Interfacing Blackfin® Processors to the Intellon INT5200 HomePlug PHY

Contributed by K.C.Kreitzer *Rev 2 – December 16, 2008*

Introduction

This EE-Note describes how to interface the ADSP-BF537 Blackfin® processor to the Intellon INT5200 HomePlug PHY. It provides the details for building an interface from an ADSP-BF537 EZ-KIT Lite® board to an Intellon EDK5200 evaluation board. It also provides an example VisualDSP++® project that implements a networked audio server/client pair. This networked audio project may be used either with HomePlug or standard 10/100 Ethernet as the physical medium.

What is HomePlug?

HomePlug is a physical layer standard for networking over home power lines. More about HomePlug may be found at the HomePlug Alliance Web site:

http://www.homeplug.org

Because HomePlug is strictly a physical layer standard, it is quite simple to interface a HomePlug PHY to a Media Access Controller (MAC) intended for an Ethernet physical layer. Many HomePlug PHYs have Media Independent Interfaces (MII) just for this purpose. That makes them ideal candidates for use with Blackfin processors that have MAC peripherals such as the ADSP-BF537 processor.

What is the INT5200?

Intellon's INT5200 is a HomePlug 1.0 PHY chip that supports line rates up to 14 Mbps. Product information on the INT5200 PHY may be found on the Intellon Web site at:

http://www.intellon.com/products/homeplug/int5200.php

The EDK5200 is Intellon's development platform for the INT5200 PHY. Documentation on the EDK5200 board may be found at:

http://www.intellon.com/support/documentation.php

Information on where to purchase EDK5200 boards or INT5200 PHYs may be found at:

http://www.intellon.com/support/distributors.php

ADSP-BF537 EZ-KIT Lite Board to Intellon EDK5200 Eval Board Interface

Wiring Harness

Connecting the ADSP-BF537 EZ-KIT Lite board to the Intellon EDK5200 evaluation board may be accomplished by building a wiring harness or by using an adapter called the EZ-HomePlug board. If a wiring harness is desired, Table 1 contains the connections required from the J3 connector on the ADSP-BF537 EZ-KIT Lite board to the EDK5200 P2 connector.

Signal Name	ADSP-BF537 EZ-KIT J3 Pin	EDK5200 Eval Board Pin (Note)
TXD0	11	16
TXD1	12	15
TXD2	13	14
TXD3	14	13
TXEN	15	18
TXCLK	16	19
TXER	N/C	20 (pull down to GND)
COL	32	17
RXD0	33	27
RXD1	34	28
RXD2	35	29
RXD3	36	30
RXDV	37	26
RXCLK	38	25
RXER	39	23
CRS	40	24
MDCLK	41	8
MDIO	42	9
MDADR0	N/C	11 (pull up to VCC or down to GND)
MDADR1	N/C	10 (pull up to VCC or down to GND
GPSI_SEL	N/C	21 (pull up to VCC)
SPIS	N/C	32 (pull up to VCC)
GND	3, 4, 24, 25, 43, 44, 87, 88	4, 12, 20, 22, 34
VCC	N/C	6, 7 (use for pull-ups)

Table 1. ADSP-BF537 EZ-KIT Lite to EDK5200 evaluation board connections

EZ-HomePlug

If a more robust solution is desired, a board called the EZ-HomePlug will interface the ADSP-BF537 EZ-KIT Lite board to the EDK5200 development board (see block diagram in Figure 1). The printed circuit board (Figure 2) may be ordered from Express PCB at `http://www.expresspcb.com` using the design files in the `.ZIP` file that accompanies this EE-Note. The bill of materials in Table 2 includes associated part numbers, which may be ordered from DigiKey at `http://www.digikey.com`. This board was designed with through-hole parts for ease of assembly. The complete assembly is shown in Figure 3.

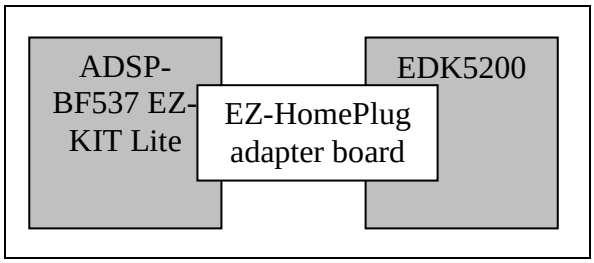

Figure 1. EZ-HomePlug block diagram

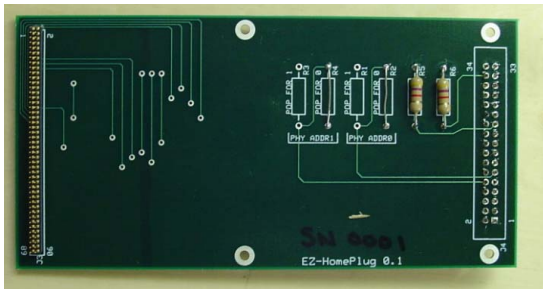

Figure 2. EZ-HomePlug photograph

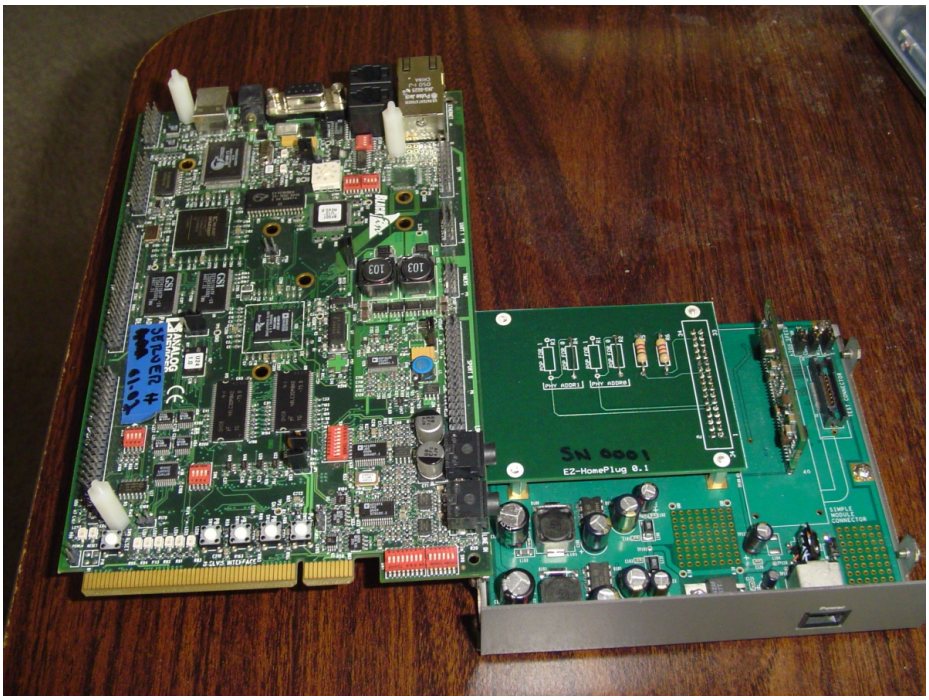

Figure 3. Assembled combination:ADSP-BF537 EZ-KIT Lite, EZ-HomePlug, and EDK5200

Description	Reference Designator	DigiKey P/N (Note)
90-pin male 0.05” pitch connector	J3	ED83100-ND (must be trimmed to 90 pins)
34-pin female 0.10” pitch connector	J4	609-2281-ND
4.7 kΩ pull-up resistors / 0 Ω pull-down resistors	R1 – R6	Any standard through-hole resistor (or jumper wire for 0 Ω)

Table 2. EZ-HomePlug bill of materials

The J3 connector must be trimmed before installation. The J3 connector is not easily found in 90-pin configurations. The ED83100-ND shown in the bill of materials is a 100-pin connector which can easily be trimmed to 90 pins.

Preparing the ADSP-BF537 EZ-KIT Lite board

Once the ADSP-BF537 EZ-KIT Lite board is connected to the EDK5200 board, either by wiring harness or by using the EZ-HomePlug board, two distinct PHY chips (SMSC LAN3C185 Ethernet PHY on the EZ-KIT Lite board and Intellon INT5200 HomePlug PHY located on the EDK5200 board) will share the ADSP-BF537 processor's MII bus. Normally, two PHYs are allowed to share a common MII bus. This is accomplished by the use of two address bits on the `MDIO` lines to identify which PHY the command and status data are associated with. This means that each PHY must have a distinct PHY address. The EZ-KIT PHY is address 01b. In order to share the MII data interface, an `ISOLATE` command bit in the MII control registers of the PHY causes the data lines to three-state. The LAN83C185 Ethernet PHY, however, does not support the use of the `ISOLATE` command bit. Unfortunately, this means that the LAN83C185 PHY must be physically removed from the EZ-KIT Lite board.

The SMSC LAN83C185 Ethernet PHY (U14) must be physically removed from the ADSP-BF537 EZ-KIT Lite board in order for the INT5200 HomePlug PHY to work properly. Note that removing U14 will adversely affect the board's warranty. Please consult the ADSP-BF537 EZ-KIT Lite Evaluation System Manual for details.

Since the LAN83C185 PHY is a quad-flat pack, it is easily removed with a small-bladed sharp knife. Use care that none of the pads are shorted after removal of the LAN83C185 PHY. Figure 4 shows the physical location on the ADSP-BF537 EZ-KIT Lite board of the component that must be removed.

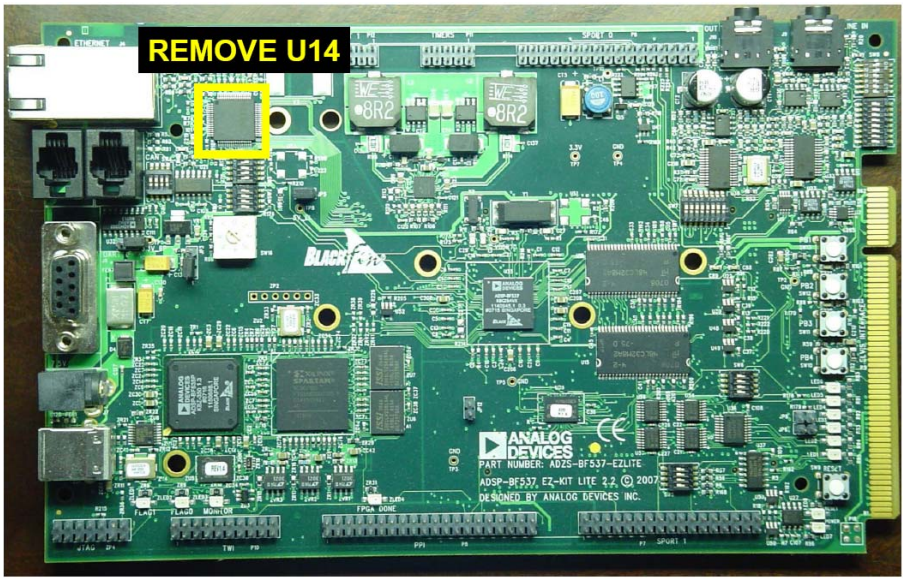

Figure 4. Remove U14 from the ADSP-BF537 EZ-KIT Lite board

Customizing the Interface for Your Own Design

Table 3 shows the signal connections directly from the ADSP-BF536/BF537 processor to the INT5200 PHY that you would use when doing your own hardware design.

- The TXER signal is not provided by the ADSP-BF537 processor and it is therefore tied to ground on the INT5200 PHY.
- The PHYINT signal on the ADSP-BF537 processor is unused. This pin may either be assigned as a GPIO or, if assigned as a PHYINT signal, should be pulled up to VCC.
- The MDADR0/1 lines should be set according to the desired PHY address.
- The GPSI_SEL, SPIS, and MODE0 lines are mode pins for the INT5200 PHY and should be set according to Table 3. This combination will place the INT5200 PHY in MII PHY mode.

Signal Name	ADSP-BF536/BF537 Signal Name (Note)	EDK5200 Signal Name (Note)
TXD0	PH0/ETxD0	B8/MII_TX0
TXD1	PH1/ETxD1	B9/MII_TX1
TXD2	PH2/ETxD2	A10/MII_TX2
TXD3	PH3/ETxD3	B11/MII_TX3
TXEN	PH4/ETxEN	C7/MII_TXEN
TXCLK	PH5/MII TxCLK	C5/MII_TXCLK
TXER	N/C	B6/MII_TX_ER (pull down to GND)
PHYINT	PH6/MII PHYINT (N/C, or pull up to VCC)	N/C
COL	PH7/COL	C6/MII_COL
RXD0	PH8/ERxD0	A2/MII_RX0
RXD1	PH9/ERxD1	B2/MII_RX1
RXD2	PH10/ERxD2	C2/MII_RX2
RXD3	PH11/ERxD3	C3/MII_RX3
RXDV	PH12/ERxDV	A4/MII_RXDV
RXCLK	PH13/ERxCLK	B3/MII_RXCLK
RXER	PH14/ERxER	C4/MII_RX_ER
CRS	PH15/MII CRS	B5/MII_CRS
MDCLK	PJ0/MDC	C9/MII_MDCLK
MDIO	PJ1/MDIO	C10/MII_MDIO
MDADR0	N/C	C8/MII_ADRSEL0 (pull up to VCC or down to GND)
MDADR1	N/C	A11/MII_ADRSEL1 (pull up to VCC or down to GND)
GPSI_SEL	N/C	E2/MII_GPSI_N (pull up to VCC)
SPIS	N/C	A3/MDI_SPIS_N (pull up to VCC)
MODE0	N/C	B1/MODE0 (pull down to GND)

Table 3. ADSP-BF537 processor to INT5200 PHY connections

Modifying the PHY Driver

VisualDSP++ development tools come with a driver for the SMSC LAN83C185 Ethernet PHY. This driver is sufficient to accommodate the Intellon INT5200 HomePlug PHY without modification.

There are, however, a couple of differences that must be accommodated when using the driver. The first is to set the PHY address for the INT5200. This is done as shown in the code box below.

```
result = adi_dev_Control (
    lanHandle,
    ADI_ETHER_CMD_BF537_SET_PHY_ADDR,
    (void *)0);
```

The second is that, due to an incompatibility in the MDIO interface of the INT5200, the MDC register reads will return shifted down by one bit. This may be accommodated as illustrated below.

```
// wait for the link to be up
#ifdef PHY_INT5200
if ( (phyregs[1] & 0x2) == 0)
#else
if ( (phyregs[1] & 0x4) == 0)
#endif
```

All of these may optionally be accommodated by writing a custom driver for the INT5200. If you wish to customize the driver specifically for the INT5200, you may do so according to the guidance supplied in the Analog Devices application note EE-315.

Networked Audio Project

A VisualDSP++ networked audio project is included in the `.ZIP` file associated with this EE-Note. It consists of two VisualDSP++ projects – one for the server and one for the client. The server takes line level audio input and sends digitized audio packets over the network. The client receives the digitized audio packets over the network and outputs line level audio.

This project may be used with either Ethernet or HomePlug physical networks. This is selected by a compile-time flag in `system.h`. Define `PHY_INT5200` for Intellon INT5200 HomePlug PHY, and undefined it for SMSC Ethernet PHY.

```
#define PHY_INT5200  // for HomePlug
--or—
#undef  PHY_INT5200  // for Ethernet
```

Details of tool versions, board versions, switch and jumper settings, and LED definitions may be found in the `readme.txt` file found with each project.

Server

Thread Description

There are three threads in this project.

The `Boot` thread initializes lwIP, System Services, and the Device Manager. It then spawns the `Audio` thread.

The Audio thread sets up a UDP socket for network transfers. It then initializes and starts the audio ADC (SPORT) driver. Finally, it spawns the AudioSend thread and keeps track of "keep alive" messages coming back from the client.

The AudioSend thread is responsible for sending the UDP audio packets across the network. It is distinct from the Audio thread because send() commands cannot be sent at the interrupt level at the time of the Audio callback function, but only at the thread level.

Data Flow

An audio source should be connected to the "line in" port of the EZ-KIT Lite board. The audio input stream comes into the ADSP-BF537 SPORT from the on-board audio ADC. This data is moved into memory via DMA by the SPORT driver. The Audio thread callback routine queues the data to the AudioSend thread and gives the buffer back to the SPORT driver. The AudioSend thread then sends the data packet to the UDP socket.

Client

Thread Description

There are four threads in this project.

The Boot thread initializes lwIP, System Services, and the Device Manager. It then spawns the Audio thread.

The Audio thread sets up a UDP socket for network transfers. It then initializes and starts the audio DAC (SPORT) driver. Finally, it spawns the AudioFlow and KeepAlive threads.

Because the SPORT clock rates on the server and client will never match exactly, some sort of flow control is necessary to prevent overflow or underflow of the data buffers. The AudioFlow thread throttles the SPORT output clock based on buffer levels checked during the SPORT driver callback found in audio.c. A periodic semaphore is used to ensure that the updates happen gradually, thus minimizing the associated audio distortion.

There are other ways to do flow control that were not used in this project. Sample insertion/deletion may be used, but the distortion may be much more severe than throttling the output sample clock. One way to minimize that effect is to do energy detection on each data buffer and make the sample insertion/deletion only at a relatively low energy point, ideally a gap or a moment of silence.

The KeepAlive thread sends "keep alive" messages back to the server.

Data Flow

Speakers or headphones should be connected to the "line out" port of the EZ-KIT Lite board. The audio packets arrive over the UDP socket in the Audio thread in a routine called RecvAudio(). This is a place holder for future audio processing on the incoming packet. Currently, it just copies it into a buffer to send to the SPORT driver. The SPORT driver then pulls data out of the buffer independently of the RecvAudio() routine placing it there. In order to maintain flow control in and out of the buffer, a check of the pointers is made in the AudioOutCallback() routine. It can mute the audio if no new packets are arriving or adjust the SPORT output clock values if the output needs to speed up or slow down. The actual adjustment of the SPORT output clock values is done in the AudioFlow thread.

Operational Notes

- The power connections for both the EZ-KIT Lite board and the EDK5200 board must be plugged in. The power supplies are independent.
- The power plugs for the EDK5200 boards should not be plugged into a surge suppressor. The surge suppressor circuitry will most likely interfere with the HomePlug signal. Either use non-surge suppressing power strips or plug directly into wall outlets.
- This has been tested successfully in power outlets in several homes and businesses in the United States. Some electrically noisy appliances such as microwave ovens or hair dryers have been observed to interfere with the HomePlug signal integrity if plugged into the same wall outlet.

References

[1] *ADSP-BF537 Blackfin Processor Hardware Reference*. Revision 3.0, December, 2007. Analog Devices, Inc.

[2] *ADSP-BF537 EZ-KIT Lite Evaluation System Manual*. Revision 2.2, May 2007, Analog Devices, Inc.

[3] *Changing the PHY in the Ethernet Driver for Blackfin Processors (EE-315).* Rev 1. June 2007. Analog Devices, Inc.

Document History

Revision	Description
Rev 2 – December 16, 2008 by K.C.Kreitzer	Added disclaimer for EZ-KIT evaluation board modifications.
Rev 1 – May 14, 2008 by K.C.Kreitzer	Initial release.