

PART II. MODLINK IoT

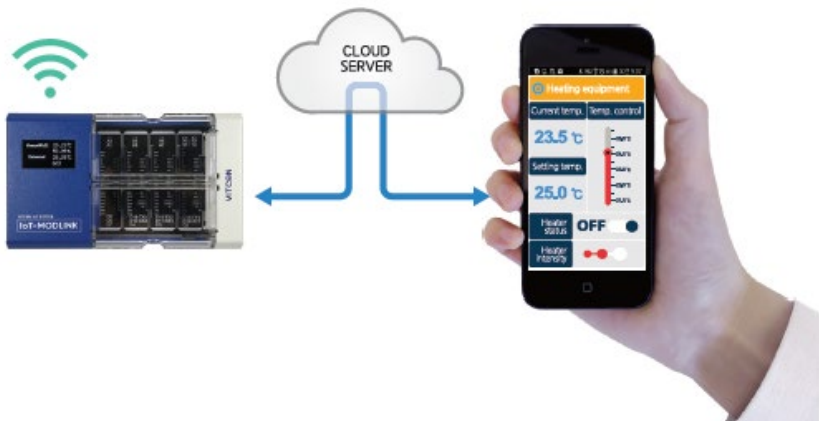
IoT 서비스 사용 설정



IoT 서비스 사용 환경

MODLINK를 사용한 IoT 환경 구성

- MODLINK 및 LINK 모듈을 사용한 End-Device
- Vitcon IoT 클라우드 서버
- 스마트폰 어플리케이션 또는 Web 기반 모니터링/제어 어플리케이션



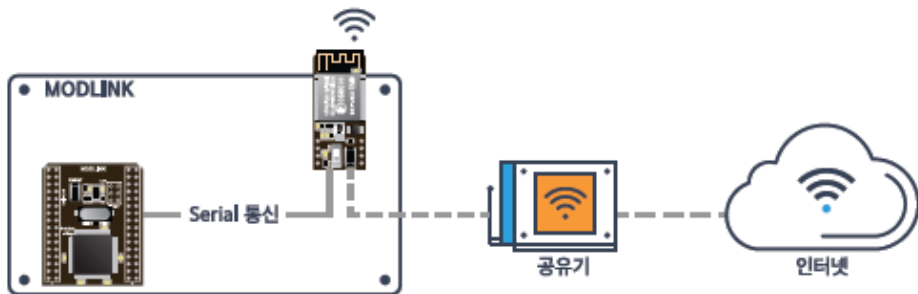
IoT 서비스 사용 설정



WiFi-LINK

IoT 구현하기 위한 네트워크 인터페이스

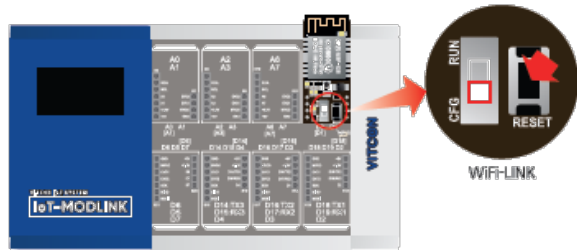
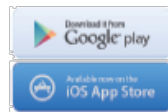
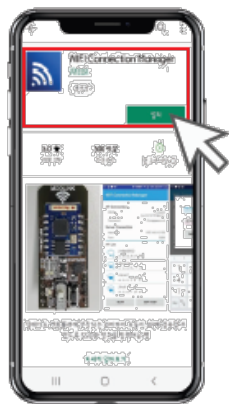
- MODLINK가 인터넷에 연결되어 클라우드 서버와의 통신 기능이 필수
- WiFi-LINK가 주변에 설치된 공유기를 이용해 장치를 인터넷에 연결
- WiFi-LINK는 MODLINK와 시리얼 통신
: 시리얼 통신이 가능한 포트에 장착
- 연결모드, AP 및 HOST 설정은 스마트폰 어플리케이션을 사용



WiFi-LINK 설정

WiFi 접속 설정

- MODLINK를 PC와 연결 후, 선택 스위치를 UPLOAD에 위치
- WiFi-LINK가 D0, D1(시리얼 통신 포트)에 장착되어 있는지 확인
- 스마트폰에서 “VITCON” 검색
: WIFI Connection Manager 설치
- WiFi-LINK의 스위치를 CFG에 맞춘 후, RESET 버튼



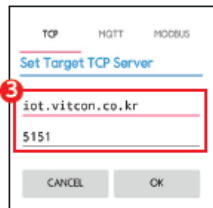
WiFi-LINK 설정

WiFi 접속 설정

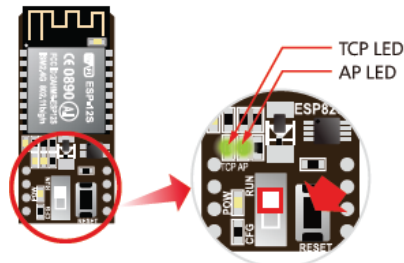
- 스마트폰 WiFi 목록에서 Vitcon_EXP12S***** 선택
: 다수의 기기가 잡힐 수 있으므로, 신호가 가장 크게 잡히는 것 선택
: 주변 MODLINK의 전원을 해제하고 접속 (순차적으로 접속)
- 설치했던 WIFI Connection Manager 어플리케이션 실행 및 설정



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



- WiFi-LINK 스위치를 RUN에 맞춘 후, RESET 버튼
: TCP와 AP LED에 녹색 불이 켜지면 정상적으로 인터넷에 연결



클라우드 서버 설정

Vitcon Cloud 서버

- <http://iot.vitcon.co.kr>에 접속 후 회원가입

The image displays two screenshots of the Vitcon Cloud website interface. The left screenshot shows the login page with fields for '이메일' (Email) and 'Password', a '로그인' (Login) button, and a '회원가입' (Sign Up) button. The right screenshot shows the registration page with fields for '이메일' (Email), '비밀번호' (Password), '비밀번호 확인' (Confirm Password), '이름' (Name), '성명' (Surname), and '휴대폰번호' (Mobile Number), along with '회원가입' (Sign Up) and '로그인' (Login) buttons.

로그인

이메일
Password
☐ 아이디 저장 ☐ 자동 로그인
로그인
회원가입
약관동의 확인

회원가입

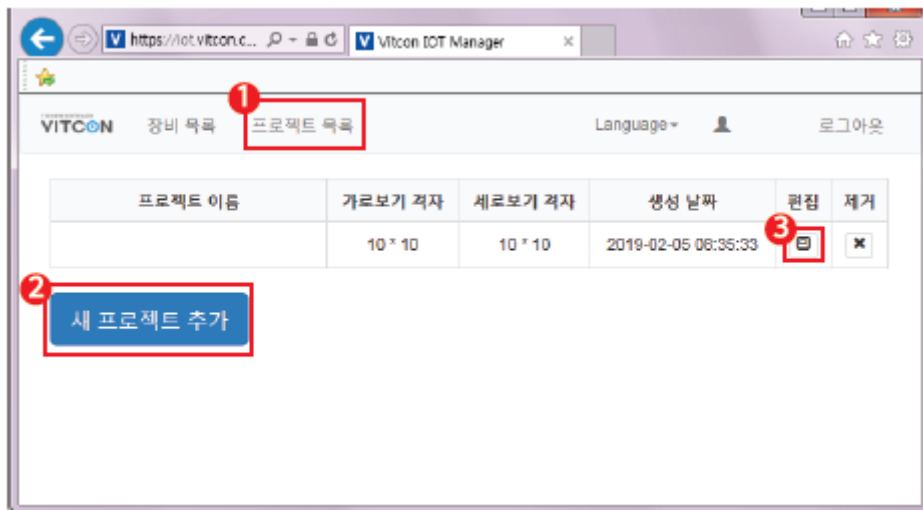
이메일
비밀번호
비밀번호 확인
이름
성명
휴대폰번호
회원가입
로그인

클라우드 서버 설정



Vitcon Cloud 서버 프로젝트 생성 및 편집

- 로그인 후, '프로젝트 목록'의 새 프로젝트 추가
: [이름, 격자 크기 제한 없음] - 원하는 이름으로 프로젝트 생성 후 '편집' 클릭



클라우드 서버 설정

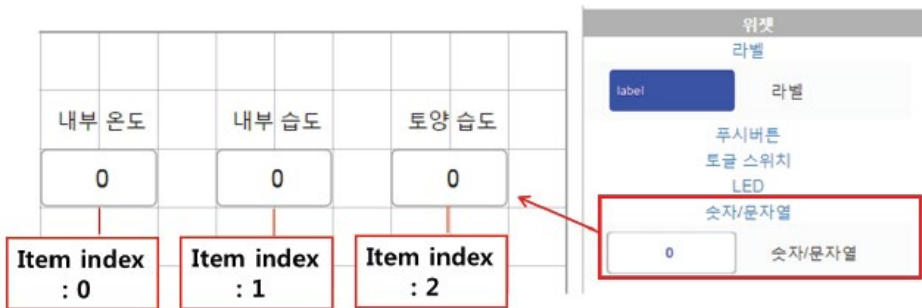


Vitcon Cloud 서버 프로젝트 생성 및 편집

- Drag & Drop 방식으로 '위젯'의 '라벨' 을 생성
- 속성에서 내부 온도, 내부 습도, 토양 습도를 각각 입력 후 저장 (글자 크기/배경색 등 선택 가능)



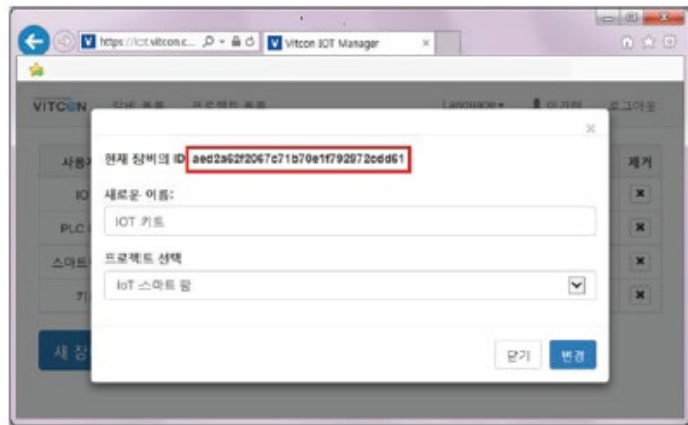
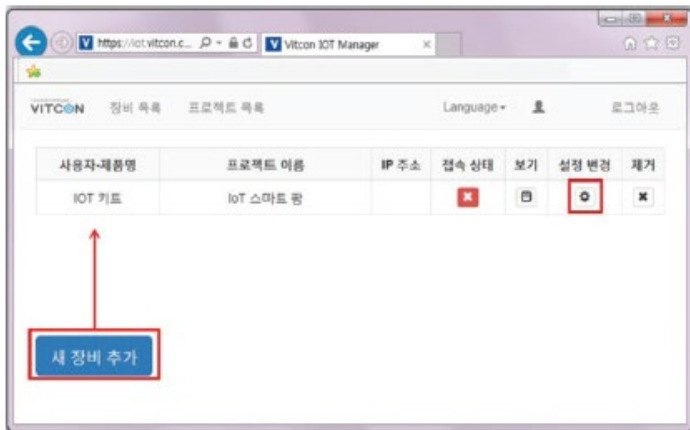
- 숫자/문자열 위젯도 같은 방식으로 생성 후 저장 - 메인화면 이동
: 위젯 생성 순서에 따라 자동으로 Item Index가 할당



클라우드 서버 설정

Vitcon Cloud 서버 프로젝트 생성 및 편집

- 장비 목록에서 새 장비를 추가
: [이름 제한 없음]을 클릭하여 프로젝트 생성 후, 설정 변경에서 만들었던 프로젝트 선택
- 설정 변경에서 현재 장비의 ID 복사 – 각 팀에서 사용하는 MODLINK의 장비 고유 ID



Example I - Vitcon IoT



IoT 서비스 사용을 위한 기본 코드

```
#include <VitconBrokerComm.h>
using namespace vitcon;

IOTItemFlo dht22_temp;           // 서버 위젯과 연동되는 vitcon namespace의 데이터 타입
IOTItemInt soilhumi;            // IOTItem[type]으로 설정

#define ITEM_COUNT 2             // 측정/제어할 IoT 아이템의 개수를 명시
IOTItem *items[ITEM_COUNT] = {&dht22_temp, &soilhumi}; // 서버 위젯 인덱스와 연동

const char device_id[ ] = "각자의 MODLINK ID";
BrokerComm comm(&Serial, device_id, items, ITEM_COUNT); //서버 통신 객체 생성

...in void loop()
dht22_temp.Set(Temp);           // MODLINK에서 측정된 값을 서버 위젯 연동 변수에 세팅
soil_humi.Set(soilhumi);        // 연동 변수에 세팅된 값이 서버로 전송되어 기록
```

Example I - IoT 모니터링



온/습도 및 토양 습도 모니터링

```
#include "DHT.h"
#include <VitconBrokerComm.h>
using namespace vitcon;

#define DHTPIN A1      // Digital pin connected to the DHT sensor
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
#define SOILHUMI A6

// Initialize DHT sensor.
DHT dht(DHTPIN, DHTTYPE);

uint32_t DataCaptureDelay = 2000; //ms
uint32_t DataCapture_ST = 0; //Start Time
int Soilhumi = 0;
float Temp;
float Humi;
/*widget label*/
IOTItemFlo dht22_temp;
IOTItemFlo dht22_humi;
IOTItemInt soilhumi;
```

Example I - IoT 모니터링



온/습도 및 토양 습도 모니터링

```
/* A set of definition for IOT items */
#define ITEM_COUNT 3

IOTItem *items[ITEM_COUNT] = { &dht22_temp, &dht22_humi, &soilhumi }; //서버 위젯 인덱스
와 연동

/* IOT server communication manager */
const char device_id[] = "복사한 현재 장비의 ID"; // Change device_id to yours
BrokerComm comm(&Serial, device_id, items, ITEM_COUNT);

void setup() {
    Serial.begin(250000); //서버와의 통신을 위한 속도 지정
    comm.SetInterval(200);

    dht.begin();
    pinMode(SOILHUMI, INPUT);
    DataCapture_ST = millis();
}
```



Example I - IoT 모니터링

온/습도 및 토양 습도 모니터링

```
void loop() {  
  Soilhumi = map(analogRead(SOILHUMI), 0, 1023, 100, 0);  
  
  if (millis() - DataCapture_ST > DataCaptureDelay) { //2초 간격으로 실행  
    Humi = dht.readHumidity();  
    // Read temperature as Celsius (the default)  
    Temp = dht.readTemperature();  
  
    DataCapture_ST = millis();  
  }  
  
  //MODLINK에서 읽은 값을 서버로 전송  
  dht22_temp.Set(Temp);  
  dht22_humi.Set(Humi);  
  soilhumi.Set(Soilhumi);  
  comm.Run();  
}
```

코드 업로드 시 주의사항!

코드를 업로드 할 시, WiFi-LINK 스위치를 CFG로 놓아야함

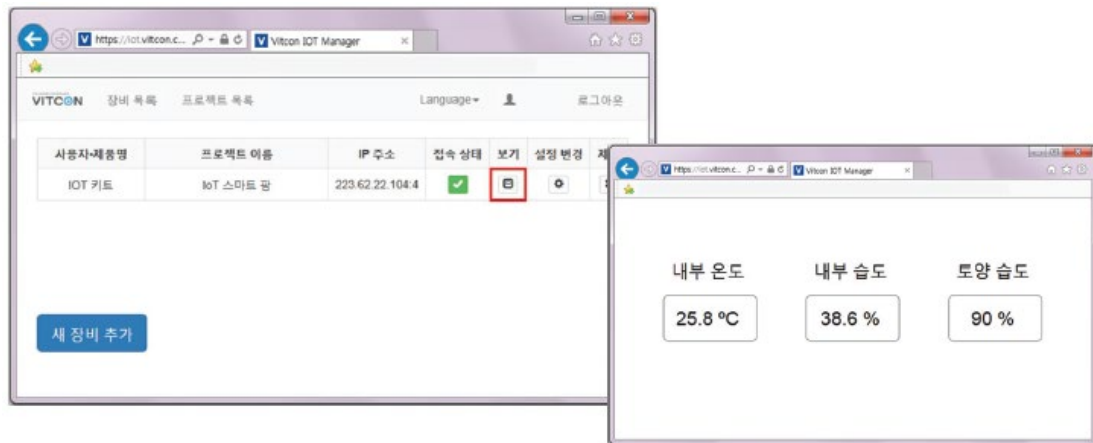
업로드 후 스위치를 다시 RUN으로 둔 후, RESET 버튼을 눌러 접속 상태 확인

Example I - IoT 모니터링



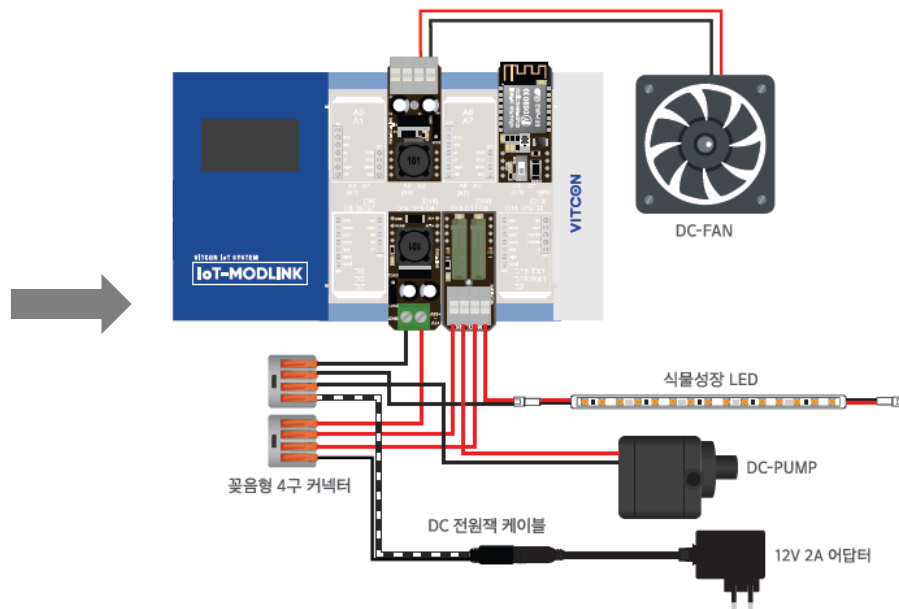
온/습도 및 토양 습도 모니터링

- Vitcon 웹 페이지에서 장비 접속 상태 확인 및 [보기]를 클릭하여 위젯 모니터링
- Play 스토어에서 VITCON을 검색하여 IoT Modlink 설치
(혹은, 스마트폰 웹 브라우저에서 iot.vitcon.co.kr에 접속)
- 로그인 하여 스마트폰으로 모니터링



A top-down view of a person's hands working on a laptop. The person has red-painted fingernails and is wearing a light blue shirt. They are typing on the laptop keyboard. To the left of the laptop, there is a white coffee cup on a saucer with a spoon, and several colorful pens (yellow, orange, pink, blue) are scattered on the dark desk surface. A stack of papers is also visible near the person's left hand. The background is a dark, textured surface.

IoT 제어 준비

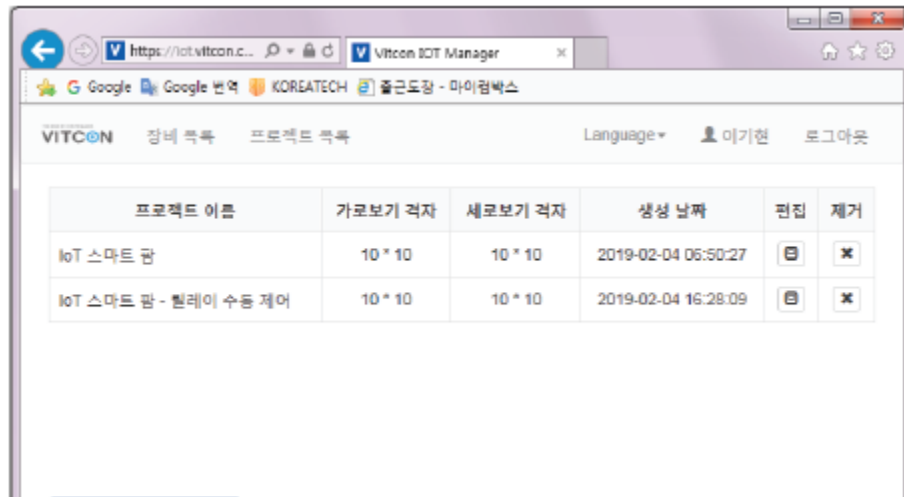


Example II - IoT 제어



서버 프로젝트 생성 및 편집

- 프로젝트 목록의 새 프로젝트 추가 후 편집 클릭

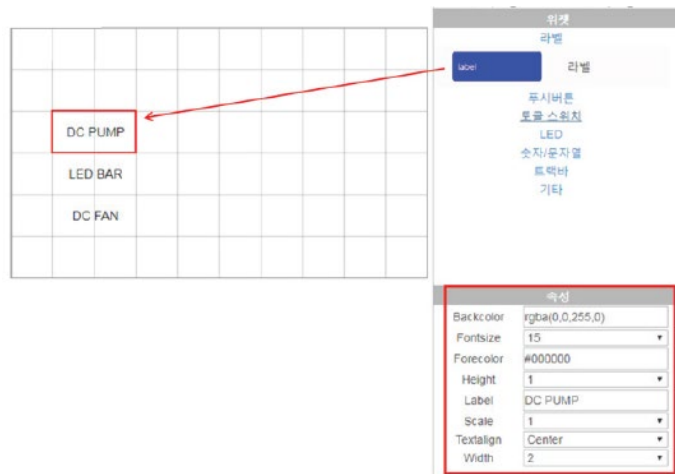
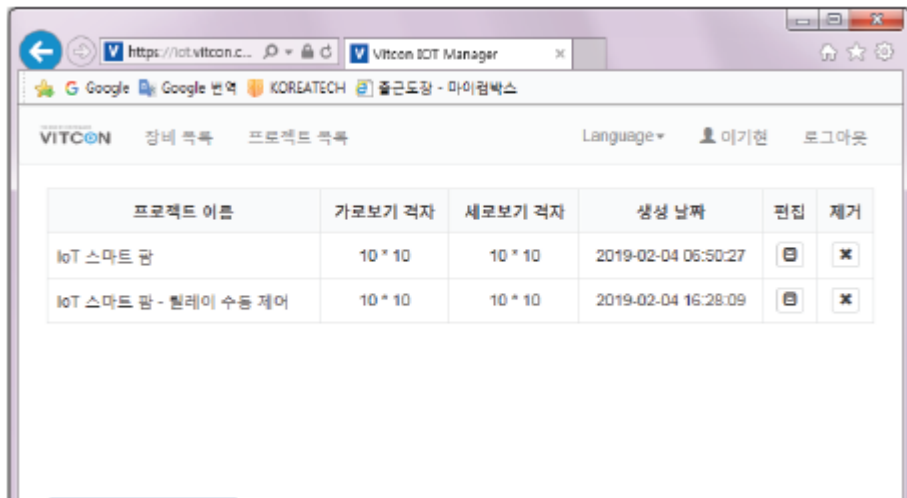


Example II - IoT 제어



서버 프로젝트 생성 및 편집

- 프로젝트 목록의 새 프로젝트 추가 후 편집 클릭
- 위젯 라벨을 생성 후 각 설정 저장



Example II - IoT 제어



서버 프로젝트 생성 및 편집

- 토글 스위치 위젯 추가



- 장비 목록에서 프로젝트 선택(Example I 과정 반복)

Example II - IoT 제어



DC FAN, DC PUMP, LED 제어

```
#include <VitconBrokerComm.h>
using namespace vitcon;
#include <SoftPWM.h>

#define LAMP 17
#define PUMP 16
SOFTPWM_DEFINE_CHANNEL(A3); //SoftPWM으로 사용할 핀 설정

bool fan_out_status;
bool pump_out_status;
bool lamp_out_status;

//manual mode일 때 FAN을 제어하는 함수
void fan_out(bool val) {
    fan_out_status = val;
}

//manual mode일 때 PUMP를 제어하는 함수
void pump_out(bool val) {
    pump_out_status = val;
}

//manual mode일 때 LAMP를 제어하는 함수
void lamp_out(bool val) {
    lamp_out_status = val;
}
```

Example II - IoT 제어



DC FAN, DC PUMP, LED 제어

```
/*widget toggle switch*/
IOTItemBin FanStatus;

IOTItemBin Fan(fan_out);

IOTItemBin PumpStatus;
IOTItemBin Pump(pump_out);

IOTItemBin LampStatus;
IOTItemBin Lamp(lamp_out);

/* A set of definition for IOT items */
#define ITEM_COUNT 6

//서버 위젯 인덱스와 연동
IOTItem *items[ITEM_COUNT] = { &FanStatus, &Fan,
                                &PumpStatus, &Pump,
                                &LampStatus, &Lamp
                                };
```

코드 업로드 시 주의사항!

코드를 업로드 할 시, WiFi-LINK 스위치를 CFG로 놓아야함

업로드 후 스위치를 다시 RUN으로 둔 후, RESET 버튼을 눌러 접속 상태 확인

Example II - IoT 제어



DC FAN, DC PUMP, LED 제어

```
/* IOT server communication manager */
const char device_id[] = "복사한 현재 장비의 ID"; // Change device_id to yours
BrokerComm comm(&Serial, device_id, items, ITEM_COUNT);

void setup() {
  Serial.begin(250000); //서버와의 통신을 위한 속도 지정
  comm.SetInterval(200);

  SoftPWM.begin(490); //PWM frequency 설정
  pinMode(LAMP, OUTPUT);
  pinMode(PUMP, OUTPUT);
}

void loop() {
  if (fan_out_status == true) SoftPWM.set(100); //duty rate 100%, DC FAN 켜짐
  else SoftPWM.set(0); //duty rate 0%, DC FAN 꺼짐
  digitalWrite(PUMP, pump_out_status);
  digitalWrite(LAMP, lamp_out_status);

  FanStatus.Set(fan_out_status);
  LampStatus.Set(digitalRead(LAMP));
  PumpStatus.Set(digitalRead(PUMP));
  comm.Run();
}
```

Example II - IoT 제어



DC FAN, DC PUMP, LED 제어

- Vitcon 웹 페이지에서 장비 접속 상태 확인 및 [보기]를 클릭하여 위젯 모니터링
- Play 스토어에서 VITCON을 검색하여 IoT Modlink 설치
(혹은, 스마트폰 웹 브라우저에서 iot.vitcon.co.kr에 접속)
- 로그인 하여 스마트폰으로 모니터링



Exercise I - IoT 모니터링/제어

장치 자동 제어

- 1) 장치 자동 제어를 ON/OFF 토글 스위치로 입력받는다.
- 2) 앞서 작성했던, 온도에 따른 DC FAN 제어와 토양 습도 조건에 따른 DC PUMP 제어를 실행/중지시키는 프로젝트를 작성





Exercise II - IoT 모니터링/제어

앞선 두 예제의 통합

- 1) 새 프로젝트를 생성하여 모니터링과 제어가 동시에 가능한 위젯 구성

Example III - 조명 타이머



서버 프로젝트 생성 및 편집

- 새 프로젝트 추가 (이전 example 과정 반복)
- 프로젝트 편집에서 라벨과 토글 스위치 위젯 추가

The image illustrates the process of creating and editing a server project, specifically for a lighting timer. It shows the following components:

- Project Name:** A field labeled "Project Name : s".
- Widget Palette:** A list of available widgets including "라벨" (Label), "부시버튼" (Push Button), "토글 스위치" (Toggle Switch), "LED", "숫자/문자열 트랙바" (Number/String Slider), and "기타" (Other).
- Initial UI Layout:** A grid-based interface with a "조명시간설정" (Lighting Time Setting) label and a toggle switch set to "OFF". Below it are three circular buttons labeled "RESET", "0 시" (0 hours), and "0 분" (0 minutes).
- Edited UI Layout:** The same grid-based interface, but with the "조명시간설정" label and toggle switch moved to the top. The toggle switch is now highlighted with a red box, and a tooltip shows "Item index: 0 Wirte index: 1". The buttons remain in the same positions.
- Widget Properties Panel:** A panel on the right showing the properties of the selected widget, including "라벨" (Label), "부시버튼" (Push Button), "토글 스위치" (Toggle Switch), "LED", "숫자/문자열 트랙바" (Number/String Slider), and "기타" (Other).

Example III - 조명 타이머



서버 프로젝트 생성 및 편집

- 푸쉬 버튼 스위치 위젯 추가



- 장비 목록에서 프로젝트 선택(Example I 과정 반복)

Example III - 조명 타이머



```
#include <VitconBrokerComm.h>
using namespace vitcon;

#define LAMP 17

bool timeset = false;
bool Interval_Mup_status;
bool Interval_Hup_status;

int Hour = 0;
int Minute = 1;
uint32_t TimeSum = 0;
uint32_t TimeCompare;

uint32_t TimePushDelay = 0;
uint32_t TimerStartTime = 0;

//Interval 설정 모드로 들어가기 위한 함수
void timeset_out(bool val) {
    timeset = val;
}

//Interval 시간 단위를 조절하는 함수
void Interval_Hup(bool val) {
    Interval_Hup_status = val;
}

//Interval 분 단위를 조절하는 함수
void Interval_Mup(bool val) {
    Interval_Mup_status = val;
}
```

Example III - 조명 타이머



//Interval을 0시 0분으로 리셋하는 함수

```
void IntervalReset(bool val) {  
    if (!timeset && val) {  
        Hour = 0;  
        Minute = 0;  
    }  
}
```

/*widget toggle switch*/

```
IOTItemBin StopStatus;  
IOTItemBin Stop(timeset_out);
```

/*widget push button*/

```
IOTItemBin IntervalHUP(Interval_Hup);  
IOTItemBin IntervalMUP(Interval_Mup);  
IOTItemBin IntervalRST(IntervalReset);
```

/*widget Num/Str*/

```
IOTItemInt label_Hinterval;  
IOTItemInt label_Minterval;
```

/* A set of definition for IOT items */

```
#define ITEM_COUNT 7
```

```
IOTItem *items[ITEM_COUNT] = { &StopStatus, &Stop,  
                                &IntervalHUP, &IntervalMUP, &IntervalRST,  
                                &label_Hinterval, &label_Minterval  
};
```

Example III - 조명 타이머



```
/* IOT server communication manager */
const char device_id[] = "복사한 현재 장비의 ID "; // Change device_id to yours
BrokerComm comm(&Serial, device_id, items, ITEM_COUNT);

void setup() {
  Serial.begin(250000);
  comm.SetInterval(200);

  pinMode(LAMP, OUTPUT); // D3핀을 출력 모드로 지정
  digitalWrite(LAMP, LOW);
}

void loop() {
  IntervalSet(timeset);

  /* a LAMP auto control */
  if (timeset) {
    if (TimeCompare % 2 == 0) {
      digitalWrite(LAMP, LOW);
    }
    else if (TimeCompare % 2 == 1) {
      digitalWrite(LAMP, HIGH);
    }
  }
  else if (!timeset) {
    digitalWrite(LAMP, LOW);
  }

  StopStatus.Set(timeset);
  label_Hinterval.Set(Hour);
  label_Minterval.Set(Minute);
  comm.Run();
}
```



Example III - 조명 타이머

```
/* Interval time set */
void IntervalSet(bool timeset) {

    if (!timeset) { //시간설정 스위치가 OFF일 때
        TimeSum = (uint32_t)(Hour * 60 + Minute) * 60 * 1000; //ms단위로 변환
        TimerStartTime = millis();

        if (millis() > TimePushDelay + 500) { //위젯 버튼 누르는 시간 딜레이 주기
            Hour += Interval_Hup_status;
            if (Hour >= 24) Hour = 0;
            Minute += Interval_Mup_status;
            if (Minute >= 60) Minute = 0;

            TimePushDelay = millis();
        }
    }
    else if (timeset) { //시간 설정 스위치가 ON일 때
        TimeCompare = (millis() - TimerStartTime) / TimeSum ;
    }
    .....
}
```

코드 업로드 시 주의사항!

코드를 업로드 할 시, WiFi-LINK 스위치를 CFG로 놓아야함

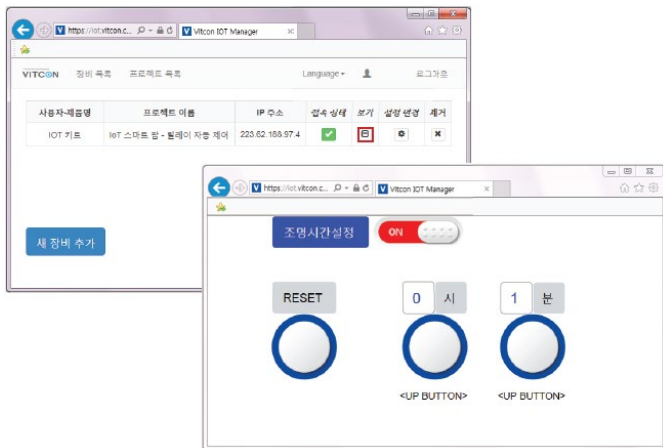
업로드 후 스위치를 다시 RUN으로 둔 후, RESET 버튼을 눌러 접속 상태 확인

Example III - 조명 타이머



타이머 기반 LED 조명 제어

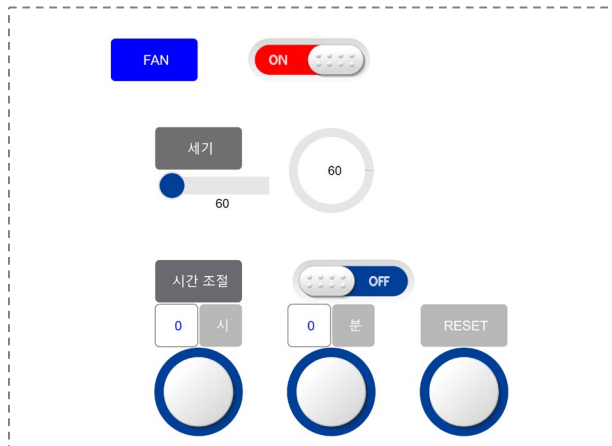
- Vitcon 웹 페이지에서 장비 접속 상태 확인 및 [보기]를 클릭하여 위젯 모니터링
- Play 스토어에서 VITCON을 검색하여 IoT Modlink 설치
(혹은, 스마트폰 웹 브라우저에서 iot.vitcon.co.kr에 접속)
- 로그인 하여 스마트폰으로 모니터링



Exercise III - IoT 모니터링/제어

FAN Remote Control

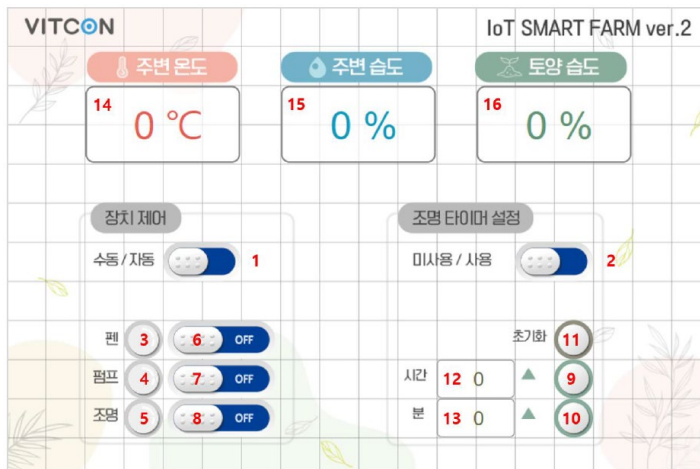
- 1) FAN ON/OFF 토글 스위치로 입력받는다.
- 2) 트랙바로 세기를 입력받고 프로그램 바로 입력된 세기를 출력한다.
- 3) 타이머 기반 LED 조명 제어와 마찬가지로 시간 조절 기능을 추가한다.
(LED와 달리 동작을 먼저 수행하고, 정해놓은 시간이 지나면 꺼지는 형태로 제작한다.)
- 4) 세기는 60 ~ 100으로 제한을 둔다.



Exercise IV - 통합 모니터링/제어 시스템

전체 실습 예제의 통합

1) 새 프로젝트를 생성한 후, 프로젝트 프리셋을 IoT 스마트팜 V2로 변경



번호	위젯 종류	Item Index Num.	Write Index Num.
1	토글 스위치	0	1
2	토글 스위치	2	3
3	LED	4	-
4	LED	6	-
5	LED	8	-
6	토글 스위치	4	5
7	토글 스위치	6	7
8	토글 스위치	8	9
9	푸쉬버튼	10	-
10	푸쉬버튼	11	-
11	푸쉬버튼	12	-
12	숫자/문자열	13	-
13	숫자/문자열	14	-
14	숫자/문자열	15	-
15	숫자/문자열	16	-
16	숫자/문자열	17	-

2) 위젯 아이디어에 따라 장치 연결 후 모니터링/제어 가능하게 연동

3) 수동/자동 제어 위젯에 따른 장치 컨트롤 구현

아두이노 AI 실습



데이터 준비

붓꽃 품종 분류 예시



Versicolor



Setosa



virginica

- Petal length, petal width, sepal length, sepal width 정보로 분류
- 붓꽃 데이터 취득 (https://en.Wikipedia.org/wiki/Iris_flower_data_set)

아두이노 AI 실습



데이터 확인

Fisher's Iris Data [\[hide\]](#)

Dataset Order	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	<i>I. setosa</i>
2	4.9	3.0	1.4	0.2	<i>I. setosa</i>
3	4.7	3.2	1.3	0.2	<i>I. setosa</i>
4	4.6	3.1	1.5	0.2	<i>I. setosa</i>
5	5.0	3.6	1.4	0.3	<i>I. setosa</i>
6	5.4	3.9	1.7	0.4	<i>I. setosa</i>
7	4.6	3.4	1.4	0.3	<i>I. setosa</i>
8	5.0	3.4	1.5	0.2	<i>I. setosa</i>
9	4.4	2.9	1.4	0.2	<i>I. setosa</i>
10	4.9	3.1	1.5	0.1	<i>I. setosa</i>

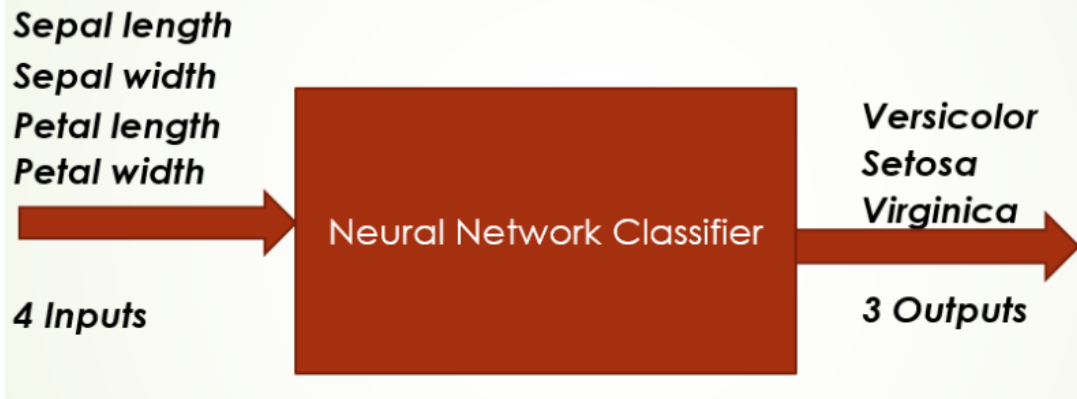
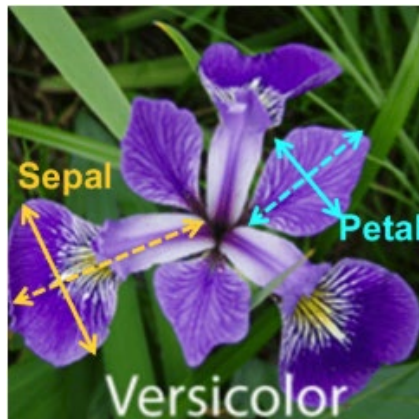
A	B	C	D	E	F	G
Sepal length	Sepal width	Petal Length	Petal Width	Setosa (output)	Versicolor (output)	Virginica (output)
6.1	2.8	4.7	1.2	0	1	0
4.8	3.1	1.6	0.2	1	0	0
5.5	3.5	1.3	0.2	1	0	0
6.4	2.7	5.3	1.9	0	0	1
6.2	2.8	4.8	1.8	0	0	1
4.4	3	1.3	0.2	1	0	0
6.5	3	5.8	2.2	0	0	1
6.1	2.9	4.7	1.4	0	1	0
4.4	3.2	1.3	0.2	1	0	0
6.6	3	4.4	1.4	0	1	0
6.9	3.1	5.1	2.3	0	0	1
5.7	4.4	1.5	0.4	1	0	0
5	3.5	1.6	0.6	1	0	0
5.4	3.4	1.7	0.2	1	0	0
4.9	2.5	4.5	1.7	0	0	1
6.3	3.4	5.6	2.4	0	0	1
5	3.4	1.6	0.4	1	0	0
5	3	1.6	0.2	1	0	0

- Output 형태를 조정하여 csv 파일로 저장
: [1, 0, 0] for Setosa, [0, 1, 0] for Versicolor, [0, 0, 1] for Virginica
- Training set (70%)와 test set(30%)로 적절히 분류

아두이노 AI 실습



ANNHub 를 사용한 Neural Network 분류기 설계



- 4가지 Input을 통해 3개의 타입 분류 Output을 출력하는 구조
- 데이터 셋이 적기 때문에 Bayesian Neural Network with Regularization 사용

아두이노 AI 실습



Step 1: IRIS training set 로드

Data Path
C:\Program Files\ANNHub\Examples\Classification Examples\IRIS\IRISTraining.csv

Training data

Sepal length	Sepal width	Petal Length	Petal Width	Setosa (output	Versicolor (out	Virginica (output
6.1	2.8	4.7	1.2	0	1	0
4.8	3.1	1.6	0.2	1	0	0
5.5	3.5	1.3	0.2	1	0	0
6.4	2.7	5.3	1.9	0	0	1
6.2	2.8	4.8	1.8	0	0	1
4.4	3	1.3	0.2	1	0	0
6.5	3	5.8	2.2	0	0	1
6.1	2.9	4.7	1.4	0	1	0
4.4	3.2	1.3	0.2	1	0	0
6.6	3	4.4	1.4	0	1	0
6.9	3.1	5.1	2.3	0	0	1
5.7	4.4	1.5	0.4	1	0	0
5	3.5	1.6	0.6	1	0	0
5.4	3.4	1.7	0.2	1	0	0
4.9	2.5	4.5	1.7	0	0	1
6.3	3.4	5.6	2.4	0	0	1
5	3.4	1.6	0.4	1	0	0
5	3	1.6	0.2	1	0	0
5	2.3	3.3	1	0	1	0
5.7	2.8	4.1	1.3	0	1	0
6	2.2	5	1.5	0	0	1
6.4	2.8	5.6	2.2	0	0	1

STEP 1: Load dataset

Configure Neural Network

아두이노 AI 실습



Step 2: Neural Network Configuration

Training Engine: Bayesian Regularization

Activation Function: Tansig

Activation Function: Logsig

Neural Network Type: Customised Type

Pre Processing: Min Max

Min: -1, Max: 1

Post Processing: Min Max

Min: 0, Max: 1

Training data ratio (%): 75

Hidden Nodes: 4

Cost Function: Mean Squared Error

Create Neural Network

Load data set

STEP 2: Configure Neural Network

Train Neural Network

- Dataset 포맷에 따라 ANNHUB가 NN 구조를 적절히 추천
 - : training algorithm
 - : hidden nodes의 수
 - : training data ratio
 - : pre-processing method
 - : post-processing method
 - : cost function
 - : activation functions
- 모델 구조에 능숙한 숙련자의 경우, 최적화를 위해 상세 설정 조정

아두이노 AI 실습



Step 3: Neural Network 학습



- Step 2의 학습 알고리즘에 따라 ANNHUB가 학습 파라미터와 stopping criteria를 적절히 추천
- 빠른 Stopping과 Bayesian Regularization은 overfitting을 적절히 막아주는 역할을 함

아두이노 AI 실습



Step 4: Neural Network 평가

Confusion Matrix

Confusion Matrix (Training)

[N=79]	Class 1	Class 2	Class 3	Accuracy	Sensitivity	Specificity
Class 1	30	0	0	100.00	100.00	100.00
Class 2	0	22	0	100.00	100.00	100.00
Class 3	0	0	27	100.00	100.00	100.00

Confusion Matrix (Validation)

[N=0]	Accuracy	Sensitivity	Specificity

Confusion Matrix (Testing)

[N=26]	Class 1	Class 2	Class 3	Accuracy	Sensitivity	Specificity
Class 1	5	0	0	100.00	100.00	100.00
Class 2	0	11	1	91.67	100.00	100.00
Class 3	0	0	9	100.00	90.00	94.12

Retrain Neural Network

STEP 4: Evaluate Neural Network

Test Neural Network

- 학습이 끝난 후, validation set과 test set에 대해 제대로 동작하는지 분석하기 위해 평가를 수행
- Bayesian Regularization을 사용했기 때문에, validation set은 필요하지 않음
- Training set의 30%를 사용
- Confusion matrix와 ROC curve 제공 : 분류기 성능과 안정성에 대한 직관성

아두이노 AI 실습



ANNHub 를 사용한 Performance Test

- 44개의 데이터 샘플(training set에 포함되지 않음) 을 사용해서 최종 테스트

Test Data Path
C:\Program Files\ANNHub\Examples\Classification Examples\IRIS\IRISTest.csv

Testing data

Output1	Output2	Output3	Predicted Output1	Predicted Output2	Predicted Output3	Mapped	Mapped	Correct
0.000000	0.000000	1.000000	0.000333	0.000397	0.999826	4	4	TRUE
0.000000	1.000000	0.000000	0.001101	0.987918	0.004761	2	2	TRUE
0.000000	0.000000	1.000000	0.000040	0.000506	0.999788	4	4	TRUE
0.000000	0.000000	1.000000	0.000046	0.000493	0.999699	4	4	TRUE
0.000000	1.000000	0.000000	0.000108	0.007081	0.992883	2	4	FALSE
0.000000	1.000000	0.000000	0.000150	0.008761	0.998273	2	4	FALSE
1.000000	0.000000	0.000000	0.998123	0.002467	0.000018	1	1	TRUE
1.000000	0.000000	0.000000	0.998079	0.002320	0.000019	1	1	TRUE
0.000000	0.000000	1.000000	0.000045	0.000607	0.999633	4	4	TRUE
0.000000	1.000000	0.000000	0.002448	0.999490	0.000047	2	2	TRUE
1.000000	0.000000	0.000000	0.998089	0.002188	0.000019	1	1	TRUE
1.000000	0.000000	0.000000	0.998022	0.002345	0.000019	1	1	TRUE
0.000000	1.000000	0.000000	0.000757	0.991847	0.003218	2	2	TRUE
1.000000	0.000000	0.000000	0.998106	0.002471	0.000018	1	1	TRUE
0.000000	1.000000	0.000000	0.007427	0.999059	0.000023	2	2	TRUE
0.000000	1.000000	0.000000	0.002231	0.999407	0.000052	2	2	TRUE
1.000000	0.000000	0.000000	0.998132	0.002773	0.000018	1	1	TRUE
1.000000	0.000000	0.000000	0.998118	0.002254	0.000018	1	1	TRUE

Accuracy (%)
93.3333

Confusion and ROC Export Result Manual Test

Retrain Neural Network STEP 5: Deploy Neural Network Export Neural Network

ROC

Confusion Matrix

[N=45]	Class 1	Class 2	Class 3	Accuracy	Sensitivity	Specificity
Class 1	15	0	0	100.00	100.00	100.00
Class 2	0	13	3	81.25	100.00	100.00
Class 3	0	0	14	100.00	82.35	90.32

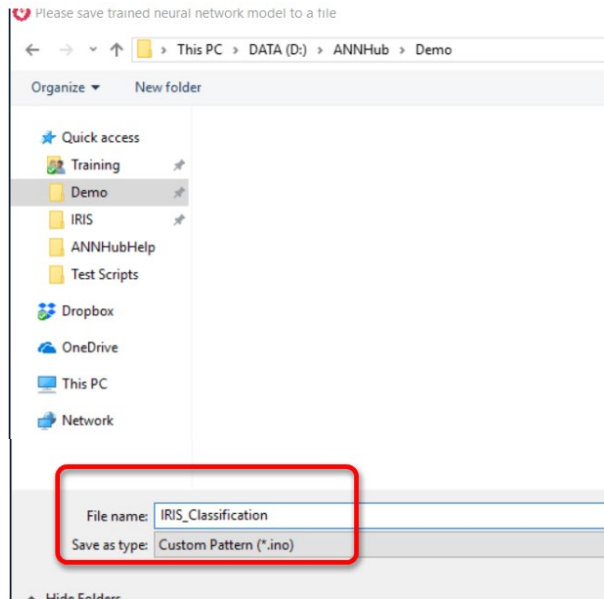
Retrain Neural Network STEP 5: Deploy Neural Network Export Neural Network

아두이노 AI 실습



학습된 NN 분류기를 아두이노에 deploy

- ANNHUB는 모델과 아두이노 코드를 바로 exporting하는 기능 제공
- 추출된 코드는 Arduino IDE에서 바로 실행 가능
- 코드를 수정하여 요구사항에 맞게 변경
- 컴파일과 업로드에 의존성(dependency) 없음
: 별도 라이브러리 없이 쉽게 업로드 가능



아두이노 AI 실습



아두이노 분류기 테스트

- 코드 업로드 후, PC에서 INPUT [Sepal length, Sepal width, Petal length, Petal width] 제공
- 아두이노가 입력에 따른 분류 OUTPUT 결과 출력

COM9

```
Welcome to ANNHUB API for Arduino by ANS Center (https://www.anscent
Please enter inputs with comma seperation.
For example, four inputs in IRIS flower classification will have the
-----
IRIS flower classification
Inputs= [4.60 ;3.20 ;1.40 ;0.20 ]
Outputs= [1.00 ;0.00 ;0.00 ]
Prediction: This is Setosa flower.
```

아두이노 AI 활용



아두이노-AI 의 단점

- 무거운 모델을 처리하기에 computing power가 부족
- 이미지, 동영상 등에 대한 처리 기능 결여

아두이노-AI 의 활용

- 텍스트 데이터 기반 분류기 탑재 후, 결과를 예측하는 데 사용
- 예시 1. 스마트 홈 전력 측정 및 경고 시스템
 - a) 아두이노에 전력 측정계를 설치
 - b) 전력 측정 패턴에 따른 이상 유무 감지 NN 모델 작성
 - c) 전력 이상 감지 NN 모델을 탑재한 아두이노가 WiFi를 통해 상태 안내
- 예시 2. 스마트 팩토리 설비 진단
 - a) 압력, 온도, 유량 등을 측정할 수 있는 아두이노 기반 End-device 제작
 - b) 압력, 온도, 유량 등에 따라 발생할 수 있는 고장 예지 NN 모델 작성
 - c) 설비 진단 모델을 탑재한 아두이노가 고장 유무 또는 예지 정보 안내