

---

# 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**  $\geq 0.8$ . 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**.

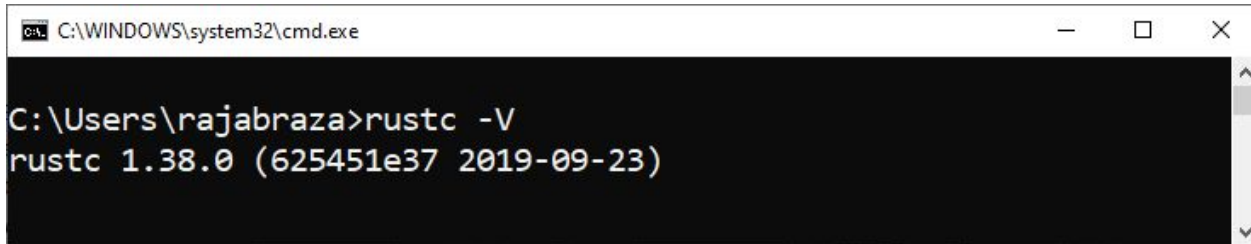


# Verification of Rust Installation



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.



```
C:\WINDOWS\system32\cmd.exe

C:\Users\rajabraza>rustc -V
rustc 1.38.0 (625451e37 2019-09-23)
```

- 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 \*

```
C:\WINDOWS\system32\cmd.exe

C:\Users\rajabrazza>cargo install itm --vers 0.3.1
  Updating crates.io index
```



# itmdump



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

```
C:\WINDOWS\system32\cmd.exe

C:\Users\rajabrazaz>cargo install itm --vers 0.3.1
  Updating crates.io index
error: binary `itmdump.exe` already exists in destination
as part of `itm v0.3.1`
Add --force to overwrite
```

3. Verifying installation by

```
C:\WINDOWS\system32\cmd.exe

C:\Users\rajabrazaz>itmdump -V
itmdump 0.3.1
```



# cargo-binutils



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

```
C:\WINDOWS\system32\cmd.exe

C:\Users\rajabraza>rustup component add llvm-tools-preview
info: component 'llvm-tools-preview' for target 'x86_64-pc-windows-msvc' is up to date
```

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

```
C:\WINDOWS\system32\cmd.exe

C:\Users\rajabraza>cargo install cargo-binutils --vers 0.1.4
    Updating crates.io index
error: binary `cargo-nm.exe` already exists in destination as part of `cargo-binutils v0.1.4`
```



# cargo-binutils



3. After successful installation, verify it.

```
C:\WINDOWS\system32\cmd.exe
C:\Users\rajabrazaz>cargo size -- -version
LLVM (http://llvm.org/):
  LLVM version 9.0.0-rust-1.38.0-stable
  DEBUG build.
  Default target: x86_64-pc-windows-msvc
  Host CPU: ivybridge
```

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](https://developer.arm.com) website and download executable (.exe) file
2. Under release section click the highlighted link to download file.

## What's new in 9-2019-q4-major

### In this release

**gcc-arm-none-eabi-9-2019-q4-major-win32.exe**

Windows 32-bit Installer (Signed for Windows 10 and later)  
(Formerly SHA2 signed binary)  
MD5: 033151c92a5cd986e4cbea058f93d91b

<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

```
C:\WINDOWS\system32\cmd.exe

C:\Users\rajabraza>arm-none-eabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-none-eabi-gcc
COLLECT_LTO_WRAPPER=c:/program\ files\ (x86)/gnu\ tools\ arm\ e
Thread model: single
gcc version 8.3.1 20190703 (release) [gcc-8-branch revision 273
027] (GNU Tools for Arm Embedded Processors 8-2019-q3-update)

C:\Users\rajabraza>_
```

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.

```
C:\WINDOWS\system32\cmd.exe

C:\Program Files (x86)\PuTTY>pscp.exe -V
pscp: Release 0.73
Build platform: 32-bit x86 Windows
Compiler: clang 7.0.0 (tags/RELEASE_700/final), emulating Visual Studio 2013 (12.0), _MSC_VER=1800
Source commit: 745ed3ad3beaf52fc623827e770b3a068b238dd5
```

**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).

**<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..

```
rajabraza@EliteBook-Folio-9470m: ~  
File Edit View Search Terminal Help  
rajabraza@EliteBook-Folio-9470m:~$ sudo apt-get install gdb-multiarch minicom openocd  
[sudo] password for rajabraza:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
minicom is already the newest version (2.7.1-1).  
openocd is already the newest version (0.10.0-4).  
gdb-multiarch is already the newest version (8.1-0ubuntu3.2).  
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.  
rajabraza@EliteBook-Folio-9470m:~$
```



# Linux : Optional packages



Following are the optional installations.

1. [Bluez](#)
2. [Rfkill](#)

```
rajabraza@EliteBook-Folio-9470m: ~  
File Edit View Search Terminal Help  
rajabraza@EliteBook-Folio-9470m:~$ sudo apt-get install bluez rfkill  
[sudo] password for rajabraza:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
bluez is already the newest version (5.48-0ubuntu3.2).  
rfkill is already the newest version (2.31.1-0.4ubuntu3.4).  
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.  
rajabraza@EliteBook-Folio-9470m:~$
```



# 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)

```
rajabraza@EliteBook-Folio-9470m: /etc/udev/rules.d
File Edit View Search Terminal Help
rajabraza@EliteBook-Folio-9470m:~$ #change directory to /etc/udev/rules.d/
rajabraza@EliteBook-Folio-9470m:~$ cd /etc/udev/rules.d/
rajabraza@EliteBook-Folio-9470m:/etc/udev/rules.d$ #touch 99-ftdi.rules
rajabraza@EliteBook-Folio-9470m:/etc/udev/rules.d$ #nano 99-ftdi.rules
rajabraza@EliteBook-Folio-9470m:/etc/udev/rules.d$ cat 99-ftdi.rules
# FT232 - USB <-> Serial Converter
ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001", MODE:="0666"
rajabraza@EliteBook-Folio-9470m:/etc/udev/rules.d$
```



# Linux : udev rules



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

```
rajabraz@EliteBook-Folio-9470m: /etc/udev/rules.d
File Edit View Search Terminal Help
rajabraz@EliteBook-Folio-9470m:/etc/udev/rules.d$ #touch 99-openocd.rules
rajabraz@EliteBook-Folio-9470m:/etc/udev/rules.d$ #nano 99-openocd.rules
rajabraz@EliteBook-Folio-9470m:/etc/udev/rules.d$ cat 99-openocd.rules
# STM32F3DISCOVERY rev A/B - ST-LINK/V2
ATTRS{idVendor}=="0483", ATTRS{idProduct}=="3748", MODE:="0666"

# STM32F3DISCOVERY rev C+ - ST-LINK/V2-1
ATTRS{idVendor}=="0483", ATTRS{idProduct}=="374b", MODE:="0666"

rajabraz@EliteBook-Folio-9470m:/etc/udev/rules.d$
```

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.

```
rajabraza@EliteBook-Folio-9470m: ~  
File Edit View Search Terminal Help  
rajabraza@EliteBook-Folio-9470m:~$ lsusb | grep -i stm  
Bus 003 Device 002: ID 0483:374b STMicroelectronics ST-LINK/V2.1 (Nucleo-F103RB)  
rajabraza@EliteBook-Folio-9470m:~$
```

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

```
rajabraza@EliteBook-Folio-9470m: ~  
File Edit View Search Terminal Help  
rajabraza@EliteBook-Folio-9470m:~$ ls -l /dev/bus/usb/003/002  
crw-rw-rw-+ 1 root root 189, 257 Dec 10 12:24 /dev/bus/usb/003/002  
rajabraza@EliteBook-Folio-9470m:~$
```



# 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.

```
rajabraza@EliteBook-Folio-9470m: ~  
File Edit View Search Terminal Help  
rajabraza@EliteBook-Folio-9470m:~$ sudo udevadm control --reload-rules  
rajabraza@EliteBook-Folio-9470m:~$
```

**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.

```
rajabraza@EliteBook-Folio-9470m: ~  
File Edit View Search Terminal Help  
rajabraza@EliteBook-Folio-9470m:~$ lsusb | grep -i ft232  
Bus 003 Device 003: ID 0403:6001 Future Technology Devices International, Ltd FT23  
2 USB-Serial (UART) IC  
rajabraza@EliteBook-Folio-9470m:~$
```

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

```
rajabraza@EliteBook-Folio-9470m: ~  
File Edit View Search Terminal Help  
rajabraza@EliteBook-Folio-9470m:~$ ls -l /dev/bus/usb/003/003  
crw-rw-rw-+ 1 root root 189, 258 Dec 10 12:58 /dev/bus/usb/003/003  
rajabraza@EliteBook-Folio-9470m:~$
```





# Establishing OpenOCD Connection



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

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

```
rajabraza@EliteBook-Folio-9470m: /tmp
File Edit View Search Terminal Help
rajabraza@EliteBook-Folio-9470m:~$ cd /tmp
rajabraza@EliteBook-Folio-9470m:/tmp$
```

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

```
rajabraza@EliteBook-Folio-9470m: /tmp
File Edit View Search Terminal Help
rajabraza@EliteBook-Folio-9470m:/tmp$ openocd -f interface/stlink-v2-1.cfg -f target/stm32f3x.cfg
Open On-Chip Debugger 0.10.0
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html
```



# Establishing OpenOCD Connection



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

```
rajabraza@EliteBook-Folio-9470m: /tmp
File Edit View Search Terminal Help
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
Info : auto-selecting first available session transport "hla_swd". To override use
'transport select <transport>'.
adapter speed: 1000 kHz
adapter_nsrst_delay: 100
Info : The selected transport took over low-level target control. The results might
differ compared to plain JTAG/SWD
none separate
Info : Unable to match requested speed 1000 kHz, using 950 kHz
Info : Unable to match requested speed 1000 kHz, using 950 kHz
Info : clock speed 950 kHz
Info : STLINK v2 JTAG v27 API v2 SWIM v15 VID 0x0483 PID 0x374B
Info : using stlink api v2
Info : Target voltage: 2.899348
Info : stm32f3x.cpu: hardware has 6 breakpoints, 4 watchpoints
```



# Summary