



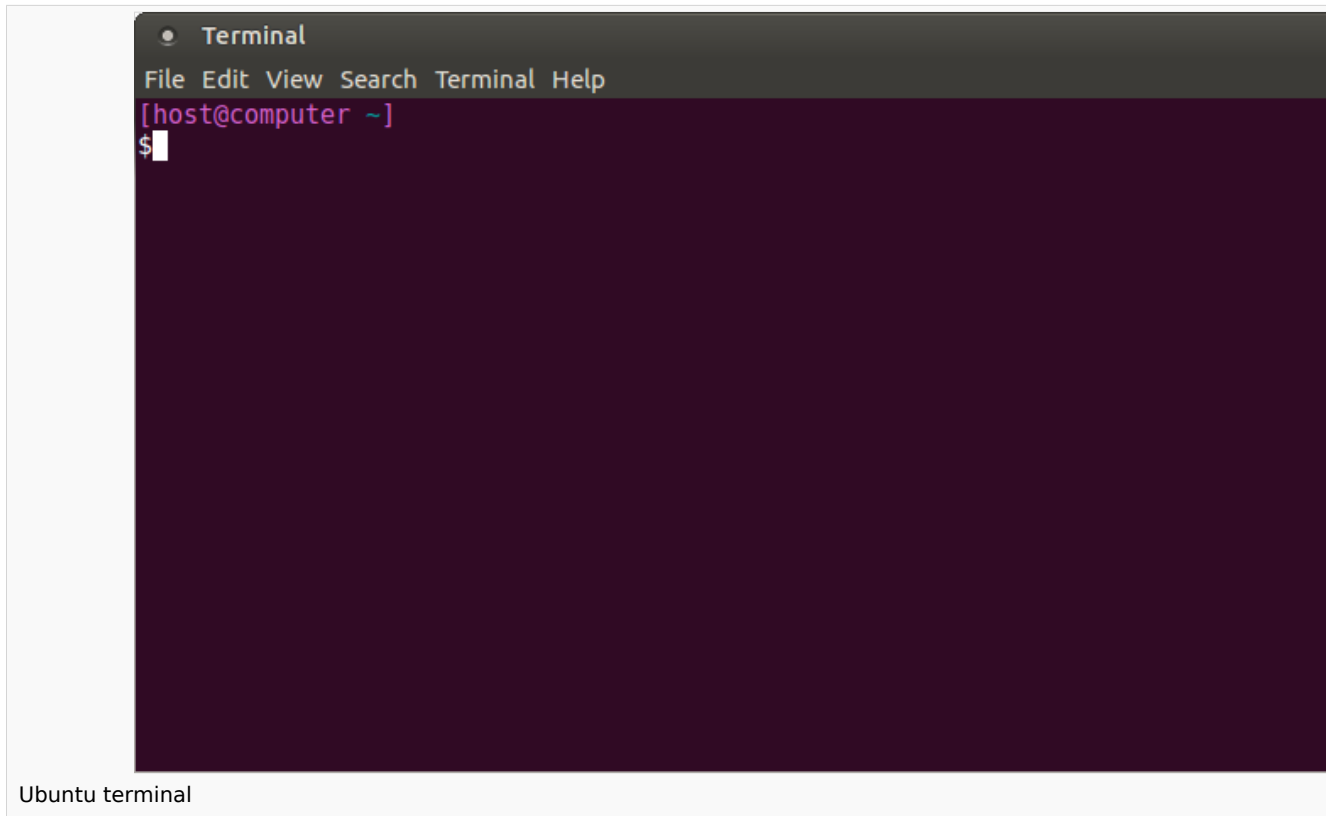
Populate the target and boot the image

Populate the target and boot the image

Stable: 06.03.2020 - 09:49 / Revision: 06.03.2020 - 09:38

1 Open a terminal

Open a terminal on the host computer.



All the commands preceded by **PC \$>** have to be executed from the host computer terminal.

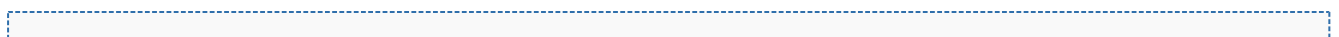
- Create your STM32MPU workspace directory on your host computer

```
PC $> mkdir $HOME/STM32MPU_workspace  
PC $> cd $HOME/STM32MPU_workspace
```

2 Check the host computer Internet access

- An Internet access through http and https protocols must be provided.

The command below allows to check for Internet access through http/https protocols:





Populate the target and boot the image

```
PC $> wget -q www.google.com && echo "Internet access over HTTP/HTTPS is OK !" || echo "No internet access over HTTP/HTTPS ! You may need to set up a proxy."
```

If an 'OK' message is returned, the network is well configured.

In such case, skip the rest of this section.

Any other likely situation indicates the need for a proxy for http/https protocols.

The best solution to set a proxy for http/https protocols is via the shell variables `http_proxy` and `https_proxy`:

```
PC $> export http_proxy=http://<MyProxyLogin>:<MyProxyPassword>@<MyProxyServerUrl>:<MyProxyPort>
PC $> export https_proxy=http://<MyProxyLogin>:<MyProxyPassword>@<MyProxyServerUrl>:<MyProxyPort>
```

Check again the Internet access with command:

```
PC $> wget -q www.google.com && echo "Internet access over HTTP/HTTPS is OK !" || echo "No internet access over HTTP/HTTPS ! You may need to set up a proxy."
```

3 Install the tools

3.1 STM32CubeProgrammer

- STM32CubeProgrammer requires 1.8 version of the Java platform.

```
PC $> sudo apt-get install openjdk-8-jre-headless
```

```
PC $> sudo update-alternatives --config java
```

Select the java-8-openjdk configuration.



- Install OpenJFX for Ubuntu@18.04

```
PC $> sudo apt purge openjfx
```

```
PC $> sudo apt install openjfx=8u161-b12-1ubuntu2 libopenjfx-jni=8u161-b12-1ubuntu2
libopenjfx-java=8u161-b12-1ubuntu2
```

```
PC $> sudo apt-mark hold openjfx libopenjfx-jni libopenjfx-java
```

- Create your STM32MPU tools directory on your host computer

```
PC $> mkdir $HOME/STM32MPU_workspace/STM32MPU-Tools
```

```
PC $> mkdir $HOME/STM32MPU_workspace/STM32MPU-Tools/STM32CubeProgrammer-x.y.z
```

- Create a temporary directory in your STM32MPU workspace



Populate the target and boot the image

```
PC $> mkdir $HOME/STM32MPU_workspace/tmp
```

- Download last **STM32CubeProgrammer** in the following directory
\$HOME/STM32MPU_workspace/tmp
- Uncompress the archive file to get the STM32CubeProgrammer installers

```
PC $> cd $HOME/STM32MPU_workspace/tmp  
PC $> unzip SetupSTM32CubeProgrammer.zip
```

- Execute the Linux installer, which guides you through the installation process. Select "\$HOME/STM32MPU-Tools /STM32CubeProgrammer-x.y.z" as the installation directory, when it's requested by the installer

```
PC $> ./SetupSTM32CubeProgrammer-x.y.z.linux
```

Screenshots, done with **STM32CubeProgrammer V2.2.0**, are given **as examples** below:



Splash screen



Populate the target and boot the image



Populate the target and boot the image



- Add the STM32CubeProgrammer binary path to your PATH environment variable

```
export PATH=$HOME/STM32MPU_workspace/STM32MPU-Tools/STM32CubeProgrammer-x.y.z/bin:$PATH
```

- Check that the STM32CubeProgrammer tool is properly installed and accessible

```
PC $> STM32_Programmer_CLI --h
-----
                STM32CubeProgrammer vx.y.z
-----
```

3.2 USB serial link

- Install the libusb on your host computer

```
PC $> sudo apt-get install libusb-1.0-0
```



Populate the target and boot the image

- To allow STM32CubeProgrammer to access the USB port through low-level commands, proceed as follows

```
PC $> cd $HOME/STM32MPU_workspace/STM32MPU-Tools/STM32CubeProgrammer-x.y.z/Drivers
/rules
PC $> sudo cp *.* /etc/udev/rules.d/
```

4 Download the image

The software package is provided AS IS, and by downloading it, you agree to be bound to the terms of the software license agreement (SLA). The detailed content licenses can be found [here](#).

4.1 For ecosystem release v1.2.0

- Create your STM32MP15 Starter Package directory on your host computer

```
PC $> mkdir $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.2.0
PC $> mkdir $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.2.0/Starter-Package
PC $> cd $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.2.0/Starter-Package
```

- Download the [STM32MP15-Ecosystem-v1.2.0 Starter Package](#) to the following directory:
\$HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.2.0/Starter-Package
- Uncompress the tarball file to get the binaries for the different partitions of the image, and the Flash layout files

```
PC $> tar xvf en.FLASH-stm32mp1-openstlinux-20-02-19.tar.xz
```

4.2 For ecosystem release v1.1.0

- Create your STM32MP15 Starter Package directory on your host computer

```
PC $> mkdir $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.1.0
PC $> mkdir $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.1.0/Starter-Package
PC $> cd $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.1.0/Starter-Package
```

- Download the [STM32MP15-Ecosystem-v1.1.0 Starter Package](#) to the following directory:
\$HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.1.0/Starter-Package
- Uncompress the tarball file to get the binaries for the different partitions of the image, and the Flash layout files

```
PC $> tar xvf en.FLASH-stm32mp1-openstlinux-4.19-thud-mp1-19-10-09.tar.xz
```

4.3 For ecosystem release v1.0.0

- Create your STM32MP15 Starter Package directory on your host computer



Populate the target and boot the image

```
PC $> mkdir $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.0.0
PC $> mkdir $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.0.0/Starter-Package
PC $> cd $HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.0.0/Starter-Package
```

- Download the [STM32MP15-Ecosystem-v1.0.0 Starter Package](#) to the following directory:
`$HOME/STM32MPU_workspace/STM32MP15-Ecosystem-v1.0.0/Starter-Package`
- Uncompress the tarball file to get the binaries for the different partitions of the image, and the Flash layout files

```
PC $> tar xvf en.FLASH-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz
```