SP Quadcopter System Tutorial

This is a short tutorial for getting the SP Quadcopter System software built and running. It is assumed that everything is done on a computer running Ubuntu Linux 14.04, however, it should be simple to adapt this tutorial to other GNU/Linux distributions. This tutorial does not cover how to assemble the hardware nor how to use the GUIs, but this information might be added in the future.

Installing Dependencies

From a terminal, run

```
sudo apt-get install build-essential qt-sdk qt4-qmake libassimp-dev
pandoc texlive-latex-base cm-super openocd git libudev-dev
libqt5serialport5-dev
```

Install a toolchain for the embedded dependencies (see here)

```
sudo apt-get remove binutils-arm-none-eabi gcc-arm-none-eabi
sudo add-apt-repository ppa:terry.guo/gcc-arm-embedded
sudo apt-get update
sudo apt-get install gcc-arm-none-eabi=4.9.3.2015q3-1trusty1
```

Add your used to the dialout group to access serial ports without being root

```
sudo adduser $USER dialout
```

Modemmanager will block the serial ports for a while after they show up. If you don't need it, uninstall it with

```
sudo apt-get remove modemmanager
```

Add this point you should be able to build and upload all the software.

Building and Installing the Simulator

From the *Linux* directory, run

```
./build
```

After the build finishes, start the simulator and QuadcopterTool with

```
./start_sim
```

QuadcopterTool

QuadcopterTool is built with the same build script in the *Linux* directory as the simulator, so if you have run it you can start QuadcopterTool with

```
./start_QuadcopterTool
```

Embedded Software

A programmer has to be connected to the embedded boards as described here

QuadcopterMain
Connect the programmer to the quadcopter mainboard. Go to the *Embedded/QuadcopterMain* directory and run

```
make upload
```

This will build and upload the firmware to the mainboard of the quadcopter.

Receiver Board
The receiver (and transmitter) board is used to connect QuadcopterTool to the quadcopters. It is based on a stm32-bv mote. Connect a programmer to it and go to the *Embedded/Range* directory. Run

```
make upload
```

Disconnect the programmer and connect a mini-USB cable to the mote. Now QuadcopterTool can connect to it and should be able to visualize and control the quadcopters that are powered on and programmed.

To control the quadcopters, a simulator remote controller can be connected to the Linux computer, which QuadcopterTool can connect to as a joystick. It is also possible to uncomment the line

```
#define ENABLE_PPM_OR_PWM
```

in the main.c file of the receiver. Then a PPM signal from a remote controller can be connected to PC6 of the stm32-bv mote, and the quadcopter can be controlled without QuadcopterTool. Further, it is also possible to run QuadcopterTool on one stm32-bv mote while running another stm32-bv mode with a PPM signal simultaneously.

Embedded Software for the Localization System

The following software and hardware is required to get the localization system running.

Border router mod
The border router with the clock sync modification runs on a stm32-bv mote. First connect a programmer to it and then go to the *Embedded/border_router_clock_sync* directory and run

```
make upload
```

Disconnect the programmer and connect a mini-USB cable to the router mote. Then, from the same directory, in order to start the router, run

```
make connect-router TTY_PORT=/dev/ttyACM0
```

This will start the border router.

Ranging Board
Connect the programmer to the ranging board of the quadcopter. Go to the *Embedded/Range* directory and open Makefile. Make sure that the *stm32-bv-copter* hardware is used and that no NODEID is selected. Then run

```
make upload
```

Anchors
The range software is for both the anchors and the ranging board, but the Makefile configuration differs. Connect the programmer to an anchor (which is a stm32-bv mote with an ultrasound transmitter connected to it). Go to the *Embedded/Range* directory and open Makefile. Make sure that the *stm32-bv* hardware is used and that NODEID for the corresponding anchor is selected. Then run

```
make upload
```

Repeat this for all the anchors.

Generating PDF Documentation

If the dependencies above are installed, PDF documentation files can be generated by running the *build_doc* script

```
./build_doc
```