Monitoring Tools File Specification, Version 1.0

Technical Report

Stefan Vogelsang, Andreas Söhnchen

Institut für Bauklimatik
Technische Universität Dresden
D-01062 Dresden, Germany
stefan.vogelsang@tu-dresden.de

Abstract

This paper describes the format of monitoring data files that are collected for external measuring sites and at laboratory experiments at the Institute for Building Climatology (IBK). The Monitoring Data Files are containers for storing time series or event driven data collected as input for transient heat and moisture transport simulations. Further applications are the documentation of real world behaviour, laboratory experiments or the collection of validation data sets for simulation results (whole building / energy consumption / HAM). The article also discusses the application interface towards measurement data verification tools as well as data storage solutions that can be used to archive *measurement* data files conveniently and efficiently.

Contents

1 Introduction

Hygrothermal and thermal transport models require a number of measurement data for input and verification purposes. Depending on model setup and simulations goals, different kinds of measured input data may be needed. The Monitoring File Format is a container collecting event driven as well as time series data for different models of measurement data use, e.g. laboratory experiments or field data collections. It serves as standard container and file-based database format for the measurement database of the Institute of Building Climatology (Institut für Bauklimatik, IBK) of the Technische Universität Dresden, Germany.

The file format was designed to cope a sensor structure of multiple tiers (Figure 1). No matter if a sensor is connected directly to the dedicated hardware, or to a logger, or to a network of loggers the file format must be able to identify every sensor in an unique way. To ensure this requirement the root level or entry point for an measurement acquiring software must be marked.

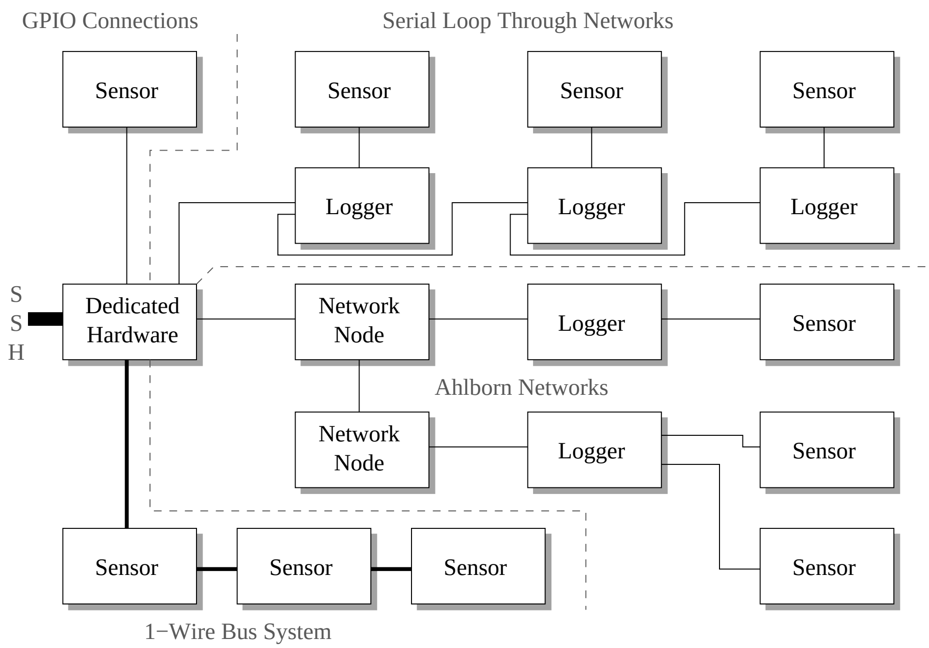

Figure 1: Examples for Measurement Networks

In consequence a unique identifier must be provided by the dedicated hardware device for directly connected measurement hardware, like 1-Wire, or GPIO connections. Ahlborn and M-Bus systems represent closed networks and thus define their own root level identifier. To gain best results with our framework and this file specification the user shall ensure that these different entry points can be uniquely identified over all measurement projects he ever might use.

To combine measurements from a variety of sources (e.g.: Ahlborn measuring networks, simple DHT-22 sensor, M-Bus Relay WebLog 250, etc.) a bunch of monitoring programs are executed on an dedicated hardware (currently Raspberry Pi B+). This common hardware synchronises itself with atomic clocks around the world and thus allows data acquirement under a common time regime with an synchronisation offset of only a few milliseconds. All time stamps defined in this specification are referred to as local time and daylight saving time is neglected. It is up to the user to decide if all dedicated hardware devices are synchronised to one specific time zone or to the according time zone a measurement is taking place.

Since each data source provides its own data format a conversion to the format described in this document is executed at the time a date is retrieved from its source. The file format is a comma separated ASCII format designed for easy manipulation and editing via plain text editors or scripts. This article describes the Monitoring File Format Version 1 with the standard file extension `csv`.

Minimalistic Example

Providing a brief overview of the measurement file format, we include a minimalistic example of a time series measurement data file. This file can be used, for example, as input to the monitoring framework described in [2].

Listing 1: Minimalistic Measurements Data File Example for a Monitoring of an Ahlborn Network

```
Channel,Rasp01_SerAhl_G01,M00,M10,Rasp01_SerAhl_G02,M01,,,M11
SensorID,Rasp01_SerAhl_G01,H2W3_AK_T,H2W3_AK_RH,Rasp01_SerAhl_G02,H2W3_MW_T,,,H2W3_MW_RH
Unit,Rasp01_SerAhl_G01,K,%,Rasp01_SerAhl_G02,C,,,%
Quantity,Rasp01_SerAhl_G01,T,rH,Rasp01_SerAhl_G02,T,,,rH
GW-Max,Rasp01_SerAhl_G01,,,Rasp01_SerAhl_G02,,,
GW-Min,Rasp01_SerAhl_G01,,,Rasp01_SerAhl_G02,,,

2016-02-15 00:00:00,Ahlborn_G01,273.15,97.8,Rasp01_SerAhl_G02,-11.5,,,5.49
2016-02-15 00:05:00,Ahlborn_G01,273.15,95,Rasp01_SerAhl_G02,-13.5,,,5.49
```

This example measurement file begins with the header (see §3) separated from the data section by a blank line. The data section in this example contains just two data lines, each line representing a specific measurement time point. Please note that the number of commas must be equal in each line of the file except the empty line that separates the header from the data section.

2 File Name Conventions

While not part of the file format specification itself, a naming convention of file names is meaningful to organize multiple measurement data files and facility file-based data storage. Each measurement data file shall be formed using the following pattern:

```
<ProjectName>_<Date>_<Time>.csv
```

`<ProjectName>` corresponds to a descriptive name of the measurement project and the corresponding measurement entity. It is primarily used to organize and identify measurement files when working directly with file names. Within measurement project file names only upper and lower case letters are valid. No non printable, or special characters, or umlauts are allowed. It is recommended to use a unique abbreviation for the whole project, a project unique acronym for the construction site, as well as an acronym for the measured construction detail to create a new ID that is unique in general for all projects to keep track of a history of changes and allow easy parsing and handling of by frameworks operating on measurement data and sensor specifications.

`<Date>` is the date when this file was created and the incorporated data was acquired. The format of a date is defined as: `<yyyy>-<mm>-<dd>`, where month and day must have two digits. Thus leading zeros must be inserted if necessary. The format includes leap years days.

<Time> is the time point when the data file was created and the first data line was included into the data set. The format of a time stamp is defined as: `<hour>-<minute>-<second>`, where hour, minute and second must have two digits. Thus leading zeros must be inserted if necessary.

3 Headers

The overall number of header lines is not restricted. All header lines are addressable by a common column index and column index 0 must contain a *LineType* that allows an interpretation of the corresponding HeaderDataValues in this line. The IBK provided libraries and data bases support a set of *LineTypes*: `Channel`, `SensorID`, `Unit`, `Quantity`, `GW-Max`, `GW-Min`, and `Description`. This set may be extended by the user to add additional header lines at will. A header line can be composed as follows:

```
<HeaderLine>:=<LineType>,<HeaderCont>
<HeaderCont>:=[,<UniqueLEID><HeaderData><HeaderCont>|,<UniqueLEID><HeaderData>]
<UniqueLEID>:=<HardwareDeviceID>_<ProtocolID>[|_<AhlbornLoggerID>|_<MBusLoggerID>]
<HeaderData>:=[,<HeaderDataValue><HeaderData>|,<HeaderDataValue>]
```

Where:

<UniqueLEID> is an identifier that must be unique for all data logging environments a user currently runs.

<LineType> is an identifier that allows an interpretation of the HeaderDataValues of the corresponding header line.

- **Channel** defines sensor hardware ids or channel numbers that address a sensor.
- **SensorID** is supposed to be used with user defined designators that clearly describes a sensor in a human readable way.
- **Unit** defines units strings for each sensor column. A list of IBK supported unit strings can found in § 5.
- **Quantity** is the name of the physical quantity a sensor data field can be assigned to.
- **GW-Min** is a lower bound for valid sensor measurement, vice versa all measurement above GW-Min are defined as valid. A corresponding data column may contain a '-' character if a sensor measures a value less than this bound (e.g. Ahlborn sensors).

GW-Max is an upper bound for valid sensor measurement, vice versa all measurement below GW-Max are defined as valid. A corresponding data column may contain a '-' character if a sensor measures a value greater than this bound (e.g. Ahlborn sensors).

Description is a user defined field to create a custom human readable header to the data provided in the data section.

<HardwareDeviceID> is a user defined unique ID defined for the dedicated hardware device that executes all monitoring tools.

<ProtocolID> is a string that represents the protocol run for this measurement. It is used to allow different interpretation of additional header lines by superordinated software layers, e.g. monitor verification framework. At IBK is currently a string from the set (`1Wire`, `DHT22`, `Event`, `SerAhl`, `MBus`).

<HeaderData> is defined as combination of an unique identifier and its corresponding data. The number of data separated by comma must match with the number of header information provided in the header for this sensor identifier.

<HeaderDataValue> is an alphanumerical value representing the column assignable header information.

To finish a header an empty line must be appended. The IBK monitoring framework utilizing this specification creates a copy of each header file on the very first initialisation of a dedicated measurement hardware. This allows a later comparison to header obtained at system runtime to ensure that missing sensors or missing networks of sensors are acknowledged every time the monitoring software instances is executed.

3.1 Specifics on Time Series Header Files

3.1.1 Monitoring files for Ahlborn Loggers

Ahlborn loggers may be used as loop through as well as a network of loggers representing multiple sensors per logger. Thus multiple Logger may be represented in one data line.

Example:

Listing 2: Example for an Ahlborn Network Header Section

```
Channel,Rasp01_SerAhl_G01,M00,M10,Rasp01_SerAhl_G02,M01,,,M11
SensorID,Rasp01_SerAhl_G01,H2W3_AK_T,H2W3_AK_RH,Rasp01_SerAhl_G02,H2W3_MW_T,,,H2W3_MW_RH
Unit,Rasp01_SerAhl_G01,K,%,Rasp01_SerAhl_G02,C,,,%
Quantity,Rasp01_SerAhl_G01,T,rH,Rasp01_SerAhl_G02,T,,,rH
GW-Max,Rasp01_SerAhl_G01,,,Rasp01_SerAhl_G02,,,
GW-Min,Rasp01_SerAhl_G01,,,Rasp01_SerAhl_G02,,,
```

3.1.2 Monitoring files for DHT 22

Hardware device utilized by IBK are restricted to 9 monitoring contacts at once. The header data value in the first header row are representing the gpio channels used to trigger the DHT22 sensors. Since this sensors do not provide an own hardware ID the dedicated hardware identifier in combination with the protocol is used to create a unique logging instance for the monitoring software and the verification framework that evaluates the measured data.

Example:

Listing 3: Example for DHT22 Header Section

```
Channel,Rasp01_DHT22,8,8,9,9
SensorID,Rasp01_DHT22,R_11_1.11_T,R_11_1.11_rH,R_11_1.12_T,R_11_1.12_rH
Unit,Rasp01_DHT22,C,%,C,%
Quantity,Rasp01_DHT22,T,rH,T,rH
```

3.1.3 Monitoring files for 1-Wire

All restrictions regarding 1-Wire bus protocol and topologies are specified by Dallas Semiconductor Inc. and Maxim Inc. [1].

Example:

Listing 4: Example for a 1Wire Header Section

```
Channel,Rasp01_1Wire,28.F5A685050000,28.DC91A0050000
SensorID,Rasp01_1Wire,R_11_1.12_TWK,R_11_1.12_TWW
Unit,Rasp01_1Wire,C,C
Quantity,Rasp01_1Wire,T,T
```

3.1.4 Monitoring files for M-Bus Modules

Example:

Listing 5: Example for an MBus Header Section

```
Channel,Rasp01_MBus_68055534-5,0,0,1,2
SensorID,Rasp01_MBus_68055534-5,SW-18-2-05-WMZ,SW-18-2-05-WMZ,SW-18-2-05-TK,SW-18-2-05-TW
Unit,Rasp01_MBus_68055534-5,kWh,m3,m3,m3
Quantity,Rasp01_MBus_68055534-5,Q,V,V,V
Description,Rasp01_MBus_68055534-5,Energie,Volumen,Volumen,Volumen
```

3.2 Specifics on Event Driven Header Files

3.2.1 Monitoring files for MCP3008

Hardware device utilized by IBK are restricted to 8 monitoring contacts at once. The header data value in the first header row are representing the channel at the MCP3008 level shifter which is connected to the dedicated hardware by SPI Bus. Since this shifter does not provide an own hardware ID the dedicated hardware identifier in combination with the protocol is used to create a unique logging instance for the monitoring software and the verification framework that evaluates the measured data.

Example:

Listing 6: Example for an Event Driven Header Section

```
Channel,Rasp01_Event,0,1,2
SensorID,Rasp01_Event,R_11_1.11_W,R_11_1.12_W,R_11_1.13_W
Unit,Rasp01_Event,-,-,-
```

4 Data Section Format Description

Data sets for a sensor network node are collected at an specific time point. Each time point is represented by a single line in the ASCII format. Line breaks Since multiple sensors in a network of sensors may provide their corresponding data for such a time point, each sensor data set (`<SensorData>`) consists of a sensor id followed by its comma separated measured data set. Since the number of network nodes or data loggers in an network is not generally restricted the number of sensor data sets is unlimited, and thus defined by recursion:

```
<DataLine> := <Date> <Time><SensorData>
<SensorData> := [,<UniqueLEID>,<Data><SensorData>|,<UniqueLEID>,<Data>]
<Data> := [,<DataValue><Data>|,<DataValue>]
```

Where:

`<Date>` is the date when the data line was captured from the sensor. The format of a date is defined as: `<yyyy>-<mm>-<dd>`, leading zeros must be inserted if necessary. The format includes leap years days.

<Time> is the time point when the data line was captured from the sensor. The format of a time stamp is defined as: `<hour>-<minute>-<second>`, where hour, minute and second must have two digits. Thus leading zeros must be inserted if necessary. Please note there is no comma between date and time!

<SensorData> is defined as combination of an sensor identifier and its corresponding data. The number of data separated by comma must match with the number of header information provided in the header for this sensor identifier.

<DataValue> is a numerical value where c-locale support is used, meaning a decimal point is represented by an '.' character.

The format is designed to allow index based access for each sensor and its data. In case of an temporarily missing logger a padding of commas must be added in such a way that all other data indices are kept synchronous to the definition in the header file.

Example:

Listing 7: Example for Data Section Line

```
2016-02-15 00:00:00,Rasp01_SerAhl_G01,273.15,45.49,Rasp01_SerAhl_G02,-11.5,,,97.8
```

5 SI Unit Strings

To allow maximum compatibility the following SI unit strings are supposed to be used by any monitoring software implementing support for this specification. This list fully compatible to all IBK related libraries and software. The following table contains all IBK supported SI unit strings and its conversions.

Si Unit	Conversions
m	* 1e+03 mm; * 1e+02 cm; * 1e+01 dm
mm/mK	
mm/m	
m3m/m3m	* 1e3 m3mm/m3m
m/s	* 100 cm/s; * 360000 cm/h; * 8.64e+06 cm/d
m/s2	
m2	* 1e+06 mm2; * 1e+04 cm2; * 1e+02 dm2
m2/s	* 10000 cm2/s; * 3600 m2/h; * 3.6e+07 cm2/h
m2/kg	
m2s/kg	
m2K/W	
m3	* 1e+09 mm3; * 1e+06 cm3; * 1e+03 dm3
m3/m2s	* 3600 m3/m2h; * 1000 dm3/m2s; * 3.6e+06 dm3/m2h
m3/m3	* 100 Vol%
m3/m3d	* 100 Vol%/d
m3/s	* 3600 m3/h; * 1000 dm3/s; * 3.6e+06 dm3/h
kg	* 1000 g; * 1e+06 mg
kg/ms	
kg/s	* 3600 kg/h; * 86400 kg/d; * 8.64e+07 g/d; * 1e6 mg/s; * 1e9 µg/s
kg/m2	/ 100 kg/dm2; * 10 g/dm2; / 10 g/cm2; * 1e6 mg/m2
kg/m2s	* 1e3 g/m2s; * 3.6e+06 g/m2h; * 3600 kg/m2h; * 1e6 mg/m2s; * 1e9 µg/m2s; * 3.6e9 mg/m2h; * 3.6e12 µg/m2h
kg/m3	/ 1e3 kg/dm3; * 1 g/dm3; / 1e3 g/cm3; * 1e3 g/m3; * 1e6 mg/m3; * 1e9 µg/m3; %0 log(kg/m3); %0 log(g/m3); %0 log(mg/m3); %0 log(µg/m3)
kg/m3s	* 1e3 g/m3s; * 3.6e+06 g/m3h; * 3600 kg/m3h; * 1e6 mg/m3s; * 1e9 µg/m3s; * 3.6e9 mg/m3h; * 3.6e12 µg/m3h
kg/m3sK	* 1e3 g/m3sK; * 3.6e+06 g/m3hK; * 3600 kg/m3hK; * 1e6 mg/m3sK; * 1e9 µg/m3sK; * 3.6e9 mg/m3hK;* 3.6e12 µg/m3hK
kg/m	* 1000 g/m; * 1 g/mm; / 1000 kg/mm
kg/kg	* 1000 g/kg; * 1e6 mg/kg
mg/l	/ 1000 g/l; / 1e6 kg/l
J	/ 1000 kJ; / 1e+06 MJ; / 3.6e+09 MWh; / 3.6e+06 kWh; / 3600 Wh
J/m2	/ 1000 kJ/m2; / 1e+06 MJ/m2; / 1e+09 GJ/m2; / 1e+02 J/dm2; / 1e+04 J/cm2; / 3.6e+06 kWh/m2

Si Unit	Conversions
J/m3	* 1 Ws/m3; / 1000 kJ/m3; / 1e+06 MJ/m3; / 1e+09 GJ/m3; / 1e+03 J/dm3; / 1e+06 J/cm3; / 3.6e+06 kWh/m3
J/m3s	/ 1000 kJ/m3s; / 1e+06 MJ/m3s; / 1000 J/dm3s; / 1e+06 J/cm3s; * 3600 J/m3h;
J/m3s	* 1 W/m3; / 1000 kW/m3; / 1e+06 MW/m3; / 1000 W/dm3; / 1e+06 W/cm3; / 1e+09 W/mm3
J/m3K	/ 1000 kJ/m3K
J/s	* 3600 J/h; * 86400 J/d; * 86.4 kJ/d; * 1 W / 1000 kW; / 1e+06 MW
J/kg	/ 1000 kJ/kg
J/kgK	/ 1000 kJ/kgK; * 1 Ws/kgK
J/K	/ 1000 kJ/K
J/m2s	* 1 W/m2; / 1000 kW/m2; / 1e+06 MW/m2; / 100 W/dm2; / 10000 W/cm2
W/Person	/ 1e3 kW/Person
W/K	
W/mK	/ 1000 kW/mK
W/m2s	* 3600 W/m2h; / 1e3 kW/m2s; / 1e6 MW/m2s; / 100 W/dm2s; / 1e4 W/cm2s
W/m2K	
W/m2K2	
W/mK2	
l/m2s	* 3600 l/m2h; * 86400 l/m2d; * 86400 mm/d; * 3600 mm/h
l/m3s	* 3600 l/m3h
s/m	
s2/m2	
s	/ 60 min; / 3600 h; / 86400 d; / 3.1536e+07 a; %0 sqrt(s); %0 sqrt(h)
s/s	/ 60 min/s; / 3600 h/s; / 86400 d/s; / 3.1536e+07 a/s
---	* 100 %
---/d	* 100 %/d
K	- 273.15 C
K/m	
1/K	
Pa	/ 10 daPa; / 100 hPa; / 100 mbar; / 1000 kPa; / 1e5 bar
1/Pa	
Pa/m	/ 1000 kPa/m
Lux	/ 1000 kLux
Rad	* 57.2958 Deg
1/m	/ 100 1/cm
logcm	
logm	
logPa	
K/Pa	
mol/kg	/ 1000 mol/g
kg/mol	* 1000 g/mol
J/mol	/ 1000 kJ/mol
kg/m2s05	* 60 kg/m2h05
1/logcm	
mol/m3	/ 1000 mol/ltr; / 1000 mol/dm3; / 1e+06 mol/cm3
mol	* 1e+03 mmol
-	
1/s	* 60 1/min; * 3600 1/h
Person/m2	
PPM	/ 1e3 PPB
A	* 1e3 mA
Ohm	* 1e3 mOhm
V	* 1e3 mV
mS	/ 1e3 S

6 Competition Law Advice

The naming of products or names of manufacturers, enterprises, or companies is solely for the purpose of information and is not a recommendation of the product or company. I do not guarantee the reliability of any named products.

7 Liability for external Links

All referenced work is solely for the purpose of providing information. The author is not responsible for the content or reliability of external references.

References

[1] Maxim Integrated. 1-wire. Webpage, February 2016. https://www.maximintegrated.com/en/products/digital/one-wire.html.

[2] Vogelsang, Stefan and Söhnchen, Andreas. Monitoring Verification Specification. Technical report, Technische Universität Dresden, 2016. InPrep.