FOSHKplugin - generic version

Even if the core task of the program is actually the conversion of incoming data from a weather station in WU or Ecowitt format to UDP for any target like Loxone Miniserver, it does offer a few functions that go beyond this.

This plugin connects various weather stations and sensors from the manufacturer Fine Offset Electronics (FOSHK) to a Loxone Miniserver or any other smarthome-system via UDP.
Actually all weather stations are supported where a custom server can be set up as a destination for transmitting the data in WU or Ecowitt format. If you can change or redirect the DNS server for the weather station, it works with any weather station which send it's data to Weather Underground or Ecowitt.
The "custom server" for the GW1000 is configured remotely by FOSHKplugin itself. For other weather stations you have to configure the custom server via WS View or the manufacturer's configuration program.

Version 0.07 - coming soon

- Fixed bug with IGNORE_EMPTY: UDP sending to Loxone did not work if IGNORE_EMPTY was deactivated
- Log output: "custom mode" renamed to "custom server"
- Troubleshooting thunderstorm warning (not every thunderstorm was reported)
- Config parsing made more robust with regard to Boolean values (mkBoolean)
- Forwards can now be activated/deactivated (FWD_ENABLE = True/False) and commented (FWD_CMT) in the config file
- Multi-instance: several instances of FOSHKplugin can now be operated in parallel - in different directories
- Support of the Ambient Weather format for incoming messages as well as forward (AMB/RAWAMB)
  in the absence of yearlyrainin, totalrainin is used and in the absence of rainratein, hourlyrainin is used
- Forward input data in Weathercloud format via GET as type WC possible
- Forward input data in Meteotemplate format via GET as type MT possible
- Preparation WH45 (PM25, PM10, CO2 sensor)
- Thunderstorm warning: number of flashes (lcount) and min. and max. Distance (ldmin and ldmax) are transmitted
- Improvement with regard to TimeOut behavior; http now has a TimeOut of 8 and UDP of 3 seconds
- Ecowitt-Forward: if totalrain is available - but no yearlyrain, yearlyrain is automatically set with the value of totalrain
- New configuration option Export\OUT_TIME = True sets the time stamp of incoming messages from the weather station to the time of receipt
- fake mode now also activated for incoming messages in WU and Ambient Weather format
- important status messages can now also be sent as push notification via pushover (update, sensor, watchdog, battery, storm and thunderstorm warnings)
- generic: display of all detected weather stations during installation via generic-FOSHKplugin-install.sh (-scanWS)

Version 0.06 - 02.08.2020

- Revision of the storm warning function, output of air pressure trend 1h / 3h and value of change of air pressure 1h / 3h
- WU-Forward from AqPM2.5 if PM-sensor is present (only pm25_ch1 is forwarded!)
- AQI calculation activated with EVAL_DATA = True and existing DP200/WH41/43
- Experimental: Forward the PM2.5 value to luftdaten.info as type LD, the ID must be entered under FWD_SID in the config file
- Battery warning via log and UDP implemented; if the supplied batt value falls below an internally defined threshold, a warning is issued
- Fixed issue with output of the name of the sensor again supplying data (SENSOR_MANDATORY)
- The status of warnings for storms, thunderstorms, sensors and batteries are stored temporarily and are therefore retentive
- WU-Forward / JSON renaming from solar radiation to solar radiation
- WU-Forward / JSON support for soil moisture sensors
- WU-Forward: keys with empty value are not transmitted
- WU-Forward: Upload of dewptf (was dewpt) and rainin (was rainratein) fixed
- new formula for dew point calculation (dewpoint) active (requires math)
- now sends a response code 200 to the sender when http data is received
- UDP message for time at wswarning changed from "time:" to "time="
- for all get / post actions: check the return value 200..202 -> ok (was 200 only)
- Text errors in help fixed
- Update of generic-FOSHKplugin-install.sh; Fixed a bug when creating the conf file
- Forward of the input data possible without conversion via UDP as type RAWUDP
- Forward of the input data possible without conversion via EW / POST as type RAWEW
- Forward of output data in CSV-format via http/POST as type CSV
- Forward of input data without conversion via http/POST as type RAWCSV
- Forward of output data as for Loxone to another target/network via UDP - with additional status
- Forward timeout handling adjusted (now 3 seconds)
- Output language can be set via LANGUAGE = DE / EN etc. in the config file
- Plugin can be terminated with a UDP command and requested to send the current status (System.shutdown, Plugin.getStatus)
- debug mode enable/disable via UDP-command Plugin.debug=enable/disable
- Separator selectable on http-GET with /RAW
- new http-GET command /STRING to get the input-line via http with selectable separator
- status messages can now be queried in /JSON and when outputting via /STRING or /UDP
- simple authentication via AUTH_PWD implemented; data and requests are only accepted via http if the specified password is contained in the URL (hint: use the PASSKEY-string in Ecowitt-mode)
- hide PASSKEY-value in Log-Files if AUTH_PWD activated
- Handling of unnecessary quotation marks in the config file adjusted
- Preparation for upcoming soil/water temperature sensor WH34 (tf_chNc, tf_battN - where N = 1..8)
- Status also available via http/GET: http://ipaddress:portnumber/FOSHKplugin/status outputs status wswarning, sensorwarning, batterywarning, ...
- Fake mode implemented: Values of an indoor sensor (WH31 / DP50) can be output as values of an outdoor sensor WH32 (temperature, air humidity)
- Updatewarning implemented, reports an available update for the weather station via log/UDP and possibly via http
- For thunderstorm warnings, tswarning is now output as status instead of tstormwarning - **Attention! This affects all outputs both via UDP and via http!**

Version 0.05 - 26.04.2020

- Sturmwarnung bleibt für 60 Minuten nach letzter Grenzwertunter-/überschreitung aktiv; Zeitraum kann via STORM_EXPIRE im Config-File angepasst werden
- Übermittlung des UV-Wertes im WU-Format angepasst, nun in Großbuchstaben UV= statt uv=
- Patch-Funktion für Weather4Loxone ist nun unabhängig von der genutzten Weather4Loxone-Version (vorhandene fetch.pl wird nicht überschrieben sondern angepasst)

Version 0.04 - 20.02.2020

- default-config angepasst - Kommentare hinter Block nicht zulässig!
- verbesserte Buttons (CSS) - Schiebeschalter nun grau bei "off" und grün bei "on"
- erweiterte CGI-Debug-Funktion; default: off; enable mit $myDebug = 1 in der index.cgi
- myDebug für zusätzliche Debug-Informationen auch im Python-Programm implementiert (default: False)
- Beschreiben der Wetterstation via WS-Set sollte nun (endlich) vollumfänglich funktionieren
- Id & Key in den Einstellungen der Wetterstation werden ignoriert und nicht vom Plugin überschrieben

Version 0.03 - 18.01.2020

- USE_METRIC wieder funktional (jetzt also auch imperiale Werte per UDP und CSV möglich)
- weitere mögliche Probleme beim Setzen der Wetterstationsparameter via WS-Set behoben (Path wird nun immer auf defaults gesetzt)
- Prüfung der nutzbaren LoxBerry-Ports (http/udp) optimiert
- Kommunikation mit der Wetterstation überarbeitet - nun jeweils 5 Versuche bei Lesen und Schreiben
- besseres Logging/Debugging bei Fehlern bei Set-WS; "buntere" und besser parse-bare Log-Files; ### entfernt
- generic: conf-File - Vorlage und Hilfstexte überarbeitet
- Ignorierliste Forward\FWD_IGNORE für Forwards eingebaut: definiert - kommasepariert - Felder, die NICHT verschickt werden sollen
- Forward\FWD_TYPE=UDP/EW/RAW für http-Forward der Werte (UDP-Ausgabezeile) an andere Ziele als WU eingeführt
- nun bis zu 10 Forwards mit unterschiedlichen Einstellungen möglich (aktuell nur im Config-File zu pflegen: Forward-1..9 analog zu Forward)
- Watchdog: kommen seit 3*eingestelltem Intervall keine Werte von der Wetterstation, Fehler melden!
  es erfolgt EINE Warnung und bei erneuter Übermittlung der Wetterstation eine Entwarnung im Log sowie per UDP:
  SID=FOSHKweather wswarning=1 last=346611722
  SID=FOSHKweather wswarning=0 last=346616459
  standardmäßig aktiv; kann im Config-File deaktiviert werden: Warning\WSDOG_WARNING=False
  Intervall kann im Config-File eingestellt werden: Warning\WSDOG_INTERVAL=3
  Warnung auch in Loxone-Vorlage enthalten
- Alarm senden (Log, UDP) wenn Sensor (auch mehrere) keine Daten liefert (etwa weil Akku/Batterie leer)
  SID=FOSHKweather sensorwarning=1 missed=wh65batt time=347196201
  SID=FOSHKweather sensorwarning=1 back=wh65batt time=347196201
  aktuell nur im Config-File zu pflegen:
  Warning\SENSOR_WARNING=True sowie Warning\SENSOR_MANDATORY="wh65batt"
- Sturmwarnung: fällt oder steigt der Luftdruck um mehr als 1.75 Hektopascal in einer Stunde, erfolgt eine Warnung vor Starkwind/Sturm
  vgl. http://www.bohlken.net/luftdruck2.htm
  es erfolgt EINE Warnung und bei Entspannung des Luftdrucks eine Entwarnung im Log und per UDP:
  SID=FOSHKweather stormwarning=1 time=346611722
  SID=FOSHKweather stormwarning=0 time=346616459
  standardmäßig aktiv; kann im Config-File deaktiviert werden: Warning\STORM_WARNING=False
  WarnDiff kann im Config-File eingestellt werden: Warning\STORM_WARNDIFF=1.75
  Warnung auch in Loxone-Vorlage enthalten
- Vorbereitung Wassersensor WH55 und Blitzsensor WH57 (noch unklar ob lightning_time = timestring oder unixtime!)
- preupgrade-Script: Upgrade-Verzeichnisse werden nun auch ohne Elternverzeichnis angelegt (mkdir -p)
- preuninstall-script entfernt; Deinstallation erfolgt bei LoxBerry ab v2.0.1.1 im uninstall-Script
- Web-Oberfläche: Anzeige der Versionsnummer eingebaut (um Nachfragen zur verwendeten Version im Fehlerfall zu minimieren)
- UDP-Versand an das Zielsystem lässt sich mit UDP_ENABLE=False abschalten
- Ignorierliste für den UDP-Versand eingeführt: Config\UDP_IGNORE (nur im Config-File zu pflegen)

Version 0.02 - 28.12.2019

- ### aus FWD-Log-Nachricht entfernt
- Umrechnung temp1f in temp1c für Innensensor auf Kanal 1 implementiert
- Timeout bei sendReboot, setWSconfig und getWSINTERVAL von 1 auf 2 Sekunden erhöht (somit sollte WS-Set sicherer funktionieren)
- Probleme beim Setzen der Wetterstationsparameter via WS-Set behoben (Id & Key werden - wenn nicht schon vorhanden - gesetzt)
- Umstellung der LoxBerry-Versionsnummerierung damit zukünftig die Auto-Update-Funktion greifen kann

Version 0.01 - 15.12.2019

- erste öffentliche Version

Features:

- accepts http messages from a weather station (DP1500, GW1000, HP1000SE, Sainlogic 7 in 1, ELV WS980WiFi, Eurochron EFWS 2900, ???) locally in WU or Ecowitt protocol via network
- does not require cloud services or internet connection
- sends the converted metric or imperial values via UDP to any host or via broadcast in the network
- saves the converted or imperial data sorted and / or extracted as CSV
- enables forwarding to up to 20 servers that are not supported by the weather station itself (e.g. Awekas, PWSWeather, Windy or Luftdaten.info, but you could also use WU in a different interval)
- can feed Meteotemplate and Weathercloud (**from v0.07**)
- can serve as an Ecowitt relay (forward in Ecowitt protocol) for Personal Weather Tablet, weewx, PWS Dashboard and any other program expecting Ecowitt-data

- can forward incoming WU and Ecowitt messages via UDP - also as a broadcast - as they come in
- is able to convert between WU and Ecowitt (within limits)
- can answer queries in WU protocol
- Integrated web server provides the last data record in http, UDP, CSV, RAW and JSON format as well as a simple website
- various watchdogs and warnings can be configured (battery, connection weather station and sensors, storm, thunderstorm, ...)
- calculates some extra data (dew point, AQI, ...)
- provides the Weather4Loxone plugin with the measured values from local weather station
- No additional software is required (WS View only for teaching new sensors or for configuring the standard forwarding services)
- also works without Loxone / LoxBerry as a systemd service on Linux-systems (a Raspi should be powerful enough) for connecting other systems (generic-FOSHKplugin.zip)
- is free of charge

The target system (e.g. Loxone Miniserver) hardly needs resources with this solution; it does not have to fetch any data or convert values - the plugin automatically sends the already converted data to the Miniserver whenever new measured values arrive from the weather station.
In addition, the measured and partly calculated values are also available to any other services via various interfaces and forwards.

Operation:
FOSHKplugin acts as a web server and returns different values depending on the requested URL.
In addition to "updateweatherstation" to accept a incoming data record in WU format (Weather Underground protocol) the integrated web server processes other http call parameters in GET: http://serverip:port/[URLpath]

URLpath	description
/CSVHDR	the field names (the header) of the last data record are output as CSV semicolon separated. If units=e is also specified, the fields for the imperial values are output.
/CSV	all reported metric values of the last data record are output as CSV semicolon separated (units=e supplies the imperial values)
/UDP	the last UDP string is output via http; with additional ?status in URL the output will also include all status.
/RAW	the data set supplied by the weather station is output unchanged via http; separator can be changed with separator=Z, where Z is a single character
/STRING	output the converted data record and the current status separated with ";" via http; by adding units=e in the URL will output with the imp. values; separator can be changed with separator=Z, where Z is a single character. With additional ?status in URL the output will also include all status. example: http://ipadresse:port/STRING?units=e?separator=, will output imp. values with comma as separator
/JSON	output via http as JSON (metric by default; by adding units=e in the URL, the output is made with the imp. values). With additional ? status in URL the output will also include all status (wswarning, sensorwarning, stormwarning, ...).
/	simple website with the current metric data in tabular form
/FOSHKplugin /state	status of the service; if active: "running"
/FOSHKplugin /status	status of the service, watchdog, missing sensor, battery, for storm, thunderstorm, ... as a simple webpage
/FOSHKplugin /debug=enable	enable debug mode for extended messages in the log file
/FOSHKplugin /debug=disable	disable debug mode for extended messages in the log file
/FOSHKplugin /pushover=enable	temporarily enable push notification via Pushover (configuration must exist!) - **from v0.07**
/FOSHKplugin /pushover=disable	temporarily disable push notification via Pushover - **from v0.07**
/FOSHKplugin /patchW4L	"Patch" a Weather4Loxone installation (copy local grabber scripts and activate local retrieval by W4L)
/FOSHKplugin /recoverW4L	Restore the original Weather4Loxone configuration before "patching"
/observations /current/json /units=m	Feedback of a WU-compatible data record with metric values (°C, kmh, mm, hPa)
/observations /current/json /units=e	Feedback of the WU-compatible data record with imperial values (°F, mph, in, inHg)

/w4l/current.dat	Feedback of a W4L-compatible current.dat: 1575925088 | Mon, Dec 09, 2019 21:58:08 +0100 | CET | Europe / Berlin | +0100 | Hohen Neuendorf || DE |||| 6.0 | 6.0 | 92 | Südsüdost | 154 | 2.88 | 5.41 | 6.0 | 1000.2 | 5.1 || 0.00 | 5.1 | 0 | 1.8 | 0.51 |||||||||||||||

If you have defined a PASSKEY in config-file to only accept incoming http-request containing this PASSKEY you always have to add something like **?auth=[PASSKEY]** to the URL. The only exception is the call of FOSHKplugin/state - this works even without authentication. This is useful if FOSHKplugin should not work in a secure local network but directly on the Internet - for example on a root server. Without this security mechanism, anyone could otherwise submit data or query or change states. Basically, however, I advise against operating on "unsafe" hosts that are freely available on the Internet.

Attention!
With HTTP-requests via the Chrome browser (at least on Android systems) problems may arise with the further processing of requests and the submission of data by the weather station.
FOSHKplugin then reports that no data has been received from the weather station for 3 intervals:

02.07.2020 06:48:17.671 **WARNING:** weather station has not reported data for more than 150 seconds (3 send-intervals)

Restarting the plugin will fix the problem again - until another request comes from Chrome on this device.
As a workaround - until a final solution - you should remove the checkmark in the Chrome browser of the relevant device under Settings /Privacy: "?Preload pages for faster browsing".
The Amazon Silk browser (Android 5.1.1) is not affected.

In POST mode, weather station data is accepted in Ecowitt format if the keyword "report" is contained in the URL. Since in Ecowitt format significantly more values can be transmitted from the weather station (such as the battery values of the sensors), I recommend this operating mode (which is also set by the plug-in for WS-Set).

Download:
generic-FOSHKplugin.zip (current stable version v0.06)

Installation:
```
# create a local directory
# sudo mkdir /opt/FOSHKplugin
#
# change into the created directory
# cd /opt/FOSHKplugin
#
# get the current version of the plugin via wget
# wget -N http://foshkplugin.phantasoft.de/files/generic-FOSHKplugin.zip
# or use a local zip-file
#
# unpack ZIP-File
# unzip generic-FOSHKplugin.zip
#
# Allow execute right for generic-FOSHKplugin-install.sh (this script)
# chmod u+x generic-FOSHKplugin-install.sh
#
# Run generic-FOSHKplugin-install.sh (this script)
# sudo ./generic-FOSHKplugin-install.sh --install
```

Configuration options

metric units - USE_METRIC:
If activated, the values for UDP dispatch and CSV export delivered by the weather station in US units are converted directly by the plugin

skip empty values - IGNORE_EMPTY:
When activated, values -9999 coming from the weather station may not be sent to the target machine via UDP

use Loxone time - LOX_TIME:
This switch determines whether the UTC time should be converted to the Loxone time. When activated, an additional field loxtime is added in Loxone-compatible time format (seconds since 01/01/2009).

optional calculations - EVAL_VALUES:
When activated, the values for dew point, wind chill temperature, heat index and perceived temperature and - if a particulate matter sensor DP200 / WH41 / WH43 is available - the current AQI value and its 24-hour average are calculated from the available measured values and those coming from the weather station Data for export processing (UDP, WU, CSV, W4L, ...) added. Values already coming from the weather station may NOT be overwritten. If the storm warning is activated, the air pressure trend and the change in air pressure are also calculated from v0.06 (for the last hour and for the last 3 hours).

optional elements - ADD_ITEMS:
appends a string with static values to the raw data line coming from the weather station; any existing variable names with the same name are overwritten. Useful to pass on a few fields (such as geolocation: lat / lon / elev or location: neighborhood) via UDP / WU / CSV / W4L etc. These fields go through the entire export processing and therefore appear in all output formats - except RAW-exports of course. This function can also be used to exclude values from the weather station from further processing. To do this, an empty value must be assigned to a variable (ie: &variable3=&variable4=value4) and "skip empty values" must be activated.
Format: &variable1=value1&variable2=value2

Log files:
Are very useful for commissioning and in the event of problems. However, you should consider whether it is really sensible to keep a permanent log. If an SD card is used as the storage medium, the SD card will eventually be written down.
Especially the export log - if a very short interval is set - can quickly become very large, because for every message coming from the weather station - depending on the configuration - an entry for UDP, forwarding (FWD) and CSV is also generated.

Forwards - Forward-1..20:
There is only one external destination for sending via "Customized Upload" in the configuration of a weather station. Since we are already using this for FOSHKplugin, you can set a forwarding to an additional service (such as Awekas) here. The plugin currently supports 20 forwarding destinations via the config file.

When specifying the URL, please note that only the measured values are added by the plugin. Any authentications or update commands must therefore already be entered here.
For an upload to Weather Underground (which is of course also possible directly via the weather station), such a line would look like this:

https://rtupdate.wunderground.com/weatherstation/updateweatherstation.php?ID=[meine ID]&PASSWORD=[my password]&action=updateraw&

I successfully tested the delivery to the services Awekas, Windy and PWSWeather:

URL for Awekas:
http://ws.awekas.at/weatherstation/updateweatherstation.php?ID=[awekasid]&PASSWORD=[awekaspassword]&

URL for Windy:
https://stations.windy.com/pws/update/[windyAPIkey]?

URL for PWSWeather:
http://www.pwsweather.com/pwsupdate/pwsupdate.php?ID=[PWS-ID]&PASSWORD=[PWS-Password]&

Other WU-compatible services should also work.
If the field remains free, no forwarding takes place.

"FWD_TYPE" defines the format in which the forwarded messages are to be sent to the weather station.
The WU format should be selected for WU-compatible servers. For other scenarios there is also the UDPGET format, in which the possibly converted metric values are sent as with UDP (but not separated by spaces but by html-conforming "&"). Virtual http inputs should be possible with this.
The EW format is experimental. Incoming messages from the weather station are converted into the Ecowitt format and forwarded in the Ecowitt protocol via HTTP mail. This means that other hosts can also be operated using the Ecowitt protocol (relay).
With type RAW, the incoming data is forwarded via http-get without conversion. In order to send the original RAW string via POST without any extension in the EW format, the RAWEW type is recommended. The RAW data can also be sent via UDP via RAWUDP. Destination-ip: destination-port must then be specified as FWD_URL. If you need to send the output data to another destination via UDP you may use the FWD_TYPE UDP. The destination address and port are defined via destination-ip: port as FWD_URL.
Also from v0.06, the values of an existing particulate matter sensor DP200 / WH41 required for the luftdaten.info service can be sent via type LD:

URL for Luftdaten:
https://api.sensor.community/v1/push-sensor-data/

The sensor ID required for registration must be entered in the config file under FWD_SID. The interval for sending the particulate matter sensor values should be configured to 150 seconds (FWD_INTERVAL = 150 in the config file). In addition to the PM2.5 value, the service also expects the PM10 value (which the DP200 / WH41 particulate matter sensor cannot deliver). Therefore, the plugin sends a dummy value of 1 for PM10.

Overview of the different forward options:

FWD_TYPE	input-Format	out-Transport	out-Format
WU	WU, EW	GET	Weather Underground (WU-->WU or EW-->WU)
RAW	WU, EW	GET	like input (WU-->WU or EW-->EW)
UDPGET	WU, EW	GET	like output to Loxone with header and possibly conversion, however, URL-compatible with "&" instead of spaces
WC	WU, EW	GET	Weathercloud **(from v0.07)**
MT	WU, EW	GET	Meteotemplate (API) - **from v0.07**
EW	WU, EW	POST	enhanced Ecowitt (WU-->EW or EW-->EW)
RAWEW	WU, EW	POST	untouched Ecowitt (EW-->EW or WU-->EW)
LD	WU, EW	POST	Luftdaten.info-Format (only PM2.5, PM10, Temp, Humidity, rel. Pressure, abs. Presssure)
CSV	WU, EW	POST	like output to Loxone with possibly conversion, with semicolon as separator instead of spaces but without header
RAWCSV	WU, EW	POST	like input (WU-->WU or EW-->EW), with semicolon as separator instead of spaces but without header
UDP	WU, EW	UDP	like output to Loxone with header and possibly conversion via UDP (FWD_URL = destination:port)
RAWUDP	WU, EW	UDP	like input-Format but transmission via UDP (EWEW via UDP or WUWU via UDP)

Data fields from the ignore list maintained under "FWD_IGNORE" are not sent for the relevant forward.

With "FWD_INTERVAL" an interval independent of the weather station can be configured (in seconds). If this field is left blank, it will be sent at the weather station's send interval.

Save as CSV:
The measurement results can also be saved as a comma-separated file (CSV). The storage location and the file name are specified under CSV_NAME. The problem with writing to SD cards already mentioned for log files also applies here. If necessary, a more suitable medium (such as NFS) should be selected here.

Field names for CSV - CSV_FIELDS:
All fields desired in the CSV are listed under CSV_FIELDS - separated by a separator (semicolon, comma or space).
Not all fields of a data record are worth saving in the CSV. The contents of the fields SID, PASSKEY, freq or model change very rarely.
By omitting these field names, these fields are excluded from storage. The order of the columns in the CSV file results from the order of the fields specified here.

CSV interval- CSV_INTERVAL:
Here you can define your own time interval for storing a data record in the CSV. If the field remains free, the transmission interval of the weather station is used.

The interval for the CSV and forwarding function cannot be smaller than the set transmission interval of the weather station, since data is only available for further processing when a data record is received from the weather station.

watchdog & warn-functions:

type	config	description
report watchdog	WSDOG_WARNING	will warn if weather station did not report within 3 send-intervals (configurable)
sensor warning	SENSOR_WARNING	will warn if data for mandatory sensor (configurable list of fields e.g. wh65batt) is missed
battery warning	BATTERY_WARNING	will warn if battery level of all known sensors is critical (pre-defined)
storm warning	STORM_WARNING	will warn if air pressure rises/drops more than 1.75 hPa/hour or 3.75hPa/3hr with expiry time of 60 minutes (all values configurable)
thunderstorm warning	TSTORM_WARNING	will warn if lightning sensor WH57/DP60 present, count of lightnings is more than 1 and distance is less or equal 20km with expiry time of 15 minutes (all values configurable)
firmware-update warning	UPD_CHECK	will warn if there's a new firmware for the weather station available

These warnings are sent via UDP and recorded in the standard log file.

Fake-Mode
The values for outside temperature and humidity normally come from either a combination sensor (WH65, WS80) or the dedicated outside sensor WH32. However, if neither a combination sensor nor a WH32 is available, values of any internal sensor DP50/WH31 (which should of course then be installed outside with appropriate weather protection) can be output as values ?of the external sensor.
In the config file you have to specify which key should be used for the respective value:

```
[Export]
OUT_TEMP=temp1f     # exchange the keyname temp1f with tempf (or use temp2f, temp3f, ...)
OUT_HUM=humidity1   # exchange the keyname humidity1 with humidity (or use humidity2, humidity3, ...)
```

Within FOSHKplugin, the substring "&temp1f =" is simply replaced by "&tempf =" and "&humidity1 =" by "&humidity =" when the Ecowitt line arrives from the weather station. The values themselves remain.
This setting is global and therefore affects all FOSHKplugin exports/forwards/outputs (except for RAW and RAWEW).
The weather station itself, of course, knows nothing of this - so the services configured there (Ecowitt, WU, WOW, etc.) still have no external values.
If you also want to output the values of the indoor sensor as outdoor sensor values for these services, the services within the weather station must be deactivated and carried out by FOSHKplugin instead.
Corresponding forwards must then be defined in the config file (the square brackets must **not** be included):

```
WU:
[Forward-11]
FWD_INTERVAL = 300
FWD_URL = https://rtupdate.wunderground.com/weatherstation/updateweatherstation.php?ID=[WU-ID]&PASSWORD=[WU-Password]
&action=updateraw&
FWD_TYPE = WU
```

```
EW:
[Forward-12]
FWD_INTERVAL = 300
FWD_URL = http://cdnrtpdate.ecowitt.net/data/report/
FWD_TYPE = EW
```

WOW:
[Forward-13]
FWD_INTERVAL = 300
FWD_URL = http://wow.metoffice.gov.uk/automaticreading?siteid=[siteid]&siteAuthenticationKey=[siteAuthenticationKey]&
FWD_TYPE = WU

Weathercloud (from v0.07):
[Forward-14]
FWD_INTERVAL = 300
FWD_URL = http://api.weathercloud.net/v01/set?wid=[weathercloudid]&key=[key]&
FWD_TYPE = WC

Meteotemplate (from v0.07):
[Forward-15]
FWD_INTERVAL = 300
FWD_URL = http://192.168.15.100/template/api.php?PASS=[meteotemplatepwd]&
FWD_TYPE = MT

Configuration-file:

```
[Config]
LOX_IP =                    # destination-IP or broadcast-address to send data to
LOX_PORT = 12340          # destination-Port - UDP-port to send data to
LB_IP =                     # local IP-address
LBU_PORT = 12341          # local Port - UDP-port to receive UDP-datagrams
LBH_PORT =                  # local-Port - HTTP-port to receive HTML-in & out
LOX_TIME = False            # adjust time base to 01.01.2009 (Loxone only)
USE_METRIC = True           # use metric instead of imperial values
IGNORE_EMPTY = True    # do not send -9999 or empty values
UDP_ENABLE = True         # set to False to disable UDP-sending
UDP_IGNORE =                # comma-separated list of fields to not send via UDP
LANGUAGE = DE               # remove or adjust to EN to use english output for wprogtxt and wnowtxt
AUTH_PWD =                  # only accept http-GET & POST if specified password is given in the URL (default: none)

[Weatherstation]
WS_IP =                     # IP-address of weather station
WS_PORT =                   # UDP-port of weather station
WS_INTERVAL = 60          # weather station will send data every n seconds (16..3600)

[Export]
EVAL_VALUES = True        # calculate some extra values on base of current data (dew point, windchill, heatIndex, feelsliketemp, AQI, ...)
ADD_ITEMS =                 # additional fixed strings to append to every raw-data-line
OUT_TEMP =                  # fake the temperature-value for outdoor-sensor with value of specific indoor-sensor e.g. temp1f
OUT_HUM =                   # fake the humidity-value for outdoor-sensor with value of specific indoor-sensor e.g. humidity1

[Forward]
FWD_URL =                   # URL of destination
FWD_INTERVAL =            # interval in seconds in which lines will be forwarded
FWD_IGNORE =              # comma-separated list of fields to not forward
FWD_TYPE =                # WU/UDP/LD/RAW/EW/RAWEW/RAWUDP - WU: WU-format; UDP: UDP-String will be forwarded (default); LD: PM2.5
luftdaten.info; EW: Ecowitt; RAWEW: Ecowitt untouched; RAW: as input; RAWUDP: RAW via UDP
FWD_SID =                   # SensorID for luftdaten.info

# you additionally can use Forward-1..9
[Forward-1]
FWD_URL =                   # URL of destination
FWD_INTERVAL =            # interval in seconds in which lines will be forwarded
FWD_IGNORE =              # comma-separated list of fields to not forward
FWD_TYPE =                # WU/UDP/LD/RAW/EW/RAWEW/RAWUDP - WU: WU-format; UDP: UDP-String will be forwarded (default); LD: PM2.5
luftdaten.info; EW: Ecowitt; RAWEW: Ecowitt untouched; RAW: as input; RAWUDP: RAW via UDP
FWD_SID =                   # SensorID for luftdaten.info

[CSV]
CSV_NAME =                          # file name for csv-file (always metric!)
CSV_FIELDS =                        # fields in desired order, separated with ; or ,
CSV_INTERVAL =                      # interval in seconds in which lines are written to the csv-file
```

```
[Warning]
WSDOG_WARNING = True                # warn if weather station did not report data within n send-intervals
WSDOG_INTERVAL = 3                  # checking interval for WSDOG_WARNING
SENSOR_WARNING = False              # warn on missing sensor data
SENSOR_MANDATORY =  wh65batt     # a comma-separated list of all mandatory sensors
BATTERY_WARNING = True              # warn if battery level of known sensors is critical
STORM_WARNING = True                # activate storm warning based on a change in air pressure
STORM_WARNDIFF = 1.75               # change of air pressure in hPa within one hour to get warning state
STORM_WARNDIFF3H = 3.75             # change of air pressure in hPa within three hours to get warning state
STORM_EXPIRE = 60                   # minutes, warning will stay active since last over- and under-range indication
TSTORM_WARNING = True               # enable thunderstorm warning (needs WH57/DP60 lightning sensor)
TSTORM_WARNCOUNT= 1                 # warn if more than n lightnings were detected
TSTORM_WARNDIST = 30                # warn only if lightning is closer than n km
TSTORM_EXPIRE = 15                  # minutes, warning will stay active since last lightning

[Logging]
logfile = REPLACEFOSHKPLUGINLOGDIR/log-foshkplugin.log                # default log - all start/stop/warn/error messages
rawfile = REPLACEFOSHKPLUGINLOGDIR/raw-foshkplugin.log                # logs raw messages coming from weather station only
sndfile = REPLACEFOSHKPLUGINLOGDIR/snd-foshkplugin.log                # logs outgoing messages from plugin (CSV, UDP, FWD)

[Update]
UPD_CHECK = True                    # enable/disable firmware-update-check for weatherstation (default: True)
UPD_INTERVAL = 86400                # interval in seconds of checking for firmware-updates (default: 86400)
UPD_URL =                           # URL for update-info (default=internal)
```

Attention!
After changes in the [Weatherstation] area, the new settings must be transferred to the weather station!
By calling ./foshkplugin.py -writeWSconfig, the weather station-specific values from config file are transmitted to the weather station and activated.

datapoints:
The names of the outgoing data points depend on the selected output system (metric or imperial). If USE_METRIC is active, the following data points are output to all configured exports (but not for format-specific forwards):

for Gateway DP1500/GW1000:
humidityin
tempinc

soil moisture sensors DP100/WH51:
soilbatt1..8 (battery in volt)
soilmoisture1..8

for Multi-Temp/Hum-sensors DP50/WH31:
batt1..8 (battery-Status; 1 = Alarm, 0 = ok)
humidity1..8
temp1..8c

for PM-sensors DP200/WH41/WH43:
pm25_avg_24h_ch1..4
pm25_ch1..4
pm25batt1..4 (Battery-Status; 5 = max)
pm25_AQI_ch1..4
pm25_AQI_avg_24h_ch1..4
pm25_AQIlvl_ch1..4 (Level 1..6)
pm25_AQIlvl_avg_24h_ch1..4 (Level 1..6)

Status/Activity/Tracker:
running (1 = startet, 0 = stopped)
loxtime (Loxone-time)
wswarning (weather station does not send data)
sensorwarning (mandatory sensor is missing)
batterywarning (Battery-warning)
stormwarning (Stormwarning)
tswarning (Thunderstorm-warning)
updatewarning (Firmware-update available)

for lightning sensor WH57/DP60:
lightning (distance last lightning in km)
lightning_time (time of last lightning Unixtime)
lightning_loxtime (time of last lightning Loxone-time)
lightning_num (count of lightnings)
wh57batt (Battery-Status; 5 = max)

for soil/water-temp-sensor WH34:
tf_chNc (temperature in °C; N=1..8)
tf_battN (battery; N=1..8)

for water sensor WH55:
leak_ch1..4 (1=Alarm, 0=ok)
leakbatt1..4 (Battery-Status; 5 = max)

for 2- or 3-wing outdoor sensor WH3000SE All-In-One or HP1000SE All-In-One (WH65):
baromabshpa (abs. air pressure in hPa)
baromhpa
baromrelhpa
dailyrainmm
dewptc
eventrainmm
feelslikec
heatindexc
hourlyrainmm
monthlyrainmm
humidity
loxtime
maxdailygust
rainratemm
solarradiation
tempc
totalrainmm
uv
weeklyrainmm
wh65batt (1 for warning, 0=ok)
windchillc
winddir
windgustkmh
windspeedkmh
yearlyrainmm
ptrend1 (air pressure-trend 1h: 1=rising, 0=equal,-1=falling)
pchange1 (air pressure change in 1h in hPa)
ptrend3 (air pressure-trend 3h: 1=rising, 0=equal, -1=falling)
pchange3 (air pressure change in 3h in hPa)
wproglvl (weather prognose level)
wprogtxt (weather prognose text)
wnowlvl (current weather level)
wnowtxt (current weather text)

As output via UDP you will get this:

```
SID=FOSHKweather dateutc=2020-06-01+13:35:54 loxtime=360257754 tempinc=27.0 humidityin=30 baromrelhpa=1022.59 baromabshpa=1017.51
tempc=25.1 humidity=30 winddir=273 windspeedkmh=1.43 windgustkmh=7.19 maxdailygust=18.36 solarradiation=677.62 uv=5 rainratemm=0.0
eventrainmm=0.0 hourlyrainmm=0.0 dailyrainmm=0.0 weeklyrainmm=0.0 monthlyrainmm=0.0 yearlyrainmm=206.4 totalrainmm=206.4 temp2c=23.3
humidity2=40 temp3c=24.8 humidity3=34 soilmoisture1=34 soilmoisture2=37 soilmoisture3=41 soilmoisture4=52 pm25_ch1=11.0
pm25_avg_24h_ch1=9.8 lightning_num=0 leak_ch1=0 wh65batt=0 batt2=0 batt3=0 soilbatt1=1.6 soilbatt2=1.6 soilbatt3=1.9 soilbatt4=1.9 pm25batt1=5
wh57batt=5 leakbatt1=5 dewptc=6.3 windchillc=25.1 feelslikec=25.1 heatindexc=24.4 pm25_AQI_ch1=46 pm25_AQIlvl_ch1=1
pm25_AQI_avg_24h_ch1=41 pm25_AQIlvl_avg_24h_ch1=1 ptrend1=-1 pchange1=-0.3 wnowlvl=3 wnowtxt=sonnig ptrend3=-1 pchange3=-1.42
wproglvl=3 wprogtxt="baldiger Regen"
```

where you can easily pickup the required fields for further processing.
Activity and Tracker messages are event-based but also include the SID-token to state these messages are coming from FOSHKplugin:

SID=FOSHKweather stormwarning=1 time=351042104
SID=FOSHKweather stormwarning=0 time=351042135
SID=FOSHKweather tswarning=1 time=359550337
SID=FOSHKweather tswarning=0 time=359551236 start=359550337 end=359551236 last=359550330
SID=FOSHKweather wswarning=1 last=360086463 time=360086560
SID=FOSHKweather wswarning=0 last=360086567 time=360086590
SID=FOSHKweather sensorwarning=1 missed=pm25batt1 time=360169470
SID=FOSHKweather sensorwarning=0 back=pm25batt1 time=360170059

If LOX_TIME is deactivated, the Unixtime appears for all the given times instead of Loxone-time.

There are a some datapoints through which the plugin can be controlled via UDP:

System.reboot	**restart the GW1000/DP1500**
Plugin.shutdown	**shutdown FOSHKplugin - if started as a system service (systemd) it will be restarted some seconds later**
Plugin.getstatus	**requests the current status values from the plugin (running, wswarning, sensorwarning, ...)**
Plugin.debug=enable	**enable debug mode for some more information in log file**
Plugin.debug=disable	**disable debug mode**

By sending a string "SID=FOSHKplugin,System.reboot" the plugin causes the GW1000 to restart. If you send the string "SID=FOSHKplugin,Plugin.getstatus", the plugin replies with the current status values for wswarning, sensorwarning, stormwarning, ...

legal notice:
I do not assume any guarantees regarding the use of this software - use is at your own risk.
Never make decisions that can lead to personal injury or property damage on the basis of this software.
Warnings generated by the program (e.g. storm or thunderstorm) can occur. However, the absence of these warnings does not imply that these things are not possible.

There is a more extensive documentation, although more Loxone-specific and currently in german only: https://foshkplugin.phantasoft.de/
However, the Google translator should be helpful: https://translate.google.de/translate?hl=de&tab=rT&sl=de&tl=en&u=https%3A%2F%2Ffoshkplugin.phantasoft.de%2F

Recipes:

change sending interval to a shorter interval than 16 seconds

```
# open a console on your Linux-machine
# make sure you have Python & the libraries we need
sudo apt-get -y install --no-upgrade python3 python3-pip
pip3 install requests

# create a new dir somewhere
sudo mkdir /opt/FOSHKplugin

# change into this new dir
cd /opt/FOSHKplugin

# get current Beta of FOSHKplugin
wget -N http://foshkplugin.phantasoft.de/files/generic-FOSHKplugin-0.0.6Beta.zip

# unzip the file
unzip generic-FOSHKplugin-0.0.6Beta.zip

# if you already know the ip address of the device you want to change you can skip next 3 steps
# the WS_PORT should always be 45000 on FOSHK-devices

# get ip address of weather station --> you will need this later as WS_IP
./foshkplugin.py -getwsip

# get port of weather station --> you will need this later as WS_PORT
./foshkplugin.py -getwsport

# get current sending interval - not needed; just for info or check
./foshkplugin.py -getwsinterval

# now set the desired sending interval to 5 seconds - where WS_IP is the ip address of your GW1000 and WS_PORT the command port (probably 45000)
# example: the IP address of your GW1000 is 192.168.0.1, interval should be 5: ./foshkplugin.py -setWSinterval 192.168.0.1 45000 5
./foshkplugin.py -setWSinterval WS_IP WS_PORT 5
```

redistribute the Ecowitt stream of a GW1000 to several weewx

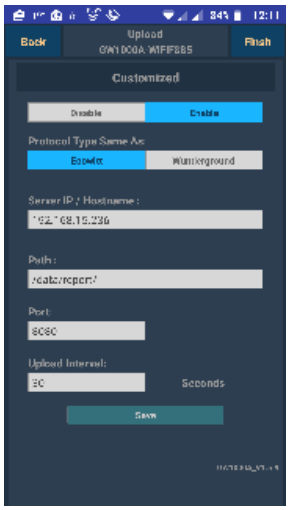
The GW1000 knows exactly **ONE** destination for a custom server upload. But if you want to feed several instances of weewx with the data from the GW1000, this also requires several GW1000.
Or you can simply use FOSHKplugin with just one GW1000 and let FOSHKpugin redistribute the incoming stream as you like!

FOSHKplugin knows 10 forwarding destinations - called Forwards.
A Forward-block must be created in the config file for each forward, in which the target, type and other conditions can be configured:

[Forward-n]
FWD_URL = # URL of destination
FWD_INTERVAL = # interval in seconds in which lines will be forwarded
FWD_IGNORE = "" # comma-separated list of fields to not forward
FWD_TYPE = "" # WU/UDP/LD/RAW/EW/RAWEW/RAWUDP - WU: WU-format; UDP: UDP-String will be forwarded (default); LD: PM2.5 luftdaten.info; EW: Ecowitt; RAWEW: Ecowitt untouched; RAW: as input; RAWUDP: RAW via UDP
FWD_SID = "" # SensorID for luftdaten.info

For the forwarding of the original Ecowitt stream, only the destination (FWD_URL) and the type (FWD_TYPE) have to be defined. The other parameters are optional.

If you want to operate 3 weewx instances with a GW1000, you have to enter the IP address of the FOSHKplugin host under **Server IP / Hostname** and the port number configured there (LBH_PORT) under **Port** via the WS View app.
/data/report/ should be configured as **Path**.



On FOSHKplugin-side you have to create 3 blocks that refer to the respective weewx instances:

[Forward-1]
FWD_URL = http://weewx-host1:weewx-port1 # URL of destination
FWD_TYPE = RAWEW # RAWEW: Ecowitt untouched

[Forward-2]
FWD_URL = http://weewx-host1:weewx-port2 # URL of destination
FWD_TYPE = RAWEW # RAWEW: Ecowitt untouched

[Forward-3]
FWD_URL = http://weewx-host1:weewx-port3 # URL of destination
FWD_TYPE = RAWEW # RAWEW: Ecowitt untouched

Don't forget to restart the systemd service after making changes to the config file: service foshkplugin restart.
As a result, when an Ecowitt string is received by the GW1000, it is forwarded in parallel and unprocessed to these defined forwards.

Example:

current situation:

GW1000-1 with ip address 192.168.1.1 configured in WS View:	Server IP / Hostname: 192.168.1.100	Port: 8001	Path: (none)
GW1000-2 with ip address 192.168.1.2 configured in WS View:	Server IP / Hostname: 192.168.1.100	Port: 8002	Path: (none)
GW1000-3 with ip address 192.168.1.3 configured in WS View:	Server IP / Hostname: 192.168.1.100	Port: 8003	Path: (none)

You install FOSHKplugin on 192.168.1.100 with LBH_PORT 8080 and create these blocks in your foshkplugin.conf:

[Forward-1] FWD_URL = http://192.168.1.100:8001 FWD_TYPE = RAWEW	 # URL of destination # RAWEW: Ecowitt untouched
[Forward-2] FWD_URL = http://192.168.1.100:8002 FWD_TYPE = RAWEW	 # URL of destination # RAWEW: Ecowitt untouched
[Forward-3] FWD_URL = http://192.168.1.100:8003 FWD_TYPE = RAWEW	 # URL of destination # RAWEW: Ecowitt untouched

restart the FOSHKplugin-service with sudo service foshkplugin restart

final situation:
Afterwards you can remove GW1000-2 and GW1000-3 and reconfigure the remaining GW1000-1 with WS View to:

Server IP / Hostname: 192.168.1.100
Port: 8080
Path: /data/report/

Now the GW1000-1 sends the Ecowitt-Stream to FOSHKplugin which redistribute this to all 3 weewx-instances. Just with **ONE** GW1000.

In addition to notification via UDP, availability as status via http and logging in the log file, important status changes (e.g. firmware update, sensor, watchdog, battery, storm and thunderstorm warnings) can also be sent to any mobile device (iOS , Android) as push notification, FOSHKplugin uses the API of Pushover.
Pushover is a one-time purchase app (no subscription!) that listens in the background for incoming messages from the pushover server.
If configured accordingly, FOSHKplugin sends these critical warnings via API call via the Internet to the Pushover cloud service, which then immediately searches for contact with the devices stored there and delivers the message immediately.
In the mobile device itself, these push notifications then arrive - depending on the setting - with or without sound and/or vibration or silently and are displayed both on the lock screen and in the notification bar. If available and configured accordingly, the notification LED also lights up or flashes.
With adjustable time schedules, you can specify the time periods during which the messages are sent silently within Pushover.
Besides the one-off purchase price for the app, there are no other hidden costs. At least if you stay below the free 7500 (!) notifications per month.
I've experimented with it for a few days now and I'm thrilled! The delivery takes place reliably and promptly - within a few (here 1-2) seconds (provided that the Internet is available).

Manual:

1. get the app Pushover from the respective store (Android, iOS)
2. Start the Pushover app and assign credentials
3. Log in via web browser: https://pushover.net/login
4. Make a note of the key under "Your User Key", this must be specified for PO_USER in the FOSHKplugin config file foshkplugin.conf
5. Generate an API token for FOSHKplugin under "Your Applications" (Create an Application / API Token) - the key specified under API Token /Key must be specified under PO_TOKEN in the FOSHKplugin config file - as app notification icon you could use this
6. Adjust config foshkplugin.conf: Pushover\PO_ENABLE=True Pushover\PO_USER="Your User Key" and Pushover\PO_TOKEN=API-TOKEN:
   [Pushover]
   PO_ENABLE = True
   PO_USER = userkey
   PO_TOKEN = token
7. Restart FOSHKplugin

From now on there should be a push notification for all important status changes.
The sending of push messages is activated with the switch PO_ENABLE = True in the config file. Sending is deactivated by default (False).
Push notifications from FOSHKplugin can also be activated and deactivated during runtime, provided the correct credentials are stored in the config file.
This is done via http via any web browser by calling up the page http://ipaddress:port/FOSHKplugin/pushover=enable (activate) or http://ipaddress:port /FOSHKplugin/pushover=disable (deactivate). The port is the port specified in the config file under LBH_PORT.
This is also possible via the UDP interface of FOSHKplugin: sending "Plugin.pushover=enable" to the IP address of the host and the port on which FOSHKplugin is running (LBU_PORT in the config file) activates the sending; "Plugin.pushover=disable" deactivates this.
Any errors when sending push notifications as well as activating/deactivating during runtime are logged in the standard log file.

In some constellations it can make sense to operate several instances of FOSHKplugin in parallel - for example, to process data from several GW1000 /DP1500/HP2551C.
Basically this is possible, but it does require a few points to be observed during the installation and, if necessary, a few adjustments to the configuration file foshkplugin.conf.
In any case, each instance must be installed in its own directory!

Installation:
The http port (LBH_PORT) and the port for incoming UDP messages (LBU_PORT) must not be used more than once on a host. Therefore, a different http port and a different UDP port must be specified for each instance.
The installation routine generic-FOSHKplugin-install.sh automatically ensures that ports are not assigned twice.

By default, a service with the name foshkplugin is created, which can be started and stopped and which starts again automatically in the event of an unscheduled termination.
When running the installation script generic-FOSHKplugin-install.sh, however, a different name can be defined for the service.
Important: a different name must be selected for the service for each instance on the same host in order to be able to operate these different services in parallel!
I recommend naming all FOSHKplugin services with "foshkplugin" - followed by a serial number or a more specific description: for example foshkplugin-GW1 or foshkplugin-Location1.

Adjustments:
If FOSHKplugin is also used to send UDP messages, it may be necessary to change the identifier for these messages so that the target system can assign the incoming messages.
Every outgoing UDP message contains a "SID = FOSHKweather" as an indicator of the data source. If you want to change this identifier, you can specify a different identifier in the config file under Config\DEF_SID.
When the UDP messages are parsed on the processing side, this can then be used as a trigger for further processing.
Alternatively, a distinction can also be made on the basis of the UDP port number (LOX_PORT).

This adjustment is not required for pure forward operation without UDP transmission.

Ambient Weather has a very modern web interface, a nice app, a connection to IFTTT, Amazon Alexa and Google Assistant and can be queried via API interface.
Access to this service requires that you also deliver your weather data there. This is possible with weather stations from Ambient Weather as well as with devices from third-party manufacturers.
A special license is required for operation with devices from third-party manufacturers: VW-ANET. This license is purchased once, is bound to a MAC address and can be used with FOSHKplugin to send data from an Ecowitt station to Ambient Weather.

To connect a GW1000 from Ecowitt (or Froggit DP1500) to Ambient Weather, the following steps are necessary:

1. Registration with Ambient Weather - create an account at https://ambientweather.net/signin
2. Purchase of the VW ANET license stating the MAC address of the GW1000 / DP1500 via https://www.ambientweather.com/amwevwamweac.html
3. Configuration of FOSHKplugin:
   A new forward must be created in the config file:

   ```
   [Forward-21]
   FWD_TYPE = RAWAMB
   FWD_CMT = Forward for Ambient Weather
   FWD_URL = https://api.ambientweather.net/endpoint?
   FWD_ENABLE = True
   FWD_STATUS = False
   FWD_INTERVAL = 180
   ```

   The device is authenticated via the PASSKEY sent through the GW1000/DP1500 automatically, but can be adjusted in the config file with FWD_SID = [PASSKEY] if necessary.

From now on, FOSHKplugin also sends the incoming data from the GW1000/DP1500 to Ambient Weather.
No additional hardware or software is required.

2bc ...