

WH1080 Utilities

Users Guide

Document Number: DAZW-WG01.002

March 2014

This document provides configuration and usage guidelines for the zxweather utilities for WH1080 compatible weather stations.

Revision/Update Information:	This is a new manual
Operating System:	Linux; Microsoft Windows NT 5.0+
Software Version:	zxweather 0.1

© Copyright David Goodwin, 2012.

Use, reproduction and modification of this document is permitted subject to the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation. See <http://www.gnu.org/copyleft/fdl.html> for full license text.

Contents

1	Introduction	1
1.1	Related Documents	1
2	wh1080	3
2.1	Command-line Arguments	3
2.2	Operations	3
2.2.1	Show Device Configuration	3
2.2.2	Dump to CSV	4
2.2.3	Show History	4
2.2.4	Update Database	6
2.3	Compilation Requirements	6
2.4	Runtime Requirements	7
3	wh1080d and wh1080svc	9
3.1	Command-line Arguments	9
3.2	Compilation Requirements	9
3.3	Runtime Requirements	9
4	wh1080memdump	11
4.1	Example	11
4.2	Compilation Requirements	11
4.3	Runtime Requirements	11
5	wh1080readdump	13
5.1	Example	13
5.2	Compilation Requirements	13
5.3	Runtime Requirements	14

Introduction

zxweather provides a number of utilities for working with weather stations compatible with the Fine Offset WH-1080. All software is available for both Linux and Microsoft Windows systems.

These utilities are:

1. wh1080
2. wh1080d
3. wh1080svc
4. wh1080memdump
5. wh1080readdump

The following chapters provide detailed information on these programs including runtime and compilation requirements, command line parameters, etc.

All software covered by this manual has been tested on the following platforms using the GNU C Compiler:

- Microsoft Windows 7 x64, 32bit builds
- Debian GNU/Linux 6.0.4 (squeeze, 32bit)

1.1 Related Documents

- DAZW-IG01 zxweather Installation Reference, version 0.1
DAZW-DB01 zxweather Database Structure, version 0.1

The wh1080 utility can perform the following operations:

- Show device configuration.
- Dump all history records to a CSV file.
- Show a specific history record or all history records.
- Update the database.

These operations are mutually exclusive - you can only do one at a time.

2.1 Command-line Arguments

Argument	Parameter	Description
-d	database	Database connection string
-u	username	Database username
-p	password	Database password
-l		Force full database load
-c		Show device configuration
-f	filename	Dump all history records to a CSV file
-r	recordid	Display recordid from the database. If the optional parameter is not supplied then all records are shown.

2.2 Operations

2.2.1 Show Device Configuration

To write device configuration information to the terminal, run supplying only the `-c` parameter:

```
wh1080 -c
```

Altering the devices configuration is not currently possible - changes must be made on the device or with other software.

2.2.2 Dump to CSV

To write all history records currently in the device to a CSV file, run `wh1080 -f filename.csv`. This will produce a CSV file with the following columns:

1. Record Number
2. Download Time
3. Time Stamp
4. Last In Set
5. Sample Time (m)
6. Indoor Relative Humidity (%)
7. Indoor Temperature (C)
8. Outdoor Relative Humidity (%)
9. Outdoor Temperature (C)
10. Absolute Pressure (hPa)
11. Average Wind Speed (m/s)
12. Gust Wind Speed
13. Wind Direction
14. Total Rain
15. Invalid Data
16. Rain Overflow
17. Status

These are the same fields stored in the database.

2.2.3 Show History

The `-r` option is used to inspect history records stored in the weather stations memory. If no record ID is specified then all history records are dumped to the terminal.

Valid record IDs are in the range 0 to 4097 (inclusive).

Examples

```
$ wh1080 -r2
Weather station record 2:
  Sample Time: 5 m
  Indoor Relative Humidity: 51%
  Indoor Temperature: 22.8 C
  Outdoor Relative Humidity: 81%
```


Outdoor Temperature: 14.0 C
Absolute Pressure: 1012.6 Hpa m/s
Average Wind Speed: 0.0 m/s
Gust Wind Speed: 0.3 m/s
Wind Direction: 0x0A (SW)
Total Rain: 79.2 mm
Status: 0x00

Note that there must not be any space between the -r option and the Record ID.

```
$ wh1080 -r
Loading all records...
Weather station record 0:
  Sample Time: 5 m
  Indoor Relative Humidity: 51%
  Indoor Temperature: 22.8 C
  Outdoor Relative Humidity: 78%
  Outdoor Temperature: 14.0 C
  Absolute Pressure: 1012.3 hPa m/s
  Average Wind Speed: 0.0 m/s
  Gust Wind Speed: 0.0 m/s
  Wind Direction: 0x06 (SE)
  Total Rain: 79.2 mm
  Status: 0x00
Weather station record 1:
  Sample Time: 5 m
  Indoor Relative Humidity: 51%
  Indoor Temperature: 22.8 C
  Outdoor Relative Humidity: 80%
  Outdoor Temperature: 13.9 C
  Absolute Pressure: 1012.3 hPa m/s
  Average Wind Speed: 0.0 m/s
  Gust Wind Speed: 0.3 m/s
  Wind Direction: 0x08 (S)
  Total Rain: 79.2 mm
  Status: 0x00
Weather station record 2:
  Sample Time: 5 m
  Indoor Relative Humidity: 51%
  Indoor Temperature: 22.8 C
  Outdoor Relative Humidity: 81%
  Outdoor Temperature: 13.8 C
  Absolute Pressure: 1012.4 hPa m/s
  Average Wind Speed: 0.0 m/s
  Gust Wind Speed: 0.3 m/s
  Wind Direction: 0x08 (S)
  Total Rain: 79.2 mm
  Status: 0x00
```

2.2.4 Update Database

This operation inserts new records into a compatible PostgreSQL database. The required database structure is described in *zxweather Database Structure* (document DAZW-DB01).

To update the database run with the database, username and password options. The format for the database option is `database@hostname:port`. For example:

```
$ wh1080 -d weatherdb@localhost:5432 -u postgres -p password
Open Device...
Connect to Database...
Connecting to database weatherdb@localhost:5432
Attempting to come up with a timestamp for current record 14.
This could take a minute...
04/10/12 22:20:13: toffset 3
04/10/12 22:20:15: toffset 3
04/10/12 22:20:17: toffset 3
04/10/12 22:20:19: toffset 4
Weather Station current record is 14 with time stamp 04/10/12 22:16:19
Fetching records 9 to 14...
Computing timestamps...
Inserting records into database...
Commit transaction...
Disconnect...
Finished.
```

Performing Full Updates

If the database is very out of date then performing a regular update as described earlier will only load some history records into the database. To avoid this you can run once supplying the `-l` option to force a full load.

The `-l` option causes every history record stored on the device to be downloaded and inserted into the database. No checks are performed to make sure the data does not already exist in the database. If you run it twice in a row with the `-l` option you will end up with two copies of every history record on the device in the database.

The only time it is safe to use the `-l` option is if the database is behind by more than 4080 records (with a 5 minute interval that would be around 14 days). This means that no record on the weather station would exist in the database.

If the database is not more than 4080 records behind then a normal update should be sufficient to bring the database up-to-date.

2.3 Compilation Requirements

The following is required to compile this program on all supported platforms:

- GNU C Compiler
- ECPG (A part of PostgreSQL)

- libecpg
- libpgtypes

Additionally, on Linux systems libusb-1.0 is required.

2.4 Runtime Requirements

On all systems libecpg and libpgtypes are required. On Linux libusb-1.0 is required. When distributing window binaries hidapi.dll must be supplied.

wh1080d and wh1080svc

wh1080d is a daemon (service in windows) which continuously feeds weather data into the database. The live_data table is updated every 48 seconds and new history records are inserted whenever they appear on the device.

wh1080d and wh1080svc both perform the same job and take the same command-line arguments. There are two of them as they are platform-specific; wh1080d is a daemon for Linux systems and wh1080svc is a service for Windows NT systems.

3.1 Command-line Arguments

Argument	Parameter	Description
-d	database	Database connection string
-u	username	Database username
-p	password	Database password
-f	filename	Log file to write messages to

All parameters are required. The log file is truncated when the daemon starts.

3.2 Compilation Requirements

The following is required to compile this program on all supported platforms:

- GNU C Compiler
- ECPG (A part of PostgreSQL)
- libecpg
- libpgtypes

Additionally, on Linux systems libusb-1.0 is required.

3.3 Runtime Requirements

On all systems libecpg and libpgtypes are required. On Linux libusb-1.0 is required. When distributing window binaries hidapi.dll must be supplied.

wh1080memdump

This utility dumps the weather stations memory to a binary file on disk. It takes only a single parameter: The name of the file to dump to. The program takes a few minutes to run.

Dump files can be converted to text using the wh1080readdump utility.

4.1 Example

```
$ wh1080memdump test.bin
WH1080 Memory Dumper v1.0
(C) Copyright David Goodwin, 2012
```

```
dumping to test.bin...
Dump complete.
```

4.2 Compilation Requirements

On all supported platforms the GNU C Compiler is required to compile the program. Additionally on Linux systems libusb-1.0 is required.

4.3 Runtime Requirements

On Linux systems libusb-1.0 is required. When distributing windows binaries, hidapi.dll must be supplied.

wh1080readdump

This utility reads memory dump files produced with wh1080memdump. It takes only one parameter: The name of the dump file to read. When run with a dump file the device configuration and all history records are written to the terminal.

5.1 Example

```
$ wh1080readdump test.bin
WH1080 Memory Dump Reader v1.0
(C) Copyright David Goodwin, 2012

Loading Device Config...
Taking only 4 bytes from block
Taking only 9 bytes from block
Taking only 30 bytes from block
Loading history data...
Taking only 4 bytes from block
Taking only 9 bytes from block
Taking only 30 bytes from block
Current sampling time interval: 5
Config flags: 0x20 0x02
INDOOR_TEMP_DEG_C OUTDOOR_TEMP_DEG_C RAINFALL_UNIT_MM
PRESSURE_UNIT_HPA WIND_SPEED_UNIT_KMH
Display format flags: 0x30 0x09
PRESSURE_ABSOLUTE WIND_SPEED_AVERAGE TIME_FORMAT_24H
DATE_FORMAT_DDMMYY TIME_SCALE_24H SHOW_DATE_COMPLETE SHOW_OUTDOOR_TEMP
SHOW_RAINFALL_1H
Alarm enable flags: 0x00 0x00 0x00
Reserved A: Disabled
Time: Disabled
Wind Direction: Disabled
Reserved B: Disabled
...
```

5.2 Compilation Requirements

Only the GNU C Compiler is required to compile this program.

5.3 Runtime Requirements

This program has no runtime requirements.

