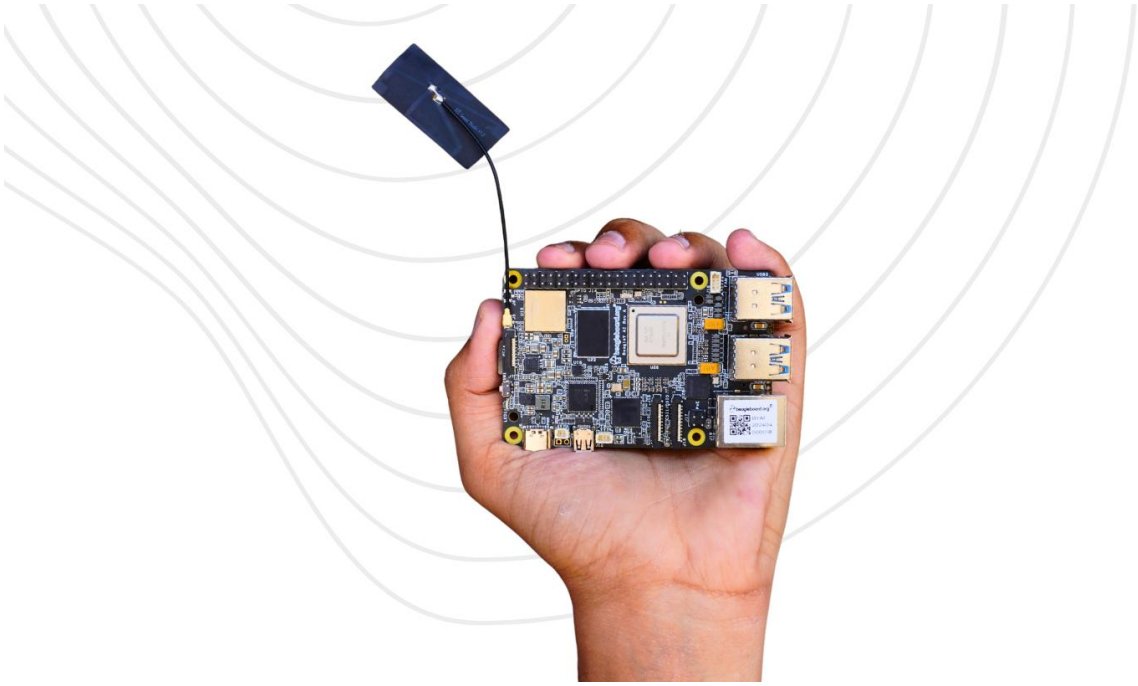
BeagleY-AI

BeagleBoard.org Foundation
Sep 06, 2024

Table of contents

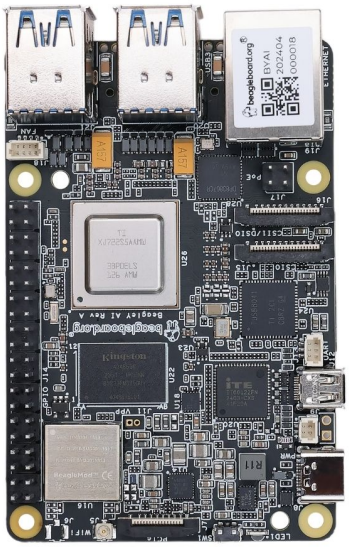
BeagleY-AI is an open-source single board computer based on the Texas Instruments AM67A Arm-based vision processor.

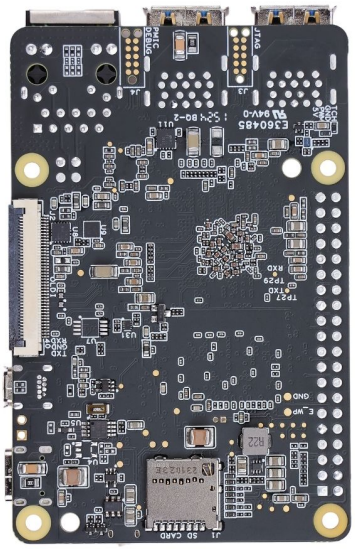


Chapter 1

Introduction

BeagleY-AI is an open-source single board computer designed for edge AI applications.





1.1 Detailed overview

BeagleY-AI is based on the Texas Instruments AM67A Arm-based vision processor. It features a quad-core 64-bit Arm®Cortex®-A53 CPU subsystem at 1.4GHz, Dual general-purpose C7x DSP with Matrix Multiply Accelerator (MMA) capable of 4 TOPs each, Arm Cortex-R5 subsystem for low-latency I/O and control, a 50 GFlop GPU, video and vision accelerators, and other specialized processing capability.

Table 1.1: BeagleY-AI features

Feature	Description
Processor	Texas Instruments AM67A, Quad 64-bit Arm® Cortex®-A53 @1.4 GHz, multiple cores including Arm/GPU processors, DSP, and vision/deep learning accelerators
RAM	4GB LPDDR4 (x32)
Wi-Fi/Bluetooth	BeagleBoard.org BeagleMod BM3301, 802.11ax Wi-Fi 6, Bluetooth Low Energy (BLE) 5.4
USB Ports	4 x USB 3.0 TypeA ports supporting simultaneous 5Gbps operation, 1 x USB 2.0 TypeC, supports USB 2.0 device mode
Ethernet	Gigabit Ethernet, with PoE+ support (requires separate PoE HAT)
Camera/Display	2 x 4-lane MIPI camera connector (one connector muxed with DSI capability)
Display Output	1 x HDMI display, 1 x OLDI display, 1 x DSI MIPI Display (DSI muxed with 1 CSI)
Real-time Clock (RTC)	Supports external coin-cell battery for power failure time retention
Debug UART	1 x 3-pin debug UART
Power	5V/3A DC power via USB-C
Power Button	On/Off included
PCIe Interface	PCI-Express® Gen3 x 1 interface for fast peripherals (requires separate M.2 HAT or other adapter)
Expansion Connector	40-pin header
Fan connector	1 x 4-pin fan connector, supports PWM control and fan speed measurement
Storage	microSD card slot with UHS-1 support
Tag Connect	1 x JTAG, 1 x External PMIC programming port

1.1.1 AM67A SoC

The AM67A scalable processor family is based on the evolutionary Jacinto™ 7 architecture, targeted at Smart Vision Camera and General Compute applications and built on extensive market knowledge accumulated over a decade of TI's leadership in the Vision processor market. The AM67A family is built for a broad set of cost-sensitive high performance compute applications in Factory Automation, Building Automation, and other markets.

Some Applications include:

- Human Machine Interface (HMI)
- Hospital patient monitoring
- Industrial PC
- Building security system
- Off-highway vehicle
- Test and measurement
- Energy storage systems
- Video Surveillance
- Machine Vision
- Industrial mobile robot (AGV/AMR)
- Front camera systems

The AM67A provides high performance compute technology for both traditional and deep learning algorithms at industry leading power/performance ratios with a high level of system integration to enable scalability and lower costs for advanced vision camera applications. Key cores include the latest Arm and GPU processors for general compute, next generation DSP with scalar and vector cores, dedicated deep learning and traditional algorithm accelerators, an integrated next generation imaging subsystem (ISP), video codec, and MCU cores. All protected by industrial-grade security hardware accelerators.

Tip: For more information about AM67A SoC you can checkout https://www.ti.com/product/AM67A

1.2 Board components location

1.2.1 Front components

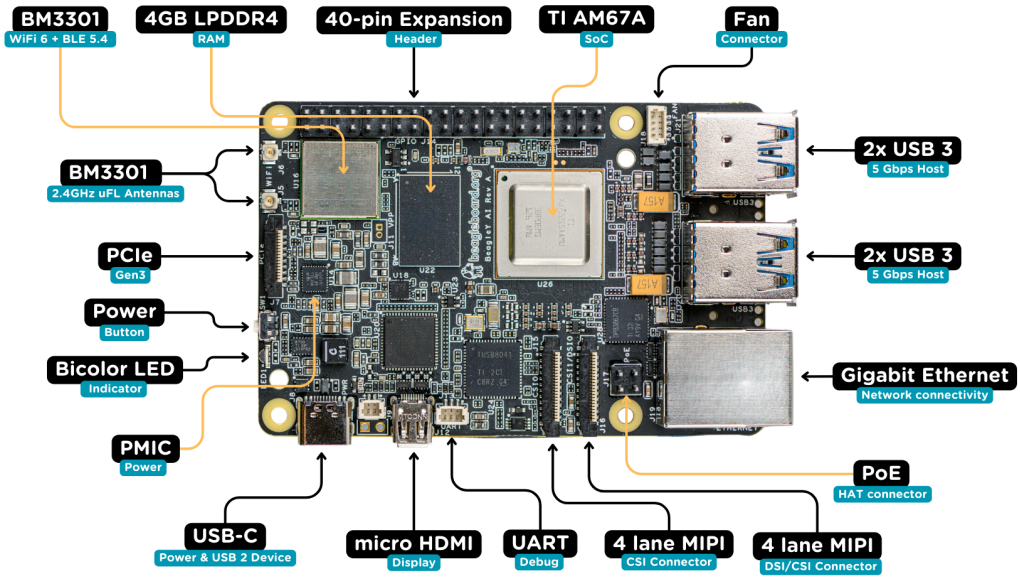


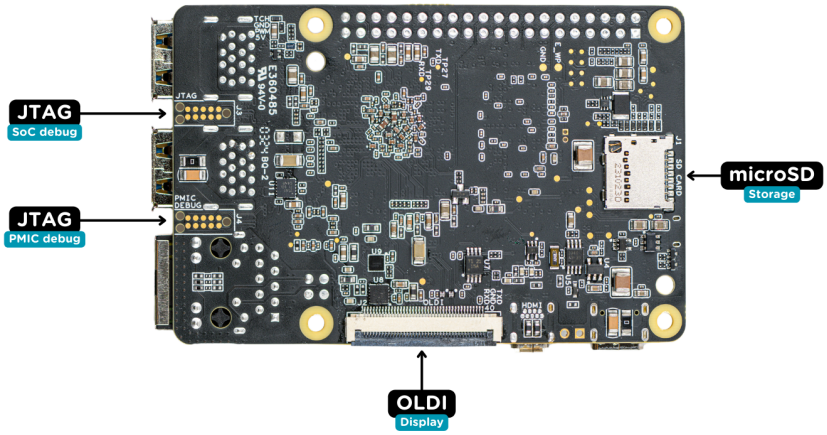
Table 1.2: BeagleY-AI board front components location

Feature	Description
WiFi/Bluetooth	BeagleBoard.org BeagleMod BM3301 with 802.11ax Wi-Fi 6 & Bluetooth Low Energy 5.4 (BLE)
RAM	4GB LPDDR4 (x32)
Expansion	40pin Expansion header compatible with HATs
SoC	TI AM67A Arm®Cortex®-A53 4 TOPS vision SoC with RGB-IR ISP for 4 cameras, machine vision, robotics, and smart HMI
Fan	4-pin fan connector
USB-A	4 x USB 3 TypeA ports supporting simultaneous 5Gbps operation host ports
Network Connectivity	Gigabit Ethernet
PoE	Power over Ethernet HAT connector
Camera/Display	1 x 4-lane MIPI camera/display transceivers, 1 x 4-lane MIPI camera
Debug UART	1 x 3-pin JST-SH 1.0mm debug UART port
Display Output	1 x HDMI display
USB-C	1 x Type-C port for power, and supports USB 2 device
PMIC	Power Management Integrated Circuit for 5V/5A DC power via USB-C with Power Delivery support
Bicolor LED	Indicator LED
Power button	ON/OFF button
PCIe	PCI-Express® Gen3 x 1 interface for fast peripherals (requires separate M.2 HAT or other adapter)

1.2.2 Back components

Table 1.3: BeagleY-AI board back components location

Feature	Description
Tag-Connect	1 x JTAG & 1 x Tag Connect for PMIC NVM Programming
Display output	1 x OLDI display
Storage	microSD card slot with support for high-speed SDR104 mode


JTAG
SoC debug
JTAG
PMIC debug
microSD
Storage
OLDI
Display

Chapter 2

BeagleY-AI Quick Start

2.1 What's included in the box?

When you purchase a BeagleY-AI, you'll get the following in the box:

1. BeagleY-AI with attached antenna.
2. Quick-start card

Todo: BeagleY-AI unboxing video

2.2 Getting started

To get started your BeagleY-AI you need the following:

1. 5V @ 3A power supply
2. MicroSD card (32GB)
3. *Boot Media (Software image)*

You may need additional accessories based on the mode of operation, you can use your BeagleY-AI in different ways.

1. *USB Tethering by directly connecting via USB type-c port*
2. *Headless connection via UART debug port*
3. *Standalone connection with Monitor and other peripherals attached*

Easiest option is to connect the board directly to your PC or Laptop using a USB type-C to type-c cable. There is only one USB type-C port on board, if you choose to use a dedicated power supply for first time setup, you may choose to access the board via any other methods listed above.

2.3 Power Supply

To power the board you can either connect it to a dedicated power supply like a mobile charger or a wall adapter that can provide 5V ≥ 3A. Checkout the docs power supply page for power supply recommendations.

Note: Instead of using a power supply or power adapter if you are using a Type-C to Type-C cable to connect the board to your laptop/PC then make sure it can supply at least 1000mA.

2.4 Boot Media (Software image)

We have two methods to prepare bootable microSD card, It is recommended to use *bb-imager*.

1. *bb-imager*
2. *Balena Etcher*

2.4.1 bb-imager

Download and install bb-imager for your operating system. Below are all the steps required to create a bootable microSD card with latest/recommended OS image for BeagleY-AI.

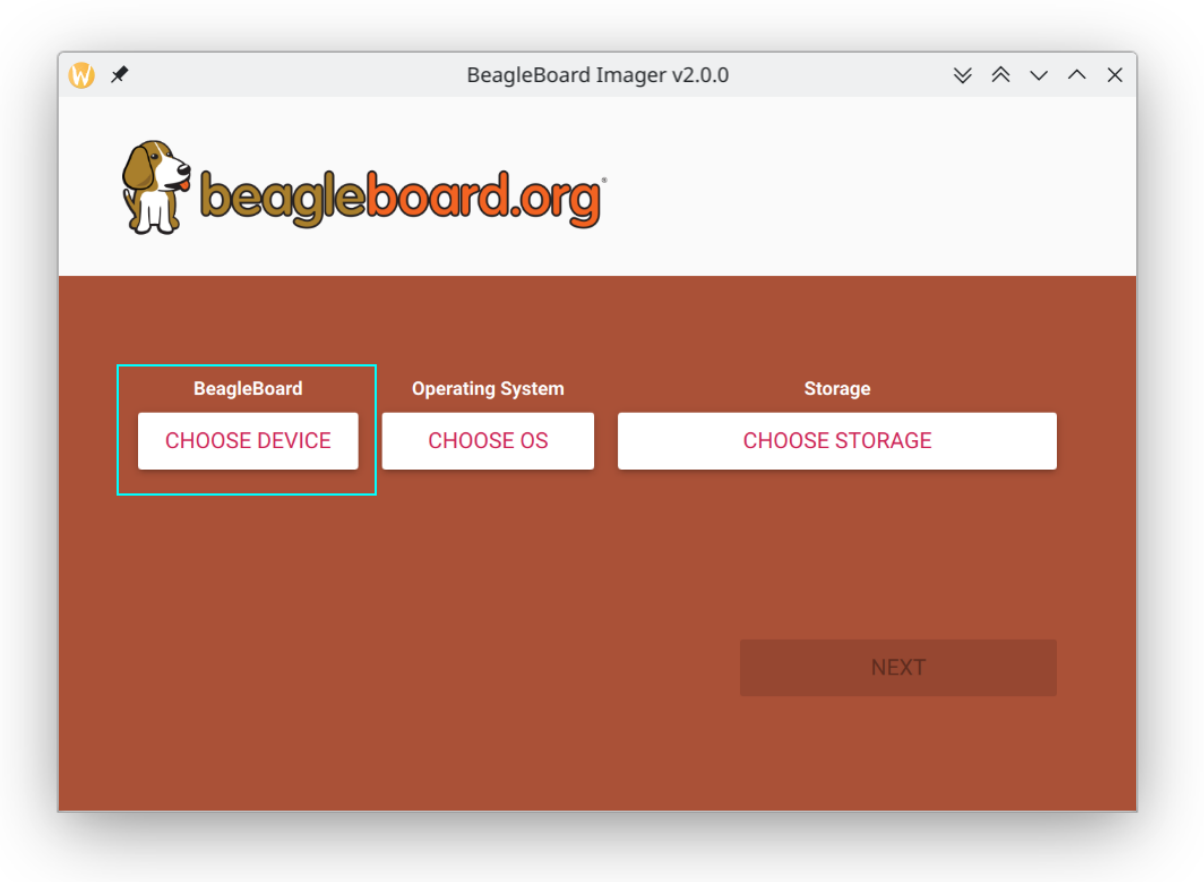


Fig. 2.1: Click on `CHOOSE DEVICE` button

2.4.2 Balena Etcher

Download and install Balena Etcher and then download the boot media from https://www.beagleboard.org/distros/beagley-ai-debian-12-5-2024-06-19-xfce. Flash it on a microSD card using Balena Etcher following the steps below:

1. Select downloaded boot media
2. Select microSD card
3. Flash!

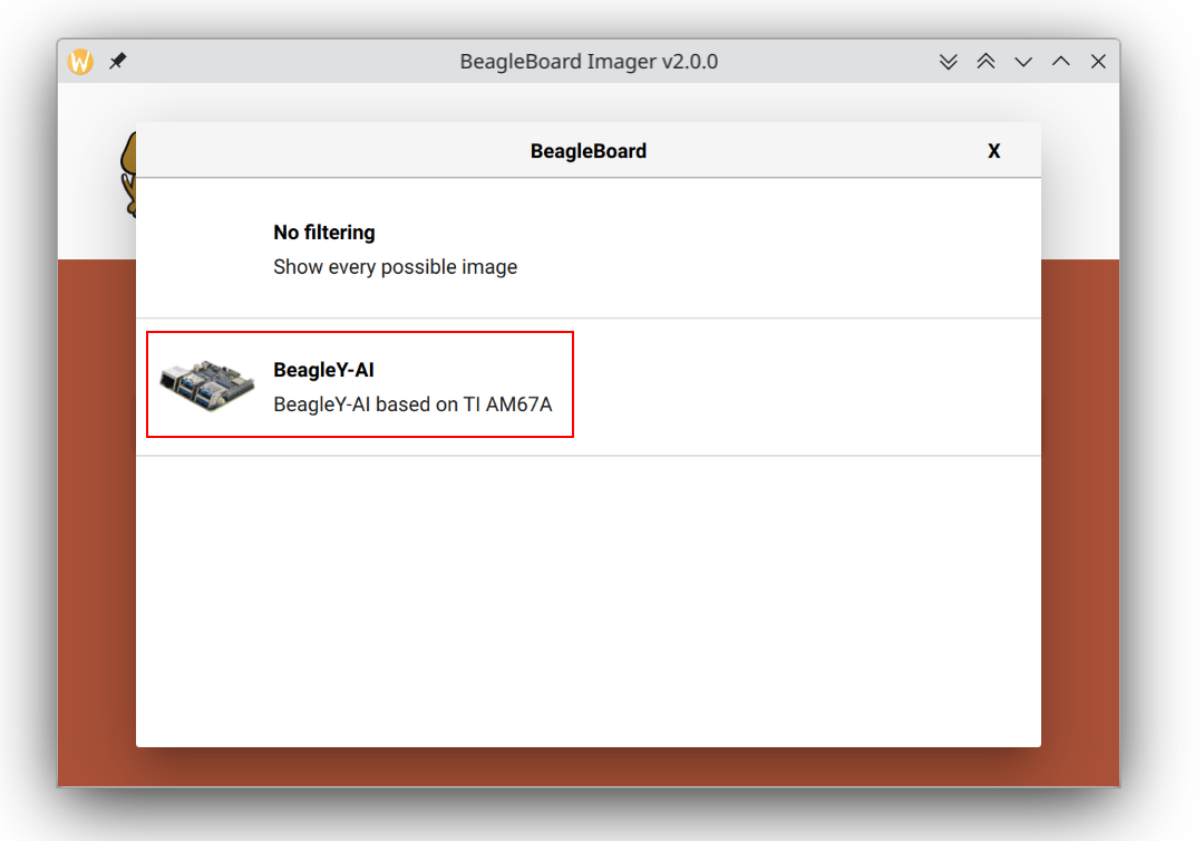


Fig. 2.2: Choose `BeagleY-AI` board

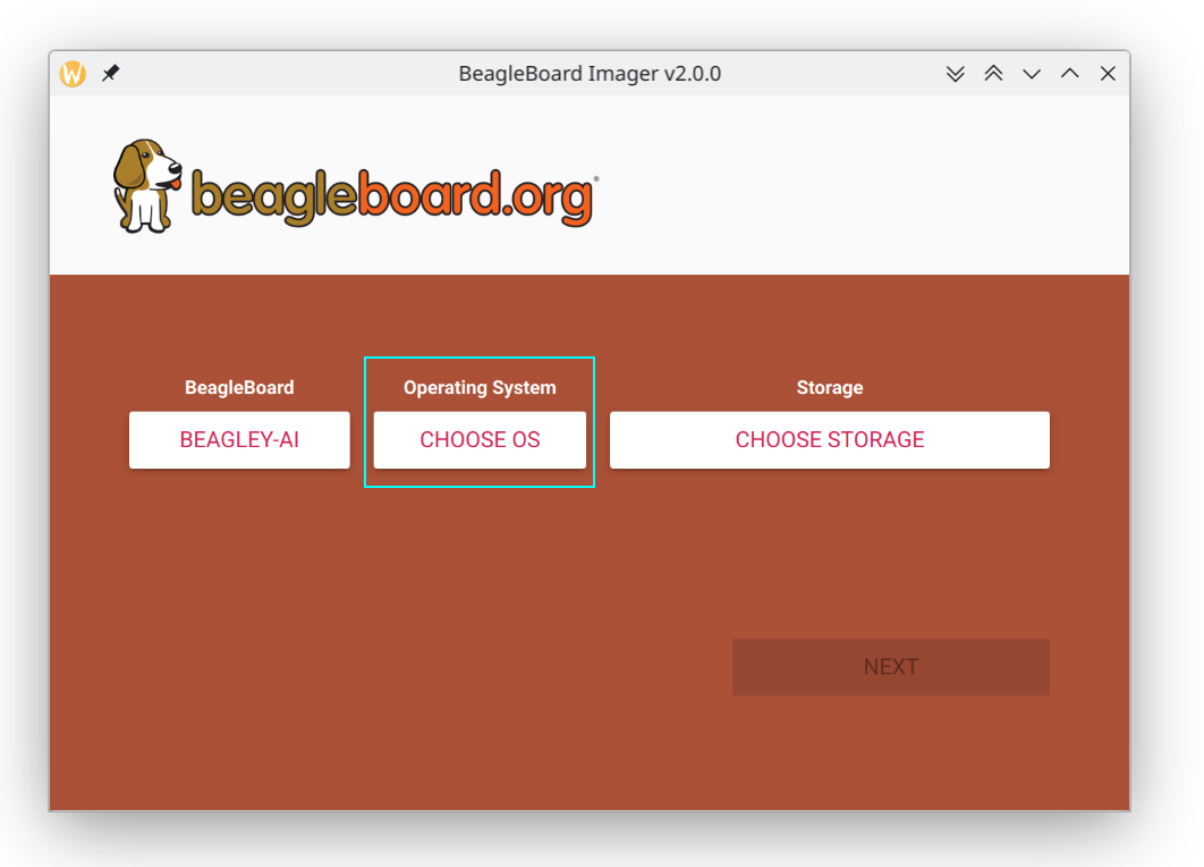


Fig. 2.3: Click on `CHOOSE OS` button

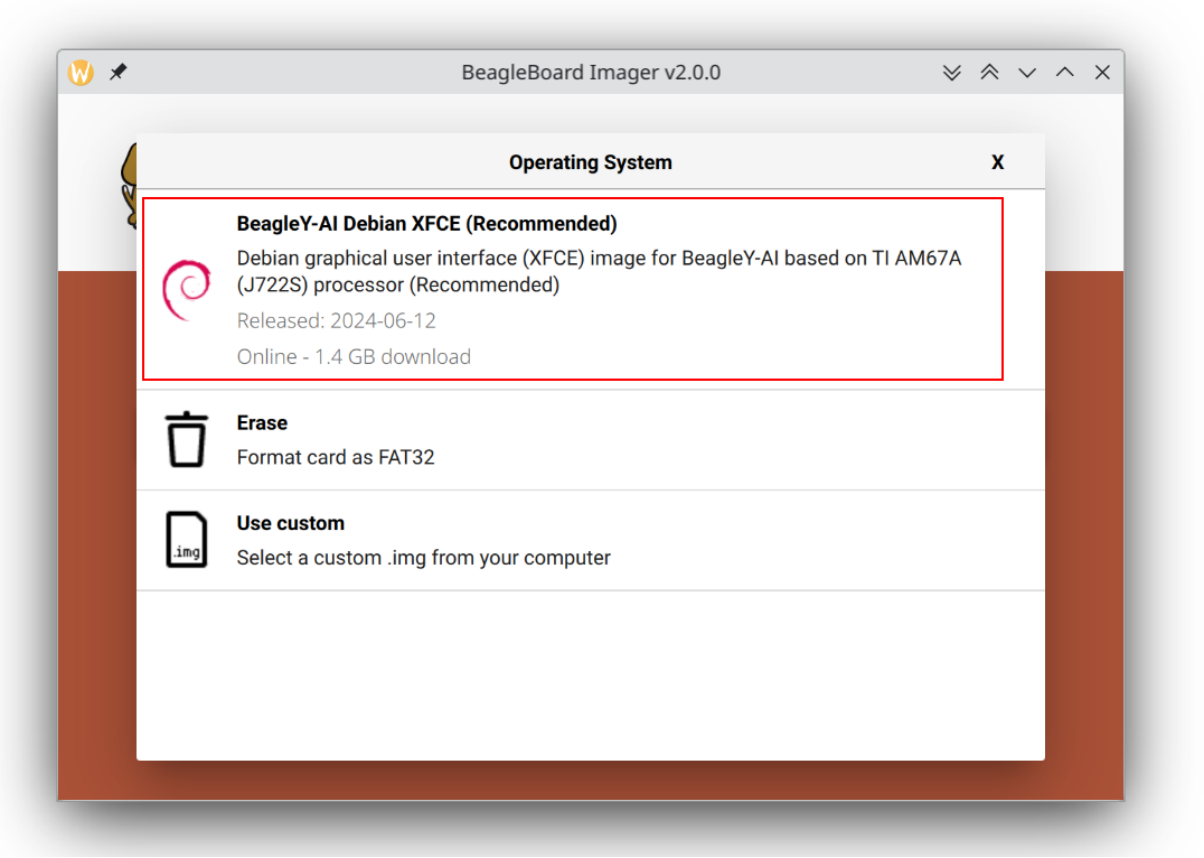


Fig. 2.4: Select `Recommended OS`

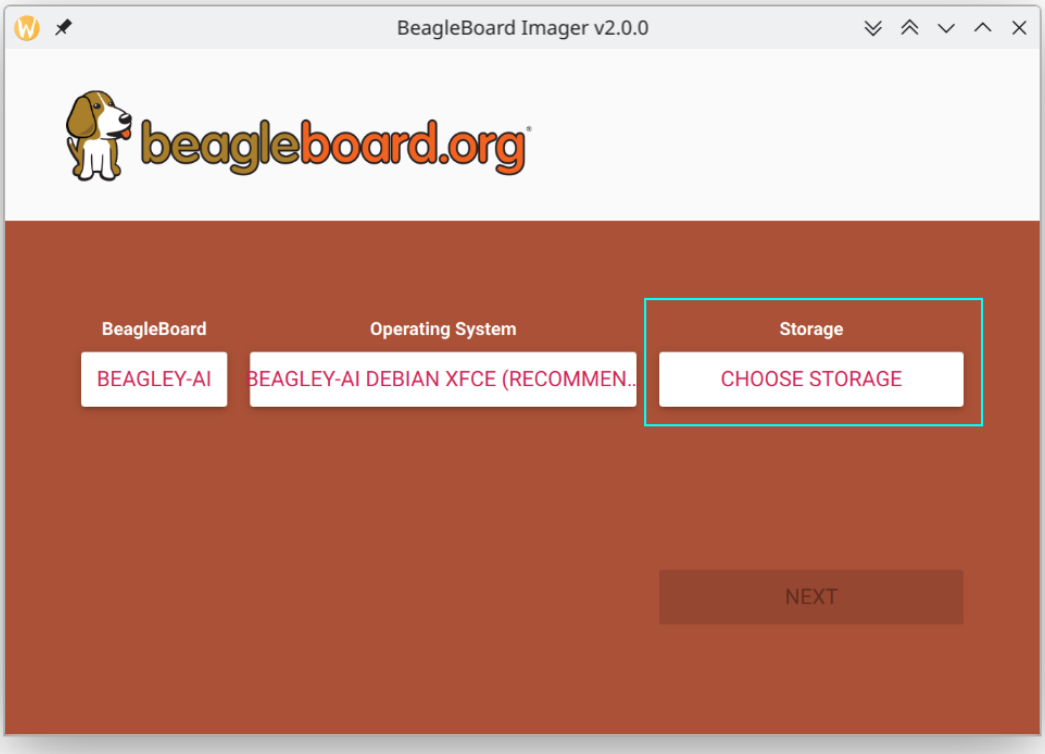


Fig. 2.5: Click on CHOOSE STORAGE button

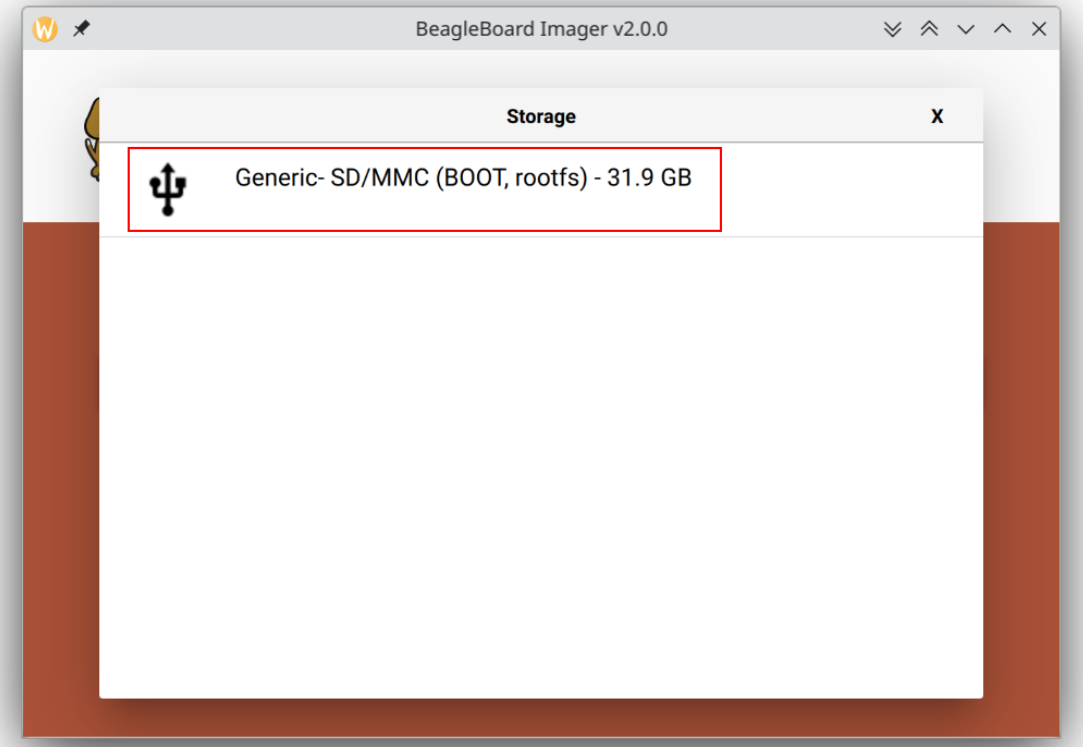


Fig. 2.6: Choose your microSD card

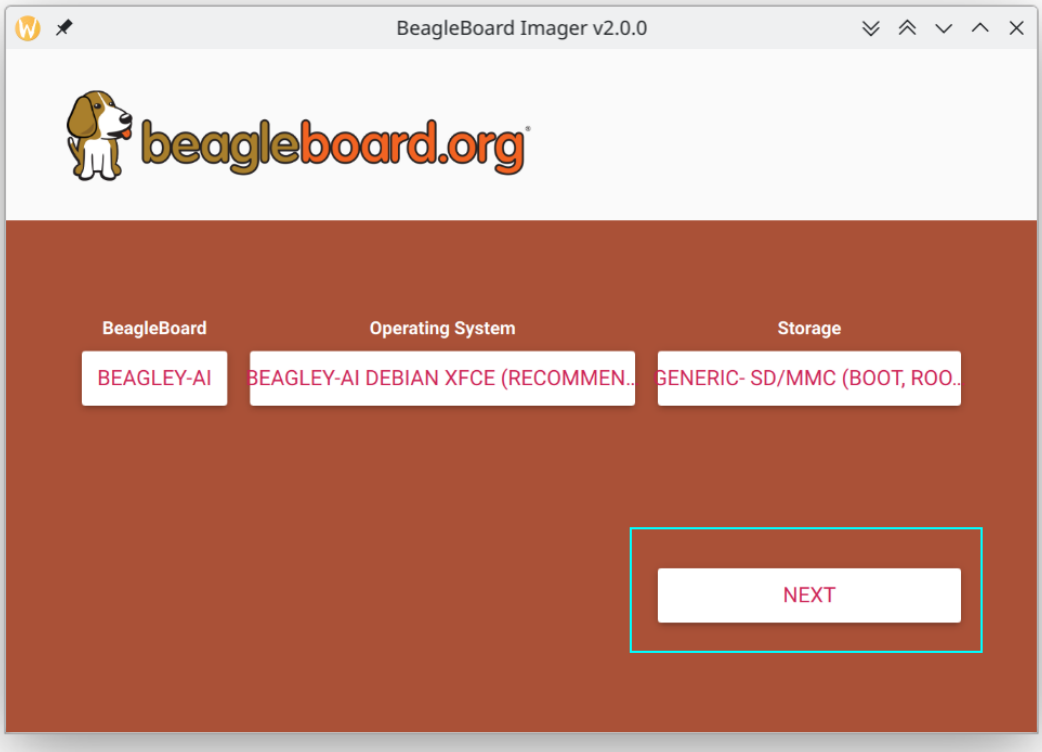


Fig. 2.7: Click on `Next` button

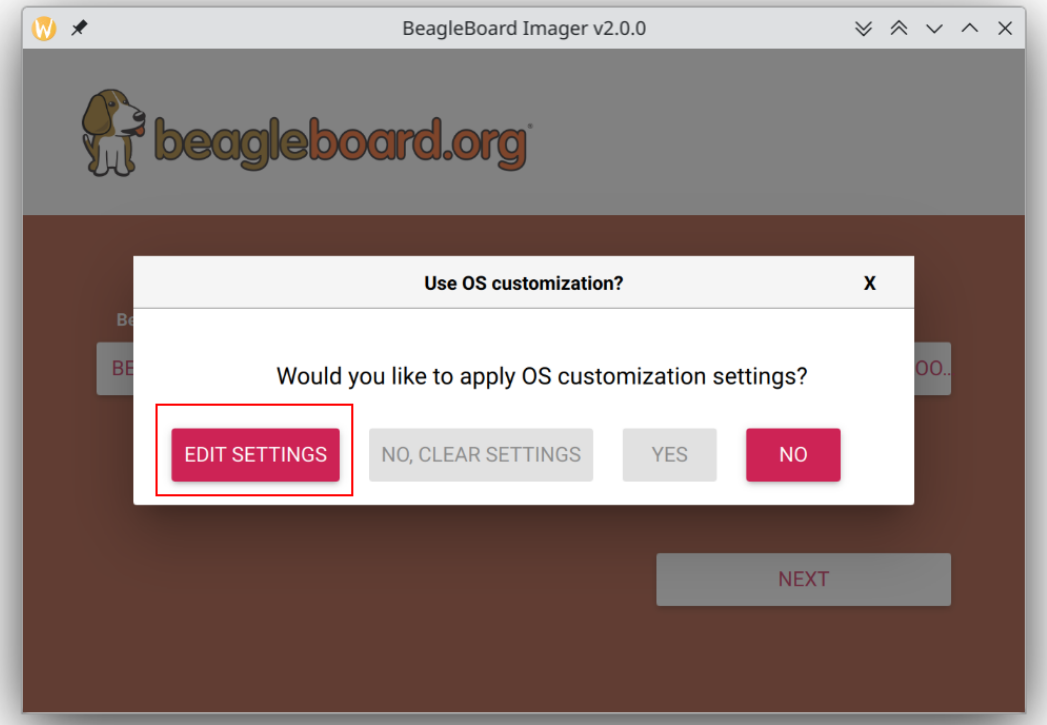


Fig. 2.8: Click on `EDIT SETTINGS` button

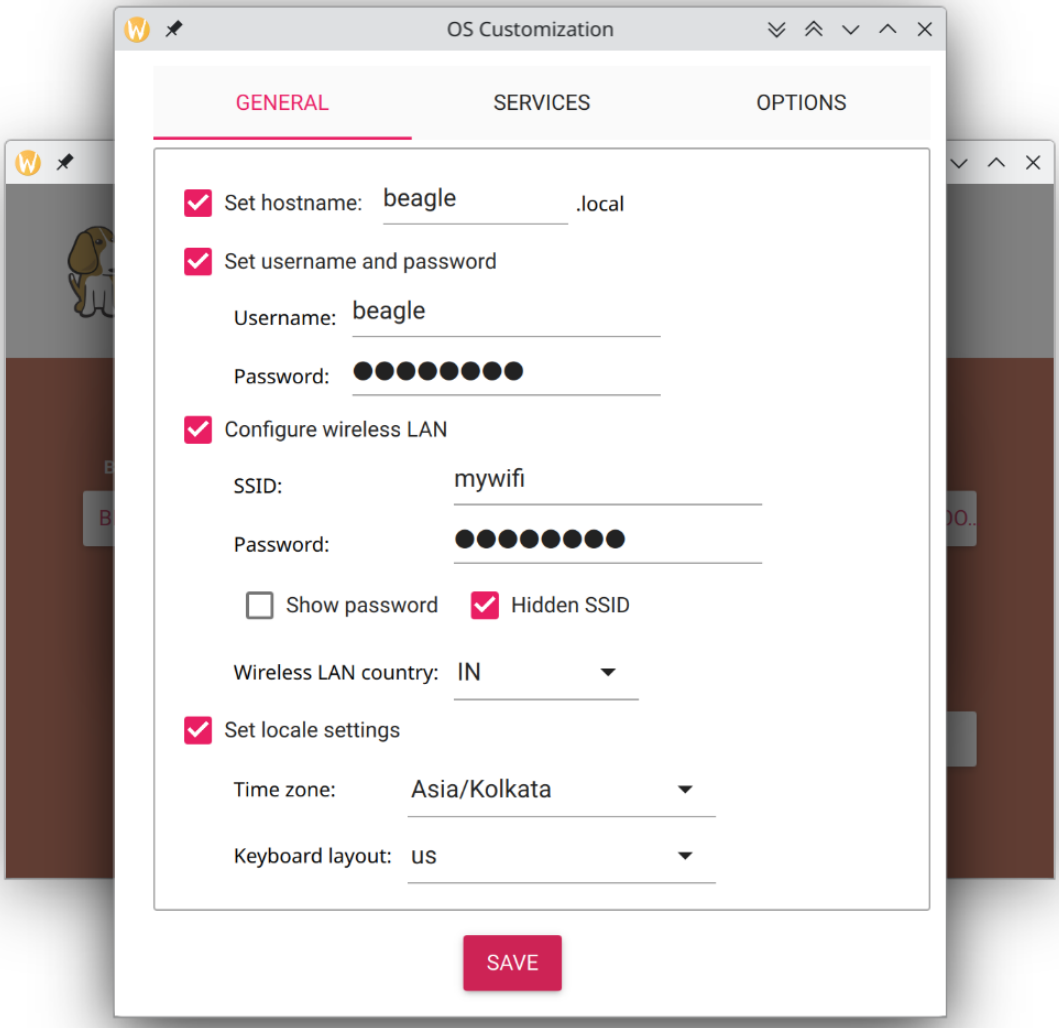


Fig. 2.9: Edit settings

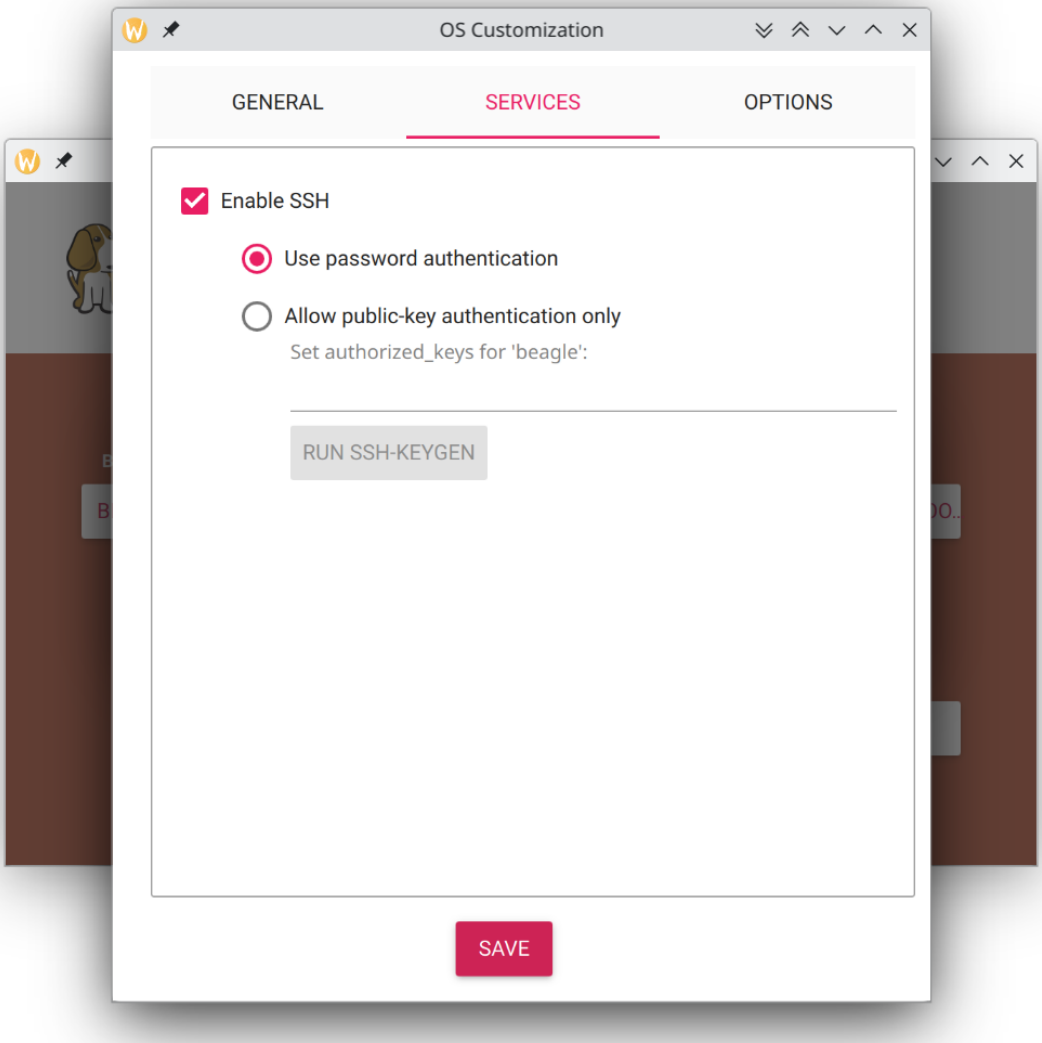


Fig. 2.10: Under SERVICES you can enable SSH

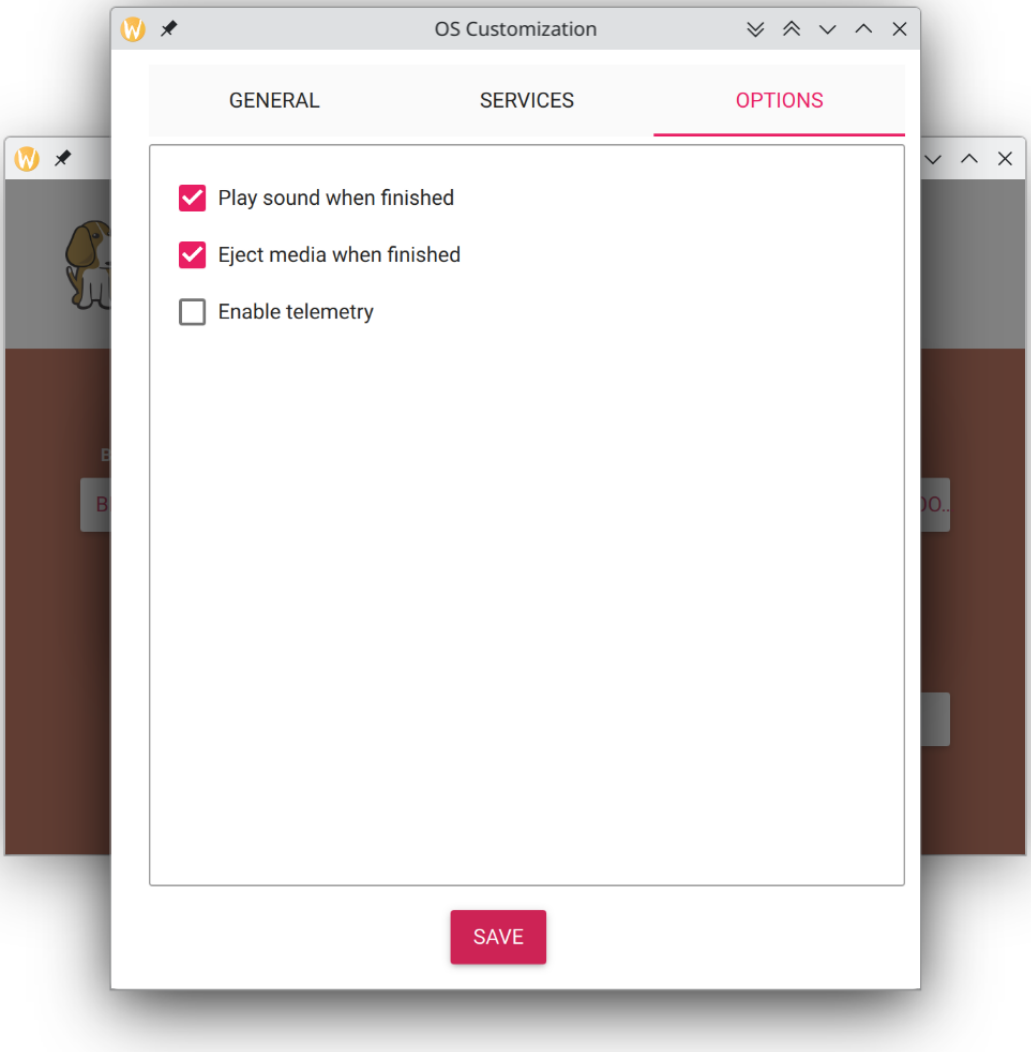


Fig. 2.11: Under OPTIONS you can enable to play sound when flashing is finished

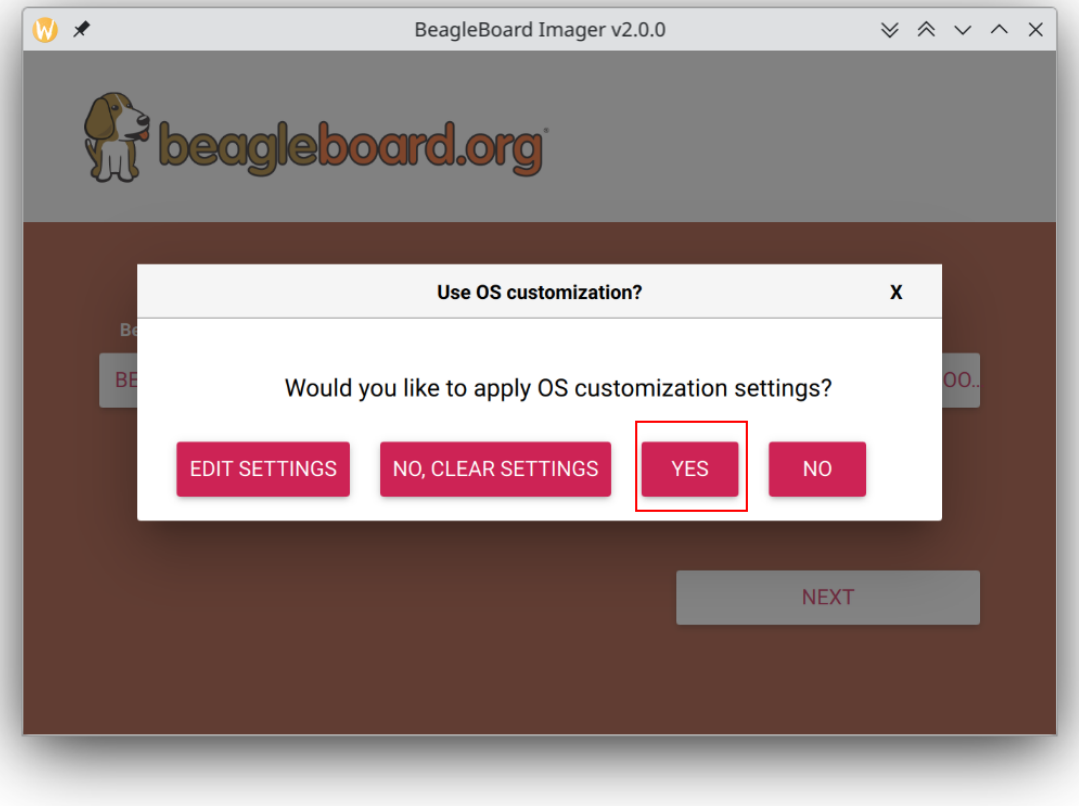


Fig. 2.12: Select YES to apply settings

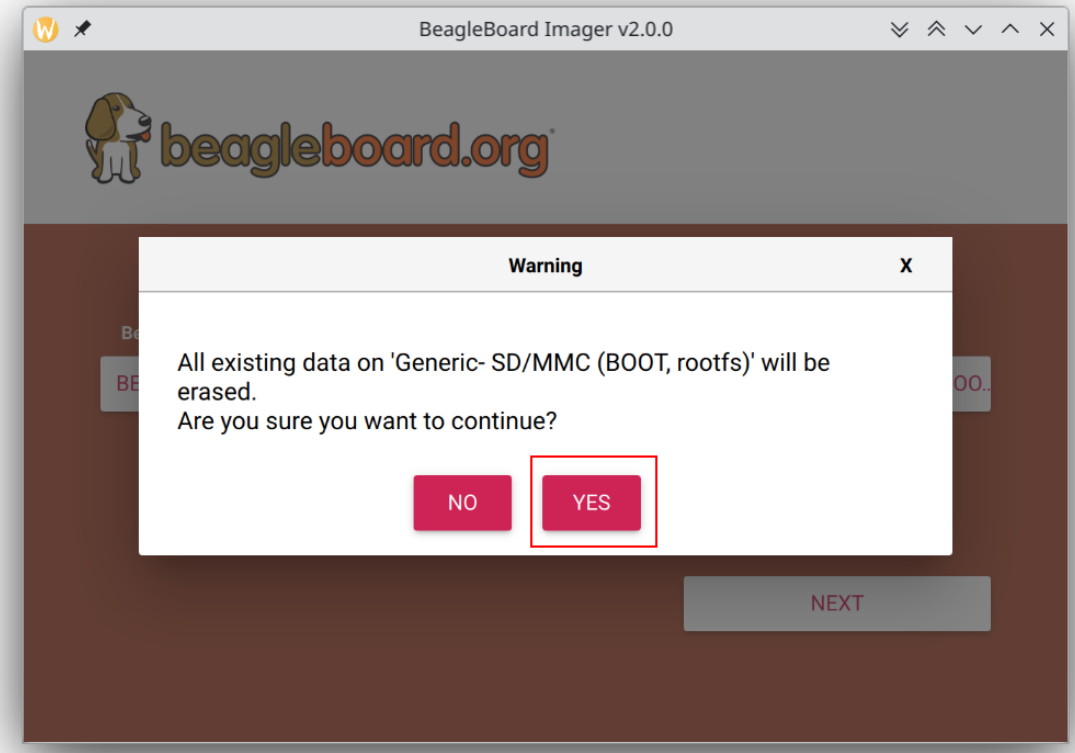


Fig. 2.13: Select YES again to confirm sdCard formatting

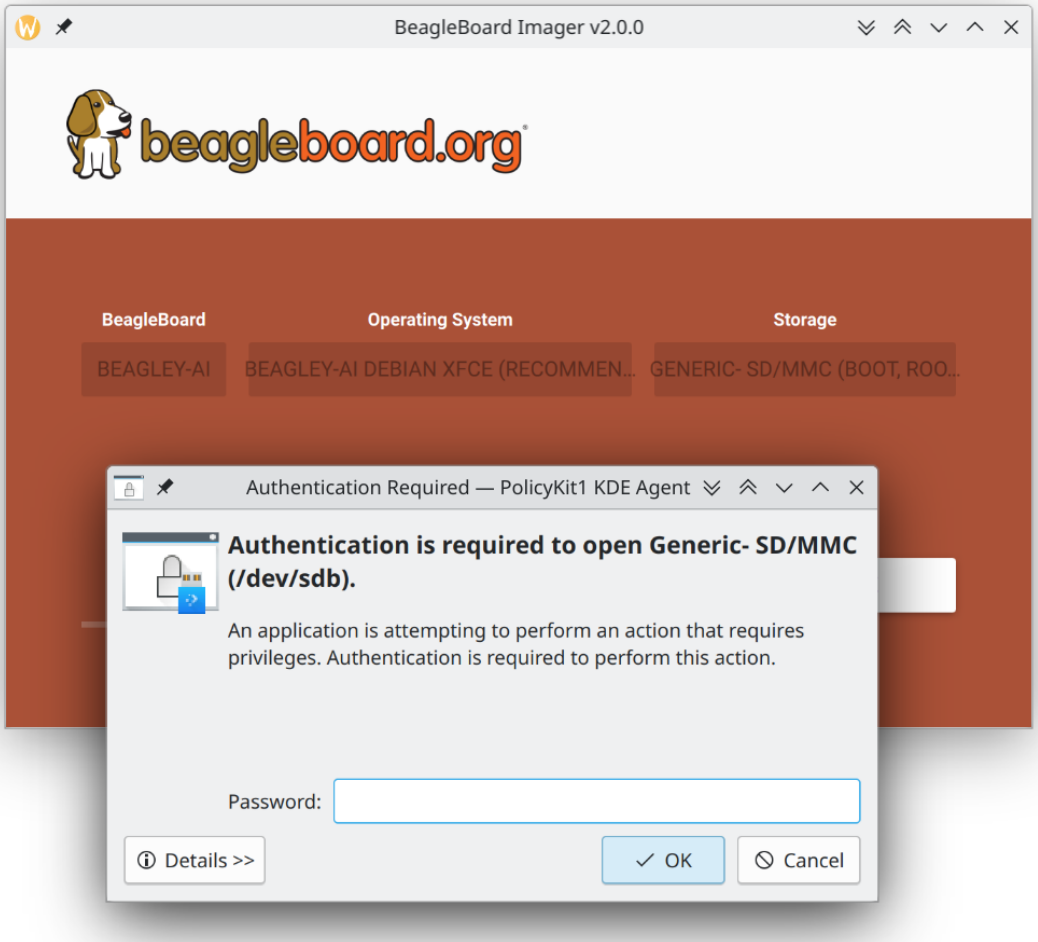


Fig. 2.14: Provide password to Authenticate the flashing process

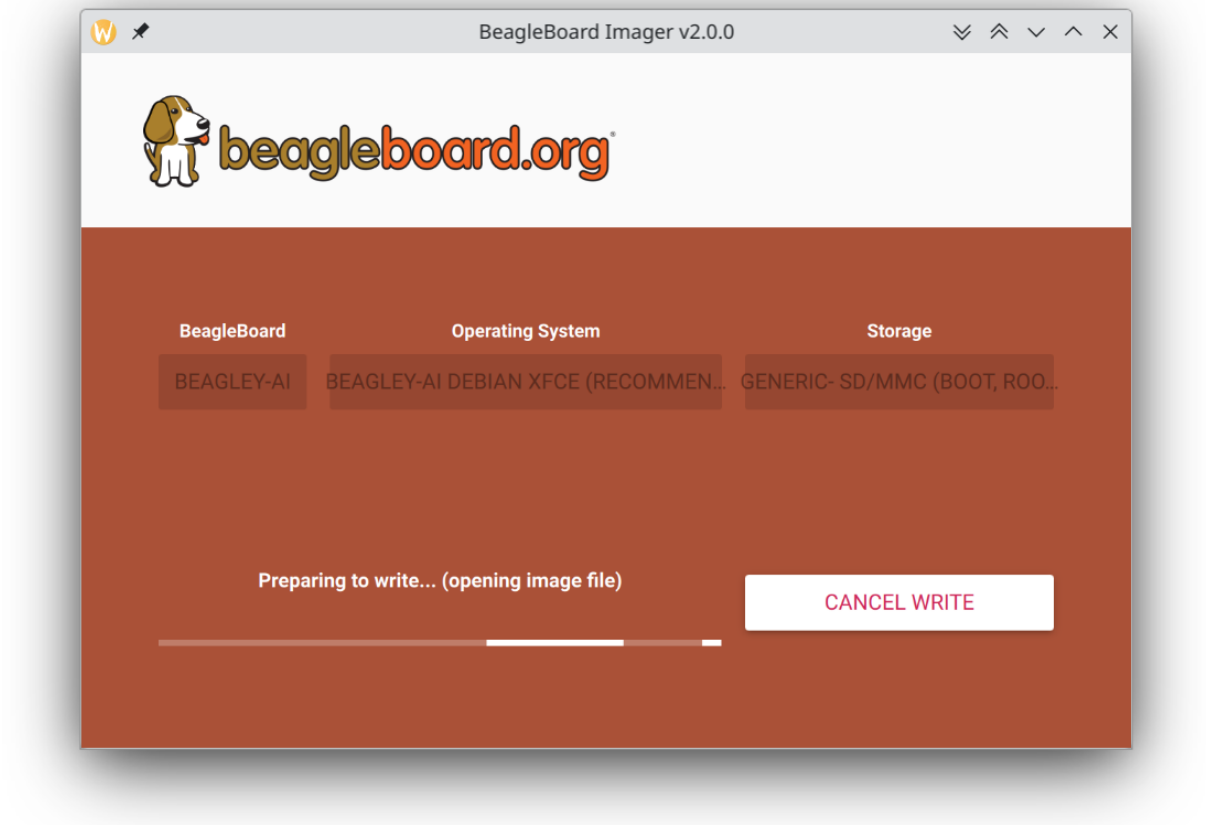


Fig. 2.15: Download image else automatically open cached image

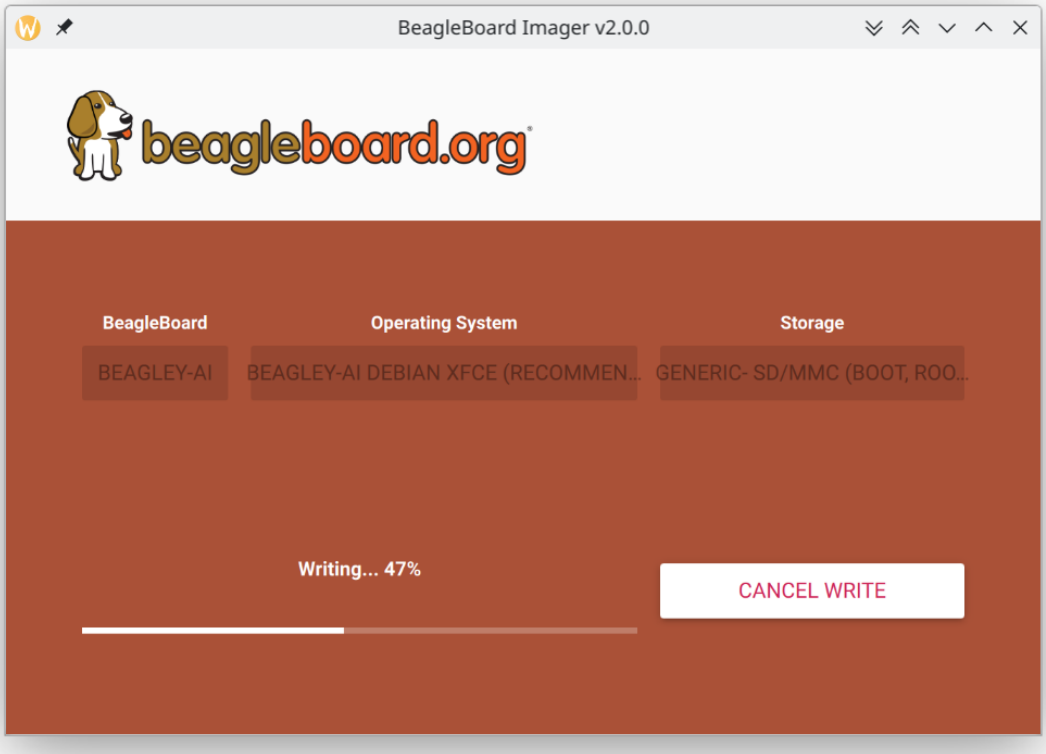


Fig. 2.16: Writing data to microSD card

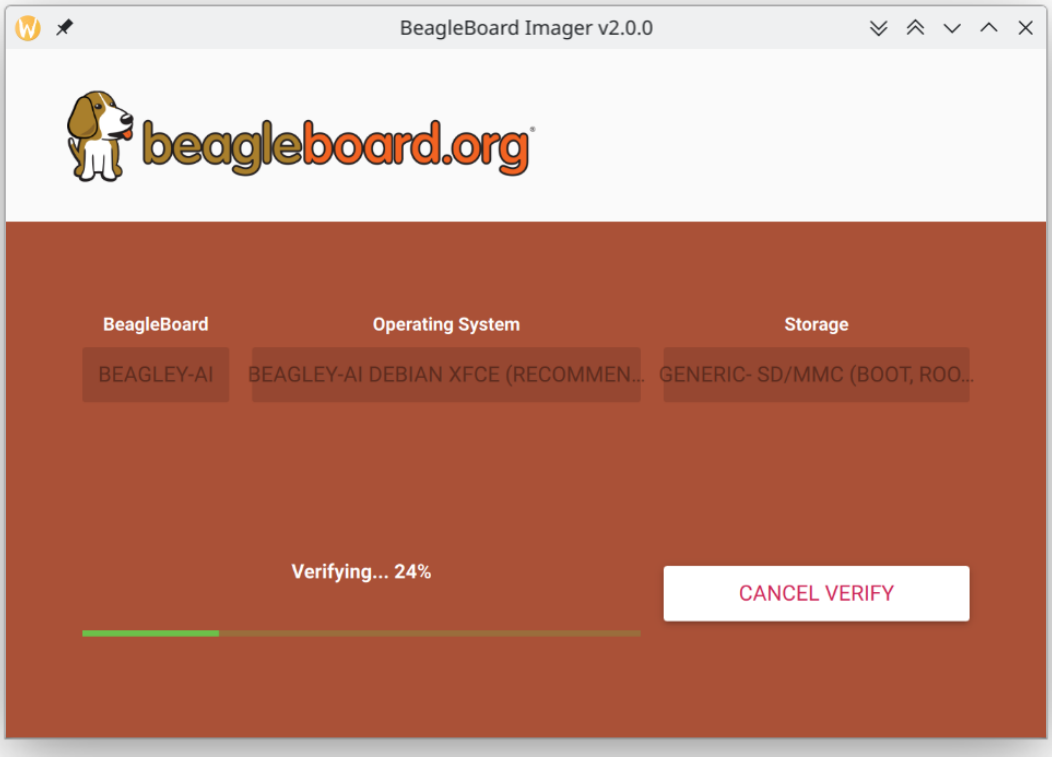


Fig. 2.17: Verifying flashed microSD card

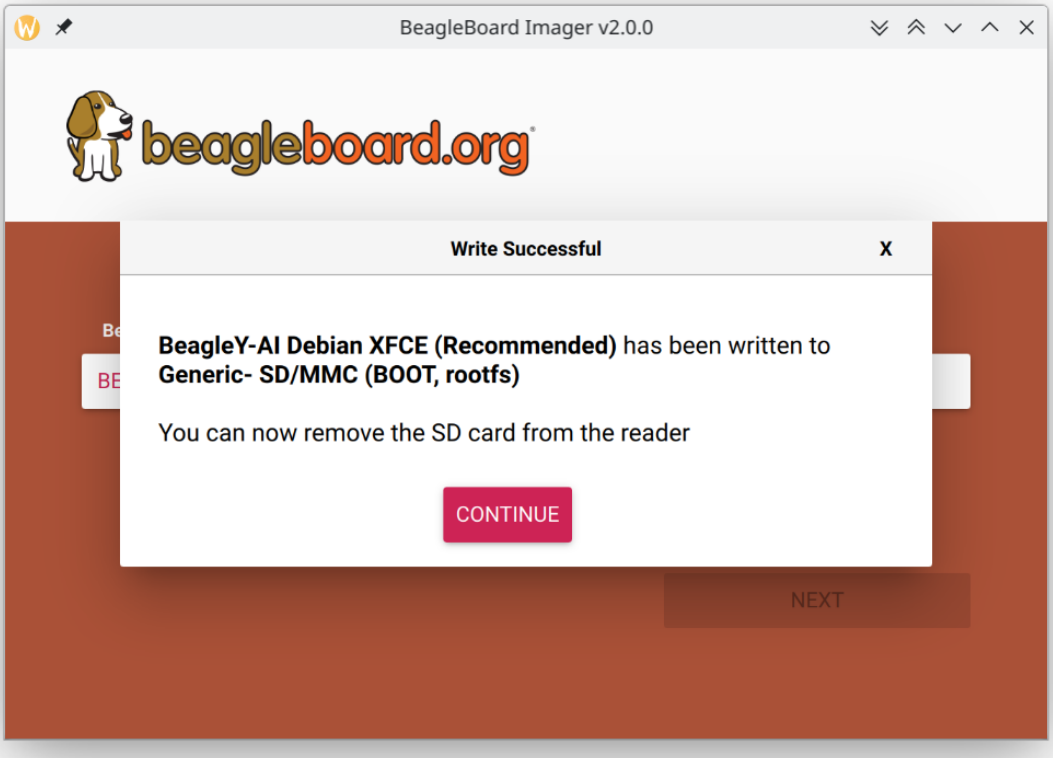


Fig. 2.18: microSD card is ready

Tip: For more detailed steps checkout the beagleboard-getting-started under support section of the documentation.

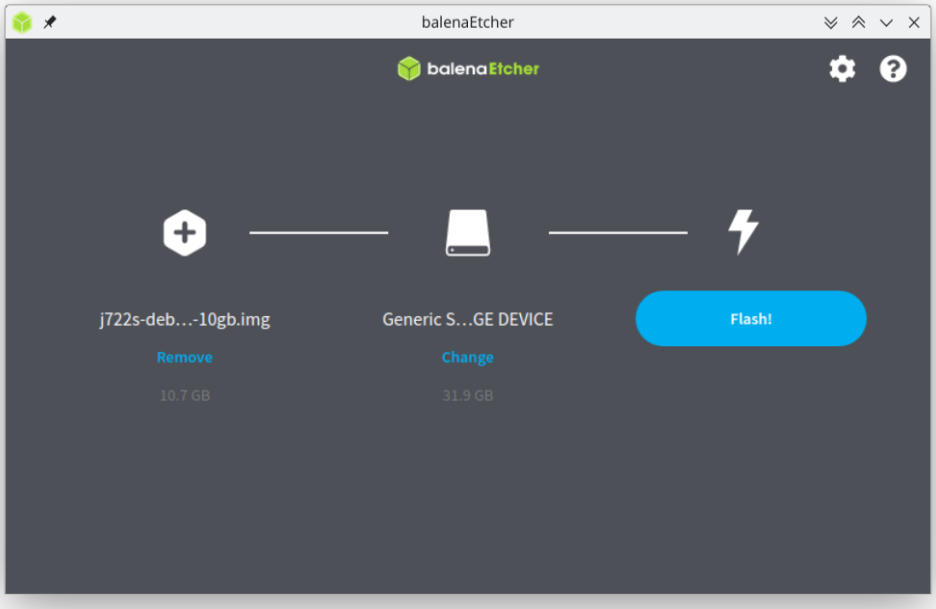


Fig. 2.19: Flashing BeagleY-AI boot image (software image) to microSD card

Once the microSD card is flashed you should see `BOOT` and `rootfs` mounted on your system as shown in image below,

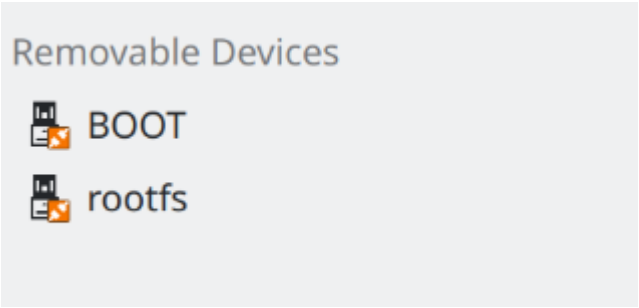


Fig. 2.20: Flashed microSD card mounted partitions

Under `BOOT` partition open `sysconf.txt` to edit login `username` and `password`.

In `sysconf.txt` file you have to edit the two lines highlighted below.

```
# user_name - Set a user name for the user (1000)
#user_name=beagle ①

# user_password - Set a password for user (1000)
#user_password=FooBar ②
```

① If `boris` is your username, update `#user_name=beagle` to `user_name=boris`

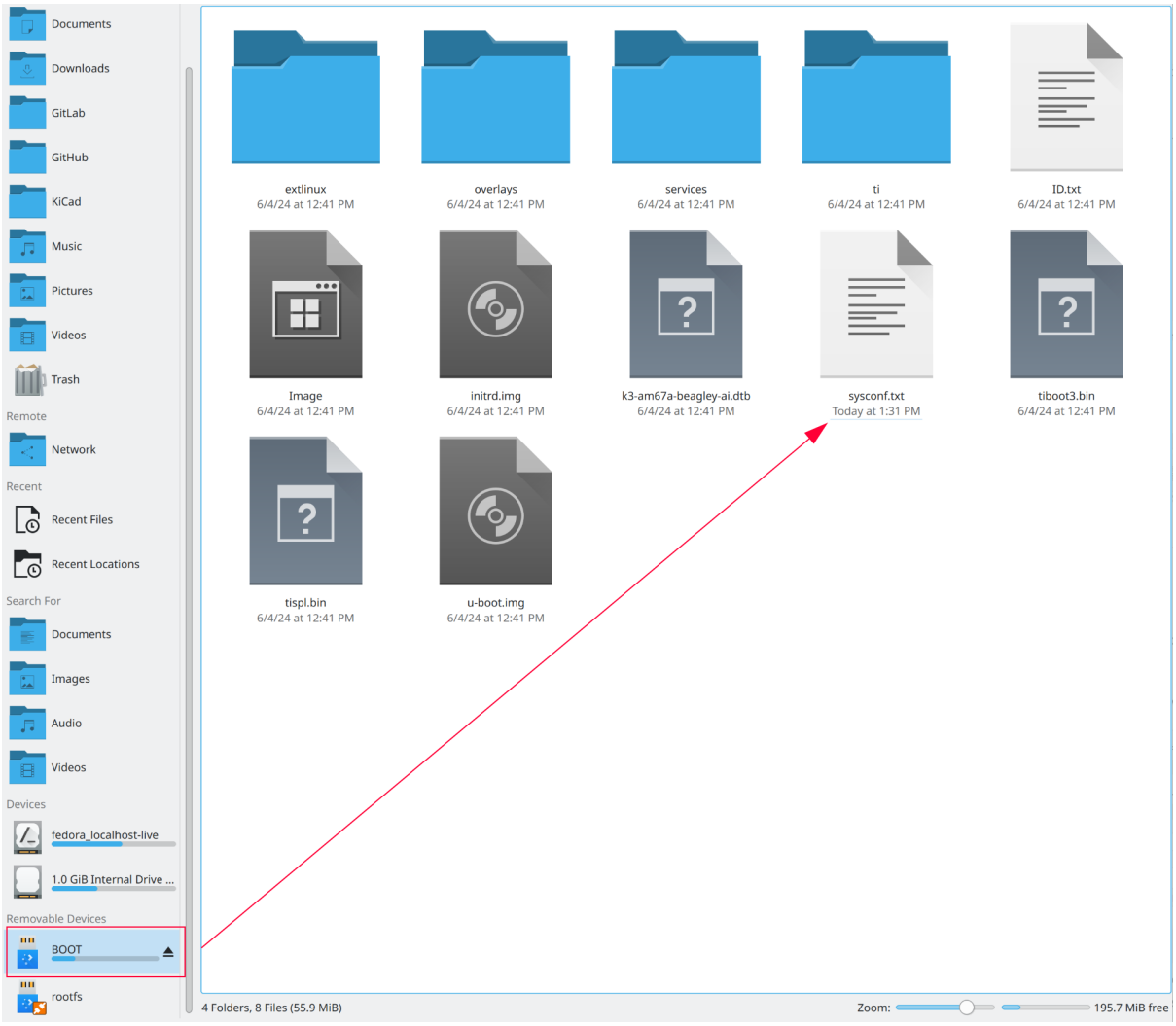


Fig. 2.21: sysconf file under BOOT partition

② If `bash` is your password, update `#user_password=FooBar` to `user_password=bash`

Important:

1. Make sure to remove `#` from `#user_name=` and `#user_password=` else the lines will be interpreted as a comment and your username & password will not be updated.
2. If you do not change your username and passord here then you will not see any output on your HDMI monitor when you do a *Standalone connection* setup.

Once username and password are updated, you can insert the microSD card into your BeagleY-AI as shown in the image below:

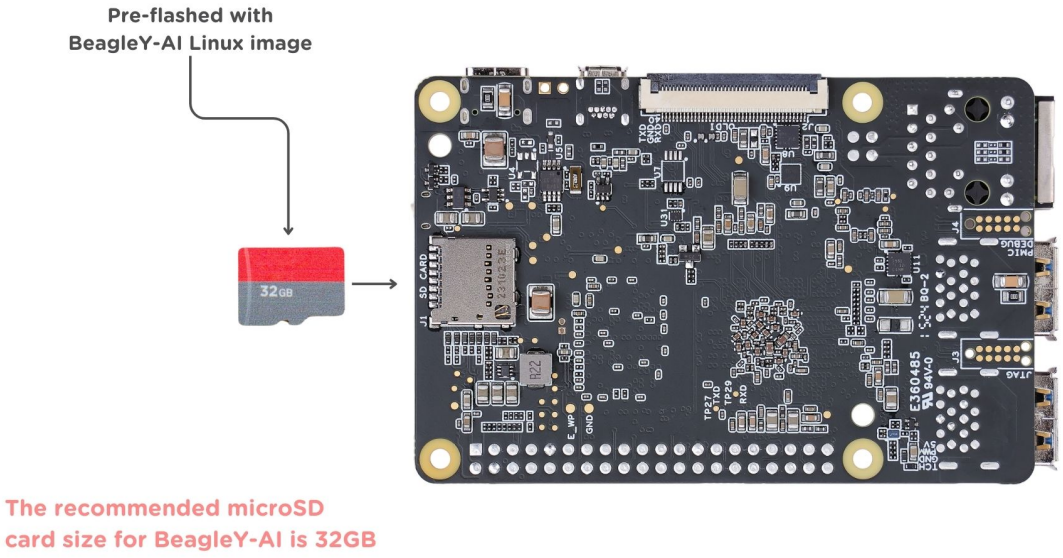


Fig. 2.22: Insert microSD card in BeagleY-AI

2.5 USB Tethering

Note: If you are using the board with a fan or running a computationally intensive task you should always power the board with a dedicated power supply that can supply 5V ≥ 3A (15W+).

As per USB standards these are the current at 5V that each downstream USB port type can (max) supply:

- USB Type-A 3.x port - 900mA (4.5W)
- USB Type-C 1.2 port - 1500mA (7.5W) to 3000mA (15W)

Thus it's recommended to use type-C to type-C cable.

To initially test your board, you can connect the board directly to your computer using a `type-C to type-C` cable shown in the image below.

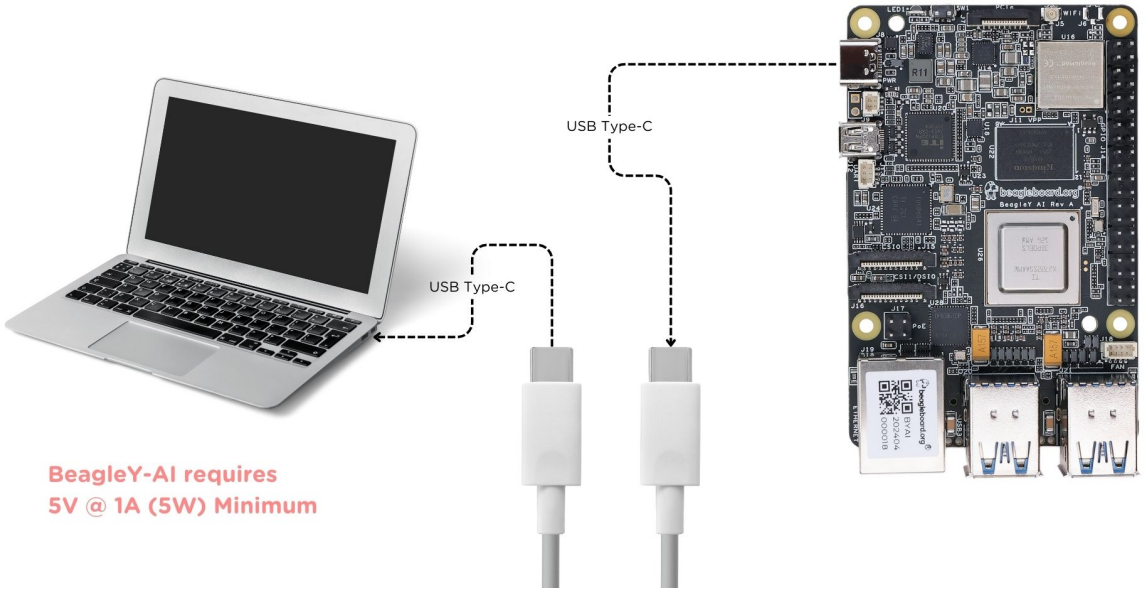


Fig. 2.23: BeagleY-AI tethered connection

2.5.1 SSH connection

After connecting, you should see the power LED glow, and soon just like with other Beagles, BeagleY-AI will create a virtual wired connection on your computer. To access the board, open up a terminal (Linux/Mac) or command prompt (Windows) and use the SSH command as shown below.

```
ssh debian@192.168.7.2
```

Important: Here `debian` is the default username, make sure to replace `debian` with the `username` you selected during *Boot Media (Software image)* preparation step.

Tip: If you are not able to find your beagle at `192.168.7.2`, checkout start-browse-to-beagle to resolve your connection issue.

Important: If you have not updated your default username and password during *Boot Media (Software image)*, you must update the default password at this step to something safer.

2.5.2 UART connection

Your BeagleY-AI board creates a UART connection (No additional hardware required) when tethered to a Laptop/PC which you can access using `Putty` of `tio`. On a linux machine it may come up as `dev/ttyACM*`, it will be different for Mac and Windows operating systems. To find serial port for your system you can checkout this guide.

- If you are on linux, try `tio` with default setting using command below,

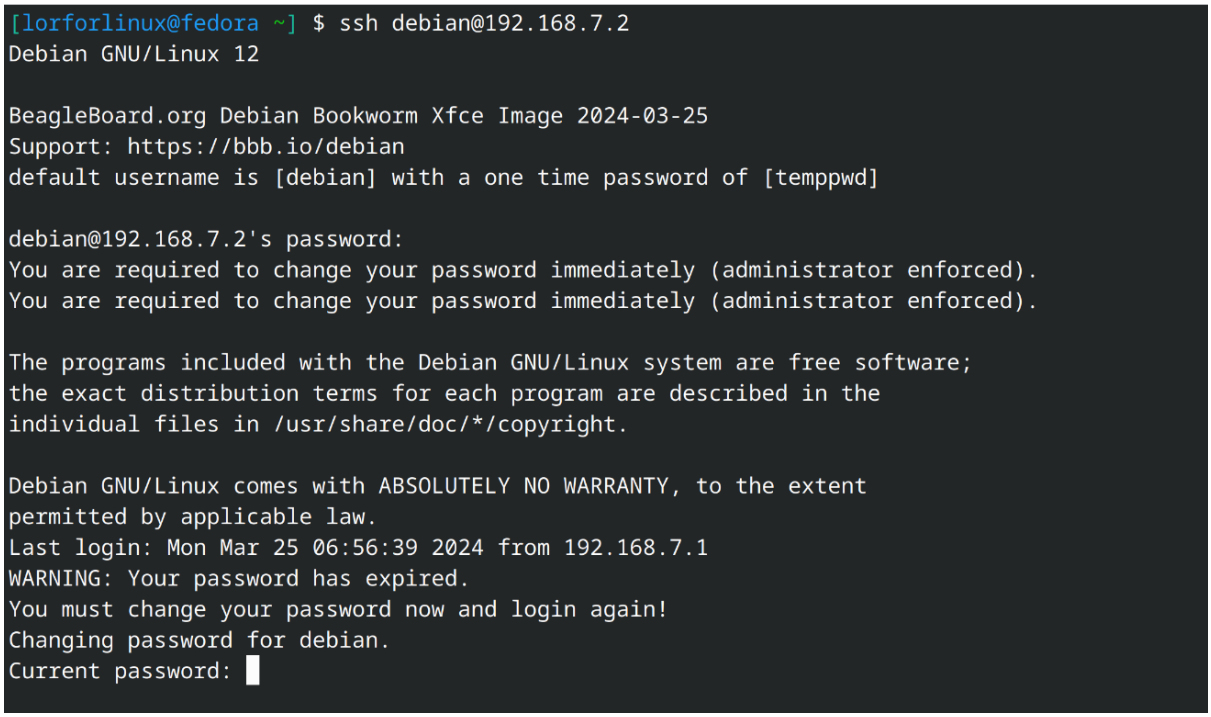


Fig. 2.24: BeagleY-AI SSH connection

```
tio /dev/ttyACM0
```

With this you have the access to BeagleY-AI terminal. Now, you can connect your board to *WiFi*, try out all the *cool demos* and explore all the other ways to access your BeagleY-AI listed below.

- *Connecting to WiFi*
- *Demos and tutorials*

2.5.3 Headless connection

If you want to run your BeagleY-AI in headless mode, you need Raspberry Pi Debug Probe or similar serial (USB to UART) adapter. Connect your UART debug probe to BeagleY-AI as shown in the image below. After making the connection you can use command line utility like `tio` on Linux of Putty on any operating system. Check *UART connection* for more information.

2.5.4 Standalone connection

Important: Make sure to update your `username` and `password` during *Boot Media (Software image)* preparation step else you'll not see any output on you HDMI monitor.

To setup your BeagleY-AI for standalone usage, you need the following additional accessories,

1. HDMI monitor
2. micro HDMI to full-size HDMI cable
3. Wireless keyboard & mice combo
4. Ethernet cable (Optional)

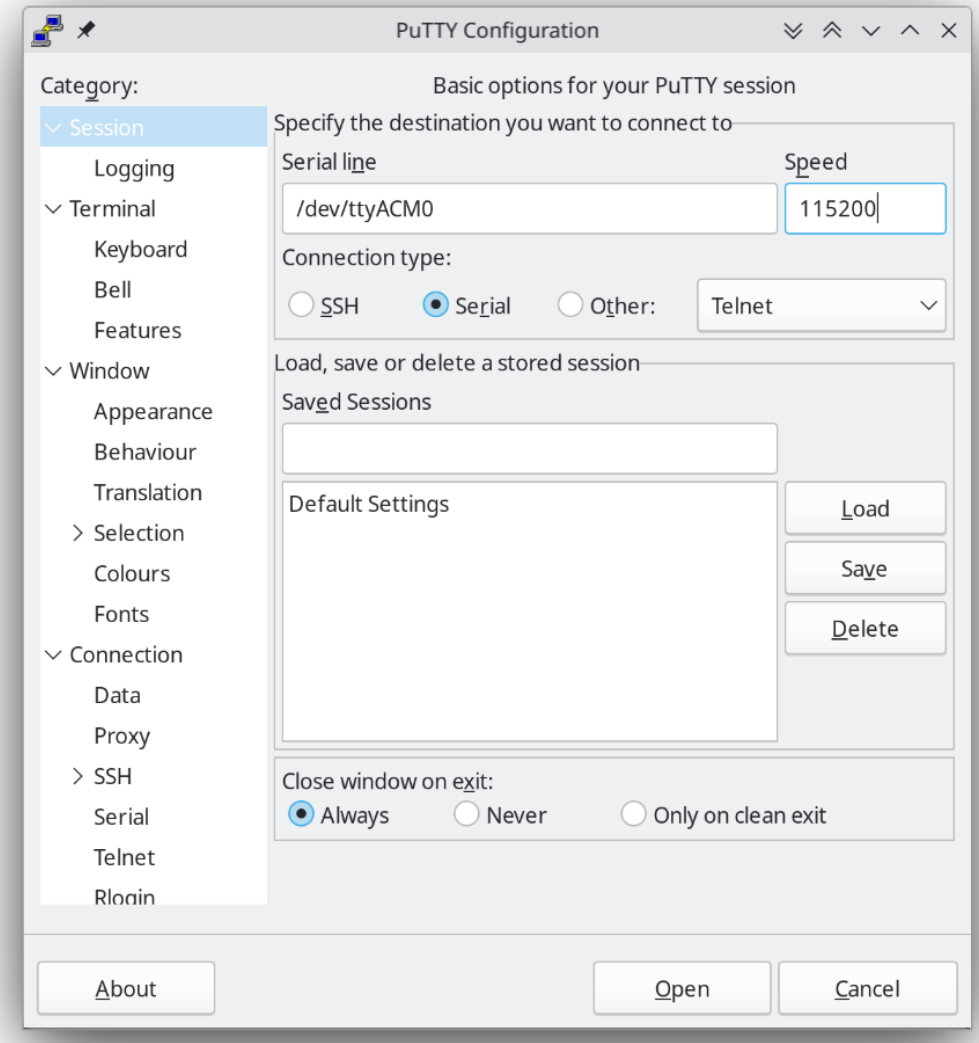


Fig. 2.25: Putty serial connection

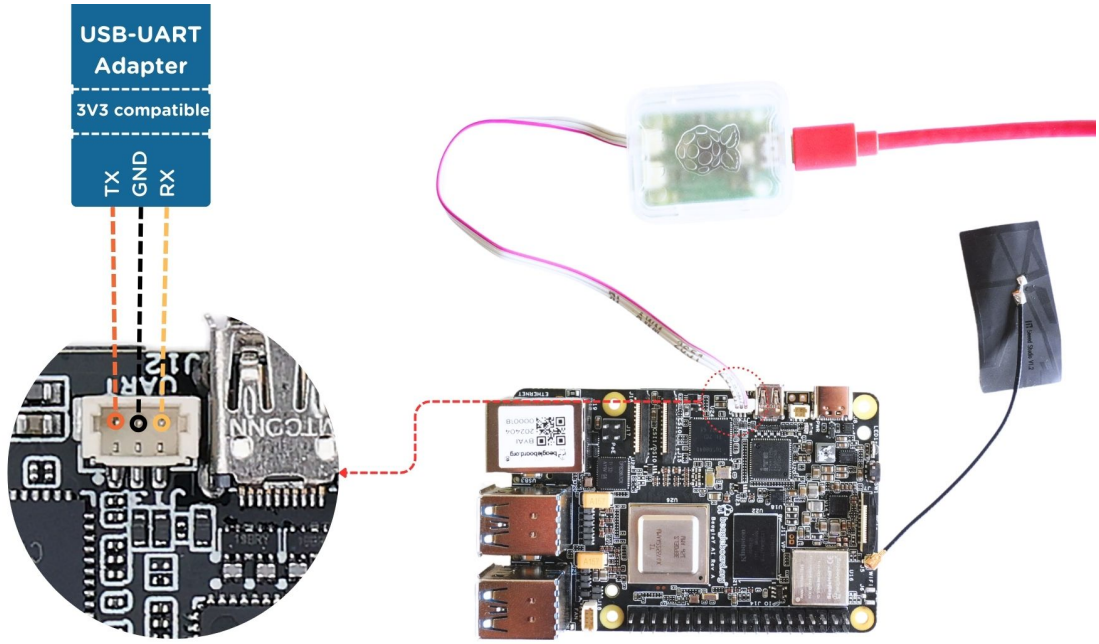


Fig. 2.26: Connecting Raspberry Pi debug probe to BeagleY-AI

Make sure you have the microSD card with boot media (software image) inserted in to the BeagleY-AI. Now connect,

1. microHDMI to BeagleY-AI and full size HDMI to monitor
2. keyboard and mice combo to one of the four USB port of BeagleY-AI
3. Power supply to USB type-c connector of BeagleY-AI

The connection diagram below provides a clear representation of all the connections,

If everything is connected properly you should see four penguins on your monitor.

When prompted, login using the credentials you updated during *Boot Media (Software image)* preparation step.

Important: You can not update login credentials at this step, you must update them during boot media (software image) micrSD card flashing or USB tethering step!

Once logged in you should see the splash screen shown in the image below:

Test network connection by running `ping 8.8.8.8`

Explore and build with your new BeagleY-AI board!

2.6 Connecting to WiFi

The onboard `BM3301` can connect to any 2.5GHz wifi access point. We have two options to connect to WiFi,

1. *nmtui*
2. *iwctl*

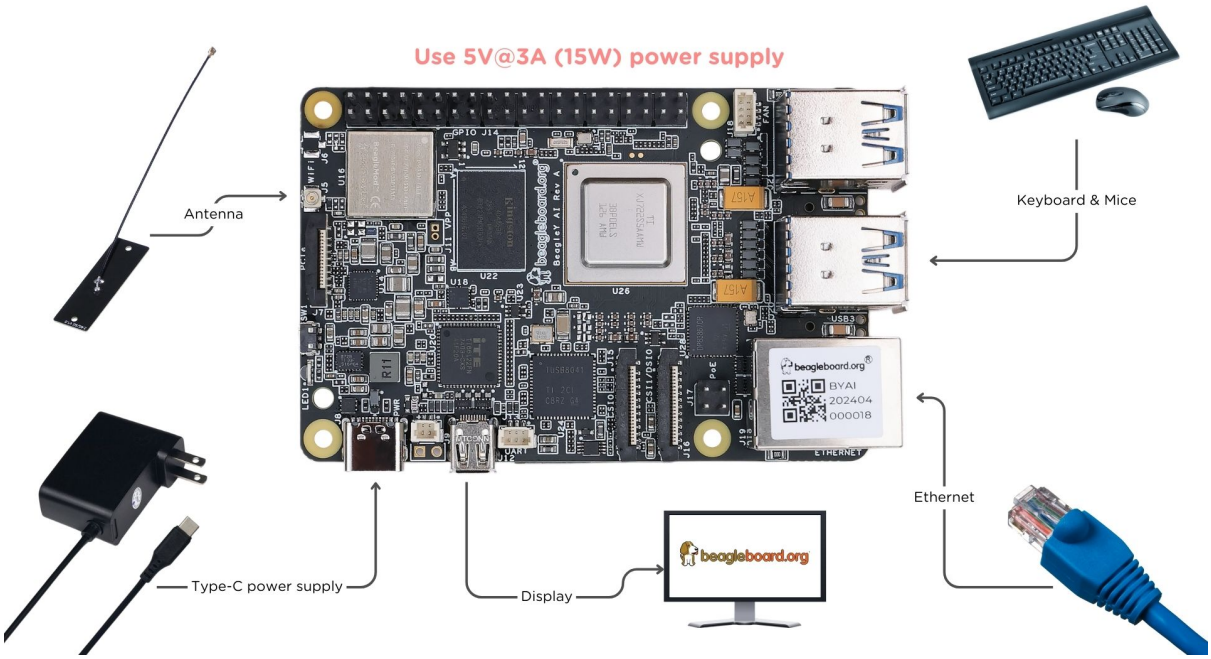


Fig. 2.27: BeagleY-AI standalone connection

Fig. 2.28: BeagleY-AI boot penguins

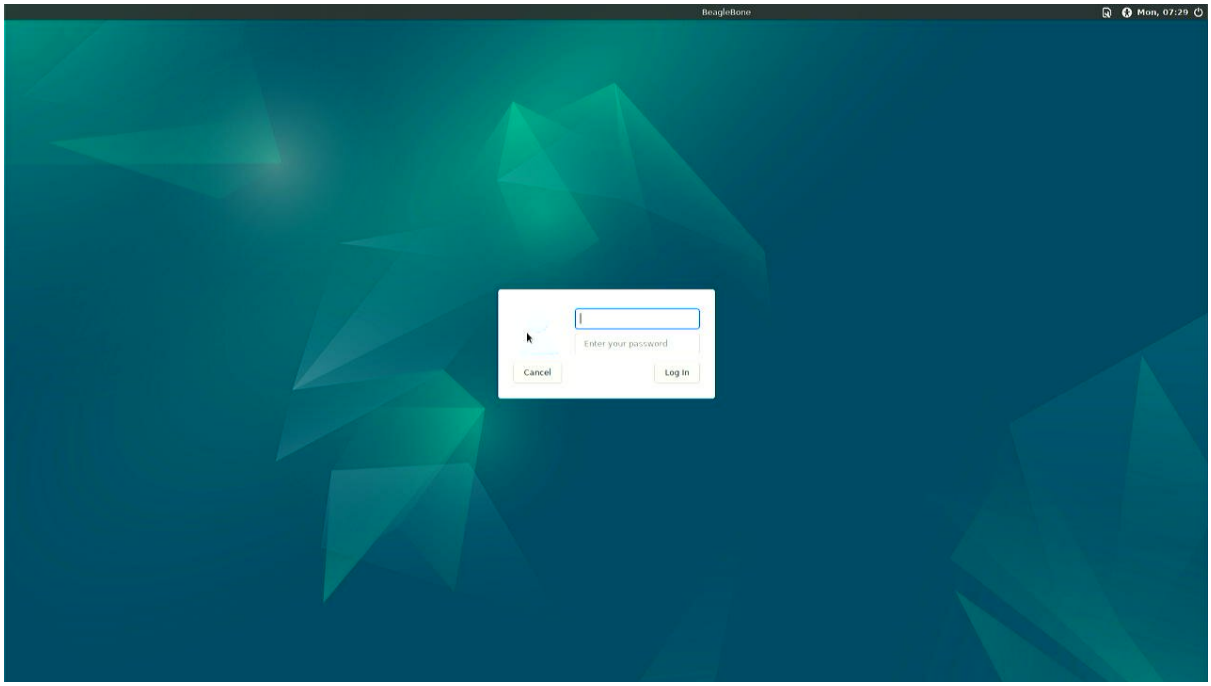


Fig. 2.29: BeagleY-AI XFCE desktop login

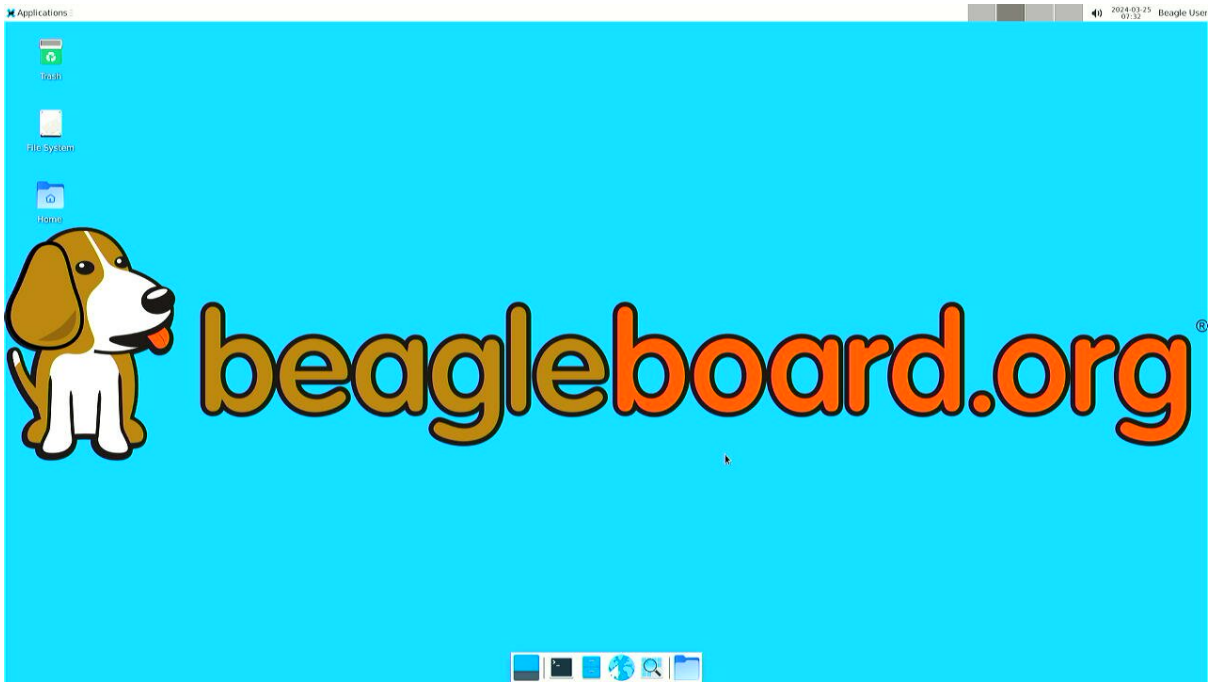


Fig. 2.30: BeagleY-AI XFCE home screen

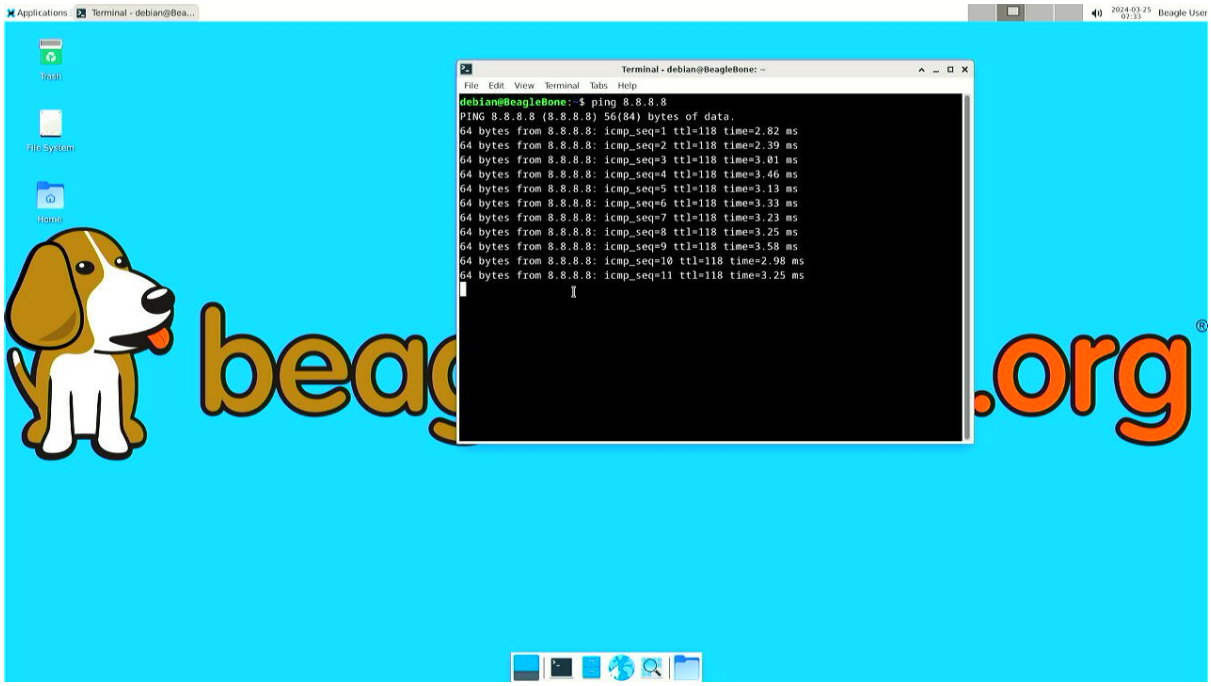


Fig. 2.31: BeagleY-AI network ping test

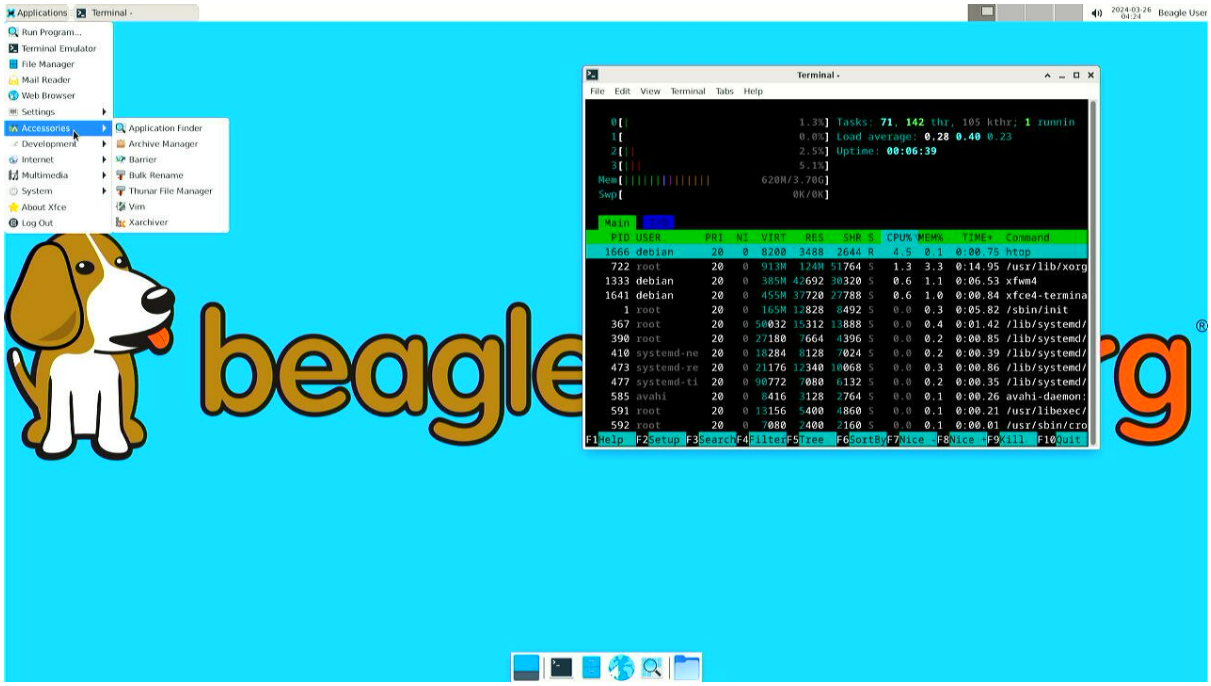


Fig. 2.32: BeagleY-AI running htop

2.6.1 nmtui

- Enable `NetworkManager`

```
sudo systemctl enable NetworkManager
```

- Start `NetworkManager`

```
sudo systemctl start NetworkManager
```

- Start `nmtui` application

```
sudo nmtui
```

- To navigate, use the `arrow keys` or press `Tab` to step forwards and press `Shift+Tab` to step back through the options. Press `Enter` to select an option. The `Space bar` toggles the status of a check box.
- You should see a screen as shown below, here you have to press `Enter` on `Acticate a connection` option to activate wired and wireless connection options.

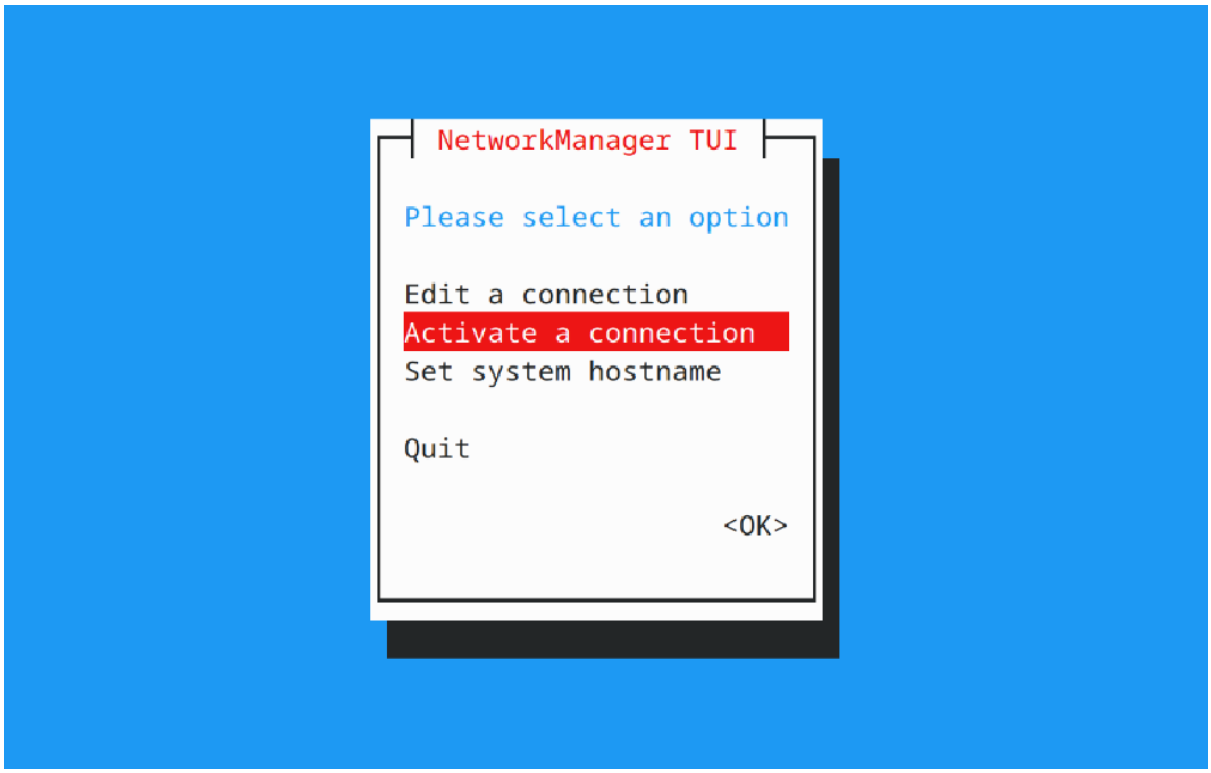


Fig. 2.33: NetworkManager TUI

There under `WiFi` section press `Enter` on desired access point and provide password to connect. When successfully connected press `Esc` to get out of the `nmtui` application window.

2.6.2 iwctl

Once board is fully booted and you have access to the shell, follow the commands below to connect to any WiFi access point,

- To list the wireless devices attached, (you should see wlan0 listed)

```
iwctl device list
```

- Scan WiFi using,

```
iwctl station wlan0 scan
```

- Get networks using,

```
iwctl station wlan0 get-networks
```

- Connect to your wifi network using,

```
iwctl --passphrase "<wifi-pass>" station wlan0 connect "<wifi-name>"
```

- Check wlan0 status with,

```
iwctl station wlan0 show
```

- To list the networks with connected WiFi marked you can again use,

```
iwctl station wlan0 get-networks
```

- Test connection with ping command,

```
ping 8.8.8.8
```

2.7 Attach cooling fan

To attached the Raspberry Pi cooling fan to BeagleY-AI you have to follow these steps,

1. Clean the surface of BeagleY-AI with a microfiber cloth or electronics safe cleaning brush.
2. Gently pull the pre-cut (blue) thermal pads from cooling fan surface and transfer them to the most heating parts of BegleY-AI like CPU and RAM.
3. Connect the fan cable, then carefully place the flat part of cooling fan on BeagleY-AI. Now, gently apply force on spring loaded push pins to securely attach the cooling fan.

2.8 Demos and Tutorials

- beagley-ai-expansion-nvme

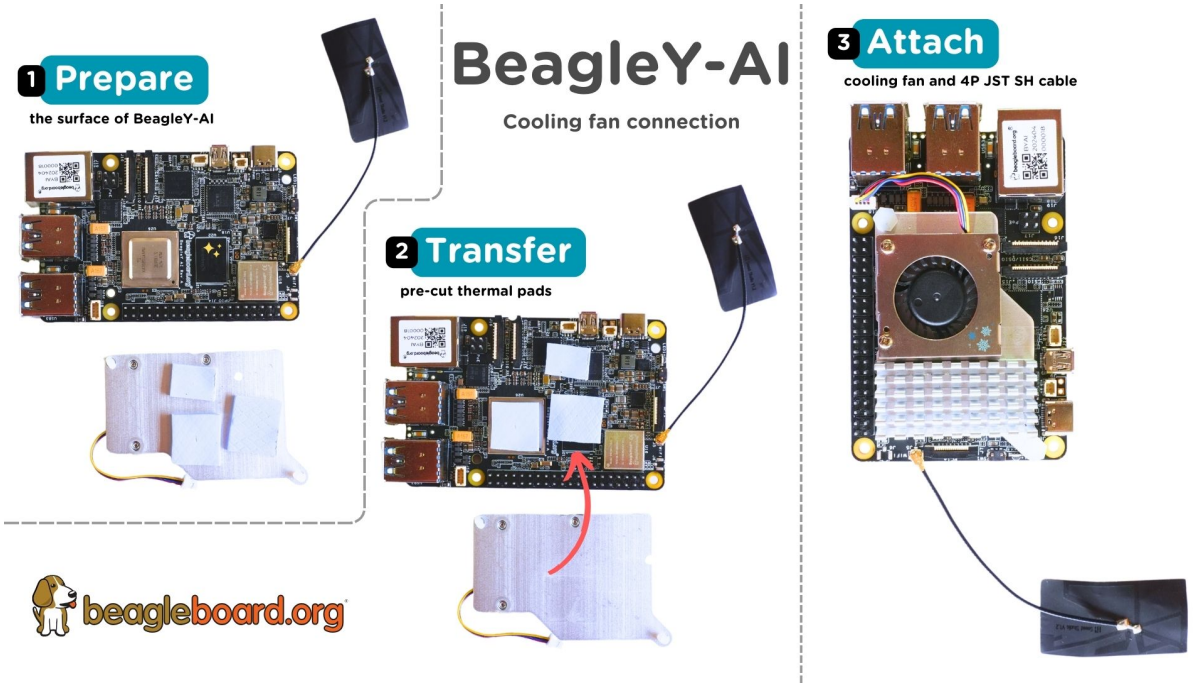


Fig. 2.34: Attaching cooling fan to BeagleY-AI

Chapter 3

Design and Specifications

Todo: Add details about all the schematic sections.

If you want to know how BeagleY-AI is designed and the detailed specifications, then this chapter is for you. We are going to attempt to provide you a short and crisp overview followed by discussing each hardware design element in detail.

Tip: For board files, 3D model, and more, you can checkout the BeagleY-AI repository on OpenBeagle.

3.1 Block Diagram and Overview

3.2 Processor

The AM67A processor from Texas Instruments is a highly integrated SoC with an Automotive pedigree. It may be referenced by TI documentation by it's superset J722s/TDA4AEN.

It's primary compute cluster revolves around 4xARM Cortex-A53 Cores running at 1.4Ghz.

An MCU subsystem consisting of an ARM Cortex-R5F running at up to 800Mhz is also available for user applications and is especially useful for real-time IO applications.

For very advanced users, two additional R5 cores are also present, but they are normally reserved for Device and Run-time Management of the SoC typically.

2x C7x DSPs with MMA support are intended for use as Deep Learning Accelerators for things like AI Vision, with up to 2TOPS each.

An Imagination BXS-4-64 GPU rounds out the compute cluster, with a dedicated video encoder/decoder available for multimedia tasks.

The SoC features advanced high speed connectivity, including USB3.1, PCIe and more.

Secure Boot is also available with the ability burn One-Time-Programmable (OTP) eFUSES by energizing the VPP test pads.

3.3 Boot Modes

The default boot mode for BeagleY-AI is the SD Card Interface. If no SD card is present, the BootROM on the AM67A SoC is going to try booting from Ethernet.

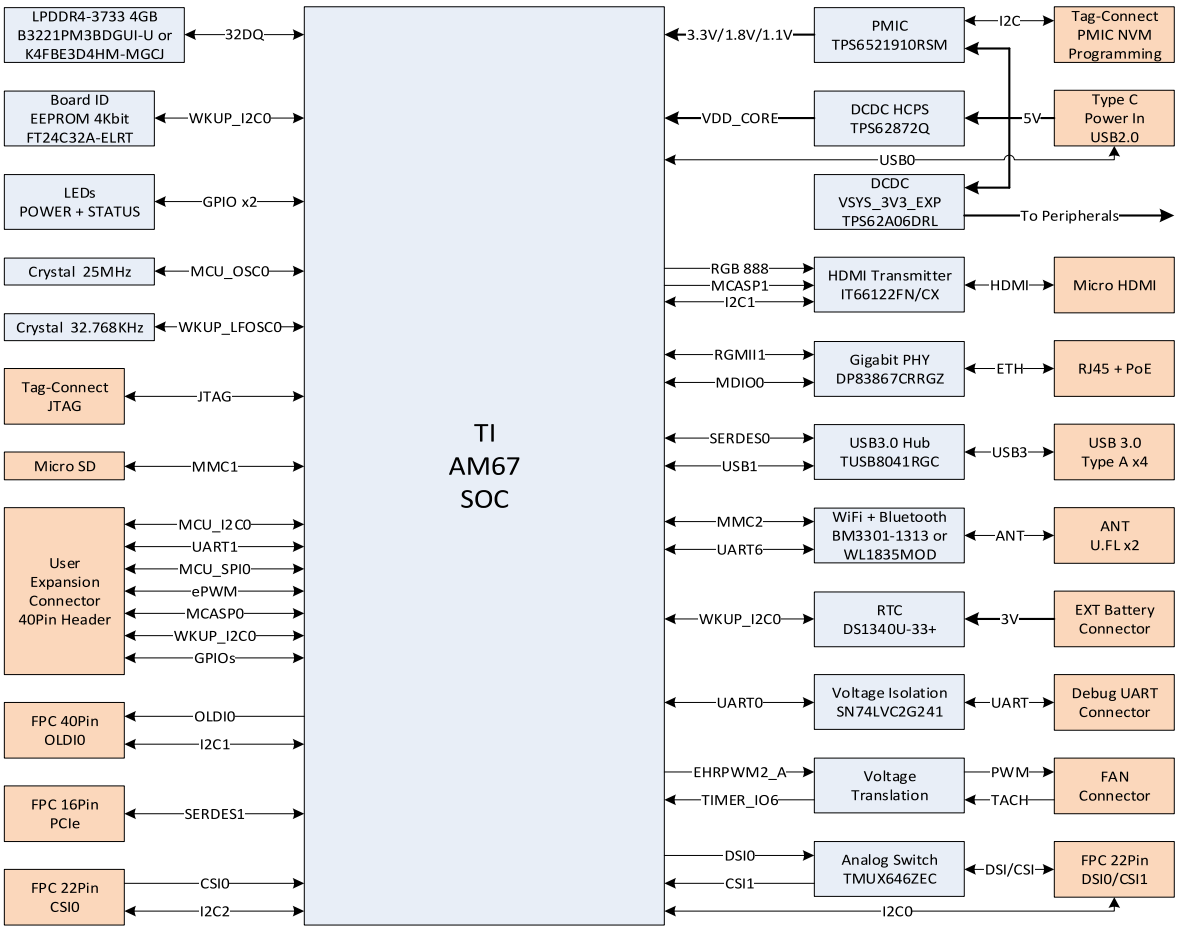


Fig. 3.1: BeagleY-AI block diagram

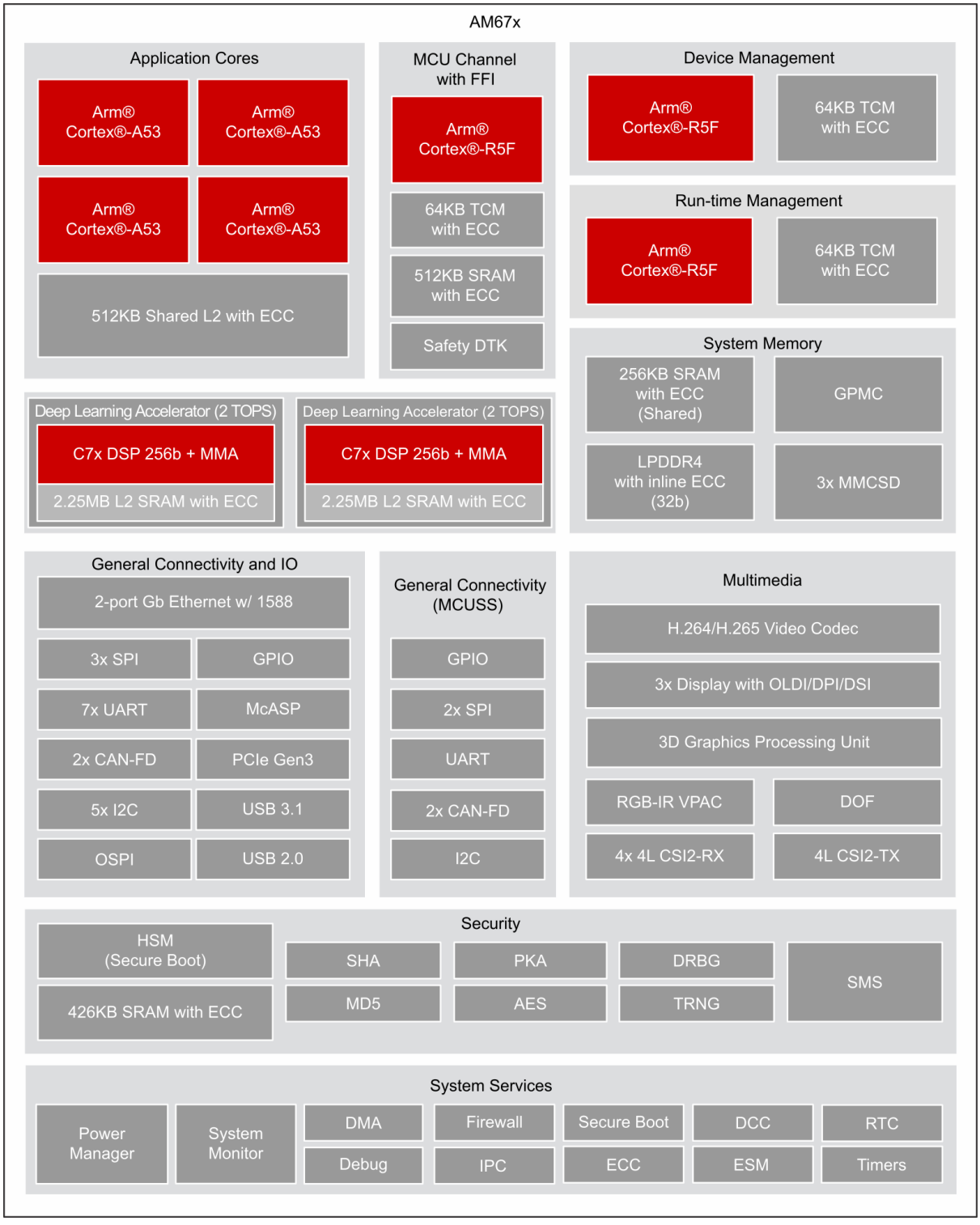


Fig. 3.2: AM67A block diagram

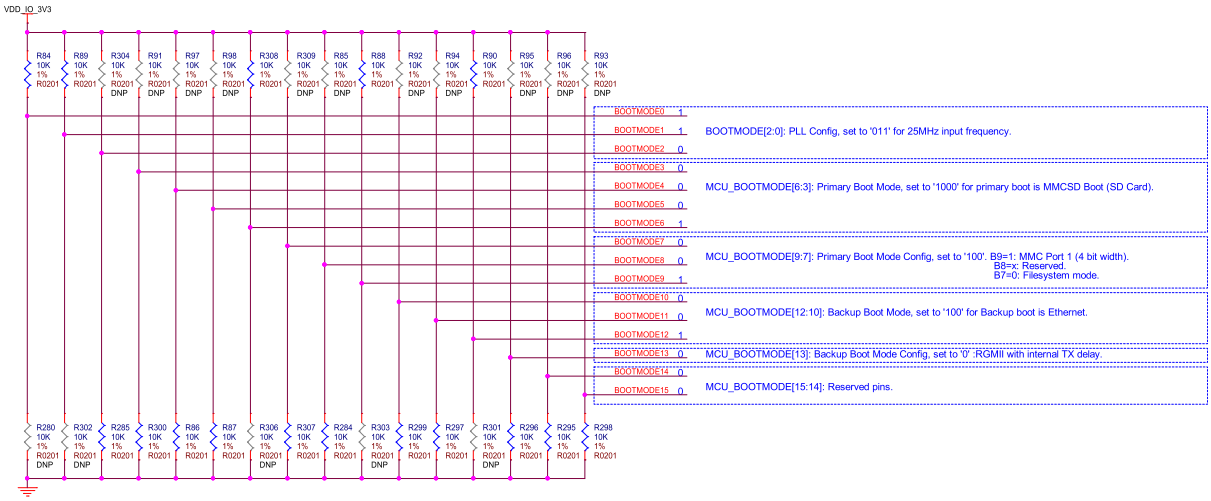


Fig. 3.3: BeagleY-AI boot modes

It is also possible to load U-Boot from the SD card and then load your main file system from another source, such as beagley-ai-expansion-nvme.

3.4 Power

BeagleY-AI's power architecture is split between the TPS65219 PMIC which handles the main logic rails and a dedicated TPS62872 high current buck regulator for the SoC core rail which defaults to 0.85V on boot.

Both PMIC and VDD_CORE regulators are highly configurable but will boot the board to "sane" defaults out of box. For advanced users, it is possible to adjust both the VDD_CORE rail as well as IO rails (voltages, timings, behavior, etc.) for applications such as low power modes where you may want to trade clock speeds for power efficiency by running the SoC Core at 0.75V for example. Be careful, as changes here could result in unexpected behavior, the board not booting or even hardware damage, so tread carefully.

Note: At the time of writing, dynamic voltage switching is not supported by the AM67A SoC.

3.5 Clocks and Resets

BeagleY's main clock source is a 25Mhz Crystal Oscillator connected to MCU_OSC0 pins.

A 32.768Khz "Slow Clock" Crystal is used on the WKUP_LFOSC0 domain.

3.5.1 USB-C Power/Data Port

The board is primarily intended to be powered via USB-C. PD Power negotiation is not done dynamically but rather by tying the CC lines to GND via 5.1KΩ resistors to indicate to the PD Source that the device requires 5V 3A. Using USB-PD power supplies rated for higher wattages is safe as they will always negotiate to the 5V 3A requested by the board.

The USB-C port is configured by default to also show up as a USB2.0 Device which exposes a serial console, ethernet gadget (for connection sharing) as well as MTP (Flash Drive) so that only one cable is required to use the board. A Type-C to Type-C cable and avoiding un-powered USB hubs is recommended due to the board's power consumption requirements. Inadequate behavior may result in brownouts/resets or other unexpected behavior.

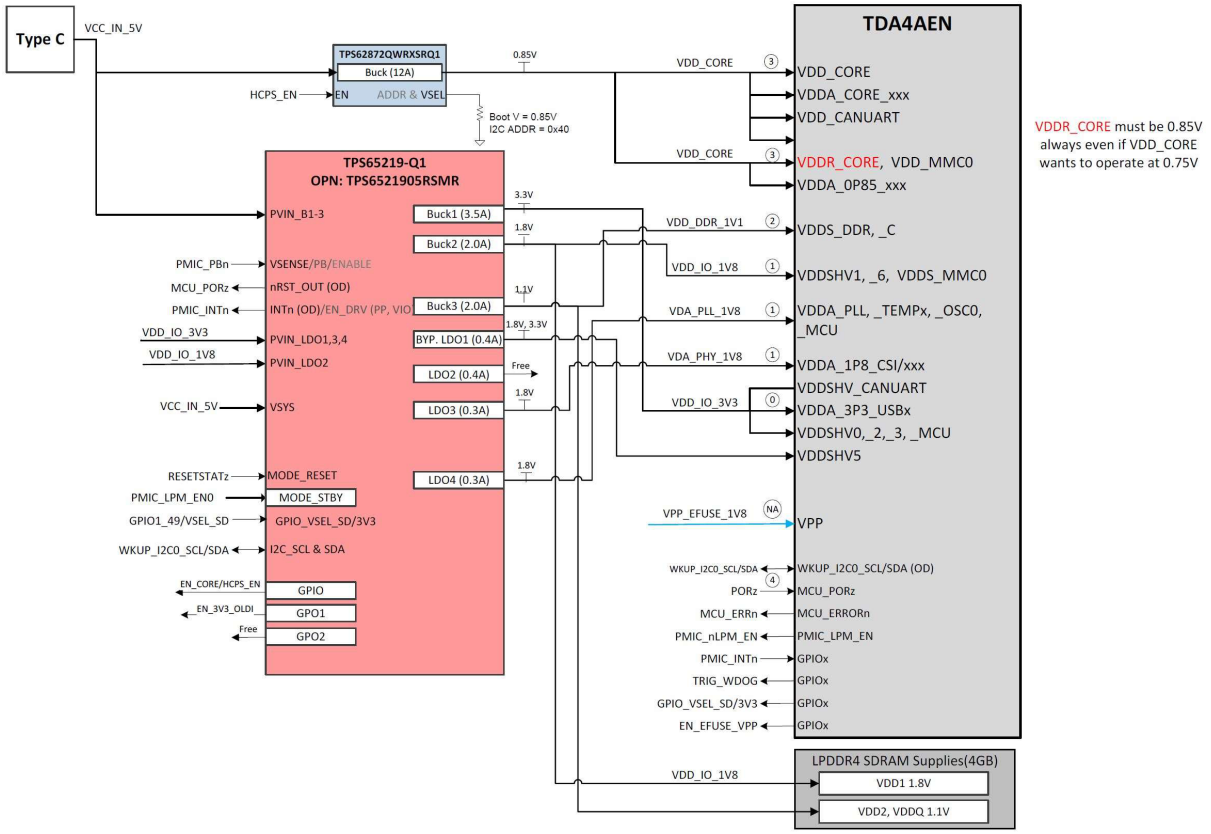


Fig. 3.4: BeagleY-AI power distribution network

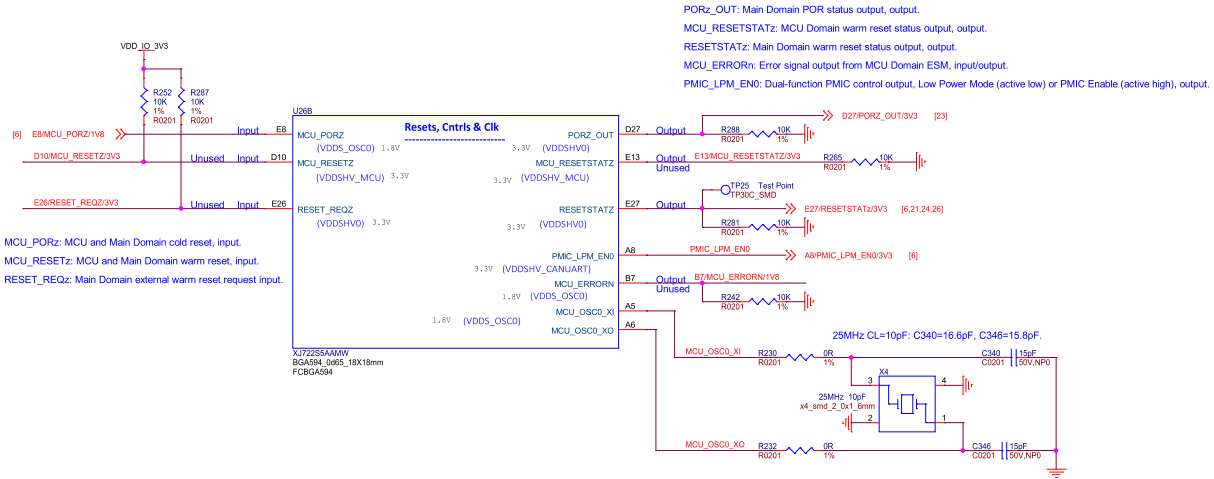


Fig. 3.5: BeagleY-AI SoC Reset, Cntrls, and Clk

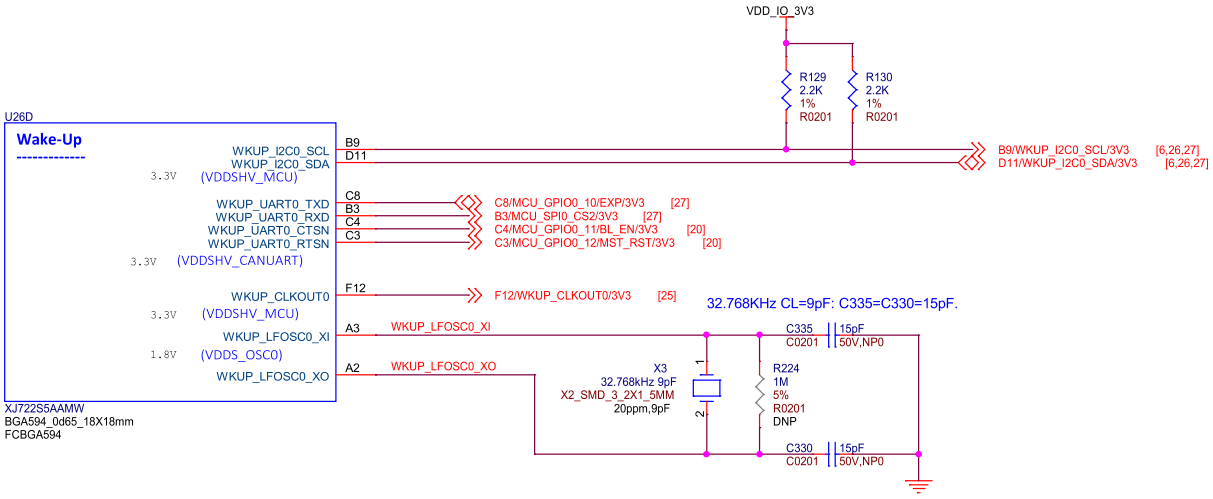


Fig. 3.6: BeagleY-AI wkup reset cntrls osc

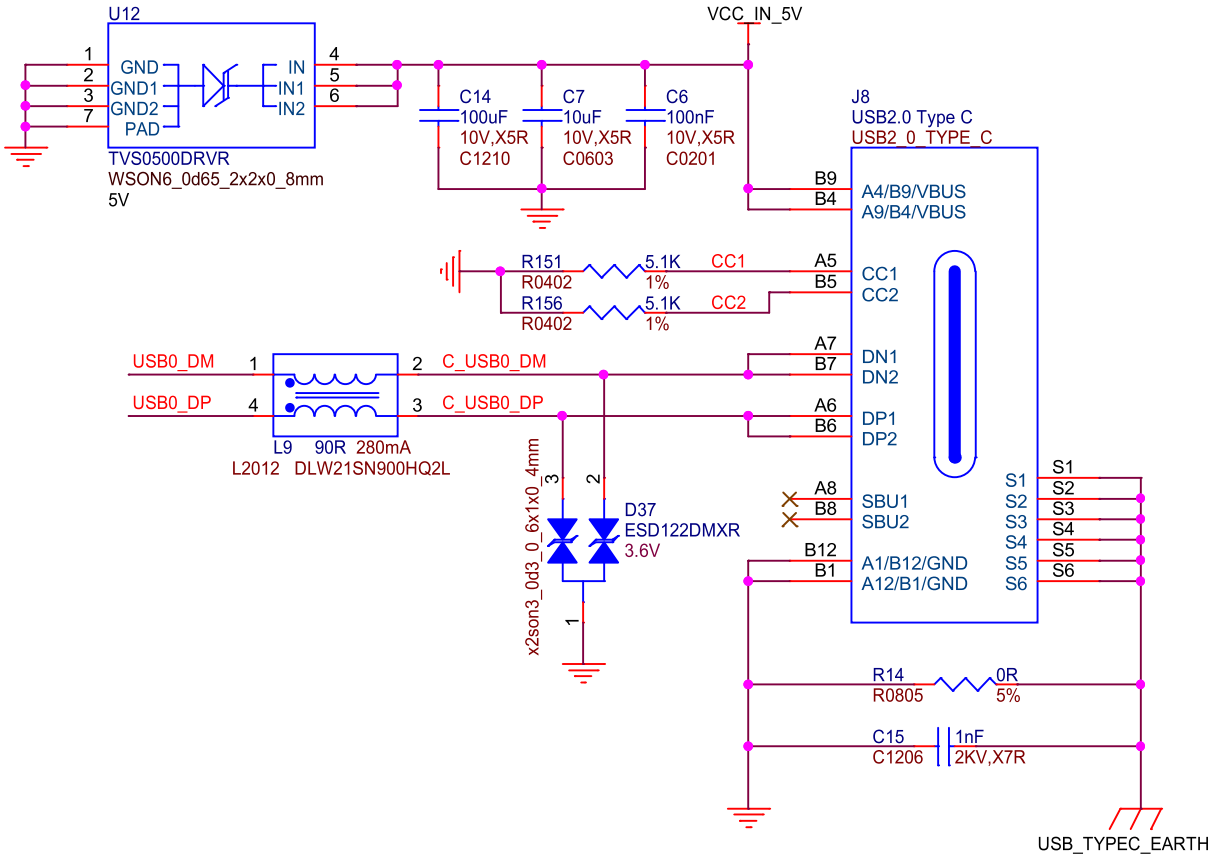


Fig. 3.7: BeagleY-AI USB-C

3.5.2 PMIC

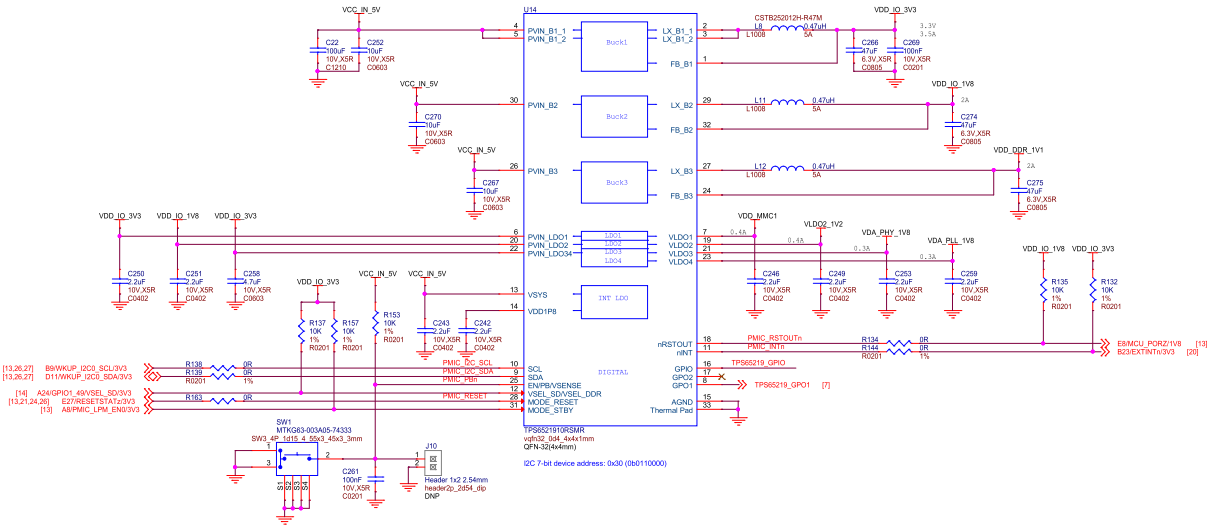


Fig. 3.8: BeagleY-AI PMIC

3.5.3 HCPS (High Current Power Stage)

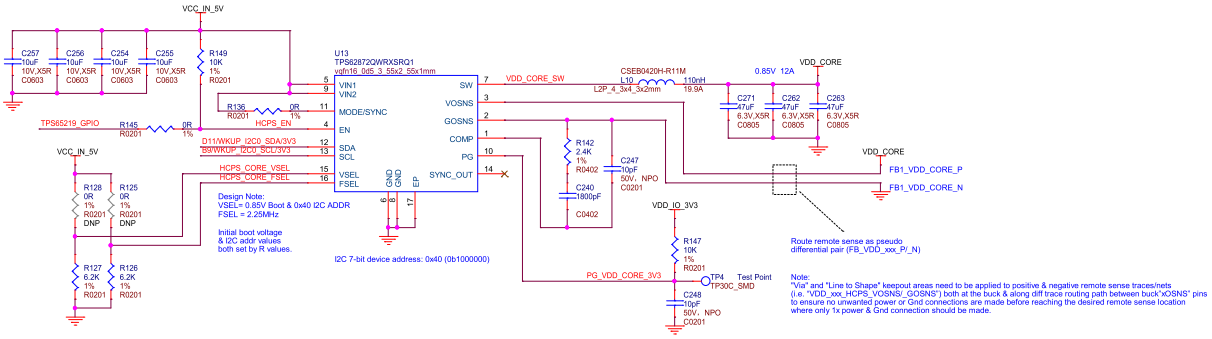


Fig. 3.9: BeagleY-AI VDD core High Current Power Stage (HCPS)

3.5.4 Analog Rail Decoupling

3.5.5 Digital Rail Decoupling

Note: Other power sections are nested within their specific interface section.

3.5.6 LDOs

While the 3.3V VDD_IO rail is provided by the PMIC, the actual “high current” VSYS 3.3V rail used on the expansion header and elsewhere in the system is provided by a discrete TPS62A06DRLR regulator.

The 2V5 Rail used by the Ethernet PHY is generated a discrete TPS74801 regulator. This regulator is fed by the 3V3 VSYS regulator.

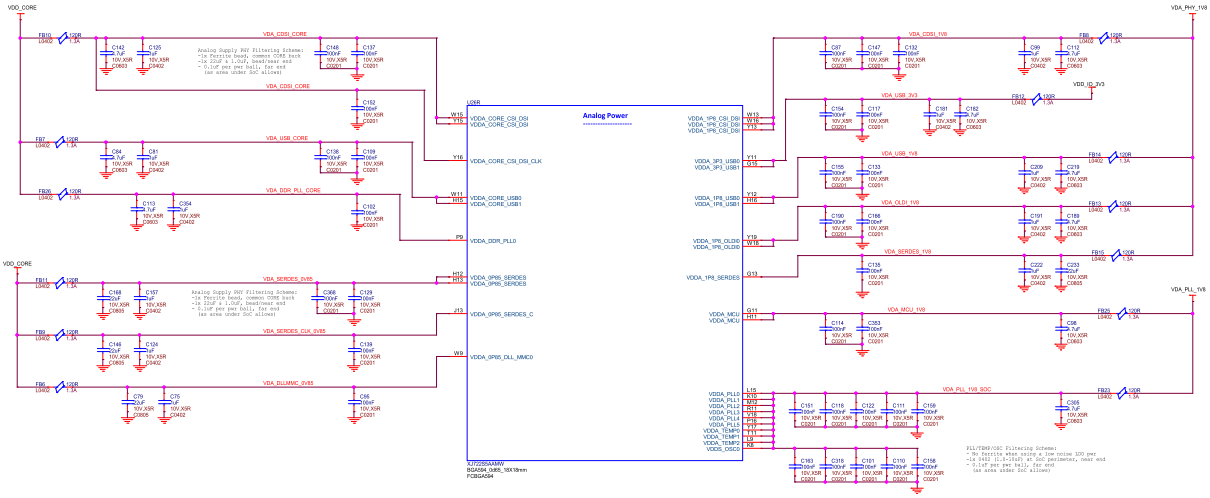


Fig. 3.10: BeagleY-AI SoC analog power rail decoupling capacitors

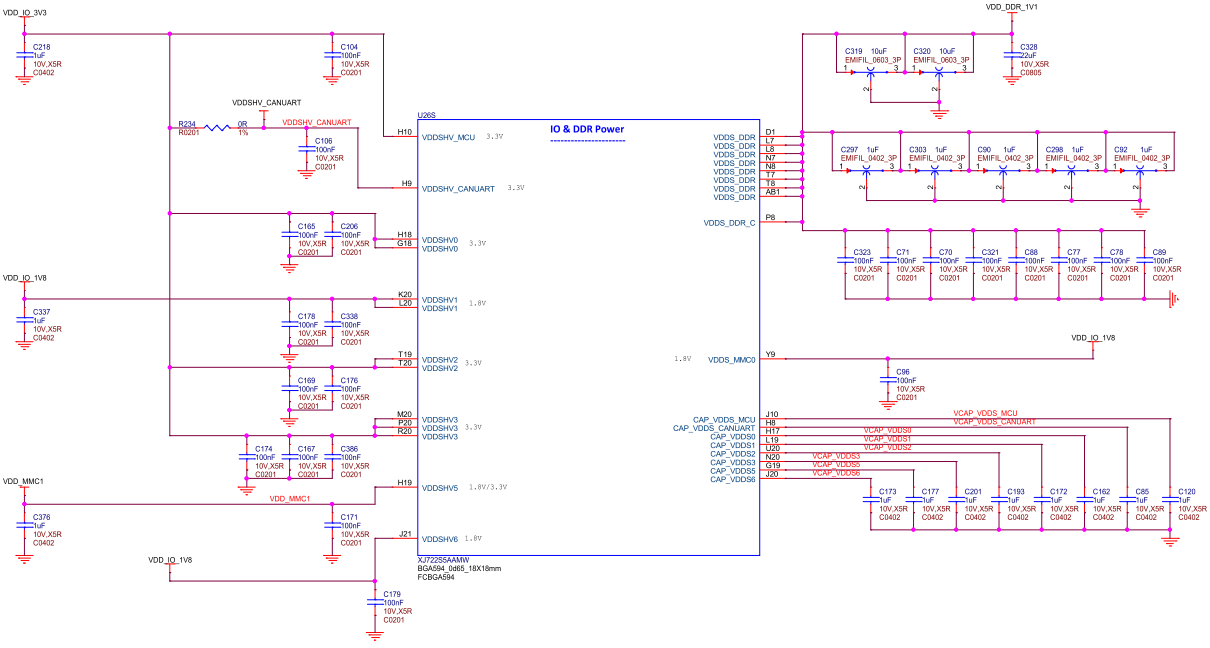


Fig. 3.11: BeagleY-AI AI SoC IO and DDR decoupling capacitors

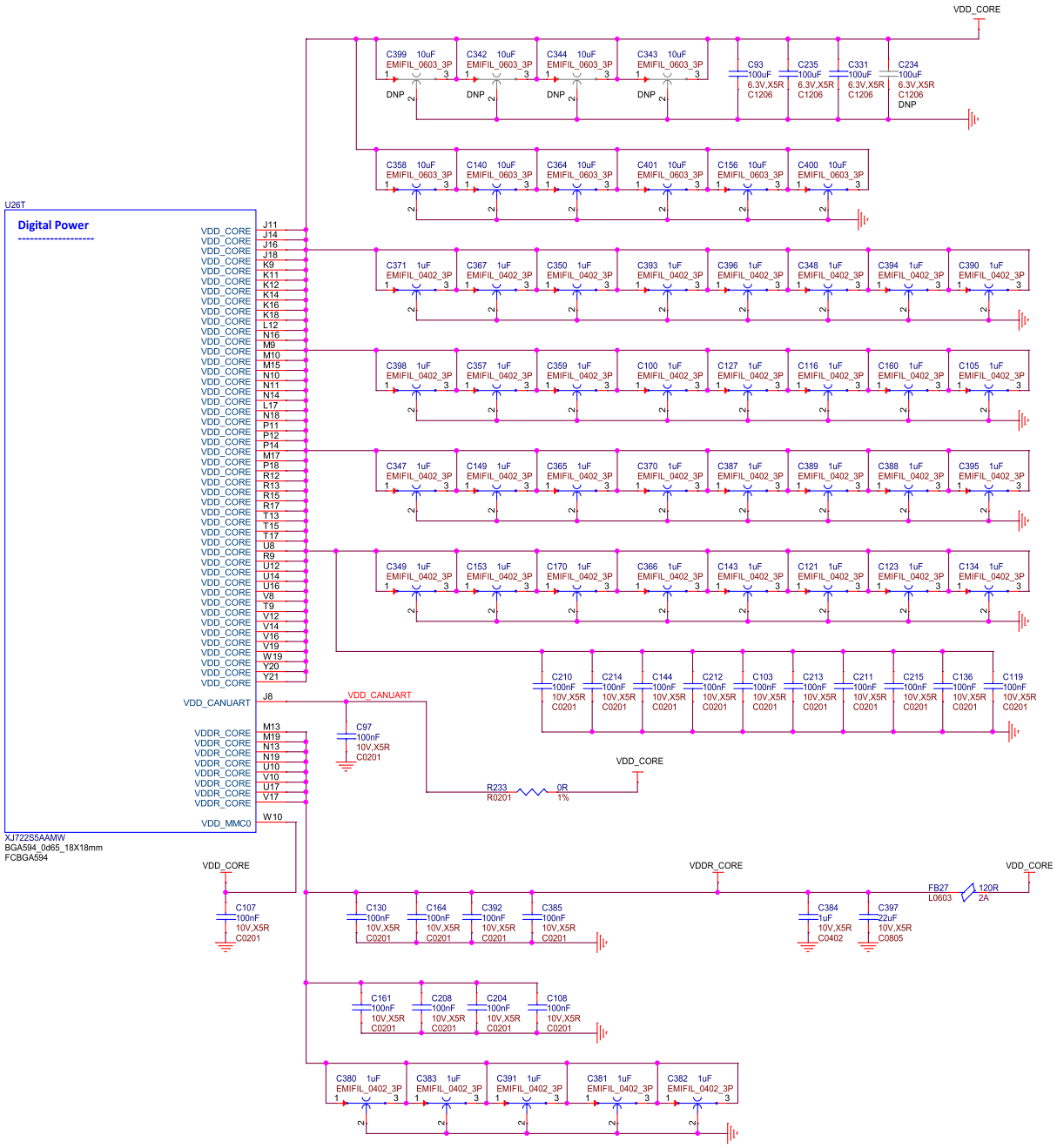


Fig. 3.12: BeagleY-AI SoC VDD & VDDR_CORE decoupling capacitors

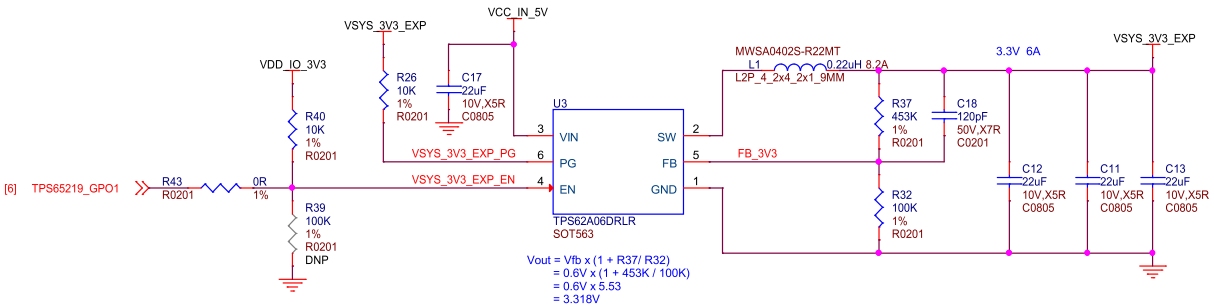


Fig. 3.13: BeagleY-AI VSYS 3V3

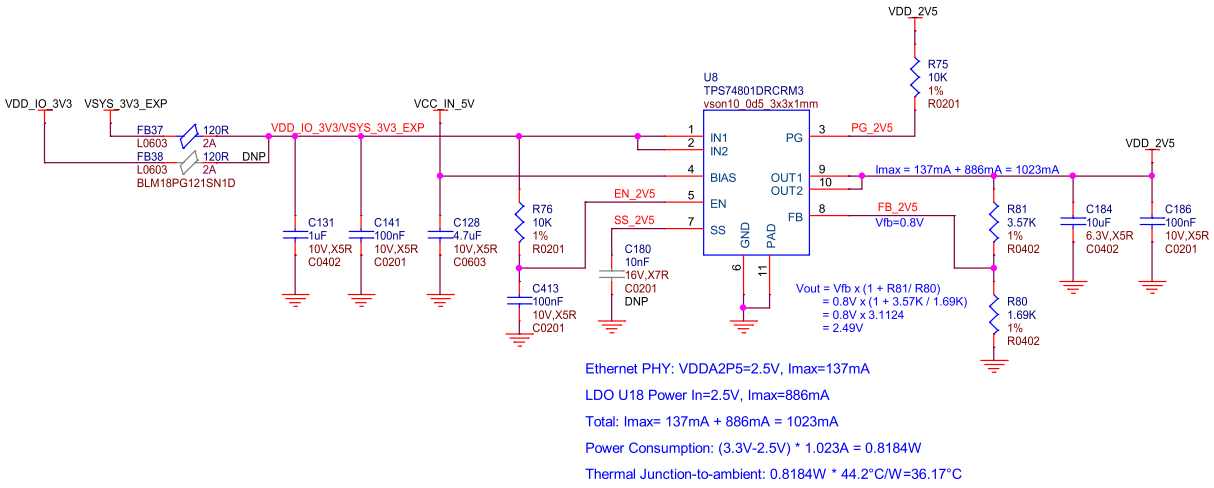


Fig. 3.14: BeagleY-AI ethernet power 3V3 to 2V5

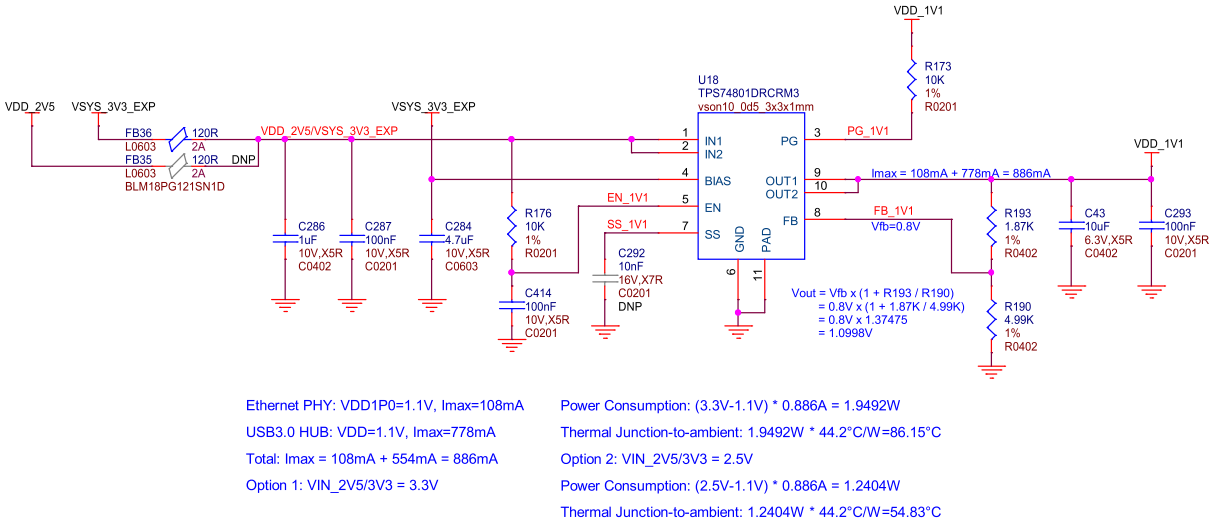


Fig. 3.15: BeagleY-AI 3V3/V5 to 1V1 LDO

The 1V1 Rail used by the PHY and USB 3.1 Hub is generated a discrete TPS74801 regulator. By default, this regulator is fed by the 3V3 VSYS regulator previously discussed.

3.6 Memory

3.6.1 RAM (LPDDR4)

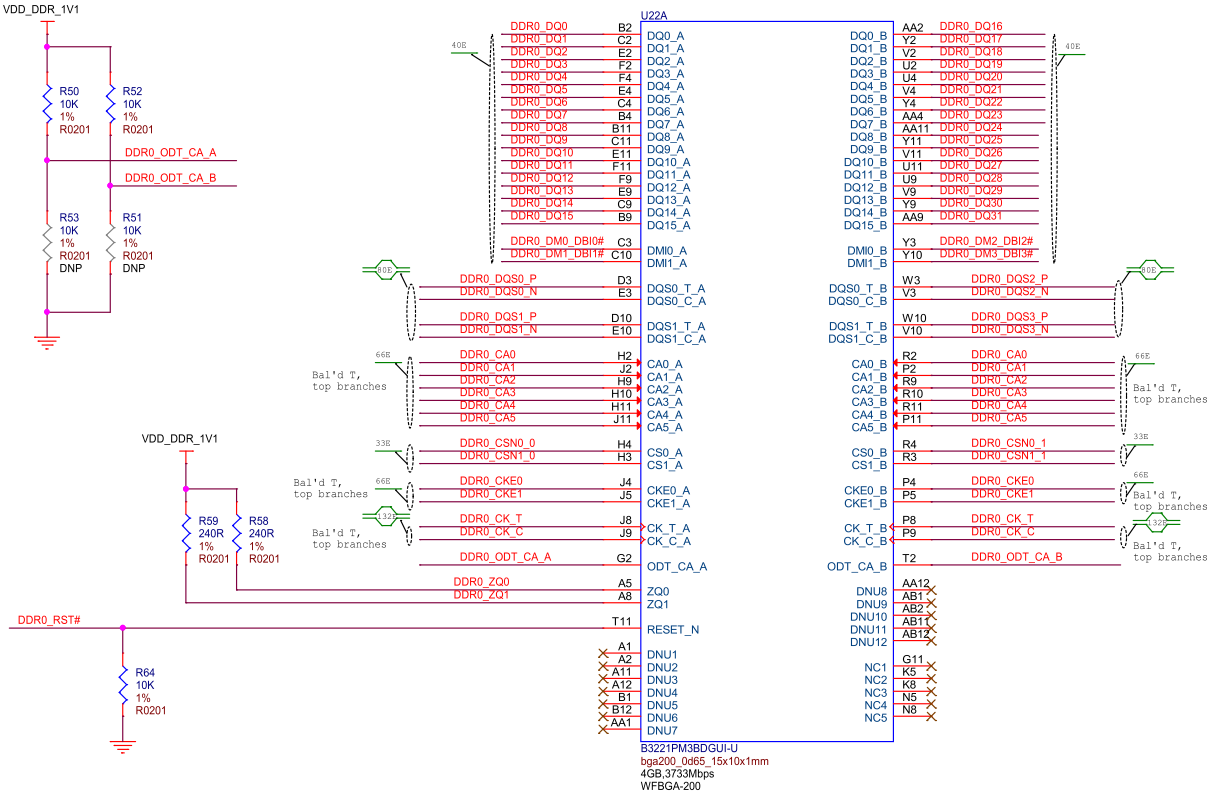


Fig. 3.16: BeagleY-AI DDR

BeagleY-AI has 4GB of Kingston x32 LPDDR4 Memory.

Todo: Add Final DDR Part Number

3.6.2 EEPROM

BeagleY-AI features an on-board FT24C32A 32Kbit I2C EEPROM for storing things like board information, manufacture date, etc.

Todo: Add details about specific EEPROM contents and formatting.

3.6.3 microSD Card

The microSD card is the primary boot interface for BeagleY-AI, it corresponds to the MMC1 interface on the AM67A SoC.

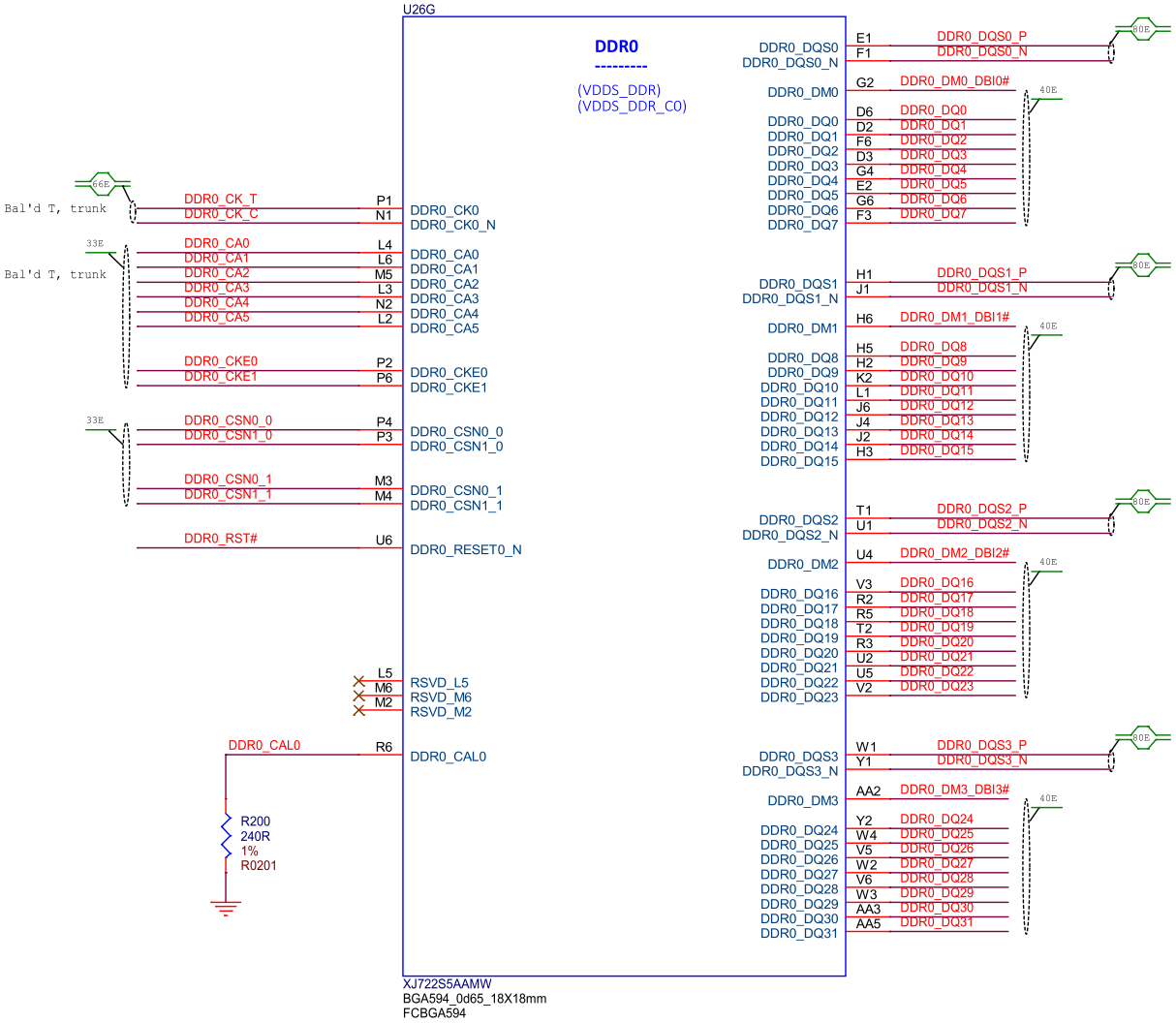


Fig. 3.17: BeagleY-AI SoC DDR0 connections

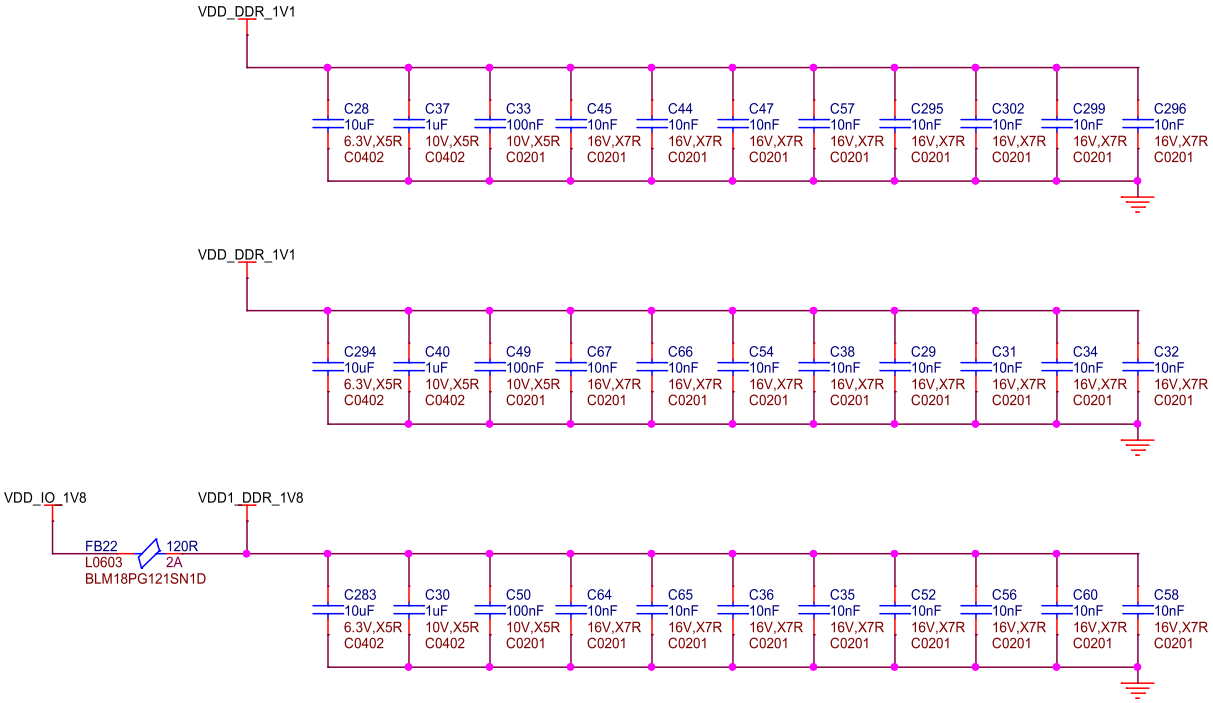


Fig. 3.18: BeagleY-AI DDR caps

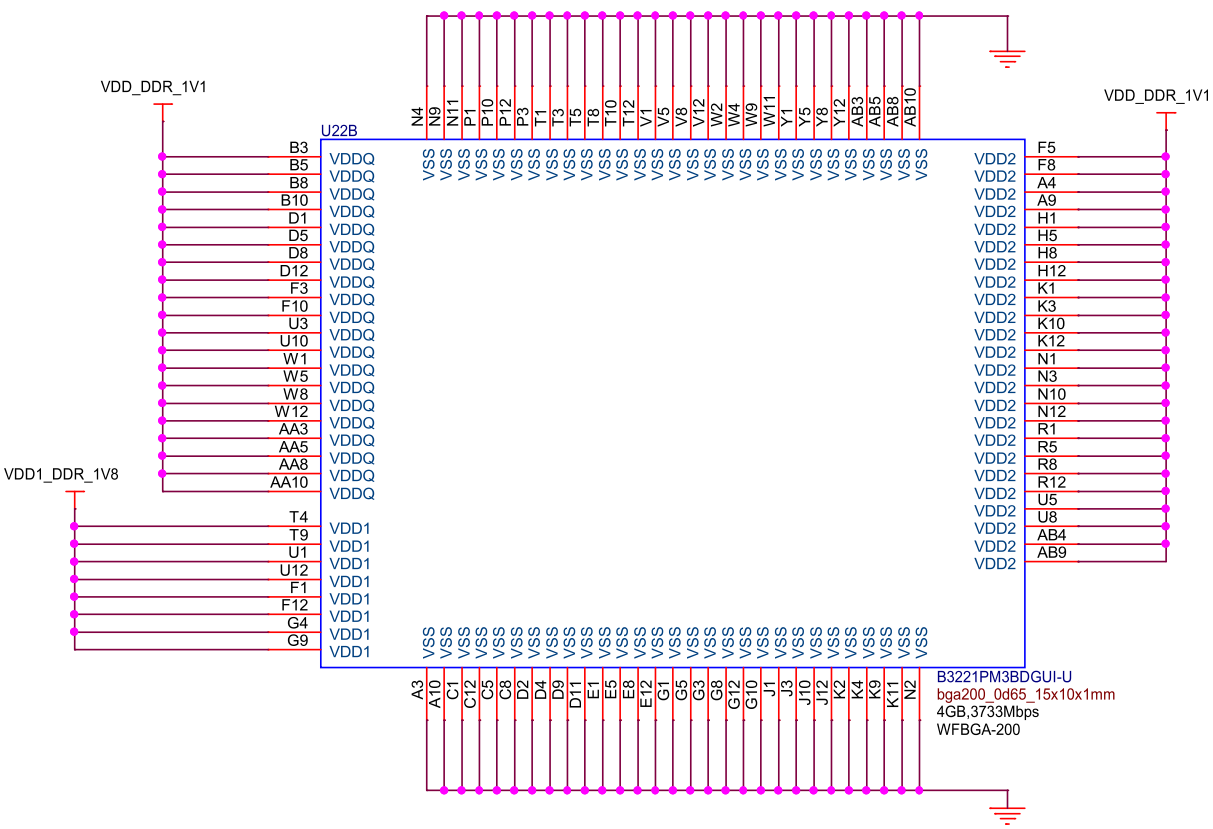


Fig. 3.19: BeagleY-AI DDR power

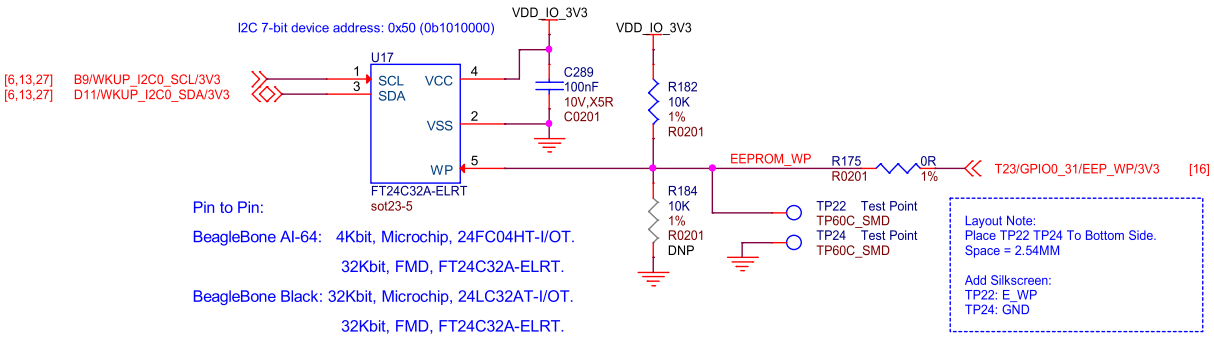


Fig. 3.20: BeagleY-AI board id eeprom

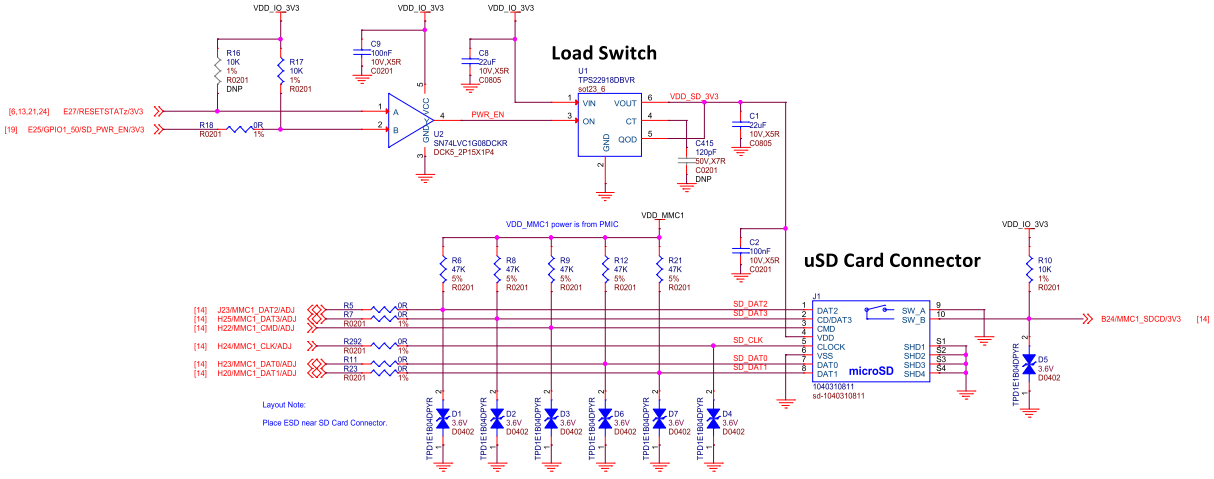


Fig. 3.21: BeagleY-AI microSD card interface

To enable UHS-1 SD card functionality (and speeds!), a load switch is provided which allows the SoC MMC1 PHY to switch the SD Card IO voltage to 1.8V.

Todo: Explain UHS-1 in more detail and add link to TRM for boot modes and resistor swap options for advanced users.

3.7 General Expansion

3.7.1 40pin Header

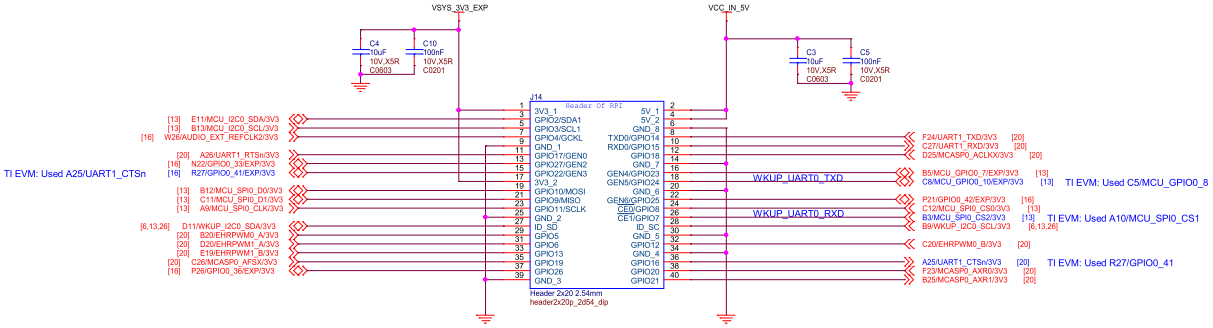


Fig. 3.22: BeagleY-AI user expansion connector

BeagleY-AI features a 40-pin GPIO Header which aims to enable compatibility with a lot of existing Raspberry Pi HAT add-on boards. See pinout.beagleboard.io for a more comprehensive view of the 40 pin GPIO header, available pin functions and tested accessories!

Todo: Add link to docs on building expansion accessories.

3.7.2 I2C

By default, 5 different I2C interfaces are exposed, all of which feature external 2.2KΩ pull-up resistors. 3 of the interfaces are used by the CSI, DSI and OLDI ports for Cameras & Displays. The remaining 2 ports are exposed on the 40pin GPIO expansion connector.

The MCU_I2C0 interface is intended as the primary external I2C interface for BeagleY-AI and matches physical pins 3 and 5 of the header. Most HATs will use these pins.

While WKUP_I2C0 is also exposed on the 40pin Header (physical pins 27 & 28), that bus is shared with several on-board devices, namely the PMIC, VDD_CORE regulator, Board ID EEPROM and RTC. As such, it is highly advisable to leave these pins unused unless you are sure you know what you are doing. These pins are normally only pinned out as a “HAT EEPROM detect” for RPi HATs that provide such functionality (of which there are very few)

See pinout.beagleboard.io/pinout/i2c for a more visual explanation.

3.7.3 USB

BeagleY-AI features a USB3.1 HUB that provides 4 total USB3.1 Ports from a single USB3.1 Gen-1 (5 Gbps) SERDES0 lane.

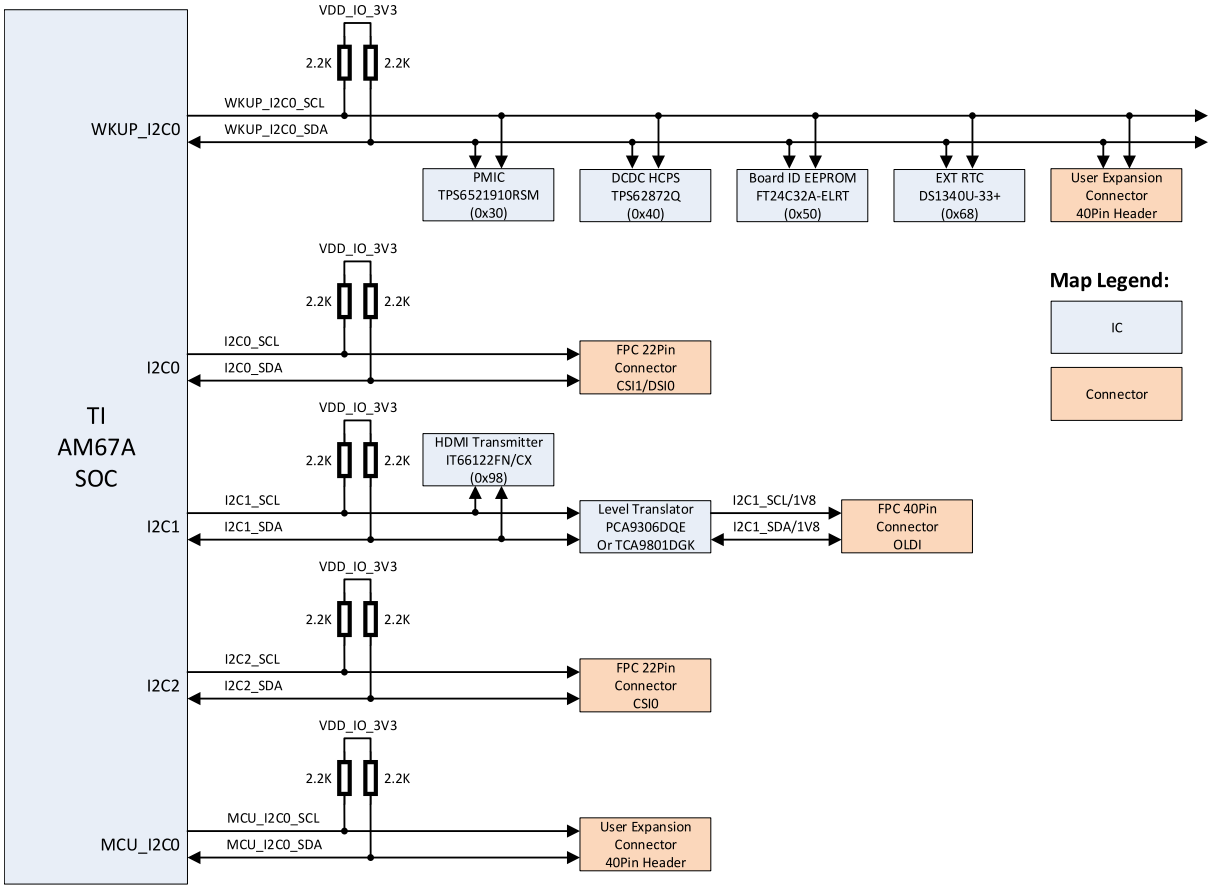


Fig. 3.23: BeagleY-AI I2C tree

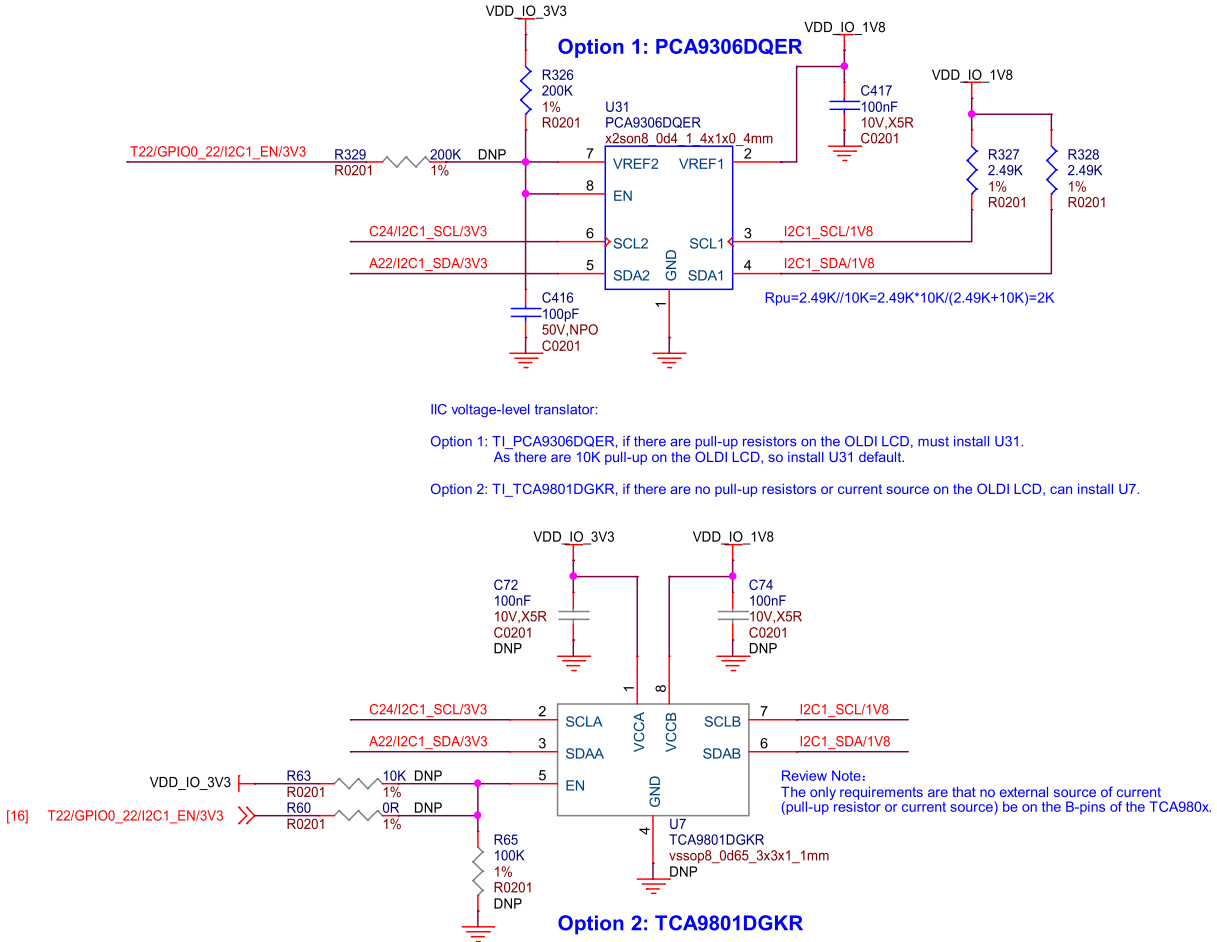


Fig. 3.24: BeagleY-AI voltage level translator

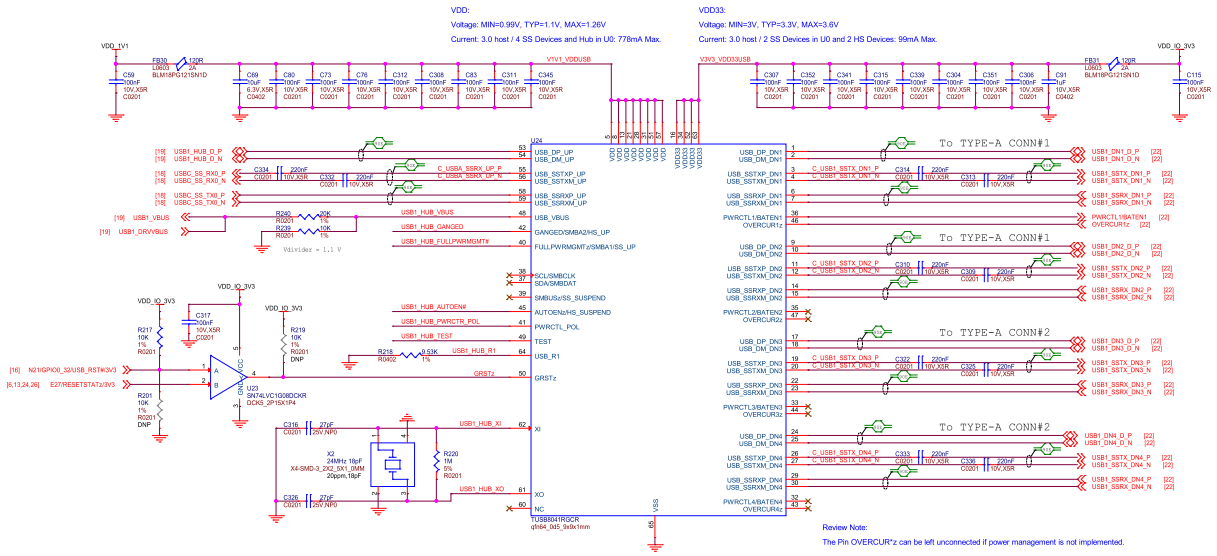


Fig. 3.25: BeagleY-AI USB3 hub

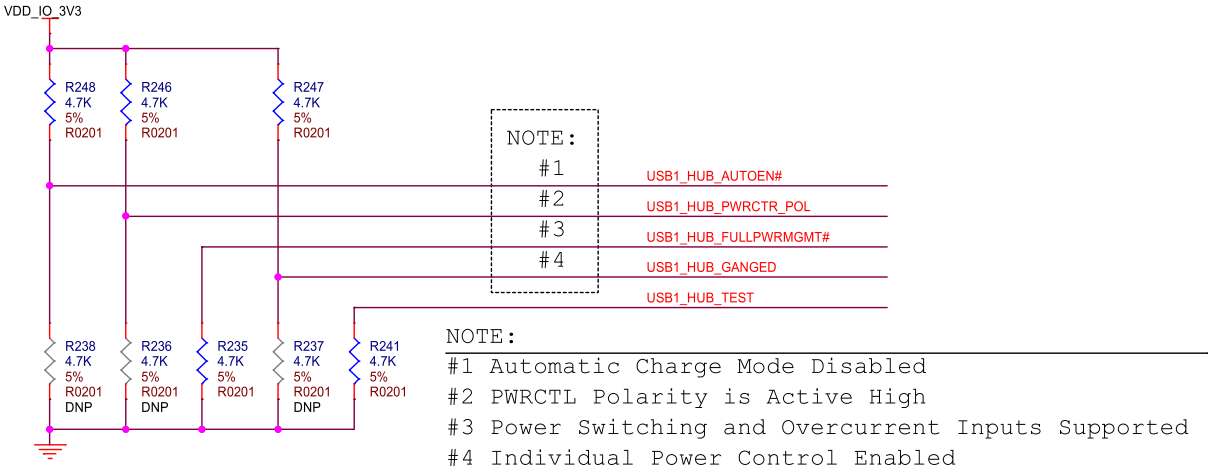


Fig. 3.26: BeagleY-AI USB hub config

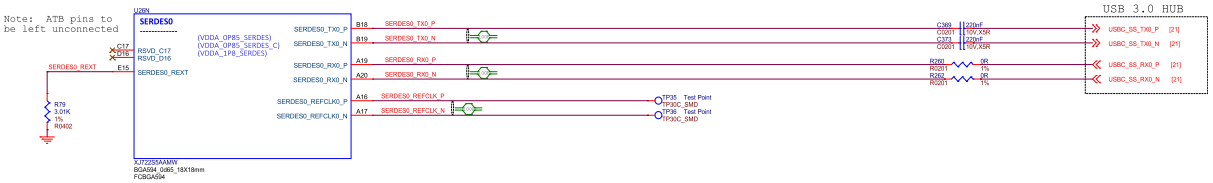


Fig. 3.27: BeagleY-AI SoC SERDES0

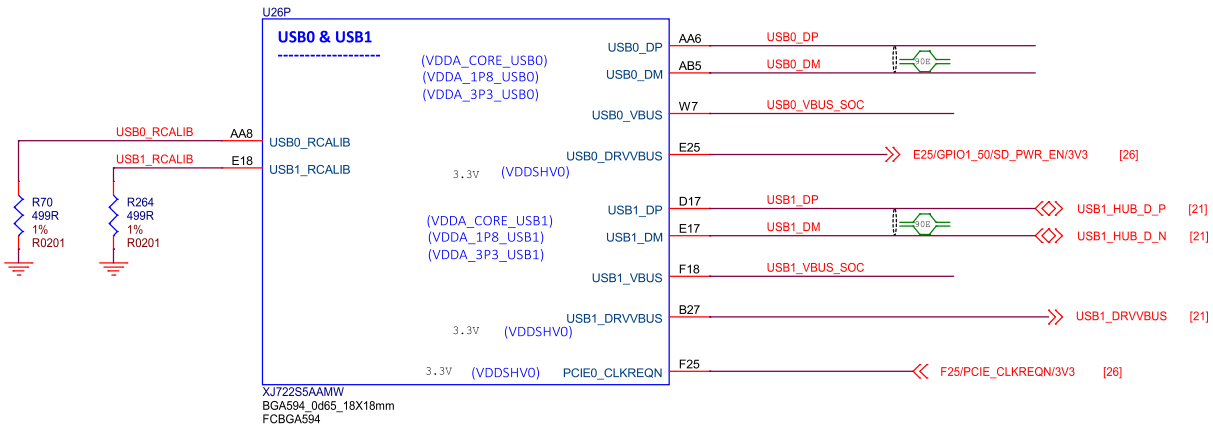


Fig. 3.28: BeagleY-AI SoC USB0 and USB1

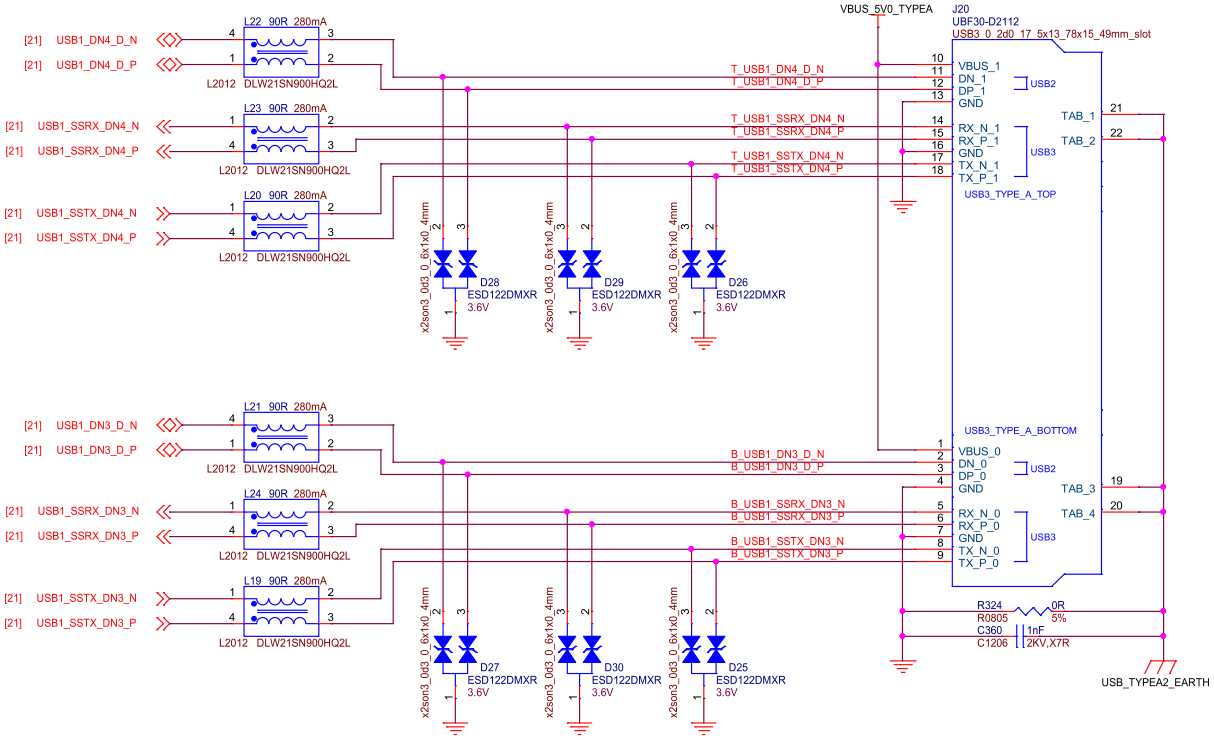


Fig. 3.29: BeagleY-AI USB-A Connector 1

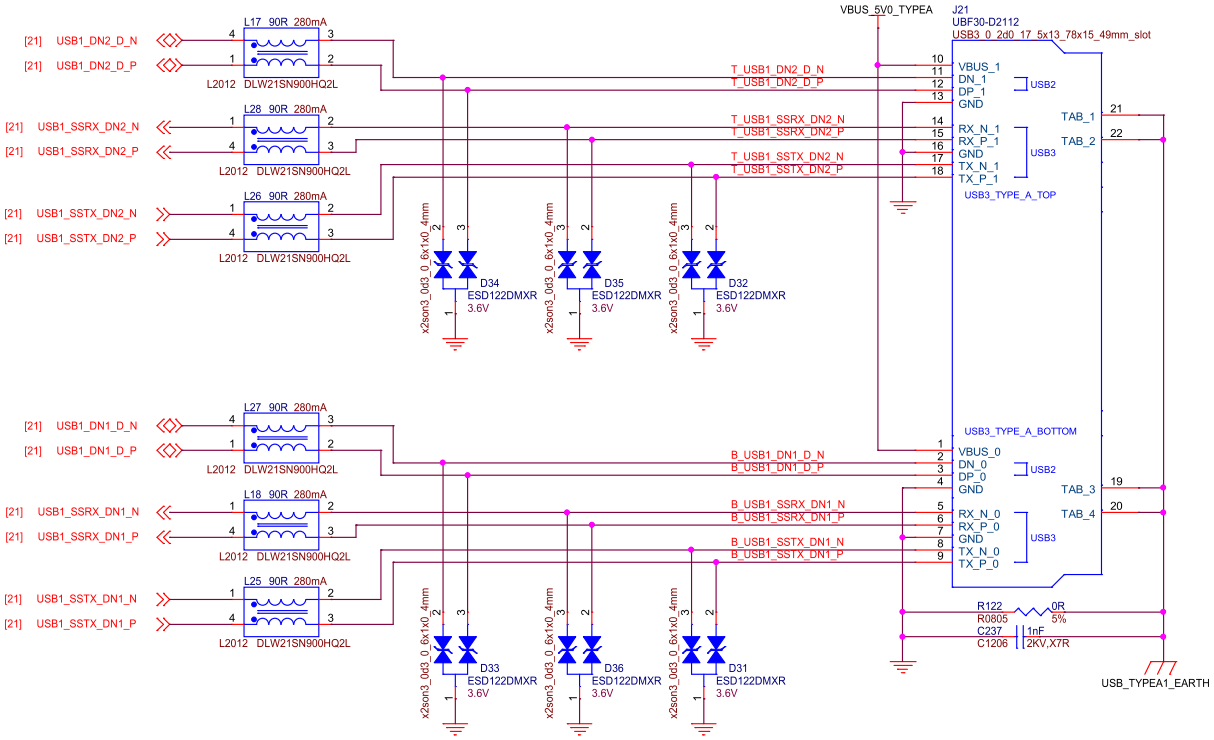


Fig. 3.30: BeagleY-AI USB-A Connector 2

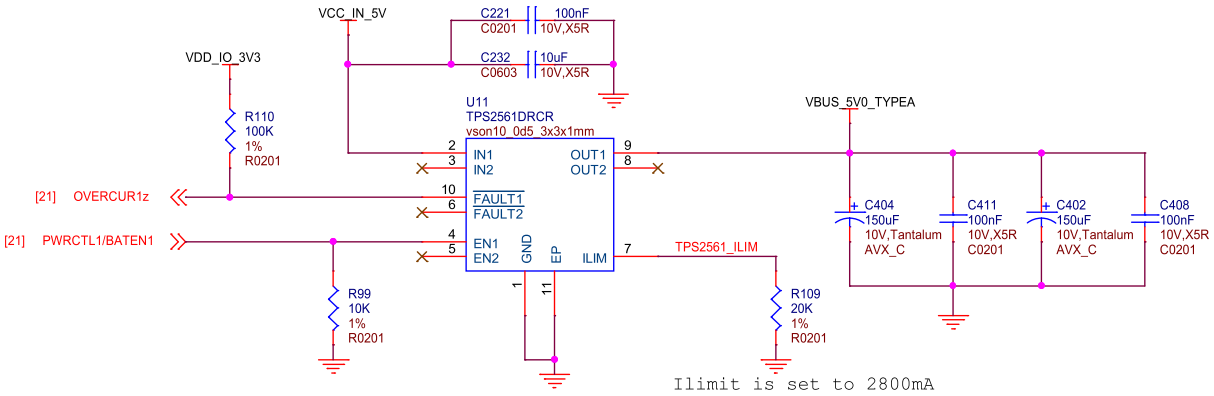


Fig. 3.31: BeagleY-AI dual USB current limiter

BeagleY-AI features a dedicated USB current limiter that will prevent the Type-A ports from drawing power in excess of 2.8A.

3.7.4 PCI Express

BeagleY-AI features an RPi 5 compatible PCIe connector rated for PCIe Gen2 x1 (5GT/s) connected to SERDES1 on AM67A.

Note: Just like the Raspberry Pi 5, while the AM67A SoC is capable of PCIe Gen3 (8GT/s), the choice of cable/connector means that some devices may not be able to run at full Gen 3 speeds and will need to be limited to Gen 2 for stable operation.

3.7.5 RTC (Real-time Clock)

BeagleY-AI has an on-board I2C RTC that can be powered by an external RTC for accurate time-keeping even when the board is powered off. For more information, see the corresponding docs page - beagley-ai-using-rtc

3.7.6 Fan Header

BeagleY-AI features a Raspberry Pi 5 compatible Fan connector. The fan is software PWM controller in Linux by default to maintain a balance between cooling and noise depending on SoC temperature.

3.8 Networking

3.8.1 WiFi / Bluetooth LE

BeagleY-AI features a Beagle BM3301 Wireless module based on the Texas Instruments CC3301 which features 2.4Ghz WiFi6 (802.11AX) and BLE 5.4

Note: 5Ghz WiFi Bands and Bluetooth Classic are not supported by the CC3301.

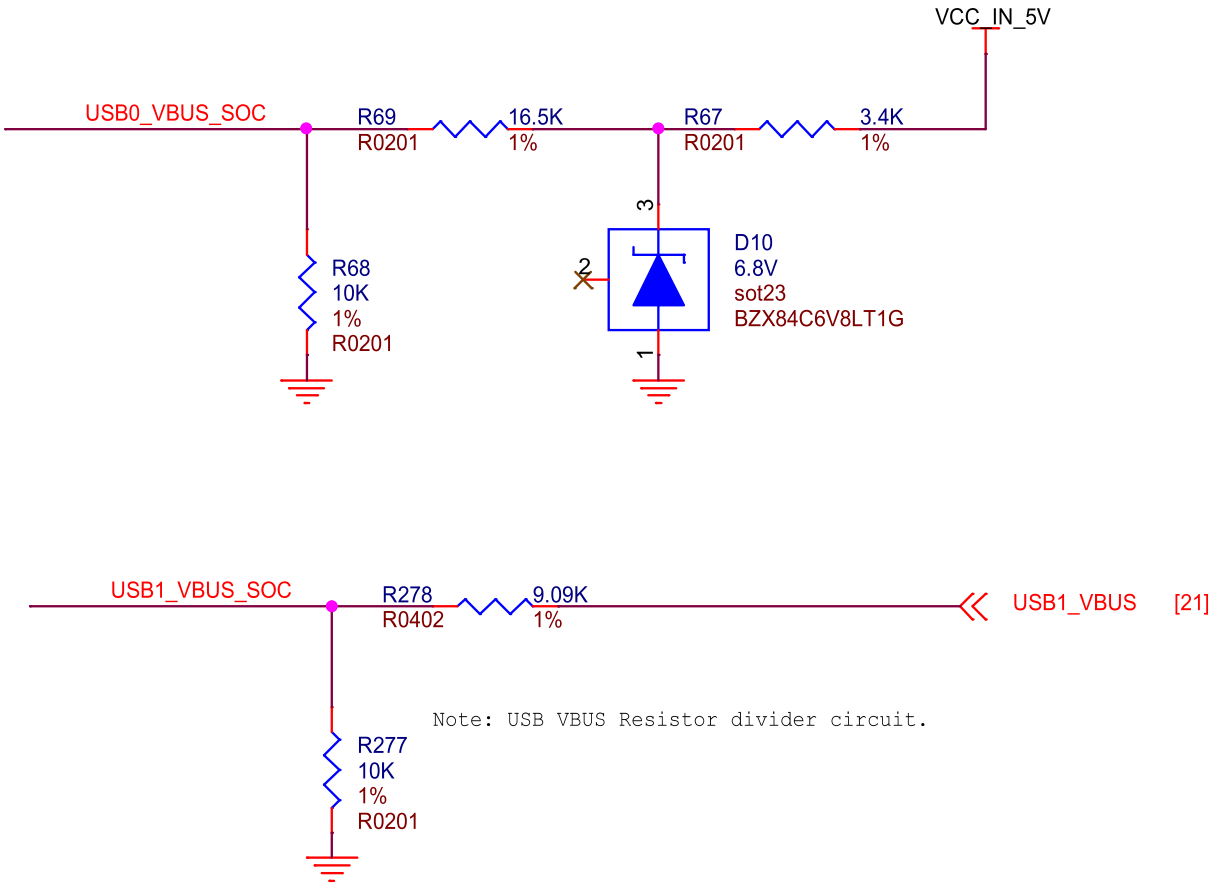


Fig. 3.32: BeagleY-AI USB VBUS resistor divider circuit

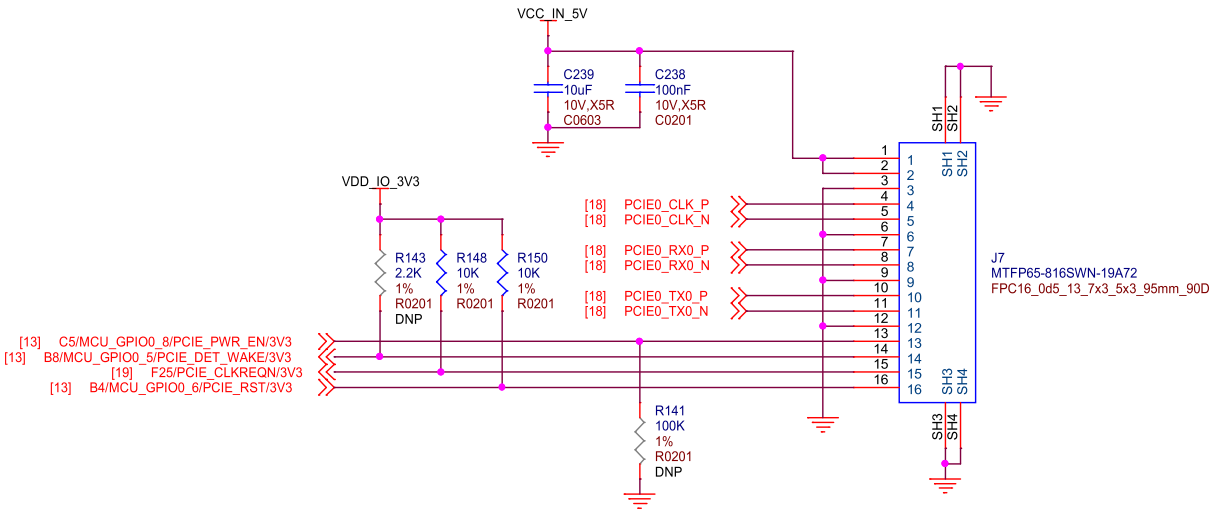


Fig. 3.33: BeagleY-AI PCIE connector

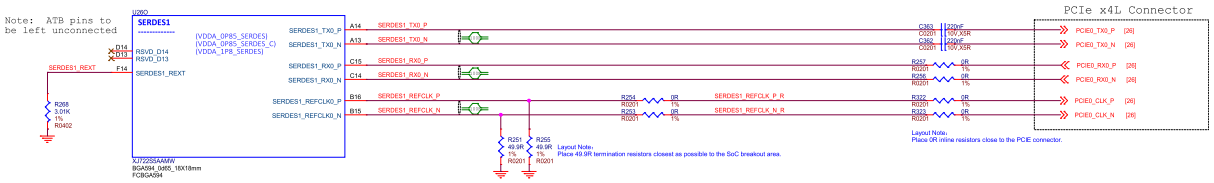


Fig. 3.34: BeagleY-AI SoC SERDES1

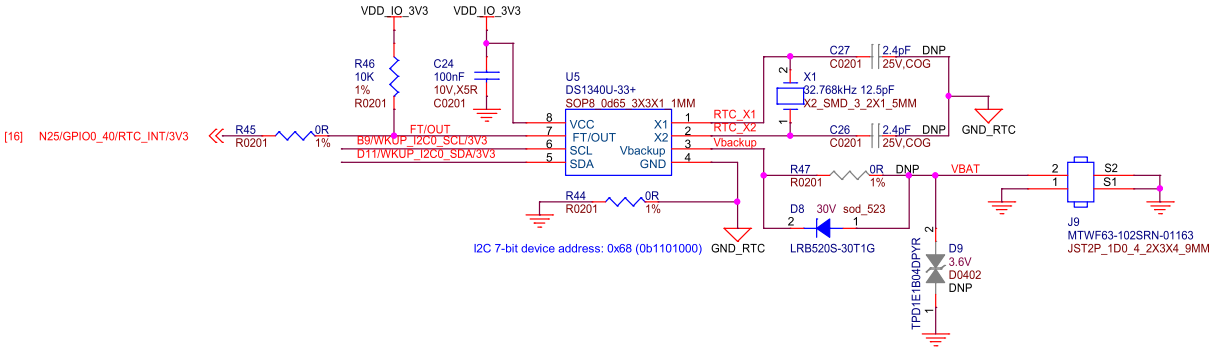


Fig. 3.35: BeagleY-AI I2C ext RTC

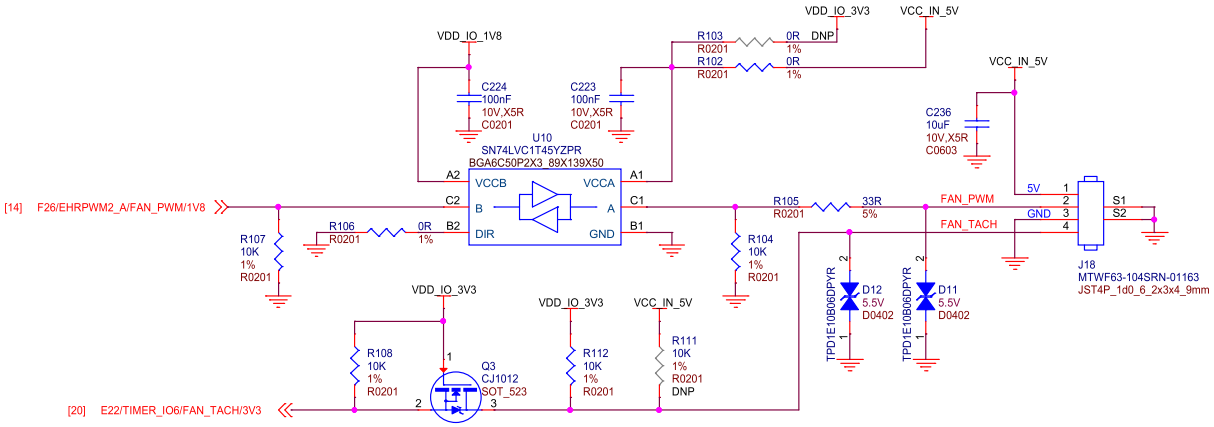


Fig. 3.36: BeagleY-AI fan connector

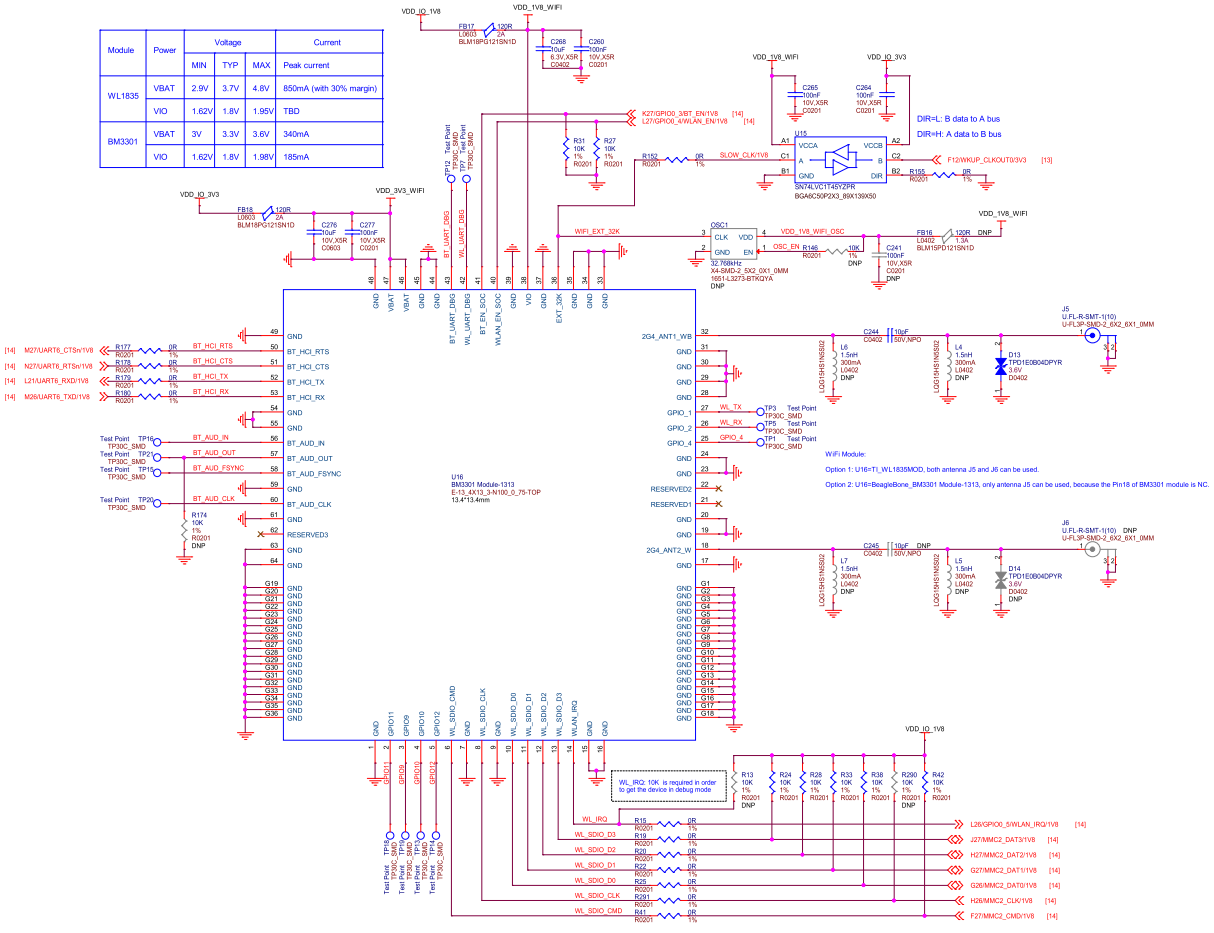


Fig. 3.37: BeagleY-AI WiFi module

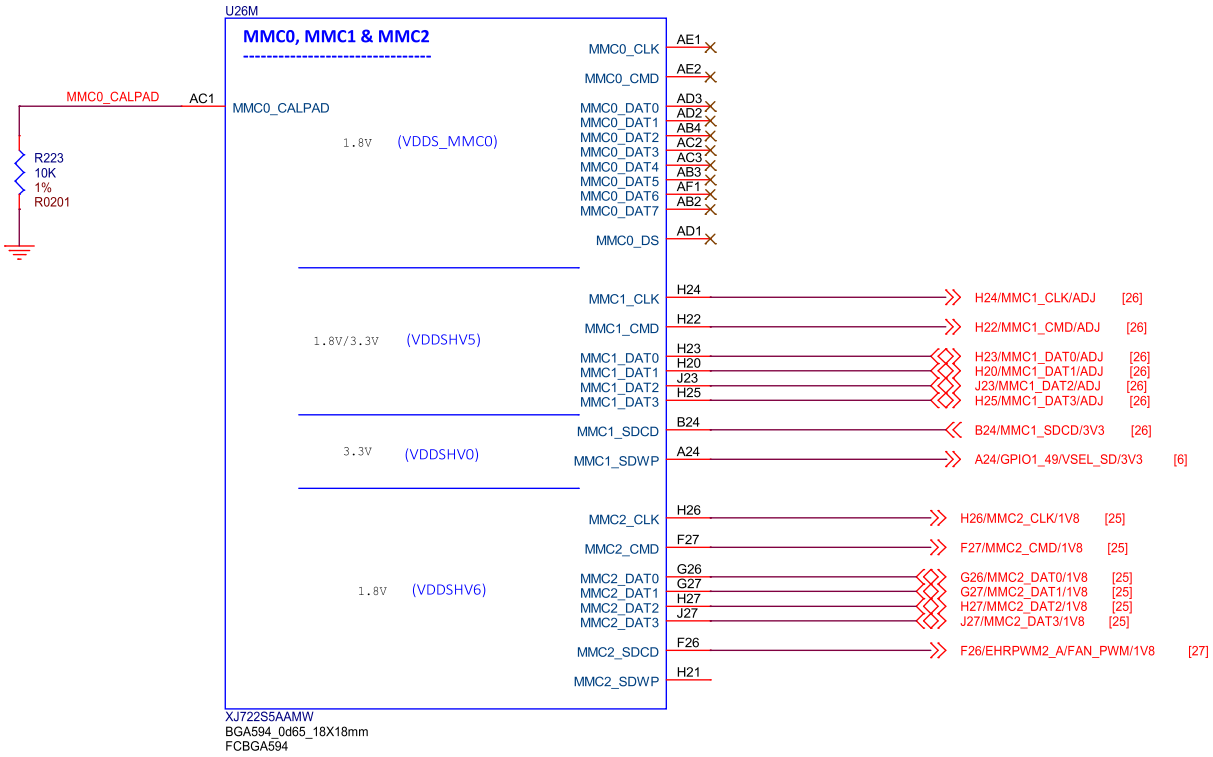


Fig. 3.38: BeagleY-AI SoC MMC0, MMC1, and MMC2

3.8.2 Ethernet

BeagleY-AI is equipped with a 1 Gb (10/100/1000) DP83867 Ethernet PHY connected over RGMII.

BeagleY-AI uses an RJ45 ethernet connector with integrated magnetics.

Optional PoE (Power over Ethernet) can also be used with compatible 3rd party HATs designed for the Raspberry Pi 5.

Note: Only Pi 5 PoE HATs are compatible, as Pi 4 and previous designs have the PoE pins in a different location.

3.9 Cameras & Displays

BeagleY-AI is capable of driving up to 3 Displays (HDMI, OLDI/LVDS & DSI) simultaneously.

- HDMI via DPI Converter up to 1920 x 1080 @60FPS
- OLDI/LVDS up to 3840 x 1080 @60FPS (Dual Link, 150-Mhz Pixel Clock)
- DSI up to 3840 x 1080 at 60fps (4 Lane MIPI® D-PHY, 300-MHz Pixel Clock)

It also features 2 CSI interfaces and can support up to 8 Cameras using Virtual Channels and V3Link.

Note: The CSI1/DSI0 22-pin port is muxed between the two interfaces like the RPi 5, meaning that you must chose if it's used as a Display or Camera port. The CSI0 22-pin connector can only be used as a Camera port.

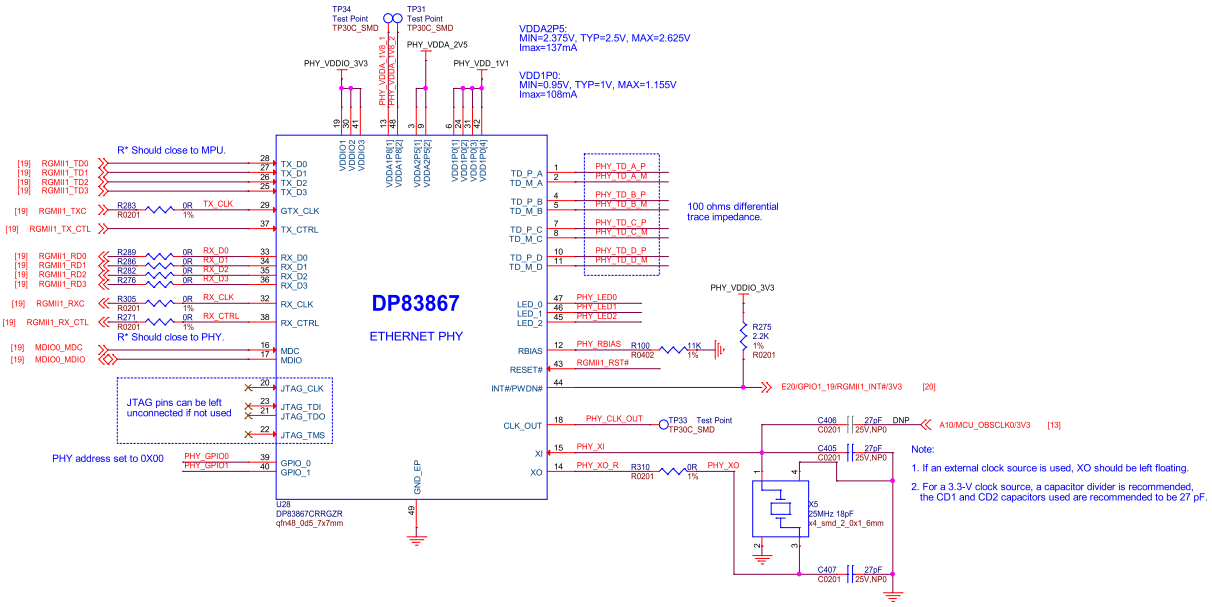


Fig. 3.39: BeagleY-AI ethernet DP83867

3.9.1 HDMI (DPI)

BeagleY-AI has a single HDMI 1.4 port capable of up to 1080p @60FPS with Audio. This is achieved using an external Parallel RGB (DPI) to HDMI converter from ITE.

Because the DPI interface is used up by the HDMI converter, it does mean that DPI is not available on the 40Pin GPIO header.

3.9.2 OLDI (LVDS)

The OLDI connector on BeagleY-AI has the same pinout as the one used by Beagle Play, meaning the same displays are compatible.

3.9.3 DSI

The DSI0 port is shared withe CSI1 and selectable via a MUX switch to maintain Pi functionality. It features the same pinout found on the 22-pin DSI connector on RPi5 and BeagleBone AI-64 and enables connectivity to existing supported DSI displays.

Please note that DSI is only available on the second of the two 22-pin “CSI” connectors.

3.9.4 CSI

To maintain a Pi compatible form factor, BeagleY-AI only exposes 2 of the 4 physical CSI interfaces of the AM67A SoC. Each CSI interfaces is MIPI® CSI-2 v1.3 + MIPI® D-PHY 1.2 with 4 Data Lanes running at up to 2.5Gbps/lane. The interface also supports up to 16 Virtual Channels for multi-camera applications using FPDLink or V3Link.

3.10 Buttons and LEDs

BeagleY-AI features a single dual-color (Red/Green) LED for Power/Status indication.

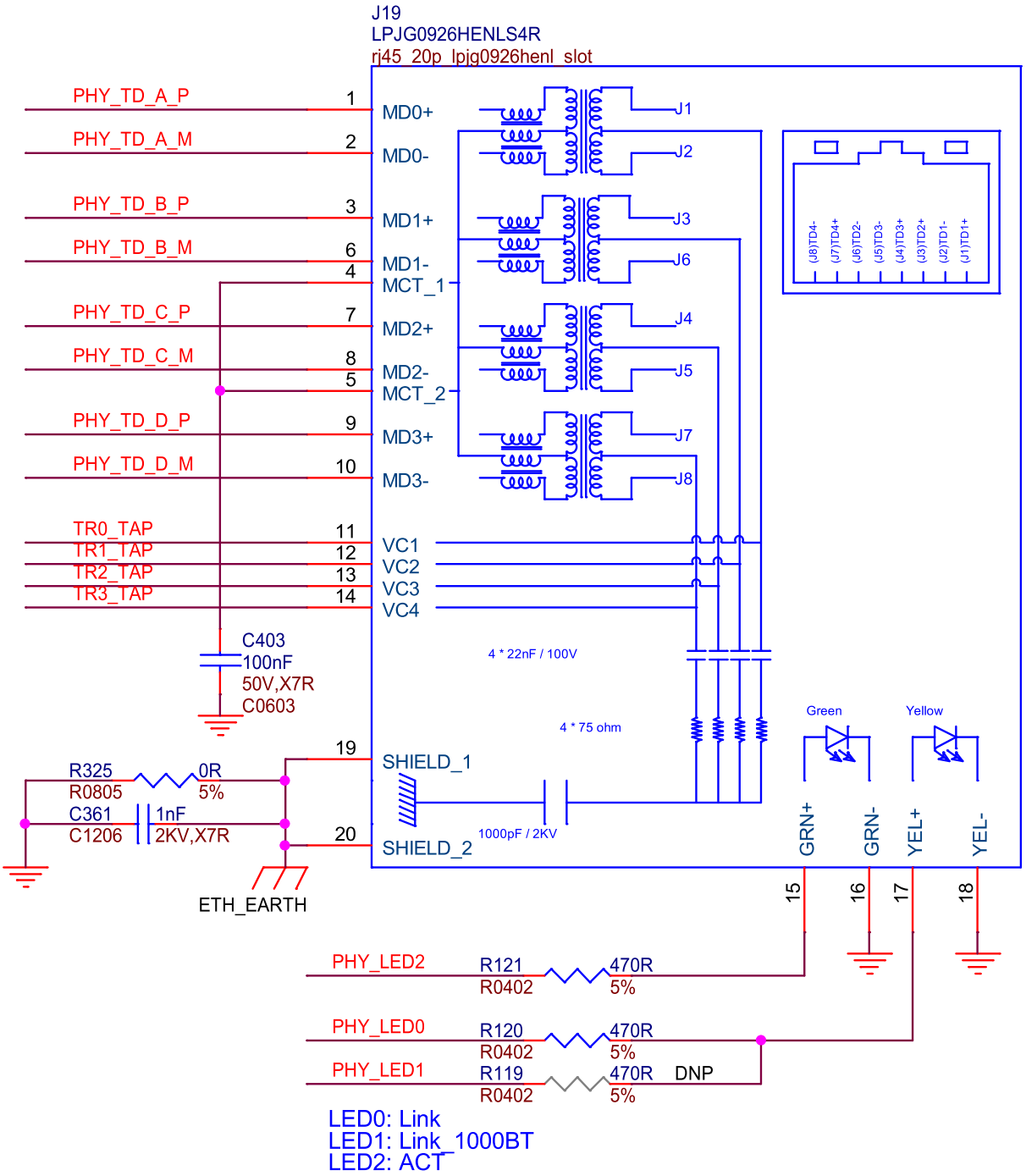


Fig. 3.40: BeagleY-AI ethernet connector

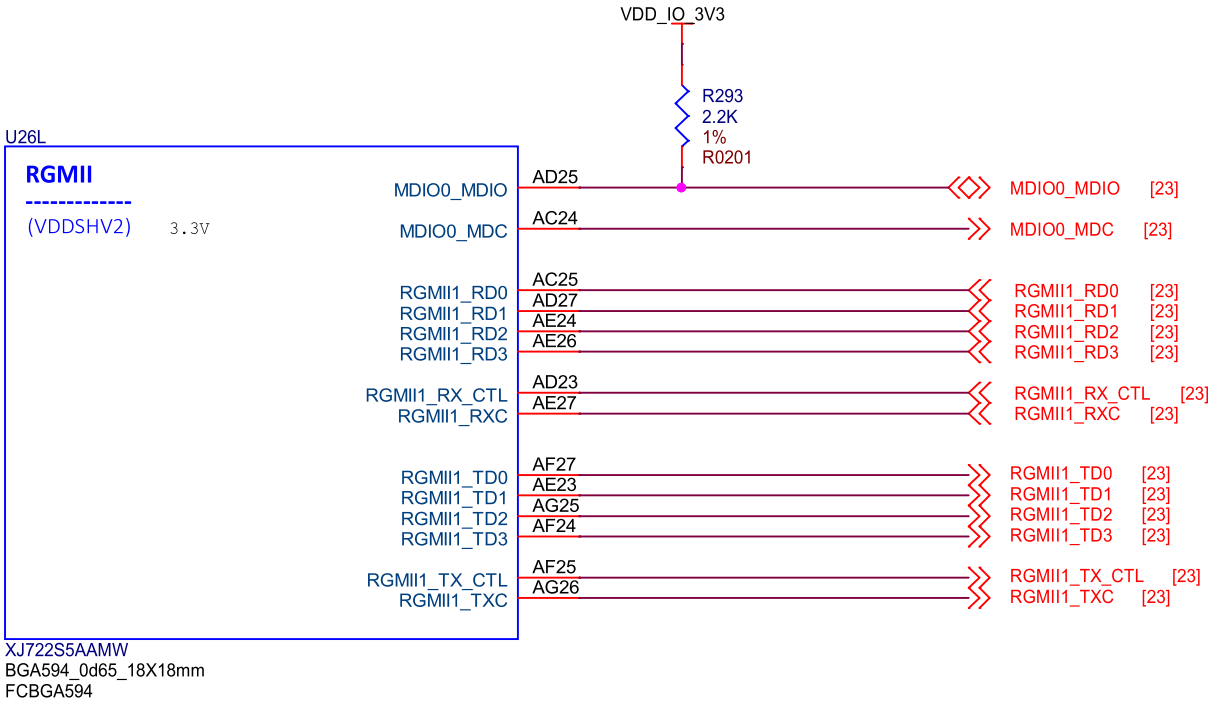


Fig. 3.41: BeagleY-AI SoC RGMII

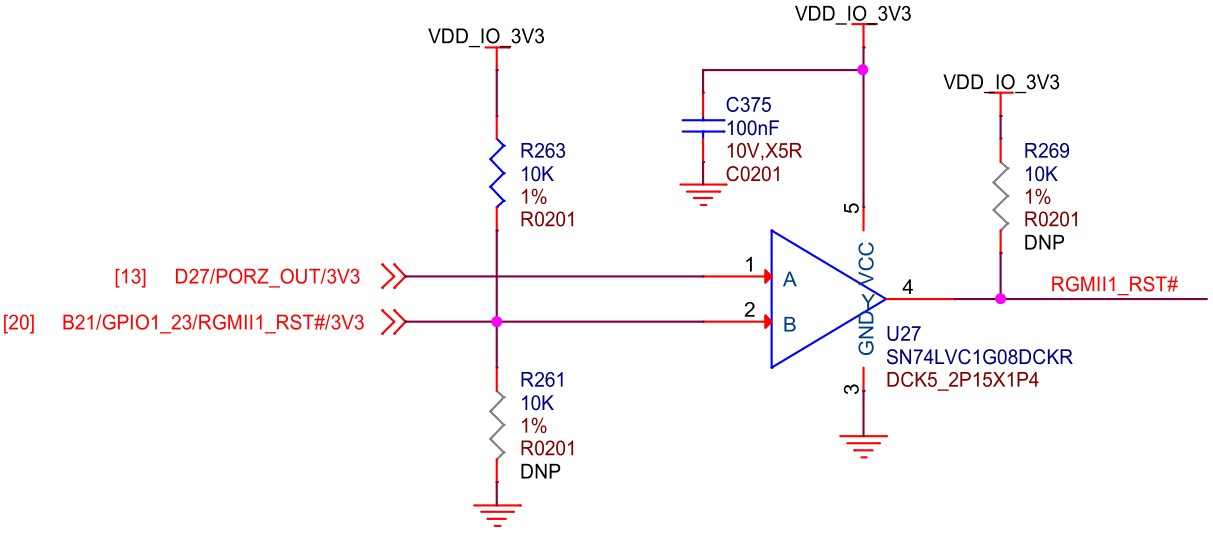


Fig. 3.42: BeagleY-AI SoC RGMII1 RST

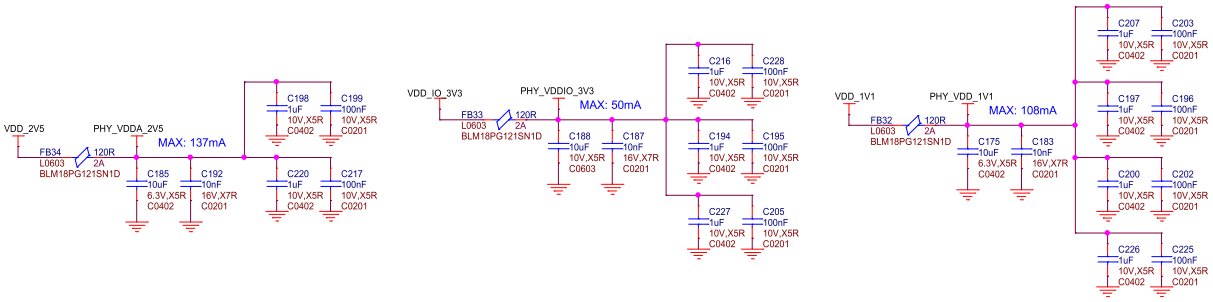


Fig. 3.43: BeagleY-AI Ethernet PHY caps

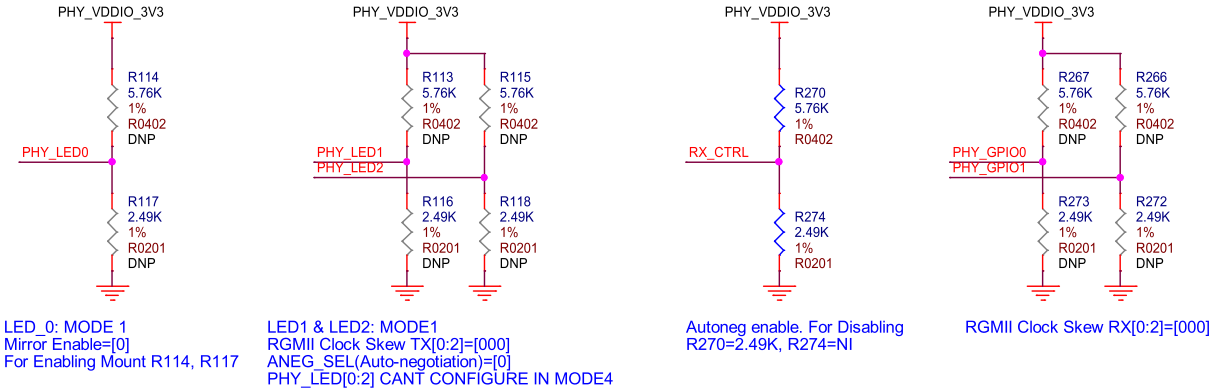


Fig. 3.44: BeagleY-AI Ethernet PHY misc

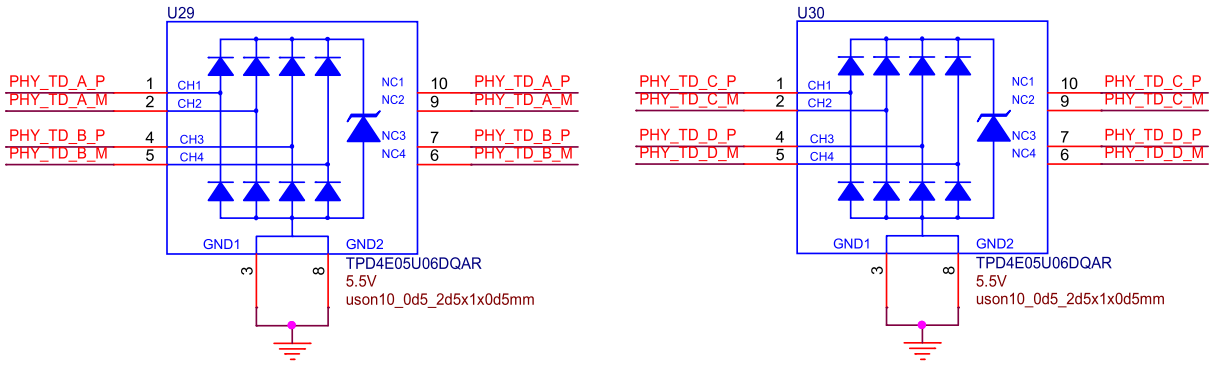


Fig. 3.45: BeagleY-AI Ethernet PHY protection

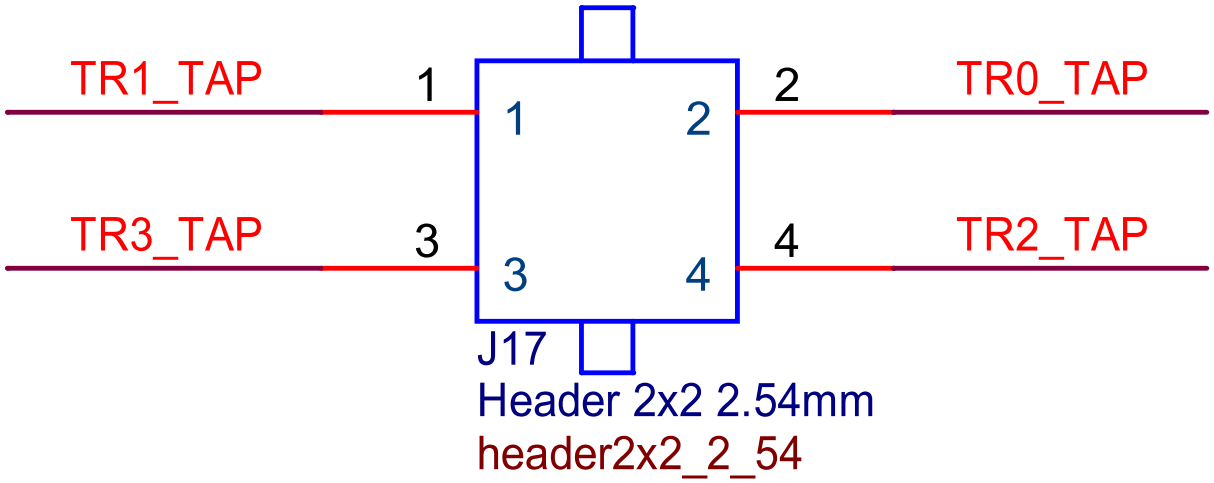


Fig. 3.46: BeagleY-AI PoE header

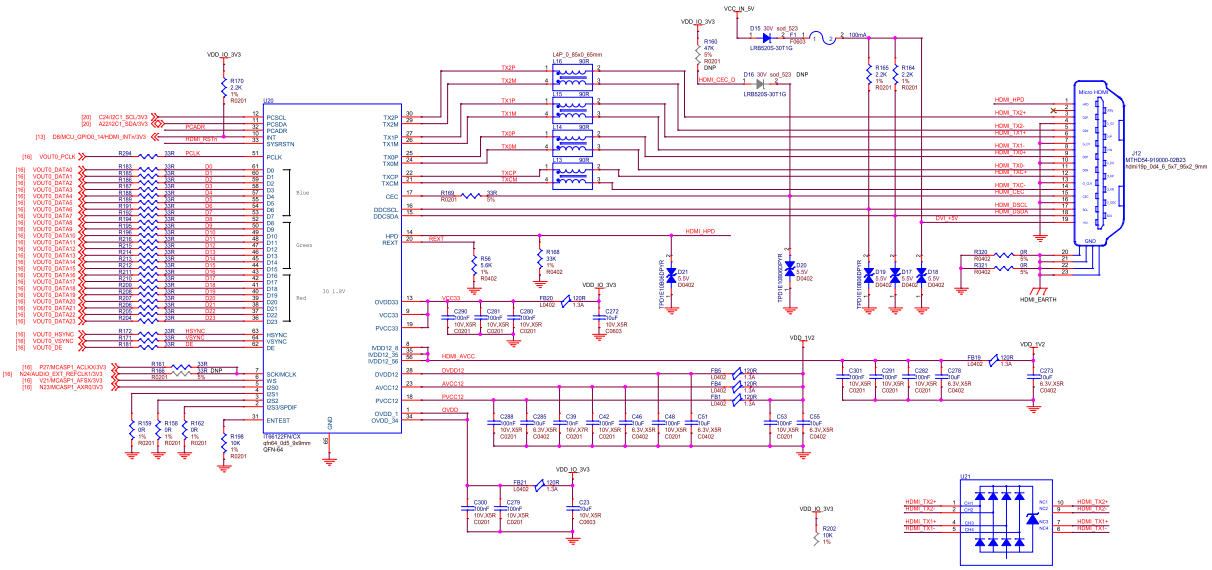


Fig. 3.47: BeagleY-AI RGB888 to HDMI

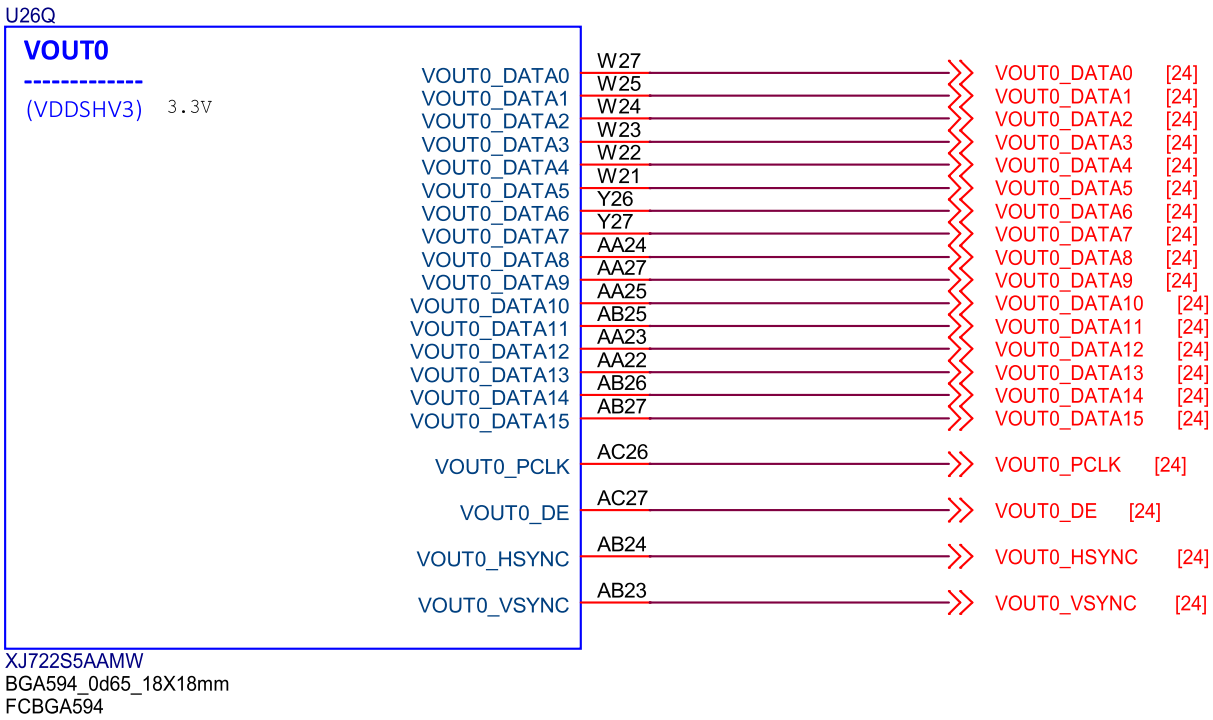


Fig. 3.48: BeagleY-AI SoC VOUT

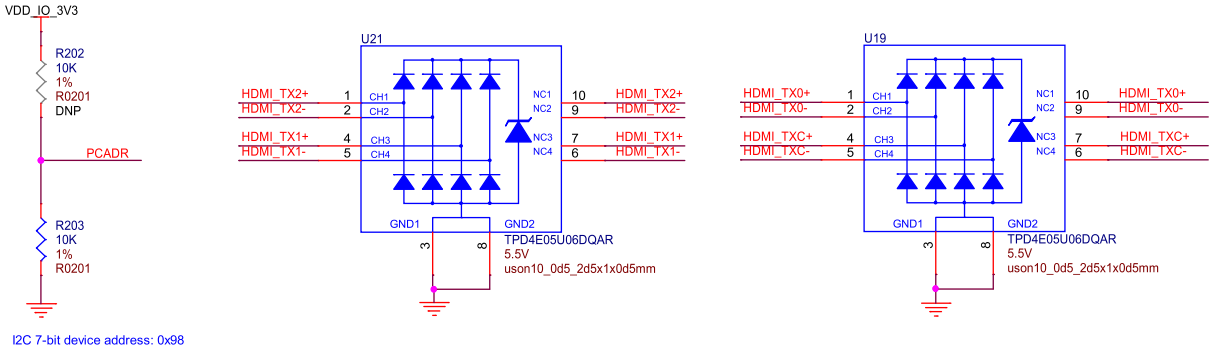


Fig. 3.49: BeagleY-AI HDMI addr protection

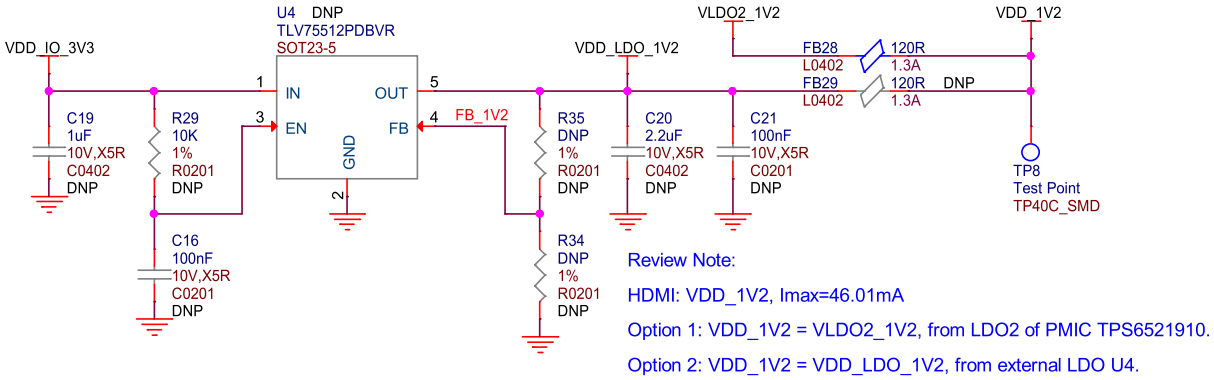


Fig. 3.50: BeagleY-AI HDMI power

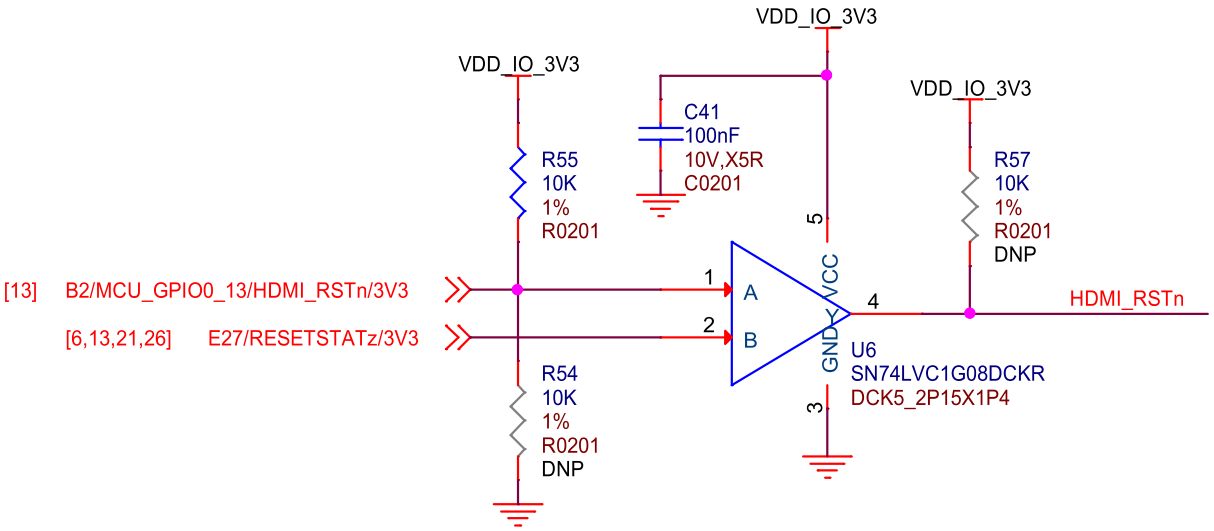


Fig. 3.51: BeagleY-AI HDMI reset

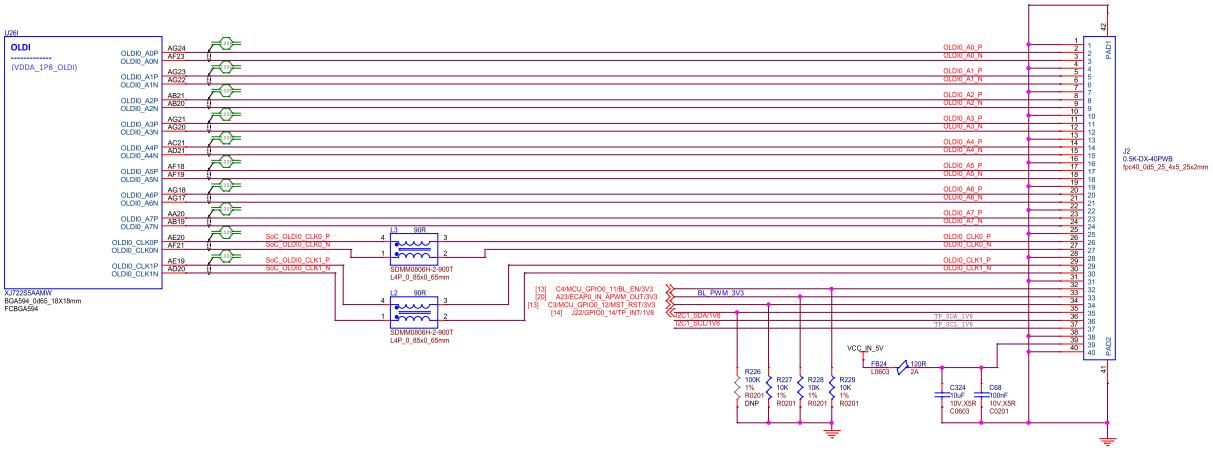


Fig. 3.52: BeagleY-AI SoC OLDI

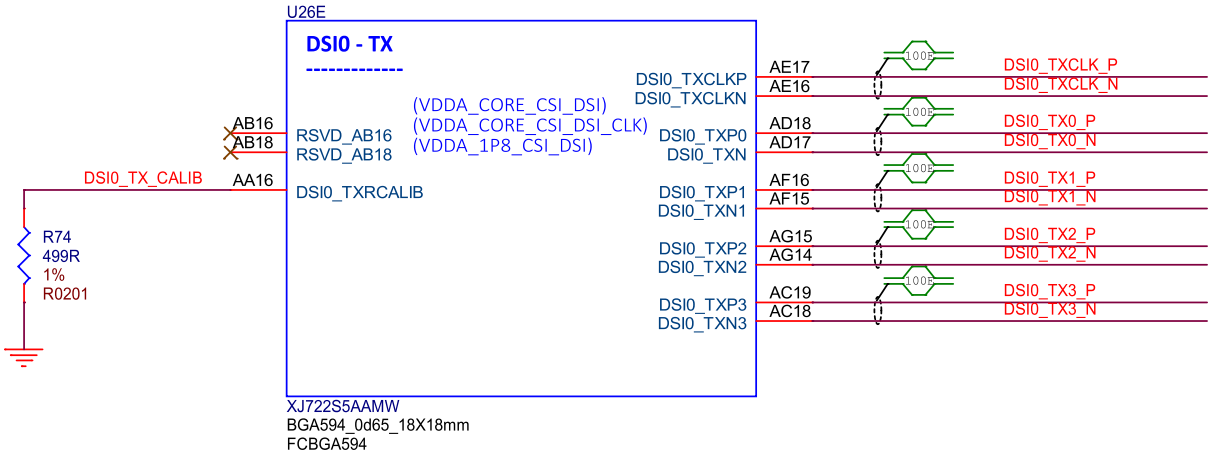


Fig. 3.53: BeagleY-AI SoC DSI0 TX connections

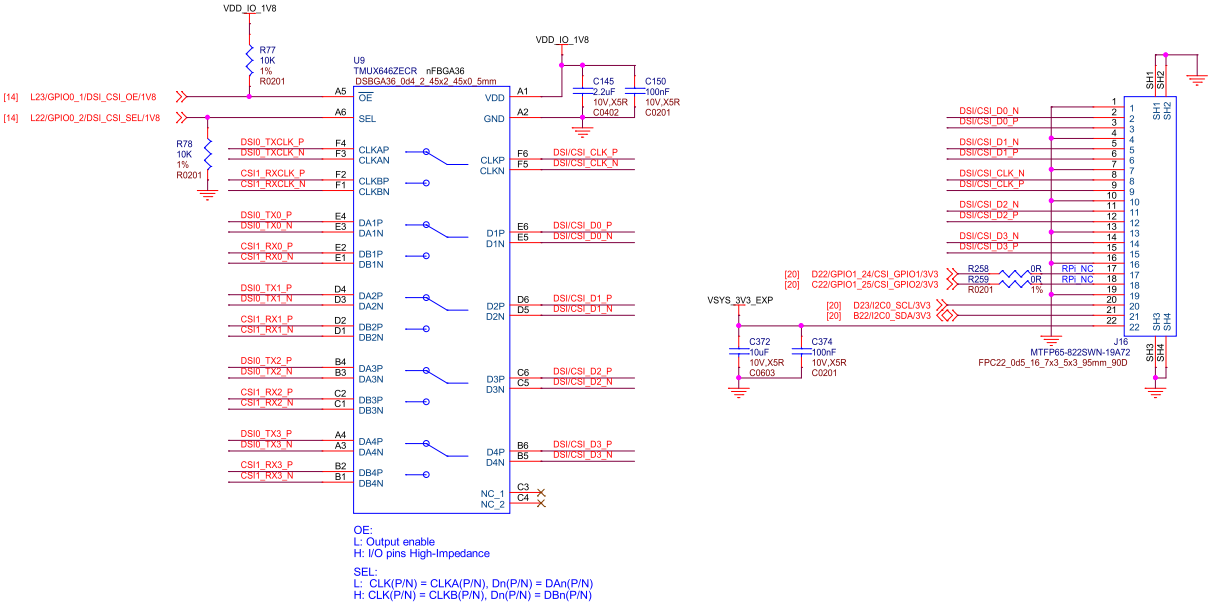


Fig. 3.54: BeagleY-AI RPI DSI/CSI

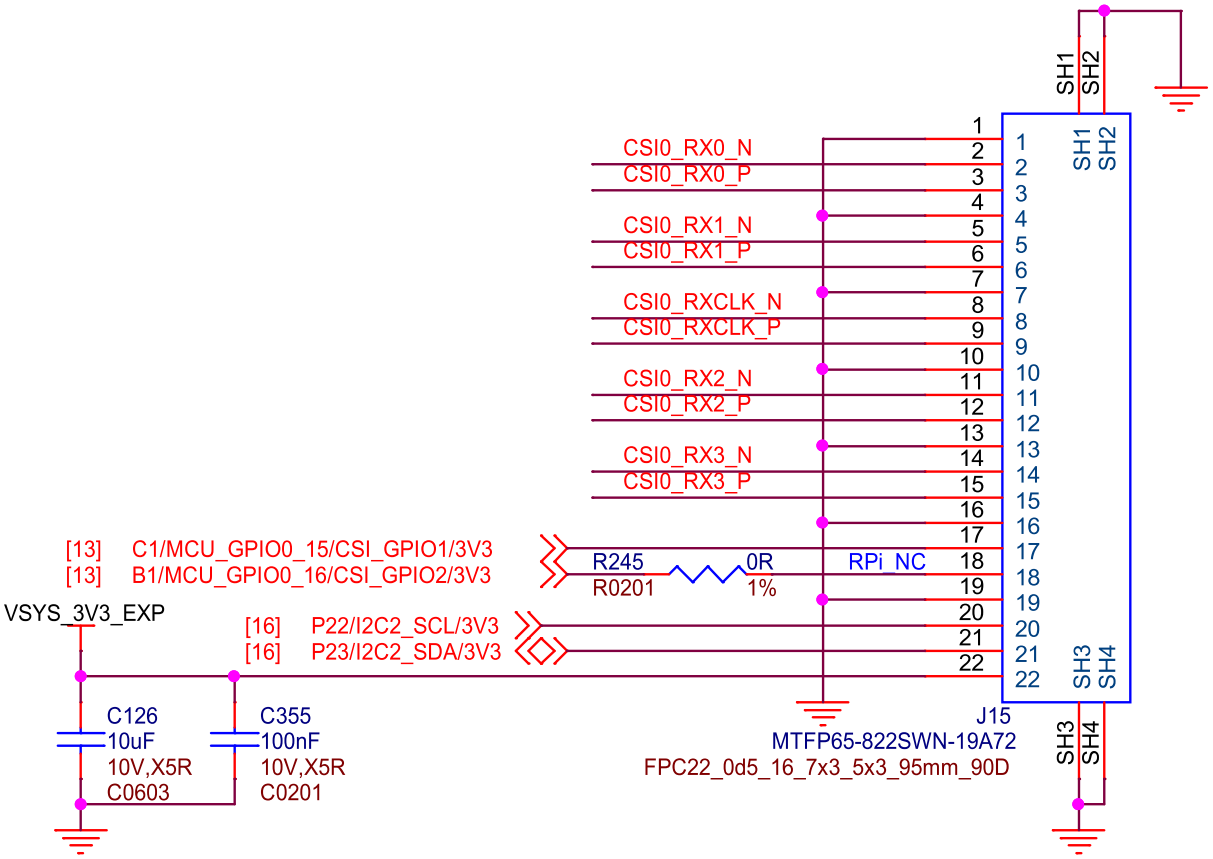


Fig. 3.55: BeagleY-AI RPI CSI

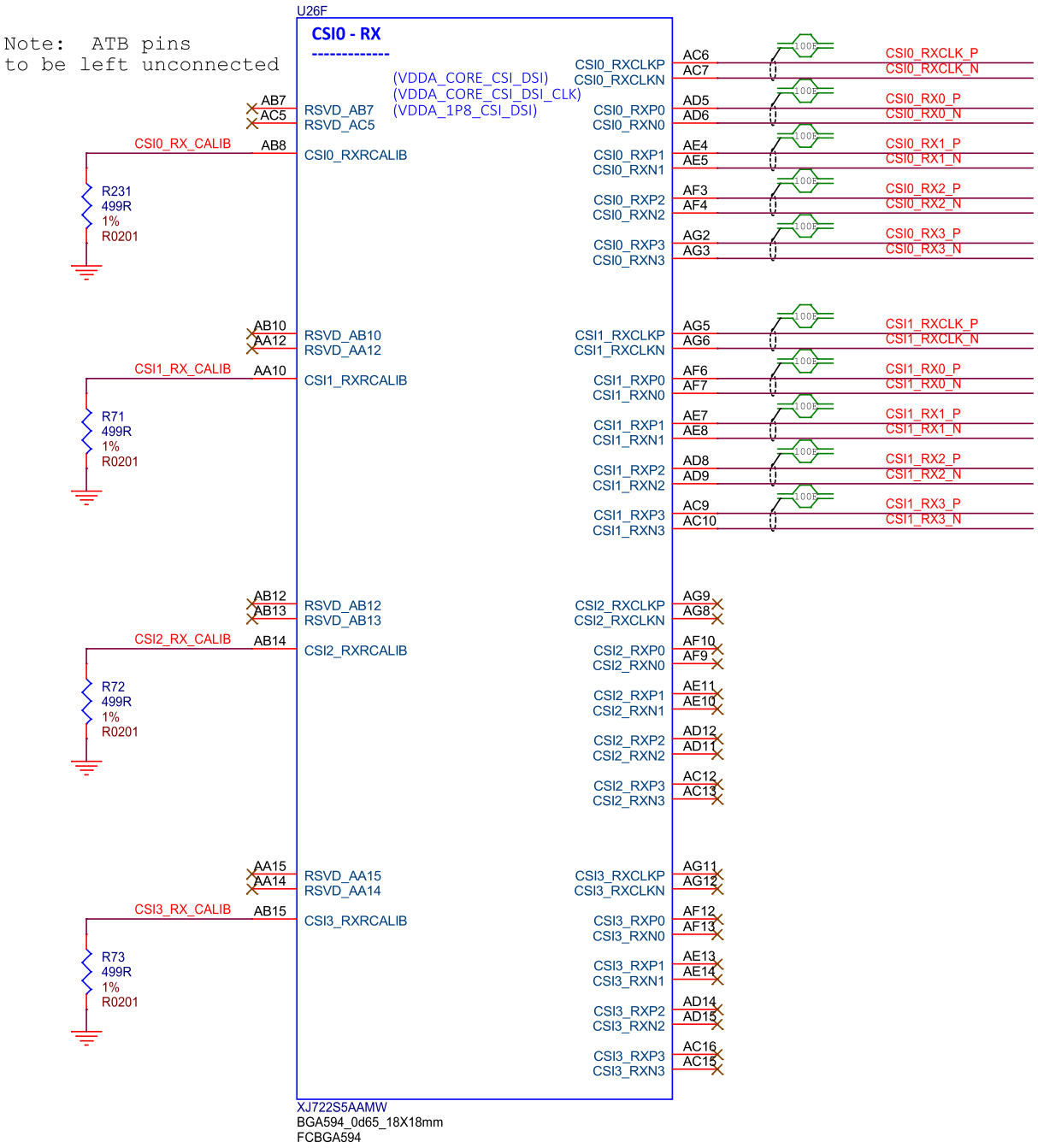


Fig. 3.56: BeagleY-AI SoC CSI1, CSI2, and CSI3

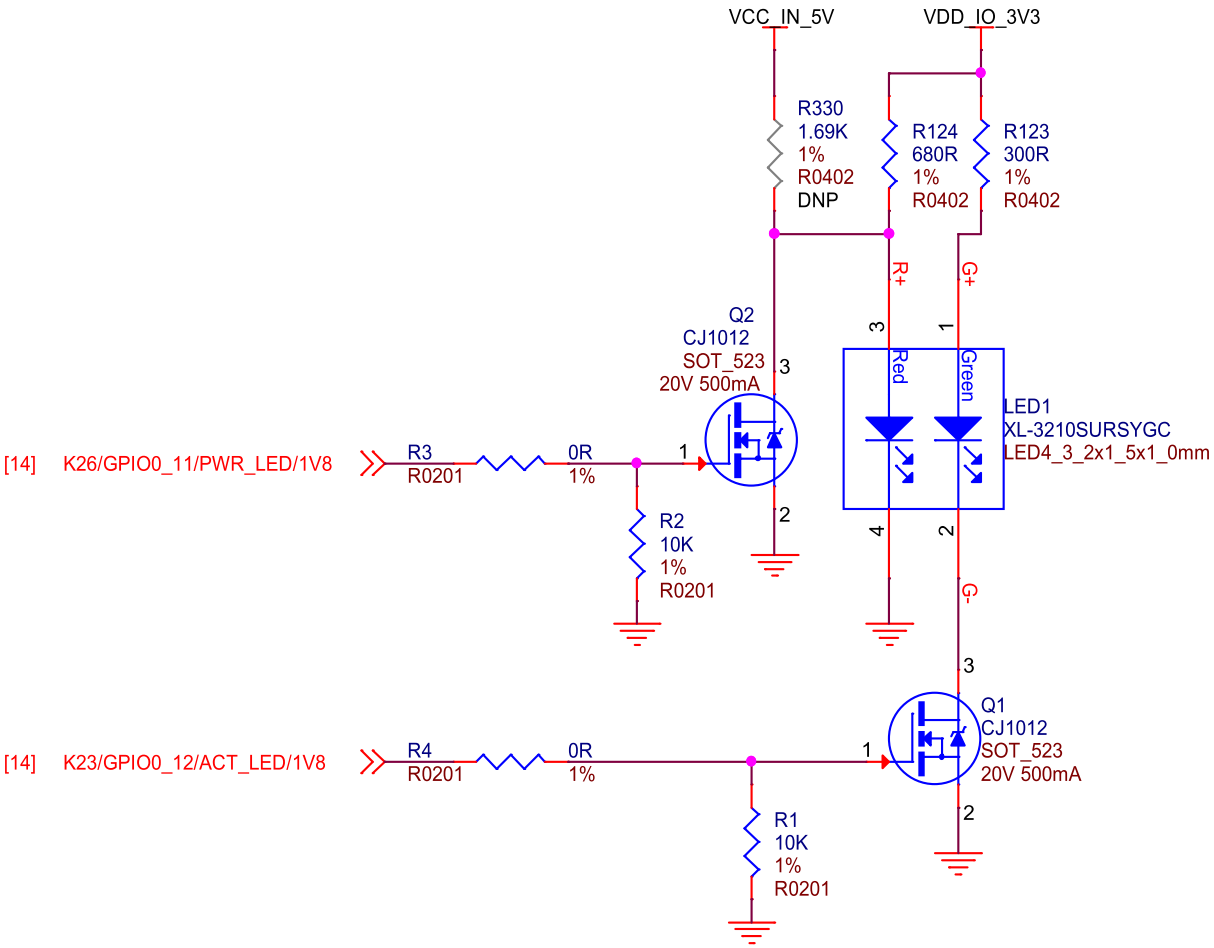


Fig. 3.57: BeagleY-AI LEDs

3.11 Debug Ports

3.11.1 JTAG Tag-Connect

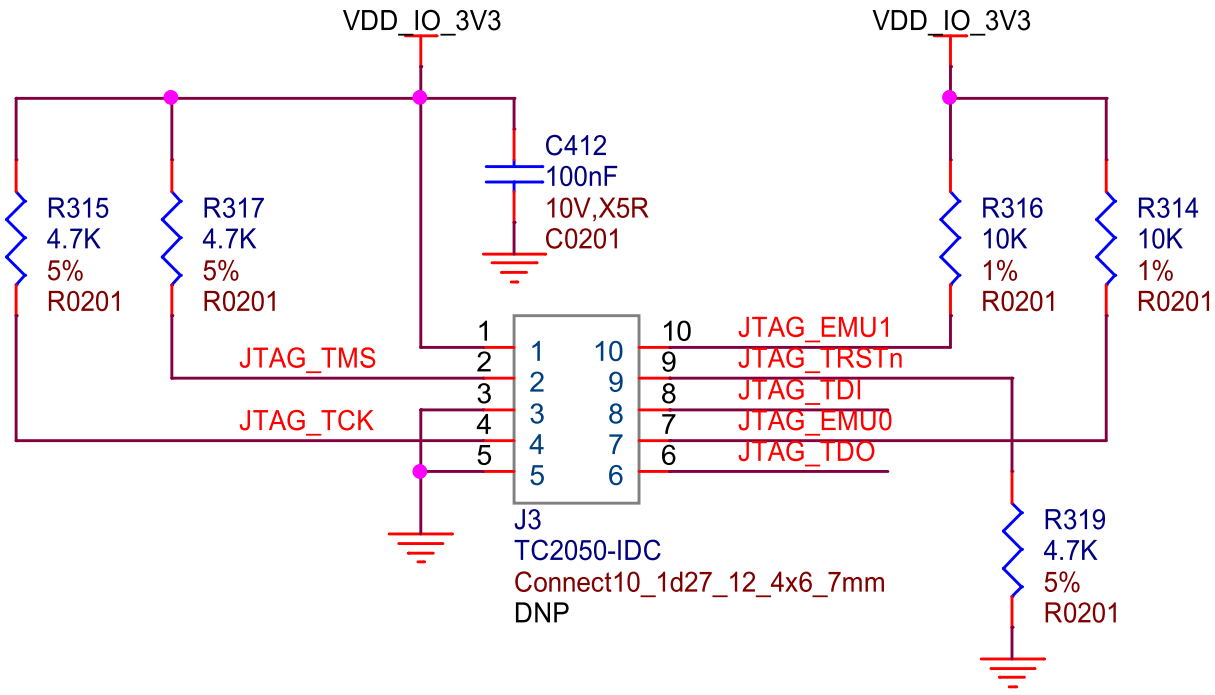


Fig. 3.58: BeagleY-AI Tag-Connect

JTAG is available on the BeagleY-AI via a 10pin Tag-Connect header located on the bottom of the board between the USB 3.0 ports.

Because of the density of the board and tight fit of the USB connectors, the standard retention clip provided by Tag-Connect will not fit. A recommended 3D printable adapter is available on Printables

3.11.2 UART

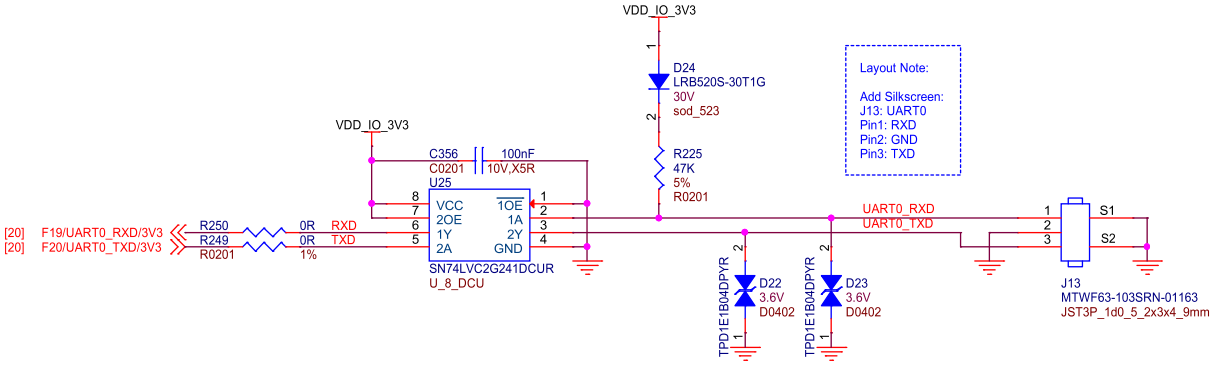


Fig. 3.59: BeagleY-AI debug UART port

By default, BeagleY-AI exposes the UART port used by UBoot & Linux on a Pi Debugger compatible JST 3pin header. The UART port used for debug can also be changed in software to use a UART available on the 40Pin GPIO header.

3.11.3 PMIC NVM Tag-Connect

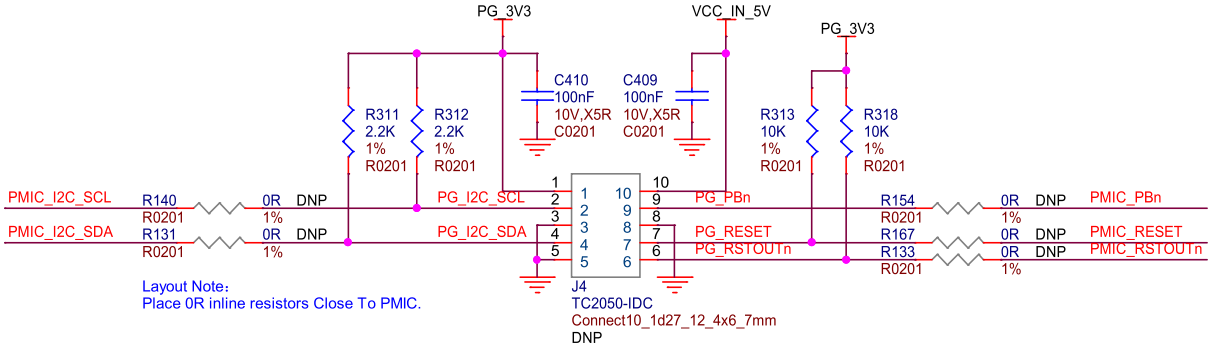


Fig. 3.60: BeagleY-AI PMIC NVM programming interface

A PMIC programming header is present on the BeagleY-AI in the form of a 10pin Tag-Connect header located on the bottom of the board between the Ethernet and USB 3.0 ports. Ensure you do not connect JTAG to this port as the pinout and interface is different. PMIC NVM programming should not be performed unless you know what you're doing. The port is mainly intended for use during manufacturing.

3.12 Miscellaneous

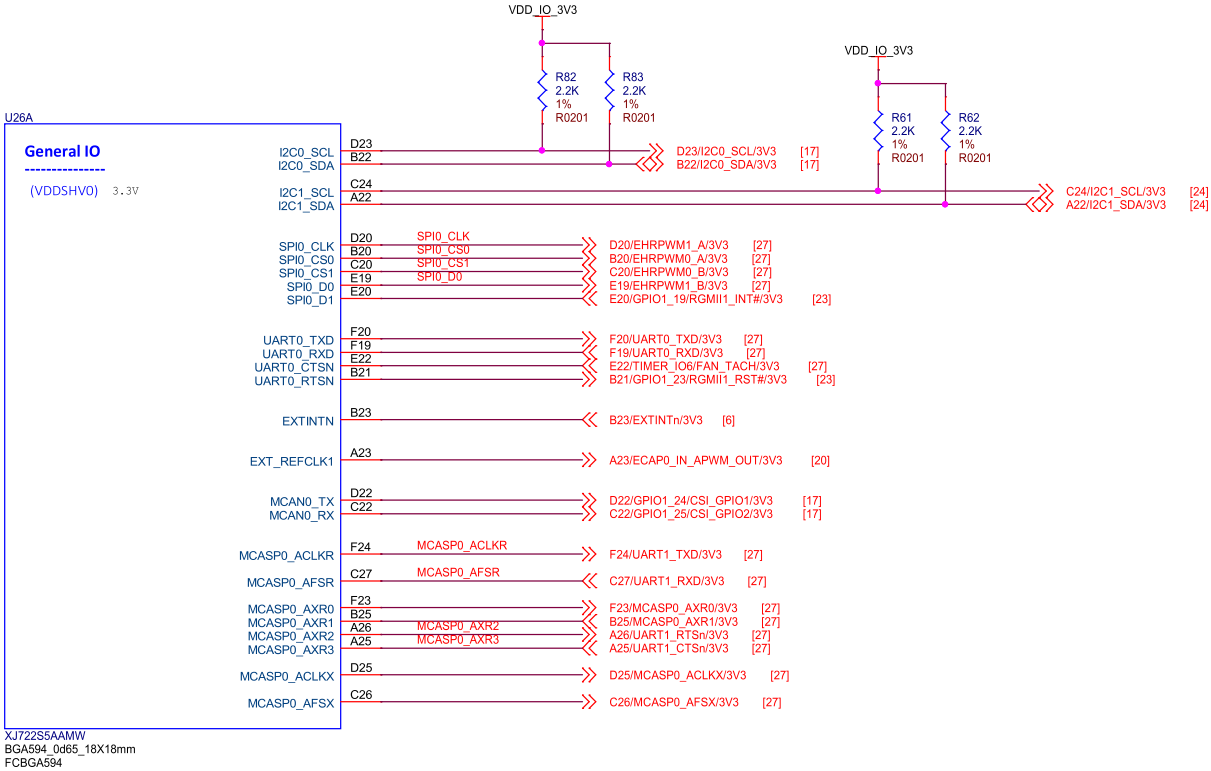


Fig. 3.61: BeagleY-AI general IO

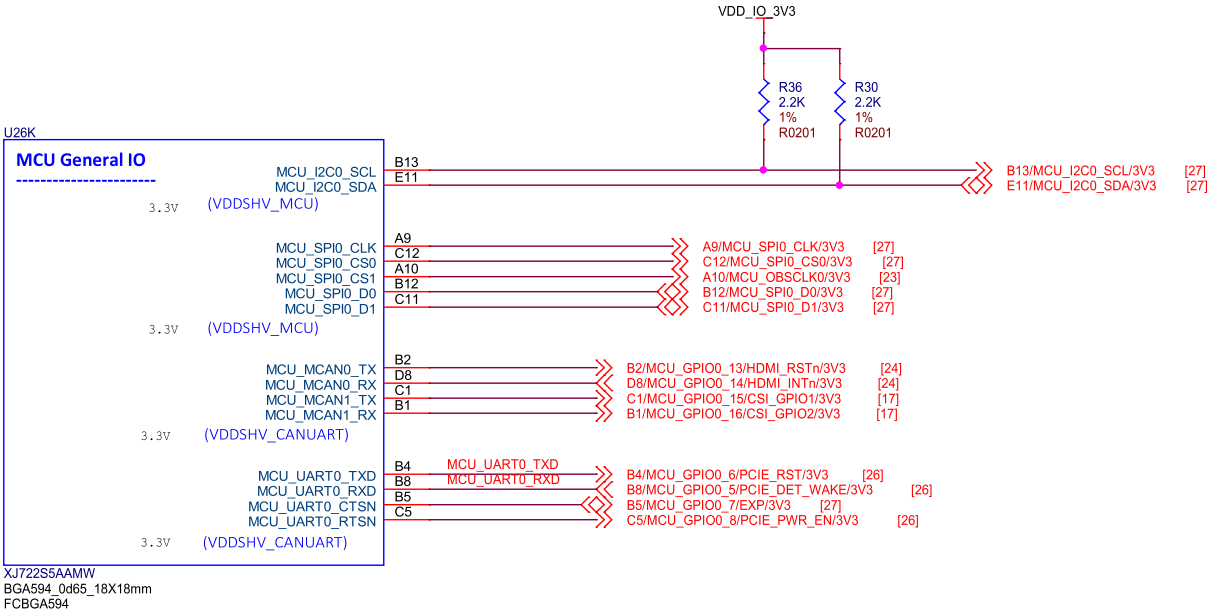


Fig. 3.62: BeagleY-AI MCU general IO

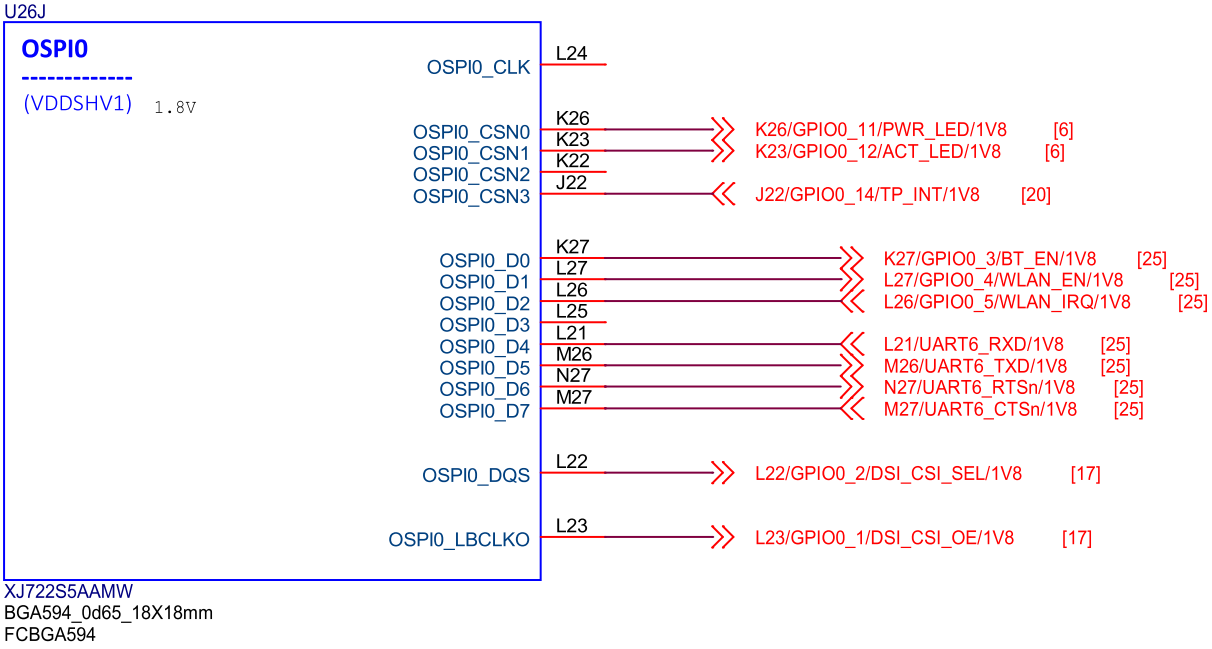


Fig. 3.63: BeagleY-AI SoC OSPI0

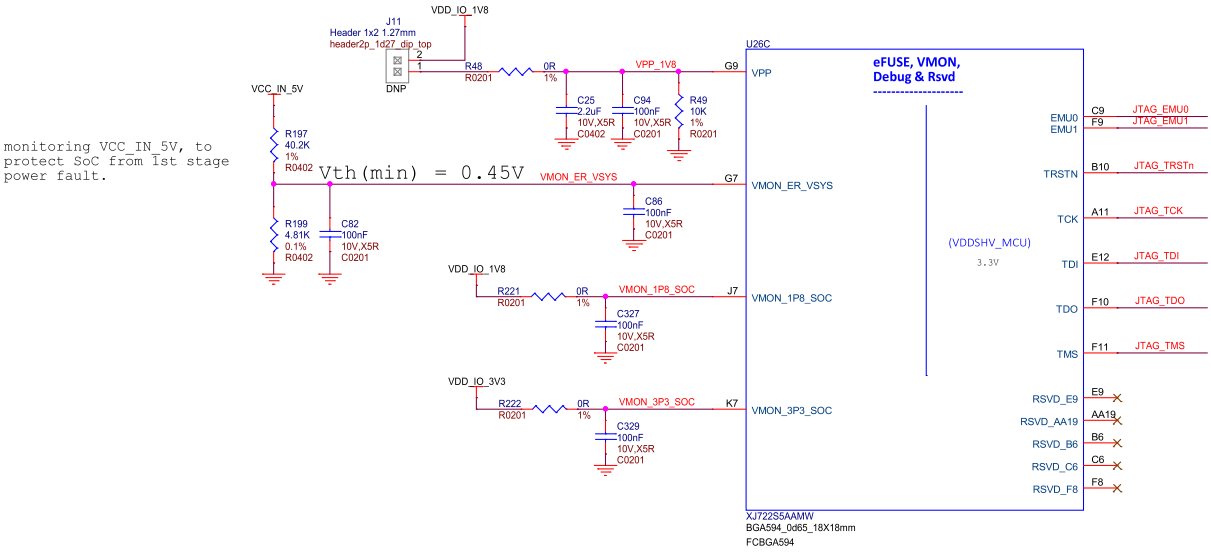


Fig. 3.64: BeagleY-AI SoC eFUSE, VMON, Debug, and RSVD

3.13 Mechanical Specifications

Todo: If there are real design elements, put those here, like clearances and other elements going into design consideration. Summary information should just go in the support page.

Table 3.1: Dimensions & weight

Parameter	Value
Size	85 x 56 x 20 mm
Max heigh	20mm
PCB Size	85 x 56 mm
PCB Layers	14 layers
PCB Thickness	1.6mm
RoHS compliant	Yes
Gross Weight	110 g
Net Weight	50 g

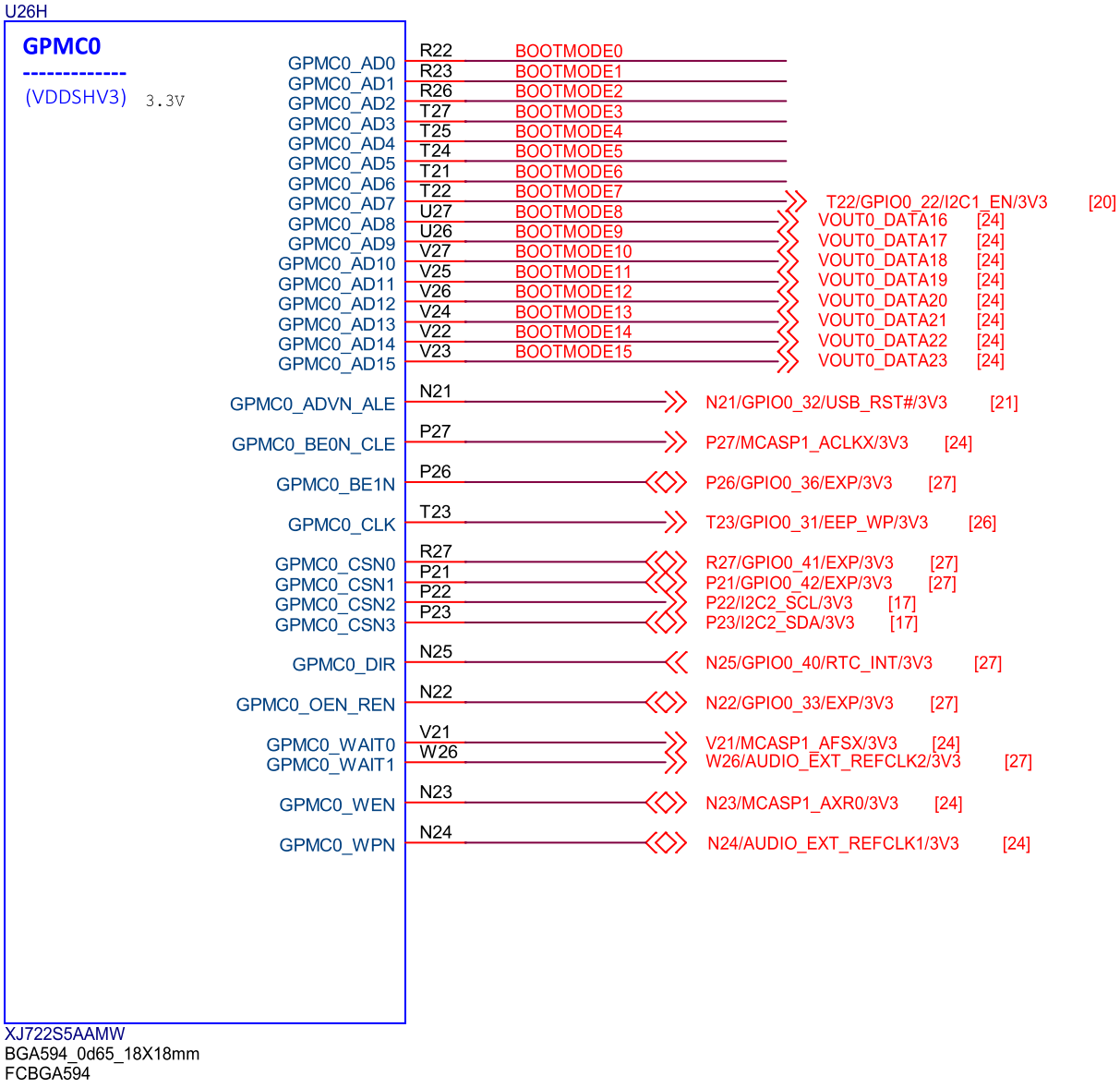


Fig. 3.65: BeagleY-AI SoC GPMC0

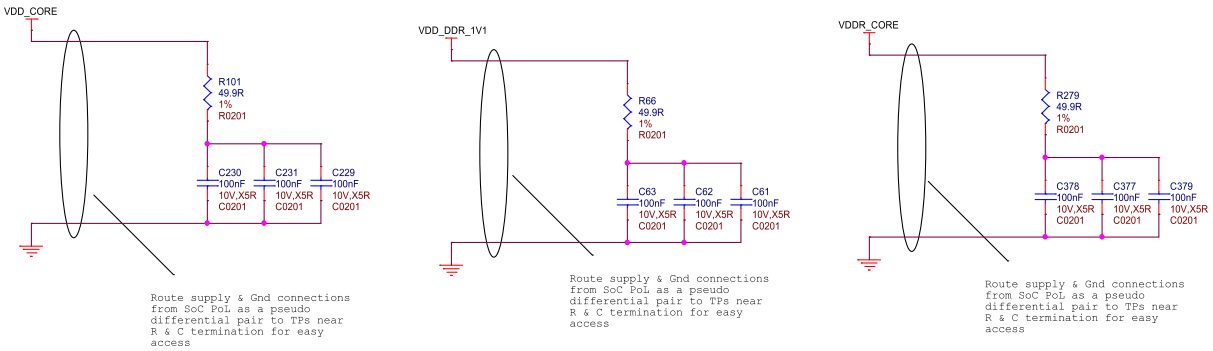


Fig. 3.66: BeagleY-AI SoC supply noise kelvin sensing

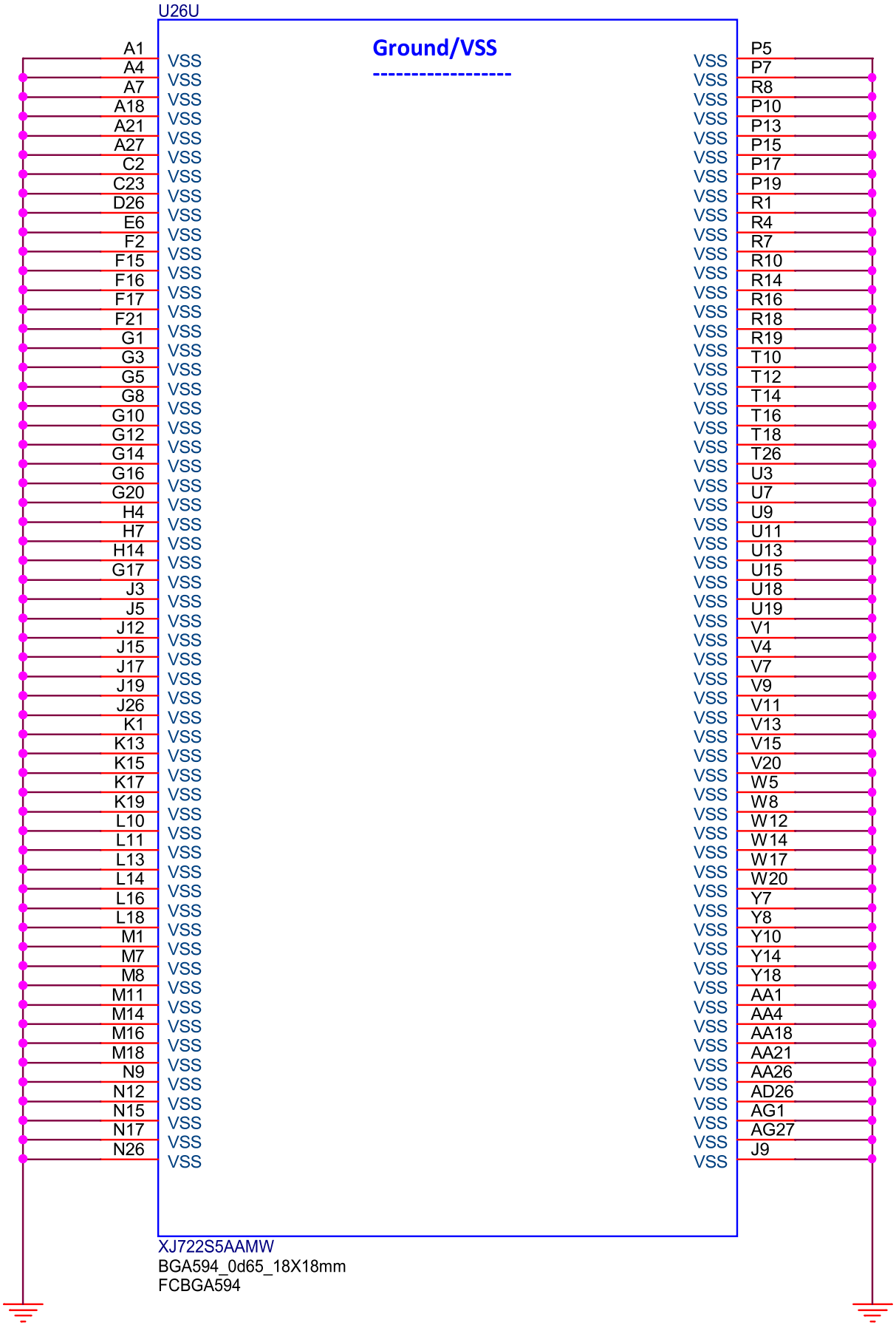


Fig. 3.67: BeagleY-AI SoC ground connections

Chapter 4

Expansion

Todo: Describe how to build expansion hardware for BeagleY-AI. This section is not about using existing add-on hardware.

4.1 PCIe

For software reference, you can see how PCIe is used on NVMe HATs.

4.2 CSI

For software reference, you can see how CSI is used on IMX219-based camera modules.

4.3 RTC

Todo: Remove this. I really don't see how "RTC" falls into our definition of what should be documented in "Expansion".

- beagley-ai-expansion-nvme
- beagley-ai-using-imx219-csi-cameras
- beagley-ai-using-rtc

Chapter 5

Demos and tutorials

Todo: Isn't including "beagley-ai" in the filename superfluous?

demos/beagley-ai-using-gpio demos/beagley-ai-using-pwm demos/beagley-ai-using-rtc demos/beagley-ai-using-i2c-oled-display demos/beagley-ai-using-i2c-adc demos/beagley-ai-pca9685-motor-drivers demos/expansion-nvme demos/connecting-imx219-csi-cameras.rst demos/beagley-ai-using-imx219-csi-cameras demos/beagley-ai-arducam-imx219-v3link-dual-camera-kit demos/beagley-ai-object-detection-tutorial

Chapter 6

Support

All support for BeagleY-AI design is through BeagleBoard.org community at BeagleBoard.org forum.

6.1 Production board boot media

Todo: Add production boot media link in `_static/epilog/production.image` and reference it here.

6.2 Certifications and export control

6.2.1 Export designations

- HS: 8471504090
- US HS: 8543708800
- UPC: 640265311062
- EU HS: 8471707000
- COO: CHINA

6.2.2 Size and weight

- Bare board dimensions: 85 x 56 x 20 mm
- Bare board weight: 50 g
- Full package dimensions: 140 x 100 x 40 mm
- Full package weight: 110g

6.3 Additional documentation

6.3.1 Hardware docs

For any hardware document like schematic diagram PDF, EDA files, issue tracker, and more you can checkout the BeagleY-AI design repository.

6.3.2 Software docs

For BeagleY-AI specific software projects you can checkout all the BeagleY-AI project repositories group.

6.3.3 Support forum

For any additional support you can submit your queries on our forum, https://forum.beagleboard.org/tag/beagley-ai

6.3.4 Pictures

6.4 Change History

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

6.4.1 Board Changes

For all changes, see https://openbeagle.org/beagley-ai/beagley-ai. Versions released into production are noted below.

Table 6.1: BeagleY-AI board change history

Rev	Changes	Date	By