

프로젝트 명세서

LwM2M 프로토콜 오픈소스 빌드 및 실습

목차

1. 프로젝트 개요..... 3

2. 과제 목표 4

3. 참고 자료 5

4. 과제 7

5. 산출물 제출 13

1. 프로젝트 개요

소개

본 프로젝트에서는 IoT 프로토콜 중 하나인 LwM2M의 Open source 중 C/C++ 기반 wakaama project를 Github에서 다운로드 받아 리눅스 환경에서 프로젝트 빌드 및 Client, Sever 간 연동을 실습하는 과제입니다.

이 과제를 통해서 교육생들은 요구사항을 분석하여 LwM2M 오픈소스로 IoT 디바이스가 Client, Server 간 통신하는 프로세스를 실습합니다.

2. 과제 목표

Windows 10 환경에서 WSL 을 설치한 뒤, ubuntu 를 추가 설치하여 임베디드 개발이 가능한 리눅스 환경을 만들어 봅니다.

Ubuntu 를 실행한 뒤 오픈 소스를 빌드 할 수 있도록 준비합니다.

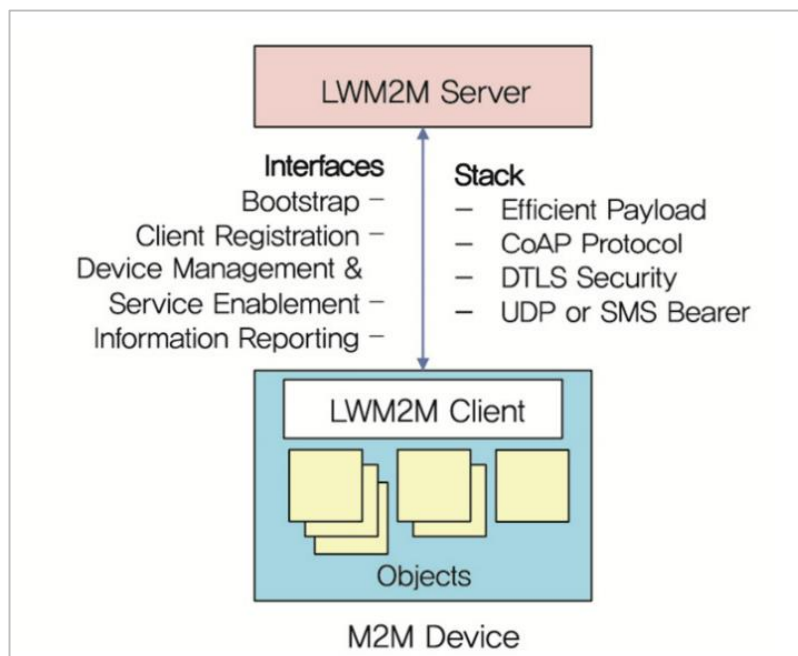
Github 에서 LwM2M 오픈소스 wakaama 를 clone 하여 server, client 실행 파일을 각각 빌드하여 만들어 봅니다.

오픈소스는 server, client 가 모두 포함되어 있으므로, 동일한 소스를 server, client 로 다운로드하여 과제를 진행해야 합니다.

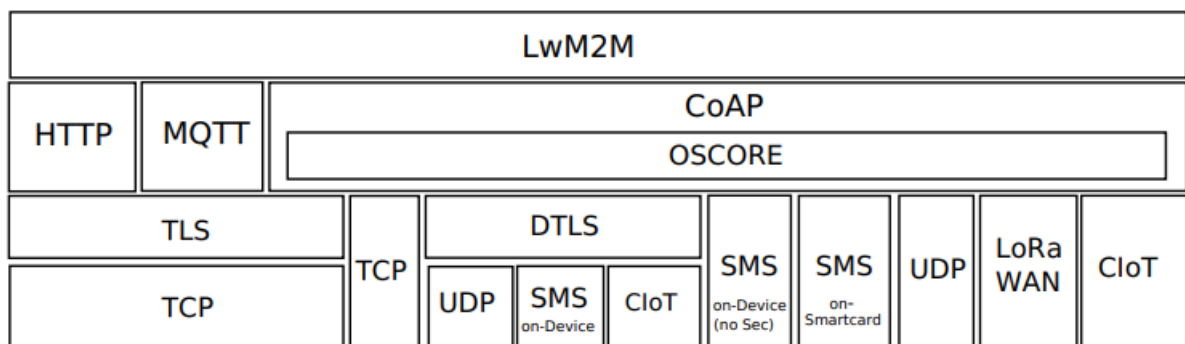
(소스를 한번만 받아 copy 하여 사용해도 무관합니다.)

3. 참고 자료

- LwM2M (Light Weight Machine To Machine)
 - 1) M2M 또는 IoT 장치 관리 및 서비스 구현을 위한 오픈 모바일 연합(OMA)의 프로토콜
 - 2) LwM2M 표준은 IoT 장치에 있는 LwM2M 클라이언트와 LwM2M 서버 간의 응용 프로그램 계층 통신 프로토콜을 정의
 - 3) CoAP 프로토콜과 보안을 위해 DTLS(Datagram TLS) 사용
 - 4) 메시지 전달은 TCP, UDP, SMS 등을 통해 전달
- LwM2M 아키텍처



- LwM2M 구조



⇒ 참고 사이트: [Lightweight M2M \(LWM2M\) - OMA SpecWorks](#)

- LwM2M 오픈 소스

1) Wakaama project (C/C++)

⇒ <https://github.com/eclipse/wakaama>

4. 과제

본 과제는 리눅스 환경 구성과 LwM2M 오픈소스를 빌드하여 테스트한 실습 내용을 캡처하여 제출해야 합니다.

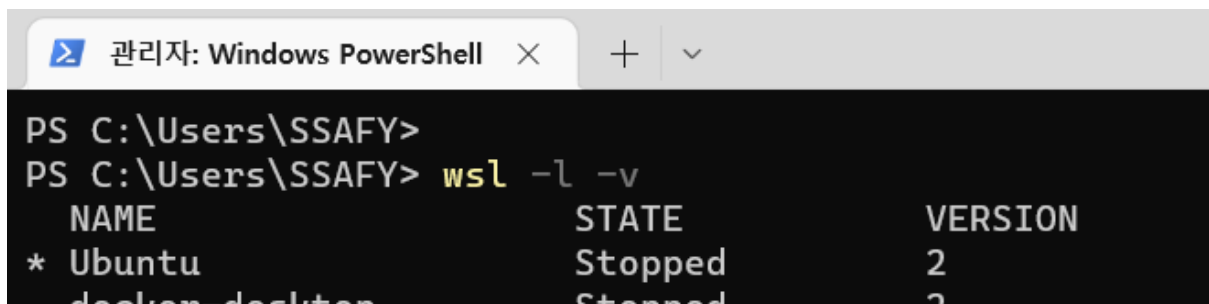
1) 리눅스 환경 구성

- a. WSL2 를 설치
- b. Ubuntu 를 설치
- c. “wsl -l -v” 명령으로 Ubuntu Version 2 확인

⇒ <https://learn.microsoft.com/ko-kr/windows/wsl/install>

⇒ <https://www.youtube.com/watch?v=anZmL2bs-xY>

(WSL2 + VSCode = 운영체제 및 시스템 프로그래밍 공부를 위한 가장 쉽고 편리한 개발환경)



The screenshot shows a Windows PowerShell window titled '관리자: Windows PowerShell'. The command 'wsl -l -v' has been executed, displaying a table of installed Linux distributions.

NAME	STATE	VERSION
* Ubuntu	Stopped	2
docker-desktop	Stopped	2

⇒ ubuntu 를 실행하여 접속

```
kosh@DESKTOP-KVCQHCD: x + v
PS C:\Users\SSAFY>
PS C:\Users\SSAFY> wsl -l -v
  NAME                STATE          VERSION
* Ubuntu              Stopped        2
  docker-desktop      Stopped        2
  docker-desktop-data Stopped        2
PS C:\Users\SSAFY> ubuntu
이제 Linux용 Windows 하위 시스템을 Microsoft Store에서 사용할 수 있습니다.
'wsl.exe --update'를 실행하거나 https://aka.ms/wslstorepage를 방문하여 업그레이드할 수 있습니다.
Microsoft Store의 WSL을 설치하면 최신 WSL 업데이트가 더 빠르게 제공됩니다.
자세한 내용은 https://aka.ms/wslstoreinfo를 방문하세요.

Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.10.102.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/kosh/.hushlogin file.
kosh@DESKTOP-KVCQHCD:~$ |
```

⇒ CMake 빌드 도구와 라이브러리 설치

“wakaama” project 는 c/c++로 구현되어 있으며, cmake로 makefile 을 생성하여 빌드하도록 만들어져 있습니다.

ubuntu 에서 cmake, gcc 를 사용할 수 있는 환경을 구성해 보십시오.

❏ sudo apt-get install build-essential libssl-dev

2) LwM2M 오픈 소스 실습

⇒ LwM2M 오픈 소스 “wakaama” project 의 최신 branch clone 하기

⇒ 오픈 소스에 Client, Server 소스가 모두 포함되어 있어, 아래 그림과 같이 Client, Server 를 두 번 Clone 하거나, Clone 받은 소스를 copy 하여 Client, Server 로 분리해야 합니다.

※ git clone -recurse-submodules <https://github.com/eclipse/wakaama.git>

```
kosh@DESKTOP-KVCQHCD: ~/lwm2m_test$ git clone --recurse-submodules https://github.com/eclipse/wakaama.git lwm2m_server
Cloning into 'lwm2m_server'...
remote: Enumerating objects: 7694, done.
remote: Counting objects: 100% (702/702), done.
remote: Compressing objects: 100% (298/298), done.
remote: Total 7694 (delta 410), reused 518 (delta 337), pack-reused 6992
Receiving objects: 100% (7694/7694), 2.98 MiB | 11.46 MiB/s, done.
Resolving deltas: 100% (5400/5400), done.
Submodule 'platforms/Linux/tinydtls' (https://github.com/eclipse/tinydtls.git) registered for path 'examples/shared/tinydtls'
Cloning into '/home/kosh/lwm2m_test/lwm2m_server/examples/shared/tinydtls'...
remote: Enumerating objects: 3853, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 3853 (delta 7), reused 11 (delta 4), pack-reused 3832
Receiving objects: 100% (3853/3853), 1.21 MiB | 1.31 MiB/s, done.
Resolving deltas: 100% (2706/2706), done.
Submodule path 'examples/shared/tinydtls': checked out '178b945e5497b813b1239e2fbf57e34b7488745f'
kosh@DESKTOP-KVCQHCD:~/lwm2m_test$ ls
lwm2m_server
```

```
kosh@DESKTOP-KVCQHCD: ~/lwm2m_test$ git clone --recurse-submodules https://github.com/eclipse/wakaama.git lwm2m_client
Cloning into 'lwm2m_client'...
remote: Enumerating objects: 7694, done.
remote: Counting objects: 100% (702/702), done.
remote: Compressing objects: 100% (298/298), done.
remote: Total 7694 (delta 410), reused 518 (delta 337), pack-reused 6992
Receiving objects: 100% (7694/7694), 2.98 MiB | 4.58 MiB/s, done.
Resolving deltas: 100% (5400/5400), done.
Submodule 'platforms/Linux/tinydtls' (https://github.com/eclipse/tinydtls.git) registered for path 'examples/shared/tinydtls'
Cloning into '/home/kosh/lwm2m_test/lwm2m_client/examples/shared/tinydtls'...
remote: Enumerating objects: 3853, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 3853 (delta 7), reused 11 (delta 4), pack-reused 3832
Receiving objects: 100% (3853/3853), 1.21 MiB | 1.55 MiB/s, done.
Resolving deltas: 100% (2706/2706), done.
Submodule path 'examples/shared/tinydtls': checked out '178b945e5497b813b1239e2fbf57e34b7488745f'
kosh@DESKTOP-KVCQHCD:~/lwm2m_test$ ls
lwm2m_client  lwm2m_server
kosh@DESKTOP-KVCQHCD:~/lwm2m_test$ |
```

※ 소스 구조

+ core	(the LWM2M engine)
+ coap	(CoAP stack adaptation)
+ er-coap-13	(Modified Erbium's CoAP engine from https://web.archive.org/web/20180316172739/http://people.inf.ethz.ch/mkovatsc/erbium.php)
+ data	(data formats serialization/deserialization)
+ tests	(test cases)
+ integration	(pytest based integration tests implementing the OMA-ETS-LightweightM2M-V1_1-20190912-D specification https://www.openmobilealliance.org/release/LightweightM2M/ETS/OMA-ETS-LightweightM2M-V1_1-20190912-D.pdf)
+ examples	
+ bootstrap_server	(a command-line LWM2M bootstrap server)
+ client	(a command-line LWM2M client with several test objects)
+ lightclient	(a very simple command-line LWM2M client with several test objects)
+ server	(a command-line LWM2M server)
+ shared	(utility functions for connection handling and command-line interface)

⇒ Server, client 소스 Ubuntu 창을 두 개 생성하여 각각 빌드하여 실행파일 만들기
(wakaama project Github README.md 참고하여 수행)

```
kosh@DESKTOP-KVCQHCD: ~/lwm2m_test/lwm2m_server$ cmake examples/server
-- The C compiler identification is GNU 11.3.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- lwm2mserver: Endiannes not set, defaulting to little endian
-- lwm2mserver: Default CoAP block size not set, using 1024
-- Configuring done
-- Generating done
-- Build files have been written to: /home/kosh/lwm2m_test/lwm2m_server
kosh@DESKTOP-KVCQHCD: ~/lwm2m_test/lwm2m_server$ make lwm2mserver

[100%] Linking C executable lwm2mserver
[100%] Built target lwm2mserver
```

```
kosh@DESKTOP-KVCQHCD: ~/lwm2m_test/lwm2m_client$ cmake examples/client
-- The C compiler identification is GNU 11.3.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- lwm2mclient: Endiannes not set, defaulting to little endian
-- lwm2mclient: Default CoAP block size not set, using 1024
-- lwm2mclient_tinydtls: Endiannes not set, defaulting to little endian
-- lwm2mclient_tinydtls: Default CoAP block size not set, using 1024
-- Found Git: /usr/bin/git (found version "2.34.1")
-- Configuring done
-- Generating done
-- Build files have been written to: /home/kosh/lwm2m_test/lwm2m_client
kosh@DESKTOP-KVCQHCD: ~/lwm2m_test/lwm2m_client$ make lwm2mclient

[100%] Linking C executable lwm2mclient
[100%] Built target lwm2mclient
```

⇒ 생성된 실행 파일을 Server, Client 순서로 실행하여 연동 확인.

```
kosh@DESKTOP-KVCQHCD: ~/lwm2m_test/lwm2m_server$ ls
CMakeCache.txt  CONTRIBUTING.md  Makefile  SECURITY.md  cmake_install.cmake  data  lwm2mserver  wakaama.cmake
CMakeFiles      LICENSE-epl-v2.0.md  NOTICE.md  TODO  coap  examples  tests
CMakeLists.txt  LICENSE.edl-v1.0.md  README.md  about.html  core  include  tools
kosh@DESKTOP-KVCQHCD: ~/lwm2m_test/lwm2m_server$ ./lwm2mserver -help
Usage: lwm2mserver [OPTION]
Launch a LWM2M server on localhost.

Options:
  -4          Use IPv4 connection. Default: IPv6 connection
  -l PORT     Set the local UDP port of the Server. Default: 5683
  -S BYTES    CoAP block size. Options: 16, 32, 64, 128, 256, 512, 1024. Default: 1024

kosh@DESKTOP-KVCQHCD: ~/lwm2m_test/lwm2m_server$ ./lwm2mserver
> help
help  Type 'help [COMMAND]' for more details on a command.
list  List registered clients.
read  Read from a client.
disc  Discover resources of a client.
write Write to a client.
update Write to a client with partial update.
time  Write time-related attributes to a client.
attr  Write value-related attributes to a client.
clear Clear attributes of a client.
exec  Execute a client resource.
del   Delete a client Object instance.
create Create an Object instance.
observe Observe from a client.
cancel Cancel an observe.
q     Quit the server.

> list
No client.

> |
```

```
> list
No client.

> 195 bytes received from [::1]:56830
44 02 B2 3C 3C B2 E0 33 B2 72 64 11 28 39 6C 77 D..<..3.rd.(9lw
6D 32 6D 3D 31 2E 31 0D 05 65 70 3D 74 65 73 74 m2m=1.1..ep=test
6C 77 6D 32 6D 63 6C 69 65 6E 74 03 62 3D 55 06 lwm2mclient.b=U.
6C 74 3D 33 30 30 FF 3C 2F 3E 3B 72 74 3D 22 6F lt=300.</>;rt="o
6D 61 2E 6C 77 6D 32 6D 22 3B 63 74 3D 31 31 30 ma.lwm2m";ct=110
2C 3C 2F 31 3E 3B 76 65 72 3D 31 2E 31 2C 3C 2F ,</1>;ver=1.1,</
31 2F 30 3E 2C 3C 2F 32 2F 30 3E 2C 3C 2F 33 2F 1/0>,</2/0>,</3/
30 3E 2C 3C 2F 34 2F 30 3E 2C 3C 2F 35 2F 30 3E 0>,</4/0>,</5/0>
2C 3C 2F 36 2F 30 3E 2C 3C 2F 37 2F 30 3E 2C 3C ,</6/0>,</7/0>,<
2F 33 31 30 32 34 3E 3B 76 65 72 3D 31 2E 30 2C /31024>;ver=1.0,
3C 2F 33 31 30 32 34 2F 31 30 3E 2C 3C 2F 33 31 </31024/10>,</31
30 32 34 2F 31 31 3E 2C 3C 2F 33 31 30 32 34 2F 024/11>,</31024/
31 32 3E 12>

New client #0 registered.
Client #0:
  name: "testlwm2mclient"
  version: "1.1"
  binding: "UDP"
  lifetime: 300 sec
  objects: /1 (1.1), /1/0, /2/0, /3/0, /4/0, /5/0, /6/0, /7/0, /31024 (1.0), /31024/10, /31024/11, /31024/12,
>
> list
Client #0:
  name: "testlwm2mclient"
  version: "1.1"
  binding: "UDP"
  lifetime: 300 sec
  objects: /1 (1.1), /1/0, /2/0, /3/0, /4/0, /5/0, /6/0, /7/0, /31024 (1.0), /31024/10, /31024/11, /31024/12,
```

위 그림은 Server 를 실행한 뒤, Client 실행하여 연동된 상태를 캡처 하였습니다.

Client 를 실행한 상태도 캡처 하여 제출해 주십시오.

5. 산출물 제출

1) 제출 위치:

<https://lab.ssafy.com/s10-study/self-project/> 의 “산출물 제출 가이드.docx” 참조

※ 챕터 4 과제에 캡처 된 그림을 참고하여 학습한 내용을 제출하십시오.