

Version 1.0.2
Jun2018



WIZnet Academy 2018

Serial to Ethernet Deep-Dive Training

James Kim

✉ jameskim@wiznet.io

시작에 앞서: 위즈네트 제품을 활용한다면 이곳만은 꼭!

WIZnet Developers

Developer Forum

<https://forum.wiznet.io>

The screenshot shows the WIZnet Developer Forum homepage. It features a navigation bar with 'Products', 'Wiki', and 'GitHub'. Below the navigation is a search bar and a user login area. The main content area includes sections for 'Announcements' (with a link to 'Notice & Forum Information'), 'Product FAQ' (listing S2E Module, TCP/IP Chip, WiFi Module, Network Module, Application Module, and Internet MCU), 'WIZnet Contest' (with links to 'Discussion on the Contest held at WIZnet #', '2014 Design Contest', and '2017 IoT Design Contest'), 'TCP/IP Chip' (listing W150A+, W1510, W5200, WS300, WS5500), 'Internet MCU' (listing W7100A, W7500, W7100A-S2E, W7500-S2E), and 'S2E Module' (listing W2100 Series, W2750 Series, and W2vSP). The page also displays several forum posts and announcements.

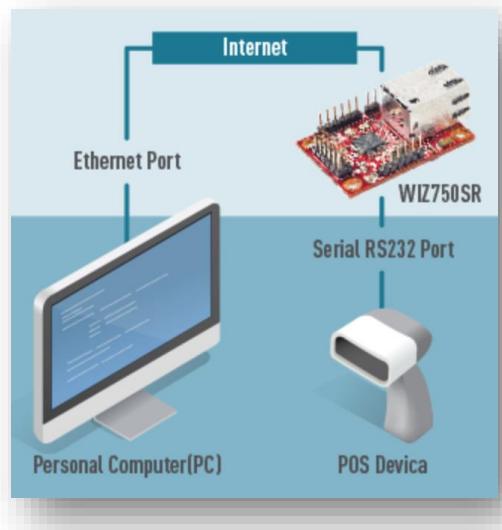
Document Wiki

<https://wizwiki.net/wiki>

The screenshot shows the WIZnet Document Wiki page for the 'WIZ750SR' module. It includes a navigation bar with 'Search', 'Recent Changes', 'Media Manager', and 'Stamp'. The main content area has a sidebar with 'Table of Contents' and a central panel featuring a 3D image of the WIZ750SR module. The page lists 'Products' (WIZ750SR), 'Documents' (Datasheet, Datasheet, Related Products, Where to Buy), 'Documents' (Project Overview, Features, Product Commands), 'Datasheet' (WIZ750SR / WIZ750SR-EVB, Hardware Specification, Schematic / Part List / Dimension, Electrical Characteristics, Connector Specification), and 'Download' (Software Download, Firmware Binary, Firmware Source code, Documentation).

시작에 앞서: 선행 강좌 소개

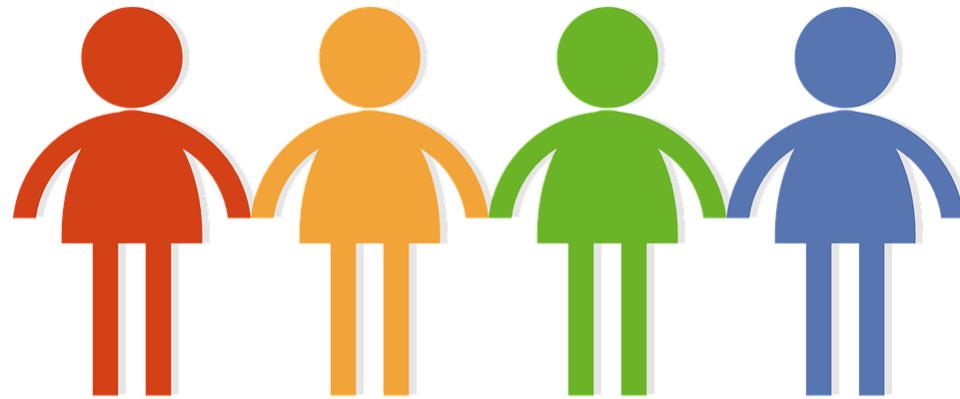
Serial to Ethernet 컨버터 강좌



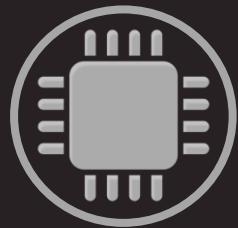
- WIZ750SR-EVB 기반 강의
- 시리얼 통신 기초
- Serial to Ethernet 컨버터 설정 및 활용
- 응용 사례 소개 및 제품 컨설팅

WIZnet Academy
<http://wiznetacademy.com>

시작에 앞서: Break the Ice



“왜 이번 강좌에 참가 하셨나요?”



WIZnet

Serial to Ethernet Converters

WIZnet S2E Converters (cont'd)

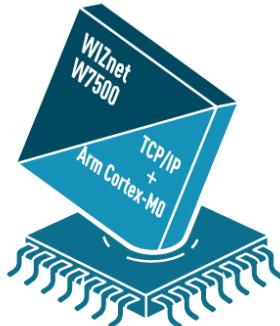
>> Product Lineup: Modules

Product Name	Product Image	UART Port	Signal Comm. Type	Serial I/F (& Pin Pitch)	Ethernet I/F	Operation Temp. (°C)	Dimension (mm)	Input Voltage
WIZ750SR-TTL		1	TTL (3.3V)	2.54mm Pitch 6x2 Pin-header	RJ45	0°C ~ 70	48 x 30 x 18	3.3V
WIZ750SR-232		1	RS-232					
WIZ750SR-485		1	RS-422/485					
WIZ750SR-100		1	TTL (3.3V)	2.00mm Pitch 1x12 Pin header	MDI (PHY I/F)	-40 ~ 85	50 x 30 x 12	3.3V
WIZ750SR-105		1	TTL (3.3V)	2.00mm Pitch 6x2 Pin-header	RJ45	-40 ~ 85	40 x 62 x 17	3.3V
WIZ750SR-110		1	RS-232	DB9	RJ45	-40 ~ 85	75 x 50 x 17	5V (DC Jack)
WIZ752SR-120		2	TTL (3.3V)	2.00mm Pitch 1x14 Pin header	MDI (PHY I/F)	-40 ~ 85	50 x 30 x 9	3.3V
WIZ752SR-125		2	RS-232	DB9	RJ45	-40 ~ 85	60 x 89 x 18	5V (DC Jack)
WIZ140SR		4	TTL (3.3V)	2.50mm Pitch 2x14 Pin header	MDI (PHY I/F)	0 ~ 70	48 x 36 x 16	3.3V
WIZ145SR		4	TTL (3.3V)	2.50mm Pitch 2x14 Pin header	RJ45	0 ~ 70	48 x 61 x 25	3.3V

WIZnet S2E Converters (cont'd)

» Product Lineup: W7500(P)-S2E Chip

- Features



- ✓ **WIZ750SR Series 펌웨어 탑재**
 - Device / MCU의 UART 인터페이스를 통해 손쉽게 Ethernet Networking 기능 제공
 - 하드웨어 구성에 따라 TTL, RS-232C, RS-422/485 serial communication 지원
 - Command mode를 통해 UART로 제품 configuration 가능
- ✓ **제품 디자인을 위한 상세 가이드 제공**
 - Hardware Design Guide & References
- ✓ **MAC address 포함 (WIZnet OUI)**
- ✓ **사용자 목적에 따른 Customized Firmware 탑재 및 공급 가능**



<W7500>
MII supported
for PHY interface



<W7500P>
10/100 Ethernet PHY
integrated

W7500 / W7500P MCU

Arm® Cortex™-M0 based microcontroller

- Up to 48MHz / 128KB Flash / 16KB RAM
- 3-UART, 2-SPI, 12bit 8ch-ADC, Timer/PWM, 53-GPIOs, SWD
- 64 TQFP Package (7x7mm)

Hardwired TCP/IP core integrated

- Hardwired TCP/IP stack + IEEE 802.3 Ethernet MAC
- 8-channel hardware sockets with 32KB buffers(SRAM)
- Supports the TCP, UDP, IPv4, ICMP, ARP, IGMP and PPPoE

WIZnet S2E Converters (cont'd)

>> Settings & Options

Operation Mode

- TCP server
- TCP client
- TCP mixed
 - TCP server / client
- UDP
 - 1:1 mode
 - 1:N mode

Network Settings

- IP (Static / DHCP)
- Local port
- Gateway
- Subnet mask
- DNS server
- Remote host / Port

Serial Settings

- Baud rate
 - Up to 230.4kbps
- Data bits
- Parity bit
- Stop bits
- Flow control
 - XON/XOFF
 - RTS/CTS

User's IO

- User IO A
 - User IO B
 - User IO C
 - User IO D
- | |
|---------------|
| - Analog In |
| - Digital In |
| - Digital Out |

Network Options

- Inactivity timer*
- Reconnection interval*
- Keep-Alive*
- Search ID code
- Connection Password

Serial Options

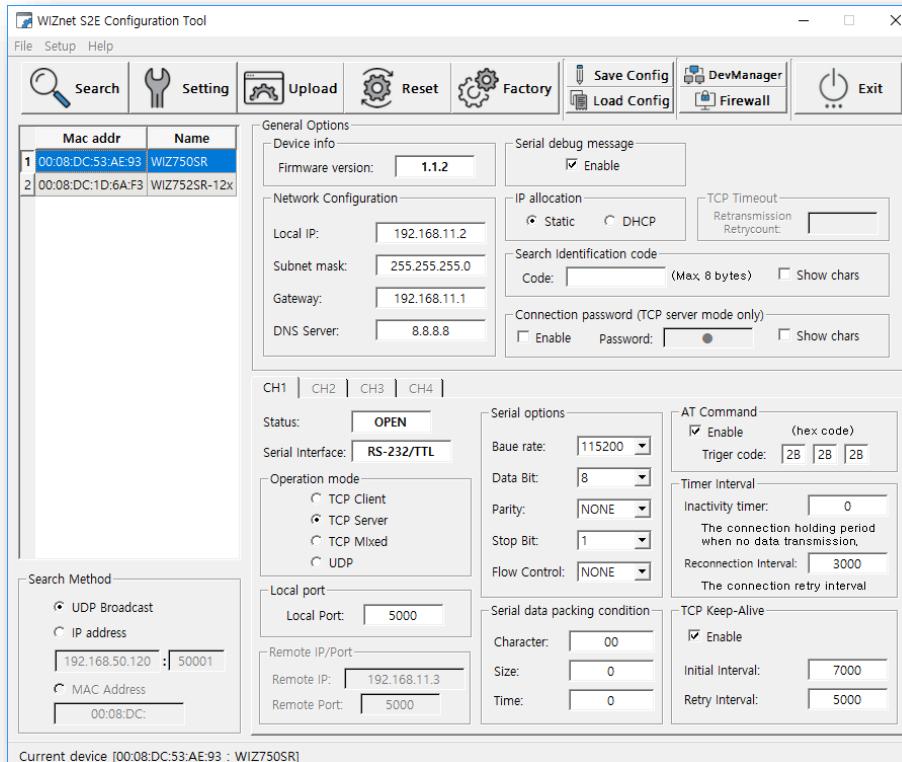
- Serial command mode*
- Data packing options*
 - Timer
 - Size
 - Char delimiter

* Settings are based on WIZ750SR 1-Port Serial to Ethernet module

WIZnet S2E Converters (cont'd)

>> Support Utilities

- WIZnet S2E Configuration Tool



Features

- WIZ750SR, WIZ752SR series 대상
- UDP broadcast / TCP unicast Search 지원
- 제품 환경 설정 기능
 - Network settings / options
 - Serial settings / options
- 네트워크 펌웨어 업데이트 기능
- Python 기반, OS 독립적으로 동작



**WIZnet-S2E-Tool-GUI
GitHub Repo.**

WIZnet S2E Converters (cont'd)

» Support Utilities

- **CLI-based Configuration Tool**

```

명령 프롬프트
C:\WIZnetTool-master>python wiz750_configTool.py -h
usage: wiz750_configTool.py [-h] [-d MACADDR] [-a] [-u FWFILE] [-s] [-c] [-r]
                            [-f] [-mnode {0,1,2,3}] [-alloc {0,1}] [--ip IP]
                            [--subnet SUBNET] [--gw GW] [--dns DNS]
                            [--port PORT] [--rip IP] [--report PORT]
                            [--baud0 BAUD0] [--data0 {0,1}]
                            [--parity0 {0,1,2}] [--stop0 {0,1}]
                            [--flow0 {0,1,2}] [--time0 TIME0] [--size0 SIZE0]
                            [--char0 CHAR0] [--baud1 BAUD1] [--data1 {0,1}]
                            [--parity1 {0,1,2}] [--stop1 {0,1}]
                            [--flow1 {0,1,2}] [--time1 TIME1] [--size1 SIZE1]
                            [--char1 CHAR1] [--it timer] [--ka {0,1}]
                            [--ki number] [--ke number] [--ri number]
                            [--rv timer] [--ra {0,1}] [--rs number]
                            [--re number] [--rr number] [--cp {0,1}] [--np pw]
                            [--sp value] [--dg {0,1}] [--te {0,1}]
                            [--ss 3-byte hex] [--setfile SETFILE]
                            [--getfile GETFILE] [-m]

<WIZnet CLI Configuration Tool>

optional arguments:
  -h, --help            show this help message and exit
  -d MACADDR, --device MACADDR
                        Device mac address to configuration
  -a, --all             Configuration about all devices (in mac_list.txt)

Firmware Upload:
  -u FWFILE, --upload FWFILE
                        Firmware upload from file

No parameter Options:
  -s, --search          Search devices (in same network)
  -c, --clear            Mac list clear
  -r, --reset            Reboot device
  -f, --factory          Factory reset

Network Configuration:
  --mnode {0,1,2,3}      Network operation mode (0: tcpclient, 1: tcpserver, 2: mixed, 3: udp)
  --alloc {0,1}           IP address allocation method (0: Static, 1: DHCP)
  --ip IP                Local ip address
  --subnet SUBNET        Subnet mask
  --gw GW                Gateway address
  --dns DNS              DNS server address
  --port PORT            Local port number
  --rip IP               Remote host IP address / Domain
  --report PORT          Remote host port number

UART #0 Configurations:
  --baud0 BAUD0          bauie rate (300 to 230400)

```

Features

- Python 기반, OS 독립적으로 동작
 - Windows / Linux / MAC OS 지원
- WIZ750SR, WIZ752SR series 지원
- Config-Tool for Windows의 전체 기능 지원
- 여러 개 제품의 동시 설정 기능 지원
- 공개 소프트웨어 (develop ver.)



WIZnet-S2E-Tool
GitHub Repo.

Serial to Ethernet 컨버터 활용 및 커스터마이즈

<실습>

- 실습 개요 및 하드웨어 구성
- ⚙️ [실습 #1] 개발 환경 구축 (Eclipse & ARM GCC)
- ⚙️ [실습 #2] WIZ750SR 시작하기 (Getting Started)
- ⚙️ [실습 #3] W7500x Project 만들기 실습
- ⚙️ [실습 #4] Standard 펌웨어 구조 & 소스 분석
- ⚙️ [실습 #5] Serial to Ethernet Customize 예제 구현 (1)
- ⚙️ [실습 #6] Serial to Ethernet Customize 예제 구현 (2)
- ⚙️ [실습 #7] Serial to Ethernet Customize 예제 구현 (3)

실습 구성 소개

구성 요소 및 하드웨어

실습 소개

>> Target: 'Serial to Ethernet' 컨버터 완전 분석

- WIZnet Serial to Ethernet 개발 환경 구축
- 프로젝트 구조 파악 및 코드 분석
- 예제 코드의 동작 확인

<개발 환경>

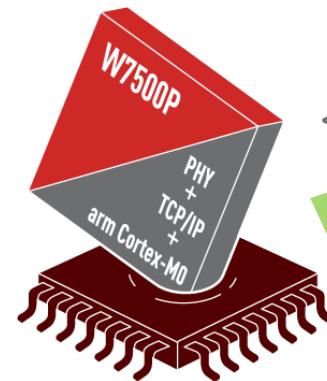


< Eclipse IDE for C/C++ Developers >



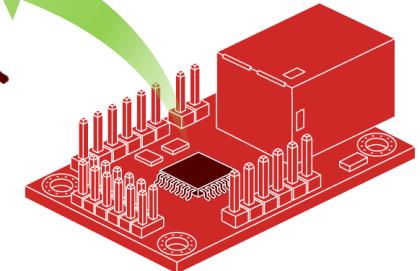
< GNU Arm Embedded Toolchain >

<하드웨어>



W7500P
iMCU Based

< WIZ750SR-EVB Kit >



실습 소개: 하드웨어 구성

>> WIZ750SR-TTL Evaluation Board Set

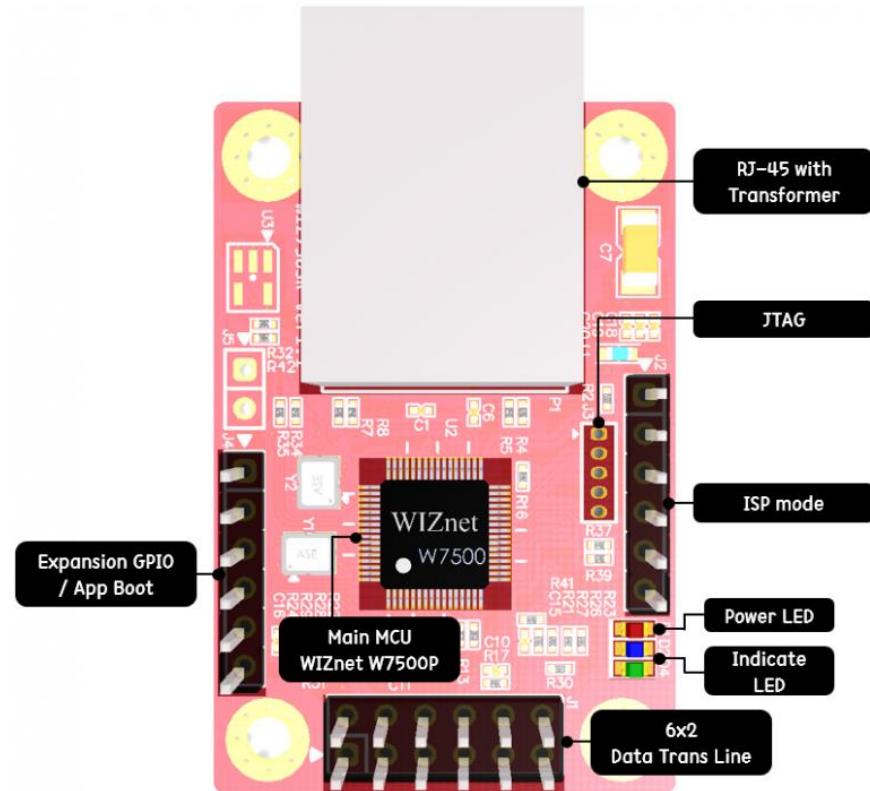


<WIZ750SR-TTL Module>

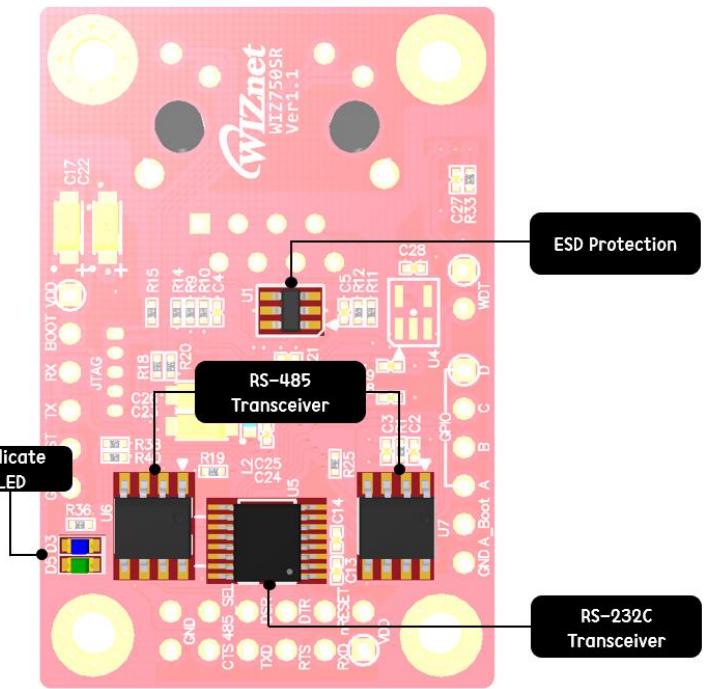


실습 소개: 하드웨어 구성 (cont'd)

>> WIZ750SR Callout



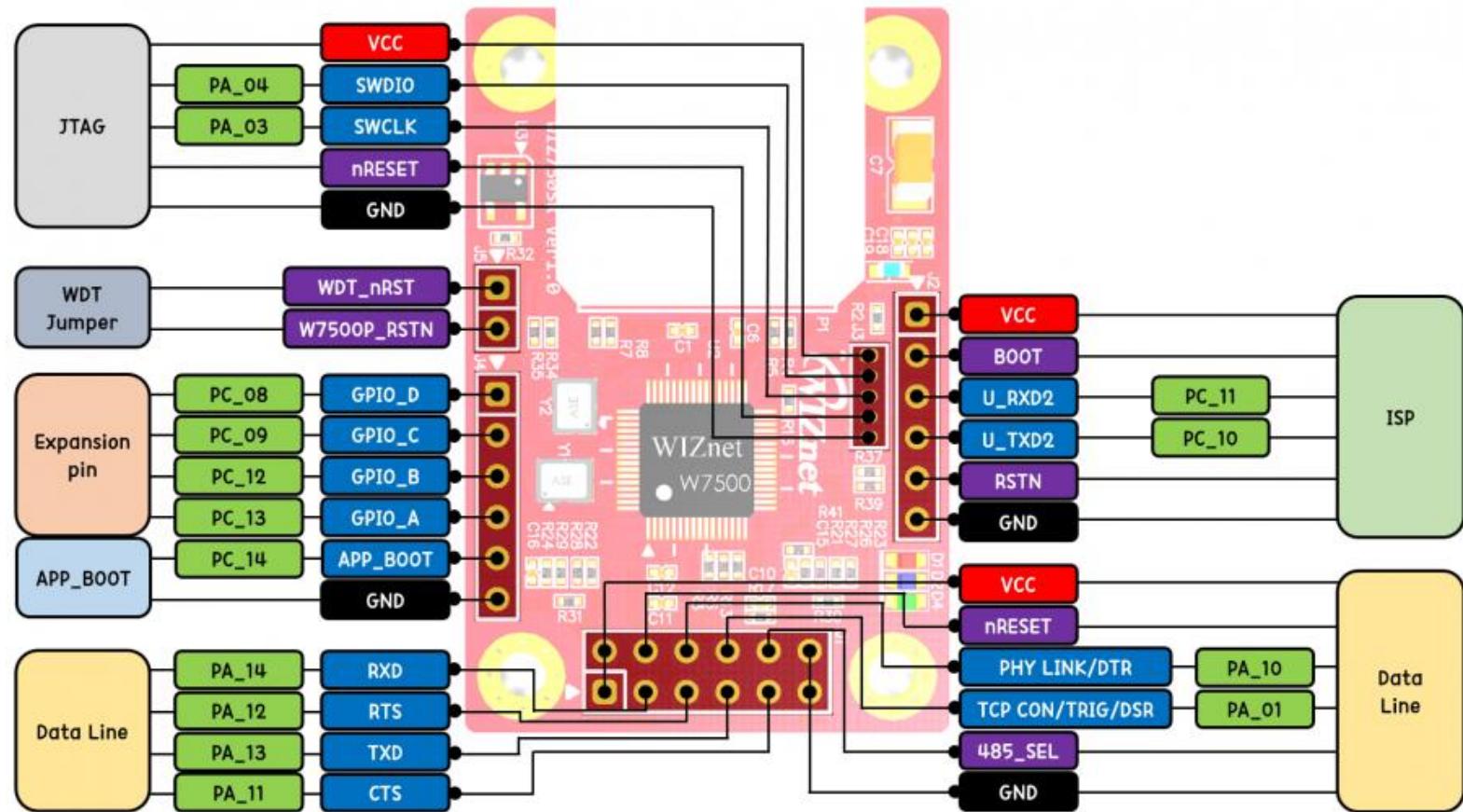
< Rev 1.1 Top >



< Rev 1.1 Bottom >

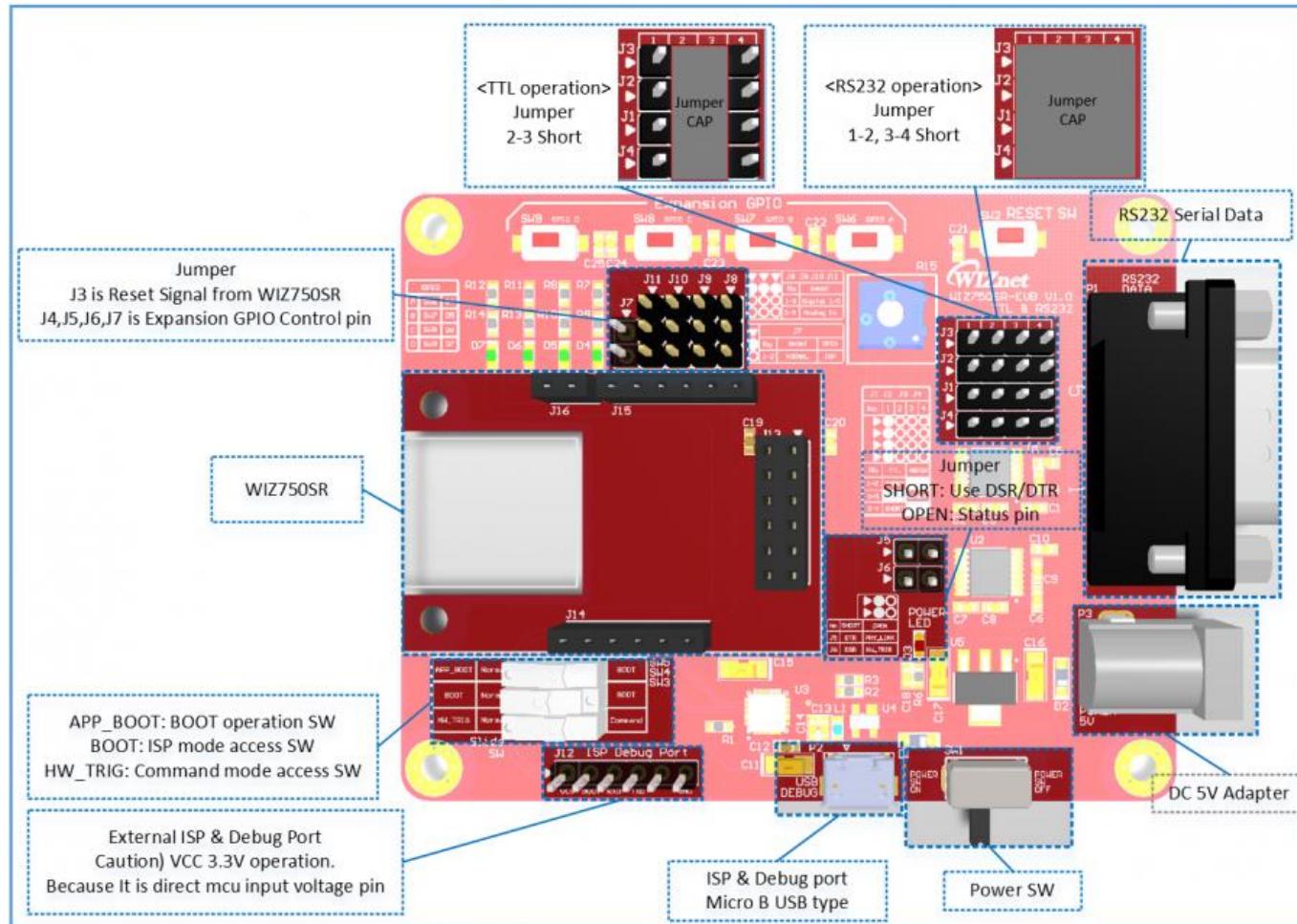
실습 소개: 하드웨어 구성 (cont'd)

>> WIZ750SR Pinout



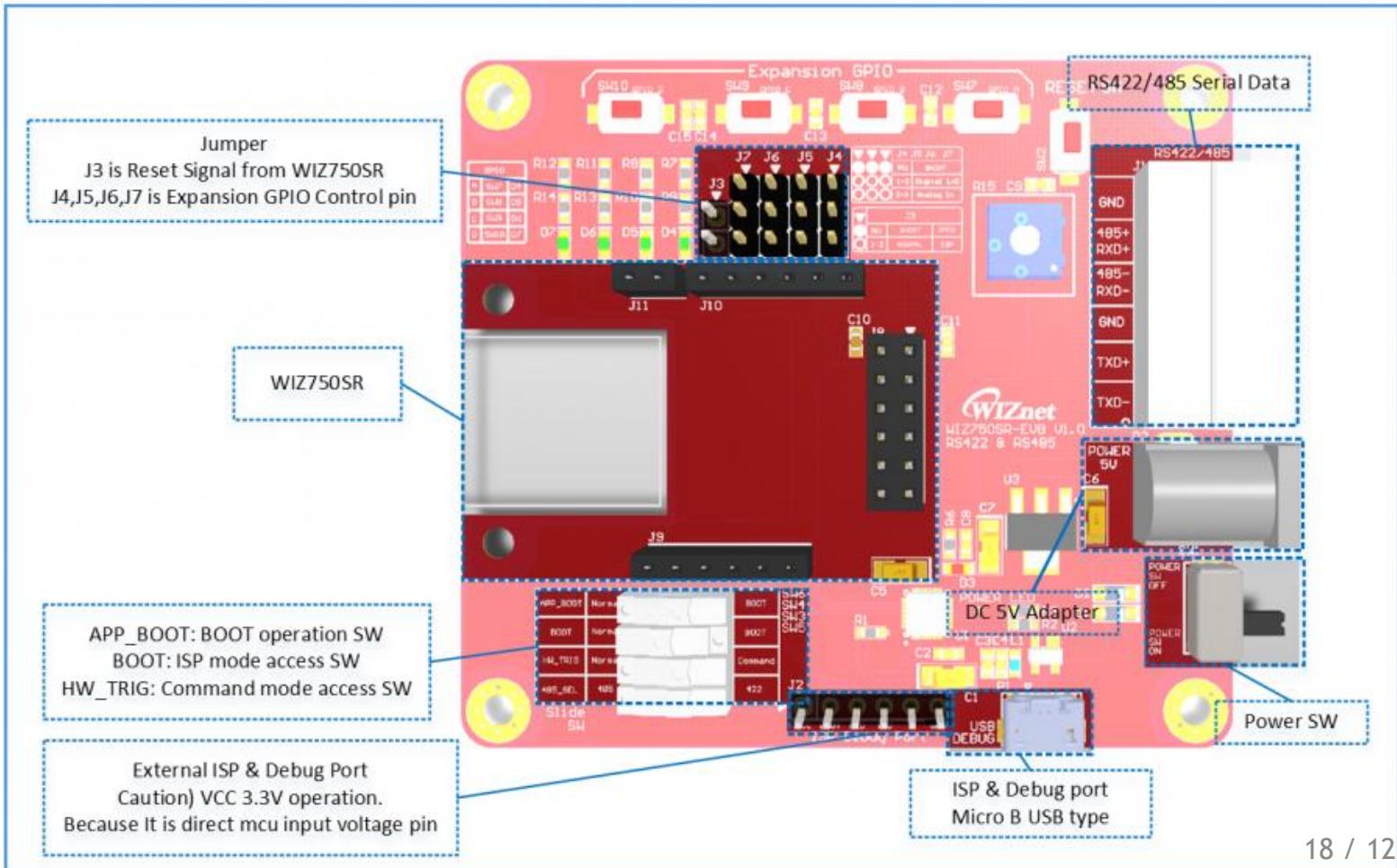
실습 소개: 하드웨어 구성 (cont'd)

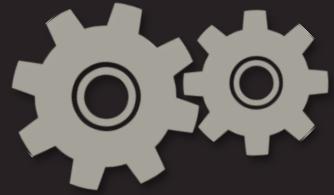
>> WIZ750SR-EVB: TTL/RS-232 Version



실습 소개: 하드웨어 구성 (cont'd)

>> WIZ750SR-EVB: RS-422/485 Version





실습 #1

개발 환경 구축

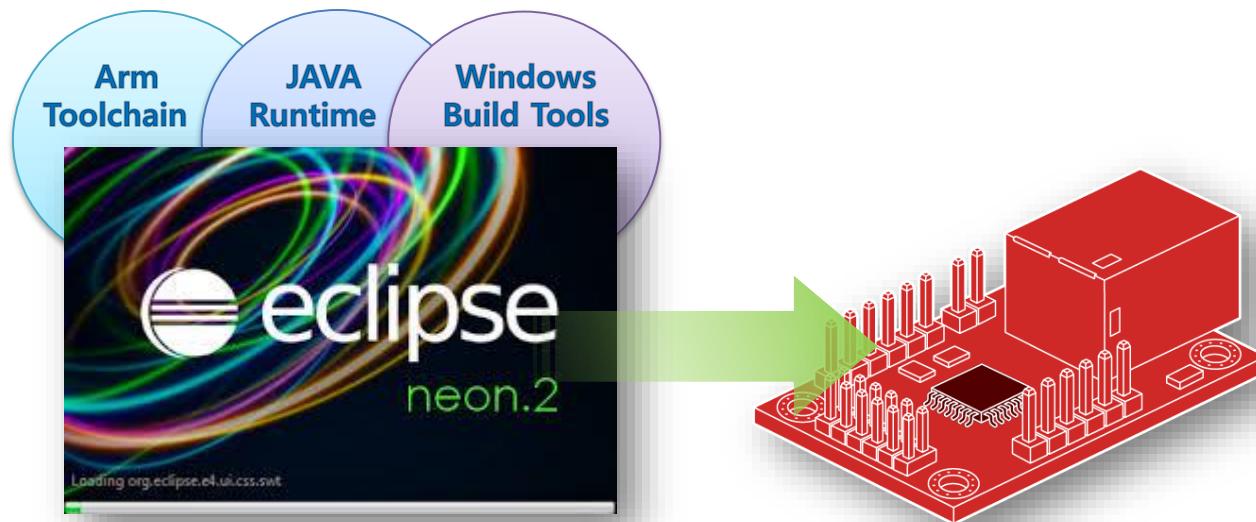
- **GNU Arm ToolChain 다운로드 및 설치**
- **환경 변수 설정 및 Build Tools 설치**
- **Eclipse IDE for C/C++ 다운로드 및 설치**
 - JAVA JRE 다운로드 및 설치
 - 관련 Plug-ins 다운로드 및 설치



실습 #1: 개발 환경 구축

>> Windows 운영체제를 위한 GCC ARM 개발 환경 구축

- [GNU Arm Embedded Toolchain](#) 다운로드 및 설치
- [JAVA Runtime Environment](#) 다운로드 및 설치
- [Windows Build Tools](#) 다운로드 및 설치
- [Eclipse IDE for C/C++](#) 다운로드 및 설치
 - 관련 Plug-ins 다운로드 및 설치



실습 #1: 개발 환경 구축 (cont'd)

>> GNU Arm Embedded Toolchain: Download

<https://launchpad.net/gcc-arm-embedded/5.0/5-2016-q2-update>

Milestone information

Project:	GNU Arm Embedded Toolchain	Series:	5.0
Version:	5-2016-q2-update	Released:	2016-06-28
Registrant:	Pietro Palazzo	Release registered:	2016-06-28
Active:	No. Drivers cannot target bugs and blueprints to this milestone.		

Activities

Assigned to you:	No blueprints or bugs assigned to you.
Assignees:	1 Thomas Preud'homme
Blueprints:	No blueprints are targeted to this milestone.
Bugs:	1 Fix Released

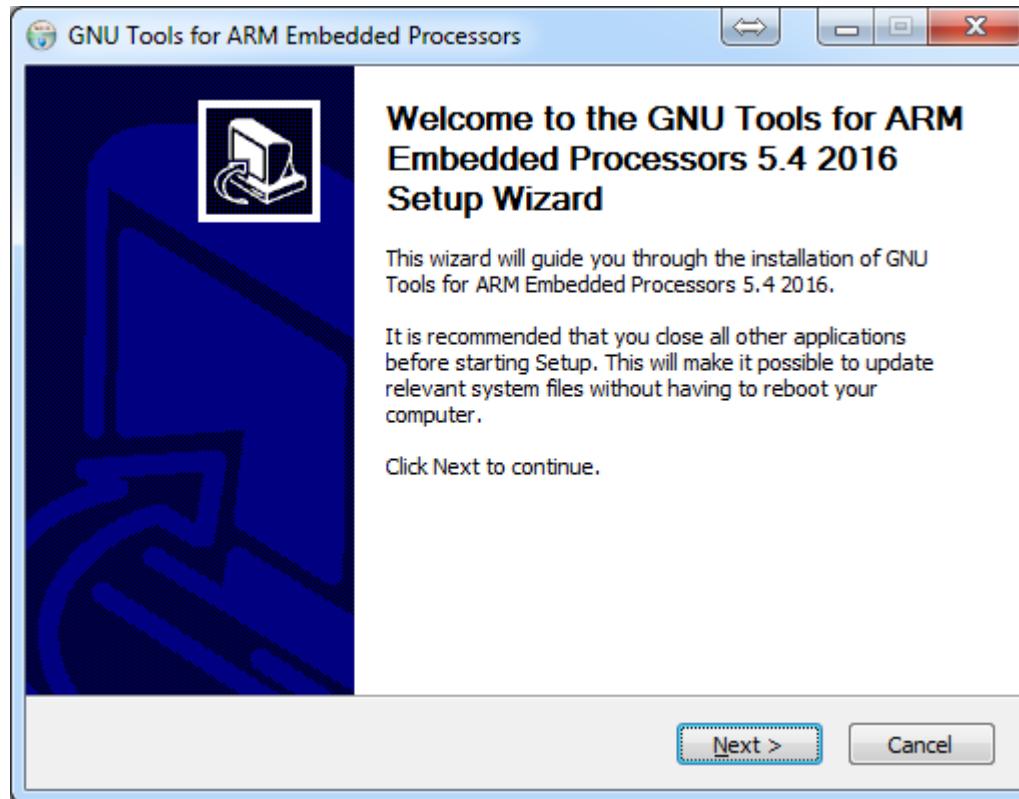
Download files for this release

After you've downloaded a file, you can verify its authenticity using its MD5 checksum. (How do I verify a download?)

File	Description	Downloads
release.txt (md5)	Release notes	3,669 last downloaded 4 days ago
gcc-arm-none-eabi-5_4-2016q2-20160622-win32.exe (md5)	Windows installer	122,601 last downloaded 24 hours ago
gcc-arm-none-eabi-5_4-2016q2-20160622-win32.zip (md5)	Windows zip package	187,654 last downloaded 24 hours ago
gcc-arm-none-eabi-5_4-2016q2-20160622-linux.tar.bz2 (md5)	Linux installation tarball	324,852 last downloaded 24 hours ago
gcc-arm-none-eabi-5_4-2016q2-20160622-mac.tar.bz2 (md5)	Mac installation tarball	84,843 last downloaded 2 days ago
gcc-arm-none-eabi-5_4-2016q2-20160622-src.tar.bz2 (md5)	Source package	55,909 last downloaded 7 days ago
How-to-build-toolchain.pdf (md5)	How to build	8,277

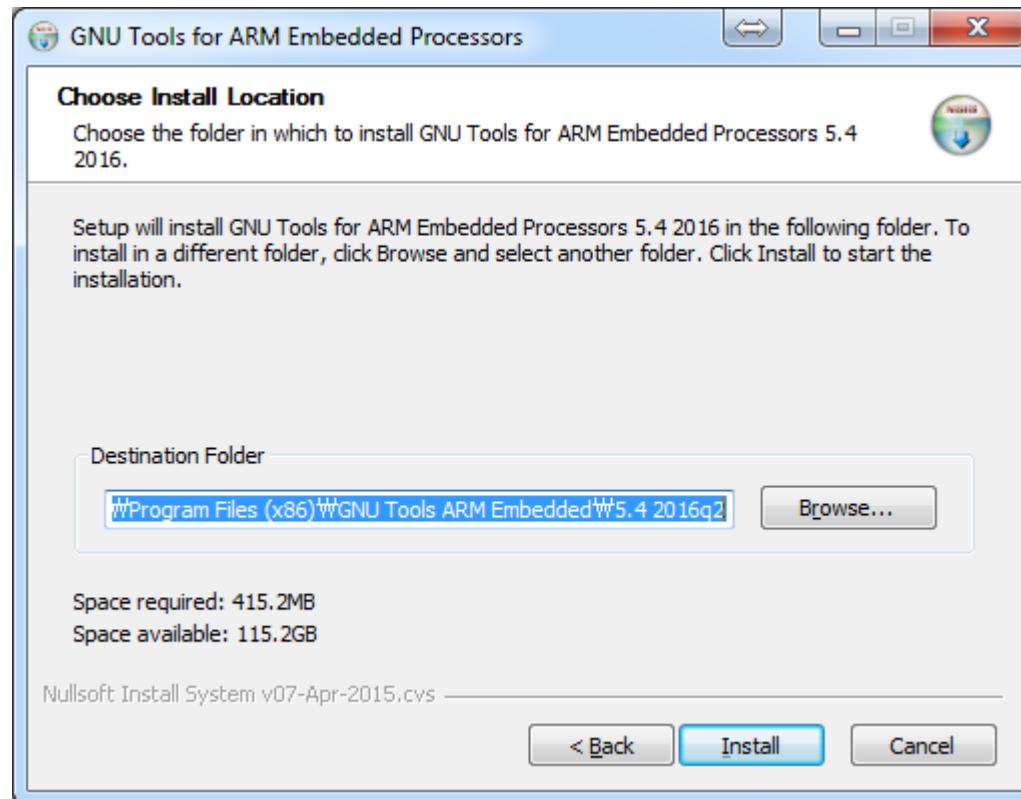
실습 #1: 개발 환경 구축 (cont'd)

>> GNU Arm Embedded Toolchain: Install (1/4)



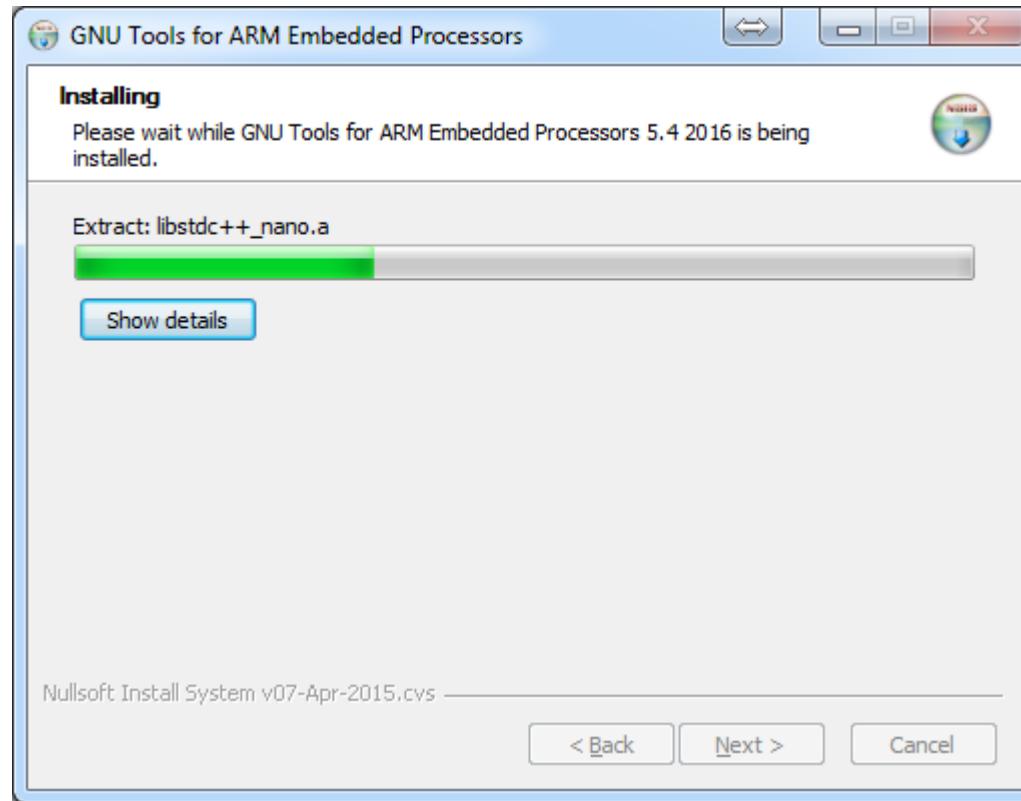
실습 #1: 개발 환경 구축 (cont'd)

>> GNU Arm Embedded Toolchain: Install (2/4)



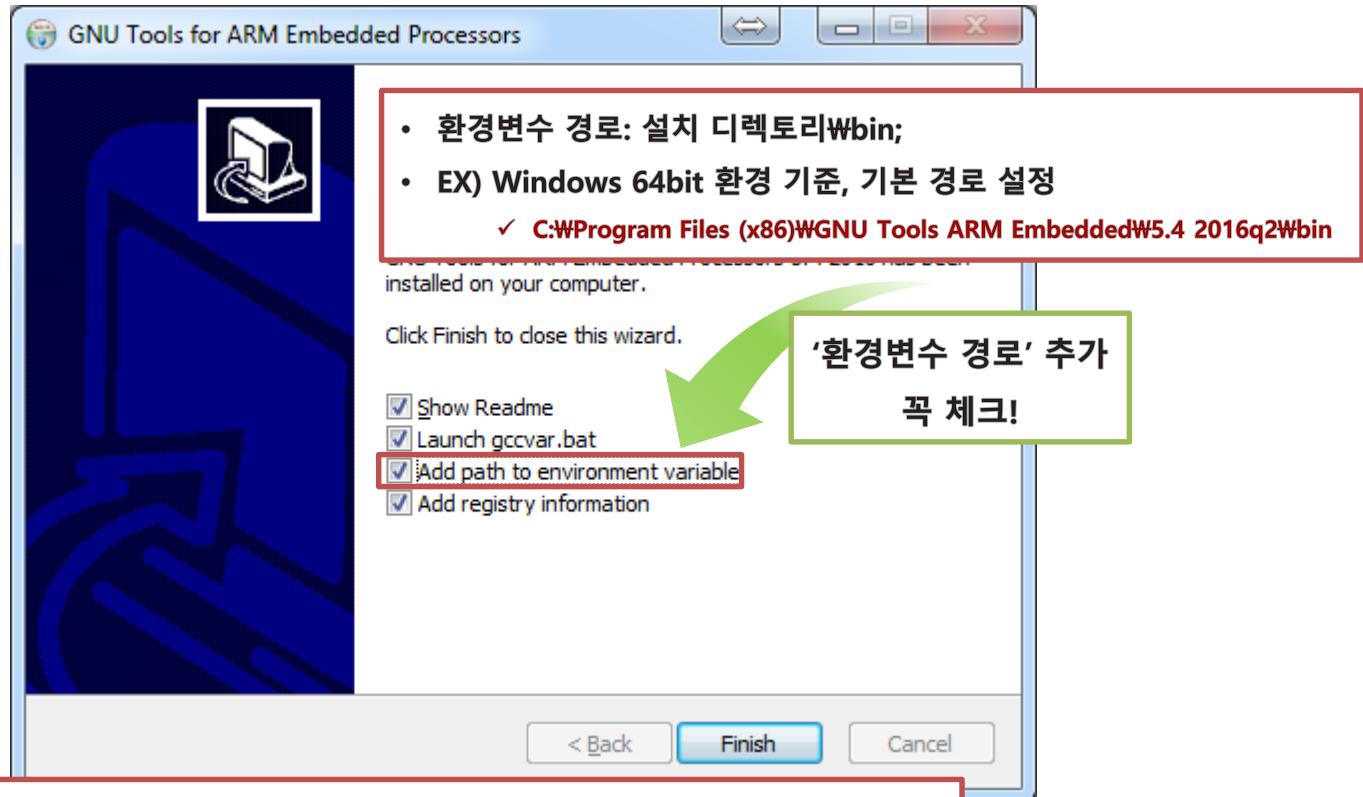
실습 #1: 개발 환경 구축 (cont'd)

>> GNU Arm Embedded Toolchain: Install (3/4)



실습 #1: 개발 환경 구축 (cont'd)

>> GNU Arm Embedded Toolchain: Install (4/4)



- 시스템의 환경변수 추가 없이 Eclipse 프로젝트에 경로 등록은 아래 링크 참고
 - ✓ Toolchain path management: <https://gnu-mcu-eclipse.github.io/toolchain/path/>
 - ✓ Arm toolchain install guide: <https://gnu-mcu-eclipse.github.io/toolchain/arm/install/>

실습 #1: 개발 환경 구축 (cont'd)

>> JAVA JRE: Download

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

The screenshot shows the Oracle Java SE Runtime Environment 8 Downloads page. On the left, there's a sidebar with links like Java SE, Java EE, Java ME, etc. The main navigation bar includes 'Overview', 'Downloads' (which is selected), 'Documentation', 'Community', 'Technologies', and 'Training'. The central content area is titled 'Java SE Runtime Environment 8 Downloads'. It asks if you want to run Java programs or develop them. If developing, it notes the JDK includes the JRE. Below this, there are two checksum links: 'JRE 8u161 Checksum' and 'JRE 8u162 Checksum'. A large green arrow points from the 'Accept License Agreement' checkbox to the 'Windows x64 Offline' download link at the bottom of the table.

Product / File Description	File Size	Download
Linux x86	63.4 MB	jre-8u161-linux-i586.rpm
Linux x86	79.29 MB	jre-8u161-linux-i586.tar.gz
Linux x64	60.4 MB	jre-8u161-linux-x64.rpm
Linux x64	76.35 MB	jre-8u161-linux-x64.tar.gz
macOS	74.17 MB	jre-8u161-macosx-x64.dmg
macOS	65.86 MB	jre-8u161-macosx-x64.tar.gz
Solaris SPARC 64-bit	52.24 MB	jre-8u161-solaris-sparcv9.tar.gz
Solaris x64	50 MB	jre-8u161-solaris-x64.tar.gz
Windows x86 Online	1.78 MB	jre-8u161-windows-i586-ifw.exe
Windows x86 Offline	61.35 MB	jre-8u161-windows-i586.exe
Windows x86	64.56 MB	jre-8u161-windows-i586.tar.gz
Windows x64 Offline	68.02 MB	jre-8u161-windows-x64.exe
Windows x64	68.58 MB	jre-8u161-windows-x64.tar.gz

실습 #1: 개발 환경 구축 (cont'd)

>> JAVA JRE: Install(1/3)



실습 #1: 개발 환경 구축 (cont'd)

>> JAVA JRE: Install(2/3)



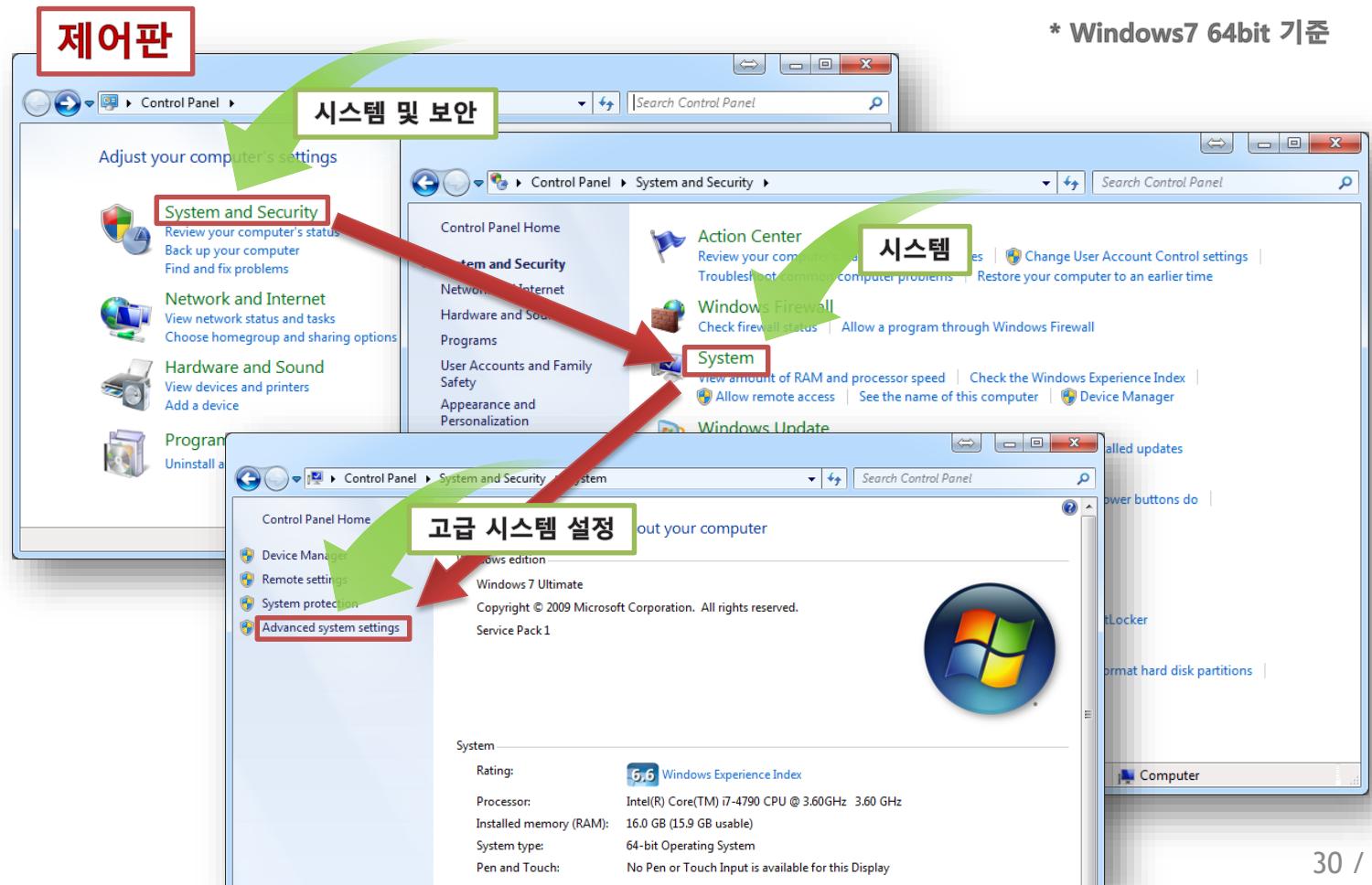
실습 #1: 개발 환경 구축 (cont'd)

>> JAVA JRE: Install(3/3)



실습 #1: 개발 환경 구축 (cont'd)

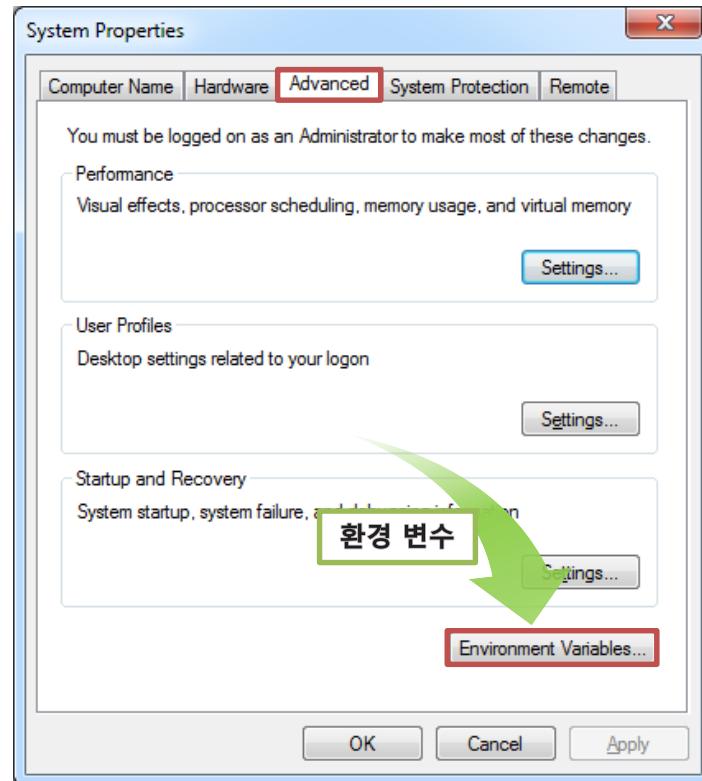
>> JAVA JRE: 환경변수 설정(1/4)



실습 #1: 개발 환경 구축 (cont'd)

>> JAVA JRE: 환경변수 설정(2/4)

* Windows7 64bit 기준



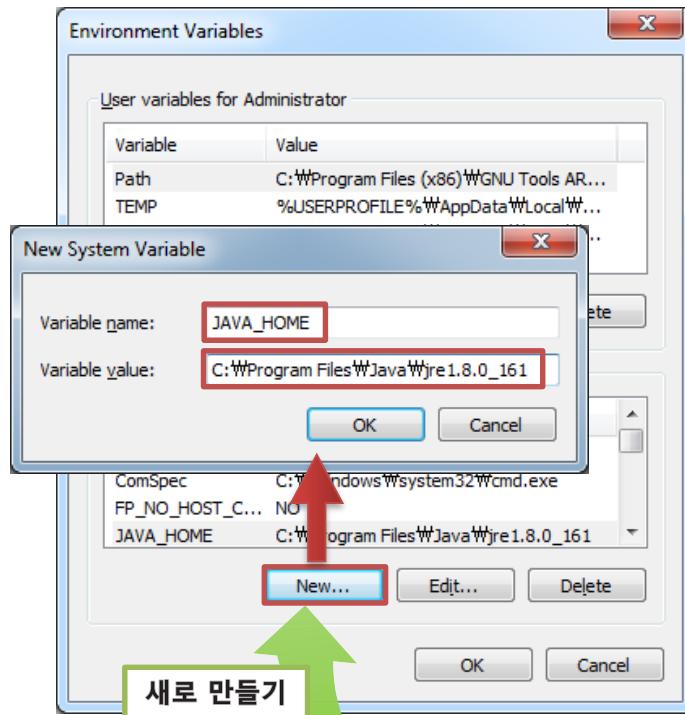
실습 #1: 개발 환경 구축 (cont'd)

>> JAVA JRE: 환경변수 설정(3/4)

* Windows7 64bit 기준

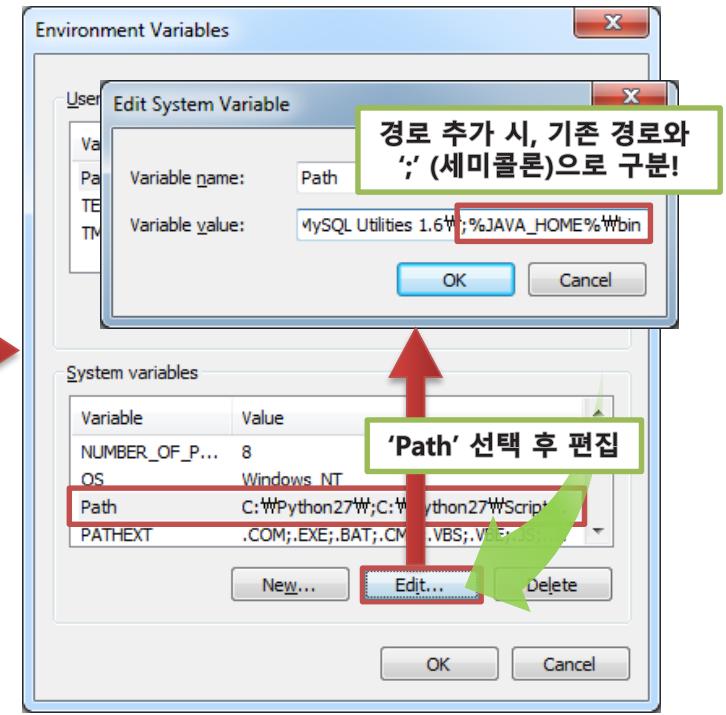
1. JAVA_HOME 변수 추가

✓ 기본 경로: C:\Program Files\Java\jre1.8.0_161



2. System Path에 JAVA_HOME의 bin 경로 추가

✓ 기본 경로: %JAVA_HOME%\bin

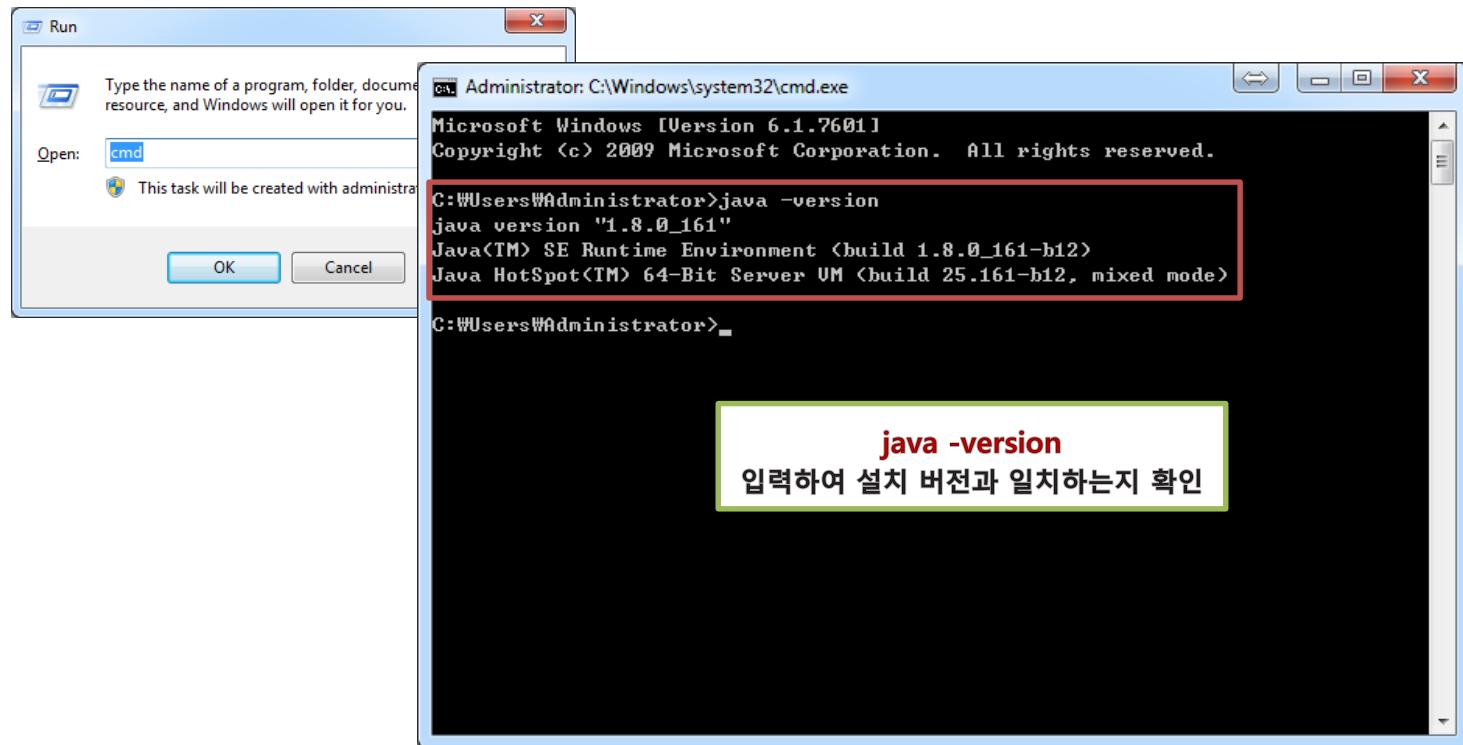


실습 #1: 개발 환경 구축 (cont'd)

>> JAVA JRE: 환경변수 설정(4/4)

* Windows7 64bit 기준

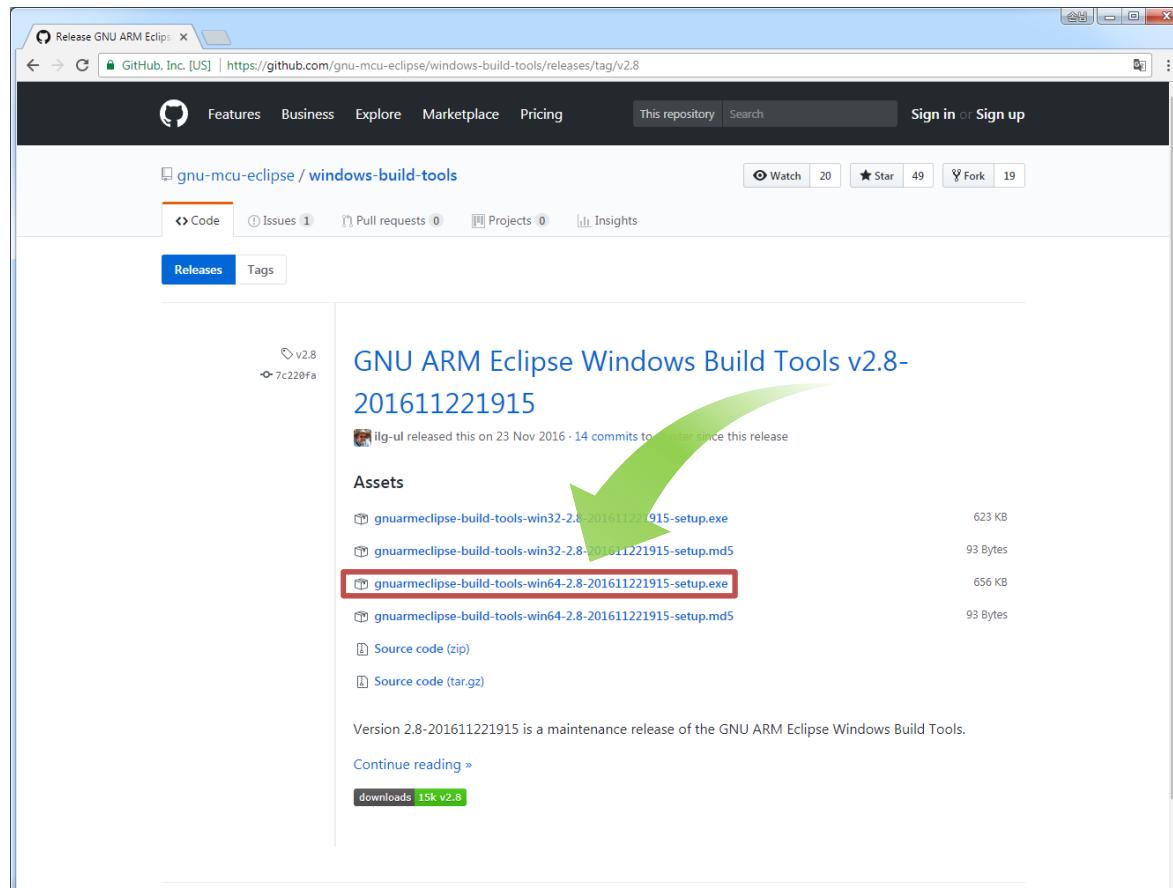
Windows key + R 키는 '실행'으로 '실행 창'을 열고,
'cmd'를 입력하여 명령 프롬프트 실행



실습 #1: 개발 환경 구축 (cont'd)

>> Windows Build Tools: Download

<https://github.com/gnu-mcu-eclipse/windows-build-tools/releases/tag/v2.8>



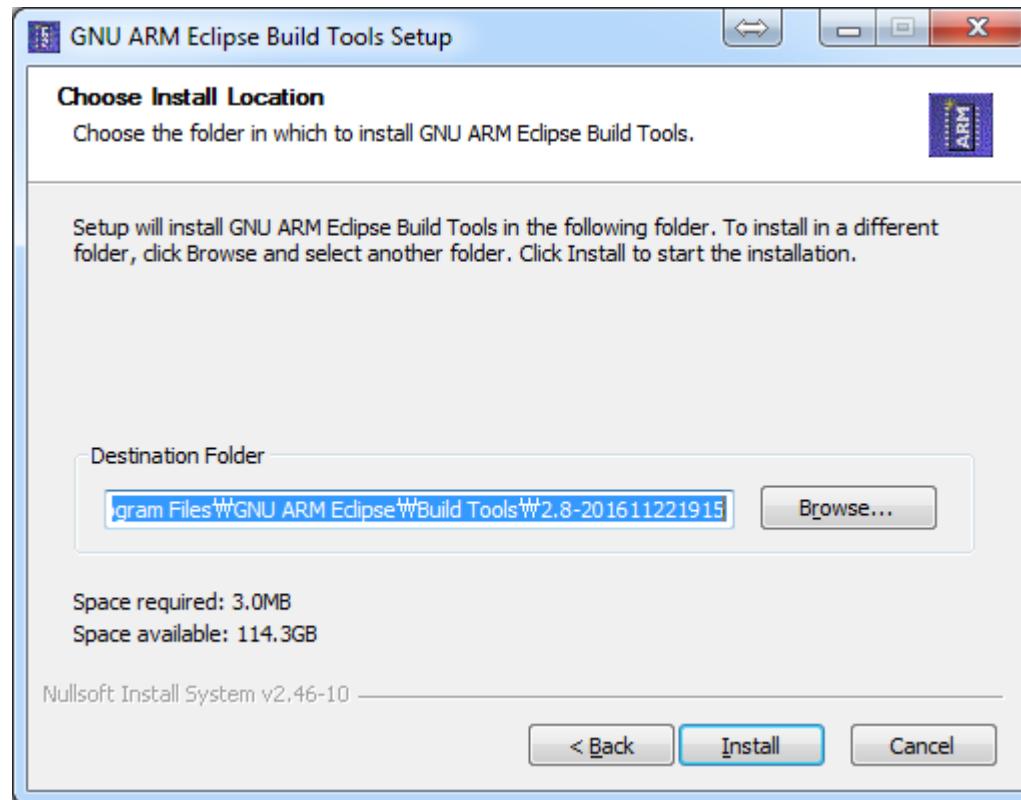
실습 #1: 개발 환경 구축 (cont'd)

>> Windows Build Tools: Install (1/3)



실습 #1: 개발 환경 구축 (cont'd)

>> Windows Build Tools: Install (1/3)

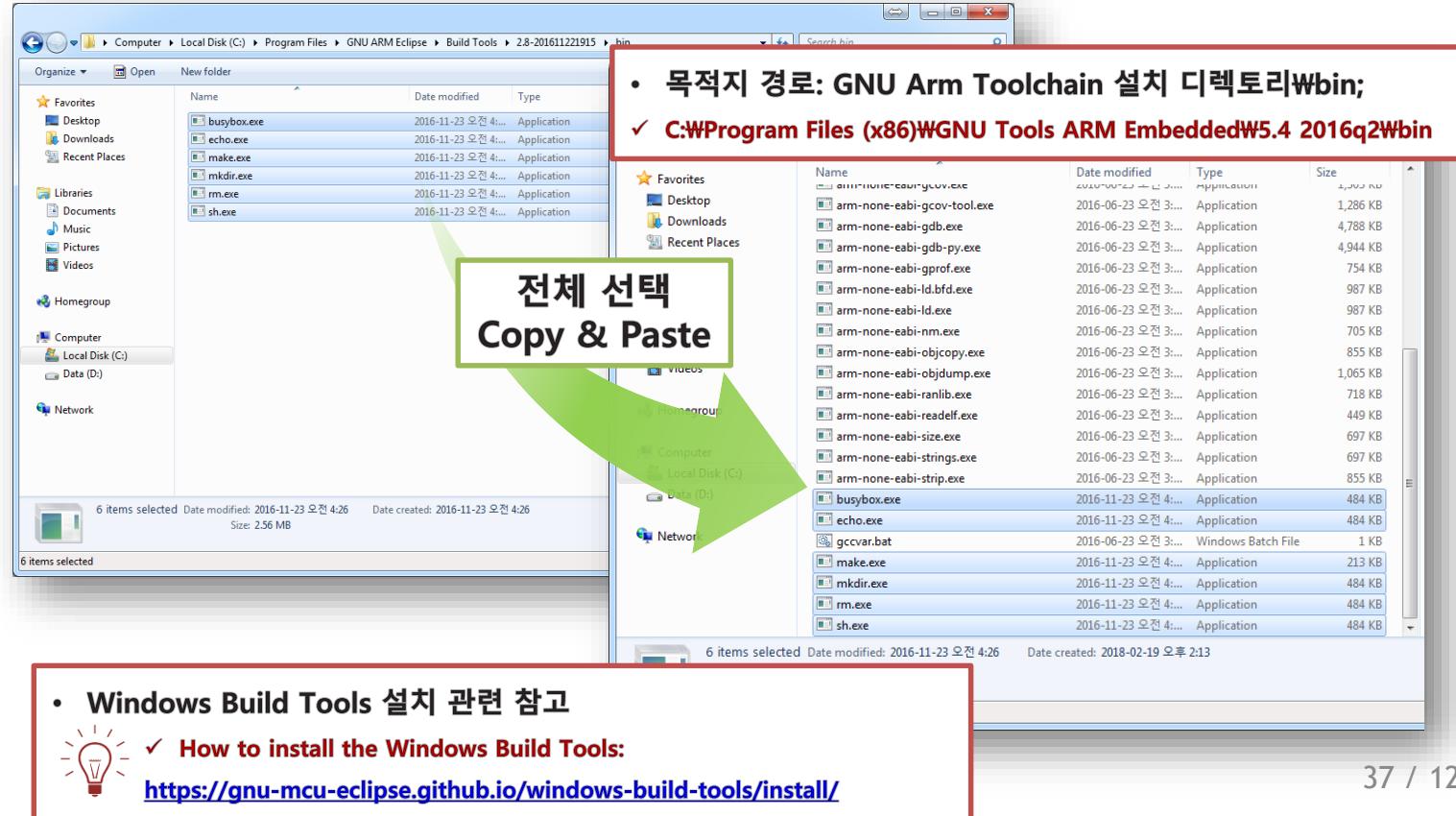


실습 #1: 개발 환경 구축 (cont'd)

>> Windows Build Tools: Install (1/3)

설치된 Window Build Tools를 GNU Arm Toolchain Wbin 폴더로 복사

✓ 기본 설치 경로: C:\Program Files\GNU ARM Eclipse\Build Tools\2.8-201611221915\bin

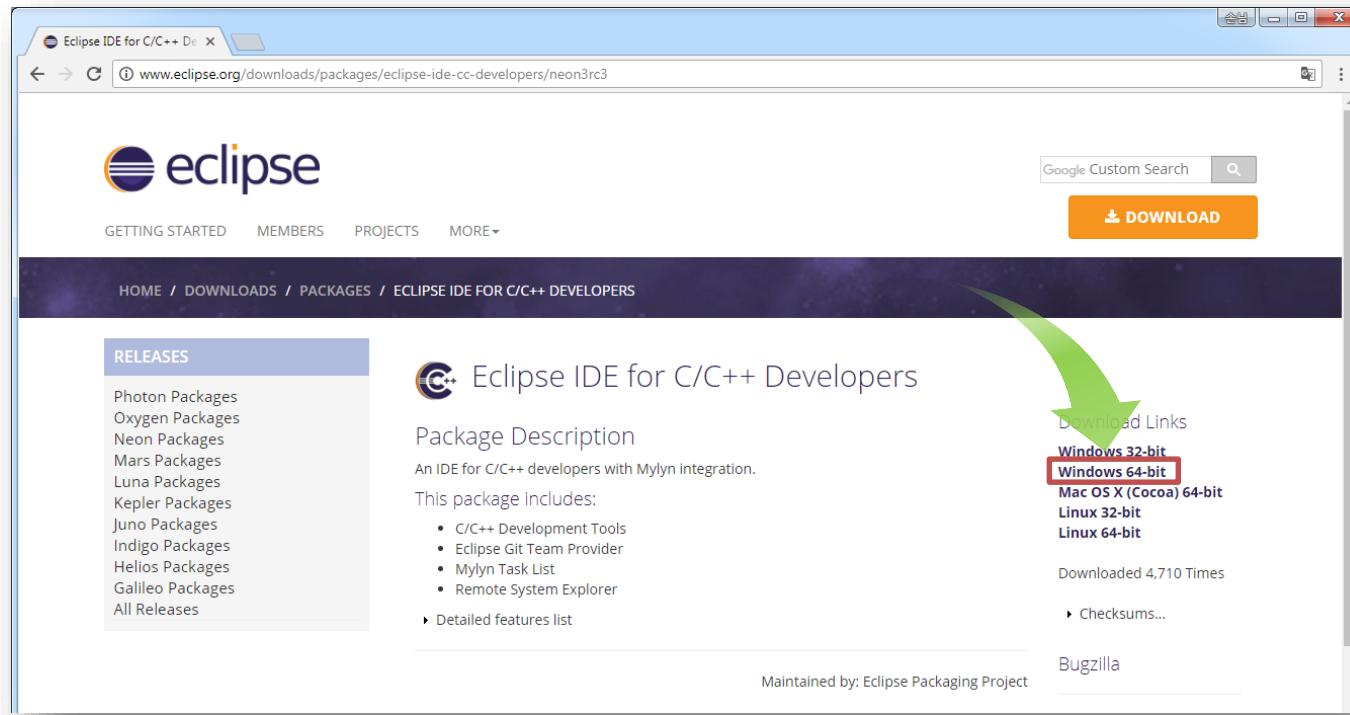


실습 #1: 개발 환경 구축 (cont'd)

>> Eclipse CDT: Download

- CDT: C/C++ Development Tooling

<http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/neon3rc3>



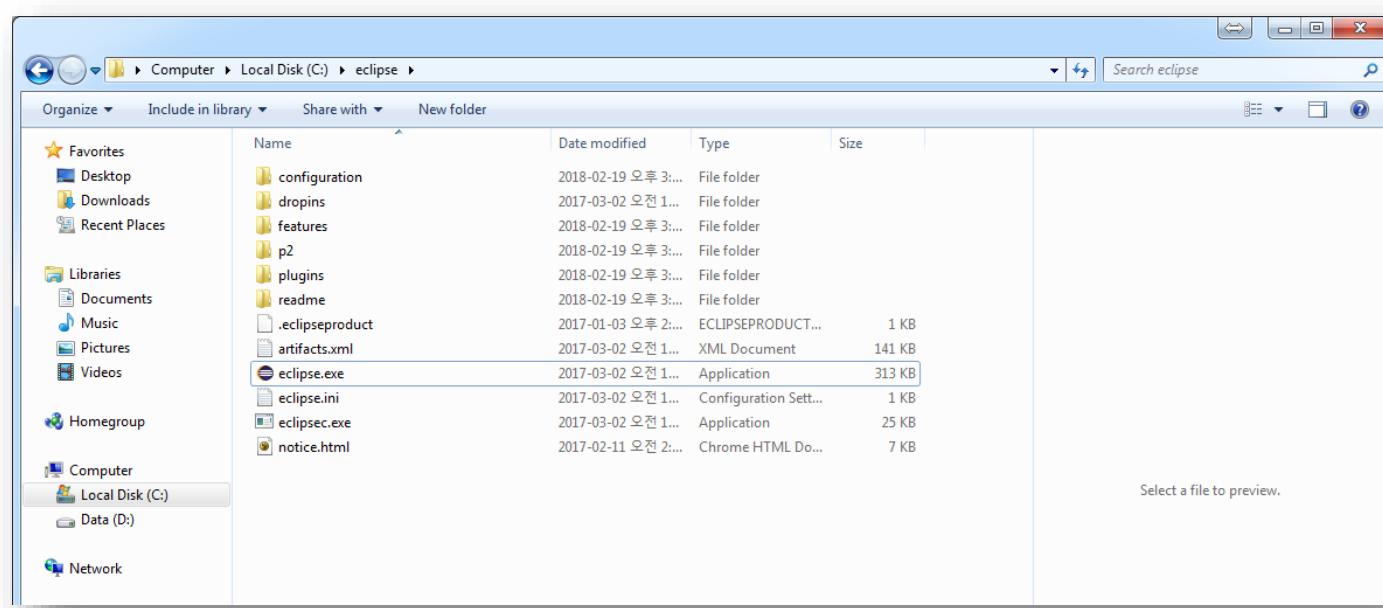
실습 #1: 개발 환경 구축 (cont'd)

>> Eclipse CDT: Install (1/2)

Download 한 Eclipse CDT를 원하는 경로에 압축 해제,

단 File path에 한글이 포함된 경로가 없도록 설치 (workspace directory도 마찬가지)

- ✓ Eclipse 예시 설치 경로: **C:\Weclipse**
- ✓ Workspace 예시 경로: **C:\Workspace**

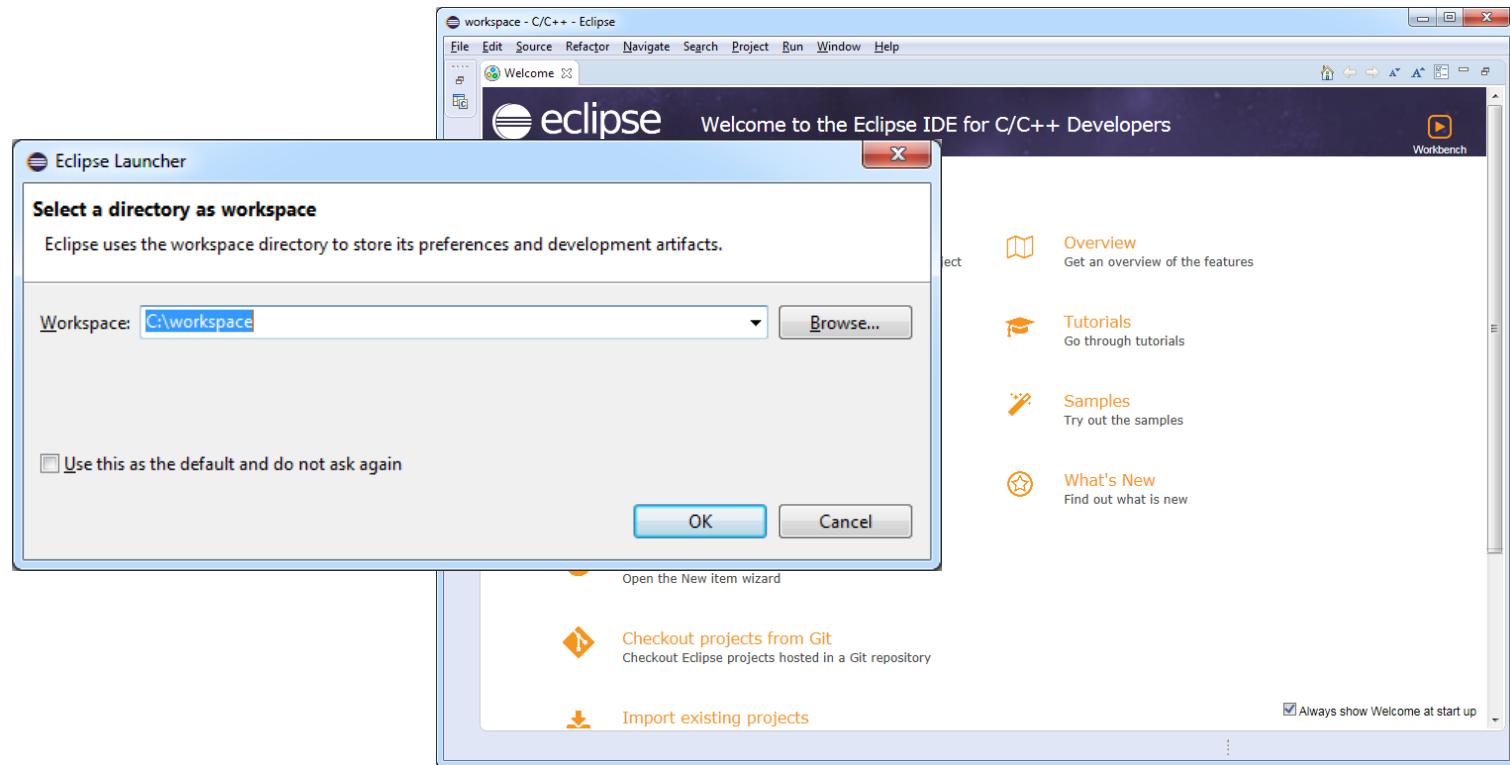


실습 #1: 개발 환경 구축 (cont'd)

>> Eclipse CDT: Install (2/2)

Eclipse 실행

✓ Workspace 예시 경로: **C:\workspace**

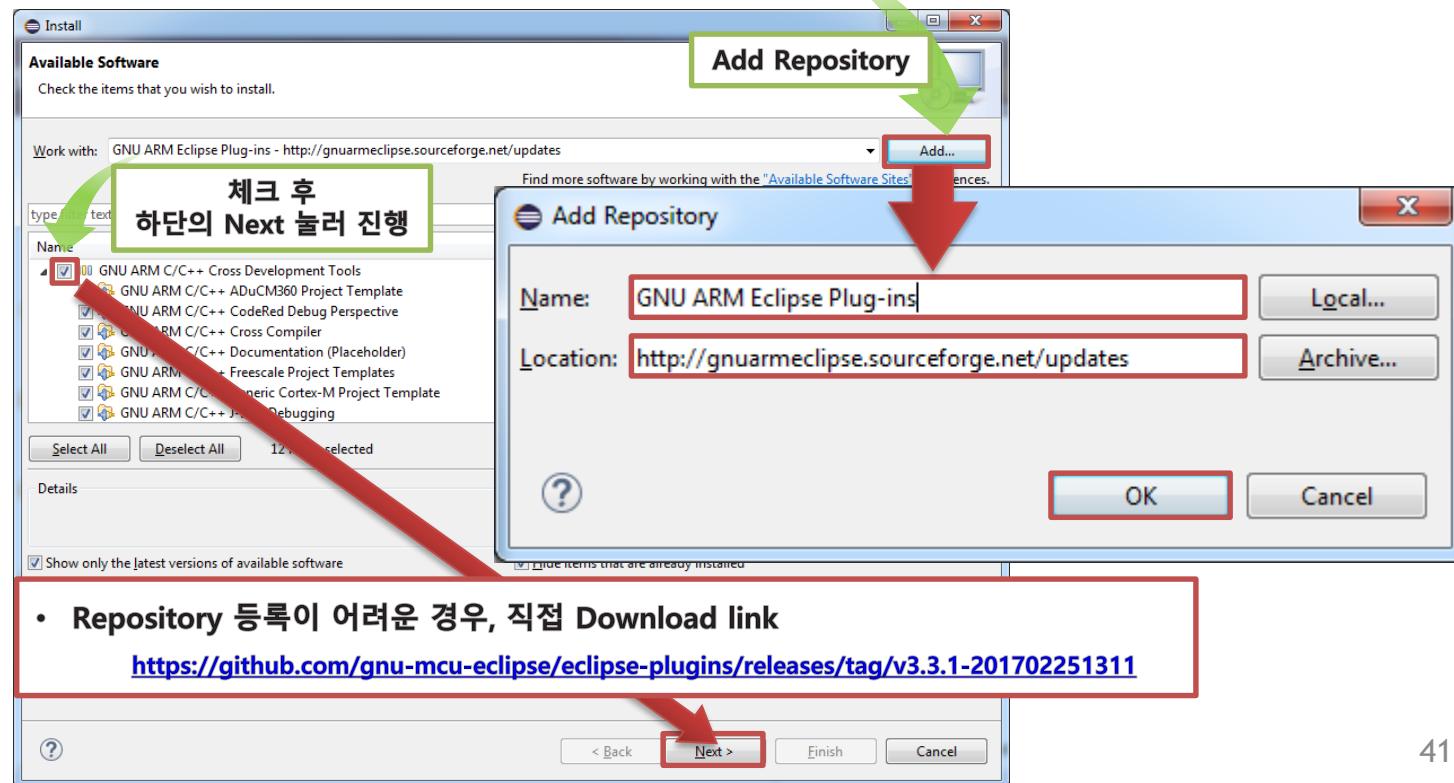


실습 #1: 개발 환경 구축 (cont'd)

>> GNU ARM Eclipse Plugin: Download & Install (1/2)

Eclipse 상단 메뉴의 [Help] – [Install New Software] 선택 후 팝업 창의 [Add...] 선택하여 Add Repo.

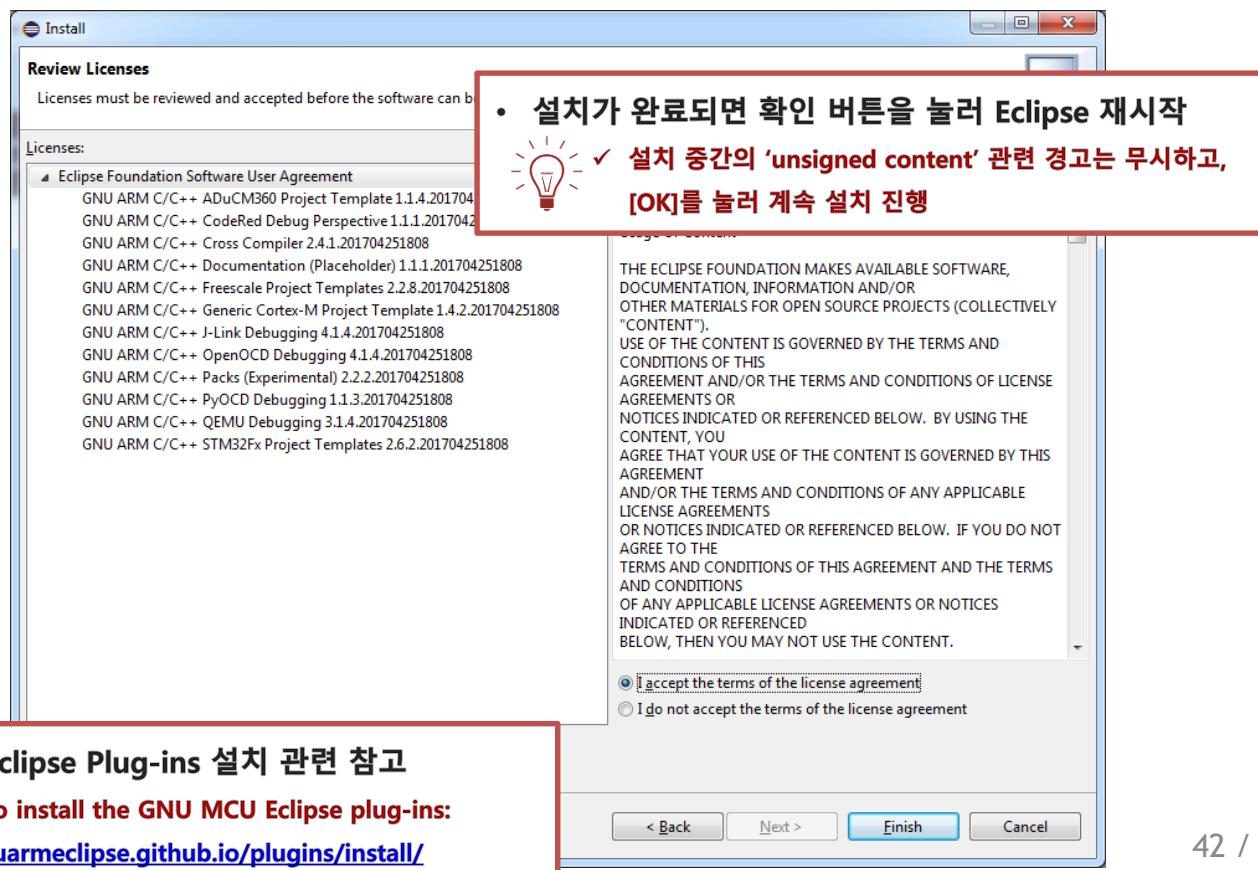
- ✓ Name: **GNU ARM Eclipse Plug-ins**
- ✓ Location: **http://gnuarmeclipse.sourceforge.net/updates**



실습 #1: 개발 환경 구축 (cont'd)

>> GNU ARM Eclipse Plugin: Download & Install (2/2)

Accept the License Agreement

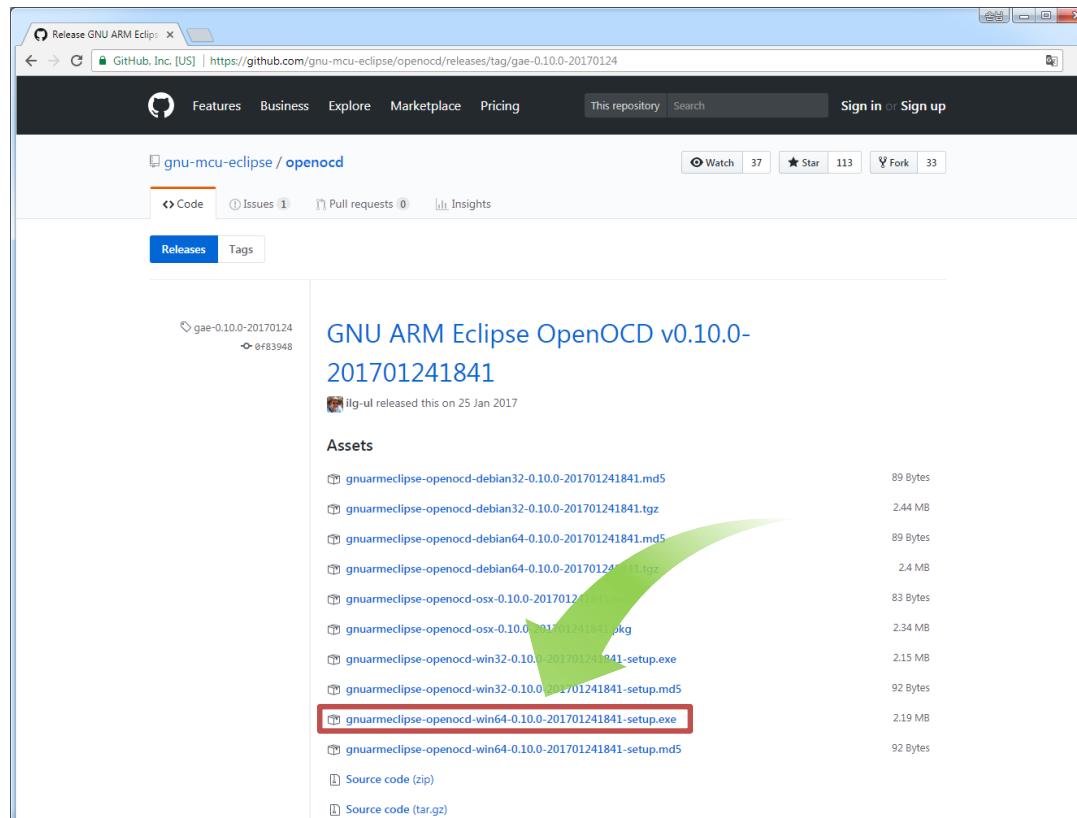


실습 #1: 개발 환경 구축 (cont'd)

>> GNU ARM Eclipse OpenOCD: Download (Option)

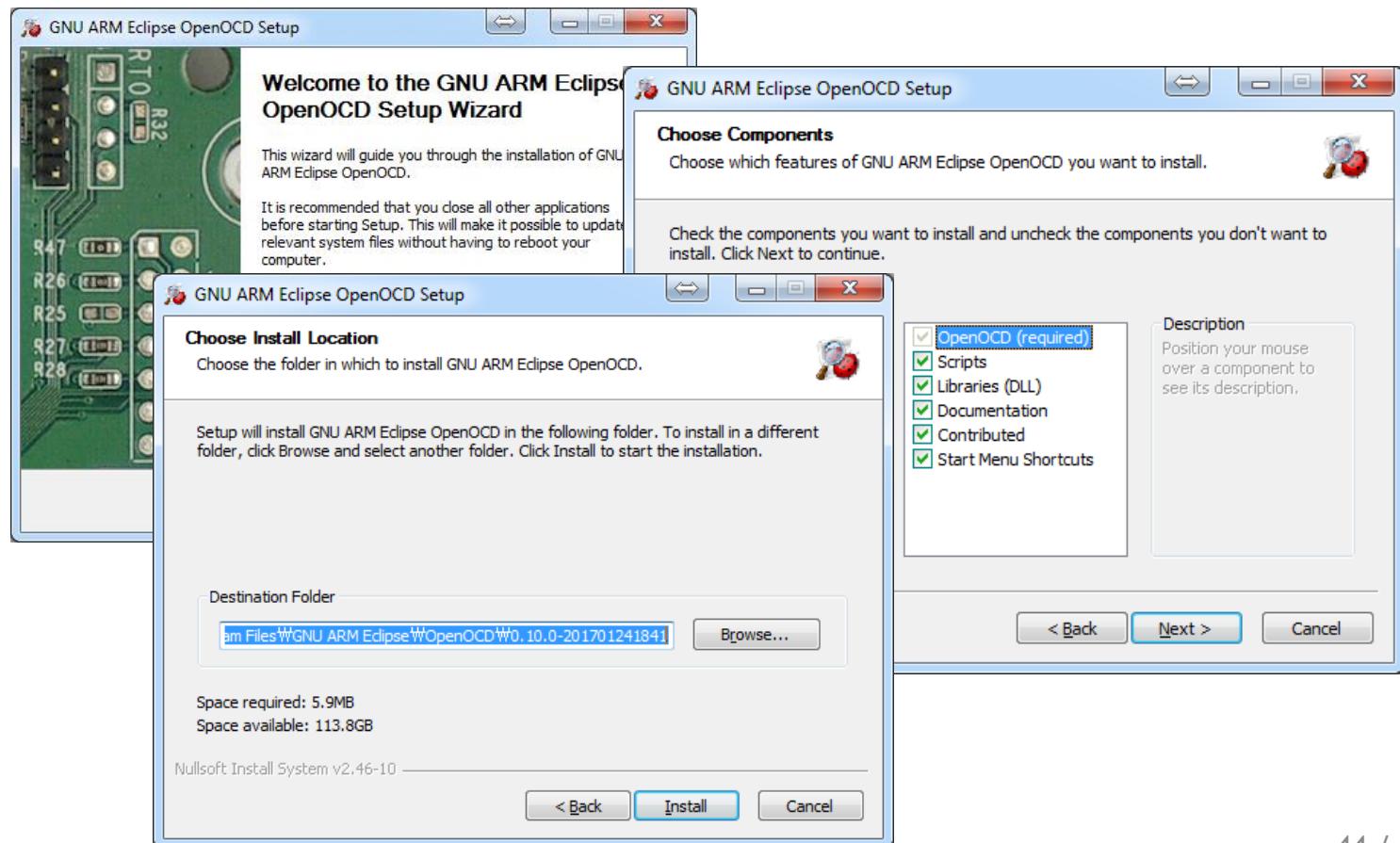
- Open source debugging 도구(Open On-Chip Debugger)

<https://github.com/gnu-mcu-eclipse/openocd/releases/tag/gae-0.10.0-20170124>



실습 #1: 개발 환경 구축 (cont'd)

>> GNU ARM Eclipse OpenOCD: Install (1/2)

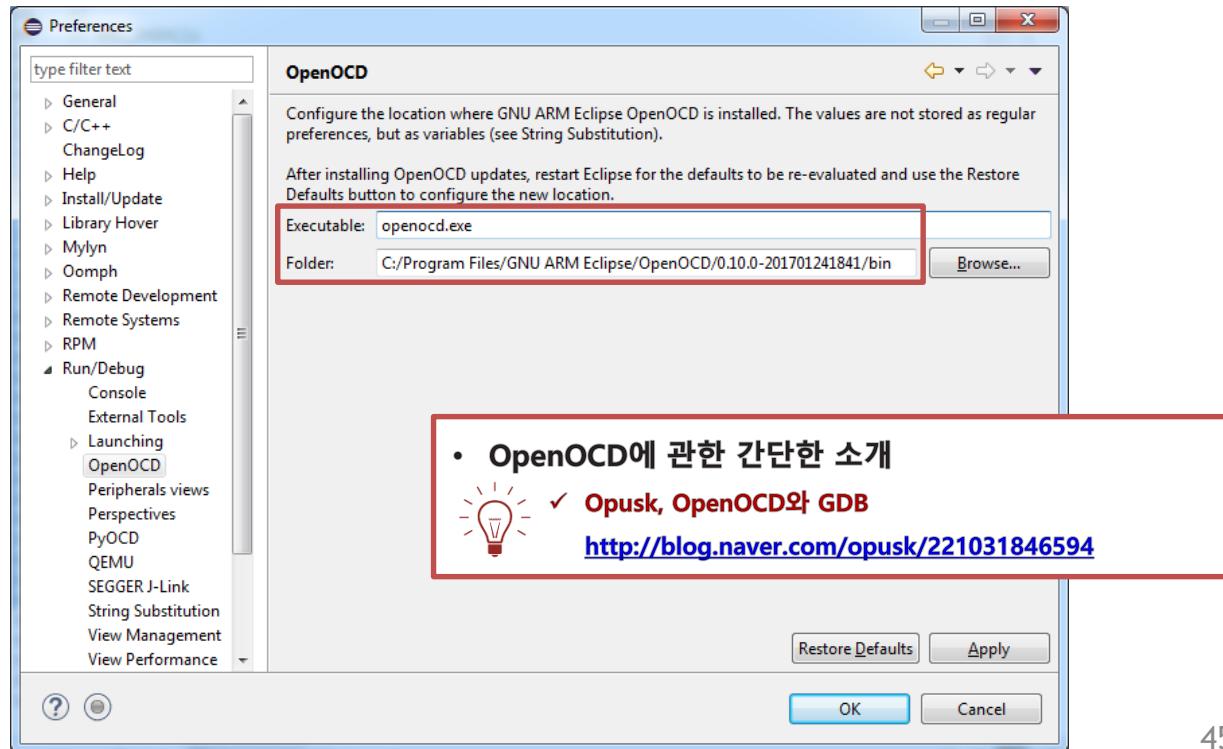


실습 #1: 개발 환경 구축 (cont'd)

>> GNU ARM Eclipse OpenOCD: Install (2/2)

Eclipse 상단 메뉴의 [Window] – [Preferences] 선택 후, 팝업 창의 [Run/Debug] 항목의 OpenOCD 경로 확인

- ✓ Executable: **openocd.exe**
- ✓ Folder: **C:/Program Files/GNU ARM Eclipse/OpenOCD/0.10.0-201701241841/bin**

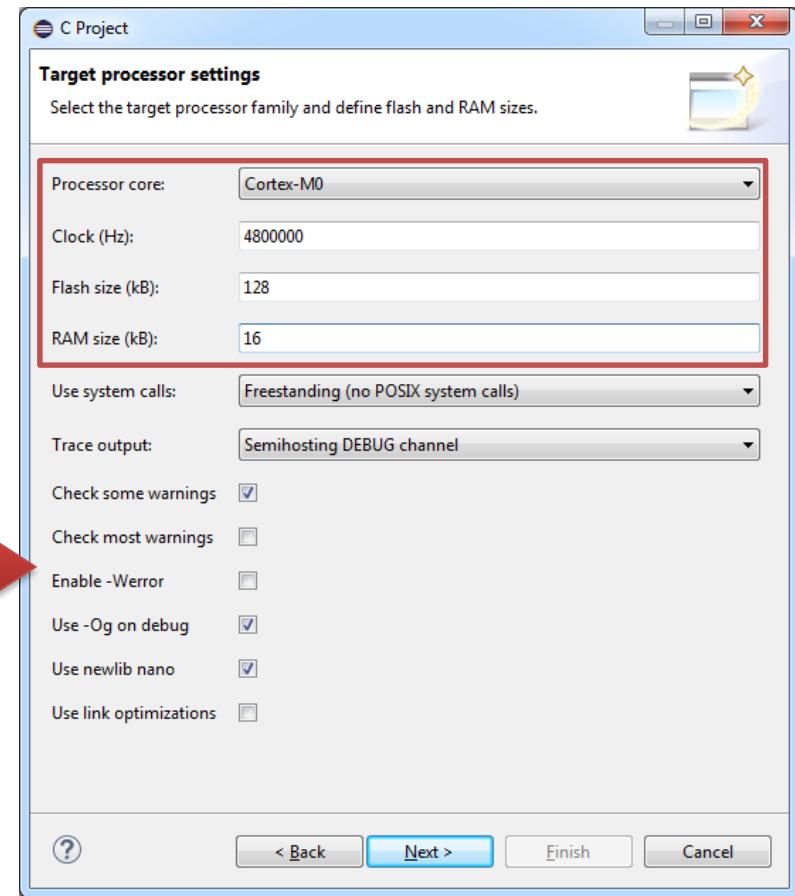
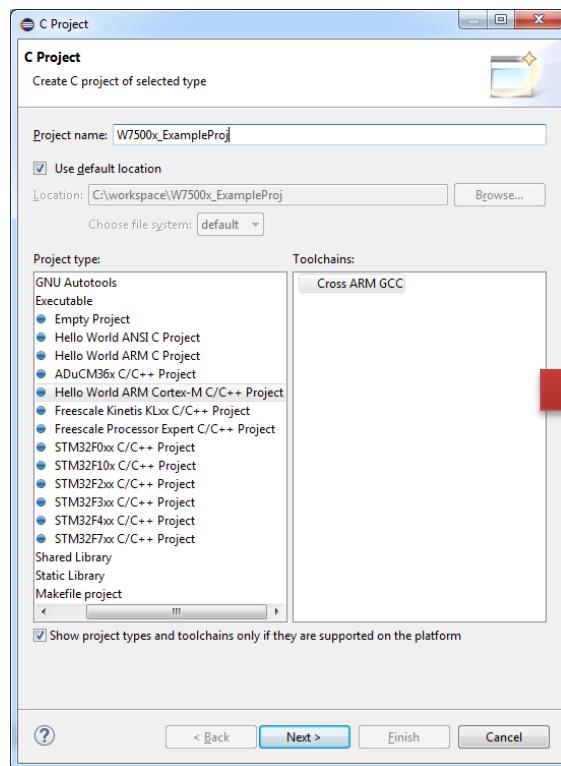


실습 #1: 개발 환경 구축 – 동작 확인

>> Eclipse CDT: W7500x Hello world (1/4)

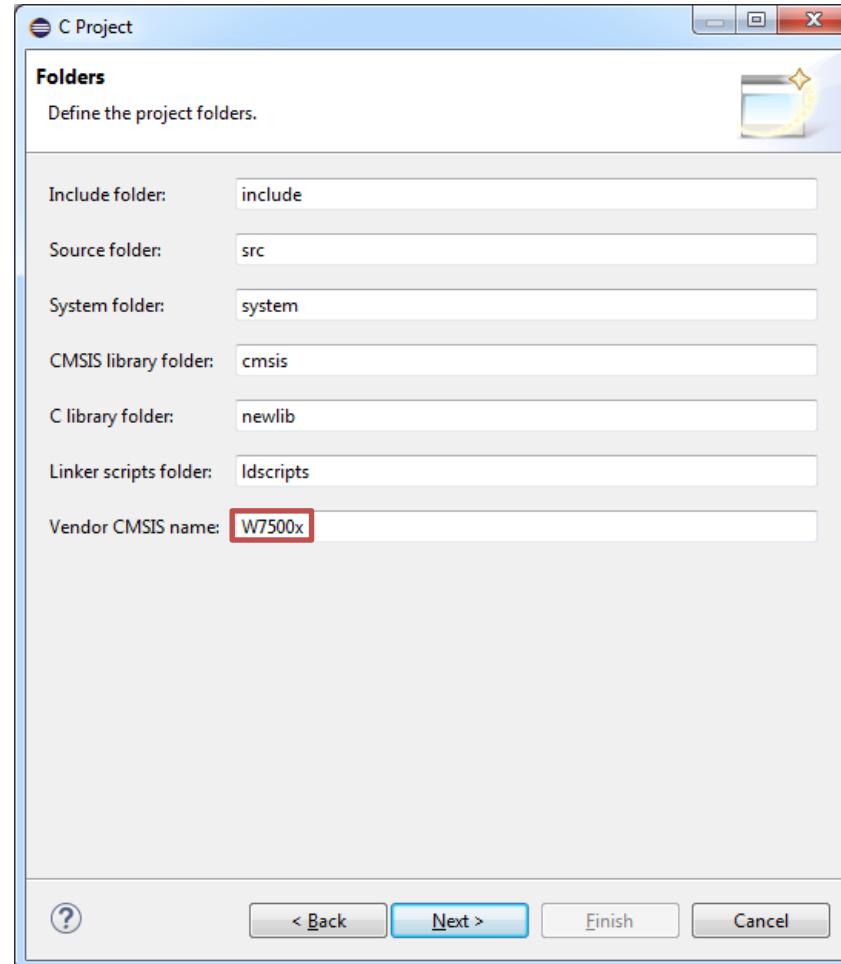
Eclipse 상단 메뉴의 [File] – [New]의 [C Project] 선택

- ✓ Executable의 Hello World ARM Cortex-M C/C++ Project
- ✓ Example Project Name: **W7500x_ExampleProj**



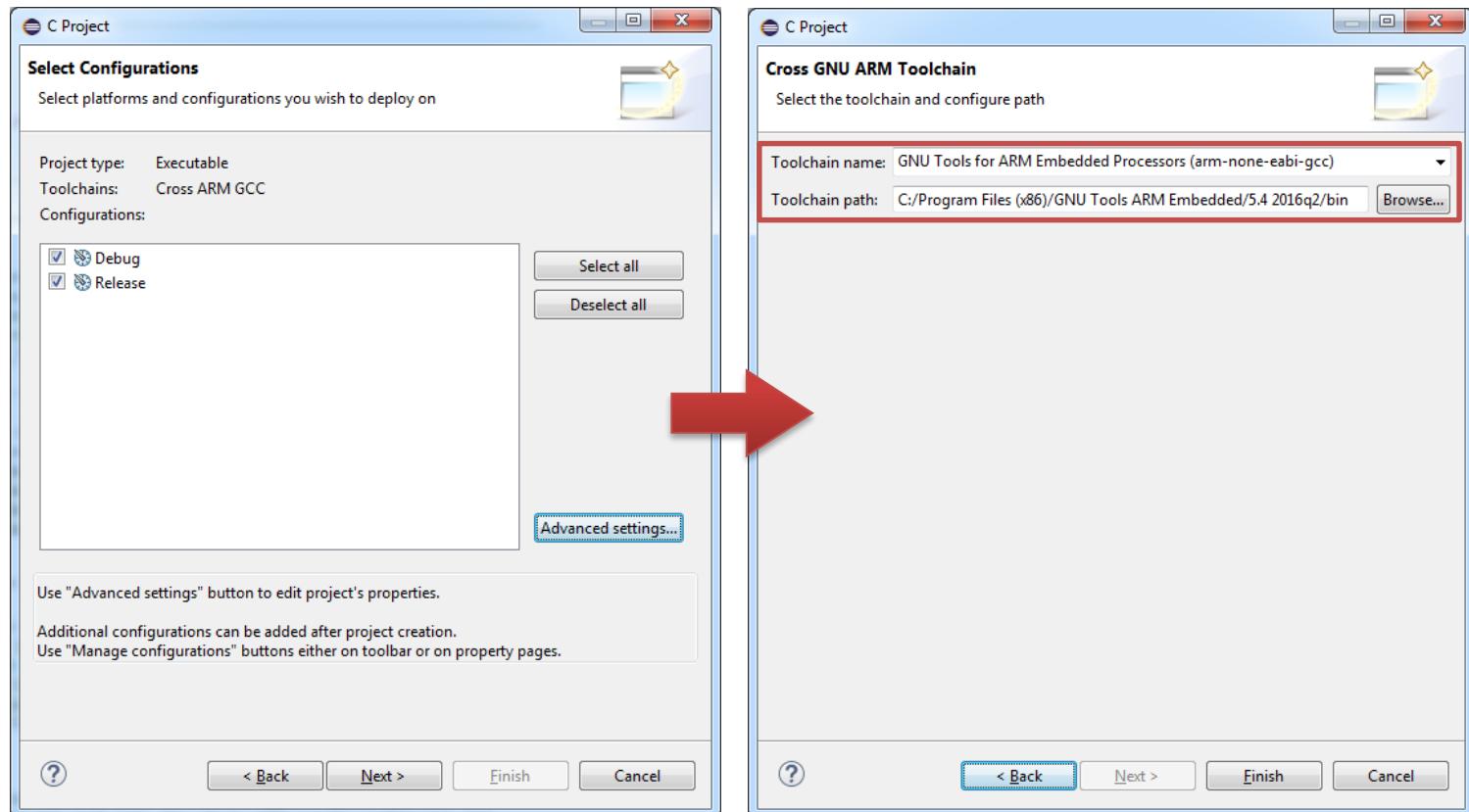
실습 #1: 개발 환경 구축 – 동작 확인 (cont'd)

>> Eclipse CDT: W7500x Hello world (2/4)



실습 #1: 개발 환경 구축 – 동작 확인 (cont'd)

>> Eclipse CDT: W7500x Hello world (3/4)



실습 #1: 개발 환경 구축 – 동작 확인 (cont'd)

>> Eclipse CDT: W7500x Hello world (4/4)

• 프로젝트 Build 후, 하단 Console의 Build Finished 확인

- ✓ 상단 [Project] 메뉴의 'Build All' 클릭
- ✓ 혹은 Ctrl + B로 프로젝트 Build 수행

The screenshot shows the Eclipse CDT interface. The code editor displays a C program for a W7500x microcontroller. The build console at the bottom shows the output of the build process, indicating a successful build.

• W7500x Eclipse 신규 프로젝트 생성은 다음 Blog 참고

✓ Opusk, [Eclipse] W7500x 프로젝트 생성하기
<https://blog.naver.com/opusk/221007777456>

16:48:11 Build Finished (took 3s.710ms)



실습 #2

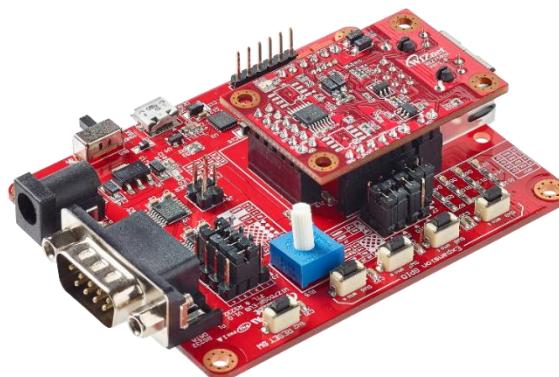
WIZ750SR 시작하기



실습 #2: WIZ750SR 시작하기

>> 실습 주제

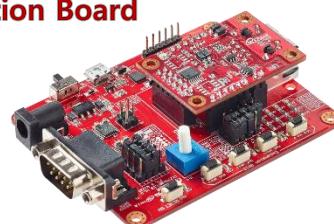
- WIZ750SR 하드웨어 구성 확인 및 테스트 프로그램 준비
- 제품 동작 테스트
 - 네트워크 환경 설정
 - 시리얼 포트 환경 설정
 - Configuration tool을 이용한 제품 검색 및 설정
 - Serial to Ethernet 동작 테스트
 - 문제 해결 가이드



실습 #2: WIZ750SR 시작하기 (cont'd)

» 테스트 환경 구축: 하드웨어 구성

**WIZ750SR-TTL
Evaluation Board**



PC or Laptop



직접 연결

**Data UART
Serial Data 송/수신**



Serial Cable

Female to female
DB9 data cable

RS-232 to USB 변환



**USB to RS232
Converter**
FTDI CHIP-X10

**Debug UART
전원 공급**



USB Cable

Micro USB Type-B

Ethernet 연결



LAN Cable

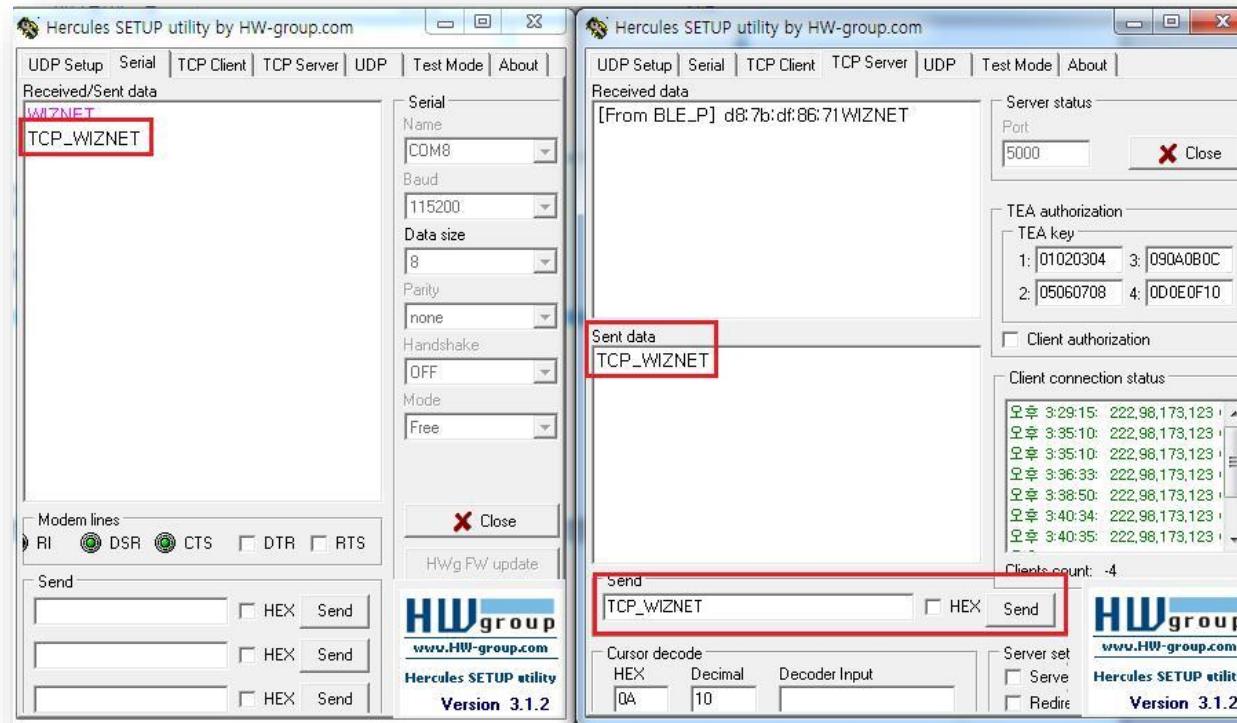
Cat5e UTP Cable

실습 #2: WIZ750SR 시작하기 (cont'd)

>> 테스트 환경 구축: Terminal 프로그램 설치

- Hercules SETUP utility: TCP Client / TCP Server / UDP / Serial Port Terminal

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

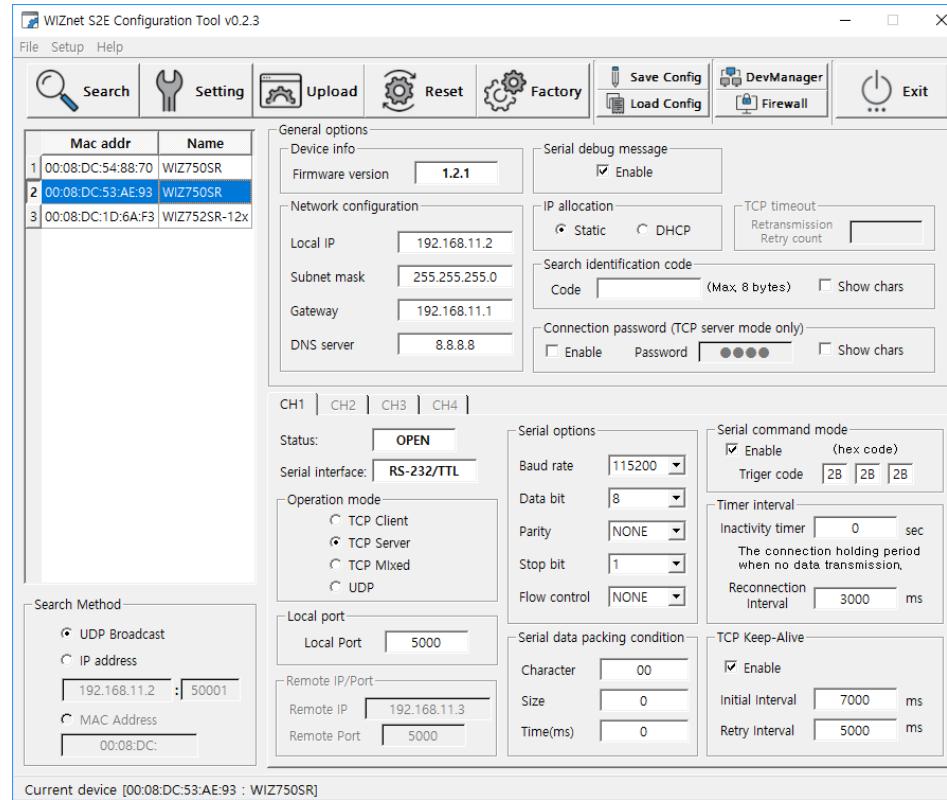


실습 #2: WIZ750SR 시작하기 (cont'd)

>> WIZnet Configuration Tool 설치

- WIZ750SR 및 WIZ752SR Series 설정용 프로그램

<https://github.com/Wiznet/WIZnet-S2E-Tool-GUI>



실습 #2: WIZ750SR 시작하기 (cont'd)

>> WIZ750SR 공장 초기 설정

Network Settings	Local	IP address	192.168.11.2	-
	Gateway address	192.168.11.1	-	-
	Subnet mask	255.255.255.0	-	-
	DNS server	8.8.8.8	Google Public DNS	-
	Port number	5000	-	-
Remote	IP address	192.168.11.3	-	-
	Port number	5000	-	-

Serial Port Settings	Data UART	115200-8-N-1 / Flow Control: None	-
	Debug UART	115200-8-N-1 / Flow Control: None	Fixed

User's I/O Settings	UserIO A	Analog / Input	Read only
	UserIO B	Digital / Input	Read only
	UserIO C	Digital / Output	Read / Write
	UserIO D	Digital / Output	Read / Write

- Operation mode: **TCP server mode**
- Debug message: **Enabled**
- Serial command mode switch: **Enabled**
- Serial command mode switch code: **+++** (hex code, [2B][2B][2B])
- Data packing option - Time: **Disabled**
- Data packing option - Size: **Disabled**
- Data packing option - Char: **Disabled**
- Inactivity Timer: **Disabled**
- Reconnection Timer: **3 second**
- Keep-Alive: **Enabled, 7-sec initial delay, 5-sec send interval**

실습 #2: WIZ750SR 시작하기 (cont'd)

>> TCP/IP 네트워크 설정 (1/4)

WIZ750SR Network Settings



Network Settings	Local	IP address	192.168.11.2	-
		Gateway address	192.168.11.1	-
		Subnet mask	255.255.255.0	-
		DNS server	8.8.8.8	Google Public DNS
		Port number	5000	-
	Remote	IP address	192.168.11.3	-
		Port number	5000	-

Ex) PC Network Settings

Network Settings	PC or laptop (= Remote)	IP address	192.168.11.3	-
		Gateway address	192.168.11.1	-
		Subnet mask	255.255.255.0	-
		Port number	5000	-

* 동일 네트워크 상의 네트워크 장치가 서로 통신하려면 동일한 IP 서브넷의 IP 주소를 가져야 함

- [Microsoft] TCP/IP 주소 지정 및 서브넷 구성 기본 사항의 이해

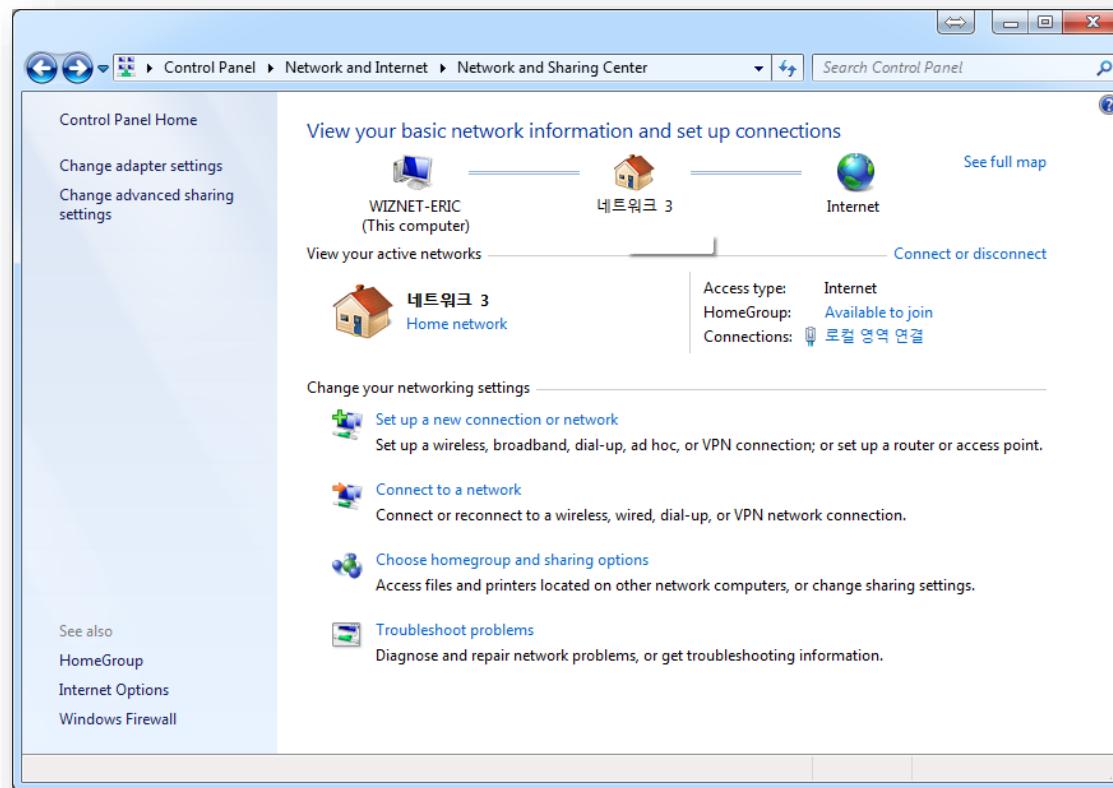
<https://support.microsoft.com/ko-kr/help/164015/understanding-tcp-ip-addressing-and-subnetting-basics>

실습 #2: WIZ750SR 시작하기 (cont'd)

>> TCP/IP 네트워크 설정 (2/4)

1. [제어판] – [네트워크 및 인터넷] – [네트워크 및 공유센터] 선택
2. [어댑터 설정 변경]의 Ethernet 네트워크 어댑터 선택 (**로컬 영역 연결**)

* Windows7 64bit 기준



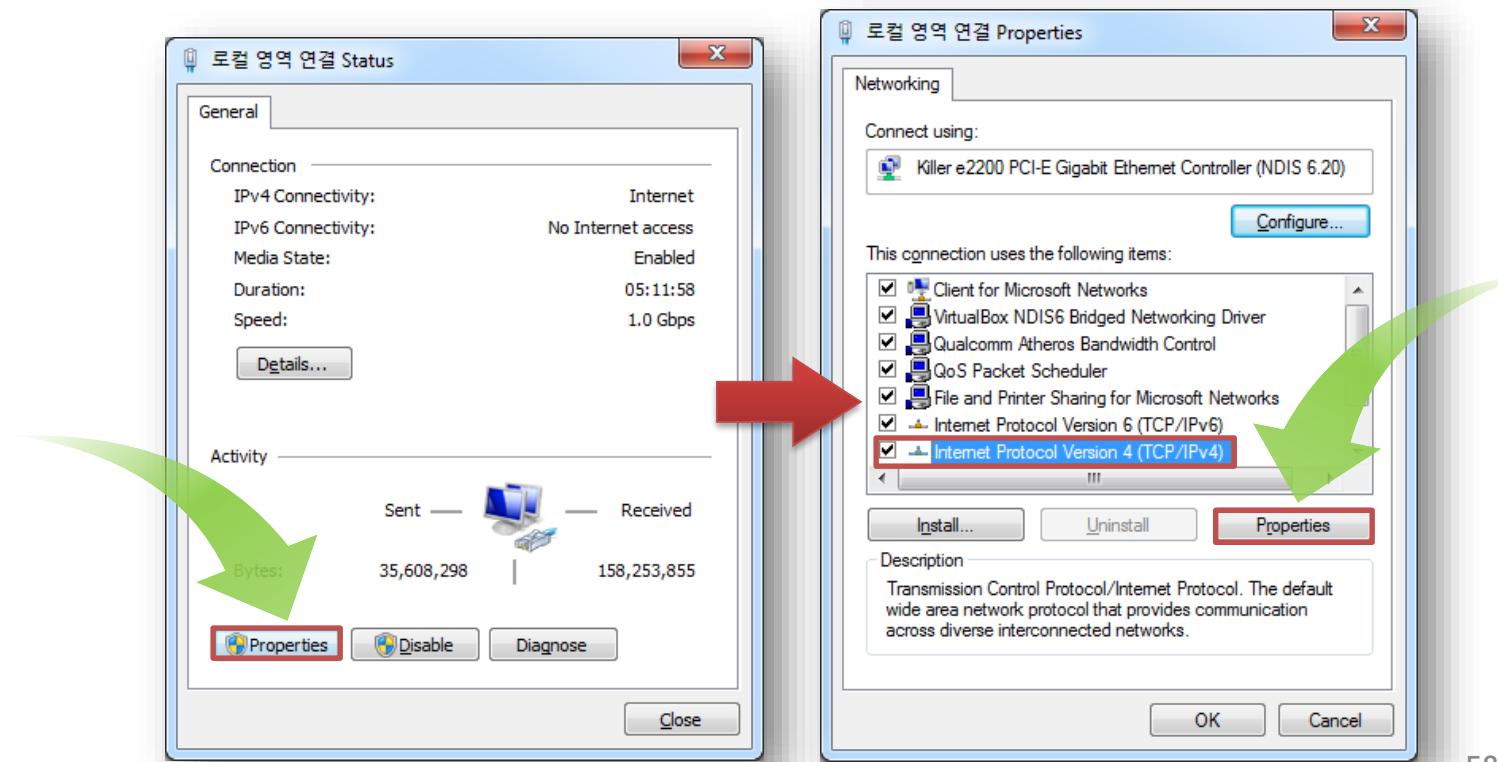
실습 #2: WIZ750SR 시작하기 (cont'd)

>> TCP/IP 네트워크 설정 (3/4)

1. [로컬 영역 연결] – [속성] 선택

* Windows7 64bit 기준

2. [로컬 영역 연결 속성]의 Internet Protocol Version 4 (TCP/IPv4) 선택 후 [속성] 선택

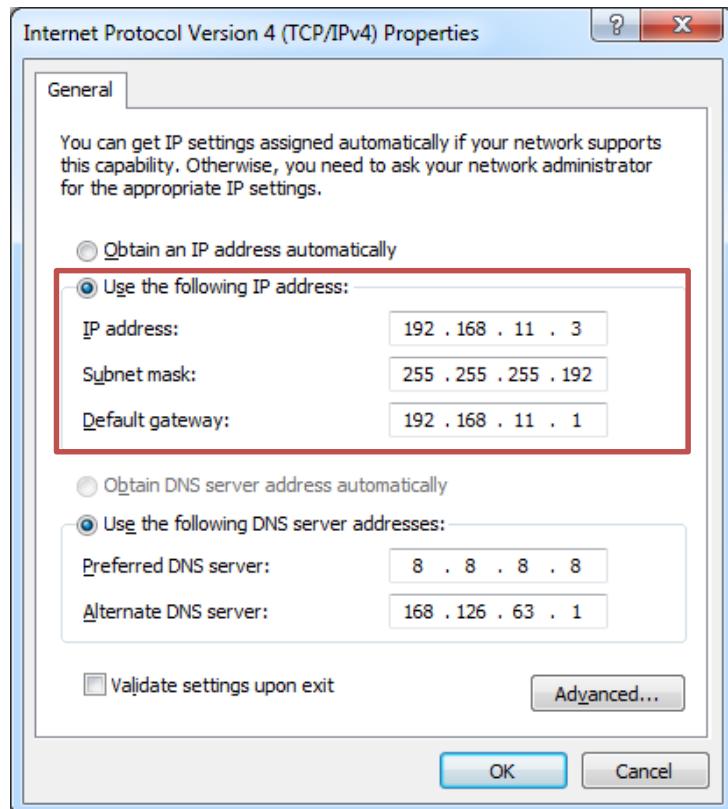


실습 #2: WIZ750SR 시작하기 (cont'd)

>> TCP/IP 네트워크 설정 (4/4)

- IP 주소 등 네트워크 정보 입력

* Windows7 64bit 기준



Ex) PC Network Settings

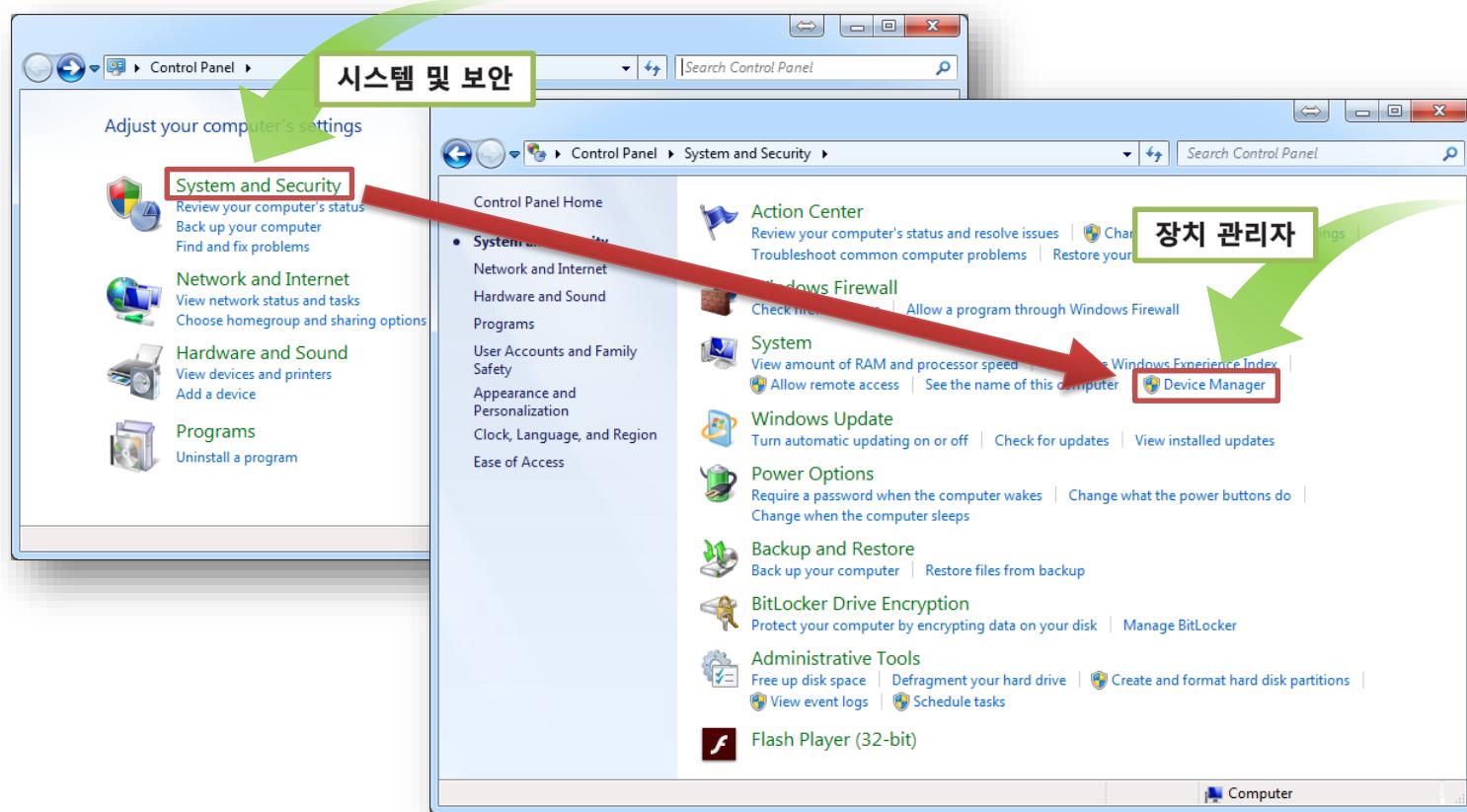
Network Settings	PC or laptop (= Remote)	IP address	192.168.11.3	-
		Gateway address	192.168.11.1	-
		Subnet mask	255.255.255.0	-
		Port number	5000	-

실습 #2: WIZ750SR 시작하기 (cont'd)

>> Serial 설정 및 COM Port 확인 (1/2)

- [제어판] – [시스템 및 보안]의 [장치 관리자] 선택

* Windows7 64bit 기준

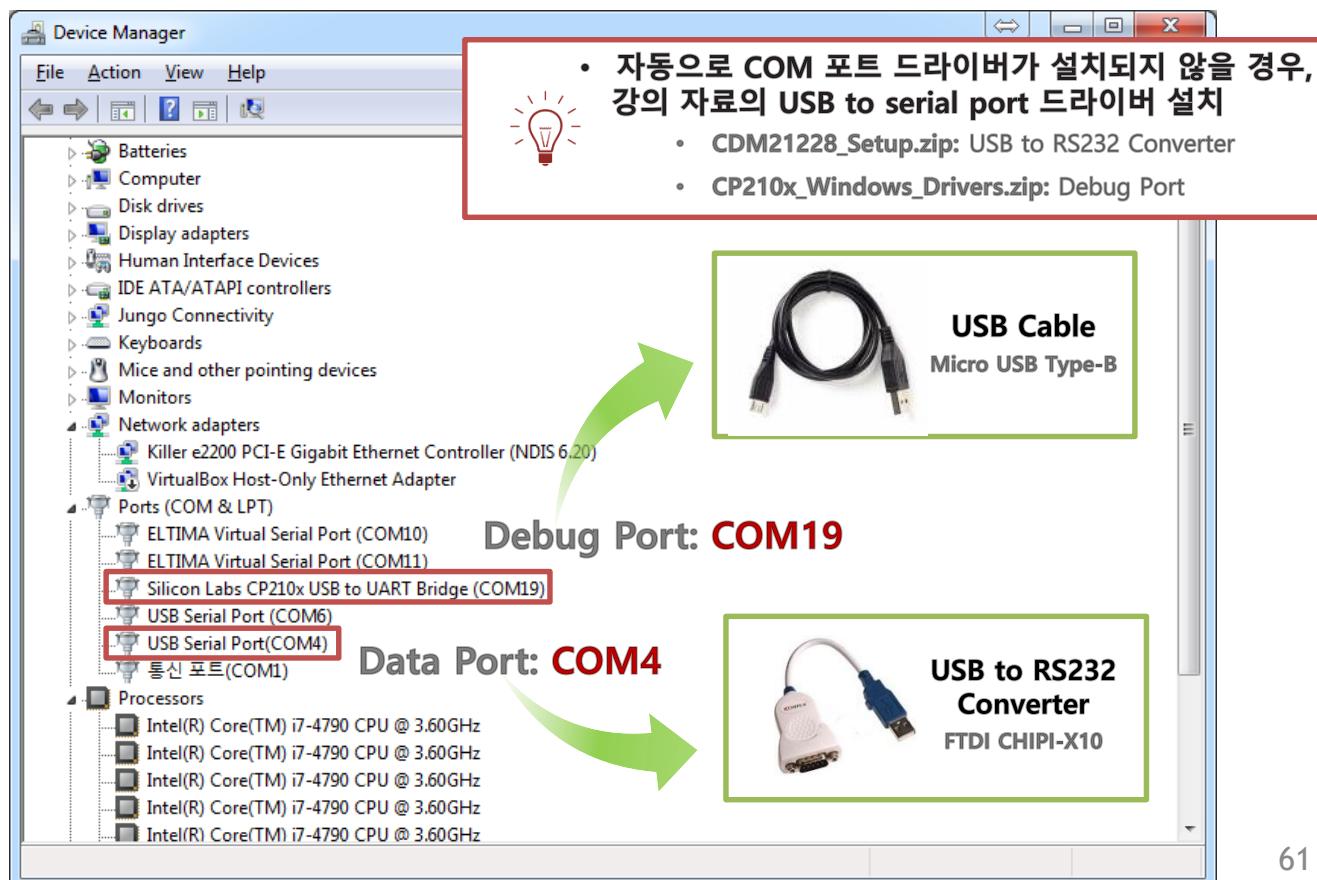


실습 #2: WIZ750SR 시작하기 (cont'd)

>> Serial 설정 및 COM Port 확인 (2/2)

- [장치 관리자] – [포트]의 COM 포트 번호 확인

* Windows7 64bit 기준

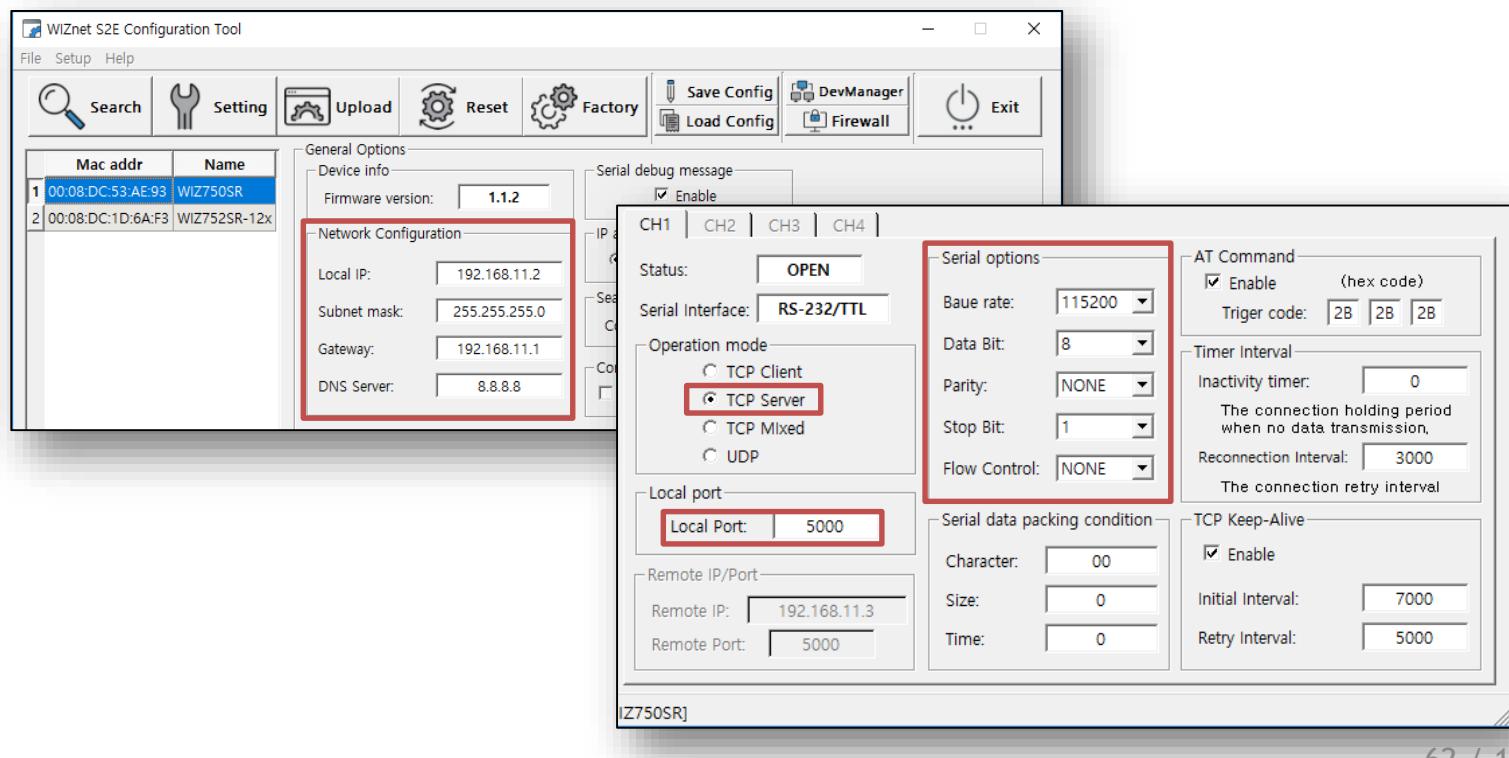


실습 #2: WIZ750SR 시작하기 (cont'd)

>> Configuration tool을 이용한 제품 검색

WIZ750SR의 네트워크 설정 및 COM 포트 설정 정보 확인

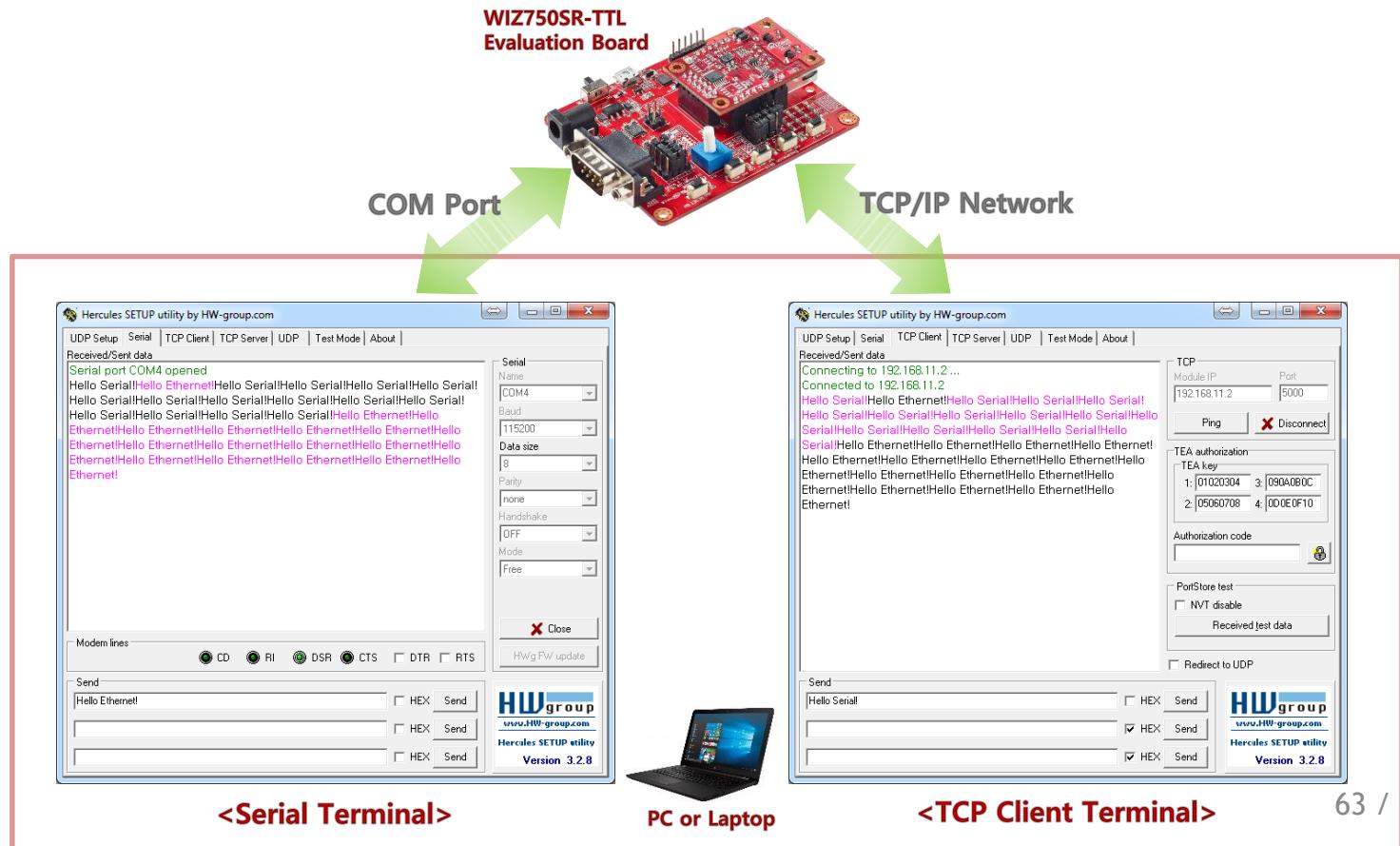
- ✓ Device의 IP 주소 및 동작 모드 정보: **192.168.11.2 : 5000, TCP Server**
- ✓ COM 포트 정보: **115200-8-N-1, None**



실습 #2: WIZ750SR 시작하기 (cont'd)

>> Serial to Ethernet 동작 테스트

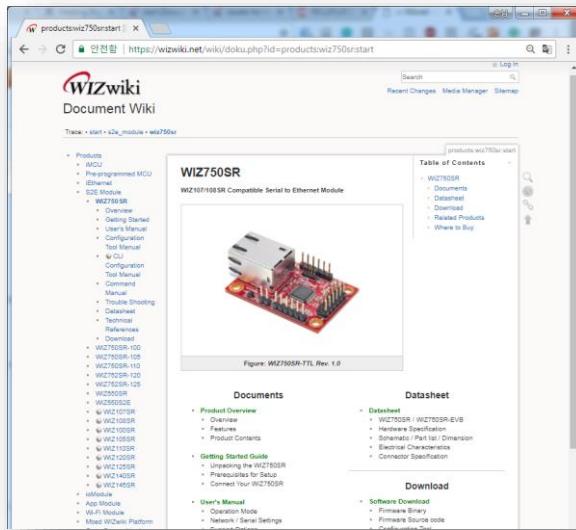
- TCP client terminal 과 Serial terminal 실행 후 WIZ750SR에 연결하여 데이터 송/수신 확인



실습 #2: WIZ750SR 시작하기 (cont'd)

>> 문제 해결 가이드

- WIZ750SR이 Configuration tool에서 검색되지 않을 때?



**Document Wiki:
WIZ750SR
Troubleshooting Guide**

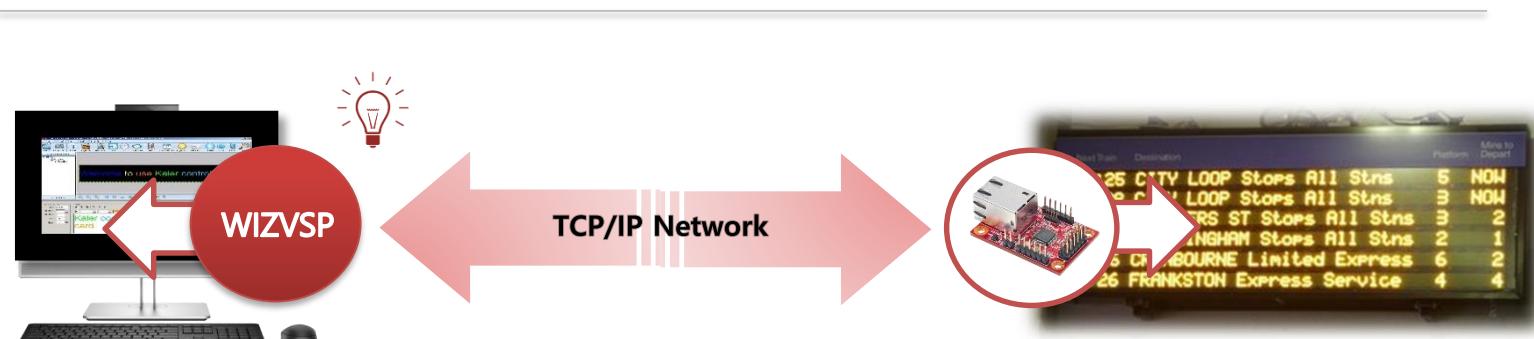
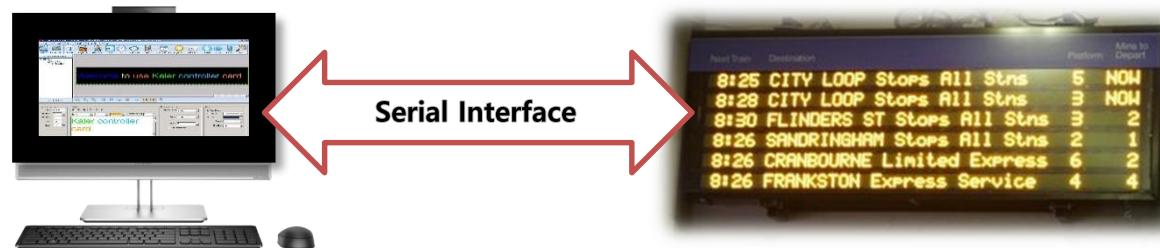
<https://wizwiki.net/wiki/doku.php?id=products:wiz750sr:troubleshooting:ko>

실습 #2: WIZ750SR 시작하기 (cont'd)

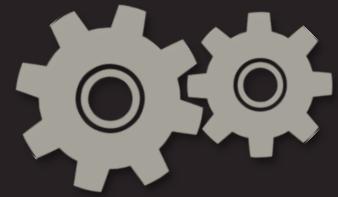
>> 여기서 잠깐: WIZVSP?

- **VSP (Virtual Serial Port program for Windows)**
- 활용 예시: LED Information Display

<Serial 연결 방식의 LED Display 제어>



<Serial to Ethernet을 활용한 원격 LED Display 제어>



실습 #3

W7500x 프로젝트 만들기



실습 #3: W7500x 프로젝트 생성하기

>> 실습 개요

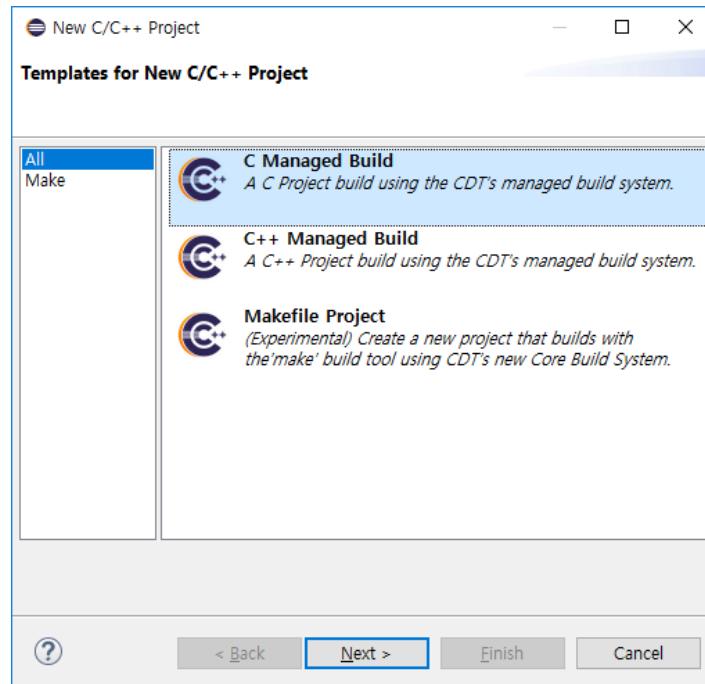
- Cortex M0용 빈 프로젝트 생성
- W7500x를 사용할 수 있는 환경 구성
- 바이너리 생성 및 다운로드
- 동작 확인



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> Eclipse에서 새 프로젝트 생성

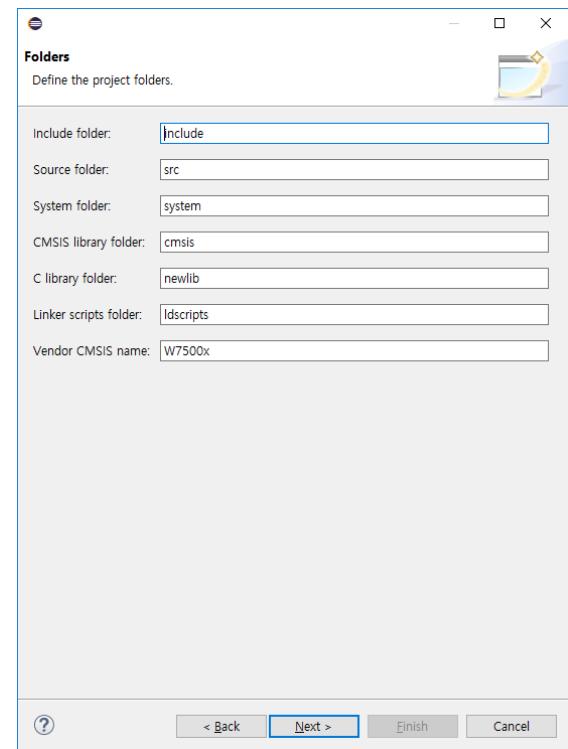
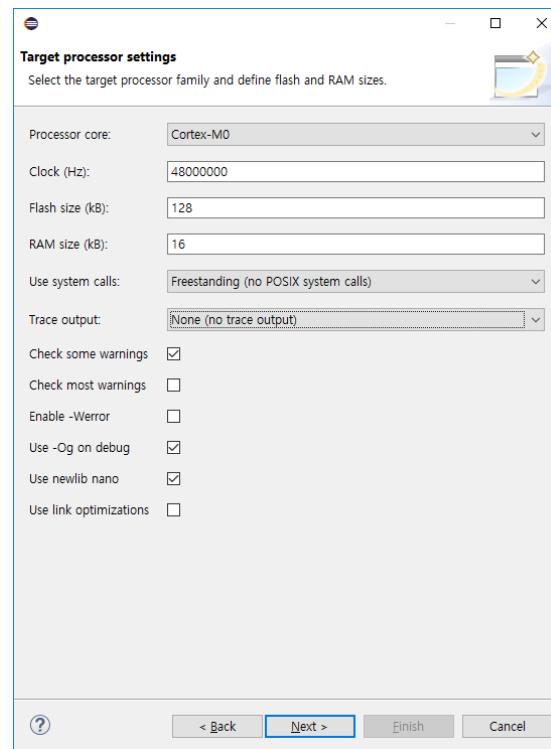
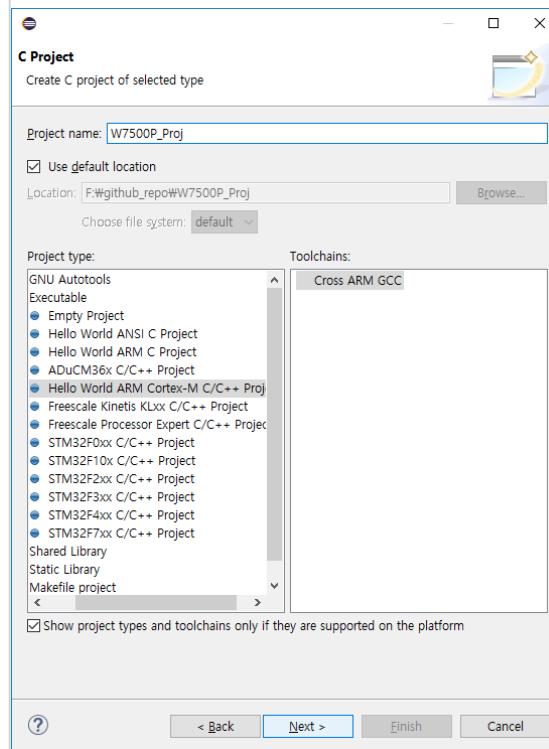
- ✓ [File]-[New]-[C/C++ Project] 선택
- ✓ 'C Managed Build' 선택 후, 'Next'



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> [Hello World ARM Cortex-M C/C++ Project] 선택

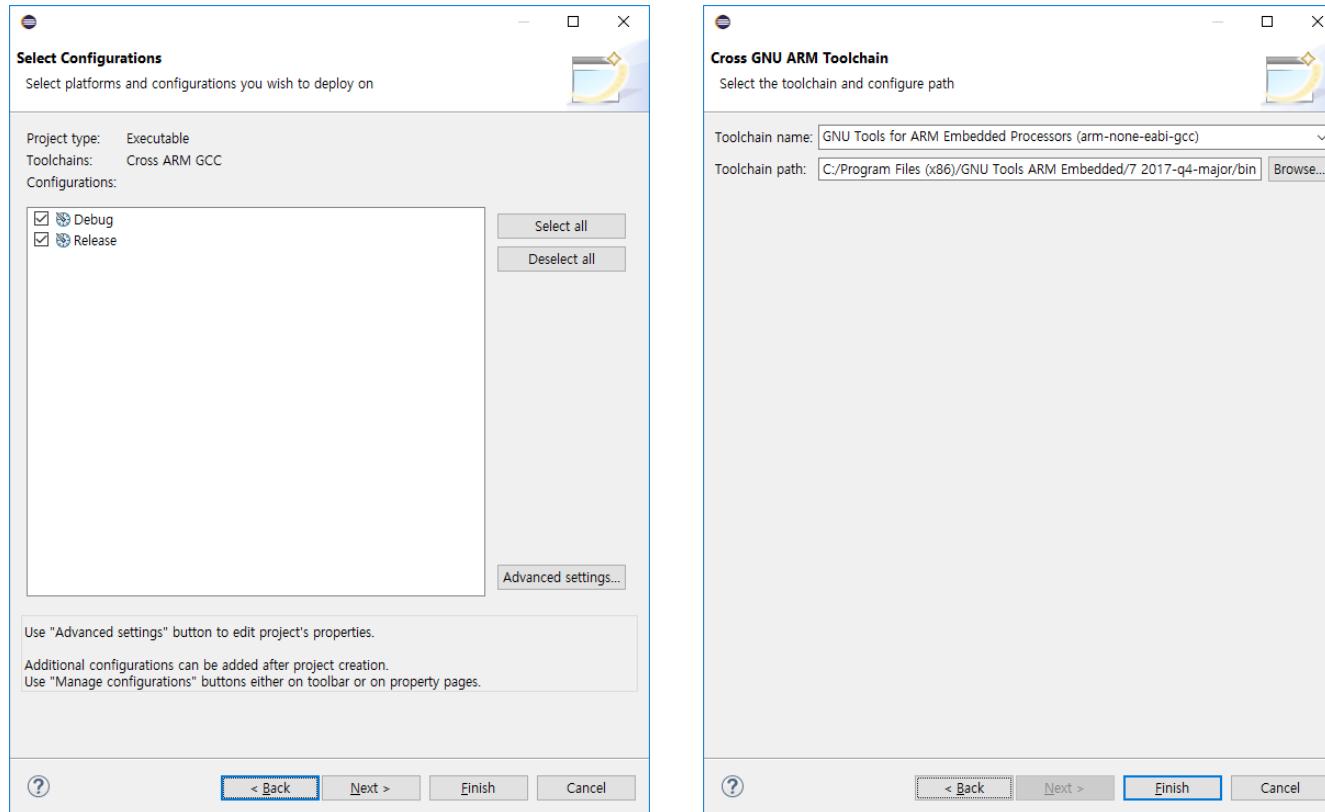
- ✓ Project Name에 원하는 이름 입력. 'W7500P_Proj' 입력
- ✓ Project type:에서 'Hello World ARM Cortex-M C/C++ Project' 선택
- ✓ 'Next' 선택



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> Select Configurations and check Toolchain path

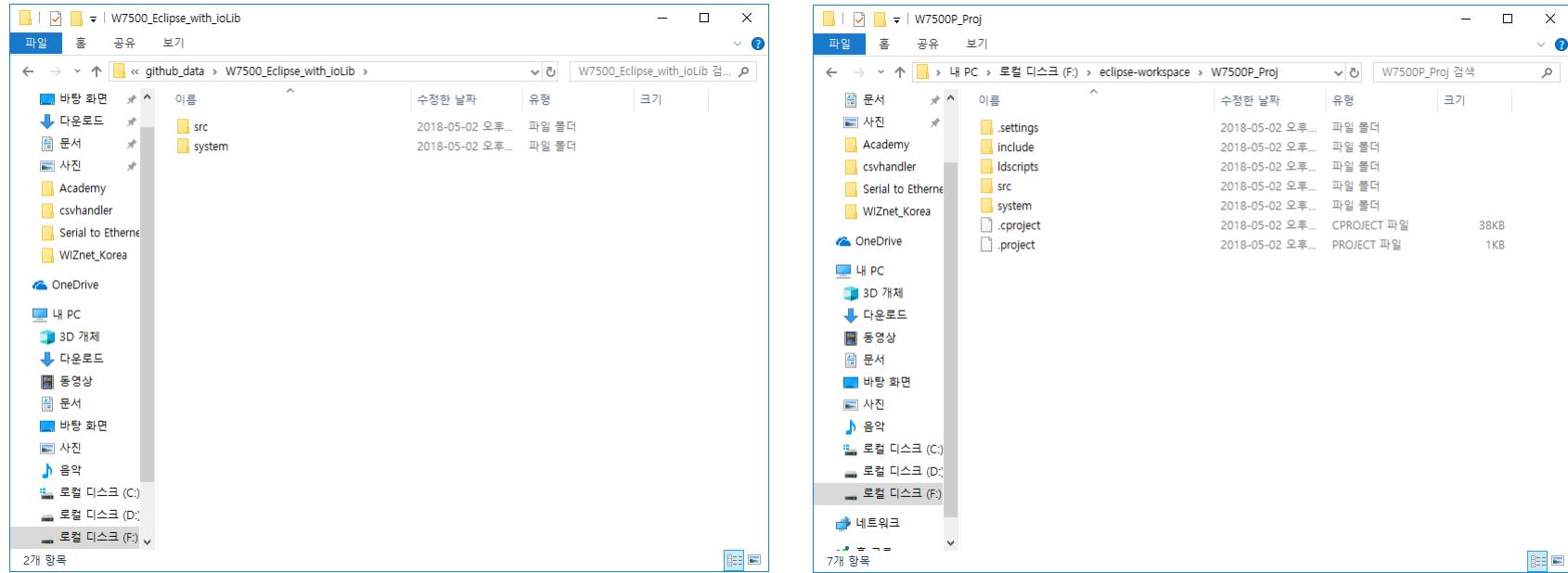
✓ Project Name에 원하는 이름 입력. 'W7500P_Proj' 입력



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> W7500x_Eclipse_with_ioLib 복사

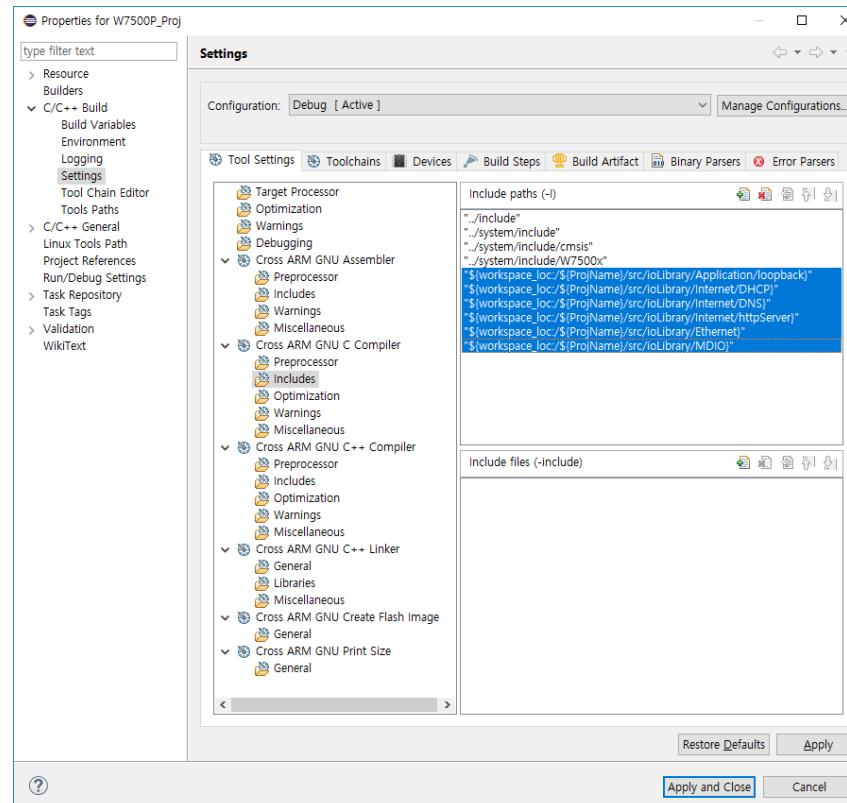
- ✓ 강의 자료 폴더 내 `github_data\W7500_Eclipse_with_ioLib` 내 `\src`, `\system` 폴더를 Eclipse 프로젝트의 `\src`, `\system` 폴더에 덮어쓴다.



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> ioLibrary Path 추가

- ✓ [Project Explorer]-[*Proj_Name*] 선택 후, 오른쪽 마우스 클릭, 'Refresh' 또는 'F5' 선택
- ✓ [Project Explorer]-[*Proj_Name*] 선택 후, 오른쪽 마우스 클릭, 'Properties' 선택
- ✓ [C/C++ Build]-[Settings] 선택. 오른쪽 Plane에서 [Tool Settings]-[Cross ARM GNU C Compiler]-[Includes] 선택



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

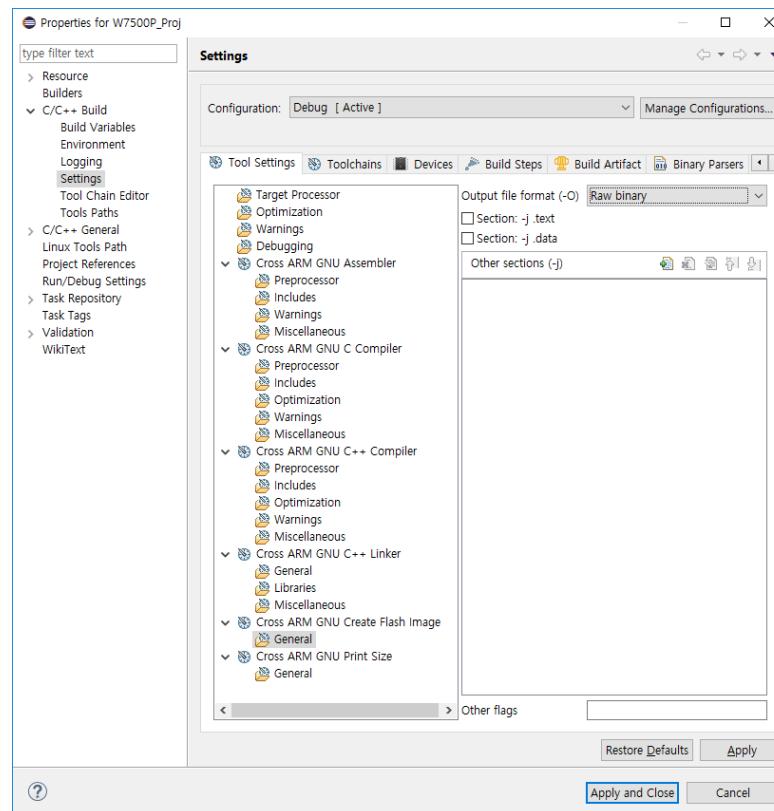
>> Timer 사용을 위한 코드 추가

- ✓ [src] 폴더에 [TimerHandler]-[PlatformHandler] 폴더 복사
- ✓ [Include] 폴더를 [TimerHandler]- [Include] 폴더로 덮어쓰기
- ✓ [system] 폴더를 [TimerHandler]- [system] 폴더로 덮어쓰기

실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> Output File Format 변경

- ✓ [Project Explorer]-[*Proj_Name*] 선택 후, 오른쪽 마우스 클릭, 'Properties' 선택
- ✓ [C/C++ Build]-[Settings] 선택. 오른쪽 Plane에서 [Tool Settings]-[Cross ARM GNU Create Flash Image]-[General] 선택
- ✓ 'Output file format (-O)'에서 'Raw binary' 선택



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> main.c Code 수정

- ✓ `#include "system_W7500x.h"` 추가
- ✓ `#include "W7500x_uart.h"` 추가

- ✓ `SystemInit_User();` 추가
- ✓ `Timer_Configuration();` 추가
- ✓ `S_UART_Init(115200);` 추가
- ✓ `trace_printf`를 `printf`로 변경

```

eclipse-workspace - W7500P_Proj/src/main.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
main.c makefile W7500x_uart.h W7500x_uart.c
10 * copy, modify, merge, publish, distribute, sublicense, and/or
11 * sell copies of the Software, and to permit persons to whom
12 * the Software is furnished to do so, subject to the following
13 * conditions:
14 *
15 * The above copyright notice and this permission notice shall be
16 * included in all copies or substantial portions of the Software.
17 *
18 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
19 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
20 * OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
21 * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
22 * HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
23 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
24 * FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
25 * OTHER DEALINGS IN THE SOFTWARE.
26 */
27
28 // -----
29
30 #include <stdio.h>
31 #include <stdlib.h>
32
33 #include "diag/Trace.h"
34
35 #include "system_W7500x.h"
36 #include "W7500x_uart.h"
37 // -----
38 // 
```

Writable Smart Insert 29 : 1

```

eclipse-workspace - W7500P_Proj/src/main.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
main.c main.c seg.c httpParser.c main.c W7500x_itc common.h
99 // HAL_Init();
100 // In this sample the SystemInit() function is just a placeholder,
101 // if you do not add the real one, the clock will remain configured with
102 // the reset value, usually a relatively low speed RC clock (8-12MHz).
103
104// Supervisory_IC_Init();
105// SystemInit();
106// SystemInit_User(DEVICE_CLOCK_SELECT, DEVICE_PLL_SOURCE_CLOCK, DEVICE_TARGET_SYSTEM_CLOCK);
107// SystemInit_User(CLOCK_SOURCE_EXTERNAL, PLL_SOURCE_12MHz, SYSTEM_CLOCK_48MHz);
108
109 Timer_Configuration();
110 S_UART_Init(115200);
111
112 SysTick_Config(GetSystemClock()/1000);
113
114 // Send a greeting to the trace device (skipped on Release).
115 printf("Hello ARM World!\r\n");
116
117 // At this stage the system clock should have already been configured
118 // at high speed.
119 printf("System clock: %lu Hz\r\n", SystemCoreClock);
120
121 int seconds = 0;
122
123 setDevtime(seconds);
124
125 // Infinite loop
126 while (1)
127 {
128     uint8_t tmpseconds;
129     delay_ms(1000);
130     seconds++;
131     if((tmpseconds = getDevtime()) != seconds)
132     {
133         seconds = tmpseconds;
134         // Count seconds on the trace device.
135         printf ("Second %d\r\n", seconds);
136     }
137 }
138 /**
139 */
140
141 // Infinite loop, never return.
<
```

Writable Smart Insert 133 : 19

실습 #3: W7500x 프로젝트 생성하기 (cont'd)

» retarget.c Code 수정

- ✓ #define USING_UART2 로 설정

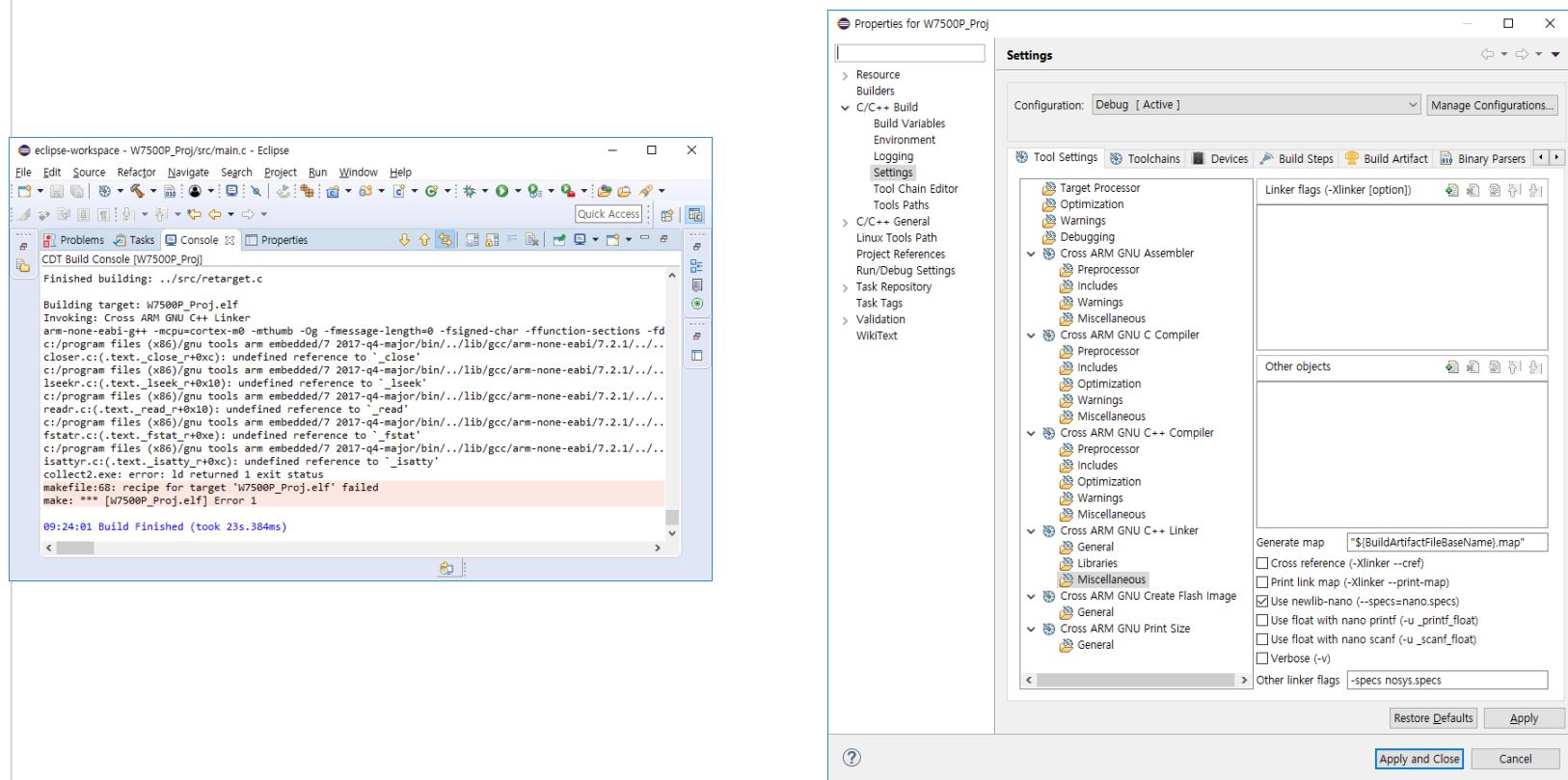
The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - W7500P_Proj/src/retarget.c - Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations like Open, Save, Find, and Build. The Quick Access button is visible. The left sidebar shows project files: main.c, retarget.c (the current editor), makefile, and W7500x_uart.h. The right sidebar shows the Package Explorer and Task List. The main editor area displays the content of retarget.c:

```
31 * <h2><center>&copy; COPYRIGHT 2015 WIZnet Co.,Ltd.</center></h2>
32 ****
33 */
34
35 #include <stdio.h>
36 #include "W7500x_uart.h"
37
38 #define USING_UART2
39
40 #if defined (USING_UART0)
41     #define UART_SEND_BYT(ch) UartPutc(UART0,ch)
42     #define UART_RECV_BYT() UartGetc(UART0)
43 #elif defined (USING_UART1)
44     #define UART_SEND_BYT(ch) UartPutc(UART1,ch)
45     #define UART_RECV_BYT() UartGetc(UART1)
46 #elif defined (USING_UART2)
47     #define UART_SEND_BYT(ch) S_UartPutc(ch)
48     #define UART_RECV_BYT() S_UartGetc()
49 #endif
50
51
52 #if defined ( __CC_ARM )
53 /* Retarget functions for ARM DS-5 Professional / Keil MDK */
54 /* **** */
55
56 #include <time.h>
57 #include <rt_msc.h>
58 #pragma import(_use_no_semihosting_sw)
59
60 struct _FILE { int handle; /* Add whatever you need here */ };
61 <
```

실습 #3: W7500x 프로젝트 생성하기 (cont'd)

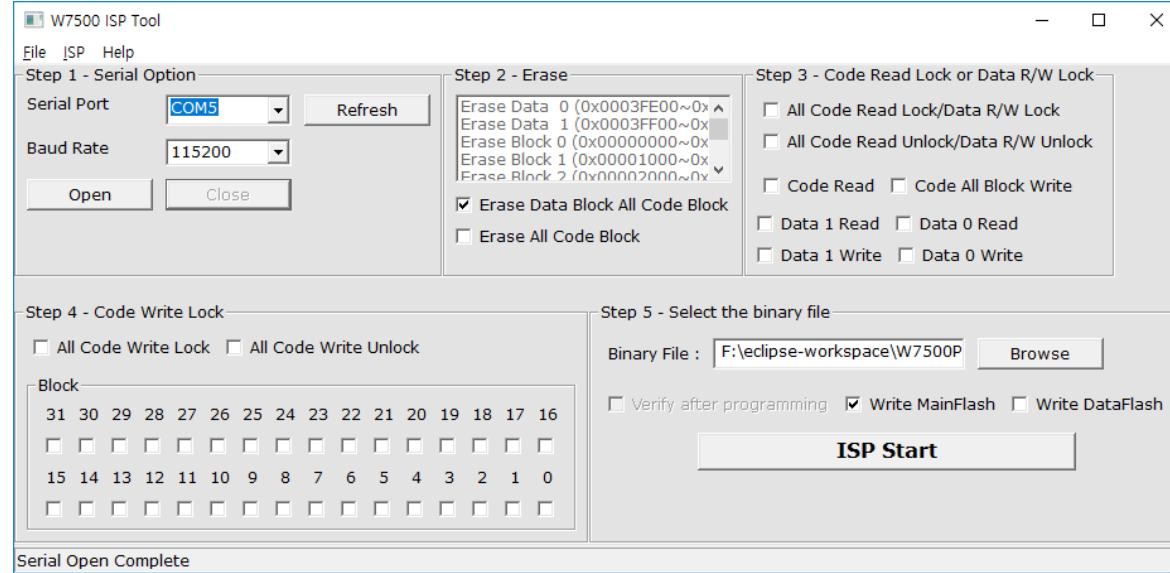
>> NewLib printf 사용을 위한 추가 옵션

- ✓ [Tool Settings]-[Cross ARM GNU C++ Linker]-[Miscellaneous] 선택
- ✓ “Other linker flags”에 “-specs nosys.specs” 추가



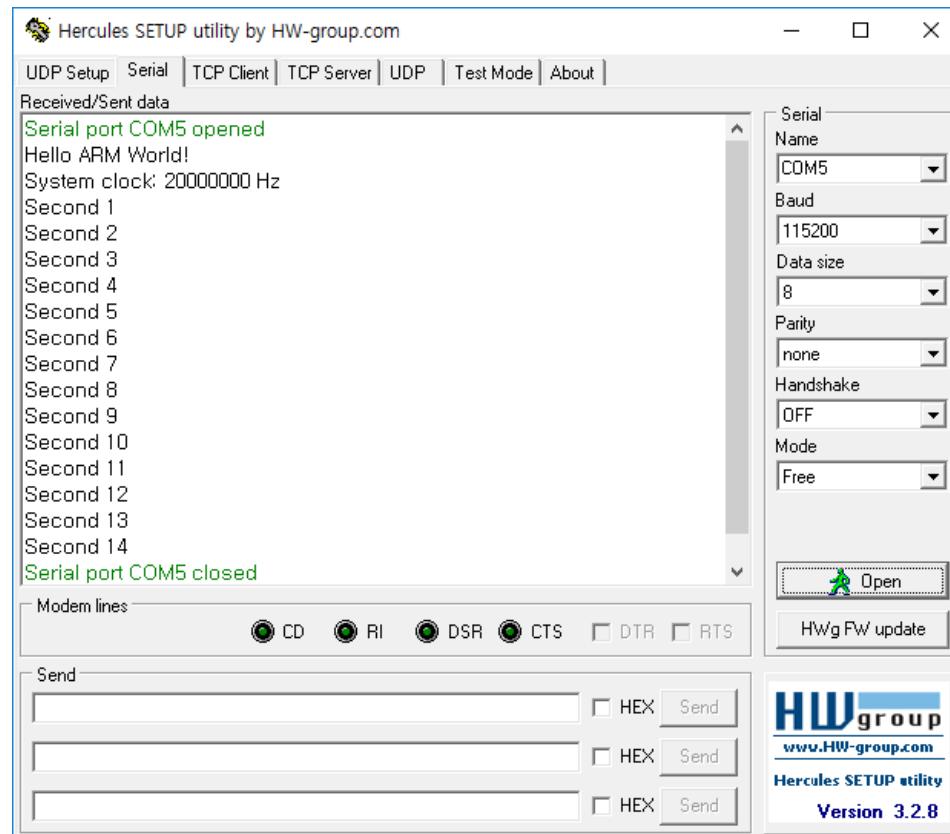
실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> W7500 ISP Tool로 Binary Download



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> Run code

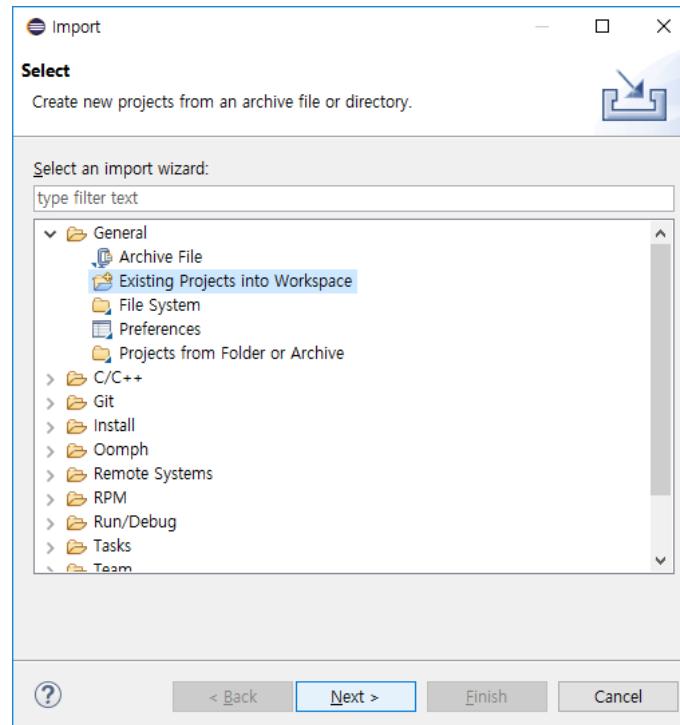




실습 #3: github 프로젝트 import 하기

>>> WIZ750SR_App_Eclipse import 하기

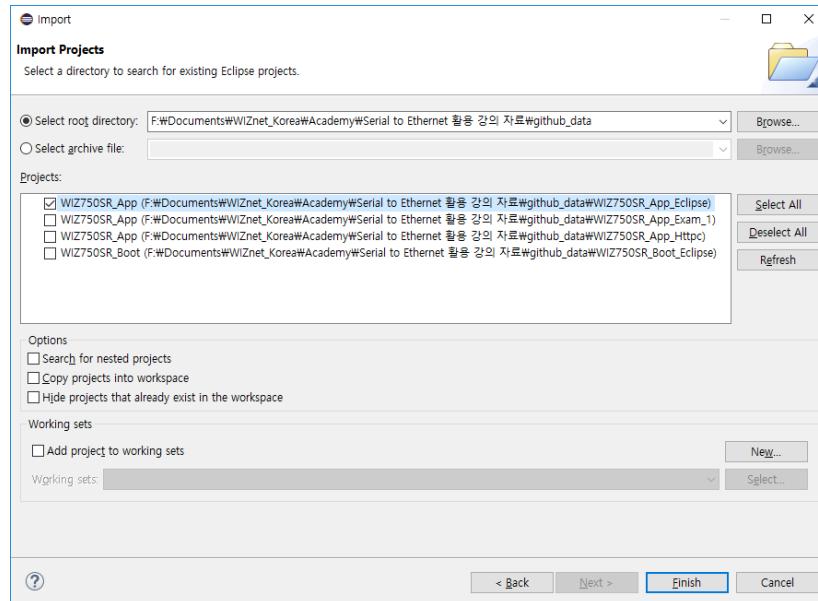
- [File]-[Import] 선택
- “Select” 대화상자에서 [General]-[Existing Project into Workspace] 선택
- “Next” 선택



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> Root Directory와 Import할 Project 지정

- ✓ Root Directory 는 강의 자료 폴더내의 “github_data” 폴더 지정
- ✓ Import할 Project는 WIZ750SR_App_Eclipse 지정
- ✓ WIZ750SR_Boot_Eclipse도 Import



The screenshot shows the Eclipse IDE interface with the 'Project Explorer' view open. The 'WIZ750SR_App' project is selected. The code editor shows a snippet of C code from 'W7500x_uart.c':

```

573     while(UARTx->FR & UART_FR_BUSY);
574     return (ch);
575 }
576
577 }
578
579
580 void UartPuts(UART_TypeDef* UARTx, uint8_t *str)
581 {
582     uint8_t ch;
583
584     do{
585         ch = *str;
586         if(ch != (uint8_t)0x00)
587         {
588             UartPutc(UARTx, ch);
589         }
590         str++;
591     }while(ch != 0);
592 }
593
594
595
596 uint8_t UartGetc(UART_TypeDef* UARTx)
597 {
598     uint8_t ch;
599
600     ch = (uint8_t)(UARTx->DR & 0xFF);
601
602     return ch;
603 }

```

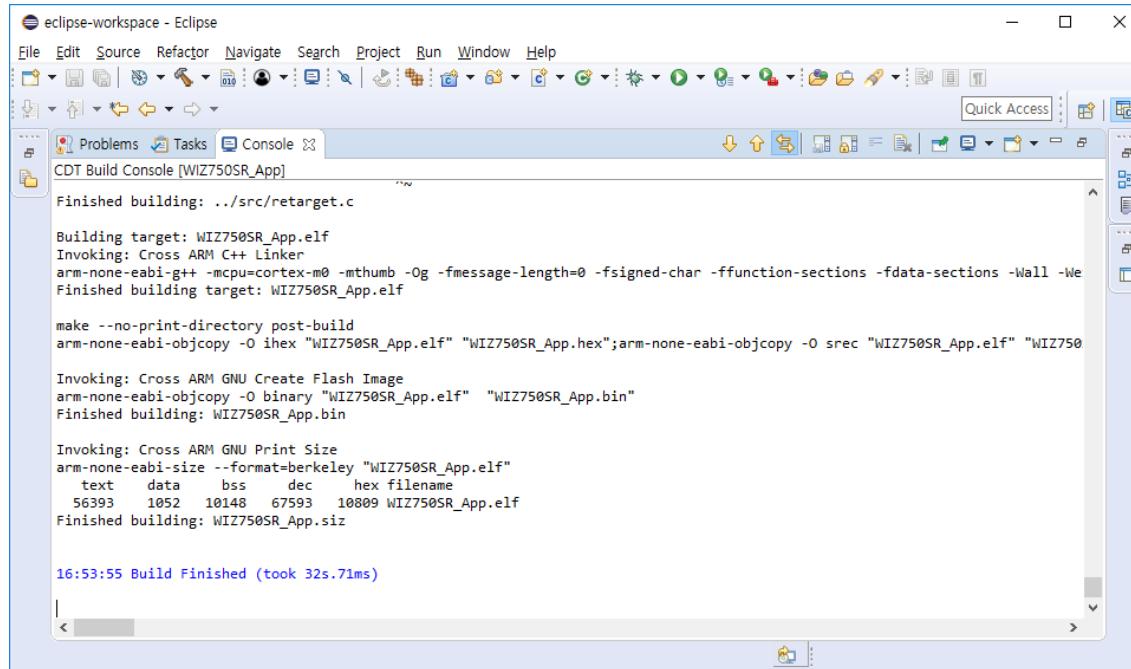
The 'Problems' view at the bottom shows 2 warnings:

Description	Resource	Path	Location
Statement has no effect: 'str++'	W7500x_uart.c	/W7500P_Proj/system/src/W7500x/W7500x_uart.c	line 591
Statement has no effect: 'str++'	W7500x_uart.c	/W7500P_Proj/system/src/W7500x/W7500x_uart.c	line 623

실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> Binary Build

- ✓ [Project]-[Build Project]를 선택해서 Build를 수행한다.
 - 오류없이 Build를 성공해야 함.



The screenshot shows the Eclipse CDT Build Console window titled "CDT Build Console [WIZ750SR_App]". The console output is as follows:

```
Finished building: ../src/retarget.c
Building target: WIZ750SR_App.elf
Invoking: Cross ARM C++ Linker
arm-none-eabi-g++ -mcpu=cortex-m0 -mthumb -Og -fmessage-length=0 -fsigned-char -ffunction-sections -fdata-sections -Wall -We
Finished building target: WIZ750SR_App.elf

make --no-print-directory post-build
arm-none-eabi-objcopy -O ihex "WIZ750SR_App.elf" "WIZ750SR_App.hex";arm-none-eabi-objcopy -O srec "WIZ750SR_App.elf" "WIZ750

Invoking: Cross ARM GNU Create Flash Image
arm-none-eabi-objcopy -O binary "WIZ750SR_App.elf" "WIZ750SR_App.bin"
Finished building: WIZ750SR_App.bin

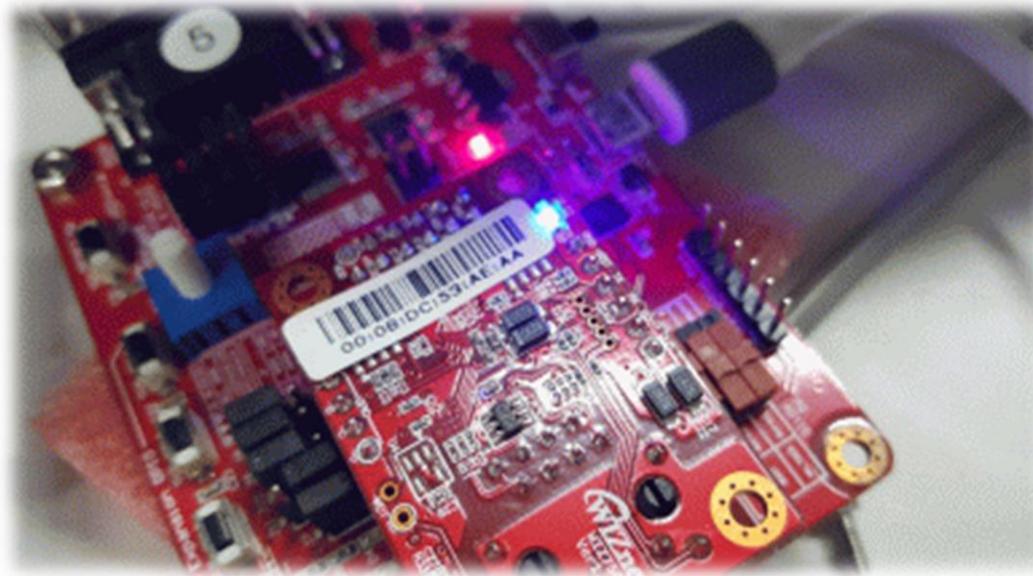
Invoking: Cross ARM GNU Print Size
arm-none-eabi-size --format=berkeley "WIZ750SR_App.elf"
    text      data      bss      dec      hex filename
  56393     1052    10148   67593   10809 WIZ750SR_App.elf
Finished building: WIZ750SR_App.size

16:53:55 Build Finished (took 32s.71ms)
```

실습 #3: Standard 펌웨어 구조 & 소스 분석

>> 여기서 잠깐: Boot vs. App-Boot vs. App

- Code flash의 0번지부터 위치하며, **Firmware update**와 **제품 복구** 등을 위해 준비된 모드
- (Application 동작과 동일한) Network Search 및 Setting 기능 지원
- 모드 진입 방법
 - WIZ750SR-EVB의 경우: 스위치 On / 모듈의 경우: APP_BOOT Pin 제어 (Active-Low, PC_14)

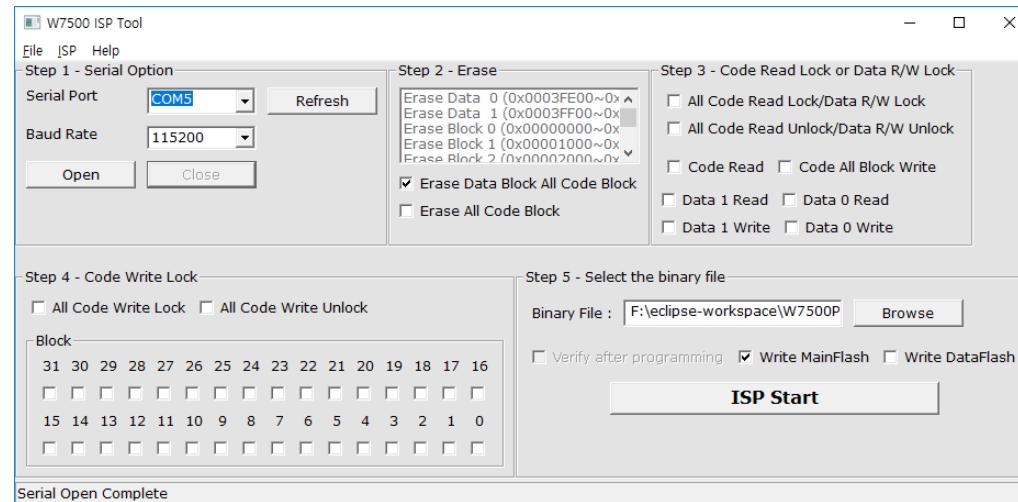


WIZ750SR의 경우, App-Boot 모드 진입 시 Status LED 1, 2가 빠르게 번갈아가며 점멸

실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> Binary Writing

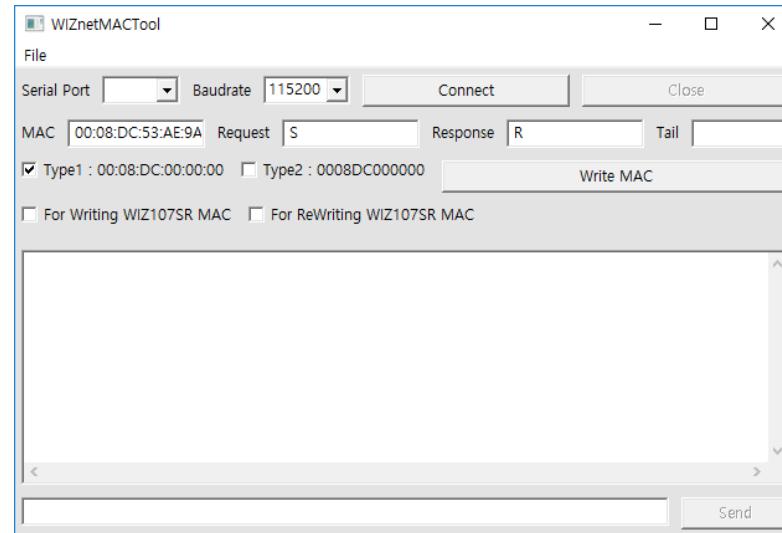
- ✓ WIZ750SR을 BOOT mode로 변경한 후,
- ✓ W7500 ISP Tool을 이용해서 Flash로 Binary Writing
 - APP_BOOT 와 APP이 통합된 Binary 필요
 - 또는 APP_BOOT Binary만 Writing. APP은 이후 Ethernet을 통해서 업로드.



실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> MAC 주소 Writing

- ✓ WIZ750SR을 Normal mode로 변경한 후,
- ✓ WizMACTool로 MAC 주소 Writing

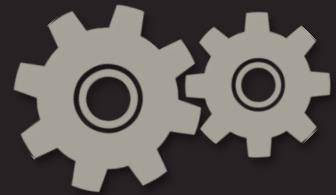


실습 #3: W7500x 프로젝트 생성하기 (cont'd)

>> 동작 확인

- ✓ Console 메시지로 동작 확인
- ✓ Tera Term이나 Hercules로 확인

```
W COM8 - Tera Term VT
File Edit Setup Control Window Help
>> WIZnet Serial to Ethernet Device
>> Firmware version: 1.2.2 Stable
=====
- Device name: WIZ750SR
- Device mode: TCP_CLIENT_MODE
- Network settings:
  - Obtaining IP settings: [Static]
  - TCP/UDP ports
    + S2E data port: [5000]
    + TCP/UDP setting port: [50001]
    + Firmware update port: [50002]
  - TCP Retransmission retry: [8]
- Search ID code:
  - Disabled; [None]
- Ethernet connection password:
  - Disabled (TCP server / mixed mode only)
- Connection timer settings:
  - Inactivity timer: [10] (sec)
  - Reconnect interval: [3000] (msec)
- Serial settings:
  - Data UART port: [UART0]
    + UART IP: [RS-232/TTL]
    + 115200-8-N-1 / Flow control: NONE
  - Debug UART port: [UART2]
    + 115200-8-N-1 / NONE (fixed)
- Serial data packing options:
  - Time: [100] (nsec)
  - Size: Disabled
  - Char: Disabled
- Serial command mode suffix code:
  - Enabled
  - [28][28][28] (Hex only)
- Hardware information: Status pins
  - Status 1: [PA_10] - PHY link
  - Status 2: [PA_01] - TCP connection
- Hardware information: User I/O pins
  - UserIO A: [PC_13] - Digital / Input
  - UserIO B: [PC_12] - Digital / Input
  - UserIO C: [PC_09] - Digital / Input
  - UserIO D: [PC_08] - Digital / Input
=====
# MAC: 00:08:0C:53:AE:99
# IP : 192.168.11.100 / Port: 5000
# GH : 192.168.11.1
# SN : 255.255.255.0
# DNS: 8.8.8.8
# Destination IP: 192.168.11.235 / Port: 50000
> SEG:TCP_CLIENT_MODE:SOCKOPEN
```



실습 #4

Standard 펌웨어 구조와
소스코드 분석



실습 #4: Standard 펌웨어 구조 & 소스 분석

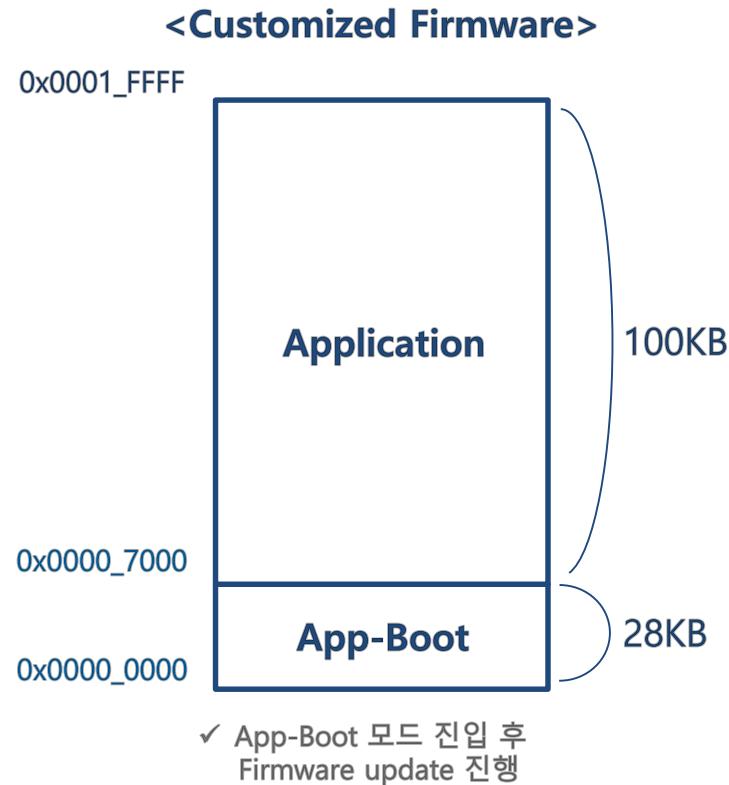
>>> 실습주제

- WIZ750SR Standard 펌웨어 구조
- 소스 코드 분석

실습 #4: Standard 펌웨어 구조 & 소스 분석

>> WIZ750SR 펌웨어: Simple Flash Memory Map

- 512Bytes의 Data Flash 함께 내장, MAC 주소 및 제품 설정 정보 저장 용도로 활용



실습 #4: App-Boot vs. App



실습 #4: Firmware Structure

»» WIZ750SR 펌웨어 폴더 구조

- MCU Independent Folder vs. MCU Dependent Folder

이름	수정한 날짜	유형
Callback	2018-05-04 오후...	파일 폴더
Configuration	2018-05-04 오후...	파일 폴더
ioLibrary	2018-05-04 오후...	파일 폴더
PlatformHandler	2018-05-04 오후...	파일 폴더
Serial_to_Ethernet	2018-05-04 오후...	파일 폴더
main.c	2018-05-04 오후...	C 소스
retarget.c	2018-04-11 오전...	C 소스
W7500x_board.c	2018-05-04 오후...	C 소스
W7500x_it.c	2018-04-11 오전...	C 소스

MCU Independent Folder

- Callback
- Configuration
 - IP 주소, Serial baud rate 등 Peripheral 초기화를 위한 설정값
- ioLibrary
 - DHCP, DNS, HTTPS 등 TCP/IP 기반 응용 프로그램
- Serial_to_Ethernet
 - Serial to Ethernet Converter 기능 파트

MCU Dependent Folder

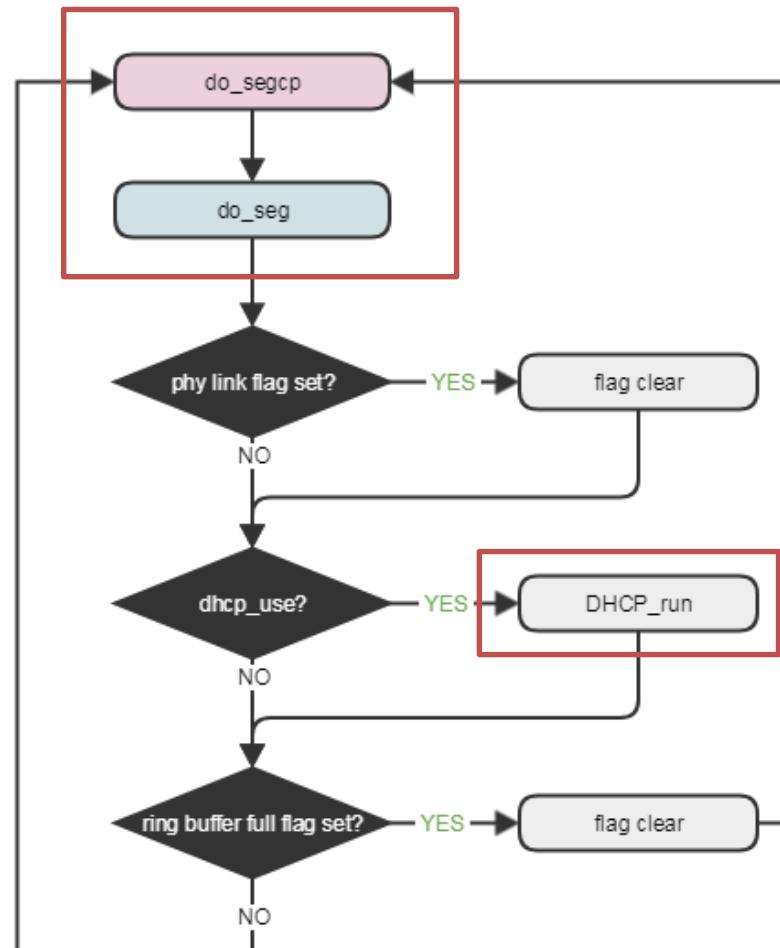
- PlatformHandler
 - MCU에 따라서 Library가 달라지는 파트
 - Flash, I2C, GPIO, UART, Timer handler
 - deviceHandler.c, .h
 - eepromHandler.c, .h
 - flashHandler.c, .h
 - gpioHandler.c, .h
 - i2cHandler.c, .h
 - storageHandler.c, .h
 - timerHandler.c, .h
 - uartHandler.c, .h

실습 #4: Standard 펌웨어 구조 & 소스 분석

»» WIZ750SR 펌웨어 main() 루틴 구조 분석

- 'S2E' Architecture Functional Block diagram
 - Initialization Code
 - TCP/IP Core Initialization
 - GPIO 등 Peripheral Initialization
 - Configuration Data Loading
 - Network Initialization
 - Main Loop
 - Flowchart: **do_seg()** function
 - Serial to Ethernet 기능 담당, 데이터 송/수신 및 제어
 - Flowchart: **do_segcp()** function
 - Configuration tool 연동, Command 송/수신 및 제어

Simple Flowchart: Main Loop



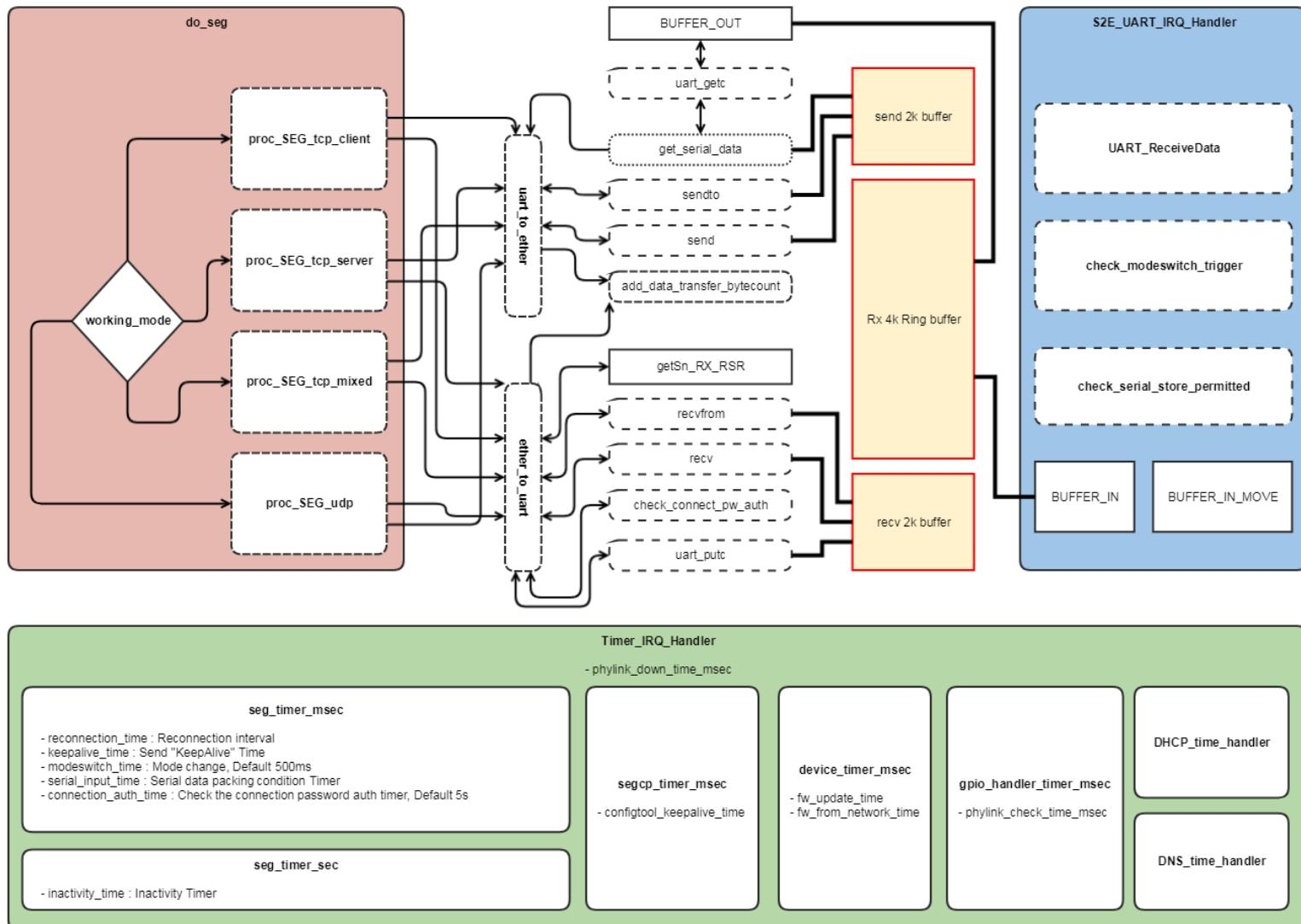
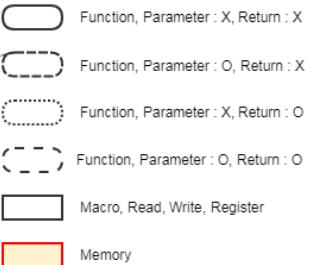
'Serial to Ethernet' Architecture Functional Block diagram

WIZnet

WIZ750SR

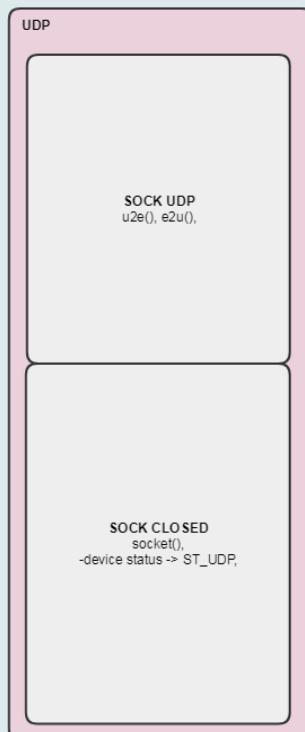
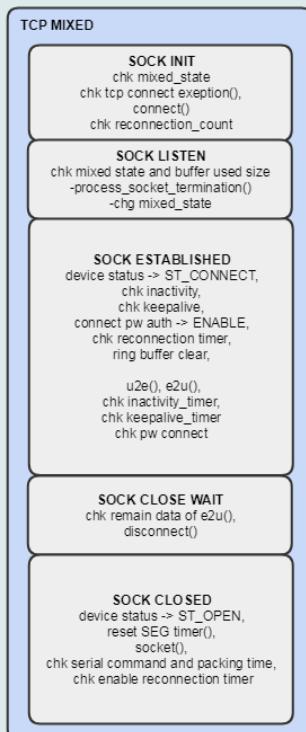
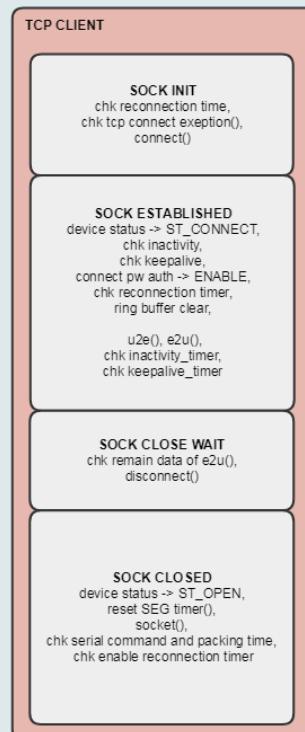
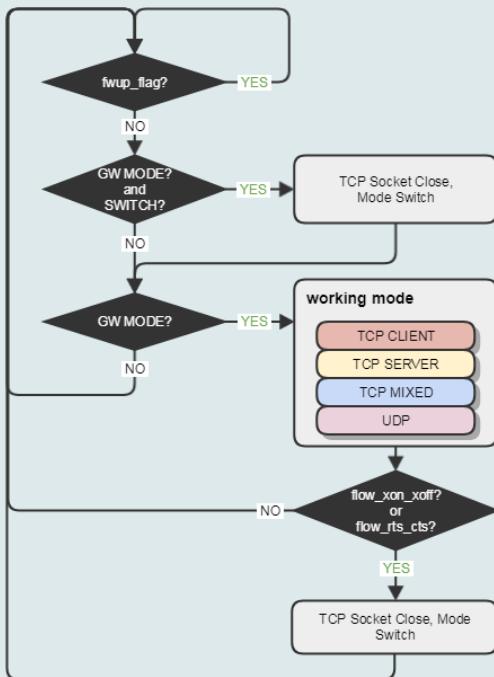
<https://github.com/Wiznet/WIZ750SR>

By Ricky

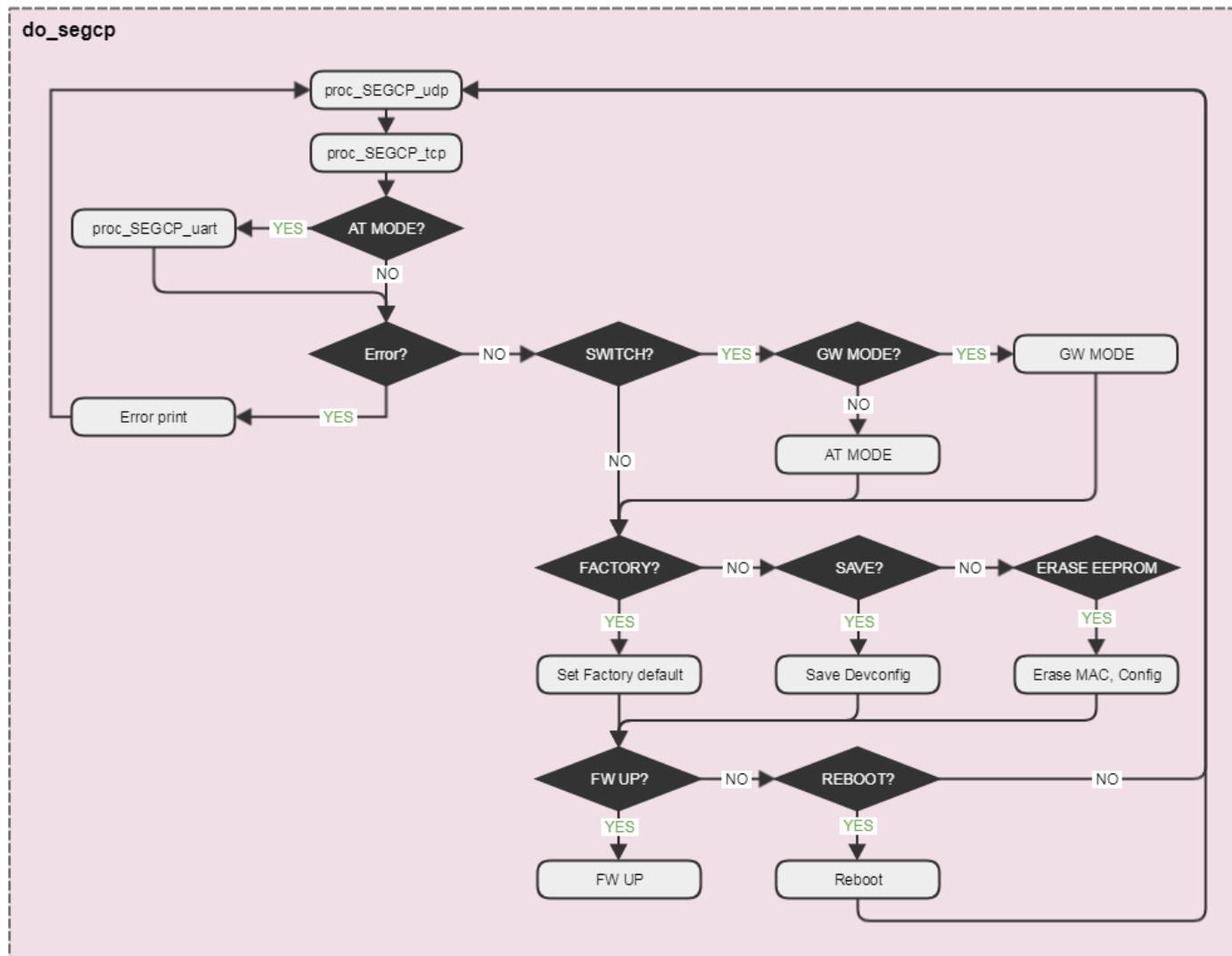


Flowchart: do_seg() function

do_seg



Flowchart: do_segcp() function



Config-Data Structure

Config-Data: __DevConfig

Type	Name	Size (bytes)
uint16_t	packet_size	2
uint8_t	module_type[]	3
uitn8_t	module_name[]	15
uint8_t	fw_version[]	3
struct	__network_info_common	18
struct	__network_info	28
struct	__serial_info	9
struct	__options	82
struct	__user_io_info	8
struct	__firmware_update	7
struct	__firmware_update_extend	74

Struct: __user_io_info

Type	Name	Size
uint16_t	user_io_enable	2
uint16_t	user_io_type	2
uint16_t	user_io_direction	2
uint16_t	user_io_status	2

Struct: __network_info_common

Type	Name	Size
uint8_t	mac[]	6
uint8_t	local_ip[]	4
uitn8_t	gateway[]	4
uint8_t	subnet[]	4

Struct: __network_info

Type	Name	Size
uint8_t	working_mode	1
uint8_t	state	1
uint8_t	remote_ip[]	4
uint16_t	local_port	2
uint16_t	remote_port	2
uint16_t	inactivity	2
uint16_t	reconnection	2
uint16_t	packing_time	2
uint8_t	packing_size	1
uint8_t	packing_delimiter[]	4
uint8_t	packing_delimiter_length	1
uint8_t	packing_data_appendix	1
uint8_t	keepalive_en	1
uint16_t	keepalive_wait_time	2
uint16_t	keepalive_retry_time	2

Struct: __serial_info

Type	Name	Size
uint8_t	uart_interface	1
uint8_t	baud_rate	1
uint8_t	data_bits	1
uint8_t	parity	1
uint8_t	stop_bits	1
uint8_t	flow_control	1
uint8_t	dtr_en	1
uint8_t	dsr_en	1
uint8_t	serial_debug_en	1

Struct: __options

Type	Name	Size
char	pw_connect[]	10
char	pw_search[]	10
uint8_t	pw_connect_en	1
uint8_t	dhcp_use	1
uint8_t	dns_use	1
uint8_t	dns_server_ip[]	4
char	dns_domain_name[]	50
uint8_t	serial_command	1
uint8_t	serial_command_echo	1
uint8_t	serial_trigger[]	3

실습 #4: Standard 펌웨어 구조 & 소스 분석

» WIZ750SR 펌웨어: 소스 코드 분석



WIZ750SR Project GitHub Repository



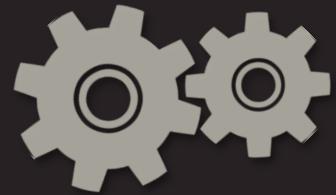
MDK-ARM (Keil uvision 5)

<https://github.com/Wiznet/WIZ750SR>

Eclipse IDE with GNU GNU Arm Embedded Toolchain

https://github.com/hkjung/WIZ750SR_App_Eclipse

https://github.com/hkjung/WIZ750SR_Boot_Eclipse



실습 #5

Serial to Ethernet Customize 예제 구현(1)

- 주기적인 사용자 I/O 모니터링 기능 추가



실습 #5: Serial to Ethernet Customize 예제 구현(1)

>> 주기적인 사용자 I/O 모니터링 기능 추가

- 동작 시나리오
 - Device는 **TCP Client mode**로 동작
 - TCP Server와 연결 이후, 서버로 Device ID와 GPIOB의 상태 전송
 - Data는 **JSON 형태**, {"DEVICE":ID, "DIN":**HIGH or LOW**} 구조
 - Device의 ID는 **DeviceName-MAC주소NIC**
 - 전송 주기는 **Time serial data packing option** 값 활용



실습 #5: S2E Customize 예제 구현(1) (cont'd)

>> 이벤트 발생을 위한 Time counter 및 Flag 추가

- WIZ750SR/Projects/S2E_App/src/Serial_to_Ethernet/seg.c
- seg_timer_msec() function에 아래 코드 삽입

```
...
if(enable_event_timer)
{
    if(netinfo->packing_time == 0)
        event_interval = DEFAULT_MODESWITCH_INTER_GAP; // 500ms
    else
        event_interval = netinfo->packing_time;

    if(event_time_cnt < event_interval)
    {
        event_time_cnt++;
    }
    else
    {
        event_time_cnt = 0;
        flag_event_occurred = 1;
    }
}
...
```

실습 #5: S2E Customize 예제 구현(1) (cont'd)

>> 사용자 Digital IO의 상태 값 전송

- WIZ750SR/Projects/S2E_App/src/Serial_to_Ethernet/seg.c
- proc_SEG_tcp_client() function 내 SOCK_ESTABLISHED 이후 동작에 아래 코드 삽입
- 데이터의 JSON 구조화? 고민X (수신의 경우 parser 활용, 송신의 경우 문자열 데이터 생성)

```
...
if(flag_event_occurred)
{
    get_user_io_val(USER_IO_B, io_val[1]);

    ex_data_len = sprintf((char *)g_send_buf,
        "{\"DEVICE\":\"%s-%02X%02X%02X\", \"DIN-B\":\"%s\"}\r\n",
        s2e->module_name,
        s2e->network_info_common.mac[3],
        s2e->network_info_common.mac[4],
        s2e->network_info_common.mac[5],
        io_val[1]?LOW_STR:HIGH_STR);

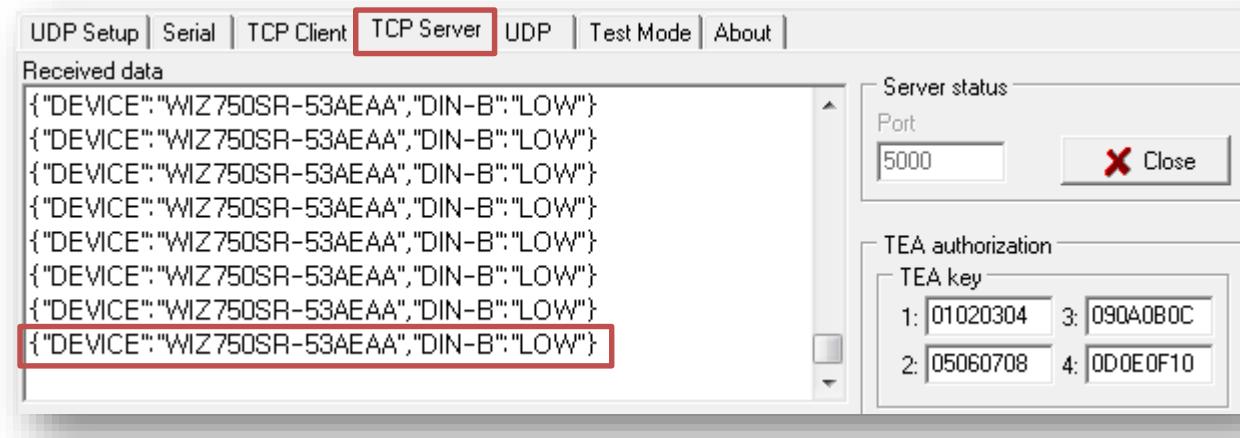
    // TCP data send to server
    ex_sent_len = (uint16_t)send(sock, g_send_buf, ex_data_len);

    // Flag Clear
    flag_event_occurred = SEG_DISABLE;
}
...
```

실습 #5: S2E Customize 예제 구현(1) (cont'd)

>> 결과

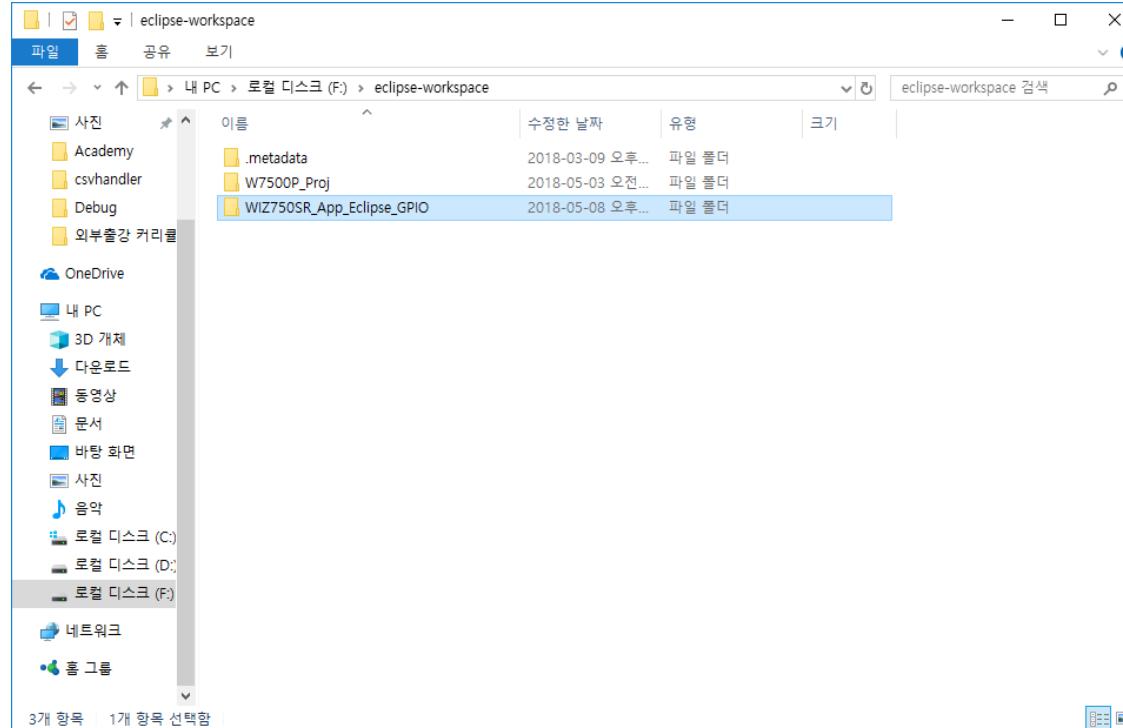
- TCP Server Terminal로 Device 정보와 IO 상태 메시지 수신 확인 가능
- WIZ750SR-EVB의 Expansion GPIO B 버튼을 눌러 IO 상태 변경 확인



실습 #5: S2E Customize 예제 구현(1) (cont'd)

>> 실습 1단계 – Copy Base Project

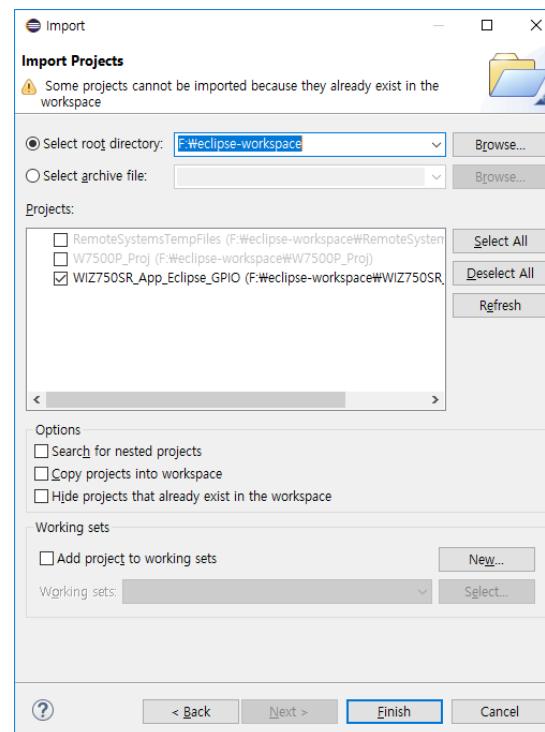
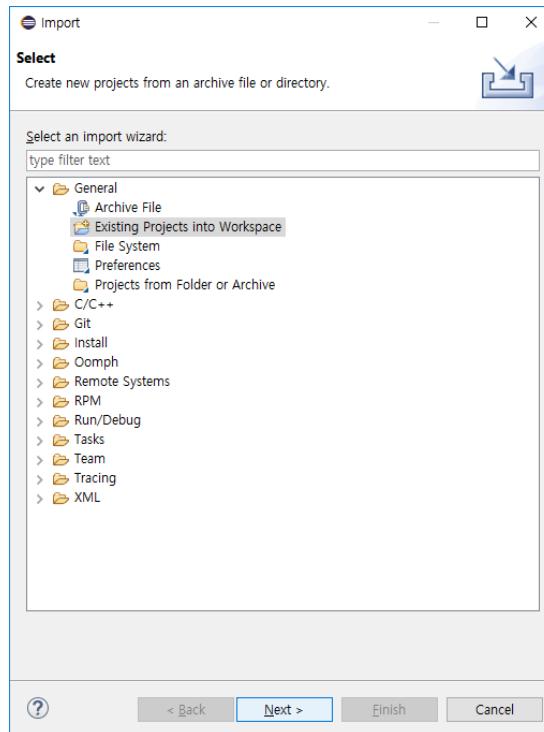
- Github에서 다운로드한 WIZ750SR_App_Eclipse 폴더를 workspace로 복사
- 폴더명을 WIZ750SR_App_Eclipse_GPIO로 변경한다.



실습 #5: S2E Customize 예제 구현(1) (cont'd)

>> 실습 2단계 – Import Project

- Eclipse에서 [File]-[Import]를 선택한 후,
- [General]-“Existing Projects into Workspace”를 선택하고 “Next” 버튼을 클릭한다.
- “Select root directory”에 프로젝트 폴더를 옮긴 상위 폴더를 지정하고,
- [Projects] 리스트에서 복사한 프로젝트 폴더를 선택하고 “Finish” 버튼을 클릭한다.



실습 #5: S2E Customize 예제 구현(1) (cont'd)

» 실습 3단계 – 전송 주기 설정을 위한 코드 추가

- seg.c 내 seg_timer_msec() 함수내에 아래 code를 추가한다

```
if(enable_event_timer)
{
    if(netinfo->packing_time == 0)
        event_interval = DEFAULT_MODESWITCH_INTER_GAP; // 500ms
    else
        event_interval = netinfo->packing_time;

    if(event_time_cnt < event_interval)
    {
        event_time_cnt++;
    }
    else
    {
        event_time_cnt = 0;
        flag_event_occurred = 1;
    }
}
```

- seg.c 내 아래 변수 선언을 추가한다

```
// added for customizing code
uint8_t enable_event_timer = SEG_DISABLE;
volatile uint16_t event_time_cnt = 0;
volatile uint16_t event_interval = 0;
uint8_t flag_event_occurred = SEG_DISABLE;
```

실습 #5: S2E Customize 예제 구현(1) (cont'd)

>> 실습 4단계 – 데이터 통신 전용 함수 생성

- TCP Client로 동작하므로 seg.c 내의 proc SEG_tcp_client() 함수를 복사해서 새로운 함수 proc_GPIO_tcp_client() 를 생성한다.
- seg.c 선언부에 void proc_GPIO_tcp_client(uint8_t sock); 을 추가한다.
- proc_GPIO_tcp_client() 함수에서 TCP Connection이 이루어지면 event_timer_enble을 ENABLE한다.

```
if(enable_event_timer)
{
    if(netinfo->packing_time == 0)
        event_interval = DEFAULT_MODESWITCH_INTER_GAP; // 500ms
    else
        event_interval = netinfo->packing_time;

    if(event_time_cnt < event_interval)
    {
        event_time_cnt++;
    }
    else
    {
        event_time_cnt = 0;
        flag_event_occurred = 1;
    }
}
```

실습 #5: S2E Customize 예제 구현(1) (cont'd)

>> 실습 5단계 – 통신처리 코드 추가

- 데이터 전송 주기가 되면,
- GPIO 상태를 읽어서 JSON 포맷의 전송 메시지를 생성하고
- 연결된 서버로 데이터를 보내는 코드를 추가한다.

```
if(flag_event_occurred)
{
    get_user_io_val(USER_IO_B, &io_val);

    ex_data_len = sprintf((char *)g_send_buf,
        "{\"DEVICE\": \"%s-%02X%02X%02X\", \"DIN-
        B\": \"%s\"}\r\n",
        s2e->module_name,
        s2e->network_info_common.mac[3],
        s2e->network_info_common.mac[4],
        s2e->network_info_common.mac[5],
        io_val?LOW_STR:HIGH_STR);

    // TCP data send to server
    ex_sent_len = (uint16_t)send(sock, g_send_buf,
        ex_data_len);

    // Flag Clear
    flag_event_occurred = SEG_DISABLE;
}
```

실습 #5: S2E Customize 예제 구현(1) (cont'd)

>> 실습 6단계 – do_seg() 함수 수정

- do_seg() 함수의 working_mode에 대한 switch 문에서 TCP_CLIENT_MODE 일 때 호출하는 함수를 proc_GPIO_tcp_client()로 변경한다.

```
if(opmode == DEVICE_GW_MODE)
{
    switch(net->working_mode)
    {
        case TCP_CLIENT_MODE:
            //proc_SEG_tcp_client(sock);
            proc_GPIO_tcp_client(sock);
            break;

        case TCP_SERVER_MODE:
            proc_SEG_tcp_server(sock);
            break;

        case TCP_MIXED_MODE:
            proc_SEG_tcp_mixed(sock);
            break;

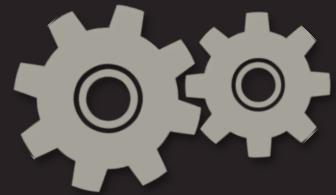
        case UDP_MODE:
            proc_SEG_udp(sock);
            break;

        default:
            break;
    }
}
```

실습 #5: S2E Customize 예제 구현(1) (cont'd)

»» 실습 6단계 – Build and Download

- Eclipse에서 [Project]-[Build Project]를 선택해서 Binary를 생성한다.



실습 #6

Serial to Ethernet Customize 예제 구현(2)

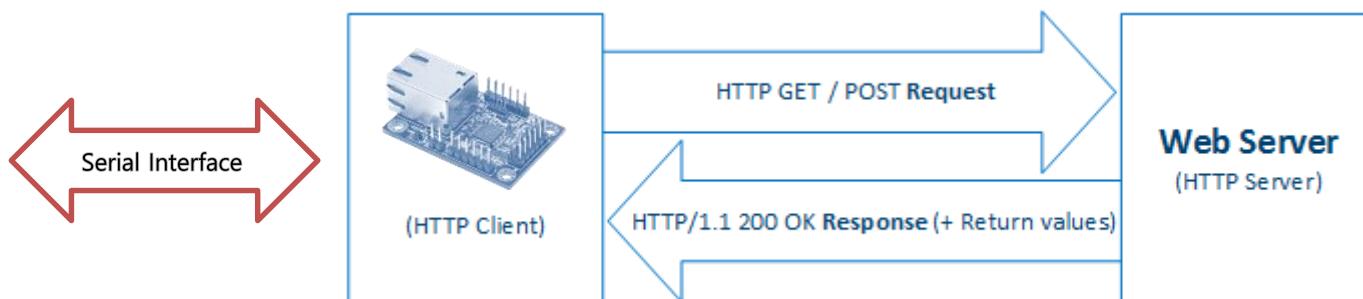
- Serial to HTTP Client 구현



실습 #6: Serial to Ethernet Customize 예제 구현(2)

>>> Serial to HTTP Client 구현

- 동작 시나리오
 - Device는 TCP Client mode로 동작
 - 사용자 응용에 따라 다양한 형태로 구현 가능
 - E.g., 주기적인 전송 / 이벤트 발생 시 전송 / ...



HTTP Client for W7500(P) GitHub Repository
https://github.com/hkjung/W7500/tree/httpClient_dev

실습 #6: S2E Customize 예제 구현(2) (cont'd)

>> HTTP Client Library: Header – Method / Content-Type

```
*****
* .HTTP·Method·String
*****
#define HTTP_GET ..... "GET"
#define HTTP_HEAD ..... "HEAD"
#define HTTP_POST ..... "POST"
#define HTTP_PUT ..... "PUT"
#define HTTP_DELETE ..... "DELETE"

*****
* .HTTP·Content-Type·String
*****
#define HTTP_CTYPE_MULTIPART_FORM ..... "multipart/form-data"
#define HTTP_CTYPE_APP_HTTP_FORM ..... "Application/x-www-form-urlencoded"
#define HTTP_CTYPE_APP_JS ..... "Application/javascript"
#define HTTP_CTYPE_APP_JSON ..... "Application/json"
#define HTTP_CTYPE_APP_XML ..... "Application/xml"
#define HTTP_CTYPE_TEXT_PLAIN ..... "text/plain"
#define HTTP_CTYPE_TEXT_HTML ..... "text/html"
#define HTTP_CTYPE_TEXT_CSS ..... "text/css"
#define HTTP_CTYPE_TEXT_JS ..... "text/javascript"
#define HTTP_CTYPE_TEXT_XML ..... "text/xml"
```

실습 #6: S2E Customize 예제 구현(2) (cont'd)

>> HTTP Client Library: Header – Request Structure / Flags

```
*****
* HTTP Request Structure Initializer
*****
#define HttpRequest_multipart_post_initializer {{(uint8_t *)HTTP_POST, NULL, NULL, (uint8_t *)HTTP_CTYPE_MULTIPART_FORM, (uint8_t *)"keep-alive", 0}}
#define HttpRequest_get_initializer {{(uint8_t *)HTTP_GET, NULL, NULL, NULL, (uint8_t *)"keep-alive", 0}}


*****
* HTTP Boundary String for Multipart/form-data
*****
#define formDataBoundary "----WebKitFormBoundaryE8pT6qqJHgRhSDDC" // example boundary string for multipart form data


// HTTP client structure
typedef struct __HttpRequest {
    uint8_t * method;
    uint8_t * uri;
    uint8_t * host;
    uint8_t * content_type;
    uint8_t * connection;
    uint32_t content_length;
} __attribute__((packed)) HttpRequest;

// HTTP client status flags
extern uint8_t httpc_isSockOpen;
extern uint8_t httpc_isConnected;
extern uint16_t httpc_isReceived;

// extern: HTTP request structure
extern HttpRequest request;
```

실습 #6: S2E Customize 예제 구현(2) (cont'd)

>> HTTP Client Library: Header – Functions

```
*****
* HTTP Client Functions
*****
uint8_t httpc_connection_handler(void); // HTTP client socket handler - for main loop, implemented in polling

uint8_t httpc_init(uint8_t sock, uint8_t * ip, uint16_t port, uint8_t * sbuf, uint8_t * rbuf); // HTTP client initialize
uint8_t httpc_connect(void); // HTTP client connect (after HTTP socket opened)
uint8_t httpc_disconnect(void);

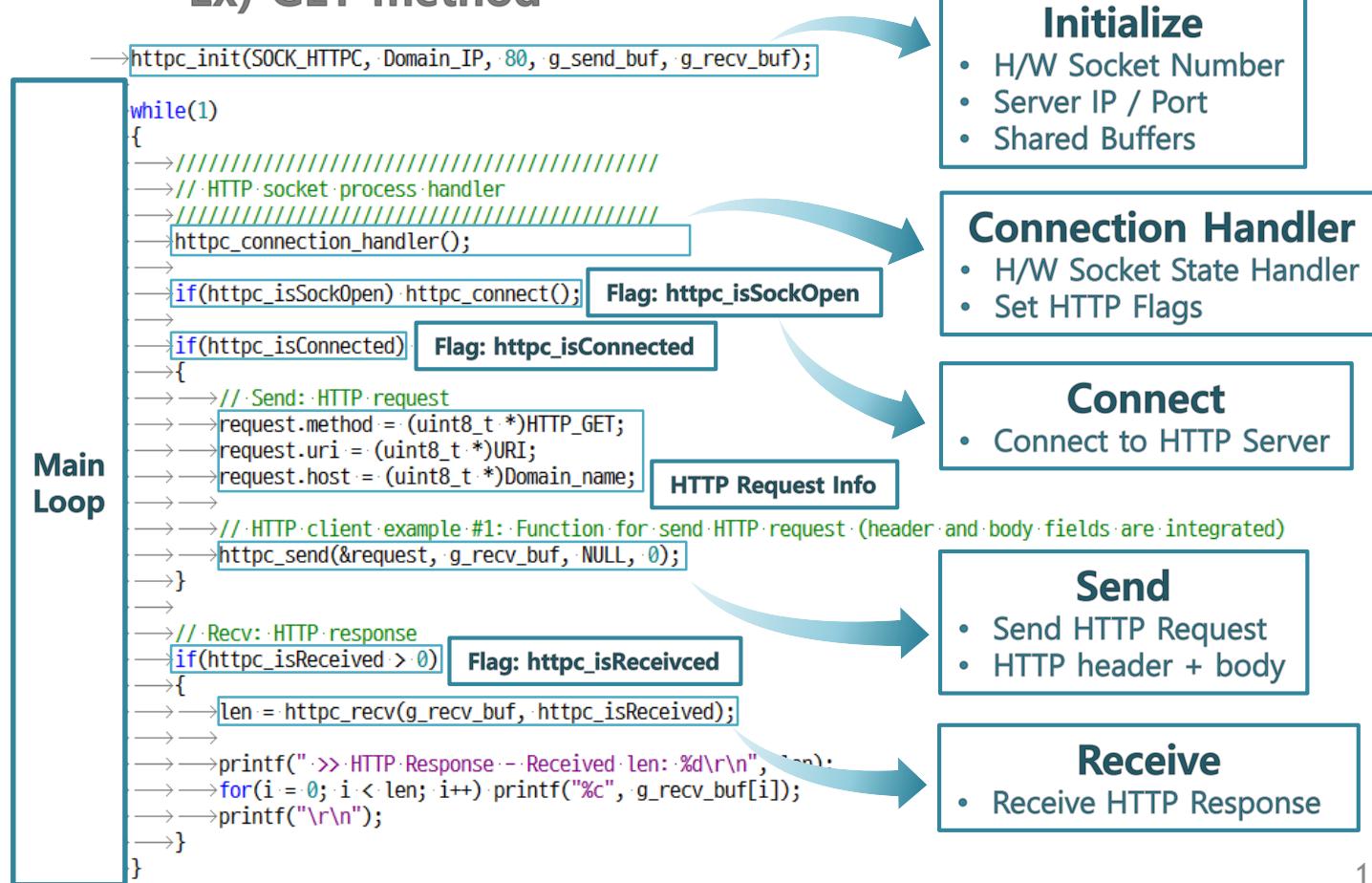
uint16_t httpc_add_customHeader_field(uint8_t * customHeader_buf, const char * name, const char * value); // Function for adding custom header fields (httpc
uint16_t httpc_send_header(HttpRequest * req, uint8_t * buf, uint8_t * customHeader_buf, uint16_t content_len); // Send the HTTP header only
uint16_t httpc_send_body(uint8_t * buf, uint16_t len); // Send the HTTP body only (have to send http request body after header sent)
uint16_t httpc_send(HttpRequest * req, uint8_t * buf, uint8_t * body, uint16_t content_len); // Send the HTTP header and body

uint16_t httpc_recv(uint8_t * buf, uint16_t len); // Receive the HTTP response header and body, User have to parse the received messages depending on needs
```

실습 #6: S2E Customize 예제 구현(2) (cont'd)

>> HTTP Client Library: Usage

- Ex) GET method



실습 #6: S2E Customize 예제 구현(2) (cont'd)

>> HTTP Client Library: Usage

- Ex) POST method

```

→ httpc_init(SOCK_HTTPC, Domain_IP, 80, g_send_buf, g_recv_buf);
→
→ while(1)
→ {
→   // HTTP socket process handler
→   httpc_connection_handler();
→
→   if(httpc_isSockOpen) httpc_connect();
→
→   if(httpc_isConnected)
→   {
→     // Send: HTTP request
→     request.method = (uint8_t *)HTTP_POST;
→     request.uri = (uint8_t *)URI;
→     request.host = (uint8_t *)Domain_name;
→
→     // HTTP client example #2: Separate functions for HTTP request - default header + body
→     httpc_send_header(&request, g_recv_buf, NULL, len);
→     httpc_send_body(g_send_buf, len); // Send HTTP request message body
→   }
→
→   // Recv: HTTP response
→   if(httpc_isReceived > 0)
→   {
→     len = httpc_recv(g_recv_buf, httpc_isReceived);
→
→     printf(" > HTTP Response -- Received len: %d\r\n", len);
→     for(i = 0; i < len; i++) printf("%c", g_recv_buf[i]);
→     printf("\r\n");
→   }
→ }
```

HTTP Request Info:

- POST method

Send HTTP Header

- with Content-Type
 - e.g., 'application/json'
- with Content-Length

Send HTTP Body

- e.g., JSON Data
 - {"power" : "on"}

실습 #6: S2E Customize 예제 구현(2) (cont'd)

>> HTTP Client Library: Usage

- Ex) Custom Header Fields

```
→ httpc_init(SOCK_HTTPC, Domain_IP, 80, g_send_buf, g_recv_buf);
→
→ while(1)
→ {
→ → // HTTP socket process handler
→ → httpc_connection_handler();
→ →
→ → if(httpc_isSockOpen) httpc_connect();
→ →
→ → if(httpc_isConnected)
→ → {
→ → → // Send: HTTP request
→ → → request.method = (uint8_t *)HTTP_POST;
→ → → request.uri = (uint8_t *)URI;
→ → → request.host = (uint8_t *)Domain_name;
→ → →
→ → → httpc_add_customHeader_field(tmpbuf, "Custom-Auth", "authkey"); // custom header field extended
→ → → httpc_add_customHeader_field(tmpbuf, "Auth-Key", "66W_vMFNYSUV6PL5op-nPXGDqSA="); // custom header field extended
→ → → httpc_send_header(&request, g_recv_buf, tmpbuf, len);
→ → → httpc_send_body(g_send_buf, len);
→ → }
→ →
→ → // Recv: HTTP response
→ → if(httpc_isReceived > 0)
→ → {
→ → → len = httpc_recv(g_recv_buf, httpc_isReceived);
→ → →
→ → → printf(" >> HTTP Response - Received len: %d\r\n", len);
→ → → for(i = 0; i < len; i++) printf("%c", g_recv_buf[i]);
→ → → printf("\r\n");
→ → }
→ }
```

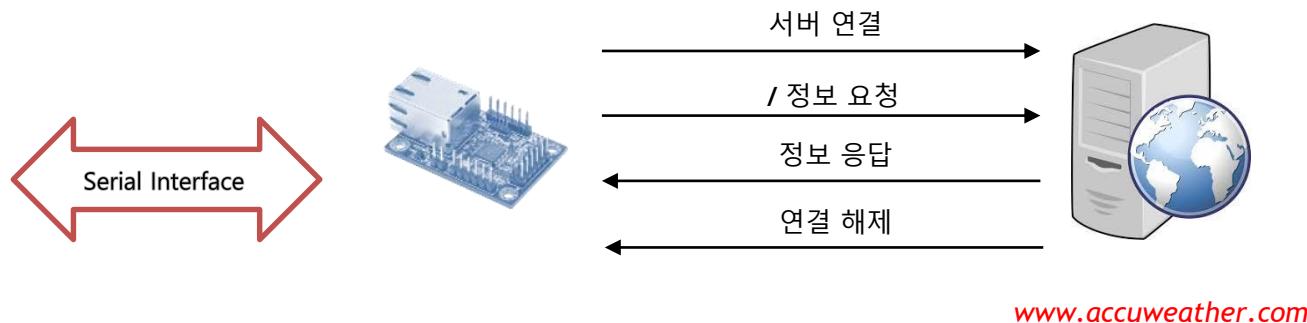
Custom Header Fields

- These functions are should be located before `httpc_send_header()` function

실습 #6: S2E Customize 예제 구현(2) (cont'd)

»» 위즈네트 www.wiznet.io 초기 페이지 가져오는 예제

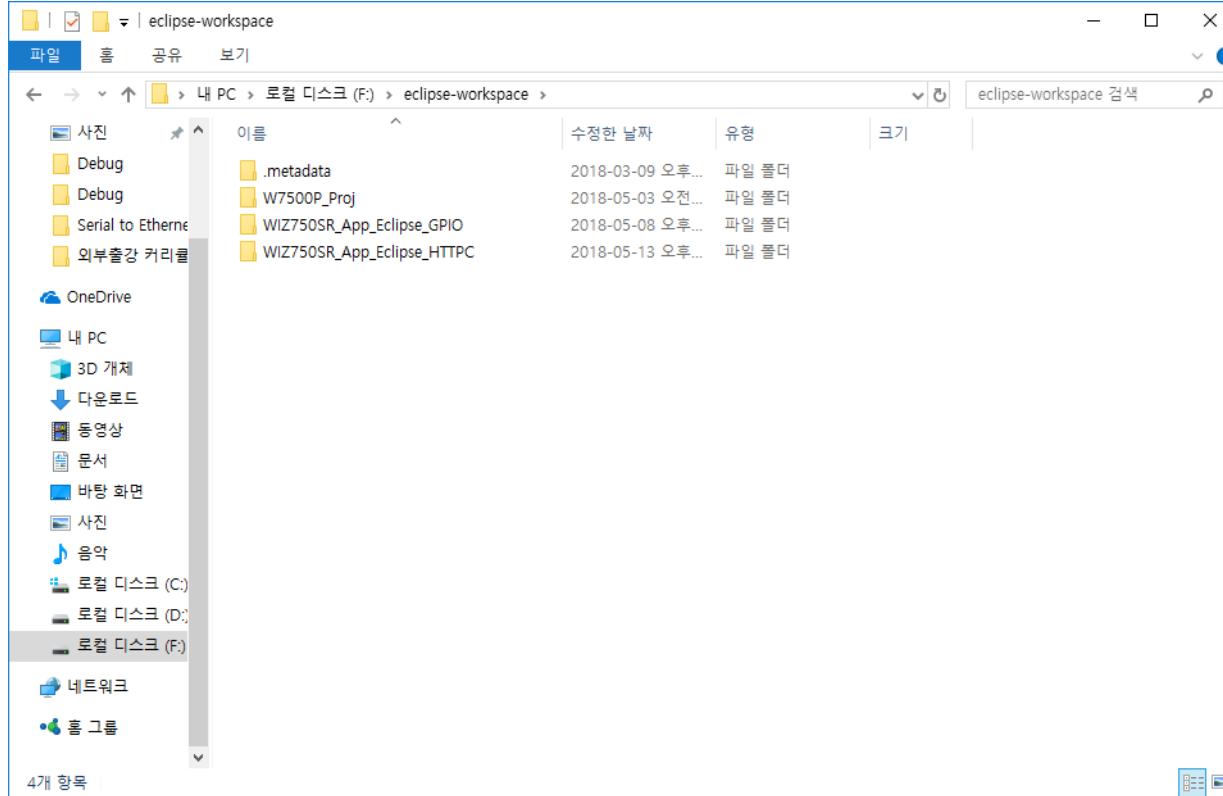
- 서버에서 웹 페이지 정보를 수신
- Console 창으로 출력



실습 #6: S2E Customize 예제 구현(1) (cont'd)

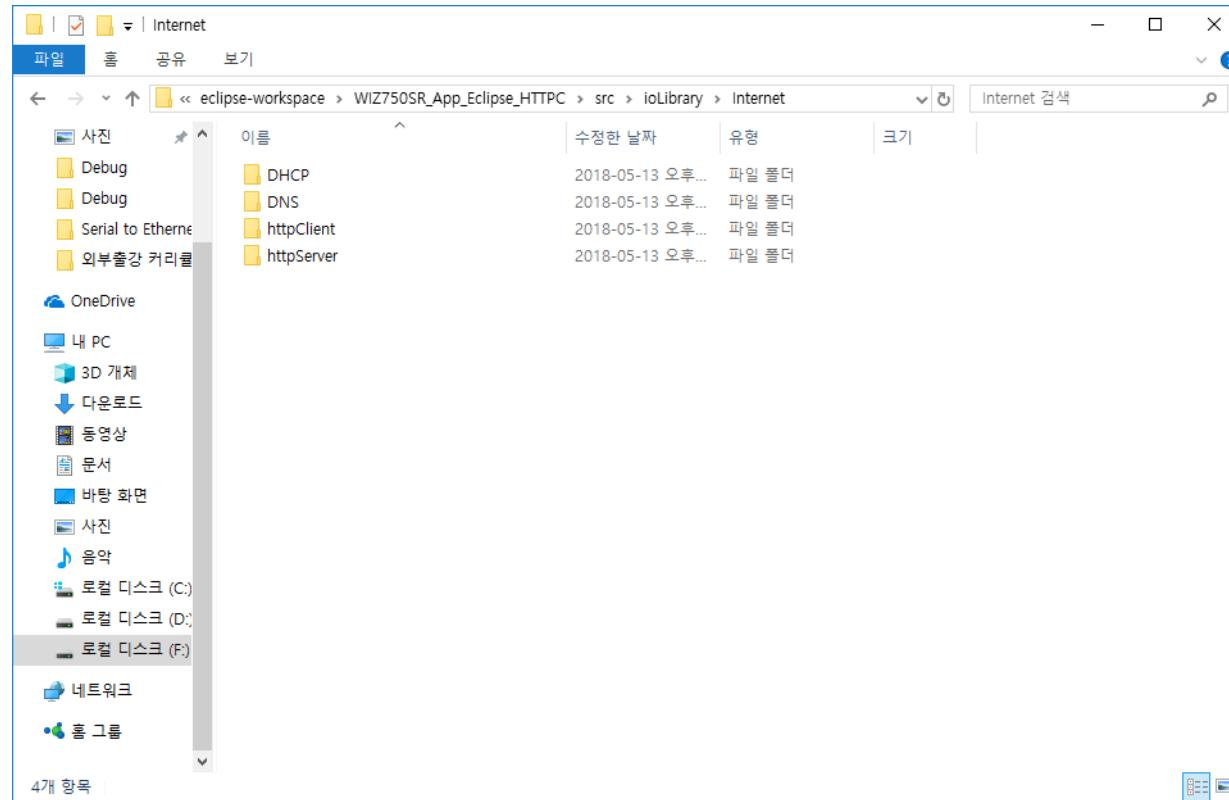
>> 실습 1단계 – Copy Base Project

- Github에서 다운로드한 WIZ750SR_App_Eclipse 폴더를 workspace로 복사
- 폴더명을 WIZ750SR_App_Eclipse_HTPC로 변경한다.



실습 #6: S2E Customize 예제 구현(1) (cont'd)

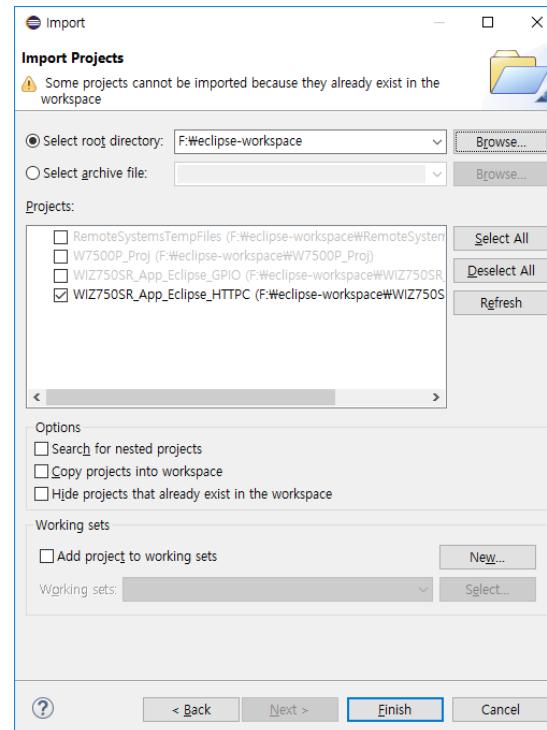
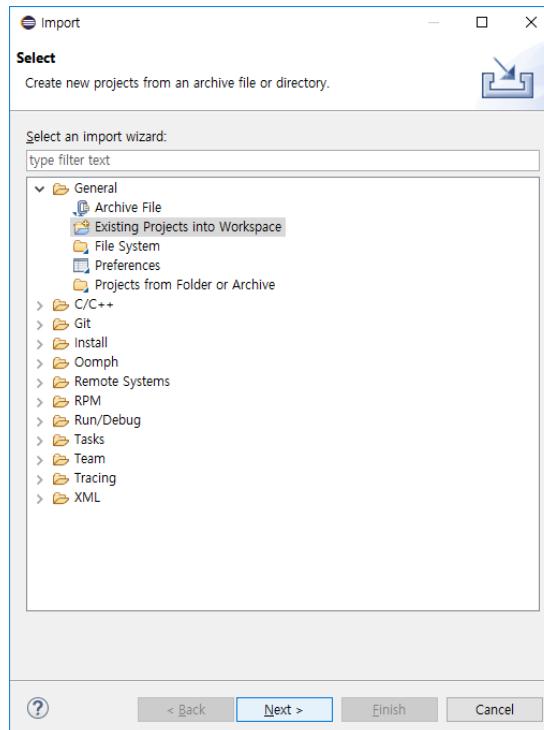
» 실습 2단계 – httpClient 폴더를 [src]-[ioLibrary]-[Internet]
하위에 복사한다.



실습 #6: S2E Customize 예제 구현(1) (cont'd)

>> 실습 3단계 – Import Project

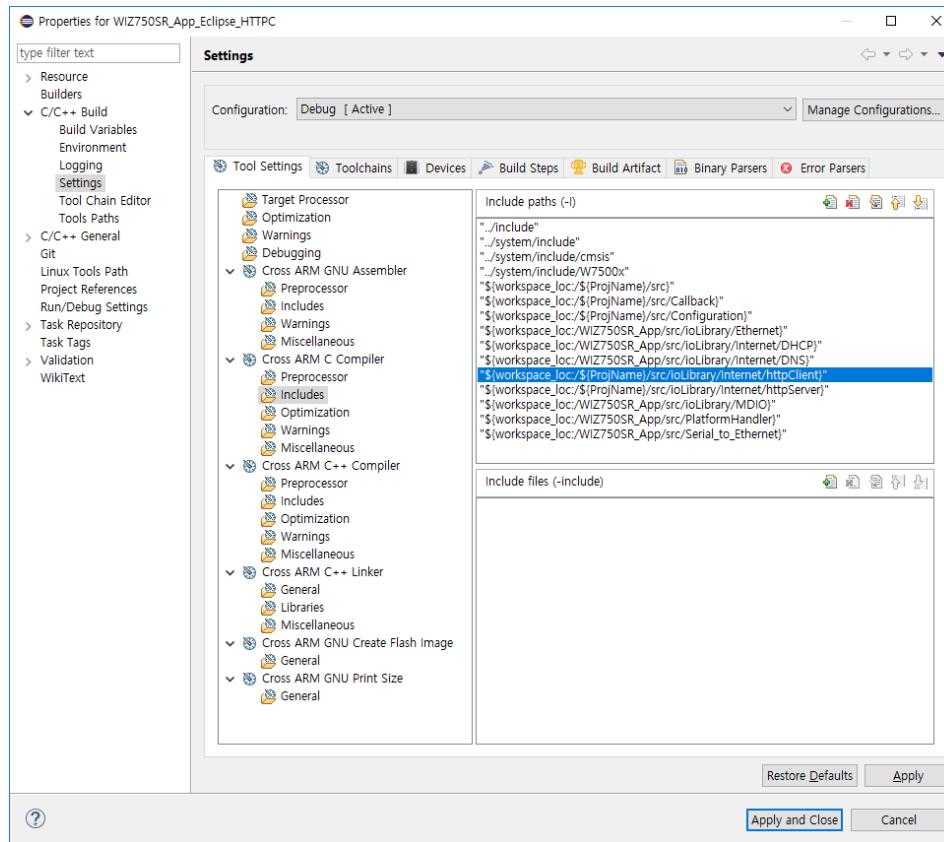
- Eclipse에서 [File]-[Import]를 선택한 후,
- [General]-“Existing Projects into Workspace”를 선택하고 “Next” 버튼을 클릭한다.
- “Select root directory”에 프로젝트 폴더를 옮긴 상위 폴더를 지정하고,
- [Projects] 리스트에서 복사한 프로젝트 폴더를 선택하고 “Finish” 버튼을 클릭한다.



실습 #6: S2E Customize 예제 구현(1) (cont'd)

» 실습 4단계 – httpClient 폴더를 Include Path에 지정한다.

- [Project]-[Properties] 선택한 후,
- [C/C++ Build]-[Settings]-[Tool Settings]-[Cross ARM C Compiler]-[Includes]를 선택하고, "httpClient"를 추가한다.



실습 #6: S2E Customize 예제 구현(1) (cont'd)

» 실습 5단계 – seg.c 수정

- #include "httpClient.h" 추가
- Domain Name 관련 변수 추가
- Proc_HTTP_client() 함수 추가
- do_seg 함수에 Proc_HTTP_clinet() 함수 호출 추가

```
#include "httpClient.h"
```

```
// Example domain name
uint8_t Domain_name[] = "www.wiznet.io";
uint8_t URI[] = "/";
uint8_t flag_sent_http_request = DISABLE;
```

```
/* Private functions prototypes -----
-----
void proc_SEG_tcp_client(uint8_t sock);
void proc_SEG_tcp_server(uint8_t sock);
void proc_SEG_tcp_mixed(uint8_t sock);
void proc_SEG_udp(uint8_t sock);
void proc_HTTP_client(uint8_t sock);
...
}
```

```
case TCP_CLIENT_MODE:
    //proc_SEG_tcp_client(sock);
    proc_HTTP_client(sock);
break;
```

```
void proc_HTTP_client(uint8_t sock)
{
    struct __options *option = (struct __options *)(&(get_DevConfig_pointer()->options));
    httpc_connection_handler();

    if(httpc_isSockOpen){
        httpc_connect();
    }
    if(httpc_isConnected){
        if(!flag_sent_http_request){
            // Send: HTTP request
            request.method = (uint8_t *)HTTP_GET;
            request.uri = (uint8_t *)URI;
            request.host = (uint8_t *)option->dns_domain_name;

            httpc_send(&request, g_recv_buf, g_send_buf, 0);
            flag_sent_http_request = ENABLE;
        }
        if(httpc_isReceived > 0){
            uint16_t len = httpc_recv(g_recv_buf, httpc_isReceived);

            printf(" >> HTTP Response - Received len: %d\r\n", len);
            printf("======\r\n");
            for(uint16_t i = 0; i < len; i++) printf("%c", g_recv_buf[i]);
            printf("\r\n");
            printf("======\r\n");
        }
    }
}
```

실습 #6: S2E Customize 예제 구현(1) (cont'd)

>> 실습 6단계 – main.c 수정

- #include "httpClient.h" 추가
- While() loop 전에 httpc_init() 함수 호출 추가

```
#include "httpClient.h"

...
httpc_init(SOCK_DATA, dev_config->network_info[0].remote_ip, 80, g_send_buf, g_recv_buf);

while(1) // main loop
{
    do_segcp();
    do_seg(SOCK_DATA);
...
}
```

실습 #6: S2E Customize 예제 구현(1) (cont'd)

>> 실습 7단계 – Build and Download

- Eclipse에서 [Project]-[Build Project]를 선택해서 Binary를 생성한다.
- WIZnet Config Tool을 이용해서 Ethernet으로 Binary를 Download 한다.



감사합니다

WIZnet Academy 2018

Serial to Ethernet Deep-Dive Training

Thank you

James Kim

 jameskim@wiznet.io