

Using the uPP EVM example applications from the BIOS PSP

Contents

Introduction

Software Prerequisites

Hardware Setup

Rebuilding the Example Applications

Running the Example Applications

See Also

Introduction

The BIOS PSP for OMAP-L138 and C6748 includes a driver for the uPP peripheral and multiple example applications that demonstrate how it is used. The first example simply operates uPP in a digital loopback mode. The other two examples, called evmA and evmB, are meant to operate together on two separate EVMs to perform real data transfers from one device to the other.

This article describes the setup and procedure necessary to run these EVM example applications. These examples are also described in the user guide distributed as part of the BIOS PSP installation. This article is meant to be a quick reference guide.

Software Prerequisites

- Code Composer Studio (<http://www.ti.com/ccs>) (v3.3, 4, or 5)
- BIOS PSP v1.30.01 or later (http://software-dl.ti.com/dsps/dsps_public_sw/psp/BIOSPSP/index.html) (**not** v2.x)
- DSP/BIOS v5.41 (http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/bios/dspbios/index.html)

Hardware Setup

Since these example applications setup two separate devices to communicate one another via uPP, you will need two EVMs. Either the OMAP-L138 or C6748 SOM can be used. When using the OMAP-L138 SOM, place the device in emulation boot mode by setting S7[5:8] to 1001.

You will also need two PCs to run a simultaneous CCS debug session on each device. The software prerequisites, above, must be met on both PCs.

Finally, you will need to connect the uPP pins from each device together. This is made more challenging by the fact that the EVM does not come with a ready-made connector to achieve this, so a breakout board or cable is necessary.

All of the uPP signals are accessible via connector J29 on the EVM. Note that this connector is also used by the optional UI board, so you will need to disconnect that board from the EVM in order to access the uPP pins. The EVM schematic indicates which uPP signal is connected to each pin of the connector, but some of the signal names are incorrectly labeled. The following table lists all of the uPP signals as named in the schematic and also gives the correct signal name and the pin on J29 where it is routed.

--	--	--

Schematic Label	uPP Signal Name	Pin on J29
UPP_CH1_START	Start A	65
UPP_CH1_ENABLE	Enable A	69
UPP_CH1_CLK	Clock A	67
UPP_CH1_WAIT	Wait A	71
UPP_CH0_START	Start B	76
UPP_CH0_ENABLE	Enable B	64
UPP_CH0_CLK	Clock B	78
UPP_CH0_WAIT	Wait B	77
UPP_XD[15:0]	XData[15:0]	13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28
UPP_CH1_D[15:0]	Data[15:0]	80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95

To properly prepare for the uPP EVM examples, every uPP pin must be connected to the same pin on the other EVM. There are four basic groups of uPP pins:

1. Channel A control pins
2. Channel B control pins
3. Data pins
4. XData pins

All of these pins should be connected together across both EVMs.

Rebuilding the Example Applications

The pre-built versions of the uPP EVM example applications are set to run at the maximum uPP clock speed of 75 MHz. This speed is probably unrealistic for use across two breakout boards (and/or a long connecting cable), so the applications must be rebuilt to operate at a lower uPP I/O clock speed. This is a straightforward process.

First, locate the uPP EVM example applications inside the BIOS PSP installation. The paths will generally look something like this:

```
C:\Program Files\Texas Instruments\pspdriers_01_30_01\packages\ti\pspiom\examples\evm6748\uppEvm\evmA
C:\Program Files\Texas Instruments\pspdriers_01_30_01\packages\ti\pspiom\examples\evm6748\uppEvm\evmB
```

In CCSv3.3, opening each project is as simple as opening the *.pjt file located in its build/ccs3 subfolder. In CCSv4 or 5, you will need to import the Eclipse project located in the build/ccs4 subfolder. You can also use the import legacy project wizard to import the CCSv3.3 *.pjt file into your workspace.

Once the project is open, you may need to configure the build settings so that CCS knows where to find the various PSP libraries and header files it needs to build the project. This should generally only be necessary in CCSv3.3.

Before building the projects, it's necessary to make one simple change to the source code. Open the file named src/uppSample_io.c (same name in both projects) and locate the uPP configuration structs near the top of the file. You should see two such structs in each source file: one for channel A and one for channel B. The seventh element in each struct sets the uPP I/O clock speed, and is initially set to 75 MHz (75000000). Change this to 10 MHz (10000000) in both structs and rebuild the application.

Running the Example Applications

Important note: Do not skip the above sections on hardware setup and rebuilding the example applications! This procedure will not work if you haven't properly setup the EVM hardware and re-configured the example applications.

To run the example applications, observe the following procedure:

1. Connect one PC to each EVM and launch CCS on both
2. Connect the uPP pins together as described in the hardware setup section, above
3. Begin a debug session for each example application on either EVM (i.e. evmA on one EVM, evmB on the other)
 - Be sure to use application binaries rebuilt according to the instructions in the previous section
4. Run the evmB application first
5. Run the evmA application second
6. Wait for the applications to finish
 - Open the DSP/BIOS message log (named trace) to view application status; the applications use LOG_printf instead of printf

If both applications complete successfully, you should see the following message in each application's message log:

```
Sample Application completed...
```

If either application fails, you should instead see error messages describing what went wrong.

See Also

- [Introduction to uPP](#)
- [Using the uPP DSP/BIOS Driver](#) (different driver; not from BIOS PSP)

<pre> {{ 1. switchcategory:MultiCore= ▪ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum ▪ For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum Please post only comments related to the article Using the uPP EVM example applications from the BIOS PSP here. </pre>	<pre> Keystone= ▪ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum ▪ For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum Please post only comments related to the article Using the uPP EVM example applications from the BIOS PSP here. </pre>	<pre> C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article Using the uPP EVM example applications from the BIOS PSP here. </pre>	<pre> DaVinci=For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article Using the uPP EVM example applications from the BIOS PSP here. </pre>	<pre> MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article Using the uPP EVM example applications from the BIOS PSP here. </pre>	<pre> OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article Using the uPP EVM example applications from the BIOS PSP here. </pre>	<pre> OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article Using the uPP EVM example applications from the BIOS PSP here. </pre>	<pre> MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article Using the uPP EVM example applications from the BIOS PSP here. </pre>	<pre> For technical support please post your questions at http://e2e.ti.com. Please post only comments about the article Using the uPP EVM example applications from the BIOS PSP here. </pre>
--	--	--	---	---	--	---	--	---

Links



[Amplifiers & Linear](#)

[Audio](#)

[Broadband RF/IF & Digital Radio](#)

[Clocks & Timers](#)

[Data Converters](#)

[DLP & MEMS](#)

[High-Reliability](#)

[Interface](#)

[Logic](#)

[Power Management](#)

[Processors](#)

- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)
- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

[Switches & Multiplexers](#)

[Temperature Sensors & Control ICs](#)

[Wireless Connectivity](#)

Retrieved from "https://processors.wiki.ti.com/index.php?title=Using_the_uPP_EVM_example_applications_from_the_BIOS_PSP&oldid=67932"

This page was last edited on 6 July 2011, at 09:37.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.