



## How to get Terminal



---

A quality version of this page, approved on 26 September 2019, was based off this revision.

2 ways can be used to connect your PC to the board and then to control your board:

- through serial link (UART / USB)
- through network connection

## Contents

1 Remote terminal via serial link (UART/USB) .....	3
1.1 On Linux <sup>®</sup> PC .....	3
1.2 On Windows <sup>®</sup> PC .....	4
2 Remote terminal over network .....	5



## 1 Remote terminal via serial link (UART/USB)

First of all, you need to connect your host PC to the board via UART.

Depending on the board, some additional materials may be needed to physically interconnect the board to the console serial port (extension board or adapter to interconnect the UART from the board to the USB PC input).

Pictures describing board connections can be found in [Category:ST boards](#)

### 1.1 On Linux® PC

On Linux, we recommend to use *Minicom*.

You need first to check that Minicom is installed on your PC.

- From a Terminal window on your PC:

```
PC $> minicom
```

- If a message indicating Minicom does not exist, then **install** it:

```
PC $> sudo apt-get install minicom
```

- To **configure** Minicom:

The UART configuration that is set with the below command has to match the one from your board: to be checked within board user manual.

```
PC $> sudo minicom -s
```

Hardware flow control is generally set to off (default value is on) and a baud rate of 115200 is usual.

- To **launch** Minicom:

```
PC $> minicom -D /dev/tty...
```

See [How to use TTY with User Terminal](#) article to find the appropriate tty instance to be used.

See below example for a standard UART (ttyS0) or STLink interface (ttyACM0):

```
PC $> minicom -D /dev/ttyS0  
PC $> minicom -D /dev/ttyACM0
```

On the terminal, the prompt is changed to:

```
root@stm32mp1:~#
```

More information on Minicom can be found at the following link: <https://help.ubuntu.com/community/Minicom>.



---

## 1.2 On Windows® PC

Any of the Windows terminal emulator applications can be used. "Tera Term" is one of them: <http://tera-term.en.lo4d.com/>.

Then, the configuration is quite simple:

- Plug the cable and start the board
- Open a terminal emulator application on your PC and configure with the serial setup menu the USB port to be used and the baud rate of the serial link (in general 115200 baud)
- More information on the board serial port can generally be found in the hardware user manual **of the board**

The setup is correct when you can send Linux commands through the terminal emulator and get feedback. You can for example reboot the board and see the boot sequence.



## 2 Remote terminal over network

The board can also be accessed via the network through Ethernet connection, using ssh.

To do so, first **get the IP address** of the board:

- if it exists, using the board user interface (Terminal window or specific application)
- using a console (see chapter [Remote terminal via Serial link \(UART/USB\)](#))

If the network connection is through Ethernet then "InterfaceNetwork" = "Eth0", else if the network connection is using Wifi then "InterfaceNetwork" = "WLAN0"

```
Board $> ifconfig InterfaceNetwork
eth0    Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:FB
        inet addr:xxx.xxx.xxx.xxx <ip address>  Bcast:xxx.xxx.xxx.xxx  Mask:xxx.xxx.xxx.
xxx
        inet6 addr: fe80::280:e1ff:fe01:14fb/64  Scope:Link
```

Then, from another terminal on a remote PC connected to the same network, use the command described in [How to perform ssh connection](#).

Universal Asynchronous Receiver/Transmitter

Linux® is a registered trademark of Linus Torvalds.

ST in-circuit debugger and programmer for the STM8 and STM32 microcontroller families (See [ST-LINK](#) for more details)

Frame Buffer (could be the Kernel framebuffer linked to the display, a GPU framebuffer, an imaging framebuffer...)