

OpenLab - Open Laboratory

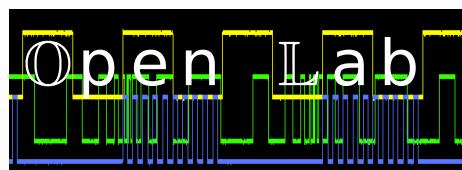
Installation Instruction V2

UAS Technikum Wien

Höchstädtplatz 6, A-1200 Vienna

I: www.technikum-wien.at

E: markus.lechner@technikum-wien.at



Contents

1	OpenLab Oscilloscope Hardware Setup	1
2	Signal-Toolkit Installation Instruction	3
2.1	Linux Distribution - Java and serial device configuration	3
2.1.1	Installing JAVA using a RPM package (usually Red Hat based distributions Fedora, Korora,...)	4
2.1.2	Installing JAVA using a tar.gz package (usually Debian based distributions Ubuntu, Mint,...)	5
2.1.3	Serial Device Permissions	6
2.1.4	Execution of the Signal-Toolkit Application	7
2.2	Windows Distribution - Java and FTDI driver	7
2.2.1	FTDI Driver Verification	7
2.2.2	Execution of the Signal-Toolkit Application	8

1 OpenLab Oscilloscope Hardware Setup

The image below illustrates an overview of the assembled OpenLab oscilloscope utilizing the FPGA DEO board.

Before you start with your measurements, please make sure that all switches (SW0, SW1, ..., SW9) are in the down position!

Furthermore, connect the PCB only to the male header GPIO1. Connect the serial FTDI cable to the hardware and subsequently to a free USB port. Finally the FPGA can be started by pressing the button SW10 (red button). Now the OpenLab Launcher can be started.

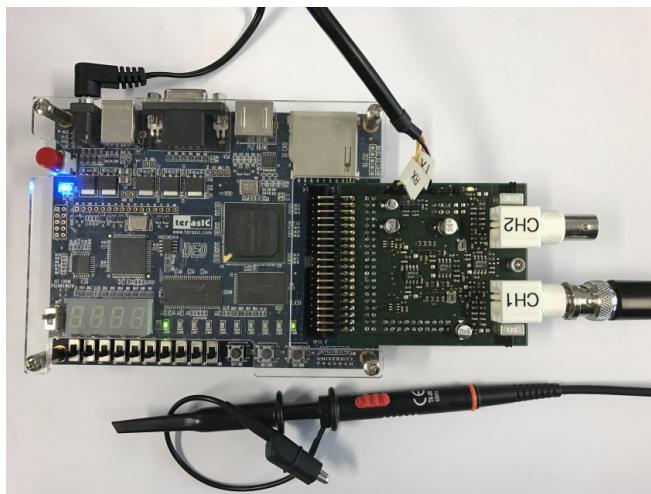
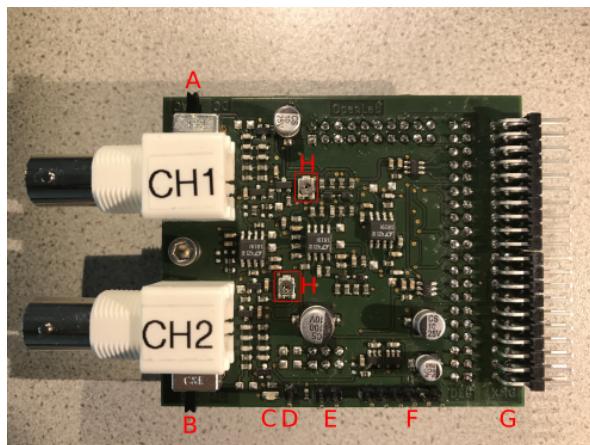


Figure 1.1: Hardware Overview

The following images are intended to give an overview of the PCB.



(a) Hardware Front View



(b) Hardware Back View

Figure 1.2: OpenLab Oscilloscope Hardware

1 OpenLab Oscilloscope Hardware Setup

- A: AC/DC Switches CH1
- B: AC/DC Switches CH2
- C: Power LED
- D: Probe Compensation
- E: GND Connector
- F: Serial Interface Connector (GND | - | - | TX | RX | -)
- G: FPGA DE0 Connector
- H: Offset Correction - **Do not adjust this trimmer!**
- I: Threaded Standoff
- J: TIVA-C Connectors
- K: XMC4500 Connector

The FTDI cable needs to be connected to the '**Serial Interface Connector**' - **H**, shown in the image below.

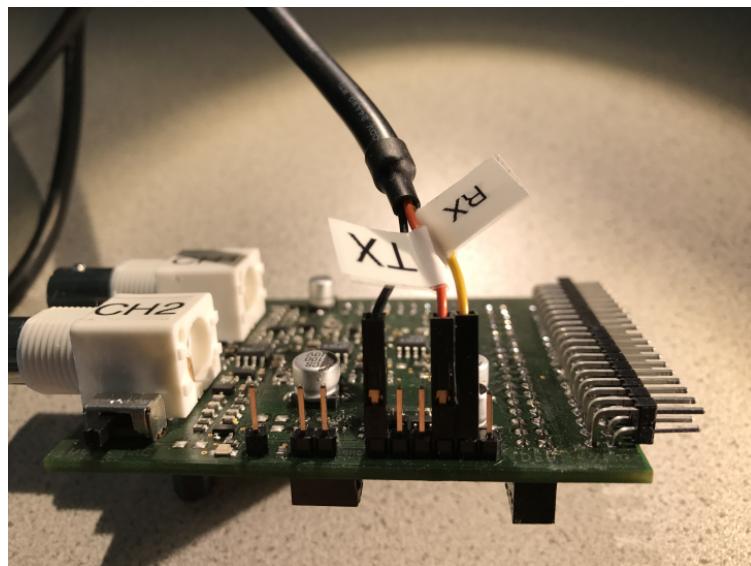


Figure 1.3: Hardware Serial Interface

2 Signal-Toolkit Installation Instruction

The following installation instruction explains in detail the necessary steps to get started with the Signal-Toolkit application. Currently, the application can be executed with the operating systems Linux and Windows. In circumstances, the installation process can only be completed if the user has administrator privileges.

If the application wants to be started on a computer of the UAS (University of Applied Sciences), please boot from the Embedded-Systems partition.

2.1 Linux Distribution - Java and serial device configuration

The first step of the installation instruction is to verify the currently utilized Java version. In Figure 2.1a the Linux terminal is used with the command **java -version** to display the configured Java environment. If the output of the terminal is *java version "1.8.0_xxx"*, continue with subsection 2.1.3. The output of the terminal is *openjdk version "1.8.0_91"*, thus it is necessary to prove which Java program is available. In order to change the default Java environment, enter the following command **sudo update-alternatives --config java**. Figure 2.1b shows that only the *openjdk version "1.8.0_91"* is available. Therefore, it is mandatory to install either the Java Runtime Environment (JRE) or the Java Developer Kit (JDK).

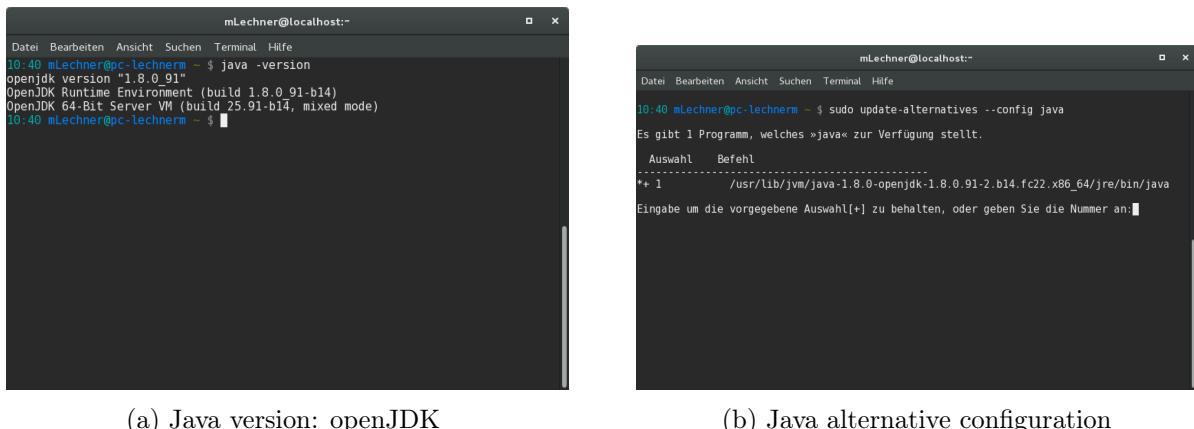


Figure 2.1: Java version and program verification

Before the installation of JRE or JDK, it is mandatory to determine the architecture of the computer. Open a terminal and enter **uname -m**. If the architecture is `x86_64` (64 Bit) the following installation guide¹ should be followed. Otherwise (32 Bit) this² installation instruction should be utilized.

¹https://java.com/en/download/help/linux_x64_install.xml

²https://java.com/en/download/help/linux_install.xml

2.1.1 Installing JAVA using a RPM package (usually Red Hat based distributions Fedora, Korora,...)

For installing the Java runtime, or developer kit (JDK), on RPM package based distributions, download the RPM file from the link mentioned in the previous section. After the download completed the package can be installed by invoking a terminal pointing to the location of the just downloaded file. By using the following command the installation process can be started: **sudo rpm -Uvh jdk-8u^{xxx}-linux-x64.rpm**. Replace the ^{xxx} with the actual version number of the Java RPM package. An example is shown in figure 2.2.

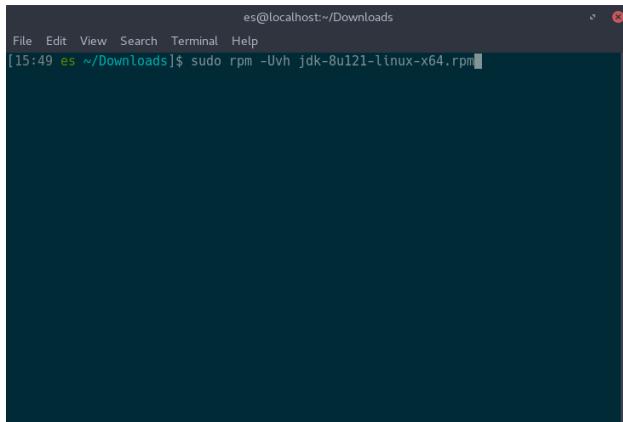


Figure 2.2: Installing the Java RPM package by invoking the RPM command

After the successful installation of JRE or JDK, it is necessary to verify again which Java programs are available, using **sudo update-alternatives --config java** illustrated in Figure 2.3.

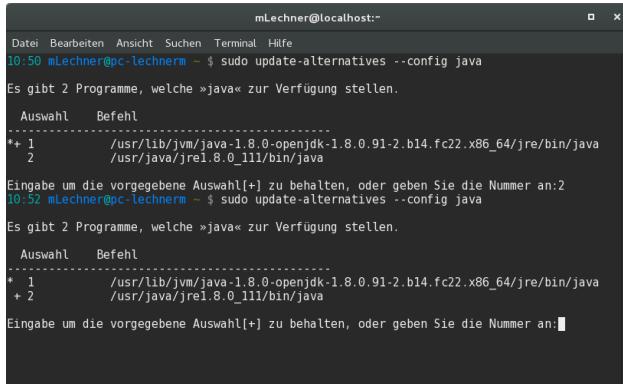
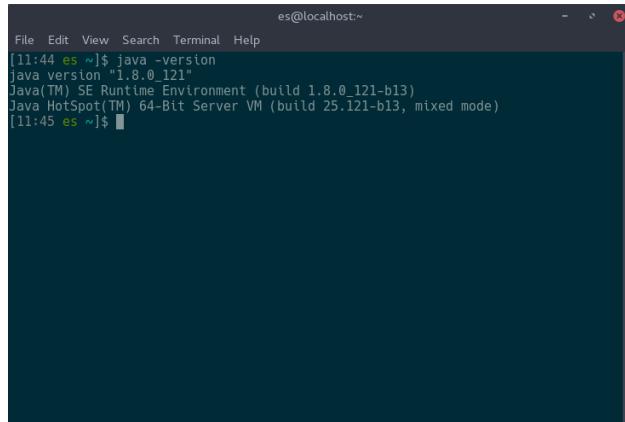


Figure 2.3: Java alternative configuration with installed JRE or JDK

Finally, it is only necessary to select, in this case, the second Java program which is *java version "1.8.0_111"*. The same command is entered again to verify whether the correct Java environment has been chosen.

2 Signal-Toolkit Installation Instruction

A successful installation of Java can be checked by entering the **java -version** command. The output should be similar to figure 2.4. (except the version number)



```
es@localhost:~$ java -version
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
[11:45 es ~]$
```

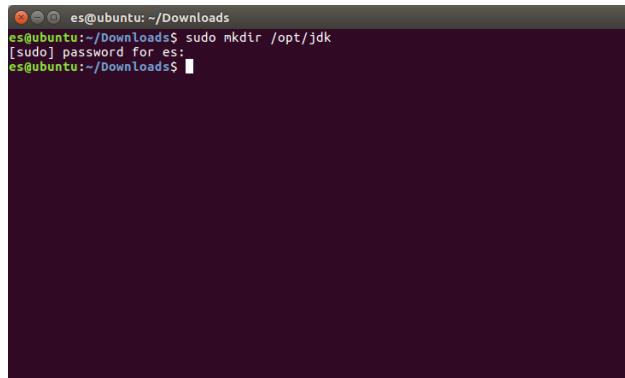
Figure 2.4: Output of "java -version" after correct installation

Verify that the output does not contain *OpenJDK* in any line.

2.1.2 Installing JAVA using a tar.gz package (usually Debian based distributions Ubuntu, Mint,...)

For installing the Java runtime, or developer kit (JDK) via a tar.gz package, download the tar.gz file from the link mentioned in 2.1. In order to copy the necessary files to a proper location, a folder has to be created using the following command: **sudo mkdir /opt/jdk**

This procedure can be seen in figure 2.5.



```
es@ubuntu:~/Downloads$ sudo mkdir /opt/jdk
[sudo] password for es:
es@ubuntu:~/Downloads$
```

Figure 2.5: Creating a folder for the Java installation using a tar.gz package

Next, unpack the tar.gz file which was previously downloaded by entering the following command into a terminal window: **sudo tar -zxf jdk-8u121-linux-x64.tar.gz -C /opt/jdk**
(assuming the terminal already points to the folder locating the tar.gz file)

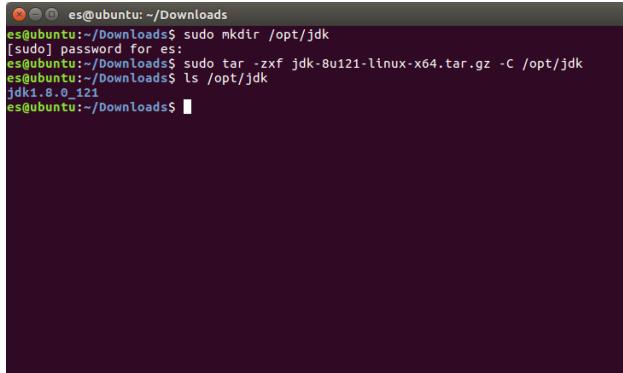
Hint: Keep in mind that the version number could differ from the example!

This command will take a while depending on the used machine. There is no indication of progress. The process is completed after the terminal draws a new line. Any errors will be displayed accordingly.

In order to verify the successful unpack process, the following command should output a folder which is named after the installed java version: **ls /opt/jdk**

2 Signal-Toolkit Installation Instruction

An example of those commands can be seen in figure 2.6.

A screenshot of a terminal window on an Ubuntu system. The user 'es' is logged in. The terminal shows the following commands being run:

```
es@ubuntu:~/Downloads$ sudo mkdir /opt/jdk
[sudo] password for es:
es@ubuntu:~/Downloads$ sudo tar -zxf jdk-8u121-linux-x64.tar.gz -C /opt/jdk
es@ubuntu:~/Downloads$ ls /opt/jdk
jdk1.8.0_121
es@ubuntu:~/Downloads$
```

The terminal window has a dark background with light-colored text. The title bar says 'es@ubuntu: ~/Downloads'. The cursor is at the end of the last command line.

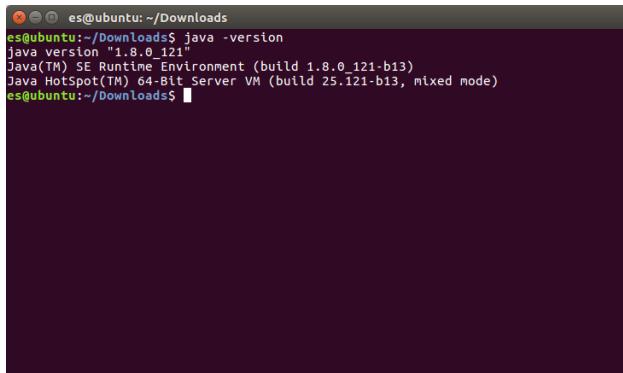
Figure 2.6: A summary of the commands used to install java utilizing a tar.gz package

The final step is to tell the system to actually use the java executable, located in the folder which was previously created, whenever the Java command is invoked. The following two commands configures the system accordingly:

- `sudo update-alternatives --install /usr/bin/java java /opt/jdk/jdk1.8.0_X/bin/java 100`
- `sudo update-alternatives --install /usr/bin/javac javac /opt/jdk/jdk1.8.0_X/bin/javac 100`

Replace X with the version number of the, to be installed, Java package.

A successful installation of Java can be checked by entering the **java -version** command. The output should be similar to figure 2.7. (except the version number)

A screenshot of a terminal window on an Ubuntu system. The user 'es' is logged in. The terminal shows the output of the `java -version` command:

```
es@ubuntu:~/Downloads$ java -version
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
es@ubuntu:~/Downloads$
```

The terminal window has a dark background with light-colored text. The title bar says 'es@ubuntu: ~/Downloads'. The cursor is at the end of the command line.

Figure 2.7: Output of "java -version" after correct installation

Verify that the output does not contain *OpenJDK* in any line.

2.1.3 Serial Device Permissions

After the installation process of Java, it is mandatory to allow a non-default user to use serial devices such as `ttyUSBx`. Hence, a terminal program needs to be opened and the following command should be entered: `usermod -a -G dialout MY_USER_NAME`. To get the current user name, enter `echo "$USER"`. A reboot is necessary before this configuration will be effective.

2.1.4 Execution of the Signal-Toolkit Application

The Signal-Toolkit application can only be launched with the command **java -jar SignalToolkit**, whereas it necessary to be in the same directory as the Signal-Toolkit application.

2.2 Windows Distribution - Java and FTDI driver

The first step of the installation instruction is to verify the currently utilized Java version. In Figure 2.8, the Windows command prompt is used with the command **java -version** to display the configured Java environment. If the output of the prompt is *java version "1.8.0_xxx"*, shown in Figure 2.8a, continue with subsection 2.2.1.

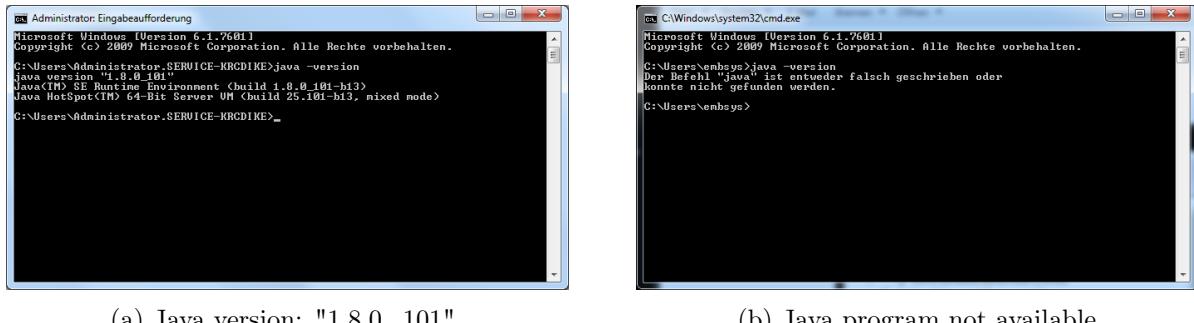


Figure 2.8: Java version verification

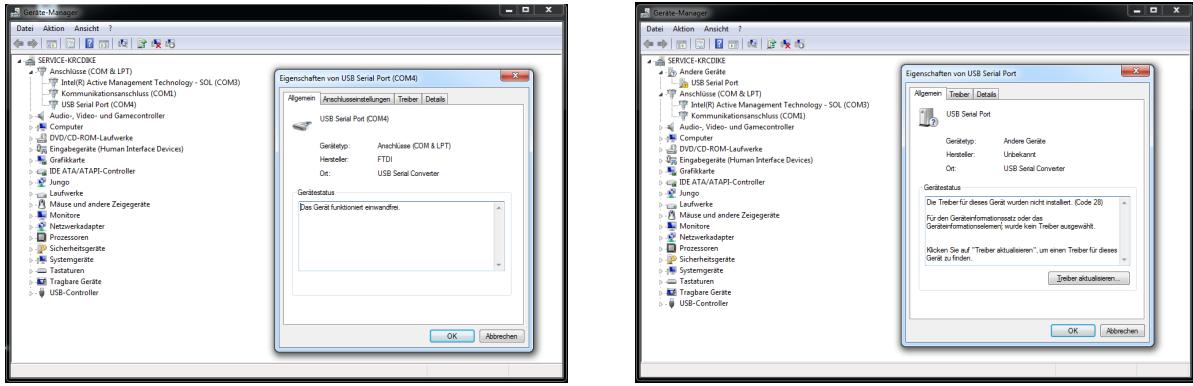
In case no Java environment is installed, the prompt outputs that the command could not be found, shown in Figure 2.8b. Therefore, it is mandatory to install either the Java Runtime Environment (JRE) or the Java Developer Kit (JDK). Before the installation of JRE or JDK, it is mandatory to determine the architecture of the computer. Open the Windows command prompt and enter **systeminfo**. Finally, follow the following installation instruction³ carefully.

2.2.1 FTDI Driver Verification

The following steps are necessary to verify whether the FTDI driver has been installed correctly. Plug the FTDI cable in a free USB port and make sure that the TXD, RXD, and GND cables are not connected to any device. In Figure 2.9a, the FTDI driver was installed correctly and the installation instruction can be continued with subsection 2.2.2.

³https://java.com/en/download/help/windows_manual_download.xml

2 Signal-Toolkit Installation Instruction



(a) Successful FDTI driver installation

(b) FDTI driver installation failed

Figure 2.9: Device Manager - FDTI driver installation verification

Figure 2.9b illustrates that the USB serial device could not be installed. Therefore, it is necessary to install the FTDI driver manually. Open the device manager and make a right click on the unknown device (USB Serial Port) and select 'Update Driver Software' from the context menu. Subsequently, click 'Browse my computer for driver software'. The driver files are in the folder 'FTDI_Driver'. After the installation process, the serial device should work without any problems.

2.2.2 Execution of the Signal-Toolkit Application

The Signal-Toolkit application can be opened with a double click. If the Signal-Toolkit is not opened correctly, make sure you utilize the *Java(TM) Platform SE binary* program to execute Java applications. The Signal-Toolkit application can also be launched with the command **java -jar SignalToolkit**, whereas it necessary to be in the same directory as the Signal-Toolkit application.