pyflexit

["Sabesto <sabesto@gmail.com>", "Martin Høy <marhoy@gmail.c

Apr 07, 2020

CONTENTS:

CHAPTER

ONE

API DOCUMENTATION

1.1 Functions and classes

Monitor and control a Flexit ventilation aggregate via Modbus.

pyflexit.**autodetect_flexit_model**(*client*, *unit*)
Automatically detect what Flexit model we're talking to.

Parameters

- **client** – A modbus client
- **unit** – The modbus slave id of the Flexit aggregate

Returns A model string, either "CI66", "Nordic" or "EcoNordic"

pyflexit.**aggregate**(*client*, *unit: int*, *model: Optional[str] = None*)
Returns the appropriate class, depending on the model.

Parameters

- **client** – A modbus client
- **unit** – The modbus slave id of the Flexit aggregate
- **model** – Optionally specify the Flexit model, otherwise we autodetect.

Returns An instance of the appropriate Flexit class

1.2 Common API

The following methods and properties are available for both CI66 and Nordic models.

class pyflexit.common.**CommonAPI**(*client*, *unit: int*)
Base class to be inherited by the different Flexit model classes.

This ensures compatibility across different Flexit models. You will not be using this class directly, it is inherited by the CI66 and Nordic classes.

class VentMode
Each subclass must define their own ventilation modes.

__init__(*client*, *unit: int*)
Initialize the object.

Parameters

- **client** – A configured and connected modbus client, as returned by `pymodbus.client.sync.ModbusSerialClient()`
- **unit** – The modbus ID of your aggregate. For the Nordic series, this number would be 1. For CI66, it is configurable with dip-switches, but it is typically 21.

property outside_air_temp
Get the measured outside air temperature.

Example

```
>>> unit.outside_air_temp
-1.2
```

property extract_air_temp
Get the measured air temperature of the air being extracted.

This corresponds to the average temperature in the house.

Example

```
>>> unit.extract_air_temp
21.3
```

property supply_air_temp
Get the measured temperature of the supply air.

This is the fresh (possibly heated) air that goes into the rooms.

Example

```
>>> unit.supply_air_temp
20.4
```

property air_temp_setpoint
Get or set the temperature setpoint for supply air.

This property can be read from and written to.

Example

```
>>> unit.air_temp_setpoint = 21
>>> unit.air_temp_setpoint
21.0
```

property vent_modes
Returns a tuple of strings with all the valid ventilation modes.

Examples

For a CI66 adapter, this would be:

```
>>> ci66_unit.vent_modes
('Off', 'Min', 'Normal', 'Max')
```

For a Nordic series aggregate, this would be:

```
>>> nordic_unit.vent_modes
('Off', 'Away', 'Home', 'High')
```

property vent_mode

This property can be used to set or get the ventilation mode.

Whet getting the current ventilation mode, a string is returned. When setting the ventilation mode, a string is expected. The string needs to be one of the values returned by the vent_modes property.

Examples

```
>>> ci66_unit.vent_mode = "Min"
>>> ci66_unit.vent_mode
'Min'
```

```
>>> nordic_unit.vent_mode = "High"
>>> nordic_unit.vent_mode
'High'
```

property heat_exchanger_speed

Get the current speed of the rotating heat exchanger.

The value is returned in percent from 0-100.

Example

```
>>> unit.heat_exchanger_speed
100
```

property electric_heater_power

Get the current power of the electric heating coil.

The value is returned in percent from 0-100.

Example

```
>>> unit.electric_heater_power
42
```

property filter_runtime

Gets the number of hours in operation since last filter change.

Example

```
>>> unit.filter_runtime
1344
```

1.3 CI66

These methods and properties are unique for aggregates with the CI66 modbus adapter:

class pyflexit.**CI66**(*client*, *unit: int*)
This class supports the Flexit CI66 modbus adapter.

The CI66 adapter is compatible with the K2, UNI 2, UNI 3 and UNI 4 aggregates.

Example

This is an example for a Flexit CI66 adapter:

```
import pyflexit
from pymodbus.client.sync import ModbusSerialClient

client = ModbusSerialClient(
    method='rtu',
    port='/dev/ttyUSB0',
    stopbits=1,
    bytesize=8,
    parity='E',
    baudrate=56000,
    timeout=2)
client.connect()
ci66_unit = pyflexit.aggregate(client, unit=21, model="CI66")
```

class VentMode
Fan modes of the CI66.

Off = 0

Min = 1

Normal = 2

Max = 3

__init__(*client*, *unit: int*)
Initialize a CI66 object.

Parameters

- **client** (*mobus client*) – A configured modbus client.
- **unit** (*int*) – The modbus ID of your CI66 adapter.

property heating_enabled
Is the heating module enabled?

Returns `True` or `False` depending on the status.

property replace_filter_alarm
Status of filter change alarm.

Returns `True` when it's time to change the filter, otherwise `False`.

1.4 Nordic

These methods and properties are unique for the Nordic series of aggregates:

class `pyflexit.`**Nordic**(*client*, *unit: int*)

Supports the Flexit Nordic models (S2, S3, S4, CL2, CL3 and CL4).

The climate centrals EcoNordic W4 and WH4 are also supported, but (currently) only the ventilation part.

Example

This is an example for a Flexit Nordic aggregate:

```
import pyflexit
from pymodbus.client.sync import ModbusSerialClient

client = ModbusSerialClient(
    method='rtu',
    port='/dev/ttyUSB0',
    stopbits=1,
    bytesize=8,
    parity='E',
    baudrate=9600,
    timeout=2)
client.connect()
nordic_unit = pyflexit.aggregate(client, unit=1, model="Nordic")
```

class VentMode

For the Nordic series, these ventilation modes are supported.

Off = 1

Away = 2

Home = 3

High = 4

__init__(*client*, *unit: int*)

Initialize the object.

Parameters

- **client** (*modbus client*) – A Mobus client from pymodbus.client
- **unit** (*int*) – The modbus ID of the Flexit unit.

property air_temp_setpoint

Get or set the temperature setpoint for supply air.

The Nordic series have two separate target temperatures, one for "Home" and one for "Away". If the current operating mode is "Away", the Away-setpoint is returned. Otherwise, the Home-setpoint is returned.

Example

```
>>> unit.air_temp_setpoint = 21
>>> unit.air_temp_setpoint
21.0
```

property home_temp_setpoint
Get or set the target temperature for supply air when in Home-mode.

In addition to the `air_temp_setpoint` property which depends on the current ventilation mode, it is also possible to get or set the "Home"-setpoint directly, using this property.

Example

```
>>> nordic_unit.away_temp_setpoint = 23
>>> nordic_unit.away_temp_setpoint
23.0
```

property away_temp_setpoint
Get or set the target temperature for supply air when in Away-mode.

In addition to the `air_temp_setpoint` property which depends on the current ventilation mode, it is also possible to get or set the "Away"-setpoint directly, using this property.

Example

```
>>> nordic_unit.away_temp_setpoint = 19
>>> nordic_unit.away_temp_setpoint
19.0
```

property exhaust_air_temp
Get exhaust air temperature.

This is the temperature of the air being blown out of the house.

property exhaust_fan_speed
Get speed of the exhaust air fan, in percent from 0-100.

property supply_fan_speed
Get speed of supply air fan, in percent from 0-100.

property filter_remaining_time
Time until filter change (hours).

property room_humidity_1
Get value of humidity sensor 1, in percent from 0-100.

This is only available if you have a CI75 adapter and a CI77 sensor.

property room_humidity_2
Get value of humidity sensor 1, in percent from 0-100.

This is only available if you have a CI75 adapter and a CI77 sensor.

property room_humidity_3
Get value of humidity sensor 1, in percent from 0-100.

This is only available if you have a CI75 adapter and a CI77 sensor.

property room_airquality
Get value of air quality sensor 1, in ppm from 0-2000.

This is only available if you have a CI75 adapter and a CI76 sensor.

property efficiency
Calculate the efficiency of the heat exchanger.

The efficiency for a counterflow heat exchanger is defined by

$$eta = fracT_{hot,in} - T_{hot,out}T_{hot,in} - T_{cold,in}$$

and will be a number between 0 and 1.

Example

```
>>> print(f"Efficiency: {nordic_unit.efficiency:2.1%}")
Efficiency: 76.9%
```

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX

Symbols

A

C

E

F

H

M

N

O

P

R

S

V