전: v10.21.0
12 Oct 12:59:58 - [info] Linux 5.4.51-v7l+ arm LE
12 Oct 13:00:00 - [info] 팔레트 노드 읽는 중
12 Oct 13:00:02 - [info] 설정 파일 : /home/pi/.node-red/settings.js
12 Oct 13:00:02 - [info] Context 저장소 : 'default' [module=memory]
12 Oct 13:00:02 - [info] 사용자 디렉토리: /home/pi/.node-red
```

이후 node-red 명령어를 사용하여 node-red 서버를 실행시켰습니다.



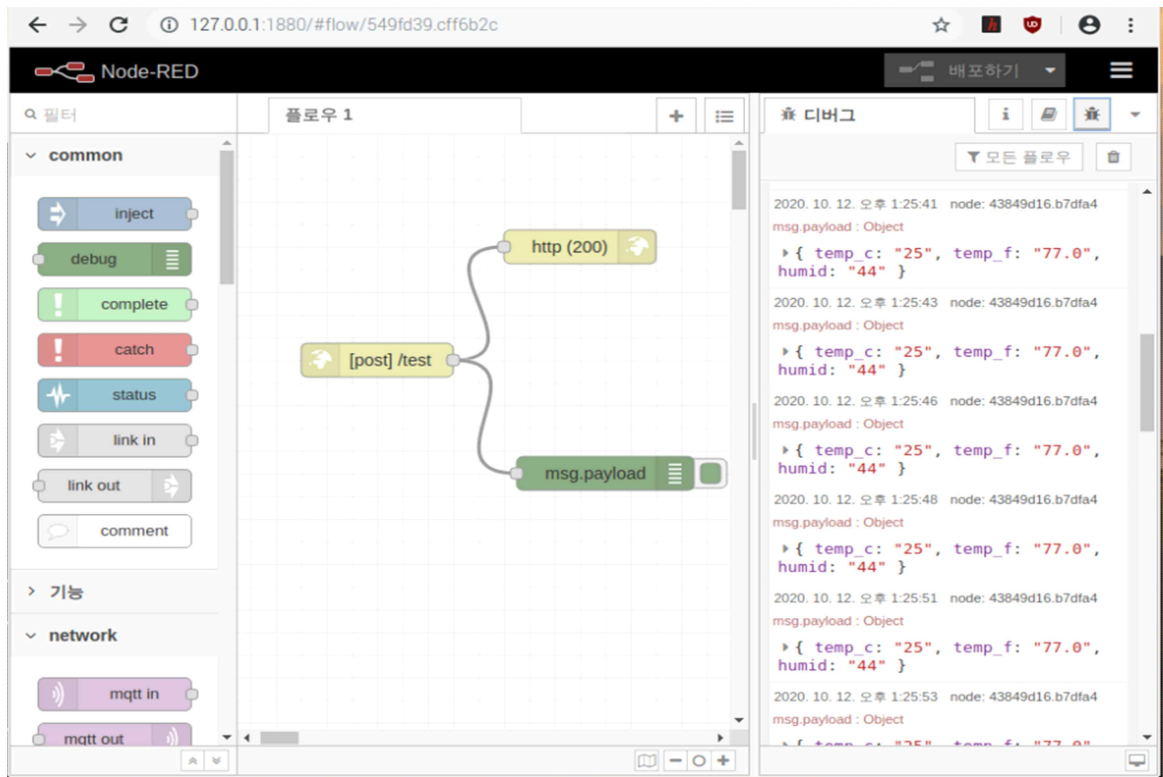
127.0.0.1:1880(로컬 호스트의 1880번 포트)로 접속한 결과는 위의 사진과 같습니다.



라즈베리파이4에서 받은 온습도 센서값을 처리해주기 위해, Node-RED에서 post방식으로 들어온 데이터를 받아주는 노드를 생성하였습니다. 127.0.0.1:1880/test로 들어온 데이터가 이 부분을 통해 처리됩니다.

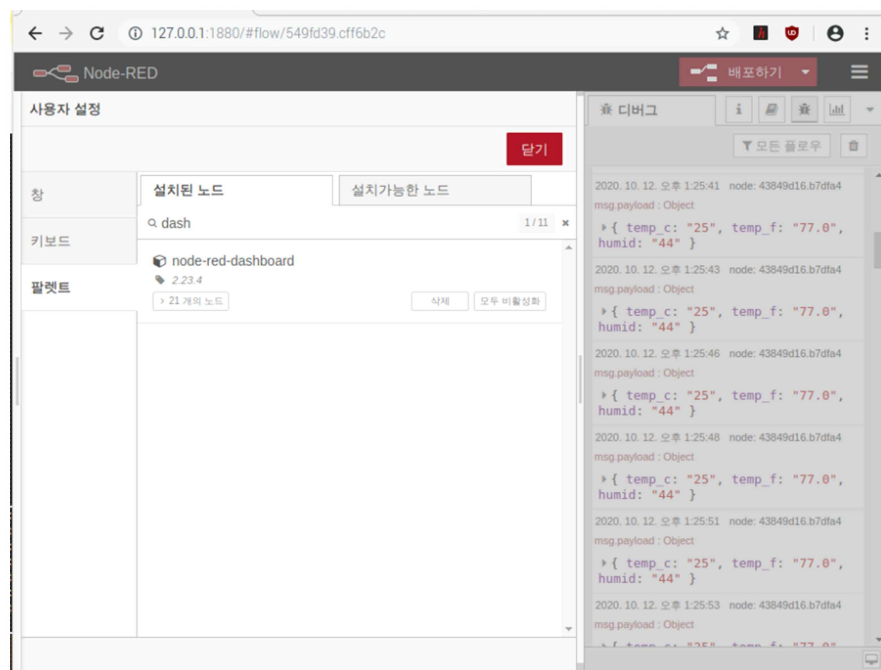


http 요청이 들어 올 경우, 응답을 해주어야 하기 때문에 http response로 상태코드 200을 반환하도록 설정해주었습니다. 또한 결과가 잘 들어오는지 확인할 수 있도록 디버깅을 연결하고 배포해주었습니다.



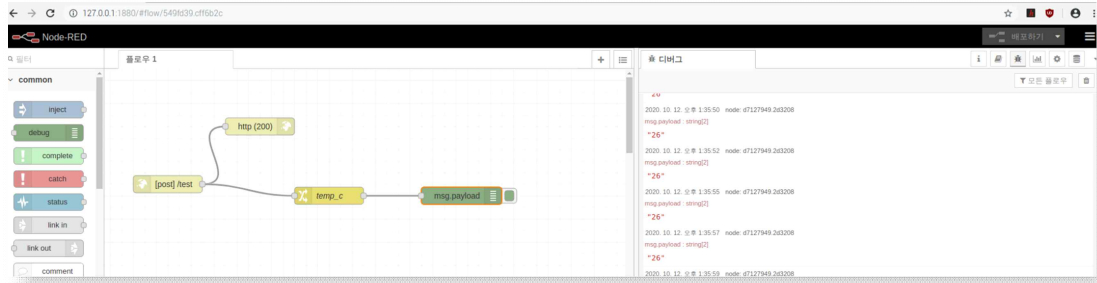
라즈베리파이4의 센서 값이 Node-RED 서버에 잘 들어온다는 사실을 알 수 있습니다.

(3) Node-RED dashboard를 통한 센서값 UI 설정

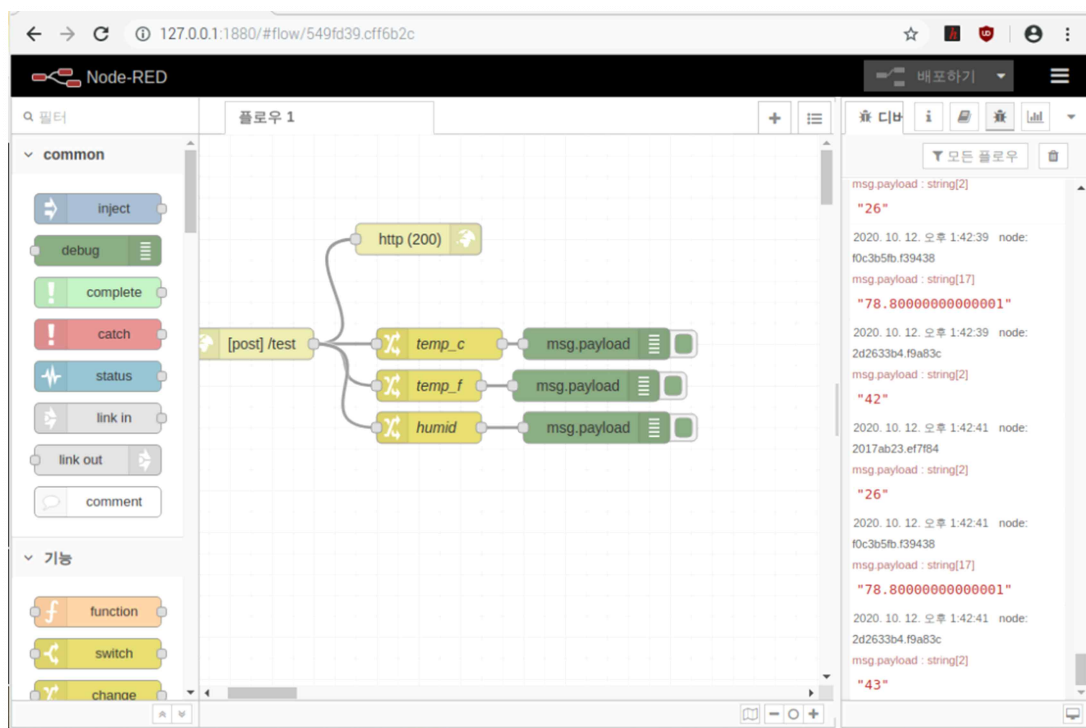


UI설정을 위해 팔레트 관리에서 설치 가능한 노드 중 node-red-dashboard를 설치해주었습니다

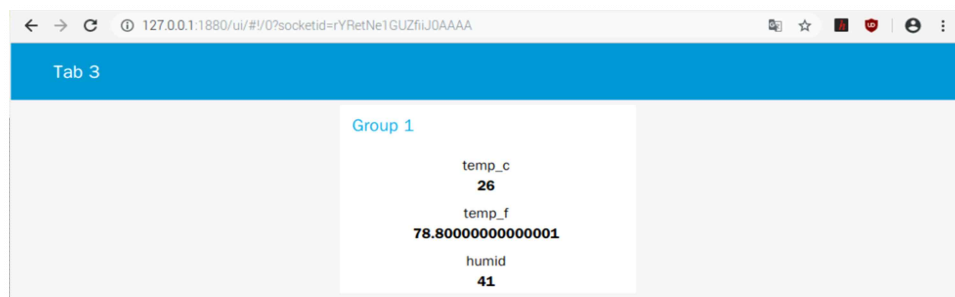
니다.



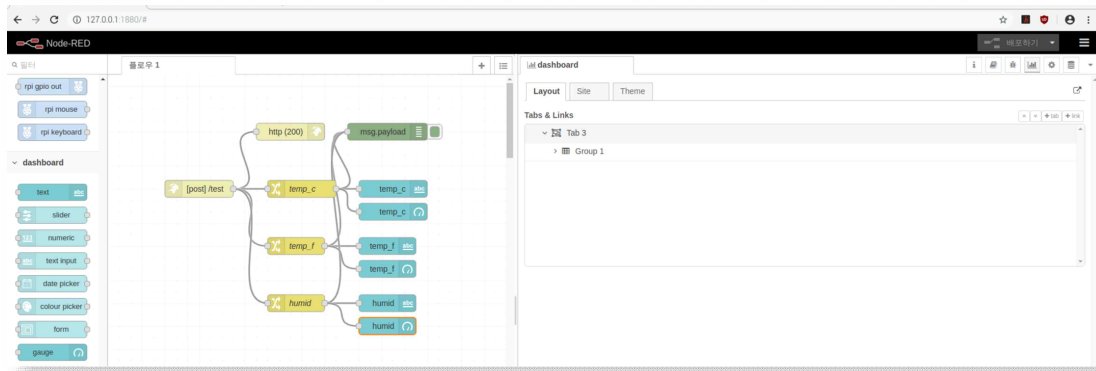
받은 데이터 중 섭씨온도만 분리하여 debug해주었고 위의 사진의 오른쪽 부분에서 제대로 출력됨을 알 수 있습니다.



같은 방식으로 섭씨온도, 화씨온도, 습도 데이터를 전부 분리하여 debug 해주었습니다.

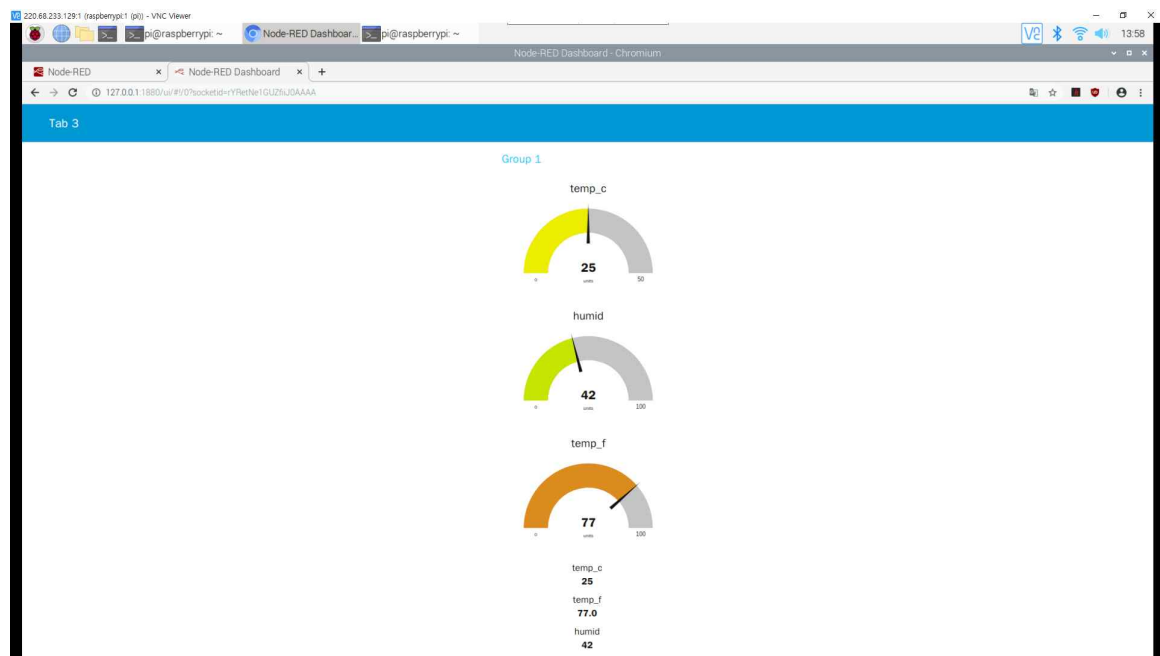


127.0.0.1:1880/ui 경로로 접속 시 결과입니다.



Gauge를 이용하여 계기판으로 UI를 표현할 수 있도록 하였습니다.

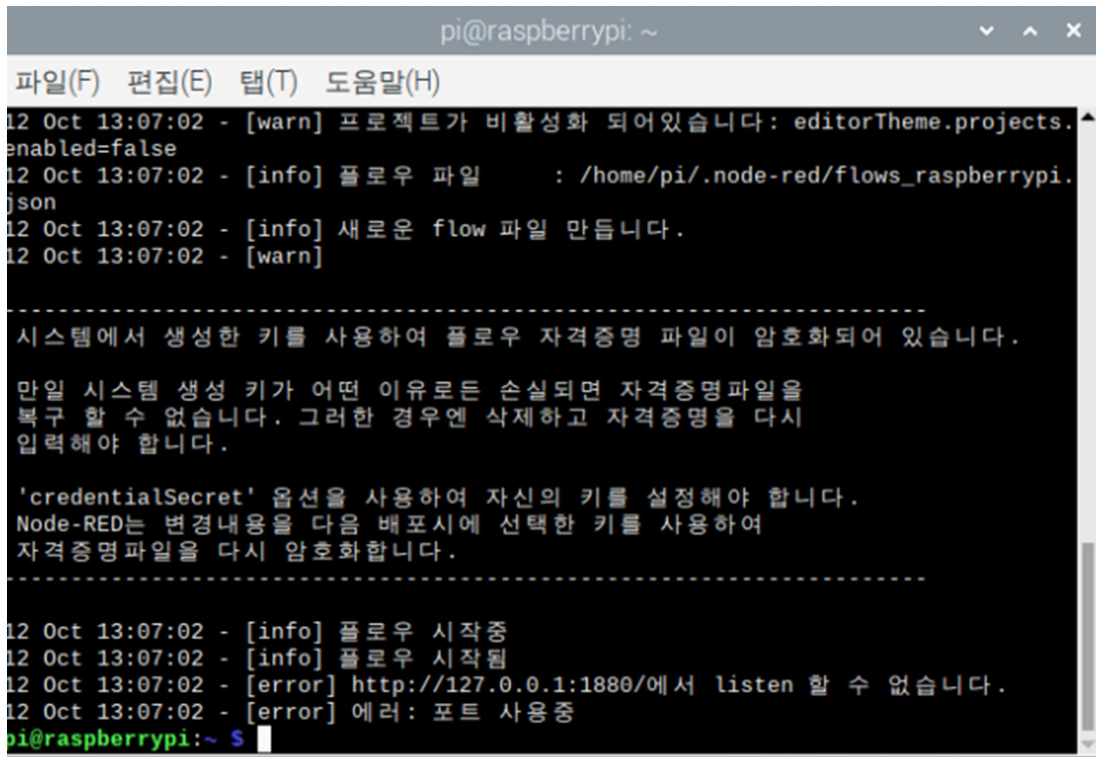
(4) 실행결과



실행결과 각각의 데이터가 계기판으로 잘 출력되었습니다.

3. 문제점 및 해결

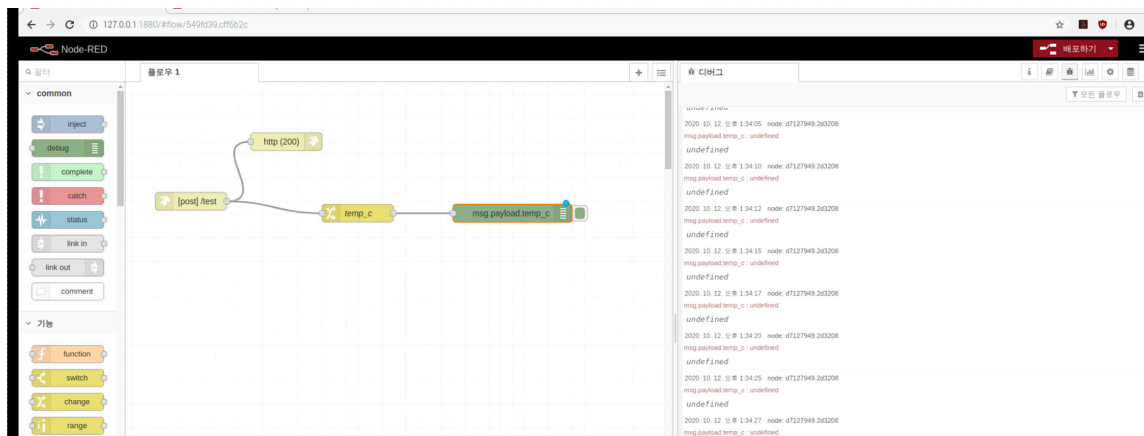
(1) Node-RED 실행 오류



```
pi@raspberrypi: ~  
파일(F) 편집(E) 탭(T) 도움말(H)  
12 Oct 13:07:02 - [warn] 프로젝트가 비활성화 되어있습니다: editorTheme.projects.  
enabled=false  
12 Oct 13:07:02 - [info] 플로우 파일      : /home/pi/.node-red/flows_raspberrypi.  
json  
12 Oct 13:07:02 - [info] 새로운 flow 파일 만듭니다.  
12 Oct 13:07:02 - [warn]  
  
-----  
시스템에서 생성한 키를 사용하여 플로우 자격증명 파일이 암호화되어 있습니다.  
  
만일 시스템 생성 키가 어떤 이유로든 손실되면 자격증명파일을  
복구 할 수 없습니다. 그러한 경우엔 삭제하고 자격증명을 다시  
입력해야 합니다.  
  
'credentialSecret' 옵션을 사용하여 자신의 키를 설정해야 합니다.  
Node-RED는 변경내용을 다음 배포시에 선택한 키를 사용하여  
자격증명파일을 다시 암호화합니다.  
-----  
  
12 Oct 13:07:02 - [info] 플로우 시작중  
12 Oct 13:07:02 - [info] 플로우 시작됨  
12 Oct 13:07:02 - [error] http://127.0.0.1:1880/에서 listen 할 수 없습니다.  
12 Oct 13:07:02 - [error] 에러: 포트 사용중  
pi@raspberrypi:~$
```

처음 node-red 명령어를 입력할 때 스크린 캡처를 위해 중지 후 다시 실행을 시켰었는데 이
미 포트를 사용 중이라는 예러가 떴습니다. 터미널을 닫고 다시 실행시켜 해결하였습니다.

(2) Node-RED 센서값 디버깅 예러



debug 블록을 처음 쓸 때, 대상을 msg.payload.temp_c로 변경하여 오류가 났습니다. 디버
깅 값이 undefind로 떴고, temp_c의 데이터가 그대로 넘어오기 때문에 msg.payload로 다시
변경해준 결과 해결되었습니다.

4. 소감

python code 내에서 import request를 하는 것만으로도 통신에 대한 준비가 된다는 부분이 신기하게 느껴졌고, 다른 언어에 비해 절차가 쉽다고 생각했습니다. 또한, node-red UI가 굉장히 편리하게 구성되어있다고 느꼈습니다. 다른 부분에 대한 응용도 시도해보고 싶습니다.