1 Software Compilation and Installation

1.1 Getting the Compiler

Open a terminal shell on your machine and execute the following commands, to install the official Raspberry Pi cross toolchain.

```
host> mkdir -p ~/Desktop/Raspberry
host> cd ~/Desktop/Raspberry
host> sudo apt-get install git bc
host> git clone https://github.com/raspberrypi/tools toolchain
host> echo PATH=\$PATH:~/Desktop/Raspberry/toolchain/arm-bcm2708/gcc-
linaro-arm-linux-gnueabihf-raspbian/bin >> ~/.bashrc
host> source ~/.bashrc
```

Check if the system recognizes the compiler.

```
host> which arm-linux-gnueabihf-gcc
```

The command output should show something like this (depending on your installation path):

```
/home/development/Desktop/Raspberry/toolchain/arm-bcm2708/gcc-linaro-arm-
linux-gnueabihf-raspbian/bin/arm-linux-gnueabihf-gcc
```

1.2 Getting the Sources

Kernel

The Moitessier HAT uses a dynamically loadable kernel device driver module. To cross compile this module, the kernel sources needs to be referenced. Keep in mind, that the kernel on your target, must have the same version as the kernel sources used for cross compilation. If the target kernel is based on a different kernel source tree, the module loading will fail. The same applies for the toolchain used for kernel creation. It must be the same to compile the device driver. You'll get an *invalid module format* error when loading the driver, if you do not take that into account.

```
host> cd ~/Desktop/Raspberry
host> git clone https://github.com/raspberrypi/linux kernel
```

Moitessier HAT

The source repository is available at GitHub.

```
host> cd ~/Desktop/Raspberry
host> git clone https://github.com/mr-rooney/moitessier.git
host> cd moitessier
```

Only once after cloning you need to initialize the cloned submodules.

```
host> git submodule update --init --recursive
```

To update the sources of the submodules at a later time do the following.

```
host> git submodule update
```

The following steps need to be repeated for each submodule. *<SUBMOUDLE>* is a placeholder for the proper submodule name. To list the submodules use the shell command *ls*.

```
host> cd <SUBMODULE>
host> git checkout master
host> cd ..
```

1.3 Updating the Sources

The kernel as well as the Moitessier HAT sources are actively maintained, so code will change in the course of time.

Before you compile the sources you should update your local source tree structure on a regular basis.

Kernel

```
host> cd ~/Desktop/Raspberry/kernel
host> git pull --tags
```

Moitessier HAT

```
host> cd ~/Desktop/Raspberry/moitessier
host> git pull origin master
host> git submodule update
```

1.4 Compilation

Kernel

You need to ensure that the Moitessier HAT sources are cross compiled for the same kernel version, that is running on your Raspberry Pi, otherwise device driver loading will fail due to wrong module format. To get the relevant information, proceed as follows.

(1) Determine the current kernel source code version.

```
pi> zcat /usr/share/doc/raspberrypi-bootloader/changelog.Debian.gz | head
```

This command will output the version of the associated kernel sources. In this example the version is *1.20180417-1*.

```
raspberrypi-firmware (1.20180417-1) stretch; urgency=medium

  * firmware as of 5db8e4e1c63178e200d6fbea23ed4a9bf4656658

 -- Serge Schneider <serge@raspberrypi.org>  Tue, 17 Apr 2018 13:07:14
+0100

raspberrypi-firmware (1.20180328-1) stretch; urgency=medium

  * firmware as of ce8652e2c743f02f04cb29f23611cbf13765483b
```

(2) Go to https://github.com/raspberrypi/linux/releases and search for *1.20180417-1*. The associated tag is *raspberrypi-kernel_1.20180417-1*.

(3) Checkout the sources referenced by the repository tag.

```
host> cd ~/Desktop/Raspberry/kernel
host> git checkout raspberrypi-kernel_1.20180417-1
```

As far as you don’t change anything on the kernel sources itself or on the kernel version used on the target system, you will only need to compile the kernel once. This step is only required to cross compile the kernel device driver of the Moitessier HAT. There is no need to copy the created kernel binary to your Raspbian system.

```
host> make distclean ARCH=arm
host> KERNEL=kernel7
host> make bcm2709_defconfig ARCH=arm
host> make -j2 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- zImage dtbs
modules
```

Moitessier HAT

The compilation/generation process is managed by the bash script *create*, located in the top level of the source directory. This script is used to compile all Moitessier HAT related sources and creates in the final step a debian package. However, you might want to build the sources step by step. In this case, use the proper scripts/Makefiles in the subdirectories. You'll get the necessary instructions in the README.md files included in the subdirectories.

If you make changes on the source code file structure or you want to use a different compiler, you might need to adopt the configuration in this script.

```
host> cd ~/Desktop/Raspberry/moitessier
host> ./create
```

1.5 Package Installation

After you’ve compiled the sources successfully, you can copy the created debian package to your Raspberry Pi using *scp* or similar. Use the command below with a proper IP address to access the Raspberry Pi in your network.

```
host> scp -r deploy/moitessier_*_armhf.deb pi@10.10.10.1:/home/pi/Desktop
```

Afterwards you need to install the package on your Raspberry Pi.

```
pi> sudo dpkg -i ~/Desktop/moitessier_*_armhf.deb
```

Your system will reboot and will load the device driver to access the Moitessier HAT automatically. No need to manually load the driver.

To check the installed package version, call the following.

```
pi> dpkg -s moitessier
```

Compatibility with OpenPlotter

If you are using OpenPlotter copy the package to the configuration directory.

```
host> scp -r deploy/moitessier_*_armhf.deb
pi@10.10.10.1:/home/pi/.config/openplotter/tools/moitessier_hat/packages
```

The package will be listed directly within the Moitesser HAT settings window in OpenPlotter. You might need to restart the OpenPlotter GUI to update the list.

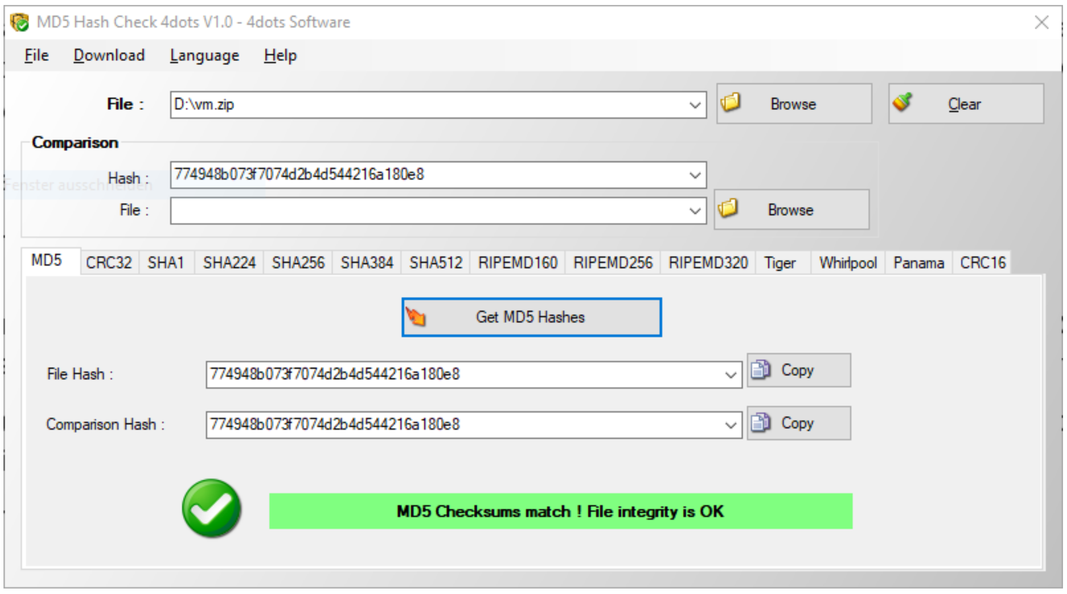
1.6 Virtual Machine

For simplicity a prepared virtual machine can be used, including the relevant sources and all the tools required to create a Raspbian compatible package. It is assumed that you have the free VMware Player or VirtualBox installed on your development host machine.

Virtual machine: http://downloads.rooco.eu/moitessier/vm.zip

MD5 hash: http://downloads.rooco.eu/moitessier/vm_md5.txt

You might want to verify the integrity of the downloaded file. Use a utility like *Hash Check 4dots*[1] to generate the MD5 hash. Compare this hash with the hash stated in *vm_md5.txt*.



File : D:\vm.zip

Hash : 774948b073f7074d2b4d544216a180e8

File :

File Hash : 774948b073f7074d2b4d544216a180e8

Comparison Hash : 774948b073f7074d2b4d544216a180e8

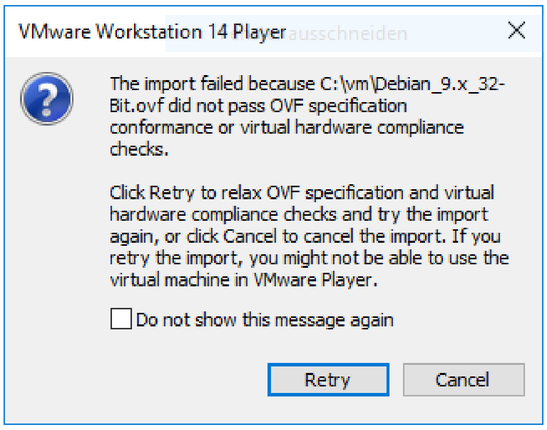
MD5 Checksums match ! File integrity is OK

Figure 1: Integrity verification with hash utility

[1] https://www.4dots-software.com/md5hashchecker/

Unpack the virtual machine, open the **.ovf* configuration file in the VMware Player / VirtualBox and start the virtual machine.

If using the VMware Player, you might get an import warning. Press the retry button to continue the import.


VMware Workstation 14 Player ausschneiden

The import failed because C:\vm\Debian_9.x_32-Bit.ovf did not pass OVF specification conformance or virtual hardware compliance checks.

Click Retry to relax OVF specification and virtual hardware compliance checks and try the import again, or click Cancel to cancel the import. If you retry the import, you might not be able to use the virtual machine in VMware Player.

☐ Do not show this message again

Figure 2: VMware Player, import warning

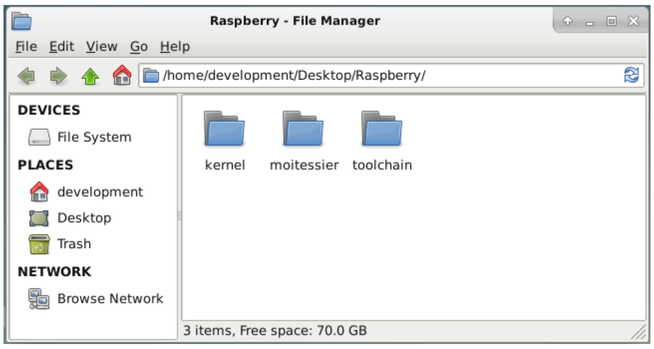
Login credentials:

- User: development
- Password: 1234

The user *development* belongs to the group *sudo*, that enables root rights required for compilation purpose.

The relevant sources can be found on the Desktop in the folder *Raspberry*:

- *kernel*: kernel sources for the Raspberry Pi
- *moitessier*: sources related to the Moitessier HAT (kernel driver, applications, scripts, etc.)
- *toolchain*: cross toolchain used to compile the kernel and Moitessier sources


Raspberry - File Manager

/home/development/Desktop/Raspberry/

3 items, Free space: 70.0 GB

Figure 3: Source folders

To compile the sources and to install the package afterwards, proceed as described in section 1.3 to 1.5.