

수소연료전지를 활용한

IoT 통신기술교육

유상현

matthew@wiznet.io



유상현
Matthew Yu

 matthew@wiznet.io

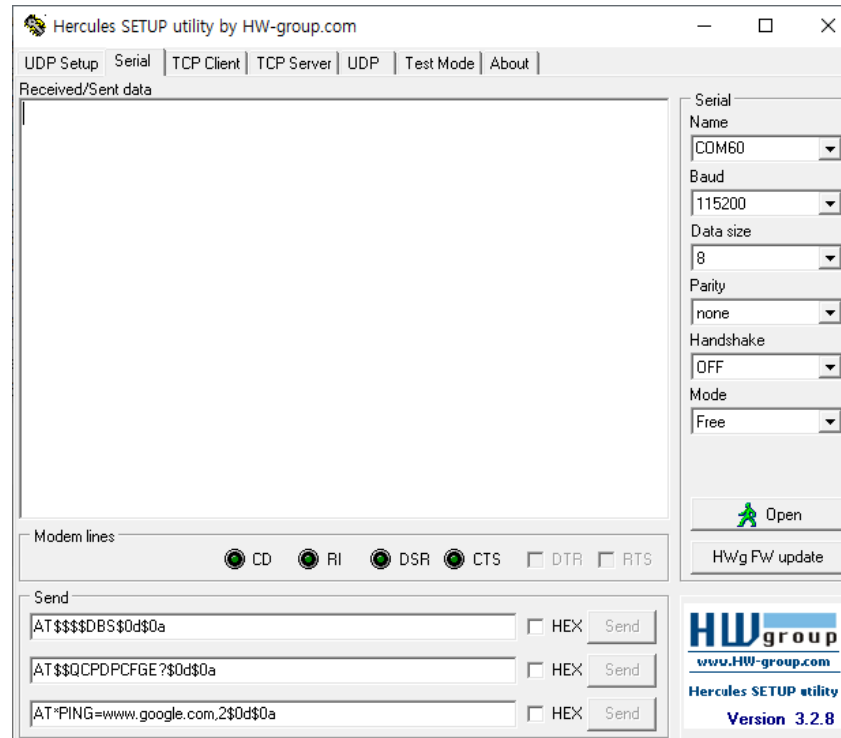
W5100S-EVB-Pico

IoT를 위한 통신 기초 개념



W5100S-EVB-Pico 통신 기초 개념

강의 시작 전 해야 할 일!

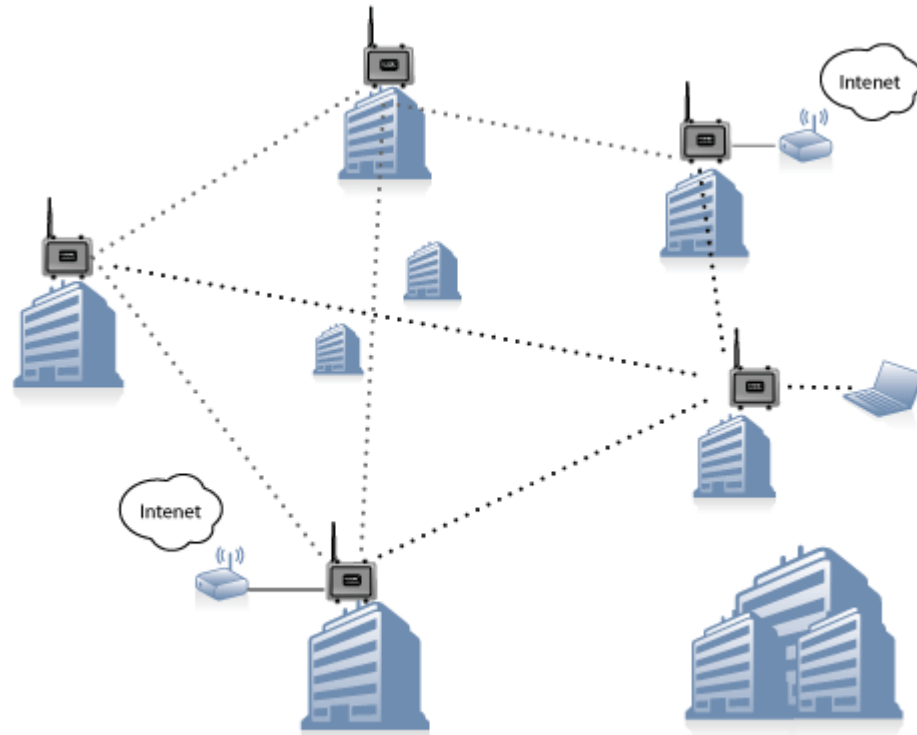


≫≫ Terminal 프로그램 다운로드

http://www.hw-group.com/products/hercules/index_en.html

W5100S-EVB-Pico 통신 기초 개념

컴퓨터 네트워크

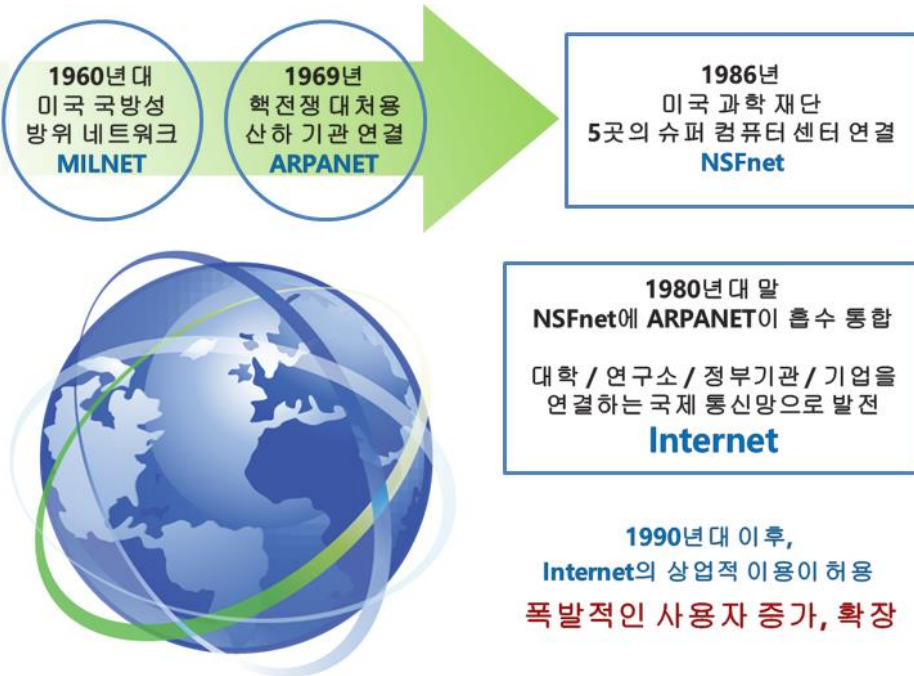


(Note: Mesh Network - all radios need to be on the same frequency)

- 정보 교환을 위해
- 서로 떨어져 있는 정보기기를
- 유/무선의 전송 매체를 통해 연결한
- 정보 통신망

W5100S-EVB-Pico 통신 기초 개념

컴퓨터 네트워크의 발전



Worldwide Web



IoT
Internet of Things

Cloud



A.I.
Deep Learning / Machine Learning ...

W5100S-EVB-Pico 통신 기초 개념

이론이 아닌 개념!

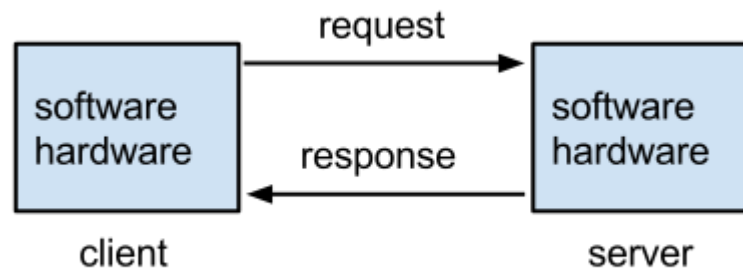
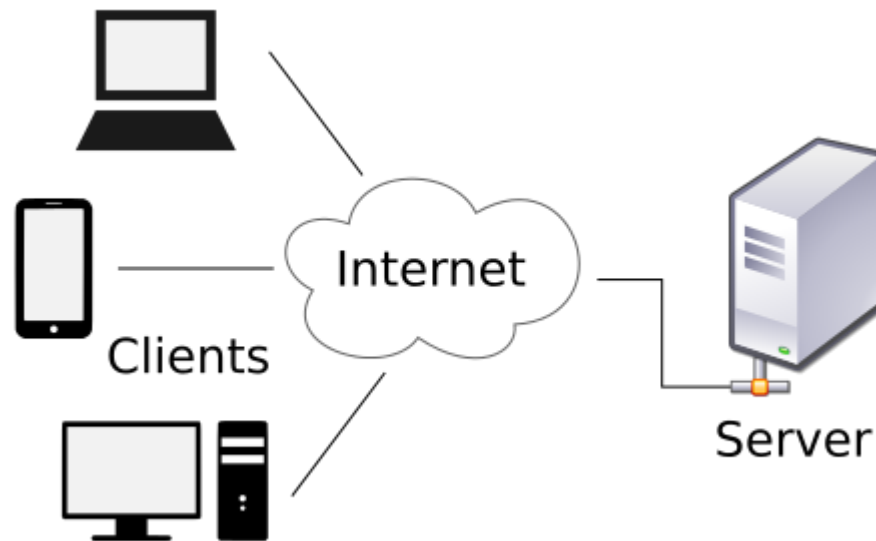
기초이론???

개념만 잡자!

W5100S-EVB-Pico 통신 기초 개념

서버와 클라이언트?

» 클라이언트는 고객, 서버는 상점주인 !?



출처 : [Wikimedia](#) (CC BY-SA)

W5100S-EVB-Pico 통신 기초 개념

IP주소는 어떻게 사용하나?

[제어판]

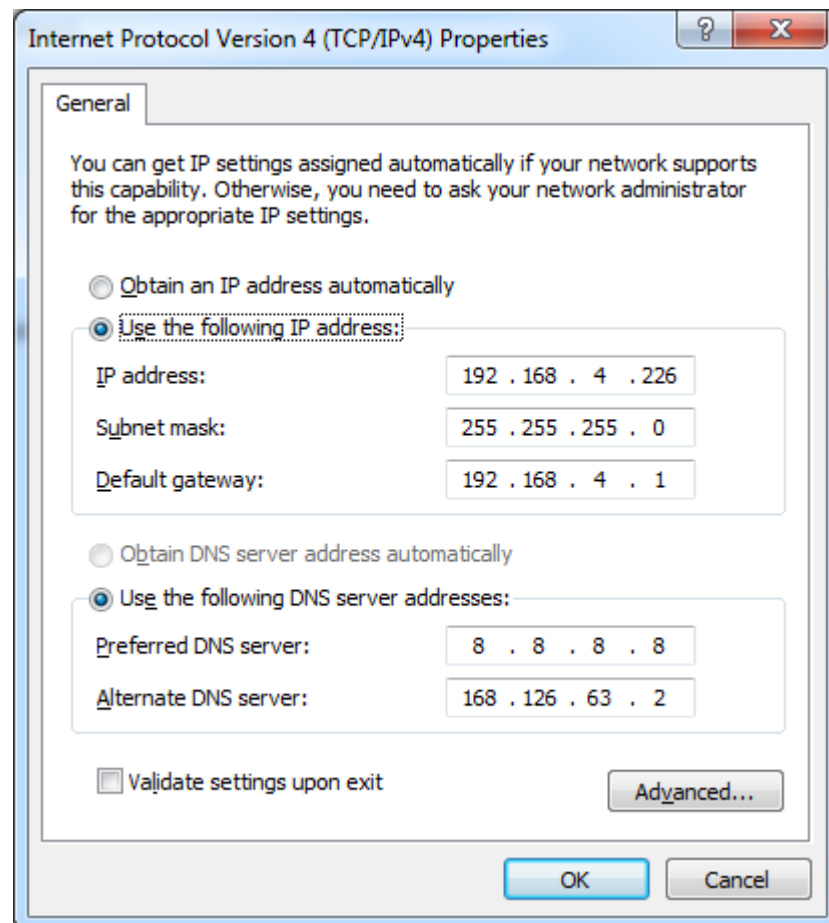
→ [네트워크 및 인터넷]

→ [네트워크 공유 센터]

→ [네트워크 연결]

→ [속성] 버튼

→ [IPv4]의 [속성]버튼



W5100S-EVB-Pico 통신 기초 개념

IP주소는 어떻게 사용하나?

➤➤ 누구인지 알기 위한 **IP주소**, **MAC주소**

IP주소 192.168.0.8 **전화번호**라고 생각해 보세요
 192.168.0.9 만약 이게 전화번호라면
 비슷한 지역에 할당된 번호겠죠?

✓ IPv4

- ✓ 총 32-bit 크기, 이론 상 42억개의 주소 할당 가능
- ✓ Classful / Classless addressing 방식 활용
- ✓ Unicast / Broadcast / Multicast 전송

✓ IPv6

- ✓ 총 128-bit 크기, 이론 상 무한대에 가까운 주소 할당 가능
- ✓ Unicast / Broadcast / Anycast 전송

MAC주소 00:08:DC:12:34:56

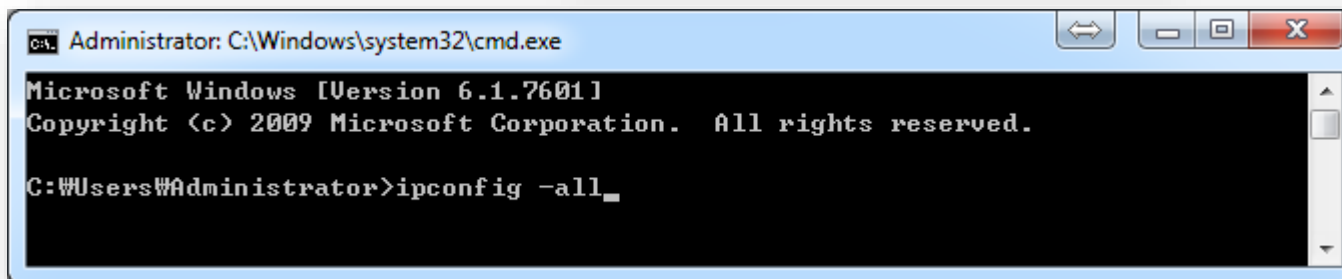
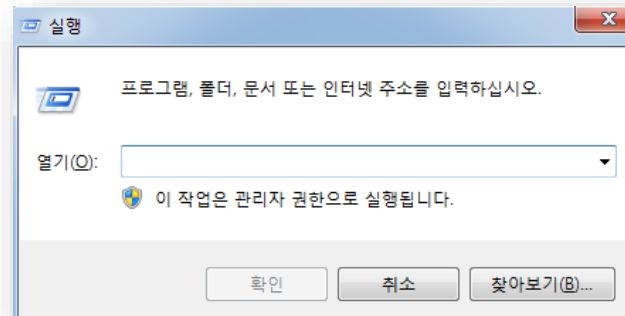
주민번호라고 생각해 보세요
만약 같은 주민번호가 존재한다면?!
누군가 내 주민번호를 도용한다면?!
[IP와 IP주소에 대해 더 알고 싶다면? 클릭!](#)

W5100S-EVB-Pico 통신 기초 개념

IP주소는 어떻게 사용하나?

➤➤ 내 PC의 NIC에 할당된 MAC 주소의 확인

1. 윈도우의 '실행' (RUN)
2. **cmd** 입력하여 Command line Interface 실행
3. **ipconfig -all**

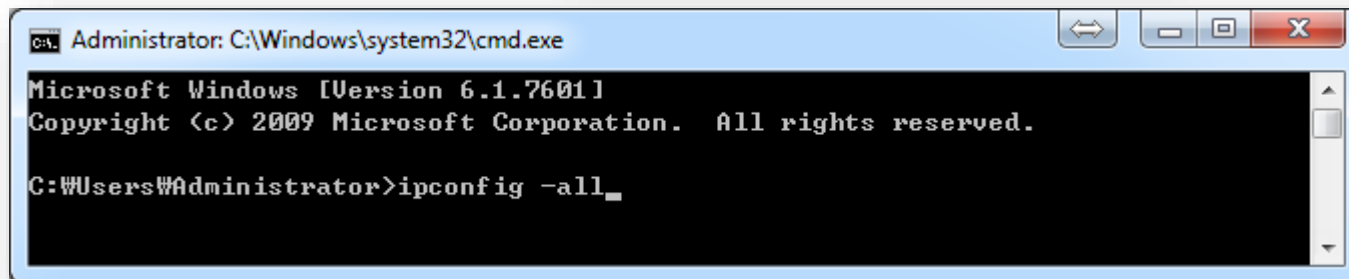
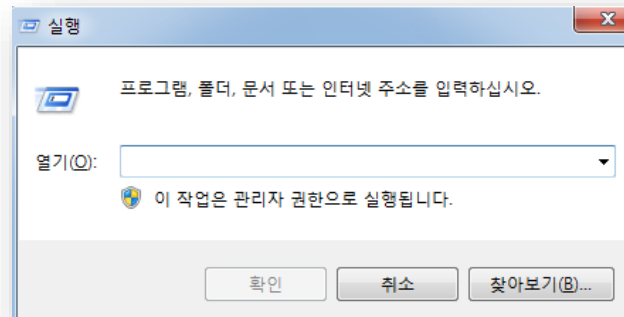


W5100S-EVB-Pico 통신 기초 개념

IP주소는 어떻게 사용하나?

➤➤ 내 PC에 기록된 네트워크 장치들의 MAC 주소 확인 (ARP)

1. 윈도우의 '실행' (RUN)
2. cmd 입력하여 Command line Interface 실행
3. arp -a



IP주소는 어떻게 사용하나?

≫≫ IP 주소 할당 방식:

- 공인 IP와 사설 IP

- 공인(Public) IP

- ✓ ICANN(국제 인터넷 주소 관리기구)를 통해 부여받은 IPv4 주소이며, 인터넷에 접속하기 위한 고유 IP 주소
 - ✓ 한국은 **KISA**(인터넷진흥원)가 ICANN으로부터 할당받은 주소를 관리
 - ✓ Class 대역에 따라 분류

- 사설(Private) IP

- ✓ 특수 목적용 IP 대역을 개인 및 회사에서 임의 할당하여 활용
 - A 클래스 : 10.0.0.0 ~ 10.255.255.255
 - B 클래스 : 172.16.0.0 ~ 172.31.255.255
 - C 클래스 : 192.168.0.0 ~ 192.168.255.255
 - ✓ 외부 망에서 사설 망 내부로 접근 할 수 없음
 - ✓ 대부분의 장비에 사설 IP를 부여하고, 공유기(라우터)의 공인 IP를 통해 외부 망에 접속하는 형태가 일반적

W5100S-EVB-Pico 통신 기초 개념

IP주소는 어떻게 사용하나?

≫≫ 우리에게엔 이름이 있다 – Domain Name

www.wiznet.co.kr

이름이라고 생각해보세요
주민등록번호랑 전화번호로
사람을 부른다고 생각해보세요...

www.wiznet.net

동명이인?!

W5100S-EVB-Pico 통신 기초 개념

IP주소는 어떻게 사용하나?

»» 우리 사람인가..? – Subnet Mask

031 – 8023 – 5699

서브넷 마스크의 기본은 A, B, C, D, E Class만 구분
이는 한정된 IP를 정해진 규칙에 따라 분배하기 위함!

여기서 한 단계 더!

우리들만의 공간을 만든다! 우리들만의 약속!

하지만,
누군가 약속을 알고
우리들만의 번호로
접근한다면?

- 나의 IP Address : 192.168.0.154
- Subnet Mask : 255.255.255.128
- 나에게 접근하려는 외부 IP Address : 192.168.0.96

- 접근 IP Address : 1100 0000 . 1010 1000 . 0000 0000 . 0110 0000
- Subnet Mask : 1111 1111 . 1111 1111 . 1111 1111 . 1000 0000
1에 해당하는 값들은 서로 match가 되어야 한다. 0은 Don't care~
- My IP Address : 1100 0000 . 1010 1000 . 0000 0000 . 1001 1010

- Result : subnet의 bit가 1인데 붉은 박스 내의 값이 match가 되지 않았다
결과적으로 나의 IP로의 접근을 허용하지 않는다.

[Subnet Mask가 궁금하다면? 클릭!](#)

W5100S-EVB-Pico 통신 기초 개념

IP주소는 어떻게 사용하나?

≫ 밖으로의 **관문**/입구 – Gateway

우리끼리만으론 재미가 없어…
외부와 접촉하려면 어떻게 하지?
밖으로 나가자!

그 문인 Gateway!
근데… 문(Gateway)이 없다면…?

≫ 전화번호도 주민번호도 모르고 이름만 알아요! – DNS

제가 만나야 할 사람이 있는데, 이름만 알아요…

DNS Server (Domain Name Service Server) ???

전세계적 전화번호부?!

[DNS 관련 동영상](#)

네트워크.. 이러쿵 저러쿵

≫≫ 포트 번호와 소켓

- 포트 번호(Port Number)

- ✓ 전송 계층 프로토콜에서 네트워크 서비스나 특정 프로세스를 식별하는 논리 단위, 통신의 종단
- ✓ 16-bit의 숫자로 표현 (0 ~ 65535)

- 포트 번호의 구분

- ✓ 0번 ~ 1023번: 잘 알려진 포트 (Well-known port)
- ✓ 1024번 ~ 49151번: 등록된 포트 (Registered port)
- ✓ 49152번 ~ 65535번: 동적 포트 (Dynamic port)

- 잘 알려진 포트 번호의 대표적 예

- | | |
|---------------------|----------------------------|
| ✓ 20 : FTP(data) | ✓ 23 : Telnet |
| ✓ 21 : FTP(control) | ✓ 53 : DNS |
| ✓ 22 : SSH | ✓ 80 : HTTP (Web Services) |

W5100S-EVB-Pico 통신 기초 개념

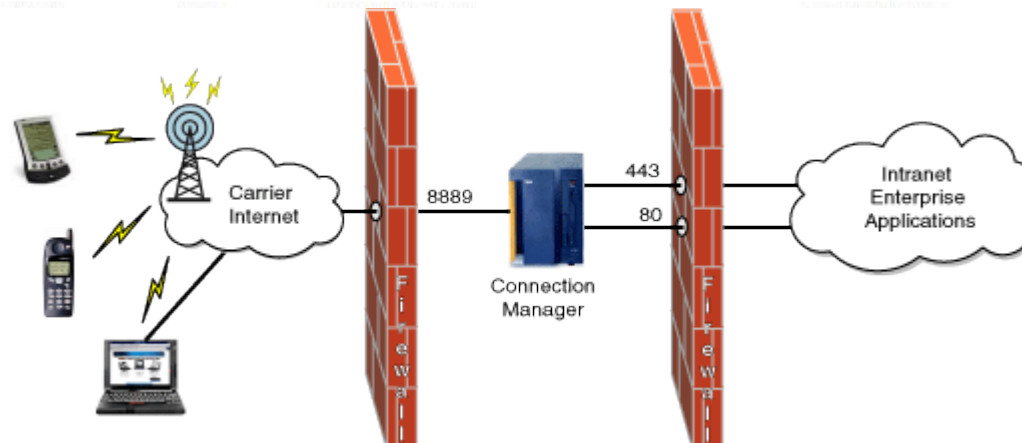
네트워크.. 이러쿵 저러쿵

»» 포트 번호와 소켓

- 참고: 방화벽(Firewall)의 기초 원리

- ✓ 허용된 포트 번호 이외의 In-bound / Out-bound 포트 번호를 차단

Port Number	Type	Protocol	RFC	Used by
80	TCP	HTTP	2616	AirPlay
443	TCP	HTTPS	-	AirPlay
554	UDP / TCP	RTSP	2326	AirPlay
3689	TCP	DAAP	-	iTunes Music Sharing / AirPlay
5297	TCP	-	-	Bonjour
5289	TCP / UDP	-	-	Bonjour
5353	UDP	MDNS	-	Bonjour / AirPlay



W5100S-EVB-Pico 통신 기초 개념

네트워크.. 이러쿵 저러쿵

>> 소켓 (Socket)

연결 접점...?
전구소켓?!



출처 : [Wikimedia](#) (CC BY-SA)

>> 포트 (Port)

항구...?
소켓에 할당한(된) 번호!

한 곳에 한대만...

1번 포트로 가면
A배를 탈 수 있습니다!

B행선지로 가실 분들은
2번 포트의 배에
탑승하세요.



출처 : [Wikimedia](#) (CC BY)

W5100S-EVB-Pico 통신 기초 개념

네트워크.. 이러쿵 저러쿵

>> DHCP

전화번호가 없어요...

잠시 임대 좀...

요청하는 곳이 DHCP Server

[DHCP 관련 동영상](#)

ID	클라이언트 이름	MAC 주소	할당된 IP	임대 시간
1	android-e[REDACTED]	[REDACTED]	192.168.4.2	01:46:05
2	android-7[REDACTED]	[REDACTED]	192.168.4.4	01:36:05

>> Static(정적) Network

난 내 번호가 있어!

좋아요. 하지만, 정보를 정확하게 등록 해야겠죠?

누군가 쓰는 번호(IP 주소)를 입력한다면? 총.돌.

우리들만의 공간에 들어가려는데 약속(Subnet)이 달라? 통신 안 됨

아... 내 문(Gateway)이 어디 있지? 밖으로 못나감

W5100S-EVB-Pico 통신 기초 개념

네트워크.. 이러쿵 저러쿵

>> TCP

당신의 데이터가 제대로 전달(**T**ransmission) 되었는지
확인하고 제어(**C**ontrol)할게요
대신 상대방이랑 연결부터 해야 되요 (Connection-Orient)
그냥... 하나의 **약속**(**P**rotocol)일 뿐이에요

신뢰성 중시

속도는 상대적(UDP에 비해)으로 느림

>> UDP

당신(**U**ser)의 데이터가 제대로 전달되었는지 책임지지 않아요
데이터(**D**atagram)의 흐름에 맡겨보죠
대신 미리 연결할 필요 없어요(Connectionless)
저도 하나의 **약속**(**P**rotocol)일 뿐

속도 중시

신뢰성은 상대적(TCP에 비해)으로 떨어짐

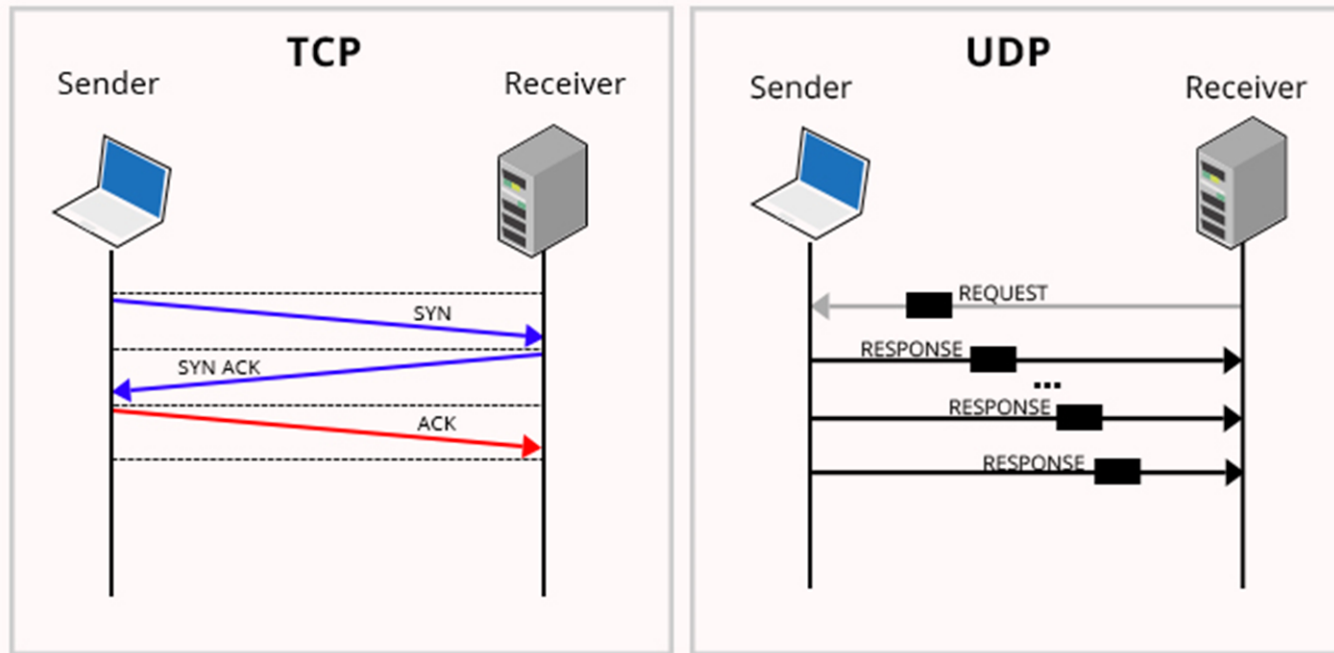
[TCP/UDP 영상](#)

[TCP vs UDP](#)

W5100S-EVB-Pico 통신 기초 개념

네트워크.. 이러쿵 저러쿵

≫ TCP vs UDP Communication



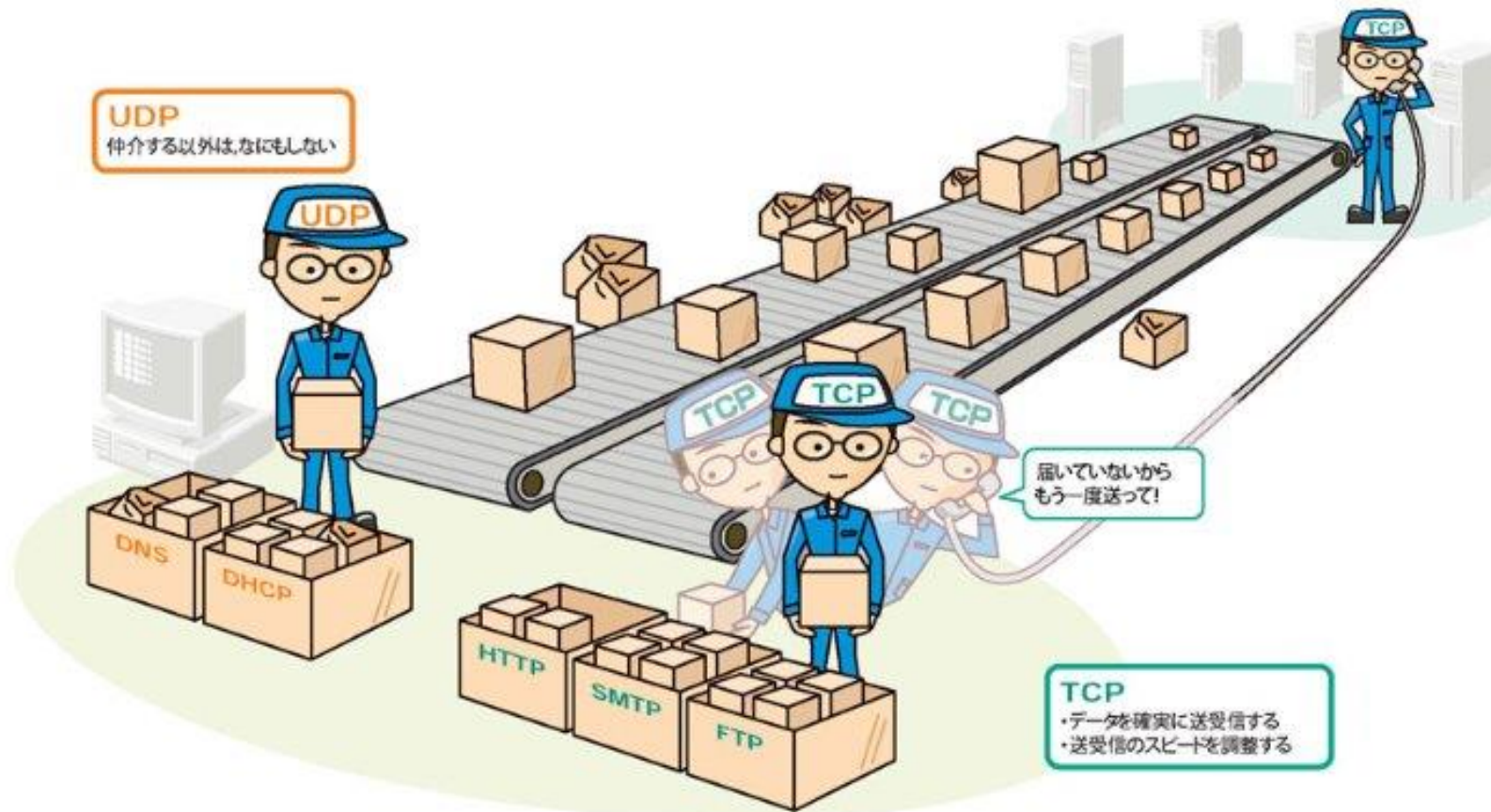
[TCP/UDP 영상](#)

[TCP vs UDP](#)

W5100S-EVB-Pico 통신 기초 개념

네트워크.. 이러쿵 저러쿵

>> TCP? UDP?



W5100S-EVB-Pico 통신 기초 개념

네트워크.. 이러쿵 저러쿵

>> TCP? UDP?

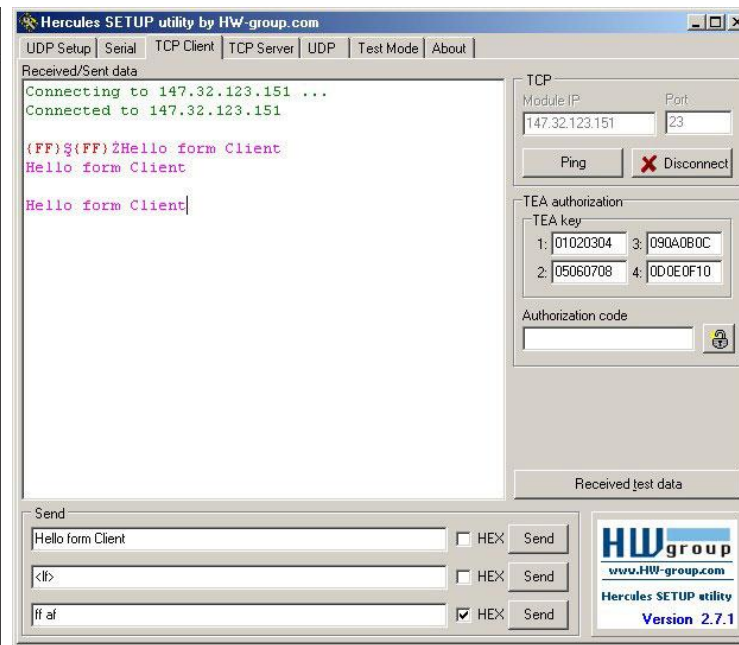
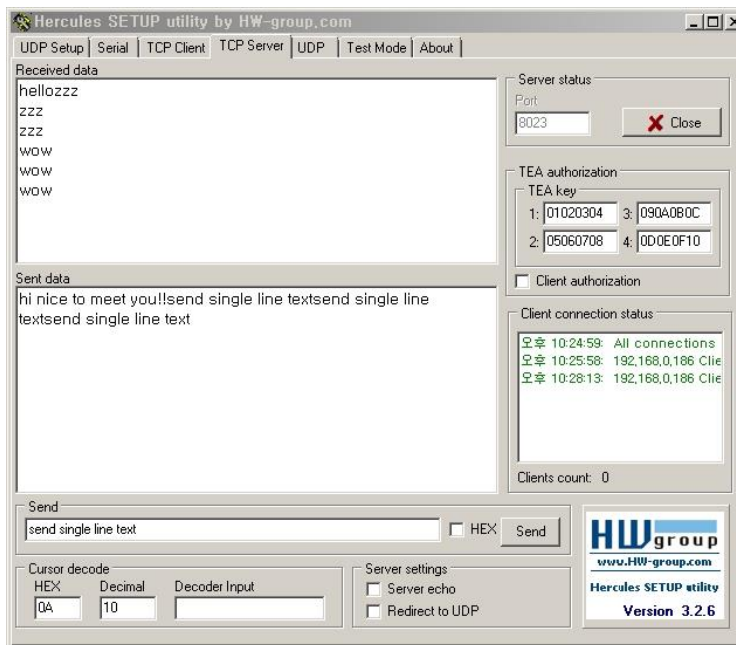
구 분	TCP	UDP
신뢰성	<ul style="list-style-type: none">Reliable (신뢰성 있는)신뢰성을 위해 Ack, Checksum 등 사용	<ul style="list-style-type: none">Unreliable (신뢰성이 없다)
연결성	<ul style="list-style-type: none">Connection-oriented (연결 지향성)Connection을 맺고 통신	<ul style="list-style-type: none">Connection less (비 연결성)
재전송	<ul style="list-style-type: none">재전송 요청함 (오류 및 패킷 손실 검출시)	<ul style="list-style-type: none">재 전송 없음
특 징	<ul style="list-style-type: none">Flow control을 위해 windowing 사용속도는 다소 느려도 신뢰성을 제공	<ul style="list-style-type: none">신뢰성은 보장하지 않지만, 고속데이터 전송실시간 전송에 적합
용 도	<ul style="list-style-type: none">신뢰성이 필요한 통신	<ul style="list-style-type: none">총 패킷수가 적은 통신동영상 및 음성 등 멀티미디어 통신

W5100S-EVB-Pico 통신 기초 개념

네트워크.. 이러쿵 저러쿵

>> TCP / UDP 데이터 교환

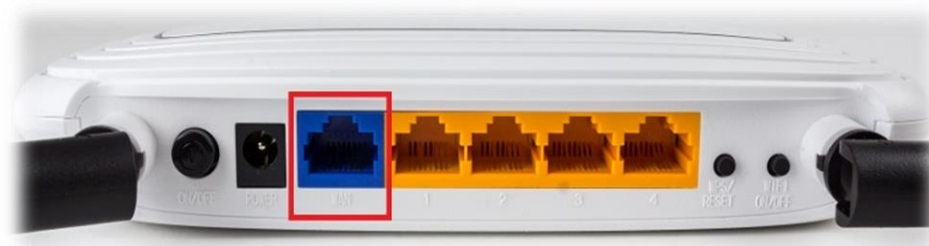
1. TCP/UDP Server와 Client 프로그램 실행
2. [Server] 포트 지정 후 **Listen** / [Client] 서버의 IP와 Port로 **Connect**
3. 데이터 **Send** / **Receive** 확인



W5100S-EVB-Pico 통신 기초 개념

네트워크.. 이러쿵 저러쿵

➤➤ 라우터 기능 - 네트워크 간 연결 도우미



출처 : [Wikimedia](#) (CC BY-SA)

➤➤ DHCP Server 기능

외부에서 들어온 회선에 할당된 IP는 하나!
온갖 기기로 같은 네트워크를 구축하려면,
수 많은 IP가 필요!

이때, DHCP로 나만의 사설 네트워크 형성!

※ 공인IP 주소 : 공식 기관([IANA](#))으로부터 할당된 유일한 IP주소

※ 사설IP 주소 : 사설로 구축한 네트워크에서 사용하는 IP주소

W5100S-EVB-Pico

IoT를 위한 Ethernet 기초 실습(1)

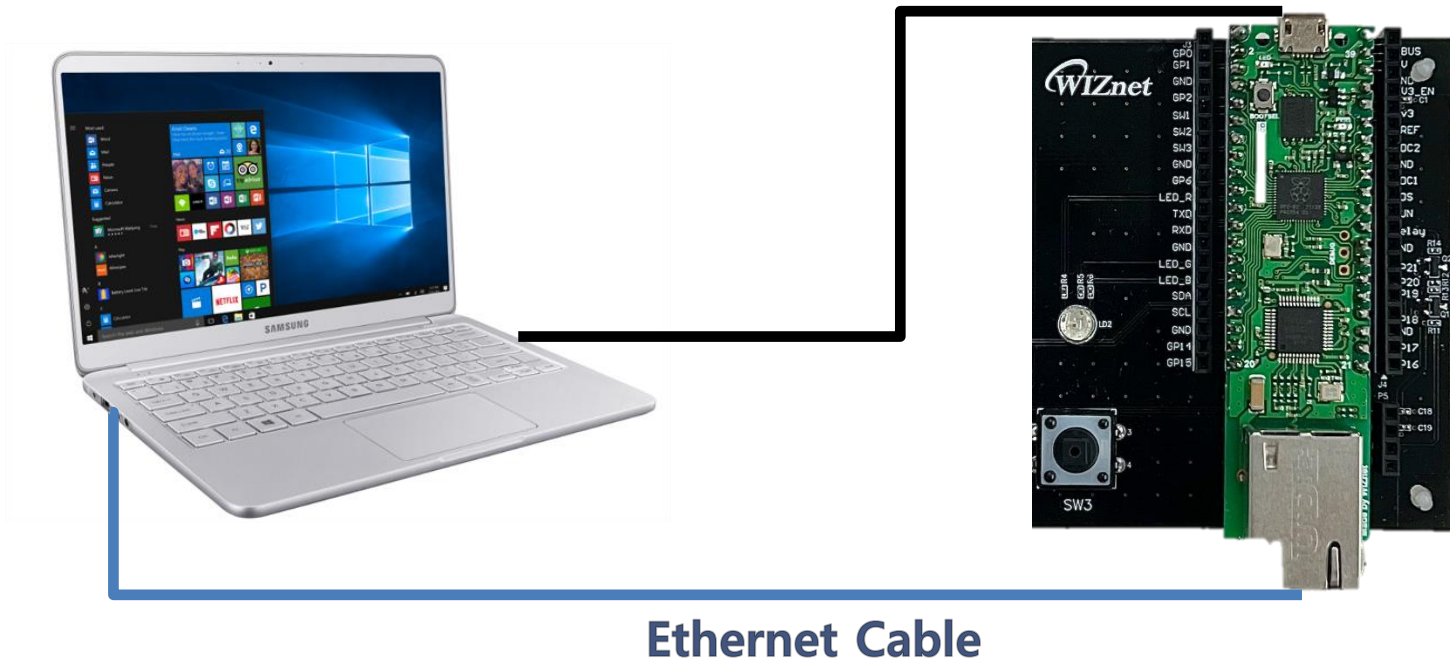


Ethernet 기초 실습 (1)

1. W5100S-EVB-Pico 연결

W5100S-EVB-Pico와 LAN케이블, USB 케이블 연결

USB Cable



Ethernet 기초 실습 (1)

2. WIZnet Arduino Ethernet Library 추가

파일 - 환경설정 - 추가적인 보드 매니저 **URLs** 수정하기

https://github.com/WIZnet-ArduinoEthernet/arduino-pico/releases/download/global/package_rp2040-ethernet_index.json

Ethernet 기초 실습 (1)

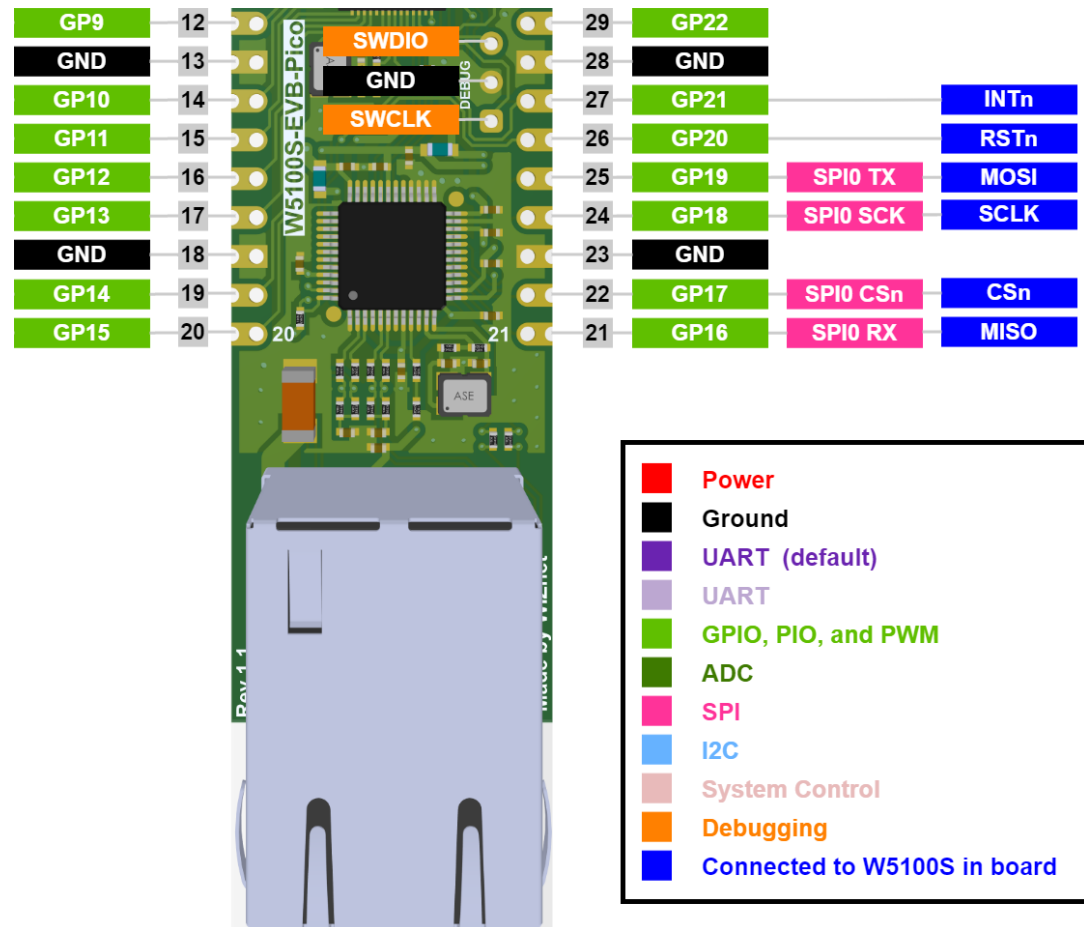
3. Ethernet Example (ChatServer)

파일 - 예제 - Ethernet - ChatServer 클릭

Ethernet.init(CS pin)

- SPI 통신의 CS Pin 으로 Init 한다.

W5100S-EVB-Pico 의 Ethernet Pin 확인



4. TCP Loopback 서버

통신 테스트에서 많이 사용되는 Loopback 서버를 만들어 보자

“ 루프백(Loopback, loop-back)이란
전기신호의 라우팅, 디지털 데이터 스트림(digital data streams), 또는 품목의 흐름이 **의도적인 가공이나 수정 없이** 원래의 장치나 장비로 **돌아가는** 것을 말한다.

이것은 주로 전송이나 수송 기반 시설을 테스트하는 수단으로 사용된다.

”

Ethernet 기초 실습 (1)

4. TCP Loopback 서버

```
#include <SPI.h>
#include <Ethernet.h>

#define SOCK_PORT      5000          //PORT

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // MAC ADDRESS.

IPAddress local_ip(192,168, 11, 177); // IP Address
IPAddress gateway(192,168, 11, 1);    // Gateway Address
IPAddress subnet(255, 255, 255, 0);   // Subnet Mask
IPAddress dns_addr(8, 8, 8, 8);       // DNS Address

EthernetServer server(SOCK_PORT);     // 소켓 생성할 때 80(Http)번 포트로 생성

void setup()
{
    Ethernet.init(17);                // CS Pin
    Serial.begin(9600);               // Serial통신 baudrate 9600로 시작

    while (!Serial);                 //Serial 모니터를 열때까지 대기

    // start the Ethernet connection:
    Serial.println("Trying to get an IP address using DHCP");
```

네트워크
환경에
맞게 설정

```
if (Ethernet.begin(mac) == 0)        // DHCP서버에 IP할당 요청
{
    // IP할당에 실패하면 Static으로 설정
    Serial.println("Failed to configure Ethernet using DHCP");
    Serial.println("Configure Static Network Information");
    // initialize the ethernet device not using DHCP:
    Ethernet.begin(mac, local_ip, dns_addr, gateway, subnet);
}
Serial.println("***** W5100S-EVB-Pico Loopback Server (BASIC) *****");
Serial.print("Device IP address : ");
Serial.print(Ethernet.localIP());    // IP주소 출력
Serial.print(" : ");
Serial.println(SOCK_PORT);           // Port 번호 출력

server.begin();                      // 서버 Listen 시작 (Socket open -> Listen 상태 돌입)
}

void loop()
{
    EthernetClient client = server.available(); // 클라이언트가 서버에 연결되었다면

    if(client.connected())
    {
        while(client.available())
        {
            char thisChar = client.read();
            server.write(thisChar);
            Serial.print(thisChar);
        }
    }
}
```


Ethernet 기초 실습 (1)

4. TCP Loopback 서버 (코드 수정)

내 네트워크 정보에 맞게 수정 (Ethernet IP 대역과 Gateway가 같아야 함)

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };    // MAC ADDRESS.

IPAddress local_ip(192,168, 11, 177);    // IP Address
IPAddress gateway(192,168, 11, 1);    // Gateway Address
IPAddress subnet(255, 255, 255, 0);    // Subnet Mask
IPAddress dns_addr(8, 8, 8, 8);    // DNS Address

EthernetServer server(SOCK_PORT);    // 소켓 생성할 때 80 (Http) 번 포트로 생성

void setup()
{
    Ethernet.init(17);    // CS Pin
    Serial.begin(9600);    // Serial통신 baudrate 9600로 시작

    while (!Serial);    //Serial 모니터를 열때까지 대기

    // start the Ethernet connection:
    Serial.println("Trying to get an IP address using DHCP");
```

Ethernet 기초 실습 (1)

5. 코드 업로드 및 Ethernet 네트워크 활성화

Ethernet 네트워크 활성화 (무선 네트워크 비활성화) 및 유선 네트워크 설정

[제어판]

→ [네트워크 및 인터넷]

→ [네트워크 공유 센터]

→ [네트워크 연결]

→ [속성] 버튼

→ [IPv4]의 [속성]버튼

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };  
  
IPAddress local_ip(192,168, 11, 177);    // IP Address  
IPAddress gateway(192,168, 11, 1);      // Gateway Address  
IPAddress subnet(255, 255, 255, 0);     // Subnet Mask  
IPAddress dns_addr(8, 8, 8, 8);         // DNS Address
```

**IP 대역대와 Gateway를 W5100S-EVB-Pico와 같게 설정
단, IP 주소가 모두 같으면 안된다.**

Ethernet 기초 실습 (1)

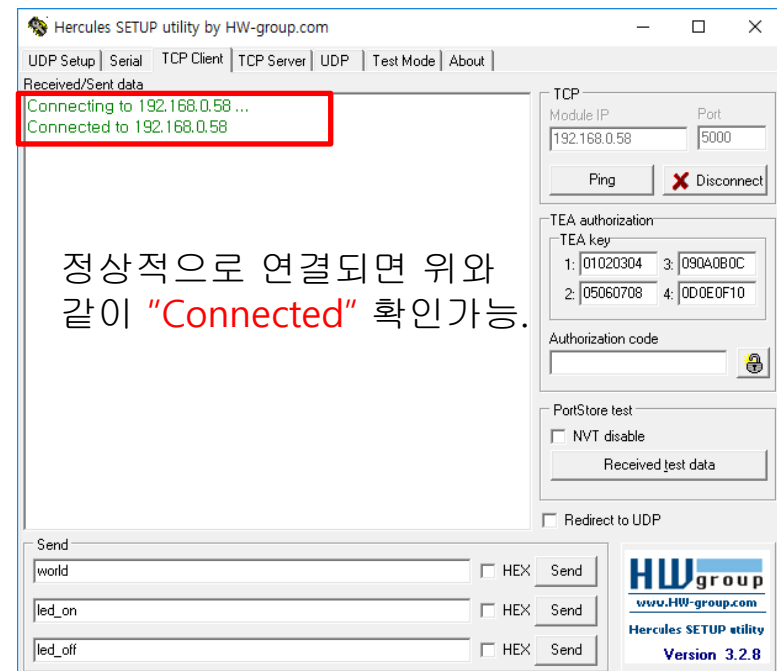
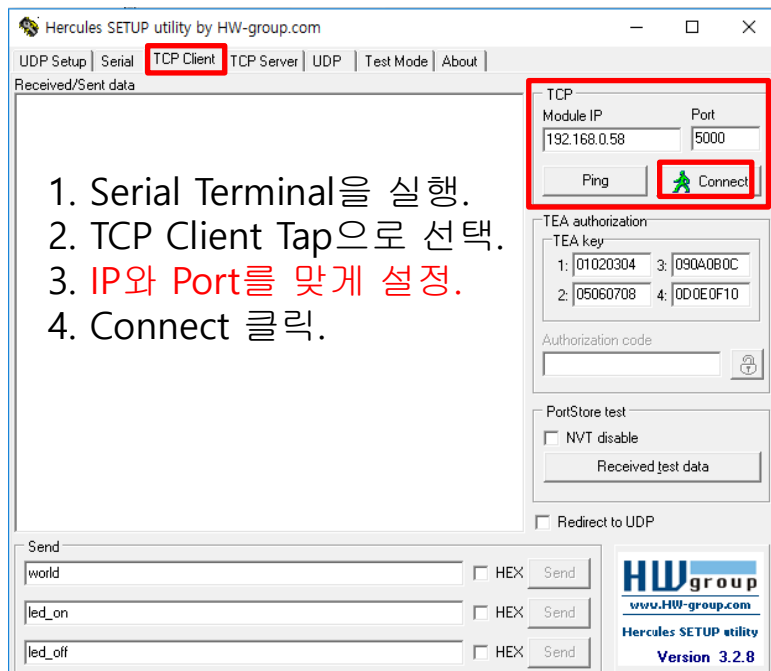
6. 업로드 및 실행 1

Client Program 실행 – ChatServer에 접속 – 데이터 전송 확인

Serial Terminal 실행

앞서 확인했던 **IP주소**와 **Port번호**를 입력합니다.

Example IP: 192.168.11.177, Port: 5000



Ethernet 기초 실습 (1)

7. 업로드 및 실행 2

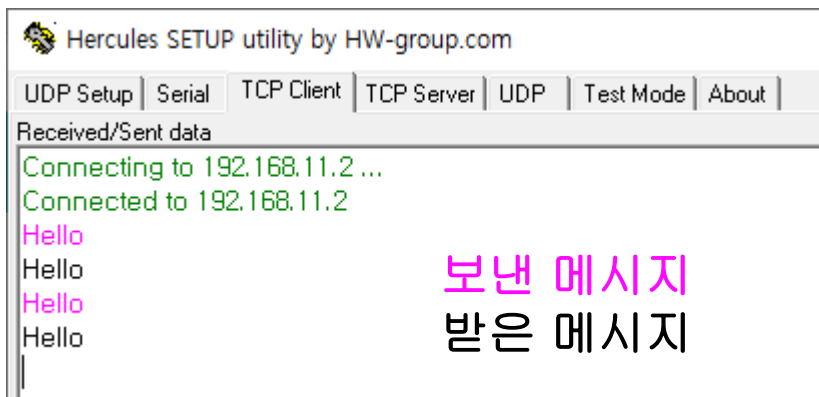
Client Program 실행 – ChatServer에 접속 – 데이터 전송 확인

➤➤ Serial Terminal 실행

앞서 확인했던 **IP주소**와 **Port번호**를 입력합니다.

Example IP: 192.168.11.177, Port: 5000

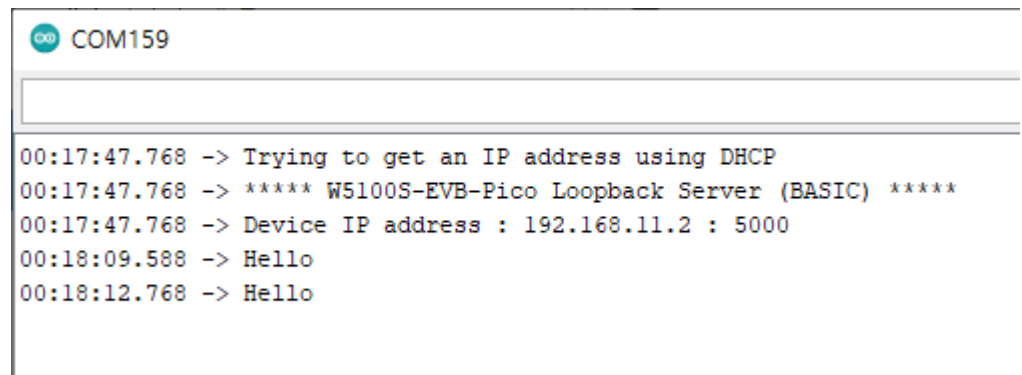
Client



```
Hercules SETUP utility by HW-group.com
UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About |
Received/Sent data
Connecting to 192.168.11.2 ...
Connected to 192.168.11.2
Hello
Hello
Hello
Hello
```

보낸 메시지
받은 메시지

Server



```
COM159
00:17:47.768 -> Trying to get an IP address using DHCP
00:17:47.768 -> ***** W5100S-EVB-Pico Loopback Server (BASIC) *****
00:17:47.768 -> Device IP address : 192.168.11.2 : 5000
00:18:09.588 -> Hello
00:18:12.768 -> Hello
```

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

≫ 시작이 중요하죠? 이 함수 안 쓰면 시작도 못합니다

begin() : 네트워크 정보 **설정**

```
38  if (Ethernet.begin(mac) == 0)    // DHCP서버에 IP할당 요청
39  {
43  // initialize the ethernet device not using DHCP:
44  Ethernet.begin(mac, local_ip, dns_addr, gateway, subnet);
```

DHCP/Static에 따라

여러 인자를 가진 함수로 설정 가능

이더넷을 사용하려면 필수!

함수는 총 5개가 있음(골라서 사용!)

localIP() : 장치 IP주소 **가져오기**

```
48  SerialUSB.print("Device IP address : ");
49  SerialUSB.println(Ethernet.localIP());
50  SerialUSB.print(" : ");
```

begin() 함수로 장치(W5100S-EVB-Pico)에
할당된 IP주소를 가져옵니다
이걸로 DHCP의 경우 어떤 IP를 할당
받았는지 알 수 있겠죠?

Ethernet class

The Ethernet class initializes the ethernet library and network settings.

- begin()
- localIP()
- maintain()

IPAddress class

The IPAddress class works with local and remote IP addressing.

- IPAddress()

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

» 네트워크 설정을 조금 더 해보고 싶다면...

maintain() : DHCP 서버에 갱신 요청

아... 전화번호를 더 써야 되는데...
더 쓰게 해주세요! 라고 **DHCP** 서버에 요청

그럼 시간이 연장됩니다
(일반적으로 임대시간의 반절이 지나면
스스로 요청하게 되어 있습니다)

IPAddress() : IP주소 설정

```
19 IPAddress local_ip(192,168, 0, 177);    // IP Address
20 IPAddress gateway(192,168, 0, 1);        // Gateway Address
21 IPAddress subnet(255, 255, 255, 0);      // Subnet Mask
22 IPAddress dns_addr(8, 8, 8, 8);          // DNS Address
```

IP주소를 설정하는데 사용합니다
'.' 대신에 **','(콤마)**로 구분하는 것이 특징

192.168.0.2 → (192, 168, 0, 2)

Ethernet class

The Ethernet class initializes the ethernet library and network settings.

- begin()
- localIP()
- maintain()

IPAddress class

The IPAddress class works with local and remote IP addressing.

- IPAddress()

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

≫ 용도에 따라 기억하자

Server : Class 자체, 직접 호출 x

EthernetServer() : 서버 **생성**

```
26  
27 EthernetServer server(SOCK_PORT);  
28
```

함수 인자로 괄호() 안에 **Port**번호 입력
0~65535 까지 가능

begin() : Server 작동 **시작**

```
52  
53 server.begin();  
54 }
```

생성한 서버 이름으로 사용
(여기에서는 **server**)
Server에게 **Client**로부터의 접속을
기다리라는 명령입니다

Server class

The Server class creates servers which can send data to and receive data from connected clients (programs running on other computers or devices).

- Server
- EthernetServer()
- begin()
- available()
- write()
- print()
- println()

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

≫≫ 용도에 따라 기억하자

available(): Client 연결 확인

```
57 void loop()
58 {
59     EthernetClient client = server.available();
```

Client가 연결되었고, 일정 데이터가 들어오면 “사용 가능한(available)” 상태로 판단합니다
즉, **Client와의 연결이 되었다고 판단**

write(): Client로 데이터 전송

```
63 char thisChar = client.read();
64 server.write(thisChar); // 수신
65 SerialUSB.print(thisChar); //
```

서버와 연결된 모든 Client에 괄호 안에 있는 데이터를 전송합니다
write(val) or write(buf, len)
val 와 **buf**의 타입은 **char or byte**

Server class

The Server class creates servers which can send data to and receive data from connected clients (programs running on other computers or devices).

- Server
- EthernetServer()
- begin()
- available()
- write()
- print()
- println()

print(): Client로 데이터 전송

println(): Client로 데이터 전송

기존 시리얼 **print** 와 유사

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

» 용도에 따라 기억하자

Client : Class 자체, 직접 호출 x

EthernetClient() : 클라이언트 **생성**

```
27  
28 EthernetClient client;  
29
```

클라이언트를 생성합니다.
선언된 class를 if문에서 사용하면,
Client의 준비상태를 확인 가능

connect() : Server에 연결 **요청**

```
89  
90 if (client.connect(server_dns, 80))  
91 {
```

Server의 IP or URL 및 Port번호를 괄호 안에 입력하여 연결을 시도합니다.

Return 값으로 결과 확인 가능

connected()는 연결여부 확인 함수

Client class

The client class creates clients that can connect to servers and send and receive data.

- Client
- EthernetClient()
- if (EthernetClient)
- connected()
- connect()
- write()
- print()
- println()
- available()
- read()
- flush()
- stop()

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

≫ 용도에 따라 기억하자

write() : 연결된 서버로 **데이터 전송**
Server와 유사

read() : 데이터 **읽기**

연결된 **서버로부터** 들어온 데이터(1byte)를 읽습니다.

참고로 이 함수는 Stream Class와 **상속관계**입니다 (관련해서 C++ 공부 필요)

available() : **수신** 데이터 유무 **확인**

```
73 ▾ {  
74     while (client.available())  
75 ▾ {
```

서버로부터 데이터를 받았다면,
그 데이터 수를 반환합니다
이를 통해 데이터가 들어왔는지
확인할 수 있습니다

Client class

The client class creates clients that can connect to servers and send and receive data.

- Client
- EthernetClient()
- if (EthernetClient)
- connected()
- connect()
- write()
- print()
- println()
- available()
- read()
- flush()
- stop()

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

» 용도에 따라 기억하자

flush() : 데이터 전송 확인

발신할 데이터가 전부 전송될 때까지
대기하도록 하는 함수입니다
즉, 자신이 보낸 데이터가 모두 전송될 때까지
다른 작업을 하지 않도록 하죠

stop() : 서버와의 연결 종료

```
86 ▾ {  
87     client.stop();  
88     SerialUSB.println(  
...
```

print() : 서버로 데이터 전송

println() : 서버로 데이터 전송

기존 시리얼 print 와 유사

Client class

The client class creates clients that can connect to servers and send and receive data.

- Client
- EthernetClient()
- if (EthernetClient)
- connected()
- connect()
- write()
- print()
- println()
- available()
- read()
- flush()
- stop()

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

» 용도에 따라 기억하자

read() 함수에서 Stream Class와 상속관계에 있다고 했습니다

따라서 Stream Class의 몇 가지 함수를 그대로 사용할 수 있습니다
(자세한 내용은 Class 상속에 관해 검색)

readStringUntil() : 특정 문자 찾기

```
76 //라인(\n = line feed)을 기준으로 문자열을 저
77 String line = client.readStringUntil('\n');
78 SerialUSB.println(line); // 실제 받아오
```

괄호() 안에 들어간 '문자(char 1byte)'를
찾아 지금까지 읽은 전체 문자열을 반환
특정 문자를 기준으로 데이터를 한번에 받
고 싶을 때 사용하면 좋습니다.

Stream

Functions

- available()
- read()
- flush()
- find()
- findUntil()
- peek()
- readBytes()
- readBytesUntil()
- readString()
- readStringUntil()
- parseInt()
- parseFloat()
- setTimeout()

Ethernet 기초 실습 (1)

8. 네트워크 설정에 관한 함수

» 찾는 방법을 알았으니 스스로 해보자!

Characters Func.

Characters

- `isAlphaNumeric()`
- `isAlpha()`
- `isAscii()`
- `isWhitespace()`
- `isControl()`
- `isDigit()`
- `isGraph()`
- `isLowerCase()`
- `isPrintable()`
- `isPunct()`
- `isSpace()`
- `isUpperCase()`
- `isHexadecimalDigit()`

millis()

Time

- `millis()`
- `micros()`
- `delay()`
- `delayMicroseconds()`

String

String Class

Description

The String class, part of the core as of version 0019, allows you to use and manipulate strings of text in more complex ways than `character arrays` do. You can concatenate Strings, append to them, search for and replace substrings, and more. It takes more memory than a simple character array, but it is also more useful.

For reference, character arrays are referred to as strings with a small s, and instances of the String class are referred to as Strings with a capital S. Note that constant strings, specified in "double quotes" are treated as char arrays, not instances of the String class.

Examples

- `StringConstructors`
- `StringAdditionOperator`
- `StringIndexOf`
- `StringAppendOperator`
- `StringLengthTrim`
- `StringCaseChanges`
- `StringReplace`
- `StringRemove`
- `StringCharacters`
- `StringStartsWithEndsWith`
- `StringComparisonOperators`
- `StringSubstring`

See Also

- `string`: character arrays
- `Variable Declaration`

Functions

- `String()`
- `charAt()`
- `compareTo()`
- `concat()`
- `c_str()`
- `endsWith()`
- `equals()`
- `equalsIgnoreCase()`
- `getBytes()`
- `indexOf()`
- `lastIndexOf()`
- `length()`
- `remove()`
- `replace()`
- `reserve()`
- `setCharAt()`
- `startsWith()`
- `substring()`
- `toCharArray()`
- `toInt()`
- `toFloat()`
- `toLowerCase()`
- `toUpperCase()`
- `trim()`

W5100S-EVB-Pico

IoT를 위한 Ethernet 기초 실습(2)



Ethernet 기초 실습 (2)

1. TCP 서버 LED Control

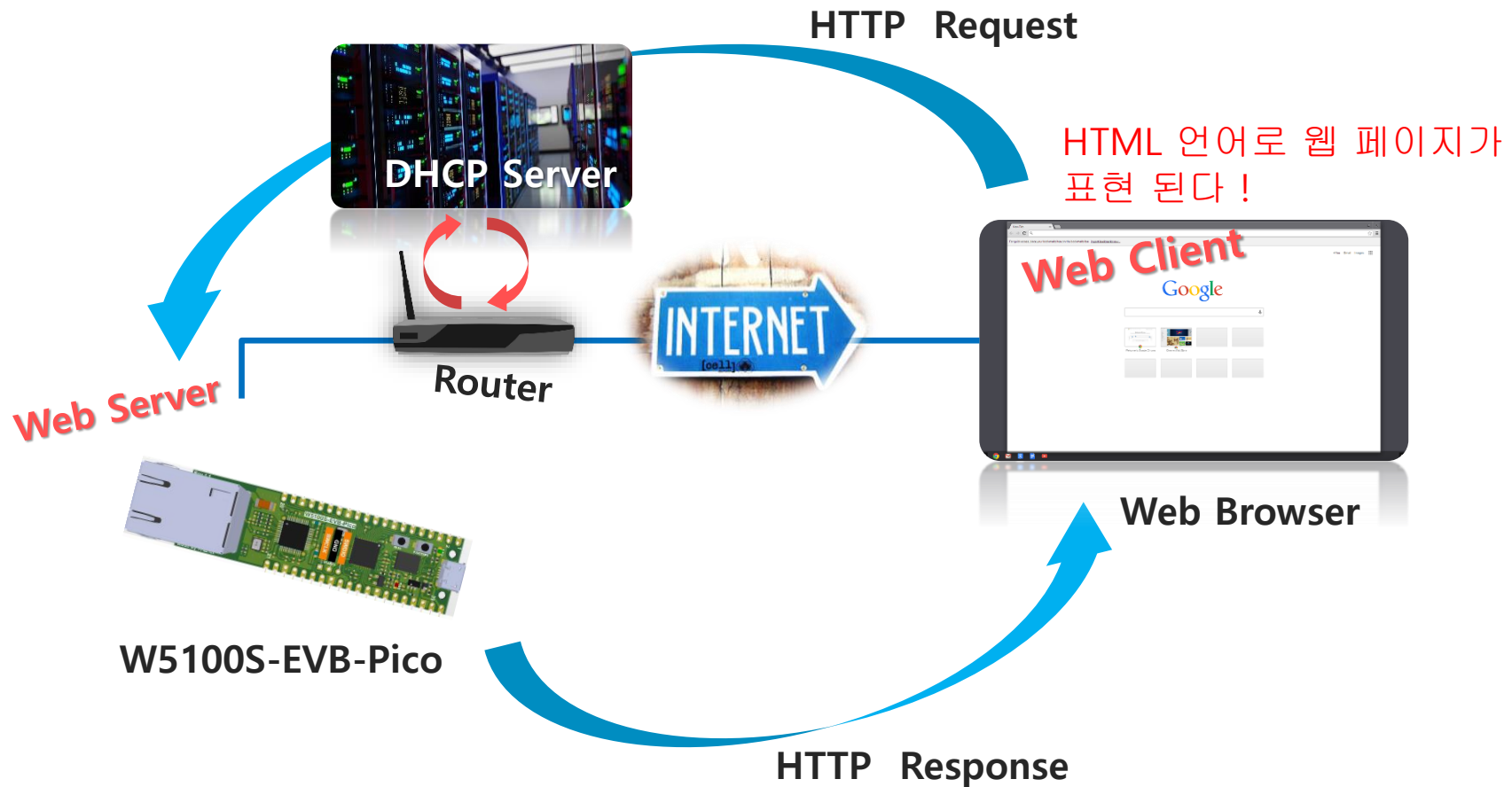
Loopback Server 코드를 이용하여 LED ON/OFF를 해봅시다!

- Client가 '0' 을 입력하면 LED OFF, Client가 '1'을 입력하면 LED ON

그 밖의 다른 값을 입력하면 LED 상태는 변화 없음

Ethernet 기초 실습 (2)

2. Web Server (LED Control)



2. Web Server (LED Control)

>> Web Server란?

- **Web Server**
 - Client로 부터 HTTP 요청을 받아, HTTP 응답하는 프로그램. Ex) Google, Naver 웹 브라우저 서비스
- **HTTP** (**H**yper **T**ext* **T**ransfer **P**rotocol)
 - Web Server와 Web Client간의 통신 약속
- **HTML** (**H**yper **T**ext **M**arkup* **L**anguage)
 - HyperText를 작성하기 위해 개발된 프로그래밍 언어
 - 웹 문서를 만들기 위해 사용하는 기본적인 프로그래밍 언어의 한 종류

Ethernet 기초 실습 (2)

2. Web Server (LED Control)

>> HTTP Method

- **GET**
 - Web Client가 Web Server로부터 값을 얻기 위한 방법
Ex) Google 검색창에 WIZnet라고 검색
- **POST**
 - Web Client가 Web Server로 변경된 값을 전달하기 위한 방법
Ex) 블로그에 글 남기기
Ex) 웹 페이지에 아두이노 센서 데이터 전송

https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Ethernet 기초 실습 (2)

2. Web Server (LED Control)

➤➤ Web Server 예제 실행

- https://github.com/matthew1205/PicoAcademy/tree/main/WebServer_LED_Con

➤➤ 내 네트워크 환경으로 값 변경 필요

웹 브라우저(**Edge, Chrome**)에서 **W5100S-EVB-Pico** 서버 주소를 입력하여
연결

Ethernet 기초 실습 (2)

2. Web Server (LED Control)

➤➤ 3 Color LED를 컨트롤 하는 것으로 변경해 보자

- https://github.com/matthew1205/PicoAcademy/tree/main/WebServer_LED_Con

Ethernet 기초 실습 (2)

3. Web Server (Sensor Data)

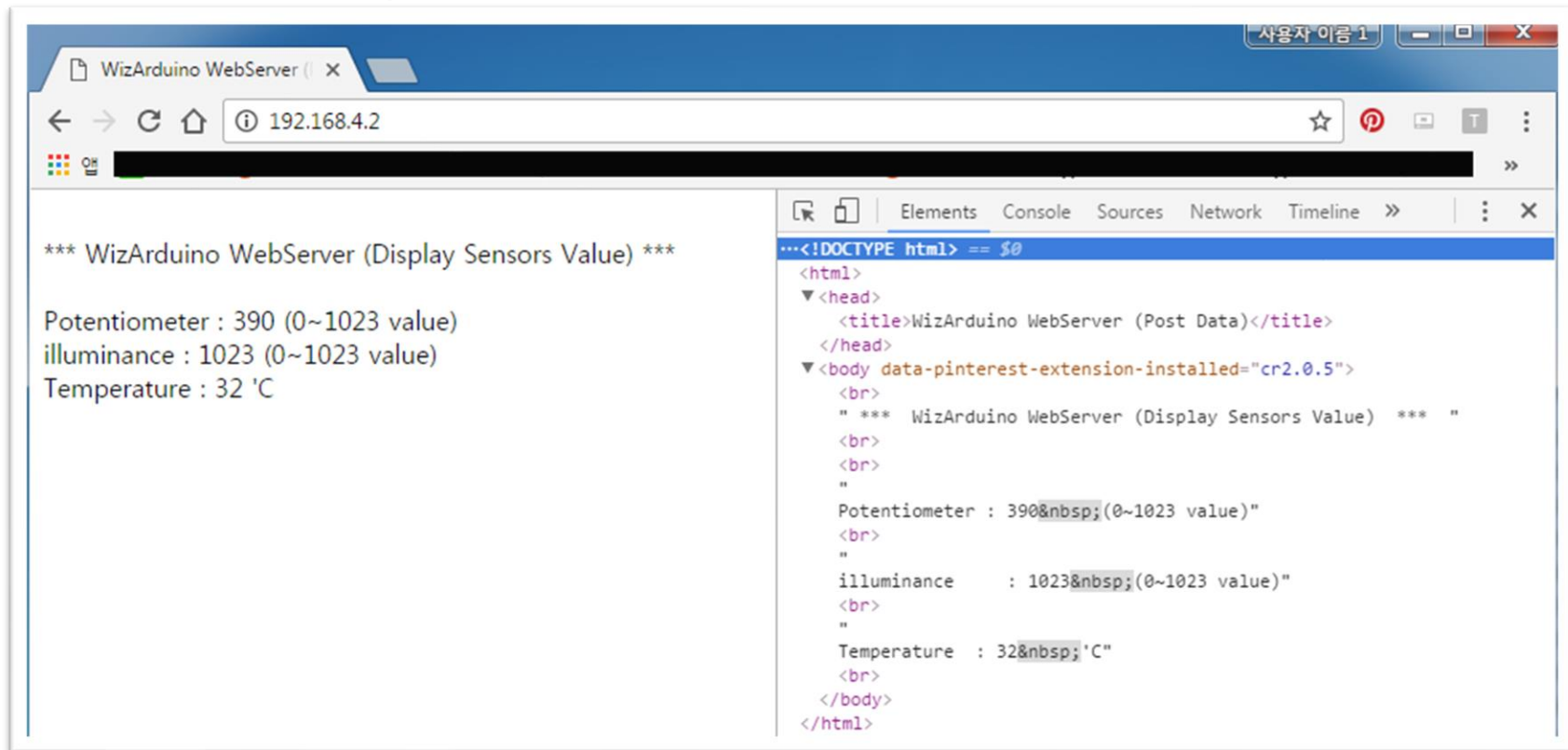
➤➤ CDS 이외에 다른 센서 값을 출력해 보자

- https://github.com/matthew1205/PicoAcademy/tree/main/WebServer_SensorValue

Ethernet 기초 실습 (2)

4. 부록 1

>> Web Page의 실체



Page 위에서 오른쪽 마우스를 클릭하고 “소스보기” or “원본보기” 클릭
Chrome 브라우저 사용자라면 단축키 **F12** !

Ethernet 기초 실습 (2)

5. 부록 2

» 인터넷 데이터 전송의 실체

The image shows a Wireshark packet capture window. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for various functions. The main display area shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 28) is expanded, showing details for Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Transport Layer Security. The packet data is displayed in hexadecimal and ASCII format at the bottom.

No.	Time	Source	Destination	Protocol	Length	Info
22	6.643316	140.82.114.26	192.168.11.11	TCP	66	443 → 14558 [ACK] Seq=1 Ack=2 Win=77 Len=0 SLE=1 SRE=2
23	6.676630	192.168.11.11	168.126.63.1	DNS	80	Standard query 0xdefe A cc-api-data.adobe.io
24	6.677944	168.126.63.1	192.168.11.11	DNS	128	Standard query response 0xdefe A cc-api-data.adobe.io A 18.176.82.97 A 18.179.143.46 A 35.73.100.104
25	6.678347	192.168.11.11	18.176.82.97	TCP	66	12626 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
26	6.715444	18.176.82.97	192.168.11.11	TCP	66	443 → 12626 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_PERM=1 WS=4096
27	6.715505	192.168.11.11	18.176.82.97	TCP	54	12626 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
28	6.716291	192.168.11.11	18.176.82.97	TLSv1.2	241	Client Hello
29	6.752544	18.176.82.97	192.168.11.11	TCP	60	443 → 12626 [ACK] Seq=1 Ack=188 Win=28672 Len=0
30	6.752544	18.176.82.97	192.168.11.11	TLSv1.2	158	Server Hello
31	6.752679	18.176.82.97	192.168.11.11	TCP	1514	443 → 12626 [ACK] Seq=105 Ack=188 Win=28672 Len=1460 [TCP segment of a reassembled PDU]
32	6.752679	18.176.82.97	192.168.11.11	TCP	1514	443 → 12626 [ACK] Seq=1565 Ack=188 Win=28672 Len=1460 [TCP segment of a reassembled PDU]
33	6.752711	192.168.11.11	18.176.82.97	TCP	54	12626 → 443 [ACK] Seq=188 Ack=3025 Win=262656 Len=0
34	6.752729	18.176.82.97	192.168.11.11	TCP	1514	443 → 12626 [ACK] Seq=3025 Ack=188 Win=28672 Len=1460 [TCP segment of a reassembled PDU]
35	6.752729	18.176.82.97	192.168.11.11	TLSv1.2	165	Certificate

▼ Frame 28: 241 bytes on wire (1928 bits), 241 bytes captured (1928 bits) on interface \Device\NPF_{DC6AD418-B5D5-420E-B0B3-E8B30EDC5282}, id 0

▼ Ethernet II, Src: Micro-St_0e:dd:47 (d8:bb:c1:0e:dd:47), Dst: EFMNetwo_be:8a:38 (88:36:6c:be:8a:38)

- Destination: EFMNetwo_be:8a:38 (88:36:6c:be:8a:38)
- Source: Micro-St_0e:dd:47 (d8:bb:c1:0e:dd:47)
- Type: IPv4 (0x0800)

▼ Internet Protocol Version 4, Src: 192.168.11.11, Dst: 18.176.82.97

▼ Transmission Control Protocol, Src Port: 12626, Dst Port: 443, Seq: 1, Ack: 1, Len: 187

▼ Transport Layer Security

0000 88 36 6c be 8a 38 d8 bb c1 0e dd 47 08 00 45 00 61 ..8...G..E.
0010 00 e3 2f d1 40 00 80 06 00 00 c0 a8 0b 0b 12 b0 ../_@.....
0020 52 61 31 52 01 bb 0b 87 5c 77 02 05 0e 4e 50 18 RaIR... \w...NP.
0030 04 02 31 9a 00 16 03 03 00 b6 01 00 00 52 03 ..1.....
0040 03 64 ad 7c 50 88 e8 dd 14 d2 92 25 a7 88 2f 13 .d|P...%.../
0050 f2 c4 04 6f de fd 50 73 42 33 5b 0f f3 8b 2c 0a ...o..Ps B3[...
0060 17 00 00 26 c0 2c c0 2b c0 30 c0 2f c0 24 c0 23 ...&...+ @ / \$ #
0070 c0 28 c0 27 c0 0a c0 09 c0 14 c0 13 00 9d 00 9c .(.....
0080 00 3d 00 3c 00 35 00 2f 00 0a 01 00 00 63 00 00 ..<S/.....c..
0090 00 19 00 17 00 00 14 63 63 2d 61 70 69 2d 64 61c c-api-da
00a0 74 61 2e 61 64 6f 62 65 2e 69 6f 00 05 00 05 01 ta.adobe .io.....
00b0 00 00 00 00 00 0a 00 00 06 00 1d 00 17 00 18
00c0 00 0b 00 02 01 00 0d 00 1a 00 18 08 04 08 05
00d0 08 06 04 01 05 01 02 01 04 03 05 03 02 03 02 02#...
00e0 06 01 06 03 00 23 00 00 17 00 00 ff 01 00 01
00f0 00

Wireshark 프로그램을 이용하면, 현재 기기의 네트워크 데이터 송수신 과정을 살펴볼 수 있다. (IoT 기기 개발에 도움을 줄 수 있음) (암호화?)