Processor Audio SDK Setup Guide

TEXAS INSTRUMENTS

Rev. 0.30

08 Dec 2016

Revision History			
Version	**Date**	**Description of changes**	**Author(s)**
0.10	23-Aug-16	Initial version	Frank Livingston
0.20	30-Aug-16	Added gmake clean to UART LLD build instructions	Frank Livingston
0.30	08-Dec-16	Updated for AR2	Frank Livingston

Table of Contents

1 Introduction

This document provides installation and setup instructions for the Alpha-2 Release (AR2) of Processor Audio SDK (PASDK) 1.0 for K2G provided by Texas Instruments, Inc. (TI). AR2 incorporates several packages: (1) the Open Source (OS) package; (2) the Firmware Deliverable (FD) package; and (3) the Dolby Atmos IP packages. The AR2 features provided by these packages are detailed below.

1. OS package
    - Audio input drivers: HDMI input, and 8-channel ADC or S/PDIF (optical only) input.
    - Audio output drivers: up to 16-channel DAC output.
    - Control I/O drivers: UART, SPI and I2C.
    - PCM decoder (executing on A15).
    - PCM encoder with DEL and VOL ASP phases.
    - DM, FIL, SRC and AE0 ASP’s.
    - Framework supporting I-topology, decoders on ARM, and ASP’s on DSP.
2. FD package
    - SNG decoder (executing on ARM).
    - BM2, DEM and GEQ3 ASP’s.
3. Dolby Atmos IP packages
    - Dolby Intrinsics 1.8.1.1.
    - Dolby Digital Plus (DDP) decoder 4.7.2.
    - MAT-PCM decoder 2.1.0 and MAT-THD decoder 3.3.2.
    - Channel Audio Renderer (CAR) 1.0.0.
    - Object Audio Renderer (OAR) 1.0.0.
    - Bass Management (BMDA) from SIDK v1.2.

2 Required Hardware

The hardware listed below is required for complete AR2 functionality:

1. Mistral K2G Rev. D EVM (SPRW274).
2. Mistral K2G EVM Audio Daughter Card Rev. B (SPRW279).
3. Momentum Data Systems (MDS) HDMI repeater kit, including:
    - MDS HSR41P HDMI Repeater module Rev. A.
    - MDS DA10X Interface board (IFB) Rev. A.
    - 5V AC to USB power supply for IFB and HSR41P.
    - Short 26- and 34-wire ribbon cables for connecting HSR41P and IFB.
    - Long 34-wire ribbon cable for connecting IFB and Audio Daughter Card.

3 Software Installation

3.1 Code Composer Studio (CCS)

AR2 was developed using CCS 6.1.2.00015 on Windows. Download the Off-line Installer for this version of CCS at the following URL:

http://processors.wiki.ti.com/index.php/Download_CCS#Code_Composer_Studio_Version_6_Downloads

Install CCS to C:\ti. During CCS installation:

- For “Processor Support”, select “Multi Core Processors (all)”
- For “Select Debug Probes”, select “TI XDS Debug Probe Support” and “Spectrum Digital Debug Probes and Boards”

AR2 was developed using the code generation tools listed below. These tools are installed as part of CCS.

- GNU ARM Tools 4.8.0.2014q3-20140805
- C6000 v8 Compiler Tools: 8.1.0

3.2 Processor SDK-RTOS

AR2 was developed using Processor SDK-RTOS 2.0.2.11 for K2G on Windows. Release information for this version of Processor SDK-RTOS is provided at the URL below, along with links to the installer:

http://software-dl.ti.com/processor-sdk-rtos/esd/K2G/02_00_02_11/index_FDS.html

Install Processor SDK-RTOS to C:\ti. Among others, this should install the tools in the following list:

- SYS/BIOS 6.45.01.29
- XDAIS 7.24.00.04
- XDC Tools 3.32.00.06

3.3 Open Source Package

The AR2 OS package is distributed using a self-extracting executable. Follow the steps below to install the software.

1. Double click the executable to initiate the installation process. Continue to the License Agreement.
2. Accept the terms of the click-wrap license agreement.
3. Change the installation destination folder to c:\ti\processor_audio_sdk_1_00_00_01 in the Choose Destination Location window. Complete the installation.

Upon completion of the installation, the processor_audio_sdk_1_00_00_01 folder contains the sub-folders described in the table below.

Folder	**Contents**
docs\	User documentation
pasdk\	PASDK OS Dev (paf\) ARM application (common\, shared\, test_arm\ DSP application (common\, shared\, test_dsp\)
psdk_cust\	IPC engineering release 3.43 (ipc_3_43_00_00_eng\) Enhanced PDK K2G 1.0.1 (pdk_k2g_1_0_1_0_eng\)
scripts\	Build scripts
tools\	Alpha communication tools

3.4 Firmware Deliverable Package

The AR2 FD package is distributed as a .zip archive. To install the software, unzip the package .zip file to c:\ti\processor_audio_sdk_1_00_00_01\pasdk\paf. The FD package root folder is “pa”, so unzipping the file should overwrite the pa folder in paf and overwrite or create the sub-folders described in the table below.

Folder	**Contents**
pa\asp\rel\	ASP IP (bm2, dem, geq3) header files
pa\build	ARM and DSP IP libraries
pa\dec\rel\sng1	Decoder IP (sng1) header files
pa\docs	User documentation

3.5 Dolby IP Packages

The AR2 Dolby IP packages are distributed using a select-extracting executable. Follow the steps below to install the software.

1. Double click the executable to initiate the installation process. Continue to the License Agreement.
2. Accept the terms of the click-wrap license agreement.
3. Accept the default installation folder, or change the folder to the desired location of the Dolby IP packages (e.g. c:\ti\processor_audio_sdk_1_00_00_01\release\dhip_package). Complete the installation.
4. Upon completion of the installation, unzip all Dolby IP .zip packages (bmda, car, oar, ddp, intrinsics, matted, and oar) to c:\ti\processor_audio_sdk_1_00_00_01\dolby_ip\dh-ip. After unzipping the Dolby IP .zip packages, the dh-ip folder should contain the sub-folders described in the table below.

Folder	**Contents**
asp\	ASP IP (bmda, car, oar) header files
build\	ARM and DSP Dolby IP libraries
dec\	Decoder IP (ddp, mat-thd) header files
DOC\	User documentation

3.6 CCS Update

Update the CCS installation as below:

1. Launch CCS and create a new workspace (e.g. c:\ti\processor_audio_sdk_1_00_00_01\pasdk\workspace_v6_1).
2. Allow installation of newly discovered products located in c:\ti.
   - Ignore any messages about unsigned content
   - Restart CCS when prompted
3. Update Keystone2 device support
   - Select Help→Installation Details from the CCS menu
   - Select Keystone2 device support and click Uninstall.

- Restart CCS when prompted.
- Select Help→Install New Software.
- Select "--All Available Sites--" from the "Work with:" drop-down menu.
- Uncheck "Show only the latest version of the available software".
- Click the right arrow next to Keystone2 Device Support to display the Keystone2 device support version options.
- Select the second version 1.1.5 Keystone2 device support option. Continue to and accept the license agreement.
- Restart CCS when prompted.

4. Update Emulators
    - Select Help→Check for updates from the CCS menu
    - Update Spectrum Digital Emulators to 5.2.0.14 or greater
    - Update TI Emulators to 6.0.407.3 or greater
    - Restart CCS when prompted
5. Update RTSC Product discovery path to access to IPC engineering release 3.43 and Enhanced PDK K2G 1.0.1
    - Select Window→Preferences
    - Select Code Composer Studio→RTSC→Products
    - Click Add next to Product Discovery Path, add folder C:\ti\processor_audio_sdk_1_00_00_01\psdk_cust

3.7 PASDK OS Dev Build and Alpha Communication Tools

Download and install the additional tools below. Sed and Python are required to build the PASDK OS DEV software contained in the ti\processor_audio_sdk_1_00_00_01\pasdk\paf folder. Python is also required for alpha command control I/O over using the PyAlpha tool. The PyAlpha tool is further described below in Section 6.4.

1. Obtain Sed 4.2.1 from http://gnuwin32.sourceforge.net/packages/sed.htm, executable installer, sed-4.2.1-setup.exe.
   Install to c:\PA_Tools\GnuWin32 (or change installation location in c:\ti\processor_audio_sdk_1_00_00_01\pasdk\setup_env.bat).
2. Obtain Python 2.7.6 from https://www.python.org/download/releases/2.7.6 executable installer, python-2.7.6.amd64.msi.
   Install to c:\PA_Tools\Python27 (or change installation location in c:\ti\processor_audio_sdk_1_00_00_01\pasdk\setup_env.bat).

4 Emulator Firmware Update

The K2G EVM on-board emulator firmware must be updated for correct emulator operation. Update the firmware as described below.

1. Connect the EVM to the host PC using the provided USB Mini B to A plug type cable. Connect the Mini B plug on the cable to XDS_USB connector J1 on the EVM.
2. Connect the EVM to power using the supplied 12V power supply.

3. Apply power to the EVM by pushing the SW1 switch to the “ON” position.
4. Open a DOS prompt on the PC. At the DOS prompt, issue these commands:
   ```
   > cd c:\ti\ccsv6\ccs_base\common\uscif\xds2xx
   > update_xds2xx.bat xds200
   ```
5. The batch file output should appear as shown below.
   ```
   Updating CPLD ...
   .
   Updating Firmware ...
   .
   Rebooting ...
   .
   Reading Configuration ...
   .
   Check swRev is 1.0.0.8 or higher
   .
   boardRev=4
   ipAddress=0.0.0.0
   ipConfig=dhcp
   ipGateway=0.0.0.0
   ipNetmask=0.0.0.0
   productClass=XDS2XX
   productName=XDS200
   serialNum=00:0E:99:03:90:0F
   swRev=1.0.0.8
   hostCPU=AM1802
   emuCtrlType=Bit bang
   extMemType=SDRAM
   portUSB=true
   portENET=false
   portWIFI=false
   portRS232=false
   EnableUSBSerial=false
   CurrentMeasure=false
   Press any key to continue . . .
   ```

5 Software Build

5.1 PASDK OS Dev Libraries

The ARM and DSP applications depend on several libraries located in the ti\processor_audio_sdk_1_00_00_01\pasdk\paf folder. Detailed instructions for building any of the libraries in this folder are contained in C:\ti\processor_audio_sdk_1_00_00_01\pasdk\paf\ReadMe.txt. However, batch files for building only the necessary libraries for the ARM/DSP applications are provided in the C:\ti\processor_audio_sdk_1_00_00_01\scripts folder. To build the required libraries:

1. Open a DOS prompt.
2. From the DOS prompt, execute:
   ```
   > cd C:\ti\processor_audio_sdk_1_00_00_01\scripts
   ```

```
> setup_env.bat
> build_paf_libs.bat
```

5.2 UART, SPI, and I2C Low Level Drivers (LLDs)

The enhanced PDK K2G 1.0.1 includes UART, SPI, and I2C serial port LLDs for control I/O. Each LLD supports CPU interrupt-callback mode and the UART LLD also supports EDMA dma-callback mode. Control I/O is implemented in the DSP application using the UART LLD in dma-callback mode. To build the serial port LLD libraries:

1. Open a DOS prompt.
2. From the DOS prompt, execute:
```
> cd C:\ti\processor_audio_sdk_1_00_00_01\scripts
> build_pdk_libs.bat
```

5.3 K2G Platform Library

Configuration and control of the K2G EVM and Audio Daughter Card require the K2G platform library supplied with the enhanced PDK K2G 1.0.1. To build the K2G platform library:

1. If necessary, re-launch CCS using the workspace created above in Section 3.6.
2. Import the “platform_lib_evmk2g” CCS project into the CCS workspace. The project is located in C:\ti\processor_audio_sdk_1_00_00_01\psdk_cust\pdk_k2g_1_0_1_0_eng\packages\ti\platform\evmk2g\platform_lib.
3. Rebuild the project using the Debug build profile.
    - Right-click “platform_lib_evmk2g” project in CCS Project Explorer, select Build Configurations→Set Active→Project→Debug.
    - Right-click “platform_lib_evmk2g” project in CCS Project Explorer, select Rebuild Project.

5.4 DSP Application

To build the DSP application:

2. Import the “test_dsp” CCS project into the CCS workspace. The project is located in C:\ti\processor_audio_sdk_1_00_00_01\pasdk\test_dsp.
3. Ensure the correct component software versions are selected by RTSC.
    - Right-click “test_dsp” project in CCS Project Explorer, select Properties
    - On left-hand side of window, Select General. Select RTSC tab on the right-hand side of General window.
    - Confirm the following are installed and selected: (1) EDMA3 LLD 2.12.1; (2) SYS/BIOS 6.45.1.29; (3) System Analyzer (UIA Target) 2.0.3.43; (4) XDAIS 7.24.0.04; and (5) k2g PDK 1.0.1
    - For IPC, select version 3.42.0.02 and then (re-)select version 3.43.0.00_eng.
    - Click OK.
4. Rebuild the project using the Debug_pkgs build profile.
    - Right-click “test_dsp” project in CCS Project Explorer, select Build Configurations→Set Active→Project→Debug_pkgs.

- Right-click "test_dsp" project in CCS Project Explorer, select Rebuild Project.

5.5 ARM Application

To build the ARM application:

1. Import the "test_arm" CCS project into the CCS workspace. The project is located in C:\ti\processor_audio_sdk_1_00_00_01\pasdk\test_arm.
2. Ensure the correct component software versions are selected by RTSC.
   - Right-click "test_arm" project in CCS Project Explorer, select Properties
   - On left-hand side of window, Select "General". Select "RTSC" tab on the right-hand side of "General" window.
   - Confirm the following are installed and selected: (1) SYS/BIOS 6.45.1.29; (2) System Analyzer (UIA Target) 2.0.3.43; and (3) XDAIS 7.24.0.04.
   - For IPC, select version 3.42.0.02 and then (re-)select version 3.43.0.00_eng.
   - Click "OK".
3. Rebuild the project using the Debug_pkgs build profile.
   - Right-click "test_dsp" project in CCS Project Explorer, select Build Configurations→Set Active→Project→Debug_pkgs.
   - Right-click "test_dsp" project in CCS Project Explorer, select Rebuild Project.

6 Application Execution and Test

6.1 I/O Configuration

AR2 supports boot- and run-time I/O configuration. Boot-time I/O configuration is determined by the at-boot I/O shortcuts in the CUS_ATBOOT_S definition in c:\ti\processor_audio_sdk_1_00_00_01\pasdk\test_dsp\application\itopo\evmk2g\atboot.c, while run-time I/O configuration is performed by invoking I/O shortcuts via alpha commands. The available Input and Output shortcuts are defined in C:\ti\processor_audio_sdk_1_00_00_01\pasdk\test_dsp\application\itopo\evmk2g\io.c and described in the tables below.

Input Configuration	**Input Shortcut**
ADC 8-ch (w/ DAC 8-ch input)	execPAIInAnalog
S/PDIF	execPAIInDigital
Stereo HDMI (PCM or DDP)	execPAIHDMIInStereo
HDMI (MAT-THD/PCM)	execPAIInHDMI

Output Configuration	**Output Shortcut**

DAC 8-ch (w/ ADC 8-ch input)	execPAIOutAnalogSlave
DAC (8-ch)	execPAIOutAnalog
DAC (12-ch)	execPAIOutAnalog12Ch
DAC (16-ch)	execPAIOutAnalog16Ch

6.2 Connect EVM

1. Turn off power to the EVM using SW1 on the EVM. Disconnect the 12V power supply.
2. Assemble the Audio Daughter Card (assembly instructions are packaged with Audio Daughter Card).
3. Attach the Audio Daughter Card to the K2G EVM. The Audio Daughter Card connects to the EVM through the JP1 “Audio Expansion” and J12 “Serial Expansion” connectors on EVM.
4. Connect the PC and K2G EVM using DB9 serial RS-232 cable delivered with EVM.
    - A USB to UART FTDI cable can be used on the PC side of the connection.
    - Attach the DB9 connector to the P1 “UART0” connector on the EVM.
    - Push the SW12-UART0_DB9_EN switch to the “ON” position.
5. Connect the EVM emulation port to the PC as described in Section 4.
6. Connect the MDS HSR41P to the MDS IFB.
    - Connect HSR41P J4 “Main” header to the IFB J5 “Main” header using the short 26-wire ribbon cable. Ensure the red wire on the ribbon cable is connected between HSR41P J4 Pin1 and IFB J5 Pin1.
    - Connect HSR41P J7 “Sec” header to the IFB J2 “Sec” header using the short 34-wire ribbon cable. Ensure the red wire on the ribbon cable is connected between HSR41P J7 Pin1 and IFB J2 Pin1.
7. Connect the MDS IFB to the Audio Daughter Card. Connect IFB J3 “McASP0” header to Audio Daughter Card “I2S Header” using the supplied long 34-wire ribbon cable. Ensure the red wire on the ribbon cable is connected between IFB J3 Pin1 and Audio Daughter Card “I2S Header” Pin1 (note the “I2S Header” is only 32 pins, so the two wires on the ribbon cable opposite Pin1 are unused).
8. Connect HDMI cable from audio source (e.g. Blu-ray player) to HSR41P J6 “IN”. Connect HDMI cable to audio sink (e.g. HDMI enabled monitor) from HSR41P J8 “HDMI OUT”.
9. Connect 5V AC to USB power supply to IFB power connector J1.
10. Re-connect the 12V power supply to the EVM. Turn on power to the EVM using SW1.

6.2.1 ADC 8-Ch Input and DAC 8-Ch Output

Connect ADC inputs to the ADC_IN0-3 input jacks on the Audio Daughter Card. Processed PCM samples will be output to the corresponding DAC output jacks DAC_OUT0-3 on the Daughter Card.

6.2.2 S/PDIF Input and DAC 8-Ch Output

Connect S/PDIF input to the MOD1 Digital Input (Optical) connector on the Audio Daughter Card. Processed PCM samples will be output to the DAC output jacks DAC_OUT0-3 on the Daughter Card.

6.2.3 HDMI Input and DAC Output

Connect HDMI input source to the HSR41P repeater module as described above. Processed PCM samples will be output to DAC output jacks DAC_OUT0-6 on the Daughter Card. The number of active DAC outputs depends on: (1) the selected Output Shortcut (execPAIOutAnalog, execPAIOutAnalog12Ch, or execPAIOutAnalog16Ch); (2) the configured ENC channel map from; and (3) the active renderer channel configuration request. For example, issue the alpha commands listed below to enable 12-channel DAC for channel-based audio using CAR with speaker layout L/R, C, Sub, Ls/Rs, Lrs/Rrs, Ltf/Rtf, Ltr/Rtr:

```
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 execPAIOutAnalog12Ch
alpha execPAIOutAnalog12Ch

> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1
writeENCChannelMapFrom16(PAF_LEFT,PAF_RGHT,PAF_LSUR,PAF_RSUR,PAF_CNTR,PAF_SUBW,PAF_LBA
K,PAF_RBAK,PAF_LTFT,PAF_RTFT,PAF_LTRR,PAF_RTRR,-3,-3,-3,-3)

> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1
writeSYSChannelConfigurationRequestSurround4LtfRtfLtrRtr_1
alpha /* none */

> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 readCARChannelCfgOverride
alpha writeCARChannelCfgOverrideUnknown
```

6.3 Load and Execute Code

Follow the steps below from within CCS to load and execute the DSP and ARM application code:

1. Click View→Target Configurations.
2. In the Target Configurations window, open the Projects→test_dsp→targetConfigs folder.
3. Right-click on K2GEVM.ccxml contained in the targetConfigs folder.
4. Select Launch Selected Configuration.
5. In the Debug window, right-click on the C66x and select Connect Target. The output from the GEL code invoked on the target connection can be observed in the Console output window. The final line of this output should read "`C66xx: GEL Output: DDR3A initialization complete`".
6. In the Debug window, right-click on the CortexA15 and select Connect Target. The output from the GEL code invoked on the target connection can be observed in the Console output window. The final line of this output should read "`CortexA15: GEL Output: A15 non secure mode entered`".
7. In the Debug Window, click on the C66x and then open Run→Load→Load Program. In the Load Program window, click on Browse project, open the "test_dsp" folder, and click on "test_dsp.out" and OK to select the program to be loaded. Click on OK in the Load Program window to load the program to the C66x.
8. In the Debug Window, click on the CortexA15, and then open Run→Load→Load Program. In the Load Program window, click on Browse project, open the "test_arm" folder, and click on "test_arm.out" and OK to select the program to be loaded. Click on OK in the Load Program window to load the program to the A15.
9. In the Debug window, click on the C66x and then Open Run→Resume to execute the DSP code.

10. In the Debug window, click on the CortexA15 and then Open Run→Resume to execute the ARM code.
11. The CIO Console output window should display memory usage statistics. The final line of output should display the memory usage summary for the MSMC SHM heap.
12. In the Debug Window, click on the CortexA15, and then open Run→Disconnect Target. This step is required for clean audio output (known issue for this release).

6.4 PyAlpha Python Scripts for Alpha Communication

AR2 supports alpha command control over UART using the PyAlpha tool located in c:\ti\processor_audio_sdk_1_00_00_01\tools. PyAlpha is invoked on the command-line in a DOS shell. PyAlpha documentation can be obtained by executing PyAlpha with a "--help" argument as below (note: the Python executable is contained in the DOS shell path for this example).

```
> cd c:\ti\processor_audio_sdk_1_00_00_01\tools
> python pyalpha --help
usage: PyAlpha [-?] [-h FILE] [-I DIR] [-p PORT] [-b BAUD] [-t DIR] [-bin DIR]
               [-r] [--fast] [--repeat N T N T]
               [ALPHAS [ALPHAS ...]]

Alpha Command Communicator

positional arguments:
  ALPHAS                Alpha commands, separated by spaces

optional arguments:
  -?, --help            show this help message and exit
  -h FILE, --header FILE
                        Add FILE_a.h to symbol definition list and FILE_a.hdM
                        to symbol decomposition list
  -I DIR, --include DIR
                        Add DIR to the include path. [default: ./alpha]
  -p PORT, --port PORT  Communication port [default: COM1]
  -b BAUD, --baud BAUD  Baud rate [default: 19200]
  -t DIR, --temp DIR    Temporary file save location [default: .]
  -bin DIR              Binary file location [default: ./bin]
  -r, --retain          Retain temporary files
  --fast                Skip post-processing
  --repeat N T N T      Repeat an alpha command N times with a T second break
                        between
```

6.5 Alpha Commands for AE0 Configuration/Status

AR2 includes the Audio Example 0 (AE0) ASP. This ASP transforms stereo (Left/Right) PCM into 3stereo (Left/Right/Center) PCM. AE0 can be included in the signal chain by uncommenting Line 63 (`63: #define _AE0_)` in C:\ti\processor_audio_sdk_1_00_00_01\pasdk\test_dsp\framework\itopo\patch.c and rebuilding the DSP application. AE0 documentation is provided in C:\ti\processor_audio_sdk_1_00_00_01\docs\DA10x_UG_AspExample.pdf, and the alpha commands contained in C:\ti\processor_audio_sdk_1_00_00_01\tools\alpha\ae_a.h can be used to configure AE0 and consult AE0 status.

In the example alpha command sequence below, AE0 is configured to generate 3stereo using a gain of 0.5 (16384 in S16Q15 format) using COM port 1. After disabling AE0, the Master Volume Control is set to -20 (= 40*0.5 dB). Note the '>' indicates the commands are issued from a DOS shell.

```
> cd c:\ti\processor_audio_sdk_1_00_00_01\tools
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 readAEMode
alpha writeAEModeDisable
```

```
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 writeAEModeEnable
alpha /* none */
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 readAEMode
alpha writeAEModeEnable
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 readAEScaleQ15
alpha wroteAEScaleQ15,0x0000
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 writeAEScaleQ15(16384)
alpha
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 readAEScaleQ15
alpha wroteAEScaleQ15,0x4000
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1
readSYSChannelConfigurationRequest
alpha 0xce20,0x2008,0x010c,0x0000,0x0000,0x0000
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1
writeSYSChannelConfigurationRequestStereo
alpha /* none */
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1
readSYSChannelConfigurationRequest
alpha 0xce20,0x2008,0x0003,0x0000,0x0000,0x0000
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1
writeSYSChannelConfigurationRequestSurround4_1
alpha /* none */
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1
readSYSChannelConfigurationRequest
alpha 0xce20,0x2008,0x010c,0x0000,0x0000,0x0000
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 writeAEModeDisable
alpha /* none */
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 readAEMode
alpha writeAEModeDisable
> python.exe pyalpha -I alpha -h pa_i13_evmk2g_io_a -p COM1 writeVOLControlMasterN(-
40)
alpha /* none */
```

7 Known Issues

- MAT-THD/PCM decoding unstable.
- Decoder status not properly shared between ARM/DSP (cache coherency issue).
- Alpha commands over UART sometimes TIMEOUT.
- SPI and I2C LLDs only support CPU interrupt-callback mode.