

Embedded Artists i.MX Platform Specific

Table of Content

1. Which Murata modules will work well with my Embedded Artists i.MX EVK?	1
2. How do I decide the correct DTB file to use for my Murata module?	2
3. I followed the Embedded Artists Getting Started Guide to bring up my Embedded Artists i.MX platform with Murata module, but the Wi-Fi interface is not showing up. What can be wrong?	3
4. Can I use a mainline Linux kernel in my Embedded Artists i.MX image?	4
5. Which Murata InterConnect module do I need to connect my Murata Wi-Fi/BT module with the Embedded Artists i.MX EVK?	5
6. I just got my Embedded Artists EVK and Murata module. How do I start?	6
7. How do I put my Embedded Artists EVK into OTG mode to flash image?	7
8. Can I transfer file over the network from my Windows host to my Embedded Artists i.MX7D EVK?	8
9. How can I intercept the Bluetooth UART communication on Embedded Artists i.MX7D EVK, for debug purpose?	12
10. Where can I find latest Embedded Artists documents to get started?.....	16
11. What are some of the advanced features provided by the Embedded Artists' i.MX 6/7/8 EVKs?	17
12. What are some of the advanced features provided by the Embedded Artists' i.MX RT EVKs?	18
13. How is the Embedded Artists' i.MX RT EVK platform powered?	19
14. How to flash Embedded Artists' i.MX8MQ COM board using UUU?	20

1. Which Murata modules will work well with my Embedded Artists i.MX EVK?

Check the table below:

	6UL	6SX	6DL	6Q	7D/DM	7ULP	8MQ	8MMini	RT1052	RT1062
1DX	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
1MW	Y	Y	Y	Y	Y	Y	Y	Y	N	N
1LV	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
1CX	N	Y	Y	Y	Y	N	Y	Y	N	N

[Back to Table of Content](#)

2. How do I decide the correct DTB file to use for my Murata module?

The DTB file to use depends on the target board and the M.2 interface to be used. Embedded Artists EVKs support both SDIO and PCIe on the M.2. Refer to the tables below for the correct dtb file name and the interface type to be used for each supported Murata module.

Target EVK	M.2 SDIO interface	M.2 PCIe interface
i.MX 6SX	imx6sxea-com-kit_v2.dtb	imx6sxea-com-kit_v2-pcie.dtb
i.MX 6Q	imx6qea-com-kit_v2.dtb	imx6qea-com-kit_v2-pcie.dtb
i.MX 6DL	imx6dlea-com-kit_v2.dtb	imx6dlea-com-kit_v2-pcie.dtb
i.MX 6UL	imx6ulea-com-kit_v2.dtb	NOT SUPPORTED
i.MX 7D	imx7dea-com-kit_v2.dtb	imx7dea-com-kit_v2-pcie.dtb
i.MX 7DM	imx7dea-ucom-kit_v2.dtb	imx7dea-ucom-kit_v2-pcie.dtb
i.MX 7ULP	imx7ulpea-ucom-kit_v2.dtb	NOT SUPPORTED
i.MX 8MQ	fsl-imx8mq-ea-com-kit_v2.dts	fsl-imx8mq-ea-com-kit_v2-pcie.dts
i.MX 8MMini	fsl-imx8mm-ea-ucom-kit_v2.dtb	fsl-imx8mm-ea-ucom-kit_v2-pcie.dtb

Murata module	Interface type
1DX	M.2 SDIO
1MW	M.2 SDIO
1LV	M.2 SDIO
1CX	M.2 PCIe

[Back to Table of Content](#)

3. I followed the Embedded Artists Getting Started Guide to bring up my Embedded Artists i.MX platform with Murata module, but the Wi-Fi interface is not showing up. What can be wrong?

Check the following:

- You are using the correct dtb file. The dtb files are different for PCIe based modules and SDIO based modules.
- You are using an image that supports Wi-Fi driver (e.g. “core-image-minimal” does not).
- In case you are trying to integrate a new module, make sure the Wi-Fi firmware, CLM Blob and NVRAM files are present in correct locations –
- The WLAN firmware should be in /lib/firmware/brcm/ folder. The naming convention for the firmware is:
“brcmfmac”+<CYW number>+<“-sdio” or “-pcie”>+“.bin”.
(e.g. brcmfmac43455-sdio.bin for 1MW)
- The CLM blob file should be in /lib/firmware/brcm/ folder. The naming convention for the CLM blob is:
“brcmfmac”+<CYW number>+<“-sdio” or “-pcie”>+“.clm_blob”.
(e.g. brcmfmac43455-sdio.clm_blob For 1MW)
- The NVRAM file should be in /lib/firmware/brcm/ folder.

[Back to Table of Content](#)

4. Can I use a mainline Linux kernel in my Embedded Artists i.MX image?

Murata / Embedded Artists do not officially support any mainline Linux kernels. It is highly recommended that you use one of the supported Linux kernels. Currently, Embedded Artists' only release is based on NXP's Linux 4.14.78 BSP and this is the only Linux version supported at this time.

[Back to Table of Content](#)

5. Which Murata InterConnect module do I need to connect my Murata Wi-Fi/BT module with the Embedded Artists i.MX EVK?

All Embedded Artists EVKs some with on-board M.2 interface. Hence no Murata InterConnect adapters are required.

[Back to Table of Content](#)

6. I just got my Embedded Artists EVK and Murata module. How do I start?

Use the Embedded Artists Getting Started documents to bring up the EVK with Murata module.

For [i.MX 6/7/8 EVKs](#)

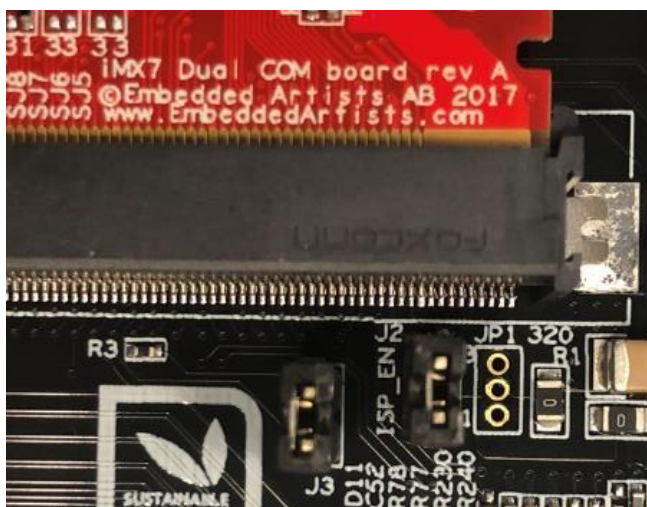
For [i.MX RT EVKs](#)

[Back to Table of Content](#)

7. How do I put my Embedded Artists EVK into OTG mode to flash image?

There are 4 steps to put the Embedded Artists EVK into OTG mode so the image can be flashed:

- i. Close the jumper J2 on the carrier board.



- ii. Connect the J11 port of the EVK to the host PC via a Micro-B to A USB cable. The host PC will use this connection to flash the image (via either MFGtool or uuu tool).
- iii. (Optional) Connect the J16 port of the EVK to the host PC via a Micro-B to A USB cable. The host PC will use this connection for console input/output (via a terminal emulator software, such as minicom or Tera Term).
- iv. Power on the EVK. It is now ready to be flashed.

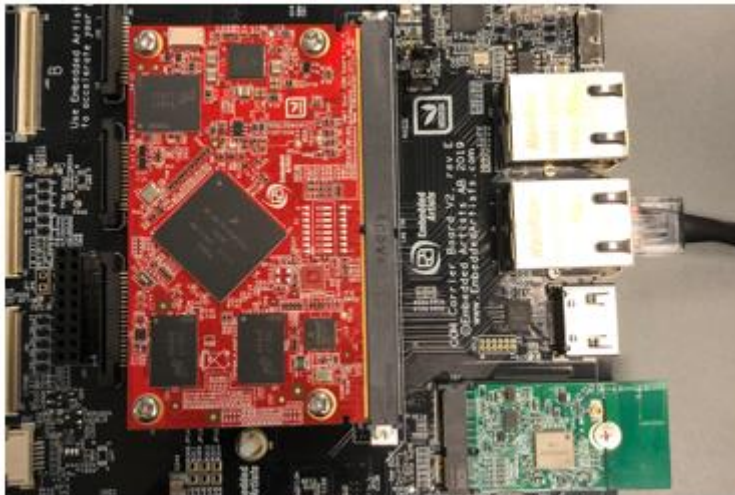
[Back to Table of Content](#)

8. Can I transfer file over the network from my Windows host to my Embedded Artists i.MX7D EVK?

Yes, you can set up SCP to transfer file easily between the two machines.
Follow the steps given below:

On the Embedded Artists i.MX 7D EVK:

- i. Connect the EVK to the router via an Ethernet cable. Use any one of the Ethernet ports on the EVK.



- ii. Open a terminal to the EVK from the host PC (connected via USB cable on J16 port)
- iii. Issue commands to the EVK to bring up the Ethernet interface and note the IP address assigned:

```
# ifconfig eth0 up
```

```
# ifconfig
```

```

COM10 - Tera Term VT
File Edit Setup Control Window Help
[ OK ] Started Connection service.
Starting Hostname Service...
Starting WPA supplicant...
[ OK ] Started Updates psplash to basic.
[ OK ] Reached target Sound Card.
[ OK ] Started WPA supplicant.
[ OK ] Reached target Network.
[ OK ] Started Update psplash to network.
Starting Terminate Ppplash Boot Screen...
[ FAILED ] Failed to start Terminate Ppplash Boot Screen.
See 'systemctl status psplash-quit.service' for details.
[ OK ] Started Hostname Service.
[ OK ] Reached target Multi-User System.
Starting Update UIMP about System Runlevel Changes...
[ OK ] Started Update UIMP about System Runlevel Changes.

MXP i.MX Release Distro 4.14-sumo inx7dea-com ttyxc0

inx7dea-com login: fec 30be0000.ethernet eth0: Link is Up - 1Gbps/Full
IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready

MXP i.MX Release Distro 4.14-sumo inx7dea-com ttyxc0

inx7dea-com login: root
Password:
root@inx7dea-com:~# ifconfig eth0 up
root@inx7dea-com:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr EA:D4:15:FF:65:80
          inet addr:192.168.1.123  Bcast:192.168.1.255  Mask:255.255.255
          inet6 addr:fe80::e001:13ff:feff:6580/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:118 errors:0 dropped:0 overruns:0 frame:0
          TX packets:156 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12077 (11.7 KiB)  TX bytes:19570 (19.1 KiB)

root@inx7dea-com:~# nano /etc/ssh/sshd_config
root@inx7dea-com:~#

```

- iv. Open the configuration file for the SSH server:
`# nano /etc/ssh/sshd_config`
- v. Add line “PermitRootLogin yes” in the file, save and exit.

```

COM10 - Tera Term VT
File Edit Setup Control Window Help
GNU nano 2.9.3 /etc/ssh/sshd_config Modified
# OpenBSD: sshd_config,v 1.80 2008/07/02 02:24:18 djm Exp #
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options change a
# default value.
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
PermitRootLogin Yes
# The default requires explicit activation of protocol 1

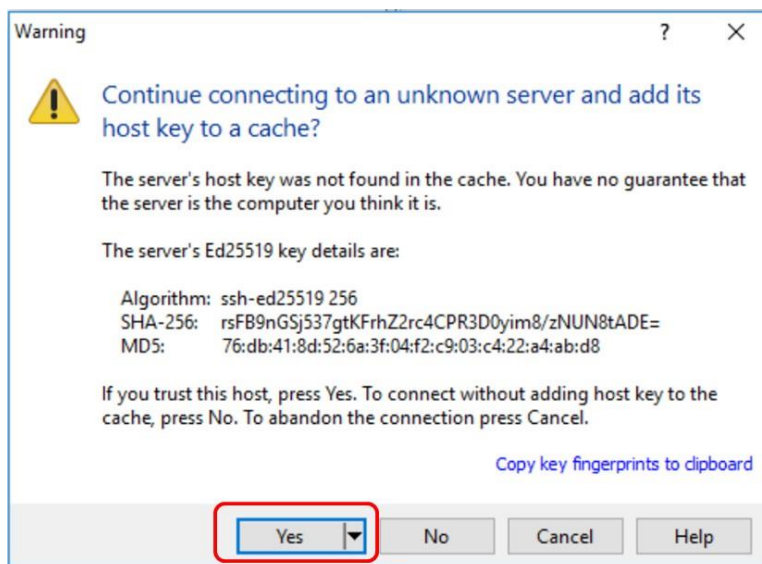
Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos
Exit      Read File  Replace   Uncut Text To Spell  Go To Line

```

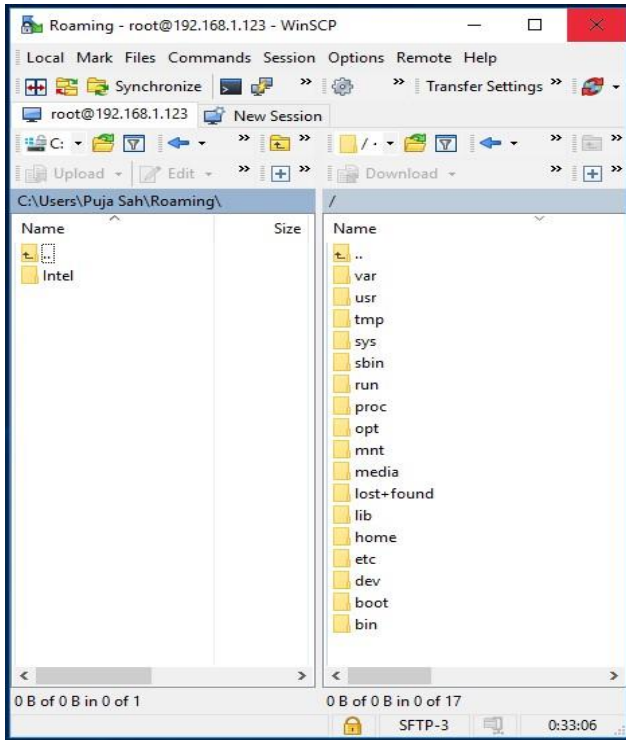
- vi. Restart the server:
`# /etc/init.d/sshd restart`

On the Windows host:

- i. Download and install **WinSCP**
- ii. Connect the host PC to the router via an Ethernet cable.
- iii. Start WinSCP and connect to the EVK:
 - a. IP Address: IP address of the EVK noted in step iii.
 - b. Username: root
 - c. Password: pass
- iv. Click 'Yes' on the pop up screen.



- v. Connection should be established. You can copy files between the two machines now.



[Back to Table of Content](#)

9. How can I intercept the Bluetooth UART communication on Embedded Artists i.MX7D EVK, for debug purpose?

The Embedded Artists EVK M.2 interface allows interception of the Bluetooth UART communication directly from a host PC by disabling the Linux access to the UART. You would need an FTDI UART to USB bridge cable and the CyBlueTool developed by Cypress.

You can download the CyBlueTool from [Windows installed](#) or [Linux installer](#), and the [latest user guide](#).

Steps:

- i. Open a terminal to the EVK from the host PC.
- ii. Power on the board and interrupt the boot process by pressing any key.
- iii. Run the command (single line):

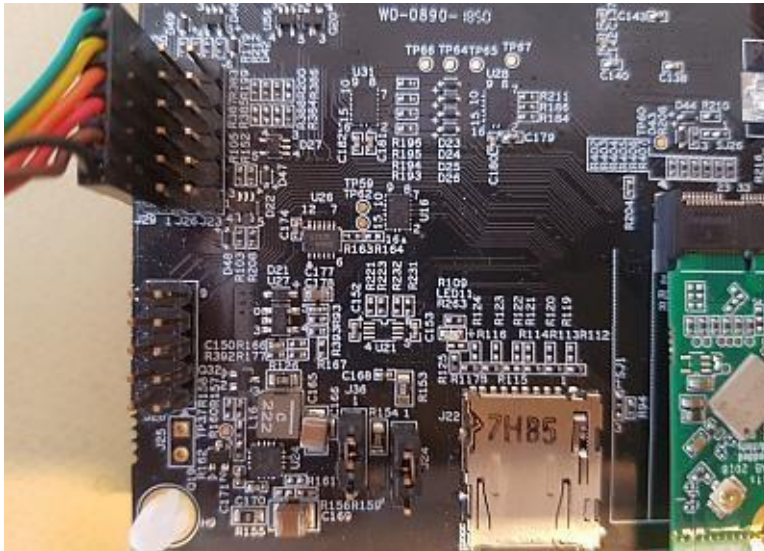

```
=> setenv cmd_custom fdt set serial1 status disabled\;fdt set /modem-reset status disabled
```

You will need to use the correct serial number. Refer to the table below:

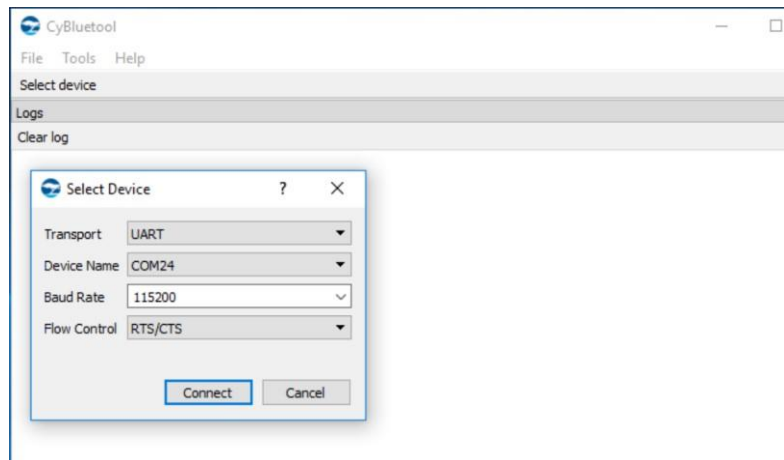
COM board	Serial name
i.MX 6SX	serial1
i.MX 6Q	serial4
i.MX 6DL	serial4
i.MX 6UL	serial1
i.MX 7D	serial1
i.MX 7DM	serial1
i.MX 8MQ	serial1

- i. Save the configuration:

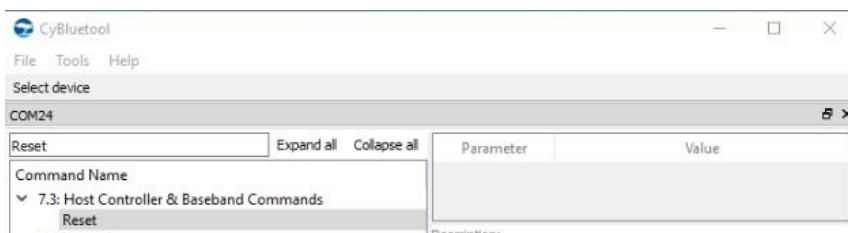

```
=> saveenv
```
- ii. Power off the EVK.
- iii. Connect an FTDI cable in connector J29 of the EVK, the other end (USB) connected to the host PC.



- iv. Power on the board and let it boot into Linux.
- v. Run the following commands to enable Bluetooth:
echo 496 > /sys/class/gpio/export
echo high > /sys/class/gpio/gpio496/direction
- ix. For i.MX 6Q and i.MX 6DL EVKs ONLY, run the following commands.
Otherwise skip this step.
echo 165 > /sys/class/gpio/export
echo low > /sys/class/gpio/gpio165/direction
- x. On the host PC, run the CyBlueTool application.
- xi. Click Select device, set the following and click 'Connect':
 - a. Transport: Select "UART".
 - b. Device name: Select the COM port of the EVK.
 - c. Baud Rate: Enter "115200" (standard baud rate).
 - d. Flow Control: Select "RTS/CTS".



- xii. Double click on 'Reset' to issue the command. Do this twice.



- xiii. Check the Log window to confirm DUT reset is successful both instances. Host PC can now communicate with the DUT using HCI controls.

```
Logs
Clear log
COM24

05/06/19 16:29:37.845 com24 c> Reset
    HCI Command
    com24@115200
    [03 0C 00]
    opcode = 0x0C03 (3075, "Reset")

05/06/19 16:29:37.876 com24 <c Reset
    HCI Command Complete Event
    com24@115200
    [0E 04]: 01 03 0C 00
    event = 0x0E (14, "Command Complete")
    Num_HCI_Command_Packets = 0x1 (1)
    Command_Opcode = 0xC03 (3075, "Reset")
    Status = 0x0 (0, "Success", "Success")

05/06/19 16:29:42.134 com24 c> Reset
    HCI Command
    com24@115200
    [03 0C 00]
    opcode = 0x0C03 (3075, "Reset")

05/06/19 16:29:42.141 com24 <c Reset
    HCI Command Complete Event
    com24@115200
    [0E 04]: 01 03 0C 00
    event = 0x0E (14, "Command Complete")
    Num_HCI_Command_Packets = 0x1 (1)
    Command_Opcode = 0xC03 (3075, "Reset")
    Status = 0x0 (0, "Success", "Success")
```

xiv. To return to normal use of UART, restart board, interrupt the boot and run the commands:

=> setenv cmd_custom

=> saveenv

[Back to Table of Content](#)

10. Where can I find latest Embedded Artists documents to get started?

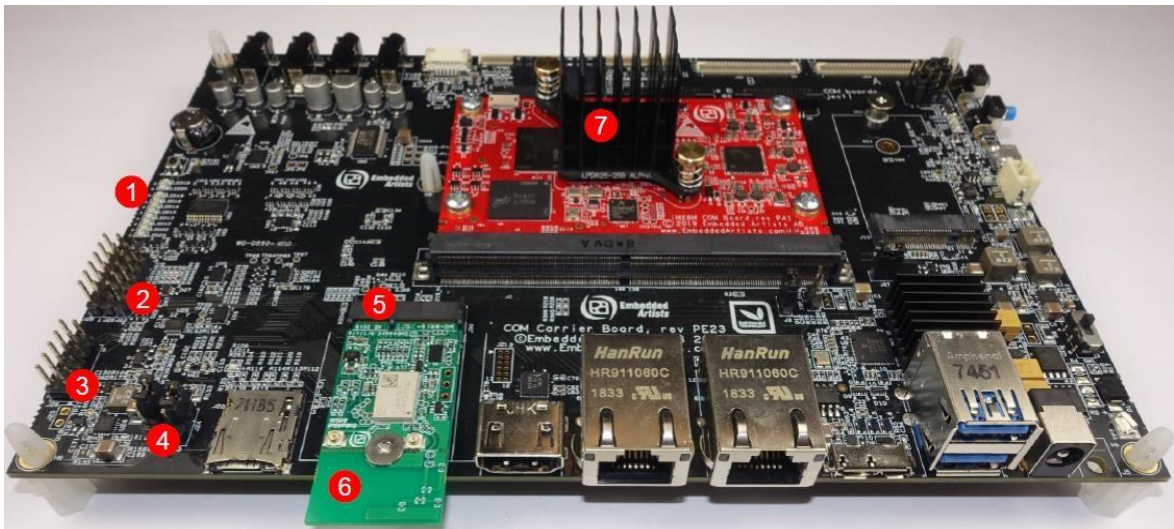
The Embedded Artists Getting [Started guide for i.MX 6/7/8 EVKs](#)

The Embedded Artists Getting [Started guide for i.MX RT EVKs](#)

[Back to Table of Content](#)

11. What are some of the advanced features provided by the Embedded Artists' i.MX 6/7/8 EVKs?

Here are some of the advanced features provided by the Embedded Artists' i.MX 6/7/8 EVKs:

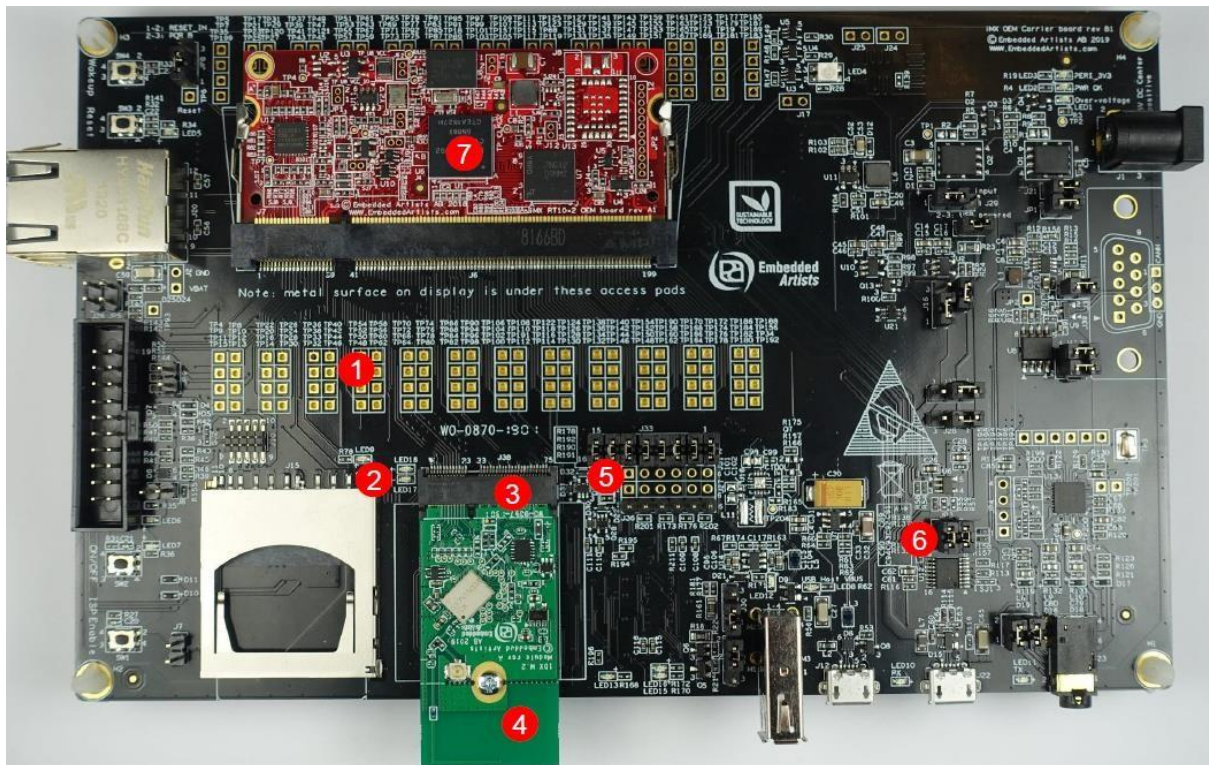


1	Operational LEDs: indicate current mode (VBAT/VDDIO/WLAN-PCIe or WLAN-SDIO etc.).
2	BT-UART and WLAN/BT TX/RX Debug UART pinout: FTDI TTL-232R USB-to-UART cable can be used to bring out the signals.
3	WLAN JTAG and External Coexistence Signalling.
4	VBAT/VDDIO jumpers: allow precise power (voltage/current) measurement.
5	M.2 interface: supports WLAN-PCIe/SDIO, BT-UART/PCM/I2S, WLAN/BT/Coexist control lines, and debug pinouts.
6	Embedded Artists 1LV (CYW43012) M.2 EVB: provides direct interconnect via M.2 connector. Can be easily switched with other options (1DX, 1MW, 1CX).
7	i.MX 8MQ COM (Computer-On-Module): can be easily switched with other options (i.MX 7ULP, i.MX 6Q, i.MX 6UL etc.).

[Back to Table of Content](#)

12. What are some of the advanced features provided by the Embedded Artists' i.MX RT EVKs?

Here are some of the advanced features provided by the Embedded Artists' i.MX RT EVKs:



1	All interface pins from i.MX RT brought out, except high speed SDIO: can be probed on M.2 EVB directly.
2	Operational LEDs: indicate power applied to M.2 EVB.
3	M.2 interface: supports WLAN-PCIe/SDIO, BT-UART/PCM/I2S, WLAN/BT/Coexist control lines, and debug pinouts.
4	Embedded Artists 1DX (CYW4343W) M.2 EVB: provides direct interconnect via M.2 connector. Can be easily switched with other options (1LV, 1MW).
5	Access to Bluetooth UART and debug signals.
6	VBAT/VDDIO jumpers to M.2 EVB: allow precise power (voltage/current) measurement.
7	i.MX RT 1062 COM (Computer-On-Module): can be easily switched with other options (i.MX RT 1062, i.MX RT 1052).

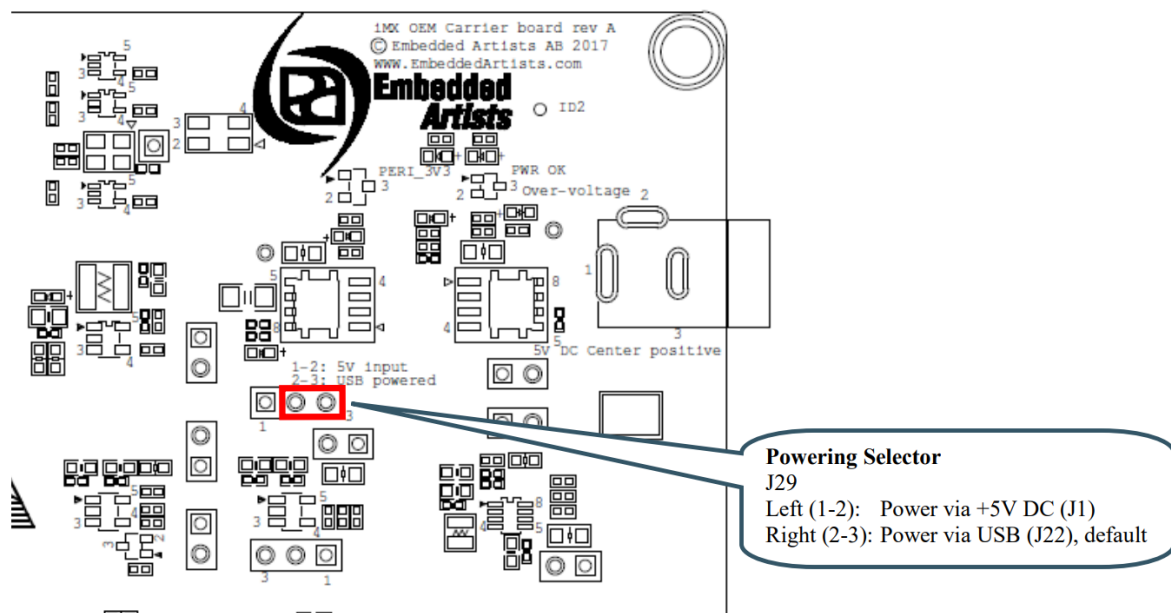
[Back to Table of Content](#)

13. How is the Embedded Artists' i.MX RT EVK platform powered?

The platform is normally powered from a PC/laptop via the included USB cable (micro-B to A) connected to the USB port (J22). However, note that not all PC/laptops can provide the needed current. A powered USB hub can be used in this case, or an external power supply.

A separate power supply is always needed in stand-alone situations or when running USB Host applications that require powering external USB devices. If USB Host is not needed (or needed with very low external current consumption), a 5V DC/1A power supply (5W) is sufficient. If USB Host is used with high external current consumption, a 5V DC/2-2.5A power supply (10-12.5W) is needed. A 2.1 mm standard power plug is provided on the EVK to power the board (J1). Centre position is the positive terminal.

It is possible to have both the USB cable and external powering connected to the board at the same time but only one of them can be used at any time. The Powering selector jumper J29 selects either +5V DC powering via J1 or USB powering via J22.



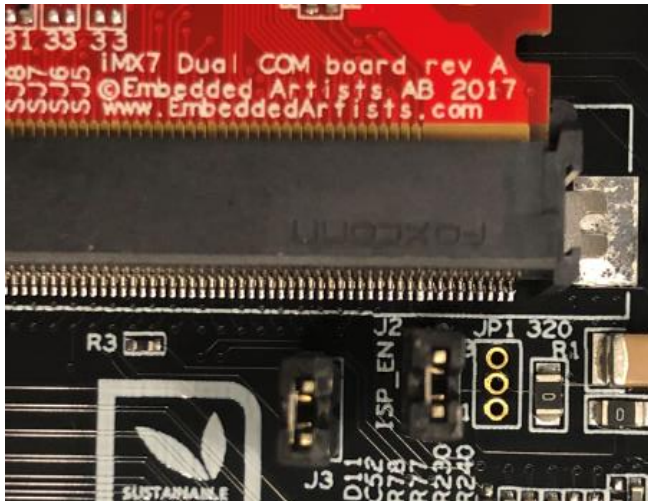
Credit: Image taken from Embedded Artists' [i.MX RT User's Guide](#).

[Back to Table of Content](#)

14. How to flash Embedded Artists' i.MX8MQ COM board using UUU?

Step 1: Download the zip file containing the UUU tool/images from Embedded Artists' [website](#).

Step 2: Close the jumper J2 on the carrier board.



Step 3: Connect the J11 port of the EVK to the host PC via a Micro-B to A USB cable. The host PC will use this connection to flash the image (via either MFGtool or uuu tool).

Step 4: (Optional) Connect the J16 port of the EVK to the host PC via a Micro-B to A USB cable. The host PC will use this connection for console input/output (via a terminal emulator software, such as minicom or Tera Term).

Step 5: Power on the EVK. It is now ready to be flashed.

Step 6: Open Tera Term and set it to the appropriate COM port.

Step 7: Unpack the downloaded zip file somewhere on your computer (Both Ubuntu and Windows can be used).

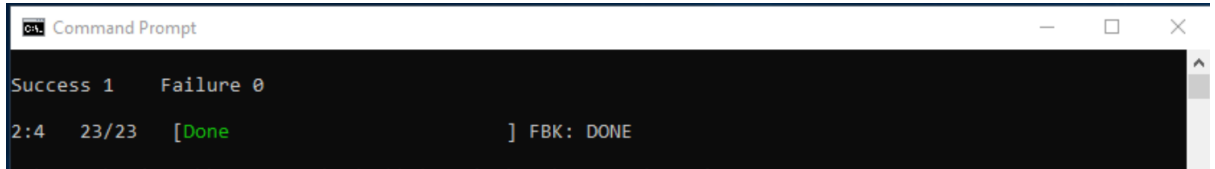
Step 8: Download [uuu](#) for Ubuntu and save the file in the same folder as the uuu README (extracted) file.

N.B. This step is required for Linux only. The Windows uuu executable is included in the downloaded zip file already.

Step 9a: To run in Windows, open command prompt and navigate to the folder where the uuu zip file was unpacked. Run the uuu tool:

```
C:/> cd c:/temp/uuu_imx8mq_com_4.14.78  
C:/temp/uuu_imx8mq_com_4.14.78> uuu.exe full_tar.uuu
```

You will see something like this in command prompt when the download is complete.

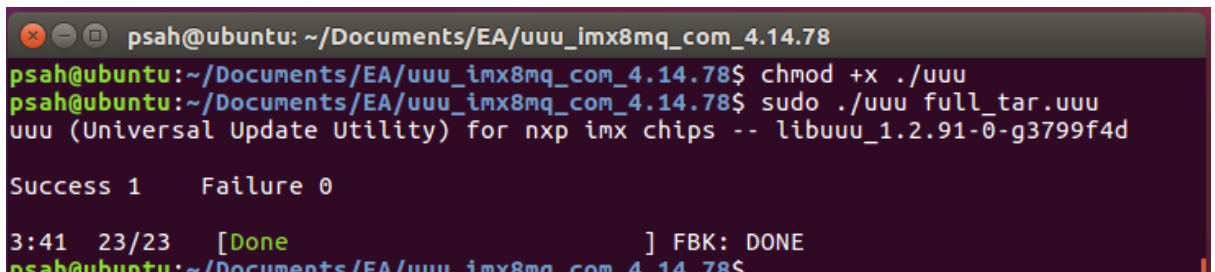


```
Command Prompt  
Success 1   Failure 0  
2:4  23/23  [Done] FBK: DONE
```

Step 9b: To run in Linux, open terminal window and navigate to the folder where the uuu zip file was unpacked. Run the uuu tool:

```
$ cd ~/uuu_imx8mq_com_4.14.78  
$ chmod +x ./uuu  
$ sudo ./uuu full_tar.uuu
```

You will see something like this in the terminal window when the download is complete.



```
psah@ubuntu: ~/Documents/EA/uuu_imx8mq_com_4.14.78  
psah@ubuntu:~/Documents/EA/uuu_imx8mq_com_4.14.78$ chmod +x ./uuu  
psah@ubuntu:~/Documents/EA/uuu_imx8mq_com_4.14.78$ sudo ./uuu full_tar.uuu  
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.2.91-0-g3799f4d  
Success 1   Failure 0  
3:41  23/23  [Done] FBK: DONE  
psah@ubuntu:~/Documents/EA/uuu_imx8mq_com_4.14.78$
```

The image has now been flashed.

[Back to Table of Content](#)