

# BeagleBone AI-64 System

## Reference Manual



---

# Table of Contents

Preface .....	iii
Glossary .....	9
1. Introduction .....	11
2. Change History .....	13
2.1. Document Change History .....	13
2.2. Board Changes .....	13
2.2.1. Rev B0 .....	13
3. Connecting Up Your BeagleBone AI-64 .....	15
3.1. 3.0.1 Methods of operation .....	15
3.2. What's In the Box .....	15
3.3. Main Connection Scenarios .....	16
3.4. Tethered To A PC .....	16
3.4.1. Connect the Cable to the Board .....	17
3.4.2. Accessing the Board as a Storage Drive .....	19
3.5. Standalone w/Display and Keyboard/Mouse .....	19
3.5.1. Required Accessories .....	20
3.5.2. Connecting Up the Board .....	20
4. BeagleBone AI-64 Overview .....	27
4.1. BeagleBone Compatibility .....	28
4.2. BeagleBone AI-64 Features and Specification .....	29
4.3. Board Component Locations .....	31
4.3.1. Board components .....	31
5. BeagleBone AI-64 High Level Specification .....	33
5.1. Processor .....	34
5.1.1. Some of the main distinguished characteristics of the device are: .....	34
5.1.2. The device is composed of the following main subsystems, across different domains of the SoC, among others: .....	35
5.1.3. The device provides a rich set of peripherals such as: .....	36
5.1.4. The device also integrates: .....	37
5.1.5. The device includes different modules for functional safety requirements support: .....	38
5.1.6. The device supports the following main security functionalities among others: .....	39

5.2. Memory .....	40
5.2.1. 4GB LPDDR4 .....	40
5.2.2. 4KB EEPROM .....	40
5.2.3. 16GB Embedded MMC .....	40
5.2.4. MicroSD Connector .....	41
5.2.5. Boot Modes .....	41
5.3. Power Management .....	42
5.4. PC USB Interface .....	42
5.5. Serial Debug Port .....	42
5.6. USB1 Host Port .....	43
5.7. Power Sources .....	43
5.8. Reset Button .....	43
5.9. Power Button .....	44
5.10. Indicators .....	44
5.11. DP Interface .....	44
5.12. Cape Board Support .....	45
6. Detailed Hardware Design .....	47
6.1. Power Section .....	48
6.1.1. TPS65941213 and TPS65941111 PMIC .....	49
6.1.2. DC Input .....	52
6.1.3. USB Power .....	53
6.1.4. Power Selection .....	53
6.1.5. Power Button .....	53
6.1.6. Battery Access Pads .....	54
6.1.7. Power Consumption .....	54
6.1.8. Processor Interfaces .....	55
6.1.9. 6.1.9 Power Rails .....	56
6.1.10. Power LED .....	58
6.1.11. TPS65941213 and TPS65941111 Power Up Process .....	59
6.1.12. Processor Control Interface .....	59
6.1.13. Low Power Mode Support .....	60
6.2. TI J721E DRA829/TDA4VM/AM752x Processor .....	61
6.2.1. Description .....	61
6.2.2. High Level Features .....	61
6.2.3. Documentation .....	62
6.2.4. Crystal Circuitry .....	62

6.2.5. Reset Circuitry .....	62
6.2.6. Memory Device .....	63
6.2.7. LPDDR4 Memory Design .....	63
6.2.8. Power Rails .....	65
6.2.9. VREF .....	65
6.3. 4GB eMMC Memory .....	65
6.3.1. eMMC Device .....	66
6.3.2. eMMC Circuit Design .....	66
6.4. Board ID EEPROM .....	67
6.5. Micro Secure Digital .....	69
6.5.1. microSD Design .....	69
6.6. User LEDs .....	69
6.7. Boot Configuration .....	70
6.7.1. Boot Configuration Design .....	70
6.8. Default Boot Options .....	71
6.9. 10/100 Ethernet .....	71
6.9.1. Ethernet Processor Interface .....	71
6.9.2. Ethernet Connector Interface .....	72
6.9.3. Ethernet PHY Power, Reset, and Clocks .....	72
6.9.4. LAN8710A Mode Pins .....	73
6.10. Display Port Interface .....	73
6.10.1. Supported Resolutions .....	74
6.10.2. Display Port Framer .....	75
6.10.3. Display Port Video Processor Interface .....	75
6.10.4. Display Port Control Processor Interface .....	75
6.10.5. Interrupt Signal .....	75
6.10.6. Audio Interface .....	75
6.10.7. Power Connections .....	75
6.10.8. uDP Connector Interface .....	75
6.11. USB Host .....	75
6.11.1. Power Switch .....	75
6.11.2. ESD Protection .....	76
6.11.3. Filter Options .....	76
6.12. PRU-ICSS .....	76
6.12.1. PRU-ICSS Features .....	76
6.12.2. PRU-ICSS Block Diagram .....	77

6.12.3. PRU-ICSS Pin Access .....	77
7. Connectors .....	81
7.1. Expansion Connectors .....	81
7.1.1. Connector P8 and P9 .....	81
7.2. Power Jack .....	86
7.3. USB Client .....	86
7.4. USB Host .....	86
7.5. Serial Header .....	87
7.6. HDMI .....	88
7.7. microSD .....	88
7.8. Ethernet .....	89
7.9. JTAG Connector .....	89
8. Cape Board Support .....	91
8.1. BeagleBone AI-64 Cape Compatibility .....	92
8.2. EEPROM .....	92
8.2.1. EEPROM Address .....	93
8.2.2. I2C Bus .....	94
8.2.3. EEPROM Write Protect .....	94
8.2.4. EEPROM Data Format .....	95
8.2.5. Pin Usage .....	100
8.3. Pin Usage Consideration .....	101
8.4. Expansion Connectors .....	101
8.4.1. Non-Stacking Headers-Single Cape .....	101
8.4.2. Main Expansion Headers-Stacking .....	103
8.4.3. Stacked Capes w/Signal Stealing .....	105
8.4.4. Retention Force .....	106
8.4.5. BeagleBone AI-64 Female Connectors .....	106
8.5. Signal Usage .....	107
8.6. Cape Power .....	107
8.6.1. Main Board Power .....	107
8.6.2. Expansion Board External Power .....	108
8.6.3. Standard Cape Size .....	109
8.6.4. Extended Cape Size .....	110
9. BeagleBone AI-64 Mechanical .....	111
9.1. Dimensions and Weight .....	111
9.2. Silkscreen and Component Locations .....	111

10. Pictures .....	113
11. Support Information .....	117
11.1. Hardware Design .....	117
11.2. Software Updates .....	117
11.3. RMA Support .....	118
11.4. Troubleshooting video output issues .....	118
11.4.1. EDID .....	118
11.4.2. DISPLAY SOURCE SELECTION .....	118
11.4.3. OUT OF SEQUENCE .....	119
11.4.4. OVERSCAN .....	119
11.4.5. Taking a Nap .....	119
11.4.6. AUDIO .....	119
11.4.7. Getting Help .....	120
Bibliography .....	121
Index .....	123



---

# Revision 0.2b

**Last PDF conversion: June 3<sup>rd</sup>, 2022**

**Maintaining editor: Deepak Khatri<sup>1</sup>**

**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://git.beagleboard.org/beagleboard/beaglebone-ai-64/-/tree/master/doc/](https://git.beagleboard.org/beagleboard/beaglebone-ai-64/-/tree/master/doc/SRM)  
[SRM](#)

---

<sup>1</sup> <mailto:lorforlinux@beagleboard.org>



---

# Preface

## 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](http://BeagleBoard.org/discuss).<sup>1</sup>

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.

## 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 BeagleBone not meet the specifications indicated in the System Reference Manual, BeagleBone may be returned within 90 days from the date of

---

<sup>1</sup> <http://beagleboard.org/discuss>

---

delivery to the distributor of purchase for a full refund. THE FOREGOING LIMITED WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY SELLER TO BUYER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. 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.

## **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.

## **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 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 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 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 BeagleBone will not result in any property damage, injury, or death, even if BeagleBone should fail to perform as described or expected.

**Certain Instructions.** It is important to operate 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 BeagleBone and/or interface electronics. Please consult the System Reference Manual before connecting any load to 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 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 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 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/discuss](https://beagleboard.org/discuss)<sup>2</sup>

To return a defective board for repair, please request an RMA at [beagleboard.org/support/rma](https://beagleboard.org/support/rma)<sup>3</sup>

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

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

---

<sup>2</sup> <https://beagleboard.org/discuss>

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



---

# Glossary

BeagleBoard.org	A Michigan, USA-based 501(c)(3) non-profit corporation.
BeagleBone	Original BeagleBone
Board	BeagleBone AI 64



---

# 1

## Introduction

---

This document is the **System Reference Manual** for 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 AI-64 / BBAI-64 / 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 frequently check the [doc folder in BBAI-64 git repository<sup>1</sup>](#) for the most up to date support documents.

---

<sup>1</sup> <https://git.beagleboard.org/beagleboard/beaglebone-ai-64/-/tree/master/doc>



---

# 2

## 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.

### 2.1. Document Change History

Table 2.1. Table 1. Change History

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

### 2.2. Board Changes

Full issue list at <https://git.beagleboard.org/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.

## ***Known issues***

- USB SuperSpeed cannot function

---

# 3

## Connecting Up Your BeagleBone AI-64

---

This section provides instructions on how to hook up your board. This beagle requires a  $5V \geq 3A$  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>

### 3.1. 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

### 3.2. What's In the Box

In the box you will find three main items as shown in [BBAI-64 box image](#).

- 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 BBBI-64 in the box (media/BBBI-in-Box.png)

### 3.3. 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://git.beagleboard.org/beagleboard/beaglebone-ai-64/wiki/Accessories><sup>1</sup>

### 3.4. 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

---

<sup>1</sup> <https://git.beagleboard.org/beagleboard/beaglebone-ai-64/wiki/Accessories>

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. [Tethered Configuration](#) figure shows this configuration.

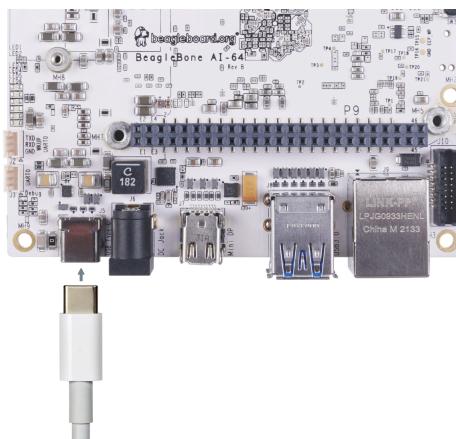


**Figure 3.1. Tethered Configuration**

At least 5V @ 3A is required to power the board. In most cases the PC may not be able to supply sufficient power for the board. You should always use an external 5VDC power supply connected to the barrel jack.

### **3.4.1. Connect the Cable to the Board**

1. Connect the type C USB cable to the board as shown in [USB Connection to the Board](#) figure. The connector is on the top side of the board near barrel jack.



**Figure 3.2. USB Connection to the Board**

2. Connect the other end 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 [Board Power LED figure](#) below.

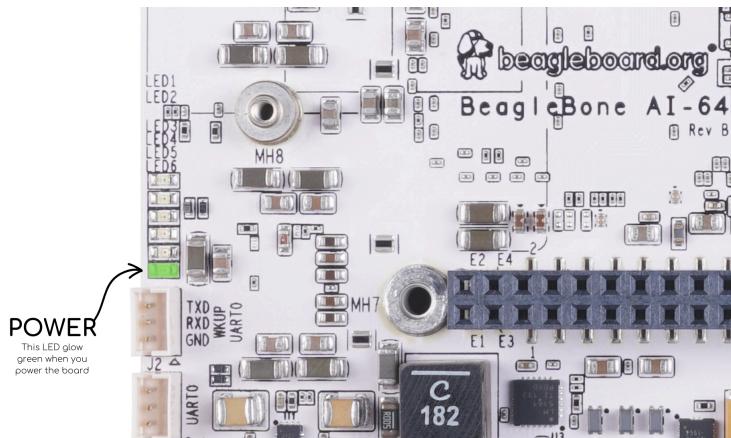


Figure 3.3. Board Power LED

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 [Board Boot Status figure](#) 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.

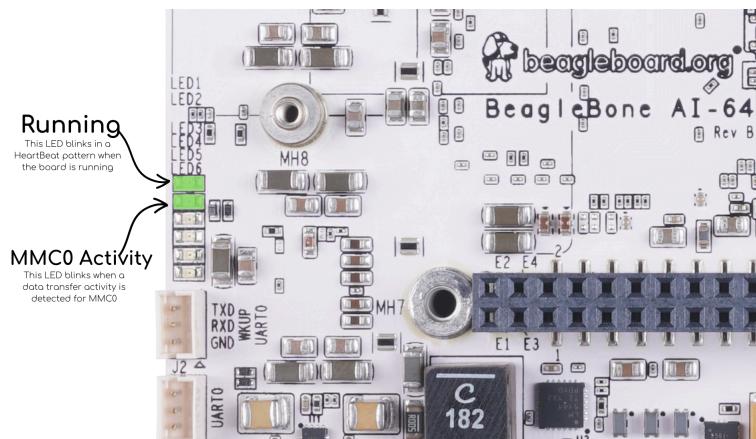


Figure 3.4. media/Board Boot Status

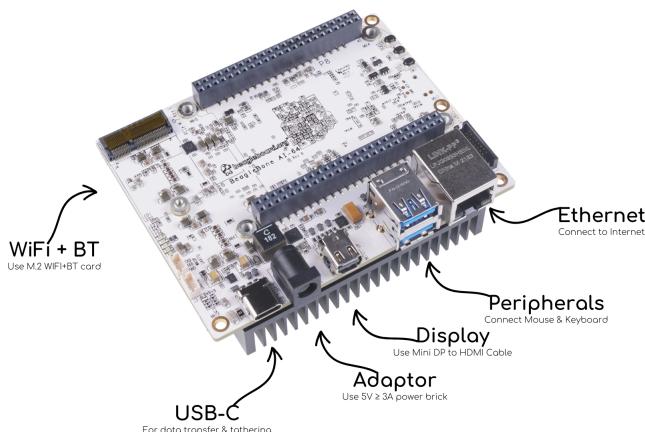
### 3.4.2. Accessing the Board as a Storage Drive

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.

### 3.5. 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 [Desktop Configuration figure](#). 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 3.5. Desktop Configuration**

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

### ***3.5.1. Required Accessories***

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

- (1) 5VDC  $\geq$  3A power supply.
- (1) Display Port or HDMI monitor (or a recommended miniDP-HDMI or miniDP-DP adapter. <https://www.amazon.com/dp/B089GF8M87> has been tested and worked beautifully.)
- (1) miniDP-DP or miniDP-HDMI cable.
- (1) USB wired/wireless 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 Bluetooth & 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 **TO-UPDATE** <https://git.beagleboard.org/beagleboard/beaglebone-ai-64/wiki/Accessories>.

### ***3.5.2. Connecting Up the Board***

1. Connect the miniDP to HDMI or miniDP to DP cable from your BBAI-64 to your monitor.

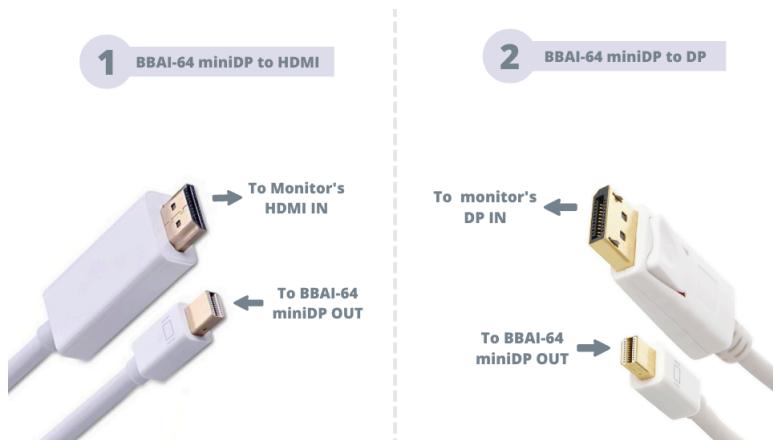


Figure 3.6. Connect miniDP/HDMI cable to BBAI-64

2. If you have an Display Port or HDMI monitor with HDMI-HDMI or DP-DP cable you can use adapters as shown in. [Display adaptors figure](#).

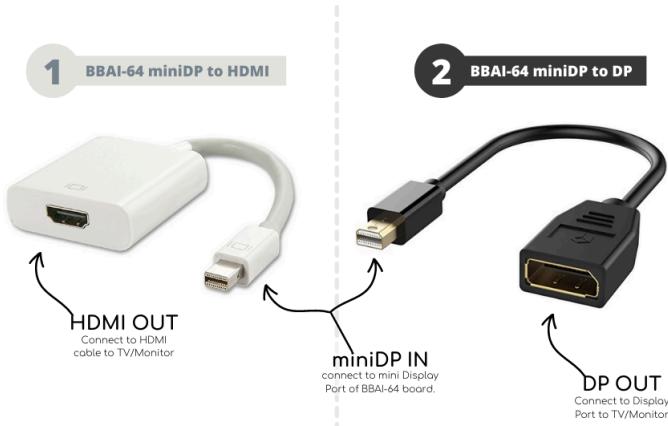
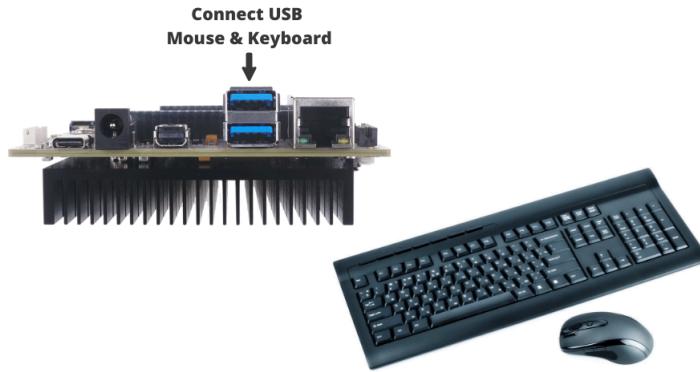


Figure 3.7. Display adaptors

3. If you have wired/wireless USB keyboard and mouse such as

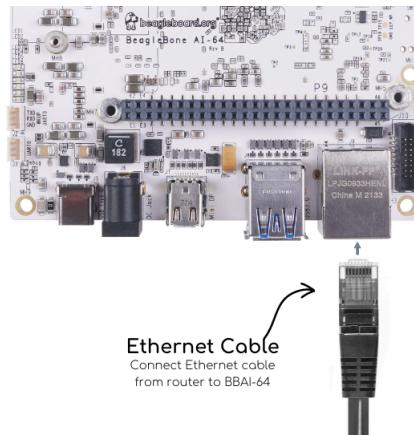
seen in [Keyboard and Mouse figure](#) below, you need to plug the receiver in the USB host port of the board as shown in [Keyboard and Mouse figure](#).



**Figure 3.8. 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 <<ethernet-cable-figure>>. Any standard 100M Ethernet cable should work.



**Figure 3.9. Ethernet Cable Connection**

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

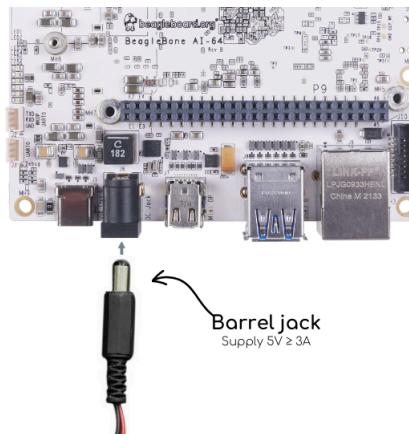


Figure 3.10. External DC Power

6. The cable needed to connect to your display is a miniDP to HDMI/DP. Connect the miniDP connector end to the board at this time. The connector is on the top side of the board as shown in [miniDP to HDMI/DP connection figure](#) below.

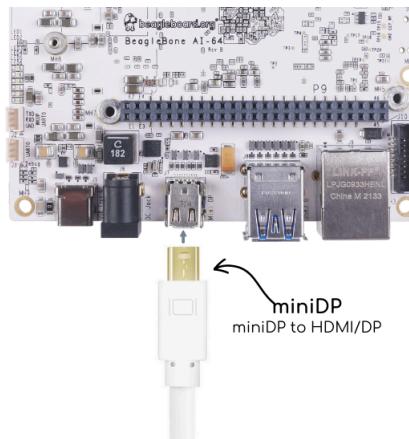


Figure 3.11. Connect miniDP Cable to the Board

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. 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.

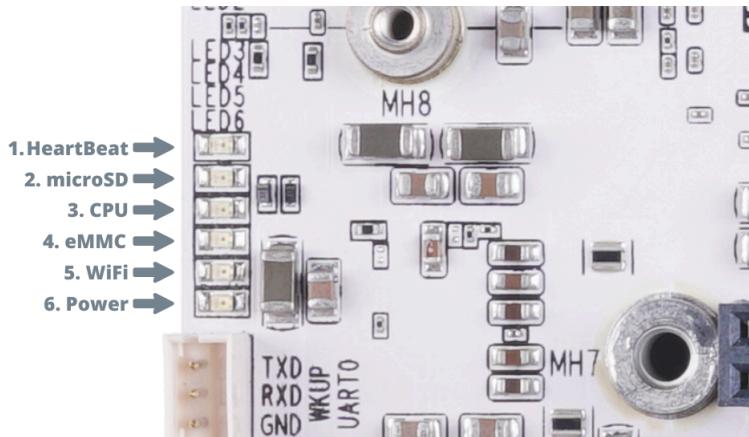


Figure 3.12. BBAI-64 LEDs

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.
- **USER4** is an activity indicator for WiFi.

## 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.

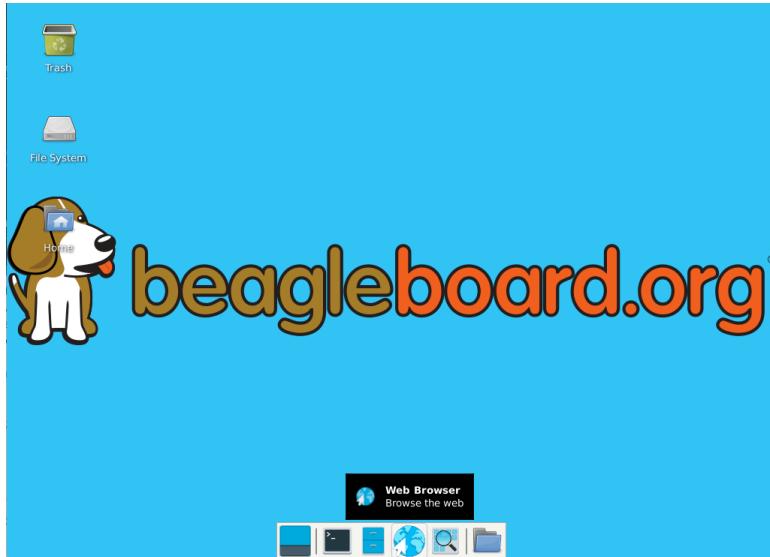


Figure 3.13. BeagleBone XFCE Desktop Screen

#### 9. Powering Down

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



---

# 4

## BeagleBone AI-64 Overview

---

BeagleBone AI-64 is the latest addition to 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 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 including the current BeagleBoard.org Foundation board members

Jason Kridner, principal of JK Embedded Consulting an independent contractor and architect for new Beagle designs. Drew Fustini, independent Linux developer Robert Nelson, applications engineer at Digi-Key Mark Yoder, professor at Rose-Hulman Institute of Technology Kathy Giori, product engineer at ZEDEDA See [bbb.io/about](http://bbb.io/about)

The BBAI-64 board is designed designed by Seeed Studio (Seeed Development Limited) under guidance from BeagleBoard.org Foundation.

## 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.

## 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 4.1. Table 2. BeagleBone AI-64 Features**

	<b>Feature</b>
<b>Processor</b>	TI J721E DRA829/TDA4VM/AM752x
<b>Graphics Engine</b>	PowerVR® Series8XE GE8430
<b>SDRAM Memory</b>	LPDDR4 (4GB) Q3222PM1WDGTK-U
<b>Onboard Flash</b>	eMMC (16GB) EMMC16G-TB29-PZ90
<b>PMIC</b>	TPS659411 and TPS659413 PMICs regulator and one additional LDO.
<b>Debug Support</b>	3 pin Serial Header
<b>Power Source</b>	USB C or DC Jack
<b>PCB</b>	This beagle is not the standard 3.4" x 2.1", It's grown in size to 4" x 3.1"
<b>Indicators</b>	1-Power, 2-Ethernet, 4-User Controllable LEDs
<b>HS USB 2.0 Client Port</b>	Access to USB0, Client mode via USB-C
<b>HS USB 2.0 Host Port</b>	Access to USB1 & USB2, Type A Socket, 1500mA LS/FS/HS
<b>Serial Port</b>	UART0 access via 3 pin 3.3V TTL micro Header.
<b>Ethernet</b>	Gigabit, RJ45
<b>SD/MMC Connector</b>	microSD , 3.3V
<b>User Input</b>	Reset Button Boot Button Power Button
<b>Video Out</b>	miniDP
<b>Audio</b>	via miniDP Interface, Stereo
<b>Expansion Connectors</b>	<b>Update required</b> Power 5V, 3.3V , VDD_ADC(1.8V) 3.3V I/O on all signals McASP0, SPI1, I2C, GPIO(69 max), LCD, GPMC, MMC1, MMC2, 7

	Feature
	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	192gm (with heatsink)
Power	Refer to <a href="#">Section 8.6.1, “Main Board Power” section</a>

## 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. Board components

[BeagleBone AI-64 board components figure](#) below shows the locations of the connectors, LEDs, and switches on the PCB layout of the board.

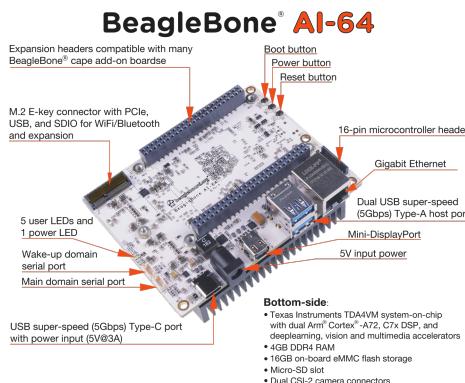


Figure 4.1. BeagleBone AI-64 board components

- **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.
- **GigaBit 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.

On bottom side we have,

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

# BeagleBone AI-64 High Level Specification

BeagleBone AI-64 Key Components figure below shows the high level block diagram of BeagleBone AI-64 board surrounding TDA4VM SoC.

BeagleBoneAI -64

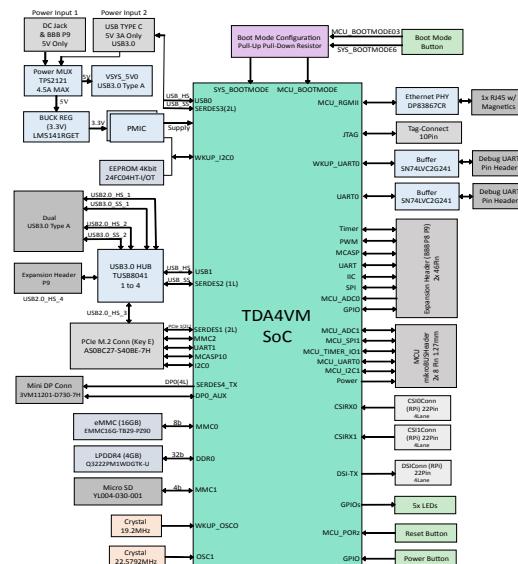


Figure 5.1. BeagleBone AI-64 Key Components

## 5.1. Processor

BeagleBone AI-64 uses TI J721E [DRA829<sup>1</sup>](#) / [TDA4VM<sup>2</sup>](#) monster of an SoC, not necessarily in terms of CPU processing power, but it has an amazing amount of features and peripherals. The DRA829 and TDA4VM SoCs are part of the K3 Multicore SoC architecture platform. The SoCs are targeted for automotive applications and aim to meet the complex processing needs of modern embedded products.

They are designed as a low power, high performance and highly integrated device architecture, adding significant enhancement on processing power, graphics capability, video and imaging processing, virtualization and coherent memory support. In addition, these SoCs support state of the art security and functional safety features. For the remaining of this section device, SoC, and processor will be used interchangeably.

### ***5.1.1. Some of the main distinguished characteristics of the device are:***

- 64-bit architecture with virtualization and coherent memory support, which leverages full processing capability of 64-bit Arm® Cortex®-A72
- Fully programmable industrial communication subsystems to enable future-proof designs for customers that need to adopt the new Gigabit Time-sensitive Networks (TSN) standards, but still need full support on legacy protocols and continuous system optimization over the product deployment
- Integration of vision hardware processing accelerators to facilitate extensive processing requirements in low power budget for automotive ADAS and machine vision applications
- Integration of a general-purpose microcontroller unit (MCU) with a dual Arm® Cortex®-R5F MCU subsystem, available for general purpose use as two cores or in lockstep, intended to help customers achieve functional safety goals for their end products
- Integration of a next-generation fixed and floating-point C71x Digital Signal Processor (DSP) that significantly boosts power over a broad range of

---

<sup>1</sup> <https://www.ti.com/product/DRA829V>

<sup>2</sup> <https://www.ti.com/product/TDA4VM>

The device is composed of the following main subsystems, across different domains of the SoC, among others:

- general signal processing tasks for both general applications and automotive functions which also incorporates advanced techniques to improve control code efficiency and ease of programming such as branch prediction, protected pipeline, precise exception and virtual memory management
- Tightly coupled Matrix Multiplication Accelerator (MMA) that extends the C71x DSP architecture's scalar and vector facilities enabling deep learning and enhance vision, analytics and wide range of general applications. The achieved total TOPS (Tera Operations Per Second) performance significantly differentiates the device for single board computer in machine vision and deep learning applications
- Key display features including flexibility to interface with different panel types (eDP, DSI, DPI) with multi-layer hardware composition
- Integration of hardware features that help applications to achieve functional safety mechanisms
- Robust security architecture with sandboxed DMSC controller managing all secure configurations with high performance client-server messaging scheme between secure DMSC and all cores
- Simplified solution for power supply management, enabling lower cost system solution (on-die bias LDOs and power good comparators for minimal power sequencing requirements consistent with low cost supply design)

### ***5.1.2. The device is composed of the following main subsystems, across different domains of the SoC, among others:***

- One dual-core 64-bit Arm Cortex-A72 microprocessor subsystem at up to 2.0 GHz and up to 24K DMIPS (Dhrystone Million Instructions per Second)
- Up to three Microcontroller Units (MCU), based on dual-core Arm Cortex-R5F processor running at up to 1.0 GHz, up to 12K DMIPS
- Up to two TMS320C66x DSP CorePac modules running at up to 1.35 GHz, up to 40 GFLOPS
- One C71x floating point, vector DSP running at up to 1.0 GHz, up to 80 GFLOPS
- One deep-learning MMA, up to 8 TOPS (8b) at 1.0 GHz

The device provides a rich set of peripherals such as:

---

- Up to two gigabit dual-core Programmable Real-Time Unit and Industrial Communication Subsystems (PRU\_ICSSG)
- Two Navigator Subsystems (NAVSS) for data movement and control
- One multi-pipeline Display Subsystem (DSS) with one MIPI® Display Serial Interface Controller (DSI) and shared MIPI D-PHY Transmitter (DPHY\_TX), one Embedded DisplayPort Transmitter (EDP) with shared Serializer/Deserializer (SERDES), and two MIPI Display Pixel Interface (DPI) ports
- Two Camera Streaming Interface Receivers (CSI\_RX\_IF) with dedicated MIPI D-PHYS (DPHY\_RX)
- One Camera Streaming Interface Transmitter (CSI\_TX\_IF) with MIPI D-PHY Transmitter (DPHY\_TX) shared with DSI
- One Vision Processing Accelerator (VPAC) with image signal processor
- One Depth and Motion Processing Accelerator (DMPAC)
- One dual-core multi-standard HD Video Decoder (DECODER)
- One dual-core multi-standard HD Video Encoder (ENCODER)
- One Graphics Processing Unit (GPU)
- One Device Management and Security Controller (DMSC)

### **5.1.3. *The device provides a rich set of peripherals such as:***

- General connectivity peripherals, including:
  - Two 12-bit general purpose Analog-to-Digital Converters (ADC)
  - Ten Inter-Integrated Circuit (I2C) interfaces
  - Three Improved Inter-Integrated Circuit (I3C) controllers
  - Eleven master/slave Multichannel Serial Peripheral Interfaces (MCSPI)
  - Twelve configurable Universal Asynchronous Receiver/Transmitter (UART) interfaces
  - Ten General-Purpose Input/Output (GPIO) modules
- High-speed interfaces, including:
  - Two Gigabit Ethernet Switch (CPSW) modules

---

The device also integrates:

- Two Dual-Role-Device (DRD) Universal Serial Bus Subsystems (USBSS) with integrated PHY
- Four Peripheral Component Interconnect express (PCIe) Gen3 subsystems
- Flash memory interfaces, including:
  - One Octal SPI (OSPI) interface and one Quad SPI (QSPI) or one QSPI and one HyperBus<sup>TM</sup>
  - One General Purpose Memory Controller (GPMC) with Error Location Module (ELM) and 8- or 16- bit-wide data bus width (supports parallel NOR or NAND FLASH devices)
  - Three Multimedia Card/Secure Digital (MMCSD) controllers
  - One Universal Flash Storage (UFS) interface
- Industrial and control interfaces, including:
  - Sixteen Controller Area Network (MCAN) interfaces with flexible data rate support
  - Three Enhanced Capture (ECAP) modules
  - Six Enhanced Pulse-Width Modulation (EPWM) subsystems
  - Three Enhanced Quadrature Encoder Pulse (EQEP) modules
- Audio peripherals, including:
  - One Audio Tracking Logic (ATL)
  - Twelve Multichannel Audio Serial Port (MCASP) modules supporting up to 16 channels with independent TX/RX clock/sync domain
- One Video Processing Front End (VPFE) interface module

#### **5.1.4. The device also integrates:**

- Power distribution, reset controls and clock management components
- Power-management techniques for device power consumption minimization:
  - Adaptive Voltage Scaling (AVS)
  - Dynamic Frequency Scaling (DFS)

---

The device includes different modules for functional safety requirements support:

---

- Gated clocks
- Multiple voltage domains
- Independently controlled power domains for major modules
- Voltage and Temperature Management (VTM) module
- Power-on Reset Generators (PRG)
- Power Sleep Controllers (PSC)
- Optimized interconnect (CBASS) architecture to enable latency-critical real time network and IO applications
- Control modules (CTRL\_MMRs) mainly associated with device top-level configurations such as:
  - IO Pad and pin multiplexing configuration
  - PLL control and associated High-Speed Dividers (HSDIV)
  - Clock selection
  - Analog function controls
- Multicore Shared Memory Controller (MSMC)
- DDR Subsystem (DDRSS) with Error Correcting Code (ECC), supporting LPDDR4
- 1KB RAM with ECC support for C71x boot vectors
- 2KB RAM with ECC support for A72 and R5F boot vectors
- 512KB On-Chip SRAM protected by ECC
- One Global Time Counter (GTC) module
- Thirty 32-bit counter timers with compare and capture modes
- Debug and trace capabilities

### ***5.1.5. The device includes different modules for functional safety requirements support:***

- MCU island with dual lock step Arm Cortex-R5F
- Safety enabled interconnect with implemented features to help with Freedom From Interference (FFI)

---

The device supports the following main security functionalities among others:

---

- Twelve Real Time Interrupt (RTI) modules with Windowed Watchdog Timer (WWDT) functionality to monitor processor cores
- Sixteen Dual-Clock Comparators (DCC) to monitor clocking sources during run-time
- Three Error Signaling Modules (ESM) to enable error monitoring
- Temperature monitoring sensors
- ECC on all critical memories
- Dedicated hardware Memory Cyclic Redundancy Check (MCRC) blocks

#### ***5.1.6. The device supports the following main security functionalities among others:***

- Secure Boot Management
- Public Key Accelerator (PKA) for large vector math operation
- Cryptographic acceleration (AES, 3DES, MD5, SHA1, SHA2-224, 256, 512 operation)
- Trusted Execution Environment (TEE)
- Secure storage support
- On-the-fly encryption and authentication support for OSPI interface

The device is partitioned into three functional domains as shown in [Device Top-level Block Diagram](#), each containing specific processing cores and peripherals:

- Wake-up (WKUP) domain
- Microcontroller (MCU) domain with one of the dual Cortex-R5 cluster
- MAIN domain

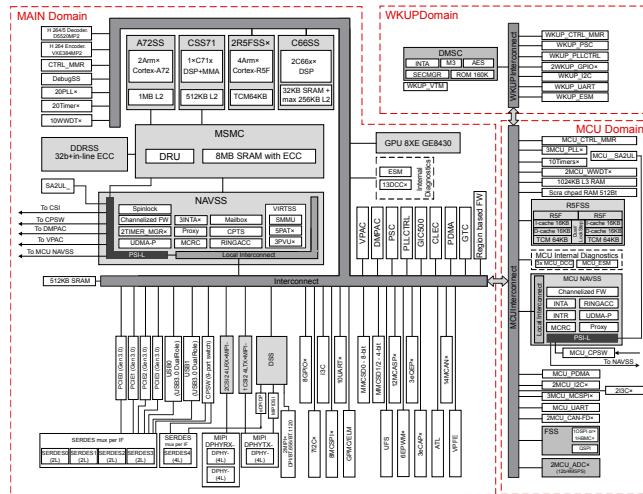


Figure 5.2. Device Top-level Block Diagram

## 5.2. Memory

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

### 5.2.1. 4GB LPDDR4

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

- Q3222PM1WDGTTK

### 5.2.2. 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 the 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.

### 5.2.3. 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.

### **5.2.4. 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.

### **5.2.5. 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.*

### 5.3. 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.

### 5.4. 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.

### 5.5. 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.

## 5.6. 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.

## 5.7. 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 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.

## 5.8. Reset Button

When pressed and released, causes a reset of the board. The reset button used on 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.

## 5.9. 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.

## 5.10. 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).

## 5.11. DP Interface

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

## 5.12. 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 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 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 BeagleBone AI-64 so those functions are not supported.

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



---

# 6

## 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.

[BeagleBone AI-64 Key Components figure](#) below is the high level block diagram of the board. For those who may be concerned, It is the same figure as shown in [Chapter 5, BeagleBone AI-64 High Level Specification](#). It is placed here again for convenience so it is closer to the topics to follow.

## BeagleBoneAI -64

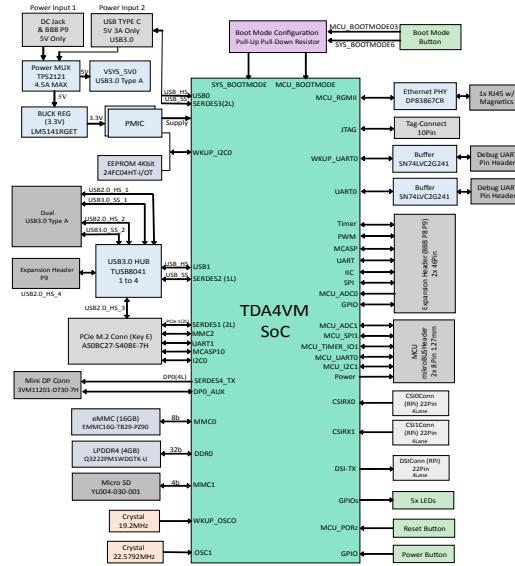


Figure 6.1. BeagleBone AI-64 Key Components

## 6.1. Power Section

[High level power block diagram](#) shows the high level block diagram of the power section of the board.

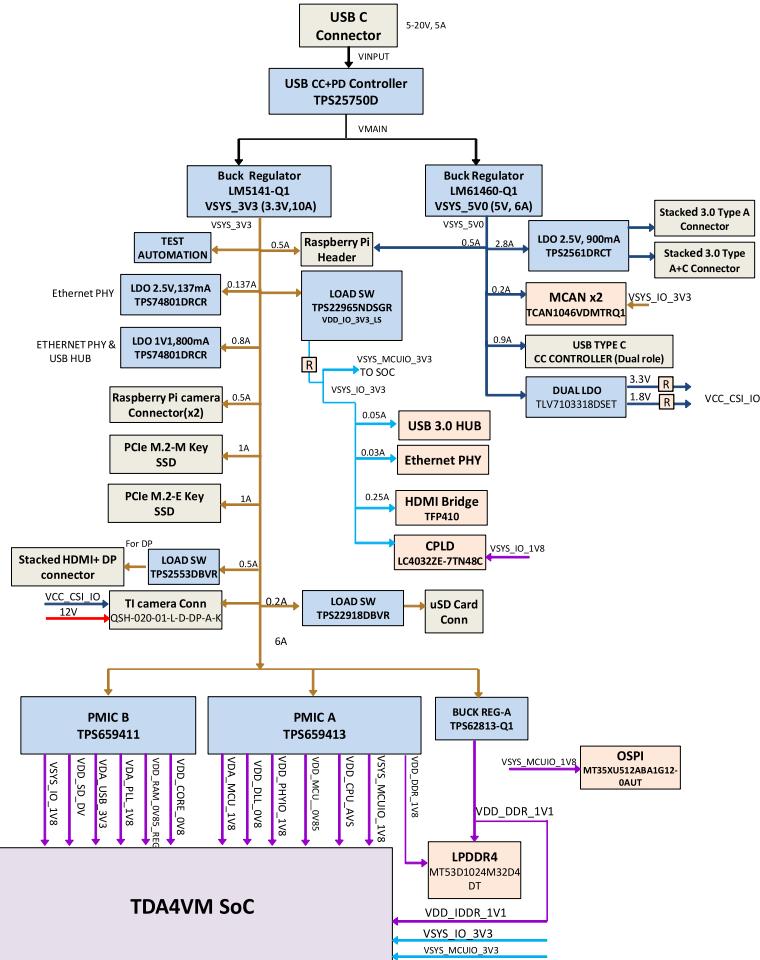


Figure 6.2. High level power block diagram

This section describes the power section of the design and all the functions performed by the **TPS65941213** and **TPS65941111**.

### 6.1.1. TPS65941213 and TPS65941111 PMIC

The main Power Management IC (PMIC) in the system is the **TPS65941213** and **TPS65941111** 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](http://en.wikipedia.org/wiki/Low-dropout_regulator).<sup>1</sup> If you want to learn more about step-down converters, you can go to [http://en.wikipedia.org/wiki/DC-to-DC\\_converter](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.

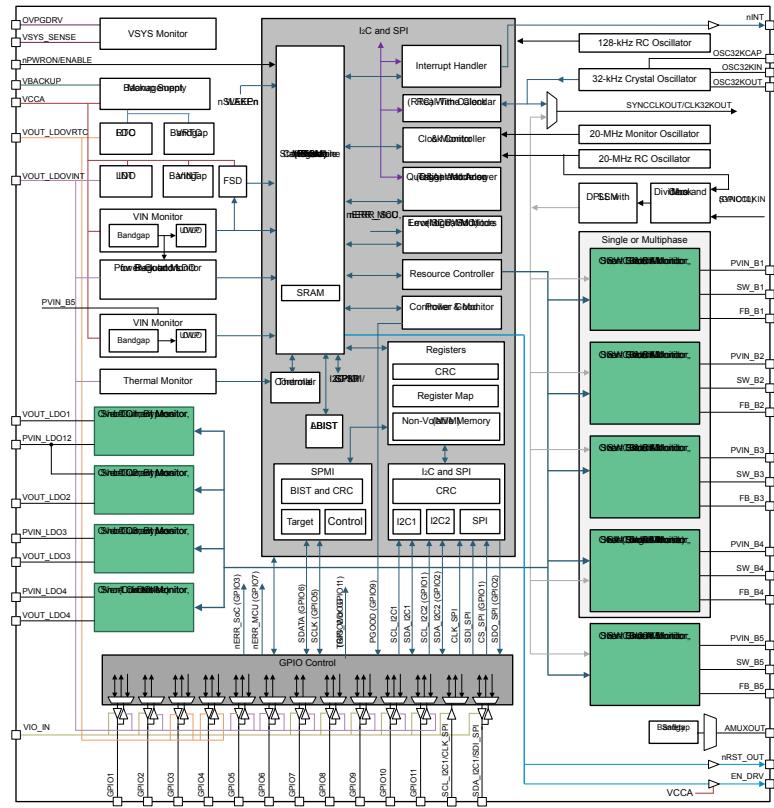
LDO1 and LDO2 are intended to support system standby mode. In normal operation, they can support up to 100mA each. LDO3 and LDO4 can support up to 285mA each.

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 **TPS65941213** and **TPS65941111** 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.

See the [TPS6594-Q1 block diagram](#) shown below for high level details for **TPS65941213** and **TPS65941111**, for more information on the, refer to <https://www.ti.com/product/TPS6594-Q1> Texas instruments product page.

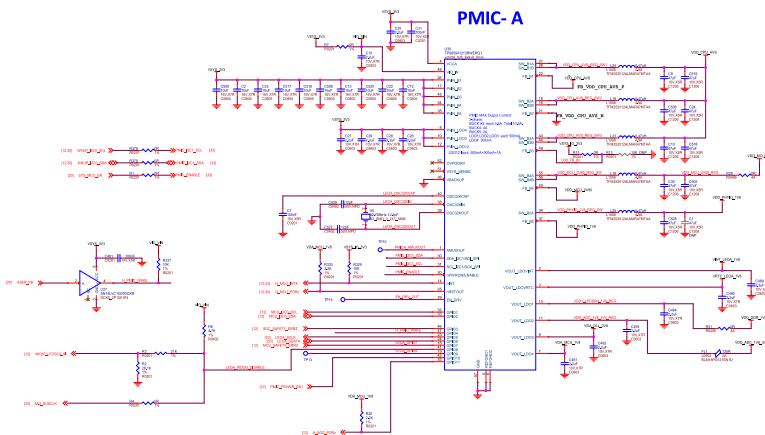
---

<sup>1</sup> [http://en.wikipedia.org/wiki/Low-dropout\\_regulator](http://en.wikipedia.org/wiki/Low-dropout_regulator)



Copyright © 2019, Texas Instruments Incorporated

Figure 6.3. TPS6594-Q1 block diagram



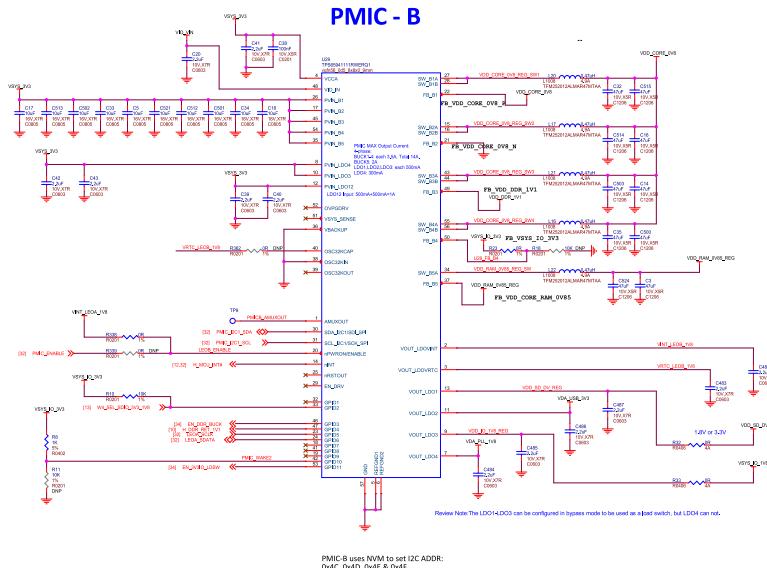


Figure 6.5. PMIC-B TPS65941111 circuit

### 6.1.2. DC Input

Figure 23 below shows how the DC input is connected to the **TPS65941213** and **TPS65941111**.

**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 **TPS65941213** and **TPS65941111**, 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 **TPS65941213** and **TPS65941111**, 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 **TPS65941213** and **TPS65941111** 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 **TPS65941213** and **TPS65941111** 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 **TPS65941213** and **TPS65941111**. 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 **TPS65941213** and **TPS65941111** 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 **TPS65941213** and **TPS65941111**. 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 6.1. Table 3. BeagleBone AI-64 Battery Pins**

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

There is no fuel gauge function provided by the **TPS65941213** and **TPS65941111**. That would need to be added if that function was required. If you want to add a fuel gauge, and 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 [TPS65941213](#) and [TPS65941111](#) 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 6.2. Table 4. BeagleBone AI-64 Power Consumption(mA@5V)**

MODE	USB	DC	DC+USB
<b>Reset</b>	TBD	TBD	TBD
<b>Idling @ UBoot</b>	210	210	210
<b>Kernel Booting (Peak)</b>	460	460	460
<b>Kernel Idling</b>	350	350	350
<b>Kernel Idling Display Blank</b>	280	280	280
<b>Loading a Webpage</b>	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 **TPS65941213** and **TPS65941111** via several different signals. Each of these signals is described below.

#### **I2C0**

I2C0 is the control interface between the processor and the **TPS65941213** and **TPS65941111**. It allows the processor to control the registers inside the **TPS65941213** and **TPS65941111** for such things as voltage scaling and switching of the input rails.

## **PMIC\_POWR\_EN**

On power up the **VDD\_RTC** rail activates first. After the RTC circuitry in the processor has activated it instructs the **TPS65941213** and **TPS65941111** to initiate a full power up cycle by activating the **PMIC\_POWR\_EN** signal by taking it HI. When powering down, the processor can take this pin low to start the power down process.

## **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.

## **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.

## **WAKEUP**

The **WAKEUP** signal from the **TPS65941213** and **TPS65941111** 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 **TPS65941213** and **TPS65941111**, such as the power button being pressed, it generates this signal.

## **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. 6.1.9 Power Rails***

[Figure 25](#) shows the connections of each of the rails from the **TPS65941213** and **TPS65941111**.

**Figure 25. Power Rails.*****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 **TPS65941213** and **TPS65941111**. It can deliver up to 250mA maximum.

***VDD\_3V3A Rail***

The **VDD\_3V3A** rail is supplied by the **TPS65941213** and **TPS65941111** and provides the 3.3V for the processor rails and can provide up to 400mA.

***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.

***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.

***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.

***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.

## ***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.

## ***Power Sequencing***

The power up process 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 **TPS65941213** and **TPS65941111** 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 **TPS65941213** and **TPS65941111** is configured. You can refer to the **TPS65941213** and **TPS65941111** datasheet for more information.

### **Figure 27. TPS65941213 and TPS65941111 Power Sequencing Timing.**

#### ***6.1.10. Power LED***

The power LED is a blue LED that will turn on once the **TPS65941213** and **TPS65941111** has finished the power up procedure. If you ever see the LED flash once, that means that the **TPS65941213** and **TPS65941111** 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. TPS65941213 and TPS65941111 Power Up Process**

Figure 28 shows the interface between the **TPS65941213 and TPS65941111** 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 **TPS65941213 and TPS65941111** connects the power to the SYS output pin which drives the switchers and LDOs in the **TPS65941213 and TPS65941111**.

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 **TPS65941213 and TPS65941111**, 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 **TPS65941213 and TPS65941111** which starts the **TPS65941213 and TPS65941111** power up process.

The **LDO\_PGOOD** signal is provided by the **TPS65941213 and TPS65941111** to the processor. As this signal is 1.8V from the **TPS65941213 and TPS65941111** by virtue of the **TPS65941213 and TPS65941111** 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 **TPS65941213 and TPS65941111**, this signal goes active releasing the processor. The LDOs on the **TPS65941213 and TPS65941111** 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 **TPS65941213 and TPS65941111** 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 **TPS65941213** and **TPS65941111** 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.

#### ***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.

#### ***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 **TPS65941213** and **TPS65941111** 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.

#### ***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 **TPS65941213** and **TPS65941111**.

## 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>.<sup>2</sup>

**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 6.3. Table 5. Processor Features**

<b>Operating Systems</b>	Linux, Android, Windows Embedded CE, QNX, ThreadX	<b>MMC/SD</b>	3
<b>Standby Power</b>	7 mW	<b>CAN</b>	2
<b>ARM CPU</b>	1 ARM Cortex-A8	<b>UART (SCI)</b>	6
<b>ARM MHz (Max.)</b>	275,500,600,800,1000	<b>ADC</b>	8-ch 12-bit
<b>ARM MIPS (Max.)</b>	1000,1200,2000	<b>PWM (Ch)</b>	3
<b>Graphics Acceleration</b>	1 3D	<b>eCAP</b>	3
<b>Other Hardware Acceleration</b>	2 PRU-ICSS, Crypto Accelerator	<b>eQEP</b>	3

<sup>2</sup> <https://www.ti.com/product/TDA4VM>

<b>On-Chip L1 Cache</b>	64 KB (ARM Cortex-A8)	<b>RTC</b>	1
<b>On-Chip L2 Cache</b>	256 KB (ARM Cortex-A8)	<b>I2C</b>	3
<b>Other On-Chip Memory</b>	128 KB	<b>McASP</b>	2
<b>Display Options</b>	LCD	<b>SPI</b>	2
<b>General Purpose Memory</b>	1 16-bit (GPMC, NAND flash, NOR Flash, SRAM)	<b>DMA (Ch)</b>	64-Ch EDMA
<b>DRAM</b>	1 16-bit (LPDDR-400, DDR2-532, DDR3-400)	<b>IO Supply (V)</b>	1.8V(ADC),3.3V
<b>USB Ports</b>	2	<b>Operating Temperature Range</b> <sup>②</sup>	-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 **TPS65941213** and **TPS65941111** 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 TPS65941213 and TPS65941111 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#:

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 **TPS65941213** and **TPS65941111**. 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\*.**

## **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.

## 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 6.4. 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: 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.

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

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.

## 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 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 **MMCO\_CMD** signal indicates that a command versus data is being sent.

There is no separate card detect pin in the microSD specification. It uses **MMCO\_DAT3** for that function. However, most microSD connectors still supply a CD function on the connectors. In 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.

## 6.6. User LEDs

There are four user LEDs on 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 6.5. 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.

## 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](http://beagleboard.org),<sup>3</sup> 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 BeagleBone AI-64, that is either eMMC or microSD based on the position of the boot switch.

### 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.

---

<sup>3</sup> <http://beagleboard.org/>

**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.

## 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 SPI0 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.

## 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 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 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.**

#### **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**.

#### **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.

## ***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.

## ***SYS\_RESET***

The reset of the LAN8710A is controlled via the **SYS\_RESETn** signal, the main board reset line.

## ***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.

## **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 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 6.6. 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***

guessing this doesn't exist on this device

### ***6.10.8. uDP Connector Interface***

insert processor Micro Display Port connector doc here

## **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.

## **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://git.beagleboard.org/beagleboard/am335x\\_pru\\_package](http://git.beagleboard.org/beagleboard/am335x_pru_package).<sup>4</sup> 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

---

<sup>4</sup> [http://git.beagleboard.org/beagleboard/am335x\\_pru\\_package](http://git.beagleboard.org/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 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 6.7. Table 11. PRU0 and PRU1 Access**

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

	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_tx <b>pr1_pru0_pru_r31_16</b> (Input)
	25	A14	GPIO3_21 <sup>a</sup>	pr1_pru0_pmu1_pmu5_pru_r31_5((Input) (Output))
	26	D16	UART1_RX	pr1_uart0_rx <b>pr1_pru1_pru_r31_16</b>
	27	C13	GPIO3_19	pr1_pru0_pmu1_pmu7_pru_r31_7 (Output) (Input)
	28	C12	SPI1_CS0	eCAP2_in_pmu2_pmu3_pmu8_pru_r31_3 (Output) (Input)
	29	B13	SPI1_D0	pr1_pru0_pmu1_pmu1_pru_r31_1 (Output) (Input)
	30	D12	SPI1_D1	pr1_pru0_pmu1_pmu2_pru_r31_2 (Output) (Input)
	31	A13	SPI1_SCLK	pr1_pru0_pmu1_pmu0_pru_r31_0 (Output) (Input)

<sup>a</sup> 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.



# Connectors

---

This section describes each of the connectors on the board.

## 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 theese 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 BY EXMODE[14:0]														
No.	NAME NUMBERINGS													
PIN ADDRESS BY EXTER 1 2 3 4 5 6 7 8 9 10 11 12 13 14 Bootstrap														
No.	NAME NUMBER													
P8_080	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	PR0	MCASP0_M1	MCASP0_M2	MCASP0_AFSX
P8_040	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR3	PR3	MCASP0_M40	AD0	MCASP0_AXR0	MCASP0_AXR1	MCASP0_AXR2
P8_050	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	MCASP0_M50	MCASP0_P2	MCASP0_P3	MCASP0_AFSX
P8_060	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	MCASP0_M60	MCASP0_P4	MCASP0_P5	MCASP0_AXR0
P8_070	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	MCASP0_M70	MCASP0_P6	MCASP0_P7	MCASP0_AXR1
P8_080	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	MCASP0_M80	MCASP0_P8	MCASP0_P9	MCASP0_AXR0
P8_090	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	MCASP0_M90	MCASP0_P10	MCASP0_P11	MCASP0_AXR0
P8_080	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	MCASP0_M80	MCASP0_P12	MCASP0_P13	MCASP0_AXR0
P8_010	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	MCASP0_M10	MCASP0_P14	MCASP0_P15	MCASP0_AXR0
P8_020	PA10	DA0	CR1	GR1	PR1	GR0	PR0	GR1	PR1	GR0	MCASP0_M20	MCASP0_P16	MCASP0_P17	MCASP0_AXR12







## 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.

## 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.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL232R_CABLES.pdf)<sup>1</sup>

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

---

<sup>1</sup> [http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS\\_TTL-232R\\_CABLES.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf)

[http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS\\_TTL-232R\\_CABLES.pdf](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**

## 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](http://www.digikey.com).<sup>2</sup>

---

<sup>2</sup> <http://www.digikey.com/>



# 8

## Cape Board Support

**BeagleBone AI-64** has the ability to accept up to four EEPROM addressable expansion boards or capes stacked onto the expansion headers. The word cape comes from the shape of the expansion board for BeagleBone boards as it is fitted around the Ethernet connector on the main board. For BeagleBone this notch acts as a key to ensure proper orientation of the cape. On AI-64 you can see a clear silkscreen marking for the cape orientation. Most of BeagleBone capes can be used with your BeagleBone AI-64 also like shown in [BeagleBone Ai Cape Placement figure](#) below.

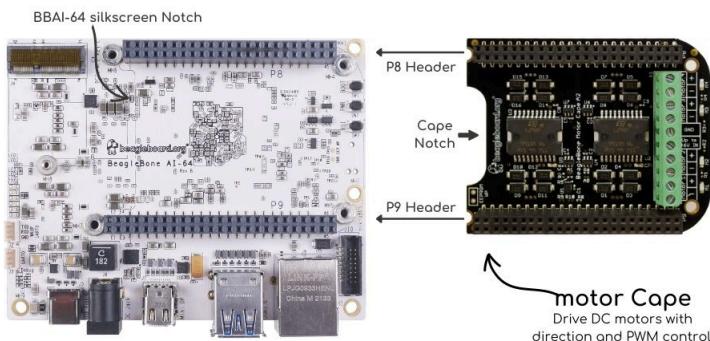


Figure 8.1. BeagleBone cape placement

This section describes the rules & guidelines for creating capes to ensure proper operation with 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.

For those wanting to create their own capes this should not put limits on the creation of capes and what they can do, but may set a few basic rules that will allow the software to administer their operation with BeagleBone AI-64. For this reason there is a lot of flexibility in the specification that we hope most people will find it liberating in the spirit of Open Source Hardware. On the other hand we are sure that there are others who 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<sup>1</sup>](#) 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.

## 8.1. BeagleBone AI-64 Cape Compatibility

The expansion headers on BeagleBone Black and BeagleBone AI-64 provides similar pin configuration options on P8 and P9 expansion header pins thus provide cape compatibility to a certain extent. Which means most BeagleBone Black capes will also be compatible with BeagleBone AI-64.



This section is still being worked on, please make sure you have the latest system reference manual (SRM).

**TODO: Add BBAI-64 LCD pins information**

**TODO: Add BBAI-64 eMMC pins information**

## 8.2. EEPROM

Each cape must have its own EEPROM containing information that will allow the software to identify the board and to configure the expansion headers pins during

---

<sup>1</sup> <https://git.beagleboard.org/beagleboard/beaglebone-ai-64/>

boot 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 BeagleBone AI-64.

The address of the EEPROM will be set via either jumpers or a dipswitch on each expansion board. [Expansion board EEPROM without write protect figure](#) below is the design of the EEPROM circuit.

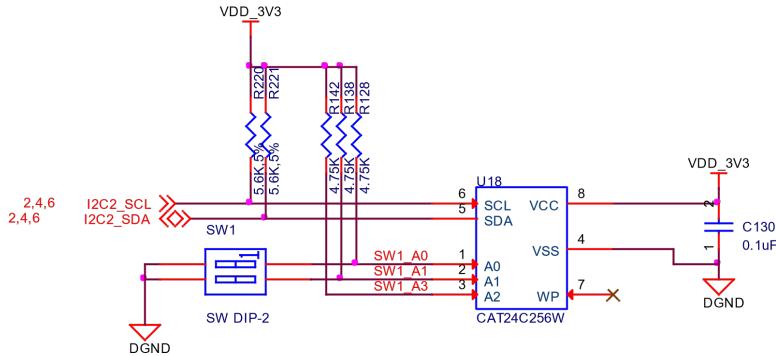


Figure 8.2. Expansion board EEPROM without write protect

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 [Expansion board EEPROM with write protect figure](#) 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**

---

## VSYS\_IO\_3V3

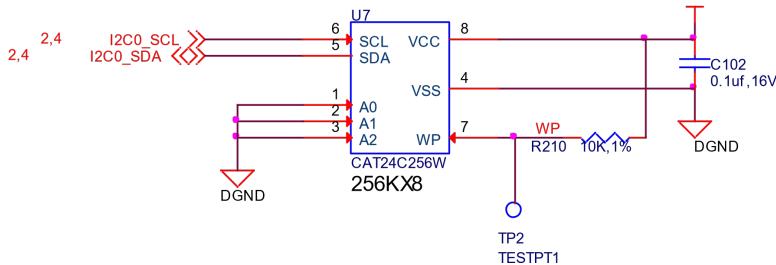


Figure 8.3. Expansion board EEPROM with write protect

#### 8.2.4. EEPROM Data Format

Expansion Board EEPROM table 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.

##### Clean/Update table

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..

Name	Offset	Size (bytes)	Contents
<b>Version</b>	<b>38</b>	<b>4</b>	<b>Hardware version code for board in ASCII. Version format is up to the developer. i.e. 02.1...00A1... .10A0</b>
<b>Manufacturer</b>	<b>42</b>	<b>16</b>	<b>ASCII name of the manufacturer. Company or individual's name.</b>
<b>Part Number</b>	<b>58</b>	<b>16</b>	<b>ASCII Characters for the part number. Up to maker of the board.</b>
<b>Number of Pins</b>	<b>74</b>	<b>2</b>	<b>Number of pins used by the daughter board including the power pins used. Decimal value of total pins 92 max, stored in HEX.</b>
<b>Serial Number</b>	<b>76</b>	<b>12</b>	<b>Serial number of the board. This is a 12 character string which is: WWYY&amp;&amp;&amp;&amp;nnnn</b>

Name	Offset	Size (bytes)	Contents
			<p>where: WW = 2 digit week of the year of production</p> <p>YY = 2 digit year of production</p> <p>&amp;&amp;&amp;=Assembly code to let the manufacturer 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 .....</p> <p>..... 1..0</p>

Name	Offset	Size (bytes)	Contents
			<p>Bit 15....Pin is used or not... 0=Unused by cape 1=Used by cape</p> <p>Bit 14-13...Pin Direction... ..1 0=Output 01=Input 11=BDIR</p> <p>Bits 12-7... Reserved..... ..should be all zeros</p> <p>Bit 6....Slew Rate .....0=Fast 1=Slow</p> <p>Bit 5....Rx Enable..... .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>
VSYS_IO_3V3 Current	236	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
DC_VDD_5V Current	238	2	Maximum current in milliamps. This is HEX value of the current in decimal 1500mA=0x05 0xDC 325mA=0x01 0x45
VSYS_5V0 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 DC_VDD_5V rail

Name	Offset	Size (bytes)	Contents
			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 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

??? shows 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.

[Add tables](#)

### 8.3. Pin Usage Consideration

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

### 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

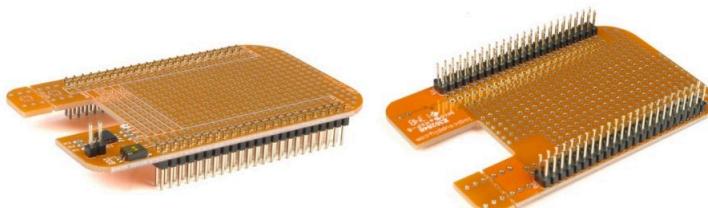
For non-stacking capes single configurations or where the cape can be the last board on the stack, the two 46 pin expansion headers use the same connectors.

[Single expansion connector figure](#) is a picture of the connector. These are dual row 23 position 2.54mm x 2.54mm connectors.



**Figure 8.4. Single expansion connector**

The connector is typically mounted on the bottom side of the board as shown in [Single cape expansion connector figure](#). 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 8.5. Single cape expansion connector on BeagleBone Proto Cape with EEPROM from onlogic**

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 [Single Cape Connectors](#) 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 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 BeagleBone AI-64

SUPPLIER	PARTNUMBER	TAIL LENGTH(in)	OVERHANG(in)
<a href="#">Major League<sup>2</sup></a>	TSHC-123-D-03-145-G-LF	.145	.004
<a href="#">Major League<sup>3</sup></a>	TSHC-123-D-03-240-G-LF	.240	.099
<a href="#">Major League<sup>4</sup></a>	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

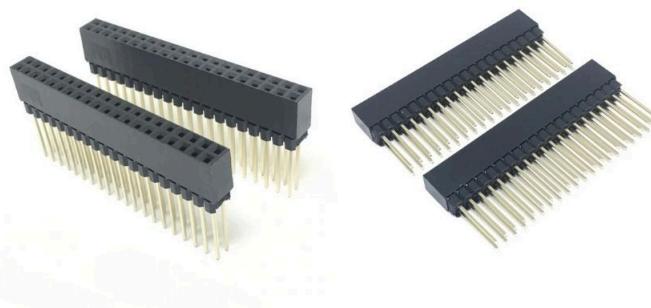
#### **8.4.2. Main Expansion Headers-Stacking**

For stacking configuration, the two 46 pin expansion headers use the same connectors. [Expansion connector figure](#) is a picture of the connector. These are dual row 23 position 2.54mm x 2.54mm connectors.

<sup>2</sup> <http://www.mlelectronics.com/>

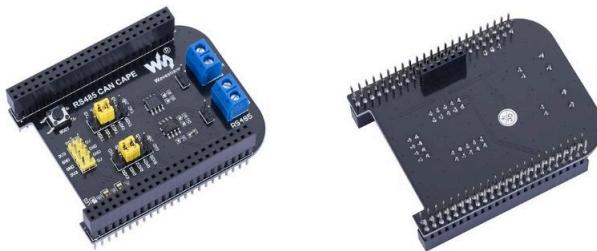
<sup>3</sup> <http://www.mlelectronics.com/>

<sup>4</sup> <http://www.mlelectronics.com/>



**Figure 8.6. Expansion Connector**

The connector is mounted on the top side of the board with longer tails to allow insertion into BeagleBone AI-64. [Stacked cape expansion connector figure](#) is the connector configuration for the 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 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 BeagleBone AI-64.

The third part listed in [Stacked cape connectors figure](#) will have insertion force issues.

SUPPLIER	PARTNUMBER	TAIL LENGTH(in)	OVERHANG(in)
<i>Major League</i> <sup>5</sup>	SSHQ-123-D-06-G-LF	.190	0.049
<i>Major League</i> <sup>6</sup>	SSHQ-123-D-08-G-LF	.390	0.249
<i>Major League</i> <sup>7</sup>	SSHQ-123-D-10-G-LF	.560	0.419

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*

??? 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 8.7. Stacked with signal stealing expansion connector figure

<sup>5</sup> <http://www.mlelectronics.com/>

<sup>6</sup> <http://www.mlelectronics.com/>

<sup>7</sup> <http://www.mlelectronics.com/>

#### 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 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 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

[Connector pin insertion depth figure](#) shows the key measurements used in calculating how much the pin extends past the contact point on the connector, what we call overhang.

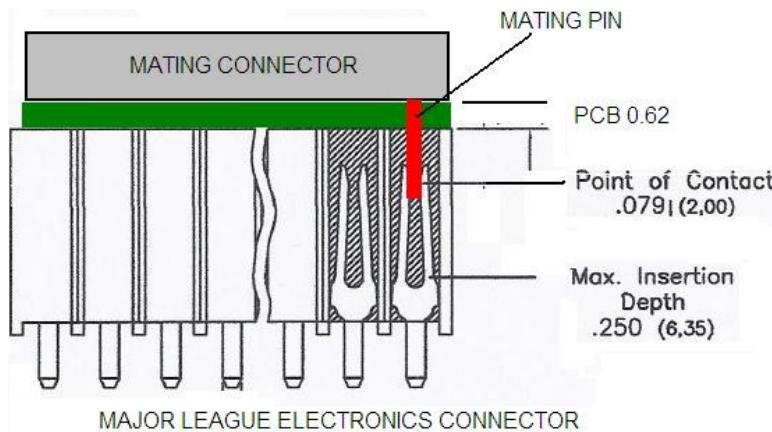


Figure 8.8. Connector Pin Insertion Depth

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 [Expansion header voltages figure](#) 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.

Current	Name	P9	P9	Name	Current
250mA	VSYS_IO_3V3		4	VSYS_IO_3V	250mA

Current	Name	P9	P9	Name	Current
1000mA	DC_VDD_5V5		6	DC_VDD_5V	1000mA
250mA	VSYS_5V0	7	8	VSYS_5V0	250mA

The **VSYS\_IO\_3V3** rail is supplied by the LDO on 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.

**DC\_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 **VSYS\_5V0** 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 **DC\_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 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 **VSYS\_IO\_3V3** rail is up.**

3. Do not apply any voltages that are generated from external sources.
  4. If voltages are generated from the DC\_VDD\_5V signal, those supplies must not become active until after the VSYS\_IO\_3V3 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.**

TODO: Add BBAI-64 cape mechanical characteristics

### **8.6.3. Standard Cape Size**

[Cape board dimensions figure](#) shows the outline of the standard cape. The dimensions are in inches.

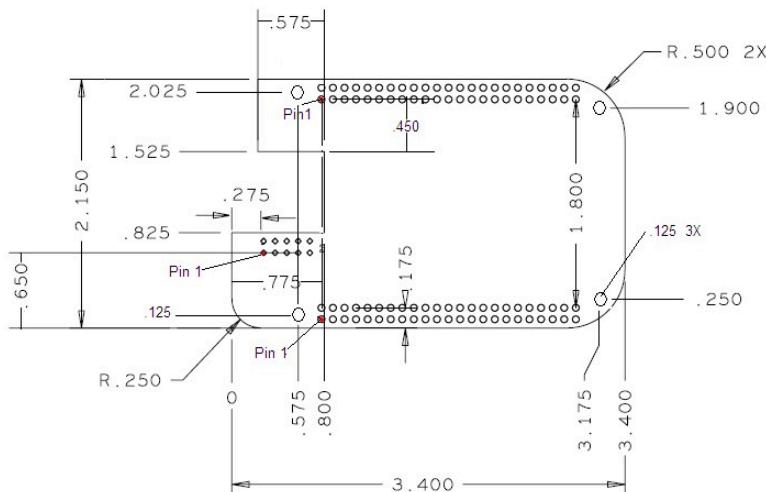


Figure 8.9. Cape board dimensions

A notch is provided for BeagleBone 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 BeagleBone board. On BeagleBone AI-64 board align it with the notch on the board silkscreen.

#### ***8.6.4. Extended Cape Size***

Capes larger than the standard board size are also allowed. A good example would be the new BeagleBone AI-64 robotics cape. There is no practical limit to the sizes of these types of boards. The notch is also optional, but it is up to the supplier to ensure that the cape is not plugged incorrectly on BeagleBone AI-64 such that damage would be caused to BeagleBone AI-64. 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.

---

# 9

## BeagleBone AI-64 Mechanical

---

### 9.1. Dimensions and Weight

Size: **TODO**

Max height: **TODO**

PCB Layers: **TODO**

PCB thickness: 2mm (0.08")

RoHS Compliant: Yes

Weight: 192gm ()

### 9.2. Silkscreen and Component Locations

**Figure 9.1. TODO: Board Dimensions**

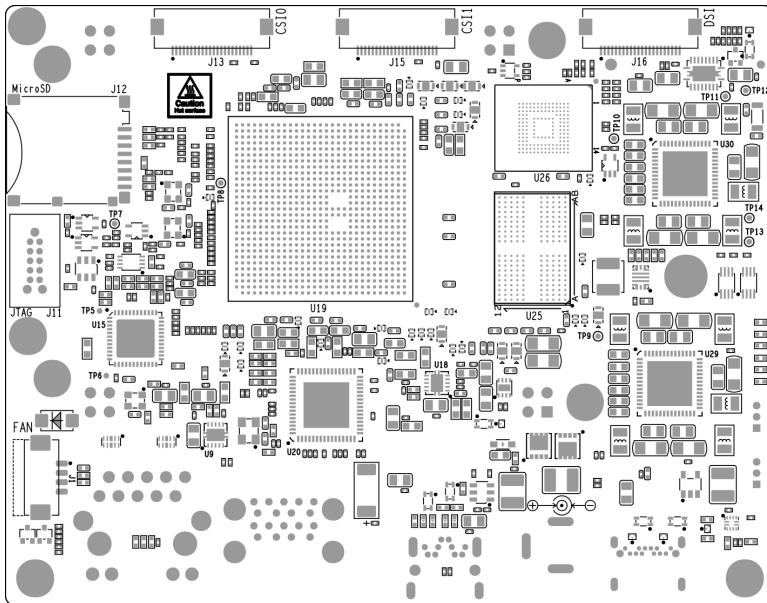
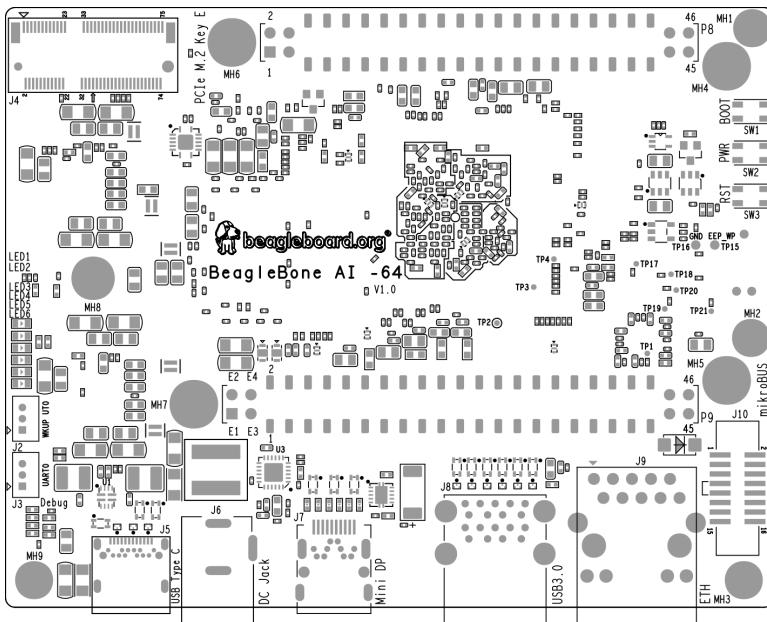


Figure 9.2. Top silkscreen



**Figure 9.3. Bottom silkscreen**

# 10

## Pictures

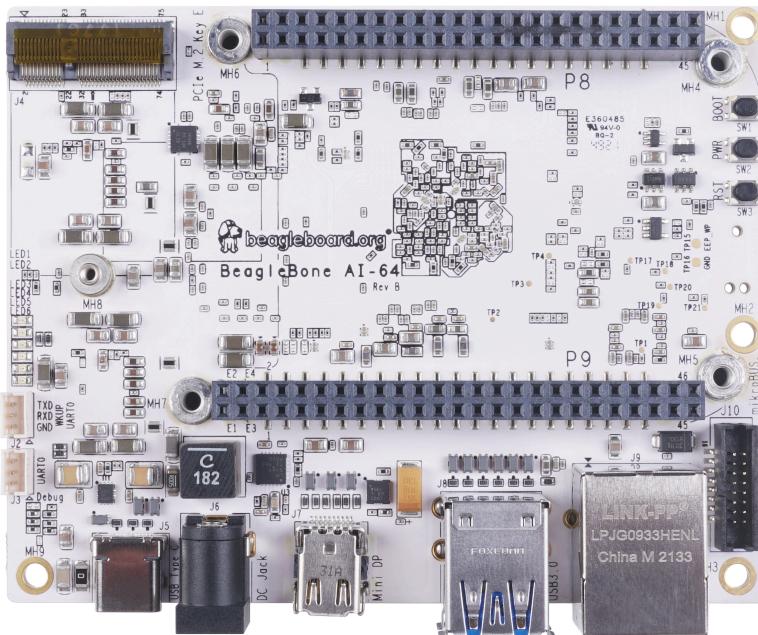


Figure 10.1. BBAI-64 front

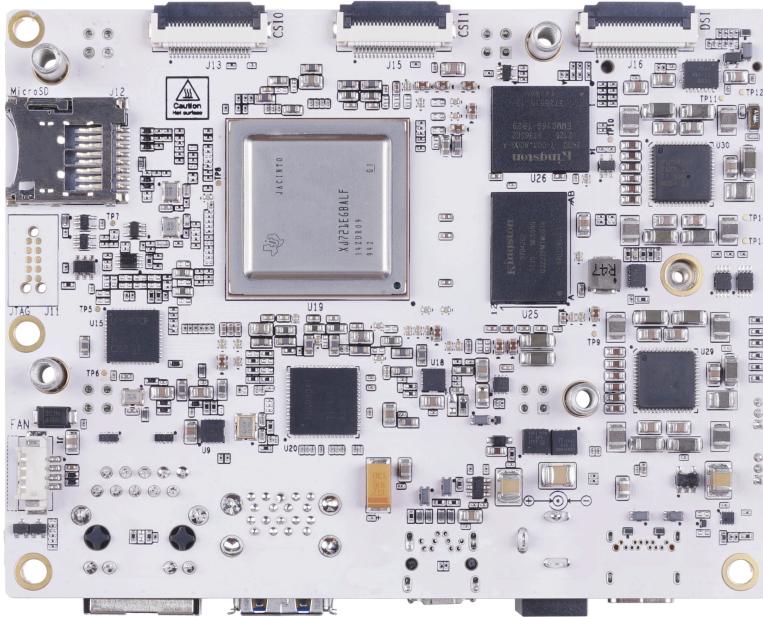
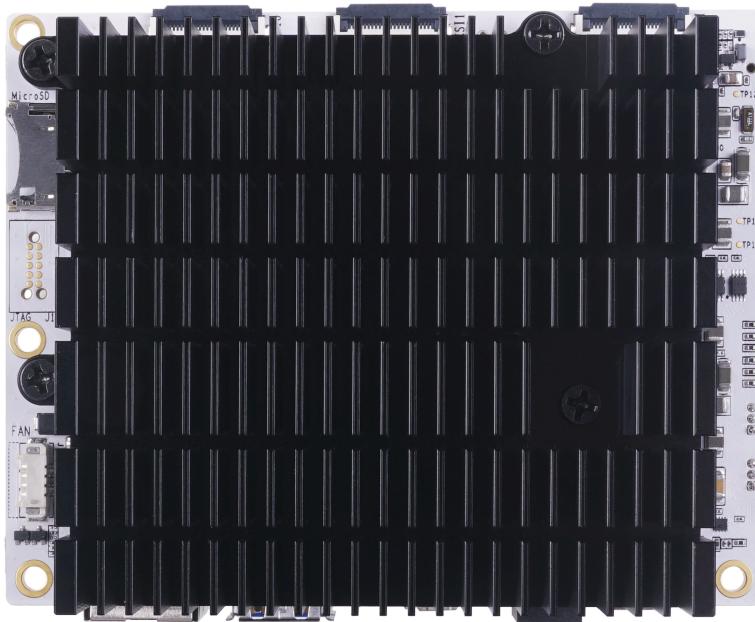


Figure 10.2. BBAI-64 back



**Figure 10.3. BBAI-64 back with heatsink**

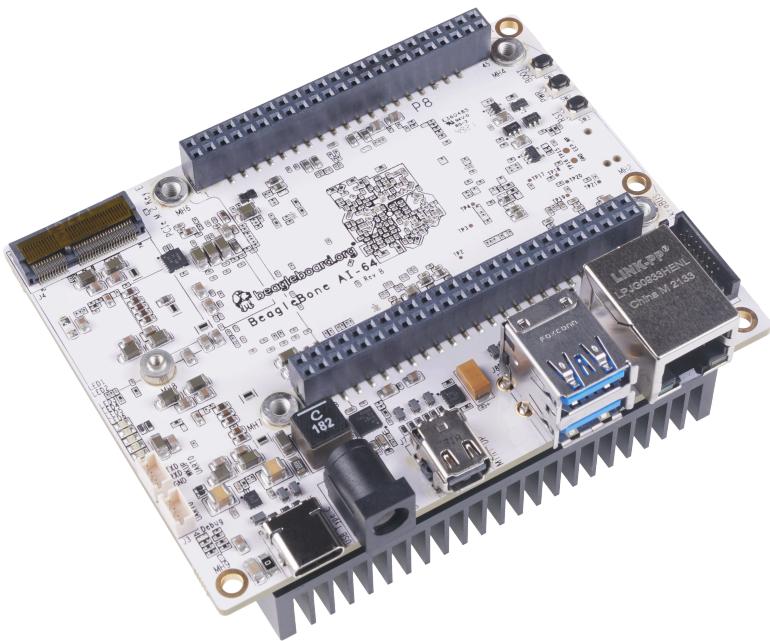


Figure 10.4. BBAI-64 front at 45° angle

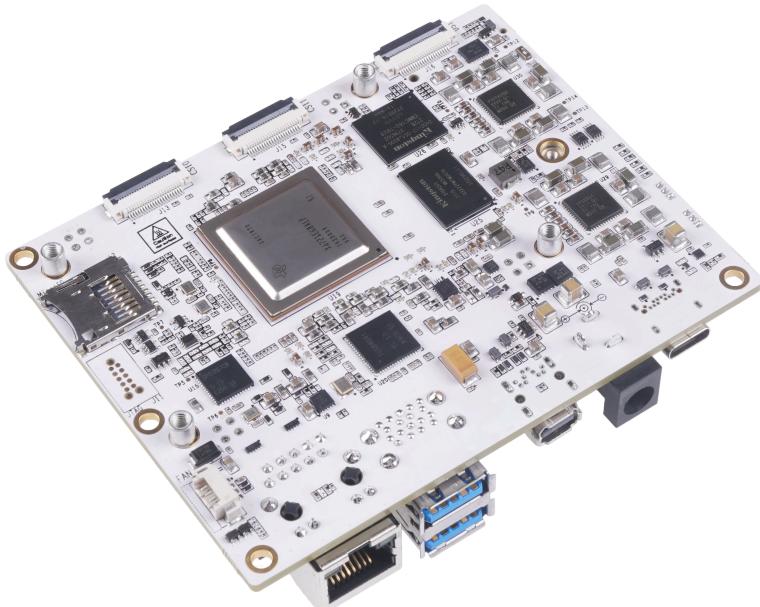
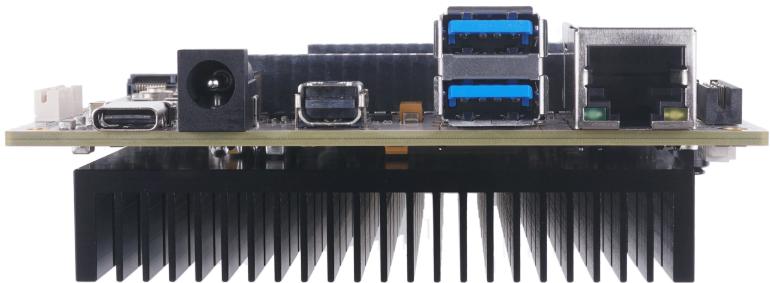


Figure 10.5. BBAI-64 back at 45° angle



**Figure 10.6. BBAI-64 back with heatsink at 45° angle**



**Figure 10.7. BBAI-64 ports**

---

# 11

## Support Information

---

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

[BeagleBoard.org forum](https://forum.beagleboard.org/)<sup>1</sup>.

### 11.1. Hardware Design

**Update required**

You can find all BBAI-64 hardware files [here](#)<sup>2</sup>.

### 11.2. Software Updates

You can download and flash the new image onto your BBAI-64 from [this](#)<sup>3</sup> image repository from Robert.

**Update required**

To see what SW revision is loaded into the eMMC check /etc/dogtag. It should look something like as shown below,

```
debian@BeagleBone:~$ cat /etc/dogtag
BeagleBoard.org Debian Bullseye Xfce Image 2022-05-26
```

---

<sup>1</sup> <https://forum.beagleboard.org/>

<sup>2</sup> <https://git.beagleboard.org/beagleboard/beaglebone-ai-64/-/tree/master/hw>

<sup>3</sup> <https://rcn-ee.net/rootfs/debian-arm64/>

## 11.3. RMA Support

### Update required

If you feel your board is defective or has issues, request an Return Merchandise Application (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.

TODO: Add serial number images

## 11.4. Troubleshooting video output issues

### TODO

#### 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](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 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, you can post your queries on [BeagleBoard.org forum<sup>4</sup>](https://forum.beagleboard.org).

---

<sup>4</sup> <https://forum.beagleboard.org/>

---

# Bibliography

TODO

## Articles

A

## Books

B



---

# Index

TODO

