SYSLIB

Release Notes

Applies to Product Release: 4.00.00.00-Alpha12-drop2
Publication Date: June 13, 2015

Texas Instruments

Document License

Contents

Release Notes

SYSLIB 4.00.00.00-Alpha

1 INTRODUCTION

1.1 Overview

This document provides the release information for the SYSLIB software package. The SYSLIB package includes the following:-

- SYSLIB Release Notes
- SYSLIB User's Guide
- Source code of all SYSLIB components
- Pre-built libraries (Little Endian) of all SYSLIB components
- API reference guide
- Software Manifest

This is an engineering tested alpha release package. Release notes from previous releases are also available in the release notes archive directory

2 RELEASE OVERVIEW

2.1 Hardware Device Support

The device and platforms tested for this release include:

- K2H
- K2K
- K2L

Please review the Device section for more details.

2.2 Components and Tools

The SYSLIB package is verified/tested using the **MCSDK 3.01.02.05** package. Please refer to the MCSDK Release notes for a list of all the component information. The following is the list of additional packages which were used to test the release:

1. SNOW3G 1.0.0.2
2. CUIA 1.01.00.06 Custom
3. UIA 2_00_03_40_eng
4. SA3GPP Enabler 3.0.0.0

The SYSLIB supports **only the RT kernel** from the MCSDK release. Please use the RT DEVKIT for the development of user space applications.

2.3 Licensing

Please refer to the software manifest

2.4 MCSDK Patches

The section documents the MCSDK Patches which need to be added to the base MCSDK release.

2.4.1 Memory Reserve Size

Please ensure that the following environment variable is defined and saved in the UBOOT environment:-

```
setenv mem_reserve 1536M
```

This will ensure that the kernel reserved the higher order 1.5GB of memory for the DSP. Failure to do so will result in the kernel overwriting DSP memory. Application developers can modify and customize the DSP & ARM memory map. The default DSP SYSLIB memory map which is released in the `SYSLIB_INSTALL_PATH/ti/platforms` assumes the above reservation.

2.4.2 UIO Kernel Module

Please refer to the MCSDK Patches (`SYSLIB_INSTALL_PATH/ti/mcsdk_patches`) directory and apply the following UIO Kernel patch:

File Name	Issue	How to patch

`uio_module_drv.ko`	SCLTE-1892: Random Kernel crashes and lockups under load. This is only applicable for Queue Pend Interrupts.	Remove and unload the kernel module
		`rmmod uio_module_drv.ko`
		`rm /lib/modules/3.10.10-rt7/extra/uio_module_drv.ko`
		TFTP the kernel module from `mcsdk_patches`

		onto the EVM into the `/lib/modules/3.10.10-rt7/extra/` directory Reboot the EVM (Driver sources available from: http://git.ti.com/cgit/cgit.cgi/keystone-linux/uio-module-drv.git/, Tag: 01.00.02.02_eng) **NOTE:** This kernel module is applicable only for the RT Kernel. Please rebuild from the source for non-RT Kernel. The module will apply to the base MCSDK released kernel. Custom kernel builds will also need to rebuild from the driver sources.

2.4.3 Installing the SA 3GPP Enabler

As mentioned above the SA3GPP enabler is a prerequisite. While installing the SA3GPP; the installer will request for the PDK Path. This will ensure that the SA3GPP Installer will be correctly found and the DSP applications will be built properly. However the installer does not update the RT Linux development kit and so the following manual steps need to be done:

- Create directory `sa3gppEnabler` under the `ARAGODIR/include/ti/drv/sa`
- Copy the `sa3gpp.h` from the `PDK_INSTALL_PATH/ti/drv/sa/sa3gppEnabler` to the `ARAGODIR/include/ti/drv/sa/sa3gppEnabler`
- Copy the `sa3gppver.h` from the `PDK_INSTALL_PATH/ti/drv/sa/sa3gppEnabler` to the `ARAGODIR/include/ti/drv/sa/sa3gppEnabler`
- Copy the library `libsa3gpp.a` from the `PDK_INSTALL_PATH/ti/drv/sa/sa3gppEnabler/lib/armv7` to the `ARAGODIR/lib` folder

NOTE: This step is required to build the NETFP Server since the NETFP Server initializes the 3GPP services. The default NETFP Server executable enables 3GPP services.

2.4.4 DTS File Updates

NOTE: Please integrate the SYSLIB released DTS files for the specific device with your application and always update the kernel DTB files and SYSLIB RMv2 DTB files. Failure to do so will result in out of the box failures.

2.4.4.1 K2H/K2K

The kernel DTS files have been modified for the following features:-

- GIC Queues 8722 to 8735 were originally reserved for the Linux kernel. These queues are not used by the Linux kernel so these have been marked as unreserved and could not be used by the ARM applications
- Wiring of the GIC Queue and INTC_SET2 interrupt queues from using the UIO module.

Along with the kernel DTS file; the SYSLIB RMv2 files have also been modified for the following features:-

- GIC Queues 8722 onwards have been marked as usable
- INTC_SET2 queues have been allocated to ARM
- Wildcarding support
- Simplified L2 and L3 QoS shapers. This is for illustration only. Customers are recommended to modify the shapers as per their requirements.

2.4.4.2 K2L

The kernel DTS files have been modified for the following features:-

- GIC Queues 546 to 559 were originally reserved for the Linux kernel. These queues are not used by the Linux kernel so these have been marked as unreserved and could not be used by the ARM applications
- Wiring of the GIC Queue and SOC_SET_1 interrupt queues from using the UIO module.

Along with the kernel DTS file; the SYSLIB RMv2 files have also been modified for the following features:-

- GIC Queues 546 onwards have been marked as usable
- SOC-SET1 queues have been allocated to ARM
- Wildcarding support
- Simplified L2 and L3 QoS shapers. This is for illustration only. Customers are recommended to modify the shapers as per their requirements.

3 What’s new

3.1 New Features

The section documents the new features supported in the release:-

3.1.1 NETFP Proxy

The NETFP Proxy module internally has been redesigned into clearly demarcated sub-modules for clarity. These changes are internal and do not affect the users of the proxy.

However application developers which are writing their own plugin need to be aware that the `NetfpProxy_pluginInit` should now return a descriptor on success. The NETFP Proxy will now invoke the plugin run function only if there is data present on the descriptor.

3.1.2 DAT Memory Logging

Memory logging is moved out from DAT module and becomes a stand along Module.

- MEMLOG instance needs to be created on a DSP core or ARM process in order for MEMLOG module to function. It can be done through Domain or calling MEMLOG create instance API.
- To enable memory logging in DAT, a MEMLOG channel needs to be created and passed to DAT when creating DAT producer.
- MEMLOG memory block information is stored in Named database. MEMLOG controller can be created on ARM to query the memory block information. It can also be used to start/stop memory logging of a logging producer.

3.2 Bug Fixes

Issue Type	Key	Summary
Bug	SCLTE-2162	Can’t recover and ARP entry from STALE (after failed PING) or FAILED state
Bug	SCLTE-1898	LTE Demos Not Supported

3.3 API changes:

3.3.1 Configuration of L3 Shaper

For consistency the `Netfp_setupL3Shaper` API has been obsoleted. Please use the `Netfp_setIfOpt` with the type set to `Netfp_Option_L3_QOS`. This mechanism now allows the ability to get/set the L3 QOS Configuration.

3.3.2 Support for multiple IPv4 address

It is now possible to add multiple IPv4 address to an interface. This implies that the definition `NETFP_MAX_IPV6_ADDRESS` is now replaced with the more generic `NETFP_MAX_IP_ADDRESS` definition.

3.4 Feature list in JIRA:

Issue Type	Key	Summary
Story	SCLTE-2189	NetFP route resolution. Functional separation
Story	SCLTE-2202	NetFP support of physical interface name change

3.5 Known Issues:

Issue Type	Key	Summary	Priority
SDOCM00114690		Memory leak in network drivers on repeated ifconfig down and up.	Major

Issue Type	Key	Summary
Bug	SCLTE-2183	StopOffload or ipsec down fails to delete Security Policy
Bug	SCLTE-2177	Support needed for offloading of Ipv6 without an IPv4 address
Bug	SCLTE-2051	Packet loss was observed during re-keying when application deletes old SA prematurely
Bug	SCLTE-2050	NetfpServer Crashes when trying to delete an already deleted fastpath
Bug	SCLTE-2019	Fixed 1GHz clock used in DAT_TIME_ELAPSED
Bug	SCLTE-1999	Netfp Proxy - Monitoring of aliased interface is not supported
Bug	SCLTE-1923	Reestablishment fails in Netfp when tried second time before the first one succeeds

Bug	SCLTE-1612	while(1) loop in msgcom code needs to be removed.

4 RELEASE BUILDING

SYSLIB release build & environment configuration scripts which are located in the SYSLIB Install directory `scripts` folder. Please setup the following environment variables:-

```
export
ARMTOOLS_INSTALL_PATH=/home/a0868491/tools/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux

export
ARAGO_INSTALL_PATH=/home/a0868491/ti/mcsdk_linux_3_01_02_05_devkit_rt/sysroots
/cortexa15hf-vfp-neon-3.8-oe-linux-gnueabi

export CGT_INSTALL_PATH=/home/a0868491/ti/cgt_7.4.8

export XDC_INSTALL_PATH=~/ti/xdctools_3_30_04_52

export PDK_INSTALL_PATH=~/ti/pdk_keystone2_3_01_02_05/packages

export SNOW3G_INSTALL_PATH=~/ti/snow3g_1_0_0_2/packages

export UIA_INSTALL_PATH=~/ti/uia_2_00_03_40_eng/packages

export INSTALL_JAMMER_INSTALL_PATH=~/tools/installjammer-1.2.15

export BIOS_INSTALL_PATH=~/ti/bios_6_40_04_47/packages

export IPC_INSTALL_PATH=~/ti/ipc_3_30_01_12/packages

export CUIA_INSTALL_PATH=~/tools/cuia_1_01_00_06Custom

export SYSLIB_DEVICE=k2h

export SYSLIB_INSTALL_PATH=~/ti/syslib_4_00_00_00_alpha12/packages
```

The environment variables are illustrative and should be modified by the customer as per their install paths. Once configured please setup the build environment by executing the following script:-

```
cd scripts

source setupenv.sh
```

This will setup the build environment and will also sanity check to make sure that all the required environment variables are configured.

4.1 Building the ARM Libraries, Servers & Unit Tests

Once the build environment is configured; please execute the following script to build the libraries for a specific device:-

```
cd scripts
source dev.sh <DEV_NAME> <ARM_BUILD> <DSP_BUILD> <DEMO_BUILD> <ARM_UNIT_TEST>
<DSP_UNIT_TEST>
```

Argument	Description
DEV_NAME	Name of the device for which the builds need to be done. Valid values are k2h, k2k and k2l
ARM_BUILD	Set to 1 to build the ARM libraries and standard SYSLIB Servers
DSP_BUILD	**Always set to 0**. To build the DSP Libraries please refer below
DEMO_BUILD	Set to 1 to build the DEMO for the specific device
ARM_UNIT_TEST	Set to 1 to build the ARM Unit Test for all the SYSLIB modules
DSP_UNIT_TEST	Set to 1 to build the DSP Unit Test for all the SYSLIB modules

Example: To rebuild the ARM Libraries/applications for K2H

```
source dev.sh k2h 1 0 0 0 0
```

Example: To build the ARM Libraries & Unit Tests for K2L

```
source dev.sh k2l 1 0 0 1 0
```

4.2 Building the DSP Libraries

Ensure that the SYSLIB_DEVICE is correctly configured in the environment variable. The example below selects the device as K2L

```
export SYSLIB_DEVICE=k2l
```

Modify the environment variable

```
export SYSLIB_INSTALL_PATH=~/ti/syslib_4_00_00_00_alpha12
```

NOTE: There is no /`packages` at the end of the SYSLIB_INSTALL_PATH

Once configured please setup the build environment again by executing the following script:-

```
cd scripts
source setupenv.sh
```

To rebuild SYSLIB DSP Libraries; please do the following from the top level directory:-

```
xdc clean –PR .
xdc –PR .
```

4.3 Building the DSP Unit Tests

DSP Unit Tests are built using the script described above. **Example:** To build all the DSP Unit Tests for K2L

```
source dev.sh k2l 0 0 0 0 1
```

5 Device Support

Please read the following section which documents details about each SYSLIB supported device:

5.1 K2H

Kernel DTS Files	**ti/runtime/resmgr/dts/k2h**
RMv2 DTS Files	ti/runtime/resmgr/dts/k2h
DSP Memory Map	ti/runtime/platforms/tmdxevm6638lxe
ARM Compilation Flags	-D_LITTLE_ENDIAN -D__ARMv7 -DDEVICE_K2 –DDEVICE_K2H -D_GNU_SOURCE -D_VIRTUAL_ADDR_SUPPORT
DSP Compilation Flags	--define=DEVICE_K2 --define=DEVICE_K2H
PA Library on DSP	var Pa = xdc.useModule('ti.drv.pa.Settings'); Pa.deviceType = “k2h”
PA Library on ARM	-lpa
SOC Sample configuration file	ti/apps/soc_init/soc_k2h.conf
NETFP Master configuration file	ti/apps/netfp_master/netfp.conf
Library & Executable Suffix	_k2h

5.2 K2K

Kernel DTS Files	**ti/runtime/resmgr/dts/k2h**
RMv2 DTS Files	ti/runtime/resmgr/dts/k2h
DSP Memory Map	ti/runtime/platforms/tmdxevm6638lxe
ARM Compilation Flags	-D_LITTLE_ENDIAN -D__ARMv7 -DDEVICE_K2 –DDEVICE_K2K -D_GNU_SOURCE -D_VIRTUAL_ADDR_SUPPORT
DSP Compilation Flags	--define=DEVICE_K2 --define=DEVICE_K2K
PA Library on DSP	var Pa = xdc.useModule('ti.drv.pa.Settings'); Pa.deviceType = “k2k”
PA Library on ARM	-lpa
SOC Sample configuration	ti/apps/soc_init/soc_k2k.conf

file	
NETFP Master configuration file	ti/apps/netfp_master/netfp.conf
Library & Executable Suffix	_k2k

5.3 K2L

Kernel DTS Files	**ti/runtime/resmgr/dts/k2l**
RMv2 DTS Files	ti/runtime/resmgr/dts/k2l
DSP Memory Map	ti/runtime/platforms/k2l
ARM Compilation Flags	-D_LITTLE_ENDIAN -D__ARMv7 -DDEVICE_K2 –DDEVICE_K2L -D_GNU_SOURCE -D_VIRTUAL_ADDR_SUPPORT
DSP Compilation Flags	--define=DEVICE_K2 --define=DEVICE_K2L

PA Library on DSP	var Pa = xdc.useModule('ti.drv.pa.Settings');
	Pa.deviceType = “k2l”

PA Library on ARM	-lpa2
SOC Sample configuration file	ti/apps/soc_init/soc_k2l.conf
NETFP Master configuration file	ti/apps/netfp_master/netfp_k2l.conf
Library & Executable Suffix	_k2l

NOTE: The PA library on K2L is different. Including the wrong library will result in run time failures.