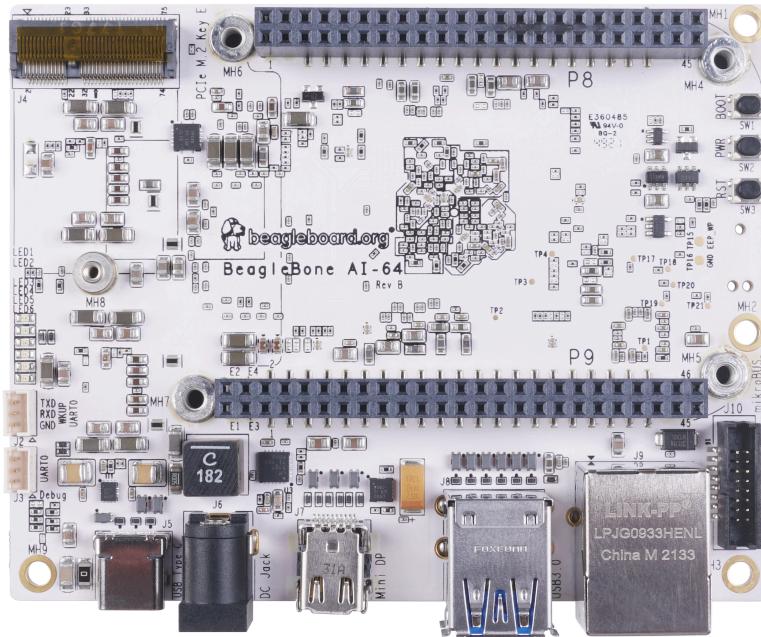

BeagleBone AI-64 System Reference Manual



Revision 1

Last PDF conversion: August 24, 2021

Maintaining editor: Deepak Khatri lorforlinux@beagleboard.org

Supply comments and errors via: <https://git.beagleboard.org/beagleboard/beaglebone-ai-64/-/issues>

All information in this document is subject to change without notice.

For an up to date version of this document refer to: <https://github.com/beagleboard/beaglebone-ai-64/System-Reference-Manual>

1. BEAGLEBONE AI-64 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.¹

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 THE PERFORMANCE OF THE DESIGN MATERIALS IS WITH YOU. SHOULD THE DESIGN MATERIALS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIRING, OR CORRECTION.

This board was designed as an evaluation and development tool. It was not designed with any other application in mind. As such, the design materials that are provided which include schematic, BOM, and PCB files, may or may not be suitable for any other purposes. If used, the design material becomes your responsibility as to whether or not it meets your specific needs or your specific applications and may require changes to meet your requirements.

2. BEAGLEBONE AI-64 ADDITIONAL TERMS

BeagleBoard.org Foundation and logo-licensed manufacturers provide the enclosed BeagleBone under the following conditions:

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies the Supplier from all claims arising from the handling or use of the goods.

Should the BeagleBone not meet the specifications indicated in the System Reference Manual, the BeagleBone may be returned within 90 days from the date of delivery to the distributor of purchase for a full refund. THE FOREGOING LIMITED WARRANTY IS THE EXCLUSIVE WARRANTY MADE

¹ <http://beagleboard.org/discuss>

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. 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.

Please read the System Reference Manual and, specifically, the Warnings and Restrictions notice in the Systems Reference Manual prior to handling the product. This notice contains important safety information about temperatures and voltages.

No license is granted under any patent right or other intellectual property right of Supplier covering or relating to any machine, process, or combination in which such Supplier products or services might be or are used. The Supplier currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. The Supplier assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

3. UNITED STATES FCC AND CANADA IC REGULATORY COMPLIANCE INFORMATION

The BeagleBone is annotated to comply with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This Class A or B digital apparatus complies with Canadian ICES-003. Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

4. BEAGLEBONE WARNINGS, RESTRICTIONS AND DISCLAIMERS

For Feasibility Evaluation Only, in Laboratory/Development Environments. BeagleBone AI-64 is not a complete product. It is intended solely for use

for preliminary feasibility evaluation in laboratory/development environments by technically qualified electronics experts who are familiar with the dangers and application risks associated with handling electrical mechanical components, systems and subsystems. It should not be used as all or part of a finished end product.

Your Sole Responsibility and Risk you acknowledge, represent, and agree that:

1. You have unique knowledge concerning Federal, State, and local regulatory requirements (including but not limited to Food and Drug Administration regulations, if applicable) which relate to your products and which relate to your use (and/or that of your employees, affiliates, contractors or designees) of the BeagleBone for evaluation, testing and other purposes.
2. You have full and exclusive responsibility to assure the safety and compliance of your products with all such laws and other applicable regulatory requirements, and also to assure the safety of any activities to be conducted by you and/or your employees, affiliates, contractors or designees, using the BeagleBone. Further, you are responsible to assure that any interfaces (electronic and/or mechanical) between the BeagleBone and any human body are designed with suitable isolation and means to safely limit accessible leakage currents to minimize the risk of electrical shock hazard.
3. Since the BeagleBone is not a completed product, it may not meet all applicable regulatory and safety compliance standards which may normally be associated with similar items. You assume full responsibility to determine and/or assure compliance with any such standards and related certifications as may be applicable. You will employ reasonable safeguards to ensure that your use of the BeagleBone will not result in any property damage, injury, or death, even if the BeagleBone should fail to perform as described or expected.

Certain Instructions. It is important to operate the BeagleBone AI-64 within Supplier's recommended specifications and environmental considerations per the user guidelines. *Exceeding the specified BeagleBone ratings (including but not limited to input and output voltage, current, power, and environmental ranges) may cause property damage, personal injury, or death. If there are questions concerning these ratings please contact the Supplier representative before connecting interface electronics including input power and intended loads. Any loads applied outside of the specified output range may result in unintended and/*

or inaccurate operation and/or possible permanent damage to the BeagleBone and/or interface electronics. Please consult the System Reference Manual before connecting any load to the BeagleBone output. If there is uncertainty as to the load specification, please contact the Supplier representative. During normal operation, some circuit components may have case temperatures greater than 60 C as long as the input and output are maintained at a normal ambient operating temperature. These components include but are not limited to linear regulators, switching transistors, pass transistors, and current sense resistors which can be identified using the BeagleBone schematic located at the link in the BeagleBone System Reference Manual. When placing measurement probes near these devices during normal operation, please be aware that these devices may be very warm to the touch. As with all electronic evaluation tools, only qualified personnel knowledgeable in electronic measurement and diagnostics normally found in development environments should use the BeagleBone.

Agreement to Defend, Indemnify and Hold Harmless. You agree to defend, indemnify" and hold the Suppliers, their licensors and their representatives harmless from and against any and all claims, damages, losses, expenses, costs and liabilities (collectively, "Claims") arising out of or in connection with any use of the BeagleBone that is not in accordance with the terms of the agreement. This obligation shall apply whether Claims arise under the law of tort or contract or any other legal theory, and even if the BeagleBone fails to perform as described or expected.

Safety-Critical or Life-Critical Applications. If you intend to evaluate the components for possible" use in safety critical applications (such as life support) where a failure of the Supplier's product would reasonably be expected to cause severe personal injury or death, such as devices which are classified as FDA Class III or similar classification, then you must specifically notify Suppliers of such intent and enter into a separate Assurance and Indemnity Agreement.

Mailing Address:

BeagleBoard.org Foundation
4467 Ascot Ct
Oakland Twp, MI 48306 U.S.A.

WARRANTY: The BeagleBone AI-64 Assembly as purchased 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](https://beagleboard.org/support)

For up to date SW images and technical information refer to <https://beagleboard.org/latest-images>

All support for this board is provided via community support at beagleboard.org/discuss²

To return a defective board for repair, please request an RMA at beagleboard.org/support/rma³

Please DO NOT return the board without approval from the RMA team first.

All boards received without RMA approval will not be worked on.

5. 1.0 Introduction

This document is the **System Reference Manual** for the BeagleBone AI-64 and covers its use and design. The board will primarily be referred to in the remainder of this document simply as the board, although it may also be referred to as the BeagleBone AI-64 as a reminder. There are also references to the original BeagleBone as well, and will be referenced as simply BeagleBone.

This design is subject to change without notice as we will work to keep improving the design as the product matures based on feedback and experience. Software updates will be frequent and will be independent of the hardware revisions and as such not result in a change in the revision number.

Make sure you check the support Wikis frequently for the most up to date information.

² <https://beagleboard.org/discuss>

³ <https://beagleboard.org/support/rma>

<https://github.com/beagleboard/beaglebone-ai-64/wiki>

6. 2.0 Change History

This section describes the change history of this document and board. Document changes are not always a result of a board change. A board change will always result in a document change.

6.1. 2.1 Document Change History

Table 1. Table 1. Change History

Rev	Changes	Date	By
A0	Preliminary	September 2021	JA
A1	Production release	December 2021	JA

6.2. 2.2 Board Changes

Full issue list at <https://github.com/beagleboard/beaglebone-ai-64/issues>

2.2.1 Rev B0

We are starting with revision B based on this being an update to the original BeagleBone AI. However, because this board ended up being so different, we've decided to name it BeagleBone AI-64, rather than simply a new revision.

This is the initial engineering prototype release of the board. We will be tracking changes from this point forward.

2.2.1.1 Known issues

- USB SuperSpeed cannot function

7. 3.0 Connecting Up Your BeagleBone AI-64

This section provides instructions on how to hook up your board. This beagle requires a 5V 3A minimum power adapter via either USB Type-C power adapter or a barrel jack power adapter.

Recommended adapters:

- <https://www.digikey.com/en/products/detail/raspberry-pi/RPI-USB-C-power-supply-White-US/10258760>

7.1. 3.0.1 Methods of operation

1. Tethered to a PC, or
2. As a standalone development platform in a desktop PC configuration with a Display Port Monitor, power supply, keyboard, and mouse

7.2. 3.1 What's In the Box

In the box you will find three main items as shown in [Figure 1](#).

- BeagleBone AI-64
- USB Type-A to USB Type-C Cable
- Instruction card with link to the support WIKI address.

This is sufficient for the tethered scenario and creates an out of box experience where the board can be used immediately with no other equipment needed.

TODO: Add BBAI-64 in the box (media/BBAI-in-Box.png)

Figure 1. In The Box.

7.3. 3.2 Main Connection Scenarios

This section will describe how to connect the board for use. This section is basically a slightly more detailed description of the Quick Start Guide that came in the box. There is also a Quick Start Guide document on the board that should also be referred to. The intent here is that someone looking to purchase the board will be able to read this section and get a good idea as to what the initial set up will be like.

The board can be configured in several different ways, but we will discuss the two most common scenarios as described in the Quick Start Guide card that comes in the box.

- Tethered to a PC via the USB cable
 - Board is accessed as a storage drive and
 - a virtual Ethernet connection.
- Standalone desktop
 - Display
 - Keyboard and mouse
 - External 5V power supply

Each of these configurations is discussed in general terms in the following sections.

For an up-to-date list of confirmed working accessories please go to <https://github.com/beagleboard/beaglebone-ai-64/wiki/Accessories>⁴

7.4. 3.3 Tethered To A PC

In this configuration, the board is powered by the PC via the provided USB cable—no other cables are required. The board is accessed either as a USB storage drive or via the browser on the PC. You need to use either Firefox or Chrome on the PC, Internet Explorer will not work properly. Figure 2 shows this configuration.



Figure 1. Tethered Configuration

⁴ <https://github.com/beagleboard/beaglebone-ai-64/wiki/Accessories>

All the power for the board is provided by the PC via the USB cable. In some instances, the PC may not be able to supply sufficient power for the board. In that case, an external 5VDC power supply can be used, but this should rarely be necessary.

3.3.1 Connect the Cable to the Board

1. Connect the small connector on the USB cable to the board as shown in [Figure 3](#).
3. The connector is on the bottom side of the board.

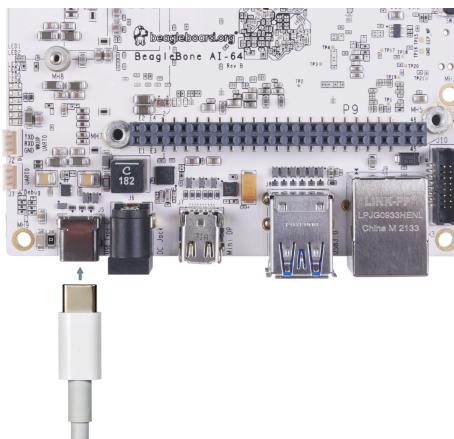


Figure 2. USB Connection to the Board

2. Connect the large connector of the USB cable to your PC or laptop USB port.
3. The board will power on and the power LED will be on as shown in [Figure 4](#) below.

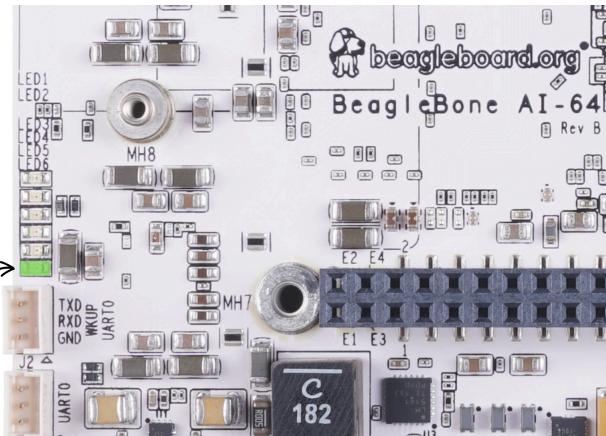


Figure 3. Board Power LED

4. When the board starts to the booting process started by the process of applying power, the LEDs will come on in sequence as shown in [Figure 5](#) below. It will take a few seconds for the status LEDs to come on, so be patient. The LEDs will be flashing in an erratic manner as it begins to boot the Linux kernel.

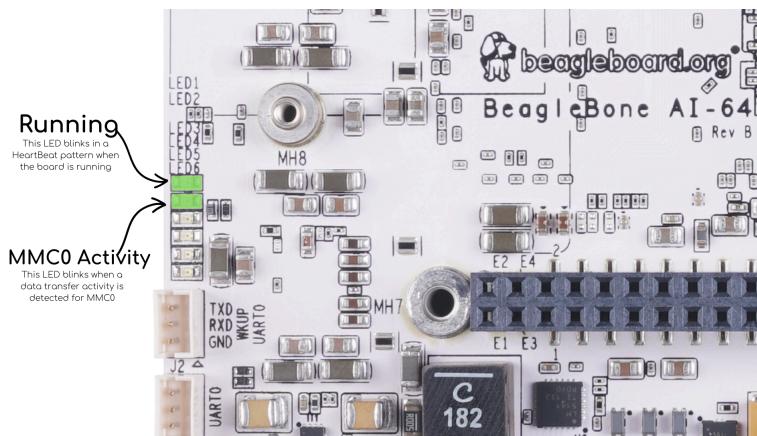


Figure 4. media/Board Boot Status

3.3.2 Accessing the Board as a Storage Drive

The board will appear around a USB Storage drive on your PC after the kernel has booted, which will take a round 10 seconds. The kernel on the board needs

to boot before the port gets enumerated. Once the board appears as a storage drive, do the following:

1. Open the USB Drive folder.
2. Click on the file named **start.htm**
3. The file will be opened by your browser on the PC and you should get a display showing the Quick Start Guide.
4. Your board is now operational! Follow the instructions on your PC screen.

7.5. 3.4 Standalone w/Display and Keyboard/Mouse

In this configuration, the board works more like a PC, totally free from any connection to a PC as shown in [Figure 6](#). It allows you to create your code to make the board do whatever you need it to do. It will however require certain common PC accessories. These accessories and instructions are described in the following section.

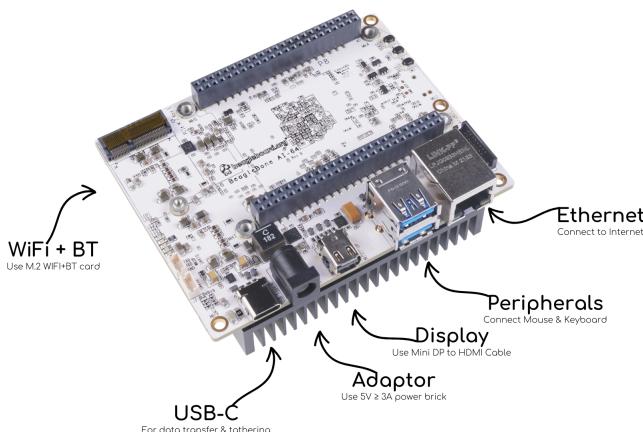


Figure 5. Desktop Configuration

Ethernet cable and M.2 WiFi + Bluetooth card are optional. They can be used if network access required.

3.4.1 Required Accessories

In order to use the board in this configuration, you will need the following accessories:

- (1) 5VDC 3A power supply.
- (1) Display Port monitor (or a recommended uDP to HDMI adapter. <https://www.amazon.com/dp/B089GF8M87> has been tested and worked beautifully.)
- (1) uDP to DP cable.
- (1) USB keyboard and mouse.
- (1) powered USB HUB (OPTIONAL). The board has only two USB Type-A host ports, so you may need to use a powered USB Hub if you wish to add additional USB devices, such as a USB WiFi adapter.
- (1) M.2 WiFi module (OPTIONAL). For wireless connections, a USB WiFi adapter or a recommended M.2 WiFi module can provide wireless networking.

For an up-to-date list of confirmed working accessories please go to <https://github.com/beagleboard/beaglebone-ai-64/wiki/Accessories>.

3.4.2 Connecting Up the Board

1. Connect the monitor

TODO

Figure 7. Connect microDP Cable to the Monitor.

TODO

2. If you have an HDMI or VGA monitor connect your adapter as shown in. [Figure 8](#) below from two perspectives.

Figure 8. microDP to HDMI Adapter.

(insert pictures.) **TODO**

3. If you have USB keyboard and mouse such as

seen in [Figure 9](#) below, you need to plug the receiver in the USB host port of the board as shown in [???](#).

Figure 9. USB Keyboard and Mouse.

4. Connect the Ethernet Cable

If you decide you want to connect to your local area network, an Ethernet cable can be used. Connect the Ethernet Cable to the Ethernet port as shown in [Figure 12](#). Any standard 100M Ethernet cable should work.

Figure 12. Ethernet Cable Connection.

TODO

5. The final step is to plug in the DC power supply to the DC power jack as shown in [Figure 13](#) below.

Figure 13. External DC Power.

TODO

6. The cable needed to connect to your display is a microHDMI to HDMI. Connect the microHDMI connector end to the board at this time. The connector is on the bottom side of the board as shown in [Figure 14](#) below.

Figure 14. Connect microDP Cable to the Board.

The connector is fairly robust, but we suggest that you not use the cable as a leash for your Beagle. Take proper care not to put too much stress on the connector or cable.

7. Booting the Board

As soon as the power is applied to the board, it will start the booting up process. When the board starts to boot the LEDs will come on in sequence as shown in [Figure 15](#) below. It will take a few seconds for the status LEDs to come on, so be patient. The LEDs will be flashing in an erratic manner as it boots the Linux kernel.

Figure 15. Board Boot Status.

While the four user LEDs can be over written and used as desired, they do have specific meanings in the image that is shipped with the board once the Linux kernel has booted.

- **USER0** is the heartbeat indicator from the Linux kernel.

- **USER1** turns on when the microSD card is being accessed
- **USER2** is an activity indicator. It turns on when the kernel is not in the idle loop.
- **USER3** turns on when the onboard eMMC is being accessed.

8. A Booted System

- a. The board will have a mouse pointer appear on the screen as it enters the Linux boot step. You may have to move the physical mouse to get the mouse pointer to appear. The system can come up in the suspend mode with the monitor in a sleep mode.
- b. After a minute or two a login screen will appear. You do not have to do anything at this point.
- c. After a minute or two the desktop will appear. It should be similar to the one shown in [Figure 16](#). HOWEVER, it will change from one release to the next, so do not expect your system to look exactly like the one in the figure, but it will be very similar.
- d. And at this point you are ready to go! [Figure 16](#) shows the desktop after booting.

Figure 16. Desktop Screen.

9. Powering Down

- a. Press the power button momentarily.
- b. The system will power down automatically.
- c. Remove the power jack.

8. 4.0 BeagleBone AI-64 Overview

The BeagleBone AI-64 is the latest addition to the BeagleBoard.org family and like its predecessors, is designed to address the Open Source Community, early adopters, and anyone interested in a low cost ARM Cortex-A8 processor based Single Board Computer (SBC).

It has been equipped with a minimum set of features to allow the user to experience the power of the processor and is not intended as a full development

platform as many of the features and interfaces supplied by the processor are not accessible from the BeagleBone AI-64 via onboard support of some interfaces. It is not a complete product designed to do any particular function. It is a foundation for experimentation and learning how to program the processor and to access the peripherals by the creation of your own software and hardware.

It also offers access to many of the interfaces and allows for the use of add-on boards called capes, to add many different combinations of features. A user may also develop their own board or add their own circuitry.

BeagleBone AI-64 is manufactured and warranted by partners listed at <https://beagleboard.org/logo> for the benefit of the community and its supporters.

Update: Jason Kridner of Texas Instruments handles the community promotions and is the spokesman for BeagleBoard.org.

Update: The board is designed by Gerald Coley of EmProDesign, a charter member of the BeagleBoard.org community.

Update: The PCB layout up through PCB revision B was done by Circuitco and Circuitco is the sole funder of its development and transition to production. Later PCB revisions have been made by Embest, a subsidiary of Avent.

Update: The Software is written and supported by the thousands of community members, including Jason Kridner, employee of Texas Instruments, and Robert Nelson, employee of DigiKey.

8.1. 4.1 BeagleBone Compatibility

TODO: The board is intended to be compatible with the original BeagleBone as much as possible. There are several areas where there are differences between the two designs. These differences are listed below, along with the reasons for the differences.

- J721E DRA829/TDA4VM/AM752x, 2GHZ, 64 Bit, processor.
 - Sorry, we just had to make it WAY faster.
- 4GB LPDDR4
 - *Cost reduction ???*

- Performance boost!!!
- Memory size increase
- Lower power
- Debug TTL serial.
 - *Cost reduction*
 - Can be added by buying a TTL to USB Cable that is widely available
 - Single largest cost reduction action taken
- EEPROM Reduced from 4KB to 1KB
 - *Cost Reduction*
- Onboard Managed NAND (eMMC)
 - 16GB
 - *Cost reduction*
 - Performance boost x8 vs. x4 bits
 - Performance boost due to deterministic properties vs. microSD card
- GPMC bus may not be accessible from the expansion headers in some cases
 - Result of eMMC on the main board
 - Signals are still routed to the expansion connector
 - If eMMC is not used, signals can be used via expansion if eMMC is held in reset
- There may be 10 less GPIO pins available
 - Result of eMMC
 - If eMMC is not used, could still be used
- The power expansion header, for battery and back-light, has been removed
 - *Cost reduction, space reduction*
 - Four pins were added to provide access to the battery charger function.
- Display Port interface onboard
 - Feature addition

- Audio and video capable
- Micro DP
- No three function USB cable
 - *Cost reduction*
- GPIO3_21 has a 24.576 MHZ clock on it.
 - This is required by the display Framer for Audio purposes. We needed to run a clock into the processor to generate the correct clock frequency. The pin on the processor was already routed to the expansion header. In order not to remove this feature on the expansion header, it was left connected. In order to use the pin as a GPIO pin, you need to disable the clock. While this disables audio to the display, the fact that you want to use this pin for something else, does the same thing.

8.2. 4.2 BeagleBone AI-64 Features and Specification

This section covers the specifications and features of the board and provides a high level description of the major components and interfaces that make up the board.

Table 2 provides a list of the features.

Table 2. Table 2. BeagleBone AI-64 Features

	Feature
Processor	TI J721E DRA829/TDA4VM/AM752x 2GHz, 4000 MIPS I'm not going to list all specifications of this monster SoC, and we'll do with J721E highlights instead: CPU Dual Cortex-A72 up to 2.0 GHz in a single cluster Up to three clusters of lockstep capable dual Cortex-R5F MCUs @ 1.0 GHz AI Accelerator / DSP Deep-learning Matrix Multiply Accelerator (MMA) @ up to 1.0 GHz (8 TOPS for 8-bit inference) C7x floating-point Vector DSP @ up to 1.0 GHz (80 GFLOPS) Up to two TMS320C66x floating-point DSPs @ up to 1.35 GHz (40 GFLOPS) Multimedia GPU – Imagination

Feature
<p>PowerVR Rogue 8XE GE8430 GPU Vision Processing Accelerator (VPAC) with image signal processor Depth and Motion Processing Accelerator (DMPAC) 1x dual-core multi-standard HD Video Decoder 1x dual-core multi-standard HD Video Encoder Memory I/F – Storage I/F – 1x Octal SPI (OSPI), GPMC for 8-/16-bit parallel NOR or NAND flash, 3x MMC/SD controllers, 1x UFS interface Two Navigator Subsystems (NAVSS) for data movement and control System MMU (SMMU) Version 3.0 and advanced virtualization capabilities. Peripherals Display – 1x eDP/DP, 1x MIPI DSI, and up to 2x DPI interfaces. Camera – 2x 4-lane MIPI CSI 2.0 Rx (camera receiver), 1x 4-lane MIPI CSI 2.0 Tx (camera transmitter) Audio – 12x MCASP (Multichannel Audio Serial Port) supporting up to 16 channels Networking Up to 2x Gigabit Industrial Communication Subsystems (ICSSG), each with dual PRUs and dual RTUs Integrated Ethernet switch supporting up to a total of 8 external ports in addition to legacy Ethernet switch of up to 2 ports. Up to 4 PCIe GEN3 controllers 2x USB 3.0 Dual-role device subsystems 16 MCANs, 12 McASP, eMMC and SD, UFS, OSPI/HyperBus memory controller, QSPI, 3x I3C, 12x I2C, 11x master/slave MCSPI, 12x UART, 10x GPIO modules Security 2x hardware accelerator blocks containing AES/DES/SHA/MD5 called SA2UL management Secure Boot Management Public Key Accelerator (PKA) for large vector math operation Trusted Execution Environment (TEE) Secure storage support On-the-fly encryption and authentication support for OSPI interface Manufacturing Process – 16-nm FinFET technology Package – 24 x 24 mm, 0.8-mm pitch, 827-pin FCBGA (ALF), enables IPC class 3 PCB routing The device is partitioned into three functional domains, each containing specific processing cores and peripherals:</p>

	Feature
	Wake-up (WKUP) domain Microcontroller (MCU) domain with one of the dual Cortex-R5 cluster MAIN domain
Graphics Engine	SGX530 3D, 20M Polygons/S See Processor data sheet at https://www.ti.com/lit/gpn/dra829v
SDRAM Memory	4GB LPDDR4 1000MHZ
Onboard Flash	16GB, 8bit Embedded MMC
PMIC	TPS659411 and TPS659413 PMICs regulator and one additional LDO.
Debug Support	3 pin Serial Header
Power Source	USB C or DC Jack
PCB	This beagle is not the standard 3.4" x 2.1", It's grown in size to 4" x 3.1"
Indicators	1-Power, 2-Ethernet, 4-User Controllable LEDs
HS USB 2.0 Client Port	Access to USB0, Client mode via USB-C
HS USB 2.0 Host Port	Access to USB1 & USB2, Type A Socket, 500mA LS/FS/HS
Serial Port	UART0 access via 3 pin 3.3V TTL micro Header. micro Header is populated cable was supplied
Ethernet	10/100, RJ45
SD/MMC Connector	microSD , 3.3V
User Input	Reset Button Boot Button Power Button
Video Out	uDP ,Unknown supported resolutions.
Audio	Via uDP Interface, Stereo
Expansion Connectors	Power 5V, 3.3V , VDD_ADC(1.8V) 3.3V I/O on all signals

	Feature
	McASP0, SPI1, I2C, GPIO(69 max), LCD, GPMC, MMC1, MMC2, 7 AIN (1.8V MAX), 4 Timers, 4 Serial Ports, CAN0, EHRPWM(0,2), XDMA Interrupt, Power button, Expansion Board ID (Up to 4 can be stacked)
Weight)
Power	Refer to Section 6.1.7 Power Consumption

8.3. 4.3 Board Component Locations

This section describes the key components on the board. It provides information on their location and function. Familiarize yourself with the various components on the board.

4.3.1 Connectors, LEDs, and Switches

[Figure 17](#) below shows the locations of the connectors, LEDs, and switches on the PCB layout of the board.

Figure 17. Connectors, LEDs and Switches.

- **DC Power** is the main DC input that accepts 5V power.
- **Power Button** alerts the processor to initiate the power down sequence and is used to power down the board.
- **10/100 Ethernet** is the connection to the LAN.
- **Serial Debug** is the serial debug port.
- **USB Client** is a USB-C connection to a PC that can also power the board.
- **BOOT switch** can be used to force a boot from the microSD card if the power is cycled on the board, removing power and reapplying the power to the board..
- There are four blue **LEDS** that can be used by the user.
- **Reset Button** allows the user to reset the processor.
- **microSD** slot is where a microSD card can be installed.
- **miniDP** connector is where the display is connected to.

- **USB Host** can be connected different USB interfaces such as Wi-Fi, BT, Keyboard, etc.

4.3.2 Key Components

[Figure 18](#) below shows the locations of the key components on the PCB layout of the board.

Figure 18. Key Components.

- **TI J721E DRA829/TDA4VM/AM752x** is the processor for the board.
- *4GB LPDDR4L is the Dual Data Rate RAM memory.
- **SMSC Ethernet PHY** is the physical interface to the network.
- **Micron eMMC** is an onboard MMC chip that holds up to 16GB of data.
- **uDP Framer** provides control for a DP HDMI or DVI-D display with an adapter.

9. 5.0 BeagleBone AI-64 High Level Specification

This section provides the high level specification of the BeagleBone AI-64.

9.1. 5.1 Block Diagram

[Figure 19](#) below is the high level block diagram of the BeagleBone AI-64.

Figure 19. BeagleBone AI-64 Key Components.

9.2. 5.2 Processor

The revision A and later boards have moved to the TI J721E DRA829/TDA4VM/AM752x device.

9.3. 5.3 Memory

Described in the following sections are the three memory devices found on the board.

5.3.1 4GB LPDDR4

A single 512Gb x16 LPDDR4 4Gb memory device is used. The memory used is:

- Q3222PM1WDGTK

5.3.2 4KB EEPROM

A single 4KB EEPROM is provided on I2C0 that holds the board information. This information includes board name, serial number, and revision information. This is not the same as the one used on the original BeagleBone. The device was changed for cost reduction reasons. It has a test point to allow the device to be programmed and otherwise to provide write protection when not grounded.

5.3.3 16GB Embedded MMC

A single 16GB embedded MMC (eMMC) device is on the board. The device connects to the MMC1 port of the processor, allowing for 8bit wide access. Default boot mode for the board will be MMC1 with an option to change it to MMC0, the SD card slot, for booting from the SD card as a result of removing and reapplying the power to the board. Simply pressing the reset button will not change the boot mode. MMC0 cannot be used in 8Bit mode because the lower data pins are located on the pins used by the Ethernet port. This does not interfere with SD card operation but it does make it unsuitable for use as an eMMC port if the 8 bit feature is needed.

5.3.4 MicroSD Connector

The board is equipped with a single microSD connector to act as the secondary boot source for the board and, if selected as such, can be the primary boot source. The connector will support larger capacity microSD cards. The microSD card is not provided with the board. Booting from MMC0 will be used to flash the eMMC in the production environment or can be used by the user to update the SW as needed.

5.3.5 Boot Modes

As mentioned earlier, there are two boot modes:

- **eMMC Boot**...This is the default boot mode and will allow for the fastest boot time and will enable the board to boot out of the box using the pre-flashed OS image without having to purchase an microSD card or an microSD card writer.
- **SD Boot**...This mode will boot from the microSD slot. This mode can be used to override what is on the eMMC device and can be used to program the eMMC when used in the manufacturing process or for field updates.

Software to support USB and serial boot modes is not provided by beagleboard.org. Please contact TI for support of this feature.

A switch is provided to allow switching between the modes.

- Holding the boot switch down during a removal and reapplication of power without a microSD card inserted will force the boot source to be the USB port and if nothing is detected on the USB client port, it will go to the serial port for download.
- Without holding the switch, the board will boot try to boot from the eMMC. If it is empty, then it will try booting from the microSD slot, followed by the serial port, and then the USB port.
- If you hold the boot switch down during the removal and reapplication of power to the board, and you have a microSD card inserted with a bootable image, the board will boot from the microSD card.

NOTE: Pressing the RESET button on the board will NOT result in a change of the boot mode. You MUST remove power and reapply power to change the boot mode. The boot pins are sampled during power on reset from the PMIC to the processor. The reset button on the board is a warm reset only and will not force a boot mode change.

9.4. 5.4 Power Management

The **TPS659411** and **TPS659413** power management device is used along with a separate LDO to provide power to the system. The **TPS659411** and **TPS659413** version provides for the proper voltages required for the LPDDR4. This is the same device as used on the original BeagleBone with the exception of the power rail configuration settings which will be changed in the internal EEPROM to the **TPS659411** and **TPS659413** to support the new voltages.

LPDDR4 requires 1.5V instead of 1.8V on the DDR2 as is the case on the original BeagleBone. The 1.8V regulator setting has been changed to 1.5V for the LPDDR4. The LDO3 3.3V rail has been changed to 1.8V to support those rails on the processor. LDO4 is still 3.3V for the 3.3V rails on the processor. An external **LDOTLV70233** provides the 3.3V rail for the rest of the board.

9.5. 5.5 PC USB Interface

The board has a miniUSB connector that connects the USB0 port to the processor. This is the same connector as used on the original BeagleBone.

9.6. 5.6 Serial Debug Port

Serial debug is provided via UART0 on the processor via a single 1x6 pin header. In order to use the interface a USB to TTL adapter will be required. The header is compatible with the one provided by FTDI and can be purchased for about 12 to 20 from various sources. Signals supported are TX and RX. None of the handshake signals are supported.

9.7. 5.7 USB1 Host Port

On the board is a single USB Type A female connector with full LS/FS/HS Host support that connects to USB1 on the processor. The port can provide power on/off control and up to 500mA of current at 5V. Under USB power, the board will not be able to supply the full 500mA, but should be sufficient to supply enough current for a lower power USB device supplying power between 50 to 100mA.

You can use a wireless keyboard/mouse configuration or you can add a HUB for standard keyboard and mouse interfacing.

9.8. 5.8 Power Sources

The board can be powered from two different sources:

- A USB port on a PC
- A 5VDC 3A power supply plugged into the DC connector.
- A power supply with a USB C connec

The USB cable is shipped with each board. This port is limited to 500mA by the Power Management IC. It is possible to change the settings in the **TPS659411** and **TPS659413** to increase this current, but only after the initial boot. And, at that point the PC most likely will complain, but you can also use a dual connector USB cable to the PC to get to 1A.

The power supply is not provided with the board but can be easily obtained from numerous sources. A 1A supply is sufficient to power the board, but if there is a cape plugged into the board or you have a power hungry device or hub plugged into the host port, then more current may be needed from the DC supply.

Power routed to the board via the expansion header could be provided from power derived on a cape. The DC supply should be well regulated and 5V +/- .25V.

9.9. 5.9 Reset Button

When pressed and released, causes a reset of the board. The reset button used on the BeagleBone AI-64 is a little larger than the one used on the original BeagleBone. It has also been moved out to the edge of the board so that it is more accessible.

9.10. 5.10 Power Button

A power button is provided near the reset button close to the Ethernet connector. This button takes advantage of the input to the PMIC for power down features. While a lot of capes have a button, it was decided to add this feature to the board to ensure everyone had access to some new features. These features include:

- Interrupt is sent to the processor to facilitate an orderly shutdown to save files and to un-mount drives.
- Provides ability to let processor put board into a sleep mode to save power.
- Can alert processor to wake up from sleep mode and restore state before sleep was entered.

If you hold the button down longer than 8 seconds, the board will power off if you release the button when the power LED turns off. If you continue to hold it, the board will power back up completing a power cycle.

We recommend that you use this method to power down the board. It will also help prevent contamination of the SD card or the eMMC.

If you do not remove the power jack, you can press the button again and the board will power up.

9.11. 5.11 Indicators

There are a total of five blue LEDs on the board.

- One blue power LED indicates that power is applied and the power management IC is up. If this LED flashes when applying power, it means that an excess current flow was detected and the PMIC has shut down.
- Four blue LEDs that can be controlled via the SW by setting GPIO pins.

In addition, there are two LEDs on the RJ45 to provide Ethernet status indication. One is yellow (100M Link up if on) and the other is green (Indicating traffic when flashing).

9.12. 5.13 DP Interface

A single uDP interface is connected to the 16 bit LCD interface on the processor.

9.13. 5.14 Cape Board Support

The BeagleBone AI-64 has the ability to accept up to four expansion boards or capes that can be stacked onto the expansion headers. The word cape comes from the shape of the board as it is fitted around the Ethernet connector on the main board. This notch acts as a key to ensure proper orientation of the cape.

The majority of capes designed for the original BeagleBone will work on the BeagleBone AI-64. The two main expansion headers will be populated on the board. There are a few exceptions where certain capabilities may not be present or are limited to the BeagleBone AI-64. These include:

- GPMC bus may NOT be available due to the use of those signals by the eMMC. If the eMMC is used for booting only and the file system is on the microSD card, then these signals could be used.

- Another option is to use the microSD or serial boot modes and not use the eMMC.
- The power expansion header is not on the BeagleBone AI-64 so those functions are not supported.

For more information on cape support refer to [Section 9.0 BeagleBone AI-64 Mechanical](#).

10. 6.0 Detailed Hardware Design

This section provides a detailed description of the Hardware design. This can be useful for interfacing, writing drivers, or using it to help modify specifics of your own design.

[Figure 20](#) below is the high level block diagram of the board. For those who may be concerned, [Figure 20](#) is the same figure as [Figure 19](#). It is placed here again for convenience so it is closer to the topics to follow.

Figure 20. BeagleBone AI-64 Block Diagram.

10.1. 6.1 Power Section

[Figure 21](#) is the high level block diagram of the power section of the board.

Figure 21. High Level Power Block Diagram.

This section describes the power section of the design and all the functions performed by the **TPS659411** and **TPS659413**.

6.1.1 TPS659411 and TPS659413 PMIC

The main Power Management IC (PMIC) in the system is the **TPS659411** and **TPS659413** which is a single chip power management IC consisting of a linear dual-input power path, three step-down converters, and four LDOs. LDO stands for Low Drop Out. If you want to know more about an LDO, you can go to http://en.wikipedia.org/wiki/Low-dropout_regulator.⁵ If you want to learn more

⁵ http://en.wikipedia.org/wiki/Low-dropout_regulator

about step-down converters, you can go to http://en.wikipedia.org/wiki/DC-to-DC_converter

The system is supplied by a USB port or DC adapter. Three high-efficiency 2.25MHz step-down converters are targeted at providing the core voltage, MPU, and memory voltage for the board.

The step-down converters enter a low power mode at light load for maximum efficiency across the widest possible range of load currents. For low-noise applications the devices can be forced into fixed frequency PWM using the I2C interface. The step-down converters allow the use of small inductors and capacitors to achieve a small footprint solution size.

LDO1 and LDO2 are intended to support system standby mode. In normal operation, they can support up to 100mA each. LDO3 and LDO4 can support up to 285mA each.

By default only LDO1 is always ON but any rail can be configured to remain up in SLEEP state. In particular the DCDC converters can remain up in a low-power PFM mode to support processor suspend mode. The **TPS659411** and **TPS659413** offers flexible power-up and power-down sequencing and several house-keeping functions such as power-good output, pushbutton monitor, hardware reset function and temperature sensor to protect the battery.

For more information on the **TPS659411** and **TPS659413**, refer to <http://www.ti.com/product/TPS659411> and [TPS659413](http://www.ti.com/product/TPS659413)[<http://www.ti.com/product/TPS659411> and [TPS659413](http://www.ti.com/product/TPS659413).]

[Figure 22](#) is the high level block diagram of the **TPS659411** and **TPS659413**.

Figure 22. TPS659411 and TPS659413 Block Diagram.

6.1.2 DC Input

[Figure 23](#) below shows how the DC input is connected to the **TPS659411** and **TPS659413**.

Figure 23. TPS65217 DC Connection.

A 5VDC supply can be used to provide power to the board. The power supply current depends on how many and what type of add-on boards are connected to the board. For typical use, a 5VDC supply rated at 1A should be sufficient. If heavier use of the expansion headers or USB host port is expected, then a higher current supply will be required.

The connector used is a 2.1MM center positive x 5.5mm outer barrel. The 5VDC rail is connected to the expansion header. It is possible to power the board via the expansion headers from an add-on card. The 5VDC is also available for use by the add-on cards when the power is supplied by the 5VDC jack on the board.

6.1.3 USB Power

The board can also be powered from the USB port. A typical USB port is limited to 500mA max. When powering from the USB port, the VDD_5V rail is not provided to the expansion headers, so capes that require the 5V rail to supply the cape direct, bypassing the **TPS659411** and **TPS659413**, will not have that rail available for use. The 5VDC supply from the USB port is provided on the SYS_5V, the one that comes from the **TPS659411** and **TPS659413**, rail of the expansion header for use by a cape. **Figure 24** is the connection of the USB power input on the PMIC.

Figure 24. USB Power Connections.

6.1.4 Power Selection

The selection of either the 5VDC or the USB as the power source is handled internally to the **TPS659411** and **TPS659413** and automatically switches to 5VDC power if both are connected. SW can change the power configuration via the I2C interface from the processor. In addition, the SW can read the **TPS659411** and **TPS659413** and determine if the board is running on the 5VDC input or the USB input. This can be beneficial to know the capability of the board to supply current for things like operating frequency and expansion cards.

It is possible to power the board from the USB input and then connect the DC power supply. The board will switch over automatically to the DC input.

6.1.5 Power Button

A power button is connected to the input of the **TPS659411** and **TPS659413**. This is a momentary switch, the same type of switch used for reset and boot selection on the board.

If you push the button the **TPS659411** and **TPS659413** will send an interrupt to the processor. It is up to the processor to then pull the **PMIC_POWER_EN** pin low at the correct time to power down the board. At this point, the PMIC is still active, assuming that the power input was not removed. Pressing the power button will cause the board to power up again if the processor puts the board in the power off mode.

In power off mode, the RTC rail is still active, keeping the RTC powered and running off the main power input. If you remove that power, then the RTC will not be powered. You also have the option of using the battery holes on the board to connect a battery if desired as discussed in the next section.

If you push and hold the button for greater than 8 seconds, the PMIC will power down. But you must release the button when the power LED turns off. Holding the button past that point will cause the board to power cycle.

6.1.6 Battery Access Pads

Four pads are provided on the board to allow access to the battery pins on the **TPS659411** and **TPS659413**. The pads can be loaded with a 4x4 header or you may just wire a battery into the pads. In addition they could provide access via a cape if desired. The four signals are listed below in [Table 3](#).

Table 3. Table 3. BeagleBone AI-64 Battery Pins

PIN	DESIGNATION	FUNCTION
BAT	TP5	Battery connection point.
SENSE	TP6	Battery voltage sense input, connect to BAT directly at the battery terminal.
TS	TP7	Temperature sense input. Connect to NTC thermistor to sense battery temperature.
GND	TP8	System ground.

There is no fuel gauge function provided by the **TPS659411** and **TPS659413**. That would need to be added if that function was required. If you want to add a fuel gauge, an option is to use 1-wire SPI or I2C device. You will need to add this using the expansion headers and place it on an expansion board.

NOTE: Refer to the TPS659411 and TPS659413 documentation before connecting anything to these pins.

6.1.7 Power Consumption

The power consumption of the board varies based on power scenarios and the board boot processes. Measurements were taken with the board in the following configuration:

- DC powered and USB powered
- monitor connected
- USB HUB
- 4GB Thumbdrive
- Ethernet connected @ 100M
- Serial debug cable connected

Table 4 is an analysis of the power consumption of the board in these various scenarios.

Table 4. BeagleBone AI-64 Power Consumption(mA@5V)

MODE	USB	DC	DC+USB
Reset	TBD	TBD	TBD
Idling @ UBoot	210	210	210
Kernel Booting (Peak)	460	460	460
Kernel Idling	350	350	350
Kernel Idling Display Blank	280	280	280
Loading a Webpage	430	430	430

The current will fluctuate as various activates occur, such as the LEDs on and microSD/eMMC accesses.

6.1.8 Processor Interfaces

The processor interacts with the **TPS659411** and **TPS659413** via several different signals. Each of these signals is described below.

6.1.8.1 I2C0

I2C0 is the control interface between the processor and the **TPS659411** and **TPS659413**. It allows the processor to control the registers inside the **TPS659411** and **TPS659413** for such things as voltage scaling and switching of the input rails.

6.1.8.2 PMIC_PWR_EN

On power up the **VDD_RTC** rail activates first. After the RTC circuitry in the processor has activated it instructs the **TPS659411** and **TPS659413** to initiate a full power up cycle by activating the **PMIC_PWR_EN** signal by taking it HI. When powering down, the processor can take this pin low to start the power down process.

6.1.8.3 LDO_GOOD

This signal connects to the **RTC_PORZn** signal, RTC power on reset. The small “n” indicates that the signal is an active low signal. Word processors seem to be unable to put a bar over a word so then is commonly used in electronics. As the RTC circuitry comes up first, this signal indicates that the LDOs, the 1.8V VRTC rail, is up and stable. This starts the power up process.

6.1.8.4 PMIC_PGOOD

Once all the rails are up, the **PMIC_PGOOD** signal goes high. This releases the **PORZn** signal on the processor which was holding the processor reset.

6.1.8.5 WAKEUP

The WAKEUP signal from the **TPS659411** and **TPS659413** is connected to the **EXT_WAKEUP** signal on the processor. This is used to wake up the processor when it is in a sleep mode. When an event is detected by the **TPS659411** and **TPS659413**, such as the power button being pressed, it generates this signal.

6.1.8.6 PMIC_INT

The **PMIC_INT** signal is an interrupt signal to the processor. Pressing the power button will send an interrupt to the processor allowing it to implement a power down mode in an orderly fashion, go into sleep mode, or cause it to wake up from a sleep mode. All of these require SW support.

6.1.9 Power Rails

Figure 25 shows the connections of each of the rails from the **TPS659411** and **TPS659413**.

Figure 25. Power Rails.

6.1.9.1 VRTC Rail

The **VRTC** rail is a 1.8V rail that is the first rail to come up in the power sequencing. It provides power to the RTC domain on the processor and the I/O rail of the **TPS659411** and **TPS659413**. It can deliver up to 250mA maximum.

6.1.9.2 VDD_3V3A Rail

The **VDD_3V3A** rail is supplied by the **TPS659411** and **TPS659413** and provides the 3.3V for the processor rails and can provide up to 400mA.

6.1.9.3 VDD_3V3B Rail

The current supplied by the **VDD_3V3A** rail is not sufficient to power all of the 3.3V rails on the board. So a second LDO is supplied, U4, a**TL5209A**, which sources the **VDD_3V3B** rail. It is powered up just after the **VDD_3V3A** rail.

6.1.9.4 VDD_1V8 Rail

The **VDD_1V8** rail can deliver up to 400mA and provides the power required for the 1.8V rails on the processor and the display framer. This rail is not accessible for use anywhere else on the board.

6.1.9.5 VDD_CORE Rail

The **VDD_CORE** rail can deliver up to 1.2A at 1.1V. This rail is not accessible for use anywhere else on the board and connects only to the processor. This rail is fixed at 1.1V and should not be adjusted by SW using the PMIC. If you do, then the processor will no longer work.

6.1.9.6 VDD_MPU Rail

The **VDD_MPU** rail can deliver up to 1.2A. This rail is not accessible for use anywhere else on the board and connects only to the processor. This rail defaults to 1.1V and can be scaled up to allow for higher frequency operation. Changing of the voltage is set via the I2C interface from the processor.

6.1.9.7 VDDS_DDR Rail

The **VDDS_DDR** rail defaults to **1.5V** to support the LPDDR4 rails and can deliver up to 1.2A. It is possible to adjust this voltage rail down to **1.35V** for lower power operation of the LPDDR4 device. Only LPDDR4 devices can support this voltage setting of 1.35V.

6.1.9.8 Power Sequencing

The power up process is consists of several stages and events. Figure 26 describes the events that make up the power up process for the processor from the PMIC. This diagram is used elsewhere to convey additional information. I saw no need to bust it up into smaller diagrams. It is from the processor datasheet supplied by Texas Instruments.

Figure 26. Power Rail Power Up Sequencing.

Figure 27 the voltage rail sequencing for the **TPS659411** and **TPS659413** as it powers up and the voltages on each rail. The power sequencing starts at 15 and then goes to one. That is the way the **TPS659411** and **TPS659413** is configured. You can refer to the TPS659411 and TPS659413 datasheet for more information.

Figure 27. TPS659411 and TPS659413 Power Sequencing Timing.

6.1.10 Power LED

The power LED is a blue LED that will turn on once the **TPS659411** and **TPS659413** has finished the power up procedure. If you ever see the LED flash once, that means that the **TPS659411** and **TPS659413** started the process and encountered an issue that caused it to shut down. The connection of the LED is shown in [Figure 25](#).

6.1.11 TPS659411 and TPS659413 Power Up Process

[Figure 28](#) shows the interface between the **TPS659411** and **TPS659413** and the processor. It is a cut from the PDF form of the schematic and reflects what is on the schematic.

Figure 28. Power Processor Interfaces.

When voltage is applied, DC or USB, the **TPS659411** and **TPS659413** connects the power to the SYS output pin which drives the switchers and LDOs in the **TPS659411** and **TPS659413**.

At power up all switchers and LDOs are off except for the **VRTC LDO** (1.8V), which provides power to the VRTC rail and controls the **RTC_PORZn** input pin to the processor, which starts the power up process of the processor. Once the RTC rail powers up, the **RTC_PORZn** pin, driven by the **LDO_PGOOD** signal from the **TPS659411** and **TPS659413**, of the processor is released.

Once the **RTC_PORZn** reset is released, the processor starts the initialization process. After the RTC stabilizes, the processor launches the rest of the power up process by activating the **PMIC_POWER_EN** signal that is connected to the **TPS659411** and **TPS659413** which starts the **TPS659411** and **TPS659413** power up process.

The **LDO_PGOOD** signal is provided by the **TPS659411** and **TPS659413** to the processor. As this signal is 1.8V from the **TPS659411** and **TPS659413** by virtue of the **TPS659411** and **TPS659413** VIO rail being set to 1.8V, and the **RTC_PORZ** signal on the processor is 3.3V, a voltage level shifter, **U4**, is used. Once the LDOs

and switchers are up on the **TPS659411** and **TPS659413**, this signal goes active releasing the processor. The LDOs on the **TPS659411** and **TPS659413** are used to power the VRTC rail on the processor.

6.1.12 Processor Control Interface

[Figure 28](#) above shows two interfaces between the processor and the **TPS659411** and **TPS659413** used for control after the power up sequence has completed.

The first is the **I2C0** bus. This allows the processor to turn on and off rails and to set the voltage levels of each regulator to support such things as voltage scaling.

The second is the interrupt signal. This allows the **TPS659411** and **TPS659413** to alert the processor when there is an event, such as when the power button is pressed. The interrupt is an open drain output which makes it easy to interface to 3.3V of the processor.

6.1.13 Low Power Mode Support

This section covers three general power down modes that are available. These modes are only described from a Hardware perspective as it relates to the HW design.

6.1.13.1 RTC Only

In this mode all rails are turned off except the **VDD_RTC**. The processor will need to turn off all the rails to enter this mode. The **VDD_RTC** staying on will keep the RTC active and provide for the wakeup interfaces to be active to respond to a wake up event.

6.1.13.2 RTC Plus DDR

In this mode all rails are turned off except the **VDD_RTC** and the **VDDS_DDR**, which powers the LPDDR4 memory. The processor will need to turn off all the rails to enter this mode. The **VDD_RTC** staying on will keep the RTC active and provide for the wakeup interfaces to be active to respond to a wake up event.

The **VDDS_DDR** rail to the LPDDR4 is provided by the 1.5V rail of the **TPS659411** and **TPS659413** and with **VDDS_DDR** active, the LPDDR4 can be placed in a

self refresh mode by the processor prior to power down which allows the memory data to be saved.

Currently, this feature is not included in the standard software release. The plan is to include it in future releases.

6.1.13.3 Voltage Scaling

For a mode where the lowest power is possible without going to sleep, this mode allows the voltage on the ARM processor to be lowered along with slowing the processor frequency down. The I₂C₀ bus is used to control the voltage scaling function in the **TPS659411** and **TPS659413**.

10.2. 6.2 TI J721E DRA829/TDA4VM/AM752x Processor

The board is designed to use the TI J721E DRA829/TDA4VM/AM752x processor in the 15 x 15 package.

6.2.1 Description

Figure 29 is a high level block diagram of the processor. For more information on the processor, go to <https://www.ti.com/product/TDA4VM>.⁶

Figure 29. Jacinto TDA4VMBZCZ Block Diagram.

6.2.2 High Level Features

Table 5 below shows a few of the high level features of the Jacinto processor.

Table 5. Table 5. Processor Features

Operating Systems	Linux, Android, Windows Embedded CE, QNX, ThreadX	MMC/SD	3
Standby Power	7 mW	CAN	2
ARM CPU	1 ARM Cortex-A8	UART (SCI)	6

⁶ <https://www.ti.com/product/TDA4VM>

ARM MHz (Max.)	275,500,600,800,1000	ADC	8-ch 12-bit
ARM MIPS (Max.)	1000,1200,2000	PWM (Ch)	3
Graphics Acceleration	1 3D	eCAP	3
Other Hardware Acceleration	2 PRU-ICSS,Crypto Accelerator	eQEP	3
On-Chip L1 Cache	64 KB (ARM Cortex-A8)	RTC	1
On-Chip L2 Cache	256 KB (ARM Cortex-A8)	I2C	3
Other On-Chip Memory	128 KB	McASP	2
Display Options	LCD	SPI	2
General Purpose Memory	1 16-bit (GPMC, NAND flash, NOR Flash, SRAM)	DMA (Ch)	64-Ch EDMA
DRAM	1 16-bit (LPDDR-400, DDR2-532, DDR3-400)	IO Supply (V)	1.8V(ADC),3.3V
USB Ports	2	Operating Temperature Range ©	-40 to 90

6.2.3 Documentation

Full documentation for the processor can be found on the TI website at <https://www.ti.com/product/TDA4VM> for the current processor used on the board. Make sure that you always use the latest datasheets and Technical Reference Manuals (TRM).

6.2.4 Crystal Circuitry

Figure 30 is the crystal circuitry for the TDA4VM processor.

Figure 30. Processor Crystals.

6.2.5 Reset Circuitry

[Figure 31](#) is the board reset circuitry. The initial power on reset is generated by the **TPS659411** and **TPS659413** power management IC. It also handles the reset for the Real Time Clock.

The board reset is the `SYS_RESETn` signal. This is connected to the `NRESET_INOUT` pin of the processor. This pin can act as an input or an output. When the reset button is pressed, it sends a warm reset to the processor and to the system.

On the revision A5D board, a change was made. On power up, the `NRESET_INOUT` signal can act as an output. In this instance it can cause the `SYS_RESETn` line to go high prematurely. In order to prevent this, the `PORZn` signal from the **TPS659411** and **TPS659413** is connected to the `SYS_RESETn` line using an open drain buffer. These ensure that the line does not momentarily go high on power up.

Figure 31. Board Reset Circuitry.

This change is also in all revisions after A5D.

LPDDR4 Memory

The BeagleBone AI-64 uses a single **MT41K256M16HA-125** 512MB LPDDR4 device from Micron that interfaces to the processor over 16 data lines, 16 address lines, and 14 control lines. On rev C we added the Kingston **KE4CN2H5A-A58** device as a source for the LPDDR4 device.

The following sections provide more details on the design.

6.2.6 Memory Device

The design supports the standard DDR3 and LPDDR4 x16 devices and is built using the LPDDR4. A single x16 device is used on the board and there is no support for two x8 devices. The DDR3 devices work at 1.5V and the LPDDR4 devices can work down to

1.35V to achieve lower power. The LPDDR4 comes in a 96-BALL FBGA package with 0.8 mil pitch. Other standard DDR3 devices can also be supported, but the

LPDDR4 is the lower power device and was chosen for its ability to work at 1.5V or 1.35V. The standard frequency that the LPDDR4 is run at on the board is 400MHZ.

6.2.7 LPDDR4 Memory Design

[Figure 32](#) is the schematic for the LPDDR4 memory device. Each of the groups of signals is described in the following lines.

Address Lines: Provide the row address for ACTIVATE commands, and the column address and auto pre-charge bit (A10) for READ/WRITE commands, to select one location out of the memory array in the respective bank. A10 sampled during a PRECHARGE command determines whether the PRECHARGE applies to one bank (A10 LOW, bank selected by BA[2:0]) or all banks (A10 HIGH). The address inputs also provide the op-code during a LOAD MODE command. Address inputs are referenced to VREFCA. A12/BC#: When enabled in the mode register (MR), A12 is sampled during READ and WRITE commands to determine whether burst chop (on-the-fly) will be performed (HIGH = BL8 or no burst chop, LOW = BC4 burst chop).

Bank Address Lines: BA[2:0] define the bank to which an ACTIVATE, READ, WRITE, or PRECHARGE command is being applied. BA[2:0] define which mode register (MR0, MR1, MR2, or MR3) is loaded during the LOAD MODE command. BA[2:0] are referenced to VREFCA.

CK and CK# Lines: are differential clock inputs. All address and control input signals are sampled on the crossing of the positive edge of CK and the negative edge of CK#. Output data strobe (DQS, DQS#) is referenced to the crossings of CK and CK#.

Clock Enable Line: CKE enables (registered HIGH) and disables (registered LOW) internal circuitry and clocks on the DRAM. The specific circuitry that is enabled/disabled is dependent upon the DDR3 SDRAM configuration and operating mode. Taking CKE LOW provides PRECHARGE power-down and SELF REFRESH operations (all banks idle) or active power-down (row active in any bank). CKE is synchronous for powerdown entry and exit and for self refresh entry. CKE is asynchronous for self refresh exit. Input buffers (excluding CK, CK#, CKE, RESET#, and ODT) are disabled during powerdown. Input buffers (excluding CKE and RESET#) are disabled during SELF REFRESH. CKE is referenced to VREFCA.

Figure 32. LPDDR4 Memory Design.

Chip Select Line: CS# enables (registered LOW) and disables (registered HIGH) the command decoder. All commands are masked when CS# is registered HIGH. CS# provides for external rank selection on systems with multiple ranks. CS# is considered part of the command code. CS# is referenced to VREFCA.

Input Data Mask Line: DM is an input mask signal for write data. Input data is masked when DM is sampled HIGH along with the input data during a write access. Although the DM ball is input-only, the DM loading is designed to match that of the DQ and DQS balls. DM is referenced to VREFDQ.

On-die Termination Line: ODT enables (registered HIGH) and disables (registered LOW) termination resistance internal to the LPDDR4 SDRAM. When enabled in normal operation, ODT is only applied to each of the following balls: DQ[7:0], DQS, DQS#, and DM for the x8; DQ[3:0], DQS, DQS#, and DM for the x4. The ODT input is ignored if disabled via the LOAD MODE command. ODT is referenced to VREFCA.

6.2.8 Power Rails

The **LPDDR4** memory device and the DDR3 rails on the processor are supplied by the **TPS659411** and **TPS659413**. Default voltage is 1.5V but can be scaled down to 1.35V if desired.

6.2.9 VREF

The **VREF** signal is generated from a voltage divider on the **VDDS_DDR** rail that powers the processor DDR rail and the LPDDR4 device itself. **Figure 33** below shows the configuration of this signal and the connection to the LPDDR4 memory device and the processor.

Figure 33. LPDDR4 VREF Design*.

10.3. 6.3 4GB eMMC Memory

The eMMC is a communication and mass data storage device that includes a Multi-MediaCard (MMC) interface, a NAND Flash component, and a controller on

an advanced 11-signal bus, which is compliant with the MMC system specification. The nonvolatile eMMC draws no power to maintain stored data, delivers high performance across a wide range of operating temperatures, and resists shock and vibration disruption.

One of the issues faced with SD cards is that across the different brands and even within the same brand, performance can vary. Cards use different controllers and different memories, all of which can have bad locations that the controller handles. But the controllers may be optimized for reads or writes. You never know what you will be getting. This can lead to varying rates of performance. The eMMC card is a known controller and when coupled with the 8bit mode, 8 bits of data instead of 4, you get double the performance which should result in quicker boot times.

The following sections describe the design and device that is used on the board to implement this interface.

6.3.1 eMMC Device

The device used is one of two different devices:

- Micron **MTFC4GLDEA 0M WT**
- Kingston **KE4CN2H5A-A58**

The package is a 153 ball WFBGA device on both devices.

6.3.2 eMMC Circuit Design

[Figure 34](#) is the design of the eMMC circuitry. The eMMC device is connected to the MMC1 port on the processor. MMC0 is still used for the microSD card as is currently done on the original BeagleBone. The size of the eMMC supplied is now 4GB.

The device runs at 3.3V both internally and the external I/O rails. The VCCI is an internal voltage rail to the device. The manufacturer recommends that a 1uF capacitor be attached to this rail, but a 2.2uF was chosen to provide a little margin.

Pullup resistors are used to increase the rise time on the signals to compensate for any capacitance on the board.

Figure 34. eMMC Memory Design.

The pins used by the eMMC1 in the boot mode are listed below in **Table 6**.

Table 6. eMMC Boot Pins.

For eMMC devices the ROM will only support raw mode. The ROM Code reads out raw sectors from image or the booting file within the file system and boots from it. In raw mode the booting image can be located at one of the four consecutive locations in the main area: offset 0x0 / 0x20000 (128 KB) / 0x40000 (256 KB) / 0x60000 (384 KB). For this reason, a booting image shall not exceed 128KB in size. However it is possible to flash a device with an image greater than 128KB starting at one of the aforementioned locations. Therefore the ROM Code does not check the image size. The only drawback is that the image will cross the subsequent image boundary. The raw mode is detected by reading sectors #0, #256, #512, #768. The content of these sectors is then verified for presence of a TOC structure. In the case of a **GP Device**, a Configuration Header (CH)**must** be located in the first sector followed by a **GP header**. The CH might be void (only containing a CHSETTINGS item for which the Valid field is zero).

The ROM only supports the 4-bit mode. After the initial boot, the switch can be made to 8-bit mode for increasing the overall performance of the eMMC interface.

10.4. 6.4 Board ID EEPROM

The BeagleBone is equipped with a single 32Kbit(4KB) 24LC32AT-I/OT EEPROM to allow the SW to identify the board. **Table 7** below defined the contents of the EEPROM.

Table 6. Table 7. EEPROM Contents

Name	Size (bytes)	Contents
Header	4	0xAA, 0x55, 0x33, EE
Board Name	8	Name for board in ASCII: A335BNLT
Version	4	Hardware version code for board in ASCII:

Name	Size (bytes)	Contents
		00A3 for Rev A3, 00A4 for Rev A4, 00A5 for Rev A5, 00A6 for Rev A6, 00B0 for Rev B, and 00C0 for Rev C.
Serial Number	12	Serial number of the board. This is a 12 character string which is: WWYY4P16nnnn where: WW = 2 digit week of the year of production YY = 2 digit year of production BBBK = BeagleBone AI-64 nnnn = incrementing board number
Configuration Option	32	Codes to show the configuration setup on this board. All FF
RSVD	6	FF FF FF FF FF FF
RSVD	6	FF FF FF FF FF FF
RSVD	6	FF FF FF FF FF FF
Available	4018	Available space for other non-volatile codes/data

[Figure 35](#) shows the new design on the EEPROM interface.

Figure 35. EEPROM Design Rev A5.

The EEPROM is accessed by the processor using the I₂C 0 bus. The **WP** pin is enabled by default. By grounding the test point, the write protection is removed.

The first 48 locations should not be written to if you choose to use the extra storage space in the EEPROM for other purposes. If you do, it could prevent the board from booting properly as the SW uses this information to determine how to set up the board.

10.5. 6.5 Micro Secure Digital

The microSD connector on the board will support a microSD card that can be used for booting or file storage on the BeagleBone AI-64.

6.5.1 microSD Design

[Figure 36](#) below is the design of the microSD interface on the board.

Figure 36. microSD Design.

The signals **MMC0-3** are the data lines for the transfer of data between the processor and the microSD connector.

The **MMC0_CLK** signal clocks the data in and out of the microSD card.

The **MMC0_CMD** signal indicates that a command versus data is being sent.

There is no separate card detect pin in the microSD specification. It uses **MMC0_DAT3** for that function. However, most microSD connectors still supply a CD function on the connectors. In the BeagleBone AI-64 design, this pin is connected to the **MMC0_SDCD** pin for use by the processor. You can also change the pin to **GPIO0_6**, which is able to wake up the processor from a sleep mode when an microSD card is inserted into the connector.

Pullup resistors are provided on the signals to increase the rise times of the signals to overcome PCB capacitance.

Power is provided from the **VDD_3V3B** rail and a 10uF capacitor is provided for filtering.

10.6. 6.6 User LEDs

There are four user LEDs on the BeagleBone AI-64. These are connected to GPIO pins on the processor. [Figure 37](#) shows the interfaces for the user LEDs.

Figure 37. User LEDs.

Resistors R71-R74 were changed to 4.75K on the revision A5B and later boards.

[Table 8](#) shows the signals used to control the four LEDs from the processor.

Table 7. Table 8. User LED Control Signals/Pins

LED	GPIO SIGNAL	PROC PIN
USR0	GPIO1_21	V15
USR1	GPIO1_22	U15
USR2	GPIO1_23	T15
USR3	GPIO1_24	V16

A logic level of "1" will cause the LEDs to turn on.

10.7. 6.7 Boot Configuration

The design supports two groups of boot options on the board. The user can switch between these modes via the Boot button. The primary boot source is the onboard eMMC device. By holding the Boot button, the user can force the board to boot from the microSD slot. This enables the eMMC to be overwritten when needed or to just boot an alternate image. The following sections describe how the boot configuration works.

In most applications, including those that use the provided demo distributions available from beagleboard.org,⁷ the processor-external boot code is composed of two stages. After the primary boot code in the processor ROM passes control, a secondary stage (secondary program loader—"SPL" or "MLO") takes over. The SPL stage initializes only the required devices to continue the boot process, and then control is transferred to the third stage "U-boot". Based on the settings of the boot pins, the ROM knows where to go and get the SPL and UBoot code. In the case of the BeagleBone AI-64, that is either eMMC or microSD based on the position of the boot switch.

⁷ <http://beagleboard.org/>

6.7.1 Boot Configuration Design

[Figure 38](#) shows the circuitry that is involved in the boot configuration process. On power up, these pins are read by the processor to determine the boot order. S2 is used to change the level of one bit from HI to LO which changes the boot order.

Figure 38. Processor Boot Configuration Design.

It is possible to override these setting via the expansion headers. But be careful not to add too much load such that it could interfere with the operation of the display interface or LCD panels. If you choose to override these settings, it is strongly recommended that you gate these signals with the **SYS_RESETn** signal. This ensures that after coming out of reset these signals are removed from the expansion pins.

10.8. 6.8 Default Boot Options

Based on the selected option found in [Figure 39](#) below, each of the boot sequences for each of the two settings is shown.

Figure 39. Processor Boot Configuration.

The first row in [Figure 39](#) is the default setting. On boot, the processor will look for the eMMC on the MMC1 port first, followed by the microSD slot on MMC0, USB0 and UART0. In the event there is no microSD card and the eMMC is empty, UART0 or USB0 could be used as the board source.

If you have a microSD card from which you need to boot from, hold the boot button down. On boot, the processor will look for the SPIO0 port first, then microSD on the MMC0 port, followed by USB0 and UART0. In the event there is no microSD card and the eMMC is empty, USB0 or UART0 could be used as the board source.

10.9. 6.9 10/100 Ethernet

The BeagleBone AI-64 is equipped with a 10/100 Ethernet interface. It uses the same PHY as is used on the original BeagleBone. The design is described in the following sections.

6.9.1 Ethernet Processor Interface

[Figure 40](#) shows the connections between the processor and the PHY. The interface is in the MII mode of operation.

Figure 40. Ethernet Processor Interface.

This is the same interface as is used on the BeagleBone. No changes were made in this design for the board.

6.9.2 Ethernet Connector Interface

The off board side of the PHY connections are shown in [Figure 41](#) below.

Figure 41. Ethernet Connector Interface.

This is the same interface as is used on the BeagleBone. No changes were made in this design for the board.

6.9.3 Ethernet PHY Power, Reset, and Clocks

[Figure 42](#) shows the power, reset, and lock connections to the **LAN8710A** PHY. Each of these areas is discussed in more detail in the following sections.

Figure 42. Ethernet PHY, Power, Reset, and Clocks.

6.9.3.1 VDD_3V3B Rail

The **VDD_3V3B** rail is the main power rail for the **LAN8710A**. It originates at the **VD_3V3B** regulator and is the primary rail that supports all of the peripherals on the board. This rail also supplies the **VDDIO** rails which set the voltage levels for all of the I/O signals between the processor and the **LAN8710A**.

6.9.3.2 VDD_PHYA Rail

A filtered version of **VDD_3V3B** rail is connected to the **VDD** rails of the **LAN8710** and the termination resistors on the Ethernet signals. It is labeled as **VDD_PHYA**.

The filtering inductor helps block transients that may be seen on the VDD_3V3B rail.

6.9.3.3 **PHY_VDDCR** Rail

The **PHY_VDDCR** rail originates inside the LAN8710A. Filter and bypass capacitors are used to filter the rail. Only circuitry inside the LAN8710A uses this rail.

6.9.3.4 **SYS_RESET**

The reset of the LAN8710A is controlled via the **SYS_RESETn** signal, the main board reset line.

6.9.3.5 **Clock Signals**

A crystal is used to create the clock for the LAN8710A. The processor uses the **RMII_RXCLK** signal to provide the clocking for the data between the processor and the LAN8710A.

6.9.4 **LAN8710A Mode Pins**

There are mode pins on the LAN8710A that sets the operational mode for the PHY when coming out of reset. These signals are also used to communicate between the processor and the LAN8710A. As a result, these signals can be driven by the processor which can cause the PHY not to be initialized correctly. To ensure that this does not happen, three low value pull up resistors are used. **Figure 43** below shows the three mode pin resistors.

Figure 43. Ethernet PHY Mode Pins.

This will set the mode to be 111, which enables all modes and enables auto-negotiation.

10.10. 6.10 Display Port Interface

The BeagleBone AI-64 has an onboard Display Port framer that converts the LCD signals and audio signals to drive a Display Port monitor. The design uses the on chip internal Display Port Framer.

The following sections provide more detail into the design of this interface.

6.10.1 Supported Resolutions

The maximum resolution supported by the BeagleBone AI-64 is 1280x1024 @ 60Hz. **Table 9** below shows the supported resolutions. Not all resolutions may work on all monitors, but these have been tested and shown to work on at least one monitor. EDID is supported on the BeagleBone AI-64. Based on the EDID reading from the connected monitor, the highest compatible resolution is selected.

Table 8. Table 9. HDMI Supported Monitor Adapter Resolutions

RESOLUTION	AUDIO
800 x 600 @60Hz	
800 x 600 @56Hz	
640 x 480 @75Hz	
640 x 480 @60Hz	YES
720 x 400 @70Hz	
1280 x 1024 @75Hz	
1024 x 768 @75Hz	
1024 x 768 @70Hz	
1024 x 768 @60Hz	
800 x 600 @75Hz	
800 x 600 @72Hz	
720 x 480 @60Hz	YES
1280 x 720 @60Hz	YES
1920x1080@24Hz	YES



The updated software image used on the Rev A5B and later boards added support for 1920x1080@24HZ.

Audio is limited to CEA supported resolutions. LCD panels only activate the audio in CEA modes. This is a function of the specification and is not something that can be fixed on the board via a hardware change or a software change.

6.10.2 Display Port Framer

insert processor Display Port framer doc here

6.10.3 Display Port Video Processor Interface

insert processor Display Port V-interface doc here

6.10.4 Display Port Control Processor Interface

insert processor Display Port C-interface doc here

6.10.5 Interrupt Signal

insert processor Display Port interrupt doc here

6.10.6 Audio Interface

insert processor Display Port audio doc here

6.10.7 Power Connections

guesing this doesn't exist on this device

6.10.8 uDP Connector Interface

insert processor Micro Display Port connector doc here

10.11. 6.11 USB Host

The board is equipped with a single USB host interface accessible from a single USB Type A female connector. [Figure 48](#) is the design of the USB Host circuitry.

Figure 48. USB Host circuit.

6.11.1 Power Switch

U8 is a switch that allows the power to the connector to be turned on or off by the processor. It also has an over current detection that can alert the processor if

the current gets too high via the **USB1_OC** signal. The power is controlled by the **USB1_DRVBUS** signal from the processor.

6.11.2 ESD Protection

U9 is the ESD protection for the signals that go to the connector.

6.11.3 Filter Options

FB7 and **FB8** were added to assist in passing the FCC emissions test. The **USB1_VBUS** signal is used by the processor to detect that the 5V is present on the connector. **FB7** is populated and **FB8** is replaced with a .1 ohm resistor.

10.12. 6.12 PRU-ICSS

The PRU-ICSS module is located inside the TDA4VM processor. Access to these pins is provided by the expansion headers and is multiplexed with other functions on the board. Access is not provided to all of the available pins.

All documentation is located at http://github.com/beagleboard/am335x_pru_package.⁸ This feature is not supported by Texas Instruments.

6.12.1 PRU-ICSS Features

The features of the PRU-ICSS include:

Two independent programmable real-time (PRU) cores:

- 32-Bit Load/Store RISC architecture
- 8K Byte instruction RAM (2K instructions) per core
- 8K Bytes data RAM per core
- 12K Bytes shared RAM
- Operating frequency of 200 MHz
- PRU operation is little endian similar to ARM processor
- All memories within PRU-ICSS support parity
- Includes Interrupt Controller for system event handling

⁸ http://github.com/beagleboard/am335x_pru_package

- Fast I/O interface
 - 16 input pins and 16 output pins per PRU core. (Not all of these are accessible on the BeagleBone AI-64).

6.12.2 PRU-ICSS Block Diagram

[Figure 49](#) is a high level block diagram of the PRU-ICSS.

Figure 49. PRU-ICSS Block Diagram.

6.12.3 PRU-ICSS Pin Access

Both PRU 0 and PRU1 are accessible from the expansion headers. Some may not be useable without first disabling functions on the board like LCD for example. Listed below is what ports can be accessed on each PRU.

PRU0

- 8 outputs or 9 inputs

PRU1

- 13 outputs or 14 inputs
- UART0_TXD, UART0_RXD, UART0_CTS, UART0_RTS

[Table 11](#) below shows which PRU-ICSS signals can be accessed on the BeagleBone AI-64 and on which connector and pins they are accessible from. Some signals are accessible on the same pins.

Table 9. Table 11. PRU0 and PRU1 Access

	PIN	PROC	NAME			
P8	11	R12	GPIO1_13		pr1_pru0_pru_r30_15 (Output)	
	12	T12	GPIO1_12		pr1_pru0_pru_r30_14 (Output)	
	15	U13	GPIO1_15		pr1_pru0_pru_r31_15 (Input)	

	16	V13	GPIO1_14	pr1_pru0_pru_r31_14 (Input)
	20	V9	GPIO1_31 pr1_pru1_ppu1r30u13pru_r31_13 (Output) (INPUT)	
	21	U9	GPIO1_30 pr1_pru1_ppu1r30u12pru_r31_12 (Output) (INPUT)	
	27	U5	GPIO2_22 pr1_pru1_ppu1r30u8_pru_r31_8 (Output) (INPUT)	
	28	V5	GPIO2_24 pr1_pru1_ppu1r30u10pru_r31_10 (Output) (INPUT)	
	29	R5	GPIO2_23 pr1_pru1_ppu1r30u9_pru_r31_9 (Output) (INPUT)	
	39	T3	GPIO2_12 pr1_pru1_ppu1r30u6_pru_r31_6 (Output) (INPUT)	
	40	T4	GPIO2_13 pr1_pru1_ppu1r30u7_pru_r31_7 (Output) (INPUT)	
	41	T1	GPIO2_10 pr1_pru1_ppu1r30u4_pru_r31_4 (Output) (INPUT)	
	42	T2	GPIO2_11 pr1_pru1_ppu1r30u5_pru_r31_5 (Output) (INPUT)	
	43	R3	GPIO2_8 pr1_pru1_ppu1r30u2_pru_r31_2 (Output) (INPUT)	
	44	R4	GPIO2_9 pr1_pru1_ppu1r30u3_pru_r31_3 (Output) (INPUT)	
	45	R1	GPIO2_6 pr1_pru1_ppu1r30u0_pru_r31_0 (Output) (INPUT)	
	46	R2	GPIO2_7 pr1_pru1_ppu1r30u1_pru_r31_1 (Output) (INPUT)	
P9	17	A16	I2C1_SCL	pr1_uart0_txd
	18	B16	I2C1_SDA	pr1_uart0_rxd

	19	D17	I2C2_SCL	pr1_uart0_rts_n		
	20	D18	I2C2_SDA	pr1_uart0_cts_n		
	21	B17	UART2_TX	pr1_uart0_rts_n		
	22	A17	UART2_RX	pr1_uart0_cts_n		
	24	D15	UART1_TX	pr1_uart0_t <u>pr1_pru0_pru_r31_16</u> (Input)		
	25	A14	GPIO3_21	^a <u>pr1_pru0_pru1_pru0_05_pru_r31_5</u> ((Input) (Output)		
	26	D16	UART1_RX	pr1_uart0_p <u>pr1_pru1_pru_r31_16</u>		
	27	C13	GPIO3_19	<u>pr1_pru0_pru1_pru0_07_pru_r31_7</u> (Output) (Input)		
	28	C12	SPI1_CS0	eCAP2_in_p <u>pr1_pru0_pru1_pru0_08_pru_r31_3</u> (Output) (Input)		
	29	B13	SPI1_D0	<u>pr1_pru0_pru1_pru0_01_pru_r31_1</u> (Output) (Input)		
	30	D12	SPI1_D1	<u>pr1_pru0_pru1_pru0_02_pru_r31_2</u> (Output) (Input)		
	31	A13	SPI1_SCLK	<u>pr1_pru0_pru1_pru0_00_pru_r31_0</u> (Output) (Input)		

^a GPIO3_21 is also the 24.576MHZ clock input to the processor to enable HDMI audio. To use this pin the oscillator must be disabled.

11. 7.0 Connectors

This section describes each of the connectors on the board.

11.1. 7.1 Expansion Connectors

The expansion interface on the board is comprised of two 46 pin connectors. All signals on the expansion headers are 3.3V unless otherwise indicated.

NOTE: Do not connect 5V logic level signals to these pins or the board will be damaged. ""

NOTE: DO NOT APPLY VOLTAGE TO ANY I/O PIN WHEN POWER IS NOT SUPPLIED TO THE BOARD. IT WILL DAMAGE THE PROCESSOR AND VOID THE WARRANTY.

NO PINS ARE TO BE DRIVEN UNTIL AFTER THE SYS_RESET LINE GOES HIGH.

Figure 50 shows the location of the expansion connectors.

Figure 50. Expansion Connector Location.

The location and spacing of the expansion headers are the same as on the original BeagleBone.

7.1.1 Connector P8 and P9

??? shows the pin bindings for **P8** and **P9** expansion headers. Signals can be connected to these connectors based on setting the pin mux on the processor, but this is the default settings on power up. The SW is responsible for setting the default function of each pin. There are some signals that have not been listed here. Refer to the processor documentation for more information on these pins and detailed descriptions of all of the pins listed. In some cases there may not be enough signals to complete a group of signals that may be required to implement a total interface.

The **BALL NUMBER** Identifier is the pin number in the processor documentation.

The **PIN No.** column is the pin number on the expansion header.

The **ADDRESS** column is the pin CONFIGURATION address??? for each pin.

The **MUXMODE[14:0] SETTINGS** are the possible pin configurations.

NOTE: DO NOT APPLY VOLTAGE TO ANY I/O PIN WHEN POWER IS NOT SUPPLIED TO THE BOARD. IT WILL DAMAGE THE PROCESSOR AND VOID THE WARRANTY.

NO PINS ARE TO BE DRIVEN UNTIL AFTER THE SYS_RESET LINE GOES HIGH.

PIN ADDRESS BAYER MODE[14:0]																		
No.	NAME	NUMBER	NAME	NUMBER	NAME	NUMBER	NAME	NUMBER	NAME	NUMBER	NAME	NUMBER	NAME	NUMBER	NAME	NUMBER	Bootstrap	
P8_080	PA10DIR	0	PR0	0	PR1	0	PR2	0	PR3	0	PR4	0	PR5	0	PR6	0	PR7	0
P8_040	PA10DIR	1	PR0	1	PR1	1	PR2	1	PR3	1	PR4	1	PR5	1	PR6	1	PR7	1
P8_050	PA10DIR	2	PR0	2	PR1	2	PR2	2	PR3	2	PR4	2	PR5	2	PR6	2	PR7	2
P8_060	PA10DIR	3	PR0	3	PR1	3	PR2	3	PR3	3	PR4	3	PR5	3	PR6	3	PR7	3
P8_070	PA10DIR	4	PR0	4	PR1	4	PR2	4	PR3	4	PR4	4	PR5	4	PR6	4	PR7	4
P8_080	PA10DIR	5	PR0	5	PR1	5	PR2	5	PR3	5	PR4	5	PR5	5	PR6	5	PR7	5
P8_090	PA10DIR	6	PR0	6	PR1	6	PR2	6	PR3	6	PR4	6	PR5	6	PR6	6	PR7	6
P8_010	PA10DIR	7	PR0	7	PR1	7	PR2	7	PR3	7	PR4	7	PR5	7	PR6	7	PR7	7
P8_020	PA10DIR	8	PR0	8	PR1	8	PR2	8	PR3	8	PR4	8	PR5	8	PR6	8	PR7	8
P8_030	PA10DIR	9	PR0	9	PR1	9	PR2	9	PR3	9	PR4	9	PR5	9	PR6	9	PR7	9
P8_040	PA10DIR	10	PR0	10	PR1	10	PR2	10	PR3	10	PR4	10	PR5	10	PR6	10	PR7	10
P8_050	PA10DIR	11	PR0	11	PR1	11	PR2	11	PR3	11	PR4	11	PR5	11	PR6	11	PR7	11
P8_060	PA10DIR	12	PR0	12	PR1	12	PR2	12	PR3	12	PR4	12	PR5	12	PR6	12	PR7	12
P8_070	PA10DIR	13	PR0	13	PR1	13	PR2	13	PR3	13	PR4	13	PR5	13	PR6	13	PR7	13
P8_080	PA10DIR	14	PR0	14	PR1	14	PR2	14	PR3	14	PR4	14	PR5	14	PR6	14	PR7	14
P8_090	PA10DIR	15	PR0	15	PR1	15	PR2	15	PR3	15	PR4	15	PR5	15	PR6	15	PR7	15
P8_010	PA10DIR	16	PR0	16	PR1	16	PR2	16	PR3	16	PR4	16	PR5	16	PR6	16	PR7	16
P8_020	PA10DIR	17	PR0	17	PR1	17	PR2	17	PR3	17	PR4	17	PR5	17	PR6	17	PR7	17
P8_030	PA10DIR	18	PR0	18	PR1	18	PR2	18	PR3	18	PR4	18	PR5	18	PR6	18	PR7	18
P8_040	PA10DIR	19	PR0	19	PR1	19	PR2	19	PR3	19	PR4	19	PR5	19	PR6	19	PR7	19
P8_050	PA10DIR	20	PR0	20	PR1	20	PR2	20	PR3	20	PR4	20	PR5	20	PR6	20	PR7	20
P8_060	PA10DIR	21	PR0	21	PR1	21	PR2	21	PR3	21	PR4	21	PR5	21	PR6	21	PR7	21
P8_070	PA10DIR	22	PR0	22	PR1	22	PR2	22	PR3	22	PR4	22	PR5	22	PR6	22	PR7	22
P8_080	PA10DIR	23	PR0	23	PR1	23	PR2	23	PR3	23	PR4	23	PR5	23	PR6	23	PR7	23
P8_090	PA10DIR	24	PR0	24	PR1	24	PR2	24	PR3	24	PR4	24	PR5	24	PR6	24	PR7	24
P8_010	PA10DIR	25	PR0	25	PR1	25	PR2	25	PR3	25	PR4	25	PR5	25	PR6	25	PR7	25
P8_020	PA10DIR	26	PR0	26	PR1	26	PR2	26	PR3	26	PR4	26	PR5	26	PR6	26	PR7	26
P8_030	PA10DIR	27	PR0	27	PR1	27	PR2	27	PR3	27	PR4	27	PR5	27	PR6	27	PR7	27
P8_040	PA10DIR	28	PR0	28	PR1	28	PR2	28	PR3	28	PR4	28	PR5	28	PR6	28	PR7	28
P8_050	PA10DIR	29	PR0	29	PR1	29	PR2	29	PR3	29	PR4	29	PR5	29	PR6	29	PR7	29
P8_060	PA10DIR	30	PR0	30	PR1	30	PR2	30	PR3	30	PR4	30	PR5	30	PR6	30	PR7	30
P8_070	PA10DIR	31	PR0	31	PR1	31	PR2	31	PR3	31	PR4	31	PR5	31	PR6	31	PR7	31
P8_080	PA10DIR	32	PR0	32	PR1	32	PR2	32	PR3	32	PR4	32	PR5	32	PR6	32	PR7	32
P8_090	PA10DIR	33	PR0	33	PR1	33	PR2	33	PR3	33	PR4	33	PR5	33	PR6	33	PR7	33
P8_010	PA10DIR	34	PR0	34	PR1	34	PR2	34	PR3	34	PR4	34	PR5	34	PR6	34	PR7	34
P8_020	PA10DIR	35	PR0	35	PR1	35	PR2	35	PR3	35	PR4	35	PR5	35	PR6	35	PR7	35
P8_030	PA10DIR	36	PR0	36	PR1	36	PR2	36	PR3	36	PR4	36	PR5	36	PR6	36	PR7	36
P8_040	PA10DIR	37	PR0	37	PR1	37	PR2	37	PR3	37	PR4	37	PR5	37	PR6	37	PR7	37
P8_050	PA10DIR	38	PR0	38	PR1	38	PR2	38	PR3	38	PR4	38	PR5	38	PR6	38	PR7	38
P8_060	PA10DIR	39	PR0	39	PR1	39	PR2	39	PR3	39	PR4	39	PR5	39	PR6	39	PR7	39
P8_070	PA10DIR	40	PR0	40	PR1	40	PR2	40	PR3	40	PR4	40	PR5	40	PR6	40	PR7	40
P8_080	PA10DIR	41	PR0	41	PR1	41	PR2	41	PR3	41	PR4	41	PR5	41	PR6	41	PR7	41
P8_090	PA10DIR	42	PR0	42	PR1	42	PR2	42	PR3	42	PR4	42	PR5	42	PR6	42	PR7	42
P8_010	PA10DIR	43	PR0	43	PR1	43	PR2	43	PR3	43	PR4	43	PR5	43	PR6	43	PR7	43
P8_020	PA10DIR	44	PR0	44	PR1	44	PR2	44	PR3	44	PR4	44	PR5	44	PR6	44	PR7	44
P8_030	PA10DIR	45	PR0	45	PR1	45	PR2	45	PR3	45	PR4	45	PR5	45	PR6	45	PR7	45
P8_040	PA10DIR	46	PR0	46	PR1	46	PR2	46	PR3	46	PR4	46	PR5	46	PR6	46	PR7	46
P8_050	PA10DIR	47	PR0	47	PR1	47	PR2	47	PR3	47	PR4	47	PR5	47	PR6	47	PR7	47
P8_060	PA10DIR	48	PR0	48	PR1	48	PR2	48	PR3	48	PR4	48	PR5	48	PR6	48	PR7	48
P8_070	PA10DIR	49	PR0	49	PR1	49	PR2	49	PR3	49	PR4	49	PR5	49	PR6	49	PR7	49
P8_080	PA10DIR	50	PR0	50	PR1	50	PR2	50	PR3	50	PR4	50	PR5	50	PR6	50	PR7	50
P8_090	PA10DIR	51	PR0	51	PR1	51	PR2	51	PR3	51	PR4	51	PR5	51	PR6	51	PR7	51
P8_010	PA10DIR	52	PR0	52	PR1	52	PR2	52	PR3	52	PR4	52	PR5	52	PR6	52	PR7	52
P8_020	PA10DIR	53	PR0	53	PR1	53	PR2	53	PR3	53	PR4	53	PR5	53	PR6	53	PR7	53
P8_030	PA10DIR	54	PR0	54	PR1	54	PR2	54	PR3	54	PR4	54	PR5	54	PR6	54	PR7	54
P8_040	PA10DIR	55	PR0	55	PR1	55	PR2	55	PR3	55	PR4	55	PR5	55	PR6	55	PR7	55
P8_050	PA10DIR	56	PR0	56	PR1	56	PR2	56	PR3	56	PR4	56	PR5	56	PR6	56	PR7	56
P8_060	PA10DIR	57	PR0	57	PR1	57	PR2	57	PR3	57	PR4	57	PR5	57	PR6	57	PR7	57
P8_070	PA10DIR	58	PR0	58	PR1	58	PR2	58	PR3	58	PR4	58	PR5	58	PR6	58	PR7	58
P8_080	PA10DIR	59	PR0	59	PR1	59	PR2	59	PR3	59	PR4	59	PR5	59	PR6	59	PR7	59
P8_090	PA10DIR	60	PR0	60	PR1	60	PR2	60	PR3	60	PR4	60	PR5	60	PR6	60	PR7	60
P8_010	PA10DIR	61	PR0	61	PR1	61	PR2	61	PR3	61	PR4	61	PR5	61	PR6	61	PR7	61
P8_020	PA10DIR	62	PR0	62	PR1	62	PR2	62	PR3	62	PR4	62	PR5	62	PR6	62	PR7	62
P8_030	PA10DIR	63	PR0	63	PR1	63	PR2	63	PR3	63	PR4	63	PR5	63	PR6	63	PR7	63
P8_040	PA10DIR	64	PR0	64	PR1	64	PR2	64	PR3	64	PR4	64	PR5	64	PR6	64	PR7	64
P8_050	PA10DIR	65	PR0	65	PR1	65	PR2	65	PR3	65	PR4	65	PR5	65	PR6	65	PR7	65
P8_060	PA10DIR	66	PR0	66	PR1	66	PR2	66	PR3	66	PR4	66	PR5	66	PR6	66	PR7	66
P8_070	PA10DIR	67	PR0	67	PR1	67	PR2	67	PR3	67	PR4	67	PR5	67	PR6	67	PR7	67
P8_080	PA10DIR	68	PR0	68	PR1	68	PR2	68	PR3	68	PR4	68	PR5	68	PR6	68	PR7	68
P8_090	PA10DIR	69	PR0	69	PR1	69	PR2	69	PR3	69	PR4	69	PR5	69	PR6	69	PR7	69
P8_010	PA10DIR	70	PR0	70	PR1	70	PR2	70	PR3	70	PR4	70	PR5	70	PR6	70	PR7	70
P8_020	PA10DIR	71	PR0	71	PR1	71	PR2	71	PR3	71	PR4	71	PR5	71	PR6	71	PR7	71
P8_030	PA10DIR	72	PR0	72	PR1	72	PR2	72	PR3	72	PR4	72	PR5	72	PR6	72	PR7	72
P8_040	PA10DIR	73	PR0	73	PR1	73	PR2	73	PR3	73	PR4	73	PR5	73	PR6	73	PR7	73
P8_050	PA10DIR	74	PR0	74	PR1	74	PR2	74	PR3	74	PR4	74	PR5	74	PR6	74	PR7	74
P8_060	PA10DIR	75	PR0	75	PR1	75	PR2	75	PR3	75	PR4	75	PR5	75	PR6	75	PR7	75
P8_070	PA10DIR	76	PR0	76	PR1	76	PR2	76	PR3	76	PR4	76	PR5	76	PR6	76	PR7	76
P8_080	PA10DIR	77	PR0	77	PR1	77	PR2	77	PR3	77	PR4	77	PR5	77	PR6	77	PR7	77
P8_090	PA10DIR	78	PR0	78	PR1	78	PR2	78	PR3	78	PR4	78	PR5	78	PR6	78	PR7	78
P8_010	PA10DIR	79	PR0	79	PR1	79	PR2	79	PR3	79	PR4	79	PR5	79	PR6	79	PR7	79
P8_020	PA10DIR	80	PR0	80	PR1	80	PR2	80	PR3	80	PR4	80	PR5	80	PR6	80	PR7	80
P8_030	PA10DIR	81	PR0	81	PR1	81	PR2	81	PR3	81	PR4	81	PR5	81	PR6	81	PR7	81
P8_040	PA10DIR	82	PR0	82	PR1	82	PR2	82	PR3	82	PR4	82	PR5	82	PR6	82	PR7	82
P8_050	PA10DIR	83	PR0	83	PR1	83	PR2	83</td										

11.2. 7.2 Power Jack

The DC power jack is located next to the RJ45 Ethernet connector as shown in [Figure 51](#). This uses the same power connector as is used on the original BeagleBone. The connector has a 2.1mm diameter center post (5VDC) and a 5.5mm diameter outer dimension on the barrel (GND).

Figure 51. 5VDC Power Jack.

The board requires a regulated 5VDC +/- .25V supply at 1A. A higher current rating may be needed if capes are plugged into the expansion headers. Using a higher current power supply will not damage the board.

11.3. 7.3 USB Client

The USB Client connector is accessible on the bottom side of the board under the row of four LEDs as shown in [Figure 52](#). It uses a 5 pin miniUSB cable, the same as is used on the original BeagleBone. The cable is provided with the board. The cable can also be used to power the board.

Figure 52. USB Client.

This port is a USB Client only interface and is intended for connection to a PC.

7.4 USB Host

There is a single USB Host connector on the board and is shown in [Figure 53](#) below.

Figure 53. USB Host Connector

The port is USB 2.0 HS compatible and can supply up to 500mA of current. If more current or ports is needed, then a HUB can be used.

7.5 Serial Header

Each board has a debug serial interface that can be accessed by using a special serial cable that is plugged into the serial header as shown in [Figure 54](#) below.

Figure 54. Serial Debug Header

Two signals are provided, TX and RX on this connector. The levels on these signals are 3.3V. In order to access these signals, a FTDI USB to Serial cable is recommended as shown in [Figure 55](#) below.

The cable can be purchased from several different places and must be the 3.3V version TTL-232R-3V3. Information on the cable itself can be found direct from FTDI at:

http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL232R_CABLES.pdf⁹

Pin 1 of the cable is the ai-64 wire. That must align with the pin 1 on the board which is designated by the white dot next to the connector on the board.

Refer to the support WIKI <http://elinux.org/BeagleBoneBlack> for more sources of this cable and other options that will work.

Table is the pinout of the connector as reflected in the schematic. It is the same as the

FTDI cable which can be found at

http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf

with the exception that only three pins are used on the board. The pin numbers are defined in **Table 14**. The signals are from the perspective of the board.

Table 14. J1 Serial Header Pins

PIN NUMBER	SIGNAL
1	Ground
4	Receive
5	Transmit

Figure 56 shows the pin location on the board.

Figure 56. Serial Header

⁹ http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf

7.6 HDMI

Access to the HDMI interface is through the HDMI connector that is located on the bottom side of the board as shown in **Figure 57** below.

Figure 57. HDMI Connector

The connector is microHDMI connector. This was done due to the space limitations we had in finding a place to fit the connector. It requires a microHDMI to HDMI cable as shown in **Figure 58** below. The cable can be purchased from several different sources.

Figure 58. HDMI Cable

7.7 microSD

A microSD connector is located on the back or bottom side of the board as shown in **Figure 59** below. The microSD card is not supplied with the board.

Figure 59. microSD Connector

When plugging in the SD card, the writing on the card should be up. Align the card with the connector and push to insert. Then release. There should be a click and the card will start to eject slightly, but it then should latch into the connector. To eject the card, push the SD card in and then remove your finger. The SD card will be ejected from the connector.

Do not pull the SD card out or you could damage the connector.

7.8 Ethernet

The board comes with a single 10/100 Ethernet interface located next to the power jack as shown in **Figure 60**.

Figure 60. Ethernet Connector

The PHY supports AutoMDX which means either a straight or a swap cable can be used

7.9 JTAG Connector

A place for an optional 20 pin CTI JTAG header is provided on the board to facilitate the SW development and debugging of the board by using various JTAG emulators. This header is not supplied standard on the board. To use this, a connector will need to be soldered onto the board.

If you need the JTAG connector you can solder it on yourself. No other components are needed. The connector is made by Samtec and the part number is FTR-110-03-G-D-06. You can purchase it from www.digikey.com.¹⁰

12. 8.0 Cape Board Support

The BeagleBone AI-64 has the ability to accept up to four expansion boards or capes that can be stacked onto the expansion headers. The word cape comes from the shape of the board as it is fitted around the Ethernet connector on the main board. This notch acts as a key to ensure proper orientation of the cape.

This section describes the rules for creating capes to ensure proper operation with the BeagleBone AI-64 and proper interoperability with other capes that are intended to coexist with each other. Co-existence is not a requirement and is in itself, something that is impossible to control or administer. But, people will be able to create capes that operate with other capes that are already available based on public information as it pertains to what pins and features each cape uses. This information will be able to be read from the EEPROM on each cape.

This section is intended as a guideline for those wanting to create their own capes. Its intent is not to put limits on the creation of capes and what they can do, but to set a few basic rules that will allow the SW to administer their operation with the BeagleBone AI-64. For this reason there is a lot of flexibility in the specification that we hope most people will find liberating and in the spirit of Open Source

¹⁰ <http://www.digikey.com/>

Hardware. I am sure there are others that would like to see tighter control, more details, more rules and much more order to the way capes are handled.

Over time, this specification will change and be updated, so please refer to the latest version of this manual prior to designing your own capes to get the latest information.

**DO NOT APPLY VOLTAGE TO ANY I/O PIN WHEN
POWER IS NOT SUPPLIED TO THE BOARD. IT WILL DAMAGE THE
PROCESSOR AND VOID THE WARRANTY.**
**NO PINS ARE TO BE DRIVEN UNTIL AFTER THE SYS_RESET LINE GOES
HIGH.**

12.1. 8.1 BeagleBone AI-64 Cape Compatibility

The main expansion headers are the same between the BeagleBone and BeagleBone AI-64. While the pins are the same, some of these pins are now used on the BeagleBone AI-64. The following sections discuss these pins.

The Power Expansion header was removed from the BeagleBone AI-64 and is not available.

PAY VERY CLOSE ATTENTION TO THIS SECTION AND READ CAREFULLY!!

8.1.1 LCD Pins

The LCD pins are used on the BeagleBone AI-64 to drive the HDMI framer. These signals are listed in **Table 15** below.

Table 15. P8 LCD Conflict Pins

PIN	PROC	NAME	MODE0
27	U5	GPIO2_22	lcd_vsync
28	V5	GPIO2_24	lcd_pclk
29	R5	GPIO2_23	lcd_hsync
30	R6	GPIO2_25	lcd_ac_bias_en
31	V4	UART5_CTSN	lcd_data14

PIN	PROC	NAME	MODE0
32	T5	UART5_RTSN	lcd_data15
33	V3	UART4_RTSN	lcd_data13
34	U4	UART3_RTSN	lcd_data11
35	V2	UART4_CTSN	lcd_data12
36	U3	UART3_CTSN	lcd_data10
37	U1	UART5_TXD	lcd_data8
38	U2	UART5_RXD	lcd_data9
39	T3	GPIO2_12	lcd_data6
40	T4	GPIO2_13	lcd_data7
41	T1	GPIO2_10	lcd_data4
42	T2	GPIO2_11	lcd_data5
43	R3	GPIO2_8	lcd_data2
44	R4	GPIO2_9	lcd_data3
45	R1	GPIO2_6	lcd_data0
46	R2	GPIO2_7	lcd_data1

If you are using these pins for other functions, there are a few things to keep in mind:

- On the HDMI Framer, these signals are all inputs so the framer will not be driving these pins.
- The HDMI framer will add a load onto these pins.
- There are small filter caps on these signals which could also change the operation of these pins if used for other functions.
- When used for other functions, the HDMI framer cannot be used.
- There is no way to power off the framer as this would result in the framer being powered through these input pins which would not be a good idea.
- These pins are also the **SYSBOOT** pins. DO NOT drive them before the **SYS_RESETN** signal goes high. If you do, the board may not boot because you would be changing the boot order of the processor.

In order to use these pins, the SW will need to reconfigure them to whatever function you need the pins to do. To keep power low, the HDMI framer should be put in a low power mode via the SW using the **I2C0** interface.

8.1.2 eMMC Pins

The BeagleBone AI-64 uses 10 pins to connect to the processor that also connect to the

P8 expansion connector. These signals are listed below in **Table 16**. The proper mode is

MODE2.

Table 16. P8 eMMC Conflict Pins

PIN	PROC	SIGNAL	MODE
22	V8	MMC1_DAT5	1
23	U8	MMC1_DAT4	1
24	V7	MMC1_DAT1	1
5	R8	MMC1_DAT2	1
4	T9	MMC1_DAT7	1
3	R9	MMC1_DAT6	1
6	T8	MMC1_DAT3	1
25	U7	MMC1_DAT0	1
20	V9	MMC1_CMD	2
21	U9	MMC1_CLK	2

If using these pins, several things need to be kept in mind when doing so:

- On the eMMC device, these signals are inputs and outputs.
- The eMMC device will add a load onto these pins.
- When used for other functions, the eMMC cannot be used. This means you must boot from the microSD slot.

- If using these pins, you need to put the eMMC into reset. This requires that the eMMC be accessible from the processor in order to set the eMMC to accept the eMMC pins.
- DO NOT drive the eMMC pins until the eMMC has been put into reset. This means that if you choose to use these pins, they must not drive any signal until enabled via Software. This requires a buffer or some other form of hold off function enabled by a GPIO pin on the expansion header.

On power up, the eMMC is NOT reset. If you hold the Boot button down, this will force a boot from the microSD. This is not convenient when a cape is plugged into the board. There are two solutions to this issue:

1. Wipe the eMMC clean. This will cause the board to default to microSD boot. If you want to use the eMMC later, it can be reprogrammed.
2. You can also tie LCD_DATA2 low on the cape during boot. This will be the same as if you were holding the boot button. However, in order to prevent unforeseen issues, you need to gate this signal with RESET, when the data is sampled. After reset goes high, the signal should be removed from the pin.

BEFORE the SW reinitializes the pins, it *MUST* put the eMMC in reset. This is done by taking eMMC_RSTn (GPIO1_20) LOW **after** the eMMC has been put into a mode to enable the reset line. This pin does not connect to the expansion header and is accessible only on the board.

DO NOT automatically drive any conflicting pins until the SW enables it. This puts the SW in control to ensure that the eMMC is in reset before the signals are used from the cape. You can use a GPIO pin for this. No, we will not designate a pin for this function. It will be determined on a cape by cape basis by the designer of the respective cape.

8.2 EEPROM

Each cape must have its own EEPROM containing information that will allow the SW to identify the board and to configure the expansion headers pins as needed. The one exception is proto boards intended for prototyping. They may or may not have an EEPROM on them. An EEPROM is required for all capes sold in order for them operate correctly when plugged into the BeagleBone AI-64.

The address of the EEPROM will be set via either jumpers or a dipswitch on each expansion board. **Figure 61** below is the design of the EEPROM circuit.

The EEPROM used is the same one as is used on the BeagleBone and the BeagleBone AI-64, a CAT24C256. The CAT24C256 is a 256 kb Serial CMOS EEPROM, internally organized as 32,768 words of 8 bits each. It features a 64-byte page write buffer and supports the Standard (100 kHz), Fast (400 kHz) and Fast-Plus (1 MHz) I2C protocol.

Figure 61. Expansion Board EEPROM Without Write Protect

The addressing of this device requires two bytes for the address which is not used on smaller size EEPROMs, which only require only one byte. Other compatible devices may be used as well. Make sure the device you select supports 16 bit addressing. The part package used is at the discretion of the cape designer.

8.2.1 EEPROM Address

In order for each cape to have a unique address, a board ID scheme is used that sets the address to be different depending on the setting of the dipswitch or jumpers on the capes. A two position dipswitch or jumpers is used to set the address pins of the EEPROM.

It is the responsibility of the user to set the proper address for each board and the position in the stack that the board occupies has nothing to do with which board gets first choice on the usage of the expansion bus signals. The process for making that determination and resolving conflicts is left up to the SW and, as of this moment in time, this method is a something of a mystery due to the new Device Tree methodology introduced in the 3.8 kernel.

Address line A2 is always tied high. This sets the allowable address range for the expansion cards to **0x54 to 0x57**. All other I2C addresses can be used by the user in the design of their capes. But, these addresses must not be used other than for the board EEPROM information. This also allows for the inclusion of EEPROM devices on the cape if needed without interfering with this EEPROM. It requires that A2 be grounded on the EEPROM not used for cape identification.

8.2.2 I2C Bus

The EEPROMs on each expansion board are connected to I2C2 on connector P9 pins 19 and 20. For this reason I2C2 must always be left connected and should not be changed by SW to remove it from the expansion header pin mux settings. If this is done, the system will be unable to detect the capes.

The I2C signals require pullup resistors. Each board must have a 5.6K resistor on these signals. With four capes installed this will result in an effective resistance of 1.4K if all capes were installed and all the resistors used were exactly 5.6K. As more capes are added the resistance is reduced to overcome capacitance added to the signals. When no capes are installed the internal pullup resistors must be activated inside the processor to prevent I2C timeouts on the I2C bus.

The I2C2 bus may also be used by capes for other functions such as I/O expansion or other I2C compatible devices that do not share the same address as the cape EEPROM.

8.2.3 EEPROM Write Protect

The design in **Figure 62** has the write protect disabled. If the write protect is not enabled, this does expose the EEPROM to being corrupted if the I2C2 bus is used on the cape and the wrong address written to. It is recommended that a write protection function be implemented and a Test Point be added that when grounded, will allow the EEPROM to be written to. To enable write operation, Pin 7 of the EEPROM must be tied to ground.

When not grounded, the pin is HI via pullup resistor R210 and therefore write protected. Whether or not Write Protect is provided is at the discretion of the cape designer.

Variable & MAC Memory

VDD_3V3B

Figure 62. Expansion Board EEPROM Write Protect

8.2.4 EEPROM Data Format

Table 17 shows the format of the contents of the expansion board EEPROM. Data is stored in Big Endian with the least significant value on the right. All addresses read as a single byte data from the EEPROM, but two byte addressing is used. ASCII values are intended to be easily read by the user when the EEPROM contents are dumped.

Table 17. Expansion Board EEPROM

Name	Offset	Size (bytes)	Contents
Header	0	4	0xAA, 0x55, 0x33, 0xEE
EEPROM Revision	4	2	Revision number of the overall format of this EEPROM in ASCII =A1
Board Name	6	32	Name of board in ASCII so user can read it when the EEPROM is dumped. Up to developer of the board as to what they call the board..
Version	38	4	Hardware version code for board in ASCII. Version format is up to the developer. i.e. 02.1...00A1... .10A0

Name	Offset	Size (bytes)	Contents
Manufacturer	42	16	ASCII name of the manufacturer. Company or individual's name.
Part Number	58	16	ASCII Characters for the part number. Up to maker of the board.
Number of Pins	74	2	Number of pins used by the daughter board including the power pins used. Decimal value of total pins 92 max, stored in HEX.
Serial Number	76	12	Serial number of the board. This is a 12 character string which is: WWYY&&&&nnnn where: WW = 2 digit week of the year of production YY = 2 digit year of production &&&=Assembly code to let the manufacturer

Name	Offset	Size (bytes)	Contents
			<p>document the assembly number or product. A way to quickly tell from reading the serial number what the board is. Up to the developer to determine. nnnn = incrementing board number for that week of production</p>
Pin Usage	88	148	<p>Two bytes for each configurable pins of the 74 pins on the expansion connectors MSB LSB</p> <p>Bit order: 15..14 ..1 1..0</p> <p>Bit 15....Pin is used or not...</p> <p>0=Unused by cape 1=Used by cape</p> <p>Bit 14-13...Pin Direction...</p> <p>..1 0=Output 01=Input 11=BDIR</p>

Name	Offset	Size (bytes)	Contents
			<p>Bits 12-7....</p> <p>Reserved.....</p> <p>..should be all zeros</p> <p>Bit 6....Slew Rate0=Fast 1=Slow</p> <p>Bit 5....Rx Enable.....</p> <p>.0=Disabled 1=Enabled</p> <p>Bit 4....Pull Up/ Dn Select... .0=Pulldown 1=PullUp</p> <p>Bit 3....Pull Up/ DN enabled... 0=Enabled 1=Disabled</p> <p>Bits 2-0 ... Mux Mode Selection... Mode 0-7</p>
VDD_3V3B Current	236	2	<p>Maximum current in milliamps. This is HEX value of the current in decimal</p> <p>1500mA=0x05 0xDC</p> <p>325mA=0x01 0x45</p>
VDD_5V Current	238	2	Maximum current in

Name	Offset	Size (bytes)	Contents
			milliamps. This is HEX value of the current in decimal 1500mA=0x05 0xDC 325mA=0x01 0x45
SYS_5V Current	240	2	Maximum current in milliamps. This is HEX value of the current in decimal 1500mA=0x05 0xDC 325mA=0x01 0x45
DC Supplied	242	2	Indicates whether or not the board is supplying voltage on the VDD_5V rail and the current rating 000=No 1-0xFFFF is the current supplied storing the decimal equivalent in HEX format
Available	244	32543	Available space for other

Name	Offset	Size (bytes)	Contents
			non-volatile codes/data to be used as needed by the manufacturer or SW driver. Could also store presets for use by SW.

8.2.5 Pin Usage

Table 18 is the locations in the EEPROM to set the I/O pin usage for the cape. It contains the value to be written to the Pad Control Registers. Details on this can be found in section 9.2.2 of the **TDA4VM Technical Reference Manual**. The table is left blank as a convenience and can be printed out and used as a template for creating a custom setting for each cape. The 16 bit integers and all 16 bit fields are to be stored in Big Endian format.

Bit 15 PIN USAGE is an indicator and should be a 1 if the pin is used or 0 if it is unused.

Bits 14-7 RESERVED is not to be used and left as 0.

Bit 6 SLEW CONTROL 0=Fast 1=Slow

Bit 5 RX Enabled 0=Disabled 1=Enabled

Bit 4 PU/PD 0=Pulldown 1=Pullup.

Bit 3 PULLUP/DN 0=Pullup/pulldown enabled

1= Pullup/pulldown disabled

Bit 2-0 MUX MODE SELECT Mode 0-7. (refer to TRM)

Refer to the TRM for proper settings of the pin MUX mode based on the signal selection to be used.

The **AIN0-6** pins do not have a pin mux setting, but they need to be set to indicate if each of the pins is used on the cape. Only bit 15 is used for the AIN signals.

Table 18. EEPROM Pin Usage

+	+	+	15	14	13	12	11	10	9	8	7	6	5
Off set	ConnName	Pin	Type	+	Reserved	+	+	S	R	P	P	Mux	Mode
			Usage					L	X	U	U	/	
88	P9-22UART2_RXD		+	+	+	+	+	+	+	+	+	+	+
90	P9-21UART2_TXD		+	+	+	+	+	+	+	+	+	+	+
92	P9-18I2C1_SDA	+	+	+	+	+	+	+	+	+	+	+	+
94	P9-17I2C1_SCL	+	+	+	+	+	+	+	+	+	+	+	+
96	P9-42GPIO0_7	+	+	+	+	+	+	+	+	+	+	+	+
98	P8-35UART4_CTSN		+	+	+	+	+	+	+	+	+	+	+
100	P8-33UART4_RTSN		+	+	+	+	+	+	+	+	+	+	+
102	P8-31UART5_CTSN		+	+	+	+	+	+	+	+	+	+	+
104	P8-32UART5_RTSN		+	+	+	+	+	+	+	+	+	+	+
106	P9-19I2C2_SCL	+	+	+	+	+	+	+	+	+	+	+	+
108	P9-20I2C2_SDA	+	+	+	+	+	+	+	+	+	+	+	+
110	P9-26UART1_RXD		+	+	+	+	+	+	+	+	+	+	+
112	P9-24UART1_TXD		+	+	+	+	+	+	+	+	+	+	+
114	P9-41CLKOUT2	+	+	+	+	+	+	+	+	+	+	+	+
116	P8-19EHRPWM2A		+	+	+	+	+	+	+	+	+	+	+
118	P8-13EHRPWM2B		+	+	+	+	+	+	+	+	+	+	+
120	P8-14GPIO0_26	+	+	+	+	+	+	+	+	+	+	+	+
122	P8-17GPIO0_27	+	+	+	+	+	+	+	+	+	+	+	+
124	P9-11UART4_RXD		+	+	+	+	+	+	+	+	+	+	+
126	P9-13UART4_TXD		+	+	+	+	+	+	+	+	+	+	+

	+	+	+	15	14	13	12	11	10	9	8	7	6	5
128	P8-25GPIO1_0	+	+	+	+	+	+	+	+	+	+	+	+	+
130	P8-24GPIO1_1	+	+	+	+	+	+	+	+	+	+	+	+	+
132	P8-5 GPIO1_2	+	+	+	+	+	+	+	+	+	+	+	+	+
134	P8-6 GPIO1_3	+	+	+	+	+	+	+	+	+	+	+	+	+
136	P8-23GPIO1_4	+	+	+	+	+	+	+	+	+	+	+	+	+
138	P8-22GPIO1_5	+	+	+	+	+	+	+	+	+	+	+	+	+
140	P8-3 GPIO1_6	+	+	+	+	+	+	+	+	+	+	+	+	+
142	P8-4 GPIO1_7	+	+	+	+	+	+	+	+	+	+	+	+	+
144	P8-12GPIO1_12	+	+	+	+	+	+	+	+	+	+	+	+	+
146	P8-11GPIO1_13	+	+	+	+	+	+	+	+	+	+	+	+	+
148	P8-16GPIO1_14	+	+	+	+	+	+	+	+	+	+	+	+	+
150	P8-15GPIO1_15	+	+	+	+	+	+	+	+	+	+	+	+	+
152	P9-15GPIO1_16	+	+	+	+	+	+	+	+	+	+	+	+	+

		15	14	13	12	11	10	9	8	7	6	5		
Off	ConnName	Pin	Type	Usage	Reserved				S	R	P	P	Mux	Mode
set								L	X	U	U	-	/	
154	P9-23GPIO1_17							E						
156	P9-14EHRPWM1A							W		P	D	E		
158	P9-16EHRPWM1B									D	N			
160	P9-12GPIO1_28													
162	P8-26GPIO1_29													
164	P8-21GPIO1_30													
166	P8-20GPIO1_31													
168	P8-18GPIO2_1													
170	P8-7 TIMER4													

			15	14	13	12	11	10	9	8	7	6	5
172	P8-9 TIMER5		+	+	+	+	+	+	+	+	+	+	+
174	P8-10TIMER6		+	+	+	+	+	+	+	+	+	+	+
176	P8-8 TIMER7		+	+	+	+	+	+	+	+	+	+	+
178	P8-45GPIO2_6		+	+	+	+	+	+	+	+	+	+	+
180	P8-46GPIO2_7		+	+	+	+	+	+	+	+	+	+	+
182	P8-43GPIO2_8		+	+	+	+	+	+	+	+	+	+	+
184	P8-44GPIO2_9		+	+	+	+	+	+	+	+	+	+	+
186	P8-41GPIO2_10	+	+	+	+	+	+	+	+	+	+	+	+
188	P8-42GPIO2_11	+	+	+	+	+	+	+	+	+	+	+	+
190	P8-39GPIO2_12	+	+	+	+	+	+	+	+	+	+	+	+
192	P8-40GPIO2_13	+	+	+	+	+	+	+	+	+	+	+	+
194	P8-37UART5_TXD		+	+	+	+	+	+	+	+	+	+	+
196	P8-38UART5_RXD		+	+	+	+	+	+	+	+	+	+	+
198	P8-36UART3_CTSN		+	+	+	+	+	+	+	+	+	+	+
200	P8-34UART3_RTSN		+	+	+	+	+	+	+	+	+	+	+
202	P8-27GPIO2_22	+	+	+	+	+	+	+	+	+	+	+	+
204	P8-29GPIO2_23	+	+	+	+	+	+	+	+	+	+	+	+
206	P8-28GPIO2_24	+	+	+	+	+	+	+	+	+	+	+	+
208	P8-30GPIO2_25	+	+	+	+	+	+	+	+	+	+	+	+
210	P9-29SPI1_D0	+	+	+	+	+	+	+	+	+	+	+	+
212	P9-30SPI1_D1	+	+	+	+	+	+	+	+	+	+	+	+
214	P9-28SPI1_CS0	+	+	+	+	+	+	+	+	+	+	+	+
216	P9-27GPIO3_19	+	+	+	+	+	+	+	+	+	+	+	+
218	P9-31SPI1_SCLK		+	+	+	+	+	+	+	+	+	+	+
220	P9-25GPIO3_21	+	+	+	+	+	+	+	+	+	+	+	+
+	+	+	15	14	13	12	11	10	9	8	7	6	5
Off	Conn	Name	Pin	Type	+	Reserved	+		S	R	P	P	Mux
set				Usage					L	X	U	U	Mode

			15	14	13	12	11	10	9	8	7	6	5
									E		-	/	
									W		P	D E	
										D	N		
+	+	+	+	0	0	0	0	0	0	0	0	0	0
222	P9-39AIN0	+											
224	P9-40AIN1	+											
226	P9-37AIN2	+											
228	P9-38AIN3	+											
230	P9-33AIN4	+											
232	P9-36AIN5	+											
234	P9-35AIN6	+											

8.3 Pin Usage Consideration

This section covers things to watch for when hooking up to certain pins on the expansion headers.

8.3.1 Boot Pins

There are 16 pins that control the boot mode of the processor that are exposed on the expansion headers. **Figure 63** below shows those signals as they appear on the BeagleBone AI-64.:

If you plan to use any of these signals, then on power up, these pins should not be driven. If you do, it can affect the boot mode of the processor and could keep the processor from booting or working correctly.

If you are designing a cape that is intended to be used as a boot source, such as a NAND board, then you should drive the pins to reconfigure the boot mode, but only at reset. After the reset phase, the signals should not be driven to allow them to be used for the other functions found on those pins. You will need to override the resistor values in order to change the settings. The DC pull-up requirement should be based on the TDA4VM Vih min voltage of 2 volts and TDA4VM maximum input

leakage current of 18uA. Also take into account any other current leakage paths on these signals which could be caused by your specific cape design.

The DC pull-down requirement should be based on the TDA4VM Vil max voltage of 0.8 volts and TDA4VM maximum input leakage current of 18uA plus any other current leakage paths on these signals.

8.4 Expansion Connectors

A combination of male and female headers is used for access to the expansion headers on the main board. There are three possible mounting configurations for the expansion headers:

- *Single*-no board stacking but can be used on the top of the stack.
- *Stacking*-up to four boards can be stacked on top of each other.
- *Stacking with signal stealing*-up to three boards can be stacked on top of each other, but certain boards will not pass on the signals they are using to prevent signal loading or use by other cards in the stack.

The following sections describe how the connectors are to be implemented and used for each of the different configurations.

8.4.1 Non-Stacking Headers-Single Cape

For non-stacking capes single configurations or where the cape can be the last board on the stack, the two 46 pin expansion headers use the same connectors.

Figure 64 is a picture of the connector. These are dual row 23 position 2.54mm x 2.54mm connectors.

Figure 64. Single Expansion Connector

The connector is typically mounted on the bottom side of the board as shown in **Figure 65**. These are very common connectors and should be easily located. You can also use two single row 23 pin headers for each of the dual row headers.

Figure 65. Single Cape Expansion Connector

It is allowed to only populate the pins you need. As this is a non-stacking configuration, there is no need for all headers to be populated. This can also reduce the overall cost of the cape. This decision is up to the cape designer.

For convenience listed in **Table 19** are some possible choices for part numbers on this connector. They have varying pin lengths and some may be more suitable than others for your use. It should be noted, that the longer the pin and the further it is inserted into the BeagleBone AI-64 connector, the harder it will be to remove due to the tension on 92 pins. This can be minimized by using shorter pins or removing those pins that are not used by your particular design. The first item in **Table 18** is on the edge and may not be the best solution. Overhang is the amount of the pin that goes past the contact point of the connector on the BeagleBone AI-64

Table 19. Single Cape Connectors

SUPPLIER	PARTNUMBER	TAIL LENGTH(in)	OVERHANG(in)
<i>Major League</i> ¹¹	TSHC-123-D-03-145-G-LF	.145	.004
<i>Major League</i> ¹²	TSHC-123-D-03-240-G-LF	.240	.099
<i>Major League</i> ¹³	TSHC-123-D-03-255-G-LF	.255	.114

The G in the part number is a plating option. Other options may be used as well as long as the contact area is gold. Other possible sources are Sullins and Samtec for these connectors. You will need to ensure the depth into the connector is sufficient

8.4.2 Main Expansion Headers-Stacking

For stacking configuration, the two 46 pin expansion headers use the same connectors. **Figure 66** is a picture of the connector. These are dual row 23 position 2.54mm x 2.54mm connectors.

¹¹ <http://www.mlelectronics.com/>

¹² <http://www.mlelectronics.com/>

¹³ <http://www.mlelectronics.com/>

Figure 66. Expansion Connector

The connector is mounted on the top side of the board with longer tails to allow insertion into the BeagleBone AI-64. **Figure 67** is the connector configuration for the connector.

Figure 67. Stacked Cape Expansion Connector

For convenience listed in **Table 18** are some possible choices for part numbers on this connector. They have varying pin lengths and some may be more suitable than others for your use. It should be noted, that the longer the pin and the further it is inserted into the BeagleBone AI-64 connector, the harder it will be to remove due to the tension on 92 pins. This can be minimized by using shorter pins. There are most likely other suppliers out there that will work for this connector as well. If anyone finds other suppliers of compatible connectors that work, let us know and they will be added to this document. The first item in**Table 19** is on the edge and may not be the best solution. Overhang is the amount of the pin that goes past the contact point of the connector on the BeagleBone AI-64.

The third part listed in **Table 20** will have insertion force issues.

Table 20. Stacked Cape Connectors

SUPPLIER	PARTNUMBER	TAIL LENGTH(in)	OVERHANG(in)
<i>Major League</i> ¹⁴	SSHQ-123-D-06-G-LF	.190	0.049
<i>Major League</i> ¹⁵	SSHQ-123-D-08-G-LF	.390	0.249
<i>Major League</i> ¹⁶	SSHQ-123-D-10-G-LF	.560	0.419

¹⁴ <http://www.mlelectronics.com/>

¹⁵ <http://www.mlelectronics.com/>

¹⁶ <http://www.mlelectronics.com/>

There are also different plating options on each of the connectors above. Gold plating on the contacts is the minimum requirement. If you choose to use a different part number for plating or availability purposes, make sure you do not select the “LT” option.

Other possible sources are Sullins and Samtec but make sure you select one that has the correct mating depth.

8.4.3 Stacked Capes w/Signal Stealing

Figure 68 is the connector configuration for stackable capes that does not provide all of the signals upwards for use by other boards. This is useful if there is an expectation that other boards could interfere with the operation of your board by exposing those signals for expansion. This configuration consists of a combination of the stacking and nonstacking style connectors.

Figure 68. Stacked w/Signal Stealing Expansion Connector

8.4.4 Retention Force

The length of the pins on the expansion header has a direct relationship to the amount of force that is used to remove a cape from the BeagleBone AI-64. The longer the pins extend into the connector the harder it is to remove. There is no rule that says that if longer pins are used, that the connector pins have to extend all the way into the mating connector on the BeagleBone AI-64, but this is controlled by the user and therefore is hard to control. We have also found that if you use gold pins, while more expensive, it makes for a smoother finish which reduces the friction.

This section will attempt to describe the tradeoffs and things to consider when selecting a connector and its pin length.

8.4.5 BeagleBone AI-64 Female Connectors

Figure 69 shows the key measurements used in calculating how much the pin extends past the contact point on the connector, what we call overhang.

Figure 69. Connector Pin Insertion Depth

To calculate the amount of the pin that extends past the Point of Contact, use the following formula:

$$\text{Overhang} = \text{Total Pin Length} - \text{PCB thickness (.062)} - \text{contact point (.079)}$$

The longer the pin extends past the contact point, the more force it will take to insert and remove the board. Removal is a greater issue than the insertion.

8.5 Signal Usage

Based on the pin muxing capabilities of the processor, each expansion pin can be configured for different functions. When in the stacking mode, it will be up to the user to ensure that any conflicts are resolved between multiple stacked cards. When stacked, the first card detected will be used to set the pin muxing of each pin. This will prevent other modes from being supported on stacked cards and may result in them being inoperative.

In [Section 7.1 Expansion Connectors](#) of this document, the functions of the pins are defined as well as the pin muxing options. Refer to this section for more information on what each pin is. To simplify things, if you use the default name as the function for each pin and use those functions, it will simplify board design and reduce conflicts with other boards.

Interoperability is up to the board suppliers and the user. This specification does not specify a fixed function on any pin and any pin can be used to the full extent of the functionality of that pin as enabled by the processor.

DO NOT APPLY VOLTAGE TO ANY I/O PIN WHEN POWER IS NOT SUPPLIED TO THE BOARD. IT WILL DAMAGE THE PROCESSOR AND VOID THE WARRANTY.

NO PINS ARE TO BE DRIVEN UNTIL AFTER THE SYS_RESET LINE GOES HIGH.

8.6 Cape Power

This section describes the power rails for the capes and their usage.

8.6.1 Main Board Power

The **Table 1** describes the voltages from the main board that are available on the expansion connectors and their ratings. All voltages are supplied by connector **P9**. The current ratings listed are per pin.

Table 21. Expansion Voltages

Current	Name	P9	P9	Name	Current
250mA	VDD_3V3B	3	4	VDD_3V3B	250mA
1000mA	VDD_5V	5	6	VDD_5V	1000mA
250mA	SYS_5V	7	8	SYS_5V	250mA

The **VDD_3V3B** rail is supplied by the LDO on the BeagleBone AI-64 and is the primary power rail for expansion boards. If the power requirement for the capes exceeds the current rating, then locally generated voltage rail can be used. It is recommended that this rail be used to power any buffers or level translators that may be used.

VDD_5V is the main power supply from the DC input jack. This voltage is not present when the board is powered via USB. The amount of current supplied by this rail is dependent upon the amount of current available. Based on the board design, this rail is limited to 1A per pin from the main board.

The **SYS_5V** rail is the main rail for the regulators on the main board. When powered from a DC supply or USB, this rail will be 5V. The available current from this rail depends on the current available from the USB and DC external supplies.

8.6.2 Expansion Board External Power

A cape can have a jack or terminals to bring in whatever voltages may be needed by that board. Care should be taken not to let this voltage be fed back into any of the expansion header pins.

It is possible to provide 5V to the main board from an expansion board. By supplying a 5V signal into the **VDD_5V** rail, the main board can be supplied. This voltage must not exceed 5V. You should not supply any voltage into any other pin of the expansion connectors. Based on the board design, this rail is limited to 1A per pin to the BeagleBone AI-64.

There are several precautions that need to be taken when working with the expansion headers to prevent damage to the board.

- 1. Do not apply any voltages to any I/O pins when the board is not powered on.**
- 2. Do not drive any external signals into the I/O pins until after the VDD_3V3B rail is up.**
- 3. Do not apply any voltages that are generated from external sources.**
- 4. If voltages are generated from the VDD_5V signal, those supplies must not become active until after the VDD_3V3B rail is up.**
- 5. If you are applying signals from other boards into the expansion headers, make sure you power the board up after you power up the BeagleBone AI-64 or make the connections after power is applied on both boards.**

Powering the processor via its I/O pins can cause damage to the processor.

8.7 Mechanical

This section provides the guidelines for the creation of expansion boards from a mechanical standpoint. Defined is a standard board size that is the same profile as the BeagleBone AI-64. It is expected that the majority of expansion boards created will be of standard size. It is possible to create boards of other sizes and in some cases this is required, as in the case of an LCD larger than the BeagleBone AI-64 board.

8.7.1 Standard Cape Size

Figure 70 is the outline of the standard cape. The dimensions are in inches.

Figure 70. Cape Board Dimensions

A slot is provided for the Ethernet connector to stick up higher than the cape when mounted. This also acts as a key function to ensure that the cape is oriented correctly. Space is also provided to allow access to the user LEDs and reset button on the main board.

Some people have inquired as to the difference in the radius of the corners of the BeagleBone AI-64 and why they are different. This is a result of having the BeagleBone fit into the Altoids style tin.

It is not required that the cape be exactly like the BeagleBone AI-64 board in this respect.

8.7.2 Extended Cape Size

Capes larger than the standard board size are also allowed. A good example would be an LCD panel. There is no practical limit to the sizes of these types of boards. The notch for the key is also not required, but it is up to the supplier of these boards to ensure that the BeagleBone AI-64 is not plugged in incorrectly in such a manner that damage would be caused to the BeagleBone AI-64 or any other capes that may be installed. Any such damage will be the responsibility of the supplier of such a cape to repair.

As with all capes, the EEPROM is required and compliance with the power requirements must be adhered to.

8.7.3 Enclosures

There are numerous enclosures being created in all different sizes and styles. The mechanical design of these enclosures is not being defined by this specification.

The ability of these designs to handle all shapes and sizes of capes, especially when you consider up to four can be mounted with all sorts of interface connectors, it is difficult to define a standard enclosure that will handle all capes already made and those yet to be defined.

If cape designers want to work together and align with one enclosure and work around it that is certainly acceptable. But we will not pick winners and we will not do anything that impedes the openness of the platform and the ability of enclosure designers and cape designers to innovate and create new concepts.

13. 9.0 BeagleBone AI-64 Mechanical

13.1. 9.1 Dimensions and Weight

Size: 3.5" x 2.15" (86.36mm x 53.34mm)

Max height: .187" (4.76mm)

PCB Layers: 6

PCB thickness: .062"

RoHS Compliant: Yes

Weight: 1.4 oz

9.2 Silkscreen and Component Locations

Figure 71. Board Dimensions

Figure 72. Component Side Silkscreen

Figure 73. Circuit Side Silkscreen

14. 10.0 Pictures

Figure 74. Top Side

Figure 75. Bottom Side

15. 11.0 Support Information

All support for this design is through the BeagleBoard.org community at:

beagleboard@googlegroups.com or

<http://beagleboard.org/discuss> .

15.1. 11.1 Hardware Design

Design documentation can be found on the eMMC of the board under the documents/hardware directory when connected using the USB cable. Provided there is:

- Schematic in PDF
- Schematic in OrCAD (Cadence Design Entry CIS 16.3)
- PCB Gerber
- PCB Layout (Allegro)
- Bill of Material
- System Reference Manual (This document).

This directory is not always kept up to date in every SW release due to the frequency of changes of the SW. The best solution is to download the files from beagleboard.org/latest-images¹⁷

We do not track SW revision of what is in the eMMC. SW is tracked separately from the HW due to the frequency of changes which would require massive relabeling of boards due to the frequent SW changes. You should always use the latest SW revision.

To see what SW revision is loaded into the eMMC follow the instructions at http://elinux.org/Beagleboard:Updating_The_Software#Checking_The_Angstrom_Image_Version

15.2. 11.2 Software Updates

It is a good idea to always use the latest software. Instructions for how to update your software to the latest version can be found at:

http://elinux.org/BeagleBoneBlack#Updating_the_eMMC_Software¹⁸

¹⁷ <https://beagleboard.org/latest-images>

¹⁸ http://elinux.org/BeagleBoneBlack#Updating_the_eMMC_Software

15.3. 11.3 RMA Support

If you feel your board is defective or has issues, request an RMA by filling out the form at <http://beagleboard.org/support/rma>. You will need the serial number and revision of the board. The serial numbers and revisions keep moving. Different boards can have different locations depending on when they were made. The following figures show the three locations of the serial and revision number.

Figure 76. Initial Serial Number and Revision Locations.

Figure 77. Second Phase Serial Number and Revision Location.

Figure 78. Third Phase Serial Number and Revision Location.

15.4. 11.4 Trouble Shooting HDMI Issues

Many people are having issues with getting HDMI to work on their TV/Display. Unfortunately, we do not have the resources to buy all the TVs and Monitors on the market today nor go to eBay and buy all of the TVs and monitors made over the last five years to thoroughly test each and every one. We are depending on community members to help us get these tested and information provided on how to get them to work.

One would think that if it worked on a lot of different TVs and monitors it would work on most if not all of them, assuming they meet the specification. However, there are other issues that could also result in these various TVs and monitors not working. The intent is that this page will be useful in navigating some of these issues. As others also find solutions, as long as we know about them, they will be added here as well. For access to the most up to date troubleshooting capabilities, go to the support wiki at http://www.elinux.org/Beagleboard:BeagleBoneBlack_HDMI

The early release of the Software had some issues in the HDMI driver. Be sure and use the latest SW to take advantage of the improvements.

http://www.elinux.org/Beagleboard:BeagleBoneBlack#Software_Resources

11.4.1 EDID

EDID is the way the board requests information from the display and determines all the resolutions that it can support. The driver on the board will then look at these timings and find the highest resolution that is compatible with the board and uses that resolution for the display. For more information on EDID, you can take a look at http://en.wikipedia.org/wiki/Extended_display_identification_data

If the board is not able to read the EDID, for whatever reason, it does not have this information. A few possible reasons for this are:

- Bad cable
- Cable not plugged in all the way on both ends
- Display not powered on. (It should still work powered off, but some displays do not).

11.4.2 DISPLAY SOURCE SELECTION

One easy thing to overlook is that you need to select the display source that matches the port you are using on the TV. Some displays may auto select, so you may need to disconnect the other inputs until you are sure the display works with the board.

11.4.3 OUT OF SEQUENCE

Sometimes the display and the board can get confused. One way to prevent this is after everything is cabled up and running, you can power cycle the display, with the board still running. You can also try resetting the board and let it reboot to resync with the TV.

11.4.4 OVERSCAN

Some displays use what is called overscan. This can be seen in TVs and not so much on Monitors. It causes the image to be missing on the edges, such that you cannot see them displayed. Some higher end displays allow you to disable overscan.

Most TVs have a mode that allows you to adjust the image. These are options like Normal, Wide, Zoom, or Fit. Normal seems to be the best option as it does not chop off the edges. The other ones will crop off the edges.

11.4.5 Taking a Nap

In some cases the board can come up in a power down/screen save mode. No display will be present. This is due to the board believing that it is asleep. To come out of this, you will need to hit the keyboard or move the mouse.

Once working, the board will time out and go back to sleep again. This can cause the display to go into a power down mode as well. You may need to turn the display back on again. Sometimes, it may take a minute or so for the display to catch up and show the image.

11.4.6 AUDIO

Audio will only work on TV resolutions. This is due to the way the specification was written. Some displays have built in speakers and others require external. Make sure you have a TV resolution and speakers are connected if they are not built in. The SW should default to a TV resolution giving audio support. The HDMI driver should default to the highest audio supported resolution.

11.4.7 Getting Help

If you need some up to date troubleshooting techniques, we have a Wiki set up at http://elinux.org/Beagleboard:BeagleBoneBlack_HDMI

16. 12.0 Footnotes