

# IoT기반 공정설비 모니터링 프로그래밍

03\_라즈베리파이 & 센서 프로그래밍

대한상공회의소 부산인력개발원

## 강 의 일 정

● IoT 기반 공정설비모니터링 프로그래밍(5d) ● 로봇 프로그래밍(AI+파이썬)(3d)

1회 차 8월 19일(월)	IoT 기반 공정설비 모니터링의 이해	IoT의 이해 / 센서 & 모니터링 시스템 / IoT네트워크	이론
2회 차 8월 20일(화)	라즈베리파이 & 센서 프로그래밍 1	Rpi의 이해 / Rpi 개발환경 구축 / 센서 프로그래밍(LED/온 · 습도/초음파...)	이론 /실습
3회 차 8월 21일(수)	라즈베리파이 & 센서 프로그래밍 2	Rpi 원격 개발환경 / 센서 응용 프로그래밍	이론 /실습
4회 차 8월 22일(목)	Node-Red 제어&모니터링 프로그래밍 1	Node-Red의 이해 / 개발환경 구축 / 센서 모니터링 및 제어	이론 /실습
5회 차 8월 26일(월)	Node-Red 제어&모니터링 프로그래밍 2	Dashboard 설계 / http통신 & DB / MQTT	이론 /실습
6회 차 8월 27일(화)	ESP32 마이크로 파이썬 로봇 프로그래밍 1	ESP32의 이해 / 마이크로 파이썬 개발환경 구축 / 로봇 프로그래밍(LEC/온 · 습도/초음파...)	이론 /실습
7회 차 8월 28일(수)	ESP32 마이크로 파이썬 로봇 프로그래밍 2	로봇 프로그래밍(DC모터/서보모터/LCD...) / 웹 서버 구축 및 제어 / AI 프로그래밍 이해	이론 /실습
8회 차 9월 4일(수)	프로젝트 & 평가	팀 프로젝트 및 평가	실습 /평가

## CONTENTS

01 Node-Red 둘러보기

02 Node-Red 기본 Node 실습

03 Node-Red 응용 Node 실습

04 Node-Red Dashboard 실습

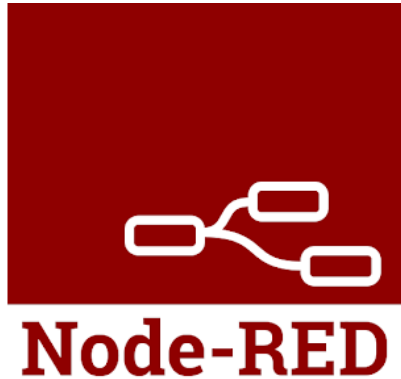
05 Casa OS 실습 / Pi-Hosted 실습

06 Umbrel OS 실습 / Cloudflare 실습

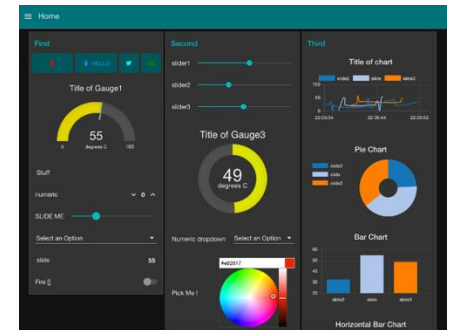
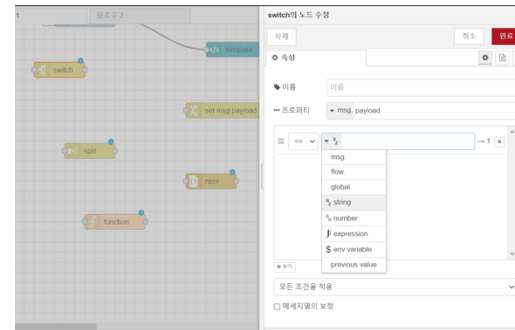
07 교재 4강/5강 실습

세계적인 첼리스트 로스트로포비치는  
기본 보잉을 하루도 빠지 않고 연습했다고 합니다.  
세계적인 대가도 가장 기본적인 연습을 게을리하지 않았습니니다.  
기본 코드를 매일 연습하는 습관은 여러분을 실력자로 인정받게  
해주는 지름길 입니다.

## Node-Red 둘러보기



- Node-RED는 플로우 기반의 하드웨어 제어 도구
- IoT기반의 플랫폼으로 활용(자체 IoT서버)
- 자체 대시보드 기능을 위젯과 함께 활용하며 단일 시스템 제어 가능
- 별도의 클라이언트 어플리케이션 없이 웹 브라우저로 편집 및 배포 가능
- Low코딩 기반으로 노드와 라인 연결을 통해 시스템 제어 가능
- 풍부하고 다양한 라이브러리를 활용하여 로직 구성 가능
- 플로우에 따라 구성 되므로 시각적 이해도가 높음



## Node-Red 둘러보기



Node-Red 기본화면

## Node-Red 둘러보기

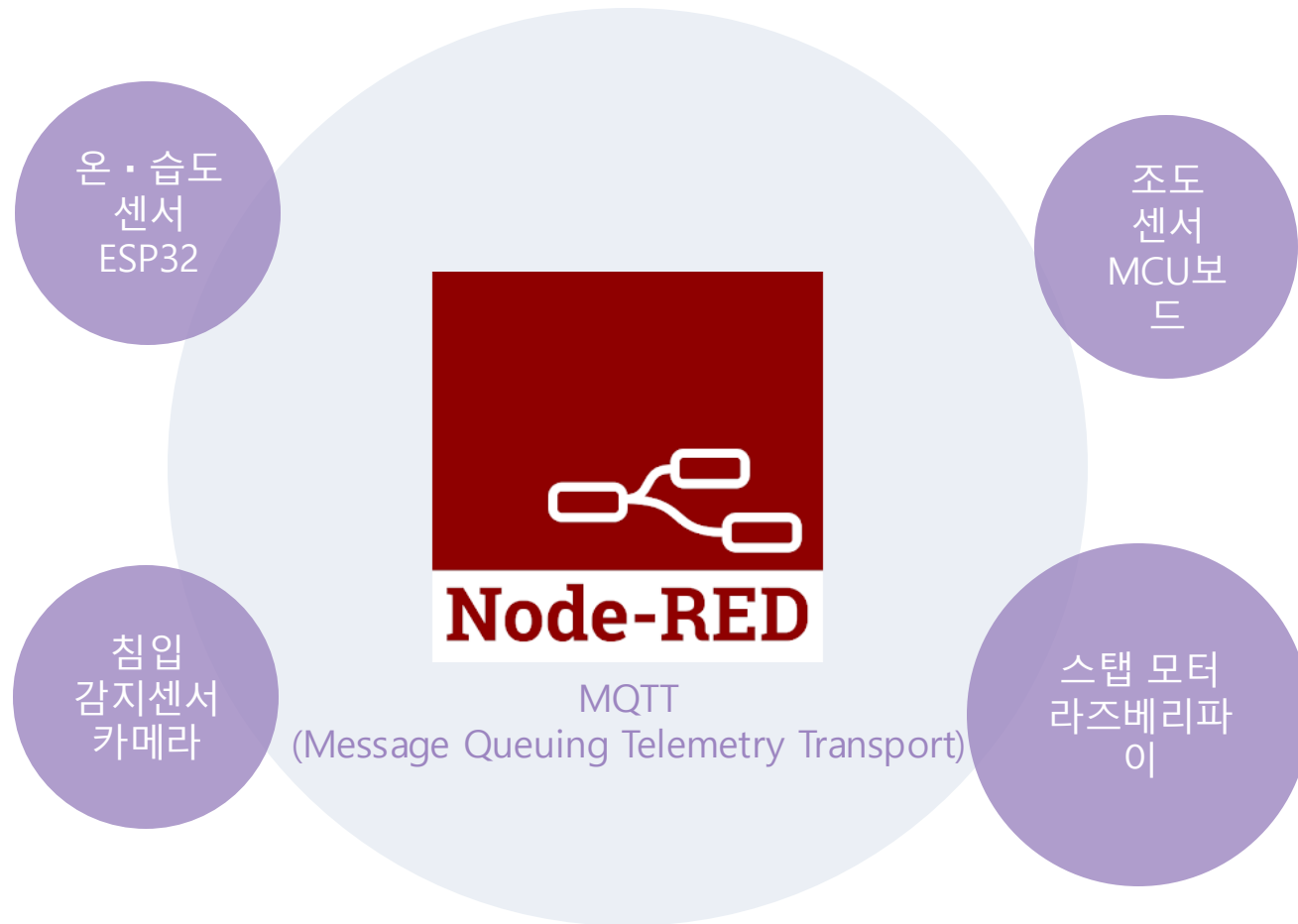
입력 값 변경

스텝 모터 작동(+90)

스텝 모터 작동(-90)

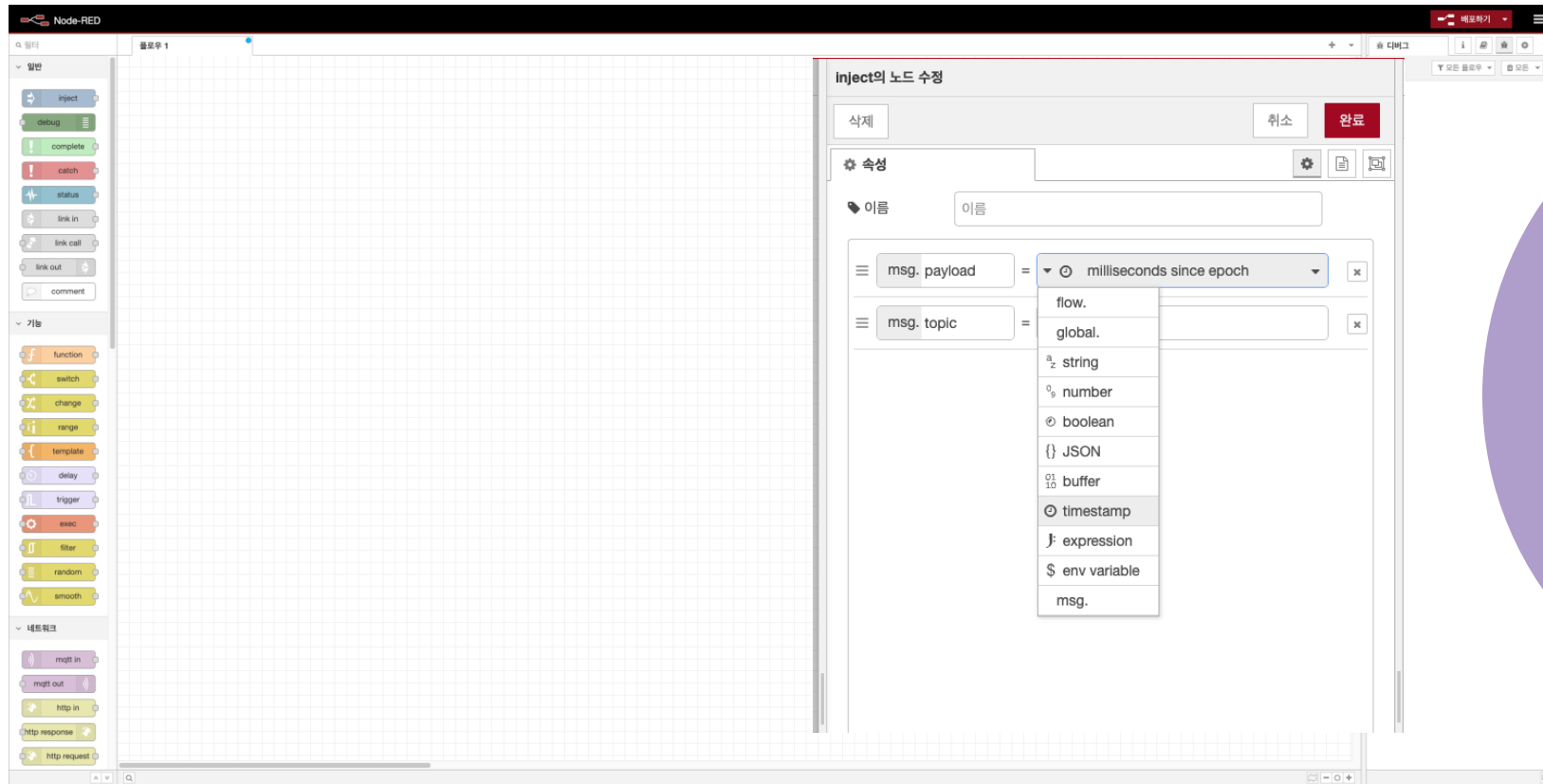
Flow 기반 설계

## Node-Red 둘러보기





## Node-Red 둘러보기



Drag & Drop 방식으로  
구현

하나의 노드에서  
다중 정보 전달

## Node-Red 둘러보기

rpi-dht22의 노드 수정

삭제 취소 완료

속성

Topic rpi-dht22

Sensor model DHT22

Pin numbering Physical pins (rev. 1)

Pin number 4

Name Name

rpi-gpio in의 노드 수정

삭제 취소 완료

속성

Pin

3.3V Power - 1	2 - 5V Power
SDA1 - GPIO02 - 3	4 - 5V Power
SCL1 - GPIO03 - 5	6 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

BCM GPIO

Resistor? none Debounce 25 mS

☐ Read initial state of pin on deploy/restart?

이름 이름

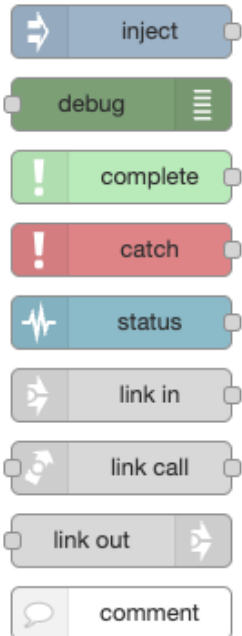
Pins in Use: 14

Tip: Only Digital Input is supported - input must be 0 or 1.

팔레트를 통해 다양한  
라이브러리를  
손쉽게 사용

## Node-Red 둘러보기

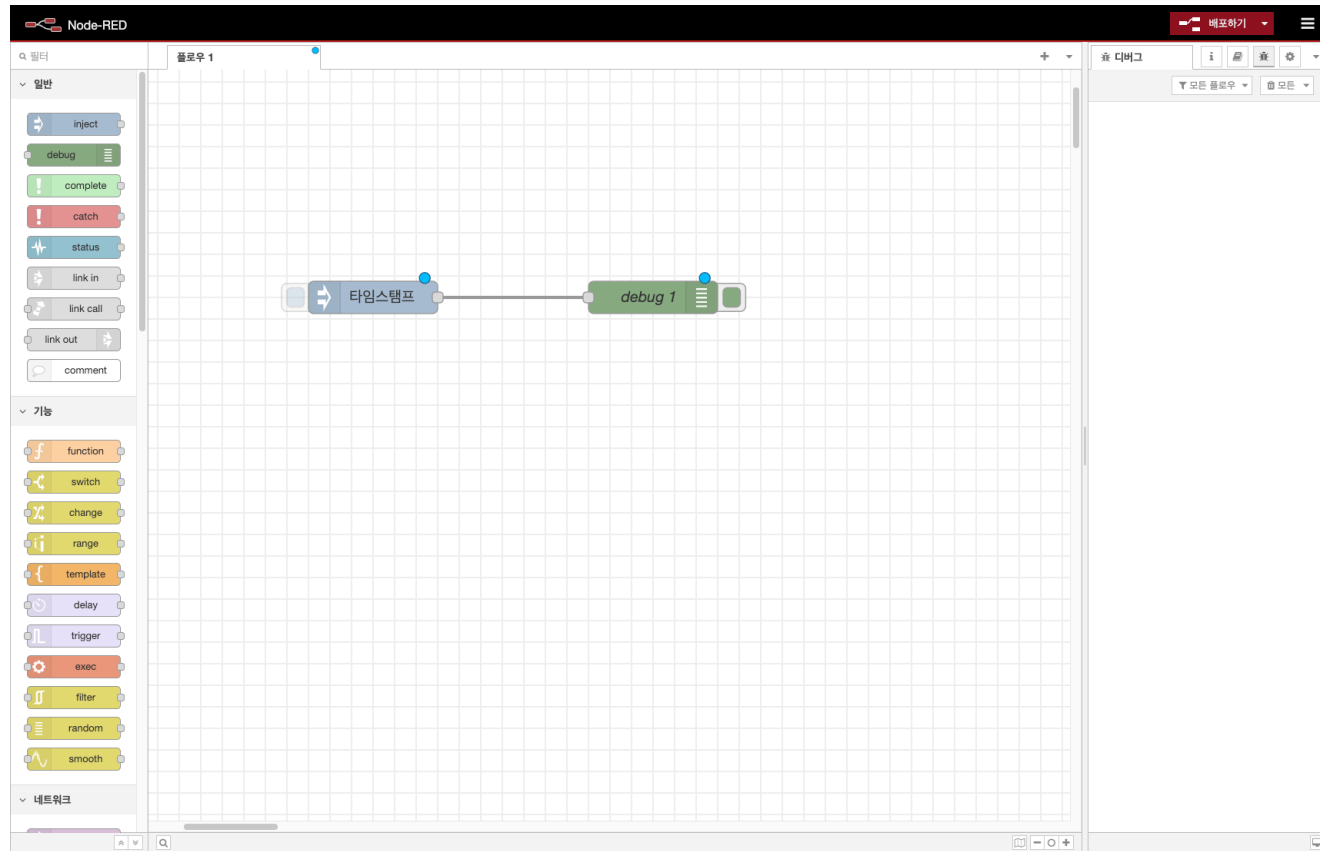
### ▼ 일반



동작이나 실행을 시키기 위한 입력 값  
동작이나 실행의 결과값 확인

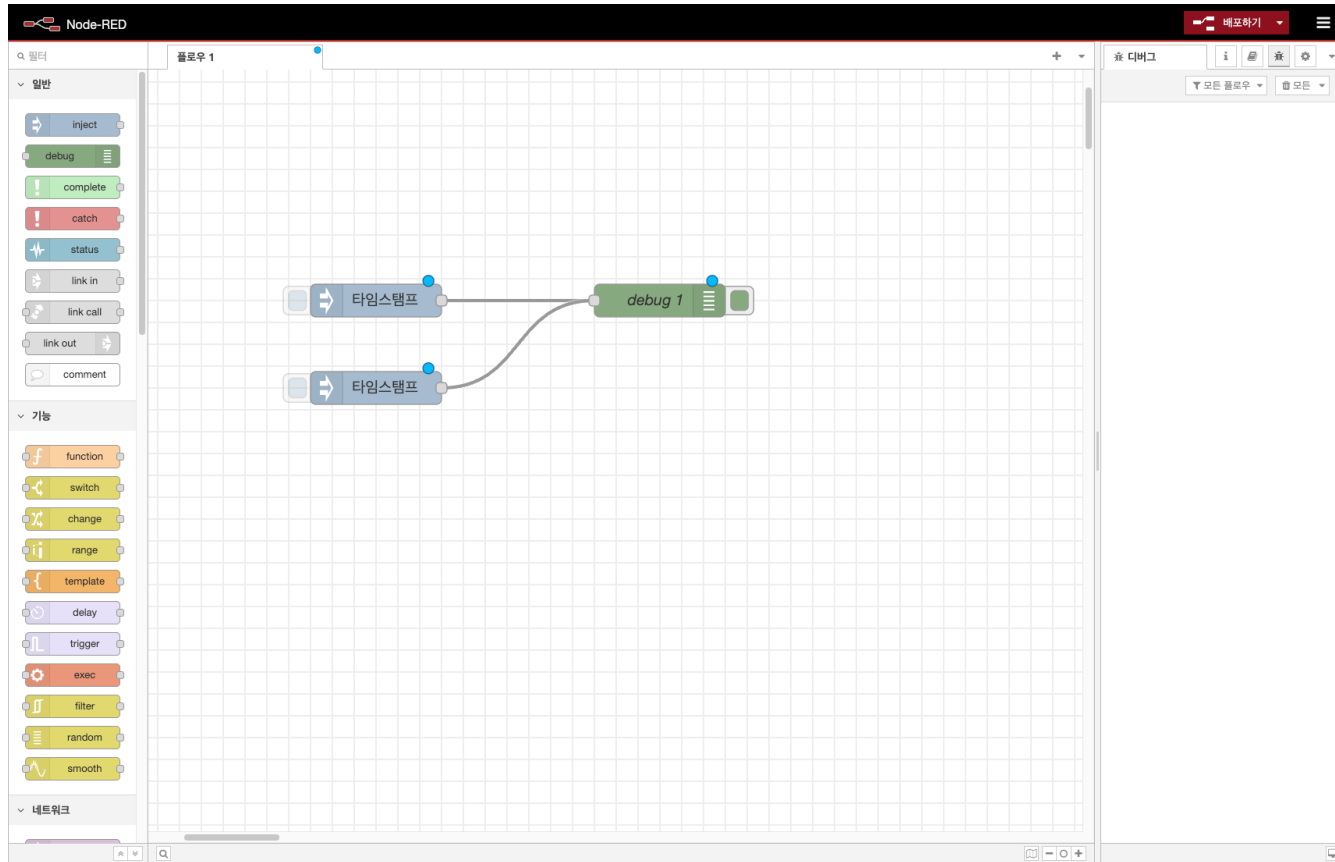
기본 실습  
인젝트와 디버그 사용

## Node-Red 둘러보기



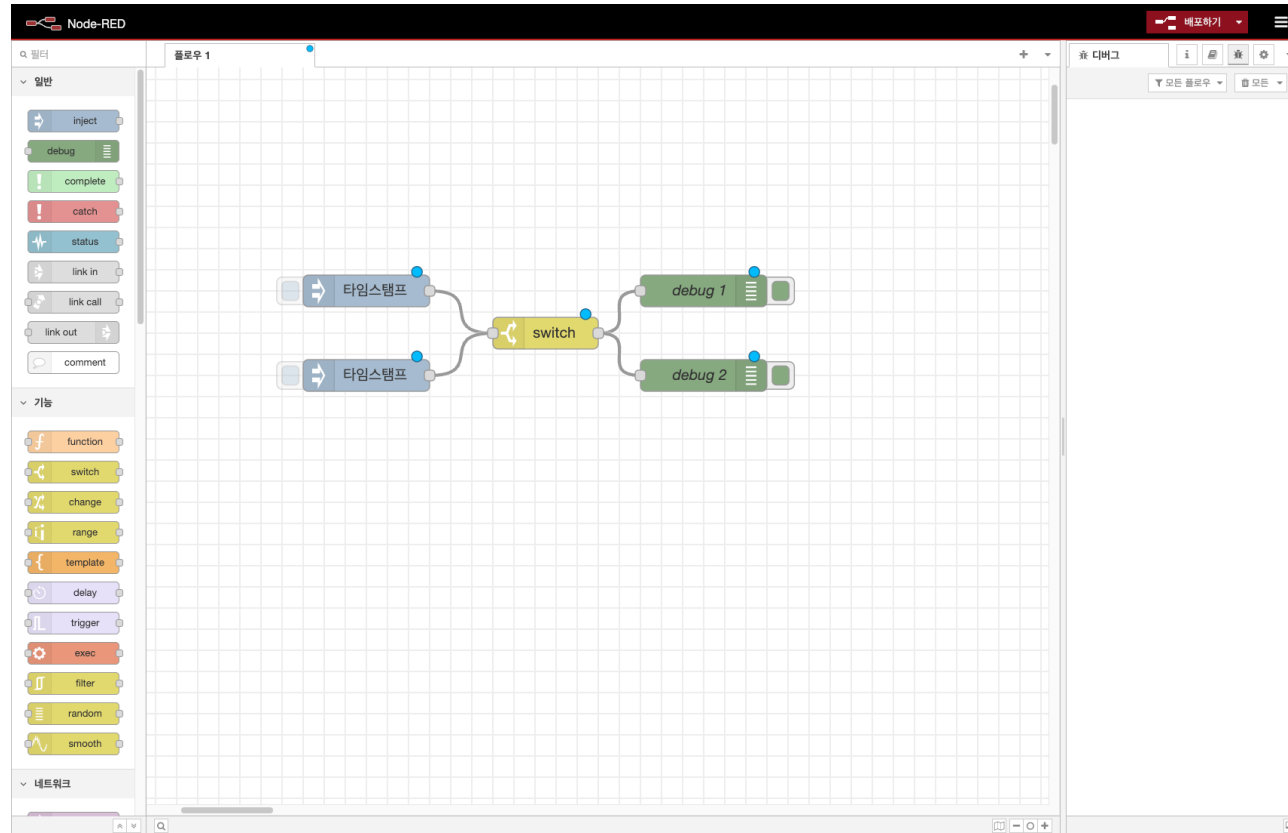
기본 실습  
인젝트와 디버그 사용

## Node-Red 둘러보기



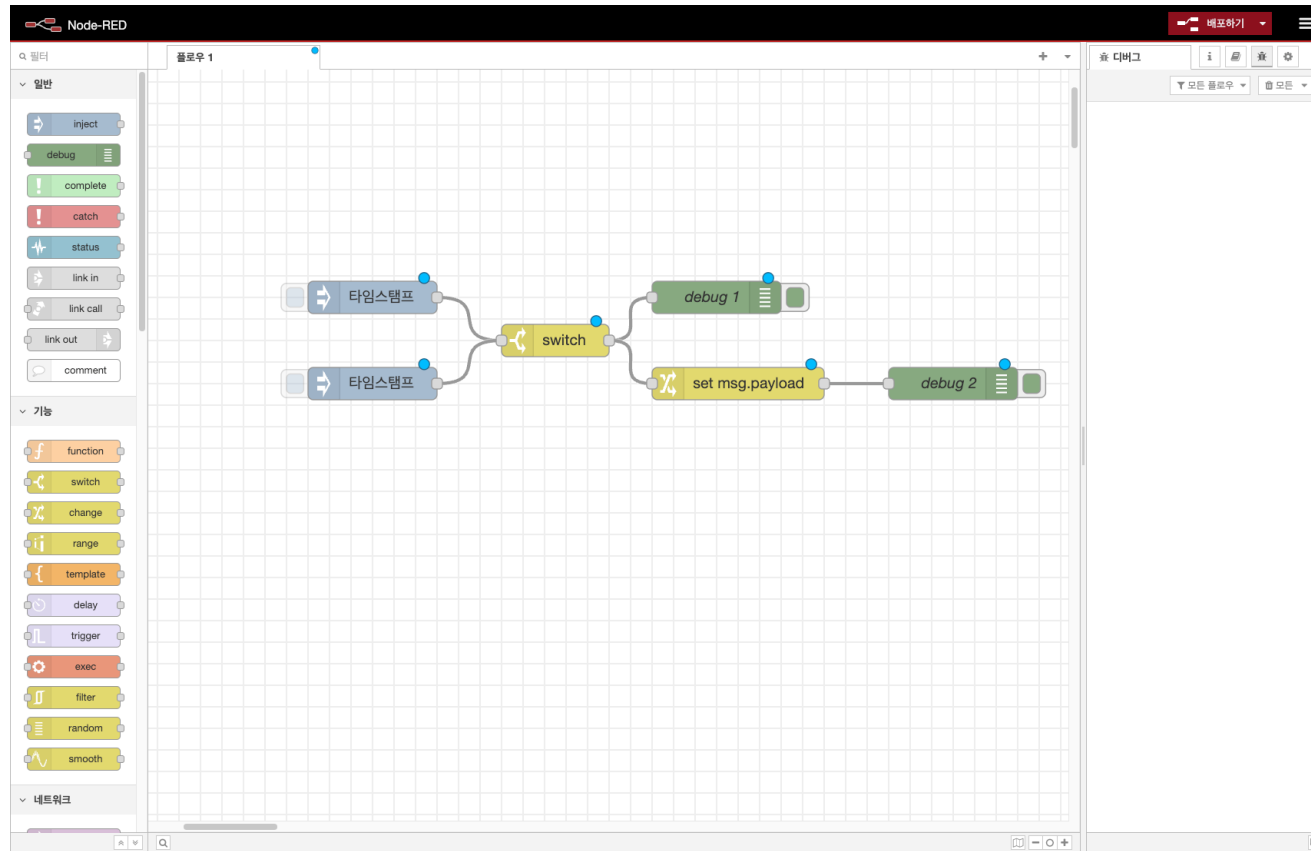
기본 실습  
다중 인젝트와 디버그  
사용

## Node-Red 둘러보기



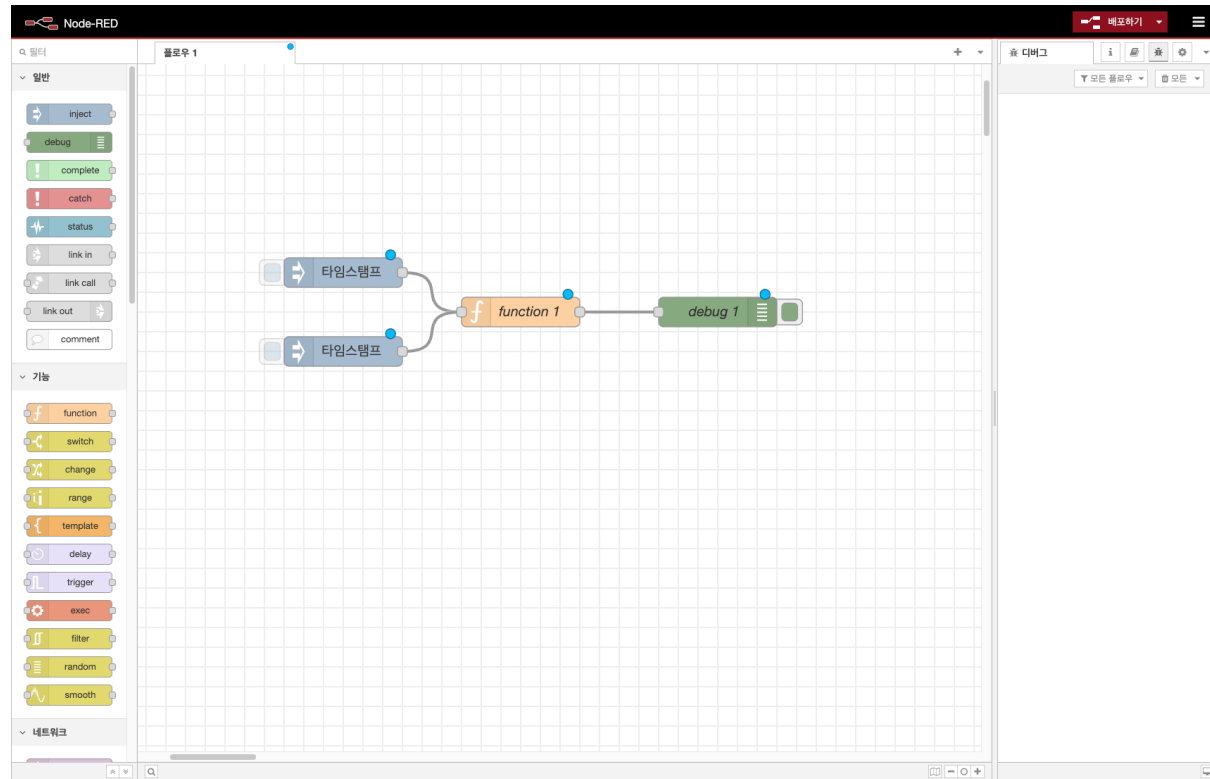
기본 실습  
스위치 활용

## Node-Red 둘러보기



기본 실습  
체인지 활용

## Node-Red 둘러보기



기본 실습  
함수 활용



## Node-Red 둘러보기

3v3 Power	1	2	5v Power
GPIO 2 (I2C1 SDA)	3	4	5v Power
GPIO 3 (I2C1 SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (UART TX)
Ground	9	10	GPIO 15 (UART RX)
GPIO 17	11	12	GPIO 18 (PCM CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3v3 Power	17	18	GPIO 24
GPIO 10 (SPI0 MOSI)	19	20	Ground
GPIO 9 (SPI0 MISO)	21	22	GPIO 25
GPIO 11 (SPI0 SCLK)	23	24	GPIO 8 (SPI0 CE0)
Ground	25	26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	28	GPIO 1 (EEPROM SCL)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM DIN)
Ground	39	40	GPIO 21 (PCM DOUT)



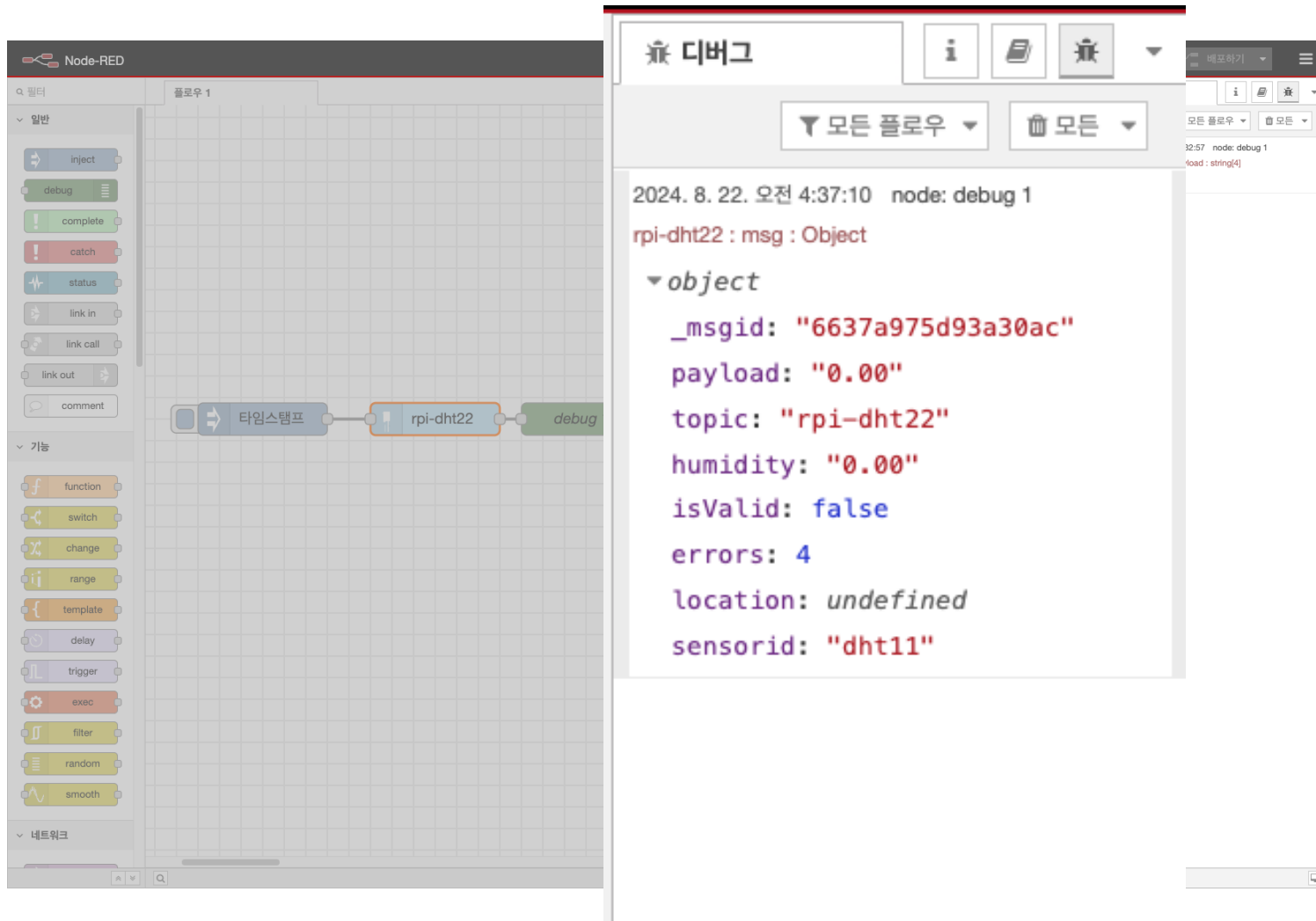
기본 실습  
GPIO 입력(DHT11)

## Node-Red 둘러보기

The screenshot displays the Node-RED web interface. On the left, the 'palette' contains various nodes categorized into '일반' (General), '기능' (Function), and '네트워크' (Network). The main workspace shows a workflow with three nodes: '타임스탬프' (Timestamp), 'rpi-dht22', and 'debug 1'. The 'rpi-dht22' node is selected, and its configuration panel is visible on the right. The configuration panel includes fields for 'Topic' (rpi-dht22), 'Sensor model' (DHT11), 'Pin numbering' (BCM GPIO), 'Pin number' (4), and 'Name' (Name). The 'debug 1' node's output is shown on the far right, displaying the message: '2024. 8. 22. 오전 4:32:57 node: debug 1 rpi-dht22 : msg.payload : string[4] "0.00"'. The interface also includes a search bar at the top left and a '배치하기' (Batch) button at the top right.

기본 실습  
GPIO 입력(DHT11)

## Node-Red 둘러보기



The screenshot shows the Node-RED web interface. On the left, the 'palette' contains various nodes categorized into '일반' (General), '기능' (Function), and '네트워크' (Network). The main workspace shows a flow with three nodes: '타임스탬프' (Timestamp), 'rpi-dht22', and 'debug'. The 'debug' node is selected, and its console is open on the right. The console displays the following information:

```
2024. 8. 22. 오전 4:37:10 node: debug 1
rpi-dht22 : msg : Object
  ▾ object
    _msgid: "6637a975d93a30ac"
    payload: "0.00"
    topic: "rpi-dht22"
    humidity: "0.00"
    isValid: false
    errors: 4
    location: undefined
    sensorid: "dht11"
```

기본 실습  
GPIO 입력(DHT11)

## Node-Red 둘러보기

The screenshot displays the Node-RED web interface. On the left, a sidebar contains various node categories: '일반' (General) with nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment; '기능' (Function) with nodes like function, switch, change, range, template, delay, trigger, exec, filter, random, and smooth; and '네트워크' (Network). The main workspace shows a workflow starting with a '타임스탬프' (Timestamp) node, followed by an 'rpi-dht22' sensor node, and ending with a 'set msg.' node. The 'change' node configuration panel is open on the right, showing the '속성' (Property) tab. It lists '이름' (Name) as '이름' and '대상' (Target) as 'msg.payload'. The '값의 이동' (Move value) dropdown is set to 'msg.humidity'. The '대상' (Target) dropdown is set to 'msg.payload'. The '배우기' (Learn) button is visible. The bottom status bar indicates '사용가능' (Available).

change의 노드 수정

삭제 취소 완료

속성

이름 이름

값의 이동 msg.humidity

대상 msg.payload

배우기

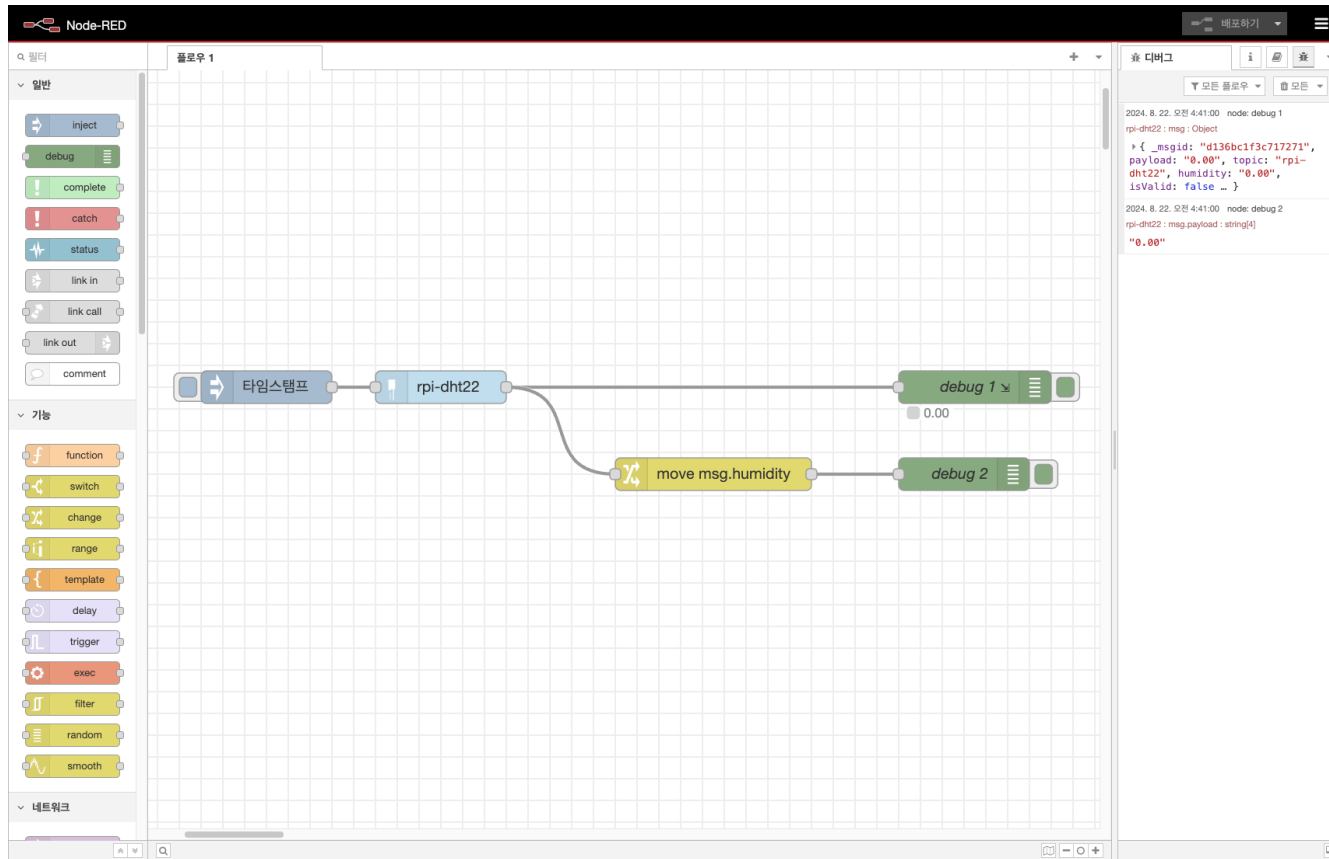
2024. 8. 22. 오전 4:37:10 noder debug 1  
rpi-dht22 : msg : Object

```
object
  _msgid: "6637a975d93a30ac"
  payload: "0.00"
  topic: "rpi-dht22"
  humidity: "0.00"
  isValid: false
  errors: 4
  location: undefined
  sensorid: "dht11"
```

사용가능

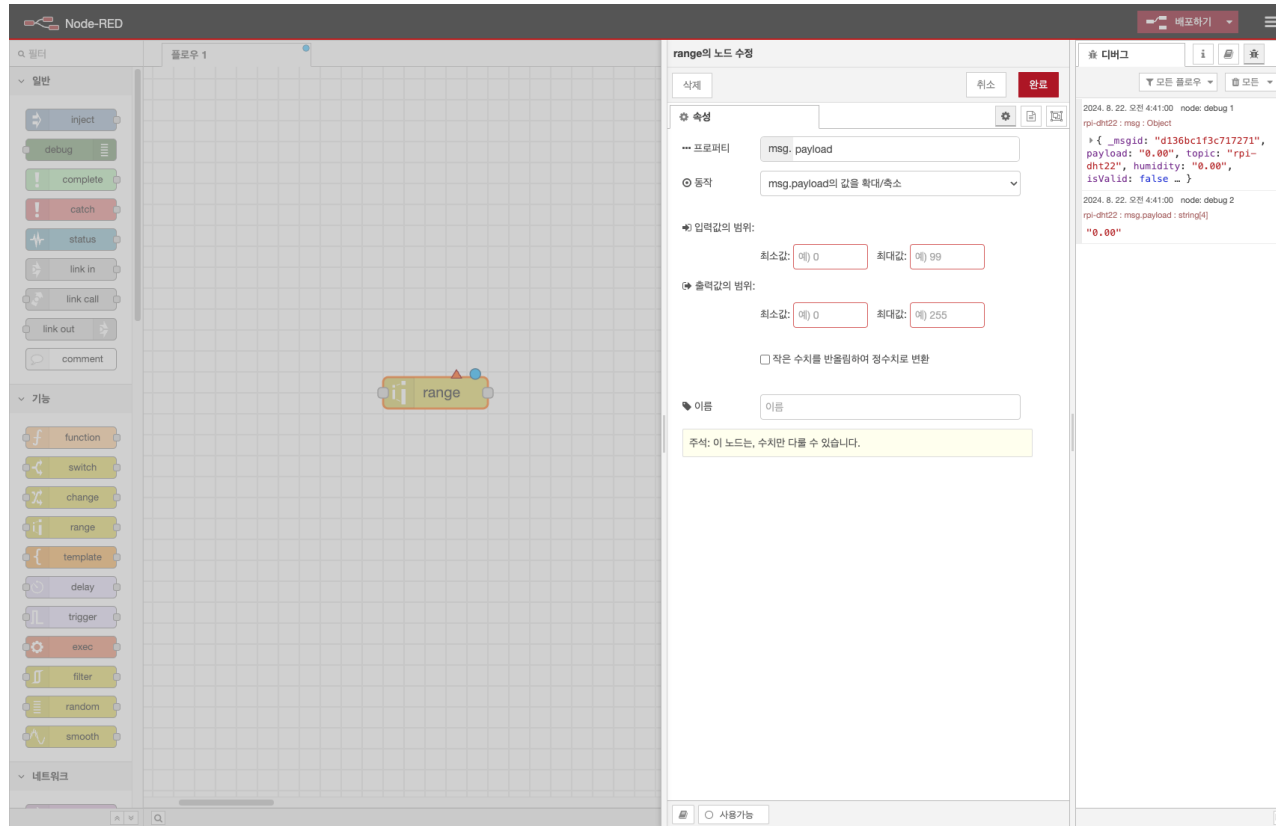
기본 실습  
습도 값 분리

## Node-Red 둘러보기



기본 실습  
습도 값 분리

## Node-Red 둘러보기



기본 실습  
범위 노드

## Node-Red 둘러보기

The screenshot displays the Node-RED web interface. On the left, a sidebar contains a search bar and two categories of nodes: '일반' (General) and '기능' (Function). The '일반' category includes nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The '기능' category includes function, switch, change, range, template, delay, trigger, exec, filter, random, and smooth. Below these is a '네트워크' (Network) section. The main workspace shows a single flow named '플로우 1' with a 'delay 5s' node. The right sidebar is divided into two panels. The top panel, titled 'delay의 노드 수정' (Edit delay node), shows settings for the selected node, including a '속성' (Property) section with a '메세지 지연' (Message delay) dropdown set to '지정된 시간 지연' (Specified time delay), a '시간' (Time) input set to '5 초' (5 seconds), and an '이름' (Name) input field. The bottom panel, titled '디버그' (Debug), shows a list of debug messages. The first message is from 'node debug 1' at 2024.8.22 04:41:00, showing a JSON object with fields like \_msgid, payload, topic, humidity, and isValid. The second message is from 'node debug 2' at the same time, showing a string payload '0.00'.

기본 실습  
지연 노드

## Node-Red 둘러보기

The screenshot displays the Node-RED web interface. On the left, the 'palette' shows various nodes categorized into '일반' (General), '기능' (Function), and '네트워크' (Network). A 'trigger 250ms' node is placed on the workspace. The right sidebar shows the 'trigger의 노드 수정' (Edit Trigger Node) configuration. The '속성' (Properties) tab is active, showing settings for the trigger. The '출신 데이터' (Source Data) is set to '1'. The '출신후의 처리' (Processing after output) is set to '지정된 시간 대기' (Wait for specified time). The '지정된 시간' (Specified time) is set to '250' milliseconds. The '재출신 데이터' (Resend Data) is set to '0'. The '초기화 조건' (Reset Condition) is set to 'msg.reset을 설정' (Set msg.reset). The '처리대상' (Processing Target) is set to '모든 메시지' (All messages). The '이름' (Name) is set to '이름'. The 'debug' console on the right shows two messages: a debug message from 'node: debug 1' and a message from 'rpi-dht22 : msg : Object' with a payload of '0.00'.

trigger 250ms

trigger의 노드 수정

속성

출신 데이터: 1

출신후의 처리: 지정된 시간 대기

지정된 시간: 250 밀리초

재출신 데이터: 0

초기화 조건: msg.reset을 설정

처리대상: 모든 메시지

이름: 이름

debug

```
2024. 8. 22. 오전 4:41:00 node: debug 1
rpi-dht22 : msg : Object
> { _msgid: "d136bc1f3c717271",
  payload: "0.00", topic: "rpi-
dht22", humidity: "0.00",
  isValid: false ... }

2024. 8. 22. 오전 4:41:00 node: debug 2
rpi-dht22 : msg.payload : string[4]
"0.00"
```

기본 실습  
트리거 노드



## Node-Red 둘러보기

The screenshot displays the Node-RED web interface. On the left, a sidebar contains various node categories: '일반' (General) with nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment; '기능' (Function) with nodes like function, switch, change, range, template, delay, trigger, exec, filter, random, and smooth; and '네트워크' (Network). The main workspace shows a single 'smooth' node placed on a grid. On the right, the 'smooth의 노드 수정' (Edit smooth node) panel is open. It includes a '삭제' (Delete) button, a '취소' (Cancel) button, and an '완료' (Done) button. The '속성' (Properties) section shows the '프로퍼티' (Property) set to 'msg.payload'. The 'Action' is 'Return the mean value'. The 'over the most recent' value is set to '10'. The 'round to' option is 'ignore'. The 'Treat' option is 'All msg as one stream.'. There is a checkbox for 'Reduce' with the label 'only emit one message per most recent N values'. A 'Name' field is present. A tip states: 'Tip: This node ONLY works with numbers.' The rightmost panel shows the '로그 디버깅' (Debug) console with two log entries from 'rpi-dht22' showing message objects and their payloads.

기본 실습  
스무스 노드

## Node-Red 둘러보기

The screenshot displays the Node-RED web interface. On the left, there are two panels: '일반' (General) and '기능' (Functions). The '일반' panel includes nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The '기능' panel includes nodes like function, switch, change, range, template, delay, trigger, exec, filter, random, and smooth. The main workspace shows a flow with three nodes: a '타임스탬프' (Timestamp) node, a '스위치' (Switch) node, and a 'debug 1' node. The flow starts with the '타임스탬프' node, which connects to the '스위치' node. The '스위치' node has two outputs: one leading to the 'debug 1' node and another leading to a '완료' (Complete) node. The '완료' node is currently disabled. On the right, the '출력 디버거' (Output Debugger) panel shows two log entries. The first entry is from 'node: debug 1' and shows a message object with fields: \_msgid, payload, topic, humidity, and isValid. The second entry is from 'node: debug 2' and shows a string value '0.00'.

기본 실습  
링크 노드

## Node-Red 둘러보기

The screenshot displays the Node-RED web interface. On the left, there are two panels: '일반' (General) and '기능' (Functions). The '일반' panel includes nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The '기능' panel includes nodes like function, switch, change, range, template, delay, trigger, exec, filter, random, and smooth. The main workspace shows a flow with three nodes: a '타임스탬프' (Timestamp) node, a '스위치' (Switch) node, and a 'debug 1' node. The flow starts with the Timestamp node, which connects to the Switch node. The Switch node has two outputs: one leading to the debug 1 node and another leading to a '완료' (Complete) node. The right sidebar shows the '출력 디버거' (Output Debugger) with two entries. The first entry shows a message object with fields: \_msgid, payload, topic, humidity, and isValid. The second entry shows a string payload: '0.00'.

기본 실습  
링크 노드

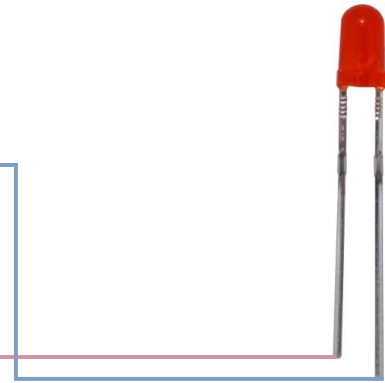
## Node-Red 둘러보기

The screenshot displays the Node-RED web interface. On the left, there are two panels: '일반' (General) and '기능' (Functions). The '일반' panel includes nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The '기능' panel includes nodes like function, switch, change, range, template, delay, trigger, exec, filter, random, and smooth. The main workspace shows a flow with three nodes: a '타임스탬프' (Timestamp) node, a '스위치' (Switch) node, and a 'debug 1' node. The flow starts with the '타임스탬프' node, which connects to the '스위치' node. The '스위치' node has two outputs: one leading to the 'debug 1' node and another leading to a '완료' (Complete) node. The '완료' node is currently disabled. On the right, the '출력 디버거' (Output Debugger) panel shows two log entries. The first entry is from 'node: debug 1' and shows a message object with fields: \_msgid, payload, topic, humidity, and isValid. The second entry is from 'node: debug 2' and shows a string value '0.00'.

기본 실습  
링크 노드

## Node-Red 둘러보기

3v3 Power	1	2	5v Power
GPIO 2 (I2C1 SDA)	3	4	5v Power
GPIO 3 (I2C1 SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (UART TX)
Ground	9	10	GPIO 15 (UART RX)
GPIO 17	11	12	GPIO 18 (PCM CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3v3 Power	17	18	GPIO 24
GPIO 10 (SPI0 MOSI)	19	20	Ground
GPIO 9 (SPI0 MISO)	21	22	GPIO 25
GPIO 11 (SPI0 SCLK)	23	24	GPIO 8 (SPI0 CE0)
Ground	25	26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	28	GPIO 1 (EEPROM SCL)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM DIN)
Ground	39	40	GPIO 21 (PCM DOUT)



기본 실습  
GPIO 출력(LED)

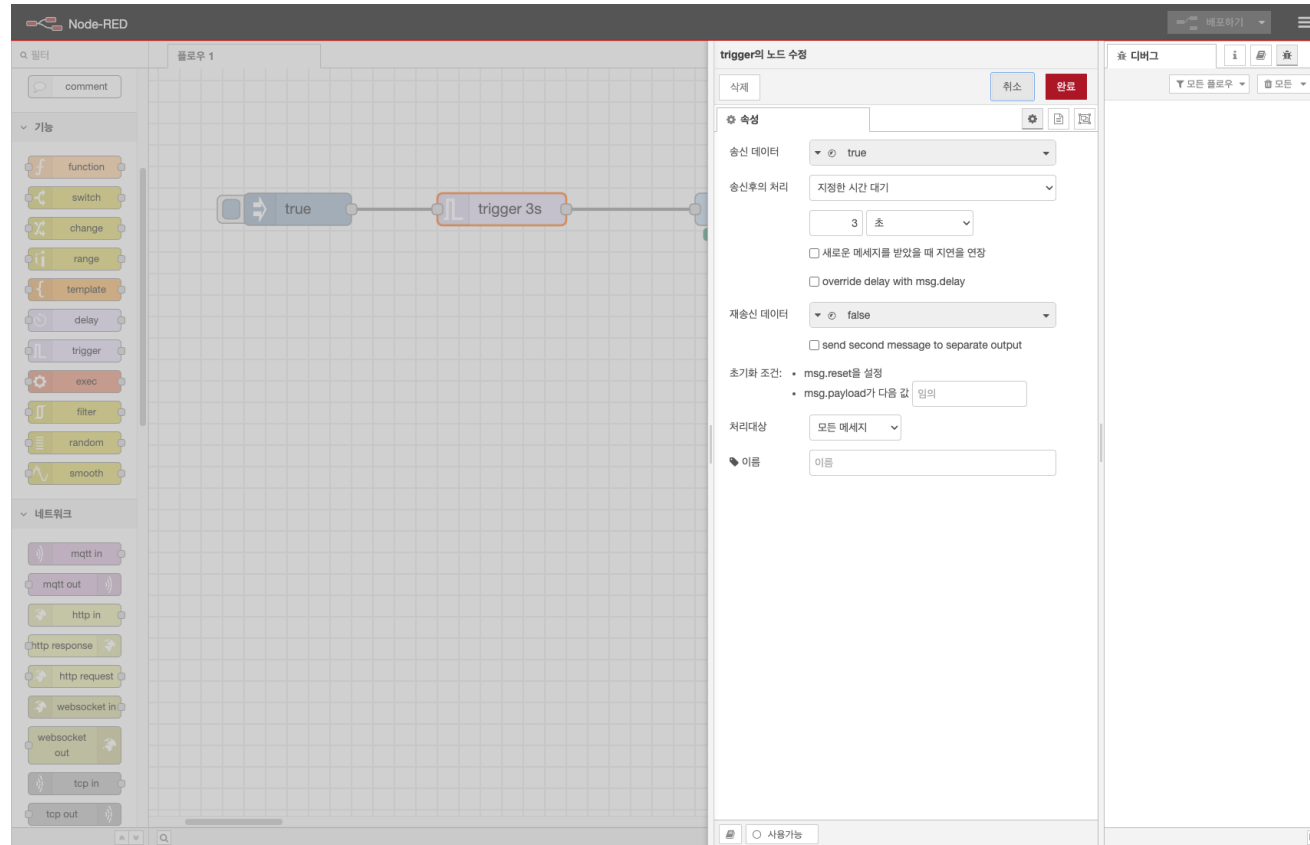
## Node-Red 둘러보기

The screenshot shows the Node-RED web interface. On the left, the 'Raspberry Pi' category is expanded, showing various nodes including 'rpi - gpio in' and 'rpi - gpio out'. The main workspace contains a workflow with two input nodes (true/false) connected to a 'PIN: 12' node. The right sidebar shows the configuration for the 'rpi-gpio out' node. The 'Pin' list is visible, with pin 12 selected. The 'BCM GPIO' is set to 18, and the 'Type' is 'Digital output'. The 'Initialise pin state?' checkbox is checked, and the 'Initial level of pin - low (0)' is set to 'low (0)'. The '이름' (Name) field is empty. The 'Pins in Use: 18' is displayed, and a tip states: 'Tip: For digital output - input must be 0 or 1.'

3.3V Power - 1	2 - 5V Power
SDA1 - GPIO2 - 3	4 - 5V Power
SCL1 - GPIO3 - 5	6 - Ground
GPIO4 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO9 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

기본 실습  
LED

## Node-Red 둘러보기



기본 실습  
LED(3Sec)

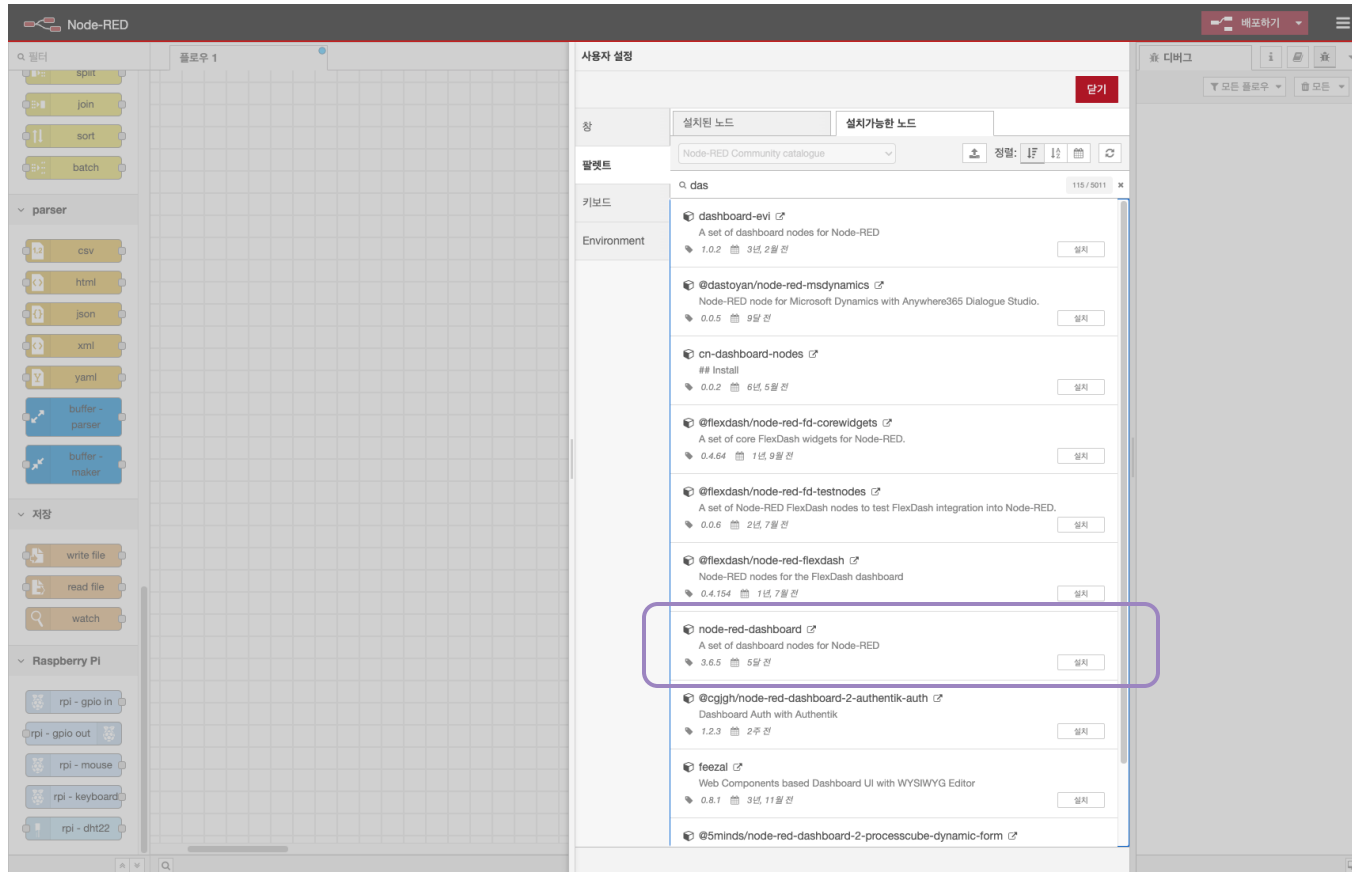
지금은?



쉬는 시간  
10min

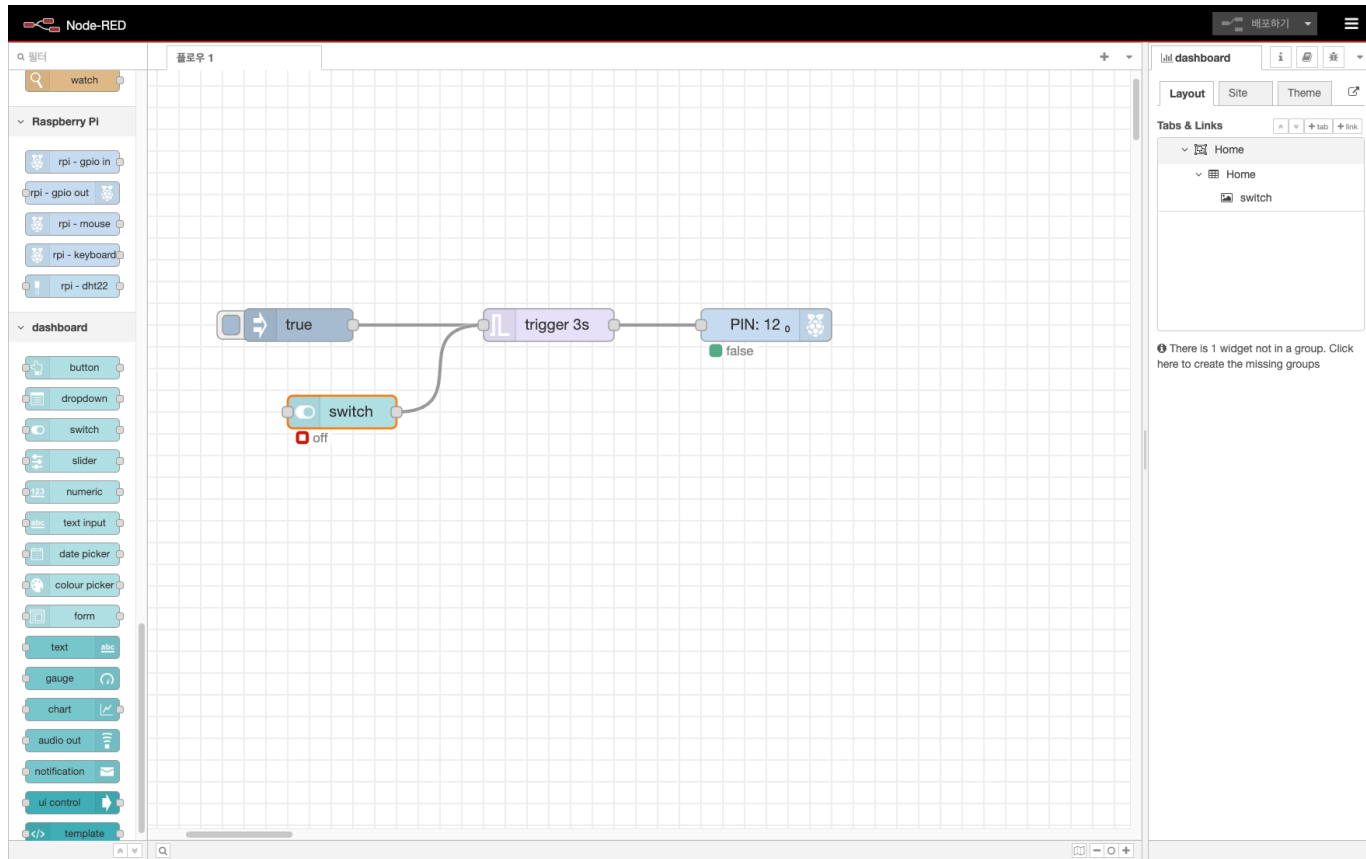


# Node-Red Dashboard



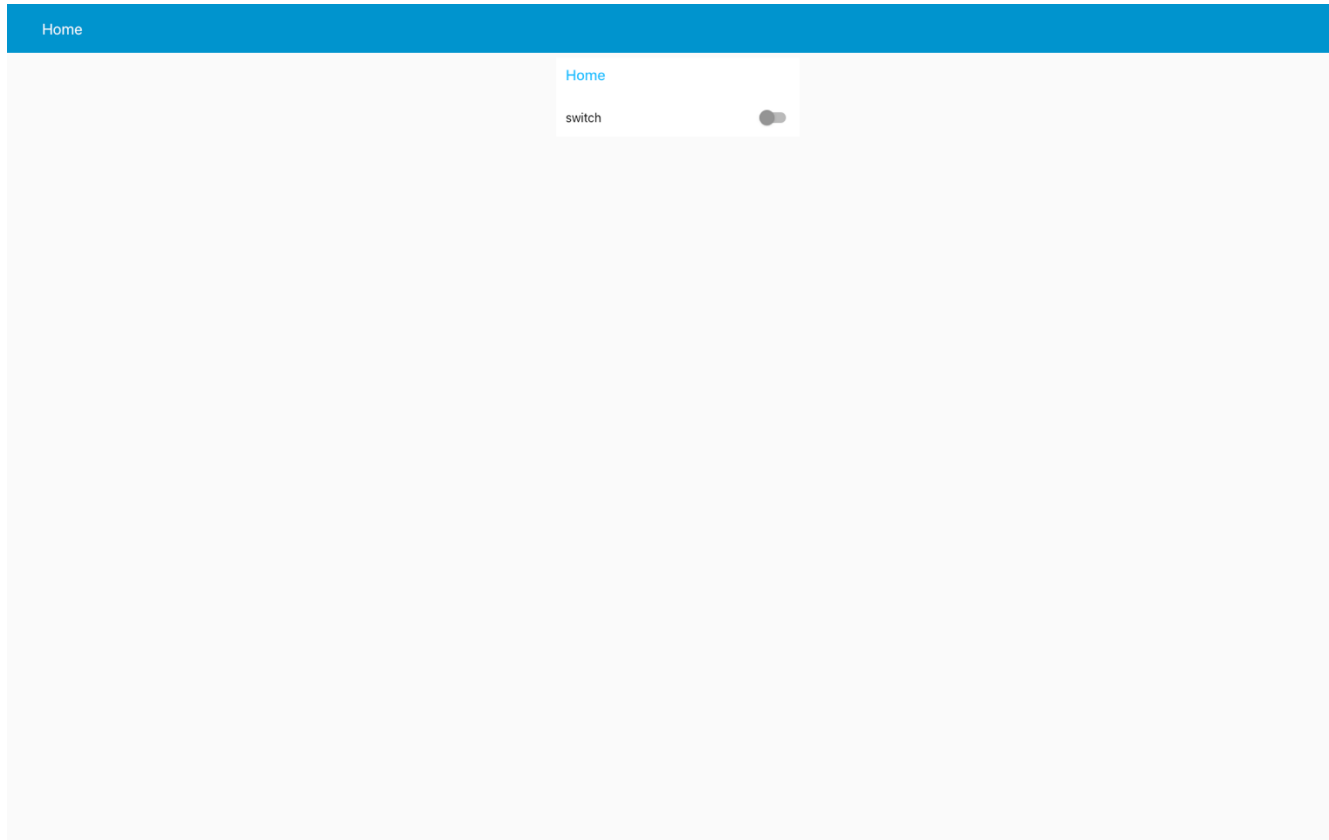
팔레트 노드 설치  
Dashboard검색

# Node-Red Dashboard



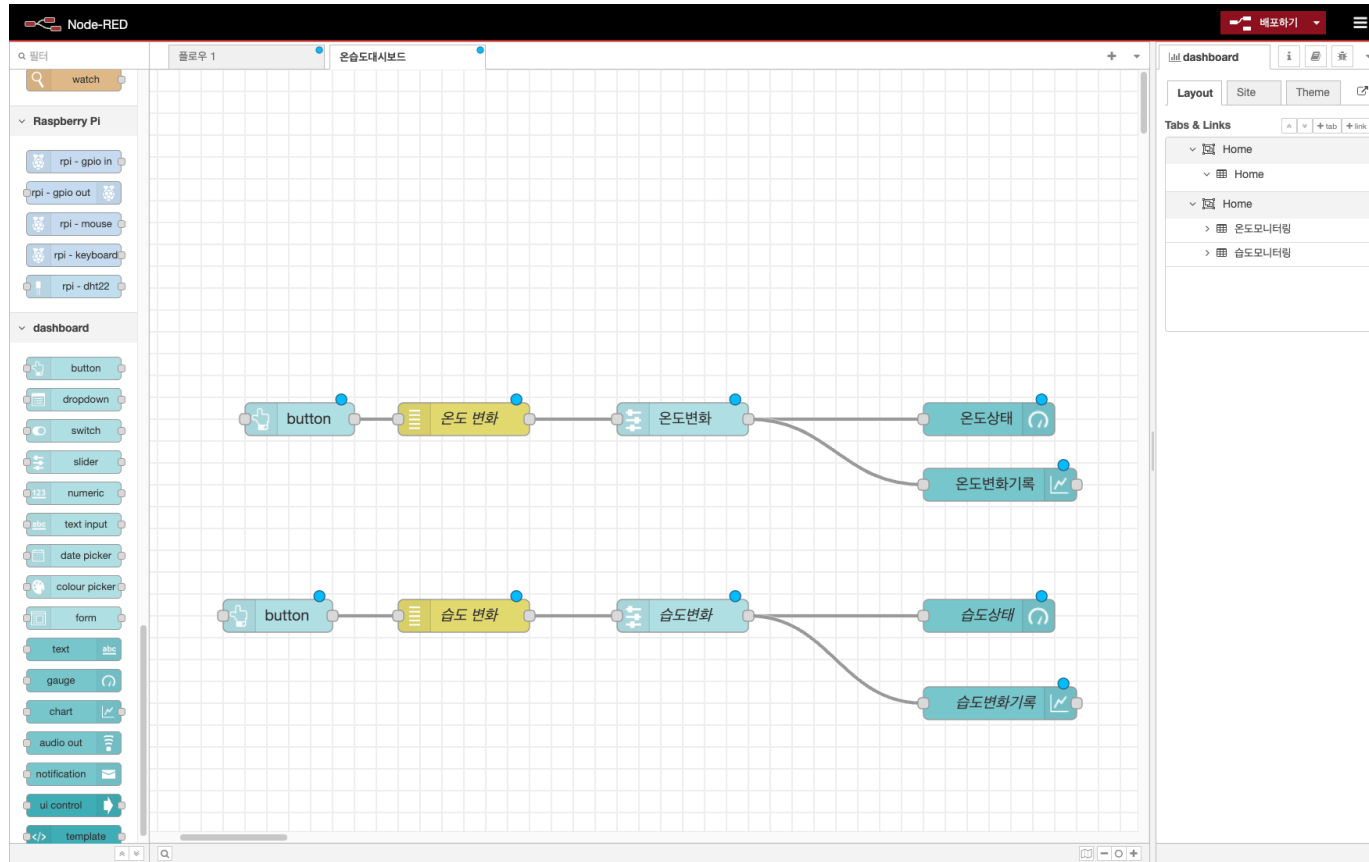
Dashboard실습  
스위치로 LED 켜고 끄기

## Node-Red Dashboard



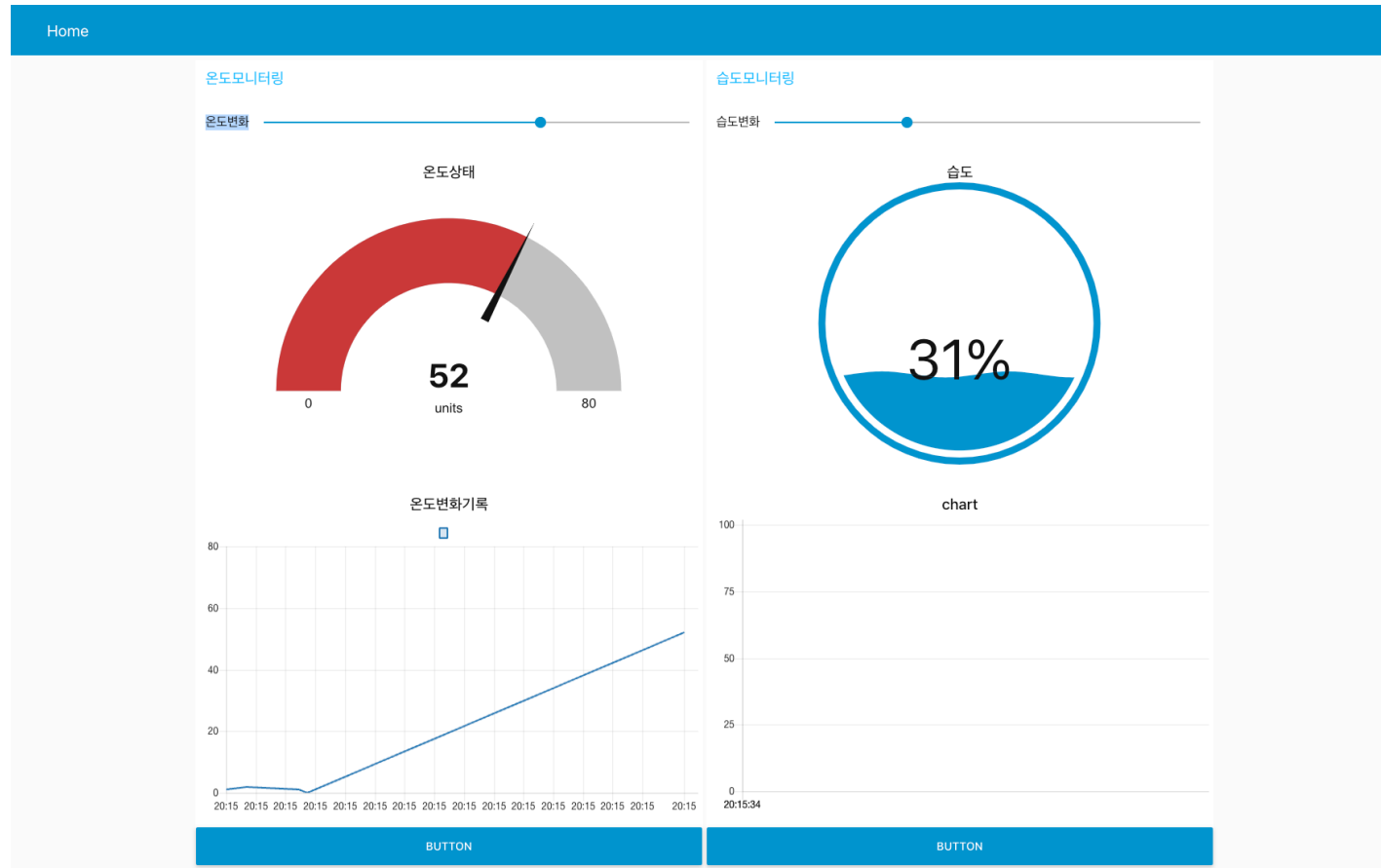
Dashboard실습  
스위치로 LED 켜고 끄기

# Node-Red Dashboard



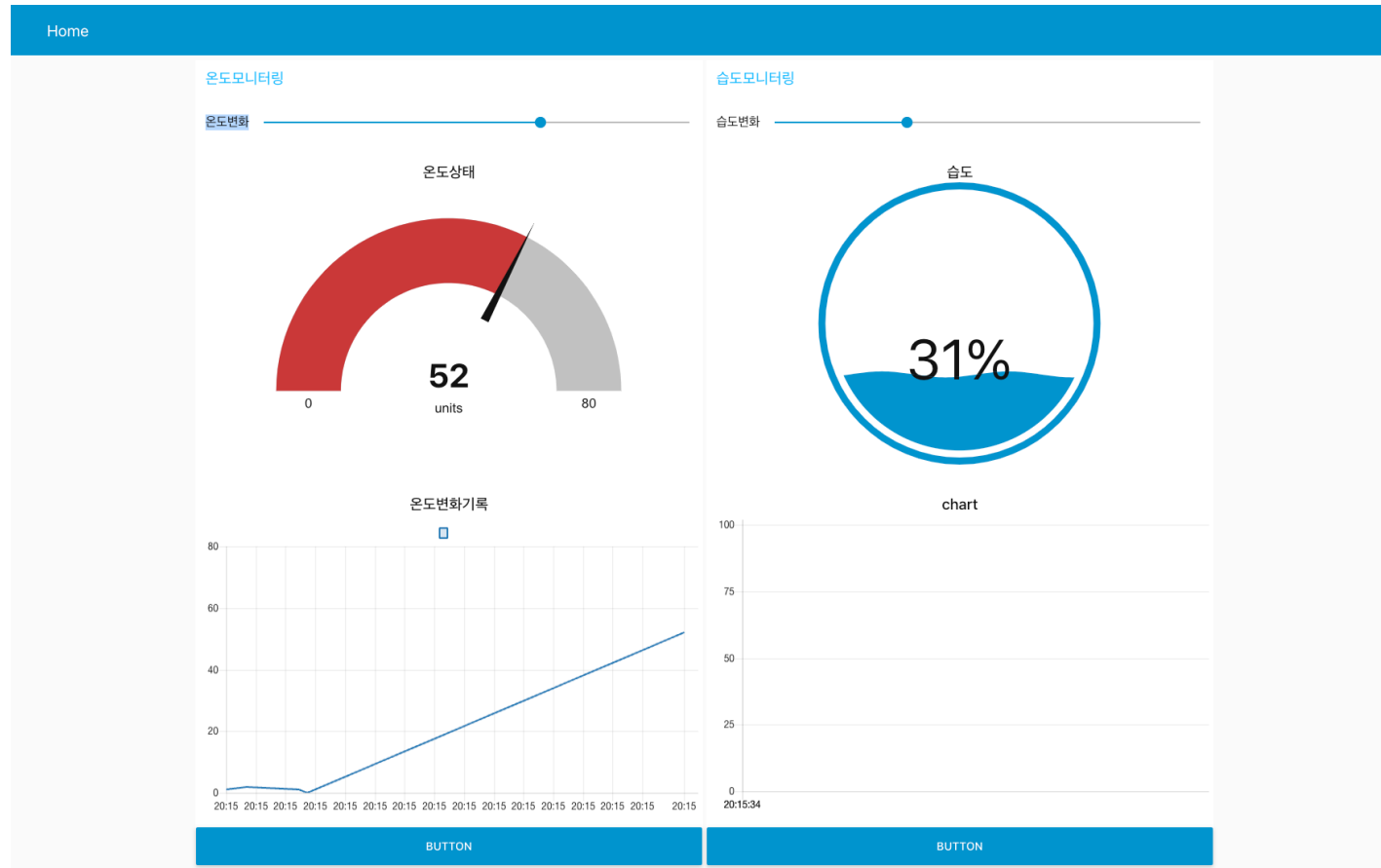
Dashboard실습  
온 · 습도 그래프

# Node-Red Dashboard



Dashboard실습  
온 · 습도 그래프

# Node-Red Dashboard



실습 과제  
DHT11에 값을 실시간으  
로  
표시하기

지금은?



쉬는 시간  
10min

CASA OS 설치 실습

CASA OS 설치 실습



지금은?



쉬는 시간  
10min

Pi-hosted 설치  
Docker / Container

지금은?



쉬는 시간  
10min

## Umbrel OS 설치 실습

Umbrel OS 설치  
Docker / Container

지금은?



쉬는 시간  
10min

교재 센서 실습

교재 센서 실습

Q & A



Q & A

## 강 의 일 정

● IoT 기반 공정설비모니터링 프로그래밍(5d) ● 로봇 프로그래밍(AI+파이썬)(3d)

1회 차 8월 19일(월)	IoT 기반 공정설비 모니터링의 이해	IoT의 이해 / 센서 & 모니터링 시스템 / IoT네트워크	이론
2회 차 8월 20일(화)	라즈베리파이 & 센서 프로그래밍 1	Rpi의 이해 / Rpi 개발환경 구축 / 센서 프로그래밍(LED/온 · 습도/초음파...)	이론 /실습
3회 차 8월 21일(수)	라즈베리파이 & 센서 프로그래밍 2	Rpi 원격 개발환경 / 센서 응용 프로그래밍	이론 /실습
4회 차 8월 22일(목)	Node-Red 제어&모니터링 프로그래밍 1	Node-Red의 이해 / 개발환경 구축 / 센서 모니터링 및 제어	이론 /실습
5회 차 8월 26일(월)	Node-Red 제어&모니터링 프로그래밍 2	Dashboard 설계 / http통신 & DB / MQTT	이론 /실습
6회 차 8월 27일(화)	ESP32 마이크로 파이썬 로봇 프로그래밍 1	ESP32의 이해 / 마이크로 파이썬 개발환경 구축 / 로봇 프로그래밍(LEC/온 · 습도/초음파...)	이론 /실습
7회 차 8월 28일(수)	ESP32 마이크로 파이썬 로봇 프로그래밍 2	로봇 프로그래밍(DC모터/서보모터/LCD...) / 웹 서버 구축 및 제어 / AI 프로그래밍 이해	이론 /실습
8회 차 9월 4일(수)	프로젝트 & 평가	팀 프로젝트 및 평가	실습 /평가