

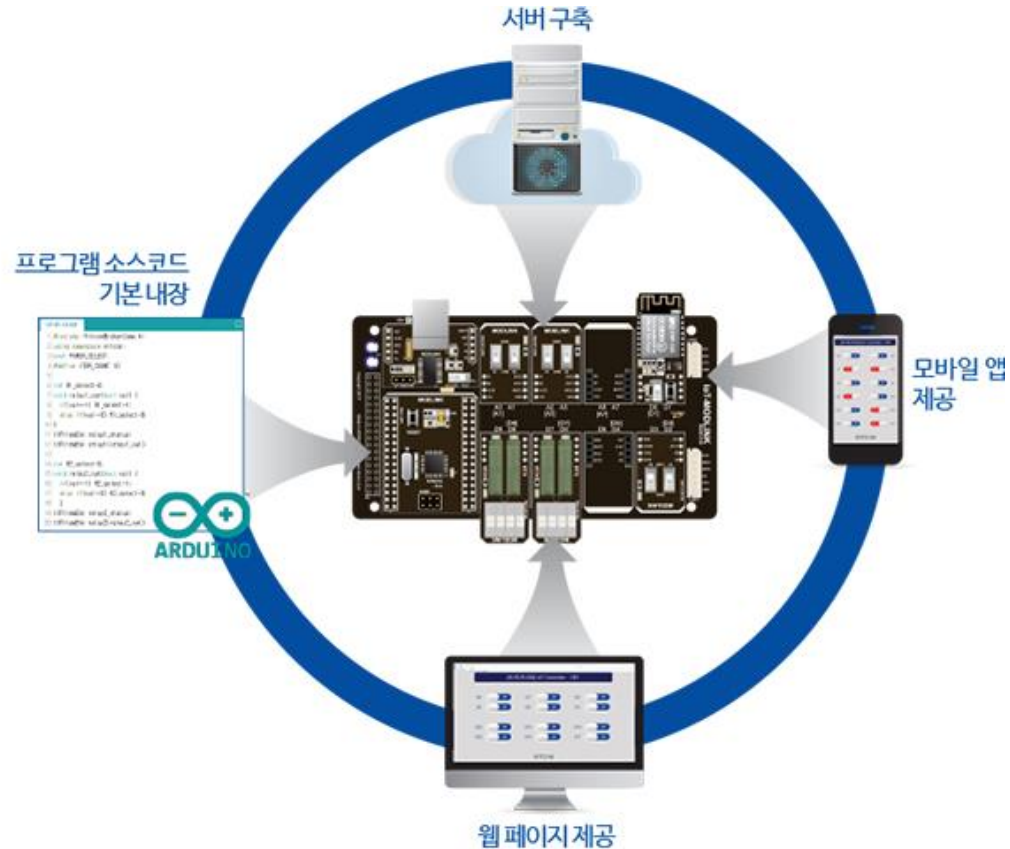
공기청정기 교육 자료

주식회사 빛컨



IoT

필요한 데이터를 생성, 수집을 위한 스마트공장의 핵심



“One-Stop IoT Solution”

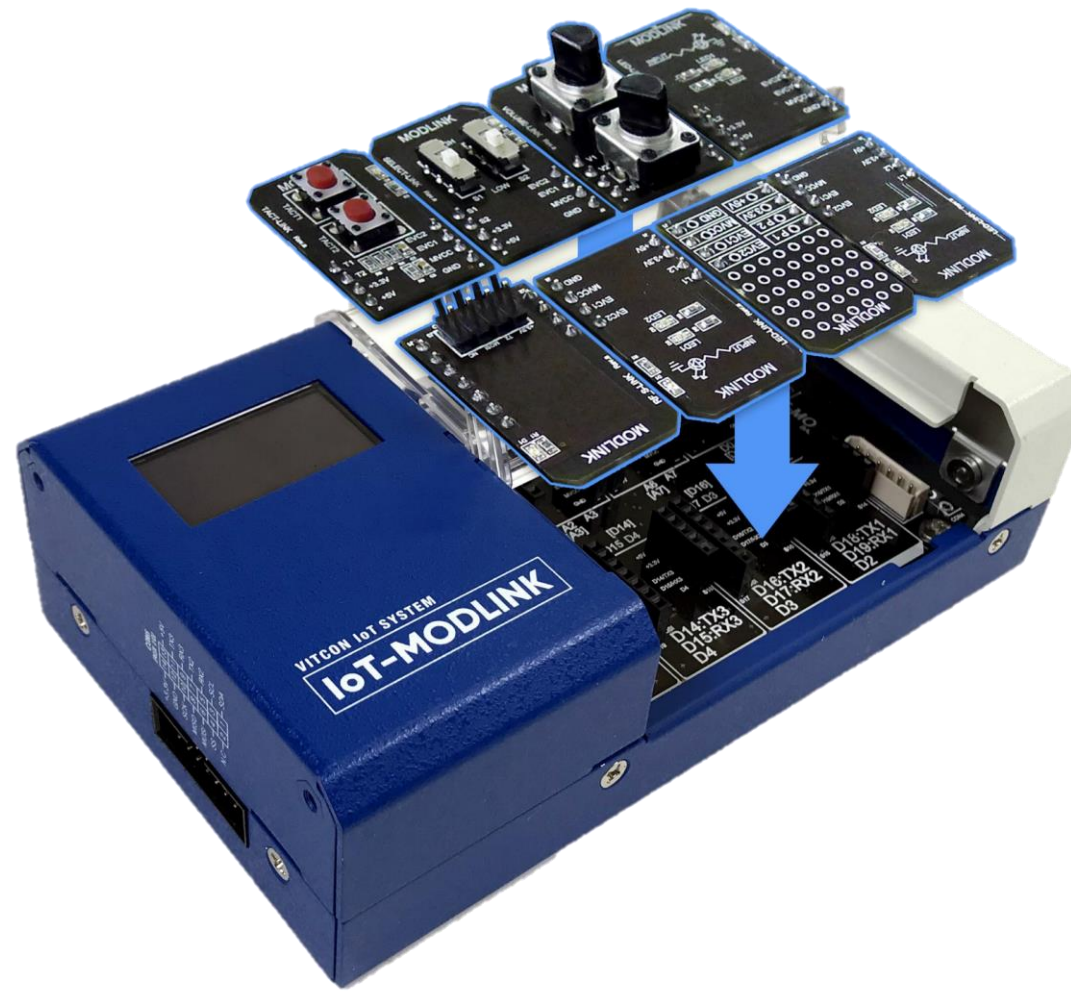
프로그램 소스코드 + 서버 구축 + 모바일 앱

WiFi 설정만으로 서버 자동 연결

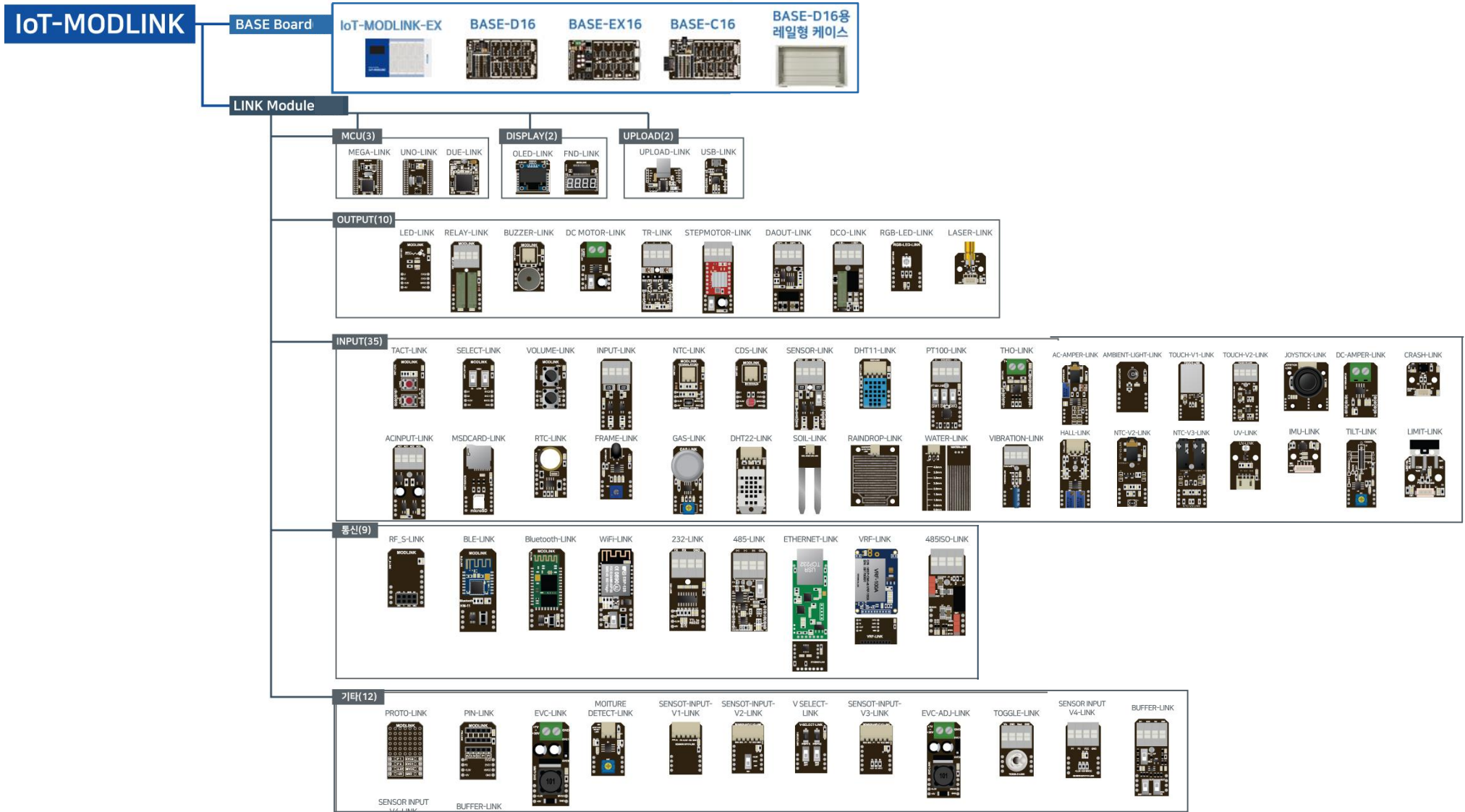
프로그램 소스 코드 제공 (기본 내장)

모바일 앱 제공

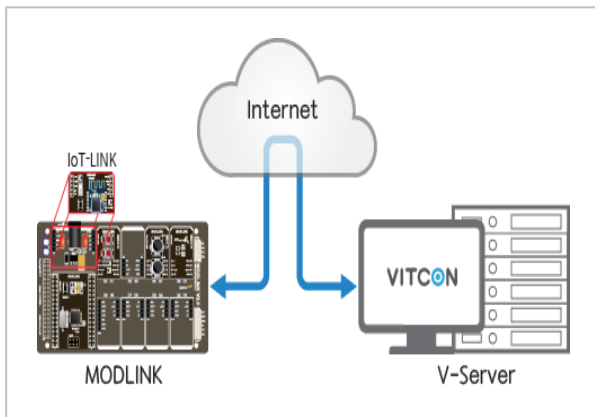
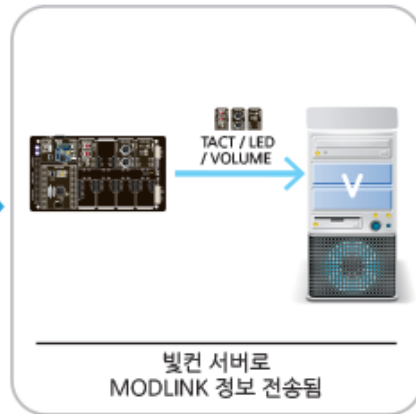
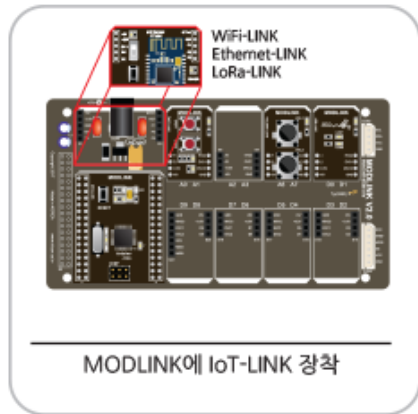
MODLINK- Overview



MODLINK- Overview



MODLINK- Overview



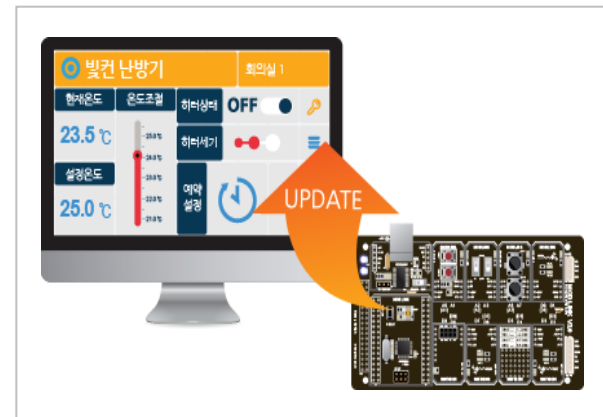
IoT-LINK 장착만으로 APP 구축



관리자페이지 위젯기능으로 편집



PC&Mobile 관리자페이지 자동생성



실시간 통신으로 자동 업데이트

기존 생산 설비 및 장비에
IoT-MODLINK를 부착만 하면 IoT가 가능해집니다.



설비 및 장비로부터 생성 및 수집된 데이터를 스마트폰으로 모니터링 및 제어할 수 있습니다.

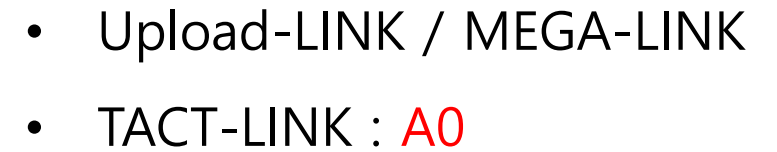
기초 ①

TACT-LINK

기초 1

아두이노 스케치의 **시리얼 모니터**를 사용해 봅시다

MODLINK 세팅



MODLINK 기초 1 - 시리얼모니터

아두이노 스케치

- 1 아두이노 스케치를 실행해주세요

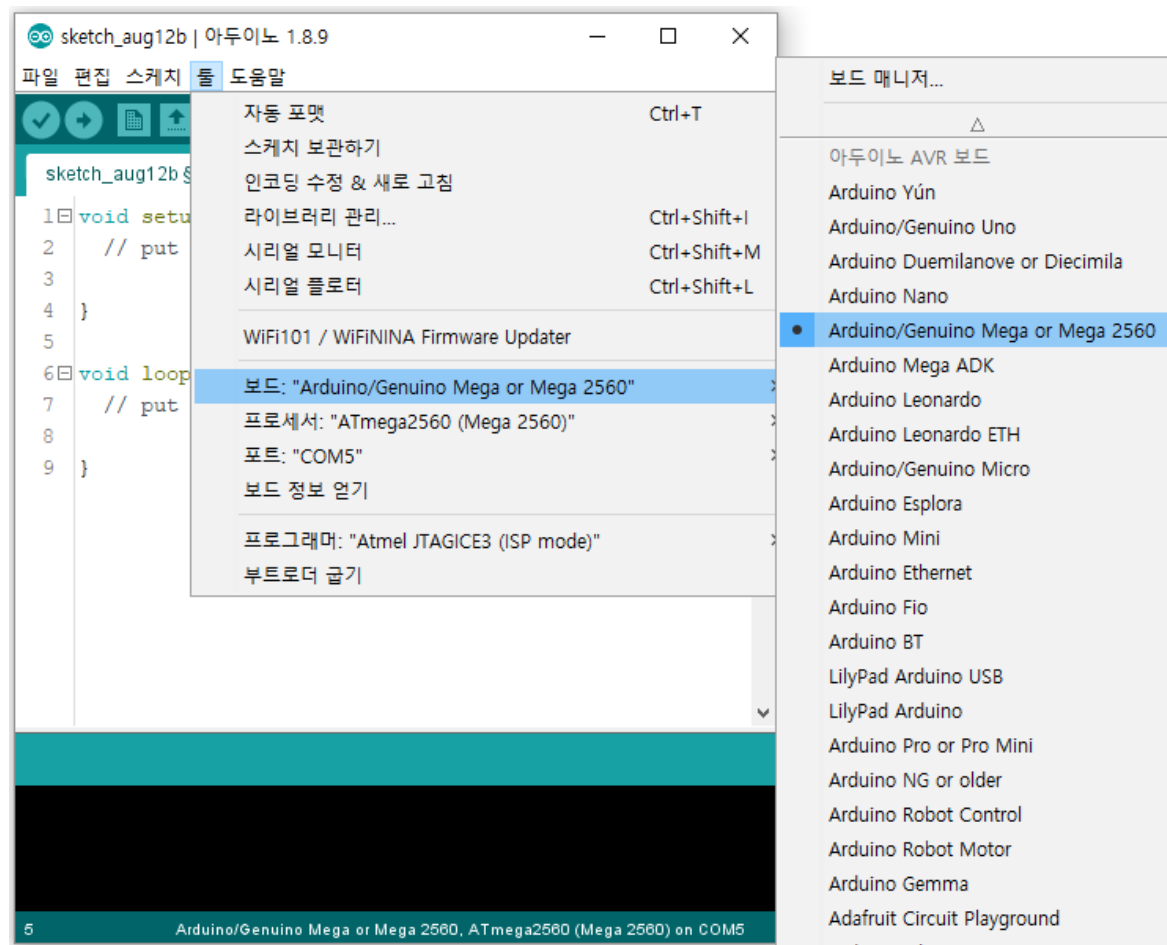


Arduino

MODLINK 기초 1 - 시리얼모니터

보드 설정

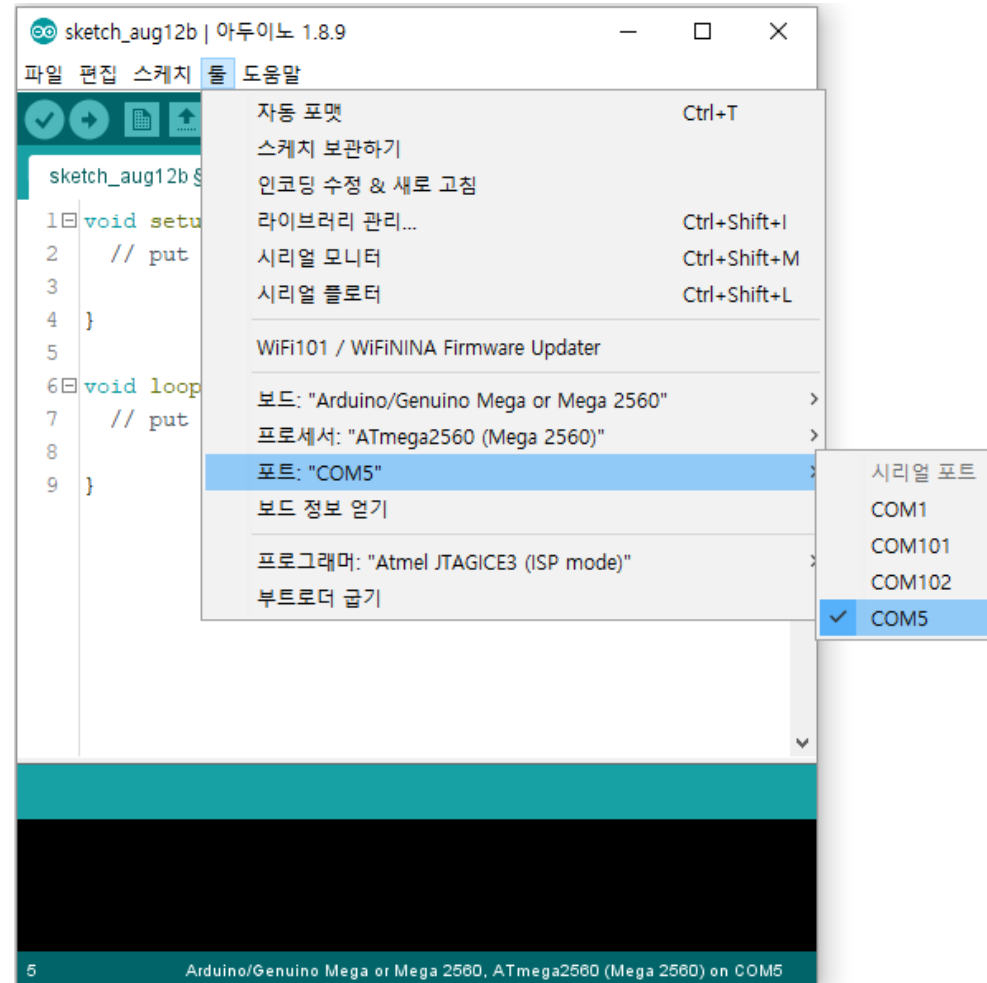
2 [툴] → [보드] → [Arduino/Genuino Mega or Mega 2560]를 선택해주세요



MODLINK 기초 1 - 시리얼모니터

포트 설정

3 [툴] → [포트] → [COM1] 제외한 시리얼 포트를 선택해주세요



MODLINK 기초 1 - 시리얼모니터

Coding

4 아두이노 스케치에 코드를 입력해주세요

```
bool buttonStatus;

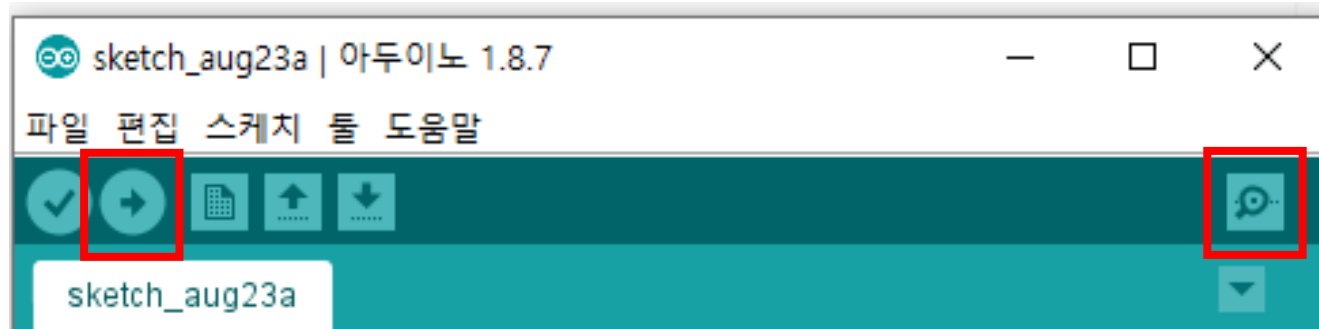
void setup() {
  Serial.begin(9600);
}

void loop() {
  buttonStatus = digitalRead(A0);
  Serial.print("A0 버튼 상태 : ");
  Serial.println(buttonStatus);
  delay(100);
}
```

MODLINK 기초 1 - 시리얼모니터

Upload

5 입력한 코드를 업로드 하고 시리얼 모니터 창을 열어 확인합니다.



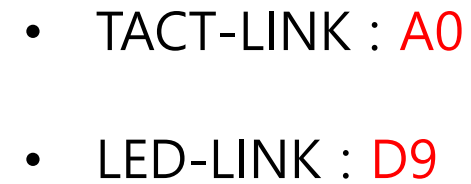
TACT-LINK

LED-LINK

기초 2

TACT 스위치를 활용하여 **LED**를 **ON/OFF** 해 봅시다.

MODLINK 세팅



MODLINK 기초 2 - 기본 입.출력

Coding

4 아두이노 스케치에 코드를 입력해주세요

```
bool buttonStatus;

void setup() {
  Serial.begin(9600);
  pinMode(A0, INPUT);
  pinMode(9, OUTPUT);
}

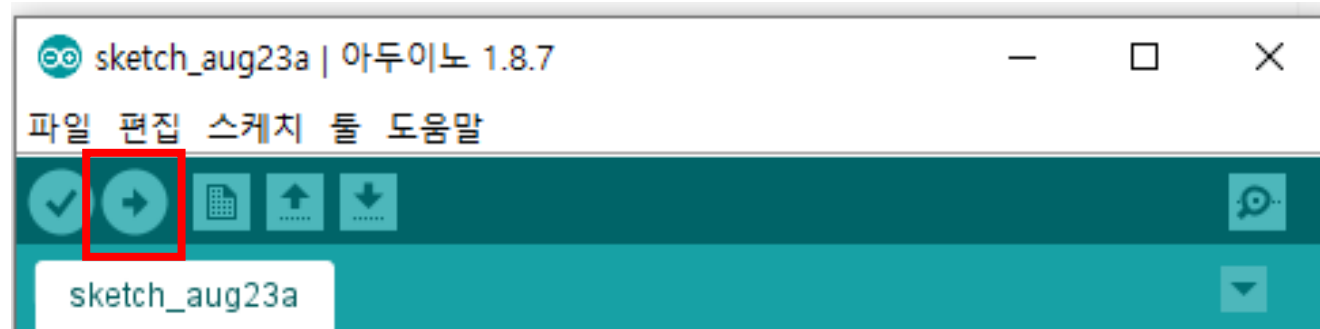
void loop() {
  buttonStatus = digitalRead(A0);

  if (buttonStatus == HIGH) {
    digitalWrite(9, HIGH);
    Serial.println("LED ON");
  }
  else if (buttonStatus == LOW) {
    digitalWrite(9, LOW);
    Serial.println("LED OFF");
  }
}
```

MODLINK 기초 2 - 기본 입.출력

Upload

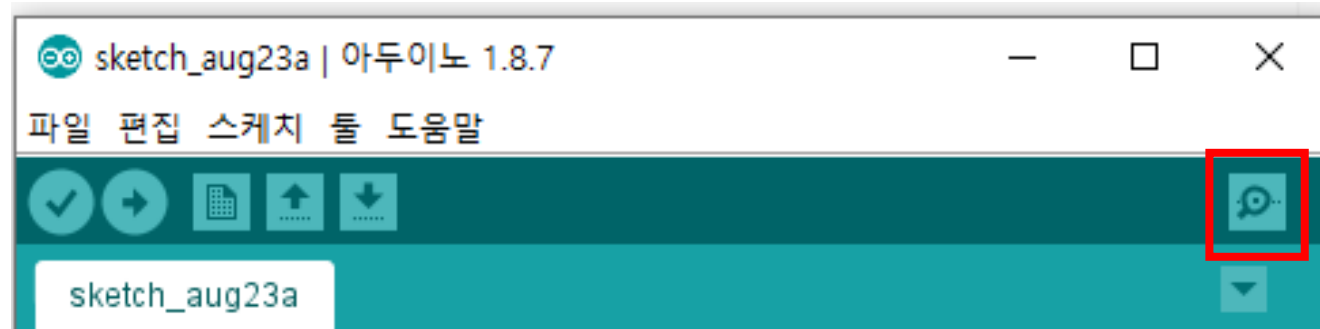
5 입력한 코드를 업로드 하주세요



MODLINK 기초 2 - 기본 입.출력

LED ON/OFF

6 TACT 스위치를 눌러 LED를 ON/OFF 해 보고, 시리얼 모니터 창에서 출력물을 확인합니다.



WiFi-LINK

LED-LINK

기초 3

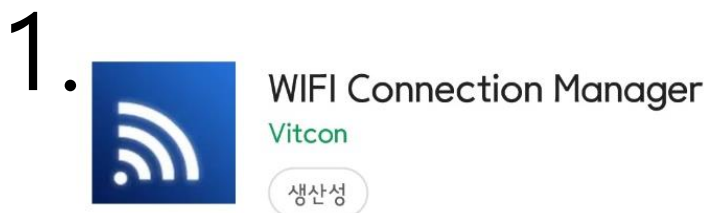
WEB 위젯 화면의 스위치를 활용하여 **LED**를 **ON/OFF** 해 봅시다.

MODLINK 기초 3 - Simple Switch

스마트폰 세팅

MODLINK에서 **생성&수집된 데이터를 서버로 보내고 원격으로 모니터링&제어**하기 위해 필요한 어플

[구글 Play 스토어] → [VITCON] 검색 → [WiFi Connection Manager] / [IoT Modlink] 설치하기 (혹은 iot.vitcon.co.kr 직접 입력(삼성 인터넷, Safari 등 웹 브라우저))



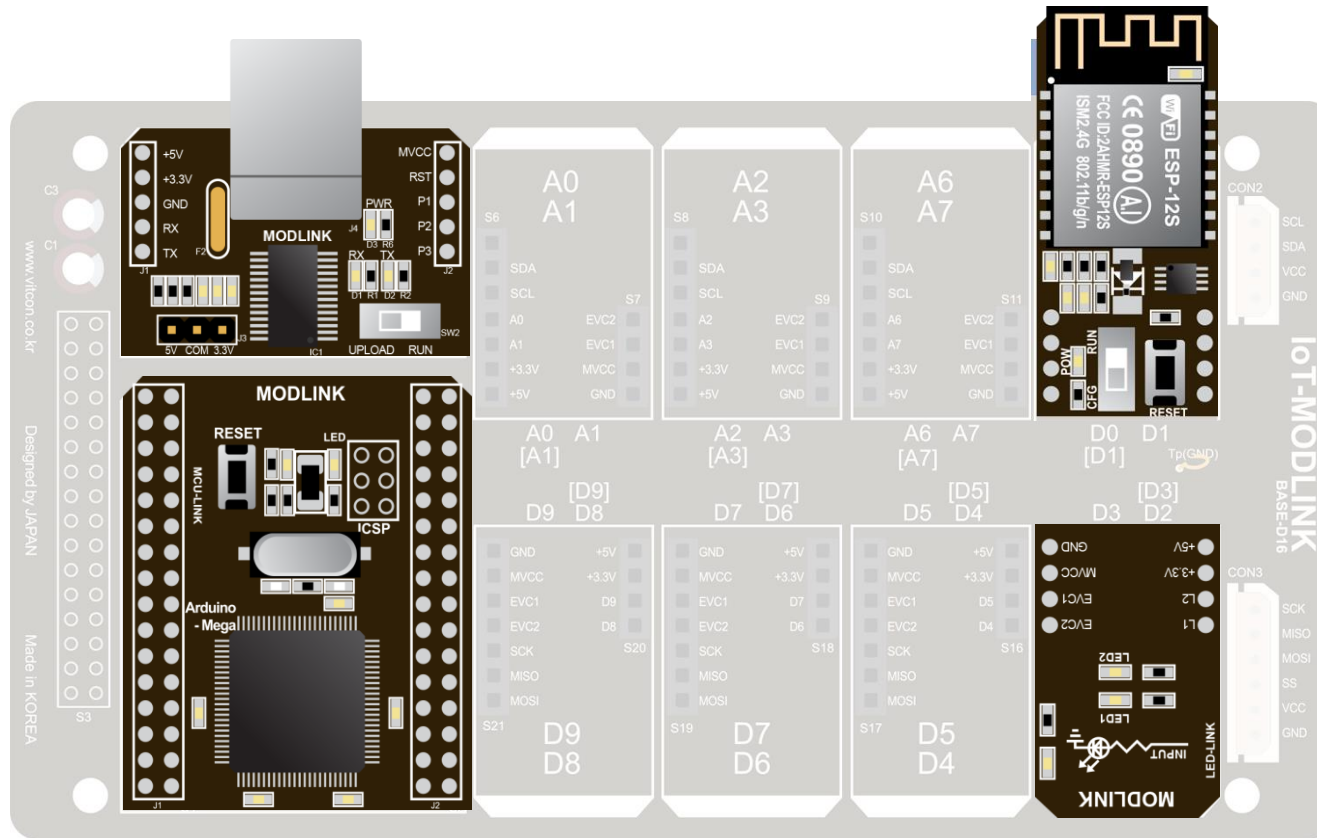
- MODLINK의 WiFi-LINK를 주변에 설치된 공유기와 연결
- 데이터를 보내고자 하는 서버의 IP와 포트번호를 설정



VITCON IoT 서버로 전송된 데이터를 위젯을 통해
모니터링 및 제어 가능

MODLINK 기초 3 - Simple Switch

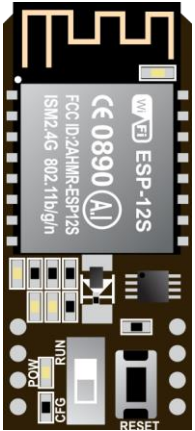
MODLINK 세팅



- LED-LINK : D2
- WIFI-LINK : D0, D1

MODLINK 기초 3 - Simple Switch

MODLINK 세팅



WiFi-LINK

주변에 설치된 공유기를 이용하여 장치를 인터넷에 연결하기 위한 링크 모듈

WiFi-LINK는 모든 시리얼 통신 포트에 사용 가능

(단, D0,D1(Serial)포트에 사용 시 프로그램을 업로드해주는 포트와 중복되므로 프로그램 업로드 시에는 CFG 모드로 변경 후 업로드)

WiFi Connection Manager 어플을 통해 주변 공유기와 연결하고 VITCON IoT 서버의 도메인과 포트 번호 설정

VITCON IoT 라이브러리를 사용하여 코딩해준 데이터가 WiFi-LINK를 통하여 설정된 VITCON IoT 서버로 전송

MODLINK 기초 3 - Simple Switch

VITCON IoT Service 사용하기

1 [http://iot.vitcon.co.kr] 접속하여 회원가입을 해주세요.

로그인

아이디 저장 자동 로그인

로그인

회원가입

[개인정보 처리방침](#)

회원가입

이메일*	<input type="text"/>
비밀번호*	<input type="password"/> <small>영문, 숫자 포함 8자 이상</small>
비밀번호 확인*	<input type="password"/> <small>비밀번호를 한번 더 입력해주세요</small>
이름*	<input type="text"/>
업체명	<input type="text"/>
휴대폰번호	<input type="text"/> <small>- 없이 입력해 주세요</small>

[회원가입](#) [돌아가기](#)

[개인정보 처리방침](#)

MODLINK 기초 3 - Simple Switch

VITCON IoT Service 사용하기

- 2 [프로젝트 목록] → '새 프로젝트 추가' → '프로젝트 이름(project1)' & '프리셋 선택(Simple Switch)'
→ '새 프로젝트 추가'를 클릭해 주세요

The screenshot shows the VITCON IoT Manager web interface. The main content area displays a table with columns for '프로젝트 이름' (Project Name), '가로보기 격자' (Grid View), '세로보기 격자' (List View), '생성 날짜' (Creation Date), '편집' (Edit), and '제거' (Delete). Below the table is a blue button labeled '새 프로젝트 추가' (Add New Project), which is highlighted with a red box.

A modal form titled '프로젝트 추가' (Add Project) is open, showing the following fields:

- 프로젝트 이름:** A text input field containing 'project1', highlighted with a red box.
- 프리셋 선택:** A dropdown menu with 'Simple Switch' selected, highlighted with a red box.
- Buttons:** At the bottom right of the modal, there are two buttons: '닫기' (Close) and '새 프로젝트 추가' (Add New Project), with the latter highlighted by a red box.

* 프리셋이란?

VITCON에서 미리 만들어 놓은 위젯 화면으로, 사용자가 직접 추가 편집할 수 있습니다.

MODLINK 기초 3 - Simple Switch

VITCON IoT Service 사용하기

3 [새 장비 추가] → 장비 이름을 입력하고 '새장비 추가' 를 클릭해 주세요.

The screenshot shows the Vitcon IoT Manager web interface. The browser address bar displays `https://iot.vitcon.co.kr/device/`. The page header includes the Vitcon logo, navigation links for '장비 목록' (Device List) and '프로젝트 목록' (Project List), a language dropdown, a user profile icon, and a '로그아웃' (Logout) button. Below the header is a table with columns: '사용자-제품명' (User-Product Name), '프로젝트 이름' (Project Name), 'IP 주소' (IP Address), '접속 상태' (Connection Status), '보기' (View), '설정 변경' (Change Settings), and '제거' (Remove). The table content area displays the message '할당된 Device ID가 없습니다.' (No assigned Device ID). A blue button labeled '새 장비 추가' (Add New Device) is highlighted with a red rectangle. A modal window titled '장비 추가' (Add Device) is open, showing a form with the label '새로운 이름:' (New Name:). The text 'TEST' is entered in the input field, which is also highlighted with a red rectangle. At the bottom of the modal, there are two buttons: '닫기' (Close) and '새 장비 추가' (Add New Device), with the latter being highlighted by a red rectangle.

사용자-제품명	프로젝트 이름	IP 주소	접속 상태	보기	설정 변경	제거
할당된 Device ID가 없습니다.						

새 장비 추가

장비 추가

새로운 이름:

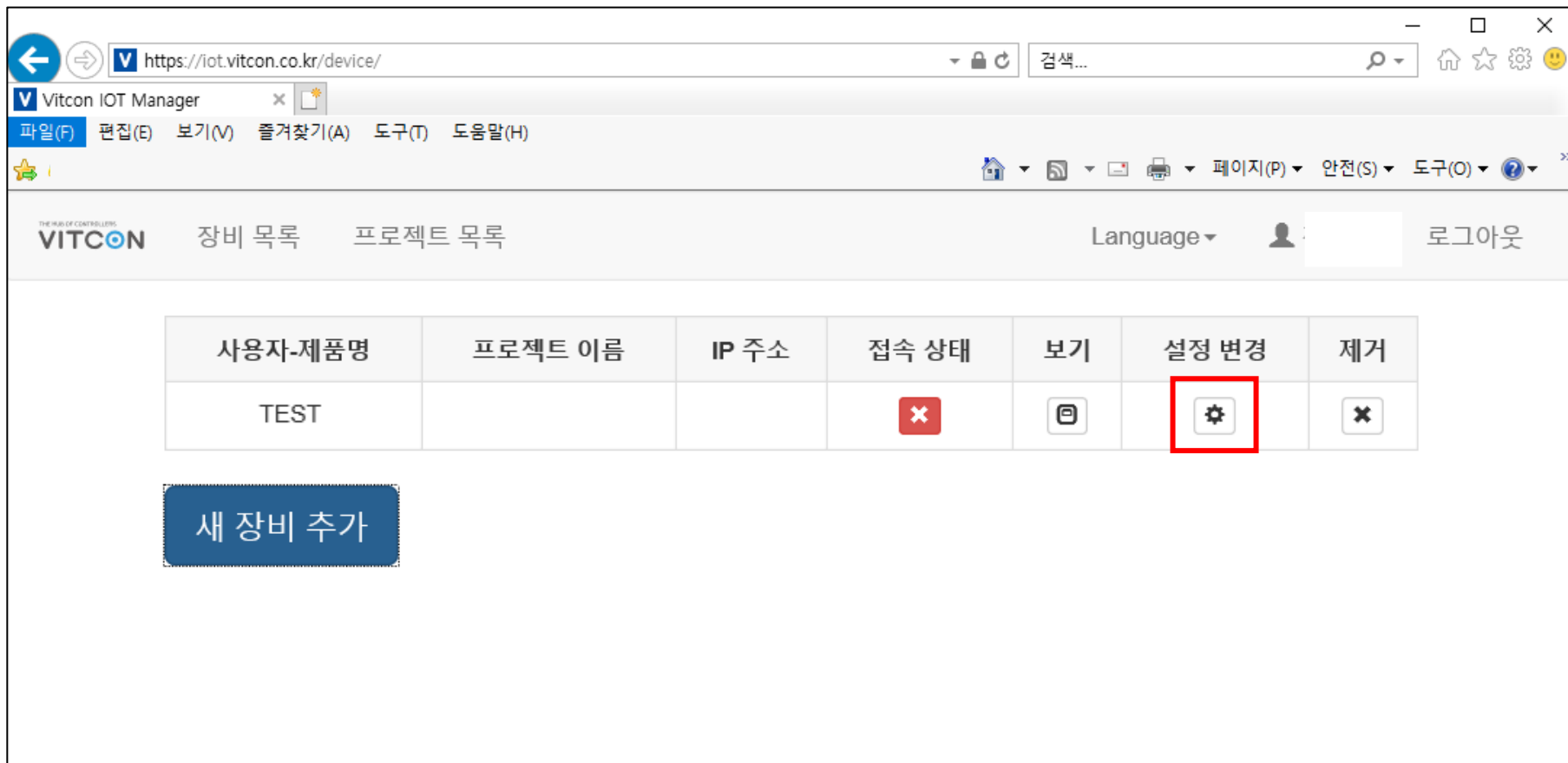
TEST

닫기 새 장비 추가

MODLINK 기초 3 - Simple Switch

VITCON IoT Service 사용하기

4 '장비 목록'에서 [설정 변경] 을 클릭해주세요



The screenshot shows the Vitcon IoT Manager web interface. The browser address bar displays <https://iot.vitcon.co.kr/device/>. The page title is 'Vitcon IOT Manager'. The navigation bar includes '파일(F)', '편집(E)', '보기(V)', '즐거찾기(A)', '도구(T)', and '도움말(H)'. The main header shows the Vitcon logo, '장비 목록' (Equipment List), '프로젝트 목록' (Project List), 'Language', a user profile icon, and '로그아웃' (Logout). The main content area features a table with the following columns: '사용자-제품명' (User-Product Name), '프로젝트 이름' (Project Name), 'IP 주소' (IP Address), '접속 상태' (Connection Status), '보기' (View), '설정 변경' (Change Settings), and '제거' (Remove). The first row of data shows 'TEST' under '사용자-제품명', an empty cell for '프로젝트 이름', an empty cell for 'IP 주소', a red 'X' icon for '접속 상태', a camera icon for '보기', a gear icon for '설정 변경' (highlighted with a red box), and a red 'X' icon for '제거'. Below the table is a blue button labeled '새 장비 추가' (Add New Equipment).

사용자-제품명	프로젝트 이름	IP 주소	접속 상태	보기	설정 변경	제거
TEST			×	📷	⚙️	×

새 장비 추가

MODLINK 기초 3 - Simple Switch

VITCON IoT Service 사용하기

5 생성한 프로젝트 이름(project1)을 [프로젝트 선택]에서 선택한 후 [현재 장비의 ID]를 복사해주세요

The screenshot shows the Vitcon IoT Manager web interface. A modal dialog is open for adding a new device. The dialog contains the following fields and elements:

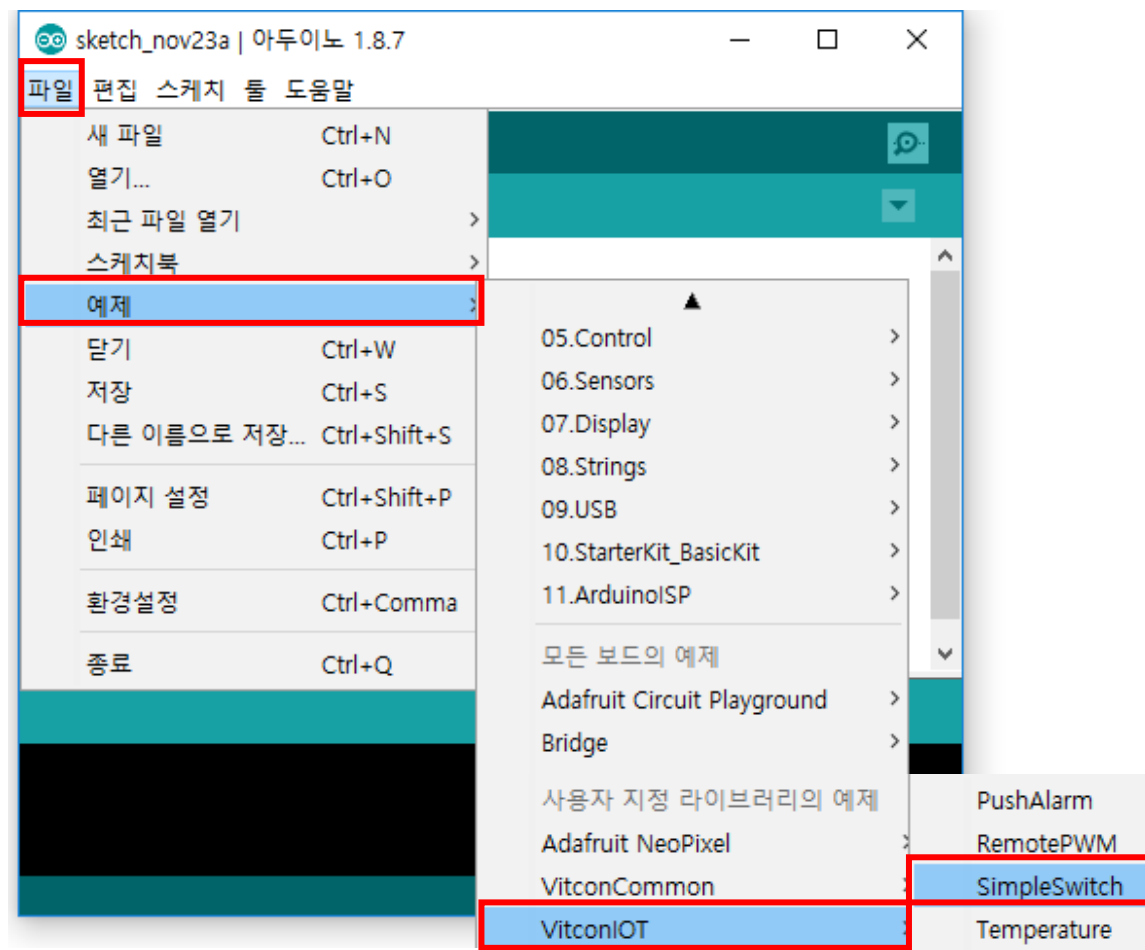
- 현재 장비의 ID:** A text field containing the ID `8710b74f630b588b4cf305297f0c4899`, which is highlighted in yellow and has a red box around it. To its right is a red text label "복사" (Copy).
- 새로운 이름:** A text input field containing the value "TEST".
- 프로젝트 선택:** A dropdown menu with "project1" selected, highlighted by a red box.
- Buttons:** At the bottom right of the dialog are two buttons: "닫기" (Close) and "변경" (Change/Save), with the "변경" button highlighted by a red box.

In the background, the main interface shows a table with the header "사용자-제품명" (User-Product Name) and one row with the value "TEST". Below the table is a blue button labeled "새 장비 추가" (Add New Device).

MODLINK 기초 3 - Simple Switch

SimpleSwitch 예제 열기

6 [파일] → [예제] → [VitconIOT] → [SimpleSwitch]를 선택해주세요



MODLINK 기초 3 - Simple Switch

Coding

```
SimpleSwitch
14 #include <VitconBrokerComm.h>
15
16 using namespace vitcon;
17
18 /* A set of definition for IOT items */
19 #define ITEM_COUNT 2
20 void relay_out(bool val) { digitalWrite(2, val); }
21 IOTItemBin relay_status;
22 IOTItemBin relay(relay_out);
23 IOTItem *items[ITEM_COUNT] = { &relay_status, &relay };
24 /* A set of definition for IOT items */
25
26 /* IOT server communication manager */
27 const char device_id[] = ""; // Change device_id to yours
28 BrokerComm comm(&Serial, device_id, items, ITEM_COUNT);
29 /* IOT server communication manager */
30
31 void setup() {
32     Serial.begin(250000);
33     pinMode(2, OUTPUT);
34     comm.SetInterval(200);
35 }
36
37 void loop() {
38     relay_status.Set(digitalRead(2));
39     comm.Run();
40 }
```

라이브러리 사용

아이템 설정

서버 위젯 인덱스와 연동

장치 ID값 입력

Modlink에서 읽은 값을 서버로 전송

MODLINK 기초 3 - Simple Switch

Coding

7 아래와 같이 코드를 수정해주세요

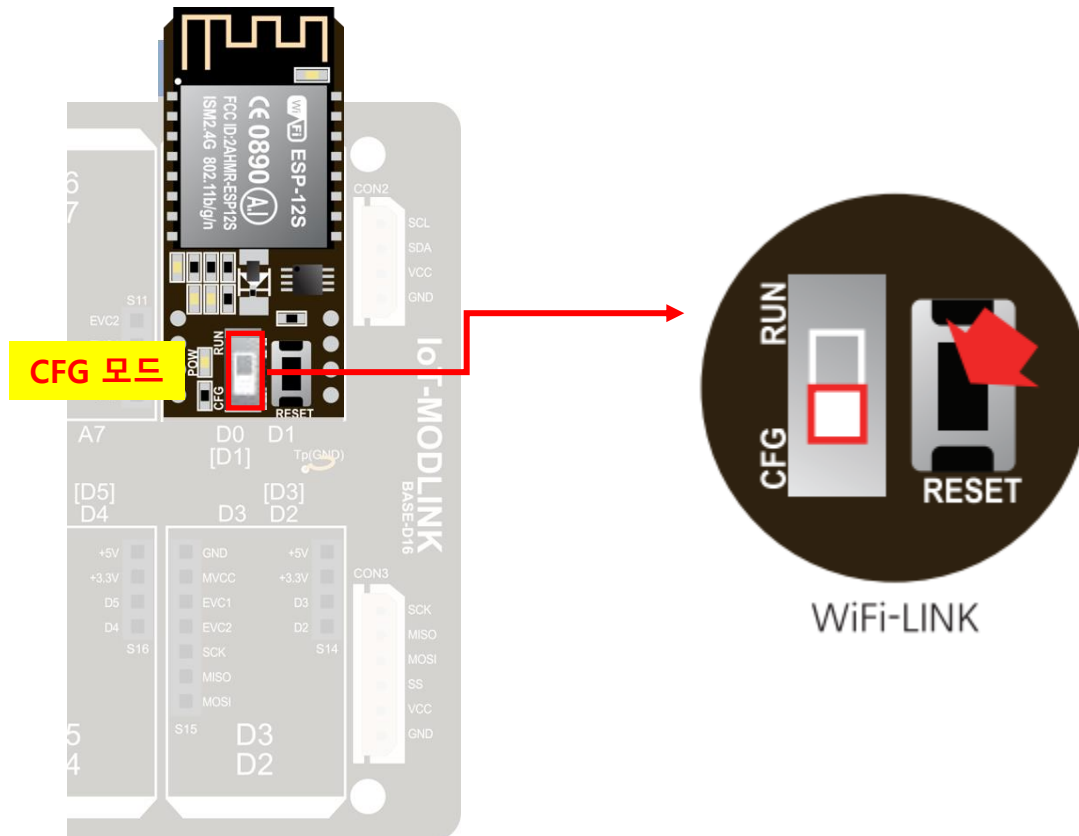
“ ” 사이에 복사한 디바이스ID를
붙여 넣어주세요

```
26  /* IOT server communication manager */
27  const char device_id[] = ""; // Change device_id to yours
28  BrokerComm comm(&Serial, device_id, items, ITEM_COUNT);
29  /* IOT server communication manager */
30
31  void setup() {
32      Serial.begin(250000);
33      pinMode (2, OUTPUT);
34      comm.SetInterval(200);
35  }
36
37  void loop() {
38      relay_status.Set(digitalRead (2)) ;
39      comm.Run ();
40  }
```

MODLINK 기초 3 - Simple Switch

Upload

8 WiFi-LINK를 **CFG(아래)**에 맞추고 **RESET** 버튼을 누른 후 코드를 **업로드** 해 주세요.



MODLINK 기초 3 - Simple Switch

WEB 모니터링 및 제어하기

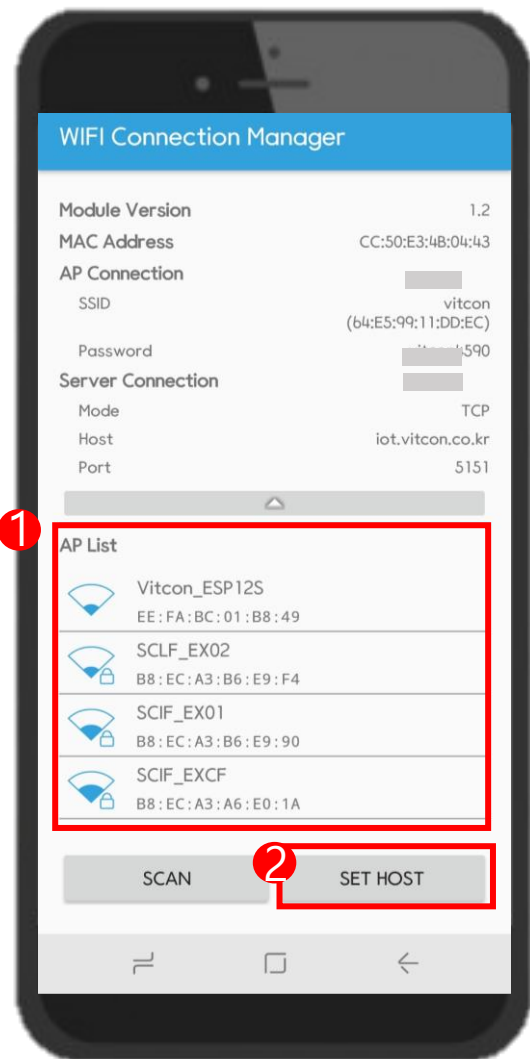
- 9 WiFi-LINK의 라벨 번호를 확인하고, 스마트폰 WiFi 목록에서 해당 번호로(Vitcon_ESP12S*****) 설정해주세요.



MODLINK 기초 3 - Simple Switch

WEB 모니터링 및 제어하기

10 WiFi Connection Manager 어플리케이션을 실행해주세요.

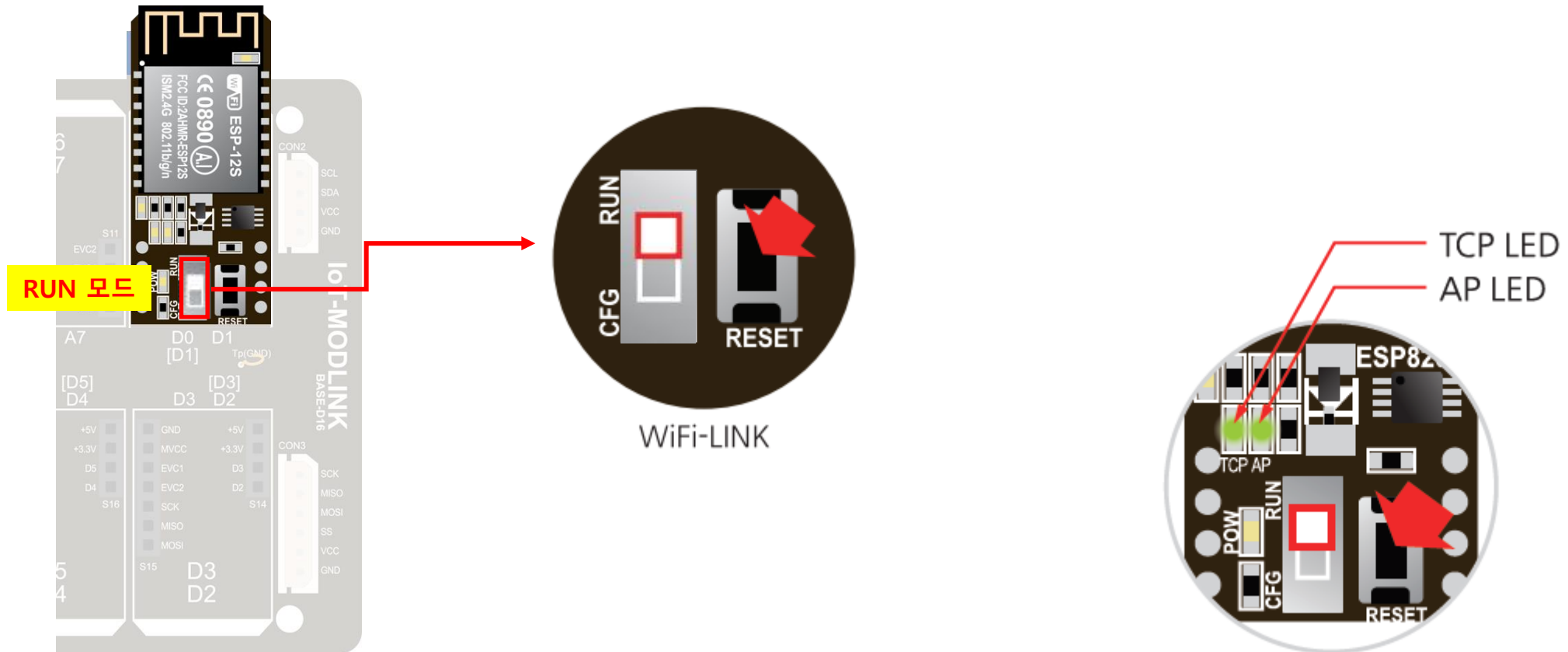


- 1) AP List에서 연결할 공유기(WiFi) 선택
- 2) SET HOST 클릭
- 3) Set Target TCP Server 입력(기본값 VITCON 서버)

MODLINK 기초 3 - Simple Switch

WEB 모니터링 및 제어하기

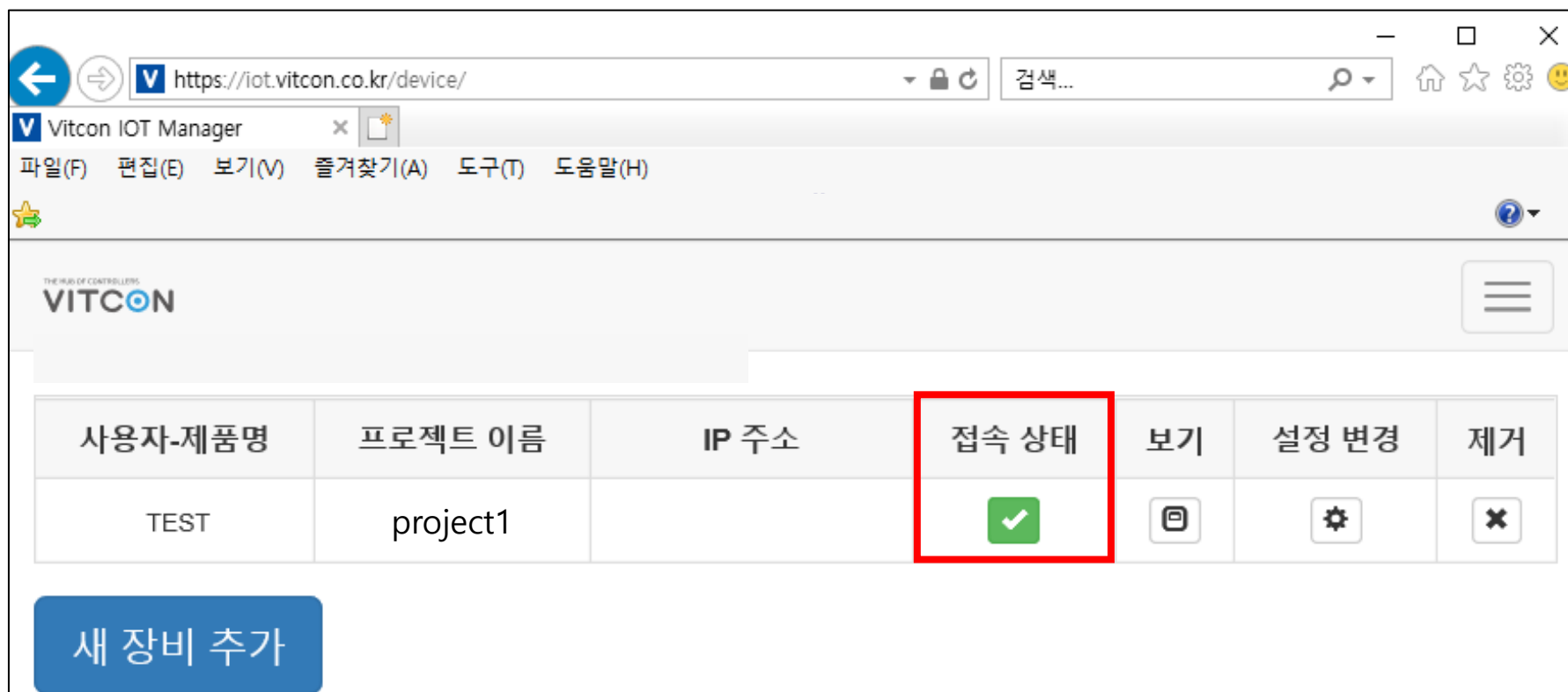
- 11 WiFi-LINK 스위치를 RUN(위)에 맞추고 RESET(검정색) 버튼을 누른 후 WiFi-LINK의 TCP, AP LED에 각각 녹색 불이 들어오는지 확인해 주세요.



MODLINK 기초 3 - Simple Switch

WEB 모니터링 및 제어하기

12 lot.vitcon.co.kr 웹 화면에서 해당 프로젝트의 **접속 상태**가 **녹색**인지 확인해주세요



The screenshot shows a web browser window with the URL <https://iot.vitcon.co.kr/device/>. The page title is "Vitcon IOT Manager". Below the header, there is a table with the following columns: "사용자-제품명", "프로젝트 이름", "IP 주소", "접속 상태", "보기", "설정 변경", and "제거". The first row of data shows "TEST" for the user-product name, "project1" for the project name, and a green checkmark in the "접속 상태" column, which is highlighted with a red box. Below the table is a blue button labeled "새 장비 추가".

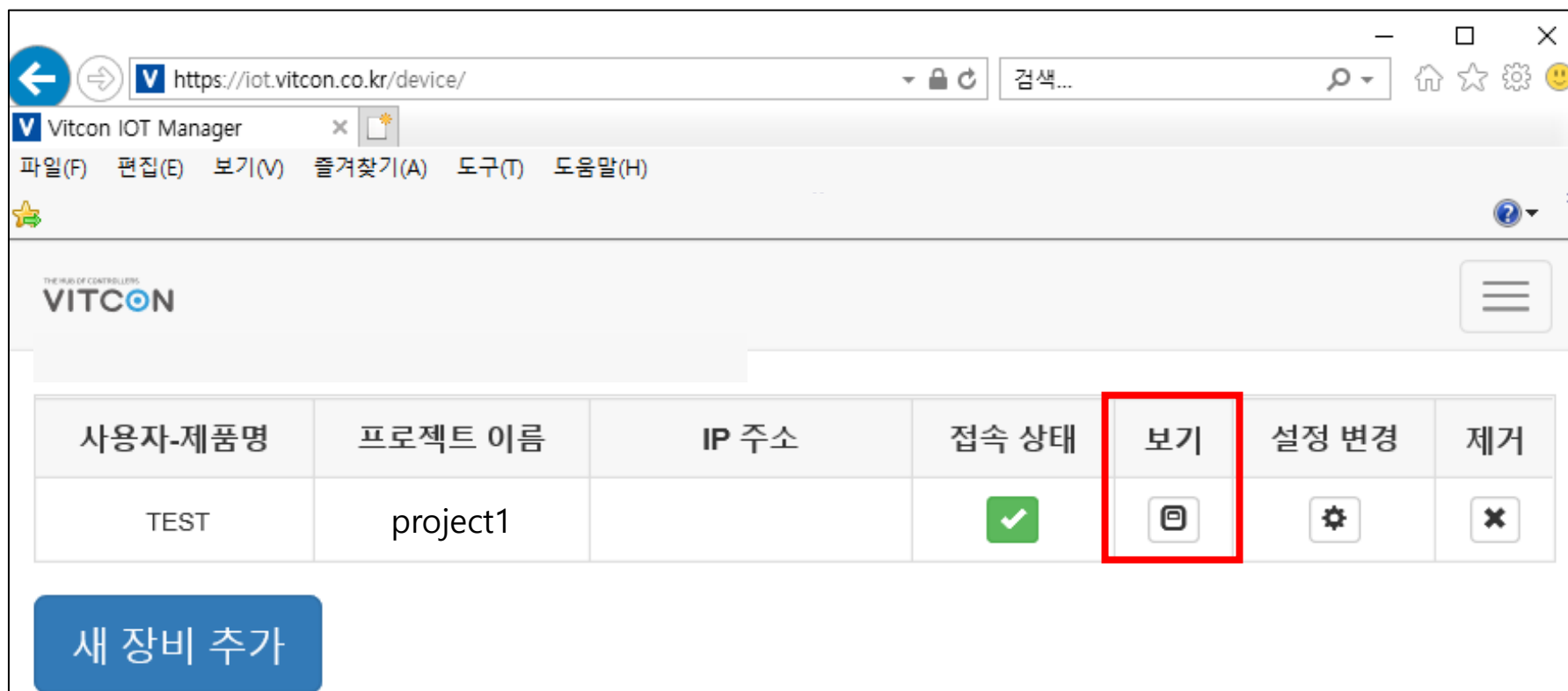
사용자-제품명	프로젝트 이름	IP 주소	접속 상태	보기	설정 변경	제거
TEST	project1		✓			

새 장비 추가




MODLINK 기초 3 - Simple Switch

WEB 모니터링 및 제어하기

13 [보기]를 클릭하여 위젯을 조작해보세요



The screenshot shows a web browser window with the URL <https://iot.vitcon.co.kr/device/>. The page title is "Vitcon IOT Manager". The main content area features a table with the following columns: "사용자-제품명" (User-Product Name), "프로젝트 이름" (Project Name), "IP 주소" (IP Address), "접속 상태" (Connection Status), "보기" (View), "설정 변경" (Change Settings), and "제거" (Remove). The first row of data shows "TEST" for the user-product name, "project1" for the project name, and a green checkmark for the connection status. The "보기" button, which contains a camera icon, is highlighted with a red rectangular box. Below the table is a blue button labeled "새 장비 추가" (Add New Device).

사용자-제품명	프로젝트 이름	IP 주소	접속 상태	보기	설정 변경	제거
TEST	project1		✓			

새 장비 추가

MODLINK 기초 3 - Simple Switch

WEB 모니터링 및 제어하기



토글 버튼
바이너리
상태확인 읽기 인덱스 1개
쓰기 인덱스 1개

위젯	
토글 스위치	
	토글 스위치
	토글 스위치
속성	
Item index	0
Scale	1
Writeindex	1

Item index : 위젯 아이템 번호(상태)

Scale : 위젯 비율 크기

Write index : 위젯 아이템 번호(제어)

실습 1

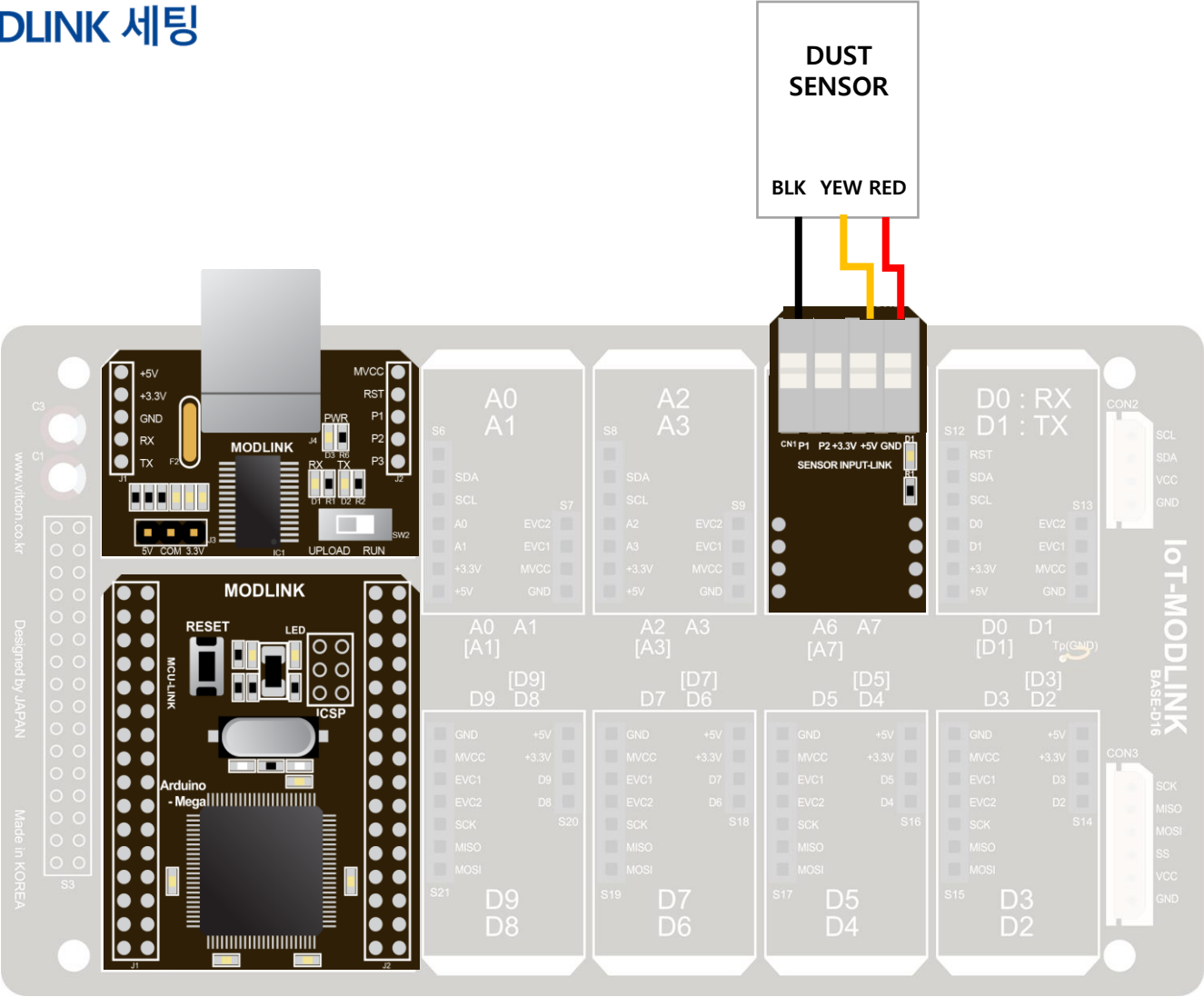
SENSOR INPUT V4-LINK

미세먼지 센서

미세먼지 측정값을 **시리얼 모니터에** 출력해봅시다

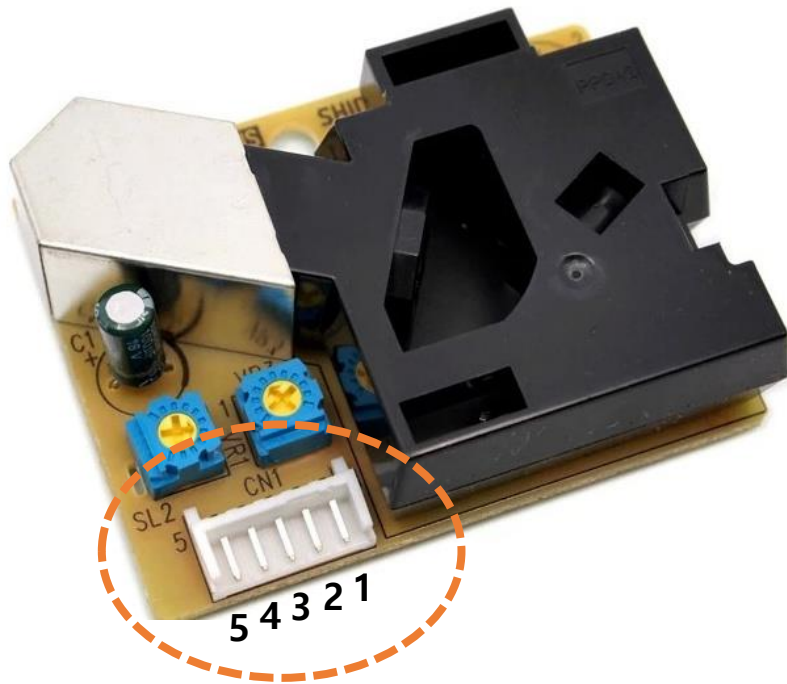
MODLINK실습 1 – 미세먼지센서

MODLINK 세팅



MODLINK실습 1 – 미세먼지센서

MODLINK 세팅



- | | |
|---------------------|----------------------------------|
| 1. COMMON (GND) | --- SENSOR-INPUT-V4-LINK의 GND 단자 |
| 2. OUTPUT(P2) | |
| 3. INPUT(5VDC 90mA) | --- SENSOR-INPUT-V4-LINK의 VCC 단자 |
| 4. OUTPUT(P1) | --- SENSOR-INPUT-V4-LINK의 P1 단자 |
| 5. INPUT(T1) | |

※ IoT 공기청정기 Kit는 1, 3, 4번 핀만 사용합니다.

MODLINK실습 1 – 미세먼지센서

Coding

1 'EDU-2.6.ino' 을 실행해주세요

```
1 #define ppd42ns A6 //먼지센서에 연결된 데이터핀
2
3 unsigned long duration;
4 unsigned long starttime;
5 unsigned long interval = 5000;//sampe 5s
6 unsigned long lowpulseoccupancy = 0;
7
8 float ratio = 0;
9 float cct = 0;           //concentration
10 float ugm3 = 0;         //The final value, in micrograms per cubic meter( $\mu\text{g}/\text{m}^3$ )
11 float pcsPerCF = 0;     //Initialize CF to 0 per particle
```

```
13 void setup() {  
14     Serial.begin(9600);  
15     pinMode(ppd42ns, INPUT); //A6번 핀을 INPUT으로 지정  
16     starttime = millis();    //get the current time;  
17 }
```

MODLINK실습 1 – 미세먼지센서

Coding

```
19 void loop() {
20     duration = pulseIn(ppd42ns, LOW); //센서가 LOW신호를 받을 때의 시간을 저장
21     lowpulseoccupancy = lowpulseoccupancy + duration;
22     //loop에서 ppd42ns의 수광 센서에서 LOW펄스가 입력된 시간을 계속해서 더한다.
23     //단위 : us
24
25     if ( millis() > interval + starttime ) //5초마다 한 번씩 실행
26     {
27         ratio = lowpulseoccupancy / (interval * 10.0);
28         //ratio 단위 : us
29
30         cct = (1.1 * pow(ratio, 3)) - (3.8 * pow(ratio, 2)) + (520 * (ratio + 0.62));
31         // using spec sheet curve
32
33         pcsPerCF = cct * 100;
34         // When the particle concentration is multiplied by 100, the CF value per particle
35
36         ugm3 = pcsPerCF / 13000;
37         // pcs/0.01cf 단위를 ug/m^3 단위로 변환 (dust measurement of micrograms per cubic meter)
38
39         ugm3 = float(int(ugm3 * 10.0f)) / 10.0f; //소수점 두 번째 자리까지 버림
40         ugm3 = min(999, ugm3); //최소값 999
```

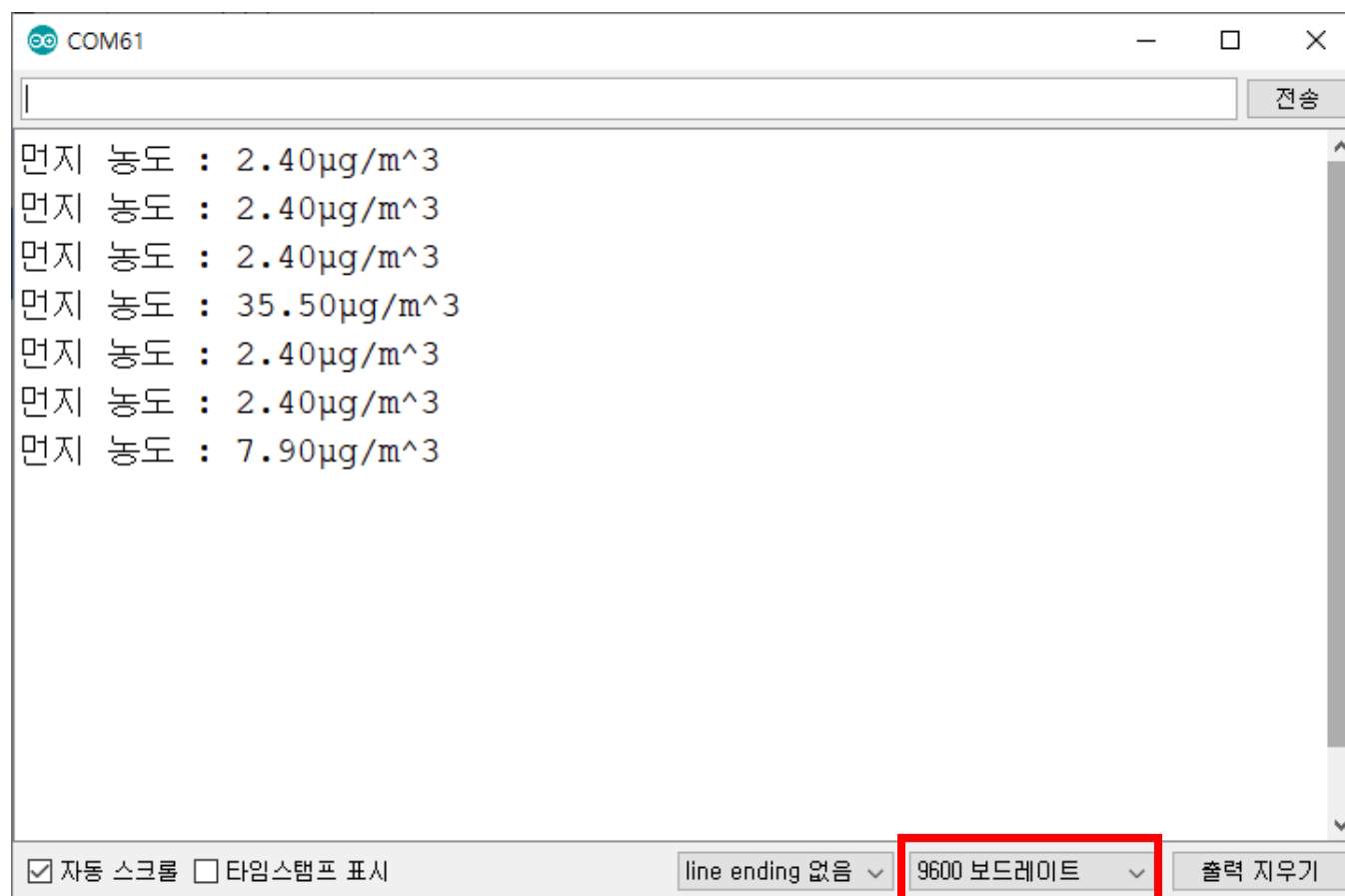


```
53     Serial.print("먼지 농도 : ");
54     Serial.print(ugm3);
55     Serial.println("µg/m^3");
56
57     lowpulseoccupancy = 0;
58     starttime = millis();
59 }
60 }
```

MODLINK실습 1 – 미세먼지센서

LED ON/OFF

2 스케치를 업로드한 후 시리얼 모니터 창에서 결과를 확인합니다.



TR-LINK

EVC-LINK

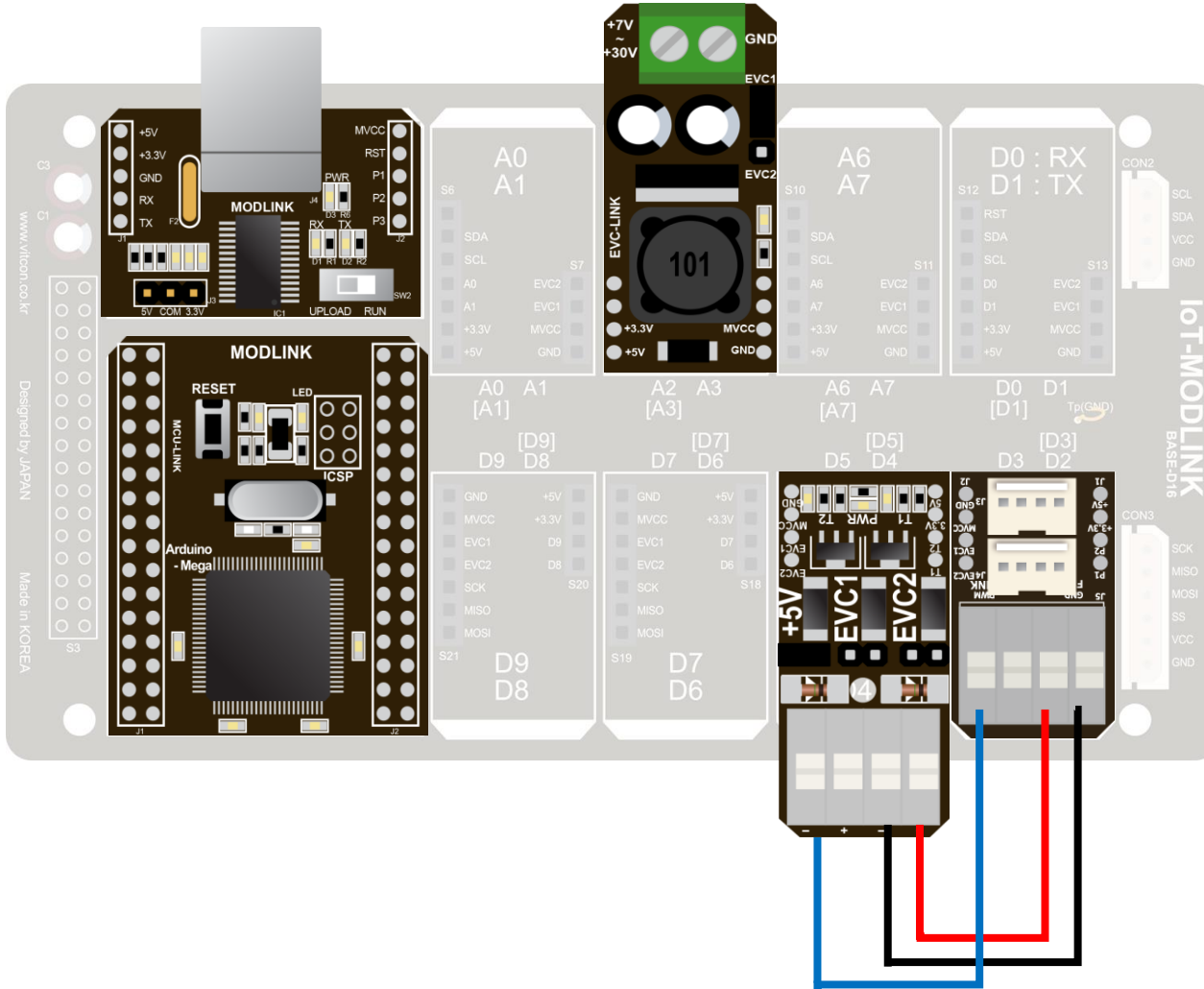
FAN Controller 1-LINK

실습 2

RGB FAN을 ON/OFF 제어해 봅시다

MODLINK실습 2 – RGB FAN

MODLINK 세팅



TR-LINK : D5, D4

EVC-LINK : (A2, A3)

FAN CONTROL 2-LINK : (D3, D2)

MODLINK실습 3 – 미세먼지센서

Coding

1 아두이노 스케치에 코드를 입력하고 업로드 해주세요

```
#define FAN_POWER 4 //팬 전원 핀
#define FAN_PWM 5   //팬 속도 조절 핀

void setup() {
  pinMode(FAN_POWER, OUTPUT); //D4를 출력 핀으로 설정
  pinMode(FAN_PWM, OUTPUT);   //D5를 출력 핀으로 설정
}

void loop()
{
  digitalWrite(FAN_POWER, HIGH); //FAN전원 ON
  analogWrite(FAN_PWM, 100);      //팬 스피드는 PWM 100으로 고정
  delay(1000);                    //2초 딜레이
  digitalWrite(FAN_POWER, HIGH); //FAN전원 OFF
  delay(5000);                    //2초 딜레이
}
```

TR-LINK

EVC-LINK

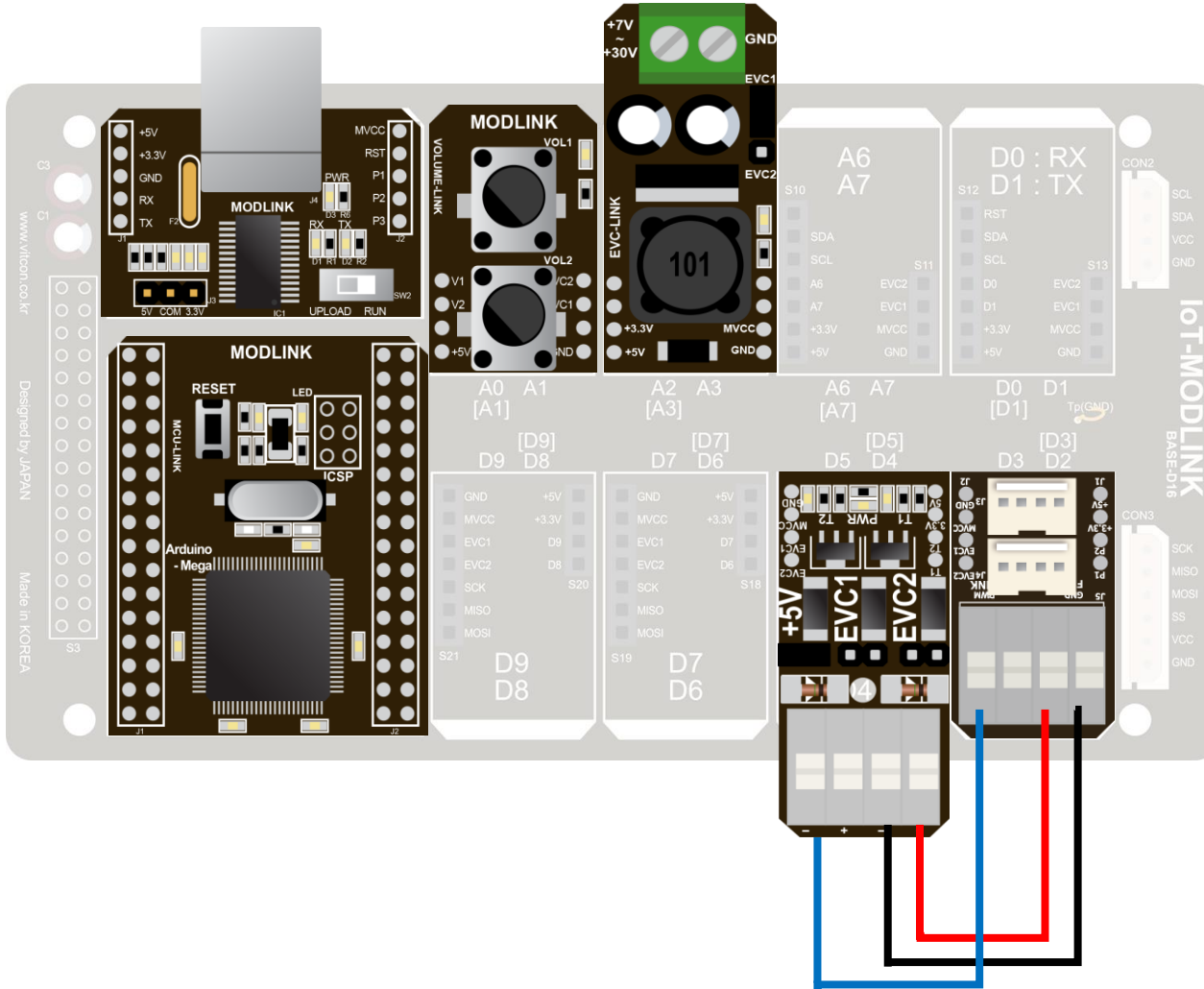
FAN Controller 1-LINK

실습 3

RGB FAN의 속도를 제어해 봅시다

MODLINK실습 3 – RGB FAN

MODLINK 세팅



VOLUME-LINK : A0

TR-LINK : D16, D17

EVC-LINK : (A2, A3)

FAN CONTROL 2-LINK : (D3, D2)

MODLINK실습 3 – RGB FAN

Coding

1 아두이노 스케치에 코드를 입력하고 업로드 해주세요

```
#define FAN_POWER 4 //팬 전원 핀
#define FAN_PWM 5   //팬 속도 조절 핀

void setup() {
  pinMode(FAN_POWER, OUTPUT); //출력 핀으로 설정
  pinMode(FAN_PWM, OUTPUT);   //출력 핀으로 설정
  digitalWrite(FAN_POWER, HIGH); //FAN전원 ON 고정
}

void loop()
{
  int FanSpeed = map(analogRead(A0), 0, 1024, 200, 30);
  //200 : slow speed, 30 : fast speed

  analogWrite(FAN_PWM, FanSpeed);
}
```


TR-LINK

EVC-LINK

FAN Controller 2-LINK

실습 4

RGB FAN의 RGB를 제어해 봅시다 -1

MODLINK 세팅



Coding

1 아두이노 스케치에 코드를 입력하고 업로드 해주세요

```
#define RED 9 //팬 Red핀
#define BLUE 7 //팬 Blue핀
#define GREEN 6 //팬 Green핀
void setup() {
  pinMode(RED, OUTPUT); //출력 핀으로 설정
  pinMode(BLUE, OUTPUT); //출력 핀으로 설정
  pinMode(GREEN, OUTPUT); //출력 핀으로 설정
}
void loop() {
  digitalWrite(RED, HIGH); //빨간색 출력
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
  delay(2000); //2초 딜레이
  digitalWrite(RED, LOW); //초록색 출력
  digitalWrite(GREEN, HIGH);
  digitalWrite(BLUE, LOW);
  delay(2000); //2초 딜레이
  digitalWrite(RED, LOW); //파란색 출력
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, HIGH);
  delay(2000); //2초 딜레이
}
```

TR-LINK

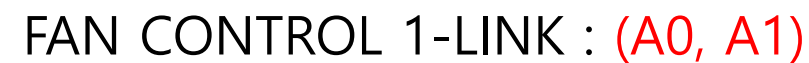
EVC-LINK

FAN Controller 2-LINK

실습 5

RGB FAN의 **RGB**를 제어해 봅시다 -2

MODLINK 세팅



EVC-LINK : (A2, A3)

TR-LINK : D9, D7, D6

1 실습4의 코드를 수정합니다.

```
#define RED 9 //팬 Red핀
#define BLUE 7 //팬 Blue핀
#define GREEN 6 //팬 Green핀

void setup() {
  pinMode(RED, OUTPUT);    //출력 핀으로 설정
  pinMode(BLUE, OUTPUT);   //출력 핀으로 설정
  pinMode(GREEN, OUTPUT);  /출력 핀으로 설정

  Serial.begin(9600);
  Serial.setTimeout(100);
}
```

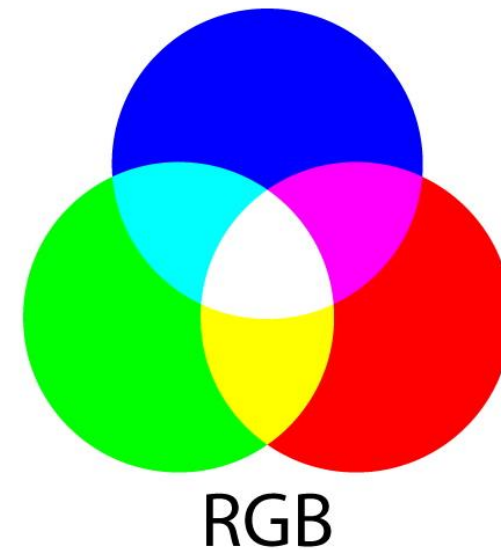
1 실습4의 코드를 수정합니다.

```
char color;
String rgbValue;

void loop() {
  if (Serial.available() > 0)
  {
    String str = Serial.readString(); //시리얼 입력 값을 문자열로 반환
    color = str.charAt(0); //문자열의 첫 문자 반환
    rgbValue = str.substring(1); //문자열의 두 번째 자리부터 반환

    int pwmValue = rgbValue.toInt(); //문자열을 int형으로 변환

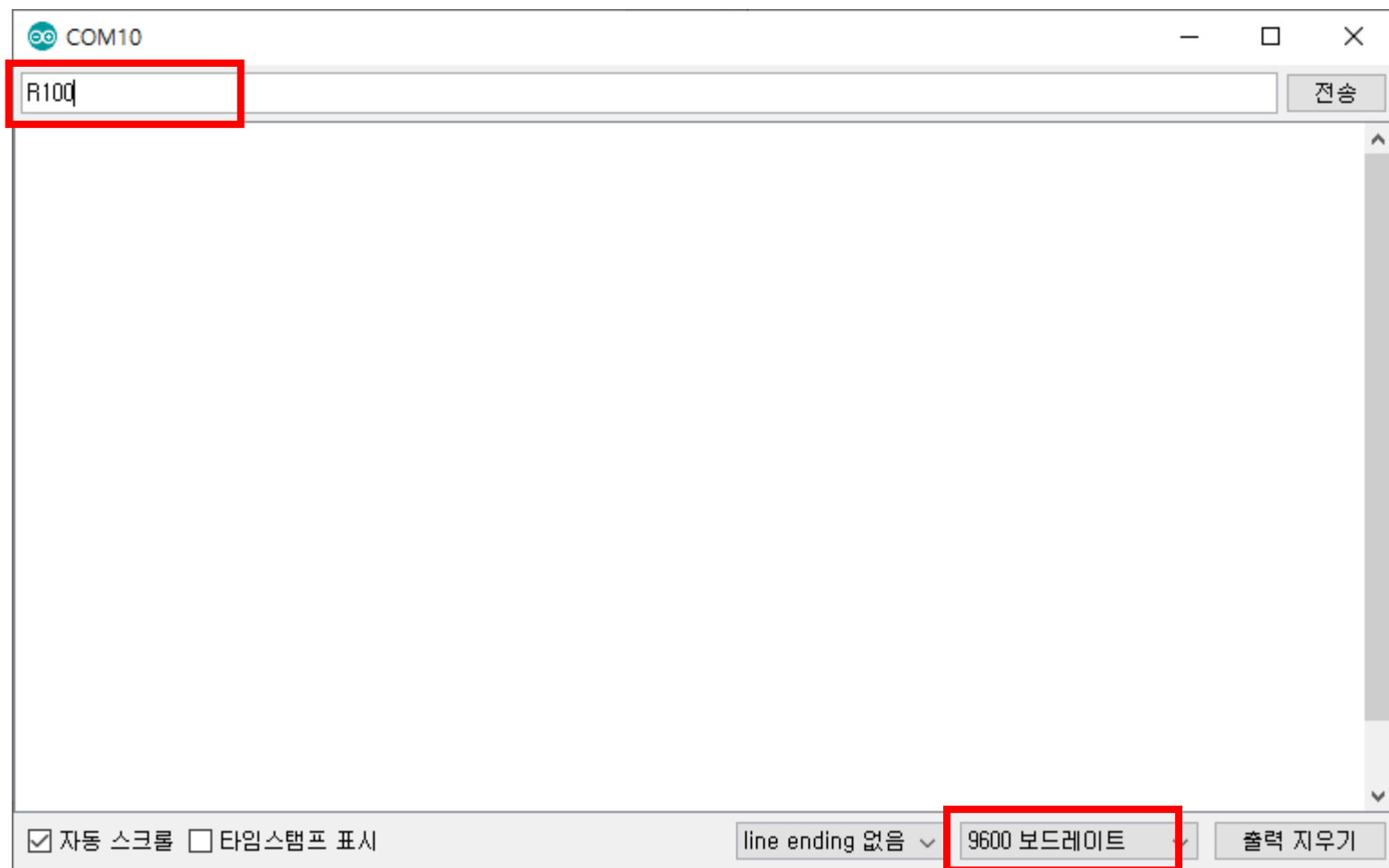
    if (color == 'R') { //첫 문자가 R일 때
      analogWrite(RED, pwmValue); //D9핀에 pwmval 출력
    }
    else if (color == 'G') { //첫 문자가 G일 때
      analogWrite(GREEN, pwmValue); //D7핀에 pwmval 출력
    }
    else if (color == 'B') { //첫 문자가 B일 때
      analogWrite(BLUE, pwmValue); //D6핀에 pwmval 출력
    }
  }
}
```



MODLINK실습 5 – RGB FAN

Coding

2 시리얼 모니터 입력창에 원하는 색의 알파벳(RED : R, GREEN : G, BLUE : B)과 PWM (0~255) 입력 후 전송합니다.

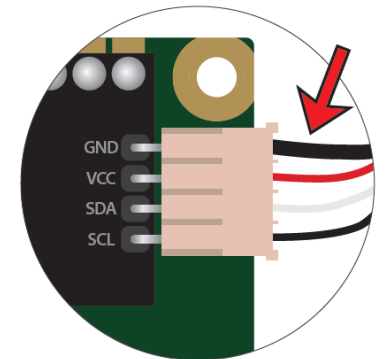
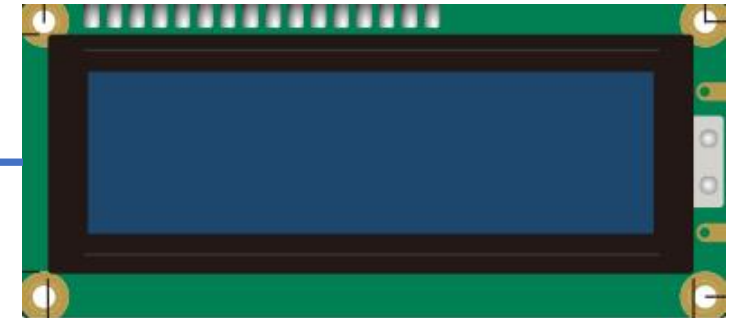


CLCD

실습 6

CLCD 에 문자를 출력해봅시다.

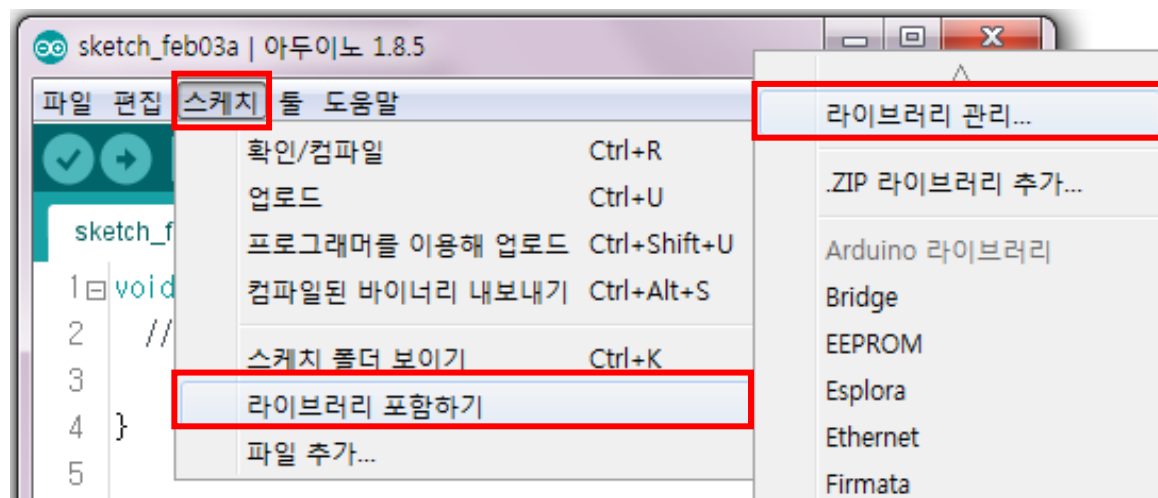
MODLINK 세팅



MODLINK실습 6 – CLCD

라이브러리 설치

1 [스케치] → [라이브러리 포함하기] → [라이브러리 관리]를 클릭해주세요.



2 [라이브러리 매니저]에서 'LiquidCrystal I2C by Frank de Brabander' 을 검색하여 설치해주세요



Coding

3 아두이노 스케치에 코드를 입력하고 업로드 해주세요

```
/*CLCD 사용을 위한 라이브러리 추가*/
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); //또는 LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup() {
  lcd.init();           //CLCD 초기화
  lcd.backlight();       //백라이트 ON (밝기는 CLCD 뒤쪽의 가변저항으로 조절합니다.)
  lcd.setCursor(0, 0);   //좌표 지정
  lcd.print("HELLO");//문자열 출력

  lcd.setCursor(0, 1);   //좌표 지정
  lcd.print("VITCON!");  //문자열 출력
}

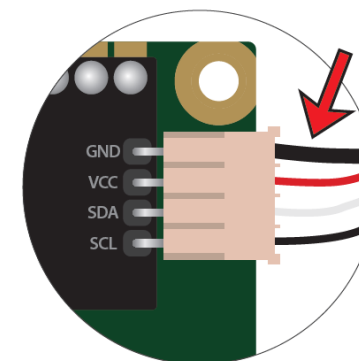
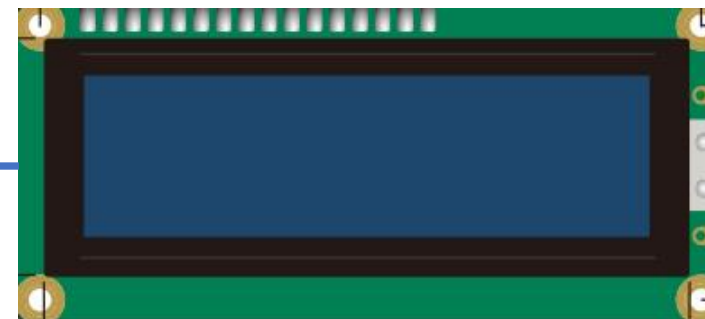
void loop() {
}
```

CLCD

실습 7

CLCD 에 이모티콘 만들어 출력하기

MODLINK 세팅



1 실습 6 코드를 수정합니다.

```
/*CLCD 사용을 위한 라이브러리 추가*/
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); //또는 LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup() {
  lcd.init();//CLCD 초기화
  lcd.backlight();          //백라이트 ON (밝기는 CLCD 뒤쪽의 가변저항으로 조절합니
                             다.)
  lcd.setCursor(0, 0);          //좌표 지정
  lcd.print("HELLO VITCON!"); //문자열 출력

  /*사용자 정의문자를 메모리에 지정, 최대 8개까지 정의 가능*/
  lcd.createChar(0, crying_eye); //우는 얼굴의 눈
  lcd.createChar(1, crying_mouth); //우는 얼굴의 입

  crying_face(); //만든 이모티콘 CLCD에 출력
}
```

MODLINK실습 7 – RGB FAN

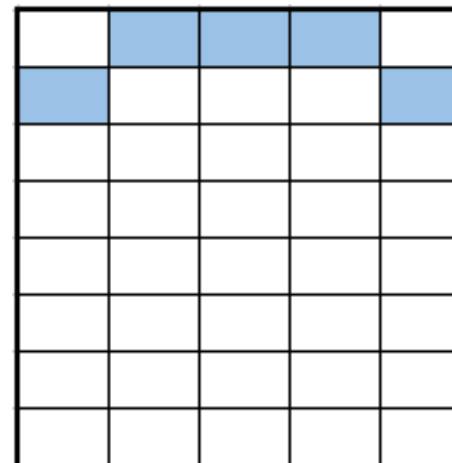
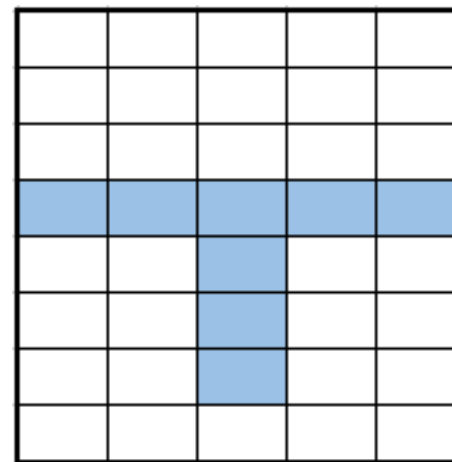
Coding

1 실습 5 코드를 수정합니다.

```
byte crying_eye[8] = { // 우는 얼굴의 눈
    B00000,
    B00000,
    B00000,
    B11111,
    B00100,
    B00100,
    B00100,
    B00000
};

byte crying_mouth[8] = { // 우는 얼굴의 입
    B01110,
    B10001,
    B00000,
    B00000,
    B00000,
    B00000,
    B00000,
    B00000
};

void setup() {
}
```



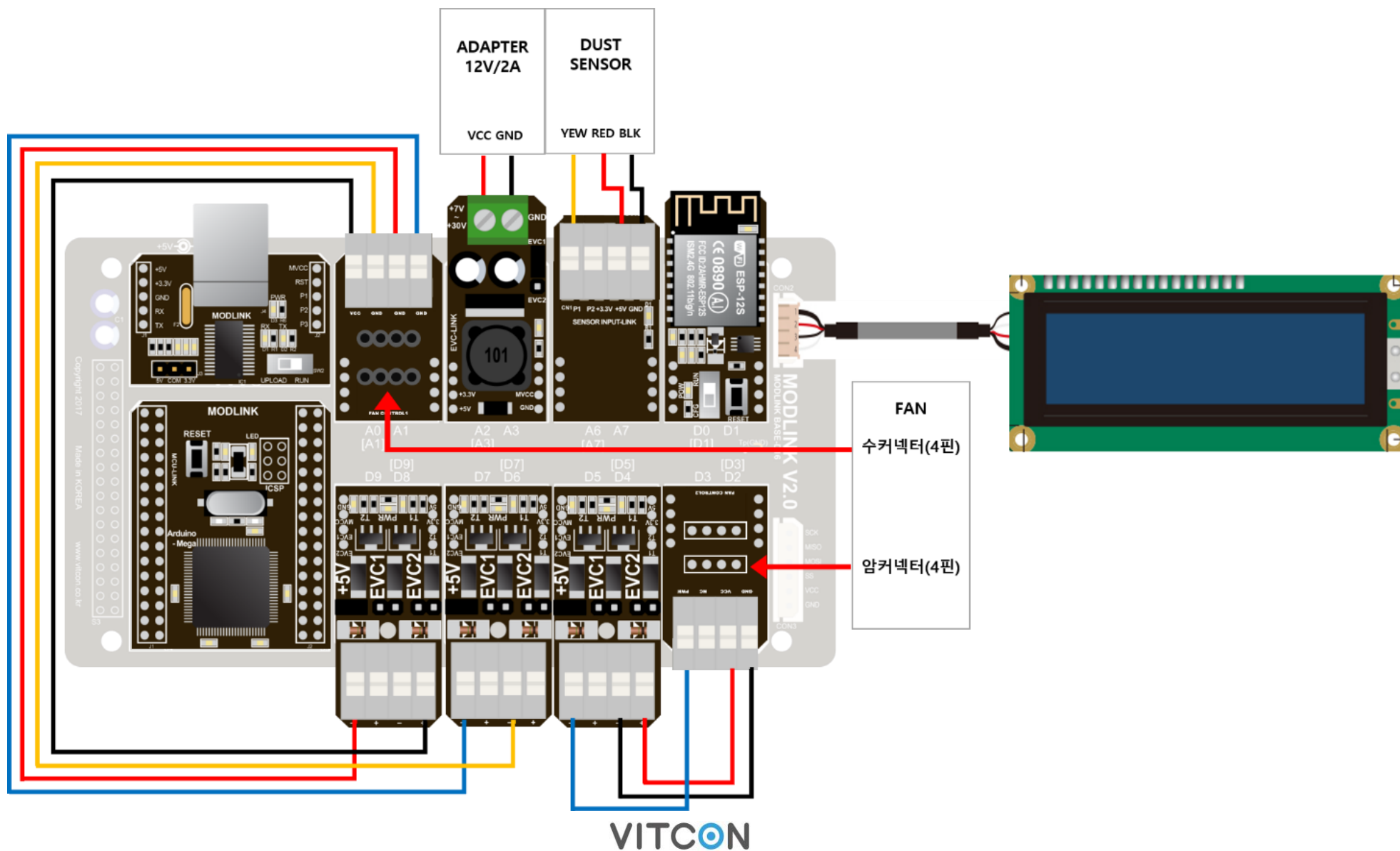
1 실습 5 코드를 수정하고 업로드합니다.

```
void loop() {  
}  
  
void crying_face()           //이모티콘 함수  
{  
  lcd.setCursor(12, 0);      //좌표 지정  
  lcd.write(0);              //사용자 정의문자 1번 출력  
  lcd.setCursor(14, 0);      //사용자 정의문자 1번 출력  
  lcd.write(0);              //사용자 정의문자 1번 출력  
  lcd.setCursor(13, 1);      //사용자 정의문자 2번 출력  
  lcd.write(1);              //사용자 정의문자 2번 출력  
}
```

실습 8 IoT 공기청정기를 만들어 봅시다

MODLINK실습 8 – 공기청정기

MODLINK 세팅



MODLINK실습 8 – 공기청정기

Coding

- 1 <https://iot.vitcon.co.kr/> 접속 및 로그인 해 주세요.
- 2 [프로젝트 목록] → [새 프로젝트 추가] -> [공기청정기] 프리셋을 선택해주세요

프로젝트 추가

프로젝트 이름:

프로젝트 이름

프리셋 선택:

(Custom)

(Custom)

0725_Test

Automatic Test System

FCU Remote

FCU Remote old

IoT Portable Toilet

IoT 공기청정기

IoT 소방 화재 감지기

IoT 스마트 팜

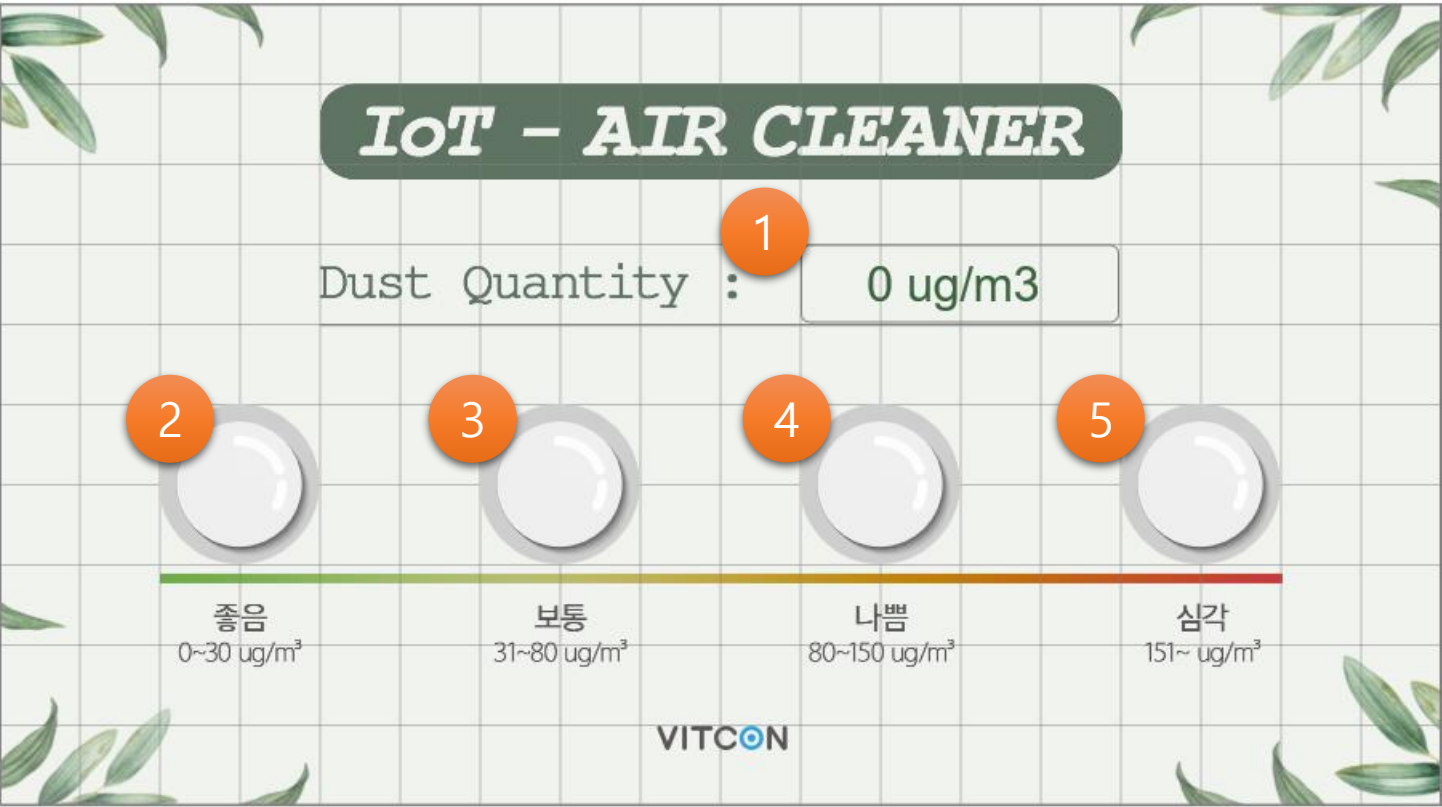
IoT 스마트 홀

LS산전 XBC-DN32H

PushAlarm

Remote PWM

3 프로젝트 수정 클릭



번호	위젯 종류	Item Index Num.	Write Index Num.
1	숫자/문자열	0	-
2	LED	1	-
3	LED	2	-
4	LED	3	-
5	LED	4	-

문자열 표시
인덱스 없음

위젯

라벨

label라벨

속성

Backcolor	#0000ff
FontSize	15
Forecolor	#ffffff
Height	1
Label	label_1_25
Scale	1
Textalign	Left
Width	3

Backcolor : 배경색

FontSize : 글씨크기

Forecolor : 글씨색

Height : 위젯 높이


Label : 사용할 문구

Scale : 위젯 비율 크기

Textalign : 글자 정렬

Width : 위젯 넓이

LED
바이너리
읽기 인덱스 1개

위젯	
<u>LED</u>	
	LED
속성	
Item index	<input type="text" value="11"/>
Backcolor	<input type="text" value="#0000ff"/>
Scale	<input type="text" value="1"/>

Item index : 위젯 아이템 번호(상태)

Backcolor : 배경색

Scale : 위젯 비율 크기

IoT 위젯 종류 – 숫자/문자열

숫자/텍스트 표시
정수/실수/문자열
읽기 인덱스 1개

위젯	
숫자/문자열	
	숫자/문자열
	0
속성	
Item index	13
BackColor	#ffffff
Firsttext	
FontSize	36 ▼
Forecolor	#000000
Height	2 ▼
Lasttext	°C
Scale	1 ▼
Width	3 ▼

Item index : 위젯 아이템 번호(상태)

BackColor : 배경색

Firsttext : 데이터 앞에 표시할 글자

FontSize : 글씨크기

Forecolor : 글씨색

Height : 위젯 높이

Lasttext : 데이터 뒤에 표시할 글자

Scale : 위젯 비율 크기

Textalign : 글자 정렬

Width : 위젯 넓이

MODLINK실습 8 – 공기청정기

Coding

1 복사해둔 Device id를 붙여넣은 후 코드를 업로드 해 주세요.

```
36 /* IOT server communication manager */  
37 const char device_id[] = ""; // Change device_id to yours
```

2 장비 접속상태 확인 및 보기를 눌러 위젯을 모니터링합니다.





감사합니다.

THE HUB OF IoT
VITCON