AUTOMOTIVE LINUX SDK BUILD GUIDE

Rev. 2.00

May 31, 2018

TelcChips

TABLE OF CONTENTS

Contents

Figures

Tables

1 Telechips Automotive Linux SDK Build Preparation

The Telechips Automotive Linux SDK (ALS) is based on the Yocto Project 2.2 Morty.
Therefore, in order to use Telechips ALS, the Yocto Project environment must be set on the host PC.

2 Introduction and composition of Yocto Project

2.1 Yocto Project

Yocto Project is a open source project which focuses on embedded linux developer.
And it uses Poky, which is a combination of OpenEmbedded project and bitbake, as build system to make Linux Images.
Using Yocto Project, bootloader, kernel, and rootfs can be built all at once.

2.2 Task Process

As shown in Figure 2.1, task procedure of Yocto Project downloads source from upstream based on meta data and performs build.
Once build is complete, package as well as image and SDK are provided as results

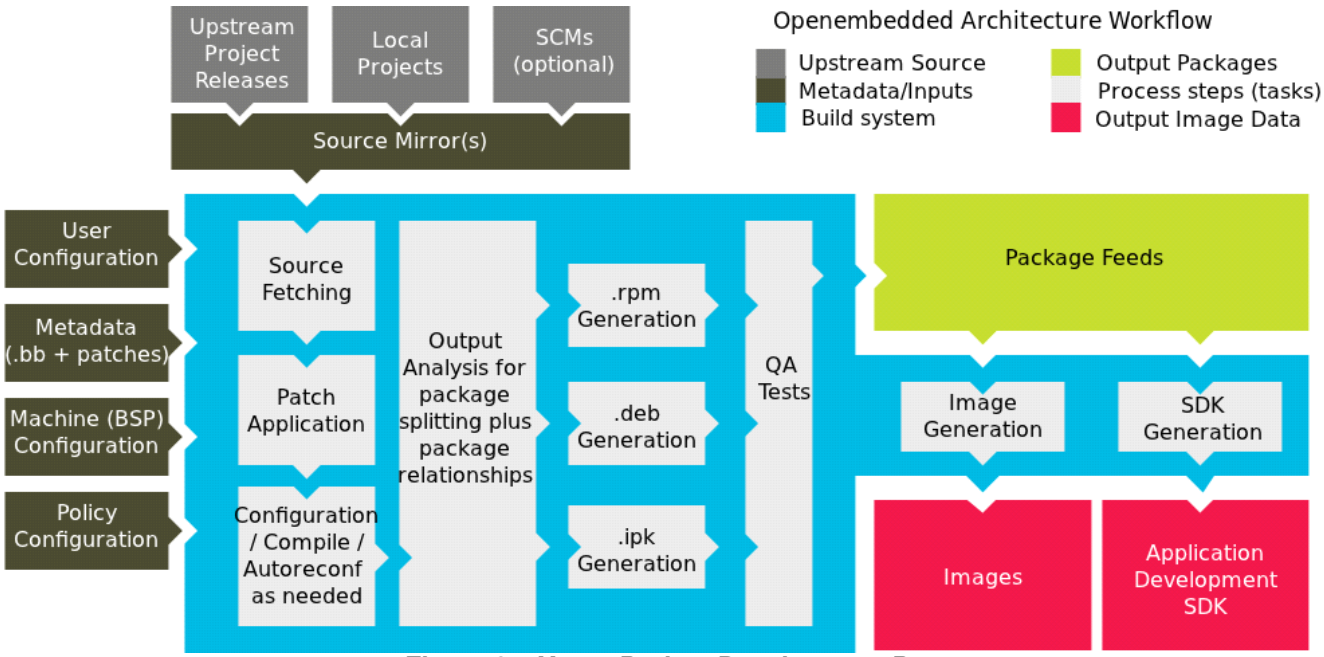

Figure 2.1 Yocto Project Development Process

3 Yocto Project Installation

3.1 Configuration for Yocto Project

Following mandatory packages must be installed on Host System (individual pc or development server) in order to use Yocto Project.
For details, refer to the following Yocto Project website.
https://www.yoctoproject.org/docs/2.4.2/yocto-project-qs/yocto-project-qs.html

3.2 Linux distribution versions supported by Yocto Project

Following Linux distribution versions are supported. Other distribution versions have not gone through verification process at Yocto project.
For details, refer to the following Yocto Project website
https://www.yoctoproject.org/docs/2.4.2/ref-manual/ref-manual.html#detailed-supported-distros

- Ubuntu 14.04 (LTS)
- Ubuntu 14.10
- Ubuntu 15.04
- Ubuntu 15.10
- Ubuntu 16.04
- Fedora release 22
- Fedora release 23
- Fedora release 24
- CentOS release 7.x
- Debian GNU/Linux 8.x (Jessie)
- openSUSE 13.2
- openSUSE 42.1

3.3 List of package required for using Yocto Project

Following packages must be installed to use Yocto Project.
For details, refer to the following Yocto Project website.
https://www.yoctoproject.org/docs/2.4.2/ref-manual/ref-manual.html#required-packages-for-the-host-development-system

Table 3.1 Required Packages for the Host Development System

Linux Distribution	Need Packages
Ubuntu and Debian	$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \ build-essential chrpath socat cpio python python3 python3-pip python3-pexpect \ xz-utils debianutils iputils-ping libsdl1.2-dev xterm
Fedora	$ sudo dnf install gawk make wget tar bzip2 gzip python3 unzip perl patch \ diffutils diffstat git cpp gcc gcc-c++ glibc-devel texinfo chrpath ccache \ perl-Data-Dumper perl-Text-ParseWords perl-Thread-Queue perl-bignum \ socat python3-pexpect findutils which file cpio python python3-pip xz \ SDL-devel xterm
openSUSE	$ sudo zypper install python gcc gcc-c++ git chrpath make wget python-xml \ diffstat makeinfo python-curses patch socat python3 python3-curses tar python3-pip \ python3-pexpect xz which libSDL-devel xterm

CentOS	$ sudo yum install –y epel-release
	$ sudo yum makecache
	$ sudo yum install gawk make wget tar bzip2 gzip python unzip perl patch \
	diffutils diffstat git cpp gcc gcc-c++ glibc-devel texinfo chrpath socat \
	perl-Data-Dumper perl-Text-ParseWords perl-Thread-Queue python3-pip \
	xz which SDL-devel xterm

4 Telechips Automotive Linux SDK Installation

Telechips Automotive Linux SDK(ALS) can be downloaded through android repo.

4.1 repo Installation

Install repo by referring to the following website.
https://source.android.com/source/downloading.html
If repo is already installed, it can be continued to be used.

4.2 Download Telechips ALS

You can download Telechips ALS using repo and script as follows.

```
~/works/Yocto/release/ALS$ repo init -u ssh://git.telechips.com/linux_ivi/manifest.git -m als_v3.0.0.xml
repo has been initialized in /home/user/works/Yocto/release
~/works/Yocto/release/ALS$ repo sync
~/works/Yocto/release/ALS$ tools/download-source-mirror.sh
This may take a long time depending on your network environment.
Continue? (Y/n) => Y
Choose ALS version
   1. als_v3.0.0
select number(1-1) => 1
Start downloading...
~/works/Yocto/release/ALS$
```

4.3 Composition of Telechips ALS

Once download is complete, following items can be checked.

Item			Explanation
document			Documents of Telechips Automotive Linux SDK
poky	meta		Yocto Project 2.4 Rocko build system
	meta-poky		
	meta-yocto-bsp		
	meta-qt5		Support Qt5 5.6.3 Layer
	meta-linaro		Support Linaro toolchain Layer
	meta-telechips	meta-bsp	Support Telechips BSP Layer
		meta-core	Recipes that require modification from the Open Source Software (OSS) used by Telechips ALS or that are not in Yocto Project 2.4.
		meta-ivi	Main configuration package and example programs of Telechips ALS.
		als-patch	Telechips common patchs
release-info			History of Telechips ALS
source-mirror			Local repository for building Telechips ALS basic class
tools			Programs and drivers for FWDN (For Windows) Yocto Project 2.4 Rocko buildtools (x86_64-buildtools-nativesdk-standalone-2.4.2.sh) Source mirror downloader script(FTP)

4.4 Yocto Project buildtools Installation

Install buildtools to configure the Yocto Project build environment for Telechips ALS as follows:
After installation is completed, you can check the following files.
Installation should be done only once.

```
~/works/Yocto/release/ALS$ tools/x86_64-buildtools-nativesdk-standalone-2.4.2.sh
Build tools installer version 2.4.2
==================================
Enter target directory for SDK (default: /opt/poky/2.4.2): ~/works/Yocto/release/ALS/buildtools
You are about to install the SDK to "/export/home4/B110141/thshin/Yocto/release/ALS/buildtools". Proceed[Y/n]? Y
Extracting SDK.......done
Setting it up...done
SDK has been successfully set up and is ready to be used.
Each time you wish to use the SDK in a new shell session, you need to source the environment setup script e.g.
 $ . /export/home4/B110141/thshin/Yocto/release/ALS/buildtools/environment-setup-x86_64-pokysdk-linux
~/works/Yocto/release/ALS$ ls buildtools/
environment-setup-x86_64-pokysdk-linux  sysroots/  version-x86_64-pokysdk-linux
~/works/Yocto/release/ALS$
```

5 Building Telechips Automotive Linux SDK

Conduct the following in order to build Automotive Linux SDK provided by Telechips.
When you switch to the build environment, several environment variables such as PATH will change, so some commands that are not related to the build may malfunction. If so, please proceed in a new shell.

5.1 Configuring the Yocto Project build environment

Before proceeding with the build, Change to the Yocto Project build environment first.

```
~/works/Yocto/release/ALS$ source buildtools/environment-setup-x86_64-pokysdk-linux
~/works/Yocto/release/ALS$
```

5.2 Bitbake Configuration

Yocto Project builds with bitbake. So you need to change to the bitbake environment.
There are two ways to set environment: auto script and user definition.

5.2.1 Configuration using Auto script

Use the als-build.sh script to use the default settings.
If you select machine as below, it automatically changes to bitbake environment.

```
~/works/Yocto/release/ALS$ source poky/als-build.sh
Choose MACHINE
   1. tcc8030
select number(1-1) => 1
machine(tcc8030) selected.
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:
      http://yoctoproject.org/documentation

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

Yocto Project common targets are:
      core-image-minimal
      core-image-sato
      meta-toolchain
      adt-installer
      meta-ide-support


Telechips common targets are:
      telechips-ivi-image-minimal
      telechips-ivi-image-gstreamer(minimal + GStreamer)
      telechips-ivi-image-qt(minimal + Qt)
      telechips-ivi-image(minimal + GStreamer + Qt)

      meta-toolchain-telechips(Application Development Toolkit)
      meta-toolchain-telechips-ivi(Application Development Toolkit include GStreamer)
      meta-toolchain-telechips-qt5(Application Development Toolkit include GStreamer and Qt5)


Telechips Automotive Linux Platform targets are:
      automotive-linux-platform-image(telechips-ivi-image + demo applications)

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

```
 or
You can also run generated Telechips images on Telechips EVM Boards(eg. tcc8030)

~/works/linux-avn/release/ALS/build/tcc8030$
```

5.2.2 User definition Configuration

For user definition settings, select the TEMPLATECONF environment variable and the build directory, machine, and so on, as follows:

5.2.2.1 Setting Environmental Variable of Template

Register environmental variable in order to use template provided by Telechips.
Use an absolute path.

```
~/works/Yocto/release/ALS$ export TEMPLATECONF=~/works/Yocto/release/ALS/poky/meta-telechips/template
```

5.2.2.2 Bitbake Configuration

Change to Bitbake build environment.
In "source poky/oe-init-build-env build/tcc8030", tcc8030 is the directory where the build proceeds.
This directory is automatically created according to the settings you have made.
5.2.2.1 is ignored if the build directory was previously used.

```
~/works/Yocto/release/ALS$ mkdir build
~/works/Yocto/release/ALS$ source poky/oe-init-build-env build/tcc8030
Yocto Project common targets are:
      core-image-minimal
      core-image-sato
      meta-toolchain
      adt-installer
      meta-ide-support


Telechips common targets are:
      telechips-ivi-image-minimal
      telechips-ivi-image-gstreamer(minimal + GStreamer)
      telechips-ivi-image-qt(minimal + Qt)
      telechips-ivi-image(minimal + GStreamer + Qt)

      meta-toolchain-telechips(Application Development Toolkit)
      meta-toolchain-telechips-ivi(Application Development Toolkit include GStreamer)
      meta-toolchain-telechips-qt5(Application Development Toolkit include GStreamer and Qt5)


Telechips Automotive Linux Platform targets are:
      automotive-linux-platform-image(telechips-ivi-image + demo applications)

You can also run generated qemu images with a command like 'runqemu qemux86'
  or
You can also run generated Telechips images on Telechips EVM Boards(eg. tcc8030)

~/works/Yocto/release/ALS/build/tcc8030$
```

5.2.2.3 Setup Configuration

Set conf/local.conf as shown below according to Target board.

Name of variable	Default value	Explanation
BB_NUMBER_THREADS	8	Maximum number of tasks that can be run simultaneously
PARALLEL_MAKE	-j 16	Option for make
MACHINE	N/A	target board setting
EXTRA_IMAGE_FEATURES	debug-tweaks read-only-rootfs	Image property selected additionally. For detailed options, refer to conf/local.conf
CORE_IMAGE_EXTRA_INSTALL	NULL	Name of package to be installed additionally on image
MY_MAC_ADDRESS	FF:FF:FF:FF:FF:FF	ethernet hw address of target board
MY_IP_ADDRESS	192.168.1.1	TCP/IP ip address of target board
MY_GATEWAY_ADDRESS	192.168.1.254	TCP/IP default gateway address of target board
DISTRO	poky-telechips-systemd	Select SDK type poky-telechips(sysvinit) poky-telechips-systemd(systemd)
INCOMPATIBLE_LICENSE	GPLv3, GPL-3.0, LGPLv3, LGPL-3.0, AGPLv3, AGPL-3.0	License type to be excluded upon SDK creation

5.2.2.4 Setup Graphics System and Qt Platform Abstraction(QPA)

Use INVITE_PLATFORM variable at conf/local.conf file in order to set linux graphics system and QPA.
Default value is Qt5/Wayland.

Graphics System and QPA	Configuration
Qt5/Wayland	INVITE_PLATFORM = “qt5” INVITE_PLATFORM += “qt5/wayland” INVITE_PLATFORM += “drm” DISTRO_FEATURES_append = “ wayland” DISTRO_FEATURES_append = “ opengl”
Qt5/EGL FS	INVITE_PLATFORM = “qt5” INVITE_PLATFORM += “qt5/eglfs” DISTRO_FEATURES_append = “ opengl”

5.3 Build automotive-linux-platform-image

Use bitbake to build image as shown below.

5.3.1 Build image type

The build image provided by ALS is as follows.

5.3.1.1 telechips-ivi-image-minimal

Only the minimum packages (dbus, alsa, …) that make up ALS are installed.

5.3.1.2 telechips-ivi-image-gstreamer

The gstreamer package is added to telechips-ivi-image-minimal.

5.3.1.3 telechips-ivi-image

The graphic manager and Qt platform are added to telechips-ivi-image-gstreamer.

5.3.1.4 automotive-linux-platform-image

The demo application is added to telechips-ivi-image.

```
~/works/Yocto/release/ALS/build/tcc8030$ bitbake automotive-linux-platform-image
Loading cache: 100% |#####################################################################| Time: 0:00:00
Loaded 1803 entries from dependency cache.
Parsing recipes: 100% |###################################################################| Time: 0:00:01
Parsing of 1257 .bb files complete (1249 cached, 8 parsed). 1813 targets, 277 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.36.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal-4.8"
TARGET_SYS           = "aarch64-telechips-linux"
MACHINE              = "tcc8030"
DISTRO               = "poky-telechips-systemd"
DISTRO_VERSION       = "2.4.2"
TUNE_FEATURES        = "aarch64 simd cortexa53"
TARGET_FPU           = ""
KBUILD_DEFCONFIG     = "tcc803x_linux_avn_reduce_defconfig"
TCMODE               = "default"
INVITE_PLATFORM      = "qt5 qt5/wayland telechips-egl drm genivi fastboot"
IMAGE_FEATURES       = " debug-tweaks read-only-rootfs"
SDKIMAGE_FEATURES    = "dev-pkgs"
GCCVERSION           = "linaro-7.2"
GLIBCVERSION         = "linaro-2.20"
meta
meta-poky            = "v3.x:928c88b446056728acd65f20a5b8b99142a1aa37"
meta-linaro-toolchain = "v3.x:5d0509563bb17775a9cb6541069cf7a75473e6cf"
meta-bsp             = "v3.x:af53c9bba3f99ab702e5819dcdfc2f30599e5dd4"
meta-qt5             = "v3.x:3dec11dd621999d19cf4f5bbc2e7013b01be7dd3"
meta-core            = "v3.x:af53c9bba3f99ab702e5819dcdfc2f30599e5dd4"
meta-qt5             = "v3.x:af53c9bba3f99ab702e5819dcdfc2f30599e5dd4"
meta-ivi
meta-genivi          = "v3.x:3023dde792730f841ea65606b63f2ba1cf98faf4"
meta-ivi             = "v3.x:be26d8ddba205b343993ca5c2017f8dba45ea49e"

Initialising tasks: 100% |################################################################| Time: 0:00:14
NOTE: Executing SetScene Tasks
NOTE: Executing RunQueue Tasks
NOTE: Tasks Summary: Attempted 4704 tasks of which 4099 didn't need to be rerun and all succeeded.
NOTE: Writing buildhistory

Summary: There were 7 WARNING messages shown.
~/works/Yocto/release/ALS/build/tcc8030$
```

5.3.2 Location of results

Once build is completed successfully, the results will be in the following locations.

```
~/works/Yocto/release/ALS/build/tcc8030$ ls -1 tmp/deploy/images/tcc8030/
automotive-linux-platform-image-tcc8030.ext4
Image-tcc8030-linux-lpd4321_sv0.1.dtb
initramfs-telechips-image-tcc8030.cpio.gz
modules-tcc8030.tgz
tc-boot-tcc8030.img
tc-boot-tcc8030_uncompressed.img
tcc803x_fwdn-tcc8030.rom
tcc803x_snor_boot.rom
~/works/Yocto/release/ALS/build/tcc8030$
```

6 FAQ

Frequently asked questions in the Yocto Project environment.

6.1 How to rebuild package

If you want to rebuild package, you can clean and build refer to bellow
There are three types of clean tasks:

6.1.1 clean and build

Deletes all output (after task do_unpack). However, shared state cache data is not cleared.

```
~/works/Yocto/release/ALS/build/tcc8030$ bitbake alsa-lib -c clean
~/works/Yocto/release/ALS/build/tcc8030$ bitbake alsa-lib
```

6.1.2 cleanall and build

It clears all data, including all results, shared state cache data, and downloaded source code.

```
~/works/Yocto/release/ALS/build/tcc8030$ bitbake alsa-lib -c cleanall
~/works/Yocto/release/ALS/build/tcc8030$ bitbake alsa-lib
```

6.1.3 cleansstate and build

Clears all output and shared state cache data.

```
~/works/Yocto/release/ALS/build/tcc8030$ bitbake alsa-lib -c cleansstate
~/works/Yocto/release/ALS/build/tcc8030$ bitbake alsa-lib
```

6.2 How to rebuild all packages

Yocto Project have not total build. If you want to rebuild all packages, you can clean each packages and build image refer to bellow

```
~/works/Yocto/release/ALS/build/tcc8030$ bitbake -c cleanall alsa-lib
~/works/Yocto/release/ALS/build/tcc8030$ bitbake -c cleanall dbus
~/works/Yocto/release/ALS/build/tcc8030$ bitbake -c cleanall gstreamer1.0
~/works/Yocto/release/ALS/build/tcc8030$ bitbake automotive-linux-platform-image -c cleanall
~/works/Yocto/release/ALS/build/tcc8030$ bitbake automotive-linux-platform-image
```

6.3 How to modify source code

The package's source located ${TMPDIR}/work/${MULTIMACH_TARGET_SYS}/${PN}/${EXTENDPE}${PV}-${PR}/git or unpacked directory name
To modify and apply the source code, you must proceed with the force option as follows:

```
~/works/Yocto/release/ALS/build/tcc8030$ pushd tmp/work/aarch64-telechips-linux/tc-launcher/1.0.0-r0/git
modify source codes
~/works/Yocto/release/ALS/build/tcc8030/ tmp/work/aarch64-telechips-linux/tc-launcher/1.0.0-r0/git$ popd
~/works/Yocto/release/ALS/build/tcc8030$ bitbake tc-launcher -f -c compile
~/works/Yocto/release/ALS/build/tcc8030$ bitbake tc-launcher
```

6.4 How to add extra package to image

If you want to add package to image, refer to below
You can set up the CORE_IMAGE_EXTRA_INSTALL variable in conf/local.conf to install additional packages to the image you are currently building.
- CORE_IMAGE_EXTRA_INSTALL += "openssh"

6.5 How to modify rootfs image without modify recipes

If you temporarily modify rootfs and create an image, run do_image with the force option as follows:

```
~/works/Yocto/release/ALS/build/tcc8030$ pushd tmp/work/tcc8030-telechips-linux/automotive-linux-platform-image/1.0-r0/rootfs/
modify rootfs
~/works/Yocto/release/ALS/build/tcc8030/ tmp/work/tcc8030-telechips-linux/automotive-linux-platform-image/1.0-r0/rootfs/ $ popd
~/works/Yocto/release/ALS/build/tcc8030$ bitbake automotive-linux-platform-image -f -c image
~/works/Yocto/release/ALS/build/tcc8030$ bitbake automotive-linux-platform-image
```

6.6 How to change from read-only to writable rootfs

The default setting of ALS rootfs is read-only.If you want to use writable rootfs, you can change IMAGE_FEATURES from conf/local.conf

- read only : EXTRA_IMAGE_FEATURES = "debug-tweaks read-only-rootfs"
- writeable : EXTRA_IMAGE_FEATURES = "debug-tweaks"

6.7 How to enable network environment

To use network in ALS, set conf / local.conf as follows

- activate kernel config and systemd network configuration : USE_NETWORK = "1"
- MAC Address configuration : MY_MAC_ADDRESS = "F4:50:EB:DF:5D:6F"
- IP Address configuration : MY_IP_ADDRESS = "192.168.21.85"
- Gateway Address configuration : MY_GATEWAY_ADDRESS = "192.168.21.254"

6.8 How to enable ssh environment

ALS supports ssh server based on openssh.OpenSSH is installed by adding ssh server to EXTRA_IMAGE_FEATURES as follows.

- EXTRA_IMAGE_FEATURES += "ssh-server-openssh"

USE_NETWORK in 6.7 is automatically activated when setting ssh server. However, IP address, MAC address and Gateway address must be set according to the user's environment.

6.9 How to install gdb environment

To use gdb with ALS, change the EXTRA_IMAGE_FEATURES and INCOMPATIBLE_LICENSE settings as follows:

- Installing GDB-related packages : EXTRA_IMAGE_FEATURES = "debug-tweaks tools-debug dbg-pkgs"
- Disable incompatible setting : #INCOMPATIBLE_LICENSE = "GPLv3 GPL-3.0 LGPLv3 LGPL-3.0 AGPLv3 AGPL-3.0"

7 Building Application Development Toolkit(ADT)

7.1 Bitbake Configuration

Change to bitbake build environment just like 5.2. If this is already done, this step is optional.

7.2 ADT type provided by ALS

ADT in ALS v2.5 supports three types.

7.2.1 meta-toolchain-telechips(Application Development Toolkit)

Packages for building basic ALS-based programs are installed.
For detailed installation packages, please refer to the following file.

- meta-telechips/meta-core/recipes-core/packagegroups/packagegroup-als-toolchain-target.bb

7.2.2 meta-toolchain-telechips-ivi(Application Development Toolkit include GStreamer)

Additional packages for building gstreamer and ALS demo applications in meta-toolchain-telechips are installed.
For detailed installation packages, please refer to the following file.

- meta-telechips/meta-ivi/recipes-core/packagegroups/packagegroup-als-ivi-toolchain-target.bb

7.2.3 meta-toolchain-telechips-qt5(Application Development Toolkit include GStreamer and Qt5)

Additional Qt5 packages are installed in meta-toolchain-telechips-ivi.
For detailed installation packages, please refer to the following file.

- meta-qt5/recipes-qt/packagegroups/packagegroup-qt5-toolchain-target.bb

7.3 Build Application Development Toolkit(ADT)

To build the ADT, use the bitbake to build as follows:

```
~/works/Yocto/release/ALS/build/tcc8030$ bitbake meta-toolchain-telechips-qt5
Loading cache: 100% |################################################################################| Time: 0:00:00
Loaded 1794 entries from dependency cache.
Parsing recipes: 100% |##############################################################################| Time: 0:00:01
Parsing of 1248 .bb files complete (1240 cached, 8 parsed). 1804 targets, 277 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.36.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal-4.8"
TARGET_SYS           = "aarch64-telechips-linux"
MACHINE              = "tcc8030"
DISTRO               = "poky-telechips-systemd"
DISTRO_VERSION       = "2.4.2"
TUNE_FEATURES        = "aarch64 simd cortexa53"
TARGET_FPU           = ""
KBUILD_DEFCONFIG     = "tcc803x_linux_avn_reduce_defconfig"
TCMODE               = "default"
INVITE_PLATFORM      = "qt5 qt5/wayland telechips-egl drm    commercial genivi fastboot"
IMAGE_FEATURES       = " debug-tweaks read-only-rootfs"
SDKIMAGE_FEATURES    = "dev-pkgs"
GCCVERSION           = "linaro-7.2"
GLIBCVERSION         = "linaro-2.20"
meta
meta-poky            = "v3.x:a51bc0490cd80029985fe9ba36cec4911b1d74c0"
meta-linaro-toolchain = "v3.x:5d0509563bb17775a9cb6541069cf7a75473e6cf"
meta-bsp             = "v3.x:af53c9bba3f99ab702e5819dcdfc2f30599e5dd4"
meta-qt5             = "v3.x:3dec11dd621999d19cf4f5bbc2e7013b01be7dd3"
meta-core            = "v3.x:af53c9bba3f99ab702e5819dcdfc2f30599e5dd4"
meta-ivi             = "v3.x:be26d8ddba205b343993ca5c2017f8dba45ea49e"
meta-qt5             = "v3.x:af53c9bba3f99ab702e5819dcdfc2f30599e5dd4"
meta-ivi
meta-genivi          = "v3.x:3023dde792730f841ea65606b63f2ba1cf98faf4"

Initialising tasks: 100% |###########################################################################| Time: 0:00:19
NOTE: Executing SetScene Tasks
NOTE: Executing RunQueue Tasks
NOTE: Tasks Summary: Attempted 6329 tasks of which 6312 didn't need to be rerun and all succeeded.
NOTE: Writing buildhistory
~/works/Yocto/release/ALS/build/tcc8030$
```

7.4 Location of result

Once build is completed successfully, the results will be located in following locations.

```
~/works/Yocto/release/ALS/build/tcc8030$ ls -1 tmp/deploy/sdk
telechips-als-v3.0.0-toolchain-aarch64-opengl-wayland-ivi-qt5-x86_64-gcc-linaro-7.2.host.manifest
telechips-als-v3.0.0-toolchain-aarch64-opengl-wayland-ivi-qt5-x86_64-gcc-linaro-7.2.sh
telechips-als-v3.0.0-toolchain-aarch64-opengl-wayland-ivi-qt5-x86_64-gcc-linaro-7.2.target.manifest
telechips-als-v3.0.0-toolchain-aarch64-opengl-wayland-ivi-qt5-x86_64-gcc-linaro-7.2.testdata.json
~/works/Yocto/release/ALS/build/tcc8030$
```

8 Relevant websites

For details of Yocto Project, refer to following documents.

8.1 Yocto Project Quick Start
https://www.yoctoproject.org/docs/2.4.2/yocto-project-qs/yocto-project-qs.html

8.2 Yocto Project Wiki
https://wiki.yoctoproject.org/wiki/Main_Page

8.3 Yocto Project Development Manual
https://www.yoctoproject.org/docs/2.4.2/dev-manual/dev-manual.html

8.4 Yocto Project Reference Manual
https://www.yoctoproject.org/docs/2.4.2/ref-manual/ref-manual.html

8.5 Yocto Project Application Developer's Guide
https://www.yoctoproject.org/docs/2.4.2/adt-manual/adt-manual.html

8.6 BitBake User Manual
https://www.yoctoproject.org/docs/2.4.2/bitbake-user-manual/bitbake-user-manual.html

9 Revision History

Date	Version	Description
2018-05-31	2.00	Update build environment Add TCC8030 Remove not supported machines
2017-09-19	1.06	Update build environment Deprecated external toolchain Add FAQ
2016-08-10	1.05	Add tcc8021-evm board Add tcc8925-lcn board remove tcc8925-evm board
2016-04-27	1.04	Update official release Add detail ddr3 type build options Add tcc8971-lcn Add tcc8972-lcn
2015-10-31	1.03	Update build options
2015-09-30	1.02	Beta Release
2015-05-27	1.01	Fixed Korean typo
2015-03.31	1.00	Initial release

DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this material invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use.

Further, Telechips, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.
No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.
This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips, Inc. does not provide any indemnification for any intellectual properties owned by third party.
Telechips, Inc. can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips, Inc. is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

COPYRIGHT STATEMENT

Important Notice

This material may include technology owned by the 3rd party licensor and the technology may be subject to its associated licenses. It is solely customer's responsibility to identify and comply with such licenses. No other licenses are granted or implied by Telechips with making available this material.

For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:
"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial. Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit http://mp3licensing.com".

For customers who use other firmware of mp3:
"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit http://mp3licensing.com".

For customers who use Digital Wave DRA solution:
"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

For customers who use DTS technology:
"This product made under license to certain U.S. patents and/or foreign counterparts."
"© 1996 – 2011 DTS, Inc. All rights reserved."

For customers who use Dolby technology:
"Supply of this Implementation of Dolby technology does not convey a license nor imply a right under any patent, or any other industrial or intellectual property right of Dolby Laboratories, to use this Implementation in any finished end-user or ready-to-use final product. It is hereby notified that a license for such use is required from Dolby Laboratories."

For customers who use MS technology:
"This product is subject to certain intellectual property rights of Microsoft and cannot be used or distributed further without the appropriate license(s) from Microsoft."