RDK-C : Extensible SDK support on Raspberrypi Yocto 3.1 dunfell build

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

Initialization/sync

```
Initialization:
$ repo init -u https://code.rdkcentral.com/r/manifests -m rdkc-nosrc.xml -b dunfell
                                ( or )
$ repo init -u https://code.rdkcentral.com/r/manifests -m rdkc-extsrc.xml -b dunfell

Download/Sync:
$ repo sync -j `nproc` --no-clone-bundle --no-tag
```

Populate eSDK

eSDK Population

```
$ MACHINE=raspberrypi3-rdk-camera source meta-cmf-raspberrypi/setup-environment
$ bitbake rdk-generic-camera-image -c populate_sdk_ext
```

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

eSDK 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

```
xxxxxxx@dvm-ch2g-yocto31-007:~/yyyyyy/31Jan21/build-raspberrypi3-rdk-camera/tmp/deploy/sdk$ ./rdk-glibc-x86_64-arm-toolchain-ext-2.0.s
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-cortexa7t2hf-neon-vfpv4-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.

eSDK Environment setup

```
# change directory to the installed path
cd ~/rdk_sdk
# setup the eSDK environemnt
source environment-setup-cortexa7t2hf-neon-vfpv4-rdk-linux-gnueabi
```

New Components

Add Component

Add Component

```
# way 1
# add a new recipe with URL
devtool add <recipe_name> <source URL>

Example:
# devtool add sctplib git://github.com/rdkcteam/usrsctplib

# way 2
# add a new recipe with external source directory
devtool add <recipe_name> <absolute path>

Example:
# devtool add sctplib /path/for/source/directory
```

Edit Component's recipe

Edit recipe's

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

Example:
# devtool edit-recipe sctplib
```

Build Components

Build component

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

Example:
# devtool build sctplib
```

Deploy binaries

Deploy binary

```
# Pre-requisite: add a new recipe using devtool
devtool deploy-target <recipe_name> <target_path>

Example:
# devtool deploy-target sctplib root@192.168.xx.xx
```

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.

Build image

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

Example:
# devtool build-image rdk-generic-camera-image
```