

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

# Set up your development workstation

You must first set up your PC workstation to use Digi Embedded Yocto. The resulting environment enables cross-platform development for ConnectCore devices using a Linux workstation.

## Requirements

- A 64-bit Linux machine with Internet connection.
- A Linux distribution supported by the Yocto Project.



Digi Embedded Yocto is developed using Ubuntu 16.04 LTS. For successful Digi Embedded Yocto projects, Digi recommends you use a Yocto Project-supported Ubuntu distribution. You can find Yocto Project-supported distributions in the Yocto Project system requirements at <http://www.yoctoproject.org/docs/2.4/ref-manual/ref-manual.html#detailed-supported-distros> .

- 8 GB of RAM.
- 8 CPU cores.
- 250 GB of free disk space. SSDs are highly recommended for faster build times.

## 1. Connect the hardware

Connect the cable.

This system boots the Linux kernel, device trees, and other boot artifacts from a TFTP server and mounts a root filesystem from an NFS server. This is ideal for development; changes to either the Linux kernel image or the root filesystem are immediately available without having to program the target device.

## 2. Install required packages

Install the following list of required packages for Ubuntu Linux distribution:

```
~> sudo apt-get install gawk wget git-core c
```



If you are using a different Linux distribution, find the list of packages at <http://www.yoctoproject.org/docs/2.4/ref-manual/ref-manual.html#required-packages-for-the-host-development-system>.



## 3. Set up a TFTP server

You also need to set up a TFTP server on your host so you can program your built images.

1. Install a TFTP server. For Ubuntu, you can execute the following command to install a TFTP server:

```
~> sudo apt-get install tftpd-hpa
```

2. After completing installation, create a directory **/tftpboot** where exported files will be located (only root user can create this directory):

```
~> sudo mkdir /tftpboot  
~> sudo chmod 1777 /tftpboot
```

3. To make sure the TFTP server is using the **/tftpboot** directory, edit the daemon's configuration file **/etc/default/tftpd-hpa** and change the constant **TFTP\_DIRECTORY** to point to your recently created **/tftpboot** folder:

```
/etc/default/tftpd-hpa  
  
TFTP_USERNAME="tftp"  
TFTP_DIRECTORY="/tftpboot"  
TFTP_ADDRESS="0.0.0.0:69"  
TFTP_OPTIONS="- - secure"
```

4. Restart the daemon so that it uses the new TFTP folder:

```
~> sudo service tftpd-hpa restart  
tftpd-hpa start/running, process 3297
```

## 4. Set up an NFS server

Booting from NFS during system development can significantly reduce idle time compared to flashing an image directly in your device.

Follow these steps in your Linux machine to install and configure the NFS server:

1. Install the NFS server:

```
~> sudo apt-get install nfs-kernel-server
```

2. Create a directory to use as root of the NFS server, for example **/exports/nfsroot-ccimx6sbc**.

```
~> sudo mkdir /exports/nfsroot-ccimx6sbc
```

3. Export the whole **/exports/nfsroot-ccimx6sbc** directory. To do so, edit the **/etc/exports** file of your development machine as root and add the following line:

```
/etc/exports
```

```
/exports/nfsroot-ccimx6sbc *(rw,no_root_s
```

4. Restart the NFS server with the following command:

```
~> sudo service nfs-kernel-server restart
```

## On this page:

1. Connect the hardware
2. Install required packages
3. Set up a TFTP server
4. Set up an NFS server

### ? This topic for another platform?

- [ConnectCore 6UL \(../cc6ul/yocto\\_t\\_set-up-workstation\)](#)
- [ConnectCore 8X \(../cc8x/yocto\\_t\\_set-up-workstation\)](#)
- [ConnectCore 6 Plus \(../cc6plus/yocto\\_t\\_set-up-workstation\)](#)

[Product page](#)

[Official site](#)

[Contact us](#)

[More documentation](#)

[About Digi](#)

[Support](#)

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

