

임베디드 SW 경진대회: 기술교육

webOS OSE 빌드 환경 구성



목차

- 1. 빌드 환경 소개
- 2. webOS OSE 빌드 환경 구성
- 3. webOS OSE 빌드
- 4. webOS OSE 플래싱

빌드 환경 소개



🐧 빌드 시스템

- Embedded Linux가 보편화 되면서 다양한 Cross Platform에 맞는 Build 요구사항 증가
- Kernel, Toolchain등 타겟에 맞게 rootfs(루트 파일 시스템) 이미지를 만들어 주는 Build 툴

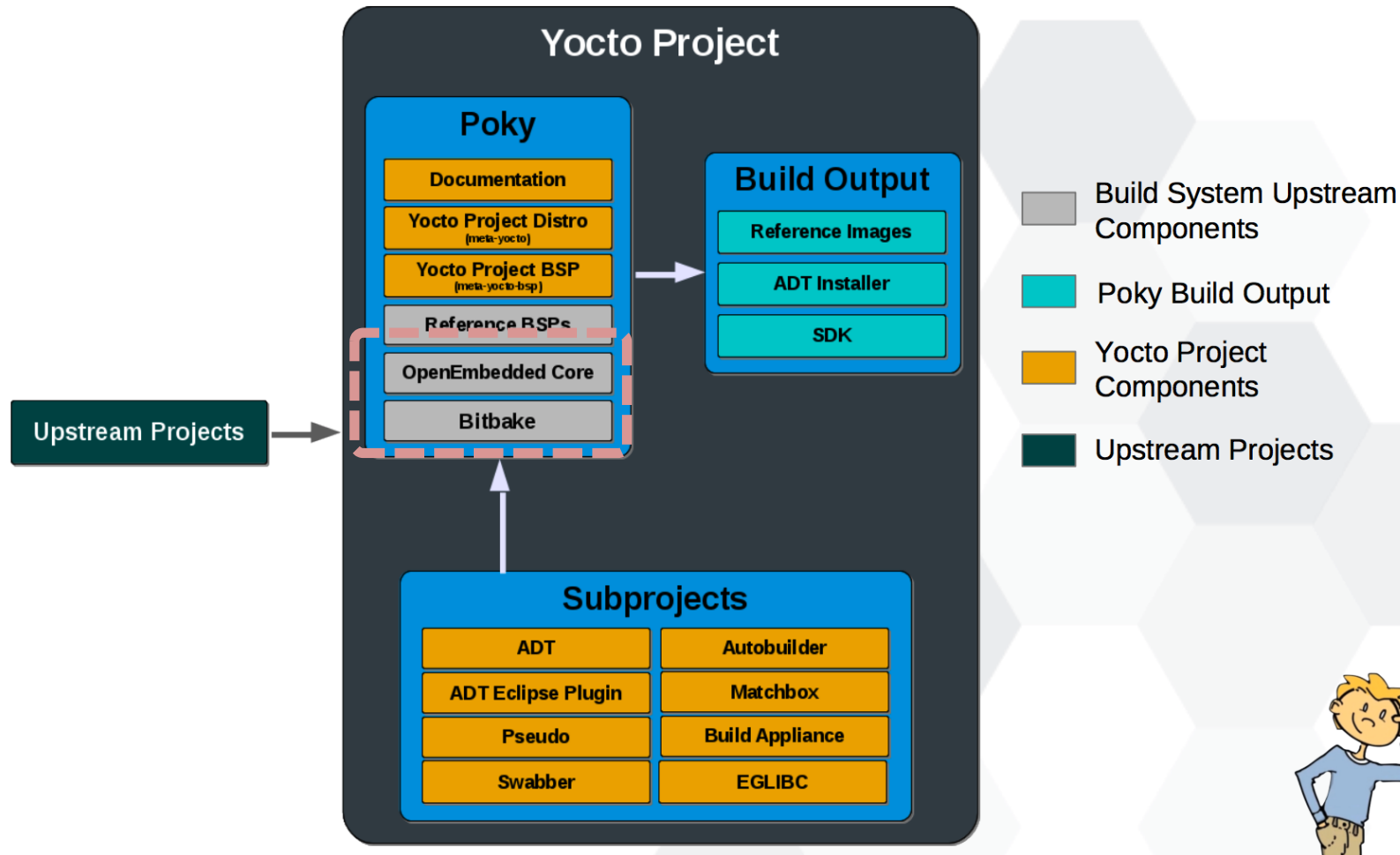
개발 환경

Yocto 프로젝트

- Linux Foundation
- 오픈 소스 협업 프로젝트 (Open Source Collaboration Project)
- 하드웨어 아키텍처(x86, ARM, MIPS, Power)에 상관없이
임베디드 제품에 대한 Linux 기반 SW 시스템을 개발자가 직접 만들 수 있도록 지원

개발 환경

우리는 Yocto 가족...



개발 환경

🐙 OpenEmbedded

- 임베디드 리눅스 배포한 구축을 위한 빌드 프레임워크
- 2003년 1월에 시작
- OE Classic에서 OE Core로 발전됨
- Python 기반의 Bitbake 빌드 툴을 사용함
- OE Core 부분을 Yocto Project와 공유함
- 다양한 Architecture, Feature Application을 제공
- 많은 형상관리 Protocol 지원: git, svn, cvs, http, ftp, ...



개발 환경

🔗 OpenEmbedded

- BitBake

- 임베디드 리눅스의 크로스 컴파일 과정을 위한 패키지과 관련파일들을 빌드하는데 사용되는 툴

- Recipe

- Bitbake는 Recipe를 참고하여 Task들을 관리, 실행



- Metadata

- Recipe의 구조화된 모음

개발 환경

BitBake

- Python과 Shell Script로 코드를 실행하는 태스크 스케줄러
(GNU make와 비슷하다고 이해하면 됨)
- 병렬 실행 지원
- OpenEmbedded와 Yocto Project에서 '빌드'에 사용함
- 메타 데이터 (.conf / .bbclass / .bb)

BitBake에 의한 빌드 작업의 수행은 ...

메타 데이터 (.conf, .bbclass, .bb & .inc)를 기반으로
'패키지'간의 'dependency'를 분석하여 작업 그래프를
만들고, 각 패키지를 빌드하고 통합하여 이미지를 생성함

개발 환경

Recipe

- 특정패키지에 대한 빌드방법을 구체화
- 패키지들의 의존성, 위치, 설정법, 컴파일, 빌드, 설치 및 제거정보를 포함
- 패키지의 메타데이터 정보를 표준화된 변수들로 가지고 있음
- .bb 파일 형식

webOS OSE 개발 환경

🐧 webOS OSE로 빌드하는 방법

```
$ git clone https://github.com/webosose/build-webos
```

```
$ cd build-webos
```

```
$ ./mcf raspberrypi3 or qemux86
```

```
$ source oe-init-build-env
```

```
$ bitbake webos-image
```

webOS OSE 개발 환경

🐧 SD Card에 플래싱 하는 방법

Image Location: BUILD/deploy/images/

Windows: Use Win32DiskImager

Linux & MacOS: Use “dd” command

webOS OSE 빌드 환경 구성



시스템 요구사항

장치 요구사항

- 라즈베리파이 3 (Model B) 보드




CPU: 1.2GHz 64-bit
RAM: 1 GB
LAN: 802.11n Wireless LAN
Bluetooth: 4.0 Support

빌드 시스템 요구사항

- Linux - Ubuntu 14.04 LTS 64-bit

	Minimum requirements	Recommended
CPU	Intel i5 dual-core with 4 threads	Intel i7 quad-core with 8 threads or higher
RAM	8 GB	16 GB or higher
Storage	HDD with 100 GB of free disk space	SSD with 100 GB of free disk space or more



Windows나 macOS에서
Linux virtual machine으로
빌드가능 ?

NO

시스템 요구사항

🐧 SW 도구

- Git

- `$ git --help` 스크립트 입력하여 확인
- 확인불가시: <https://help.github.com/articles/set-up-git>
또는 `$ sudo apt-get install git`

- Python

- `$ python3 --version` 스크립트 입력하여 확인
- 확인불가시 <https://www.python.org/>
또는 `$ sudo apt-get install python3`

webOS OSE 빌드



webOS OSE 빌드 따라하기

🐙 Repository 복사해오기

- 작업 디렉토리 생성
 - `$ mkdir work` 스크립트 입력
 - `$ cd work` (작업디렉토리로 이동)
- Git Repository에서 build-webOS 복사해오기
 - `work$ git clone https://github.com/webosose/build-webos.git`
- 설치요소 확인
 - `work$ cd build-webos` (디렉토리 이동)
 - `work/build-webos$ sudo scripts/prerequisites.sh`

webOS OSE 빌드 따라하기

🐼 \$ vi scripts/prerequisites.sh

```
apt-get update
```

```
# These are essential to pass OE sanity test
```

```
# locales, because utf8 is needed with newer bitbake which uses python3
```

```
essential="\
```

```
    build-essential \
```

```
    chrpath \
```

```
    cpio \
```

```
    diffstat \
```

```
    gawk \
```

```
    git \
```

```
    iputils-ping \
```

```
    locales \
```

```
    lsb-release \
```

```
    python \
```

```
    python2.7 \
```

```
    python3 \
```

```
    texinfo \
```

```
    wget \
```

```
"
```

- (vi종료) :q

webOS OSE 빌드 따라하기

🐧 타겟 디바이스 선택하여 빌드 구성

- QEMU

- work/build-webos 경로 확인
- `work/build-webos$./mcf qemux86` 스크립트 입력



라즈베리파이3을 타겟으로 잡으려면 ?

```
$ ./mcf raspberrypi3
```

🐧 이미지 빌드

- Full Build

- `work/build-webos$ source oe-init-build-env`
- `work/build-webos$ bitbake webos-image`

= `work/build-webos$ make webos-image`



나는...디버깅을 하면서 개발하고 싶은데...
GDB 디버거 같은 툴을 포함해서 빌드할
수는 없나요?

```
$ source oe-init-build-env
```

```
$ bitbake webos-image-devel
```

webOS OSE 빌드 따라하기

🔗 타겟 디바이스 선택하여 빌드 구성

- QEMU

- work/build-webos 경로 확인
- `work/build-webos$./mcf qemux86` 스크립트 입력



라즈베리파이3을 타겟으로 잡으려면 ?

```
work/build-webos$ ./mcf raspberrypi3
```



mcf의 옵션을 지정하지 않으면 기본적으로 Build 할 Host PC의 자원을 최대한으로 사용함

호스트 코어의 2배수 권장

- b : BB_NUMBER_THREADS, 동시에 실행되는 BitBake 스레드의 최대 개수
- p : PARALLEL_MAKE, 동시에 실행되는 컴파일 작업 수

webOS OSE 빌드 따라하기

🔥 빌드 고고 !!

```
Currently 28 running tasks (3588 of 6204) 57% |#####|
0: glibc-locale-2.24-r0 do_package_write_ipk - 262s (pid 10039)
1: libtool-cross-2.4.6-r0 do_configure - 33s (pid 30908)

3: node-inspector-0.7.5-r2 do_compile - 17s (pid 9460)
4: openssl-1.0.2j-r0 do_compile - 16s (pid 9937)
5: gcc-runtime-6.4.0-r0 do_package_write_ipk - 12s (pid 14047)
6: linux-yocto-1_4.8.26+gitAUTOINC+1c60e003c7_27efc3ba68-r0 do_unpack - 9s (pid 17374)
7: rosidl-generator-cpp-git-r0 do_package_write_ipk - 2s (pid 24671)
8: fastcdr-git-r0 do_package_write_ipk - 2s (pid 24807)
9: c-ares-1.13.0+gitrAUTOINC+3be1924221-r0 do_package - 2s (pid 25393)
10: rmw-implementation-cmake-git-r0 do_package_write_ipk - 2s (pid 25961)
11: python-cmake-module-git-r0 do_package_write_ipk - 2s (pid 26168)
12: libaio-0.3.110-r0 do_package_write_ipk - 2s (pid 26211)
13: jemalloc-0.20080828a-0webos9-1-r1 do_package_write_ipk - 1s (pid 26651)
14: configd-data-1.0.0-1-r1 do_package_write_ipk - 1s (pid 27304)
15: pps-tools-0.0.0+gitAUTOINC+0deb9c7e13-r0 do_package_write_ipk - 1s (pid 27399)
16: connman-conf-1.0-r2 do_package_write_ipk - 1s (pid 27622)
17: busybox-1.24.1-r0 do_compile - 1s (pid 27295)
18: libsandbox-2.0.0-1-r2 do_package_write_ipk - 1s (pid 27698)
```

webOS OSE 빌드 따라하기

🔗 빌드 결과 파일

```
kw.choi@sdk03:~/test/build-webos/BUILD/deploy/images/qemux86$ ls
bzImage-linux-yocto-qemux86.bin
webos-image-qemux86.hdddirect
webos-image-qemux86-master-20181030005332.ext4
webos-image-qemux86-master-20181030005332.hdddirect
webos-image-qemux86-master-20181030005332.manifest
webos-image-qemux86-master-20181030005332.qemuboot.conf
webos-image-qemux86-master-20181030005332.tar.gz
webos-image-qemux86-master-20181030005332.vmdk
webos-image-qemux86.qemuboot.conf
webos-image-qemux86.rootfs.ext4
webos-image-qemux86.rootfs.manifest
webos-image-qemux86.rootfs.tar.gz
webos-image-qemux86.vmdk
```

→ Pre-built Emulator Package 세션에서...

```
kw.choi@sdk03:~/test/build-webos/BUILD/deploy/images/qemux86$ ~
```

webOS OSE 빌드 따라하기

클린 빌드 (Cleaning)

- `work/build-webos$ rm -rf BUILD`
- `work/build-webos$./mcf .status`

개별 컴포넌트 - 빌드 & 클리닝

- 개별 컴포넌트 빌드 = `work/build-webos$ make <component-name>`

- `work/build-webos$ source oe-init-build-env`
- `work/build-webos$ bitbake <component-name>`

- 개별 컴포넌트 클리닝

- `work/build-webos$ source oe-init-build-env`
- `work/build-webos$ bitbake -c clean <component-name>`

webOS OSE 플래싱



이미 빌드 된 webOS OSE 이미지 내려받기

🔗 webos-ports 저장소에서 받기

=> <http://build.webos-ports.org/webosose/raspberrypi3/>

Index of /webosose/raspberrypi3/

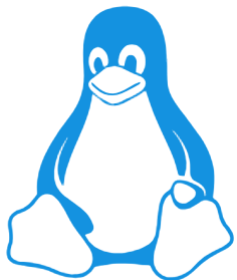
../		
webos-image-raspberrypi3-master-20180612152938...	12-Jun-2018 15:37	210648932
webos-image-raspberrypi3-master-20180615150841...	15-Jun-2018 15:47	210702664
webos-image-raspberrypi3-master-20180629112134...	29-Jun-2018 11:30	210708679
webos-image-raspberrypi3-master-20180725184349...	25-Jul-2018 23:30	210608323
webos-image-raspberrypi3-master-20180814034519...	14-Aug-2018 09:09	250012436
webos-image-raspberrypi3-master-20180814034519...	14-Aug-2018 09:07	41943040
webos-image-raspberrypi3-master-20180925122909...	25-Sep-2018 12:40	255068727
webos-image-raspberrypi3-master-20180925122909...	25-Sep-2018 12:38	41943040
webos-image-raspberrypi3-master-20190524062618...	24-May-2019 09:56	282949990
webos-image-raspberrypi3-master-20190524062618...	24-May-2019 09:51	41943040
webos-image-raspberrypi3-master-20190614082029...	14-Jun-2019 10:10	293983792
webos-image-raspberrypi3-master-20190614082029...	14-Jun-2019 10:08	41943040
webos-image-raspberrypi3-master-2019062620252...	27-Jun-2019 02:01	299299331
webos-image-raspberrypi3-master-2019062620252...	27-Jun-2019 01:58	41943040

webOS OSE 플래싱 소개

🐧 이미지 플래싱(Flashing) 방법



Win32DiskImager



- 경로 확인 (`.rpi-sdimg`)

- `$ cd BUILD/deploy/images/raspberrypi3`

- SD카드 Device 이름 확인

- `$ sudo fdisk -l`

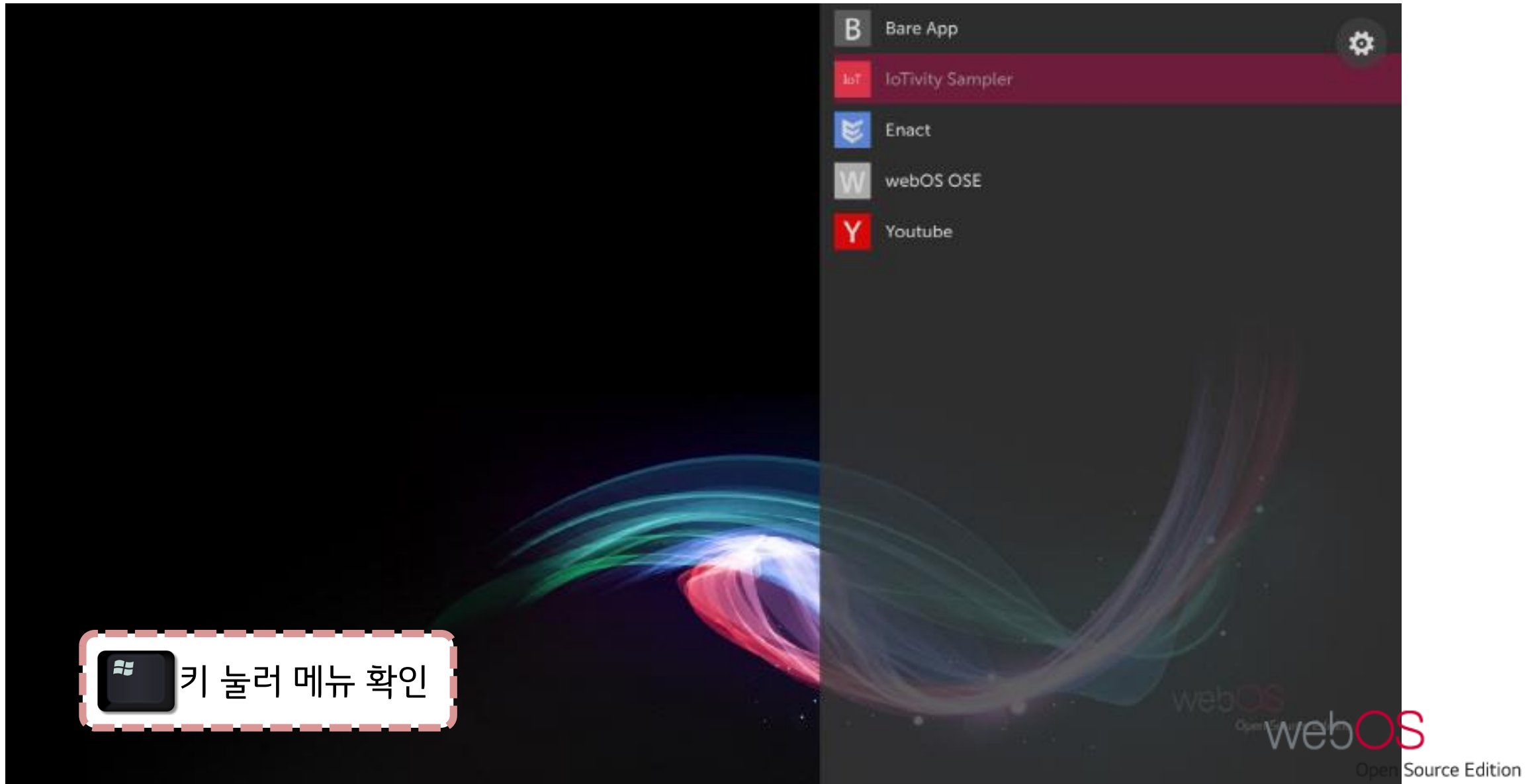
- 플래싱

- `$ sudo umount /dev/sdXn`

- `$ sudo dd bs=4M if=./**.rpi-sdimg of=/dev/sdX`

- `$ sudo umount /dev/sdXn`

webOS OSE 화면





webOS Open Source Edition
Overview

Getting Started

Setting Up

Requirements

Building webOS Open Source
Edition

Flashing webOS Open Source
Edition

Setting up Networking

Setting up for GDB
Debugging

Contributing

Roadmap

Video

Release Notes

Requirements

Version added 16-Jan-2018 | Modified 11-Oct-2018

Before you set up an environment for webOS Open Source Edition (OSE) development, make sure that you prepare the target device and systems that meet the following requirements.



webOS OSE cannot be built directly on the target device. You must use a separate Linux desktop with Ubuntu 14.04 64-bit installed. For more details, refer to the [Build System Requirements](#).

Contents

- Target Device Requirements
- Build System Requirements
 - Operating System
 - Hardware
 - Software Tools
- Host Machine Requirements
 - Operating System
 - Tools for Enact-based App Development

webOS OSE 개발자 사이트

DISCOVER > SETTING UP

<http://webosose.org/discover/setting>

Target Device Requirements

webOS OSE is optimized for **Raspberry Pi 3**. To test apps and components on Raspberry Pi 3, we recommend that you prepare the following set of hardware and peripheral devices.

- Raspberry Pi 3
- microSD card (8 GB or larger) and microSD card reader device
- HDMI-compliant monitor and cable
- Input devices such as a keyboard and a mouse
- Ethernet cable and internet connection

ABOUT webOS ?

QUESTIONS ?

? ANSWERS



webOS

Open Source Edition

