7

2024년 10월 15일 화요일 오후 3:34

Yocto를 이용한 개발

기존 BSP 위에서 빌드

BSP 레이어는 특정 하드웨어 장치나 장치 제품군에 대해 지원 가능하도록 YOCTO에 추가

BSP 레이어 이름은 meta- 접두어로 시작하고 그 뒤에 시스템 이름이 옴
타깃에 장치에 가장 적합한 BSP를 찾는 것은 Yocto를 사용

<기존 BSP를 빌드>

- **Yocto의 dunfell 릴리스 복제**

```
ysj@build-master:~$ git clone -b dunfell git://git.openembedded.org/meta-openembedded
Cloning into 'meta-openembedded'...
remote: Enumerating objects: 252945, done.
remote: Counting objects: 100% (511/511), done.
remote: Compressing objects: 100% (308/308), done.
remote: Total 252945 (delta 263), reused 320 (delta 191), pack-reused 252434
Receiving objects: 100% (252945/252945), 62.87 MiB | 3.33 MiB/s, done.
Resolving deltas: 100% (155076/155076), done.
```

```
ysj@build-master:~$ git clone --branch dunfell git://git.yoctoproject.org/meta-raspberrypi
Cloning into 'meta-raspberrypi'...
remote: Enumerating objects: 11313, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 11313 (delta 14), reused 0 (delta 0), pack-reused 11277
Receiving objects: 100% (11313/11313), 3.50 MiB | 1.65 MiB/s, done.
Resolving deltas: 100% (6553/6553), done.
```

- **Yocto의 일반 이미지 레시피**

a. Yocto를 복제한 디렉터리로 이동
b. 표준 이미지 레시피가 있는 디렉터리로 이동
c. 코어 이미지 레시피 리스트 확인

```
ysj@build-master:~$ cd poky/
ysj@build-master:~/poky$ cd meta/recipes-core/images
ysj@build-master:~/poky/meta/recipes-core/images$ ls -l core*
-rw-rw-r-- 1 ysj ysj  148 10월  14 14:04 core-image-base.bb
-rw-rw-r-- 1 ysj ysj  174 10월  14 14:04 core-image-minimal-dev.bb
-rw-rw-r-- 1 ysj ysj 1091 10월  14 14:04 core-image-minimal-initramfs.bb
-rw-rw-r-- 1 ysj ysj  270 10월  14 14:04 core-image-minimal-mtdutils.bb
-rw-rw-r-- 1 ysj ysj  332 10월  14 14:04 core-image-minimal.bb
-rw-rw-r-- 1 ysj ysj 1669 10월  14 14:04 core-image-tiny-initramfs.bb
```

d. Core-image-base 레시피 확인

```
ysj@build-master:~/poky/meta/recipes-core/images$ cat core-image-base.bb
SUMMARY = "A console-only image that fully supports the target device \
hardware."

IMAGE_FEATURES += "splash"

LICENSE = "MIT"

inherit core-image
```

e. Core-image-minimal 레시피 확인

```
ysj@build-master:~/poky/meta/recipes-core/images$ cat core-image-minimal.bb
SUMMARY = "A small image just capable of allowing a device to boot."

IMAGE_INSTALL = "packagegroup-core-boot ${CORE_IMAGE_EXTRA_INSTALL}"

IMAGE_LINGUAS = " "

LICENSE = "MIT"

inherit core-image

IMAGE_ROOTFS_SIZE ?= "8192"
IMAGE_ROOTFS_EXTRA_SPACE_append = "${@bb.utils.contains("DISTRO_FEATURES", "systemd", " + 4096", "" ,d)}"
```

f. Core-image-minimal-dev 레시피 확인

```
ysj@build-master:~/poky/meta/recipes-core/images$ cat core-image-minimal-dev.bb
require core-image-minimal.bb

DESCRIPTION = "A small image just capable of allowing a device to boot and \
is suitable for development work."

IMAGE_FEATURES += "dev-pkgs"
```

g. Poky/meta 아래의 classes 디렉터리로 이동
h. Core-image 클래스 파일 확인

```
ysj@build-master:~/poky/meta/recipes-core/images$ cd ../../classes
ysj@build-master:~/poky/meta/classes$ cat core-image.bbclass
```

```
# Available IMAGE_FEATURES:
#
# - x11                 - X server
# - x11-base            - X server with minimal environment
# - x11-sato            - OpenedHand Sato environment
# - tools-debug         - debugging tools
# - eclipse-debug       - Eclipse remote debugging support
# - tools-profile       - profiling tools
# - tools-testapps      - tools usable to make some device tests
# - tools-sdk           - SDK (C/C++ compiler, autotools, etc.)
# - nfs-server          - NFS server
# - nfs-client          - NFS client
# - ssh-server-dropbear - SSH server (dropbear)
# - ssh-server-openssh  - SSH server (openssh)
# - hwcodecs            - Install hardware acceleration codecs
# - package-management  - installs package management tools and preserves the package manager database
# - debug-tweaks        - makes an image suitable for development, e.g. allowing passwordless root logins
#   - empty-root-password
#   - allow-empty-password
#   - allow-root-login
#   - post-install-logging
# - dev-pkgs            - development packages (headers, etc.) for all installed packages in the rootfs
# - dbg-pkgs            - debug symbol packages for all installed packages in the rootfs
# - doc-pkgs            - documentation packages for all installed packages in the rootfs
# - bash-completion-pkgs - bash-completion packages for recipes using bash-completion bbclass
# - ptest-pkgs          - ptest packages for all ptest-enabled recipes
# - read-only-rootfs    - tweaks an image to support read-only rootfs
# - stateless-rootfs    - systemctl-native not run, image populated by systemd at runtime
# - splash              - bootup splash screen
```

- 라즈베리 파이 4용 core-image-base에 WIFI와 블루투스를 지원할 수 있도록 세팅

a. Yocto 를 복제한 디렉터리의 한 단계 상위 레벨로 이동
b. meta-raspberrypi BSP 레이어 내부의 디렉터리로 이동
c. 라즈베리 파이 이미지 레시피 리스트 확인

```
ysj@build-master:~/poky/meta/classes$ cd ../../..
ysj@build-master:~$ cd meta-raspberrypi/recipes-core//images
ysj@build-master:~/meta-raspberrypi/recipes-core/images$ ls -l
```

```
ysj@build-master:~/meta-raspberrypi/recipes-core/images$ ls -l
total 12
-rw-rw-r-- 1 ysj ysj 360 10월  16 11:07 rpi-basic-image.bb
-rw-rw-r-- 1 ysj ysj 279 10월  16 11:07 rpi-hwup-image.bb
-rw-rw-r-- 1 ysj ysj 163 10월  16 11:07 rpi-test-image.bb
ysj@build-master:~/meta-raspberrypi/recipes-core/images$
```

d. rpi-test-image 레시피 확인

```
ysj@build-master:~/meta-raspberrypi/recipes-core/images$ cat rpi-test-image.bb
# Base this image on core-image-base
include recipes-core/images/core-image-base.bb

COMPATIBLE_MACHINE = "^rpi$"

IMAGE_INSTALL_append = " packagegroup-rpi-test"
```

=> IMAGE_INSTALL 변수가 packagegroup-rpi-test 를 추가하고 해당 패키지를 이미지에 포함하기 위해 재정의

e. Meta-raspberrypi/recipes-core 아래에 있는 packagegroups 디렉터리로 이동
f. Packagegroup-rpi-test 레시피 확인

```
ysj@build-master:~/meta-raspberrypi/recipes-core/images$ cd ../packagegroups
ysj@build-master:~/meta-raspberrypi/recipes-core/packagegroups$ cat packagegroup-rpi-test.bb
DESCRIPTION = "RaspberryPi Test Packagegroup"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"

PACKAGE_ARCH = "${MACHINE_ARCH}"

inherit packagegroup

COMPATIBLE_MACHINE = "^rpi$"

OMXPLAYER  = "${@bb.utils.contains('MACHINE_FEATURES', 'vc4graphics', '', 'omxplayer', d)}"

RDEPENDS_${PN} = "\
    ${OMXPLAYER} \
    bcm2835-tests \
    rpio \
    rpi-gpio \
    pi-blaster \
    python3-rtimu \
    python3-sense-hat \
    connman \
    connman-client \
    wireless-regdb-static \
    bluez5 \
"

RRECOMMENDS_${PN} = "\
    ${@bb.utils.contains("BBFILE_COLLECTIONS", "meta-multimedia", "bigbuckbunny-1080p bigbuckbunny-480p bigbuckbunny-720p", "", d)} \
    ${MACHINE_EXTRA_RRECOMMENDS} \
"
```

=> connman, connman-client, bluez5 패키지는 Wi-fi와 블루투스가 전부 활성화 되도록 런타임 의존성 목록에 포함되어 있음

- **라즈베리 파이 4용 rpi-test-image 빌드**

1. Yocto를 복제한 디렉터리의 한 단계 상위 레벨로 이동 후 bitbake 환경 설정

```
ysj@build-master:~/meta-raspberrypi/recipes-core/packagegroups$ cd ../../..
ysj@build-master:~$ source poky/oe-init-build-env build-rpi
You had no conf/local.conf file. This configuration file has therefore been
created for you with some default values. You may wish to edit it to, for
example, select a different MACHINE (target hardware). See conf/local.conf
for more information as common configuration options are commented.

You had no conf/bblayers.conf file. This configuration file has therefore been
created for you with some default values. To add additional metadata layers
into your configuration please add entries to conf/bblayers.conf.

The Yocto Project has extensive documentation about OE including a reference
manual which can be found at:
    https://docs.yoctoproject.org

For more information about OpenEmbedded see their website:
    https://www.openembedded.org/


### Shell environment set up for builds. ###

You can now run 'bitbake <target>'

Common targets are:
    core-image-minimal
    core-image-sato
    meta-toolchain
    meta-ide-support

You can also run generated qemu images with a command like 'runqemu qemux86'

Other commonly useful commands are:
 - 'devtool' and 'recipetool' handle common recipe tasks
 - 'bitbake-layers' handles common layer tasks
 - 'oe-pkgdata-util' handles common target package tasks
ysj@build-master:~/build-rpi$ cd ..
```

2. 이미지에 다음 레이어 추가

```
ysj@build-master:~/build-rpi$ bitbake-layers add-layer ../meta-openembedded/meta-python
NOTE: Starting bitbake server...
ysj@build-master:~/build-rpi$ bitbake-layers add-layer ../meta-openembedded/meta-networking
NOTE: Starting bitbake server...
ysj@build-master:~/build-rpi$ bitbake-layers add-layer ../meta-openembedded/meta-multimedia
NOTE: Starting bitbake server...
ysj@build-master:~/build-rpi$ bitbake-layers add-layer ../meta-raspberrypi
NOTE: Starting bitbake server...
ysj@build-master:~/build-rpi$ ls
```

3. 필요한 모든 레이어가 이미지에 추가됐는지 확인

```
ysj@build-master:~/build-rpi$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer                 path                                      priority
==========================================================================
meta                  /home/ysj/poky/meta                       5
meta-poky             /home/ysj/poky/meta-poky                  5
meta-yocto-bsp        /home/ysj/poky/meta-yocto-bsp             5
meta-oe               /home/ysj/meta-openembedded/meta-oe       6
meta-raspberrypi      /home/ysj/meta-raspberrypi                9
meta-python           /home/ysj/meta-openembedded/meta-python   7
meta-networking       /home/ysj/meta-openembedded/meta-networking  5
meta-multimedia       /home/ysj/meta-openembedded/meta-multimedia  6
ysj@build-master:~/build-rpi$
```

4. 이전 bitbake-layers add-layer 명령문을 실행한 것과 관련해 bblayers.conf의 변경 사항 확인

```
ysj@build-master:~/build-rpi$ cat conf/bblayers.conf
# POKY_BBLAYERS_CONF_VERSION is increased each time build/conf/bblayers.conf
# changes incompatibly
POKY_BBLAYERS_CONF_VERSION = "2"

BBPATH = "${TOPDIR}"
BBFILES ?= ""

BBLAYERS ?= " \
  /home/ysj/poky/meta \
  /home/ysj/poky/meta-poky \
  /home/ysj/poky/meta-yocto-bsp \
  /home/ysj/meta-openembedded/meta-oe \
  /home/ysj/meta-raspberrypi \
  /home/ysj/meta-openembedded/meta-python \
  /home/ysj/meta-openembedded/meta-networking \
  /home/ysj/meta-openembedded/meta-multimedia \
  "
```

5. Meta-raspberrypi BSP 레이어에서 지원하는 시스템 확인

```
ysj@build-master:~/build-rpi$ cat conf/bblayers.conf
# POKY_BBLAYERS_CONF_VERSION is increased each time build/conf/bblayers.conf
# changes incompatibly
POKY_BBLAYERS_CONF_VERSION = "2"

BBPATH = "${TOPDIR}"
BBFILES ?= ""

BBLAYERS ?= " \
  /home/ysj/poky/meta \
  /home/ysj/poky/meta-poky \
  /home/ysj/poky/meta-yocto-bsp \
  /home/ysj/meta-openembedded/meta-oe \
  /home/ysj/meta-raspberrypi \
  /home/ysj/meta-openembedded/meta-python \
  /home/ysj/meta-openembedded/meta-networking \
  /home/ysj/meta-openembedded/meta-multimedia \
  "
ysj@build-master:~/build-rpi$ ls ../meta-raspberrypi/conf/machine/
include             raspberrypi-cm3.conf  raspberrypi0-wifi.conf  raspberrypi2.conf     raspberrypi3.conf     raspberrypi4.conf
raspberrypi-cm.conf  raspberrypi.conf      raspberrypi0.conf       raspberrypi3-64.conf  raspberrypi4-64.conf
```

6. Conf/local.conf 파일에 다음 줄을 추가

```
MACHINE = "raspberrypi4-64"
```

7. Ssh-server-openssh를 conf/local.conf 파일의 EXTRA_IMAGE_FEATURES 목록에 추가

```
EXTRA_IMAGE_FEATURES ?= "debug-tweaks ssh-server-openssh"
```

로컬 네트워크 액세스를 위해 이미지에 SSH 서버 추가

8. 이미지 빌드

```
ysj@build-master:~/build-rpi/conf$ bitbake rpi-test-image
Loading cache: 100% |#########################################################################################
Loaded 3283 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "aarch64-poky-linux"
MACHINE              = "raspberrypi4-64"
DISTRO               = "poky"
DISTRO_VERSION       = "3.1.33"
TUNE_FEATURES        = "aarch64 cortexa72 crc crypto"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp       = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-oe              = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
meta-raspberrypi     = "dunfell:2081e1bb9a44025db7297bfd5d024977d42191ed"
meta-python
meta-networking
meta-multimedia      = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"

Initialising tasks: 100% |####################################################################################
Sstate summary: Wanted 226 Found 0 Missed 226 Current 1553 (0% match, 87% complete)
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 4549 tasks of which 4108 didn't need to be rerun and all succeeded.
```

-> tmp/deploy/iamges/raspberrypi4-64 디렉터리에 rpi-test-image-raspberrypi4-64.rootf.wic.bz2 파일 생성

```
ysj@build-master:~/build-rpi$ ls -l tmp/deploy/images/raspberrypi4-64/rpi-test*wic.bz2
-rw-rw-r-- 2 ysj ysj 81165375 10월  16 15:23 tmp/deploy/images/raspberrypi4-64/rpi-test-image-raspberrypi4-64-20241016061628.rootfs.wic.bz2
lrwxrwxrwx 2 ysj ysj       60 10월  16 15:23 tmp/deploy/images/raspberrypi4-64/rpi-test-image-raspberrypi4-64.wic.bz2 -> rpi-test-image-raspberrypi4-64-20241016061628.rootfs.wic.bz2
```

<WI-FI 제어>

1. 빌드한 이미지의 호스트 이름은 raspberrypi4-64 이므로 다음과 같이 root로 장치에 ssh로 접속할 수 있음

```
ysj@build-master:~$ ssh 192.168.21.217
ysj@192.168.21.217's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-82-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

81 updates can be applied immediately.
추가 업데이트를 확인하려면 apt list --upgradable 을 실행하세요.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
```

2. Wifi 드라이버가 탑재돼 있는지 확인

```
ysj@build-master:~$ lsmod | grep 80211
mac80211             1249280  1 iwlmvm
libarc4                16384  1 mac80211
cfg80211              970752  3 iwlmvm,iwlwifi,mac80211
```

3. Connman-client 구동
4. Wi-Fi 실행
5. 연결 에이전트로 connmanctl 등록
6. Wi-Fi 네트워크 스캔
7. 모든 사용 가능한 Wi-Fi 네트워크 목록 확인

```
ysj@build-master:~$ connmanctl
connmanctl> enable wifi
Enabled wifi
connmanctl> agent on
Agent registered
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired                ethernet_c025a5d34391_cable
                         wifi_546ceb6ff117_hidden_managed_psk
    Kyungshin FREE       wifi_546ceb6ff117_4b79756e677368696e2046524545_managed_psk
    DIRECT-FMA11705168msIP wifi_546ceb6ff117_4449524543542d464d4131313730353136386d734950_managed_psk
    jin10                wifi_546ceb6ff117_6a696e3130_managed_psk
    KT_GiGA_AE99         wifi_546ceb6ff117_4b545f476947415f41453939_managed_psk
    Public WiFi Free     wifi_546ceb6ff117_5075626c6963205769466920462726565_managed_none
    Public WiFi Secure   wifi_546ceb6ff117_5075626c6963205769466920536563757265_managed_ieee8021x
    MI 9                 wifi_546ceb6ff117_4d492039_managed_psk
    Morris Kim           wifi_546ceb6ff117_4d6f72726973204b696d_managed_psk
```

8. Wi-Fi 네트워크에 접속
9. 다시 서비스 확인
10. Connman-client 종료

```
connmanctl> connect wifi_546ceb6ff117_4b79756e677368696e2046524545_managed_psk
Agent RequestInput wifi_546ceb6ff117_4b79756e677368696e2046524545_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory ]
Passphrase? 0327259300
Connected wifi_546ceb6ff117_4b79756e677368696e2046524545_managed_psk
connmanctl> services
*AR Wired               ethernet_c025a5d34391_cable
*AR Kyungshin FREE      wifi_546ceb6ff117_4b79756e677368696e2046524545_managed_psk
                        wifi_546ceb6ff117_hidden_managed_psk
    DIRECT-FMA11705168msIP wifi_546ceb6ff117_4449524543542d464d413131373035313638 6d734950_managed_psk
    jin10               wifi_546ceb6ff117_6a696e3130_managed_psk
    MI 9                wifi_546ceb6ff117_4d492039_managed_psk
    KT_GiGA_AE99        wifi_546ceb6ff117_4b545f47694741 5f41453939_managed_psk
    Morris Kim          wifi_546ceb6ff117_4d6f72726973204b696d_managed_psk
connmanctl> quit
ysj@build-master:~$ rudtls12!
rudtls12!: command not found
ysj@build-master:~$ quit
```

<블루투스 제어>

1. 라즈베리 파이 4전원을 켜고 ssh로 접속한 뒤 블루투스 드라이버가 내장돼 있는지 확인
2. HCI UART 드라이버를 초기화 후 Connman-client 구동
3. 블루투스 실행
4. Connmanctl 종료
5. 디폴트 에이전트를 요청
6. 컨트롤러 전원을 켠 후 컨트롤러에 대한 정보 표시

```
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Oct 22 09:16:44 2024 from 192.168.31.26
ysj@build-master:~$ lsmod | grep bluetooth
bluetooth             696320  33 btrtl,btintel,btbcm,bnep,btusb,rfcomm
ecdh_generic           16384  2 bluetooth
ysj@build-master:~$ btuart
btuart: command not found
ysj@build-master:~$ connmanctl
connmanctl> enable bluetooth
Error bluetooth: Already enabled
connmanctl> quit
ysj@build-master:~$ bluetoothctl
Agent registered
[CHG] Controller 54:6C:EB:6F:F1:1B Pairable: yes
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# show
Controller 54:6C:EB:6F:F1:1B (public)
        Name: build-master
        Alias: build-master
        Class: 0x002c0104
        Powered: yes
        Discoverable: no
        DiscoverableTimeout: 0x00000000
        Pairable: yes
        UUID: A/V Remote Control        (0000110e-0000-1000-8000-00805f9b34fb)
        UUID: Audio Source              (0000110a-0000-1000-8000-00805f9b34fb)
        UUID: PnP Information           (00001200-0000-1000-8000-00805f9b34fb)
        UUID: Audio Sink                (0000110b-0000-1000-8000-00805f9b34fb)
        UUID: Headset                   (00001108-0000-1000-8000-00805f9b34fb)
        UUID: A/V Remote Control Target (0000110c-0000-1000-8000-00805f9b34fb)
        UUID: Generic Access Profile    (00001800-0000-1000-8000-00805f9b34fb)
        UUID: Generic Attribute Profile (00001801-0000-1000-8000-00805f9b34fb)
        UUID: Headset AG                (00001112-0000-1000-8000-00805f9b34fb)
        UUID: Handsfree                 (0000111e-0000-1000-8000-00805f9b34fb)
        Modalias: usb:v1D6Bp0246d0535
        Discovering: no
Advertising Features:
        ActiveInstances: 0x00
        SupportedInstances: 0x08
        SupportedIncludes: tx-power
        SupportedIncludes: appearance
        SupportedIncludes: local-name
        SupportedSecondaryChannels: 1M
        SupportedSecondaryChannels: 2M
        SupportedSecondaryChannels: Coded
```

7. 블루투스 장치 검색 시작 (scan on)
8. 블루투스 장치 검색 중지 (scan off)

```
        SupportedSecondaryChannels: Coded
[bluetooth]# scan on
Discovery started
[CHG] Controller 54:6C:EB:6F:F1:1B Discovering: yes
[NEW] Device D0:C2:4E:64:1B:27 [TV] Samsung UA8070 75 TV
[NEW] Device 41:A3:98:51:E0:06 41-A3-98-51-E0-06
[NEW] Device 7E:4F:F0:9C:E3:5E 7E-4F-F0-9C-E3-5E
[NEW] Device F0:20:FF:BA:D9:74 KSA2401-0068
[NEW] Device 63:05:85:D4:BF:EA 63-05-85-D4-BF-EA
[NEW] Device 18:CC:18:CA:84:0F A11704780
[NEW] Device 44:4C:81:19:9B:93 44-4C-81-19-9B-93
[NEW] Device 44:5A:77:D3:FB:DD 44-5A-77-D3-FB-DD
[NEW] Device 2B:09:73:64:7F:0E 2B-09-73-64-7F-0E
[NEW] Device 66:A4:33:A3:94:02 66-A4-33-A3-94-02
[NEW] Device 1C:99:57:F3:DF:CB A11704985
[NEW] Device 74:56:D5:71:6E:D7 74-56-D5-71-6E-D7
[NEW] Device 79:17:AC:23:F2:7A 79-17-AC-23-F2-7A
[CHG] Device 63:05:85:D4:BF:EA ManufacturerData Key: 0x004c
[CHG] Device 63:05:85:D4:BF:EA ManufacturerData Value:
  01 00 00 00 20 00 00 00 20 00 00 00 00 00 00 00  .... ... .......
  00                                               .
[NEW] Device 51:EB:63:46:51:A1 51-EB-63-46-51-A1
[NEW] Device 67:C2:F4:58:45:07 67-C2-F4-58-45-07
[NEW] Device 10:39:17:39:5C:3D 10-39-17-39-5C-3D
[NEW] Device 60:E3:2B:C3:A5:D1 A11705168
[NEW] Device 7E:F9:42:4E:32:AA 7E-F9-42-4E-32-AA
[NEW] Device 59:25:92:21:E8:33 59-25-92-21-E8-33
[NEW] Device 14:F6:D8:E8:80:59 A11704576
[NEW] Device 51:90:BA:75:99:65 51-90-BA-75-99-65
[CHG] Device D0:C2:4E:64:1B:27 ManufacturerData Key: 0xff19
[CHG] Device D0:C2:4E:64:1B:27 ManufacturerData Value:
  00 75 00 09 01 00 00 00 06 01 00 00 00 00 00 00  .u..............
```

```
[NEW] Device A4:F9:33:30:58:35 A4-F9-33-30-58-35
[bluetooth]# scan off
Discovery stopped
[CHG] Controller 54:6C:EB:6F:F1:1B Discovering: no
[CHG] Device A4:F9:33:30:58:35 RSSI is nil
[CHG] Device 4D:D4:86:46:EB:68 RSSI is nil
```

9. 스마트폰과 페어링 시도 (yes를 입력하기 전 페어링 요청 수락)
10. 스마트폰 연결
11. 서비스 승인 메시지 표시되면 yes 입력

```
[CHG] Device 4D:D4:86:46:EB:68 RSSI is nil
[bluetooth]# pair 04:68:65:4B:FD:77
Attempting to pair with 04:68:65:4B:FD:77
[CHG] Device 04:68:65:4B:FD:77 Connected: yes
Request confirmation
[agent] Confirm passkey 261077 (yes/no): Request canceled
Failed to pair: org.bluez.Error.AuthenticationFailed
[CHG] Device 04:68:65:4B:FD:77 Connected: no
[bluetooth]# yes
```

스마트폰 연결부터는 실패

<커스텀 레이어 추가>

새로운 제품의 프로토타입을 만드는 경우 conf/local.conf의 IMAGE_INSTALL_append 변수에 할당된 목록에 패키지를 추가해 커스텀 이미지 생성할 수 있음

새로운 레시피로 커스텀 레이어 만들기

1. Yocto 복제한 디렉터리의 상위 레벨로 이동한 뒤 bitbake 환경 설정

```
ysj@build-master:~/poky$ source oe-init-build-env build-rpi
You had no conf/local.conf file. This configuration file has therefore been
created for you with some default values. You may wish to edit it to, for
example, select a different MACHINE (target hardware). See conf/local.conf
for more information as common configuration options are commented.

You had no conf/bblayers.conf file. This configuration file has therefore been
created for you with some default values. To add additional metadata layers
into your configuration please add entries to conf/bblayers.conf.

The Yocto Project has extensive documentation about OE including a reference
manual which can be found at:
    https://docs.yoctoproject.org

For more information about OpenEmbedded see their website:
    https://www.openembedded.org/


### Shell environment set up for builds. ###

You can now run 'bitbake <target>'

Common targets are:
    core-image-minimal
    core-image-sato
    meta-toolchain
    meta-ide-support

You can also run generated qemu images with a command like 'runqemu qemux86'

Other commonly useful commands are:
 - 'devtool' and 'recipetool' handle common recipe tasks
 - 'bitbake-layers' handles common layer tasks
 - 'oe-pkgdata-util' handles common target package tasks
ysj@build-master:~/poky/build-rpi$ ls
```

2. 애플리케이션에 대한 새 레이어 생성
3. 새 레이어 디렉터리로 이동한 뒤 레이어 파일 구조 검사 (tree)
4. Recipes-examples 및 examples 디렉러티, examples 레시피 파일 이름 변경 후 확인

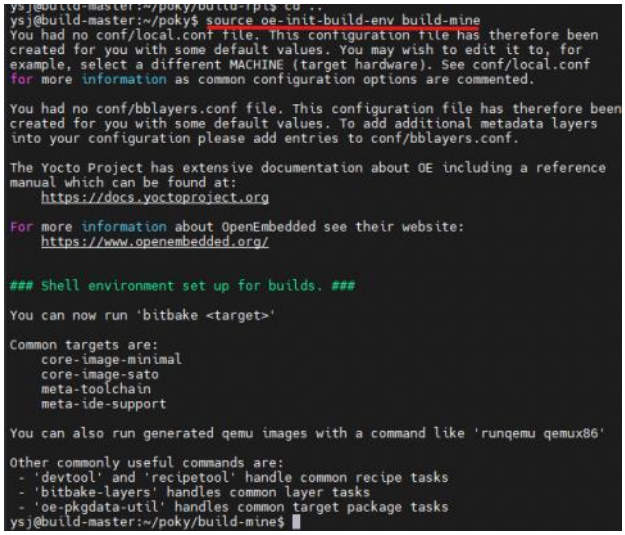
```
ysj@build-master:~/poky/build-rpi$ bitbake-layers create-layer ../meta-gattd
NOTE: Starting bitbake server...
Add your new layer with 'bitbake-layers add-layer ../meta-gattd'
ysj@build-master:~/poky/build-rpi$ d ../meta-gattd/
d: command not found
ysj@build-master:~/poky/build-rpi$ cd ../meta-gattd/
ysj@build-master:~/poky/meta-gattd$ tree
.
├── COPYING.MIT
├── README
├── conf
│   └── layer.conf
└── recipes-example
    └── example
        └── example_0.1.bb

3 directories, 4 files
ysj@build-master:~/poky/meta-gattd$ mv recipes-example recipes-gattd
ysj@build-master:~/poky/meta-gattd$ ls
COPYING.MIT  README  conf  recipes-gattd
ysj@build-master:~/poky/meta-gattd$ cd recipes-gattd/
ysj@build-master:~/poky/meta-gattd/recipes-gattd$ mv example gattd
ysj@build-master:~/poky/meta-gattd/recipes-gattd$ cd gattd/
ysj@build-master:~/poky/meta-gattd/recipes-gattd/gattd$ mv example_0.1.bb gattd_0.1.bb
ysj@build-master:~/poky/meta-gattd/recipes-gattd/gattd$ ls
gattd_0.1.bb
ysj@build-master:~/poky/meta-gattd/recipes-gattd/gattd$ cat gattd_0.1.bb
SUMMARY = "bitbake-layers recipe"
DESCRIPTION = "Recipe created by bitbake-layers"
LICENSE = "MIT"

python do_display_banner() {
    bb.plain("***********************************************");
    bb.plain("*                                             *");
    bb.plain("*  Example recipe created by bitbake-layers   *");
    bb.plain("*                                             *");
    bb.plain("***********************************************");
}

addtask display_banner before do_build
ysj@build-master:~/poky/meta-gattd/recipes-gattd/gattd$ ls
gattd_0.1.bb
ysj@build-master:~/poky/meta-gattd/recipes-gattd/gattd$
```

Devtool로 변경 사항 캡처

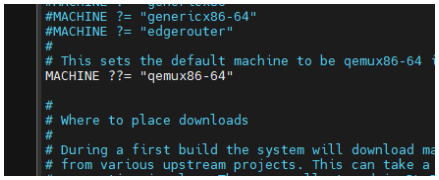
<개발 워크플로>

트리 내 레시피를 수정하는 대신 새 레이어에서 작업을 진행하고 있는지 확인 필요 (그렇지 않으면, 쉽게 덮어 쓰게 되고 지금까지 작업했던 시간을 잃어버릴 수 있음)

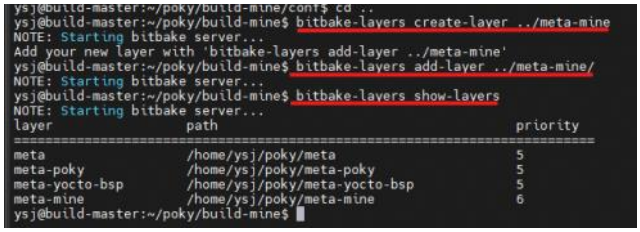
1. Yocto 상위 폴더로 이동 후 bitbake 환경 설정



2. Conf/local.conf 에서 MACHINE 설정



3. 새로운 레이어 생성 후 새 레이어 추가
4. 원하는 위치에 새로운 레이어가 생성됐는지 확인



전체 이미지 빌드

```
meta-mine             /home/ysj/poky/meta-mine                6
ysj@build-master:~/poky/build-mine$ devtool build-image core-image-full-cmdline
NOTE: Starting bitbake server...
INFO: Creating workspace layer in /home/ysj/poky/build-mine/workspace
NOTE: Reconnecting to bitbake server...
NOTE: Retrying server connection (#1)...
Parsing recipes: 100% |##########################################################################################################################
Parsing of 777 .bb files complete (0 cached, 777 parsed). 1333 targets, 40 skipped, 0 masked, 0 errors.
WARNING: No packages to add, building image core-image-full-cmdline unmodified
Loading cache: 100% |############################################################################################################################
Loaded 1333 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "ubuntu-20.04"
TARGET_SYS           = "x86_64-poky-linux"
MACHINE              = "qemux86-64"
DISTRO               = "poky"
DISTRO_VERSION       = "3.1.33"
TUNE_FEATURES        = "m64 core2"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp
meta-mine
workspace            = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"

NOTE: Fetching uninative binary shim http://downloads.yoctoproject.org/releases/uninative/4.4/x86_64-nativesdk-libc-4.4.tar.xz;sha256sum
l check PREMIRRORS first)
Initialising tasks: 100% |#######################################################################################################################
Sstate summary: Wanted 1439 Found 0 Missed 1439 Current 0 (0% match, 0% complete)
NOTE: Executing Tasks
WARNING: pkgconfig-native-0.29.2+gitAUTOINC+edf8e6f0ea-r0 do_fetch: Failed to fetch URL git://gitlab.freedesktop.org/pkg-config/pkg-conf
WARNING: popt-native-1.16-r3 do_fetch: Failed to fetch URL http://anduin.linuxfromscratch.org/BLFS/popt/popt-1.16.tar.gz, attempting MIR
WARNING: gnu-config-native-20200117+gitAUTOINC+5256817ace-r0 do_fetch: Failed to fetch URL git://git.savannah.gnu.org/git/config.git;pro
WARNING: ncurses-native-6.2-r0 do_fetch: Failed to fetch URL git://salsa.debian.org/debian/ncurses.git;protocol=https;branch=master, att
WARNING: shared-mime-info-native-1.15-r0 do_fetch: Failed to fetch URL git://gitlab.freedesktop.org/xdg/shared-mime-info.git;protocol=ht
WARNING: procps-3.3.16-r0 do_fetch: Failed to fetch URL git://gitlab.com/procps-ng/procps.git;protocol=https;branch=master, attempting M
WARNING: gnome-desktop-testing-2018.1-r0 do_fetch: Failed to fetch URL git://gitlab.gnome.org/GNOME/gnome-desktop-testing.git;protocol=h
WARNING: ca-certificates-20211016-r0 do_fetch: Failed to fetch URL git://salsa.debian.org/debian/ca-certificates.git;protocol=https;bran
WARNING: psmisc-23.3-r0 do_fetch: Failed to fetch URL git://gitlab.com/psmisc/psmisc.git;protocol=https;branch=master, attempting MIRROR
WARNING: rpcbind-1.2.5-r0 do_fetch: Failed to fetch URL https://downloads.sourceforge.net/rpcbind/rpcbind-1.2.5.tar.bz2, attempting MIRR
WARNING: zip-3.0-r2 do_fetch: Failed to fetch URL https://downloads.sourceforge.net/infozip/Zip%203.x%20%28latest%29/3.0/zip30.tar.gz, a
WARNING: cracklib-2.9.5-r0 do_fetch: Failed to fetch URL https://downloads.sourceforge.net/cracklib/cracklib-2.9.5.tar.gz, attempting MI
NOTE: Tasks Summary: Attempted 3794 tasks of which 8 didn't need to be rerun and all succeeded.

Summary: There were 12 WARNING messages shown.
INFO: Successfully built core-image-full-cmdline. You can find output files in /home/ysj/poky/build-mine/tmp/deploy/images/qemux86-64
ysj@build-master:~/poky/build-mine$
```

<새로운 레시피 생성>

1. Yocto 디렉터리 복제한 디렉터리의 상위 레벨로 이동 후 bitbake 환경 설정

```
ysj@build-master:~/poky/meta-mine$ cd ..
ysj@build-master:~/poky$ source oe-init-build-env  build-mine

### Shell environment set up for builds. ###

You can now run 'bitbake <target>'

Common targets are:
    core-image-minimal
    core-image-sato
    meta-toolchain
    meta-ide-support

You can also run generated qemu images with a command like 'runqemu qemux86'

Other commonly useful commands are:
 - 'devtool' and 'recipetool' handle common recipe tasks
 - 'bitbake-layers' handles common layer tasks
 - 'oe-pkgdata-util' handles common target package tasks
```

2. 릴리스된 소스 tar 압축 파일의 URL로 devtool add 실행

```
  meta-oe              1.4.5
ysj@build-master:~/poky/build-mine$ devtool add https://github.com/containers/bubblewrap/releases/download/v0.4.1/bubblewrap-0.4.1.tar.xz
NOTE: Starting bitbake server...
NOTE: Starting bitbake server...
INFO: Fetching https://github.com/containers/bubblewrap/releases/download/v0.4.1/bubblewrap-0.4.1.tar.xz...
Loading cache: 100% |############################################################################################################################
Loaded 2958 entries from dependency cache.
Parsing recipes: 100% |##########################################################################################################################
Parsing of 1908 .bb files complete (1905 cached, 3 parsed). 2961 targets, 80 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "x86_64-poky-linux"
MACHINE              = "qemux86-64"
DISTRO               = "poky"
DISTRO_VERSION       = "3.1.33"
TUNE_FEATURES        = "m64 core2"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp       = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-oe
meta-python          = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
workspace            = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"

Initialising tasks: 100% |#######################################################################################################################
Sstate summary: Wanted 0 Found 0 Missed 0 Current 18 (0% match, 100% complete)
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 83 tasks of which 81 didn't need to be rerun and all succeeded.
```

3. Bubblewrap_0.4.1.bb 끝에 다음 줄 추가 새로운 레시피 빌드

```
ysj@build-master:~/poky/build-mine$ devtool edit-recipe bubblewrap
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-mine$ devtool add https://github.com/c
```

devtool은 bubblewrap_0.4.1.bb를 편집기에서 오픈

```
# WARNING: the following LICENSE and LIC_FILES_CHKSUM values are best guesses - it is
# your responsibility to verify that the values are complete and correct.
LICENSE = "LGPLv2"
LIC_FILES_CHKSUM = "file://COPYING;md5=5f30f0716dfdd0d91eb439ebec522ec2"

SRC_URI = "https://github.com/containers/bubblewrap/releases/download/v${PV}/bubblewrap-${PV}.tar.xz"
SRC_URI[md5sum] = "1104b0e43006f22076b5057c129939c8"
SRC_URI[sha1sum] = "00e121950ea494fcd9cfbe23971c0938d6be6755"
SRC_URI[sha256sum] = "b9c69b9b1c61a608f34325c8e1a495229bacf6e4a07cbb0c80cf7a814d7ccc03"
SRC_URI[sha384sum] = "68c9ae7f0c8a13540ecd9b6647f307253022acca5267853ed5c497c6ea236f08fd9ce51016b8595eaa0af66f7bde92cf"
SRC_URI[sha512sum] = "0ff46dc0fda2d0cffbb36cc52ff43951b30cbd835a42cc56806acbbbd827796bfadbb1cfafc84d6b47a72c031ca44abe1c377acc0cc25fe3b33e854f5f687d35"

# NOTE: the following prog dependencies are unknown, ignoring: xsltproc
# NOTE: unable to map the following pkg-config dependencies: libselinux
#       (this is based on recipes that have previously been built and packaged)
DEPENDS = "libcap bash-completion"

# NOTE: if this software is not capable of being built in a separate build directory
# from the source, you should replace autotools with autotools-brokensep in the
# inherit line
inherit pkgconfig autotools

# Specify any options you want to pass to the configure script using EXTRA_OECONF:
EXTRA_OECONF = ""
FILES_${PN} += "/usr/share/*"
```

<레시피로 빌드된 소스 수정>

몇가지 작은 코드 변경을 하기 위해 jq를 직접 패치하는 법 (devtool modify 필요)

1. Build-mine 환경 설정에서 몇 개의 레이어 삭제
2. 깃허브에서 meta-openembedded 저장소 복제
3. 이미지에 meta-oe와 meta-mine 레이어 추가
4. 모든 레이어가 이미지에 추가됐는지 확인

```
-bash: syntax error near unexpected token `('
ysj@build-master:~/poky/build-mine$ bitbake-layers remove-layer workspace
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-mine$ bitbake-layers remove-layer meta-mine
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-mine$ git clone -b dunfell https://github.com/openembedded/meta-openembedded.git ../meta-openembedded
Cloning into '../meta-openembedded'...
remote: Enumerating objects: 253587, done.
remote: Counting objects: 100% (7098/7098), done.
remote: Compressing objects: 100% (2320/2320), done.
remote: Total 253587 (delta 5130), reused 5802 (delta 4730), pack-reused 246489 (from 1)
Receiving objects: 100% (253587/253587), 69.17 MiB | 11.03 MiB/s, done.
Resolving deltas: 100% (152099/152099), done.
ysj@build-master:~/poky/build-mine$ bitbake-layers add-layer ../meta-openembedded/meta-oe/
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-mine$ bitbake-layers add-layer ../meta-mine
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-mine$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer                 path                                      priority
==========================================================================
meta                  /home/ysj/poky/meta                       5
meta-poky             /home/ysj/poky/meta-poky                  5
meta-yocto-bsp        /home/ysj/poky/meta-yocto-bsp             5
meta-oe               /home/ysj/poky/meta-openembedded/meta-oe  6
meta-mine             /home/ysj/poky/meta-mine                  6
ysj@build-master:~/poky/build-mine$ ls
```

5. Onig 패키지는 jq의 런타임 의존성이 있으므로 conf/local.conf에 다음 줄을 추가

```
#MACHINE ?= "genericx86-64"
#MACHINE ?= "edgerouter"
/home/ysj/meta-gattd/
default machine to be q
MACHINE ??= "qemux86-64"
IMAGE_INSTALL_append = " onig"
#
# Where to place downloads
#
```

6. 이미지 재빌드

```
ysj@build-master:~/poky/build-mine/conf$ devtool build-image core-image-full-cmdline
NOTE: Starting bitbake server...
NOTE: Reconnecting to bitbake server...
NOTE: Retrying server connection (#1)...
Parsing recipes: 100% |#########################################################################################################################
Parsing of 1539 .bb files complete (0 cached, 1539 parsed). 2322 targets, 74 skipped, 0 masked, 0 errors.
WARNING: No packages to add, building image core-image-full-cmdline unmodified
NOTE: Reconnecting to bitbake server...
NOTE: Previous bitbake instance shutting down?, waiting to retry...
NOTE: Retrying server connection (#1)...
Loading cache: 100% |###########################################################################################################################
Loaded 2322 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION            = "1.46.0"
BUILD_SYS             = "x86_64-linux"
NATIVELSBSTRING       = "universal"
TARGET_SYS            = "x86_64-poky-linux"
MACHINE               = "qemux86-64"
DISTRO                = "poky"
DISTRO_VERSION        = "3.1.33"
TUNE_FEATURES         = "m64 core2"
TARGET_FPU            = ""
meta
meta-poky
meta-yocto-bsp        = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-oe               = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
meta-mine
workspace             = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"

Initialising tasks: 100% |######################################################################################################################
Sstate summary: Wanted 15 Found 5 Missed 10 Current 1431 (33% match, 99% complete)
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 3812 tasks of which 3784 didn't need to be rerun and all succeeded.
INFO: Successfully built core-image-full-cmdline. You can find output files in /home/ysj/poky/build-mine/tmp/deploy/images/qemux86-64
```

<레시피를 최신 버전으로 업데이트>

1. 먼저 build-mine 환경 설정에서 몇 개의 레이어를 삭제
2. 이미지에 meta-python과 meta-mine 레이어 추가

```
ERROR: Failed to copy script to 192.168.21.217 - rerun with -s to get a complet
ysj@build-master:~/poky/build-mine$ bitbake-layers remove-layer workspace
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-mine$ bitbake-layers remove-layer meta-mine
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-mine$ bitbake-layers add-layer ../meta-openembedd
```

```
ysj@build-master:~/poky/build-mine$ bitbake-layers add-layer ../meta-mine
NOTE: Starting bitbake server...
```

```
COPYING.MIT  README  classes  conf  licenses  recipes-connectivity  recipes-core  recipes-devtools
ysj@build-master:~/poky/build-mine$ bitbake-layers add-layer ../meta-openembedded/meta-python
NOTE: Starting bitbake server...
```

3. 필요한 레이어들이 이미지에 추가됐는지 확인

```
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-mine$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer                 path                                      priority
==========================================================================
meta                  /home/ysj/poky/meta                       5
meta-poky             /home/ysj/poky/meta-poky                  5
meta-yocto-bsp        /home/ysj/poky/meta-yocto-bsp             5
meta-oe               /home/ysj/poky/meta-openembedded/meta-oe  6
meta-mine             /home/ysj/poky/meta-mine                  6
meta-python           /home/ysj/poky/meta-openembedded/meta-python  7
ysj@build-master:~/poky/build-mine$
```

4. 사용가능한 파이썬 모듈 확인

```
ysj@build-master:~/poky/build-mine$ bitbake -s | grep ^python3
python3                                          :3.8.18-r0
python3-absl                                      :0.7.0-r0
python3-absl-native                               :0.7.0-r0
python3-aenum                                     :2.2.3-r0
python3-aenum-native                              :2.2.3-r0
python3-aiofiles                                  :0.4.0-r0
python3-aiohttp                                   :3.6.2-r0
python3-alembic                                   :1.4.2-r0
python3-ansi2html                                 :1.5.2-r0
python3-anyjson                                   :0.3.3-r0
python3-appdirs                                   :1.4.4-r0
python3-apply-defaults                            :0.1.4-r0
python3-apply-defaults-native                     :0.1.4-r0
python3-arpeggio                                  :1.9.2-r0
python3-arpeggio-native                           :1.9.2-r0
python3-asn1crypto                                :1.3.0-r0
python3-asn1crypto-native                         :1.3.0-r0
python3-astor                                     :0.8.1-r0
python3-astor-native                              :0.8.1-r0
python3-astroid                                   :2.3.3-r0
python3-async                                     :0.6.2-r0
python3-async-native                              :0.6.2-r0
python3-async-timeout                             :3.0.1-r0
python3-atomicwrites                              :1.3.0-r0
python3-attr                                      :0.3.1-r0
python3-attrs                                    :19.3.0-r0
python3-attrs-native                             :19.3.0-r0
python3-automat                                   :0.8.0-r0
python3-aws-iot-device-sdk-python                 :1.4.8-r0
python3-aws-iot-device-sdk-python-native              :1.4.8-r0
python3-babel                                    :2.8.0-r0
python3-babel-native                             :2.8.0-r0
python3-backports-functools-lru-cache              :1.6.1-r0
python3-bandit                                    :1.6.2-r0
python3-bandit-native                             :1.6.2-r0
python3-bcrypt                                    :3.1.7-r0
python3-beautifulsoup4                            :4.8.2-r0
python3-beautifulsoup4-native                     :4.8.2-r0
python3-behave                                    :1.2.6-r0
python3-bitarray                                  :1.2.1-r0
python3-bitarray-native                           :1.2.1-r0
python3-blinker                                    :1.4-r0
python3-booleanpy                                  :3.7-r0
python3-booleanpy-native                           :3.7-r0
```

/home/ysj/meta-gattd/

5. Conf/local.conf 파일을 검색해 python3와 python3-flask가 빌드 되고 이미지에 설치 되고 있는지 확인

   없는 경우 다음 줄을 추가

```
#MACHINE ?= "edgerouter"
#
# This sets the default machine to be qemux86-64 if no other machine
MACHINE ??= "qemux86-64"
IMAGE_INSTALL_append = " onig"
IMAGE_INSTALL_append = " python3 python3-flask"
#
# Where to place downloads
#
```

6. 이미지 재빌드

   Flask는 파이썬 3라이브러리 이므로, 이미지 업그레이드하려면 파이썬 3, Flask, Flask 의존성 포함해야 함
   다음과 같은 방법으로 진행

```
ysj@build-master:~/poky/build-mine$ devtool build-image core-image-full-cmdline
NOTE: Starting bitbake server...
NOTE: Reconnecting to bitbake server...
NOTE: Retrying server connection (#1)...
Parsing recipes: 100% |#########################################################################################################################################################
Parsing of 1907 .bb files complete (0 cached, 1907 parsed). 2960 targets, 80 skipped, 0 masked, 0 errors.
INFO: Building image core-image-full-cmdline with the following additional packages: jq
NOTE: Reconnecting to bitbake server...
NOTE: Retrying server connection (#1)...
NOTE: Reconnecting to bitbake server...
NOTE: Previous bitbake instance shutting down?, waiting to retry...
NOTE: Retrying server connection (#2)...
Loading cache: 100% |###########################################################################################################################################################
Loaded 2959 entries from dependency cache.
Parsing recipes: 100% |#########################################################################################################################################################
Parsing of 1907 .bb files complete (1905 cached, 2 parsed). 2960 targets, 80 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "x86_64-poky-linux"
MACHINE              = "qemux86-64"
DISTRO               = "poky"
DISTRO_VERSION       = "3.1.33"
TUNE_FEATURES        = "m64 core2"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp       = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-oe              = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
meta-mine            = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-python          = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
workspace            = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"

Initialising tasks: 100% |######################################################################################################################################################
Sstate summary: Wanted 53 Found 1 Missed 52 Current 1442 (1% match, 96% complete)
NOTE: Executing Tasks
NOTE: jq: compiling from external source tree /home/ysj/poky/build-mine/workspace/sources/jq
NOTE: Tasks Summary: Attempted 3929 tasks of which 3802 didn't need to be rerun and all succeeded.
INFO: Successfully built core-image-full-cmdline. You can find output files in /home/ysj/poky/build-mine/tmp/deploy/images/qemux86-64
ysj@build-master:~/poky/build-mine$
```

업그레이드 방법

1. 패키지 이름과 업데이트할 타깃 버전으로 devtool upgrade를 실행

```
ysj@build-master:~$ git config --global user.email ysj0930@kyungshin.co.kr
ysj@build-master:~$ devtool upgrade python3-flask --version 1.1.2
NOTE: Starting bitbake server...
NOTE: Reconnecting to bitbake server...
NOTE: Retrying server connection (#1)...
Loading cache: 100% |###########################################################################################################################################################
Loaded 2959 entries from dependency cache.
Parsing recipes: 100% |#########################################################################################################################################################
Parsing of 1907 .bb files complete (1906 cached, 1 parsed). 2960 targets, 80 skipped, 0 masked, 0 errors.
INFO: Extracting current version source...
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "x86_64-poky-linux"
MACHINE              = "qemux86-64"
DISTRO               = "poky"
DISTRO_VERSION       = "3.1.33"
TUNE_FEATURES        = "m64 core2"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp       = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-oe              = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
meta-mine            = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-python          = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
workspace            = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"

Initialising tasks: 100% |######################################################################################################################################################
Sstate summary: Wanted 0 Found 0 Missed 0 Current 20 (0% match, 100% complete)
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 93 tasks of which 90 didn't need to be rerun and all succeeded.
INFO: Extracting upgraded version source...
INFO: Fetching https://files.pythonhosted.org/packages/source/F/Flask/Flask-1.1.2.tar.gz...
Loading cache: 100% |###########################################################################################################################################################
Loaded 2959 entries from dependency cache.
Parsing recipes: 100% |#########################################################################################################################################################
Parsing of 1908 .bb files complete (1905 cached, 3 parsed). 2961 targets, 80 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "x86_64-poky-linux"
MACHINE              = "qemux86-64"
DISTRO               = "poky"
DISTRO_VERSION       = "3.1.33"
TUNE_FEATURES        = "m64 core2"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp       = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-oe              = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
meta-mine            = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
```

=> devtool은 파일을 편집기로 open

2. 새로운 레시피 빌드

```
ERROR: No recipe named 'python-flask' in your workspace
ysj@build-master:~/poky/build-mine$ devtool build python3-flask
NOTE: Starting bitbake server...
NOTE: Reconnecting to bitbake server...
NOTE: Retrying server connection (#1)...
Loading cache: 100% |#########################################################################################################
Loaded 2958 entries from dependency cache.
Parsing recipes: 100% |#######################################################################################################
Parsing of 1908 .bb files complete (1905 cached, 3 parsed). 2961 targets, 80 skipped, 0 masked, 0 errors.
NOTE: Reconnecting to bitbake server...
NOTE: Previous bitbake instance shutting down?, waiting to retry...
NOTE: Retrying server connection (#1)...
Loading cache: 100% |#########################################################################################################
Loaded 2958 entries from dependency cache.
Parsing recipes: 100% |#######################################################################################################
Parsing of 1908 .bb files complete (1905 cached, 3 parsed). 2961 targets, 80 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "x86_64-poky-linux"
MACHINE              = "qemux86-64"
DISTRO               = "poky"
DISTRO_VERSION       = "3.1.33"
TUNE_FEATURES        = "m64 core2"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp       = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"
meta-oe
meta-python          = "dunfell:01358b6d705071cc0ac5aefa7670ab235709729a"
workspace            = "dunfell:63d05fc061006bf1a88630d6d91cdc76ea33fbf2"

Initialising tasks: 100% |####################################################################################################
Sstate summary: Wanted 0 Found 0 Missed 0 Current 204 (0% match, 100% complete)
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 783 tasks of which 783 didn't need to be rerun and all succeeded.
ysj@build-master:~/poky/build-mine$
```

나만의 배포판 빌드

자신의 배포판을 구축하는 3단계 프로세스

1. 새로운 배포 레이어 생성
2. 배포판 설정 파일 생성
3. 배포판에 더 많은 레시피 추가

<새로운 배포 레이어 생성>

1. Yocto를 복제한 상위 디렉터리로 이동 후 bitbake 환경 설정
2. Build-rpi 환경에서 emta-gattd 레이어 삭제

```
ysj@build-master:~/poky$ source oe-init-build-env build-rpi/

### Shell environment set up for builds. ###

You can now run 'bitbake <target>'

Common targets are:
    core-image-minimal
    core-image-sato
    meta-toolchain
    meta-ide-support

You can also run generated qemu images with a command like 'runqemu qemux86'

Other commonly useful commands are:
 - 'devtool' and 'recipetool' handle common recipe tasks
 - 'bitbake-layers' handles common layer tasks
 - 'oe-pkgdata-util' handles common target package tasks
ysj@build-master:~/poky/build-rpi$ birbake-layers remove-layer meta-gattd
birbake-layers: command not found
ysj@build-master:~/poky/build-rpi$ bitbake-layers remove-layer meta-gattd
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-rpi$ ls
```

3. 배포판의 새로운 레이어 생성 후 build-rpi 환경 설정에 새 레이어 추가

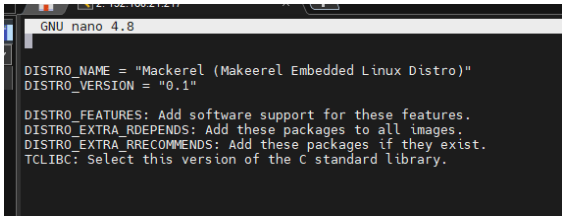
```
bitbake-cookerdaemon.log  cache  conf  tmp
ysj@build-master:~/poky/build-rpi$ bitbake-layers create-layer ../meta-mackerel
NOTE: Starting bitbake server...
Add your new layer with 'bitbake-layers add-layer ../meta-mackerel'
ysj@build-master:~/poky/build-rpi$ bitbake-layers add-layer ../meta-mackerel
NOTE: Starting bitbake server...
ysj@build-master:~/poky/build-rpi$ bitbake-layers show-layer
NOTE: Starting bitbake server...
usage: bitbake-layers [-d] [-q] [-F] [--color COLOR] [-h] <subcommand> ...
bitbake-layers: error: argument <subcommand>: invalid choice: 'show-layer' (choose from
, 'show-recipes', 'show-appends', 'show-cross-depends', 'create-layer')
ysj@build-master:~/poky/build-rpi$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer                 path                                      priority
==========================================================================
meta                  /home/ysj/poky/meta                       5
meta-poky             /home/ysj/poky/meta-poky                  5
meta-yocto-bsp        /home/ysj/poky/meta-yocto-bsp             5
meta-mackerel         /home/ysj/poky/meta-mackerel              6
ysj@build-master:~/poky/build-rpi$ ls
bitbake-cookerdaemon.log  cache  conf  tmp
ysj@build-master:~/poky/build-rpi$ cd conf
ysj@build-master:~/poky/build-rpi/conf$ sudo nano local.conf
ysj@build-master:~/poky/build-rpi/conf$
```

예제의 배포판 이름: mackerel
자체 배포판의 레이어를 생성하면 배포 정책을 패키지 레시피와 별도로 유지

<배포판 환경 설정>

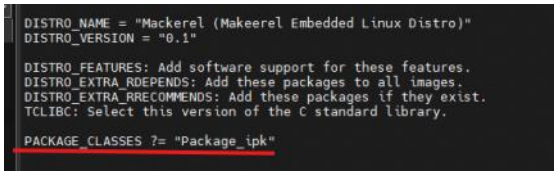
meta-mackerel 배포판 레이어의 conf/distro 디렉터리에 배포판 설정 파일 생성 후 배포판과 동일한 이름을 지정

Conf/distro/mackerel.conf 에서 필요한 변수 설정

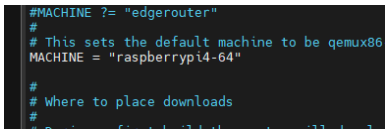


<런타임 패키지 관리>

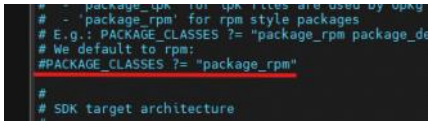
배포판 이미지에 패키지 관리자를 포함하면 안전한 무선 업데이트와 신속한 애플리케이션 개발이 가능해짐
원격 서버에서 패키지를 가져와 타깃 장치에 설치할 수 있는 것을 '런타임 패키지 관리' 라고 함



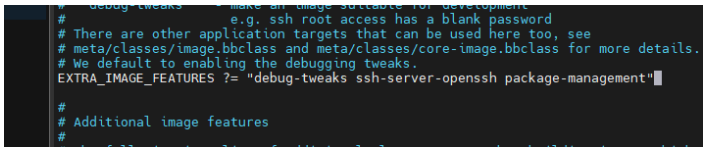
추가



라즈베리 파이로 되어있는지 확인

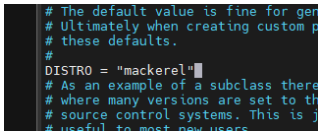


PACKAGE_CLASSES 주석 처리



패키지 관리를 활성화 하기 위해 conf/local.conf에 있는 EXTRA_IMAGE_FEATURES 목록에 package-management 추가

=> 여기까지 현재 빌드의 모든 패키지가 포함된 패키지 데이터베이스가 배포판 이미지에 설치



Conf/local/conf 파일에 있는 DISTRO 변수를 배포판 이름으로 설정

```
$ bitbake -c clean rpi-test-image
$ bitbake rpi-test-image
```

배포판 빌드

```
$ ls tmp-glibc/deploy/images/raspberrypi4-64/rpi-testimage*wic.bz2
```

완성된 이미지는 디렉터리에 배치

```
$ ssh root@raspberrypi4-64.local
```

Etcher를 사용해 이미지를 마이크로 SD 카드에 쓰고 라즈베리파이 4에 부팅
이더넷과 SSH 연결