

SYSLIB

Release Notes

Applies to Product Release: 4.00.00.00-Alpha8
Publication Date: December 19, 2014



Document License

This work is licensed under the Creative Commons Attribution-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Copyright (C) 2012 Texas Instruments Incorporated - <http://www.ti.com>

Contents

1	INTRODUCTION.....	1
1.1	Overview.....	1
2	RELEASE OVERVIEW	1
2.1	Hardware Device Support.....	1
2.2	Components and Tools.....	1
2.3	Licensing	2
2.4	MCSDK Patches	2
3	What's new.....	3
3.1	New Features.....	3
3.2	Bug Fixes.....	4
3.3	API changes:	4
4	RELEASE BUILDING.....	4

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 (both Big and 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:

- TMDXEVM6638Ixe

2.2 Components and Tools

The SYSLIB package is verified/tested using the MCSDK 3.1.00.03 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 1_03_06_17_eng](#)
4. [SA3GPP Enabler 3.0.0.0](#)

The SYSLIB supports only the RT kernel from the MCSDK release. Similarly 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

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.

Please refer to the MCSDK patches (`SYSLIB_INSTALL_PATH/ti/mcsdk_patches`) directory and manually apply the following patches:

File Name	Issue	How to patch
libhplib-no-osal_k2h.so.1.0.0	HPLIB bitmap allocator overruns and corrupts allocated memory	TFTP the shared library object onto the EVM into the <code>/usr/lib</code> directory.
libqmss_k2h.so.1.0.0	<ol style="list-style-type: none"> 1. Incorrect programming of the QMSS Memory region 2. Accumulator programming race condition 	TFTP the shared library object onto the EVM into the <code>/usr/lib</code> directory.
ti.drv.qmss.ae66	<ol style="list-style-type: none"> 1. Incorrect programming of 	Copy the files and overwrite the default library files which are present in the

ti.drv.qmss.ae66e	the QMSS Memory region 2. Accumulator programming race condition	PDK_INSTALL_PATH/ti/drv/qmss/lib/k2h/c66
-------------------	--	--

3 What's new

3.1 New Features

The following features were added to this release:

1. NETFP IPv6 support:

The NETFP module now supports IPv6. This allows application to create/delete IPv6 fast paths and allow applications the ability to send/receive UDP/GTPU traffic over an IPv6 socket. The NETFP module supports all the combinations of tunnels:

- 6in6
- 4in6
- 6in4
- 4in4

2. NETFP Source base routing:

The NETFP module now allows applications to route packets not only on the basis of the destination IP address but also on the basis of the source IP address. The feature required that all routing decisions are now taken by the NETFP Proxy while consulting the Linux routing tables. In order to support this feature there have been some changes in the API behavior listed below.

3. DAT Memory Logging

Memory logging feature for a DAT producer is added in this release. In addition to shipping to NETFP socket, a producer buffer can also be logged into a contiguous memory block given by application. Memory block can be dumped into a file from Linux and converted to a SA readable bin file by post-processing script. A producer controller should be created on ARM side to get the memory block information. Start/stop logging APIs are provided to stop logging before dumping the memory and re-start after it is done.

4. NETFP 3GPP Re-establishment

The NETFP module now supports re-establishment. To re-establish a user, the application must perform the 3 step process for each active LTE channel.

- Suspend LTE channel
- Reconfigure LTE channel
- Resume LTE channel

5. Support for FAPI 2.0

The FAPI Tracing library has been updated to use the FAPI 2.0 interface definition.

3.2 Bug Fixes

1. PKTLIB: Release data buffer ownership

It was observed that with L2 caches enabled and using a BIOS `HeapMem` heap; the data buffers allocated to a PKTLIB heap could be corrupted. This was because the BIOS `HeapMem` object stores meta-information inside the data buffer with respect to the amount of free memory. When the buffer is passed to the PKTLIB heap the cache line is dirty and could result in the data buffer being overwritten as a result of standard cache eviction. The PKTLIB module now ensures that the data buffers in the cache & memory are synchronized before pushing the packets to the hardware queues.

3.3 API changes:

1. Netfp_addFastPath

While adding an Egress fast path to the NETFP module it is now required to pass the route handle to the used in the fast path configuration. It is recommended that the route handle be derived using the `Netfp_isRouteResolved` API.

3.4 Feature list in JIRA :

The release feature list accounted for in TI JIRA database:

Key	Summary
SCLTE-915	SCLTE-911 DAT for syslib 4.0: LTE Demo
SCLTE-1426	Re-Establishment Support
SCLTE-1218	FastPath fragmentation handling.
SCLTE-1226	IPSec in tunnel mode. Support for IPv4 (outer), IPv6 (inner)

SCLTE-1225	IP Fragmentation. IPv4 and IPv6 support
SCLTE-1224	IPSec in tunnel mode. Support for IPv6 (outer), IPv6 (inner)
SCLTE-1223	IP Reassembly. IPv4 and IPv6 support
SCLTE-1062	IPSec in tunnel mode. Support for IPv6 (outer), IPv4 (inner)
SCLTE-1401	DAT for Syslib 4.0: LogSync on ARM & dynamic logger instance support
SCLTE-1399	DAT for Syslib 4.0: Adding support for memory logging
SCLTE-1060	NetFP Networking. Multiple Routing tables
SCLTE-1061	NetFP Networking. Source based routing
SCLTE-1493	FAPI Tracing verbosity control
SCLTE-1492	FAPI Tracing object deletion
SCLTE-1491	FAPI Tracing Streaming
SCLTE-1490	FAPI Tracing object instantiation
SCLTE-1489	FAPI tracing library support for FAPI 2.0
SCLTE-914	SCLTE-911 DAT for syslib 4.0: Implemetation/Validation
SCLTE-911	DAT redesign for syslib 4.0

4 RELEASE BUILDING

Please install the MCSDK 3.01.00.03 release on a Linux machine. Please setup the following environment variables:-

```
export ARMTTOOLS_INSTALL_PATH=~/.tools/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux

export
ARAGO_INSTALL_PATH=~/.ti/mcsdk_3_01_00_03_devkit_rt/sysroots/cortexa15hf-vfp-neon-3.8-oe-linux-gnueabi

export CGT_INSTALL_PATH=~/.ti/cgt_7.4.4

export XDC_INSTALL_PATH=~/.ti/xdctools_3_25_06_96

export PDK_INSTALL_PATH=~/.ti/pdk_keystone2_3_01_00_03/packages

export SYSLIB_INSTALL_PATH=~/.ti/syslib4.x/packages

export SNOW3G_INSTALL_PATH=~/.ti/snow3g_1_0_0_2/packages

export UIA_INSTALL_PATH=~/.ti/uia_1_03_06_17_eng/packages

export INSTALL_JAMMER_INSTALL_PATH=~/.tools/installjammer-1.2.15

export CUIA_INSTALL_PATH=~/.tools/cuia_1_01_00_06Custom

export SYSLIB_DEVICE=k2h
```

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:-
SYSLIB_INSTALL_PATH/scripts/setupenv.sh. This will setup the build environment and will also sanity check to make sure that all the required environment variables are configured.

Please execute the release script (SYSLIB_INSTALL_PATH/scripts/release.sh) as follows:-

```
source ./release.sh 1 0 1
```

The script takes 3 arguments:-

1. Argument1: Build the SYSLIB ARM libraries.
2. Argument2: Build the SYSLIB DSP libraries. [Should always be set to 0]
3. Argument3: Build the SYSLIB LTE Demo

To rebuild the DSP libraries; please execute the following path from the SYSLIB_INSTALL_PATH

```
xdc clean -PR .  
xdc -PR .
```

For information on how to build the DSP and ARM unit tests and for execution instructions please refer to the SYSLIB Unit Test documentation.