Installation Instructions for CppRTA on Raspberry Pi 3

- Download offline version of the NOOBS distribution (.zip, approx. 1,1GB)
  - https://www.raspberrypi.org/downloads/noobs/ [checked March 2017]

- Format the SD card (follow instructions from "INSTRUCTIONS-README.txt" inside .zip archive)

- Unpack NOOBS .zip archive and copy it on the SD card (no subfolders, see "INSTRUCTIONS-README.txt")

- Plug in the SD card, attach a mouse, keyboard and a screen via the USB and HDMI connectors to your Raspberry Pi 3

- Start your RasPi by connecting a 5V, min. 2A power supply via the micro USB connector

- The NOOBS installation routine should appear on the screen.
  - IMPORTANT: Choose correct typeset and installation language at the bottom

- Install Raspibian [RECOMMENDED] (full Debian-like operating system)
  This will take approx. 15-20 minutes, treat yourself to a cup of coffee / tea

- Reboot RasPi to load right language settings.

- Congratulations! Your RasPi is ready to go. Now we have to install several things. For this you need an internet connection.
  - Plug in LAN cable via the appropriate connector or connect with a valid WLAN
  - If you need EDUROAM connection, follow the instructions at the end of this document

- Open command line, update package list and software with commands
  - sudo apt-get update
  - sudo apt-get upgrade

- Install cmake and qt creator IDE with
  - sudo apt-get install cmake
  - sudo apt-get install qtcreator

  NOTE: Using the QtCreator IDE is not required, but recommended. Alternatively, you may use eclipse or code without an IDE (experienced users only). Using QtCreator IDE produces no depencencies to any qt library, it's only a nice IDE which understands cmake on-the-fly and gives the best out-of-the-box-experience for building CppRTA.
  Jaja, so is das

- Install necessary audio reading / streaming libraries and zip. I highly recommend building the audio dependencies from source (don't worry, that's easy):
  - sudo apt-get install libasound2-dev (for ALSA driver development packages)
  - sudo apt-get install zip (for unzipping things)
  - Download port audio, mpg123 and sndfile soruce code from
  https://sourceforge.net/projects/mpg123/files/ [checked March 2017]
  http://www.mega-nerd.com/libsndfile/#Download [checked March 2017]
  http://www.portaudio.com/download.html [checked March 2017]
  - Unpack them and navigate on the command line to the source code folders with
  cd *DirectoryPathHere*
  type following commands after navigating in each source code folder:
  - sudo ./configure
  - sudo make
  - sudo make install
  - This configures and builds all dependencies and libraries for your RasPi. The compilation may take several minutes, be patient for the last time.

- Reboot, download CppRTA and unpack it to a nice working directory (/home/pi/workspace may be a good idea)
  https://github.com/HJ100490/MARTA [checked March 2017]

- Start your IDE and load the project
  - In QtCreator, navigate to the directory and open the CMakeList.txt file for project configuration. Follow the instructions, run cmake and finish configuration. Enjoy!
  - Important: If QtCreator "can't find a working toolkit", it may be unable to find the gcc compiler. Add it manually via the "compiler" slider as gcc compiler, it is located here: /usr/bin/gcc

- Build CppRTA. This will produce a library and a test executable based on MARTAtst.c . Adapt the sound file path inside MARTAtst.c to a valid sound file
  - I recommend to check everything is working properly by executing the test programm. When using the streamFromFileExample, you should hear a lowpassed version of your sound file.

- Now you can start coding! You may like to start by editing my example or build a complete new project based on CppRTA.
  - Note1: You need to link your CppRTA based project against portaudio, mpg123 and sndfile libraries. You may wand to work with Cmake (it's really cool and cross-platform). Check out my CmakeLists.txt file for basic commands / an idea how to start.
  - Note2: You can use / link against MARTA, which is a C-Wrapper for CppRTA. So you can write plain C code.
  - Note3: It is not required to use the MARTA C-Wrapper. If you build C++ projects, you can directly use the CppRTA source files to call the C++ object and its methods directly.

For more explanations, check out the readme in github / MARTA folder.
If there are still any questions or bug reports, you can contact me via email:
<hagenvontronje1[at]gmx[point]de>.

- If you need Eduroam connection in Germany (Telekom root CA2 certificate):

  - Download .crt certificate data ("https://www.uni-giessen.de/fbz/svc/hrz/svc/ident/zertifikat/crt/dt-root-ca-2.crt/view" [checked March 2017])

  - Copy it to your RasPi (via USB-stick, SD-card, ...). Place it in "/home/pi/" IMPORTANT: File name must be exactly "deutsche-telekom-root-ca-2.crt"

  - Open command line (left upper corner), type "sudo leafpad /etc/wpa_supplicant/wpa_supplicant.conf". This opens a text file.

  - substitute the whole text in the file with the text below, fill in your identity and password (For example identity: "ab1234@hs-woe.de", pw: "123abc")

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
ssid="eduroam"
proto=RSN
key_mgmt=WPA-EAP
pairwise=CCMP
auth_alg=OPEN
eap=PEAP
identity="ab1234@hs-woe.de"
anonymous_identity="anonymous@hs-woe.de"
password="123abc"
ca_cert="/home/pi/deutsche-telekom-root-ca-2.crt"
phase2="auth=MSCHAPv2"
}
```

  - Reboot your RasPi and enjoy!