

---

FWS  
Flight Weather Station

---

Projektpraktikum

Johannes Kasberger  
0616782

Markus Klein  
0726101

# Contents

<b>Contents</b>	<b>I</b>
<b>List of Figures</b>	<b>I</b>
<b>1 Introduction</b>	<b>2</b>
1.1 About FWS . . . . .	2
<b>2 Slave</b>	<b>3</b>
<b>3 Master</b>	<b>4</b>
3.1 About the Master . . . . .	4
3.2 View . . . . .	5
3.3 Configuration . . . . .	6
3.4 Stations . . . . .	II
3.5 History . . . . .	II

# List of Figures

3.1 Quick overview about the stations and their status . . . . .	5
3.2 Configuration of the basic settings . . . . .	6
3.3 Adds a new station to the master . . . . .	6
3.4 Current Wind direction plot . . . . .	IO
3.5 last 24 hours of two seperat parameters in one plot . . . . .	II
3.6 Axis description of direction parameter . . . . .	II

## CHAPTER I

# **Introduction**

### **I.I About FWS**

## CHAPTER 2

# **Slave**

## CHAPTER 3

# Master

### 3.1 About the Master

The master is written Java and uses the jamod Library<sup>1</sup> for the communication with the sensors over the modbus protocol. The view is created with the SWT Library<sup>2</sup>. It's separated in several parts:

- View (see 3.2)
- Configuration (see 3.3)
- Stations (see 3.4)
- History (see 3.5)

It's needed to create a configuration file before collection process of the measurement values could be started. A sensor must be added to the master before it can be used. Each sensor has a individual name and ip address and it's referred as a station. In the configuration phase it's necessary to specify the kind of data that the stations collect. The data that is transferred from the station to the master is called Input Parameter. To configure the sensor it's possible to define configuration Parameters. These values are transferred from the master to the sensor. To map the parameters to the memory in the station they must be bound to addresses. This happens individually for each station. So it's possible to collect different data from different stations. It's also possible to configure the kind of plots that are generated (see 3.3).

After the configuration of the stations the collection process could be started. Each station is polled in its own thread. One collector thread collects the data from all stations, generates a text file with the information

---

<sup>1</sup><http://jamod.sourceforge.net/>

<sup>2</sup><http://www.eclipse.org/swt/>

about the current values from the stations and draws the plots. The time between the polling of the stations and between generation of the output files can also be configured. As soon the current day changes the values from the last day are aggregated to a history value. This daily history is kept for one year. The values from the station are kept for two days so it's possible to plot hourly data.

## 3.2 View

The view is created with the help of SWT. SWT uses the os drawing apis to draw the widgets. So the look and feel of the application is very good integrated in the os it runs in.

### Screenshots

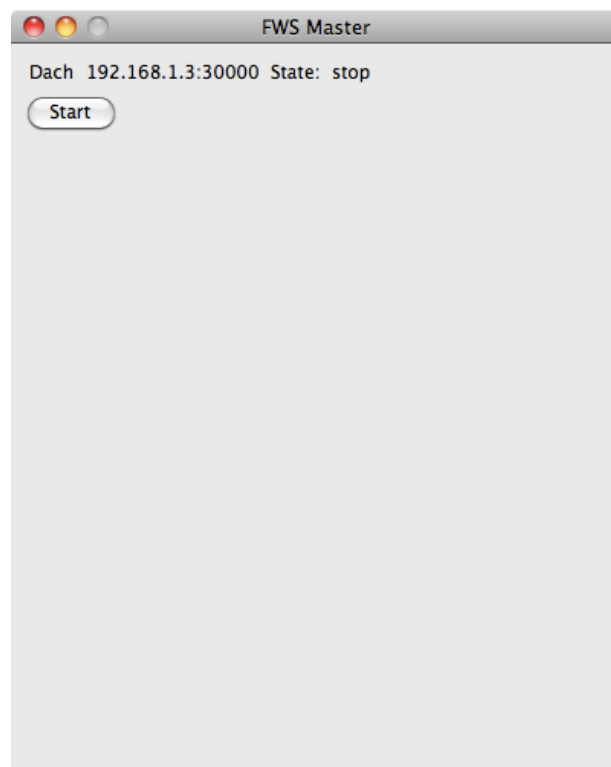


Figure 3.1: Quick overview about the stations and their status

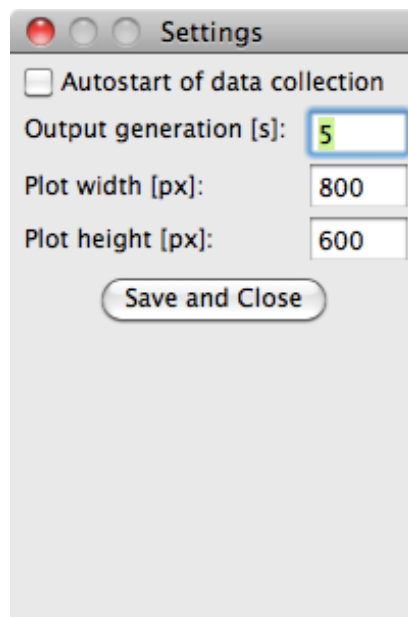


Figure 3.2: Configuration of the basic settings

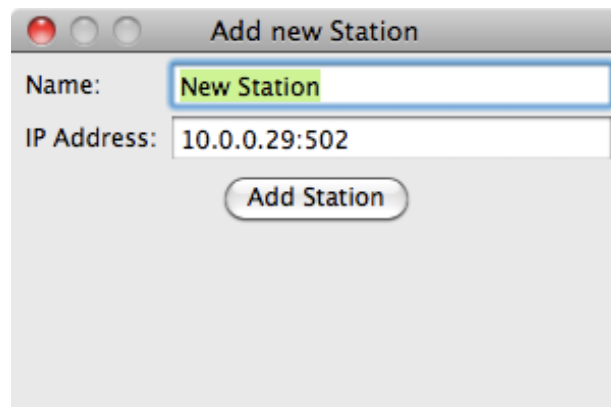


Figure 3.3: Adds a new station to the master

### 3.3 Configuration

The necessary views for the configuration are 3.3, 3.2,

The result of the configuration phase is an xml file with all settings in it.  
See example in Listing 3.1

Listing 3.1: Sample settings file

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```

<fws_config>
<path>/home/user/FWSMaster/output</path>
<generatortime>5</generatortime>
<autostart>>false</autostart>
<plotwidth>800</plotwidth>
<plotheight>600</plotheight>
<parameter typ="config">
<name>Messintervall</name>
</parameter>
<parameter typ="input">
<name>Temperatur</name>
<unit>°C</unit>
<format>6553.6</format>
<history>MAX</history>
</parameter>
<parameter typ="input">
<name>Windrichtung</name>
<unit>Richtung</unit>
<format>65536</format>
<history>AVG</history>
</parameter>
<station intervall="2" ip="192.168.1.3:30000" name="Dach">
<binding active="true" address="3" parameter="Messintervall"
    transfered="false" type="config" value="1000"/>
<binding active="true" address="0" parameter="Temperatur" plotconfig="
    d4;1h24;" type="input"/>
<binding active="true" address="1" parameter="Windrichtung" plotconfig
    ="d4;c1;h24;" type="input"/>
</station>
</fws_config>

```

## Parameters

There are two types of parameters. Input Parameters and Configuration Parameters. Each parameter has a unique name and a boolean value if it's enabled. The name is the identifier of the parameters so it has to be unique. Beside of these common attributes the configuration and input parameters have further attributes.



### Input Parameter

Additional attributes of an input parameter:

- Unit
- Format
- History Function

**Unit** The Unit is just used for the description in the generated files. Available units are:

- speed  $\frac{m}{s}$
- speed  $\frac{km}{h}$
- frequency  $Hz$
- direction
- temperature  $^{\circ}C$

**Format** The transferred value from the station is a 16 bit integer value. To be able to display floating point numbers it's possible to set the output format to the desired form. Example: The temperature equals  $23.3^{\circ}C$ . The station measures the temperature and converts it to 233 to have an integer value. On the master the output format is set to the format with one digit before the decimal point. When such an input parameter is used it's converted back to 23.3.

**History Function** There are three different history functions available. For more details about these functions and how they are used please read 3.5.

- average
- minimum
- maximum

### Configuration Parameter

Additional attributes of a configuration parameter:

- value

**Value** A configuration parameter is a value that is transferred from the master to the slave. The value that is transferred is saved in the attribute value. This value must be an 16 Bit integer value.

## Plots

The plots are generated with the help of the jFreeChart Library<sup>3</sup>. The plots can be configured for every binding. Each binding can have several plots assigned. It's also possible to plot more than one data into one plot. The plots are configured with a string. The simplest configuration looks like C h24; With that configuration one plot is generated and the data for this plot are the values from the last 24 hours. The plots are generated in the output directory that can be set from the user.

It's possible to use different data ranges for the plots. To allow the use to choose one range are three different time bases available:

- c - current
- h - hours
- d - days

**Current** Use the last values for the plot. Currently this is only implemented for the wind direction. Results in a plot like Figure 3.4.

**Hours** Use the last values of the last hours for the plot. The amount of the hours is specified in the plot configuration. It's the number after the timebase. Example of an 24 hour plot see Figure 3.5.

**Days** Same plot as with the hours timebase but the values from the last days are used. For more details about the history see chapter 3.5.

## Configuration Syntax

Each plot is specified by three different parts: [Number]CharacterNumber;

**First Number: ID** This number is optional and controls if more than one input parameter is drawn in one plot. All configured plots with the same number are drawn in the same plot.

---

<sup>3</sup><http://www.jfree.org/jfreechart/>

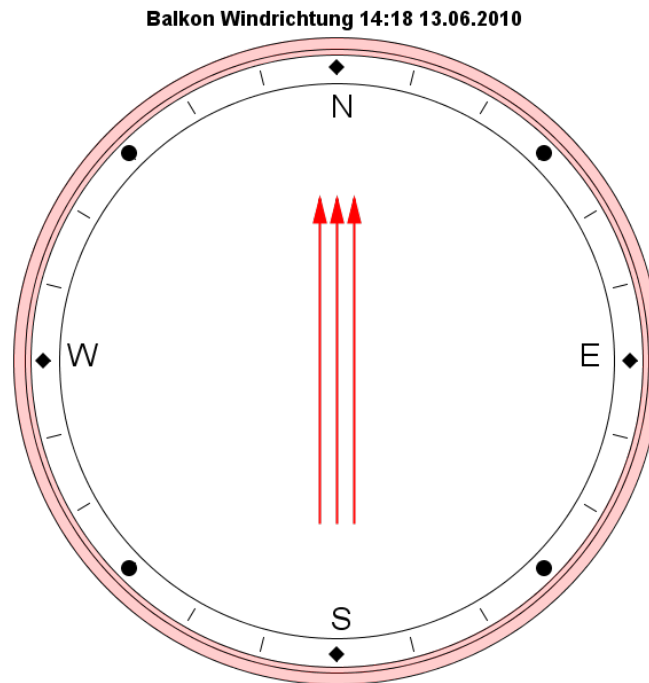


Figure 3.4: Current Wind direction plot

**Character: Timebase** Defines the timebase. Explained in paragraphs above.

**Second Number: Ammount of data** The second number defines how much data will be in the plot. d4; will plot the last four days.

**End of Configuration** Each configuration must end with an `;`.

**Example** For the configuration string `h24;1h24;d30;c1;d365;` following plots are generated:

- Last 24 hours
- Last 24 hours with other data with ID 1
- Last 30 days
- Current value
- Last 365 days

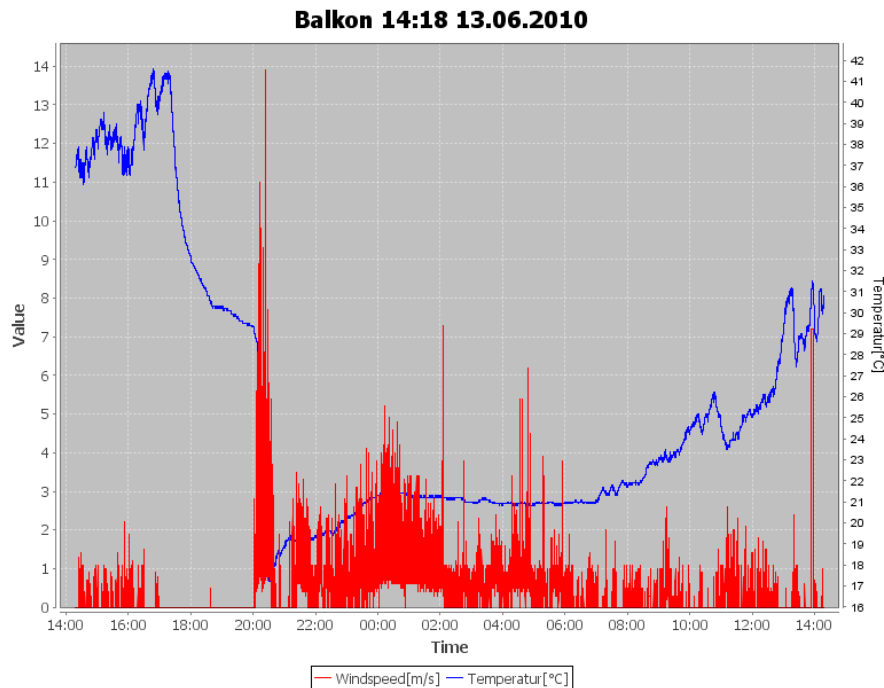


Figure 3.5: last 24 hours of two separat parameters in one plot

### Detailed Information about the plots

If two differnt data sets should be plotted in one plot there two axis with own scale generated. When more that three data sets should be drawn in one plot they share one axis. So it's advisable to generate more plots with two datasets each in it to kepp the plots tidy.

The direction parameters values are not connected with a line. Otherwise the plot would be unreadable if the wind directions changes a lot.

The direction parameter values are mapped to a description of the direction. So  $0^\circ$  result in N(orth) see figure 3.6

## 3.4 Stations

## 3.5 History

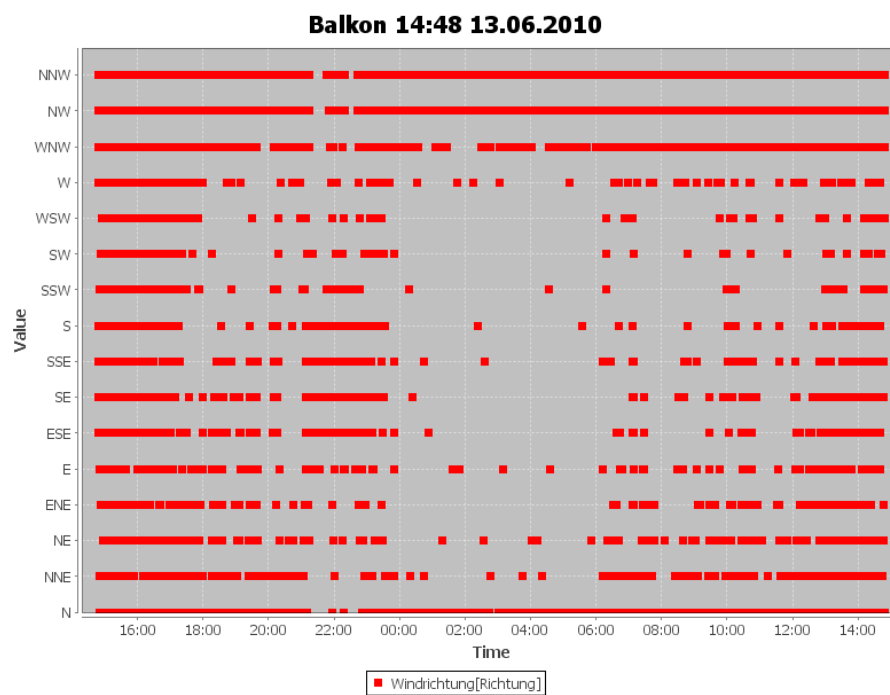


Figure 3.6: Axis description of direction parameter