RPI 4B Model Reference Platform with 64bit Arch

- Host Setup
- Repo Setup
- Yocto Build Steps
- Validated Functionalities
- OEM Side Changes Required
- For DBUS Enabling support
- System Testing
    - E2E Validation Test Results
- Performance comparison metrics between RPi4 64 bit Vs 32 bit

Host Setup

The OpenEmbedded build system should be able to run on Ubuntu 18.04 distribution/other compatible linux distribution with the following versions for Git, tar, and Python.

- Git 1.8.3.1 or greater
- tar 1.27 or greater
- Python 3.4.0 or greater
- Coreutils(E.g realpath)

Note: You should also have about 50 Gbytes of free disk space for building images.

The essential packages you need for a supported Ubuntu or Debian distribution are shown in the following command:

```
$ 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 python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev \
  pylint3 xterm bmap-tools

$ sudo apt-get install git cmake autoconf texinfo openjdk-8-jdk openjdk-8-jre \
  m4 libtool libtool-bin curl pkg-config lib32z1 doxygen
```

Repo Setup

In order to use Yocto build system, the repo tool must be properly installed on the machine.

To install Repo make sure you have a /bin directory in your home directory and that it is included in your path

Repo Setup Steps

```
$ mkdir ~/bin
$ PATH=~/bin:$PATH
Download the repo tool and ensure that it is executable
$ curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

Note: it is also recommended to put credentials in ~/.netrc when interacting with repo.

A sample **~/.netrc** file is illustrated below

.netrc

```
machine code.rdkcentral.com login YOUR_USERNAME password YOUR_PASSWORD
```

Yocto Build Steps

To build, follow below instructions

Build Steps

```
$ mkdir <workspace dir>
$ cd <workspace dir>
$ repo init -u https://code.rdkcentral.com/r/manifests -b dunfell -m rdkb-extsrc.xml
$ repo sync -j`nproc` --no-clone-bundle
$ MACHINE=raspberrypi4-64-rdk-broadband source meta-cmf-raspberrypi/setup-environment
$ bitbake rdk-generic-broadband-image
```

Flashing Procedure

Following command can be used to flash the RPI image to sd card using linux machine . bmap tool should be available in linux

Flash command

```
bzip2 -d <path to ImageName.wic.bz2>
sudo -E bmaptool copy --nobmap <path to ImageName.wic> <path to SD card space>

Example:

$ bzip2 -d rdk-generic-broadband-image-raspberrypi4-64-rdk-broadband.wic.bz2
$ sudo -E bmaptool copy --nobmap rdk-generic-broadband-image-raspberrypi4-64-rdk-broadband.wic /dev/sdb
```

Validated Functionalities

No	Feature	Supported
1	LAN Connected Devices-Ethernet	✅
2	WAN Connected Devices-Wi-Fi	✅
3	Parental Control	✅
4	Firewall settings	✅
5	Advanced Config: Port Triggering	❌
6	Advanced Config: Port Forwarding	✅
7	Advanced Config: Remote Management	✅
8	Advanced Config: DMZ	✅
9	Xfinity Wi-Fi 2.4/5 GHz – Public Hotspot	✅
10	Test and Diagnostics	✅
11	Local WebUI Configuration	✅
12	Factory Reset	✅
13	DHCP /Reserved IP	✅
14	EthWan	❌
15	Eth Agent	✅
16	2.4 GHz Band Support	✅
17	5 GHz Band Support	✅
18	Bridge Mode Support	✅
19	Persistent Storage Management	✅
20	WebPA for Comcast,community	✅

21	Lost and Found	✗
22	Bluetooth	✗
22	Harvester Support	✗
23	TR-69	✗
24	SNMP	✓
27	Boot time data measurement	✗
28	Wireless Protection Setup(WPS)	✓
29	Captive Portal	✓
30	Wi-Fi MAC Filtering	✗
31	Log Rotation Support	✓
32	Firmware Upgrade Support	✓
33	Multiboot Support	✗
34	Telemetry Support	✓
35	IPV6	✗
36	Log level control through CcspLogAgent	✓
37	Band Steering using RSSI	✓
38	Backup and Restore	✗
39	SelfHeal	✓
40	Password-Reset	✓
41	WebUI JST	✓
42	Telemetry 2	✓
43	RdkWanManager	✓
44	RdkFwUpgradeManager	✓

OEM Side Changes Required

Global CFLAGS

NOTE: This can be added in ccsp_common inc files. So, will be available for all ccsp components.

```
CFLAGS_append_aarch64 = " -D_64BIT_ARCH_SUPPORT_ "
```

For DBUS Enabling support

By default, rbus is enabled in rpi4 64b build. For dbus enabling in rpi4 64b build, please follow the below steps,

```
Delete the below lines from below file's,

File path : ~/meta-cmf-raspberrypi/meta-rdk-broadband/recipes-ccsp/util/utopia.bbappend
    touch \/nvram\/rbus_support ' ${D}${sysconfdir}/utopia/utopia_init.sh

File path : ~/meta-cmf-raspberrypi/recipes-core/dbus/dbus_%.bbappend
do_install_append_aarch64_broadband ()
{
# Removing dbus service in 64bit arch alone
 rm ${D}${systemd_system_unitdir}/dbus.service
}
```

System Testing

E2E Validation Test Results

System Test Plan and Report - Broadband - 2021 - M10

System Test Plan and Report - RDK Broadband - 2022 - M1

Performance comparison metrics between RPi4 64 bit Vs 32 bit

- **Performance comparison metrics-RPi4-64 bit Vs 32 bit.ods**