NVIDIA JETSON TX1/TX2 MODULE EEPROM LAYOUT

DA_07991-001 | December 1, 2017

Release 28.2
Application Note



TABLE OF CONTENTS

Jetson TX1/TX2 Module EEPROM Layout..... 3
Configuration of Vendor-Specified MAC Addresses..... 4
Value of the CRC-8 Byte..... 4

Legal Information..... 6

Jetson TX1/TX2 Module EEPROM Layout

This topic describes the layout of the Jetson™ TX1 and TX2 module EEPROM.

All numeric values are little-endian, i.e. the low-addressed byte contains the least significant digit and the high-addressed byte contains the most significant digit.

MAC addresses are also little-endian. For example, for the MAC address 00:04:4b:01:02:03, the low-addressed byte contains 0x03 and high-addressed byte contains 0x00.

The following table describes the layout of the EEPROM.

Bytes	Value	Notes
0-1		Board ID, EEPROM format version.
2-3		Length of board ID data; no longer supported.
4-19		Reserved for future use.
20-49		Product Part Number*, used for asset tracking. A character string in the format `699-8nnnn-1000-vvv r.m`, where: • `699` is a fixed string. • `8` is the board class, which indicates that the Jetson TX1 is a mobile device. • `nnnn` is the board ID. 2180 is the Jetson TX1 board ID 3310 is the Jetson TX2 board ID • `1000` is the Jetson TX1/TX2 SKU. • *vvv* is a three-digit number, the version. • *r* is a single capital letter, the manufacturing major revision. • *M* is a single decimal digit, the manufacturing minor revision. The character data occupies 22 bytes. The field is padded to its full length of 40 bytes with NULs. An example value is `699-82180-1000-000 C.0`. The version number is 000, and the manufacturing major revision is C, and the minor revision number is 0.
50-55		Factory default Wi-Fi MAC address.
56-61		Factory default BT MAC address.
62-67		Secondary Wi-Fi MAC address.
68-73		Factory default Gigabit Ethernet MAC address.
74-88		Asset tracking number, a unique string corresponding to the

Bytes	Value	Notes
		number on the device's identifying sticker. A character string padded with NUL characters.
89–149		**Reserved for future use.**
150–153	'NVCB'	Block signature; stands for "NVIDIA Configuration Block."
154–155	28	Length of this struct from block signature to end. Value is subject to change.
156–157	'M1'	Format of following MAC address data: "MAC address, field format version 1."
158–159	0x0000	Version.
160–165		Vendor-specified Wi-Fi MAC address.
166–171		Vendor-specified BT MAC address.
172–177		Vendor-specified Gigabit Ethernet MAC address. Last field in the struct whose length is in bytes 154-155.
178–254		**Reserved for future use.**
255		CRC-8 computed for bytes 0–254.
Note		
* This part number is subject to change without notice. However, in the event of a major change with customer impact, a PCN (Product Change Notification) will be issued.		

Configuration of Vendor-Specified MAC Addresses

To configure the vendor-specified MAC addresses, use the following procedure.

If either of the "Verify" steps does not produce the expected result, the EEPROM has been corrupted or the device is malfunctioning. Identify and correct the problem before you proceed.

1. Read the EEPROM data from the 256-byte block at I2C bus 2, address 0x50.
2. Verify the EEPROM's CRC-8 checksum. Compute the checksum for bytes 0-254, using the procedure in **Value of the CRC-8 Byte**. The computed checksum should match the value in byte 255.
3. Verify that bytes 150-153 and 156-157 contain the signature values shown in the table above.
4. Update the MAC addresses in bytes 160-177 (see table). Remember that the MAC addresses are stored in little-endian order, the reverse of normal reading order.
5. Recompute the checksum, using the updated MAC addresses. Store the new checksum in byte 255.

Value of the CRC-8 Byte

The CRC is a single byte stored in byte 255, the last byte of EEPROM. It is computed using the CRC-8 algorithm in the following sample code.

```
def AddToCRC(b, crc):
```

```
b2 = b
if (b < 0):
b2 = b + 256
for i in range(8):
odd = ((b2^crc) & 1) == 1
crc >>= 1
b2 >>= 1
if (odd):
crc ^= 0x8C # This means crc ^= 140.
return crc
```

For each byte `B` of EEPROM content:

```
crc = AddToCRC(B, crc)
```

Legal Information

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OR CONDITION OF TITLE, MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE AND ON-INFRINGEMENT, ARE HEREBY EXCLUDED TO THE MAXIMUM EXTENT PERMITTED BY LAW.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, CUDA, Tegra, and Vibrante are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

The Android robot is reproduced or modified from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License.

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

ARM, AMBA, and ARM Powered are registered trademarks of ARM Limited. Cortex, MPCore and Mali are trademarks of ARM Limited. All other brands or product names are the property of their respective holders. "ARM" is used to represent ARM Holdings plc; its operating company ARM Limited; and the regional subsidiaries ARM Inc.; ARM KK; ARM Korea Limited.; ARM Taiwan Limited; ARM France SAS; ARM Consulting (Shanghai) Co. Ltd.; ARM Germany GmbH; ARM Embedded Technologies Pvt. Ltd.; ARM Norway, AS and ARM Sweden AB.

Copyright

© 2017 by NVIDIA Corporation. All rights reserved