



WeeWx plugin for WeatherDuino Pro2 Plus and WeatherDuino Logger addon

by engolling

21.01.2019

Get all the data captured by the WeatherDuino Pro2 Plus system into the WeeWx weather software.

In combination with the WeatherDuino logging addon you can import all the processed signals to WeeWx. Therefore, the database schema of WeeWx is extended to hold all of the selected signals.

To use the WeeWx plugin you have to install the WeatherDuino Logger addon first.

How to use and install the WeatherDuino logging addon please refer to the [forum thread](#).

Features:

- Get extra signals in full precision into WeeWx
- Get signals of multiple wind- and rain sensors into WeeWx
- Get signals of the AQM into WeeWx

Be careful, the plugin is in a beta phase and if your conditions are not the same it was tested with, it might result to crash WeeWx.

Installation

First copy the plugin file in the "BIN ROOT" in the folder "user".

To find the "BIN ROOT" please refer to the [WeeWx user guide](#).

Since the plugin reads the "WeeWx_Exp.txt" output file of the WeatherDuino logging script **you have to specify the absolute path to this file**. At the moment it is hardcoded on **three** positions:

First in line 10

```
#First read units from unit line of the export file
filename = '/home/pi/WeatherDuino/WeeWx_Exp.txt'
with open(filename) as f:
    for line,row in enumerate(f.readlines()):
        if line == 0:
            names = row.strip().split(";")
        if line == 1:
            unit_groups = row.strip().split(";")
```

Second in line 81

```
super(WeeWxService, self).__init__(engine, config_dict)
d = config_dict.get('WeatherDuino_logger_service', {})
self.filename = d.get('filename', '/home/pi/WeatherDuino/WeeWx_Exp.txt')
syslog.syslog(syslog.LOG_INFO, "WeatherDuino: using %s" % self.filename)
self.bind(weewx.NEW_ARCHIVE_RECORD, self.read_file)
```

Third in line 112

```
import schemas.wview
filename = '/home/pi/WeatherDuino/WeeWx_Exp.txt'

with open(filename) as f:
```

The first part is responsible to provide the correct units in WeeWx.

Be aware, each unit group that is defined in the transmission layout file in the logging script and appears in the “WeeWx_Exp.txt” file has to be described in the standard unit set of WeeWx or in this part of the configuration!

The second part does the actual data handling and passes it to the WeeWx archive.

The third part is for updating the database scheme. This part is only needed once at the beginning and whenever you change anything in the layout file of the logging script.

Before starting over to make the first modifications in WeeWx please check that the “WeeWx_Exp.txt” file is written properly and contains all your defined data. **Make sure that there are no signal names containing a “ ” space character.** This will make WeeWx crashing while generating the database.

To get the plugin working properly the weewx.conf file has to be modified.

The first step is to extend the database with the new schema, therefore uncomment the old schema and refer to the new one.

[DataBindings]

```
[[wx_binding]]
# The database must match one of the sections in [Databases].
# This is likely to be the only option you would want to change.
database = archive_sqlite
        #database = archive_mysql
# The name of the table within the database
table_name = archive
# The manager handles aggregation of data for historical summaries
manager = weewx.wxmanager.WXDaySummaryManager
# The schema defines the structure of the database.
# It is *only* used when the database is created.
#schema = schemas.wview.schema
schema = user.WeeWx_WeatherDuino_Logger_plugin_04.schema_WeatherDuino
```

Next, add the data collecting service to the engine section:

```
[Engine]

[[Services]]
# This section specifies the services that should be run. They are
# grouped by type, and the order of services within each group
# determines the order in which the services will be run.
prep_services = weewx.engine.StdTimeSynch
data_services = user.WeeWx_WeatherDuino_Logger_plugin_04.WeeWxService,
process_services = weewx.engine.StdConvert, weewx.engine.StdCalibrate, weewx.engine.StdQC,
weewx.wxservices.StdWXCalculate
archive_services = weewx.engine.StdArchive
restful_services = weewx.restx.StdStationRegistry, weewx.restx.StdWunderground,
weewx.restx.StdPWSweather, weewx.restx.StdCWOP, weewx.restx.StdWOW,
weewx.restx.StdAWEKAS
report_services = weewx.engine.StdPrint, weewx.engine.StdReport
```

After that, all modifications of the weewx.conf are done and the new database have to be generated. Run the following command while being in the path of the configuration directory:

```
wee_database weewx.conf --reconfigure
```

This will generate a new extended database named *weewx.sdb_new*. When using SQLite as database system you have to rename the new generated database to replace the initial one.

```
cd SQLITE_ROOT
mv weewx.sdb_new weewx.sdb
```

At the end, run the following command while being in the path of the configuration directory

```
wee_database weewx.conf --rebuild-daily
```

to rebuild the daily summaries.

See also the [WeeWx customization guide](#).

So that's it – as long as the WeatherDuino Logging script and WeeWx is running all extra data will be imported and archived in the WeeWx database. The signals are also available under its names for importing it as tags in the skin files or for generating some graphs.