

7주차 실습 보고서

Node-red Maria DB 연결

과목명: 웹기반임베디드시스템
교수님: 류대현 교수님
작성날짜: 2020년 10월 19일
작성자: 전자소프트웨어학과 201910063 정윤미
목차
1. 실습방향 ----- 1
2. 실습
(1) MariaDB 설치 및 세팅 ----- 1
(2) Node-Red - MariaDB 연동 ----- 3
(3) 실습결과 ----- 6
3. 문제점 및 해결
(1) MariaDB install 에러 ----- 7
(2) NodeRed DB 커넥트 오류 ----- 7
4. 소감 ----- 7

1. 실습방향

- 5-6주차의 진행사항을 바탕으로 Node-RED와 MariaDB를 연결해줍니다.

2. 실습

(1) MariaDB 설치 및 세팅

```
pi@raspberrypi:~$ sudo apt-get install mariadb-server
```

패키지 목록을 읽는 중입니다 ... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다 ... 완료

sudo apt-get install mariadb-server 커멘트를 입력하여 mariadb를 설치합니다. mariaDB는 mysql기반 관계형 데이터베이스입니다.

```
pi@raspberrypi:~$ sudo service mysql start
pi@raspberrypi:~$ sudo service mysql status
● mariadb.service - MariaDB 10.3.23 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-10-19 12:25:30 KST; 2min 50s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 2239 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 31 (limit: 4915)
   CGroup: /system.slice/mariadb.service
           └─2239 /usr/sbin/mysqld

10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: Phase 6/7: Checking and upgrading tables
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: Running 'mysqlcheck' with connection arguments: --socket='/var/run/mysqld/mysqld.sock'
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: # Connecting to localhost...
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: # Disconnecting from localhost...
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: Processing databases
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: information_schema
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: performance_schema
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: Phase 7/7: Running 'FLUSH PRIVILEGES'
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2277]: OK
10% 19 12:25:32 raspberrypi /etc/mysql/debian-start[2335]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
lines 1-21/21 (END)
```

설치 후 sudo service mysql start 커멘트를 통해 실행결과 잘 설치되어 돌아가고 있음을 알 수 있습니다.

```
pi@raspberrypi:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 49
Server version: 10.3.23-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> set password for 'root'@'localhost'=password('1q2w3e4r');
Query OK, 0 rows affected, 1 warning (0.001 sec)

MariaDB [(none)]> CREATE user 'raspi_user'@'%' IDENTIFIED BY '1q2w3e4r';
Query OK, 0 rows affected (0.001 sec)
```

root계정으로 접속하여 set password for 'root'@'localhost'=password('')를 통해 root계정의 비밀번호를 생성하였고 CREATE user 명령어를 통해 어디에서든지 접근이 가능한 raspi_user계정을 생성하고 비밀번호를 설정해 주었습니다.

```
MariaDB [(none)]> CREATE DATABASE raspi_dht11 CHARACTER SET utf8 collate utf8_general_ci;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON raspi_dht11.* to 'raspi_user'@'%';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that correspon
MariaDB [(none)]> GRANT ALL PRIVILEGES ON raspi_dht11.* to 'raspi_user'@'%';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES
-> ;
Query OK, 0 rows affected (0.001 sec)
```

CREATE DATABASE raspi_dht11 CHARACTER SET utf8 collate utf8_general_ci;
 raspi_dht11라는 이름의 데이터베이스를 생성하고 데이터베이스의 기본 언어 인코딩을 utf-8으로 맞추어주었습니다. 또한, raspi_user 계정에 raspi_dht11 데이터베이스의 모든 권한을 설정했습니다. 환경설정을 변경하고 재시작 없이 사용하기 위해 FLUSH PRIVILEGES; 명령어를 입력하였습니다.

```
pi@raspberrypi:~ $ sudo mysql -u raspi_user -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 52
Server version: 10.3.23-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| raspi_dht11 |
+-----+
2 rows in set (0.001 sec)

MariaDB [(none)]> use raspi_dht11;
Database changed
MariaDB [raspi_dht11]> CREATE TABLE collect_data(sensor varchar(30) not null, collect_time datetime not null, temp_c float, temp_f float, humidity float);
Query OK, 0 rows affected (0.026 sec)

MariaDB [raspi_dht11]> show tables;
+-----+
| Tables_in_raspi_dht11 |
+-----+
| collect_data |
+-----+
1 row in set (0.001 sec)

MariaDB [raspi_dht11]> INSERT INTO collect_data VALUES('dht11','2020-10-09 14:15:14',27.4,77,50);
Query OK, 1 row affected (0.009 sec)

MariaDB [raspi_dht11]> SELECT * FROM collect_data;
+-----+-----+-----+-----+-----+
| sensor | collect_time | temp_c | temp_f | humidity |
+-----+-----+-----+-----+-----+
| dht11 | 2020-10-09 14:15:14 | 27.4 | 77 | 50 |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

raspi_user 계정으로 접속하여 raspi_dht11 데이터 베이스에 collect_data 테이블을 생성합니다. 이때, sensor 칼럼은 최대 30까지의 바이트를 사용하는 텍스트 문자열 데이터이고, collect_time은 시간을, temp_c와 temp_f, humidity는 소수점 아래 7자리까지를 표현하는 실수입니다. not null은 해당 필드에 NULL값을 저장할 수 없음을 의미합니다. insert 명령어를 사용하여 collect_data 테이블에 임의의 데이터를 넣어 확인했습니다.

```
MariaDB [raspi_dht11]> SELECT * FROM collect_data;
+-----+-----+-----+-----+-----+
| sensor | time | temp_c | temp_f | humidity |
+-----+-----+-----+-----+-----+
| dht11 | 2020-10-09 14:15:14 | 27.4 | 77 | 50 |
+-----+-----+-----+-----+-----+
1 row in set (0.002 sec)

MariaDB [raspi_dht11]> ALTER TABLE collect_data MODIFY COLUMN time datetime AFTER humidity;
Query OK, 0 rows affected (0.113 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [raspi_dht11]> SELECT * FROM collect_data;
+-----+-----+-----+-----+-----+
| sensor | temp_c | temp_f | humidity | time |
+-----+-----+-----+-----+-----+
| dht11 | 27.4 | 77 | 50 | 2020-10-09 14:15:14 |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

ALTER TABLE [테이블명] MODIFY COLUMN [칼럼명] [자료형] AFTER [다른칼럼];

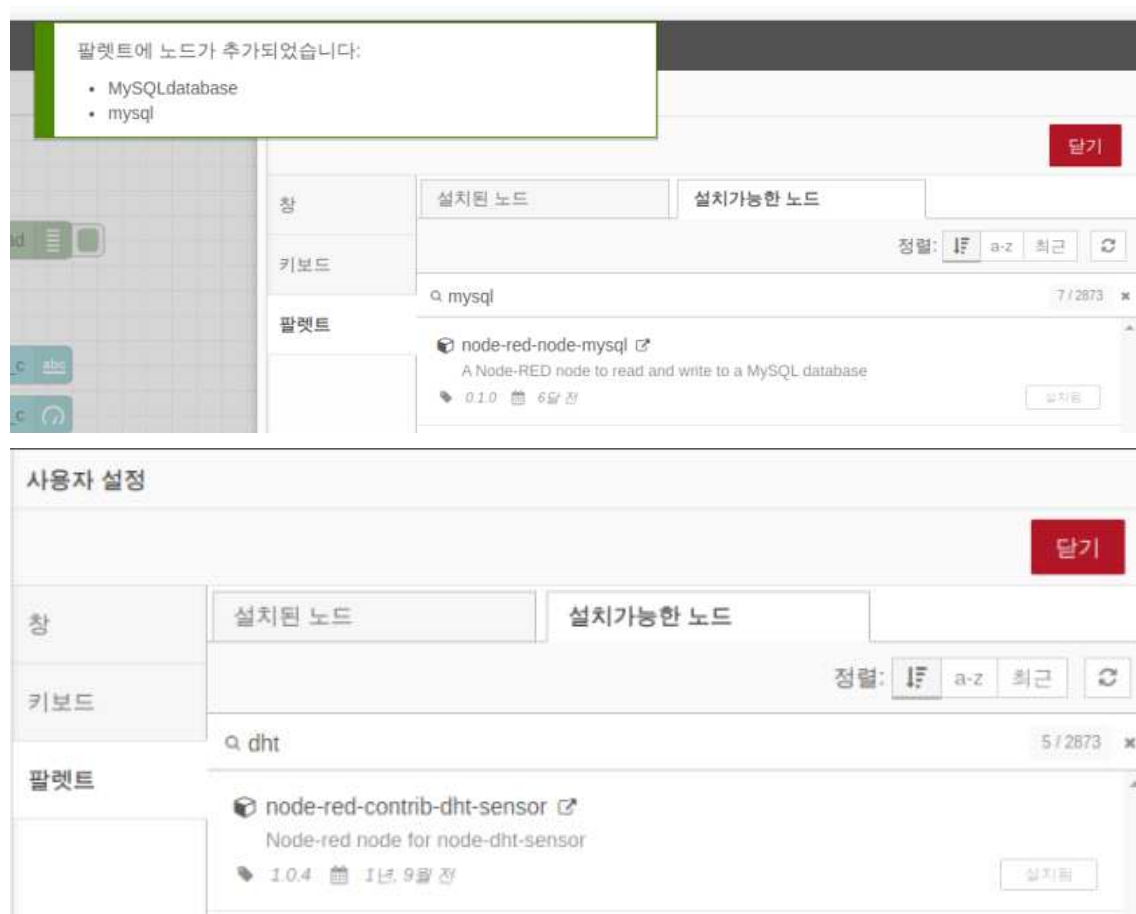
기존에 sensor 칼럼 오른쪽에 위치해있던 time 테이블을 humidity의 오른쪽으로 이동시켜 주었습니다.

```
MariaDB [raspi_dht11]> describe collect_data;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| sensor_id | varchar(30) | YES |     | NULL |       |
| temp_c    | float      | YES |     | NULL |       |
| temp_f    | float      | YES |     | NULL |       |
| humidity  | float      | YES |     | NULL |       |
| time      | datetime  | YES |     | NULL |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.003 sec)

MariaDB [raspi_dht11]> truncate collect_data;
Query OK, 0 rows affected (0.038 sec)
```

describe 명령어를 통해 최종적으로 만들어진 테이블의 속성을 확인하고 truncate를 사용하여 기존에 넣어둔 임의의 샘플 데이터를 지워주었습니다.

(2) NodeRed - MariaDB 연동

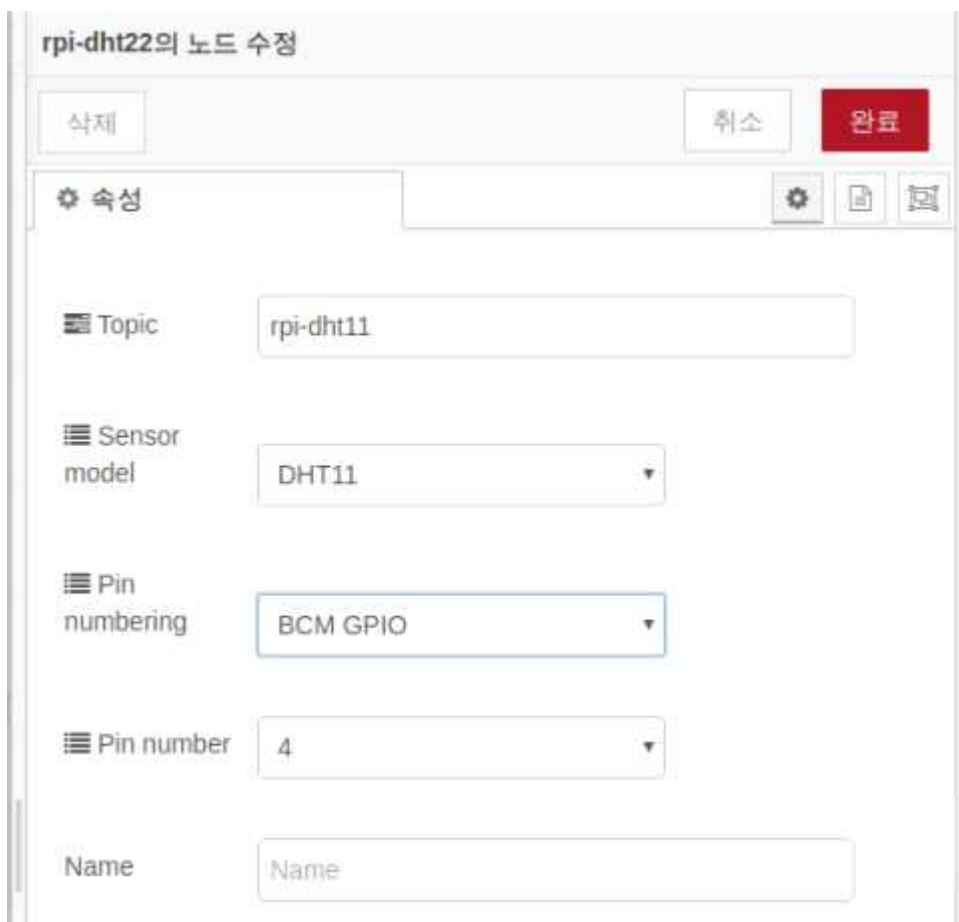


팔레트에 mysql 노드와 dht sensor 노드를 추가하였습니다.



A Node-RED configuration window for a delay node. It features a '+ 추가' (Add) button at the top left. Below it, there is a checkbox labeled 'Node-RED시작의' followed by a text input '0.1' and the text '초 후, 아래를 시행' (After 0.1 seconds, execute below). Further down, there is a '반복' (Repeat) button and a dropdown menu labeled '지정한 시간간격' (Specified time interval). At the bottom, there is a '시간각격' (Time interval) section with a numeric input '3', a small up/down arrow, and a unit dropdown menu set to '초' (seconds).

timestamp 노드의 설정입니다. 3초마다 실행되게 하였습니다.



A Node-RED configuration window titled 'rpi-dht22의 노드 수정' (Edit rpi-dht22 node). It has buttons for '삭제' (Delete), '취소' (Cancel), and '완료' (Done). Under the '속성' (Properties) tab, there are several settings: 'Topic' is set to 'rpi-dht11'; 'Sensor model' is a dropdown menu set to 'DHT11'; 'Pin numbering' is a dropdown menu set to 'BCM GPIO'; 'Pin number' is a dropdown menu set to '4'; and 'Name' is a text input field containing 'Name'.

dht 센서 노드의 설정입니다. 4번 핀에서 데이터를 받아오게 하였고 기본설정은 dht22로 맞추어있기 때문에 dht11로 변경하였습니다.

mysql의 노드 수정 > MySQLdatabase의 설정노드 추가

취소 추가

속성

Host 127.0.0.1

Port 3306

User respi_user

Password

Database raspi_dht11

Timezone

Name Name

DB 노드의 설정입니다. (1)에서 만들어둔 데이터베이스와 유저 계정의 비밀번호를 입력하여 추가했습니다.

function의 노드 수정

삭제 취소 완료

속성

이름 이름

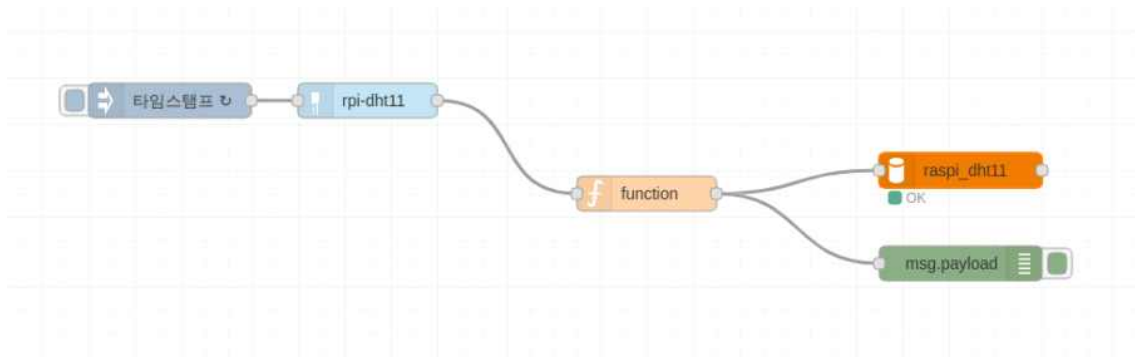
Setup 코드 Close

```

1 var sensor = "" + msg.topic+"";
2 // 센서ID 변수에 저장 할때 ""를 붙여주는 이유는 INSERT 문은 문자열을 집어넣을 때
3 // 'asd' 나 "asd" 식으로 씌워줄표나 줄 따옴표를 붙여줘야하기 때문이다
4 var timezoneOffset = new Date().getTimezoneOffset() * 60000;
5 var timezoneDate = new Date(Date.now() - timezoneOffset);
6 var time = timezoneDate.toISOString().slice(0, 19).replace('T', ' ');
7 // time의 toISOString() 구문을 사용해 시간을 가져오는데 앞의 2문장은 현재 시간으로 조정하기
8 // 위한 부분이다. https://from2020.tistory.com/12 자세하건 이 링크에 잘 설명되어 있다.
9
10 // DB의 datetime 타입에 맞추기 위한 과정으로 DB 컬럼타입을 varchar 로 고정하면
11 // 이 과정은 스킵해 도된다. var time = new Date(); 만 사용해도 될어진다.
12
13 var temp_c = msg.payload;
14 // 섭씨 데이터 변수에 저장
15 var temp_f = temp_c * (9/5) + 32;
16 // 화씨 데이터 변수에 저장
17 var humid = msg.humidity;
18 // 습도데이터 변수에 저장
19 msg.topic="INSERT INTO collect_data (`sensor_id`,`temp_c`,`temp_f`,`humidity`,`time`)";
20 // DB 에 명령을 집어넣을 때 꼭 topic 에 담아서 전송하라고 나와있음
21 // 파이썬에서 DB 연동할 때와 쿼리문은 같이 작성하면 되는데 걸림사이에 `` 붙여야한다.
22 msg.topic=msg.topic+VALUES("+sensor+",\`"+temp_c+"\`,`"+temp_f+"\`,`"+humid+"\`,`"+time+"`"+
23 // 값사이에 `,\` `를 넣는 이유는 문자열 형식을 INSERT 문에 넣어야할 경우 쌍따옴표나 줄 따옴표// 를 붙여서 넣
24 return msg;
25

```

function 노드의 설정입니다.



전체 구성도입니다.

(3) 실행 결과

```

pi@raspberrypi: ~
파일(F) 편집(E) 탭(T) 도움말(H)

MariaDB [raspi_dht11]> select * from collect_data
-> ;
+-----+-----+-----+-----+-----+
| sensor_id | temp_c | temp_f | humidity | time                |
+-----+-----+-----+-----+-----+
| rpi-dht11 | 25     | 77     | 43       | 2020-10-19 14:55:27 |
| rpi-dht11 | 25     | 77     | 43       | 2020-10-19 14:55:28 |
| rpi-dht11 | 25     | 77     | 43       | 2020-10-19 14:55:31 |
| rpi-dht11 | 25     | 77     | 42       | 2020-10-19 14:55:34 |
| rpi-dht11 | 25     | 77     | 43       | 2020-10-19 14:55:37 |
| rpi-dht11 | 25     | 77     | 43       | 2020-10-19 14:55:40 |
| rpi-dht11 | 25     | 77     | 44       | 2020-10-19 14:55:43 |
| rpi-dht11 | 25     | 77     | 43       | 2020-10-19 14:55:46 |
| rpi-dht11 | 25     | 77     | 43       | 2020-10-19 14:55:49 |
| rpi-dht11 | 25     | 77     | 43       | 2020-10-19 14:55:52 |
| rpi-dht11 | 25     | 77     | 42       | 2020-10-19 14:55:55 |
| rpi-dht11 | 25     | 77     | 42       | 2020-10-19 14:55:58 |
| rpi-dht11 | 25     | 77     | 42       | 2020-10-19 14:56:01 |
| rpi-dht11 | 25     | 77     | 42       | 2020-10-19 14:56:04 |
| rpi-dht11 | 25     | 77     | 42       | 2020-10-19 14:56:09 |
| rpi-dht11 | 25     | 77     | 42       | 2020-10-19 14:56:10 |
| rpi-dht11 | 25     | 77     | 42       | 2020-10-19 14:56:13 |
| rpi-dht11 | 25     | 77     | 41       | 2020-10-19 14:56:16 |
  
```

node-red에서 배포하기 버튼을 클릭한 뒤의 모습입니다. 터미널에서 확인한 결과 데이터가 잘 들어오고 있음을 알 수 있습니다.

3. 문제점 및 해결

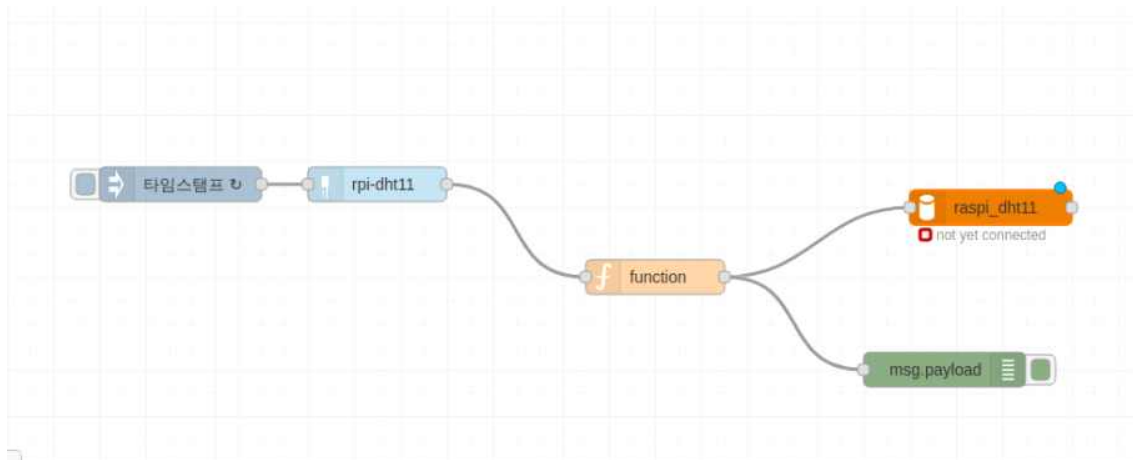
(1) MariaDB install 에러

```
pi@raspberrypi:~$ sudo apt-get install mariadb-server
E: dpkg가 중단되었습니다. 수동으로 'sudo dpkg --configure -a' 명령을 실행해 문제점을 바로잡으십시오.
pi@raspberrypi:~$ sudo apt-get install mariadb-server
E: dpkg가 중단되었습니다. 수동으로 'sudo dpkg --configure -a' 명령을 실행해 문제점을 바로잡으십시오.
pi@raspberrypi:~$ sudo dapt-get install mariadb-server
sudo: dapt-get: 명령이 없습니다
pi@raspberrypi:~$ sudo apt install mariadb-server
E: dpkg가 중단되었습니다. 수동으로 'sudo dpkg --configure -a' 명령을 실행해 문제점을 바로잡으십시오.
pi@raspberrypi:~$ sudo apt-get mariadb-server
E: 잘못된 작업 mariadb-server
pi@raspberrypi:~$

pi@raspberrypi:~$ sudo dpkg --configure -a
Processing triggers for mime-support (3.62) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.31.4-3) ...
Processing triggers for desktop-file-utils (0.23-4) ...
```

처음 mariaDB install 시 dpkg가 중단되는 에러가 떴고 아래 안내와 같이 sudo dpkg --configure -a 명령어를 실행하여 문제를 해결하였습니다.

(2) NodeRed DB 커넥트 에러



node-red에서 DB node를 설정할 때 db와 연결되지 않는 오류가 발생하였고, 확인결과, 데이터베이스의 이름 작성에서 오타가 났다는 사실을 알게 되었습니다. 오타를 수정한 뒤 에러가 해결되었습니다.

4. 소감

관계형 데이터베이스를 많이 접해보지 못하여 데이터베이스를 만드는 명령어를 작성할 때 이해가 되지 않는 부분이 많았지만 재밌었습니다. node-red와 mariadb를 연결하는 부분은 비교적 쉽게 따라갔지만 mariadb와 같은 관계형 데이터베이스에 대한 공부가 좀 더 필요할 것 같다는 생각이 들었습니다.