

Extensible SDK (eSDK) support for Turris Omnia Phase2

Tables of Contents

- [Yocto Build](#)
 - [Host Machine Setup](#)
 - [Yocto workspace setup](#)
 - [Populate eSDK](#)
- [eSDK Installation](#)
 - [Host Configuration](#)
 - [Installation](#)
- [Build from eSDK](#)
 - [Environment Setup](#)
 - [New Components](#)
 - [Add Component](#)
 - [Modify Component](#)
 - [Edit Component's recipe](#)
 - [Build Component](#)
 - [Build Image](#)

Yocto Build

Host Machine Setup

Hardware requirements:

- Ubuntu 18.04 desktop machine
- RAM - 8 GB or more
- Memory - a minimum of 100 GB free space

Refer to the link for host machine setup: [Host Setup](#)

Yocto workspace setup

build setup

```
repo init -u https://code.rdkcentral.com/r/manifests -m rdkb-turris-extsrc.xml -b dunfell
repo sync -j4 --no-clone-bundle
```

apply bug fixes

Note: The above bug fixes changes are temporary given as instructions. This is not required once the change is get merged into the appropriate layer.

Populate eSDK

- broadband turris gateway

image build

```
MACHINE=turris source meta-turris/setup-environment
bitbake rdk-generic-broadband-image -c populate_sdk_ext
```

The generated eSDK installer resides under the directory build-<MACHINE>/tmp/deploy/sdk/rdk-glibc-x86_64-arm-toolchain-ext-2.0.sh of the Yocto workspace.

Note: The generated eSDK installer script usually occupies 1.5 to 2.5 GB

eSDK Installation

The installer can be done on any x86_64 Linux machines.

Host Configuration

Refer to the [RDKCentral's credential configuration](#) to setup RDK Central's credentials in the machine where eSDK to be installed. This is to access repositories by the RDK Yocto recipes that reside in the eSDK installer.

Installation

```
# run the installer script file
# installer asks for a directory to install (default directory ~/rdk_sdk)
# installer asks permission to proceed
./rdk-glibc-x86_64-arm-toolchain-ext-2.0.sh
```

Below is the result (terminal output) of the installation

```
xxxuser@yyy-machine-003:~/turris/broadband$ build-turris/tmp/deploy/sdk/rdk-glibc-x86_64-arm-toolchain-ext-2.0.sh
RDK (A Yocto Project based Distro) Extensible SDK installer version 2.0
=====
Enter target directory for SDK (default: ~/rdk_sdk):
You are about to install the SDK to "/home/xxxuser/rdk_sdk". Proceed [Y/n]? Y
Extracting
SDK.....
done
Setting it up...
Extracting buildtools...
Preparing build system...

Parsing recipes: 100% |#####| Time: 0:00:52
Initialising tasks: 100% |#####| Time: 0:00:00
Checking sstate mirror object availability: 100% |#####| Time: 0:00:00
Loading cache: 100% |#####| Time: 0:00:00
Initialising tasks: 100% |#####| Time: 0:00:00
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.

$ . /home/xxxuser/rdk_sdk/environment-setup-armv7ahf-neon-rdk-linux-gnueabi

SDK Installation Done.
```

Build from eSDK

This section covers how to use the eSDK for component build and image build

Environment Setup

The installer can be run on any x86_64 Linux based machines.

```
# change directory to the installed path
cd ~/rdk_sdk
# setup the eSDK environemnt
source environment-setup-armv7ahf-neon-rdk-linux-gnueabi
```

New Components

Add Component

```
# way 1
# add a new recipe with URL
devtool add <recipe_name> <source URL>
# eg. devtool add mosquitto "http://mosquitto.org/files/source/mosquitto-1.6.10.tar.gz"

# way 2
# add a new recipe with external source directory
devtool add <recipe_name> <absolute path>
# eg. devtool add wireless-tools /path/for/source/directory
```

Modify Component

```
# modify existing recipe's URL or srctree or to add patches
devtool modify <recipe_name> <new URL>
# eg. devtool modify mosquitto "http://mosquitto.org/files/source/mosquitto-1.6.10.tar.gz"
```

Edit Component's recipe

```
# modify recipe from an editor
devtool edit-recipe <recipe_name>
# eg. devtool edit-recipe mosquitto
```

Build Components

```
# Pre-requisite: add a new recipe using devtool
devtool build <recipe_name>
# eg. devtool build mosquitto
```

Build Image

A complete RDK image can be generated from the eSDK installer.

Note: The same image where the eSDK populated can be generated here.

```
# devtool command to build image from eSDK
# <IMAGE> - rdk-generic-broadband-image
devtool build-image <IMAGE>
```

Note: Stabilized image is generated in Phase2

Known Issues

1. Taskhash mismatch error may appear while generating image from sdk, but that would not break the build.

Flashing procedure

[Turris Omnia Reference Platform: Flashing Instruction](#)