

**Embedded**[\(../../index.html\)](#)

Boot images

Network booting

You must first set up your PC workstation to use Digi Embedded Yocto. Follow the instructions at [Set up your development workstation \(yocto_t_set-up-workstation.html\)](#).

1. Open a serial connection

1. Open a serial connection using any terminal program such as Tera Term, Minicom, Coolterm, or HyperTerminal. This documentation demonstrates using Minicom to work with the device command line.

Use the following settings:

Parameter	Value
Port	Serial port where the device is connected
Baud rate	115200
Data bits	8
Parity	None
Stop bits	1
Flow control	None

2. Reset the device by pressing the reset button on the board. Then immediately press any key in the serial terminal to stop the auto-boot process. The U-Boot bootloader prompt displays:

```

----
U-Boot dub-2017.03-r2.1-git+g3e65181 (Nov 06 2017)

CPU:   Freescale i.MX6Q rev1.5 1200 MHz (running at 1200 MHz)
CPU:   Extended Commercial temperature grade -40 to 105 C
Reset cause: POR
I2C:   ready
DRAM:  1 GiB
MMC:   FSL_SDHC: 0 (eMMC), FSL_SDHC: 1
In:     serial
Out:    serial
Err:    serial
Model: Digi International ConnectCore 6 Single Board Computer
ConnectCore 6 SOM variant 0x02: Consumer quad-core
Board: ConnectCore 6 SBC, version 3, ID 129
Boot device: MMC4
PMIC:  DA9063, Device: 0x61, Variant: 0x50, Revision: 0x00
Net:    FEC [PRIME]
Hit any key to stop autoboot:  0
=>
----

```

2. Prepare the device artifacts

1. Get the Yocto firmware images to boot from network:
 - **<kernel-file>.bin**
 - **<device-tree-file>.dtb**
 - **<rootfs-file>.rootfs.tar.bz2**
 - a. After building the Yocto firmware, you can find the image files inside the project directory at:
 <project_folder>/tmp/deploy/images/<platform>
 - b. You can download Digi provided pre-built images from:
 - For ConnectCore 6 SBC:
 ftp://ftp1.digi.com/support/digiembeddedyocto/2.4/r2/images/ccimx6sbc/x11/
 (ftp://ftp1.digi.com/support/digiembeddedyocto/2.4/r2/images/ccimx6sbc/x11/)

yocto/2.4/r2/images/ccimx6sbc/x11/)

2. Untar the root filesystem tarball (***.rootfs.tar.bz2**) in the NFS exported directory of your development workstation. See [Set up your development workstation \(yocto_t_set-up-workstation.html\)](#).

```
~> sudo tar xvpf image.rootfs.tar.bz2 -C
```



3. Copy the kernel ***.bin** file to the TFTP exported directory of your development workstation.

```
~> sudo cp image.bin /tftpboot
```

4. Copy the device tree ***.dtb** file to the TFTP exported directory of your development workstation.

```
~> sudo cp image.dtb /tftpboot
```



Read the **board_id** and **soc_family** of your platform from U-Boot to find out the appropriate device tree file for your platform. For example:

```
=> printenv board_id
board_id=129
=> printenv soc_family
soc_family=imx6q
=>
```

The dtb file name of your device device tree for your platform follows the format:
zImage—4.9-r0.2-<soc_family>-<platform>-id<board_id>-<date>.dtb

3. Configure your device's network settings

1. Get a dynamic IP for your target:

```
=> setenv autoload no
=> dhcp
```

or you can set a static IP:

```
=> setenv ipaddr 192.168.115.222
```

2. Configure the IP of the development workstation with TFTP and NFS servers installed. See [Set up your development workstation \(yocto_t_set-up-workstation.html\)](#):

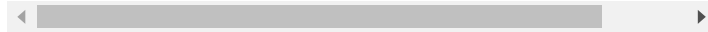
```
=> setenv serverip 192.168.115.1
```

4. Boot from network

Boot from TFTP+NFS

1. Set the directory with the rootfs to mount. This directory is the one exported via NFS in your development workstation. See [Set up an NFS server \(yocto_t_set-up-workstation.html#set-up-an-nfs-server\)](#).

```
=> setenv rootpath /exports/nfsroot-ccim
```



2. Specify the device tree (***.dtb**) file name. This is the name of the ***.dtb** file you copied to the TFTP exported directory of your development workstation.

```
=> setenv fdt_file image.dtb
```

3. Establish the kernel file (***.bin**) name. This is the name of the ***.bin** file you copied to the TFTP exported directory of your development workstation.

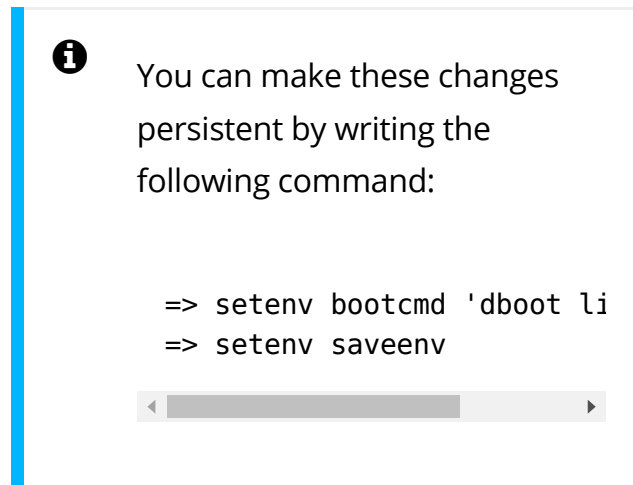
```
=> setenv zimage image.bin
```

4. Save the changes.

```
=> saveenv
```

5. Boot from TFTP.

```
=> dboot linux tftp
```



The target now loads the kernel and device tree from the TFTP server and the root filesystem from the NFS server.

Boot entirely from NFS

To avoid using TFTP for kernel and device tree files and boot everything from NFS, copy the kernel ***.bin** and device tree ***.dtb** files to the NFS-exported directory of your development workstation (instead of to the TFTP directory). See [Set up an NFS server \(yocto t set-up-workstation.html#set-up-an-nfs-server\)](https://yocto.t-set-up-workstation.html#set-up-an-nfs-server).

```
=> dboot linux nfs
```

On this page:

Network booting

1. Open a serial connection
2. Prepare the device artifacts
3. Configure your device's network settings
4. Boot from network

? This topic for another platform?

- [ConnectCore 6UL \(../cc6ul/yocto t boot-images\)](#)
- [ConnectCore 8X \(../cc8x/yocto t boot-images\)](#)
- [ConnectCore 6 Plus \(../cc6plus/yocto t boot-images\)](#)

[Product page](#)

[Official site](#)

[Contact us](#)

[More documentation](#)

[About Digi](#)

[Support](#)

©2019 Digi International Inc. All rights reserved.
Site last generated: Apr 30, 2019

