

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		<p>Product Part Number, used for asset tracking. A character string in the format 699-8nnnn-1000-vvv r.m, where:</p> <ul style="list-style-type: none">• 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. <p>The character data occupies 22 bytes. The field is padded to its full length of 40 bytes with NULs.</p> <p>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.</p>
50-55		Factory default Wi-Fi MAC address.
56-61		Factory default BT MAC address.
62-67		Secondary Wi-Fi MAC address.

Bytes	Value	Notes
68-73		Factory default Gigabit Ethernet MAC address.
74-88		Asset tracking number, a unique string corresponding to the 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.

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)
```