

Introduction

This document provides a basic guideline for using IDT's Evaluation Software Development Kit ("EVAL_SDK_4_1"). This open-source software is freely downloadable from my.IDT.com and provides the C-code APIs and example test code for all of IDT's sRIO-based devices including the 70K2000 Pre-Processing Switch (PPS), 80KSW0001 PPS Gen 2, 80KSW0002 Central Packet Switch (CPS-16), 80KSW0003 (CPS-8), 80KSW0004 (CPS-12), and 10G Serial Buffer (80KSBR201). This document provides an overview of the software layering, API interface, and a quick explanation on how to set up and running out of the box with the example tests.

The software environment is Texas Instruments' (TI) Code Composer Studio (CCS) 3.3, Freescale CodeWarrior version 5.7.0 or higher version. The target board can be IDT's AMC70K2000, TI's TMS320C6455 DSK or Mezzanine board, or Freescale MSC8144 ADS board, or customized board with proper TI/Freescale DSP device. The CPS / PPS device may be on the AMC70K2000, IDT's PPS / CPS evaluation carrier (aka. "pumpkin board" such as EVC80KSW0002) or customized board. The EVAL_SDK also supports IDT's Serial Buffer daughter module for AMC70K2000.

Most hardware mentioned in this application note can be purchased from IDT's partner, Silicon Turnkey Express (STx).

Additional documentation for IDT devices can be found for the associated device portal page on my.IDT.com. For any other support please contact the Technical Support directly. Contact info is shown at the end of this document.

Software Layering

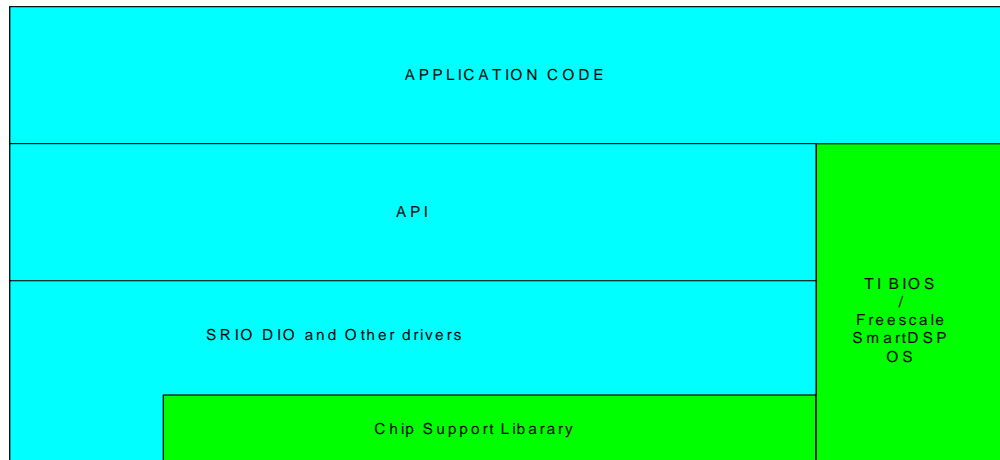


Figure 1.1 SDK Software Layering

As shown in the diagram, IDT provides device access drivers, API, application examples. Both Chip Support Library (CSL) and BIOS / SmartDSP OS are provided by TI or Freescale. Full documentation including API description, data structures, and inputs and outputs for IDT's APIs are provided in the `INSTALL_DIR\CPS\CAPI\docs\html\index.html` for each of the IDT devices. Refer to `INSTALL_DIR\SRIO_SDK\SRIO_DIO\docs\html\index.html` for the TI DSP sRIO driver documentation.

Notes

API Interface

The API provides a set of functions to program and manage the PPS, CPS, and Serial Buffer devices. These functions are written in C language that can be called by application program. The use of IDT's API software is optional, but it provides a quick platform to get the PPS, CPS, and Serial Buffer up and running. The code is open source so that the user may customize it or simply use the code as reference for developing custom API calls. These APIs are written such that they can easily be ported to other environments/processors.

The use of the API is very simple. Simply include top level header file (pps_capi.h for PPS/PPS Gen-2/CPS, or sb_capi.h for Serial Buffer) that exports all the API functions. There are documents describing the detailed functions, input and output parameters, and data structures.

INSTALL_DIR\CPS\CAPI\docs\html\index.html is the main documentation file for PPS/PPS Gen-2/CPS, which provides the user easy navigation to find all necessary information. Here is a list of some frequently used functions:

- ◆ *PPS_Device_Create*
This function creates a handle that points to the API device, all the future communications will use this handle to locate functions.
- ◆ *PPS_Device_Register*
This function instantiates the appropriate APIs according to the device type. For example, it registers CPS drivers if the device type is CPS.
- ◆ *PPS_Device_Destroy*
This function releases the API instance.
- ◆ *Configure_Quads*
This function configures the specified sRIO QUAD. The QUAD control information will be written into this QUAD control register.
- ◆ *Configure_Ports*
This call passes the appropriate data structure to the appropriate port's control CSRs per sRIO standard requirements.
- ◆ *PPS_Add_Delete_Routes*
This function adds or removes a route for a specified destination ID and route number. For CPS and PPS Gen 2, it supports both common and individual port route tables. For PPS, there is only one common route table available.
- ◆ *PPS_Add_Mcast_Route*
This function adds user-specified individual member ports to a multicast (MC) list.
- ◆ *PPS_Delete_Mcast_Route*
This function removes user-specified individual member ports from a multicast (MC) list.
- ◆ *PPS_Add_Delete_Err_Source*
This function enables or disables one or more error reporting sources in CPS or PPS. There are multiple functional reporting sources in CPS/PPS. For detailed error source code, please refer to the CPS/PPS user's manual.
- ◆ *PPS_Create_Special_Err_Filter*
This function enables a special error filter which allows the Error Management block of the CPS or PPS to filter errors coming from particular sources, and to take special action for particular errors. For more error handling information refer to the User's Manual Error Handling chapter for the CPS or PPS.
- ◆ *PPS_Read_Err_FIFO*
This function reads the contents of the Error log memory. This allows the user to determine if any errors have occurred, the source of the error, etc.

For Serial Buffer API functions, please refer to INSTALL_DIR\SB\CAPI\docs\html\index.html.

Notes

Running the Examples

Where to find the examples:

There are examples for PPS, CPS, PPS Gen 2, and Serial Buffer.

The following directories are respectively for the TI TMS320C6455/6482 DSP:

INSTALL_DIR\examples\pps_capi_demo

INSTALL_DIR\examples\pps_cps_pumpkin_capi_demo

INSTALL_DIR\examples\pps_pl_pumpkin_capi_demo

INSTALL_DIR\examples\SB_capi_demo

INSTALL_DIR\examples\sb2_amc_traffic_gen

The following directories are for the Freescale MSC8144 DSP:

INSTALL_DIR\examples\Freescale

Please refer to "INSTALL_DIR\release_note_4_1.txt" for more detailed info.

What the examples do:

These examples demonstrate the basic functions of PPS on AMC70K2000 or Evaluation Carrier, PPS Gen 2 on AMC80KSW0001 or Evaluation Carrier, CPS on Evaluation Carrier and Serial Buffer on Evaluation Carrier. For these platforms, the examples will run as-is. They can readily be re-configured to run on any other platform. We take a look at the PPS CAPI example. Application_Main_Thread in main.c is the task created by BIOS. This top level procedure provides the entry of all our example test applications.

```
Void application_Main_Thread() {
BasicAccessTest();
BasicAccessEnhTest();
BasicPpscTest();
DomRoutingTest();
EnhGppscTest();
}
```

- ◆ *User should comment out any tests they do not want to run.*
- ◆ *BasicAccessTest demonstrates how to initialize the DSP's sRIO peripheral, configure PPS sRIO port, routing table and then copies data from one DSP memory location to another by routing (looping back) Swrite/Nwrite packets through the switch, then verify data by Nreads. This example is preconfigured to run on DSP0 on the AMC70K2000. It can readily be modified through simple port configuration and number change to run on a TI Mezzanine board connected to an EVC70K2000 Pumpkin board.*
- ◆ *BasicAccessEnhTest configures the Quad in enhanced mode, then copies data from one DSP memory location to another by Swrite/Nwrite through the switch, then verifies by Nread*
- ◆ *BasicPPScTest configures the PPSc to receive Swrite packet, manipulate the samples and build the output frame by Swrite packet. Basically it demonstrates the PPSc capability*
- ◆ *EnhGPPScTest configures a set of PPScs as a group and does the same processing for incoming Swrite. Each PPSc of the group have individual output ports*
- ◆ *DomRoutingTest configures the routing table and sends packets through the switch*
- ◆ *BasicMasterModeTest in Serial Buffer example Configures the Serial buffer Queue in Master mode and then sends SWRITE packets through the queue*

Notes

Environment setup

For basic TI CCS setup and emulator driver configuration instructions for the AMC70K2000, please refer to the "AMC70K2000 Quick Start Guide" on my.IDT.com. These instructions are generically applicable to other boards and hardware configurations (such as the TI Mezzanine board connected to an evaluation carrier). Note that there are a wide variety of vendors and emulator types to choose from in support of TI's DSPs; configuration procedures for each emulator type are unique. Please consult the emulator-specific documentation for full configuration procedure and details. Detailed CCS configuration instructions should be obtained through Texas Instruments. Note: macro.ini should be copied to the CCS_INSTALL_DIR\cc\bin directory and update the "PATH" variables.

The detailed CodeWarrior configuration instructions should be obtained through Freescale.

Run the examples

Running the examples is straight forward.

Here is the basic flow and sequence to run a example for TI DSP.

- ◆ *Setup the hardware environment, it can be AMC/DSK powered by an Evaluation Carrier board or a customized board, or customized stand alone board.*
- ◆ *Startup CCS and open workshop for the correct DSP. Pay attention to which DSP number is selected.*
- ◆ *Open the project. Check the compile option, select DEBUG for AMC70K2000 or AMC80KSW0001, or DSK_DEBUG for TI EVM.*
- ◆ *Check the switch device ID and connected switch port number (the switch port number to which the target DSP is connected.)*
- ◆ *Ensure the DSP's port speed defined in the example test matches that of the speed used by the switch.*
- ◆ *Ensure the DSP's port configuration matches the quad configuration of the switch. For "Enhanced Quads" ensure the DSP is configure for 4 independent 1x ports (defined in the example code). Note that a mismatch of 4 1x port with a 1 4x port will NOT initialize correctly and the program will stall waiting for link initialization.*
- ◆ *Compile and build the program*
- ◆ *Load the out file*
- ◆ *Debug the code, you can set breakpoint, step in the function, etc. In the debug mode the output message window will give you a result*

Here is the basic flow and sequence to run a example for Freescale DSP.

- ◆ *Setup the hardware environment*
- ◆ *Startup CodeWarrior*
- ◆ *Open the project (*.mcp)*
- ◆ *Click "Run"*

FAQ

- ◆ *Why does it happen that when I open the project it can not find some files?*
 - This may happen if the CCS installation directory is different. The simplest way to solve this is to find the files and add them to the project. Ensure the CCS environment points to the appropriate directory. Ensure macro.ini is in the cclbin directory and "path" variables are set correctly.
- ◆ *Is there any difference when sRIO packet target address points to IRAM or DRAM?*
 - There is almost no difference. IRAM is suggested.
- ◆ *What should be done if the CCS can not find a GEL file?*
 - Find this file and set it for CCS

Notes

- ◆ *Is DSP connected to standard port or enhanced port?*
 - Check the default mode of the connected port according to the device's data sheet or user's manual. Simply select the mode in software by macro definition.
- ◆ *Can I run the API on multiple DSPs?*
 - Yes.
- ◆ *Can I run the API on multiple DSPs?*
 - Yes.
- ◆ *Does routing table configuration API working for all your switches?*
 - Yes. The APIs will appropriate configure routes for any standard sRIO switch.
- ◆ *Are your sRIO switch API standard? Will they work with any of your switches or preprocessing switches?*
 - Yes
- ◆ *How do I use API to reconfigure a QUAD without losing packet?*
 - Please refer to IDT's "Hot Swap" application note on my.IDT.com
- ◆ *How do I select the sRIO or I²C access mode?*
 - Use Attach_Dev_Access_Methods function. Set PPS_ACCESS = I2C_ACCESS or PPS_ACCESS = SRIO_ACCESS.
- ◆ *Who can I contact if I'm still having technical difficulties?*
 - Please see below for contact information for IDT technical support.

References

1. IDT 70K2000 Pre-Processing Switch Datasheet
2. IDT 70K2000 User's Manual
3. IDT80KSW0001 Datasheetl
4. IDT80KSW0001 User's Manual
5. IDT 80KSW0002 Datasheetl
6. IDT 80KSW0002 User's Manual
7. IDT 80KSBR201 Advanced Datasheet
8. IDT AMC70K2000 Quick Start Guide
9. IDT AMC70K2000 User's Guide
10. IDT EVC70K2000 Flyer
11. Texas Instruments TMS320C6455 Serial RapidIO User's Guide
12. Texas Instruments TMS320C6455 Chip Support Library API Reference Guide
13. Freescale Instruments MSC8144 Reference Manual

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.