Chapter 3

Setting up a development environment

Documentation

Documentation is key to know nitty gritty of microcontrollers. Following are the resources we will be required much in this course:

- STM32F3DISCOVERY User Manual
- STM32F303VC Datasheet
- STM32F303VC Reference Manual
- LSM303DLHC
- L3GD20

Tools Required

Following are the tools will be required in this course.

- **Rust V - 1.31** (or newer).
- **itmdump V - 0.3.1**
- **OpenOCD** >=0.8. Tested versions: v0.9.0 and v0.10.0
- **arm-none-eabi-gdb V - 7.12** (or newer highly recommended). Tested versions: 7.10, 7.11, 7.12 and 8.1.
- **Cargo-binutils V - 0.1.4** (or newer).
- **minicom** on **Linux** and **macOS**. Tested version: 2.7.
- **PuTTY** on **Windows**.

Verification of Rust Installation

Since you are here, means you must have already installed rust. For the sake of confirmation of the version installed.

1. Open command prompt/terminal (in linux) and run this command.

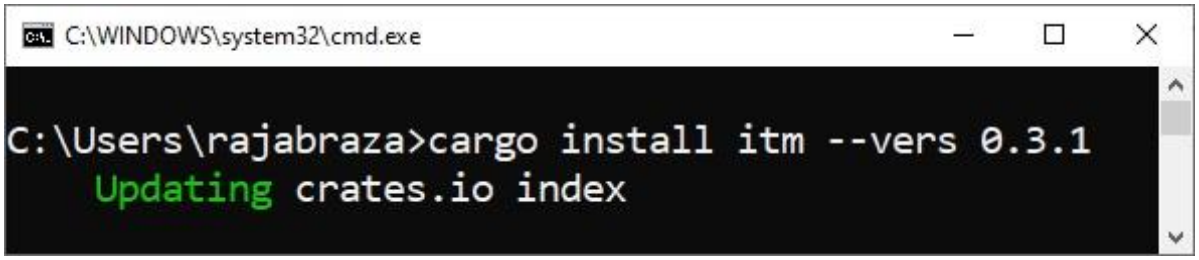
- In this machine version 1.38.0 of rust is installed. All okay here.

Itmdump

ITM -> Instrumentation Trace Macrocell (Communication Protocol)

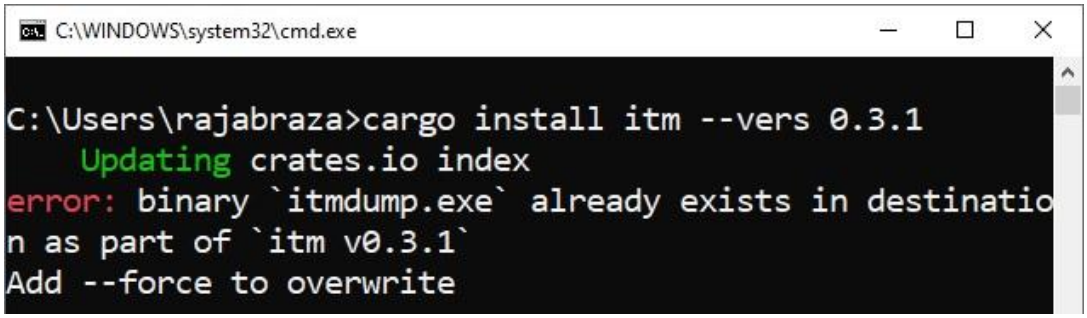
For installing itmdump you need to follow the steps below:

1. On command prompt/terminal *



itmdump

2. I have installed already, therefore i got this output. You probably got different (ending up installed successfully).

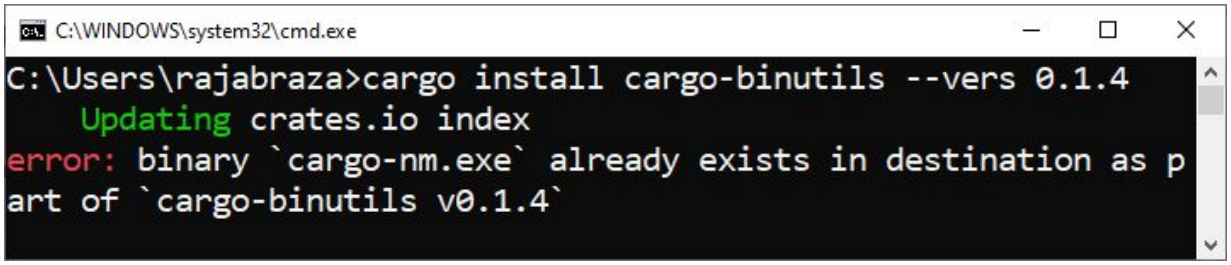


3. Verifying installation by

cargo-binutils

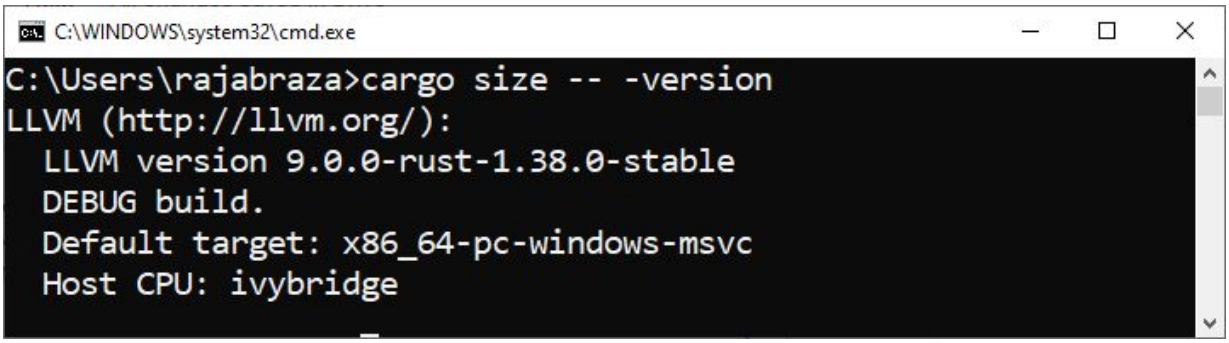
1. Before installing binary utilities, first we have to install llvm tools

2. After successful installation of llvm, we will install binary utilities.



cargo-binutils

3. After successful installation, verify it.



P.S. It's good habit to verify things once done.

Note : Also the output may slight differ depending upon your system architecture and OS.

OS Specific Instructions

OS specific instructions

So far, the instructions we performed was independent of Operating systems but now the remaining instructions will be performed are OS specific. Therefore we will cater the cases of:

1. Windows
2. Linux

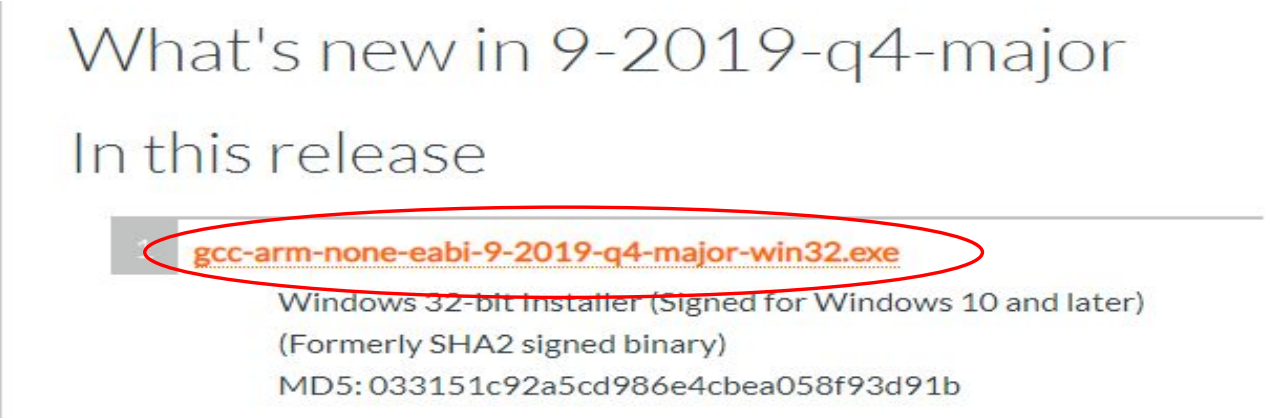
Windows

In Windows following installations will be required:

1. GDB
2. OpenOCD
3. PuTTy
4. ST-LINK Driver

arm-none-eabi-gdb

1. First thing first, visit arm website and download executable (.exe) file
2. Under release section click the highlighted link to download file.

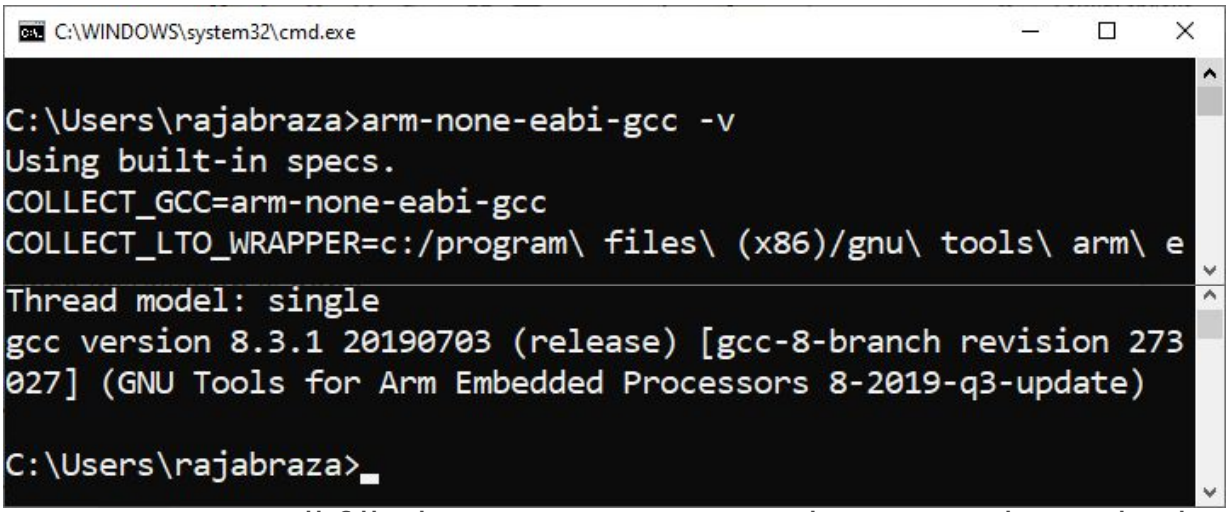


https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads

arm-none-eabi-gdb (cont)

3. Upon successful download locate the file and launch executable.

4. Once installation is done



Long output will filled-up your screen, ending up with similar lines as above.

PuTTy

Get puTTy right from here.

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html


(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

MSI ('Windows Installer')

32-bit:	putty-0.73-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.73-installer.msi	(or by FTP)	(signature)

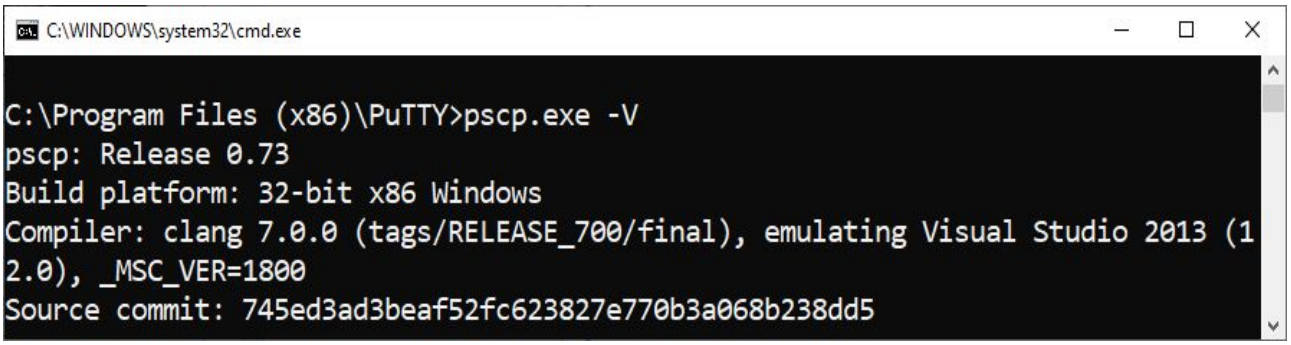
Unix source archive

.tar.gz:	putty-0.73.tar.gz	(or by FTP)	(signature)

2. Once done with download, run the setup.

PuTTy

3. Upon successful installation, verify the installation.



Note: Run the above command from the directory where PuTTy installed.

ST-LINK USB driver

Go get your driver from here.

https://www.st.com/en/development-tools/stsw-link009.html

1. Create account and download driver.
2. Extract compressed folder and launch the file "**stlink_winusb_install.bat**" as administrator.

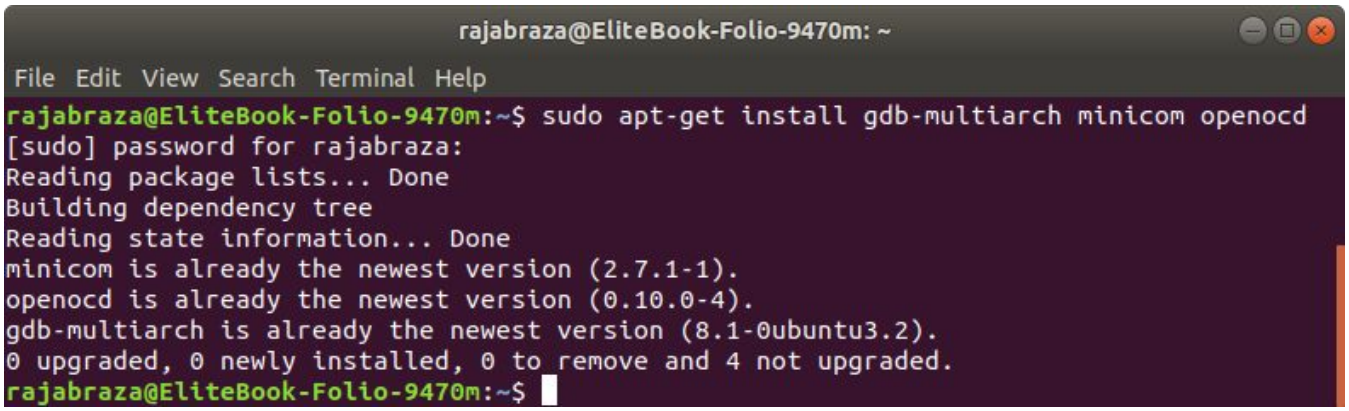
This is it, you are all set for embedded programming. It may be not that simple as it sounds. Do some extra efforts!

Linux : Required packages

In Linux following installations are required.*

1. GDB
2. OpenOCD
3. Minicom

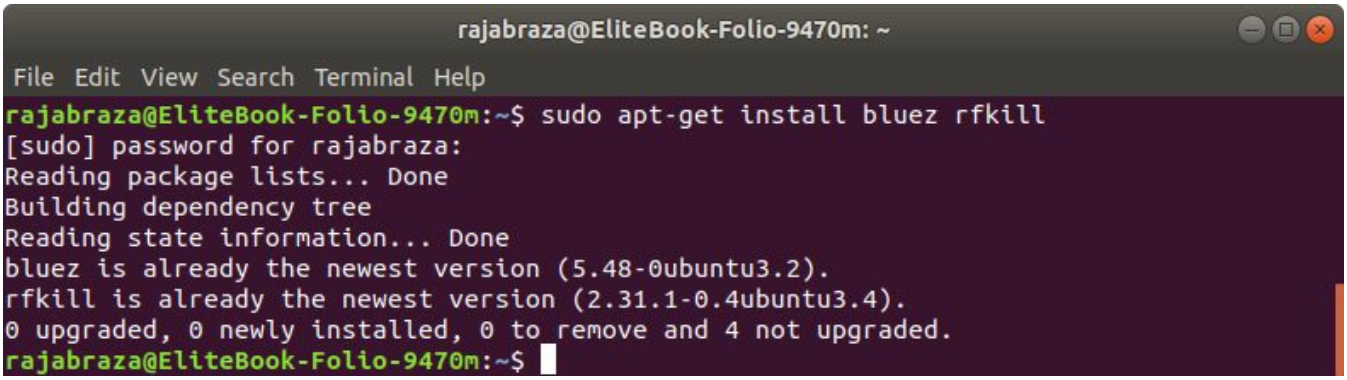
In Linux things are much simple. Like 1,2,3..



Linux : Optional packages

Following are the optional installations.

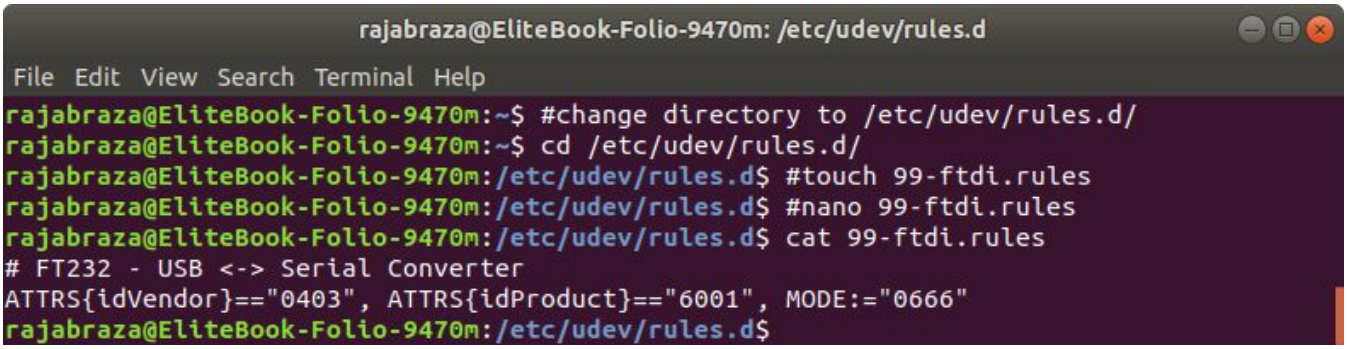
1. Bluez
2. Rfkill



Linux : udev rules

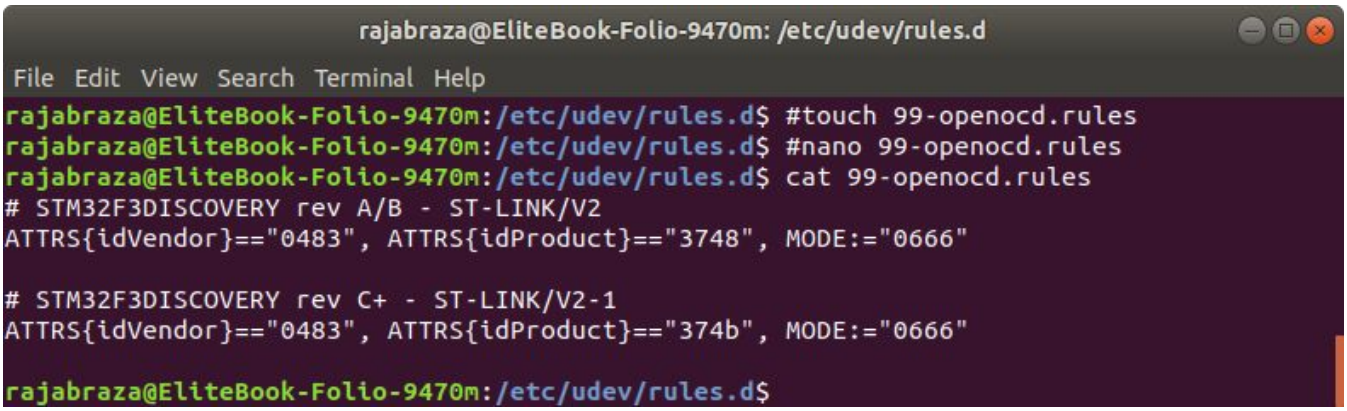
The purpose of setting up these rules is to let you use USB devices without **sudo** privilege.

1. First rule for ftdi (serial module)



Linux : udev rules

2. Second rule is for ST-LINK debugger (F3 USB Port)



Finall, you just need to run one last command.

```
$ sudo udevadm control --reload-rules
```

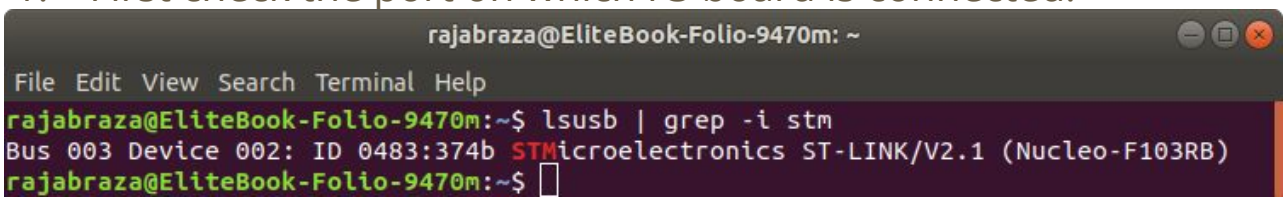
Now, you are equipped with all the tools required.

Verify the installation

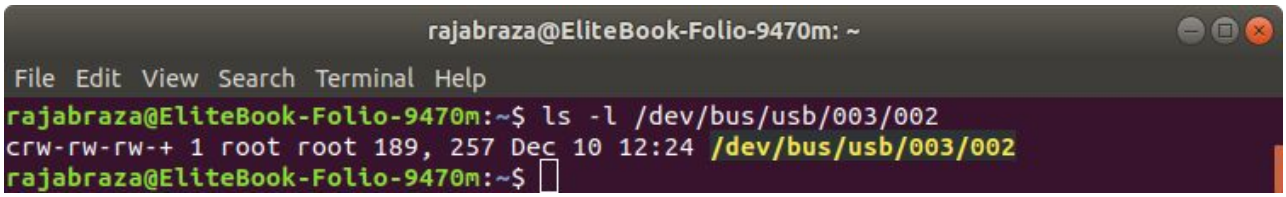
Checking F3 board permissions

Verify permission first, for that:

1. First check the port on which F3 board is connected.

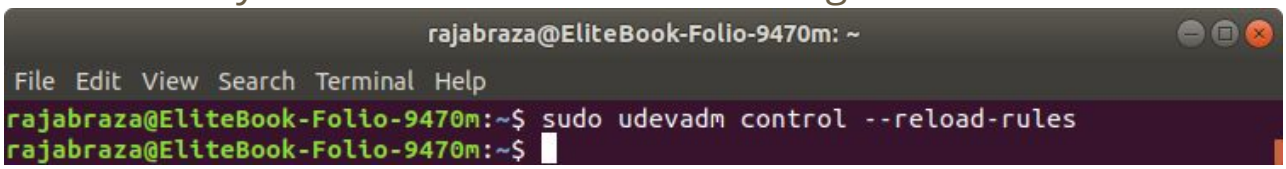


2. The above result shows F3 board connected on **bus 003** as **device 002**. Now for checking permissions for this device.



Checking F3 board permissions

3. If you get the same result then no need to do extra work, but if it differs then you need to check udev-rules again and then reload them.

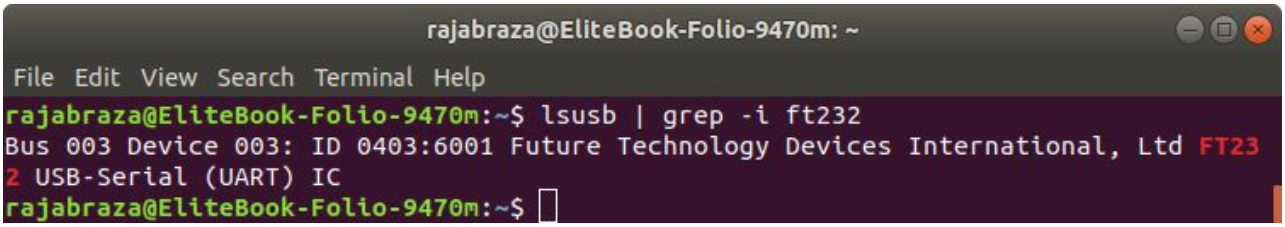


Note : Aboe command won't show any output, so after running it just repeat the steps.

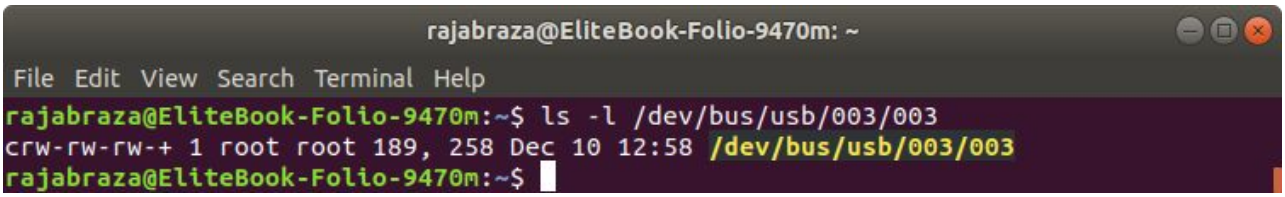
Checking ftdi permissions

Verify permission first, for that:

1. First check the port on which ftdi module is connected.



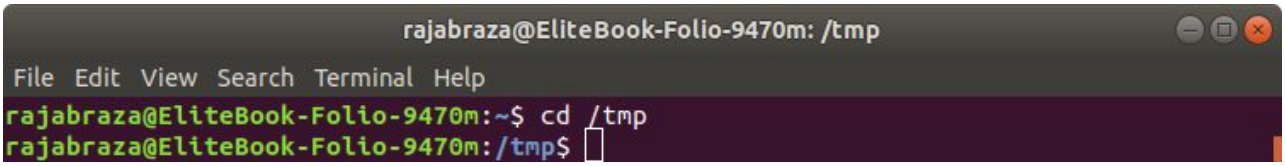
2. The above result shows ftdi module connected on **bus 003** as **device 003**. Now for checking permissions for this device.



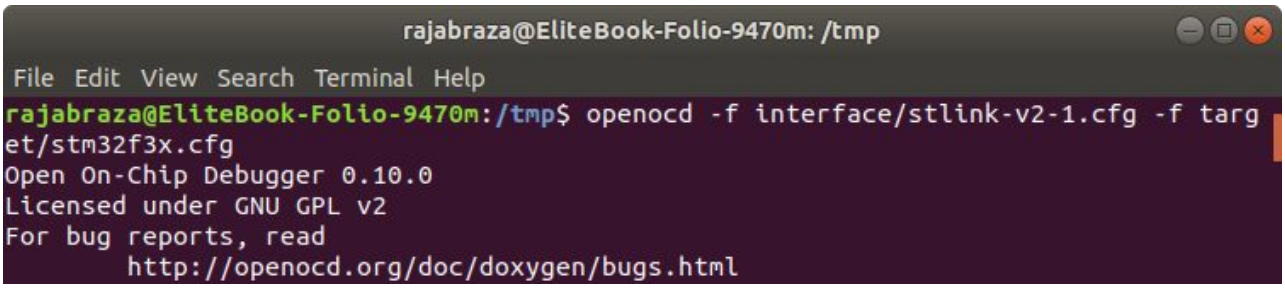
Establishing OpenOCD Connection

We have already discussed the purpose of using OpenOCD. For establishing a connection between host and debugging target.

1. Change you current directory to **/tmp**

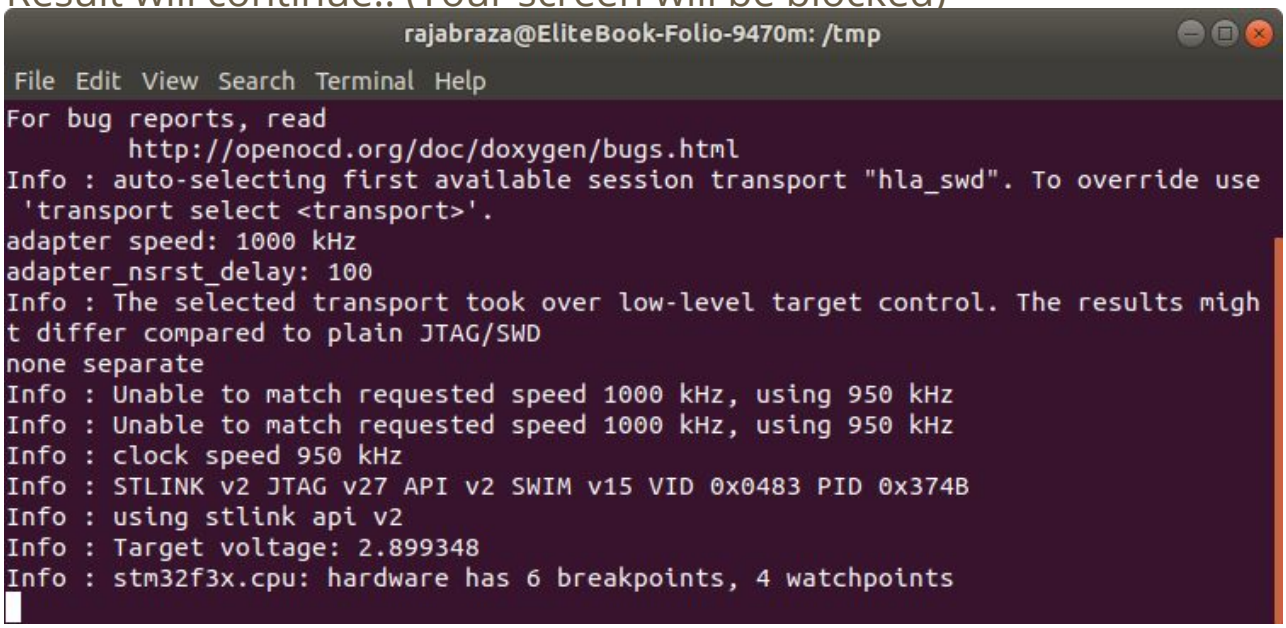


2. Next run this command to establish link between devices.



Establishing OpenOCD Connection

Result will continue.. (Your screen will be blocked)



Summary