

How to Monitor Zehnder ComfoAir Q Ventilation Units with PRTG Network Monitor



In our house we have two ventilation units from Swiss/German vendor Zehnder, one [ComfoAir Q600](#) and one [ComfoAir Q350](#). Since most of our energy and house

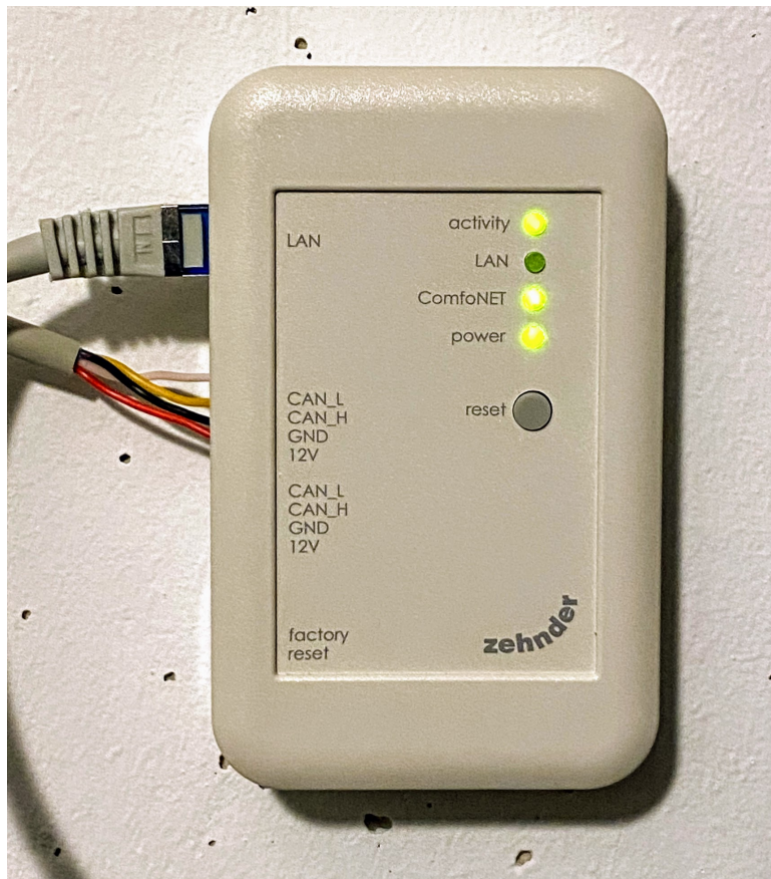
tech systems are already monitored by my [PRTG](#) instance (including [solar system](#) and [heatpump](#)) to be shown on [my dashboard](#), I also wanted to monitor our ventilation system, too.

Which turned out to be harder than expected...



Connecting the fan units to the network

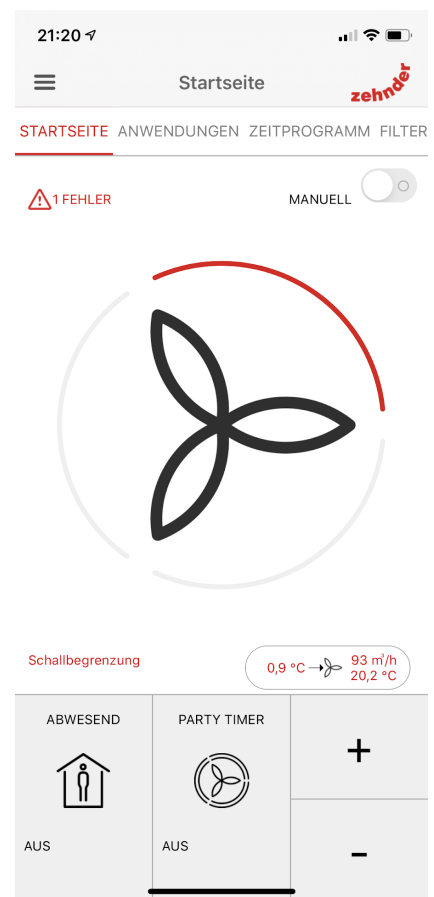
To connect the ventilation units to my Ethernet network I purchased and connected two [ComfoConnect LAN C](#) modules which are intended for use with the Zehnder's official [ComfoControl](#) Android/iOS apps.



Now I could control the systems from my smart phone.

OK, that was easy.

But now it became much more complicated because of the fact, that the API calls to monitor/control the ComfoConnect Lan C Module are closed source. Zehnder wants the end user to use their official ComfoControl Android/iOS apps. Nevertheless, some geeks had (thankfully) way too much time and implemented a very nice reverse engineering library to monitor your ComfoConnect fans in Python and its available on GitHub: [PyComfoConnect](https://github.com/PyComfoConnect).



Zehnder ComfoControl App (iOS)

PyComfoConnect Open Source Library

PyComfoConnect is an open source library developed by Michaël Arnauts, an IOT enthusiast from Belgium, that makes it possible, to monitor the Q350/Q450/Q600 systems connected to the ComfoConnect Lan C Module.

It's not perfect as it is limited in its functionalities compared to the official ComfoControl apps (there are not that many commands to control the fans yet), but it is enough to monitor your ventilation units. Thanks Michaël!

Developing the code for PRTG

It became clear that I need some help with this because my Python knowledge is limited. So I asked Ferhat Aram for help to make the PyComfoConnect library usable in PRTG. As [PRTG has native support for executing Python scripts](#) to retrieve data from sensors, there was a good chance that this could work.

The goal therefore was to create a Python script, that a) outputs each sensor data as channel in PRTG and b) is reusable.

And well, what can I say: With some tweaks, trial and error (and skills) Ferhat got this to work! He built a wrapper script, that passes the IP and PIN of the Zehnder unit from PRTG to the library, then listens for a brief amount of time to the push messages of the module and afterwards translates the data into PRTG's XML format.

After Ferhat got the main thing to work I was able to adjust some details myself (like sensible channel limits) even with my limited Python experience! Thanks Ferhat!

Installation

So finally we can now share the code and explain how to use it in your PRTG installation:

- Note: Since I am using the [cloud version of PRTG](#) I am using several remote probes on Windows PCs to monitor our home. So I had to do the following steps on the remote probe system, not the PRTG core server system itself.
- Prepare the Python environment:

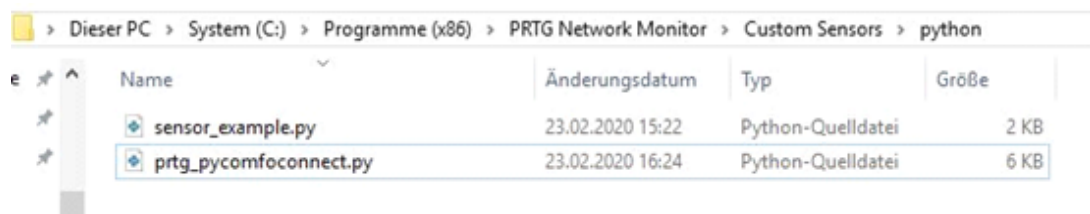
- Open cmd command line and navigate to the Python directory of PRTG, usually:

```
>cd C:\Program Files (x86)\PRTG Network Monitor\python
```

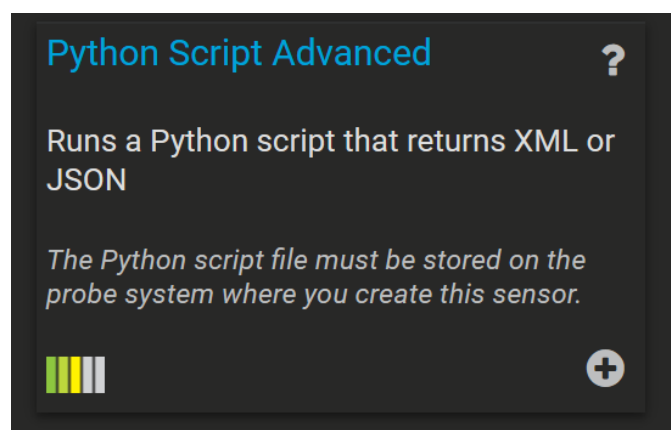
- In the console type:

```
python.exe -m pip install pycomfoconnect
```

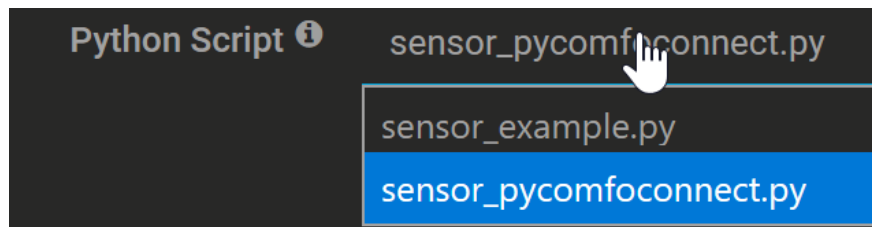
- Install and set up the Zehnder LAN C hardware module:
 - Make sure that your ComfoConnect Lan C module is in the same network as your PRTG/Remote Probe that will execute the Python script
- Download our [sensor_pycomfoconnect.py](#) script from [my GitHub repository](#)
- Copy the script into the `<PRTG/Remote Probe Installation Folder>\Custom Sensors\python\` directory



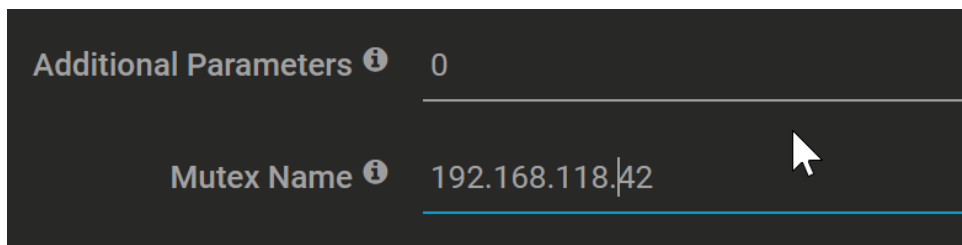
- Log into your PRTG's web interface, create a device for your ventilation system with the IP address used by the ComfoConnect Lan C Module
- Then create a new sensor for this device
- In the sensor creation process, where you are prompt to select a sensor type, search for the “python” and select “Python Script Advanced”:



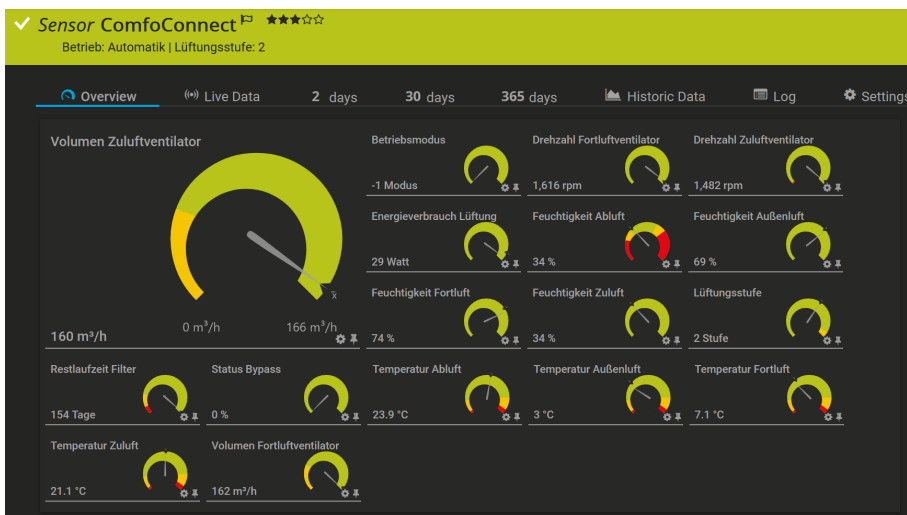
- Set up the sensor by selecting the `sensor_pycomfoconnect.py` script.



- There are two settings that you must provide data for:
 - In the “Additional Parameters” field you need to provide the PIN of the ventilation system (default is “0”).
 - And please use a unique string for the “Mutex Name” field (I would recommend using the IP address) to make sure that only one sensor at any time is accessing each LAN module, because it does not support multiple concurrent connections.



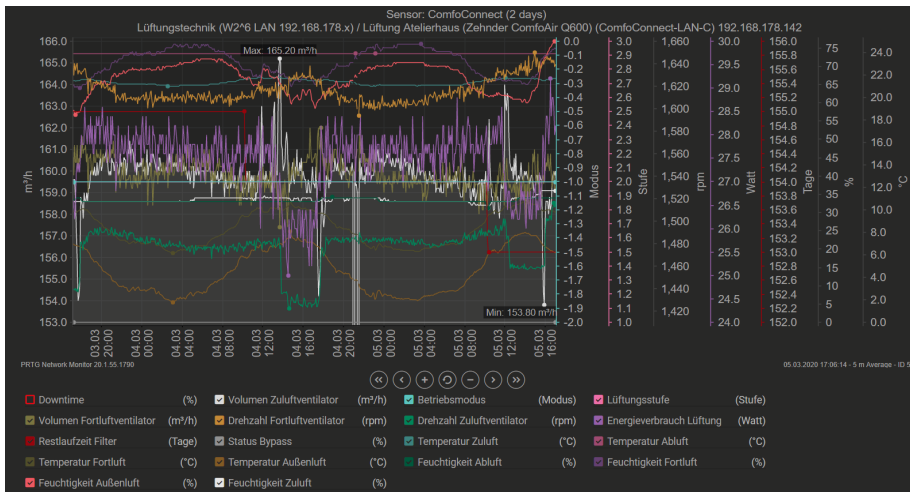
Et voilà. now PRTG is monitoring my ventilation systems!



The sensor provides a lot of data (currently in German):

- Betriebsmodus
- Energieverbrauch Lüftung
- Lüftungsstufe
- Restlaufzeit Filter
- Status Bypass

- Drehzahl Fortluftventilator/Zuluftventilator
- Feuchtigkeit Abluft/Außenluft/Fortluft/Zuluft
- Temperatur Abluft/Außenluft/Fortluft/Zuluft
- Volumen Fortluftventilator/Zuluftventilator



Limitations/Future Development

There is one limitation to this sensor: We have not been able to read/analyze the error messages from the ventilation system, if there are any. Hopefully either Zehnder publishes its API or the Python library will be updated in the future.

Share this:



Author: Dirk Paessler

Founder and Chairman, Paessler AG; Founder and Executive, Carbon Drawdown Initiative

[View all posts by Dirk Paessler](#)

March 5, 2020 / Smart Home and IoT /

Dirk Paessler's Personal Blog /

