BeagleBoard Rev B7
System Reference Manual

Revision 0.2
November 17, 2008

beagleboard.org

THIS DOCUMENT

For any questions, concerns, or issues submit them to gerald@BeagleBoard.org

BEAGLEBOARD DESIGN

These design materials referred to in this document are ***NOT SUPPORTED*** and **DO NOT** constitute a reference design. Only "community" support is allowed via resources at BeagleBoard.org/discuss.

We mean it, these design materials may be totally unsuitable for any purposes.

BeagleBoard.org provides the enclosed product(s) under the following conditions:

This evaluation board/kit is intended for use for **ENGINEERING DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY** and is not considered by BeagleBoard.org to be a finished end-product fit for general consumer use. Persons handling the product(s) must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards. This evaluation board/kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL, and therefore may not meet the technical requirements of these directives or other related directives.

Should this evaluation board/kit not meet the specifications indicated in the User's Guide, the board/kit may be returned within 30 days from the date of delivery for a full refund. THE FOREGOING WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY SELLER TO BUYER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies BeagleBoard.org from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

BeagleBoard.org currently deals with a variety of customers for products, and therefore our arrangement with the user **is not exclusive**. BeagleBoard.org assumes **no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein**.

Please read the User's Guide and, specifically, the Warnings and Restrictions notice in the User's Guide prior to handling the product. This notice contains important safety information about temperatures and voltages. For additional information on BeagleBoard.org environmental and/or safety programs, please contact visit BeagleBoard.org.

No license is granted under any patent right or other intellectual property right of BeagleBoard.org covering or relating to any machine, process, or combination in which such BeagleBoard.org products or services might be or are used.

Mailing Address:

BeagleBoard.org
675 North Glenville #195
Richardson, TX 75081

WARRANTY: *The BeagleBoard is warranted against defects in materials and workmanship for a period of 90 days from purchase. This warranty does not cover any problems occurring as a result of improper use, modifications, exposure to water, excessive voltages, abuse, or accidents. All boards will be returned via standard mail if an issue is found. If no issue is found or express return is needed, the customer will pay all shipping costs.*

Before returning the board, please visit BeagleBoard.org/support

Please refer to sections 12 and 13 of this document for the board checkout procedures and troubleshooting guides.

To return a defective board, please request an RMA at http://beagleboard.org/support/rma .

Table of Contents

beagleboard.org

Figures

beagleboard.org

Tables

1.0 Introduction

This document is the System Reference Manual for the BeagleBoard, a low cost OMAP3530 based board supported through BeagleBoard.org. Previously this document was known as the Hardware Reference Manual, but the name has been changed to more accurately reflect its contents which now include system setup, debugging, and software components. This document provides detailed information on the overall design and usage of the BeagleBoard from the System perspective.

The key sections in this document are:

Section 2.0– Change History
> Provides tracking for the changes made to the System Reference Manual.

Section 3.0– Definitions and References
> This section provides definitions for commonly used terms and acronyms.

Section 4.0– Overview
> This is a high level overview of the BeagleBoard.

Section 5.0– Specification
> Provided here are the features and electrical specifications of the BeagleBoard.

Section 6.0-Product Contents
> Describes what the BeagleBoard package looks like and what is included in the box.

Section 7.0– Hookup
> Covered here is how to connect the various cables to the BeagleBoard.

Section 8.0– System Architecture and Design
> This section provides information on the overall architecture and design of the BeagleBoard. This is a very detailed section that goes into the design of each circuit on the board.

Section 9.0– Connector Pinouts and Cables
> The section describes each connector and cable used in the system. This will allow the user to create cables or purchase cables or to perform debugging as needed.

Section 10.0– BeagleBoard Accessories
> Covered in this section are a few of the accessories that may be used with BeagleBoard. This is not an exhaustive list, but does provide an idea of the types of cables and accessories and how to find them. It also provides a definition of what they need to be. It does not guarantee that these devices will work on all OS implementations.

Section 11.0 – Mechanical
> Information is provided here on the dimensions of the BeagleBoard.

Section 12.0 – Board Verification
> A description is provided on how to setup the board and using the verification process and SW, verify that the board is functional.

Section 13.0 – Troubleshooting

Here is where you can find tips on troubleshooting the setup of the BeagleBoard.

Section 14.0- Known Issues

This section describes the known issues with the current revision of the BeagleBoard.

Section 15.0- BeagleBoard Components

These are the top and bottom side silkscreen of the BeagleBoard showing the location of the components.

Section 16.0- BeagleBoard Schematics

These are the schematics for the BeagleBoard itself .

Section 17.0- Bill Of Material

This section describes where to get the latest Bill of Material for the BeagleBoard.

Section 18.0- BeagleBoard PCB Information

This section describes where to get the PCB file information for the BeagleBoard.

2.0 Change History

2.1 Change History

Table 1 tracks the changes made for each revision of this document.

Table 1. Change History

Rev	Changes	Date	By
B5	1. Added more information to explain the operation of the User button. 2. Added more detail on the removal of the USB Host port 3. Added alternate BeagleBoard LOGO. 4. Removed capacitor C70 from the board to improve the rise/fall time of the 32KHz clock. 5. Added a note to indicate that the expansion header is not installed. 6. Added a detailed section covering the board verification process. 7. Added BOM section with link. 8. Added links to the PCB and Schematic information on BeagleBoard.org. 9. Changed the name to "BeagleBoard System Reference Manual"	9/4/2008	GC
B6	1. In Tables 17 and 21, pins 6 and 10 had the B and C columns swapped. 2. Fixed typing errors 3. Updated to reflect changes made in the Rev B1 PCB to change U9 and U11 package. 4. Corrected several other typos on page 54. 5. Updated table 18.	10/15/08	GC
B7	1. Changed OMAP3530 processor to revision ES3.0	11/17/08	GC
B7.2	1. Corrected typo in section 12.10.2 2. Added more information to section 12.5 to say SERAIL instead of UART.	1/4/09	GC

2.2   Revision A vs. B4

The following bullets describe the differences between the Rev A and Rev B4
BeagleBoards.

o The DC power connector is now functional on the Rev B4 board.

o The USB Host has been removed due to issues with the PCB layout. While
   working, the USB Host port was not reliable and caused the PHY to lockup under
   heavy loads.

o EMU0 jumper was removed. This was required on the Rev A board when using
   the older version of the OMAP3530 processor. It is no longer needed on the
   version of silicon used on the Rev B board.

o Added a capacitor to the VBUS signal. There were some cases reported that
   certain USB hubs would not connect due to noise on the VBUS rail.

o The LEDS USR0 and USR1 can now be controlled separately. The previous
   version had the signals shorted together.

2.3   Revision B4 vs. B5

The following bullets describe the differences between the **Rev B4** and **Rev B5**
BeagleBoards.
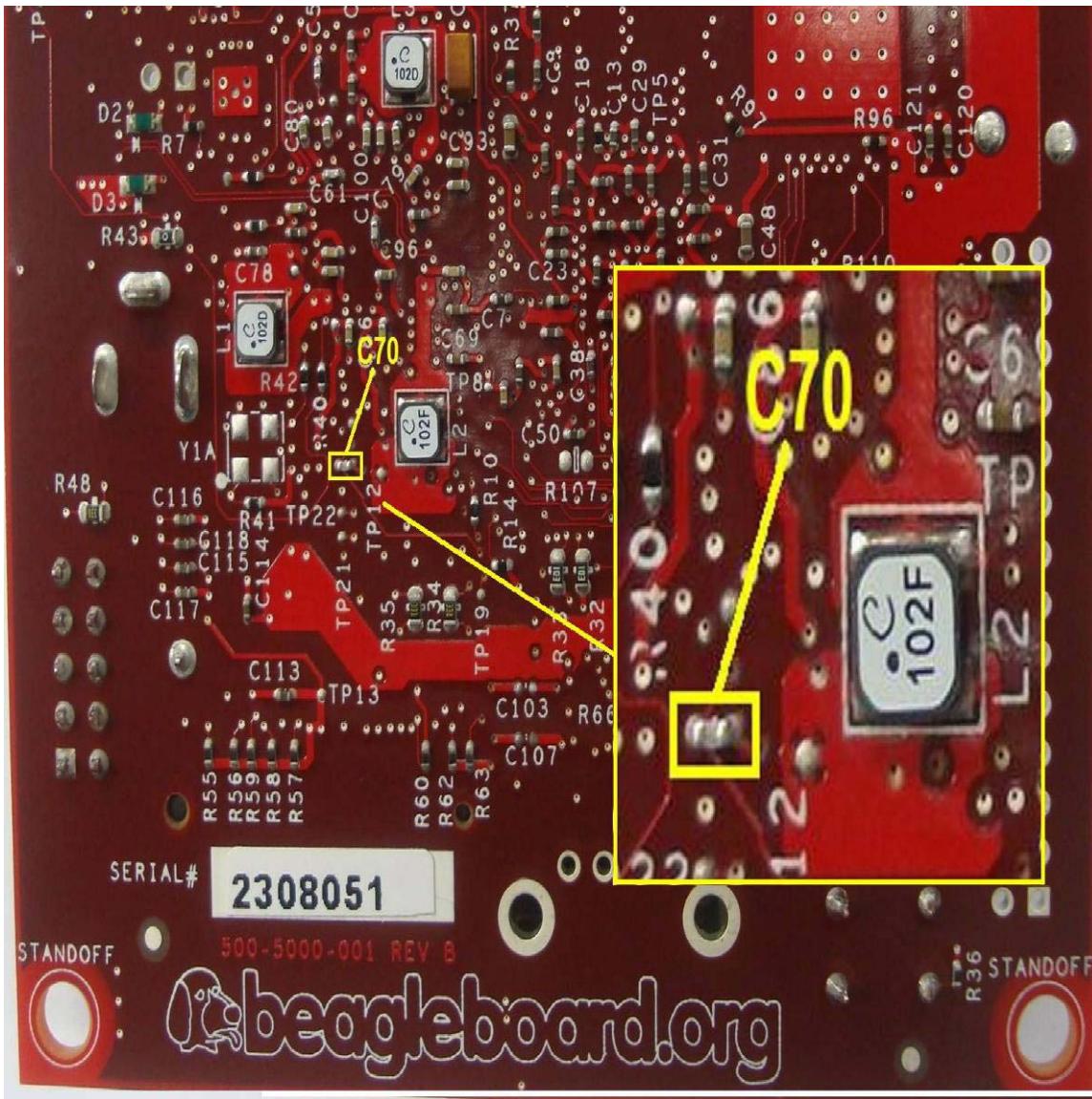
o Capacitor C70 was removed to improve the 32KHz clock rise and fall time. This
   fixes the GPT1 timer issue. This change can be easily made by the board owner
   using a soldering iron.

2.4   Revision B5 vs. B6

The following bullets describe the differences between the **Rev B5** and **Rev B6**
BeagleBoards.

o The package used for U9 and U11 was changed in the PCB. This was to address
   the issue we had with failures due to issues with the package. No electrical
   changes were made.
o The Dx capacitor, which was hand mounted, was incorporated into the PCB
   layout. No electrical changes were made.
o The name of the TWL4030 was replaced with TWL4030 to reflect the catalog
   version of the device. The TWL4030 is still being used on the board.

o **Table 17** and **Table 21** were corrected to reflect the proper naming of the pins. No impact on the schematic.
o The OMAP3530 ES3.0 is being used on the B6 revision where B5 and earlier used the ES3.0.

Location of C70

2.5   Revision B6 vs. B7

The Revision B7 boards are built with OMAP3530 Rev ES3.0 silicon versus the ES2.1 on the Rev B6 board. There are no additional features on the BeagleBoard as a result of this change.

3.0 Definitions and References

3.1 Definitions

SD- Secure Digital
SDIO- Secure Digital Input Output
MMC- Multimedia Card
MDDR- Mobile Dual Data Rate
SDRAM- Synchronous Dual Access Memory
OMAP3530- The CortexA8 based System on a Chip from Texas Instruments.

4.0 BeagleBoard Overview

The BeagleBoard is an OMAP3530 platform designed specifically to address the Open Source Community. It has been equipped with a minimum set of features to allow the user to experience the power of the OMAP3530 and is not intended as a full development platform as many of the features and interfaces supplied by the OMAP3530 are not accessible from BeagleBoard. By utilizing standard interfaces, the BeagleBoard is highly extensible to add many features and interfaces.

4.1 BeagleBoard Usage Scenarios

The **Figure 1** provides an example of a few of the various usage scenarios for the BeagleBoard.
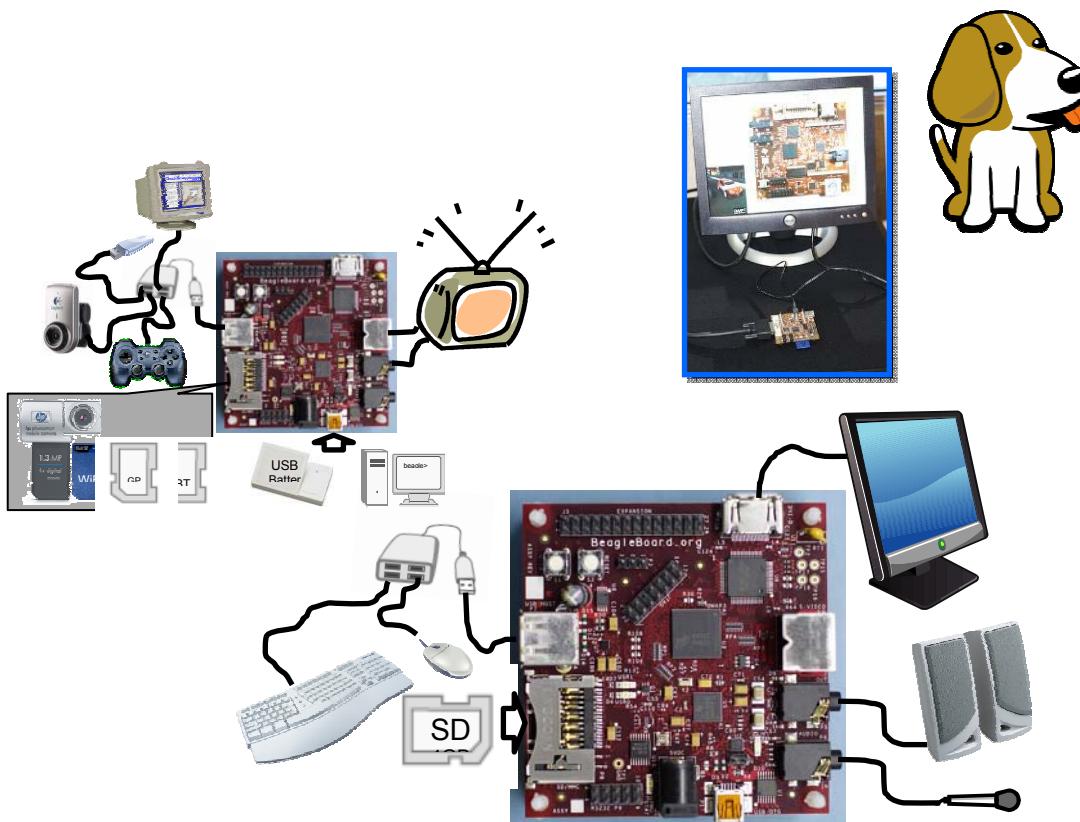
Figure 1. BeagleBoard Usage Scenarios

5.0 BeagleBoard Specification

This section covers the specifications of the BeagleBoard and it also provides a high level description of the major components and interfaces that make up the BeagleBoard.

5.1 BeagleBoard Features

Table 2 provides a list of the BeagleBoard's features.

Table 2. BeagleBoard Features

	Feature	
Processor	OMAP3530530 ES3.0	
POP Memory	Micron	
	2Gb NAND (256MB)	1Gb MDDR SDRAM (128MB)
TWL4030 PMIC	Power Regulators	
	Audio CODEC	

	Reset	
	USB OTG PHY	
Debug Support	14-pin JTAG	GPIO Pins
	UART	LEDs
PCB	3.1" x 3.0" (78.74 x 76.2mm)	6 layers
Indicators	Power	2-User
	PMU	
HS USB 2.0 OTG Port	Mini AB USB connector	
	TWL4030 I/F	
	MiniAB	
Audio Connectors	3.5mm	3.5mm
	L+R out	L+R Stereo In
SD/MMC Connector	6 in 1 SD/MMC/SDIO	4/8 bit support, Dual voltage
User Interface	1-User defined button	Reset Button
Video	DVI-D	S-Video
Power Connector	USB Power	DC Power
Expansion Connector (Not Populated)	Power (5V & 1.8V)	UART
	McBSP	McSPI
	I2C	GPIO
	MMC	

The following sections provide more detail on each feature and components on the BeagleBoard.

5.2   OMAP Processor

The BeagleBoard uses the OMAP3530 version ES3.0 and comes in a .4mm pitch POP package. POP (Package on Package) is a technique where the memory, NAND and SDRAM, are mounted on top of the OMAP3530. For this reason, when looking at the BeagleBoard, you will not find an actual part labeled OMAP3530.

5.3   Memory

The Micron POP memory is used on BeagleBoard and is mounted on top of the processor as mentioned. The key function of the POP memory is to provide:

   o   2Gb NAND x 16 (256MB)
   o   1Gb MDDR SDRAM x32 (128MB @ 166MHz)

No other memory devices are on the BeagleBoard. It is possible however, that additional memory can be added to BeagleBoard by installing a NAND based device in the SD/MMC slot or use the USB OTG port and a powered USB hub to drive a USB Thumb drive or hard drive. Support for this is dependent upon driver support in the OS.

5.4 TWL4030 Based Power Management

The TWL4030 is used to provide power to the BeagleBoard with the exception of the 3.3V regulator which is used to provide power to the DVI-D encoder and RS232 driver. In addition to the power it also provides:

- o Stereo Audio Out
- o Stereo Audio in
- o Power on reset
- o USB OTG PHY
- o Status LED

5.5 HS USB 2.0 OTG Port

The HS USB OTG port is the primary power source and communication link for the BeagleBoard and derives power from the PC over the USB cable. The client port is limited in most cases to 500mA by the PC. A single PC USB port is sufficient to power the BeagleBoard. If additional devices are connected to the expansion bus and the 5V rail is used, then the power required could exceed that supplied by a USB port or Hub.

It is possible to take this to 1A by using a Y cable if additional power is needed for either the USB host port or an expansion card. **Figure 2** shows and example of the Y-Cable for the USB.
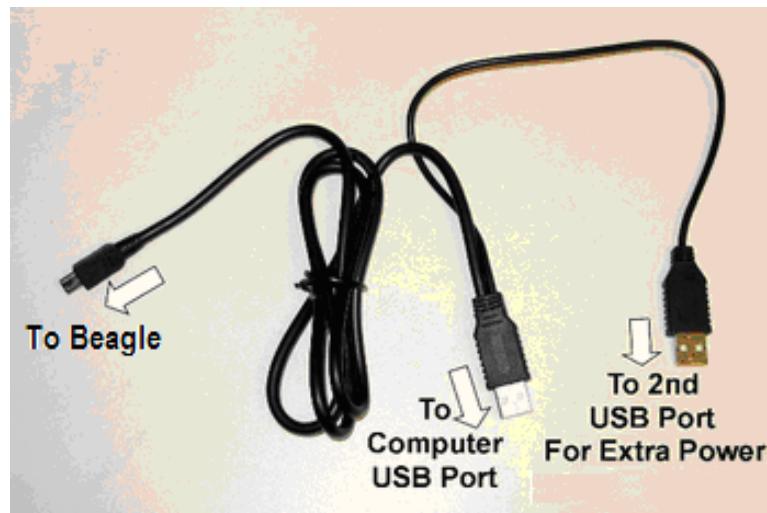
Figure 2.   USB Y-Cable

The BeagleBoard requires a single minAB to USB A cable or as mentioned a Y-Cable can be used if needed. There is an option to provide external power to the BeagleBoard using a 5V DC supply and is discussed later in this section.

5.6   Stereo Audio Output Connector

A 3.5mm standard stereo output audio jack is provided to access the stereo output of the onboard audio CODEC. The Audio CODEC is provided by the TWL4030.

5.7   Stereo Audio In connector

A 3.5mm standard stereo audio input jack is provided to access the stereo output of the onboard audio CODEC.

5.8   S-Video Connector

A 4 pin DIN connector is provided to access the S-Video output of the BeagleBoard. This is a separate output from the OMAP processor and can contain different video output data from what is found on the DVI-D output.

It will support NTSC or PAL format output to a standard TV. The default is NTSC, but can be changed via the Software.

5.9   DVI-D Connector

The BeagleBoard can drive a LCD panel equipped with a DVI-D digital input. This is the standard LCD panel interface of the OMAP3530 and will support 24b color output. DDC2B (Display Data Channel) or EDID (Enhanced Display ID) support over I2C is provided in order to allow for the identification of the LCD monitor type and settings.

The BeagleBoard is equipped with a DVI-D connector that uses an HDMI connector that was selected for its small size. It does not support the HDMI interface and is used to provide the DVI-D interface only. The user must use a HDMI to DVI-D cable or adapter to connect to a LCD monitor. This cable or adapter is not provided with the BeagleBoard.

5.10   SD/MMC 6 in 1 Connector

A 6 in 1 SD/MMC connector is provided as a means for expansion and can support such devices as:

  o   WiFi Cards
  o   Camera
  o   Bluetooth Cards

- o   GPS Modules
- o   SD Memory Cards
- o   MMC Memory Cards
- o   SDIO Cards
- o   MMCMobile cards
- o   RS-MMC Cards
- o   miniSD Cards

It supports the MMC4.0 (MMC+) standard and can boot from MMC or SD cards. It will support both 4 and 8 bit cards, but 8 Bit cards are 1.8V only and the boot mode supports a 3V card. **In order to boot from MMC/SD the card must be a 3V 4 bit card.**

One of the nice features is that the OMAP3530 can be booted from the SD/MMC. By holding the User button and forcing a reset, the BeagleBoard will boot from the SD/MMC.

5.11   Reset Button

When pressed and released, causes a full power on reset of the BeagleBoard. It should be noted that currently, the reset will not work when the Linux kernel is running. To reset the board from the kernel operation, a power cycle is required.

5.12   User/Boot Button

A button is provided on the BeagleBoard to provide two functions:

- •   Force a change in the boot sequence of the OMAP3530.
- •   Used as an application button that can be used by SW as needed.

When used in conjunction with the RESET button, it will force a change to the order in which boot sources are checked as viable boot sources.

If the button is pressed while the RESET button is released, the sequence becomes:

- o   USB
- o   UART
- o   MMC1
- o   NAND

Even though the NAND may have a program in it, if a card is placed in the MMC slot, it will try to boot from it first. If it is not there, it will boot from NAND.

There is also the option to have a serial download application that will program the NAND if connected to the serial or USB ports. In this scenario the internal ROM will stop on either the serial or USB port and start the download process from there. It does require an application to be run on the host PC in order to perform this function.

If the user button is not pressed at reset, the sequence in which the internal ROM looks for viable boot sources is as follows:

- o NAND
- o USB
- o UART3
- o MMC1

In this case, NAND overrides every option and will always boot from NAND if there is data in the NAND. If the NAND is empty, then the other sources are available to be used based on the boot order.

5.13   Indicators

There are three green LEDs on the BeagleBoard that can be controlled by the user.

- o One on the TWL4030 that is programmed via the I2C interface
- o Two on the OMAP3530 Processor controlled via GPIO pins

There is a fourth LED on the BeagleBoard that provides an indication that power is supplied to the board.

5.14   Power Connector

Power will be supplied via the USB OTG connector and if a need arises for additional power, such as when a board is added to the expansion connectors, a larger wall supply 5V can be plugged into the optional power jack. When the wall supply is plugged in, it will remove the power path from the USB connector and will be the power source for the whole board. The power supply is not provided with the BeagleBoard.

When using the USB OTG port in the host mode, the DC supply must be connected as the USB port will be used to provide limited power to the hub at a maximum of 100mA, so a hub must be powered. The 100mA is not impacted by having a higher amperage supply plugged into the DC power jack. The 100mA is a function of the OTG port itself.

WARNING: DO NOT PLUG IN ANYTHING BUT 5V TO THE DC CONNECTOR OR THE BOARD WILL BE DAMAGED!

Make sure the DC supply is regulated and a clean supply.

5.15   JTAG Connector

A 14 pin JTAG header is provided on the BeagleBoard to facilitate the SW development and debugging of the board by using various JTAG emulators. The interface is at 1.8V on all signals. Only 1.8V Levels are supported. **DO NOT expose the JTAG header to 3.3V.**

5.16   RS232 Header

Support for RS232 via UART3 is provided by a 10 pin header on the BeagleBoard for access to an onboard RS232 transceiver. It does require an IDC to DB9 flat cable, which is not provided, to access the serial port.

5.17   Expansion Headers

An option for a single 28 pin header is provided on the board to allow for the connection of various expansion cards that could be developed by the users or other sources. Due to multiplexing, different signals can be provided on each pin providing more that 24 actual signal accesses. This header is not populated on the BeagleBoard so that based on the usage scenario it can be populated as needed (Top, Bottom, Top right angle, or Bottom Right angle).

5.18   BeagleBoard Mechanical Specifications

```
Size:              3.0" x 3.1"
Max height:        TBM
Layers:            6
PCB thickness:     .062"
RoHS Compliant:    Yes
Weight:            TBW
```

5.19   Electrical Specifications

Table 3 is the electrical specification of the external interfaces to the BeagleBoard.

Table 3.   BeagleBoard Electrical Specification

Specification	Min	Typ	Max	Unit
Power				
Input Voltage USB		5	5.2	V
Current USB		350		mA
Input Voltage DC	4.8	5	5.2	V
Current DC		350		mA
Expansion Voltage (5V)	4.8	5	5.2	V

Current (Depends on source current available)		1		A
Expansion Voltage (1.8V)	1.75	1.8	1.85	V
Current			100	mA
USB Client				
High Speed Mode			480	Mb/S
Full Speed Mode			12.5	Mb/S
Low Speed Mode			1.5	Mb/S
RS232				
Transmit				
High Level Output Voltage		5	5.4	V
Low Level output voltage		-5	-5.5	V
Output impedance		+/-35	+/-60	mA
Maximum data rate	250			Kbit/S
Receive				
High level Input Voltage	-2.7	-3.2		V
Lo Level Input Voltage			.4	
Input resistance	3	5	7	Kohms
JTAG				
Realview ICE Tool			30	MHz
XDS560			30	MHz
XDS510			30	MHz
Lauterbach(tm)			30	MHz
SD/MMC				
Voltage Mode 1.8V	1.71	1.8	1.89	V
Voltage Mode 3.0V	2.7	3.0		V
Current			220	mA
Clock			48	MHz
DVI-D				
Pixel Clock Frequency	25		65	MHz
High level output voltage		3.3		V
Swing output voltage	400		600	mVp-p
Maximum resolution			1024 x 768	
S-Video				
Full scale output voltage (75ohm load)	.7	.88	1	V
Offset voltage		50		mV
Output Impedance	67.5	75	82.5	Ohms
Audio In				
Peak-to-peak single-ended input voltage (0 dBFs)			1.5	Vpp
Total harmonic distortion (sine wave @ 1.02 kHz @ -1 dBFs)		-80	-75	dB
Total harmonic distortion (sine wave @ 1.02 kHz) 2 0 Hz to 20 kHz, A-weighted audio, Gain = 0 dB		-85	-78	dB
Audio Out				
Load Impedance @100 pF	14	16		ohms
Maximum Output Power (At 0.53 Vrms differential output voltage and load impedance = 16 Ohms)		17.56		mW
Peak-to-Peak output voltage			1.5	Vpp
Total Harmonic Distortion @ 0 dBFs		-80	-75	dB
Idle channel noise (20Hz to 20KHz)		-90	-85	dB

beagleboard.org

6.0   Product Contents

Under this section is a description of what comes in the box when the BeagleBoard is purchased.

6.1   BeagleBoard In the Box

The final packaged product will contain the following:

- o 1 Box
- o 1 BeagleBoard in an ESD Bag

<u>**NO CABLES ARE PROVIDED WITH THE BEAGLEBOARD.**</u>

Figure 3.   The Box

Figure 4.   Box Contents

6.2   Software on the BeagleBoard

The board ships with U-Boot and X-Loader flashed onto the BeagleBoard.

6.3   Repair

If you feel the board is in need of repair, follow the RMA Request process found at
http://beagleboard.org/support/rma

7.0   BeagleBoard Hookup

This section provides an overview of all of the connectors on the BeagleBoard and how
they should be used.

7.1   Connecting USB OTG

The USB OTG port connects to the PC host and uses a miniAB cable through which
power is provided to the BeagleBoard. If desired, the BeagleBoard may also be
connected to a self powered USB hub.

 Figure 5 shows where the cable is connected to the BeagleBoard.

If the OTG Port is to be used as a Host, the ID pin must be grounded. This means that
you must have a 5 pin cable connected to the OTG port on the BeagleBoard and you must
use a USB powered HUB.
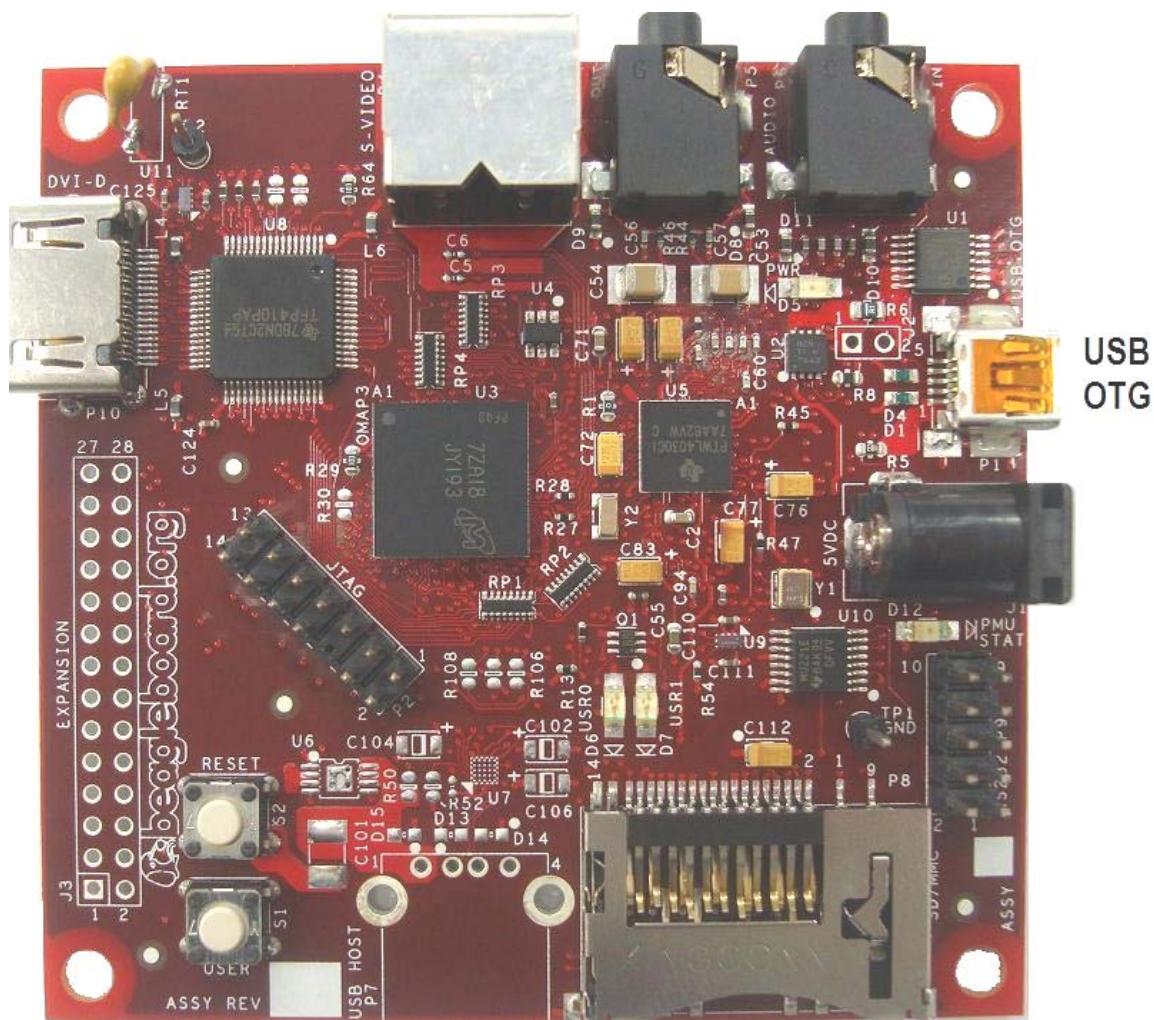
Figure 5.   USB OTG Connection

7.2   Connecting Optional Power

An optional DC supply can be used to power the BeagleBoard by plugging it into the power jack of the BeagleBoard. The power supply is not provided with the BeagleBoard, but can be obtained from various sources. You need to make sure the supply is a regulated 5V supply. **Figure 6** shows where to install the power supply into the power jack.
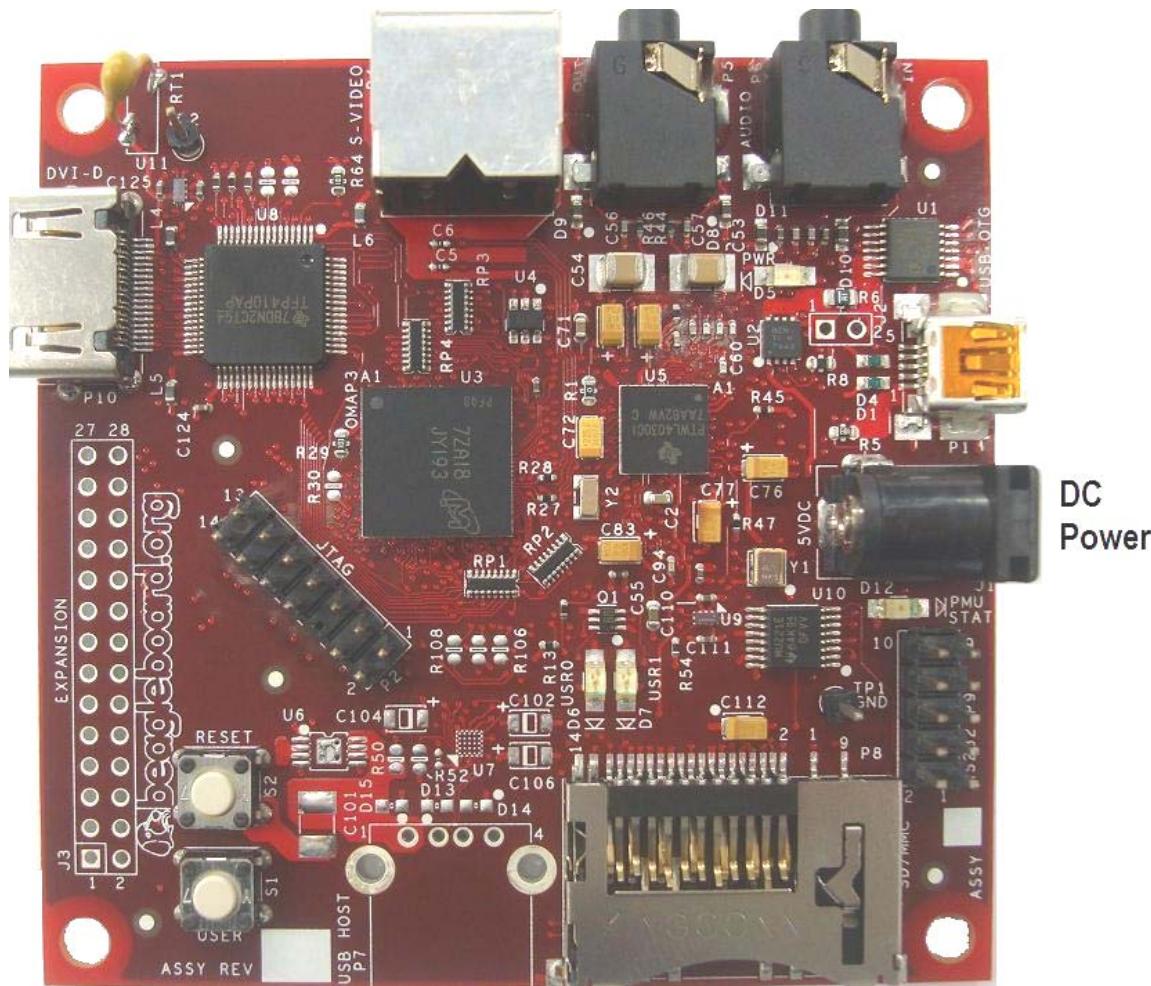
Figure 6.   DC Power Connection

The power supply must have a 2.1mm I.D x 5.5mm O.D. x 9.5mm and can be either straight or right angle. Connecting anything other than 5V will result in damage to the board.

If you are using the USB OTG port in the OTG or host mode, you must have an externa; DC supply powering the BeagleBoard.

7.3   Connecting JTAG

A JTAG emulator can be used for advanced debugging by connecting it to the JTAG header on the BeagleBoard. Only the 14pin version of the JTAG is supported and if a 20pin version is needed, you will to contact your emulator supplier for the appropriate adapter. **Figure 7** shows the connection of the **JTAG** cable to the BeagleBoard.
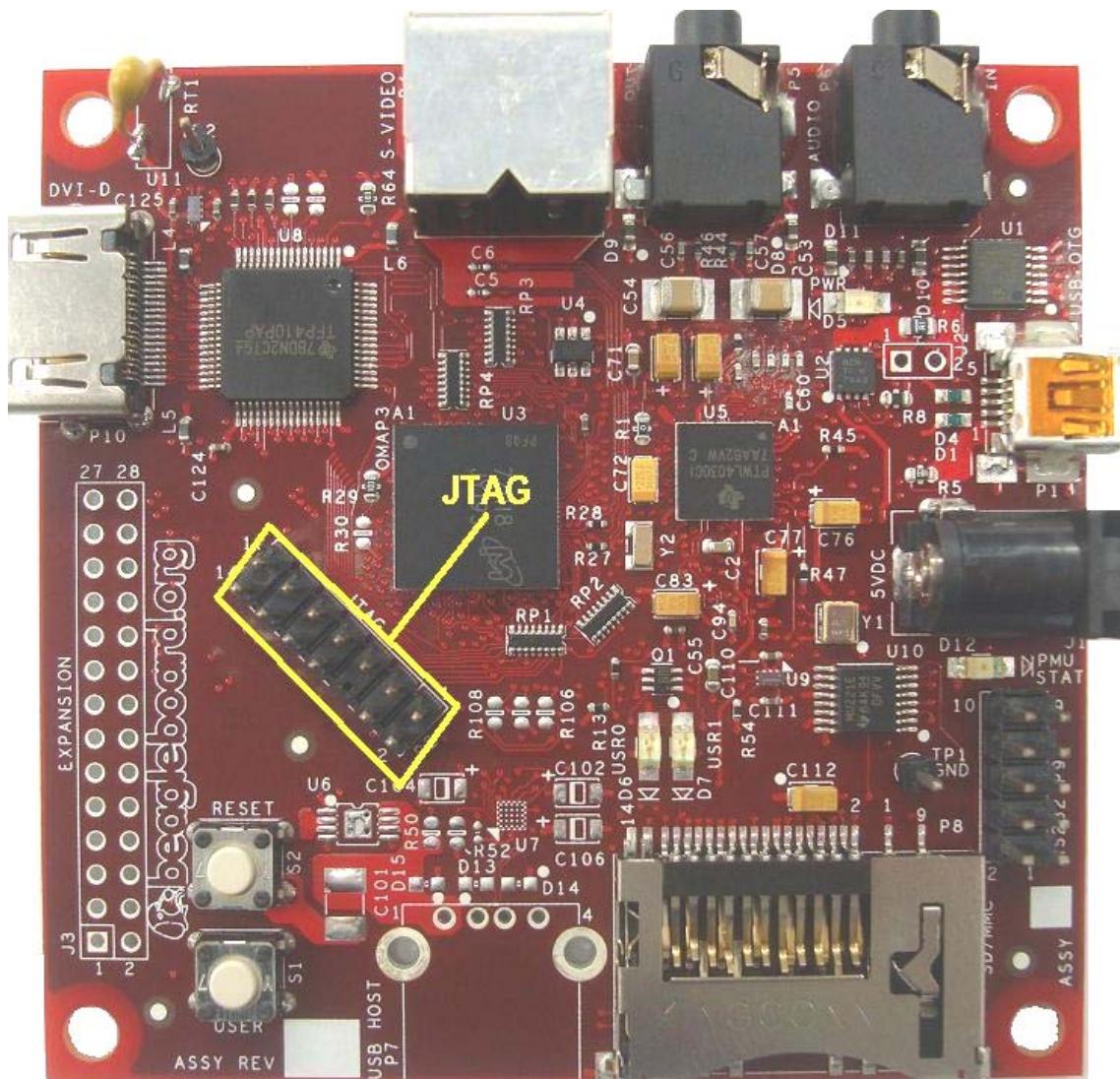
Figure 7.   BeagleBoard JTAG Connection

DO NOT expose the JTAG header to 3.3V. It supports 1.8V only.

7.4   Connecting Serial Cable

In order to access the serial port of the BeagleBoard a flat cable is required to connect to a PC. The adapter will not plug directly into the PC and will require an external Female to Female twisted cable (Null Modem) in order to connect it to the PC. The ribbon cable is not supplied with the BeagleBoard but can be obtained from numerous sources. **Figure 8** shows where the ribbon cable is to be installed.
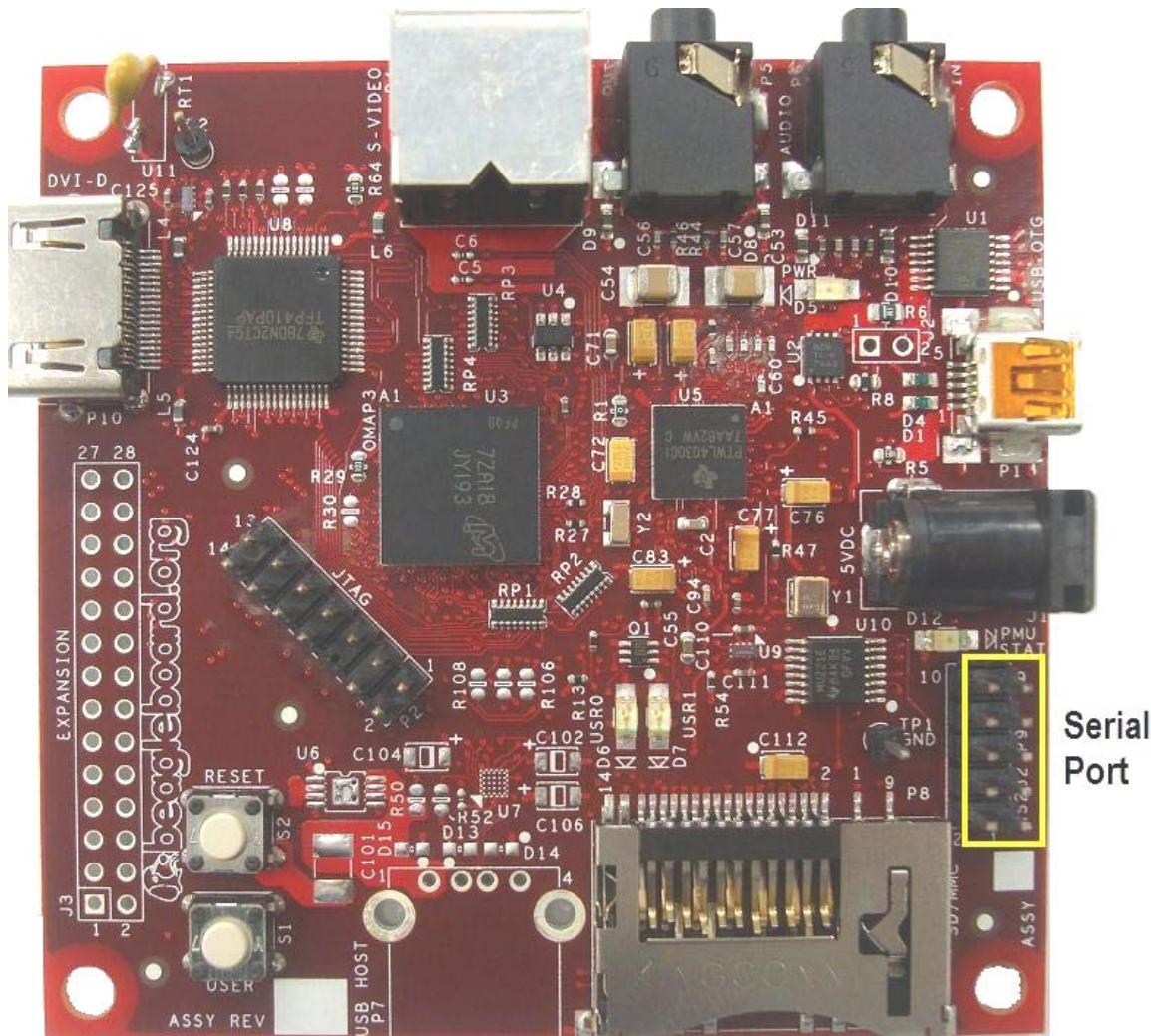
Figure 8.   BeagleBoard Serial Cable Connection

7.5   Connecting S-Video

An S-Video cable can be connected to the BeagleBoard and from there is can be connected to a TV or monitor that supports an S-Video input. This cable is not supplied with the BeagleBoard. **Figure 9** shows the connector for the S-Video cable.
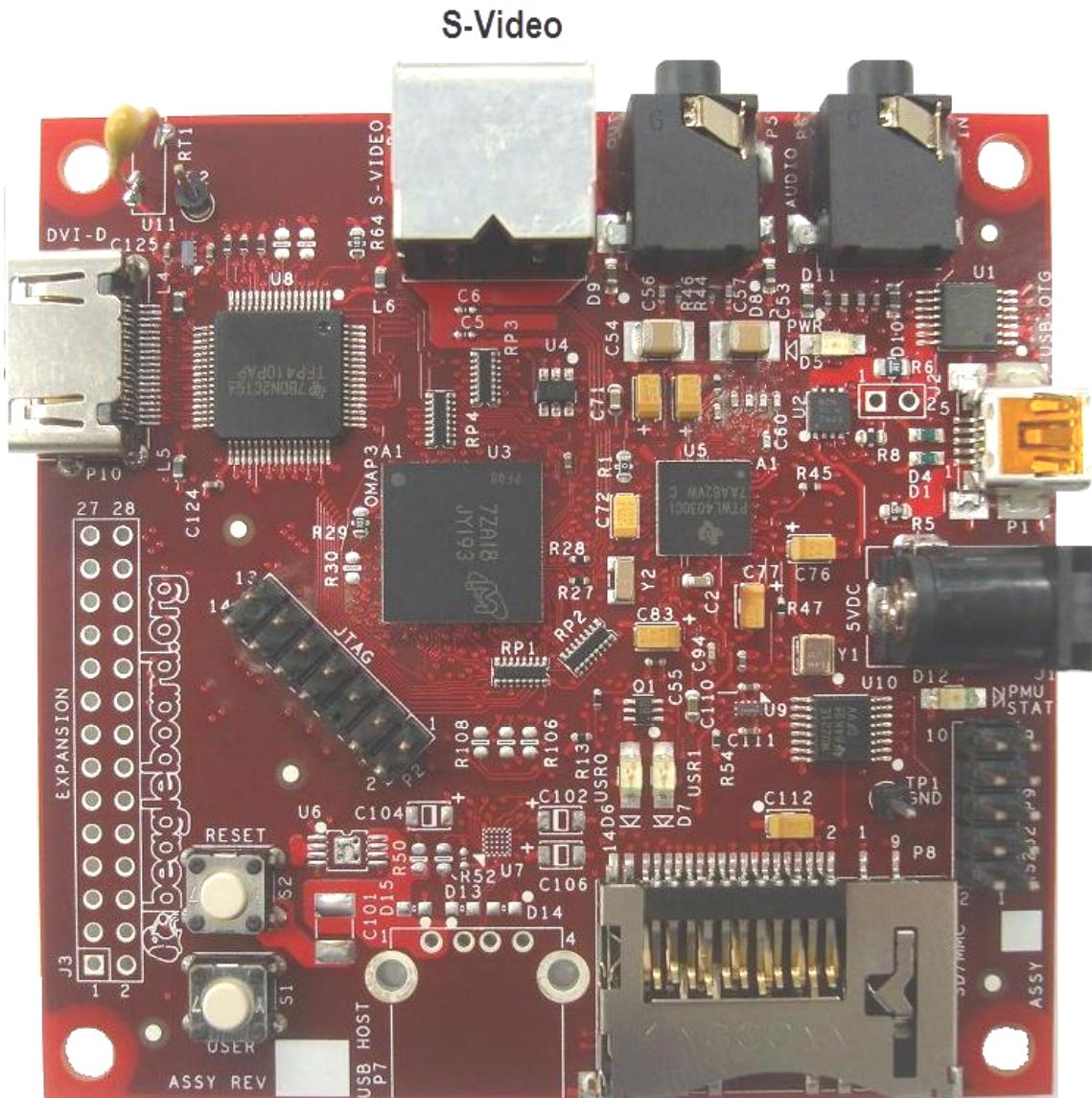
Figure 9.   BeagleBoard S-Video Connection

7.6   Connecting DVI-D Cable

In order to connect the DVI-D output to a monitor, a HDMI to DVI-D cable is required. This cable is not supplied with BeagleBoard but can be obtained through numerous sources. **Figure 10** shows the proper connection point for the cable.
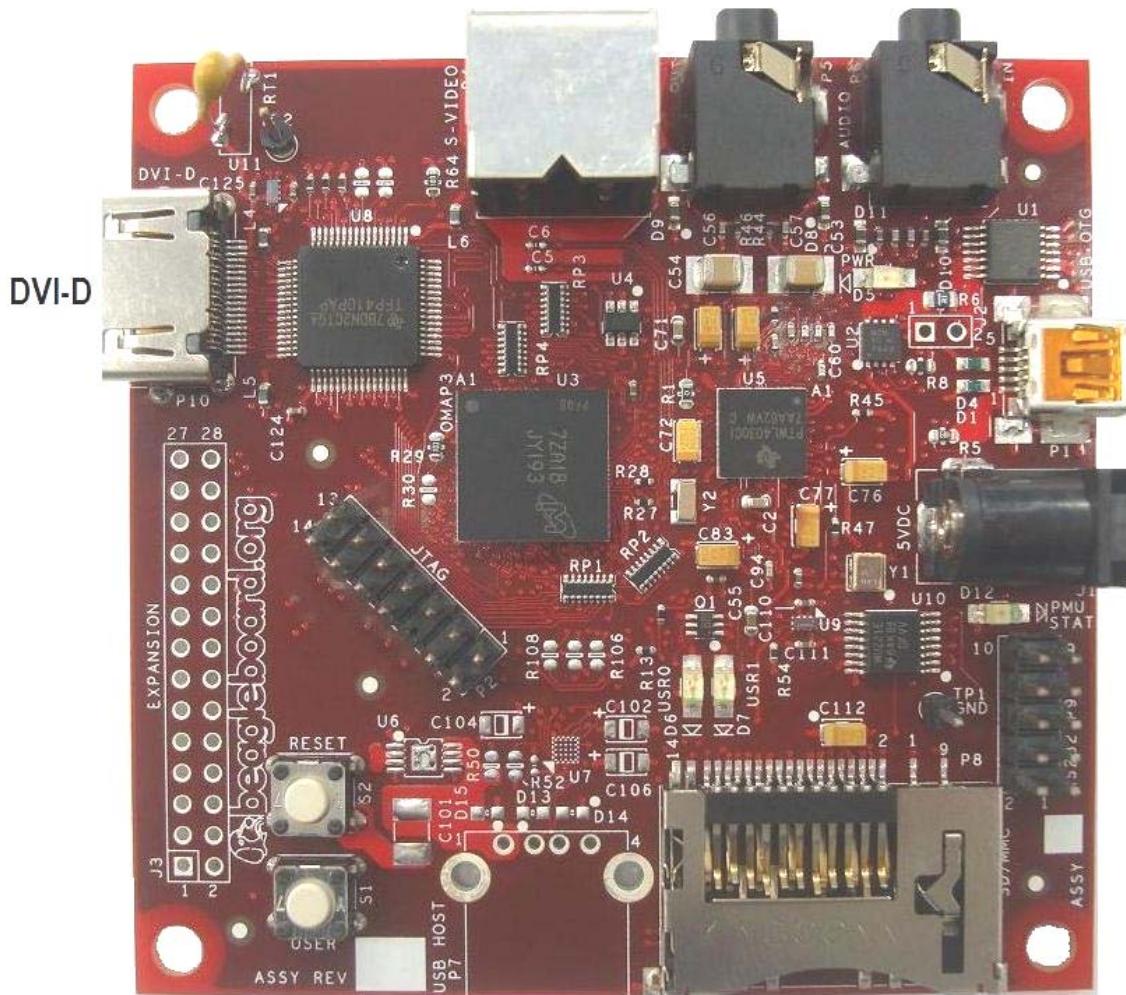
Figure 10.   BeagleBoard DVI-D Connection

<p style="text-align:center"><u>HDMI is not supported on the BeagleBoard.</u></p>

7.7   Connecting Stereo Out Cable

An external Audio output device, such as external stereo powered speakers, can be connected to the BeagleBoard via a 3.5mm jack. The audio cables are not provided with BeagleBoard, but can be obtained from just about anywhere. **Figure 12** shows how the cable connected to the stereo out jack.
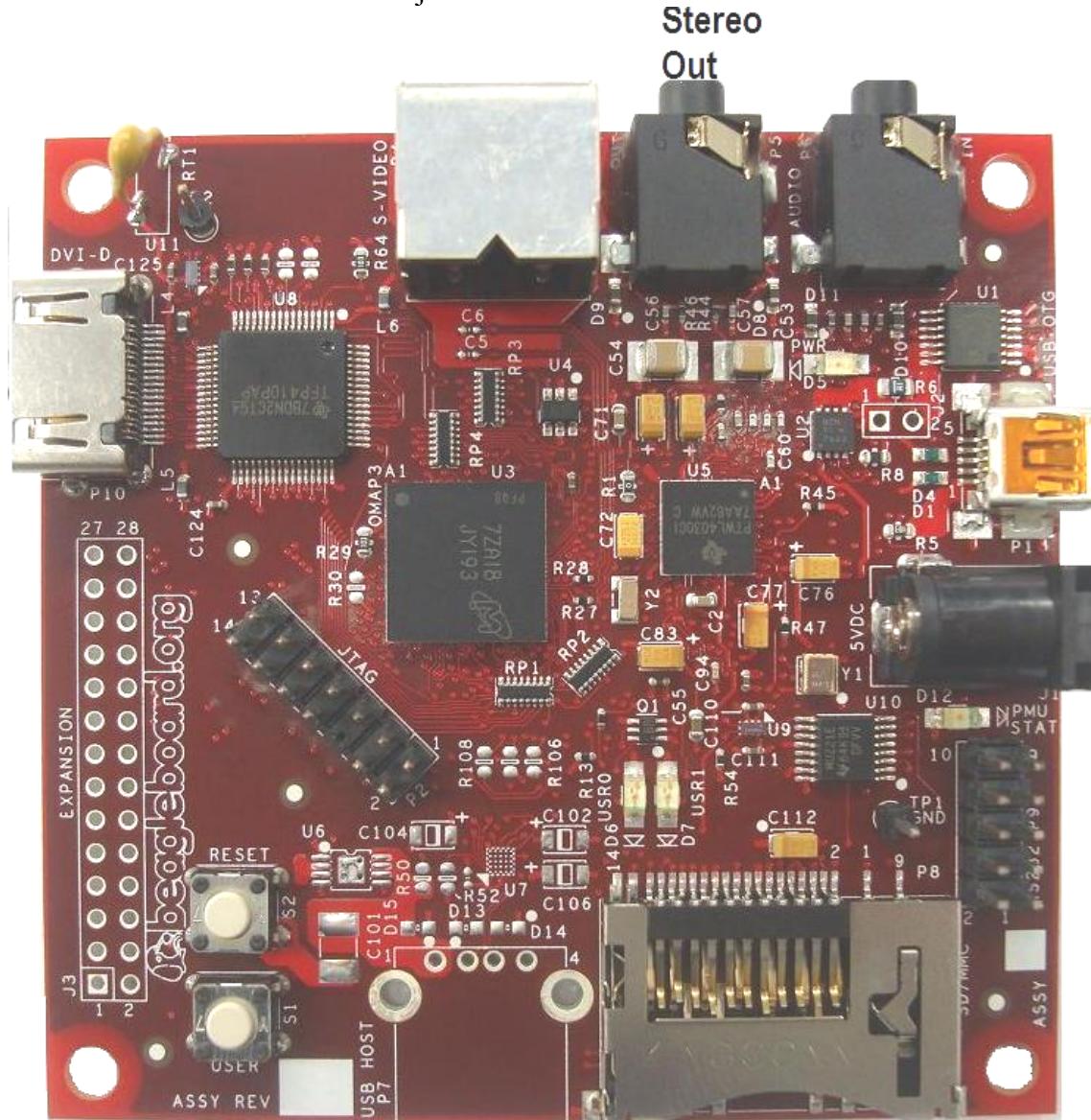
Figure 11.   BeagleBoard Audio Cable Connection

7.8   Connecting Stereo In Cable

External Audio input devices, such as a powered microphone or the audio output of a PC or MP3 player, can be connected to the via a 3.5mm jack. The audio cables are not provided with BeagleBoard, but can be obtained from just about any source. **Figure 13** shows how the cable is connected to the stereo input jack.
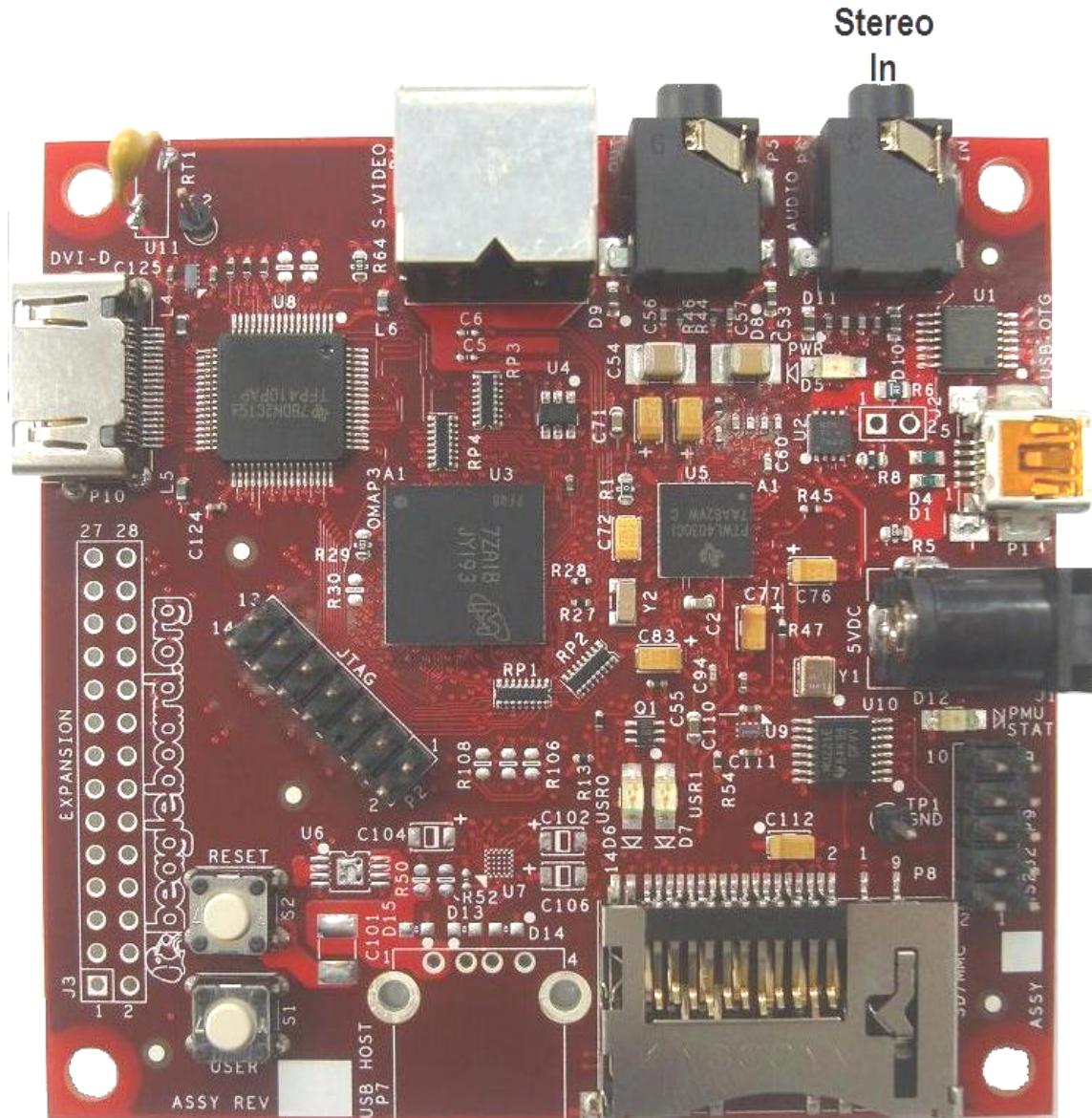
Figure 12.   BeagleBoard Audio Cable Connection

7.9   Indicator Locations

There are four green indicators on the BeagleBoard. One of them, POWER, indicates that the main supply is active. The other three can be controlled by the software. **Figure 14** shows the location of each indicator.
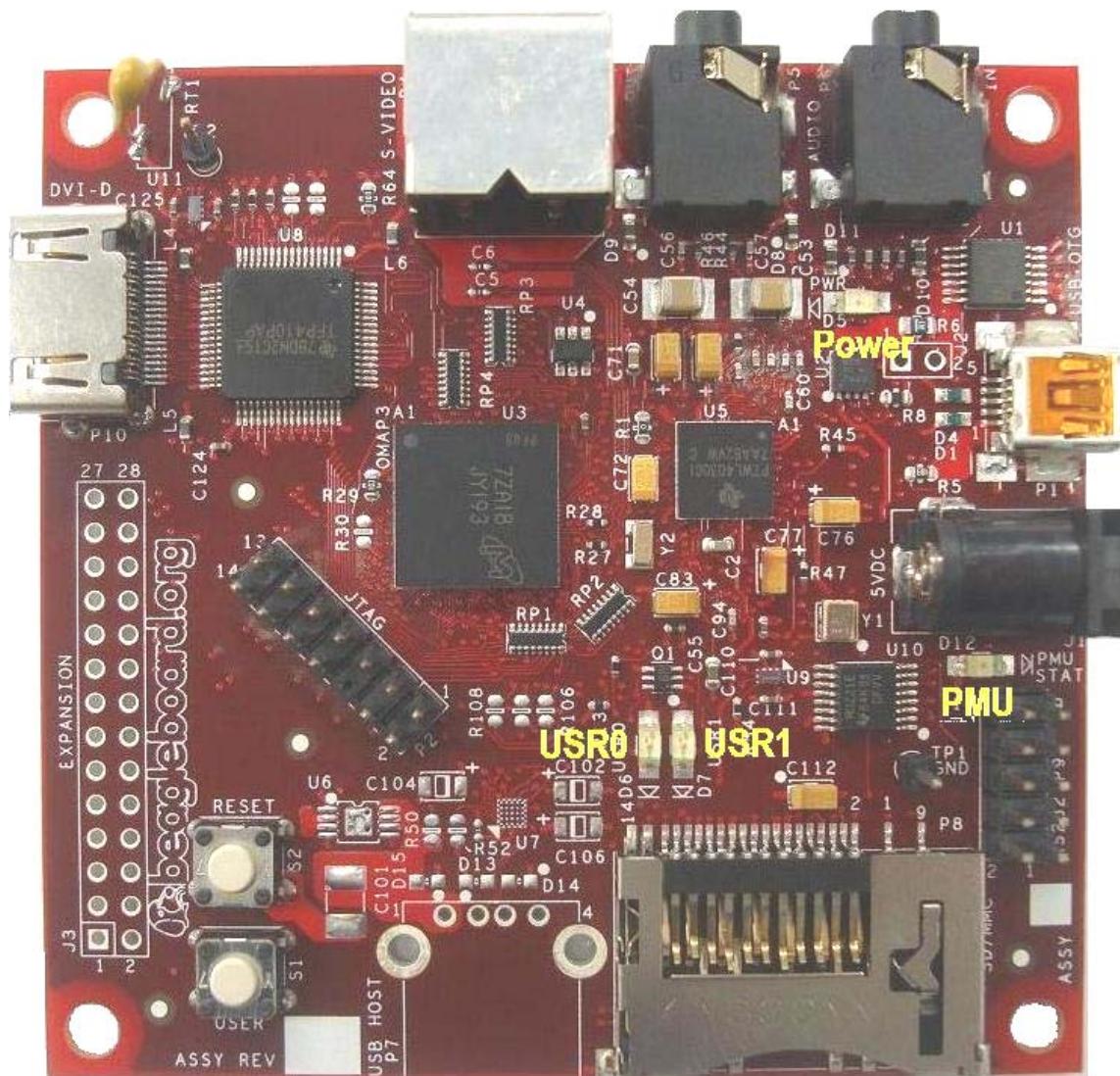
Figure 13.   BeagleBoard Indicator Location

7.10   Button Locations

There are two buttons on the BeagleBoard; the **RESET** button when pressed will force a full board reset and the **USER** button which can be used by the SW for user interaction. If the user holds the **USER** button down while pressing and releasing the **RESET** button, the BeagleBoard will enter the ROM boot loader mode. **Figure 15** shows the location of the buttons.
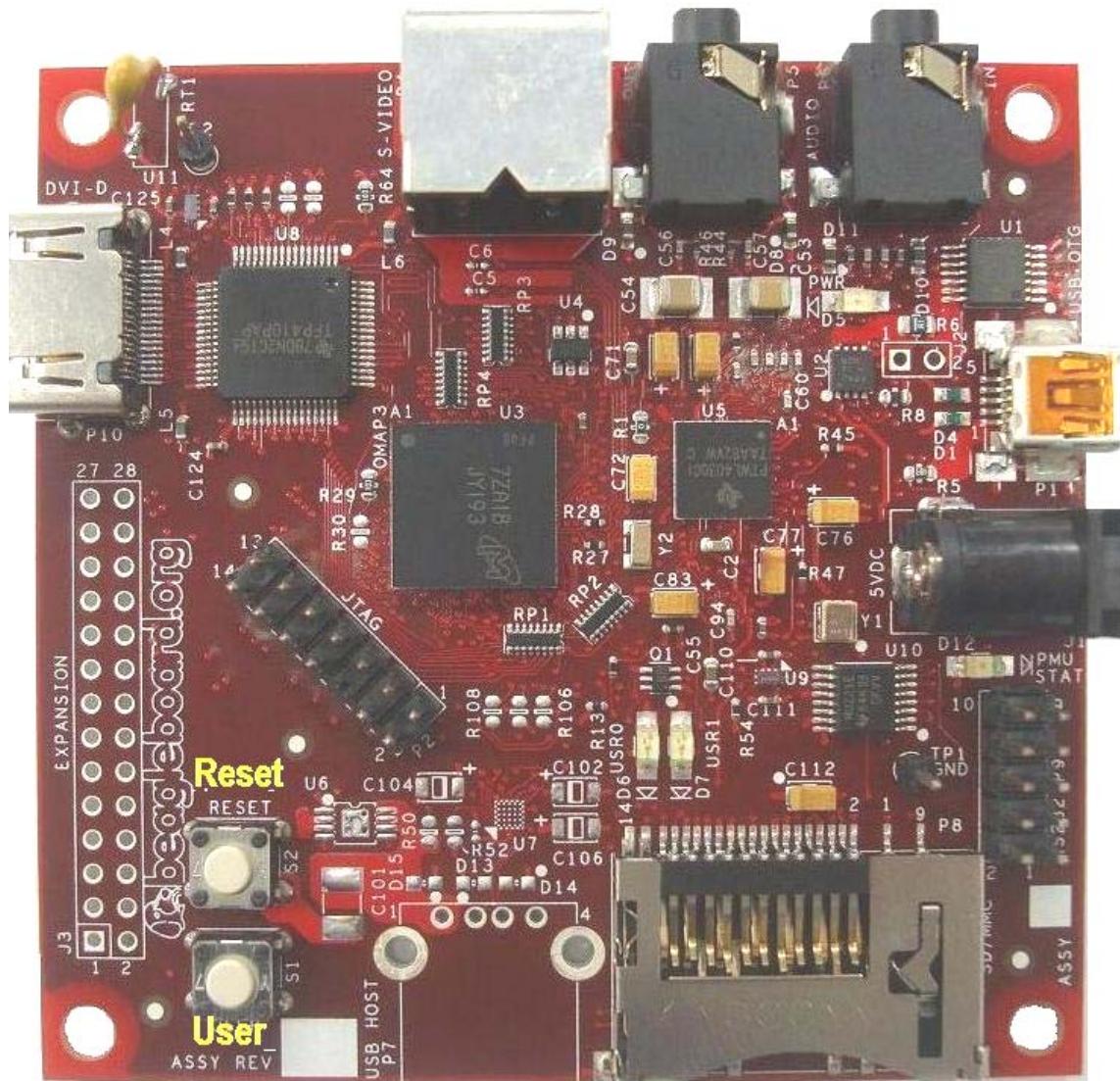
Figure 14.   BeagleBoard Button Location

7.11   SD/MMC Connection

The SD/MMC connector can be used for Memory or SDIO type cards. This is a full size connector and will support various cards. Whether a particular card is supported or not, is dependent on the available SW drivers. **Figure 16** shows the location of the SD/MMC connector.
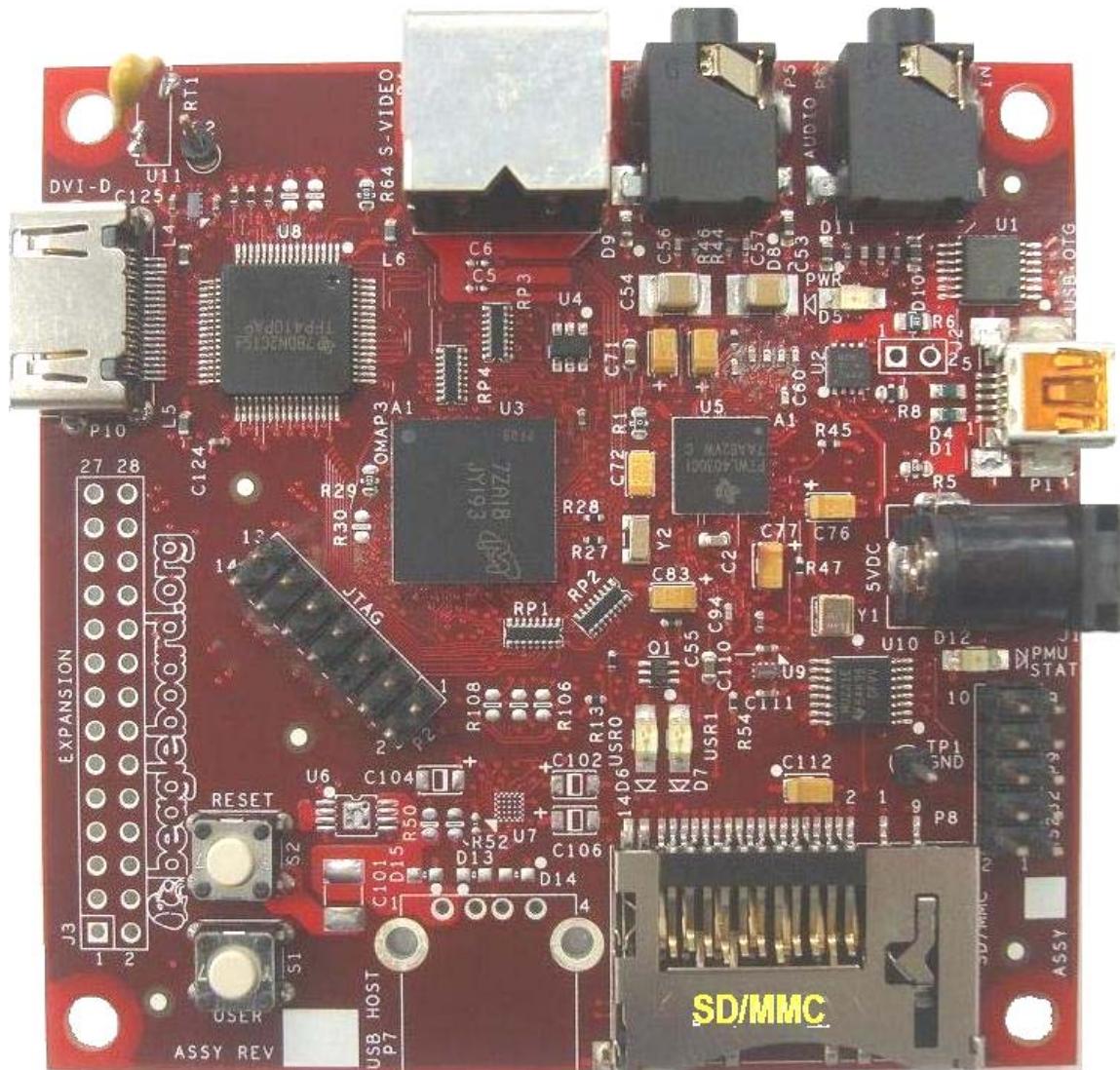
Figure 15.   BeagleBoard SD/MMC Location

8.0   BeagleBoard System Architecture and Design

This section provides a high level description of the design of the BeagleBoard and its overall architecture.

8.1   System Block Diagram

Figure 16 is the high level block diagram of the BeagleBoard. If you will notice, the block diagram is configured to match the component placement of the BeagleBoard.
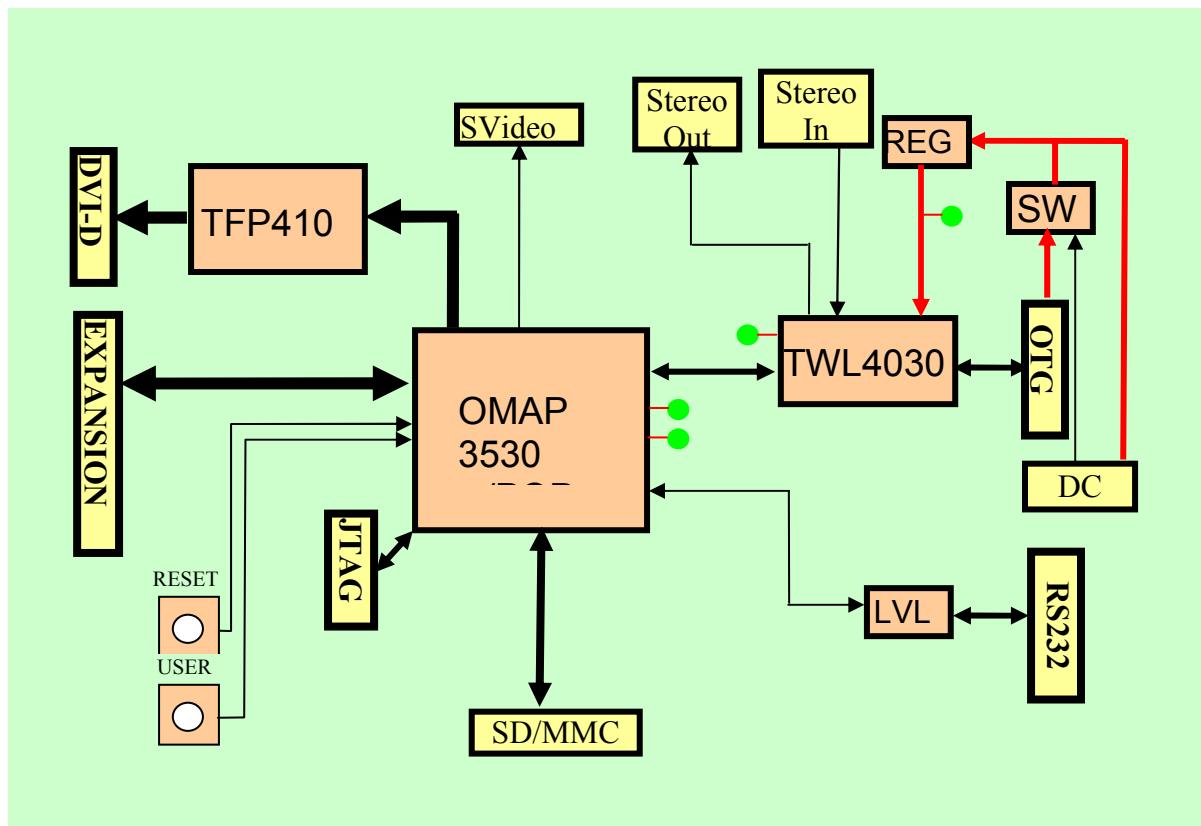
Figure 16.   BeagleBoard High Level Block Diagram

Figure 17 shows the location of the components as shown in the block diagram and is of the full assembly. The expansion and DC connectors are not provided on the Revision A version of the BeagleBoard. The Revision B versions contain the DC connector, but not the USB Host port.
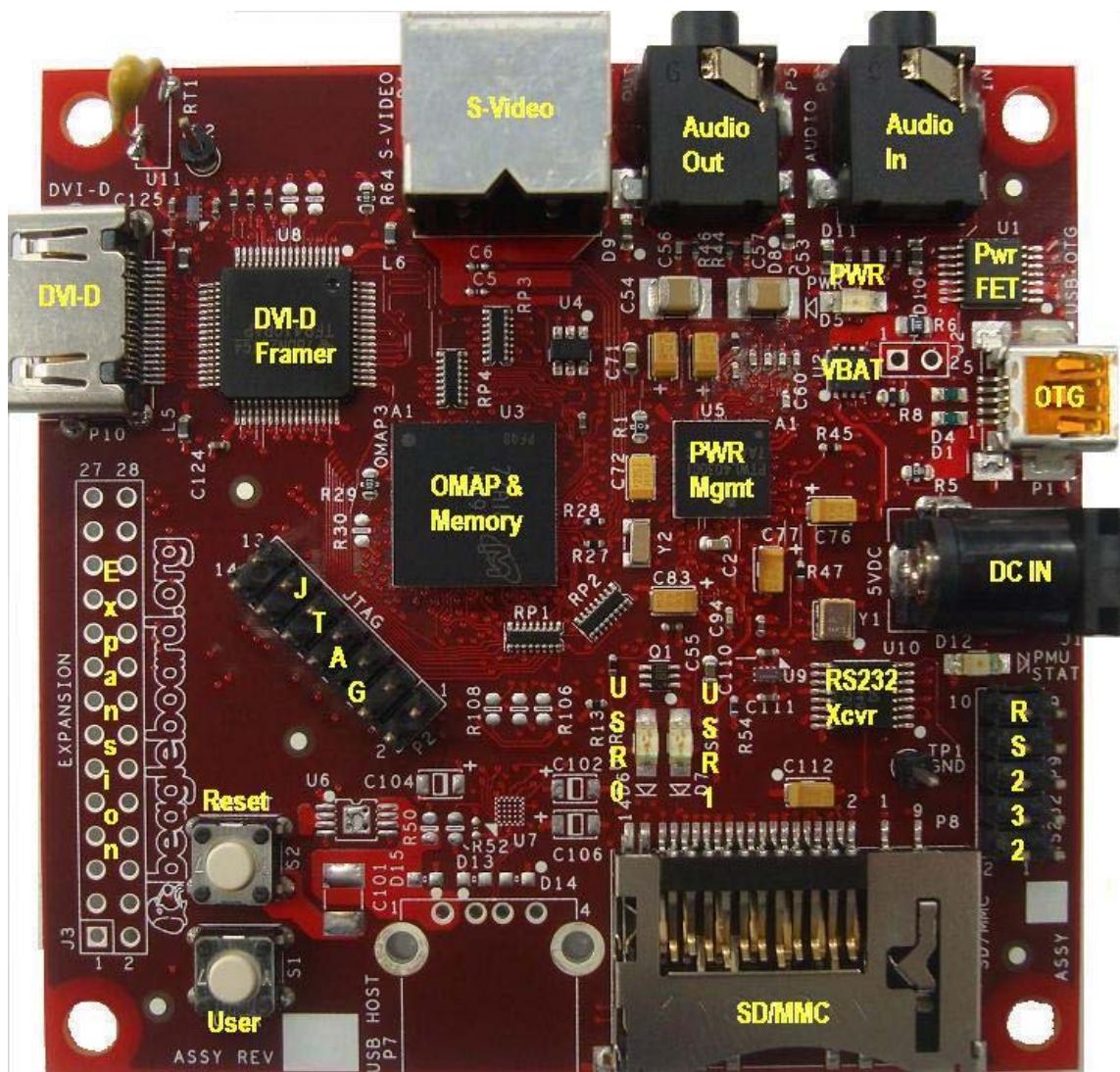
Figure 17. BeagleBoard Top Side Components

There are no key components on the back of the BeagleBoard, but **Figure 18** has been provided for completeness.
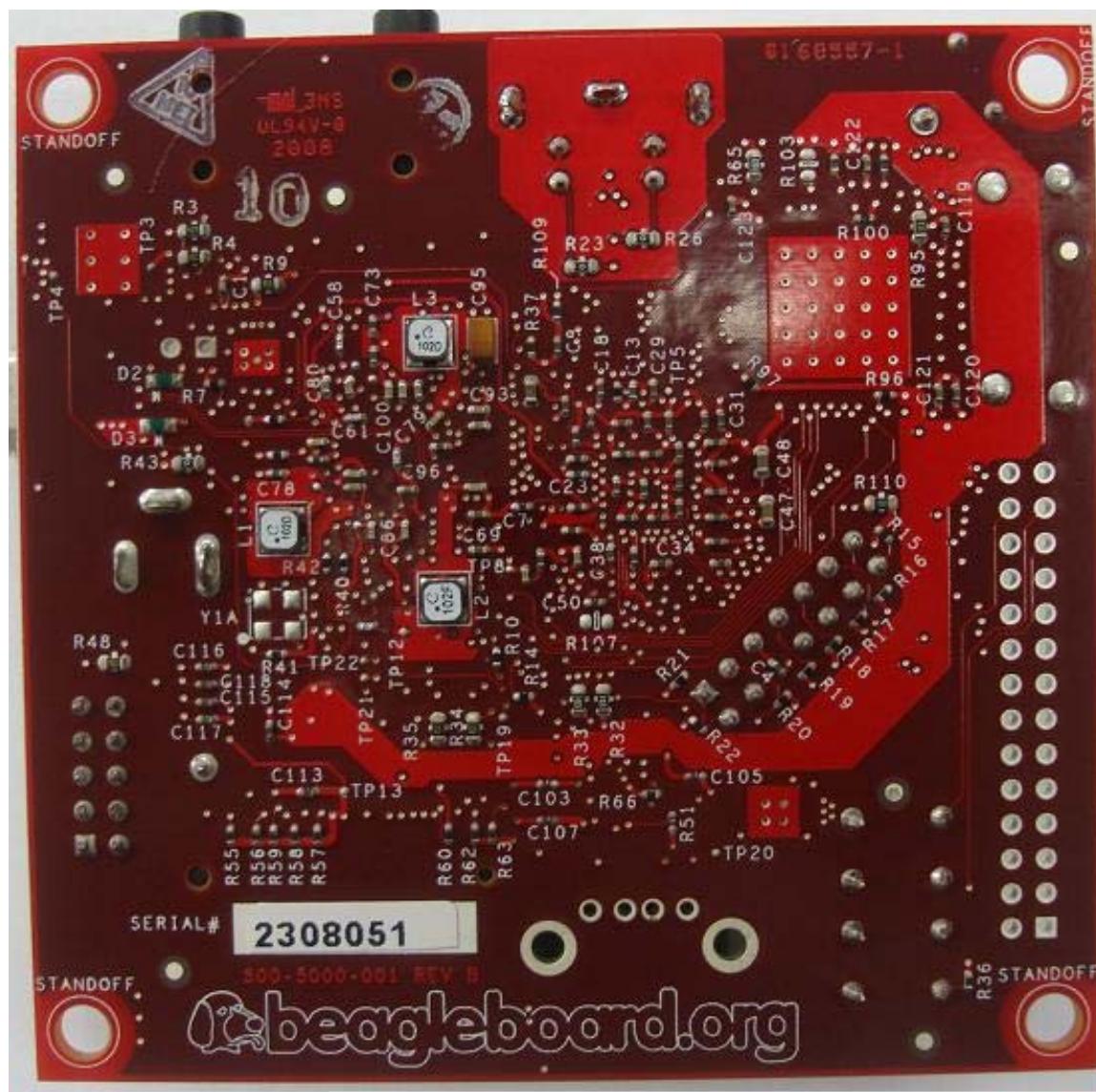
Figure 18.   BeagleBoard Backside Components

This remainder of this section describes in detail the architecture and design of the BeagleBoard.

You will notice certain things in this section.

o The schematic has been created for each section showing only the pertinent components and their connections.
o The pin names differ from the actual schematic. For ease of reading, the names have been truncated to only show the specific functions of that pin as used in the design.

8.2   Input Power

There are two possible sources of the 5V required by the BeagleBoard. It can come from the USB OTG port connected to a PC, powered USB HUB, or a 5V DC supply. The USB supply is sufficient to power the BeagleBoard. However, depending on the load needed by the expansion port on BeagleBoard, additional power may be required. This is where the DC supply comes in to play.

WARNING: DO NOT PLUG IN ANYTHING BUT **5V** TO THE DC CONNECTOR OR THE BOARD WILL BE DAMAGED!

It should also be noted that if an OTG configuration is used, for example tying two BeagleBoards together via a UBS OTG cable, both of the BeagleBoards must be powered by the DC supply. If the OTG port is used as a Host port, then the DC supply must also be used.

Figure 19 is the design of the power input section.
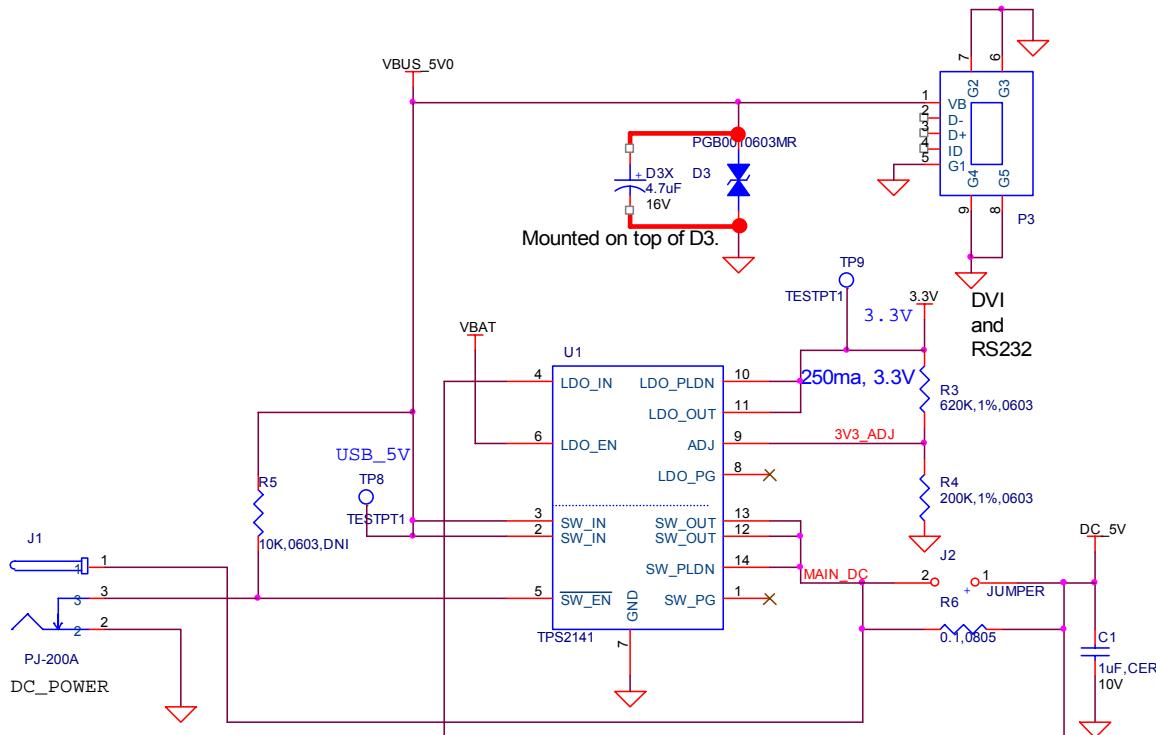
Figure 19.   Input Power Section

8.2.1   USB DC Source

The USB specification requires that the current consumed prior to enumeration be limited to 100mA @ 5V (500mW). The 5V DC from the USB is routed through the **TPS214**1 switch to insure that this requirement is met as uncharged capacitors on the BeagleBoard can exhibit a large current drain during start up that could exceed this requirement. The **TPS2141** is a USB 2.0 Specification-compatible IC containing a dual-current limiting power switch and an adjustable low dropout regulator (LDO). Both the switch and LDO limit inrush current by controlling the turn on slew rate. The dual-current-limiting feature of the switch allows USB peripherals to utilize high-value capacitance at the output of the switch, while keeping the inrush current low.

During turn on, the switch limits the current delivered to the capacitive load to less than 100 mA. When the output voltage from the switch reaches about 93% of the input voltage, the switch power good output goes high, and the switch current limit increases to 800mA (minimum), at which point higher current loads can be turned on. The higher current limit provides short circuit protection while allowing the peripheral to draw maximum current from the USB bus.

When in the USB powered mode and no DC supply is connected, the **TPS2141** is enabled, allowing the power to be supplied to the board through the integrated switch inside the **TPS2141**.

New on the REV B4 and B5 design is the addition of a 4.7uf capacitor across **D3.** This is provided to insure that the voltage supplied by the **TWL4030** is clean when in the host mode. There have been some cases where the hub would not connect due to excessive noise on the supply.

8.2.2   Wall Supply Source

A wall supply can be used to provide power to the board. A regulated 5V DC supply of at least 500mA is required. It needs to have a 2.1mm plug with a center hot configuration.

> **WARNING:** DO NOT PLUG IN ANYTHING BUT **5V** TO THE DC CONNECTOR OR THE BOARD WILL BE DAMAGED!

In the event that a higher DC load is required due to the addition of a Daughtercard a higher current supply can be used. The maximum current should not exceed 2A.

8.2.3   DC Source Control

Unlike when powering from the USB OTG port, in the case of the DC voltage, the current limiting is not required. As long as the DC supply is not connected, the switch for the USB is enabled. When the DC supply is plugged in, the switch is disabled because the

ground is removed from **pin 5** of the **TPS2141**. This insures that the 5V from the USB is not connected by disabling the internal FET. In the case where there is no USB plugged in, there is no 5V available to be routed so the removal of the pullup in **pin 5** has no affect.

When in the DC mode of operation, the USB OTG can be used in the Host or Client modes. The **TWL4030** will be responsible for handling the supply of the **VBUS_5V0** rail in the OTG or Host modes. As this is limited to 100mA, a powered hub must be used to support peripherals on the OTG port.

8.2.4   3.3V Supply

The **TPS2141** has an integrated 3.3V LDO which is being used to supply the **3.3V** as required on the BeagleBoard for the **DVI-D** interface and the **UART**. The input to the LDO is supplied by the main **DC_5V**. This insures that the power to the LDO can be supplied by either the USB or the DC wall supply and that the current measurement includes the 3.3V supply.
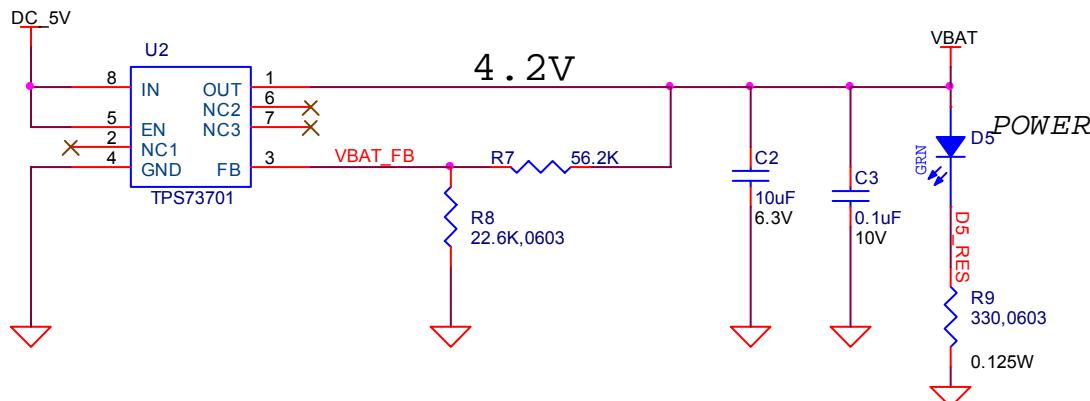
8.2.5   Current Measurement

Jumper **J2** is a set of pads that can facilitate the installation of a .1 x .1 header. This allows for the voltage drop across the resistor to be measured, providing a way to measure the current consumption of the BeagleBoard from the main voltage rails, either USB or DC. The resistor, **R6**, is a .1 ohm resistor across which the voltage is measured. The reading you get is .1mV per mA of current.

8.3   Power Conditioning

This circuitry regulates the DC input to a nominal 4.2VDC level. This is required in order to meet the maximum DC voltage level as specified by the TWL4030 Power Management device which is 4.7V. Using 4.2V gives us some margin and meets the nominal 4.2V rating of the TWL4030.

Figure 20 is the power conditioning section of the BeagleBoard.

Figure 20. Power Conditioning

The **TWL4030** provides the main power rails to the board and has a maximum limit of 4.8V on its VBAT input and a nominal of 4.2V. **U2**, the **TPS73701**, is used to convert the DC_5V, which can come from a DC wall supply or the USB, to 4.2V to meet this requirement. The **TPS737701** is a linear low-dropout (LDO) voltage regulator and is thermal shutdown and current limit protected. It has the ability to deliver 1A of current, although this is far and above the requirements of the board. By adjusting the values of **R7** and **R8**, the actual voltage can be adjusted if needed. The LED **D5** is an indication that the 4.2V is present.

8.4   TWL4030 Reset and Power Management

The **TWL4030** supplies several key functions on the BeagleBoard. This section covers a portion of those functions centered on the power and reset functions. Included in this section is:

- o   Main Core Voltages
- o   Peripheral Voltages
- o   Power Sequencing
- o   Reset

The other functions are covered in other sections in this document and are grouped by their overall board functions. The explanation of the various regulators found on the TWL4030 is based upon how they are used in the board design and are not intended to reflect the overall capability of the TWL4030 device. Please refer to the TWL4030 documents for a full explanation of the device operation.

8.4.1   Main Core Voltages

The **TWL4030** supplies the three main voltage rails for the **OMAP3530** processor and the board:

- o   VOCORE_1V3 (1.2V)
- o   VDD2　　　　　(1.3V)
- o   VIO_1V8　　　　(1.8V)

The **VOCORE_1V3** defaults to **1.2V** at power up, but can be adjusted by software to the **1.3V** level. **Figure 21** is the interfacing of the TWL4030 to the system as it provides the three main rails.
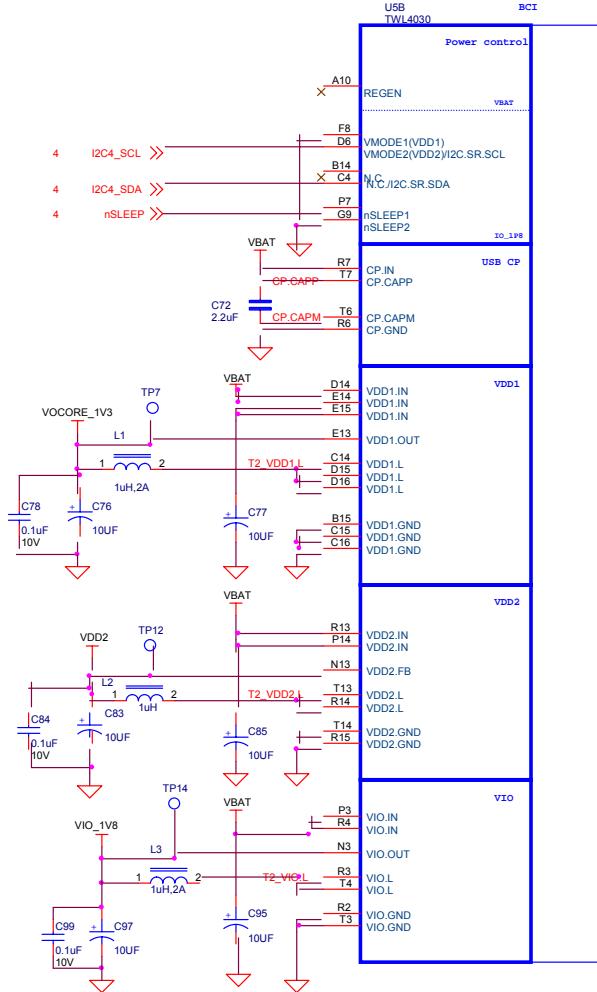
Figure 21.   Main Core Voltages

8.4.2   Main DC Input

The main supply to the **TWL4030** for the main rails is the **VBAT** rail which is a nominal 4.2V. Each rail has a filter cap of **10uF** connected to each of the three inputs. A **.1uF** cap is also provided for high frequency noise filtering.

8.4.3   OMAP3530 I2C Control

The various components in the TWL4030 are controlled from the OMAP3530 via the I2C interface. I2C_0 is used to control the TWL4030 device.

8.4.4   Smart Reflex

VDD1 and **VDD2** regulators on the **TWL4030** provide SmartReflex-compliant voltage management. The SmartReflex controller in the **OMAP3530** interfaces with the **TWL4030** counterpart through the use of a dedicated **I2C** bus. The **OMAP3530** computes the required voltage and informs the **TWL4030** using the SmartReflex I2C interface.

SmartReflex control of the **VDD1** and **VDD**2 regulators can be enabled by setting the SMARTREFLEX_ENABLE bit (DCDC_GLOBAL_CFG[3]) to 1. To perform **VDD1** voltage control through the SmartReflex interface, the TWL4030 provides the VDD1_SR_CONTROL register. The MODE field of the VDD1_SR_CONTROL register can be set to 0 to put VDD1 in an ACTIVE state; setting the field to 1 moves **VDD1** to a SLEEP state. **VDD1** output voltage can be programmed by setting the VSEL field of the VDD1_SR_ CONTROL register. The **VDD1** output voltage is given by VSEL*12.5 mV + 600 mV.

8.4.5   VOCORE_1V3

The **VOCORE_1V3** rail is supplied by the **VDD1** regulator of the **TWL4030**. The **VDD1** regulator is a 1.1A stepdown power converter with configurable output voltage between 0.6 V and 1.45 V in steps of 12.5 mV. This regulator is used to power the **OMAP3530** core.

The **OMAP3530** can request the **TWL4030** to scale the **VDD1** output voltage to reduce power consumption. The default output voltage at power-up depends on the boot mode settings, which in the case of the BeagleBoard is 1.2V. The output voltage of the **VDD1** regulator can be scaled by software or hardware by setting the ENABLE_VMODE bit (VDD1_VMODE_CFG[0]). In each of these modes, the output voltage ramp can be single-step or multiple-step, depending on the value of the STEP_REG field of the VDD1_STEP[4:0] register. The VOCORE_1V3 rail should be set to 1.3V after boot up.

Apart from these modes, the **VDD1** output voltage can also be controlled by the **OMAP3530** through the SmartReflex I2C interface between the OMAP3530 and the **TWL4030**. The default voltage scaling method selected at reset is a software-controlled mode. Regardless of the mode used, **VDD1** can be configured to the same output voltage in sleep mode as in active mode by programming the DCDC_SLP bit of the VDD1_VMODE_CFG[2] register to 0. When the DCDC_SLP bit is 1, the sleep mode

output voltage of **VDD1** equals the floor voltage that corresponds to the VFLOOR field (VDD1_VFLOOR[6:0]).

8.4.6 VDD2

The **VDD2** voltage rail is generated by the **TWL4030** using the **VDD2** regulator. The **VDD2** regulator is a stepdown converter with a configurable output voltage of between 0.6 V and 1.45 V and is used to power the OMAP3530 core. **VDD2** differs from **VDD1** in its current load capabilities with an output current rating of 600 mA in active mode.

The **VDD2** provides different voltage regulation schemes. When **VDD2** is controlled by the VMODE2 signal or with the SmartReflex interface, the range of output voltage is 0.6 V to 1.45 V. The use of the VMODE2 signal and the VDD2_VMODE_CFG, VDD2_STEP, VDD2_FLOOR, and VDD2_ROOF registers is similar to the use of the corresponding signals and registers for **VDD1**. **VDD2** shares the same SmartReflex I2C bus to provide voltage regulation. The VDD2_SR_CONTROL register is provided for controlling the **VDD2** output voltage in SmartReflex mode.

When the **VDD2** is used in software-control mode, the VSEL (VDD2_ DEDICATED[4:0]) field can be programmed to provide output voltages of between 0.6 V and 1.45 V. The output voltage for a given value of the VSEL field is given by VSEL*12.5 mV + 600 mV. If the VSEL field is programmed so that the output voltage computes to more than 1.45 V, the TWL4030 sets the **VDD2** output voltage to 1.5 V.

8.4.7 VIO_1V8

The **VIO_1V8** rail is generated by the TWL4030 VIO regulator. The VIO output is a stepdown converter with a choice of two output voltage settings: 1.8 V or 1.85 V. The voltage is set by configuring the VSEL bit (VIO_VSEL[0]). When the VSEL bit is set to 0, the output voltage is 1.8 V, and when it is set to 1, the output voltage is 1.85 V.

When the **TWL4030** resets, the default value of this LDO is 1.80 V; the OMAP3530 must write 1 to the VSEL field to change the output to 1.85 V. The default for the BeagleBoard is 1.8V. This regulator output is used to supply power to the system memories and I/O ports. It is one of the first power supplies to be switched on in the power-up sequence. VIO does not support the SmartReflex voltage control schemes. VIO can be put into sleep or off mode by configuring the SLEEP_STATE and OFF_STATE fields of the VIO_REMAP register.

8.5 Peripheral Voltages

There are five additional voltages used by the system that are generated by the **TWL4030**. These are:

- o VDD_PLL2
- o VDD_PLL1
- o VDAC_1V8
- o VDD_SIM
- o VMMC1

Figure 22 shows the peripheral voltages supplied by the **TWL4030**.
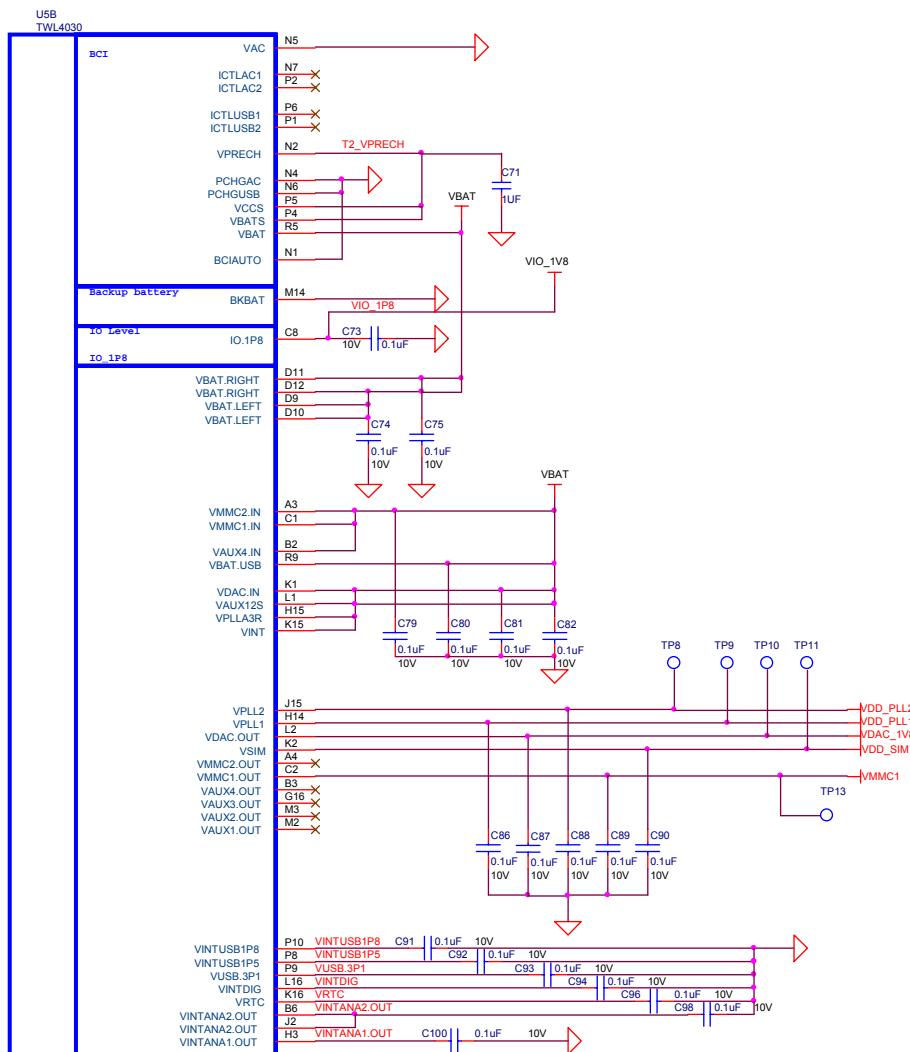
Figure 22.   Peripheral Voltages

8.5.1   VDD_PLL2

This programmable LDO is used to power the OMAP3530 PLL circuitry. The **VPLL2**
LDO can be configured through the I2C interface to provide output voltage levels of 1.0
V, 1.2 V, 1.3 V, or 1.8 V, based on the value of the VSEL field
(VPLLI_DEDICATED[3:0]). On the board this rail is used to power DVI output for pins

DSS_DATA(0:5), DSS_DATA(10:15) and DSS_DATA(22:23). The VPLL2 must be set to 1.8V for proper operation of the **DVI-D** interface.

8.5.2 VDD_PLL1

The VPLL1 programmable LDO regulator is low-noise, linear regulator used for the OMAP3530 PLL supply. The VDD_PLL1 rail is initialized to 1.8V.

8.5.3 VDAC_1V8

The **VDAC** programmable LDO regulator is a high-PSRR, low-noise, linear regulator that powers the OMAP3530 dual-video DAC. It is controllable with registers via I2C and can be powered down if needed. The **VDAC** LDO can be configured to provide 1.2V, 1.3 V, or 1.8 V in on power mode, based on the value of the VSEL field (VDAC_DEDICATED[3:0]). The **VDAC_1V8** rail should be set to 1.8V for the BeagleBoard.

8.5.4 VDD_SIM

This voltage regulator is a programmable, low dropout, linear voltage regulator supplying the bottom 4 bits of the 8 bit **SD/MMC** card slot. The VSEL field (VSIM_DEDICATED[3:0]) can be programmed to provide output voltage of 1.0 V, 1.2 V, 1.3 V, 1.8 V, 2.8 V, or 3.0 V and can deliver up to 50mA. The default output voltage of this LDO as directed by the **TWL4030** boot pins is 1.8V.

8.5.5 VMMC1

The **VMMC1** LDO regulator is a programmable linear voltage converter that powers the MMC1 slot snd includes a discharge resistor and overcurrent protection (short-circuit). This LDO regulator can also be turned off automatically when the MMC card extraction is detected. The VMMC1 LDO is powered from the main **VBAT** rail. The **VMMC1** rail defaults to 3.0V as directed by the **TWL4030** boot pins and will deliver up to 220mA. It can be set to 3.0V in the event 3V cards are being used.

8.5.6 Boot Configuration

The boot configuration pins on the **TWL4030** determine the power sequence of the device. For the OMAP3530 support, the boot pin configuration is fixed at:

- o **BOOT0** tied to VBAT
- o **BOOT1** tied to Ground.

8.5.7   Power Sequencing

Based on the boot configuration pins, the **TWL4030** knows the type of OMAP processor that it needs to support, in this case the OMAP3530. The voltages are ramped in a sequence that is compatible with the OMAP3530 processor. **Figure 23** is the sequence that the power rails, clocks, and reset signal come up.
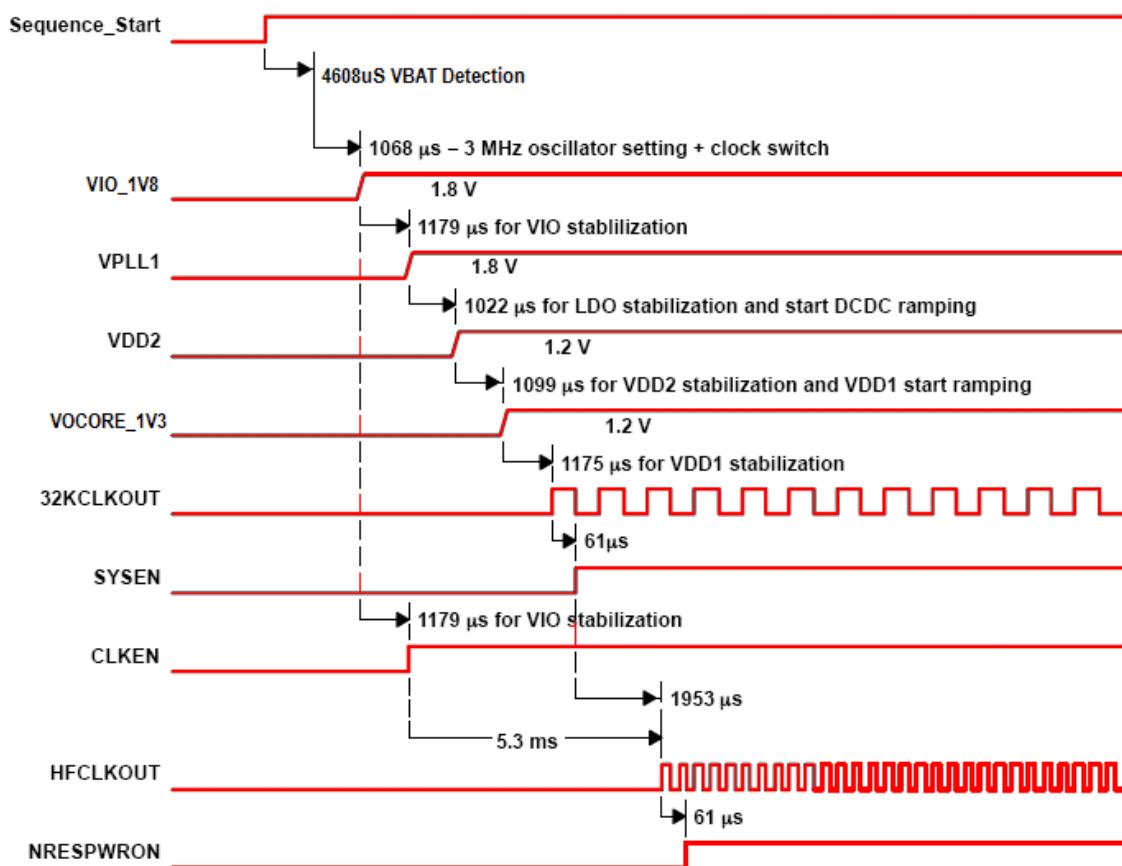
Figure 23.   Power Sequencing

8.5.8 Reset Signals

The BeagleBoard uses two distinct reset circuits:

- o Warm Reset
- o Cold Reset

Figure 24 shows the connections for the Warm and Cold Reset.
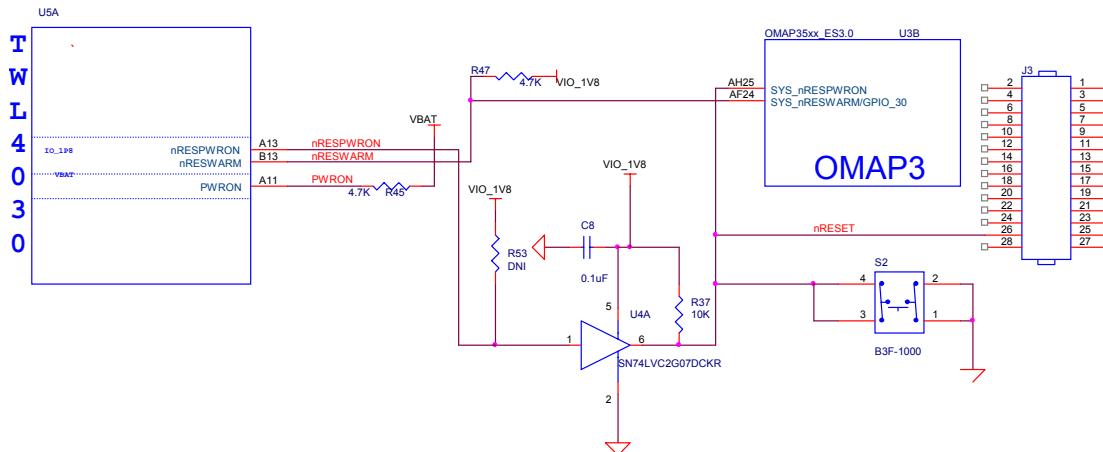
Figure 24. Reset Circuitry

8.5.8.1 *Warm Reset*

The warm reset is generated by the **OMAP3530** processor on power up. The **nRESWARM** signal is a bidirectional reset. When an internal reset occurs, **nRESWARM** goes low and resets all the peripherals and the **TWL4030**. The **TWL4030** can be configured to perform a warm reset of the device to bring it into a known defined state by detecting a request for a warm reset on the **NRESWARM** pin. The minimum duration of the pulse on the **nRESWARM** pin should be two 32-kHz clock cycles. The **nRESWARM** output is open-drain; consequently, an external pullup resistor is required. There is no way for the user to generate a warm reset on the BeagleBoard.

8.5.8.2 *Cold Reset*

On power up as shown in **Figure 24**, the **TWL4030** generates **nRESPWRON**, power on reset. The signal from the **TWL4030** is an output only and is not an open drain signal. By running the signal through a buffer, **SN74LVC2G07**, the signal becomes open drain, which requires a pullup on the signal. This will allow the **nRESPWRON** signal to be pulled low, by pressing the reset switch **S2**, to force a reset to the **OMAP3530** processor and to any device on the expansion card that require a reset.

It also allows for the reset signal to be pulled low or held low for an extended time by circuitry on the expansion card if needed.

8.5.8.3   *PWRON*

You will notice another signal on the **TWL4030** called **PWRON**. This signal is referenced in the **TWL4030** documentation. In the BeagleBoard design it is not used but it is pulled high to insure the desired operation is maintained.

8.6   OMAP3530 Processor

The heart of BeagleBoard is the OMAP3530 processor. **Figure 25** is a high level block diagram of the OMAP3530.
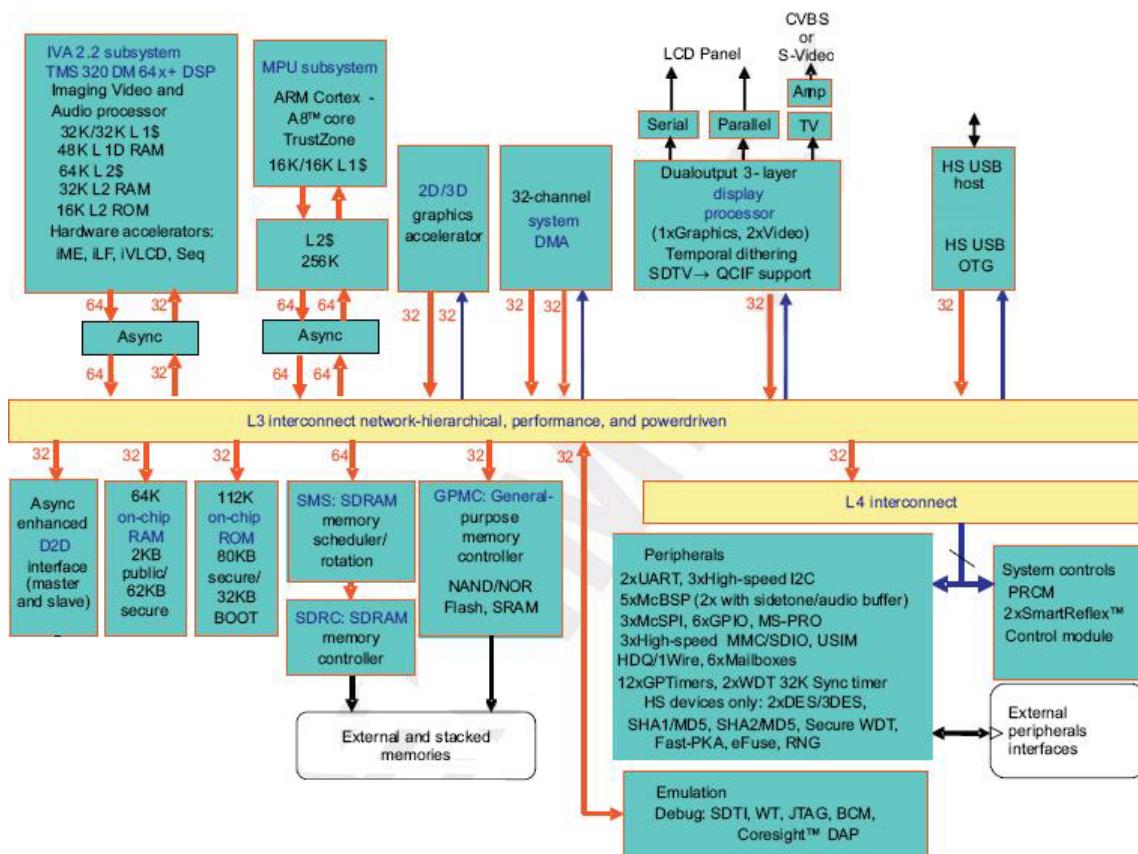
Figure 25.   OMAP3530 Block Diagram

8.6.1   Overview

The OMAP3530 high-performance, multimedia application device is based on the enhanced OMAP™ 3 architecture and is integrated on TI's advanced 65-nm process technology. The OMAP3530 architecture is configured with different sets of features in different tier devices. Some features are not available in the lower-tier devices. For more information, refer to the OMAP3530 Technical Reference Manual (TRM).The architecture is designed to provide best-in-class video, image, and graphics processing sufficient to various applications.

The OMAP3530 supports high-level operating systems (OSs), such as:
   o   Windows CE
   o   Linux
   o   Others

This OMAP3530 device includes state-of-the-art power-management techniques required for high-performance low power products.
The OMAP3530 supports the following functions and interfaces on the BeagleBoard:

   o   Microprocessor unit (MPU) subsystem based on the ARM Cortex-A8™ microprocessor
   o   POP Memory interface
        o   1Gb MDDR (128Mbytes)
        o   2Gb NAND Flash (256 Mbytes)
   o   24 Bit RGB Display interface (DSS)
   o   SD/MMC interface (2)
   o   USB OTG interface
   o   NTSC/PAL/S-Video output
   o   Power management
   o   Serial interface
   o   I^{2}C interface
   o   I^{2}S Audio interface (McBSP2)
   o   Expansion McBSP1
   o   JTAG debugging interface

8.6.2   SDRAM Bus

The SDRAM bus is not accessible on the BeagleBoard. Its connectivity is limited to the POP memory access on the top of the OMAP3530 and therefore is only accessible by the SDRAM memory.

The base address for the DDR SDRAM in the POP device is **0x8000 0000**.

8.6.3   GPMC Bus

The GPMC bus is not accessible on the BeagleBoard. Its connectivity is limited to the POP memory access on the top of the OMAP3530 and therefore is only accessible by the NAND memory.

The memory on the GPMC bus is NAND and therefore will support the classical NAND interface. The address of the memory space is programmable. Please consult the Software Reference for more information.

8.6.4   DSS Bus

The display subsystem provides the logic to display a video frame from the memory frame buffer in either SDRAM on a liquid-crystal display (LCD) panel via the DVI-D interface. The DSS is configured in the 24 bit mode.

8.6.5   McBSP2

The multi-channel buffered serial port (McBSP) McBSP2 provides a full-duplex direct serial interface between the OMAP3530 and the audio CODEC in the **TWL4030** using the I2S format. Only four signals are supported on the McBSP2 port. **Figure 26** is a depiction of McBSP2.
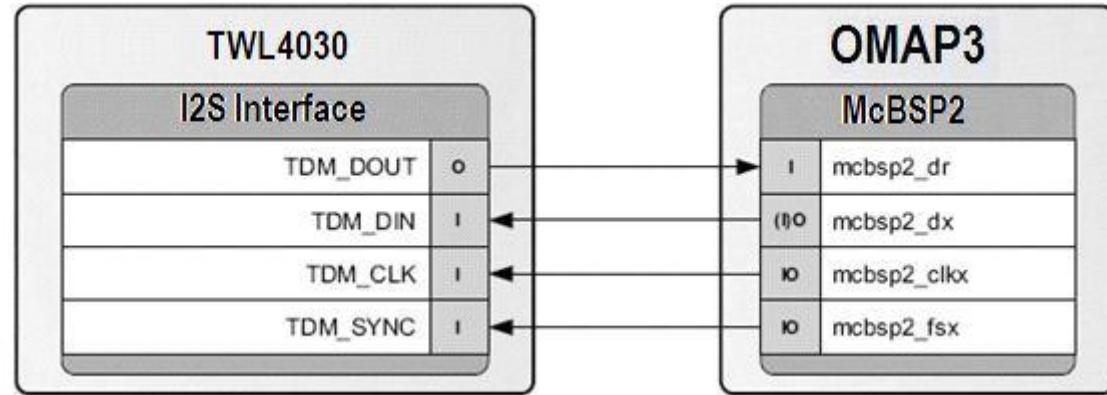
Figure 26.   McBSP2 Interface

8.6.6   McBSP1

McBSP1 provides a full-duplex direct serial interface between the OMAP3530 and the expansion interface. There are 6 signals supported on McBSP1, unlike the 4 signals on the other ports. **Figure 27** is a diagram of McBSP1.
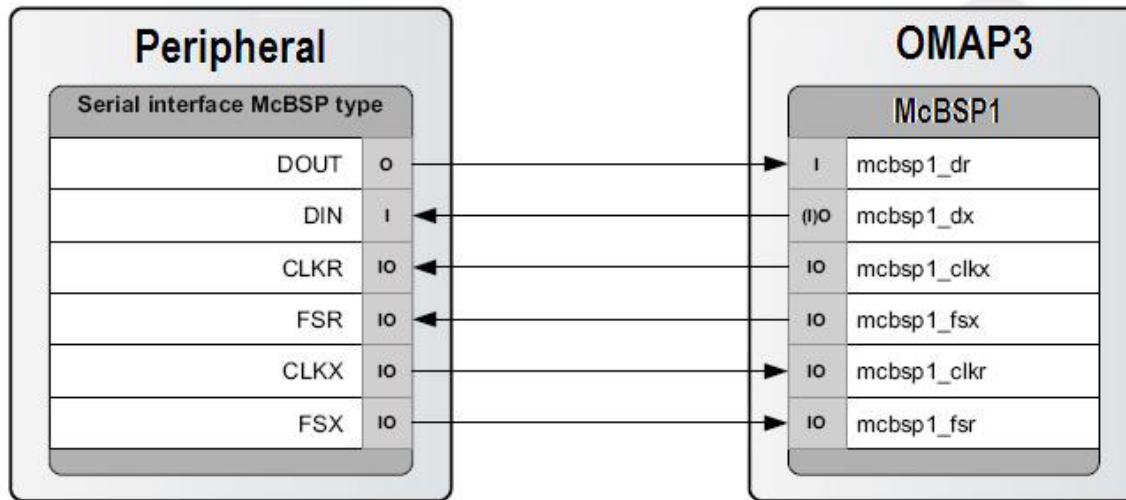
Figure 27.   McBSP1 Interface

8.6.7   McBSP3

McBSP3 provides a full-duplex direct serial interface between the OMAP3530 and the expansion interface. **Figure 28** is a diagram of McBSP3.
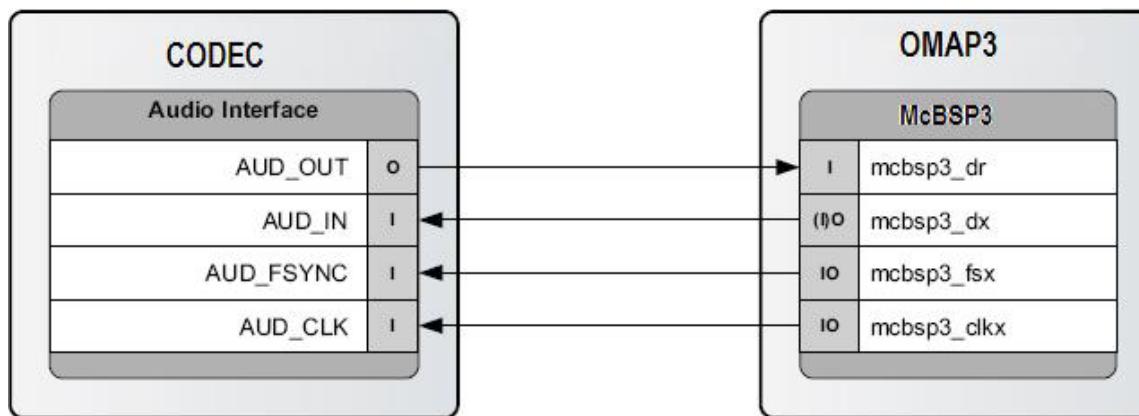
Figure 28.   McBSP3 Interface

8.6.8   Pin Muxing

On the OMAP3530, the majority of pins have multiple configurations that the pin can be set to. In essence, the pin can become different signals depending on how they are set in the software. In order for the BeagleBoard to operate, the pins used must be set to the correct signal. In some cases, the default signal is the correct signal. Each pin can have a

maximum of 7 options on the pin. This is called the pin mode and is indicated by a three bit values (0:3).

In the case of the signals going to the expansion connector, the settings required for those pins depends on how they are to be used. For an explanation of the options, please refer to the Expansion Header section.

Table 4 is a list of all of the signals used on the OMAP3530 for the BeagleBoard and the required mode setting for each pin. Where the default setting is needed, it will be indicated. The USER notation under mode indicates that this is an expansion signal and can be set at the discretion of the user. A FIXED indicates that there is only one function for that signal and that it cannot be changed,

Table 4.   OMAP3530 Pin Muxing Settings

Signal	Mode
DSS	Default
MMC1	Default
MMC2	User
UART3	Default
GPMC	Default
UART1	Default
I2C1	Default
I2C2	Default
I2C3	Default
I2C4	Default
JTAG	FIXED
TV_OUT	Default
SYS_nRESPWRON	Default
SYS_nRESWARM	Default
SYS_nIRQ	Default
SYS_OFF	Default
SYS_CLKOUT	Default
SYS_CLKOUT2	Default
SYS_CLKREQ	Default
SYS_XTALIN	FIXED
GPIO_149	4
GPIO_150	4
McBSP1	Default
McBSP2	User
McBSP3	Default

8.6.9   GPIO Mapping

There are a number of GPIO that are used on the BeagleBoard design. **Table 5** shows which of these GPIO pins are used in the design and whether they are inputs or outputs. While GPIO pins can be used as interrupts, the table only covers the GPIO pin mode. If it is an interrupt, then it is covered in the interrupt section.

![beagleboard.org logo]

Table 5.   OMAP3530 GPIO Pins

OMAP PIN	INT/GPIO	I/O	Signal	USAGE
AA9	GPIO_149	O	LED_GPIO149	Controls User LED0
W8	GPIO_150	O	LED_GPIO149	Controls User LED1
AH8	GPIO_29	I	MMC1_WP	SD/MMC card slot Write protect
J25	GPIO_170	O	DVI_PUP	Controls the DVI-D interface. A Hi = DVI-D enabled.
AE21	GPIO_7	I	SYSBOOT_5	Used to put the device in the boot mode or as a user button input

Other signals, such as those that connect to the expansion connector, may also be set as a GPIO pin. For information on those, refer to the Expansion Connector section.

8.6.10   Interrupt Mapping

There are a small number of pins on the OMAP3530 that act as interrupt. Some of these interrupts are connected to the TWL4030 and their status is reflected through the main TWL4030 interrupt. **Table 6** lists the interrupts.

Table 6.   OMAP3530 Interrupt Pins

TWL4030 Pin	OMAP PIN	INT/GPIO	USAGE
	AF26	SYS_nIRQ	Interrupt from the TWL4030
	AH8	GPIO_29	SD Write protect lead. Can be polled or set to an interrupt.
P12		GPIO0	MMC1 card detect input. Goes to the OMAP3530 over the SYS_nIRQ pin.

8.7   POP Memory Device

The OMAP3530 uses what is called POP (Package-on-Package) memory. The memory is a MCP (Multi Chip Package) that contains both the Mobile DDR SDRAM and the NAND Flash. **Figure 29** shows the POP Memory concept.

Figure 29.   POP Memory

The Memory device mounts on top of the OMAP3530 device. The configuration used on the board is a 2Gb NAND Flash plus 1Gb MDDR SDRAM device from Micron.

8.8   System Clocks

There are three clocks needed for the operation of the BeagleBoard, 32KHz, 26MHz and McBSP_CLKS. **Figure 30** shows the components that make up the System Clocks.
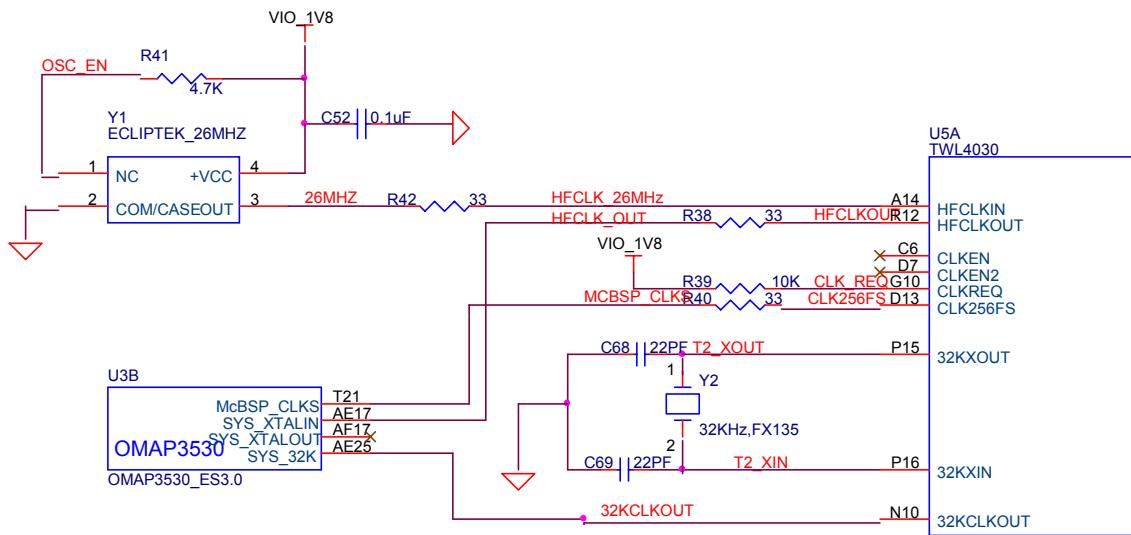
Figure 30.   System Clocks

8.8.1   32KHz Clock

The 32KHz clock is needed for the TWL4030 and the **OMAP3530** and is provided by the **TWL4030** via the external 32KHz crystal, **Y2**. The **TWL4030** has a separate output from the crystal to drive the OMAP3530 that buffers the resulting 32-kHz signal and provides it as **32KCLKOUT**, which is provided to the OMAP3530 on ball **AE25**. The default mode of the 3**2KCLKOUT** signal is active, but it can be disabled if desired under SW control.

The 32.768-kHz clock drives the RTC embedded in the **TWL4030**. The RTC is not enabled by default; the host processor must set the correct date and time to enable the RTC.

8.8.2   26MHz Clock

This section describes the 26MHz clock section of the BeagleBoard.

![beagleboard.org]

8.8.2.1   26MHz Source

The **26MHz** clock is provided by an onboard oscillator, **Y1**. The **TWL4030** receives the external **HFCLKIN** signal on ball **A14** and uses it to synchronize or generate the clocks required to operate the TWL4030 subsystems. The **TWL4030** must have this clock in order to function to the point where it can power up the BeagleBoard. This is the reason the **26MHz** clock is routed through the TWL4030.

8.8.2.2   TWL4030 Setup

When the TWL4030 enters an active state, the OMAP3530 must immediately indicate the **HFCLKIN** frequency (26 MHz) by setting the HFCLK_FREQ bit field (bits [1:0]) in the CFG_BOOT register of the TWL4030. HFCLK_FREQ has a default of not programmed, and in that condition, the USB subsection does not work, the three DCDC switching supplies (VIO, VDD1, and VDD2) operate from their free-running 3-MHz (RC) oscillators, and the PWR registers are accessed at a default 1.5-M byte. HFCLK_FREQ must be set by the OMAP3530 during the initial power-up sequence. On BeagleBoard, this is done by the internal boot ROM on startup.

8.8.2.3   OMAP3530 26MHz

The 26MHz clock for the **OMAP3530** is provided by the TWL4030 on ball **R12** through **R38**, a 33 ohm resistor is providing to minimize any reflections on the clock line. The clock signal enters via ball **AE17** on the **OMAP3530**.

8.8.3   McBSP_CLKS

An additional clock is also provided by the **TWL4030** called **McBSP_CL**KS. This clock is provided to the OMAP3530 in order to insure synchronization of the I2S interface between the **OMAP3530** and the **TWL4030**.

8.9   USB OTG Port

The main USB port on the BeagleBoard is a USB OTG (On-the-Go) port. It can be used as an OTG port or Client port. The main use is as a client port, as that is the mode that will supply the power needed to power the BeagleBoard.

> ## NOTE: In order to use the OTG or the Host mode, the BeagleBoard must be powered from the DC supply.

8.9.1   USB OTG Overview

USB OTG is a supplement to the USB 2.0 specification. The standard USB uses a master/slave architecture, a USB host acting as a master and a USB peripheral acting as a slave. Only the USB host can schedule the configuration and data transfers over the link.

The USB peripherals cannot initiate data transfers, they only respond to instructions given by a host.

USB OTG works differently in that gadgets don't need to be pure peripherals because they can sometimes act as hosts. An example might be connecting a USB keyboard or printer to BeagleBoard or a USB printer that knows how to grab documents from certain peripherals and print them. The USB OTG compatible devices are able to initiate the session, control the connection and exchange Host/Peripheral roles between each other.

The USB OTG supplement does not prevent the use of a hub, but it describes role swapping only in the case of a one-to-one connection where two OTG devices are directly connected. If a standard hub is used, the supplement notes that using it will lead to losing USB OTG role-swap capabilities making one device as the Default-Host and the other as the Default-Peripheral until the hub is disconnected.

The combination of the **OMAP3530** and the **TWL4030** allows the BeagleBoard to work as an OTG device if desired. The primary mode of operation however, is intended to be a client mode in order to pull power from the USB host which is typically a PC. As the Rev B does not have a Host USB port, this port will be used as a Host port in many applications.

8.9.2   USB OTG Design

Figure 31 is the design of the USB OTG port on the BeagleBoard.
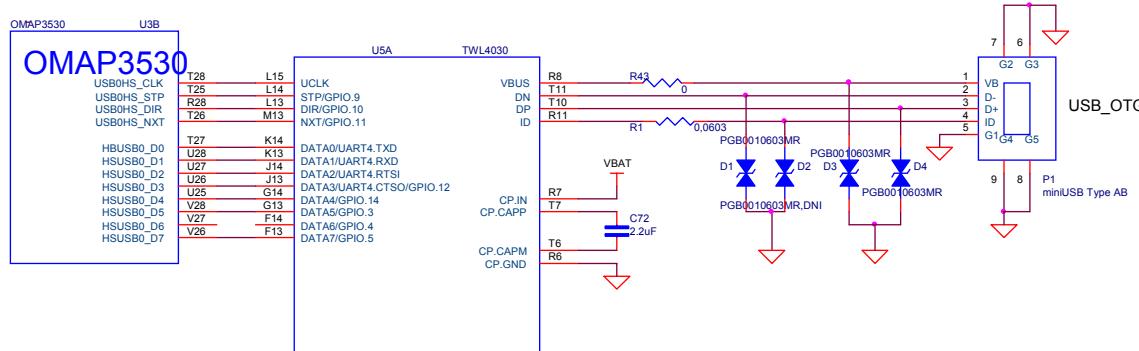
Figure 31.   USB Client Design

8.9.3   OTG ULPI Interface

ULPI is an interface standard for high-speed USB 2.0 systems. It defines an interface between USB link controller (OMAP3530) and the TWL4030 that drives the actual bus. ULPI stands for UTMI+ low pin interface and is designed specifically to reduce the pin

count of discrete high-speed USB PHYs. Pin count reductions minimize the cost and footprint of the PHY chip on the PCB and reduce the number of pins dedicated to USB for the link controller.
.

Unlike full- and low-speed USB systems, which utilize serial interfaces, high-speed requires a parallel interface between the controller and PHY in order to run the bus at 480Mbps. This leads to a corresponding increase in complexity and pin count. The ULPI used on the BeagleBoard keeps this down to only 12 signals because it combines just three control signals, plus clock, with a 8-bit bi-directional data bus. This bus is also used for the USB packet transmission and for accessing register data in the ULPI PHY.

8.9.3.1   OMAP3530 Interface

The controller for the ULPI interface is the OMAP3530. It provides all of the required signals to drive the interface. **Table 7** describes the signals from the OMAP3530 that are used for the USB OTG interface.

Table 7.   OMAP3530 ULPI Interface

Signal	Description	Type	Ball
hsusb0_clk	Dedicated for external transceiver 60-MHz clock input from PHY	I	T28
hsusb0_stp	Dedicated for external transceiver Stop signal	O	T25
hsusb0_dir	Dedicated for external transceiver Data direction control from PHY	I	R28
hsusb0_nxt	Dedicated for external transceiver Next signal from PHY	I	T26
hsusb0_data0	Transceiver Bidirectional data bus	I/O	T27
hsusb0_data1	Transceiver Bidirectional data bus	I/O	U28
hsusb0_data2	Transceiver Bidirectional data bus	I/O	U27
hsusb0_data3	Transceiver Bidirectional data bus	I/O	U26
hsusb0_data4	Transceiver Bidirectional data bus	I/O	U25
hsusb0_data5	Transceiver Bidirectional data bus	I/O	V28
hsusb0_data6	Transceiver Bidirectional data bus	I/O	V27
hsusb0_data7	Transceiver Bidirectional data bus	I/O	V26

8.9.3.2   TWL4030 Interface

The TWL4030 USB interfaces to the OAMP3 over the ULPI interface. **Table 8** is a list of the signals used on the TWL4030 for the ULPI interface.

Table 8.   OMAP3530 ULPI Interface

Signal	Description	Type	Ball
UCLK	High speed USB clock	I/O	L15
STP	High speed USB stop	I	L14
DIR	High speed USB dir	O	L13
NXT	High speed USB direction	O	M1

beagleboard.org

DATA0	High speed USB Data bit 0	I/O	K14
DATA1	High speed USB Data bit 0	I/O	K13
DATA2	High speed USB Data bit 0	I/O	J14
DATA3	High speed USB Data bit 0	I/O	J13
DATA4	High speed USB Data bit 0	I/O	G14
DATA5	High speed USB Data bit 0	I/O	G13
DATA6	High speed USB Data bit 0	I/O	F14
DATA7	High speed USB Data bit 0	I/O	F13

8.9.4   OTG Charge Pump

When the TWL4030 acts as an A-device, the USB charge pump is used to provide 4.8 V/100 mA to the VBUS pin. When the TWL4030 acts as a B-device, the USB charge pump is in high impedance. If used in the OTG mode as an A-device, the BeagleBoard will need to be powered from the DC supply. If acting as a B-device, there will not be a voltage source on the USB OTG port to drive the BeagleBoard. **Table 9** describes the charge pump pins.

Table 9.   USB OTG Charge Pump Pins

Signal	Description	Type	Ball
CP.IN	The charge pump input voltage. Connected to VBAT.	Power	R7
CP.CAPP	The charge pump flying capacitor plus.	O	L14
CP.CAPM	The charge pump flying capacitor minus.	O	T6
CP.GND	The charge pump ground.	GND	R6

The charge pump is powered by the **VBAT** voltage rail. The charge pump generates a 4.8-V (nominal) power supply voltage to the **VBUS** pin. The input voltage range is 2.7 V to 4.5 V so the 4.2V VBAT is within this range. The charge pump operating frequency is 1 MHz. The charge pump integrates a short-circuit current limitation at 450 mA.

8.9.5   OTG USB Connector

The OTG USB interface is accessed through the miniAB USB connector.

8.9.6   OTG USB Protection

Each lead on the USB port has ESD protection. In order or the interface to meet the USB 2.0 Specification Eye Diagram, these protection devices must be low capacitance.

8.10   SD/MMC

The board provides and SD/MMC interface for using cards such as MMC memory cards and SDIO cards, such as cameras and Wireless LAN.

The connector supports 7 different types of cards.
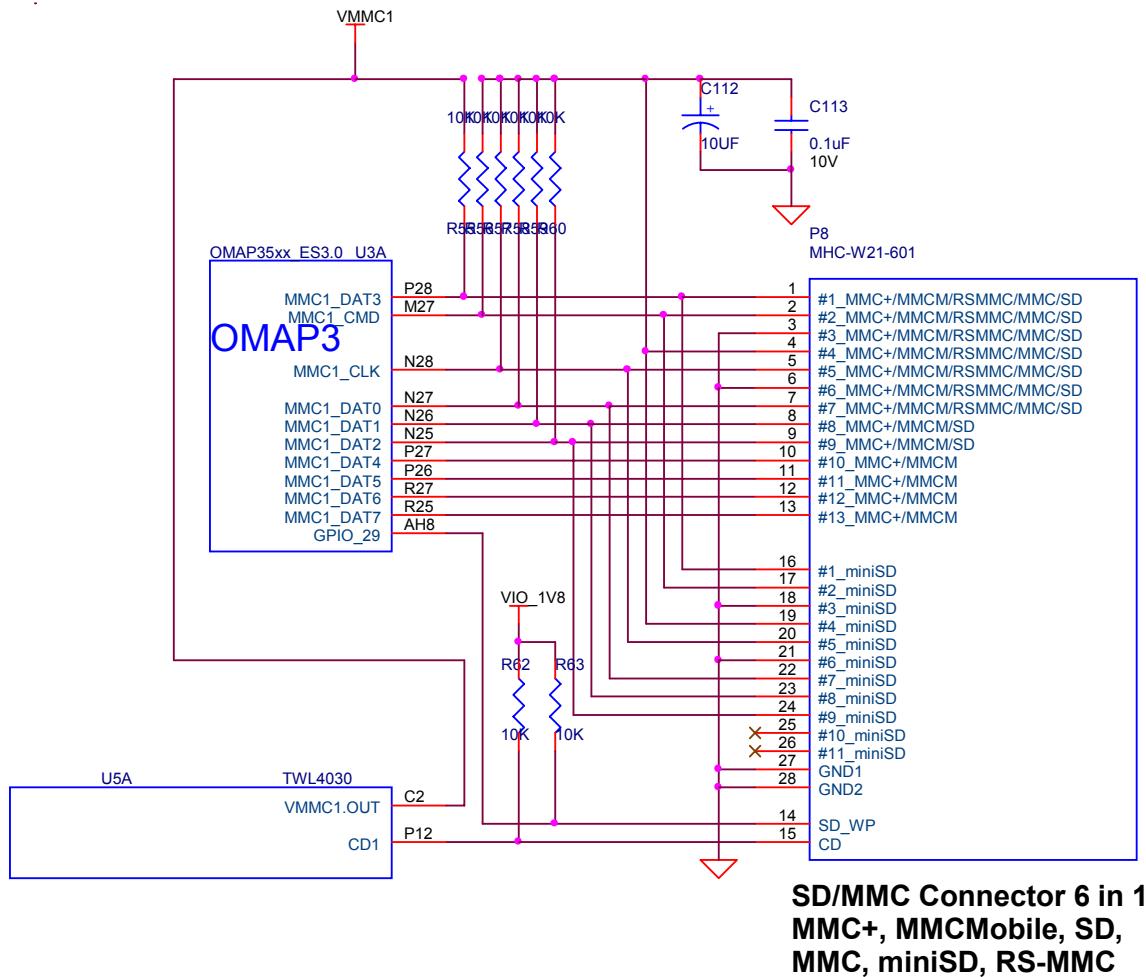
- o **SD-** Secure Digital (SD) is a flash memory card format developed by Matsushita, SanDisk and Toshiba for use in portable devices. As of 2007, SD card capacities range from 8 MB to 16 GB. Several companies have announced SD cards with 32 GB. Cards with 4-32 GB are considered high-capacity. The format has proven to be very popular. However, compatibility issues between older devices and the newer 4 GB and larger cards and the SDHC format have caused considerable confusion for some users. SD card have a write protect tab to prevent the data from being overwritten. SD supports 1-bit SD, 4-bit SD, and SPI modes.

- o **miniSD-** Has the same features as the SD with the exceptions that it is in a smaller size and the support for 4-bit mode is optional amongst suppliers.

- o **SDIO** - SDIO stands for Secure Digital Input Output. SD slots can actually be used for more than flash memory cards. Devices that support **SDIO** can use small devices designed for the SD form factor, like GPS receivers, Wi-Fi or Bluetooth adapters, modems, Ethernet adapters, barcode readers, IrDA adapters, FM radio tuners, TV tuners, RFID readers, digital cameras, or other mass storage media such as hard drives. SDIO cards are fully compatible with SD Memory Card host controller (including mechanical, electrical, power, signaling and software). When an SDIO card is inserted into a non SDIO-aware host, it will cause no physical damage or disruption to device or host controller. It should be noted that SPI bus topology is mandatory for SDIO, unlike SD Memory and most of the SD Memory commands are not supported in SDIO. **Figure 32** is an example of a SDIO camera card.

Figure 32.   Example of an SDIO Card

o **MMC**- The Multi Media Card (**MMC**) is a flash memory card standard. Unveiled in 1997 by Siemens AG and SanDisk, it is based on Toshiba's NAND-based flash memory, and is therefore much smaller than earlier systems based on Intel NOR-based memory such as CompactFlash. MMC is about the size of a postage stamp: 24 mm x 32 mm x 1.4 mm. MMC originally used a 1-bit serial interface, but newer versions of the specification allow transfers of 4 at a time. MMCs are currently available in sizes up to and including 4 GB an 8 GB models.

o **MMCplus**- The version 4.x of the MMC standard, introduced in 2005, brought in two very significant changes to compete against SD cards. These were support for running at higher speeds (26MHz, 52MHz) than the original MMC (20MHz) or SD (25MHz, 50MHz). Version 4.x cards are fully backward compatible with existing readers but require updated hardware/software to use their new capabilities; even though the 4 bit wide bus and high-speed modes of operation are deliberately electrically compatible with SD, the initialization protocol is different, so firmware/software updates are required to allow these features to be enabled when the card is used in an SD reader.

o **MMCmobile** – Is basically the same as MMCplus except that it supports 8 bit data mode.

o **RS-MMC** –This alternate form factor is known as Reduced-Size MultiMediaCard, or RS-MMC, and was introduced in 2004. This form factor is a smaller form factor, of about half the size: 24 mm × 18 mm × 1.4 mm. RS-MMCs are simply smaller MMCs. RS-MMCs are currently available in sizes up to and including 4 GB. Nokia used to use RS-MMC in the Nokia 770 Internet Tablet. **Figure 33** is a side by side comparison of the RS-MMC and MMC card.

Figure 33.   RS-MMC and Card

Figure 34 is the SD/MMC interface design on the BeagleBoard.

Figure 34.   SD/MMC Interface

8.10.1 MMC Power

The SD/MMC connector is supplied power from the **TWL4030** using the **VMMC1** rail. The default setting on this rail is 3.0V as set by the Boot ROM and under SW control, can be set to 1.80V for use with 1.8V cards. The maximum current this rail can provide is 220mA as determined by the TWL4030 regulator. Maximum current can be limited by the overall current available from the USB interface of the PC.

8.10.2 OMAP3530 Interface

There are no external buffers required for the SD/MC operation. The OMAP3530
provides all of the required interfaces for the SD/MMC interface.

The main features of the MMC/SD/SDIO host controller are:

o   Full compliance with MMC command/response sets as defined in the *Multimedia
     Card System Specification*, v4.0
o   Full compliance with SD command/response sets as defined in the *SD Memory
     Card Specifications*, v1.10d
o   Full compliance with SDIO command/response sets and interrupt/read-wait mode
     as defined in the *SDIO Card Specification, Part E1*, v1.10
o   Compliance with sets as defined in the *SD Card Specification, Part A2, SD Host
     Controller Standard Specification*, v1.00
o   Full compliance with MMC bus testing procedure as defined in the *Multimedia
     Card System Specification*, v4.0
o   Full compliance with CE-ATA command/response sets as defined in the *CE-ATA
     Standard Specification*
o   Full compliance with ATA for MMCA specification
o   Flexible architecture allowing support for new command structure
o   Support:
         –   1-bit or 4-bit transfer mode specifications for SD and SDIO cards
         –   1-bit, 4-bit, or 8-bit transfer mode specifications for MMC cards
o   Built-in 1024-byte buffer for read or write
o   32-bit-wide access bus to maximize bus throughput
o   Single interrupt line for multiple interrupt source events
o   Two slave DMA channels (1 for TX, 1 for RX)
o   Programmable clock generation
o   Support SDIO Read Wait and Suspend/Resume functions
o   Support Stop at block gap
o   Support command completion signal (CCS) and command completion signal
     disable (CCSD) management as specified in the *CE-ATA Standard Specification*

The known limitations are as follows:

o   No built-in hardware support for error correction codes (ECC). See the
     Multimedia Card System Specification, v4.0, and the *SD Memory Card
     Specifications*, v1.10d, for details about ECC.
o   The maximum block size defined in the *SD Memory Card Specifications*, v1.10d
     that the host driver can read and write to the buffer in the host controller is 2048
     bytes. MMC supports a maximum block size of 1024 bytes. Up to 512 byte
     transfers, the buffer in MMC is considered as a double buffering with ping-pong
     management; half of the buffer can be written while the other part is read. For 512
     to 1024 byte transfers, the entire buffer is dedicated to the transfer (read only or
     write only).

Table 10 provides a description of the signals on the MMC card.

Table 10.   SD/MMC OMAP Signals

Signal Name	Description	I/O	Pin
MMC1_CLK	SD/MMC Clock output.	O	N28
MMC1_CMD	SD/MMC Command pin	I/O	M27
MMC1_DAT(0..7)	SD/MMC Data pins	I/O	N27,N26,N25,P28,P27, P26,R27,R25
MMC_WP	Write Protect detect	I	AH8

8.10.3   Card Detect

When a card is inserted into the SD/MMC connector, the Card Detect pin is grounded. This is detected on pin **P12** of the **TWL4030**. An interrupt, if enabled, is sent to the **OMAP3530** via the interrupt pin. The SW can be written such that the system comes out of sleep or a reduced frequency mode when the card is detected.

8.10.4   Write Protect

If an SD card is inserted into the SD/MMC connector and the write protect pin is active, the Write Detect pin is grounded. This is detected **GPIO_29** of the OMAP3530. The SW can then determine if the card is write protected and act accordingly.

8.10.5   8 Bit Mode

The BeagleBoard also supports the new 8-bit cards. The upper 4 bits are supplied by the VDD_SIM power rail and as such the 8-bit mode is only supported in 1.8V modes. This requires that both the VMMC1 and VDD_SIM rails must be set to 1.8V when using 8 bit cards.

8.10.6   Booting From SD/MMC Cards

The ROM code supports booting from MMC and SD cards with some limitations:

o Support for MMC/SD cards compliant with the Multimedia Card System Specification v4.2 from the MMCA Technical Committee and the Secure Digital

I/O Card Specification v2.0 from the SD Association. Including high-capacity (size >2GB) cards: HC-SD and HC MMC.

o 3-V power supply, 3-V I/O voltage on port 1
o Initial 1-bit MMC mode, 4-bit SD mode.
o Clock frequency:
    − Identification mode: 400 kHz
    − Data transfer mode: 20 MHz
o Only one card connected to the bus
o FAT12/16/32 support, with or without master boot sector (MBR).

The high-speed MMC/SD/SDIO host controllers handle the physical layer while the ROM code handles the simplified logical protocol layer (read-only protocol). A limited range of commands is implemented in the ROM code. The MMC/SD specification defines two operating voltages for standard or high-speed cards. The ROM code only supports standard operating voltage range (3-V) (both modes supported). The ROM code reads out a booting file from the card file system and boots from it.

8.11   Audio Interface

The BeagleBoard supports stereo in and out through the **TWL4030** which provides the audio CODEC.

Figure 35 is the Audio circuitry design on the BeagleBoard.
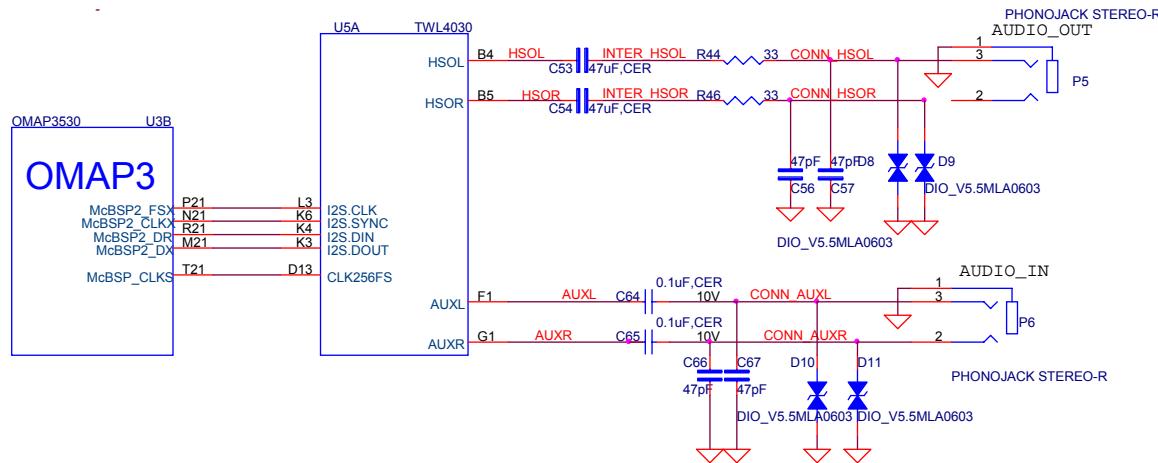
Figure 35.   Audio Circuitry

8.11.1 OMAP3530 Audio Interface

There are five McBSP modules called McBSP1 through McBSP5 on the OMAP3530. **McBSP2** provides a full-duplex, direct serial interface between CODEC inside the

TWL4030. It supports the I2S format to the TWL4030. In **Table 11** are the signals used on the OMAP3530 to interface to the CODEC.

Table 11.   OMAP3530 Audio Signals

Signal Name	Description	I/O	Pin
mcbsp2_dr	Received serial data	I	R21
mcbsp2_dx	Transmitted serial data	I/O	M21
mcbsp2_clkx	Combined serial clock	I/O	N21
mcbsp2_fsx	Combined frame synchronization	I/O	P21
Mcbsp_clks	External clock input. Used to synchronize with the TWL4030	I	T21

8.11.2 TWL4030 Audio Interface

The **TWL4030** acts as a master or a slave for the I2S interface. If the **TWL4030** is the master, it must provide the frame synchronization (I2S_SYNC) and bit clock (I2S_CLK) to the **OMAP3530**. If it is the slave, the **TWL4030** receives frame synchronization and bit clock. The TWL4030 supports the I2S, left-justified, and right-justified data formats, but doesn't support the TDM slave mode.

In **Table 12** are all the signals used to interface to the OMAP3530.

Table 12.   OMAP3530 Audio Signals

Signal Name	Description	I/O	Pin
I2S.CLK	Clock signal (audio port)	I/O	L3
I2S.SYNC	Synchronization signal (audio port)	IO	K6
I2S.DIN	Data receive (audio port)	I	K4
I2S. DOUT	Data transmit (audio port)	O	K3
CLK256FS	Synchronization frame sync to the OMAP3530	O	D13

8.11.3 Audio Output Jack
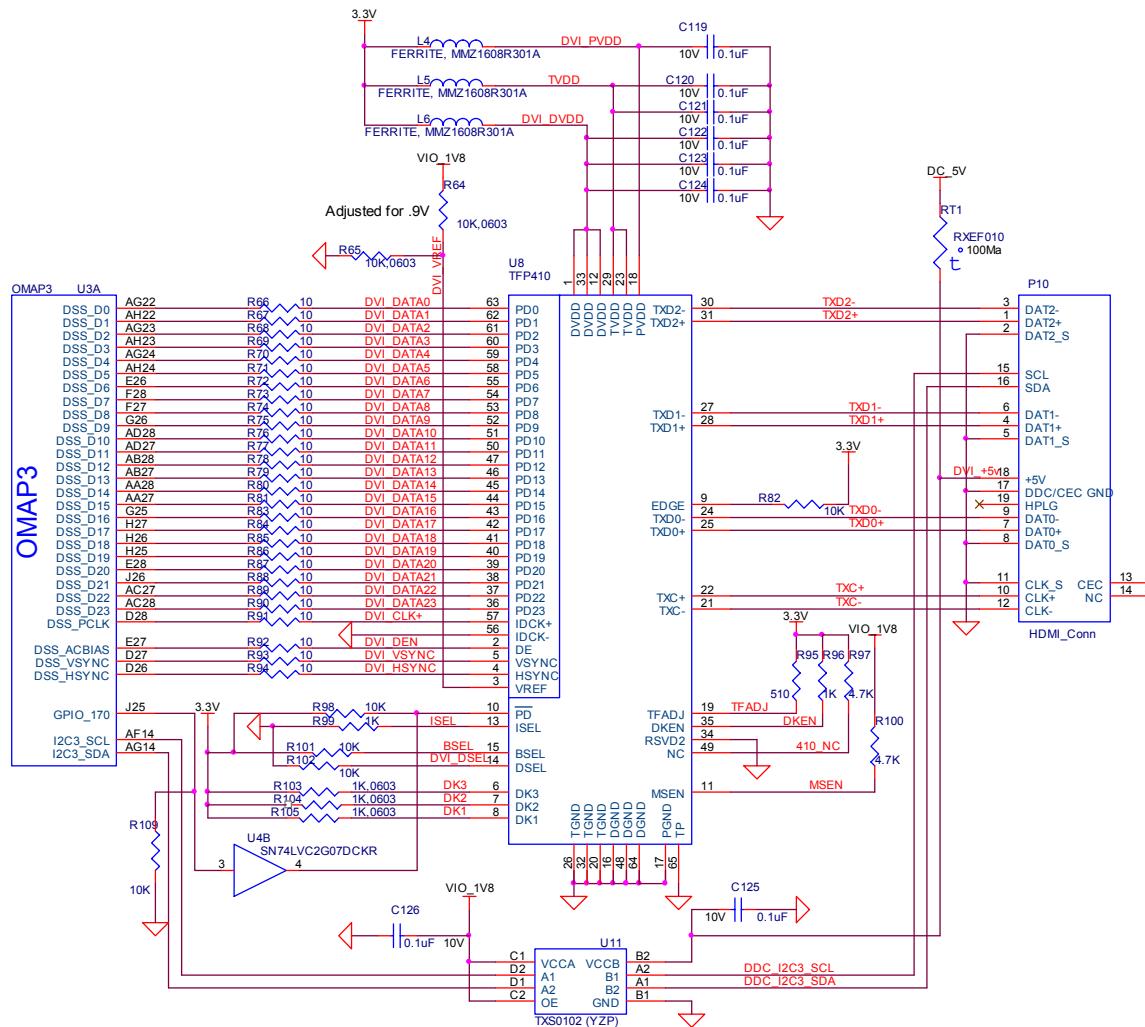
A single 3.5mm jack is provided on BeagleBoard to support external stereo audio output devices such as headphones and powered speakers.

8.11.4 Audio Input Jack

A single 3.5mm jack is supplied to support external audio inputs including stereo or mono.

8.12 DVI Interface

The LCD interface on the **OMAP3530** is accessible from the **DVI-D** interface connector on the board. **Figure 36** is the DVI-D design.

Figure 36.   DVI-D Interface

8.12.1 OMAP3530 LCD Interface

The main driver for the DVI-D interface originates at the **OMAP3530** via the **DSS** pins. The OMAP3530 provides 24 bits of data to the DVI-D framer chip, **TFP410**. There are

three other signals used to control the DVI-D that originate at the **OMAP3530**. These are I2C3_SCL, I2C3_SDA, and GPIO_170. All of the signals used are described in **Table 13.**

Table 13.   OMAP3530 LCD Signals

Signal Name	Description	Type	Ball
dss_pclk	LCD Pixel Clock	O	D28
dss_hsync	LCD Horizontal Synchronization	O	D26
dss_vsync	LCD Vertical Synchronization	O	D27
dss_acbias	Pixel data enable (TFT) output	O	E27
dss_data0	LCD Pixel Data bit 0	O	AG22
dss_data1	LCD Pixel Data bit 1	O	AH22
dss_data2	LCD Pixel Data bit 2	O	AG23
dss_data3	LCD Pixel Data bit 3	O	AH23
dss_data4	LCD Pixel Data bit 4	O	AG24
dss_data5	LCD Pixel Data bit 5	O	AH24
dss_data6	LCD Pixel Data bit 6	O	E26
dss_data7	LCD Pixel Data bit 7	O	F28
dss_data8	LCD Pixel Data bit 8	O	F27
dss_data9	LCD Pixel Data bit 9	O	G26
dss_data10	LCD Pixel Data bit 10	O	AD28
dss_data11	LCD Pixel Data bit 11	O	AD27
dss_data12	LCD Pixel Data bit 12	O	AB28
dss_data13	LCD Pixel Data bit 13	O	AB2
dss_data14	LCD Pixel Data bit 14	O	AA28
dss_data15	LCD Pixel Data bit 15	O	AA27
dss_data16 IO	LCD Pixel Data bit 16	O	G25
dss_data17	LCD Pixel Data bit 17	O	H27
dss_data18	LCD Pixel Data bit 18	O	H26
dss_data19	LCD Pixel Data bit 19	O	H25
dss_data20	LCD Pixel Data bit 20	O	E28
dss_data21	LCD Pixel Data bit 21	O	J26
dss_data22	LCD Pixel Data bit 22	O	AC27
dss_data23	LCD Pixel Data bit 23	O	AC28
GPIO_170	Powers down the TFP410 when Lo. TFP410 is active when Hi.	O	J25
I2C3_SCL	I2C3 clock line. Used to communicate with the monitor to determine setting information.	I/O	AF14
I2C3_SDA	I2C3 data line. Used to communicate with the monitor to determine setting information.	I/O	AG14

10ohm series resistors are provide in the signal path to minimize reflections in the high frequency signals from the **OMAP3530** to the **TFP410**. These resistors are in the form of resistor packs on the BeagleBoard. The maximum clock frequency of these signals is 65MHz.

8.12.2 OMAP3530 LCD Power

In order for the DSS outputs to operate correctly out of the OMAP3530, two voltage rails must be active, **VIO_1V8 and VDD_PLL2**. Both of these rails are controlled by the TWL4030 and must be set to 1.8V. By default, **VDD_PLL2** is not turned and must be activated by SW. Otherwise some of the bits will not have power supplied to them.

8.12.3 TFP410 Framer

The **TFP410** provides a universal interface to allow a glue-less connection to provide the DVI-D digital interface to drive external LCD panels. The adjustable 1.1-V to 1.8-V digital interface provides a low-EMI, high-speed bus that connects seamlessly with the 1.8V and 24-bit interface output by the **OMAP3530**. The DVI interface on the BeagleBoard supports flat panel display resolutions up to XGA at 65 MHz in 24-bit true color pixel format.

Table 14 is a description of all of the interface and control pins on the **TFP410** and how they are used on BeagleBoard.

Table 14.   TFP410 Interface Signals

Signal Name	Description	Type	Ball
DATA[23:12]	The upper 12 bits of the 24-bit pixel bus.	I	36–47
DATA[11:0]	The bottom 12 bits of the 24-bit pixel bus.	I	50–55.56-53
IDCK+	Single ended clock input.	I	57
IDCK-	Tied to ground to support the single ended mode.	I	56
DE	Data enable. During active video (DE = high), the transmitter encodes pixel data, DATA[23:0]. During the blanking interval (DE = low), the transmitter encodes HSYNC and VSYNC.	I	2
HSYNC	Horizontal sync input	I	4
VSYNC	Vertical sync input	I	5
DK3	These three inputs are the de-skew inputs DK[3:1], used to	I	6
DK2	adjust the setup and hold times of the pixel data inputs	I	7
DK1	DATA[23:0], relative to the clock input IDCK±.	I	8
MSEN	A low level indicates a powered on receiver is detected at the differential outputs. A high level indicates a powered on receiver is not detected..	O	11
ISEL	This pin disables the I2C mode on chip. Configuration is specified by the configuration pins (BSEL, DSEL, EDGE, VREF) and state pins (PD, DKEN).	I	13
BSEL	Selects the 24bit and single-edge clock mode.	I	13
DSEL	Lo to select the single ended clock mode.	I	14
EDGE	A high level selects the primary latch to occur on the rising edge of the input clock IDCK	I	9
DKEN	A HI level enables the de-skew controlled by DK[1:3]	I	35
V_{REF}	Sets the level of the input signals from the OMAP3530.	I	3
PD	A HI selects normal operation and a LO selects the powerdown mode.	I	10

TGADJ	This pin controls the amplitude of the DVI output voltage swing, determined by the value of the pullup resistor RTFADJ connected to 3.3V.	I	19

8.12.4 TFP410 Power

Power to the TFP410 is supplied from the 3.3V regulator in **U1**, the **TPS2141**. In order to insure a noise free signal, there are three inductors, **L4**, **L5**, and **L6** that are used to filter the 3.3V rail into the TFP410.

8.12.5 TFP410 Control Pins

There are twelve control pins that set up the TFP410 to operate with the **OMAP3530**. Most of these pins are set by HW and do not require any intervention by the **OMAP3530** to set them.

8.12.5.1   ISEL

The **ISEL** pin is pulled LO via **R99** to place the TFP410 in the control pin mode with the I2C feature disabled. This allows the other modes for the TFP410 to be set by the other control pins.

8.12.5.2   BSEL

The **BSEL** pin is pulled HI to select the 24 bit mode for the Pixel Data interface from the OMAP3530..

8.12.5.3   DSEL

The **DSEL** pin is pulled low to select the single ended clock mode from the **OMAP3530**.

8.12.5.4   EDGE

The **EDGE** signal is pulled HI through **R82** to select the rising edge on the IDCK+ lead which is the pixel clock from the **OMAP3530**.

8.12.5.5   DKEN

The **DKEN** signal is pulled HI to enable the de-skew pins. The de-skew pins, **DK1-DK3,** are pulled low by the internal pulldown resistors in the **TFP410**. This is the default mode of operation. If desired, the resistors can be installed to pull the signals high. However, it is not expected that any of the resistors will need to be installed. The DK1-DK3 pins adjust the timing of the clock as it relates to the data signals.

8.12.5.6   MSEN

The **MSEN** signal, when low, indicates that there is a powered monitor plugged into the DVI-D connector. This signal is not connected to the **OMAP3530** and is provided as a test point only.

8.12.5.7   VREF

The **VREF** signal sets the voltage level of the **DATA**, **VSYNC**, **HSYNC**, **DE**, and **IDCK+** leads from the OMAP3530. As the **OMAP3530** is 1.8V, the level is set to .9V by **R64** and **R65**.

8.12.5.8   PD

The **PD** signal originates from the **OMAP3530** on the GPIO_170 pin. Because the **PD** signal on the **TFP410** is 3.3V referenced, this signal must be converted to **3.3V**. This is done by **U4**, **SN74LVC2G07**, a non-inverting open drain buffer. If the **GPIO_170** pin is HI, then the open drain signal is inactive, causing the signal to be pulled HI by **R98**. When **GPIO_170** is taken low, the output of **U4** will also go LO, placing the **TFP410** in the power down mode. Even though **U4** is running at 1.8V to match the **OMAP3530**, the output will support being pulled up to **3.3V**. On power up, the TFP410 is disabled by **R109**, a 10K resistor. When the **OMAP3530** powers on, pin **J25** comes in a safe mode, meaning it is not being driven. R109 insures that the signal is pulled LO, putting the TFP410 in the power down mode.

8.12.5.9   TFADJ

The **TFADJ** signal controls the amplitude of the DVI output voltage swing, determined by the value of R95.

8.12.5.10   RSVD2

This unused pin is terminated to ground as directed by the TFP410 data manual.

8.12.5.11   NC

This unused pin is pulled HI as directed by the TFP410 data manual.

8.12.6   DVI-D Connector

In order to minimize board size, a HDMI connector was selected for the DVI-D connection. The BeagleBoard does not support HDMI but only the DVI-D component of HDMI. The Cable is not supplied with the BeagleBoard but is available from numerous cable suppliers and is required to connect a display to the BeagleBoard.

![beagleboard.org logo]

8.12.6.1 Shield Wire

Each signal has a shield wire that is used in the cable to provide signal protection for each differential pair. This signal is tied directly to ground.

8.12.6.2 DAT0+/DAT0-

The differential signal pair **DAT0+/DAT0-** transmits the 8-bit blue pixel data during active video and HSYNC and VSYNC during the blanking interval.

8.12.6.3 DAT1+/DAT1-

The differential signal pair **DAT1+/DAT1-** transmits the 8-bit green pixel data during active video.

8.12.6.4 DAT2+/DAT2-

The differential signal pair **DAT2+/DAT2-** transmits the 8-bit red pixel data during active.

8.12.6.5 TXC+/TXC-

The differential signal pair **TXC+/TXC-** transmits the differential clock from the TFP410.

8.12.6.6 DDC Channel

The **Display Data Channel** or **DDC (**sometimes referred to as EDID Enhanced Display ID**)** is a digital connection between a computer display and the **OMAP3530** that allows the display specifications to be read by the **OMAP3530**. The standard was created by the Video Electronics Standards Association (VESA). The current version of DDC, called DDC2B, is based on the I²C bus. The monitor contains a read-only memory (ROM) chip programmed by the manufacturer with information about the graphics modes that the monitor can display. This interface in the LCD panel is powered by the +5V pin on the connector through **RT1**, a resetable fuse. As the **OMAP3530** is 1.8V I/O, the I2C bus is level translated by **U11**, a **TXS0102**. It provides for a split rail to allow the signals to communicate. Inside of **TXS0102** is a pullup on each signal, removing the need for an external resistor.

8.13   S-Video

A single S-Video port is provided on the BeagleBoard. **Figure 37** is the design of the S-Video interface.
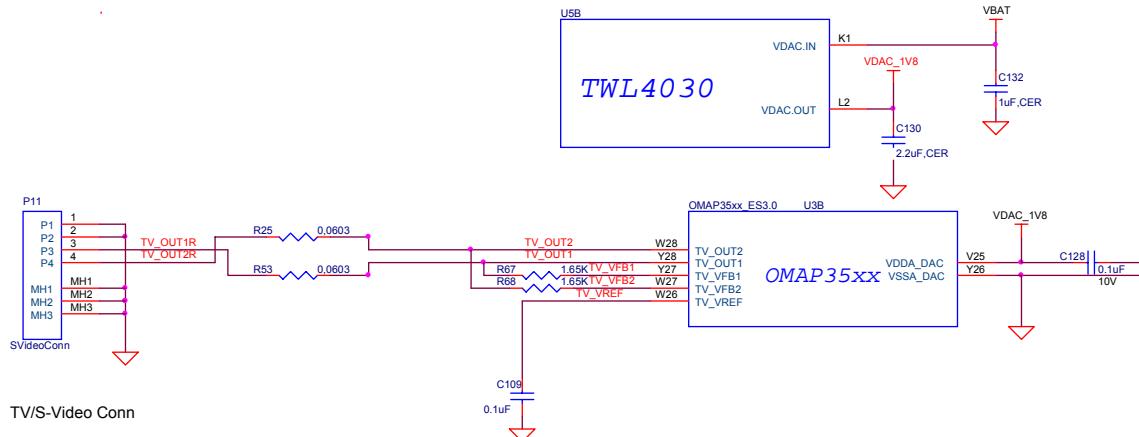
Figure 37.   S-Video Interface

Table 15 is the list of the signals on the S-Video interface and their definitions.

Table 15.   TFP410 Interface Signals

Signal	I/O	Description
tv_out1	O	TV analog output composite
tv_out2	O	TV analog output S-VIDEO
tv_vref	I	Reference output voltage from internal bandgap
tv_vfb1	O	Amplifier feedback node
tv_vfb2	O	Amplifier feedback node

You will notice a bank of two resistors, **R23**, and **R25**, on the S-Video output. These are zero ohm resistors and are not required in the design. They are a hold over from the previous version of the board and were left there just in case they were needed. Notice that the signal names are in a different order than they are on the **OMAP3530**. So, while it appears that the signals are swapped in that they cross, in reality they do not.

Power to the internal DAC is supplied by the TWL4030 via the VDAC_1V8 rail. **Figure 37** reflects the filtering that is used on these rails, including the input VBAT rail.

8.14 RS232 Port

A single RS232 port is provided on the BeagleBoard. It provides access to the TX and RX lines of **UART3** on the OMAP3530. **Figure 38** shows the design of the RS232 port.
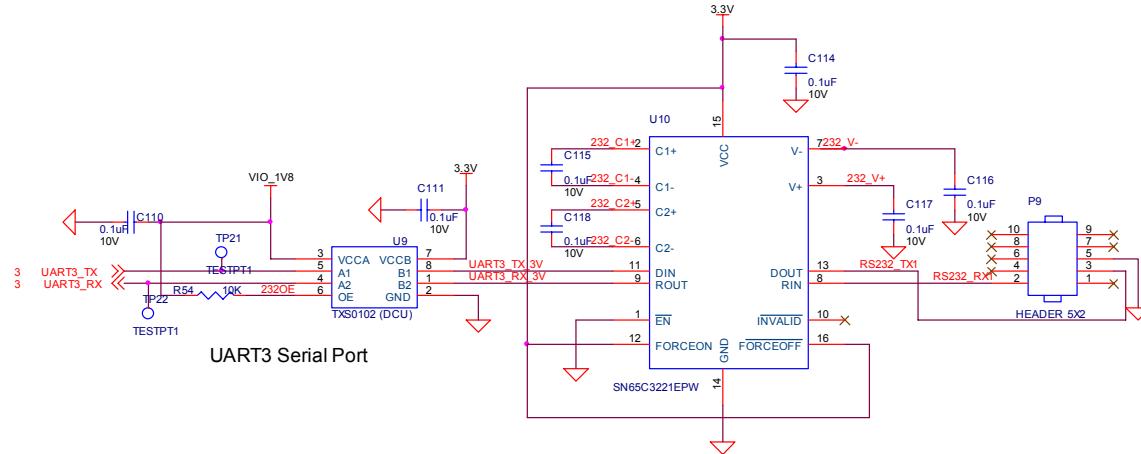
Figure 38.   RS232 Interface Design

Figure 38 reflects the new package, DCU that was incorporated in the layout for Rev B6. Refer to Rev B5 for the Rev B5 and older that do not use the new package.

8.14.1 OMAP3530 Interface

Two lines, **UART3_Tx** and **UART3_Rx**, are provided by the OMAP3530. The **UART3** function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate and also supports auto bauding.

8.14.2 OMAP3530 Level Translator

All of the I/O levels from the OMAP3530 are **1.8V** while the transceiver used runs at 3.3V. This requires that the voltage levels be translated. This is accomplished by the TXS0102 which is a two-bit noninverting translator that uses two separate configurable power-supply rails. The A port tracks VCCA, 1.8V and the B port tracks VCCB, 3.3V. This allows for low-voltage bidirectional translation between the two voltage nodes. When the output-enable (OE) input is low, all outputs are placed in the high-impedance state. In this design, the OE is tied high via a 10K ohm resistor to insure that it is always on.

8.14.3 RS232 Transceiver

The RS232 transceiver used is the SN65C3221. The SN65C3221 consist of one line driver, one line receiver, and a dual charge-pump circuit with ±15-kV IEC ESD

protection pin to pin (serial-port connection pins, including GND). These devices provide the electrical interface between an asynchronous communication controller and the serial-port connector. The charge pump and four small external capacitors allow operation from a single 3-V to 5.5-V supply. The **SN65C3221** operates at data signaling rates up to 1 Mbit/s and a driver output slew rate of 24 V/ms to 150 V/ms. While the **OMAP3530** can easily drive a 1Mbit/S rate, your results may vary based on cabling, distance, and the loads and drive capability on the other end of the RS232 port.

The transceiver is powered from the 3.3V rail and is active at power up. This allows the port to be used for UART based peripheral booting over the port.

8.14.4   Connector

Access to the RS232 port is through a 10pin header, **P9**. Connection to the header is through a 10 pin IDC to 9 pin D-sub cable. This header requires the use of an ATI-Everex type cable. This is the only cable that will work. This cable is readily available from a number of sources and is commonly found on many PC motherboards and is not supplied with the BeagleBoard. **Figure 39** is a picture of what the cable assembly looks like.
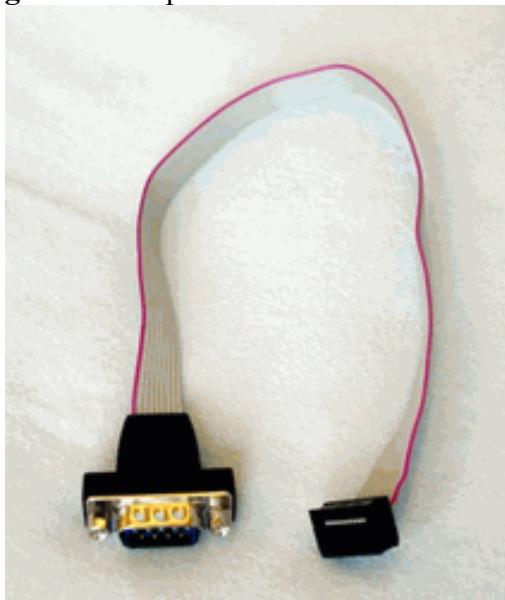
Figure 39.   RS232 Cable

When purchasing, make sure the ATI-Everex or pass through cable is ordered.

8.15   Indicators

There are four green indicators on the BeagleBoard:

- o Power
- o PMU_STAT

  o USER0
  o USER1

Three of these are programmable under SW control and the fourth one is tied to the main power rail.

Figure 40 shows the connection of all of these indicators.
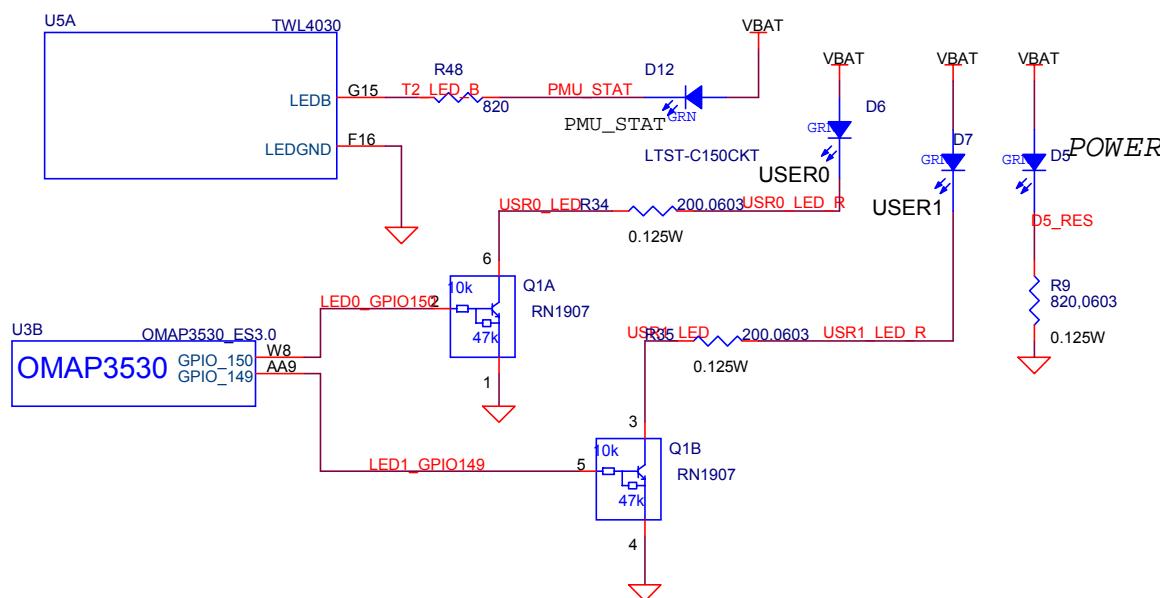
Figure 40. Indicator Design

8.15.1 Power Indicator

This indicator, **D7**, connects across the **VBAT** supply and ground. It indicates that the entire power path is supplying the power to the board. The **VBAT** regulator can be driven from either the USB Client port or an external 5VDC power supply. Indicator **D7** does not indicate where one or the other is being used to supply the main power to the board.

8.15.2 PMU Status Indicator

This output is driven from the **TWL4030** using the **LED.B** output. The **TWL4030** provides LED driver circuitry to power two LED circuits that can provide user indicators. The first circuit can provide up to 160 mA and the second, 50 mA. Each LED circuit is independently controllable for basic power (on/off) control and illumination level (using PWM). The second driver, **LED.B**, is used to drive an LED that is connected to the VBAT rail through a resistor.

The PWM inside the TWL4030 can be used to alter the brightness of the LED if desired or it can be turned on or off by the **OMAP3530** using the I2C bus. The PWM is programmable, register-controlled, duty cycle based on a nominal 4-Hz cycle which is

derived from an internal 32-kHz clock. It is possible to set the LED to flash automatically without SW control if desired.

8.15.3   User Indicators

There are two user LEDs that can be driven directly from a GPIO pin on the **OMAP3530**. These can be used for any purpose by the SW. The output level of the **OMAP3530** is 1.8V and the current sink capability is not enough to drive an LED with any level of brightness. A transistor pair, **RN1907** is used to drive the LEDs from the **VBAT** rail. A logic level of 1 will turn the LED on.

In the REV A version, the LEDS were shorted, not allowing them to be controlled separately. That has been fixed on the REV B version.

8.16   JTAG

A JTAG header is provided to allow for advanced debugging on the BeagleBoard by using a JTAG based debugger **Figure 41** shows the interconnection to the OMAP3530 processor.
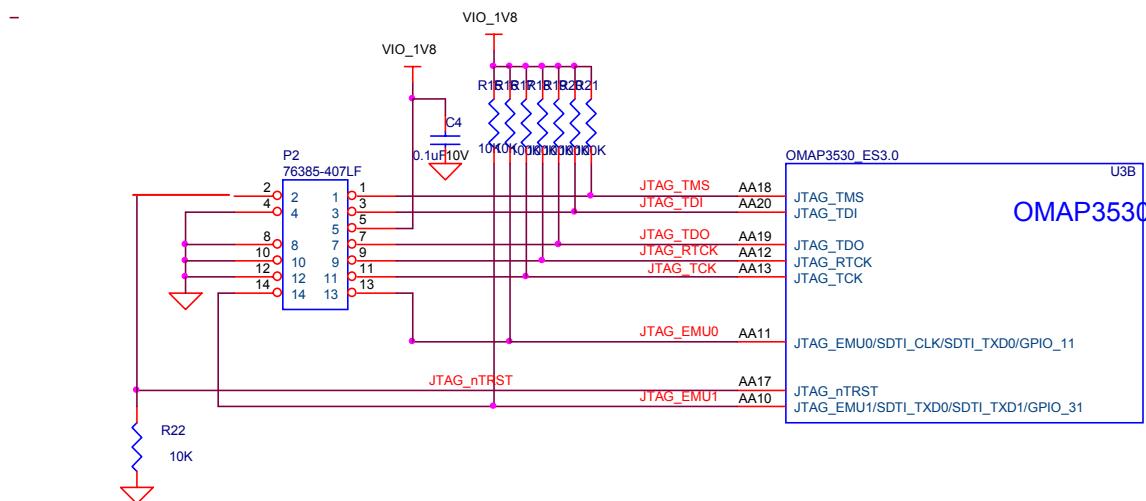
Figure 41.   JTAG Interface

On the REV A version, a jumper was provided to allow the EMU0 line to be grounded. The version of the OMAP3 processor used on the REV B does not have this requirement. For this reason, the jumper was removed.

8.16.1 OMAP3530 Interface

The JTAG interface connects directly to the OMAP processor. All signals are a 1.8V
level. **Table 16** describes the signals on the JTAG connector.

Table 16.   JTAG Signals

Signal	Description	I/O
JTAG_TMS	Test mode select	I/O
JTAG_TDI	Test data input	I
JTAG_TDO	Test Data Output	O
JTAG_RTCK	ARM Clock Emulation	O
JTAG_TCK	Test Clock	I
JTAG_nTRST	Test reset	I
JTAG_EMU0	Test emulation 0	I/O
JTAG_EMU1	Test emulation 1	I/O

8.16.2 Connector

The JTAG interface uses a 14 pin connector. All JTAG emulator modules should be able
to support this interface. Contact your emulator supplier for further information or if an
adapter is needed.

8.17   Expansion Header

The expansion header is provided to allow a limited number of functions to be added to
the board via the addition of a daughtercard.

Figure 42 is the design of the expansion connector and the interfaces to the OMAP3530.
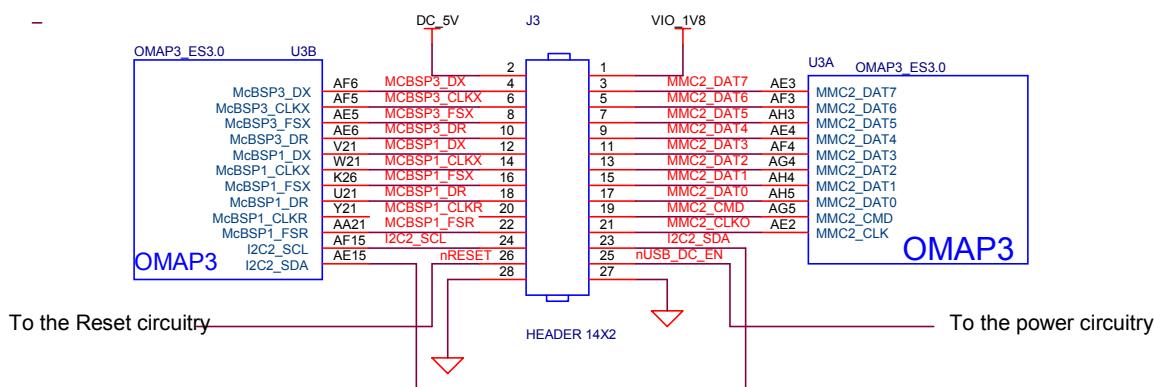
Figure 42.   Expansion Header

NOTE: The Expansion header itself is NOT provided on the BeagleBoard. This is user installed option. This header is not populated on the BeagleBoard so that based on the usage scenario; it can be populated as needed (Top, Bottom, Top right angle, or Bottom Right angle). The user should take care in installing this header.

CAUTION: The voltage levels on the expansion header are 1.8V. Exposure of these signals to a higher voltage will result in damage to the board and a voiding of the warranty.

8.17.1 OMAP3530 Interface

The main purpose of the expansion connector is to route additional signals from the **OMAP3530** processor. **Table 17** shows all of the signals that are on the expansion header. As the **OMAP3530** has a multiplexing feature, multiple signals can be connected to certain pins to add additional options as it pertains to the signal available. The different columns in **Table 17** show what other signals can be accessed by setting the mux control register in the **OMAP3530**.

Table 17.   Expansion Connector Signals

Pin	Option A	Option B	Option C	Option D
1	VIO_1V8			
2	DC_5V			
3	MMC2_DAT7	GPIO_139		
4	McBSP3_DX	GPIO_140	UART2_CTS	
5	MMC2_DAT6	GPIO_138		
6	McBSP3_CLKX	GPIO_142	UART2_TX	
7	MMC2_DAT5	GPIO_137		
8	McBSP3_FSX	GPIO_143	UART2_RX	
9	MMC2_DAT4	GPIO_136		
10	McBSP3_DR	GPIO_141	UART2_RTS	
11	MMC2_DAT3	McSPI3_CS0	GPIO_135	
12	McBSP1_DX	McSPI4_SIMO	McBSP3_DX	GPIO_158
13	MMC2_DAT2	McSPI3_CS1	GPIO_134	
14	McBSP1_CLKX	McBSP3_CLKX	GPIO_162	
15	MMC2_DAT1	GPIO_133		
16	McBSP1_FSX	McSPI4_CS0	McBSP3_FSX	GPIO_161
17	MMC2_DAT0	McSPI3_SOMI	GPIO_132	
18	McBSP1_DR	McSPI4_SOMI	McBSP3_DR	GPIO_159
19	MMC2_CMD	McSPI3_SIMO	GPIO_131	
20	McBSP1_CLKR	McSPI4_CLK	SIM_CD	GPIO_156
21	MMC2_CLKO	McSPI3_CLK	GPIO_130	
22	McBSP1_FSR			GPIO_157
23	I2C2_SDA	GPIO_183		
24	I2C2_SCL	GPIO_168		
25	REGEN			
26	nRESET			
27	GND			
28	GND			

8.17.2 Expansion Signals

This section provides more detail on each of the signals available on the expansion connector. They are grouped by functions in **Table 18** along with a description of each signal.

If you use these signals in their respective groups and that is the only function you use, all of the signals are available. Whether or not the signals you need are all available, depends on the muxing function on a per pin bases. Only one signal per pin is available at any one time.

Table 18.   Expansion Connector Signals

Signal	Description	I/O	Pin
SD/MMC Port 2			
MMC2_DAT7	SD/MMC data pin 7.	I/O	3
MMC2_DAT6	SD/MMC data pin 6.	I/O	5
MMC2_DAT5	SD/MMC data pin 5.	I/O	7
MMC2_DAT4	SD/MMC data pin 4.	I/O	9
MMC2_DAT3	SD/MMC data pin 3.	I/O	11
MMC2_DAT2	SD/MMC data pin 2.	I/O	13
MMC2_DAT1	SD/MMC data pin 1.	I/O	15
MMC2_DAT0	SD/MMC data pin 0.	I/O	17
MMC2_CMD	SD/MMC command signal.	I/O	19
MMC_CLKO	SD/MMC clock signal.	O	21
McBSP Port 1			
McBSP1_DR	Multi channel buffered serial port receive	I	18
McBSP1_CLKS	---	N/A	N/A
McBSP1_FSR	Multi channel buffered serial port transmit frame sync RCV	I/O	22
McBSP1_DX	Multi channel buffered serial port transmit	I/O	12
McBSP1_CLKX	Multi channel buffered serial port transmit clock	I/O	14
McBSP1_FSX	Multi channel buffered serial port transmit frame sync XMT	I/O	16
McBSP1_CLKR	Multi channel buffered serial port receive clock	I/O	20
I2C Port 2			
I2C2_SDA	I2C data line.	IOD	23
I2C2_SCL	I2C clock line	IOD	24
McBSP Port 3			
McBSP3_DR	Multi channel buffered serial port receive	I	10,18
McBSP3_DX	Multi channel buffered serial port transmit	I/O	4,12
McBSP3_CLKX	Multi channel buffered serial port receive clock	I/O	6,14
McBSP3_FSX	Multi channel buffered serial port frame sync transmit	I/O	8,16
General Purpose I/O Pins			
GPIO_130	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	21
GPIO_131	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	19
GPIO_132	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	17
GPIO_133	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	15
GPIO_134	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	13
GPIO_135	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	11
GPIO_136	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	9
GPIO_137	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	7
GPIO_138	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	5
GPIO_139	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	3
GPIO_140	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	4
GPIO_141	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	10
GPIO_142	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	6
GPIO_143	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	8
GPIO_156	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	20
GPIO_158	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	12
GPIO_159	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	18

GPIO_161	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	16
GPIO_162	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	14
GPIO_168	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	24
GPIO_183	General Purpose Input/Output pin. Can be used as an interrupt pin.	I/O	23
McSPI Port 3			
McSPI3_CS0	Multi channel SPI chip select 0	O	11
McSPI3_CS1	Multi channel SPI chip select 1	O	13
McSPI3_SIMO	Multi channel SPI slave in master out	I/O	19
McSPI3_SOMI	Multi channel SPI slave out master in	I/O	17
McSPI3_CLK	Multi channel SPI clock	I/O	21
McSPI Port 4			
McSPI4_SIMO	Multi channel SPI slave in master out	I/O	12
McSPI4_SOMI	Multi channel SPI slave out master in	I/)	18
McSPI4_CS0	Multi channel SPI chip select 0	O	16
McSPI4_CLK	Multi channel SPI clock	I/O	20
UART Port 2			
UART2_CTS	UART clear to send.	I/O	4
UART2_RTS	UART request to send	O	10
UART2_RX	UART receive	I	8
UART2_TX	UART transmit	O	6

8.17.3 Power

The expansion connector provides two power rails. The first is the **VIO_1.8V** rail which is supplied by the **TWL4030**. This rail is limited in the current it can supply from the **TWL4030** and what remains from the current consumed by the BeagleBoard and is intended to be used to provide a rail for voltage level conversion only. It is not intended to power a lot of circuitry on the expansion board. All signals from the BeagleBoard are at 1.8V.

The other rail is the **DC_5V**. The same restriction exits on this rail as mentioned in the USB section. The amount of available power to an expansion board depends on the available power from the DC supply or the USB supply from the PC.

8.17.4 Reset

The **nRESET** signal is the main board reset signal. When the board powers up, this signal will act as an input to reset circuitry on the expansion board. After power up, a system reset can be generated by the expansion board by taking this signal low. This signal is a 1.8V level signal.

8.17.5 Power Control

There is an additional open-drain signal on the connector called **REGEN**. The purpose of this signal is to provide a means to control power circuitry on the expansion card to turn on and off the voltages. This insures that the power on the board is turned on at the appropriate time. Depending on what circuitry is provided on the expansion card, an additional delay may be needed to be added before the circuitry is activated. Refer to the **OMAP3530** and **TWL4030** documentation for more information.

9.0   Connector Pinouts and Cables

This section provides a definition of the pinouts and cables to be used with all of the connectors and headers on the BeagleBoard.

<u>**THERE ARE NO CABLES SUPPLIED WITH THE BEAGLEBOARD.**</u>

9.1   Power Connector

Figure 43 is a picture of the BeagleBoard power connector with the pins identified. The supply must have a 2.1mm center hot connector with a 5.5mm outside diameter.
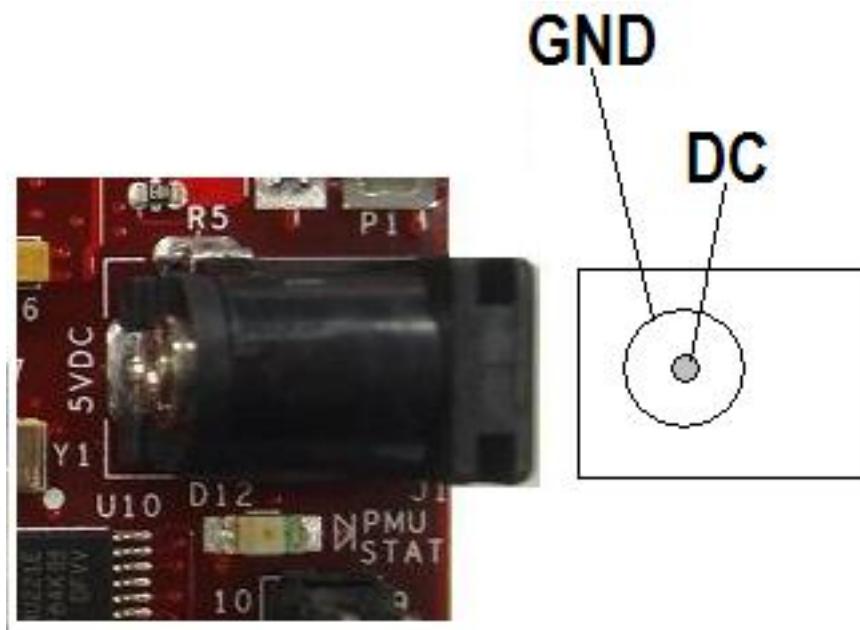
Figure 43.   Power Connector

The supply must be at least 500mA with a maximum of 2A. If the expansion connector is used, more power will be required depending on the load of the devices connected to the expansion connector.

<u>WARNING:</u> DO NOT PLUG IN ANYTHING BUT **5V** TO THE DC CONNECTOR OR THE BOARD WILL BE DAMAGED!

beagleboard.org

9.2   USB OTG

Figure 44 is a picture of the BeagleBoard USB OTG connector with the pins identified.

Figure 44.   USB OTG Connector

9.3   S-Video

Figure 45 is the S-Video connector on the BeagleBoard.
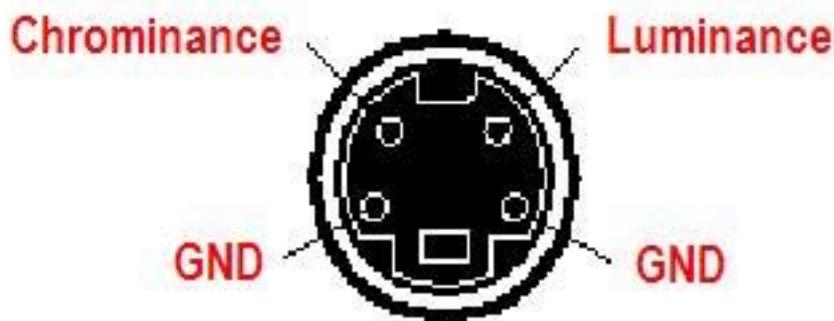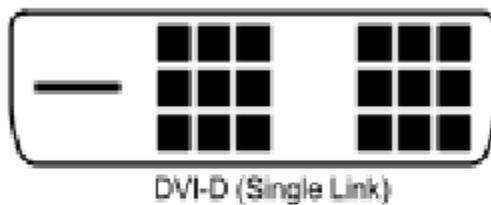
Figure 45.   S-Video Connector

9.4   DVI-D

Figure 46 is the pinout of the DVI-D connector on BeagleBoard.
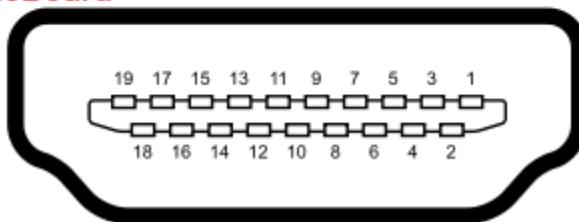
Figure 46.   DVI-D Connector

Table 19 is the pin numbering of the two ends of the cable as it relates to the signals used in the DVI-D interface itself.

Table 19.   DVI-D to HDMI Cable

SIGNAL	DVI-D PIN#	DVI-D PIN#
DATA 2-	1	3
DATA 2+	2	1
SHIELD	3	2
	4	
	5	
DDS CLOCK	6	15
DDS DATA	7	16
	8	
DATA 1-	9	6
DATA 1+	10	4
SHIELD	11	5
	12	
	13	
5V	14	18
GROUND (5V)	15	17
	16	

SIGNAL	DVI-D PIN#	DVI-D PIN#
DATA 0-	17	9
DATA 0+	18	7
SHIELD	19	5
	20	
	21	
	22	
CLOCK+	23	10
CLOCK-	24	12

Figure 47 is the cable to be used to connect to an LCD monitor.

Figure 47.   DVI-D Cable

9.5   Audio Connections

Figure 48 is the audio input jack required to connect to the BeagleBoard.
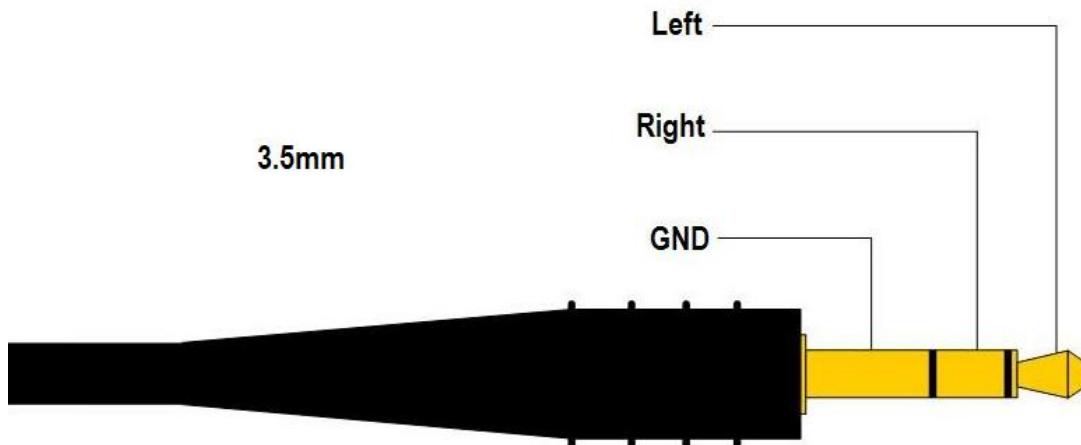
Figure 48.   Audio In Plug

Figure 49 is the actual connector used on the BeagleBoard.

Figure 49.   Audio In Plug

9.6   Audio Out

Figure 50 is the audio out jack required to connect to the BeagleBoard.
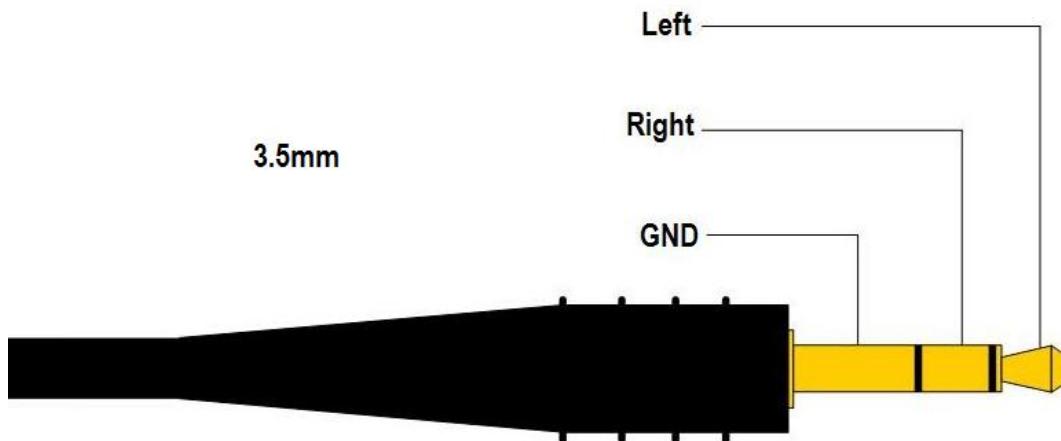
Figure 50.   Audio Out Plug

Figure 51 is the actual connector used on the BeagleBoard.

Figure 51.   Audio In Plug

9.7   JTAG

Figure 52 is the JTAG connector pin out showing the pin numbering.
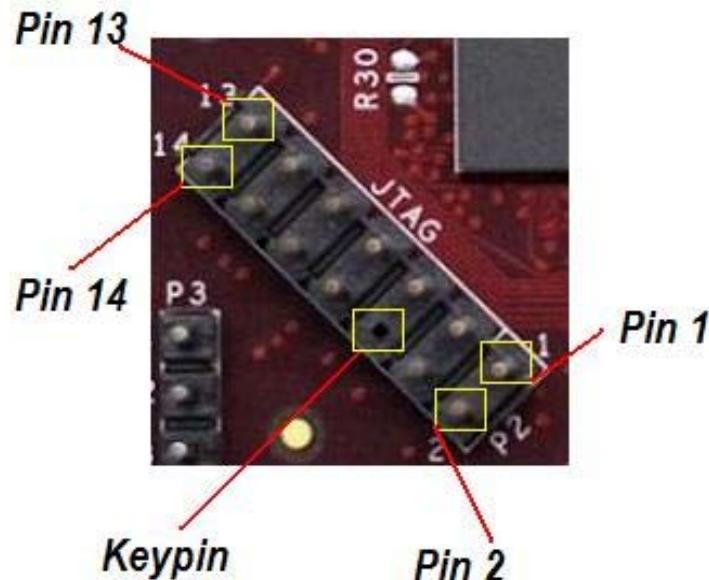
Figure 52.   JTAG Connector Pinout

Table 20 gives a definition of each of the signals on the JTAG header.

Table 20.   JTAG Signals

Pin	Signal	Description	I/O
1	JTAG_TMS	Test mode select	I/O
3	JTAG_TDI	Test data input	I
7	JTAG_TDO	Test Data Output	O
9	JTAG_RTCK	ARM Clock Emulation	O
11	JTAG_TCK	Test Clock	I
2	JTAG_nTRST	Test reset	I
13	JTAG_EMU0	Test emulation 0	I/O
14	JTAG_EMU1	Test emulation 1	I/O
5	VIO	Voltage pin	PWR
4,8,10,12,14	GND	Ground	PWR

All of the signals are 1.8V only. The JTAG emulator must support 1.8V signals for use on the BeagleBoard.

If a 20 pin connector is provided on the JTAG emulator, then a 20 pin to 14 pin adapters must be used. You may also use emulators that are either equipped with a 14 pin connector or are universal in nature.

![beagleboard.org]

Figure 51 shows an example of a 14 pin to 20 pin adapter.
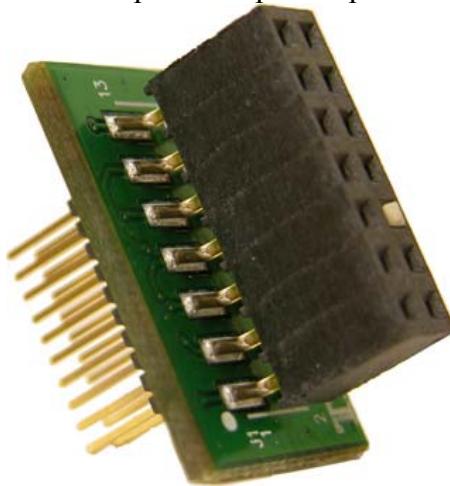
Figure 53.   JTAG 14 to 20 Pin Adapter

Figure 52 shows how the JTAG cable is to be routed when connected to the BeagleBoard.
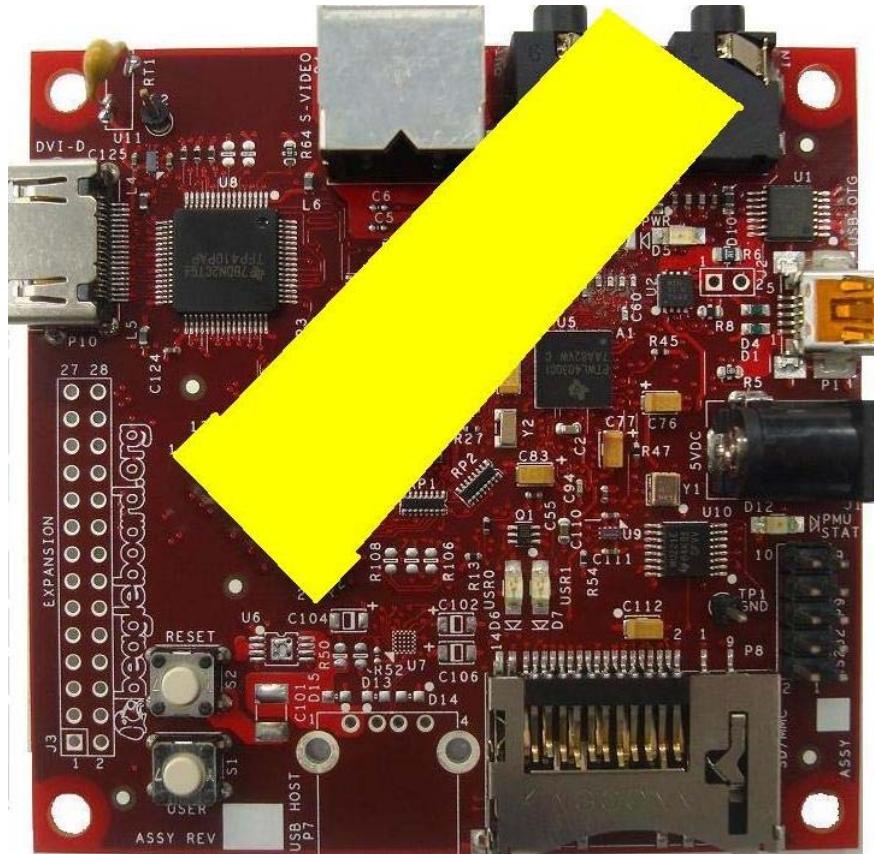
Figure 54.   JTAG Connector Pinout

9.8   RS232

Figure 53 is the RS232 header on the BeagleBoard with the pin numbers identified.
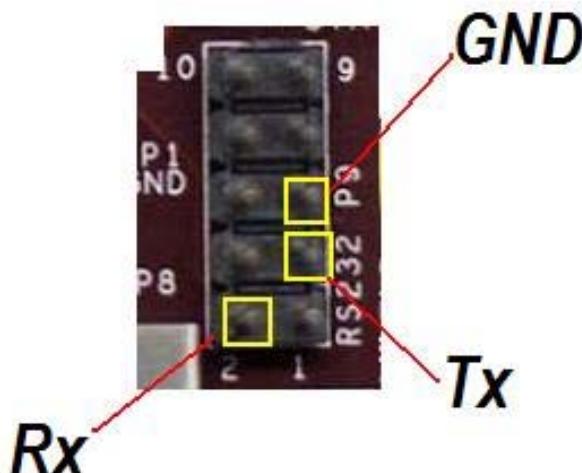
Figure 55.   RS232 Header

Figure 54 is the cable that is required in order to access the RS232 header. This cable can be purchased from various sources and is referred to as the ATI/Everex type cable.
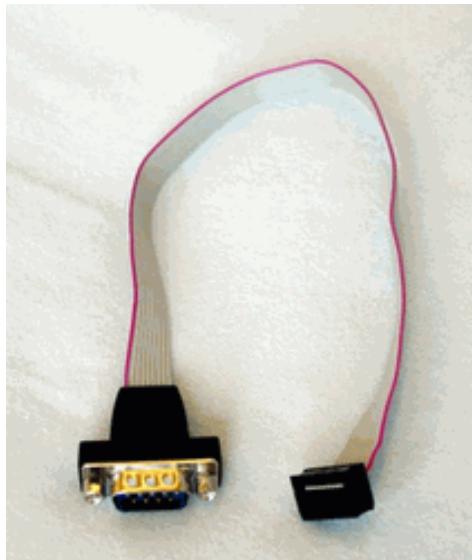
Figure 56.   RS232 Flat Cable

9.9   Expansion

Figure 54 is the pinout of the expansion header on the BeagleBoard. It is a standard 14 x 2 header with .1" (2.54mm) x .2"(5.08).

Figure 57.   Expansion Header

Table 21 is the signals on the Expansion Connector.

Table 21.   Expansion Connector Signals

Pin	Option A	Option B	Option C	Option D
1	VIO_1V8			
2	DC_5V			
3	MMC2_DAT7	GPIO_139		
4	McBSP3_DX	GPIO_140	UART2_CTS	
5	MMC2_DAT6	GPIO_138		
6	McBSP3_CLKX	GPIO_142	UART2_TX	
7	MMC2_DAT5	GPIO_137		
8	McBSP3_FSX	GPIO_143	UART2_RX	
9	MMC2_DAT4	GPIO_136		
10	McBSP3_DR	GPIO_141	UART2_RTS	
11	MMC2_DAT3	McSPI3_CS0	GPIO_135	
12	McBSP1_DX	McSPI4_SIMO	McBSP3_DX	GPIO_158
13	MMC2_DAT2	McSPI3_CS1	GPIO_134	
14	McBSP1_CLKX	McBSP3_CLKX	GPIO_162	
15	MMC2_DAT1	GPIO_133		
16	McBSP1_FSX	McSPI4_CS0	McBSP3_FSX	GPIO_161
17	MMC2_DAT0	McSPI3_SOMI	GPIO_132	
18	McBSP1_DR	McSPI4_SOMI	McBSP3_DR	GPIO_159
19	MMC2_CMD	McSPI3_SIMO	GPIO_131	
20	McBSP1_CLKR	McSPI4_CLK	SIM_CD	GPIO_156
21	MMC2_CLKO	McSPI3_CLK	GPIO_130	
22	McBSP1_FSR			
23	I2C2_SDA	GPIO_183		
24	I2C2_SCL	GPIO_168		
25	REGEN			
26	nRESET			
27	GND			
28	GND			

![beagleboard.org logo]

10.0   BeagleBoard Accessories

Throughout this manual various items are mentioned as not being provided with the standard BeagleBoard package or as options to extend the features of the BeagleBoard. The concept behind BeagleBoard is that different features and functions can be added to BeagleBoard by bringing your own peripherals. This has several key advantages:

- o User can choose which peripherals to add.
- o User can choose the brand of peripherals based on driver availability and ability to acquire the particular peripheral
- o User can add these peripherals at a lower cost than if they were integrated into the BeagleBoard.

This section covers these accessories and add-ons and provides information on where they may be obtained. Obviously things can change very quickly as it relates to devices that may be available. Please check BeagleBoard.org for an up to date listing of these peripherals.

Inclusion of any products in this section does not guarantee that they will operate with all SW releases. It is up to the user to find the appropriate drivers for each of these products. Information provided here is intended to expose the capabilities of what can be done with the BeagleBoard and how it can be expanded.

All pricing information provided is subject to change an din most cases is likely to be lower depending on the products purchased and from where they are purchased.

Covered in this section are the following accessories:

- o DC Power Supplies
- o Serial Ribbon cable
- o USB Hubs
- o USB Thumb Drives
- o DVI-D Cables
- o DVI-D Monitors
- o SD/MMC Cards
- o USB to Ethernet
- o USB to WiFi
- o USB Bluetooth
- o Expansion Cards

NO CABLES OR POWER SUPPLIES ARE PROVIDED WITH THE BEAGLEBOARD.

10.1   DC Power Supply

Tabletop or wall plug supplies can be used to power BeagleBoard. **Table 22** provides the specifications for the BeagleBoard DC supply. Supplies that provide additional current than what is specified can be used if additional current is needed for add on accessories. The amount specified is equal to that supplied by a USB port.

Table 22.   DC Power Supply Specifications

Specification	Requirement	Unit
Voltage	5.0	V
Current	500mA (minimum)	mA
Connector	2.1mm x 5.5mm Center hot	

It is recommended that a supply higher than 500mA be used if higher current peripherals are expected to be used or if expansion boards are added.

Table 23 lists some power supplies that will work with the BeagleBoard.

Table 23.   DC Power Supplies

Part #	Manufacturer	Supplier	Price
DCU090050E2961	RELIAPRO	Jameco	$5
EPS050100-P6P	CUI	Digi-Key	$7
DPS050200UPS-P5P-SZ	CUI	Digi-Key	$16

Figure 58 is a picture of the type of power supply that will be used on the BeagleBoard.

Figure 58.   DC Power Supply

10.2   Serial Ribbon Cable

Figure 59 is the serial ribbon cable for the BeagleBoard.
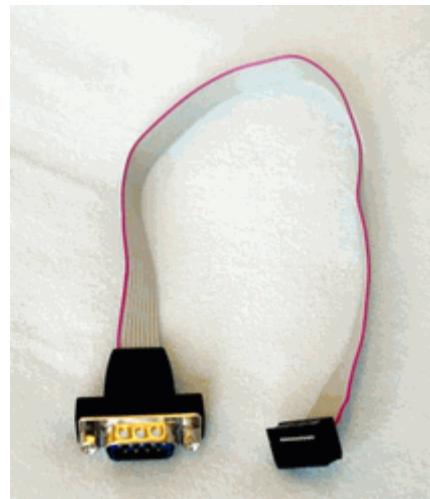
Figure 59.   RS232 Cable

If you like, you can also make your own cable. **Figure 58** shows the internal wiring of the cable in **Figure 60**.
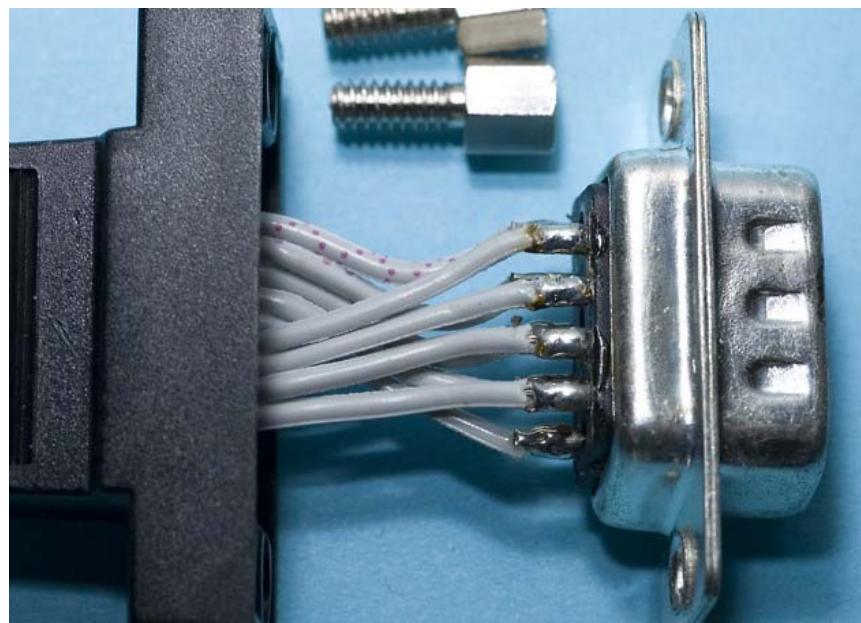
Figure 60.   RS232 Cable

Table 24 shows the pinout of the ribbon cable connector.

Table 24.   Cable Pinout

Ribbon Cable	DB9
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

10.3   USB Hubs

There are no known or anticipated issues with USB hubs. However, it should be noted that a self powered hub is highly recommended. **Table 25** is a list of Hubs that have been tested on the BeagleBoard.

Table 25.   USB Hubs Tested

Supplier	Part Number
IOGEAR	GUH274
D-Link	DUB-H4 2.0
Vakoss	TC-204-NS

10.4   DVI Cables

In order to connect the DVI-D interface to a LCD monitor, a HDMI to DVI-D cable is required. **Figure 61** is a picture of a HDMI to DVI-D cable.

Figure 61.   HDMI to DVI-D Cable

10.5   DVI-D Monitors

There are many monitors that can be used with the BeagleBoard. With the integrated EDID feature, timing data is collected from the monitor to enable the SW to adjust its timings. **Table 26** shows a short list of the monitors that have been tested to date on the BeagleBoard at the 1024x768 resolution. Please check on BeagleBoard.org for an up to date listing of the DVI-D monitors as well as information on the availability of drivers.

Table 26.   DVI-D Monitors Tested

Manufacturer	Part Number	Status
Dell	2407WFPb	Tested
Insignia	NS-LCD15	Tested
Dell	1708FP	Tested

10.6   SD/MMC Cards

Table 27 is a list of SD/MMC cards that have been tested on BeagleBoard. Please check BeagleBoard.org for an up to date listing of the SD/MMC cards that have been tested as well as

information on the availability of drivers if required.

Table 27.   SD/MMC Cards Tested

Manufacturer	Type	Part Number	Status
Patriot	SD	1GB	Tested
Microcenter	SD	1GB/2GB	Tested

10.7   USB to Ethernet

There are several USB to Ethernet adapters on the market and **Figure 62** shows a few of these devices. These devices can easily add Ethernet connectivity to BeagleBoard by using the USB OTG port in the host. This will require a special cable to convert the miniAB connector to a Type A or a hub can also be used. These are provided as examples only. Check BeagleBoard.org for information on devices that have drivers available for them.

Figure 62.   USB to Ethernet Adapters

Table 28 provides examples of USB to Ethernet Adapters that **might** be used with the BeagleBoard. This list has not been verified. Inclusion of these items in the table does not guarantee that they will work, but is provided as examples only. Please check BeagleBoard.org for an up to date listing of the USB to Ethernet devices as well as information on the availability of drivers.

Table 28.   USB to Ethernet Adapters

Product	Manufacturer	Status
ASOHOUSB	Airlink	Not Tested
TU-ET100C 10/100Mbps	TRENDnet	Not Tested
SABRENT	NB-USB20	Not Tested
Zonet	ZUN2210	Not Tested
StarTech	USB2105S	Not Tested

MOSCHIP is the silicon provider for USB to Ethernet devices. The product that has been tested uses the 7830 from MOSCHIP and has a vendor ID of 9710 and a product ID of 7830. The devices above that are based upon the MOSCHIP device are highlighted in red.

10.8   USB to WiFi

There are several USB to WiFi adapters on the market and **Figure 63** shows a few of these devices. These devices can easily add WiFi connectivity to BeagleBoard by using the USB OTG port in the host mode. This will require a special cable to convert the miniAB connector to a Type A or a hub can also be used. These are provided as examples only. Check BeagleBoard.org for information on devices that have drivers available for them.

Figure 63.   USB to WiFi

Table 29 provides a list of USB to WiFi adapters that could be used with the BeagleBoard. Inclusion of these items in the table does not guarantee that they will work, but is provided as examples only. Please check BeagleBoard.org for an up to date listing of the USB to Wifi devices as well as information on the availability of drivers.

Table 29.   USB to WiFI Adapters

Product	Manufacturer	Status
4410-00-00AF	Zoom	Not Tested
HWUG1	Hawkins	Not Tested
TEW-429Uf	Trendnet	Not Tested

It should be noted that the availability of Linux drivers for various WiFi devices is limited. Before purchasing a particular device, please verify the availability of drivers for that device.

10.9   USB to Bluetooth

There are several USB to Bluetooth adapters on the market and **Figure 64** shows a few of these devices. These devices can easily add Bluetooth connectivity to BeagleBoard by using the USB OTG port in the host mode. This will require a special cable to convert the miniAB connector to a Type A or a hub can also be used. These are provided as examples only. Check BeagleBoard.org for information on devices that have drivers available for them and their test status.

Figure 64. USB to Bluetooth

Table 30 provides a list of USB to Bluetooth adapters that could be used with the BeagleBoard. Inclusion of these items in the table does not guarantee that they will work, but is provided as examples only. Please check BeagleBoard.org for an up to date listing of the USB to Bluetooth devices as well as information on the availability of drivers.

Table 30. USB to Bluetooth Adapters

Product	Manufacturer
TBW-105UB	Trendnet
ABT-200	Airlink
F8T012-1	Belkin

10.10 Expansion Cards

This section is reserved for future use as expansion cards are released for the BeagleBoard by various sources.

11.0   Mechanical Information

11.1   BeagleBoard Dimensions

This section provides information on the mechanical aspect of the BeagleBoard. **Figure 65** is the dimensions of the BeagleBoard.
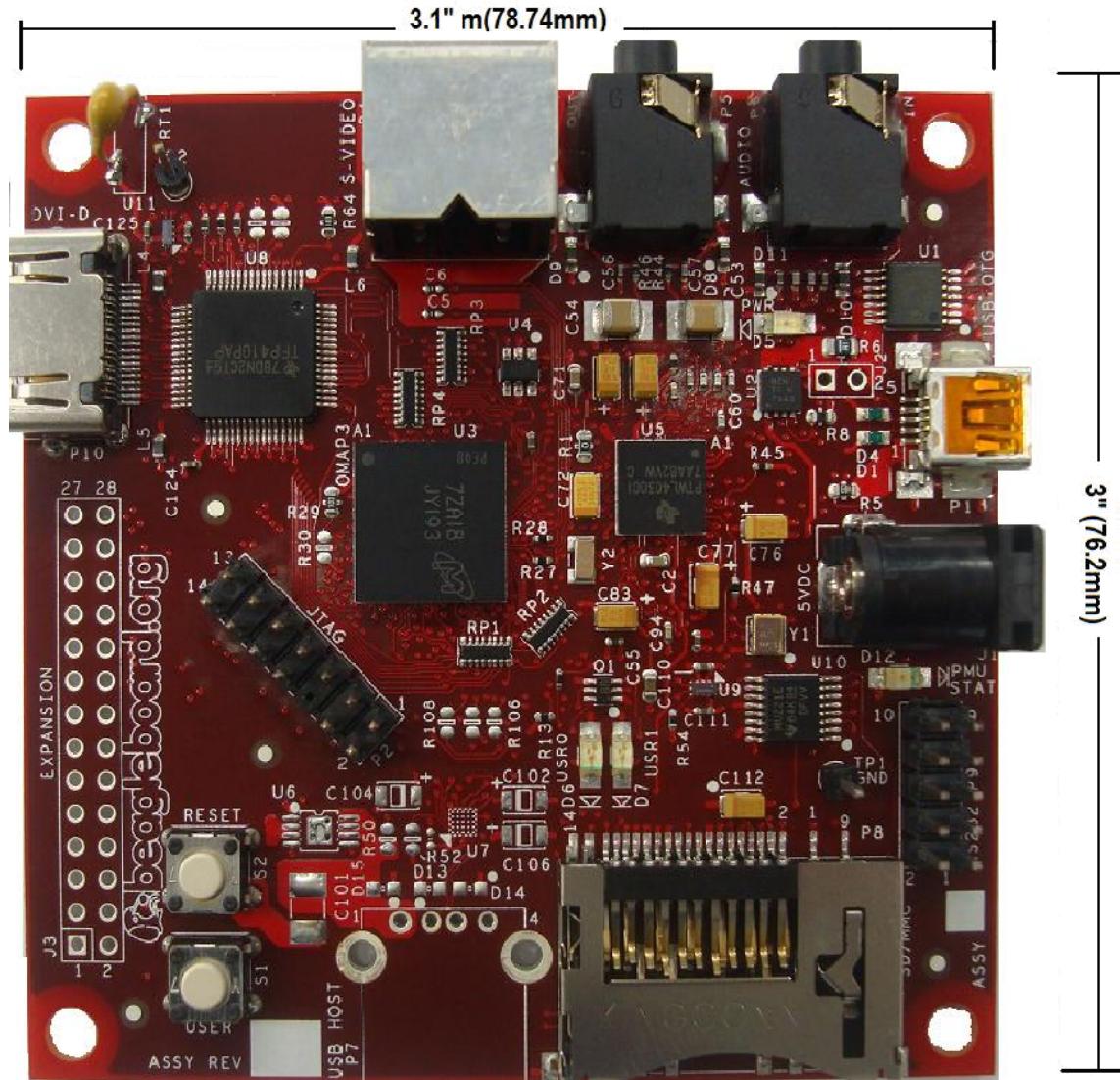
Figure 65.   BeagleBoard Dimension Drawing

11.2   BeagleBoard Daughter Card Information

This section provides information on what is required from a mechanical aspect to create a daughter card for the BeagleBoard. Users are free to create their own cards for private or commercial use. The concept of a standard card size for these cards is not being set down. The examples provided in this section show what is possible and is provided as a starting point. The idea is not to limit the possibilities.

11.2.1   Stacked Daughtercard Card

One method is to provide a daughtercard stacked onto the board. This can be either mounted on top of the BeagleBoard or under the BeagleBoard. **Figure 66** and **Figure 6**7 shows these two strategies.
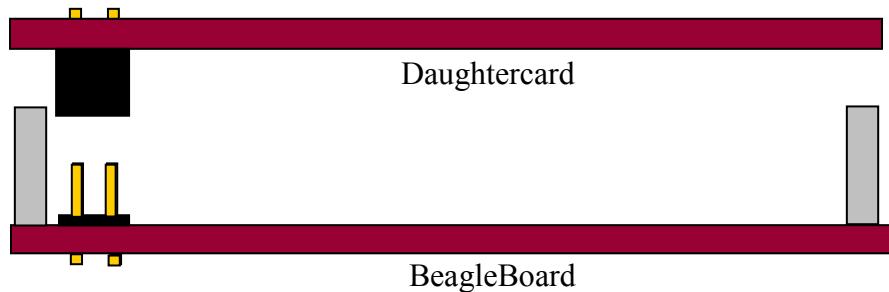
Figure 66.   BeagleBoard Top Stacked Daughter Card
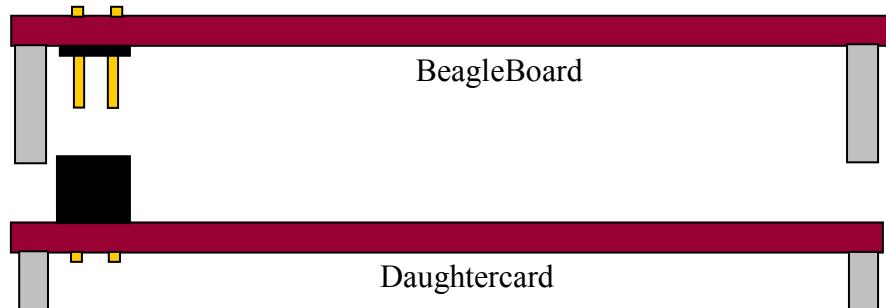
Figure 67.   BeagleBoard Bottom Stacked Daughter Card

11.2.2   Offset Daughter Card Information

Another option is to create a daughtercard that plugs in from the side. **Figure 68** and **Figure 69** shows the offset daughtercard using a board to board connector system.

Figure 68.   BeagleBoard Offset Daughter Card Side
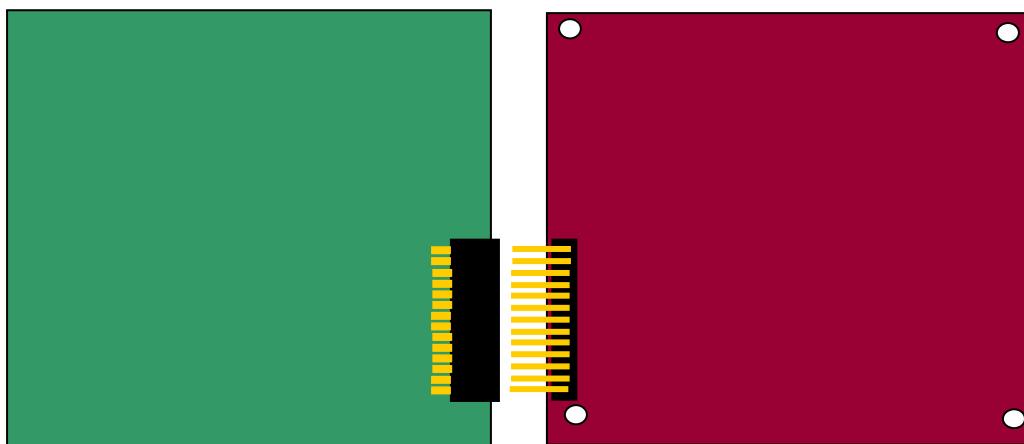
Figure 69.   BeagleBoard Offset Daughter Card Top

11.2.3   Ribbon Cable Daughter Card Information

Another method is to use a ribbon cable to connect the two boards together. **Figure 70** shows this concept.
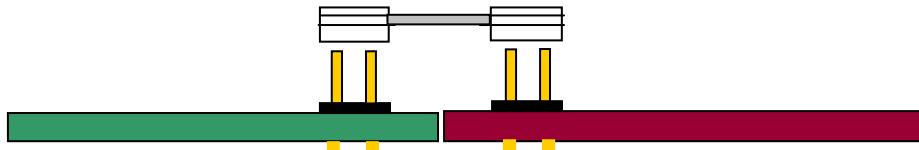
Figure 70.   Ribbon Cable Daughter Card

12.0   Board Verification

This section provides a step by step process to be followed to verify that the hardware is working. This is the same basic process the board is taken through in production testing.

For an up to date listing of common questions and their answers, please refer to
http://elinux.org/BeagleBoardFAQ

12.1   Equipment

To run these tests you will need the following components:

- o   BeagleBoard
- o   5V DC supply with a 2.1mm I.D. and 5.5mm O.D. connector
- o   SD Card
- o   PC
- o   USB miniA to A cable
- o   USB HUB
- o   DVI-D Monitor
- o   DVI-D to HDMI cable
- o   Speakers
- o   3.5mm stereo cable with connectors on both ends
- o   DB9 Null-Modem Cable
- o   DB9 to IDC-10 cable ATI/Everex configuration

12.2   Out of the Box

Each BeagleBoard comes pre-loaded with the XLoader and UBoot in Flash. When powered up, it will do the following:

1. Plug in either a USB cable to the board and then to a PC or plug in a 5V power supply.
2. Power LED (D5) will turn on.
3. On the terminal window the following will be printed:

   Texas Instruments X-Loader 1.41
   Starting OS Bootloader...

   U-Boot 1.3.3 (Jul 10 2008 - 16:33:09)

   OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
   OMAP3 Beagle Board + LPDDR/NAND
   DRAM: 128 MB
   NAND: 256 MiB

   In: serial

Out: serial
Err: serial

4. At this point the following LEDS will turn on:

   o User0
   o USR1
   o PMU

5. Then the following will be sent to the terminal window and you will hear a tone from the left speaker:

Audio Tone on Speakers ... complete
OMAP3 beagleboard.org #

6. The BeagleBoard.org log will be sent out of the DVI port and the color bars will appear on the S-Video Port.

12.3   SD Card Configuration

In order to boot from the SD card, it must be formatted and the files loaded. The following steps explain that process.

1. Format the MMC/SD Card for FAT32 File System using the HP USB Disk Storage Format Tool 2.0.6: http://selfdestruct.net/misc/usbboot/SP27213.exe
2. Insert the Card writer/reader into the Windows machine.
3. Insert MMC/SD card into the card reader/writer
4. Open the HP USB Disk Storage Format Tool.
5. Select "FAT32 as File System".
6. Click on "Start".
7. After formatting is done Click "OK"
8. Copy the following files on to MMC in the exact order listed. COPY THE MLO FIRST! Make sure you name the file as indicated in the **BOLD** type.

   o MLO as **MLO**
   o x-load.bin.ift for NAND as **x-load.bin.ift**
   o u-boot to flash onto NAND as **flash-uboot.bin**
   o u-boot.bin for MMC boot as **u-boot.bin**
   o ramdisk Image as **rd-ext2.bin**
   o Kernel (uImage) as **uImage**
   o Sample Video File as **HARRY.YUV**

12.4   Setup

This step sets up the board for the tests to follow.

1. Make sure Beagle power is in OFF state by removing the 5V supply and the USB host connection.
2. Connect the IDC UART cable the BeagleBoard and using a Null-Modem serial cable, connect it to a SERIAL port on a Window/Linux/Mac machine.
3. Have a terminal program, such as TeraTerm, HyperTerminal, or Minicom, running on the host machine.
4. Configure the terminal program for (BAUD RATE - 115200, DATA - 8 bit, PARITY- none, STOP - 1bit, FLOW CONTROL - none)
5. Insert the MMC/SD card (that is prepared as described above) into MMC/SD slot on Beagle Board.
6. Connect a LCD Monitor to DVI/HDMI port on Beagle Board.
7. Connect an externally powered speaker to audio out jack on Beagle Board.
8. Connect a Line-in cable from PC or any player to Audio In jack on Beagle Board.
9. Connect a TV (NTSC-M) to S-video port.
10. Power ON LCD, TV and audio speakers.
11. Connect one end of USB mini B to USB A cable to Beagle board. Do NOT **connect the USB A side to Host machine yet.**
12. If you have Windows PC as a host machine, copy the Linux.inf RNDIS driver configuration file to Host machine. Generally windows will have one.

12.5   Factory Boot Verification

The BeagleBoard comes pre-Flashed with the Xloader and UBoot in Flash. This step verifies that the board will boot properly. If the board has been flashed and the default code removed, then you should proceed to the next step.

1. Connect the USB cable to the Host PC,
2. The power LED should come on.
3. On the terminal window the following should be printed out by the BeagleBoard:

Texas Instruments X-Loader 1.41
Starting OS Bootloader...

U-Boot 1.3.3 (Jul 10 2008 - 16:33:09)

OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
OMAP3 Beagle Board + LPDDR/NAND
DRAM: 128 MB
NAND: 256 MiB

In: serial
Out: serial
Err: serial
Audio Tone on Speakers ... complete

OMAP3 beagleboard.org #

4. You should hear a tone out of the left channel of your speakers.
5. The USER LEDS and the PMU LED should be on.
6. The S-Video output should display color bars.
7. The DVI-D monitor should display the BeagleBoard.org Logo.

12.6 Board SD Boot

This test will force the BeagleBoard to boot from the SD card instead of the onboard
Flash.

1. Press and hold the USER button while pressing and releasing the RESET button.
2. The following should be printed to the terminal window:

> *40T*
>
> *Texas Instruments X-Loader 1.41*
> *Starting on with MMC*
> *Reading boot sector*
>
> *717372 Bytes Read from MMC*
> *Starting OS Bootloader from MMC...*
>
>
> *U-Boot 1.3.3 (Jul 10 2008 - 16:30:47)*
>
> *OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz*
> *OMAP3 Beagle Board + LPDDR/NAND*
> *DRAM: 128 MB*
> *NAND: 256 MiB*
> **** Warning - bad CRC or NAND, using default environment*
>
> *In: serial*
> *Out: serial*
> *Err: serial*
> *Audio Tone on Speakers ... complete*
> *Hit any key to stop autoboot:* 0

3. You should hear a tone out of the left channel of your speakers.
4. The USER LEDS and the PMU LED should be on.

12.7 Factory Boot Reinstall

This section tells you how to restore the information in the Flash to the factory default.
This is the same test that is run in production when the board is new. The erase process is

not required in this case because the Flash is clean. It is required if the Flash has previously been flashed with data.

1. At the OMAP3 beagleboard.org # prompt type: *NAND Unlock*
2. The board should print out the following:

> *device 0 whole chip*
> *nand_unlock: start: 00000000, length: 268435456!*
> *NAND flash successfully unlocked*
> *OMAP3 beagleboard.org #*

3. Then type: *nand ecc hw*
4. Then type : *nand erase 0 0x3f0000*
5. The BeagleBoard will print the following to the terminal window:

> *NAND erase: device 0 offset 0x0, size 0x3f0000*
>
> *Erasing at 0x0 --   3% complete.*
> *Erasing at 0x20000 --   6% complete.*
> *Erasing at 0x40000 --   9% complete.*
> *Erasing at 0x60000 -- 12% complete.*
> *Erasing at 0x80000 -- 15% complete.*
> *Erasing at 0xa0000 -- 19% complete.*
> *Erasing at 0xc0000 -- 22% complete.*
> *Erasing at 0xe0000 -- 25% complete.*
> *Erasing at 0x100000 -- 28% complete.*
> *Erasing at 0x120000 -- 31% complete.*
> *Erasing at 0x140000 -- 34% complete.*
> *Erasing at 0x160000 -- 38% complete.*
> *Erasing at 0x180000 -- 41% complete.*
> *Erasing at 0x1a0000 -- 44% complete.*
> *Erasing at 0x1c0000 -- 47% complete.*
> *Erasing at 0x1e0000 -- 50% complete.*
> *Erasing at 0x200000 -- 53% complete.*
> *Erasing at 0x220000 -- 57% complete.*
> *Erasing at 0x240000 -- 60% complete.*
> *Erasing at 0x260000 -- 63% complete.*
> *Erasing at 0x280000 -- 66% complete.*
> *Erasing at 0x2a0000 -- 69% complete.*
> *Erasing at 0x2c0000 -- 73% complete.*
> *Erasing at 0x2e0000 -- 76% complete.*
> *Erasing at 0x300000 -- 79% complete.*
> *Erasing at 0x320000 -- 82% complete.*
> *Erasing at 0x340000 -- 85% complete.*
> *Erasing at 0x360000 -- 88% complete.*
> *Erasing at 0x380000 -- 92% complete.*
> *Erasing at 0x3a0000 -- 95% complete.*
> *Erasing at 0x3c0000 -- 98% complete.*
> *Erasing at 0x3e0000 -- 101% complete.*
> *OK*
> *OMAP3 beagleboard.org #*

6. At this point the Flash is now erased.

7. Insert the SD card that was created earlier in this section.
8. Press and release the RESET button on the board.
9. The board will boot form the SD card and reprogram the flash. The board will print the following to the terminal window.

Texas Instruments X-Loader 1.41
Starting on with MMC
Reading boot sector

717372 Bytes Read from MMC
Starting OS Bootloader from MMC...

U-Boot 1.3.3 (Jul 10 2008 - 16:30:47)

OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
OMAP3 Beagle Board + LPDDR/NAND
DRAM: 128 MB
NAND: 256 MiB

*** Warning - bad CRC or NAND, using default environment

In: serial
Out: serial
Err: serial
Audio Tone on Speakers ... complete
Hit any key to stop autoboot: 3
reading x-load.bin.ift

9808 bytes read
device 0 whole chip
nand_unlock: start: 00000000, length: 268435456!
NAND flash successfully unlocked

NAND erase: device 0 offset 0x0, size 0x80000

Erasing at 0x0 -- 25% complete.
Erasing at 0x20000 -- 50% complete.
Erasing at 0x40000 -- 75% complete.
Erasing at 0x60000 -- 100% complete.
OK

NAND write: device 0 offset 0x0, size 0x80000

Writing data at 0x0 --   0% complete.
Writing data at 0x1000 --   1% complete.
Writing data at 0x2800 --   2% complete.
Writing data at 0x3800 --   3% complete.
Writing data at 0x5000 --   4% complete.
Writing data at 0x6000 --   5% complete.
Writing data at 0x7800 --   6% complete.
Writing data at 0x8800 --   7% complete.

Writing data at 0xa000 --   8% complete.
Writing data at 0xb800 --   9% complete.
Writing data at 0xc800 -- 10% complete.
Writing data at 0xe000 -- 11% complete.
Writing data at 0xf000 -- 12% complete.
                    :
                    :
                    :
Writing data at 0x7e800 -- 99% complete.
Writing data at 0x7f800 -- 100% complete.
 524288 bytes written: OK
reading flash-uboot.bin

717116 bytes read
device 0 whole chip
nand_unlock: start: 00000000, length: 268435456!
NAND flash successfully unlocked

NAND erase: device 0 offset 0x80000, size 0x160000

Erasing at 0x80000 --   9% complete.
Erasing at 0xa0000 -- 18% complete.
Erasing at 0xc0000 -- 27% complete.
Erasing at 0xe0000 -- 36% complete.
Erasing at 0x100000 -- 45% complete.
Erasing at 0x120000 -- 54% complete.
Erasing at 0x140000 -- 63% complete.
Erasing at 0x160000 -- 72% complete.
Erasing at 0x180000 -- 81% complete.
Erasing at 0x1a0000 -- 90% complete.
Erasing at 0x1c0000 -- 100% complete.
OK

NAND write: device 0 offset 0x80000, size 0x160000

Writing data at 0x80000 --   0% complete.
Writing data at 0x83800 --   1% complete.
                    :
                    :
                    :
                    :
Writing data at 0x1d8800 -- 98% complete.
Writing data at 0x1dc000 -- 99% complete.
Writing data at 0x1df800 -- 100% complete.
 1441792 bytes written: OK
OMAP3 beagleboard.org #

10. Press and release the RESET button and insure that the board reboots to the
BeagleBoard.org# prompt.

12.8   Setting Up The Boot Parameters

In order to boot the Linux Kernel, certain parameters need to be placed in Flash. This section describes the commands to set this up.

1. Type the following command: setenv bootargs console=ttyS2,115200n8 ramdisk_size=8192 root=/dev/ram0 rw rootfstype=ext2 initrd=0x81600000,8M nohz=0ff
2. Type the following command next: setenv bootcmd 'mmcinit;fatload mmc 0 0x80300000 uImage;fatload mmc 0 0x81600000 rd-ext2.bin;bootm 0x80300000';
3. To save these commands in Flash, type : saveenv

12.9   Booting the Kernel

This section describes how to boot the kernel from the SD card. In order to complete this section, you must have completed section 12.7.

1. At the BeagleBoard.org# prompt type *run bootcmd*
2. The BeagleBoard will output the following:

> *reading uImage*
>
> *1856680 bytes read*
> *reading rd-ext2.bin*
>
> *3394477 bytes read*

3. At this point, the booting process will start. The following will be printed to the terminal:

> *## Booting kernel from Legacy Image at 8000 ...*
> *   Imageme: Linux-2.6.18-omap3*
> *   Image Type: ARM Linux Kernel Image (uncompressed)*
>
> *   Data Size: 1856616 Bytes = 1.8 MB*
> *   Load Address: 80008000*
> *   Entry Point: 80008000*
>
> *   Verifying Checksum ... OK*
> *   Loading Kernel Image ... OK*
> *OK*
>
> *Starting kernel ...*
>
>
> *Uncompressing*
> *Linux...*
> *.. done, booting the kernel.*
> *<5>Linux version 2.6.22.18-omap3 (root@fedoraserver) (gcc version 4.2.1*
> *(CodeSourcery Sourcery G++ Lite 2007q3-51)) #1 Thu Jul 24 15:29:36 IST 2008*

CPU: ARMv7 Processor [411fc082] revision 2 (ARMv7), cr=00c5387f

Machine: OMAP3 Beagle board

Memory policy: ECC disabled, Data cache writeback

<7>On node 0 totalpages: 32768

<7> DMA zone: 256 pages used for memmap

<7> DMA zone: 0 pages reserved

<7> DMA zone: 32512 pages, LIFO batch:7

<7> Normal zone: 0 pages used for memmap

<6>OMAP3430ES1

<6>SRAM: Mapped pa 0x40200000 to va 0xd7000000 size: 0x100000

CPU0: D VIPT write-through cache

CPU0: cache: 768 bytes, associativity 1, 8 byte lines, 64 sets

Built 1 zonelists. Total pages: 32512

*<5>Kernel command line: console=ttyS2,115200n8 ramdisk_size=8192
root=/dev/ram0 rw rootfstype=ext2 initrd=0x81600000,8M nohz=0ff*

<6>GPMC revision 5.0

<6>IRQ: Found an INTC at 0xd8200000 (revision 4.0) with 96 interrupts

<6>Total of 96 interrupts on 1 active controller

<6>OMAP34xx GPIO hardware version 2.5

PID hash table entries: 512 (order: 9, 2048 bytes)

Console: colour dummy device 80x30

Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)

Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)

<6>Memory: 128MB 0MB = 128MB total

<5>Memory: 117760KB available (3496K code, 312K data, 132K init)

<7>Calibrating delay loop... 499.92 BogoMIPS (lpj=1949696)

Mount-cache hash table entries: 512

<6>CPU: Testing write buffer coherency: ok

<6>NET: Registered protocol family 16

<3>TWL4030: I2C Client[3] is not initialized[515]

<3>TWL4030: I2C Client[3] is not initialized[464]

<6>SmartReflex driver initialized
<6>OMAP DMA hardware revision 4.0
<6>OMAP Display hardware version 2.0
<6>i2c_omap i2c_omap.1: bus 1 rev3.12 at 2600 kHz
<6>i2c_omap i2c_omap.2: bus 2 rev3.12 at 100 kHz
<6>i2c_omap i2c_omap.3: bus 3 rev3.12 at 400 kHz
<6>TWL4030: TRY attach Slave TWL4030-ID0 on Adapter OMAP I2C adapter [1]
<6>TWL4030: TRY attach Slave TWL4030-ID1 on Adapter OMAP I2C adapter [1]
<6>TWL4030: TRY attach Slave TWL4030-ID2 on Adapter OMAP I2C adapter [1]
<6>TWL4030: TRY attach Slave TWL4030-ID3 on Adapter OMAP I2C adapter [1]
<6>TWL4030 Power Companion Active
<6><6>TWL4030: Driver registration complete.
<6>TWL4030 GPIO Demux: IRQ Range 376 to 386, Initialization Success
<6>Initialized TWL4030 USB module<5>SCSI subsystem initialized
<6>usbcore: registered new interface driver usbfs
<6>usbcore: registered new interface driver hub
<6>usbcore: registered new device driver usb
<6>musb_hdrc: version 2.2a/db-0.5.2, pio, otg (peripheral+host), debug=0
<7>musb_hdrc: ConfigData=0xde (UTMI-8, dyn FIFOs, bulk combine (X), bulk split (X), HB-ISO Rx (X), HB-ISO Tx (X), SoftConn)
<7>musb_hdrc: MHDRC RTL version 1.400
<7>musb_hdrc: setup fifo_mode 4
<7>musb_hdrc: 29/31 max ep, 15424/16384 memory
<7>musb_hdrc: hw_ep 0shared, max 64
<7>musb_hdrc: hw_ep 1tx, max 512
<7>musb_hdrc: hw_ep 1rx, max 512
<7>musb_hdrc: hw_ep 2tx, max 512
<7>musb_hdrc: hw_ep 2rx, max 512
<7>musb_hdrc: hw_ep 3tx, max 512
<7>musb_hdrc: hw_ep 3rx, max 512
<7>musb_hdrc: hw_ep 4tx, max 512
<7>musb_hdrc: hw_ep 4rx, max 512
<7>musb_hdrc: hw_ep 5tx, max 512
<7>musb_hdrc: hw_ep 5rx, max 512
<7>musb_hdrc: hw_ep 6tx, max 512
<7>musb_hdrc: hw_ep 6rx, max 512
<7>musb_hdrc: hw_ep 7tx, max 512
<7>musb_hdrc: hw_ep 7rx, max 512
<7>musb_hdrc: hw_ep 8tx, max 512
<7>musb_hdrc: hw_ep 8rx, max 512
<7>musb_hdrc: hw_ep 9tx, max 512
<7>musb_hdrc: hw_ep 9rx, max 512
<7>musb_hdrc: hw_ep 10tx, max 512
<7>musb_hdrc: hw_ep 10rx, max 512

<7>musb_hdrc: hw_ep 11tx, max 512
<7>musb_hdrc: hw_ep 11rx, max 512
<7>musb_hdrc: hw_ep 12tx, max 512
<7>musb_hdrc: hw_ep 12rx, max 512
<7>musb_hdrc: hw_ep 13tx, max 512
<7>musb_hdrc: hw_ep 13rx, max 512
<7>musb_hdrc: hw_ep 14shared, max 1024
<7>musb_hdrc: hw_ep 15shared, max 1024
<6>musb_hdrc: USB OTG mode controller at c8800000 using PIO, IRQ 92
<6>Time: 32k_counter clocksource has been installed.
<6>Switched to high resolution mode on CPU 0
<6>NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
<6>TCP: Hash tables configured (established 4096 bind 4096)
<6>TCP reno registered
<6>checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
<6>Freeing initrd memory: 8192K
<3>Power Management for TI OMAP.
<6>cpuidle: using driver omap3_idle
<3>create_proc_entry succeeded
<3>create_proc_entry succeeded
<4>NetWinder Floating Point Emulator V0.97 (double precision)
<5>VFS: Disk quotas dquot_6.5.1
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
<6>JFFS2 version 2.2. (NAND) Â© 2001-2006 Red Hat, Inc.
<6>io scheduler noop registered
<6>io scheduler anticipatory registered (default)
<6>io scheduler deadline registered
<6>io scheduler cfq registered
<7>omap2_disp_outLCD panel 1024x768
<4>timeout waiting for frame-done interrupt
<7>omap2_disp_outTV 640x480 interlaced
<6>omap24xxfb: Options "<NULL>"
Console: switching to colour frame buffer device 128x48
<6>omap24xxfb: fb0 frame buffer device
<6>omap24xxfb: display mode 1024x768x16 hsync 22kHz vsync
28Hz<6>omap_rng omap_rng: OMAP Random Number Generator ver. 70
<6>OMAP Watchdog Timer Rev 0x31: initial timeout 60 sec
<6>Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing enabled
<6>serial8250.0: ttyS0 at MMIO 0x4806a000 (irq = 72) is a ST16654
<6>serial8250.0: ttyS1 at MMIO 0x4806c000 (irq = 73) is a ST16654
<6>serial8250.0: ttyS2 at MMIO 0x49020000 (irq = 74) is a ST16654
Linux version 2.6.22.18-omap3 (root@fedoraserver) (gcc version 4.2.1
(CodeSourcery Sourcery G++ Lite 2007q3-51)) #1 Thu Jul 24 15:29:36 IST 2008
CPU: ARMv7 Processor [411fc082] revision 2 (ARMv7), cr=00c5387f
Machine: OMAP3 Beagle board
Memory policy: ECC disabled, Data cache writeback
OMAP3430ES1
SRAM: Mapped pa 0x40200000 to va 0xd7000000 size: 0x100000
CPU0: D VIPT write-through cache
CPU0: cache: 768 bytes, associativity 1, 8 byte lines, 64 sets
Built 1 zonelists. Total pages: 32512

Kernel command line: console=ttyS2,115200n8 ramdisk_size=8192
root=/dev/ram0 rw rootfstype=ext2 initrd=0x81600000,8M nohz=0ff
GPMC revision 5.0
IRQ: Found an INTC at 0xd8200000 (revision 4.0) with 96 interrupts
Total of 96 interrupts on 1 active controller
OMAP34xx GPIO hardware version 2.5
PID hash table entries: 512 (order: 9, 2048 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 128MB 0MB = 128MB total
Memory: 117760KB available (3496K code, 312K data, 132K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16
TWL4030: I2C Client[3] is not initialized[515]
TWL4030: I2C Client[3] is not initialized[464]
SmartReflex driver initialized
OMAP DMA hardware revision 4.0
OMAP Display hardware version 2.0
i2c_omap i2c_omap.1: bus 1 rev3.12 at 2600 kHz
i2c_omap i2c_omap.2: bus 2 rev3.12 at 100 kHz
i2c_omap i2c_omap.3: bus 3 rev3.12 at 400 kHz
TWL4030: TRY attach Slave TWL4030-ID0 on Adapter OMAP I2C adapter [1]
TWL4030: TRY attach Slave TWL4030-ID1 on Adapter OMAP I2C adapter [1]
TWL4030: TRY attach Slave TWL4030-ID2 on Adapter OMAP I2C adapter [1]
TWL4030: TRY attach Slave TWL4030-ID3 on Adapter OMAP I2C adapter [1]
TWL4030 Power Companion Active
<6>TWL4030: Driver registration complete.
TWL4030 GPIO Demux: IRQ Range 376 to 386, Initialization Success
Initialized TWL4030 USB module<5>SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
musb_hdrc: version 2.2a/db-0.5.2, pio, otg (peripheral+host), debug=0
musb_hdrc: USB OTG mode controller at c8800000 using PIO, IRQ 92
Time: 32k_counter clocksource has been installed.
Switched to high resolution mode on CPU 0
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP reno registered
checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
Freeing initrd memory: 8192K
Power Management for TI OMAP.
cpuidle: using driver omap3_idle
create_proc_entry succeeded
create_proc_entry succeeded
NetWinder Floating Point Emulator V0.97 (double precision)
VFS: Disk quotas dquot_6.5.1
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
JFFS2 version 2.2. (NAND) Â© 2001-2006 Red Hat, Inc.
io scheduler noop registered

io scheduler anticipatory registered (default)
io scheduler deadline registered
io scheduler cfq registered
timeout waiting for frame-done interrupt
omap24xxfb: Options "<NULL>"
Console: switching to colour frame buffer device 128x48
omap24xxfb: fb0 frame buffer device
omap24xxfb: display mode 1024x768x16 hsync 22kHz vsync 28Hz<6>omap_rng
omap_rng: OMAP Random Number Generator ver. 70
OMAP Watchdog Timer Rev 0x31: initial timeout 60 sec
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing enabled
serial8250.0: ttyS0 at MMIO 0x4806a000 (irq = 72) is a ST16654
serial8250.0: ttyS1 at MMIO 0x4806c000 (irq = 73) is a ST16654
serial8250.0: ttyS2 at MMIO 0x49020000 (irq = 74) is a ST16654
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize

RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
<6>loop: module loaded
loop: module loaded
<6>usbcore: registered new interface driver asix
usbcore: registered new interface driver asix
<6>usbcore: registered new interface driver cdc_ether
usbcore: registered new interface driver cdc_ether
<6>usbcore: registered new interface driver net1080
usbcore: registered new interface driver net1080
<6>usbcore: registered new interface driver cdc_subset
usbcore: registered new interface driver cdc_subset
<6>usbcore: registered new interface driver zaurus
usbcore: registered new interface driver zaurus
<6>Linux video capture interface: v2.00
Linux video capture interface: v2.00
<6>omap24xxvout: registered device video1 [v4l2]
omap24xxvout: registered device video1 [v4l2]
<6>omap24xxvout: registered device video2 [v4l2]
omap24xxvout: registered device video2 [v4l2]
<6>sn9c102: V4L2 driver for SN9C1xx PC Camera Controllers v1:1.47pre49
sn9c102: V4L2 driver for SN9C1xx PC Camera Controllers v1:1.47pre49
<6>usbcore: registered new interface driver sn9c102
usbcore: registered new interface driver sn9c102
<6>i2c /dev entries driver
i2c /dev entries driver
<6>omap2-nand driver initializing
omap2-nand driver initializing
<6>NAND device: Manufacturer ID: 0x2c, Chip ID: 0xba (Micron NAND 256MiB 1,8V 16-bit)
NAND device: Manufacturer ID: 0x2c, Chip ID: 0xba (Micron NAND 256MiB 1,8V 16-bit)
<5>Creating 5 MTD partitions on "omap2-nand.0":

Creating 5 MTD partitions on "omap2-nand.0":
<5>0x00000000-0x00080000 : "X-Loader-NAND"
0x00000000-0x00080000 : "X-Loader-NAND"
<5>0x00080000-0x00260000 : "U-Boot-NAND"
0x00080000-0x00260000 : "U-Boot-NAND"
<5>0x00260000-0x00280000 : "Boot Env-NAND"

0x00260000-0x00280000 : "Boot Env-NAND"
<5>0x00280000-0x00780000 : "Kernel-NAND"
0x00280000-0x00780000 : "Kernel-NAND"
<5>0x00780000-0x10000000 : "File System - NAND"
0x00780000-0x10000000 : "File System - NAND"
<5>usbmon: debugfs is not available
usbmon: debugfs is not available
<6>usbcore: registered new interface driver cdc_acm
usbcore: registered new interface driver cdc_acm
<6>drivers/usb/class/cdc-acm.c: v0.25:USB Abstract Control Model driver for USB modems and ISDN adapters
drivers/usb/class/cdc-acm.c: v0.25:USB Abstract Control Model driver for USB modems and ISDN adapters
<6>Initializing USB Mass Storage driver...
Initializing USB Mass Storage driver...
<6>usbcore: registered new interface driver usb-storage
usbcore: registered new interface driver usb-storage
<6>USB Mass Storage support registered.
USB Mass Storage support registered.
<4>ether gadget: using random self ethernet address
ether gadget: using random self ethernet address
<4>ether gadget: using random host ethernet address
ether gadget: using random host ethernet address
<6>usb0: Ethernet Gadget, version: May Day 2005
usb0: Ethernet Gadget, version: May Day 2005
<6>usb0: using musb_hdrc, OUT ep1out IN ep1in STATUS ep2in
usb0: using musb_hdrc, OUT ep1out IN ep1in STATUS ep2in
<6>usb0: MAC 46:f1:66:40:f2:85
usb0: MAC 46:f1:66:40:f2:85
<6>usb0: HOST MAC ea:e6:16:d9:84:12
usb0: HOST MAC ea:e6:16:d9:84:12
<6>usb0: RNDIS ready
usb0: RNDIS ready
<6>musb_hdrc musb_hdrc.0: MUSB HDRC host driver
musb_hdrc musb_hdrc.0: MUSB HDRC host driver
<6>musb_hdrc musb_hdrc.0: new USB bus registered, assigned bus number 1
musb_hdrc musb_hdrc.0: new USB bus registered, assigned bus number 1
<6>usb usb1: configuration #1 chosen from 1 choice
usb usb1: configuration #1 chosen from 1 choice
<6>hub 1-0:1.0: USB hub found
hub 1-0:1.0: USB hub found
<6>hub 1-0:1.0: 1 port detected
hub 1-0:1.0: 1 port detected
<6>mice: PS/2 mouse device common for all mice
mice: PS/2 mouse device common for all mice
<6>usbcore: registered new interface driver usbhid
usbcore: registered new interface driver usbhid
<6>drivers/hid/usbhid/hid-core.c: v2.6:USB HID core driver

drivers/hid/usbhid/hid-core.c: v2.6:USB HID core driver
<6>Advanced Linux Sound Architecture Driver Version 1.0.14 (Thu May 31 09:03:25 2007 UTC).
Advanced Linux Sound Architecture Driver Version 1.0.14 (Thu May 31 09:03:25 2007 UTC).

<6>OMAP3430 TWL4030 Audio Support: OMAP3430 TWL4030 Audio Support:
Chip Rev[0x2f] Initialized
Chip Rev[0x2f] Initialized
<6>audio support initialized
audio support initialized
<6>usbcore: registered new interface driver snd-usb-audio
usbcore: registered new interface driver snd-usb-audio
<6>ALSA device list:
ALSA device list:
<6> #0: TWL4030
* #0: TWL4030*
<6>TCP cubic registered
TCP cubic registered
<6>NET: Registered protocol family 1
NET: Registered protocol family 1
<6>NET: Registered protocol family 17
NET: Registered protocol family 17
<6>NET: Registered protocol family 15
NET: Registered protocol family 15
<6>VFP support v0.3: VFP support v0.3: implementor 41 architecture 3 part 30
variant c rev 1
implementor 41 architecture 3 part 30 variant c rev 1
<6>cpuidle: using governor menu
cpuidle: using governor menu
<5>RAMDISK: Compressed image found at block 0
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem).
VFS: Mounted root (ext2 filesystem).
<6>Freeing init memory: 132K
Freeing init memory: 132K
init started: BusyBox v1.9.0 (2008-02-26 15:40:22 IST)
starting pid 261, tty '': '/etc/init.d/rcS'
<4>mmc0: host does not support reading read-only switch. assuming write-
enable.
mmc0: host does not support reading read-only switch. assuming write-enable.
<6>mmcblk0: mmc0:b368 SD   1997312KiB
mmcblk0: mmc0:b368 SD   1997312KiB
<6> mmcblk0: mmcblk0: p1 p1

*   beagleboard.org (v0.90) : System initialization...*

*   Kernel release : Linux 2.6.22.18-omap3*
*   Kernel version : #1 Thu Jul 24 15:29:36 IST 2008*

Mounting /proc   : [SUCCESS]
Mounting /sys   : [SUCCESS]
Mounting /dev   : [SUCCESS]
Mounting /dev/pts   : [SUCCESS]
Enabling hot-plug   : [SUCCESS]
Populating /dev   : [SUCCESS]
Mounting other filesystems : [SUCCESS]
Starting syslogd   : Jan 1 00:00:06 beagleboard syslog.info syslogd
started: BusyBox v1.9.0
[SUCCESS]

Starting telnetd   : [SUCCESS]

System initialization complete.

Please press Enter to activate this console.

4. Hit the Enter key and you will have an active Linux console.

12.10   OTG Peripheral Test

This test requires that the board be powered from the USB and connected to a Host PC. This uses the ping test to verify connection to the Host PC. Described below are two sections. The first section is required if this is the first time that the BeagleBoard has been connected to the host. If this has already been done, then you can skip to the next section which covers the actual test.

12.10.1 Loading Host Driver

This section covers the procedure for connecting to a Windows based PC.

1. When the BeagleBoard connects to the PC it will detect a new device and ask for the driver. Select the Linux.inf as the driver, if you don't find it automatically, then you can copy one from Linux Kernel Source folder (2.6_kernel_revb-v2.tar.gz), from the path "2.6_kernel/Documentation/usb/linux.inf"

2. On the Windows PC, bring up Network Connections
3. Look for the Device Name: Linux USB Ethernet/RNDIS Gadget.
4. Right-mouse click for Properties.
5. Scroll down to Internet Protocol (TCP/IP), select it
6. Press the Properties button.
7. Select an IP address close to the one selected for the Beagle Board. This example uses an IP address of 192.168.1.5 and a Subnet mask of 255.255.255.0.

12.10.2 Running the Test

This runs the ping test on the BeagleBoard.

1. Mount the SD card by typing:

    [root@beagleboard /]# mount -t vfat /dev/mmcblk0p1 /mnt/mmc/

2. Change the directory by typing:

    [root@beagleboard /]# cd /mnt/mmc/

3. Configure a static IP address for Beagle Board with the ifconfig command. The example configures an IP address of 192.168.1.1 with a subnet mask of 255.255.255.0. Type:

[root@beagleboard mmc]# ifconfig usb0 192.168.1.1 netmask 255.255.255.0

4. On the terminal emulator connected to the Beagle Board use ping to test the connection. Type the following:

[root@beagleboard mmc]# ping 192.168.1.5

5. The Beagle Board will print the following to the terminal window:

PING 192.168.1.5 (192.168.1.5): 56 data bytes
64 bytes from 192.168.1.5: seq=0 ttl=128 time=0.885 ms
64 bytes from 192.168.1.5: seq=1 ttl=128 time=0.977 ms
64 bytes from 192.168.1.5: seq=2 ttl=128 time=0.977 ms

6. To stop the test type a <CTRL-C> character. The BeagleBoard should print the following to the terminal window:

--- 192.168.1.5 ping statistics ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 0.885/0.969/1.679 ms
[root@beagleboard mmc]#

The packet count that will be displayed will reflect the actual number of packers that were sent and received. This will vary depending on how long you wait before stopping the test.

7. To close the test type the following:

 [root@beagleboard mmc]#ifconfig usb0 down

12.11   EDID Test

This test will display the EDID (Enhanced Display ID) from the DVI-D monitor by using the I2C interface on the DVI-D connector. This test will only run when in the UBoot mode which the board enters just after a reset or power up.

1. Type the following commands:
     OMAP3 beagleboard.org # ibus 2 0x64
     OMAP3 beagleboard.org# imd 0x50 0 100

2. Something similar to the following will be displayed:

     0000: 00 ff ff ff ff ff ff 00 10 ac 24 40 5a 39 41 41   ....$@Z9AA

```
0010: 1f 11 01 03 80 22 1b 78 ee ae a5 a6 54 4c 99 26   ....."x....TL.&
0020: 14 50 54 a5 4b 00 71 4f 81 80 01 01 01 01 01 01   .PT.K.qO........
0030: 01 01 01 01 01 01 30 2a 00 98 51 00 2a 40 30 70   ......0*..Q.*@0p
0040: 13 00 52 0e 11 00 00 1e 00 00 00 ff 00 50 4d 30   ..R..........PM0
0050: 36 31 37 38 32 41 41 39 5a 0a 00 00 00 fc 00 44   61782AA9Z......D
0060: 45 4c 4c 20 31 37 30 38 46 50 0a 20 00 00 00 fd   ELL 1708FP. ....
0070: 00 38 4c 1e 51 0e 00 0a 20 20 20 20 20 20 00 36   .8L.Q...     .6
```

Note the words "DELL 1708FP" which is the ID of the monitor in this example. It will be different based on the display manufacturer..

For more detailed information on the full EDID format, refer to:

http://en.wikipedia.org/wiki/EDID

12.12   DVI-D Test

This test plays a short video clip to the DVI-D monitor.

1. Make sure that you are in the MMC directory. If you are, proceed to step 3. If not, then type the following command:

    root@beagleboard /]# mount -t vfat /dev/mmcblk0p1 /mnt/mmc/cd /mnt/mmc/

2. Change the directory by typing:

    [root@beagleboard /]# cd /mnt/mmc/

3. Type the following command:

    [root@beagleboard mmc]# stream_video

4. It will display a 320x240 video of Stephane Edberg playing Tennis on DVI screen

12.13   S-Video Test

1. Make sure that you are in the MMC directory. If you are, proceed to step 3. If not, then type the following command:

    root@beagleboard /]# mount -t vfat /dev/mmcblk0p1 /mnt/mmc/cd /mnt/mmc/

2. Change the directory by typing:

    [root@beagleboard /]# cd /mnt/mmc/

3. Type the following command:

[root@beagleboard mmc]# echo 'tv' > /sys/class/ display_control/omap_disp_control/ video1

4. Type the following command to start the video:

[root@beagleboard mmc]# stream_video

5. Should display a 320x240 video of Stephane Edberg playing Tennis on TV

12.14 Audio Test

The audio test is divided into two test, one for audio in and one for audio out. Audio is recorded into the audio in port and then played out the audio out port.

12.14.1 Audio In

1. Make Sure your player is running and Audio Line in is connected to board.
2. Make sure that you are in the MMC directory. If you are, proceed to step 4. If not, then type the following command:

    root@beagleboard /]# mount -t vfat /dev/mmcblk0p1 /mnt/mmc/cd /mnt/mmc/

3. Change the directory by typing:

    [root@beagleboard /]# cd /mnt/mmc/

4. Type the following command:
    [root@beagleboard mmc]# arecord -t wav -c 2 -r 44100 -f S16_LE -v k

5. Following output is expected on the terminal window:

    Recording WAVE 'k' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
    Plug PCM: Hardware PCM card 0 'TWL4030' device 0 subdevice 0
    Its setup is:
      stream   : CAPTURE
      access   : RW_INTERLEAVED
      format   : S16_LE
      subformat   : STD
      channels   : 2
      rate   : 44100
      exact rate : 44100 (44100/1)
      msbits   : 16
      buffer_size : 32768
      period_size : 2048
      period_time : 46439
      tick_time   : 7812
      tstamp_mode : NONE
      period_step : 1

```
    sleep_min    : 0
    avail_min    : 2048
    xfer_align   : 2048
    start_threshold  : 1
    stop_threshold   : 32768
    silence_threshold: 0
    silence_size : 0
    boundary     : 1073741824
```

6. When you think you want to stop just press <CONTROL+C>.

12.14.2 Audio Out

<u>**NOTE**</u>: It is expected that you have previously recorded an audio file to be played.

1. Type the following command:
        root@beagleboard mmc]# aplay -t wav -c 2 -r 44100 -f S16_LE -v k

2. The recorded audio should be heard on the Speakers,

3. The following output is expected on terminal window:

```
        Playing WAVE 'k' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
        Plug PCM: Hardware PCM card 0 'TWL4030' device 0 subdevice 0
        Its setup is:
         stream      : PLAYBACK
         access      : RW_INTERLEAVED
         format      : S16_LE
         subformat   : STD
         channels    : 2
         rate        : 44100
         exact rate  : 44100 (44100/1)
         msbits      : 16
         buffer_size : 32768
         period_size : 2048
         period_time : 46439
         tick_time   : 7812
         tstamp_mode : NONE
         period_step : 1
         sleep_min   : 0
         avail_min   : 2048
         xfer_align  : 2048
         start_threshold  : 32768
         stop_threshold   : 32768
         silence_threshold: 0
         silence_size : 0
         boundary     : 1073741824
```


7. To stop the audio just press <CONTROL+C>.

12.15   USB Host Test

This test runs on the OTG port in the Host mode. It requires that a Powered USB hub be used, and that the Hub and device (Keyboard or mouse) be connected when the Linux OS is booted. This section is broken down into two sections, one for the mouse and the other for the keyboard.

12.15.1 Keyboard Test

NOTE: This test is run after the OS is booted with the Hub and Keyboard plugged in.

1. Make sure that you are in the MMC directory. If you are, proceed to step 3. If not, then type the following command:

   root@beagleboard /]# mount -t vfat /dev/mmcblk0p1 /mnt/mmc/cd /mnt/mmc/

2. Change the directory by typing:

   [root@beagleboard /]# cd /mnt/mmc/

3. Type the following command:

   root@beagleboard mmc]# evtest /dev/input/event1

4. Press a Key on USB Keyboard and look for a printout in the terminal window.

   Example if "a" is pressed the following output is seen on Console:

   Event: time 1657.754638, type 1 (Key), code 30 (A), value 1
   Event: time 1657.754638, -------------- Report Sync ------------
   Event: time 1657.964599, type 1 (Key), code 30 (A), value 0
   Event: time 1657.964599, -------------- Report Sync ------------

5. Press <CONTROL+C> to stop the test.

12.15.2 Mouse Test

NOTE: This test is run after the OS is booted with the Hub and mouse plugged in.

1. Make sure that you are in the MMC directory. If you are, proceed to step 3. If not, then type the following command:

   root@beagleboard /]# mount -t vfat /dev/mmcblk0p1 /mnt/mmc/cd /mnt/mmc/

2. Change the directory by typing:

> [root@beagleboard /]# cd /mnt/mmc/

3. Type the following command:

> root@beagleboard mmc]# `evtest /dev/input/event0`

4. Press mouse button and look for a printout in the terminal window.

> Example if Left button is pressed and released the following lines should get displayed on console
>
> Event: time 1871.724792, -------------- Report Sync -----------
> Event: time 1873.804687, type 1 (Key), code 272 (LeftBtn), value 1
> Event: time 1873.804687, -------------- Report Sync -----------
> Event: time 1873.964660, type 1 (Key), code 272 (LeftBtn), value 0
> Event: time 1873.964660, -------------- Report Sync -----------

5. Moving the Mouse also results in Console messages

> Event: time 1959.120635, -------------- Report Sync -----------
> Event: time 1959.130676, type 2 (Relative), code 0 (X), value -21
> Event: time 1959.130676, -------------- Report Sync -----------
> Event: time 1959.140625, type 2 (Relative), code 0 (X), value -16

6. Press <CONTROL+C> to stop the test

13.0   Troubleshooting

This section will provide assistance in troubleshooting the BeagleBoard in the event there are questions raised as to what the sate of the BeagleBoard is. This may be due to a HW failure or the SW not initializing things properly during development. Also provided is a section of know issues. Be sure and check with BeagleBoard.org for any updates.

For an up to date listing of common questions and their answers, please refer to http://elinux.org/BeagleBoardFAQ

13.1   Access Points

This section covers the various access points where various signals and voltages can be measured.

13.1.1 Voltage Points

Figure 71 shows the test points for the various voltages on BeagleBoard.
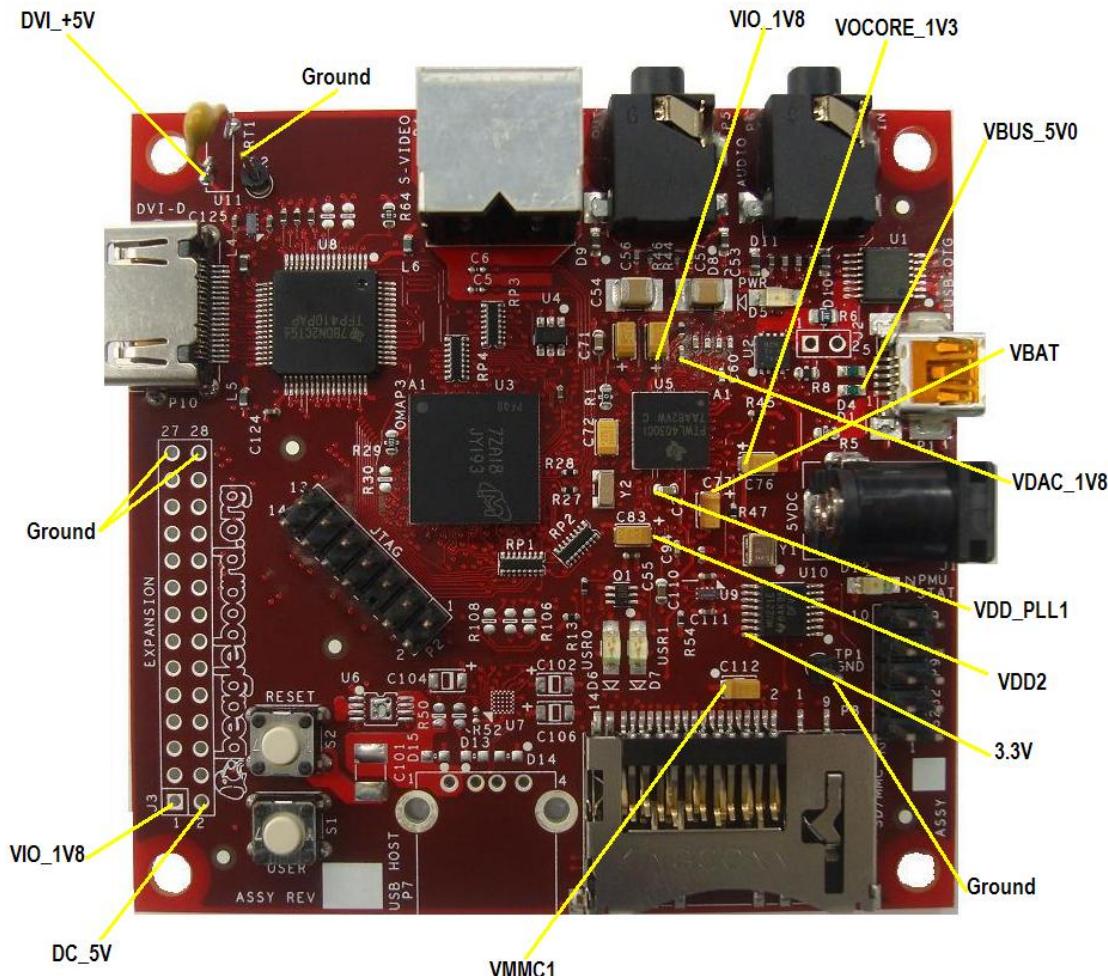

Figure 71.   BeagleBoard Voltage Access Points

Some of these voltages may not be present depending on the state of the TWL4030 as set by the OMAP3530. Others may be at different voltage levels depending on the same factor.

Table 31 provides the ranges of the voltages and the definition of the conditions as applicable.

Table 31.   Voltages

Voltage	Min	Nom	Max	Conditions
VIO_1V8	1.78	1.8	1.81	
VDD_SIM	1.78	1.8	1.81	
VBUS_5V0	4.9	5.0	5.5	From the host PC. May be lower or higher.
VOCORE_1V3	1.15	1.2	1.25	Can be set via SW. Voltage levels may vary.

VBAT	4.1	4.2	4.3	
VDAC_1V8	1.78	1.8	1.81	
VDD_PLL1	1.78	1.8	1.81	
VDD_PLL2	1.78	1.8	1.81	
VDD2	1.15	1.2	1.25	
3.3V	3.28	3.3	3.32	
VMMC1 (3V)	2.9	3.0	3.1	3.0V at power up. Can be set to via SW.
VMMC1(1.8V)	1.78	1.8	1.81	

13.1.2 Signal Access Points

Figure 72 shows the access points for various signals on BeagleBoard.
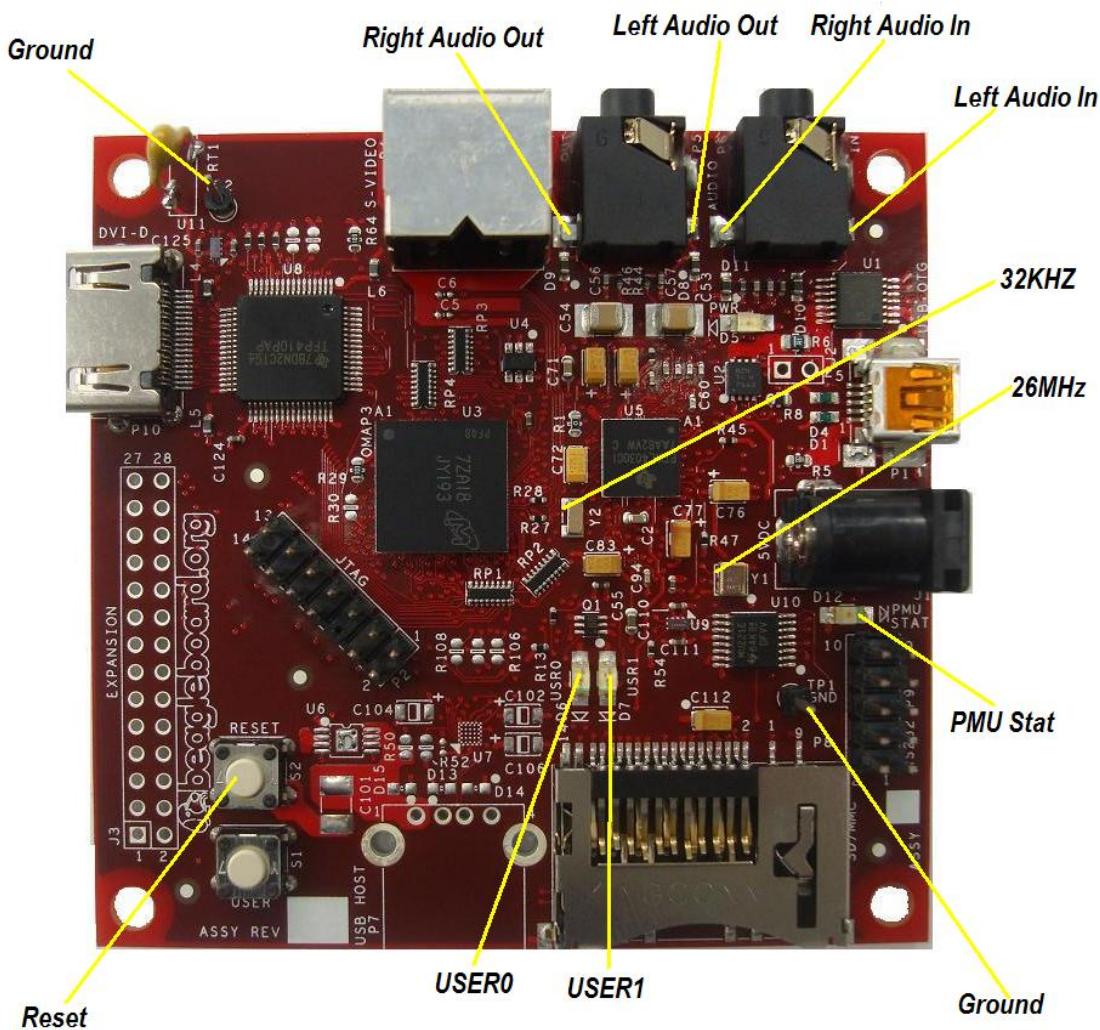
Figure 72.   BeagleBoard Signal Access Points

13.2   Troubleshooting Guide

Table 32 provides a list of possible failure modes and conditions and suggestions on how to diagnose them and ultimate determine whether the HW is operational or not.

Table 32.   Troubleshooting

Symptoms	Possible Problem	Action
JTAG does not connect.	Verify that the Power LED is on.	If off and running over USB, the PC may have shut down the voltage due to excessive current as related to what it is capable of providing. Remove the USB cable and re insert. If running on a DC supply make sure that voltage is being supplied.
	JTAG interface needs to be reset	Reset the BeagleBoard.
UBoot does not start, and no activity on the RS232 monitor.	Incorrect serial cable configuration.	Verify orientation of the RS232 flat cable
		Check for the right null modem cable.
	If a 40T is displayed over the serial cable, processor is booting. Issue could be the SD/MMC card.	Make sure the SD/MMC card is installed all they way into the connector.
		Make sure the card is formatted correctly and that the MLO file is the first file written to the SD card.
USB Host Connection Issues	Cheap USB Cable. OTG cables are typically not designed for higher current. The expect 100mA max.	Measure the voltage at the card to determine the voltage drop across the cable. If it the level is below 4.35V, the nUSB power is not guaranteed to work,

13.3   Serial Port Issues

We have had several serial port issues in the field caused by different issues. This section attempts to provide a step by step process to identify what the issue is.

The main thing to keep in mind is that the PC and the BeagleBoard connectors are wired the same. In order for them to talk, they must have a null modem cable to connect them.

The following sections provide steps to help identify the issue.

For additional help on debugging serial issues, refer to the FAQ at
http://elinux.org/BeagleBoardFAQ#Serial_connection_.231

13.3.1   First Step

1. Review the wiring of your IDC10 to DB9M serial adapter. Only the TX, RX and GND signals are used.
2. Make sure that the cable is plugged in correctly. The red stripe should be at the bottom next to pin1 of P9. Some cables may have the flat cable extending away from the BeagleBoard and others may be extending toward the middle of the BeagleBoard. **Figure 73** shows the proper orientation of the IDC serial cable.
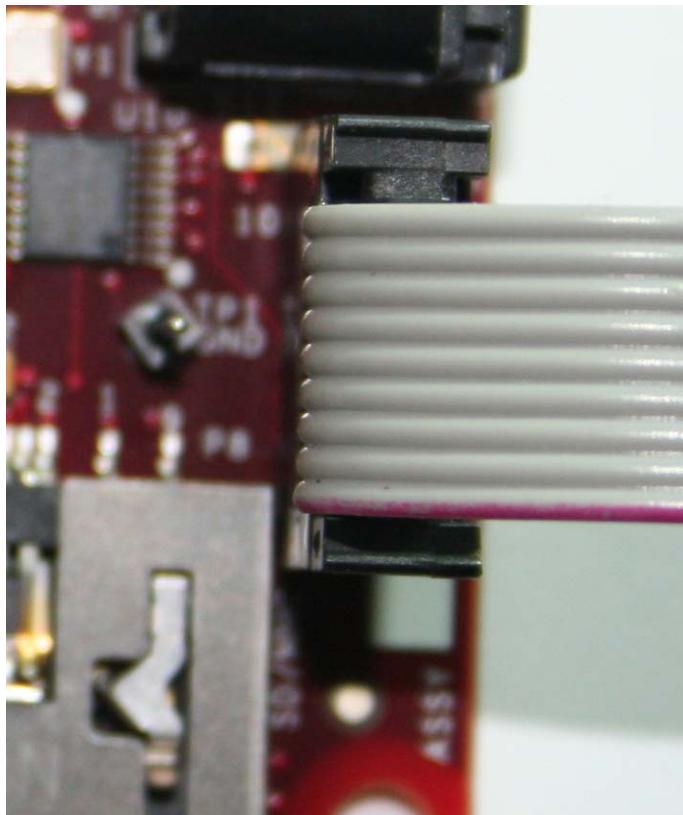
Figure 73.   BeagleBoard Serial Cable Orientation

![beagleboard.org]

3. You must have a Null Modem cable to connect to a PC. This results in the TX and RX leads being swapped, connecting the TX of the BeagleBoard to RX of the PC and RX of the BeagleBoard to TX of the PC. This cable also must be a female to female cable as the connectors on the BeagleBoard and PC are male. **Figure 74** shows the DB9 male connector and **Figure 75** shows the Null Modem Cable.

Figure 74. DB9 Male Connector
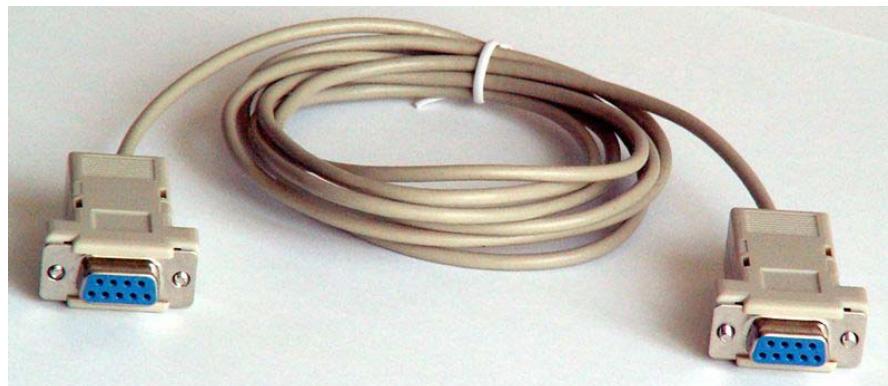
Figure 75. DB9 Null Modem Cable

4. If you have an ohmmeter, you can measure to see if the pins are swapped between pins 2 and 3 from each end of the cable.

13.3.2 Second Step

A simple test to verify that the cables you are using are correct to create a loopback on the cable. This checks the IDC cable and the null modem cable for connections.

1. Connect a wire across the TX and RX leads (Pins 2 and Pins 3) of the cable that plugs into the BeagleBoard (IDC Cable). **Figure 76** shows how this is done.
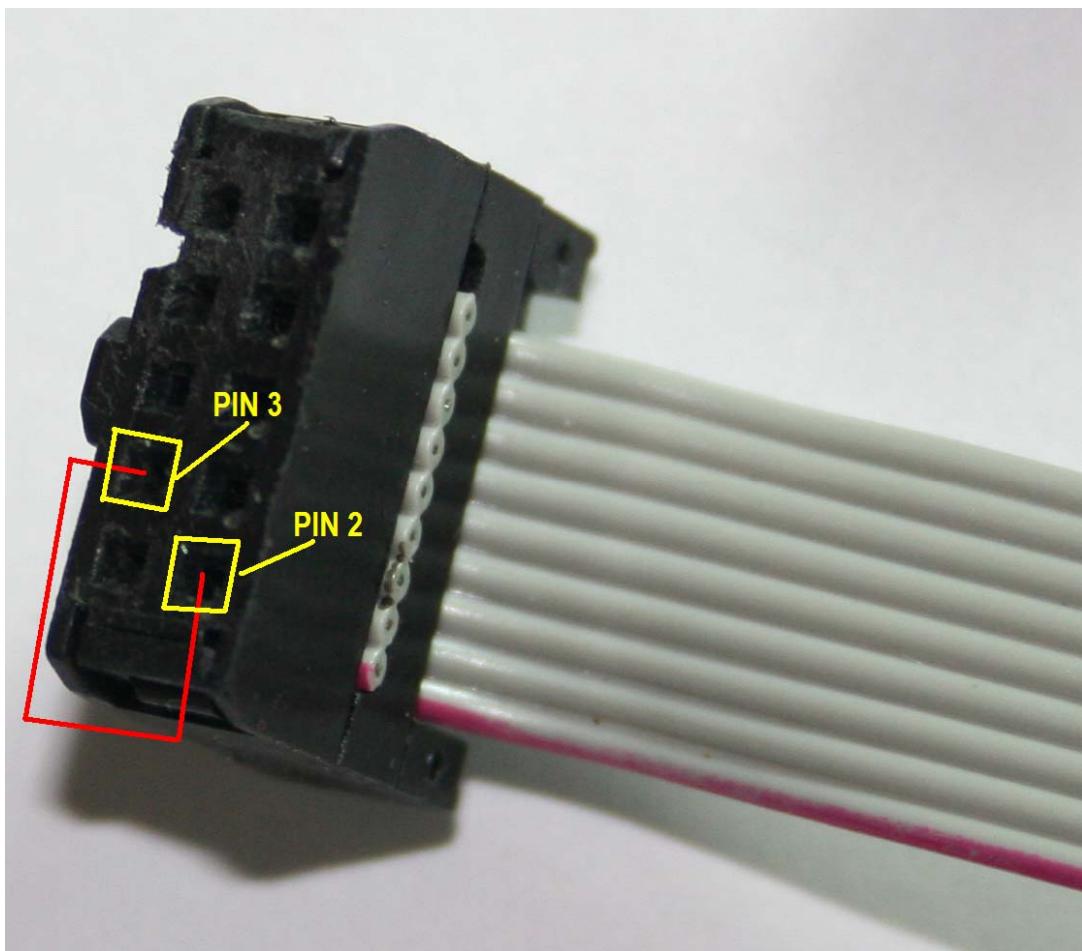
Figure 76.   Serial Cable Loopback

2. On your terminal start typing.
3. If the correct characters are echoed back, then the cables are in the proper configuration. Note that this checks the electrical connection only. If the terminal program is set wrong, then serial port will still not work.

13.3.3 Third Step

1. Make sure your terminal settings are correct.

   o BAUD RATE: 115200
   o DATA: 8 bit
   o PARITY: none
   o STOP: 1bit
   o FLOW CONTROL: none (Critical)

   Make sure that the Flow Control is set to none.

13.3.4   Fourth Step

If everything checks out OK on the previous steps, then the issue may be on the BeagleBoard. Follow the steps below to determine that state of the BeagleBoard.

1. Apply power to the board.
2. LED D5 should come on indicating that power is on.
3. LEDs USR0 and USR1 will come on once the board runs UBOOT.
4. By this time data should be printed to the terminal window.
5. Below are a couple of scenarios we have seen:

   o BeagleBoard sends data but cannot receive data
   o No data is sent at all
   o No data is sent, but it can be received.

If any of these issues are present, then there is a chance that the serial driver has failed. This is an issue with the level shifter, U9, on the board that we have seen fail after 48 hours of operation. The vast majority of boards with this issue are being screened out at the manufacturing stages, but some of the early shipment of boards could still exhibit this issue.

If this is the case, complete the RMA process at http://beagleboard.org/support/rma

14.0   Known Issues

This section provides information on any know issues with the BeagleBoard HW and the overall status. **Table 33** provides a list of the know issues on the BeagleBoard.

Table 33.   Known Issues

Affected Revision	Issue	Description	Workaround	Final Fix
B4	USB Host Port not working	The USB host port has been removed due to layout and other issues on the board that resulted in instability..	None	Rev. C
B4	Intermittent lockup of the serial port	The serial port will stop working when running Linux.	Remove C70 from the board	Rev B5
B5	USB Host Port not working	The USB host port has been removed due to layout and other issues on the board that resulted in instability..	None	Rev. C
B6	USB Host Port not working	The USB host port has been removed due to layout and other issues on the board that resulted in instability..	None	Rev. C

15.0 PCB Component Locations

Figures 77 and **Figure 78** contain the bottom and top side component locations of the BeagleBoard.
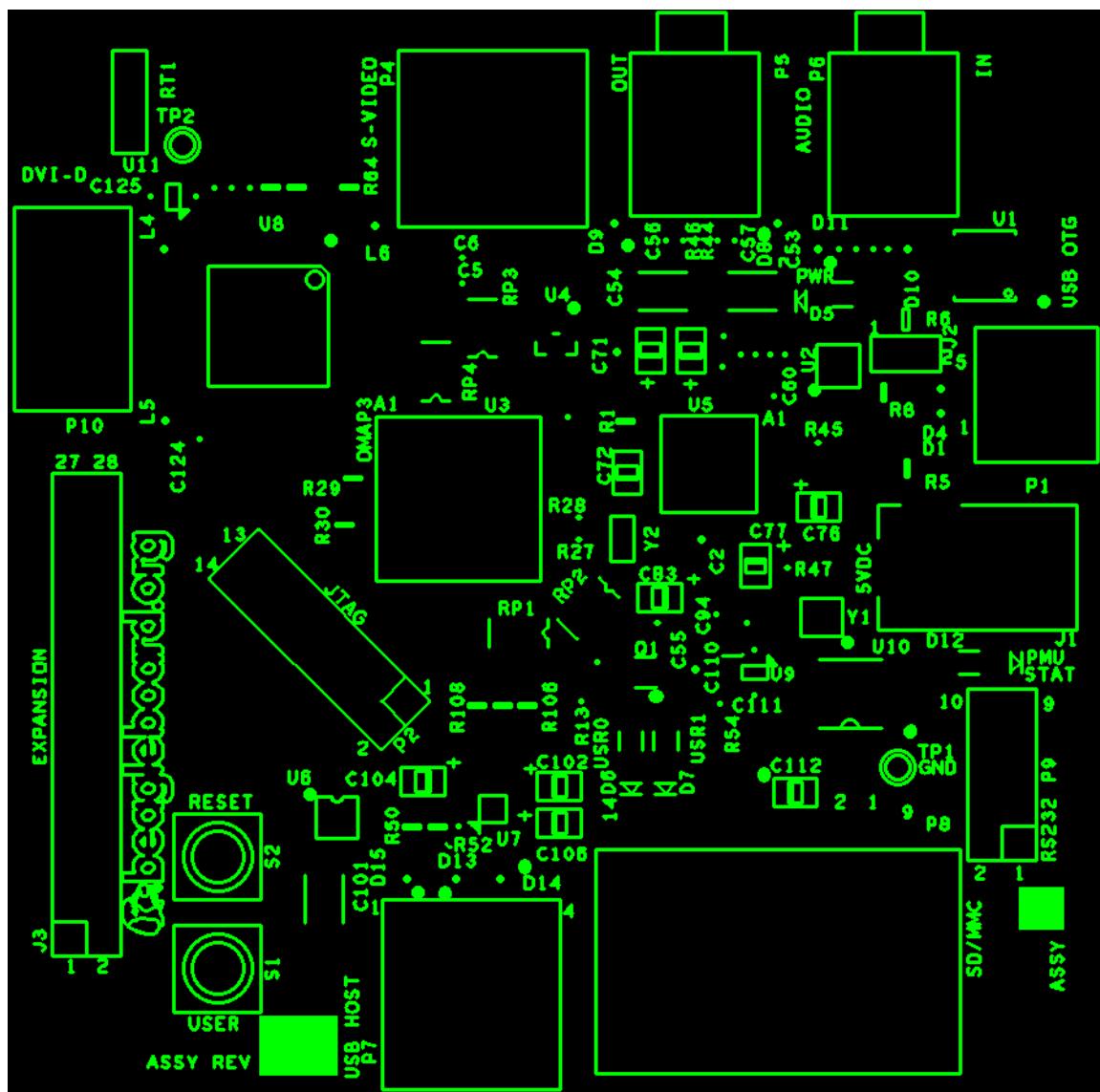
Figure 77. BeagleBoard Top Side Components
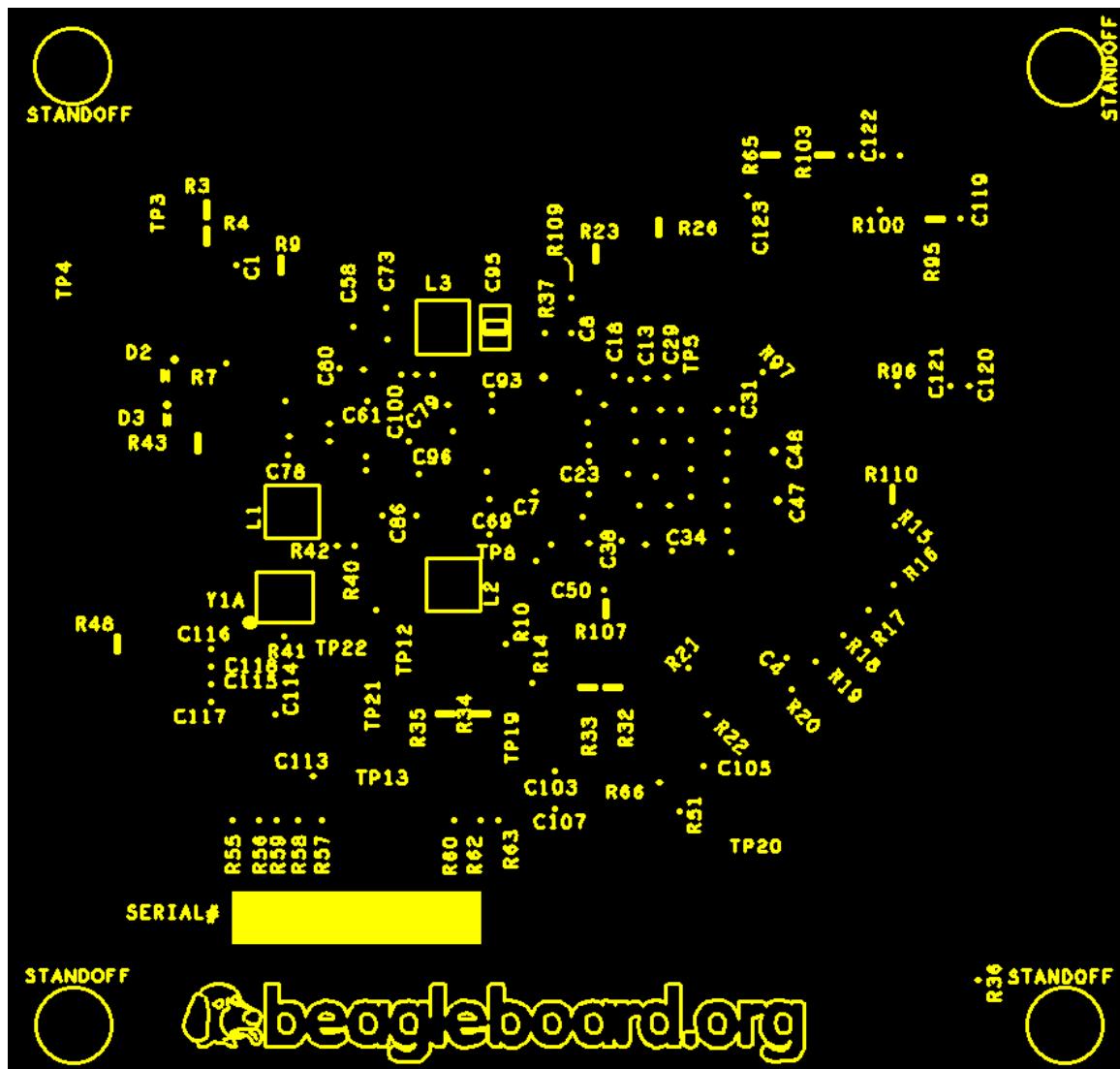
Figure 78.   BeagleBoard Bottom Side Components

The reference designators in green are not on the PCB. These were added to the figure.

16.0   Schematics

The following pages contain the PDF schematics for the BeagleBoard. This manual will be periodically updated, but for the latest documentations be sure and check BeagleBoard.org for the latest schematics.

OrCAD source files are provided for BeagleBoard on BeagleBoard.org at the following link.

http://www.beagleboard.org/uploads/BEAGLE_ORCAD_B4.zip

NOTE: For revision B5, C70 is not installed.

*These design materials are ***NOT SUPPORTED*** and **DO NOT** constitute a reference design. Only "community" support is allowed via resources at BeagleBoard.org/discuss. THERE IS NO WARRANTY FOR THE DESIGN MATERIALS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN MATERIALS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN MATERIALS IS WITH YOU. SHOULD THE DESIGN MATERIALS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.*
We mean it, these design materials may be totally unsuitable for any purposes.

REV	Description	DATE	BY
B1	1. Added 1K resistor to the GPMC_WAIT0 line. 2. Change power connector symbol to fix the swapped pins. 3. Add plated through mounting holes to P10 and provide soldering pad on the back side. 4. Fix hole size on P8 and make them plated through with a soldering pad on the back side. 5. Changed C101 to 47uf, 25V. 6. Removed TP15, TP16, TP17, and TP18. 7. USER0 and USER1 LEDs are wired wrong. Corrected net naming error on schematic. 8. Added 10K pullup to USB1HS_nCS signal. 9. Deleted P3, R53, R24, and R25.	4/8/08	GC
B2	1. Removed the USB host components from this assembly as the USB host on this layout is not reliable.	6/3/08	GC
B3	1. Added capacitors back in to reestablish noise margins.	6/18/08	GC
B4	1. Added 4.7uf across D3 to improve noise levels on OTG VBus. Some hubs would not work without this capacitor.	6/25/08	GC
B5	1. Removed capacitor C70 making it a DNI. Removal improves the rise/fall time of the 32KHz clock.	8/13/08	GC
B6	1. Replace U9 and U11 packages to address the issues we have had with failure of these devices. 2. Incorporated the Dx capacitor into the PCB as C127. 3. Changed the part number of the TWL4030 to TPS659501 to reflect the commercially available version of the device.	9/22/08	GC

Title	Beagle-Cover Page
Size	B
Document Number	450-5000-001
Date	Monday, October 13, 2008
Sheet	1 of 10
Rev	B6

Beagle-USB Client and Power

400-5000-001

Monday, October 13, 2008

Sheet 2 of 10

Beagle-OMAP3530 Processor (1 of 3)

Document Number
450-5000-001

Date: Monday, October 13, 2008

Sheet 3 of 10

Title: Beagle-OMAP3530 Processor (3 of 3)
Document Number: 450-5000-001
Date: Monday, October 13, 2008
Size: C   Rev: B6
Sheet: 5 of 10

Beagle-TPS65950 (1 of 2)

Title: Beagle-TPS65950 (2 of 2)

Document Number: 450-5000-001

Date: Monday, October 13, 2008

Sheet 7 of 10

These components are not installed except as noted.

Installed.

Expansion Connector

USB HOST INTERFACE

Power Control

Host Connector

Title: Beagle-USB Host and Expansion Connector.

Size: B

Document Number: 450-5000-001

Rev: B6

Date: Monday, October 13, 2008

Sheet 8 of 10

DVI-D Interface

Title: Beagle-DVI-D Interface

Size: B

Document Number: 450-5000-001

Rev: B6

Date: Monday, October 13, 2008

Sheet 10 of 10

17.0   Bills of Material

The Bill of Material for the Beagle Board is provided at BeagleBoard.org at the following location:

http://www.beagleboard.org/uploads/Beagle_BOM_B4.xls

*These design materials are *NOT SUPPORTED* and DO NOT constitute a reference design. Only "community" support is allowed via resources at BeagleBoard.org/discuss. THERE IS NO WARRANTY FOR THE DESIGN MATERIALS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN MATERIALS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN MATERIALS IS WITH YOU. SHOULD THE DESIGN MATERIALS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.*
We mean it, these design materials may be totally unsuitable for any purposes.

18.0   PCB Information

The following pages contain the PDF PCB layers for the BeagleBoard. Gerber files and Allegro source files are available on BeagleBoard.org at the following address.
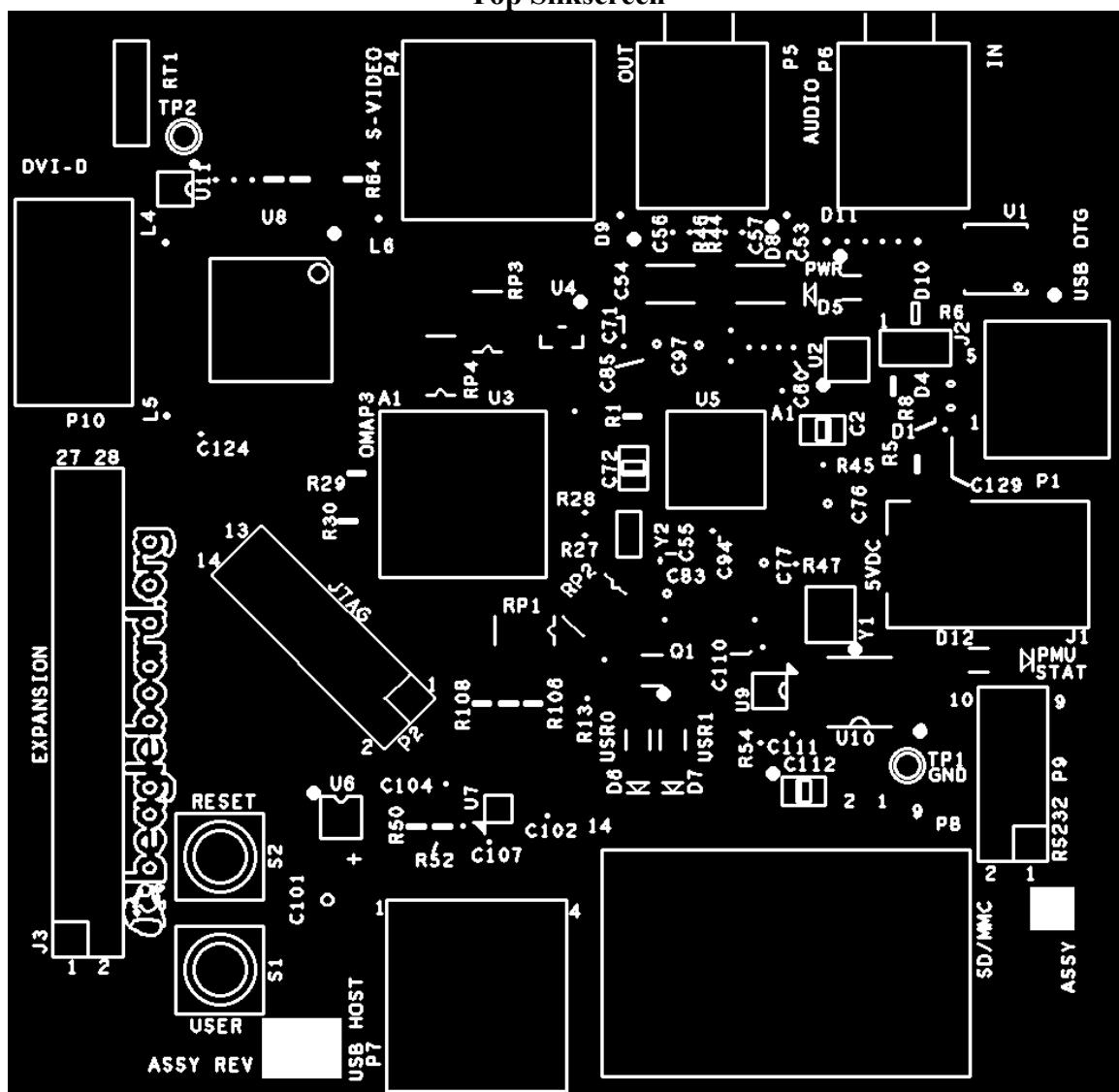
http://www.beagleboard.org/uploads/Beagle_Allegro_B.zip

http://www.beagleboard.org/uploads/Beagle_Gerbers_B.zip

These design materials are ****NOT SUPPORTED**** *and* ***DO NOT*** *constitute a reference design. Only "community" support is allowed via resources at BeagleBoard.org/discuss. THERE IS NO WARRANTY FOR THE DESIGN MATERIALS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN MATERIALS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN MATERIALS IS WITH YOU. SHOULD THE DESIGN MATERIALS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.*
*We mean it, these design materials may be totally unsuitable for any purposes.*

Top Silkscreen

Layer 1
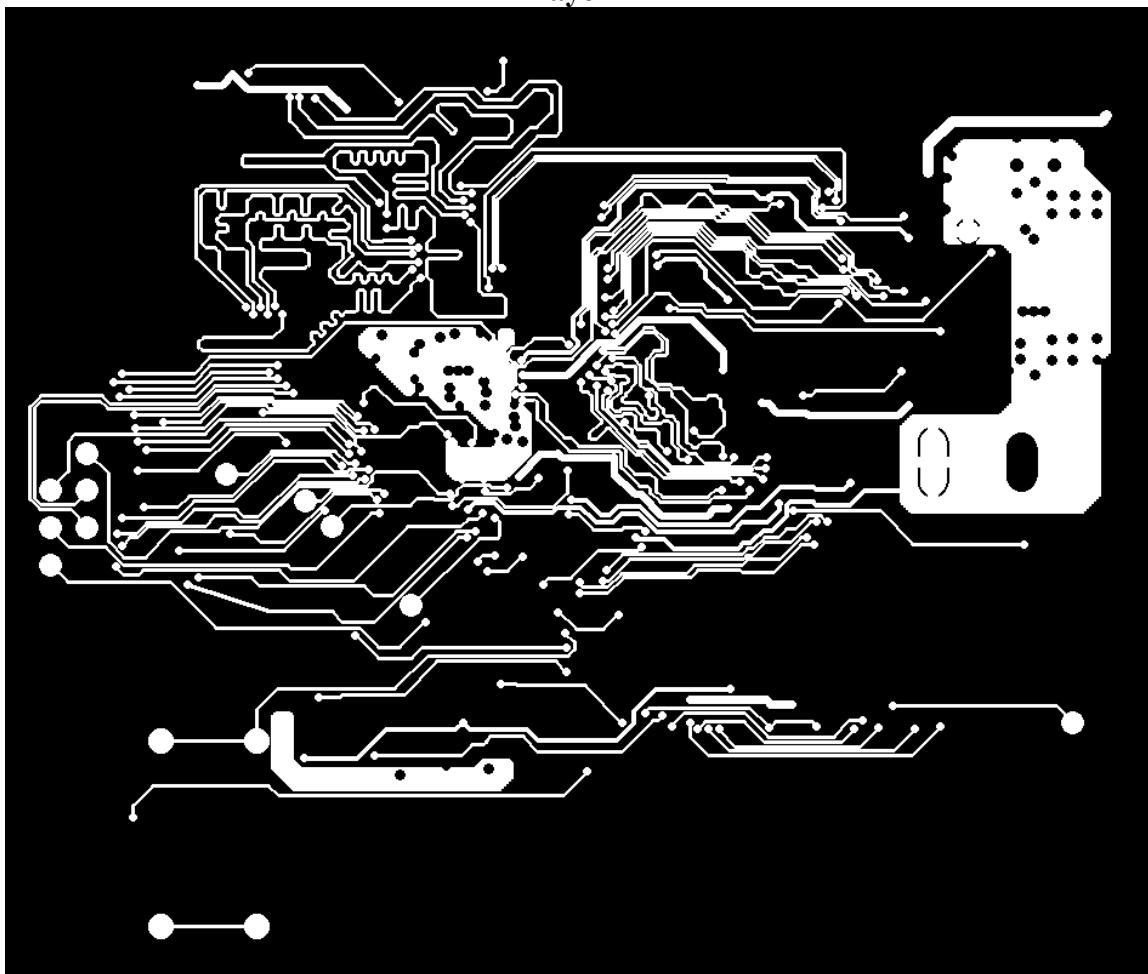
Layer 2
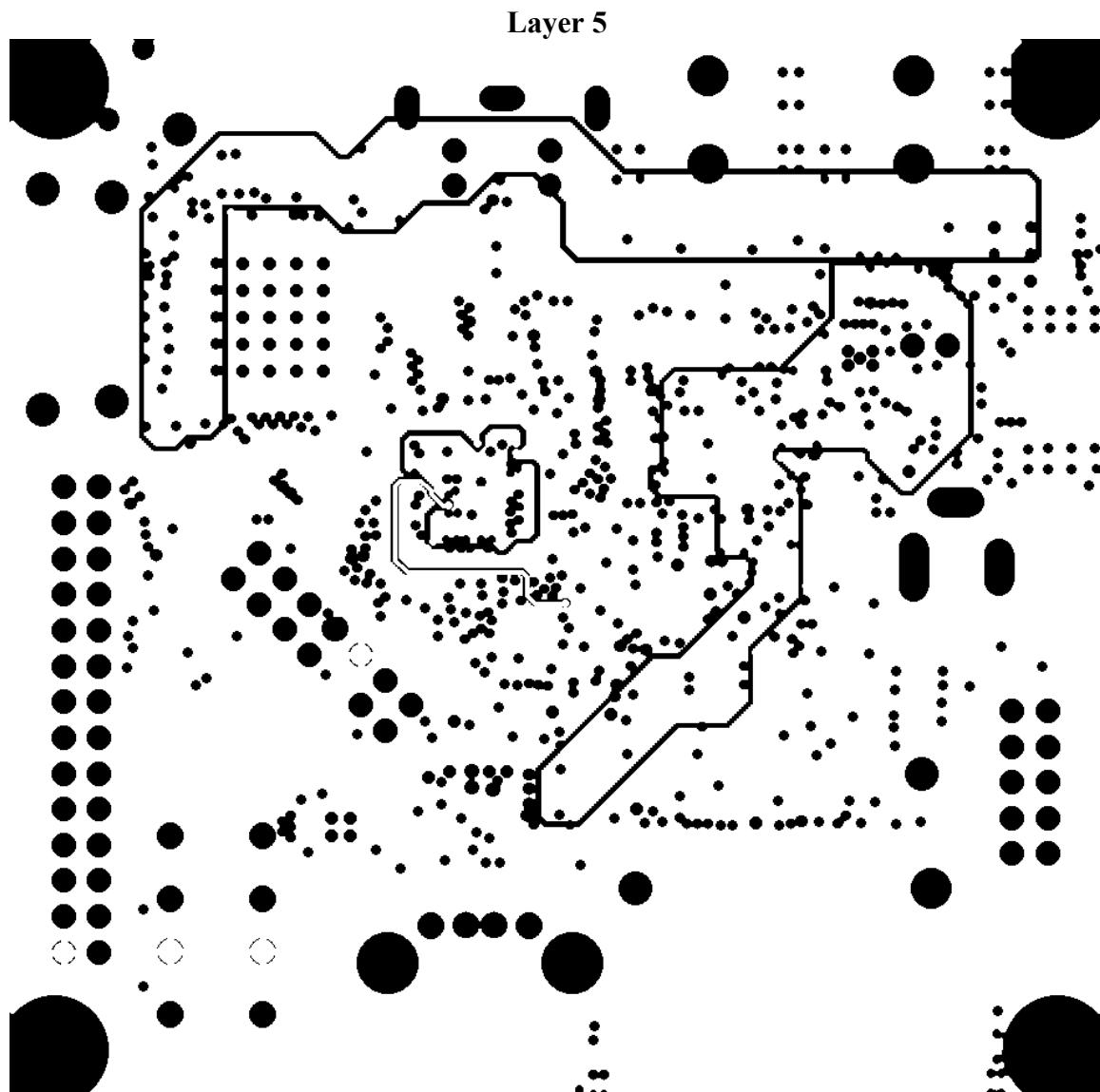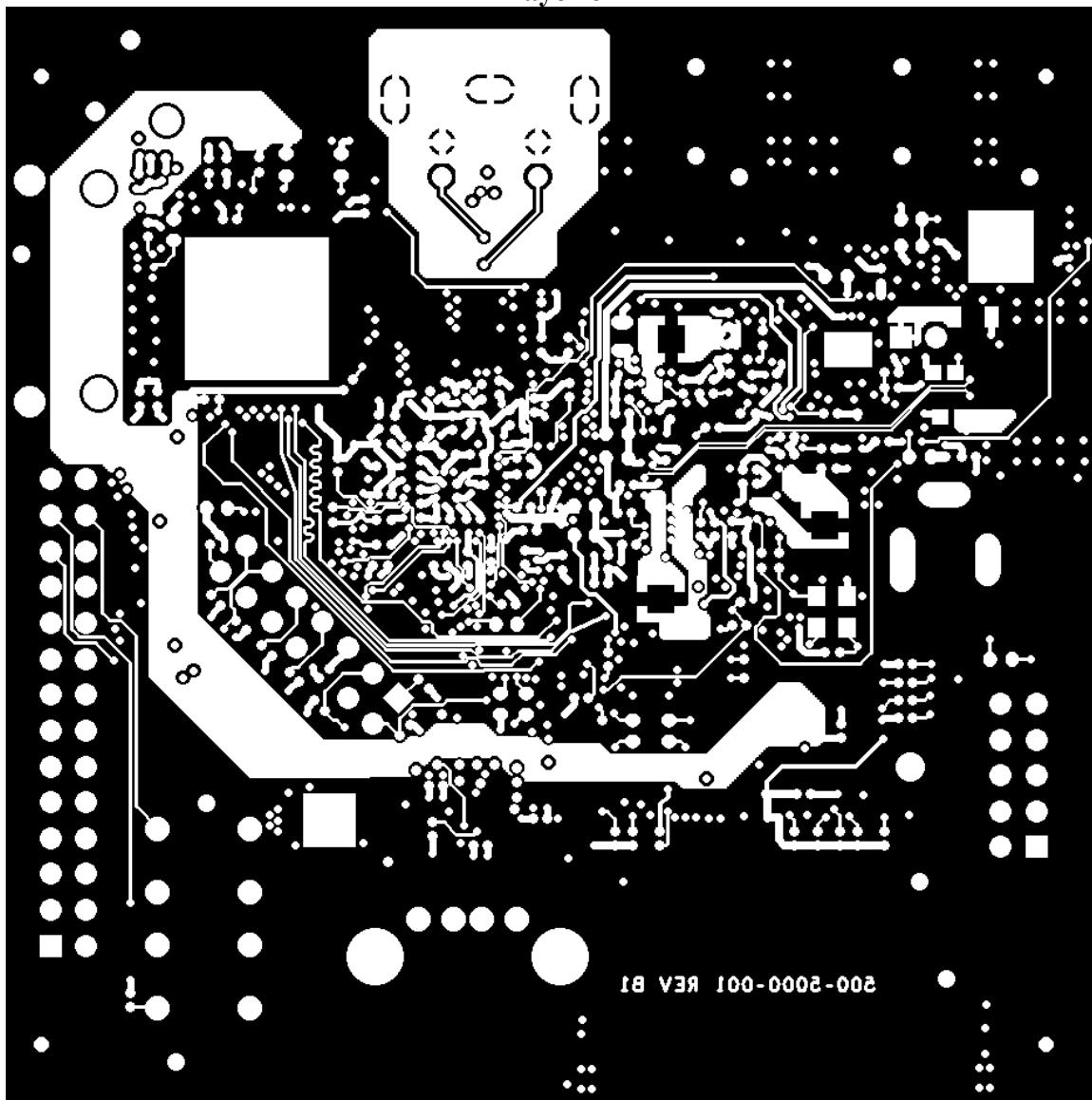
Layer 3

Layer 4

Layer 5

Layer 6

Bottom Silkscreen