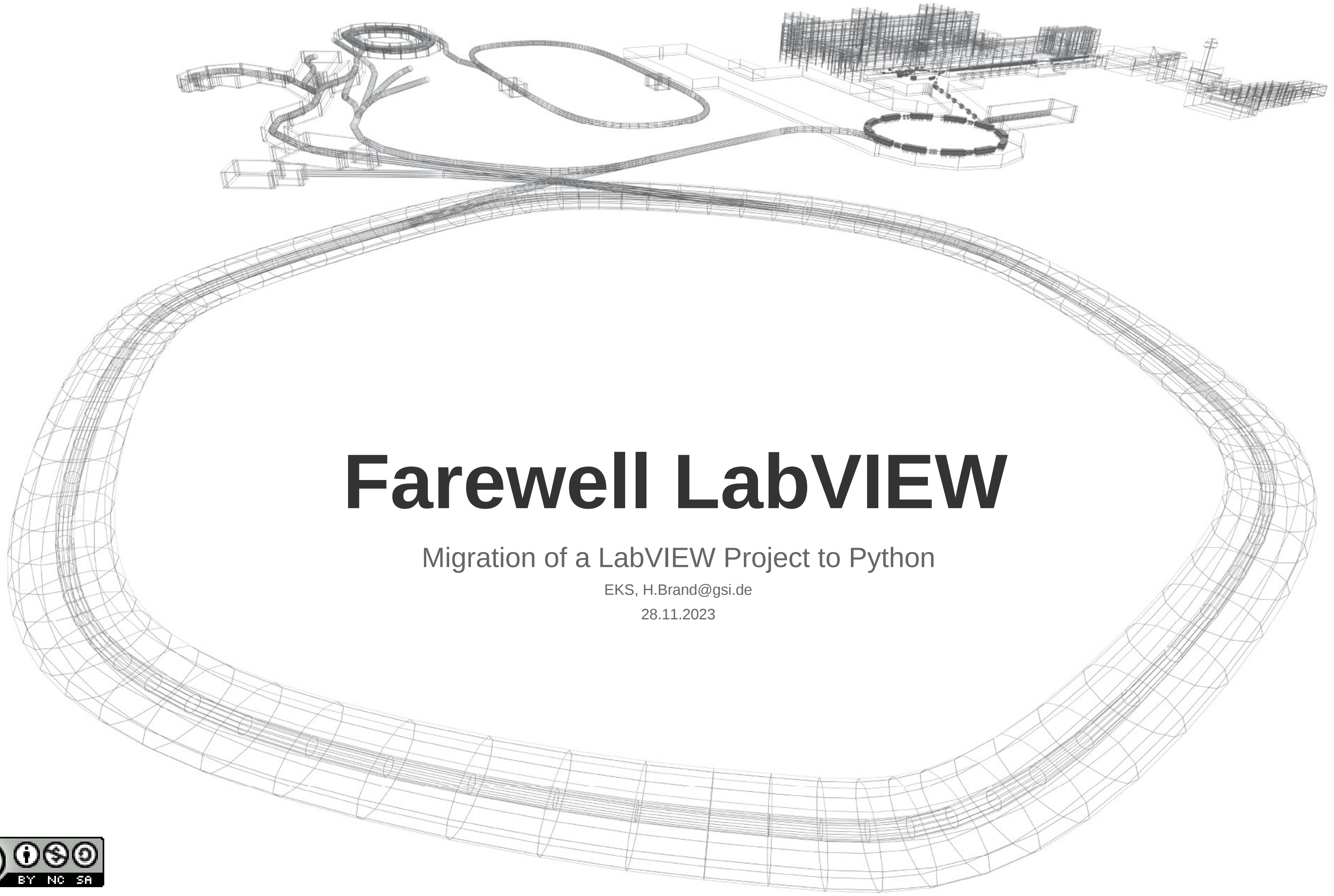


Farewell LabVIEW

Migration of a LabVIEW Project to Python

EKS, H.Brand@gsi.de

28.11.2023

Agenda

- Status & Zukunft
- Migrationsprojekt
  - LabVIEW RT basiertes Ausheizen des ESR/SIS
    - Python / pykka / PyAcdaq / paho-mqtt
    - Mosquitto / MQTT-Explorer / Telegraf / InfluxDB / Grafana

Empfehlung und Abkündigung

- Brief vom 23. April 2021
  - *„Empfehlungen zur Nutzung der Lizenzen“, H. Brand, EEL*
- Um mindestens das finanzielle Risiko zu begrenzen, werden in Rücksprache mit unserer stellvertretenden Forschungsdirektorin
  Y. Leifels folgende Empfehlungen zur Nutzung der verfügbaren Lizenzen vorgeschlagen:
  - Entwickler sollen in ihren Aktivitäten nicht eingeschränkt werden.
  - Alle verfügbaren Module mögen eingesetzt werden, um Anwendungen möglichst effizient zu entwickeln.
  - Entwickler mögen auf modulare Software-Architekturen achten und mögliche Migrationsstrategien berücksichtigen, um einen Vendor-Lock zu vermeiden.
  - Anwendungen sollen möglichst in ausführbare Programme (Executable) kompiliert werden. Sie sollen nicht dauerhaft in der Entwicklungsumgebung laufen.
    - Für viele Programme ist eine kostenfreie Runtime-Lizenz ausreichend. Das kann helfen, die Anzahl der benötigen Lizenzen niedrig zu halten.
    - Manche NI Software Module benötigen für das Ausführen eines Executable eine Deployment Lizenz. Während der Vertragslaufzeit ist das unproblematisch, sie wird vom NI-VLM gewährt. Dabei wird je eine Entwicklungslizenz für den Computer gezählt. Nur bei Beendigung des Vertrags würden dann entsprechende einmalige Kosten anfallen, die aber deutlich niedriger sind als eine Entwicklungsumgebung und nicht verfallen.
    - Zusammen mit einem Versionskontrollsystem, z.B. Git (git.gsi.de) kann es auch helfen, eine geordnetes Release-Management zu organisieren.
- Am 15.11.2023: Ankündigung des Endes des NI Enterprise Agreements zum 14.12.2025.
  - GSI/FAIR/HIM hat meiner Ansicht nach ausreichend viele Linzenzen im Bestand.
    Stand: 16.11.2023; Ausnahme: Vision,Deployment.
  - Noch keinen ernsthaften Widerspruch erhalten.

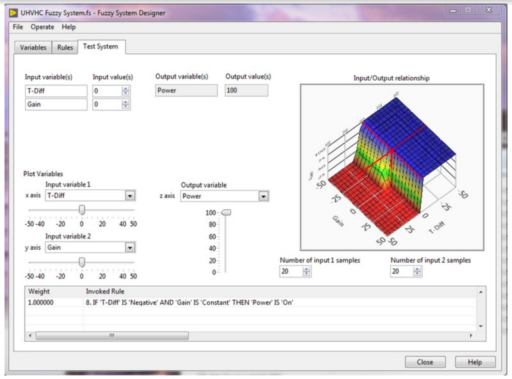
Alternativen zu NI

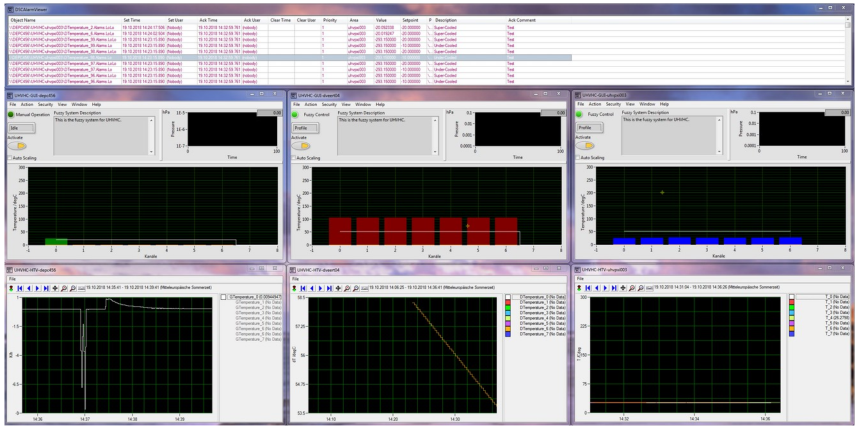
- Wie kann ein Vendor-Lock vermieden werden?
  - Benutzen offener Netzwerk-Schnittstellen für den Fernzugriff und offene Datenformate
  - Erstellen von Executables
    - Deployment-Lizenzen fallen nur einmalig bei der Beschaffung an.
  - Damit könnte man NI SW einfrieren und mit den bereits erworbenen Lizenzen eine Weile überleben.
- Welche Alternativen gibt es?
  - Bei GSI/FAIR:
    - EPICS, Python, PyAcdaq/Pykka, Tkinter/QT, FESA/Java, Siemens TIA, Beckhoff/TwinCat, Pures C/C++
    - NI LV-DSC -> EPICS (ALH/Archiver); MQTT / InfluxDB / Grafana
    - NI-Vision -> OpenCV; Other stuff: Open Source Libraries
    - NI-RT wird als RT praktisch nicht benutzt.
    - NI-FPGA: Expertise für FPGA Programmierung gibt es bei GSI.
      - https://git.gsi.de/EKS/Python/MUPPET
      - https://git.gsi.de/EE-LV/Drivers/MUPPET
  - Community: DOOCS, TANGO, TINE

Vacuum Heating System (Bakeout)

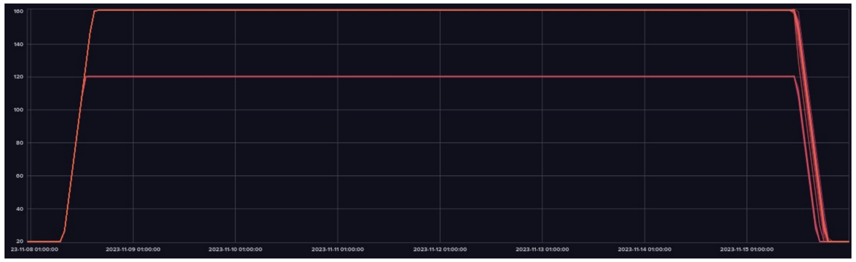
Anforderungen: Ausheizen der Strahlführung von ERS/SIS

- 3x PXIe basiertes LabVIEW 2018 RT System
  - NI PXI-8861RT Controller
    - LabVIEW mit AF/CS++ und Shared Variablen mit DSC
  - 4 Slots NI-6345 mit je 64 Kanälen analoge Eingänge für die Temperaturmessung
  - 4 Slots NI-6513 mit je 64 Kanälen digitale Ausgänge für die PWM-Ansteuerung der Heizungsnetzteile
  - Fuzzy Regelsystem


UHVHC Fuzzy System.fs - Fuzzy System Designer
File Operate Help
Variables Rules Test System
Input variable(s) Input value(s) Output variable(s) Output value(s)
T-Diff 0 Power 100
Gain 0
Input/Output relationship
Plot Variables
Input variable 1
x axis T-Diff
Input variable 2
y axis Gain
Output variable
z axis Power
Number of input 1 samples 20
Number of input 2 samples 20
Weight Invoked Rule
1.000000 8. IF 'T-Diff' IS 'Negative' AND 'Gain' IS 'Constant' THEN 'Power' IS 'On'
Close Help

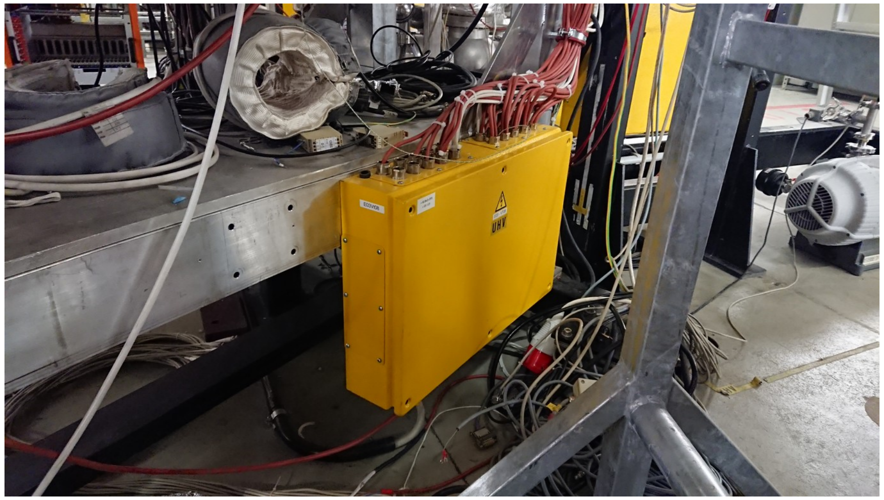


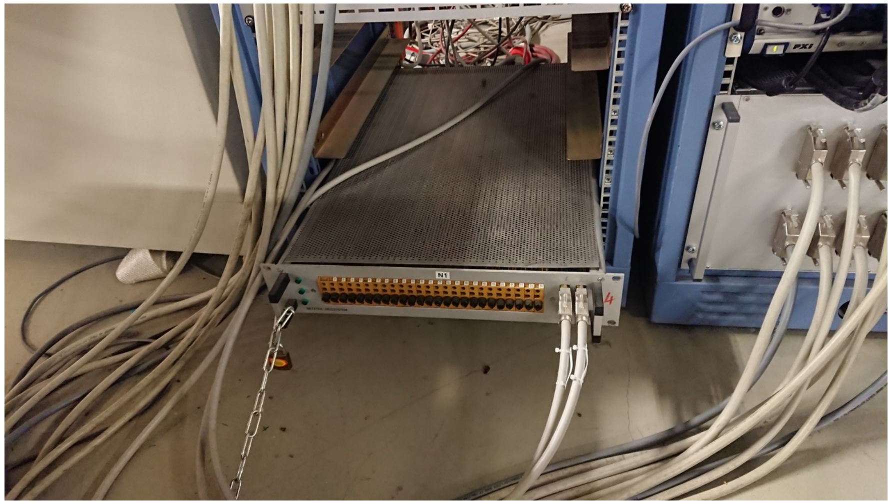
- LV-GUI zur Visualisierung auf Windows.
  - Zustangsanzeige und Logging
  - Alarme & Historische Daten (DSC)
  - Laden der Konfiguration für einen Heizzyklus
    - Heizprofil T-Soll(Kanal, Zeit) wird durch CSV-Datei vorgegeben.

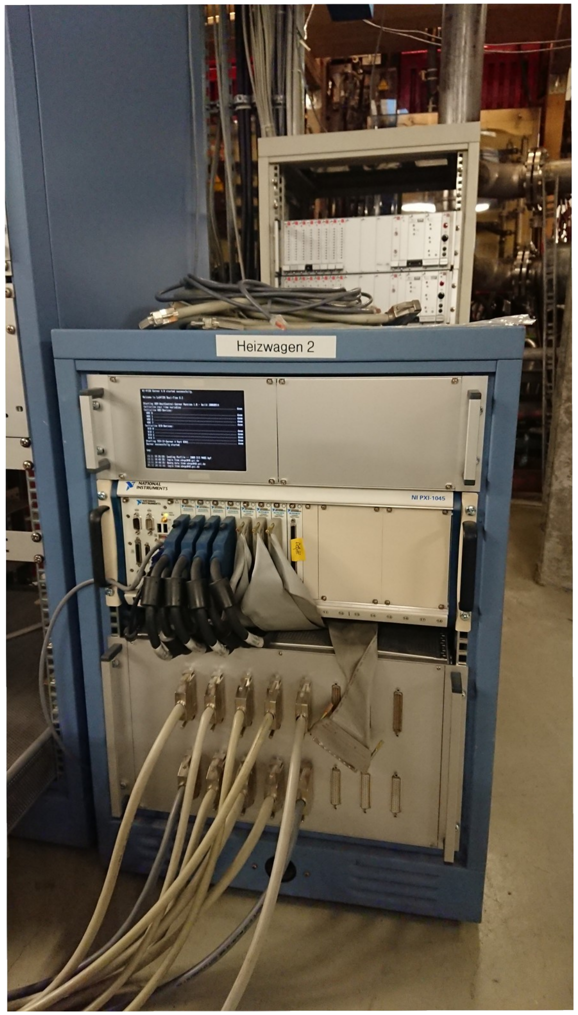


Vacuum Heating System (Bakeout)

Bilder von der Hardware







LV Migration Proposal → Open Source

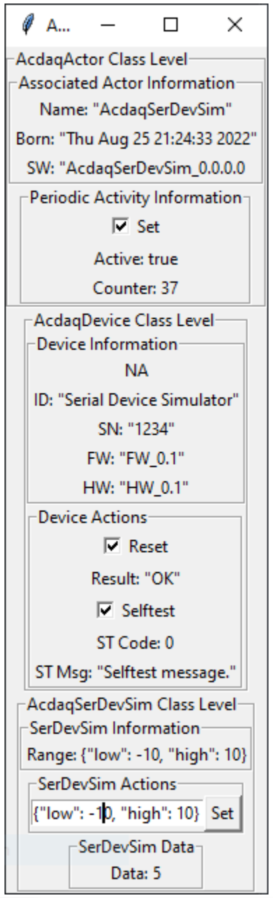
- Measurement & Automation Explorer
  - Maybe necessary or useful for NI hardware configuration and test.
- LabVIEW → Python
  - Real Time → Not really necessary
  - **AF/CS++** → Pykka/PyAcdaq
  - Shared Variables → MQTT / Mosquitto Broker
  - Distributed System Manager → MQTT Explorer
  - Data Logging and Supervisory Control
    - Data Logging → Telegraf / InfluxDB
    - Alarming
      - → Telegraf / InfluxDB-Task → MQTT
      - → Telegraf / Prometheus + Alertmanager → MQTT (to be tested)
  - GUI → Tkinter / QT / Grafana
  - Multi-threading → Multi-threading (Pykka)/ Multi-processing (Fuzzy)

Python: Pykka

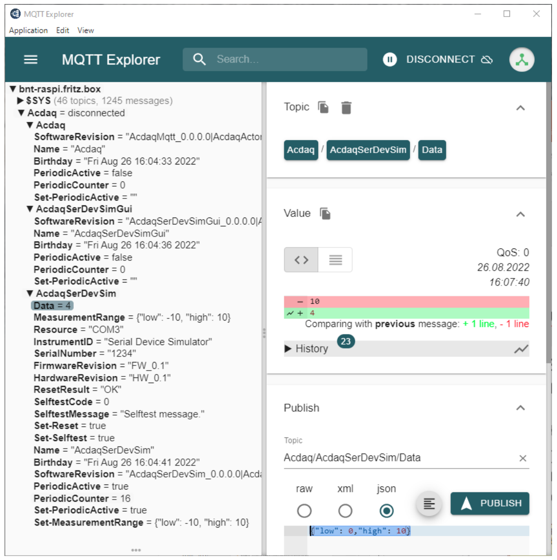
- Pykka: https://pykka.readthedocs.io/en/stable/
  - *Pykka is a Python implementation of the actor model. The actor model introduces some simple rules to control the sharing of state and cooperation between execution units, which makes it easier to build concurrent applications.*
  - **Inspiration**
    - *Much of the naming of concepts and methods in Pykka is taken from the Akka project which implements actors on the JVM. Though, Pykka does not aim to be a Python port of Akka, and supports far fewer features.*
    - *Notably, Pykka* ***does not*** *support the following features:*
      - *Supervision: Linking actors, supervisors, or supervisor groups.*
      - *Remoting: Communicating with actors running on other hosts.*
      - *Routers: Pykka does not come with a set of predefined message routers, though you may make your own actors for routing messages.*
- *Very similar to NI Actor Framework*

PyAcdaq Overview

- PyAcdaq is a Python 3.10.x library for Automation Control & Data Acquisition System.
  - It is meant as a possible substitute (Plan B) for our LabVIEW based *CS* / **CS++** frameworks or other LabVIEW Applications.
    - Git: https://git.gsi.de/EE-LV/ACDAQ/PyAcdaq
    - MQTT Explorer
    - Example GUI
    - UML class Diagrams


A...

AcdaqActor Class Level
Associated Actor Information
Name: "AcdaqSerDevSim"
Born: "Thu Aug 25 21:24:33 2022"
SW: "AcdaqSerDevSim_0.0.0.0"
Periodic Activity Information
☑ Set
Active: true
Counter: 37
AcdaqDevice Class Level
Device Information
NA
ID: "Serial Device Simulator"
SN: "1234"
FW: "FW_0.1"
HW: "HW_0.1"
Device Actions
☑ Reset
Result: "OK"
☑ Selftest
ST Code: 0
ST Msg: "Selftest message."
AcdaqSerDevSim Class Level
SerDevSim Information
Range: {"low": -10, "high": 10}
SerDevSim Actions
{"low": -10, "high": 10} Set
SerDevSim Data
Data: 5



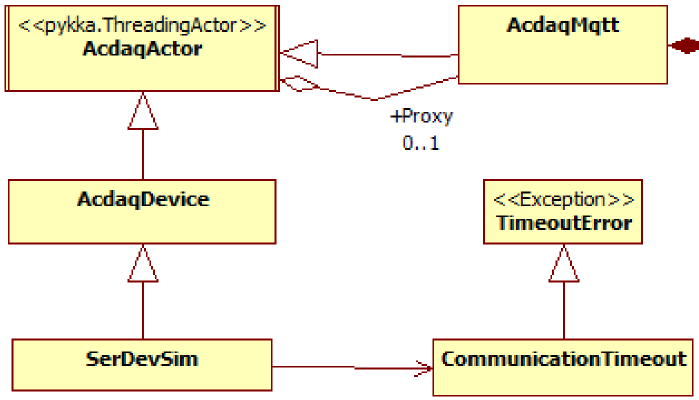
```
▼ bnt-raspi.fritz.box
  ▶ $SYS (46 topics, 1245 messages)
  ▼ Acdaq = disconnected
    ▼ Acdaq
      SoftwareRevision = "AcdaqMqtt_0.0.0.0|AcdaqActor
      Name = "Acdaq"
      Birthday = "Fri Aug 26 16:04:33 2022"
      PeriodicActive = false
      PeriodicCounter = 0
      Set-PeriodicActive = ""
    ▼ AcdaqSerDevSimGui
      SoftwareRevision = "AcdaqSerDevSimGui_0.0.0.0|A
      Name = "AcdaqSerDevSimGui"
      Birthday = "Fri Aug 26 16:04:36 2022"
      PeriodicActive = false
      PeriodicCounter = 0
      Set-PeriodicActive = ""
    ▼ AcdaqSerDevSim
      Data = 4
      MeasurementRange = {"low": -10, "high": 10}
      Resource = "COM3"
      InstrumentID = "Serial Device Simulator"
      SerialNumber = "1234"
      FirmwareRevision = "FW_0.1"
      HardwareRevision = "HW_0.1"
      ResetResult = "OK"
      SelftestCode = 0
      SelftestMessage = "Selftest message."
      Set-Reset = true
      Set-Selftest = true
      Name = "AcdaqSerDevSim"
      Birthday = "Fri Aug 26 16:04:41 2022"
      SoftwareRevision = "AcdaqSerDevSim_0.0.0.0|Acda
      PeriodicActive = true
      PeriodicCounter = 16
      Set-PeriodicActive = true
      Set-MeasurementRange = {"low": -10, "high": 10}
```

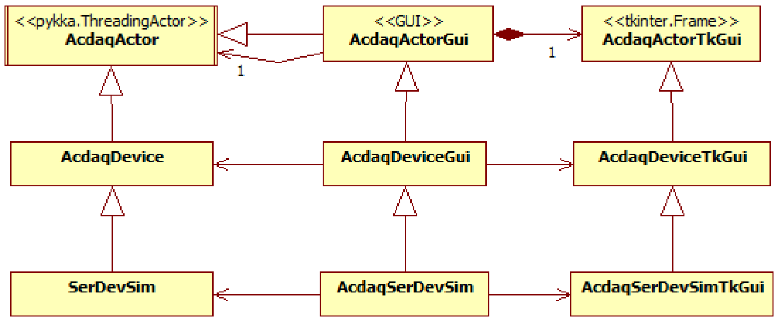
```
 − 10
 + 4
Comparing with previous message: + 1 line, - 1 line
```

Topic: Acdaq/AcdaqSerDevSim/Data

raw ( ) xml ( ) [✓ json]

```
{"low": 0,"high": 10}
```



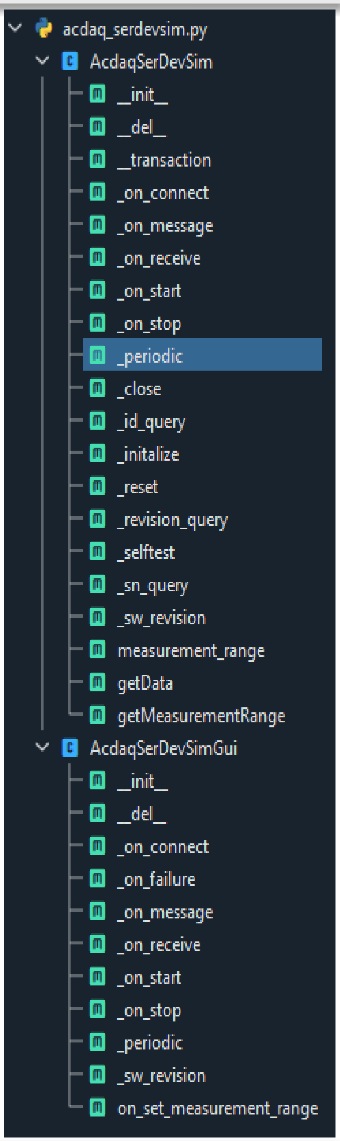


AcdaqSerDevSim: Code Examples

https://git.gsi.de/EKS/Python/ACDAQ/PyAcdaq/-/tree/master/examples/scpi_device

```
# Protected override method
def _initalize(self,
               resource,
               id_query_flag,
               reset_flag,
               reset_options,
               selftest_flag):
    """
    Open serial connection and initialize instrument.

    Returns
    -------
    int
        Error code.
    """
    logging.debug(self._name +
        ' SerDevSim _initialize() Initializing connection.')
    try:
        self.__ser = serial.Serial(resource, timeout=1)
        if id_query_flag:
            self.id_query()
        self.revision_query()
        if reset_flag:
            self.reset(reset_options)
        if selftest_flag:
            self.selftest()
        self.measurement_range(self._range)
    except acdaq_exceptions.CommunicationTimeout as e:
        raise e
    pass
```

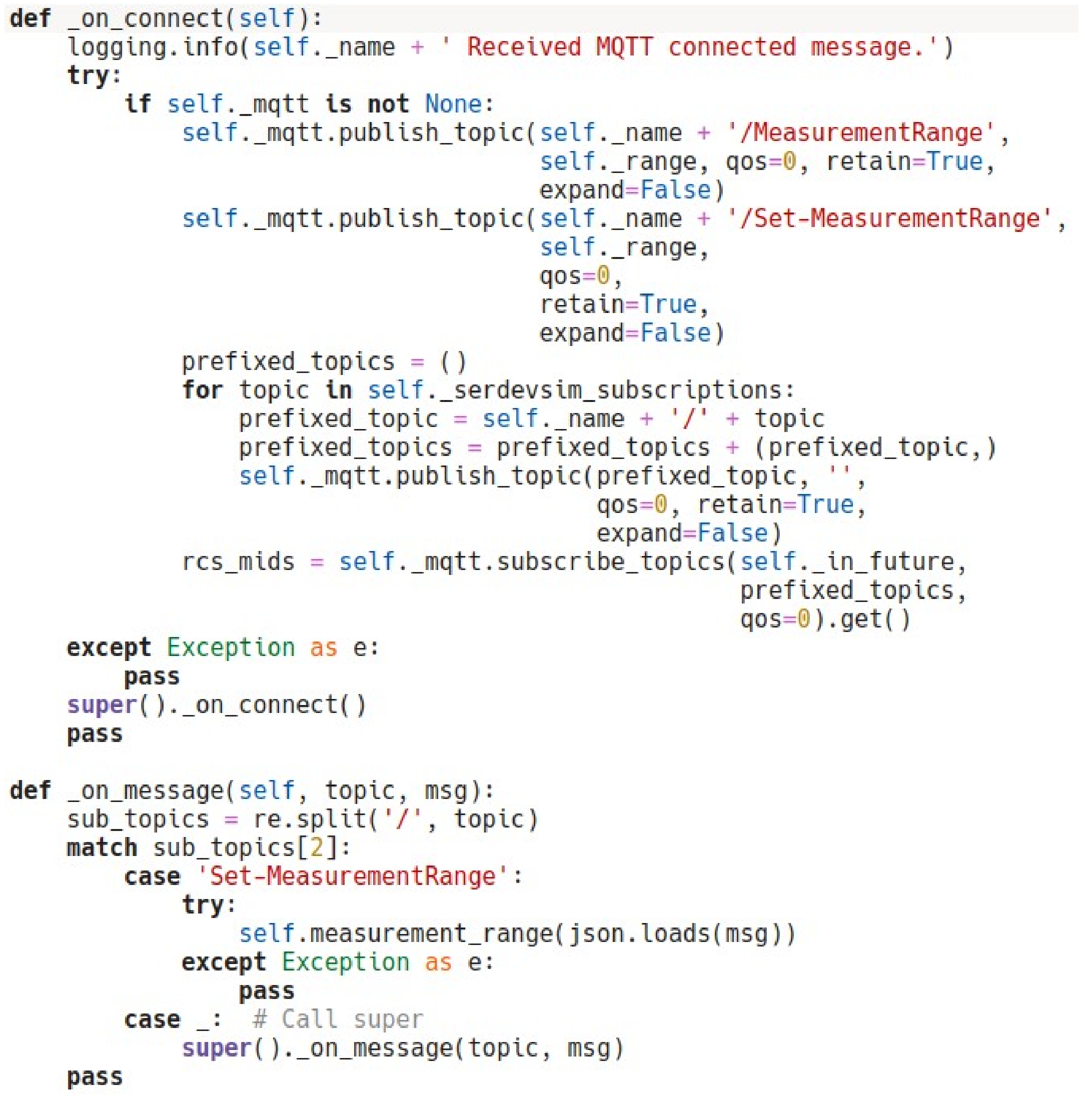
```
# Protected overide method
def _periodic(self):
    """
    Implementation of periodic actions.

    Read data from instrument.

    Returns
    -------
    None.
    """
    try:
        self._data = int(self.__transaction('READ?'))
        logging.debug(self._name + " Data=„
            + str(self._data))
        if self._mqtt is not None:
            self._mqtt.publish_topic(self._name +
        '/Data', self._data,
         qos=0, retain=False, expand=False)
    except acdaq_exceptions.CommunicationTimeout as e:
       logging.error(self._name + ' SerDevSim _periodic()
       Communication timed out for\
       command:{}'.format(e.command))
        raise e
    pass
```



```
acdaq_serdevsim.py
  AcdaqSerDevSim
    _init_
    _del_
    _transaction
    _on_connect
    _on_message
    _on_receive
    _on_start
    _on_stop
    _periodic
    _close
    _id_query
    _initalize
    _reset
    _revision_query
    _selftest
    _sn_query
    _sw_revision
    measurement_range
    getData
    getMeasurementRange
  AcdaqSerDevSimGui
    _init_
    _del_
    _on_connect
    _on_failure
    _on_message
    _on_receive
    _on_start
    _on_stop
    _periodic
    _sw_revision
    on_set_measurement_range
```

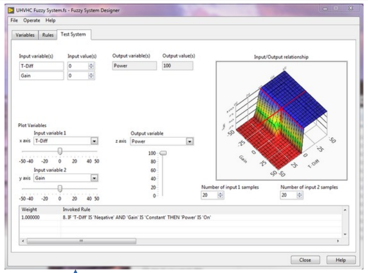
AcdaqSerDevSim: Code Examples

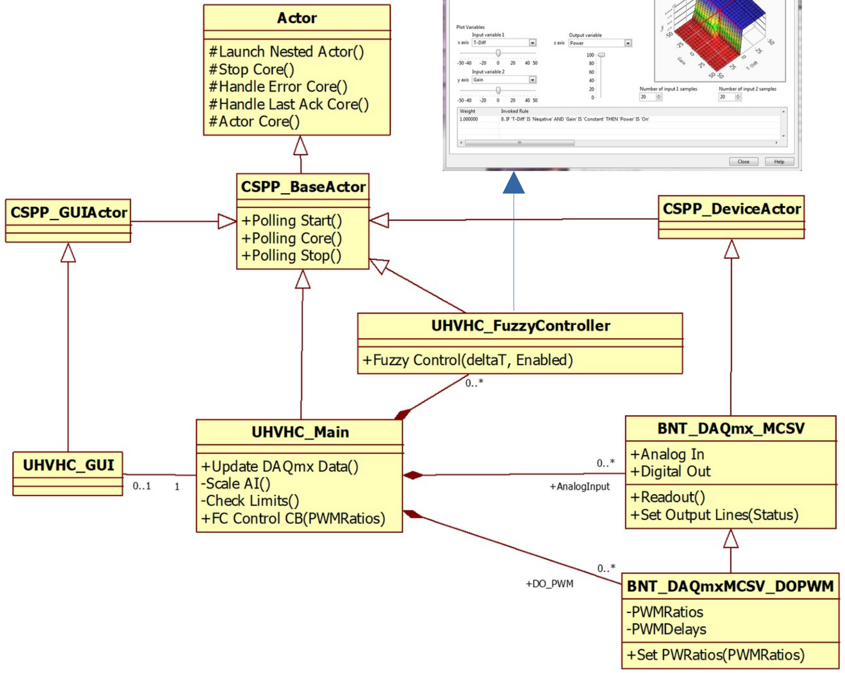
https://git.gsi.de/EKS/Python/ACDAQ/PyAcdaq/-/tree/master/examples/scpi_device

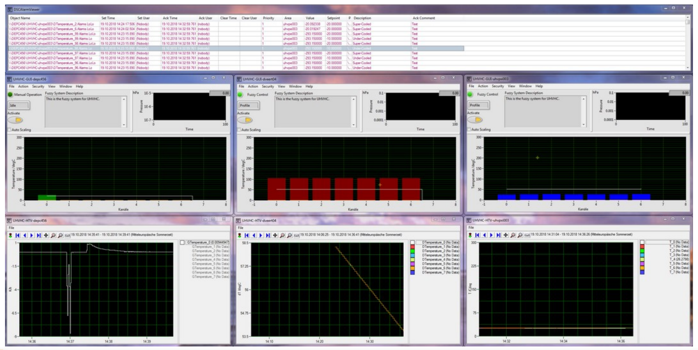


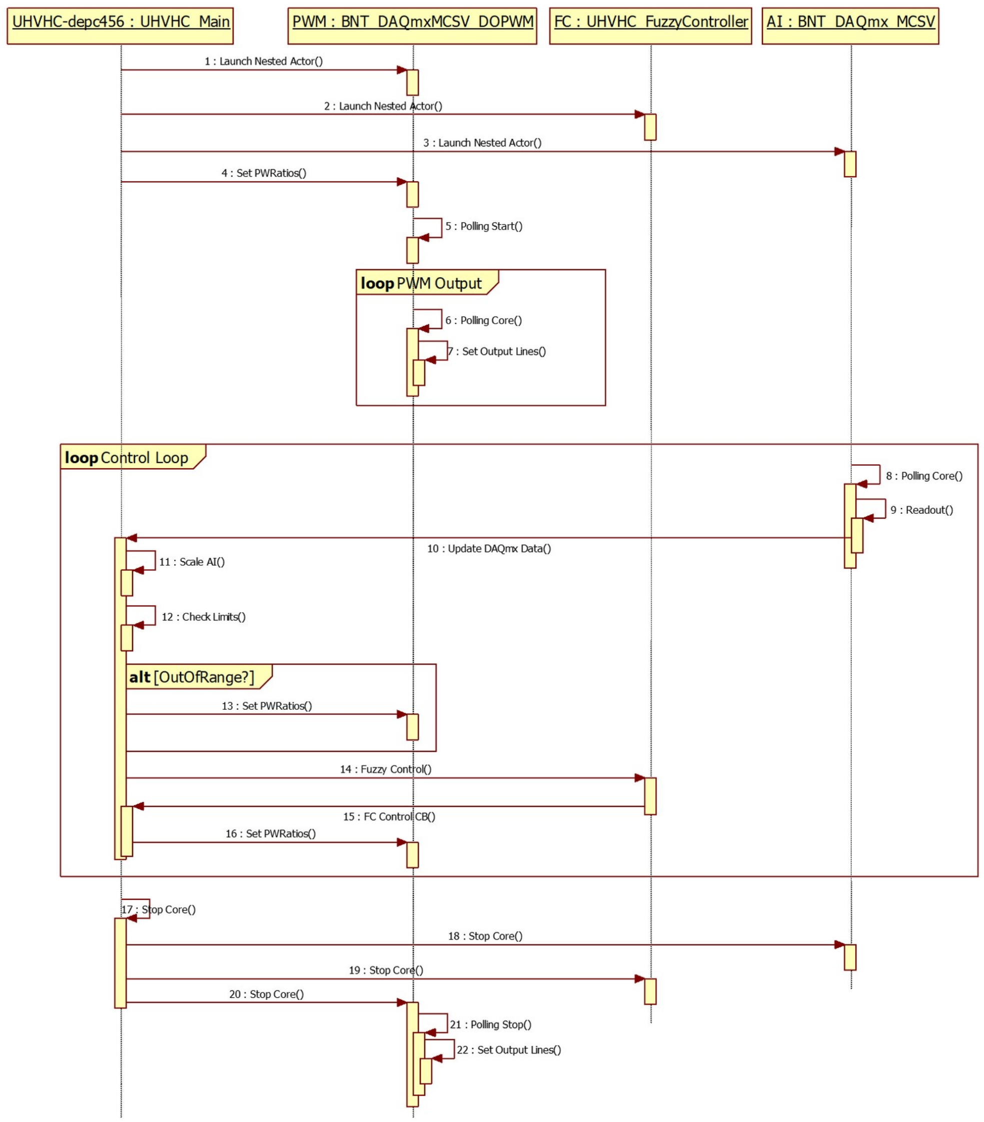
```python
def _on_connect(self):
    logging.info(self._name + ' Received MQTT connected message.')
    try:
        if self._mqtt is not None:
            self._mqtt.publish_topic(self._name + '/MeasurementRange',
                                     self._range, qos=0, retain=True,
                                     expand=False)
            self._mqtt.publish_topic(self._name + '/Set-MeasurementRange',
                                     self._range,
                                     qos=0,
                                     retain=True,
                                     expand=False)
            prefixed_topics = ()
            for topic in self._serdevsim_subscriptions:
                prefixed_topic = self._name + '/' + topic
                prefixed_topics = prefixed_topics + (prefixed_topic,)
                self._mqtt.publish_topic(prefixed_topic, '',
                                         qos=0, retain=True,
                                         expand=False)
            rcs_mids = self._mqtt.subscribe_topics(self._in_future,
                                                   prefixed_topics,
                                                   qos=0).get()

    except Exception as e:
        pass
    super()._on_connect()
    pass

def _on_message(self, topic, msg):
    sub_topics = re.split('/', topic)
    match sub_topics[2]:
        case 'Set-MeasurementRange':
            try:
                self.measurement_range(json.loads(msg))
            except Exception as e:
                pass
        case _:  # Call super
            super()._on_message(topic, msg)
    pass
```

Migrationsprojekt: LabVIEW RT basiertes Ausheizen des ESR/SIS
https://git.gsi.de/EE-LV/CSPP/GSI_VAC/VacuumHeating


UHVHC Fuzzy System.fs - Fuzzy System Designer
File Operate Help
Variables Rules Test System
Input/Output relationship
Close Help

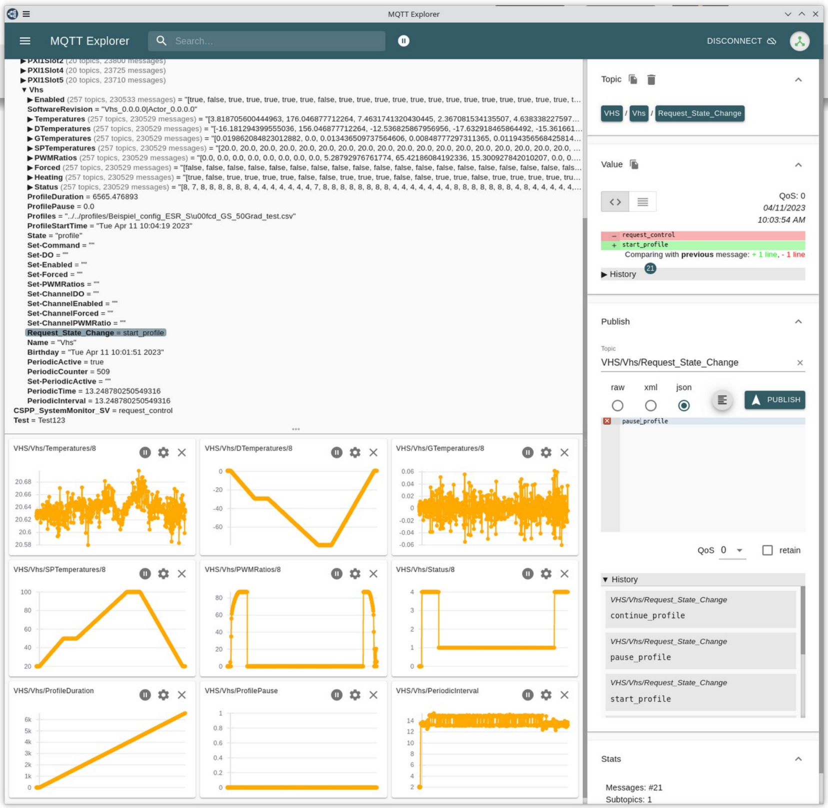






Vacuum Heating System (Bakeout)

- PyAcdaq/pykka based project for the GSI **V**acuum **H**eating **S**ystem (VHS) to be used at ESR and SIS.
  - https://git.gsi.de/EKS/Python/ACDAQ/VacuumHeatingSystem
- It is utilizing PXIe HW from NI. It measures temperatures from analog inputs and applies a fuzzy system to calculate the digital output pulse width ratio for the heating power supplies.
- It demonstrate a migration path from LabVIEW RT and DSC to
  - Python / pykka / PyAcdaq / ***nidaqmx***
  - python-statemachine / simpful / paho-mqtt
  - numpy / pandas / matplotlib
  - Mosquitto / MQTT-Explorer / Telegraf / InfluxDB / Grafana.
- Temperature Profile execution with no HW connected is working.
- Comissioning with connected power supplies was planned for May 2023. Done!



```
▶ PXI1Slot4 (20 topics, 23725 messages)
▶ PXI1Slot5 (20 topics, 23710 messages)
▼ Vhs
  ▶ Enabled (257 topics, 230533 messages) = "[true, false, true, true, true, true, true, false, true, true, true, true, true, true, true, true, true, true, true, true, t...
  SoftwareRevision = "Vhs_0.0.0.0|Actor_0.0.0.0"
  ▶ Temperatures (257 topics, 230529 messages) = "[3.818705600444963, 176.046877712264, 7.4631741320430445, 2.367081534135507, 4.638338227597...
  ▶ DTemperatures (257 topics, 230529 messages) = "[-16.181294399555036, 156.046877712264, -12.536825867956956, -17.632918465864492, -15.361661...
  ▶ GTemperatures (257 topics, 230529 messages) = "[0.019862084823012882, 0.0, 0.013436509737564606, 0.00848777297311365, 0.01194356568425814...
  ▶ SPTemperatures (257 topics, 230529 messages) = "[20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, ...
  ▶ PWMRatios (257 topics, 230529 messages) = "[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 5.28792976761774, 65.42186084192336, 15.300927842010207, 0.0, 0....
  ▶ Forced (257 topics, 230529 messages) = "[false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, fals...
  ▶ Heating (257 topics, 230529 messages) = "[true, false, true, true, true, true, false, false, true, true, true, false, false, true, true, false, true, true, true, true, tru...
  ▶ Status (257 topics, 230529 messages) = "[8, 7, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 4, 7, 8, 8, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 4, 8, 8, 8, 8, 8, 8, 4, 8, 4, 4, 4, 4, 4,...
  ProfileDuration = 6565.476893
  ProfilePause = 0.0
  Profiles = "../../profiles/Beispiel_config_ESR_S\u00fcd_GS_50Grad_test.csv"
  ProfileStartTime = "Tue Apr 11 10:04:19 2023"
  State = "profile"
  Set-Command = ""
  Set-DO = ""
  Set-Enabled = ""
  Set-Forced = ""
  Set-PWMRatios = ""
  Set-ChannelDO = ""
  Set-ChannelEnabled = ""
  Set-ChannelForced = ""
  Set-ChannelPWMRatio = ""
  Request_State_Change = start_profile
  Name = "Vhs"
  Birthday = "Tue Apr 11 10:01:51 2023"
  PeriodicActive = true
  PeriodicCounter = 509
  Set-PeriodicActive = ""
  PeriodicTime = 13.248780250549316
  PeriodicInterval = 13.248780250549316
CSPP_SystemMonitor_SV = request_control
Test = Test123
```

Topic: VHS / Vhs / Request_State_Change

Value — QoS: 0, 04/11/2023 10:03:54 AM

```
- request_control
+ start_profile
```

Comparing with previous message: + 1 line, - 1 line

Publish — Topic: VHS/Vhs/Request_State_Change — raw / xml / json [✓ json] — PUBLISH

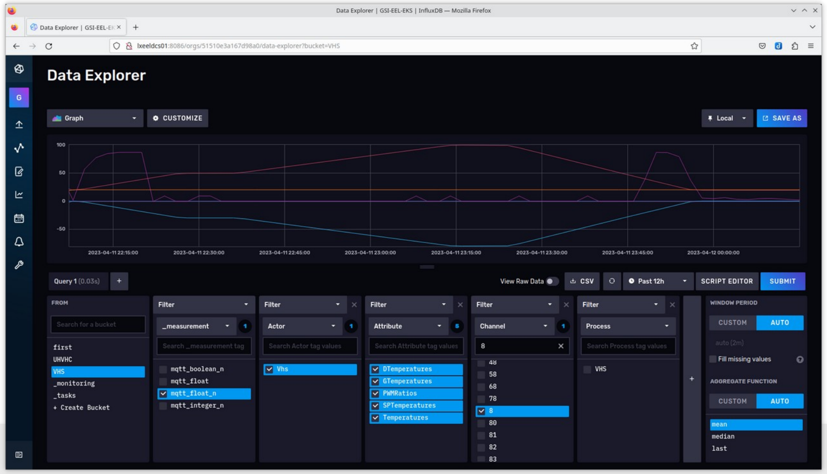
```
pause_profile
```

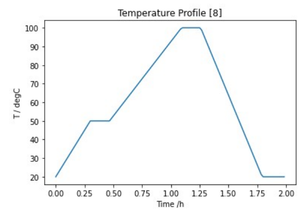
QoS 0 — retain

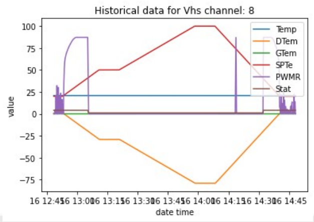
History

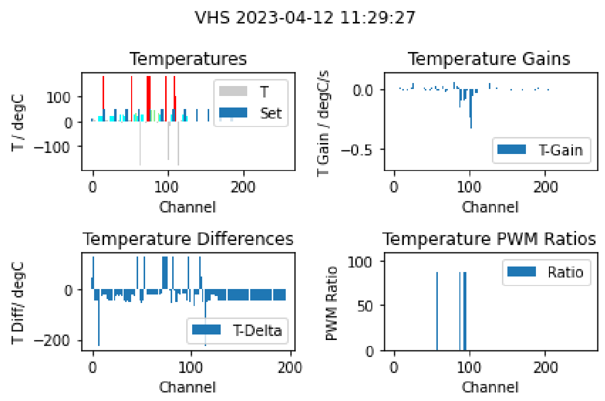
```
VHS/Vhs/Request_State_Change
continue_profile
VHS/Vhs/Request_State_Change
pause_profile
VHS/Vhs/Request_State_Change
start_profile
```

Stats — Messages: #21 — Subtopics: 1

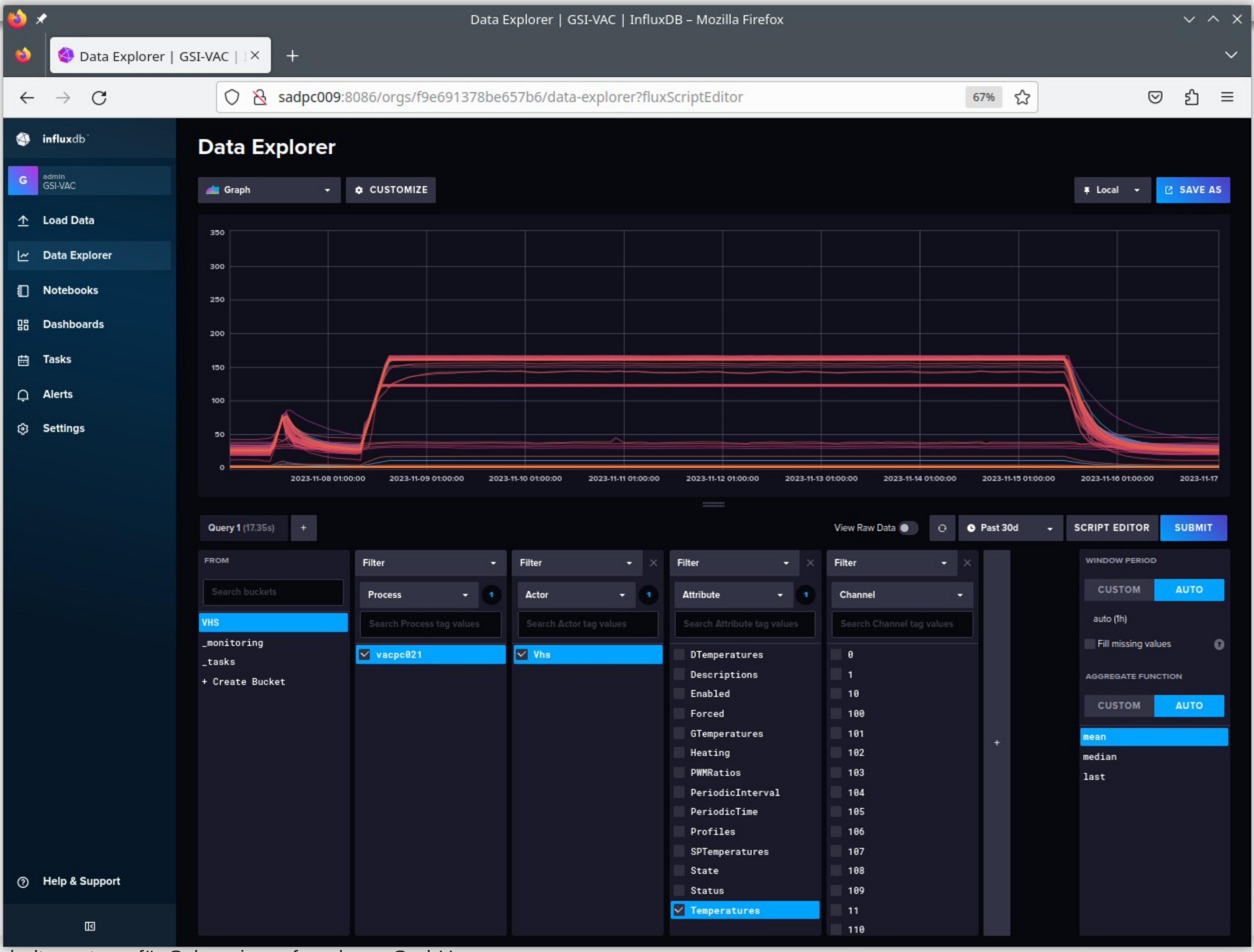




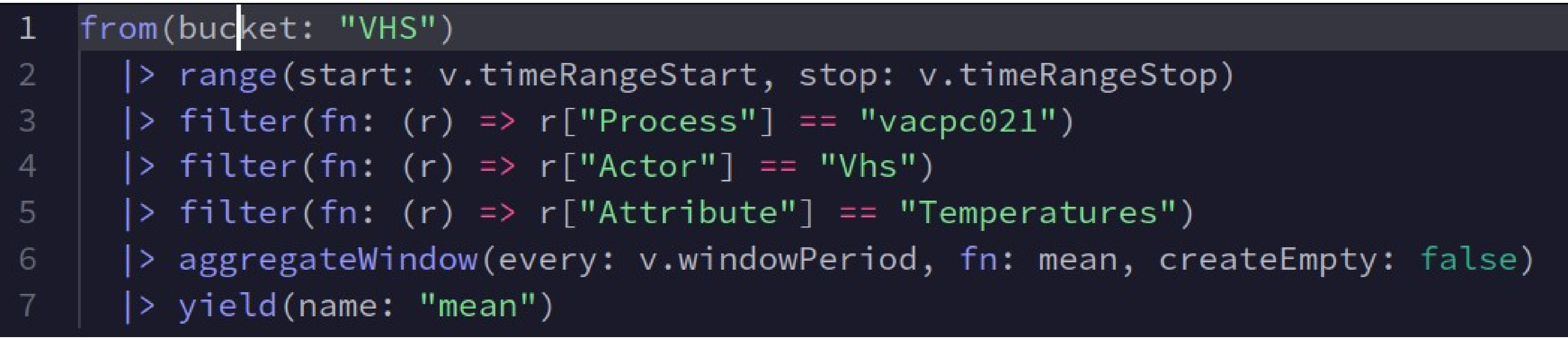




InfluxDB: Temperaturen
07.11. - 17.11.2023

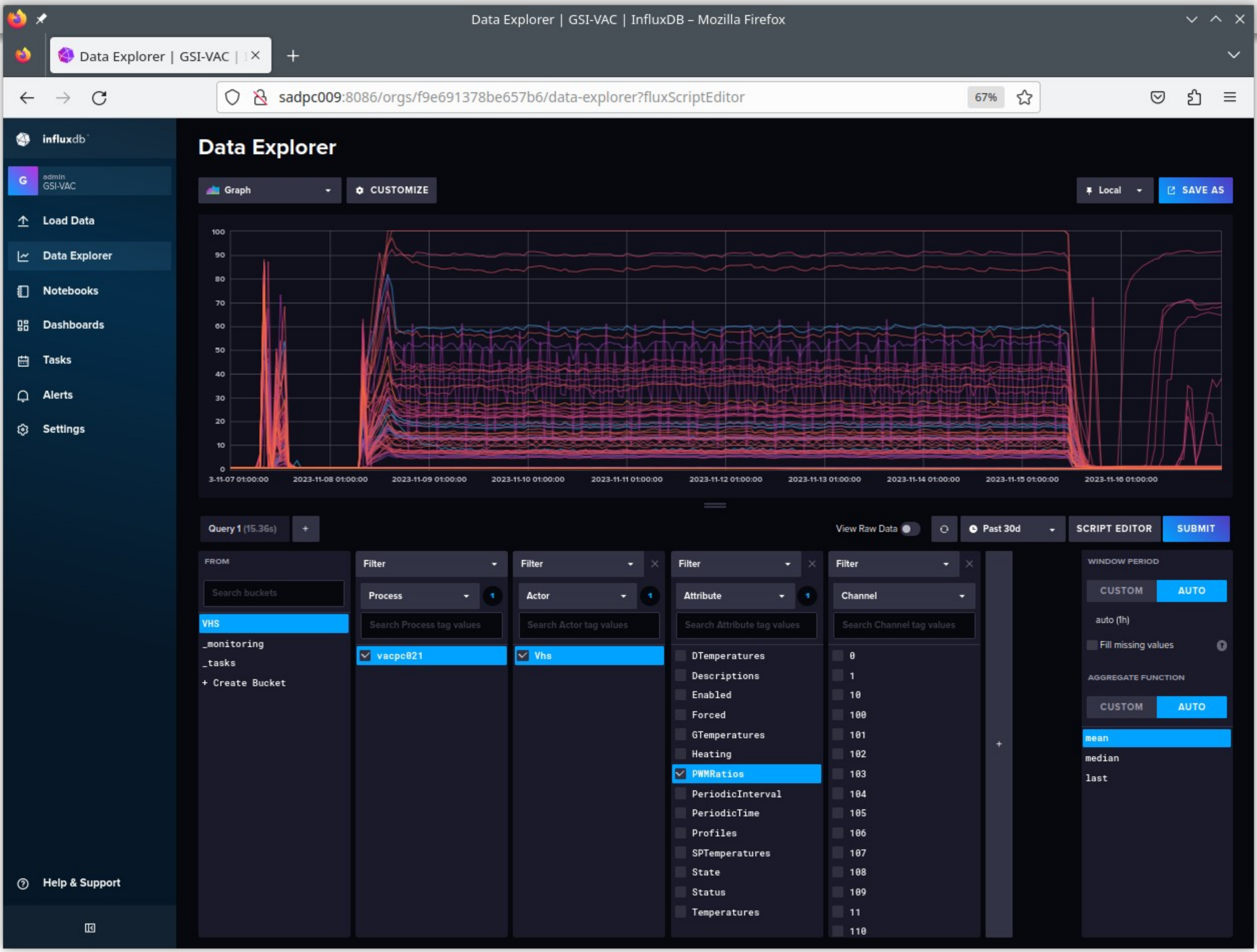


InfluxDB: Temperaturen, Flux-Query-Builder
https://docs.influxdata.com/flux/v0/



```
from(bucket: "VHS")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["Process"] == "vacpc021")
  |> filter(fn: (r) => r["Actor"] == "Vhs")
  |> filter(fn: (r) => r["Attribute"] == "Temperatures")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

InfluxDB: PWMRatios
07.11. - 17.11.2023


Data Explorer | GSI-VAC | InfluxDB – Mozilla Firefox

sadpc009:8086/orgs/f9e691378be657b6/data-explorer?fluxScriptEditor

Data Explorer

Graph · CUSTOMIZE · Local · SAVE AS

Query 1 (15.36s) · View Raw Data · Past 30d · SCRIPT EDITOR · SUBMIT

FROM	Filter	Filter	Filter	Filter
VHS	Process: vacpc021	Actor: Vhs	Attribute: PWMRatios	Channel

Attribute tag values: DTemperatures, Descriptions, Enabled, Forced, GTemperatures, Heating, PWMRatios, PeriodicInterval, PeriodicTime, Profiles, SPTemperatures, State, Status, Temperatures

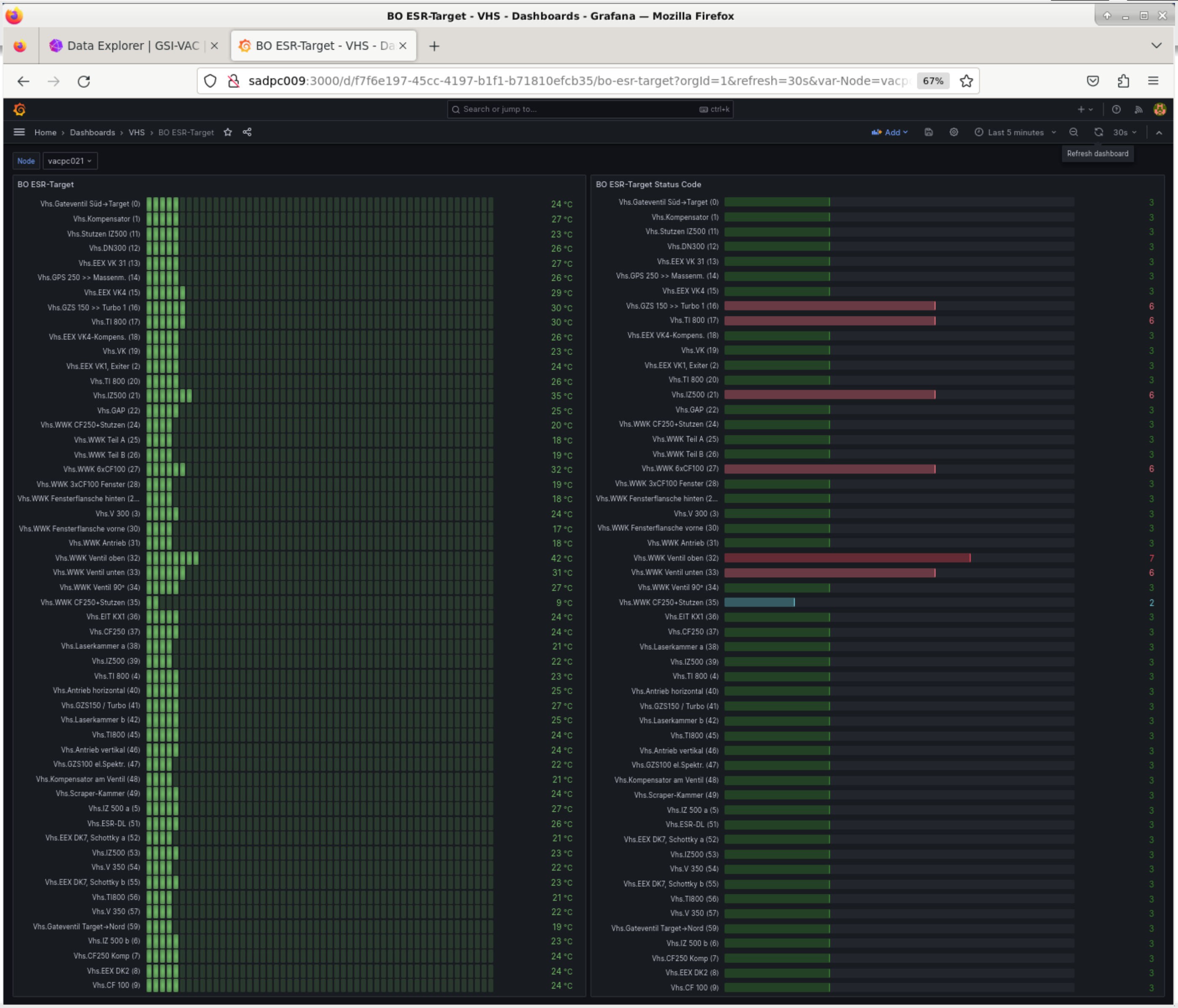
Channel tag values: 0, 1, 10, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 11, 110

WINDOW PERIOD: AUTO — auto (1h)

Fill missing values

AGGREGATE FUNCTION: AUTO — mean, median, last

Grafana
Bar View of Temperature and Status with Channel names



BO ESR-Target - VHS - Dashboards - Grafana — Mozilla Firefox

Data Explorer | GSI-VAC | BO ESR-Target - VHS - Da

sadpc009:3000/d/f7f6e197-45cc-4197-b1f1-b71810efcb35/bo-esr-target?orgId=1&refresh=30s&var-Node=vacp 67%

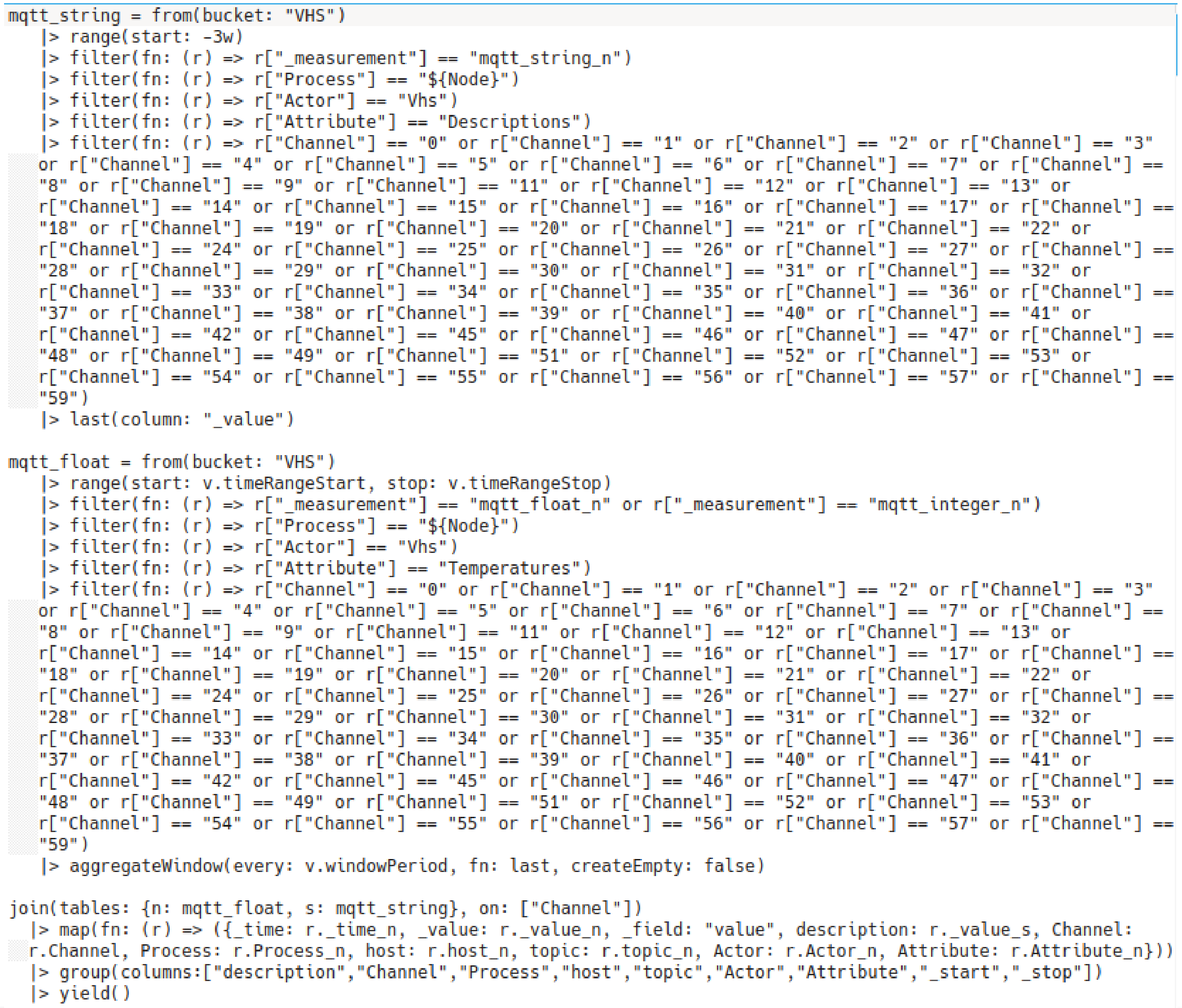
Home › Dashboards › VHS › BO ESR-Target

Last 5 minutes 30s

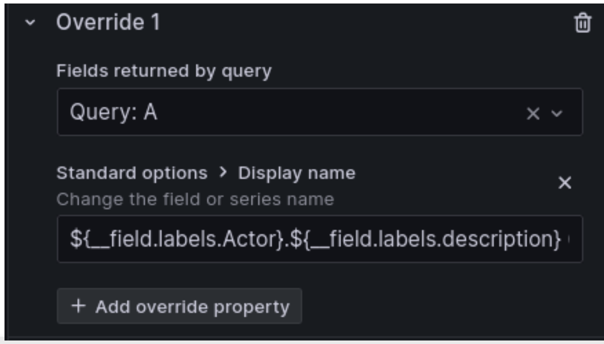
Node vacpc021

BO ESR-Target		BO ESR-Target Status Code
Vhs.Gateventil Süd→Target (0)	24 °C	3
Vhs.Kompensator (1)	27 °C	3
Vhs.Stutzen IZ500 (11)	23 °C	3
Vhs.DN300 (12)	26 °C	3
Vhs.EEX VK 31 (13)	27 °C	3
Vhs.GPS 250 >> Massenm. (14)	26 °C	3
Vhs.EEX VK4 (15)	29 °C	3
Vhs.GZS 150 >> Turbo 1 (16)	30 °C	6
Vhs.TI 800 (17)	30 °C	6
Vhs.EEX VK4-Kompens. (18)	26 °C	3
Vhs.VK (19)	23 °C	3
Vhs.EEX VK1, Exiter (2)	24 °C	3
Vhs.TI 800 (20)	26 °C	3
Vhs.IZ500 (21)	35 °C	6
Vhs.GAP (22)	25 °C	3
Vhs.WWK CF250+Stutzen (24)	20 °C	3
Vhs.WWK Teil A (25)	18 °C	3
Vhs.WWK Teil B (26)	19 °C	3
Vhs.WWK 6xCF100 (27)	32 °C	6
Vhs.WWK 3xCF100 Fenster (28)	19 °C	3
Vhs.WWK Fensterflansche hinten (2...	18 °C	3
Vhs.V 300 (3)	24 °C	3
Vhs.WWK Fensterflansche vorne (30)	17 °C	3
Vhs.WWK Antrieb (31)	18 °C	3
Vhs.WWK Ventil oben (32)	42 °C	7
Vhs.WWK Ventil unten (33)	31 °C	6
Vhs.WWK Ventil 90° (34)	27 °C	3
Vhs.WWK CF250+Stutzen (35)	9 °C	2
Vhs.EIT KX1 (36)	24 °C	3
Vhs.CF250 (37)	24 °C	3
Vhs.Laserkammer a (38)	21 °C	3
Vhs.IZ500 (39)	22 °C	3
Vhs.TI 800 (4)	23 °C	3
Vhs.Antrieb horizontal (40)	25 °C	3
Vhs.GZS150 / Turbo (41)	27 °C	3
Vhs.Laserkammer b (42)	25 °C	3
Vhs.TI800 (45)	24 °C	3
Vhs.Antrieb vertikal (46)	24 °C	3
Vhs.GZS100 el.Spektr. (47)	22 °C	3
Vhs.Kompensator am Ventil (48)	21 °C	3
Vhs.Scraper-Kammer (49)	24 °C	3
Vhs.IZ 500 a (5)	27 °C	3
Vhs.ESR-DL (51)	26 °C	3
Vhs.EEX DK7, Schottky a (52)	21 °C	3
Vhs.IZ500 (53)	23 °C	3
Vhs.V 350 (54)	22 °C	3
Vhs.EEX DK7, Schottky b (55)	23 °C	3
Vhs.TI800 (56)	21 °C	3
Vhs.V 350 (57)	22 °C	3
Vhs.Gateventil Target→Nord (59)	19 °C	3
Vhs.IZ 500 b (6)	23 °C	3
Vhs.CF250 Komp (7)	24 °C	3
Vhs.EEX DK2 (8)	24 °C	3
Vhs.CF 100 (9)	24 °C	3

Grafana
Bar View of Temperature Query



```
mqtt_string = from(bucket: "VHS")
   |> range(start: -3w)
   |> filter(fn: (r) => r["_measurement"] == "mqtt_string_n")
   |> filter(fn: (r) => r["Process"] == "${Node}")
   |> filter(fn: (r) => r["Actor"] == "Vhs")
   |> filter(fn: (r) => r["Attribute"] == "Descriptions")
   |> filter(fn: (r) => r["Channel"] == "0" or r["Channel"] == "1" or r["Channel"] == "2" or r["Channel"] == "3"
   or r["Channel"] == "4" or r["Channel"] == "5" or r["Channel"] == "6" or r["Channel"] == "7" or r["Channel"] ==
   "8" or r["Channel"] == "9" or r["Channel"] == "11" or r["Channel"] == "12" or r["Channel"] == "13" or
   r["Channel"] == "14" or r["Channel"] == "15" or r["Channel"] == "16" or r["Channel"] == "17" or r["Channel"] ==
   "18" or r["Channel"] == "19" or r["Channel"] == "20" or r["Channel"] == "21" or r["Channel"] == "22" or
   r["Channel"] == "24" or r["Channel"] == "25" or r["Channel"] == "26" or r["Channel"] == "27" or r["Channel"] ==
   "28" or r["Channel"] == "29" or r["Channel"] == "30" or r["Channel"] == "31" or r["Channel"] == "32" or
   r["Channel"] == "33" or r["Channel"] == "34" or r["Channel"] == "35" or r["Channel"] == "36" or r["Channel"] ==
   "37" or r["Channel"] == "38" or r["Channel"] == "39" or r["Channel"] == "40" or r["Channel"] == "41" or
   r["Channel"] == "42" or r["Channel"] == "45" or r["Channel"] == "46" or r["Channel"] == "47" or r["Channel"] ==
   "48" or r["Channel"] == "49" or r["Channel"] == "51" or r["Channel"] == "52" or r["Channel"] == "53" or
   r["Channel"] == "54" or r["Channel"] == "55" or r["Channel"] == "56" or r["Channel"] == "57" or r["Channel"] ==
   "59")
   |> last(column: "_value")

mqtt_float = from(bucket: "VHS")
   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
   |> filter(fn: (r) => r["_measurement"] == "mqtt_float_n" or r["_measurement"] == "mqtt_integer_n")
   |> filter(fn: (r) => r["Process"] == "${Node}")
   |> filter(fn: (r) => r["Actor"] == "Vhs")
   |> filter(fn: (r) => r["Attribute"] == "Temperatures")
   |> filter(fn: (r) => r["Channel"] == "0" or r["Channel"] == "1" or r["Channel"] == "2" or r["Channel"] == "3"
   or r["Channel"] == "4" or r["Channel"] == "5" or r["Channel"] == "6" or r["Channel"] == "7" or r["Channel"] ==
   "8" or r["Channel"] == "9" or r["Channel"] == "11" or r["Channel"] == "12" or r["Channel"] == "13" or
   r["Channel"] == "14" or r["Channel"] == "15" or r["Channel"] == "16" or r["Channel"] == "17" or r["Channel"] ==
   "18" or r["Channel"] == "19" or r["Channel"] == "20" or r["Channel"] == "21" or r["Channel"] == "22" or
   r["Channel"] == "24" or r["Channel"] == "25" or r["Channel"] == "26" or r["Channel"] == "27" or r["Channel"] ==
   "28" or r["Channel"] == "29" or r["Channel"] == "30" or r["Channel"] == "31" or r["Channel"] == "32" or
   r["Channel"] == "33" or r["Channel"] == "34" or r["Channel"] == "35" or r["Channel"] == "36" or r["Channel"] ==
   "37" or r["Channel"] == "38" or r["Channel"] == "39" or r["Channel"] == "40" or r["Channel"] == "41" or
   r["Channel"] == "42" or r["Channel"] == "45" or r["Channel"] == "46" or r["Channel"] == "47" or r["Channel"] ==
   "48" or r["Channel"] == "49" or r["Channel"] == "51" or r["Channel"] == "52" or r["Channel"] == "53" or
   r["Channel"] == "54" or r["Channel"] == "55" or r["Channel"] == "56" or r["Channel"] == "57" or r["Channel"] ==
   "59")
   |> aggregateWindow(every: v.windowPeriod, fn: last, createEmpty: false)

join(tables: {n: mqtt_float, s: mqtt_string}, on: ["Channel"])
  |> map(fn: (r) => ({_time: r._time_n, _value: r._value_n, _field: "value", description: r._value_s, Channel:
  r.Channel, Process: r.Process_n, host: r.host_n, topic: r.topic_n, Actor: r.Actor_n, Attribute: r.Attribute_n}))
  |> group(columns:["description","Channel","Process","host","topic","Actor","Attribute","_start","_stop"])
  |> yield()
```



Override 1

Fields returned by query

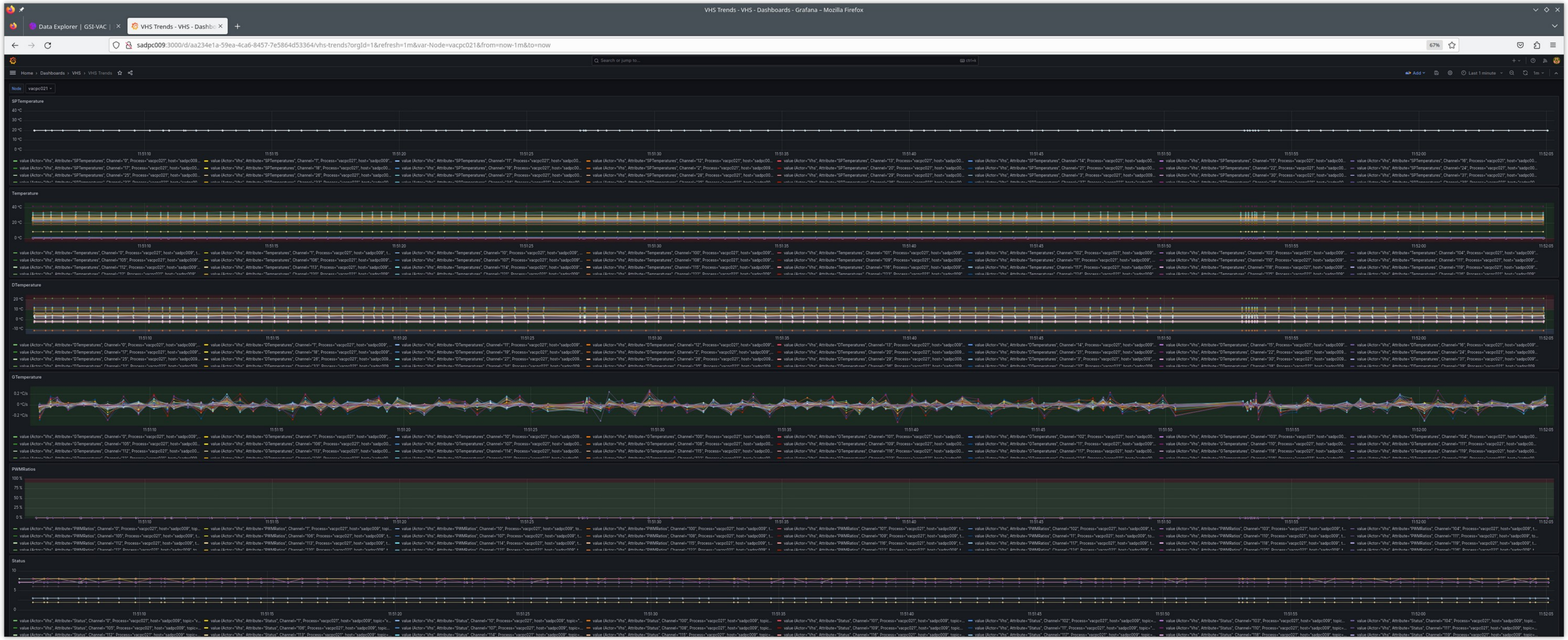
Query: A

Standard options › Display name

Change the field or series name

${__field.labels.Actor}.${__field.labels.description}

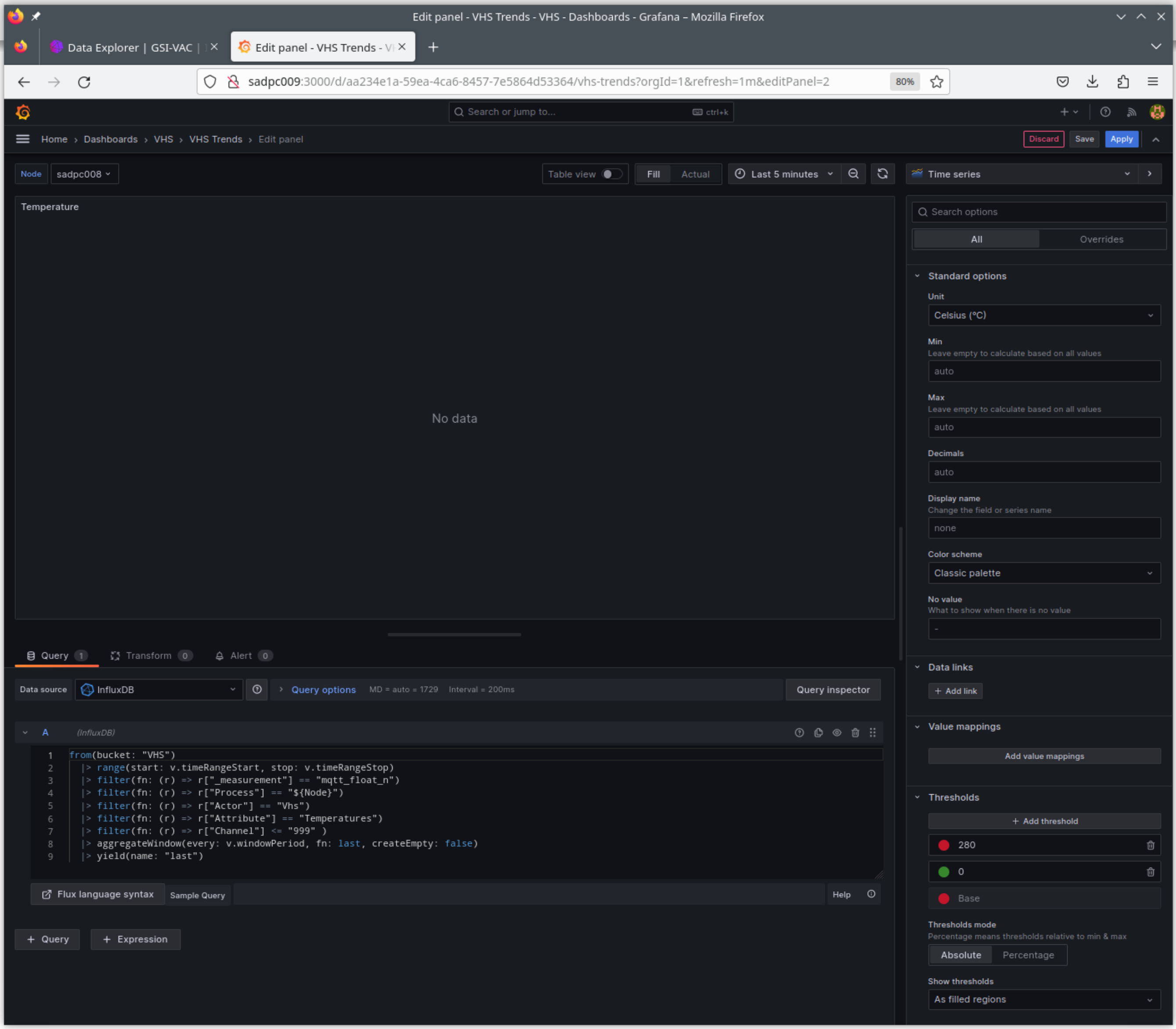
Grafana
Trend View


VHS Trends - VHS - Dashboards - Grafana – Mozilla Firefox
Data Explorer | GSI-VAC
VHS Trends - VHS - Dashbo
Home › Dashboards › VHS › VHS Trends
Node
vacpc021
SPTemperature
Temperature
DTemperature
GTemperature
PWMRatios
Status

sadpc009:3000/d/aa234e1a-59ea-4ca6-8457-7e5864d53364/vhs-trends?orgId=1&refresh=1m&var-Node=vacpc021&from=now-1m&to=now

```
value {Actor="Vhs", Attribute="SPTemperatures", Channel="0", Process="vacpc021", host="sadpc009...
value {Actor="Vhs", Attribute="SPTemperatures", Channel="1", Process="vacpc021", host="sadpc009"...
value {Actor="Vhs", Attribute="SPTemperatures", Channel="17", Process="vacpc021", host="sadpc00...
value {Actor="Vhs", Attribute="SPTemperatures", Channel="18", Process="vacpc021", host="sadpc00...
value {Actor="Vhs", Attribute="SPTemperatures", Channel="25", Process="vacpc021", host="sadpc00...
value {Actor="Vhs", Attribute="SPTemperatures", Channel="26", Process="vacpc021", host="sadpc00...
```

Grafana
Trend View Configuration



Node: sadpc008

Data source: InfluxDB — MD = auto = 1729 — Interval = 200ms

A (InfluxDB)

```
from(bucket: "VHS")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "mqtt_float_n")
  |> filter(fn: (r) => r["Process"] == "${Node}")
  |> filter(fn: (r) => r["Actor"] == "Vhs")
  |> filter(fn: (r) => r["Attribute"] == "Temperatures")
  |> filter(fn: (r) => r["Channel"] <= "999" )
  |> aggregateWindow(every: v.windowPeriod, fn: last, createEmpty: false)
  |> yield(name: "last")
```

Standard options

Unit: Celsius (°C)

Min — Leave empty to calculate based on all values: auto

Max — Leave empty to calculate based on all values: auto

Decimals: auto

Display name — Change the field or series name: none

Color scheme: Classic palette

No value — What to show when there is no value: -

Data links

Value mappings

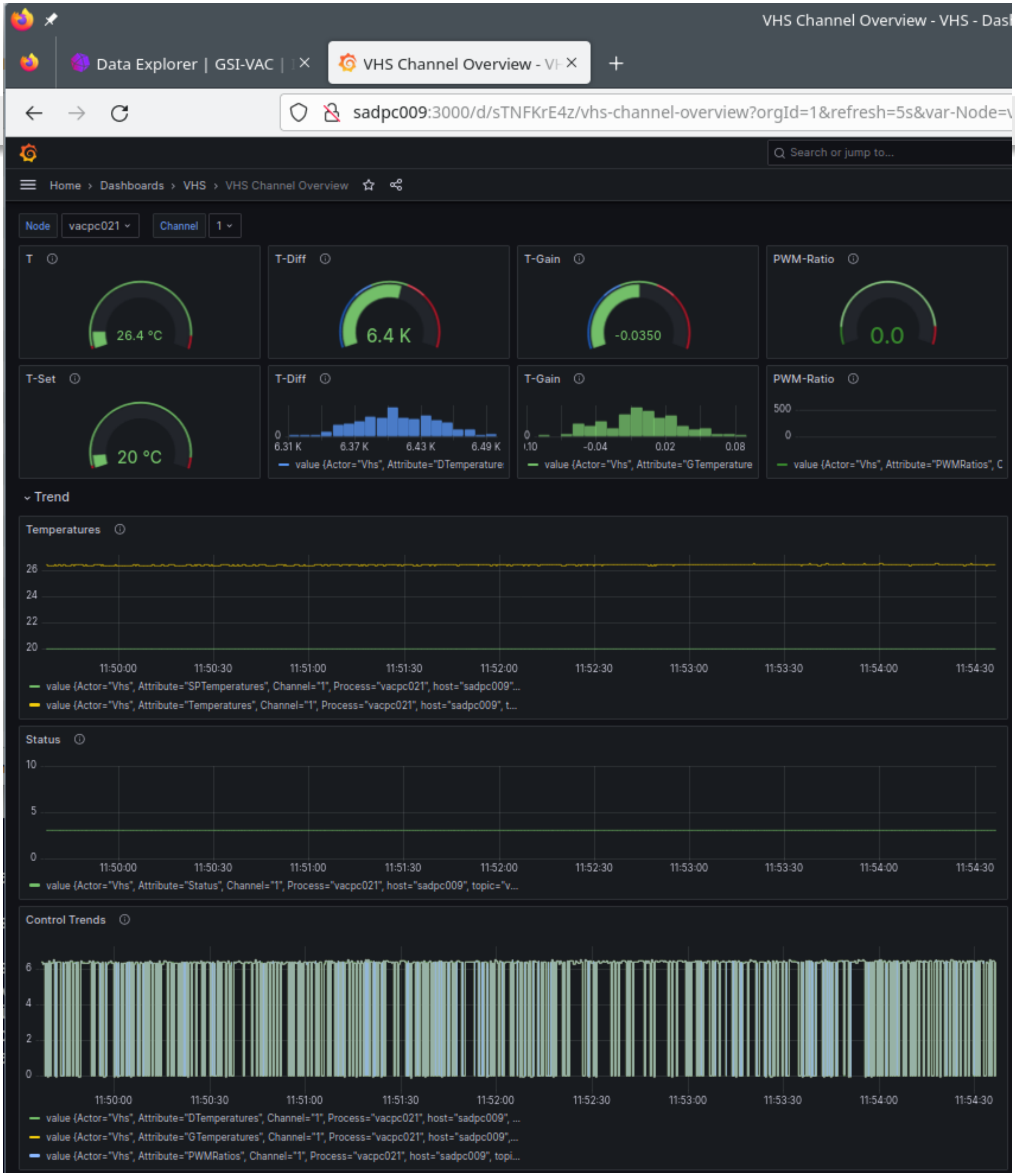
Thresholds

- 🟥 280
- 🟩 0
- 🟥 Base

Thresholds mode — Percentage means thresholds relative to min & max: [✓ Absolute] Percentage

Show thresholds: As filled regions

Grafana Channel View



sadpc009:3000/d/sTNFKrE4z/vhs-channel-overview?orgId=1&refresh=5s&var-Node=v

— value {Actor="Vhs", Attribute="DTemperature

— value {Actor="Vhs", Attribute="GTemperature

— value {Actor="Vhs", Attribute="PWMRatios", C

— value {Actor="Vhs", Attribute="SPTemperatures", Channel="1", Process="vacpc021", host="sadpc009"...

— value {Actor="Vhs", Attribute="Temperatures", Channel="1", Process="vacpc021", host="sadpc009", t...

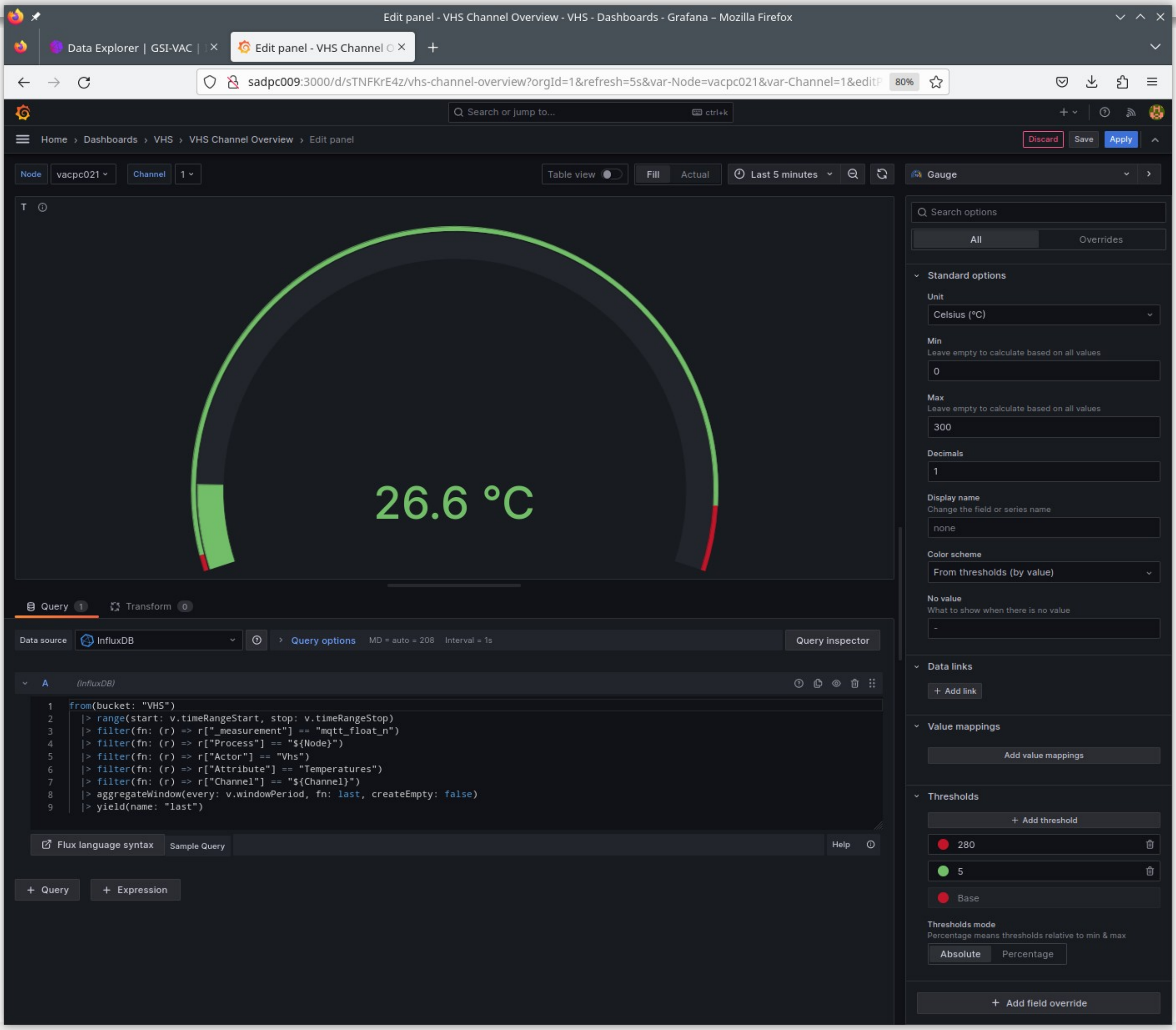
— value {Actor="Vhs", Attribute="Status", Channel="1", Process="vacpc021", host="sadpc009", topic="v...

— value {Actor="Vhs", Attribute="DTemperatures", Channel="1", Process="vacpc021", host="sadpc009", ...

— value {Actor="Vhs", Attribute="GTemperatures", Channel="1", Process="vacpc021", host="sadpc009",...

— value {Actor="Vhs", Attribute="PWMRatios", Channel="1", Process="vacpc021", host="sadpc009", topi...

Grafana
Channel View Configuration



Edit panel - VHS Channel Overview - VHS - Dashboards - Grafana – Mozilla Firefox

sadpc009:3000/d/sTNFKrE4z/vhs-channel-overview?orgId=1&refresh=5s&var-Node=vacpc021&var-Channel=1&edit

Home › Dashboards › VHS › VHS Channel Overview › Edit panel

Node vacpc021 · Channel 1 · Last 5 minutes · Gauge

26.6 °C

Data source: InfluxDB · Query options: MD = auto = 208, Interval = 1s

```
from(bucket: "VHS")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "mqtt_float_n")
  |> filter(fn: (r) => r["Process"] == "${Node}")
  |> filter(fn: (r) => r["Actor"] == "Vhs")
  |> filter(fn: (r) => r["Attribute"] == "Temperatures")
  |> filter(fn: (r) => r["Channel"] == "${Channel}")
  |> aggregateWindow(every: v.windowPeriod, fn: last, createEmpty: false)
  |> yield(name: "last")
```

Standard options
- Unit: Celsius (°C)
- Min (Leave empty to calculate based on all values): 0
- Max (Leave empty to calculate based on all values): 300
- Decimals: 1
- Display name (Change the field or series name): none
- Color scheme: From thresholds (by value)
- No value (What to show when there is no value): -

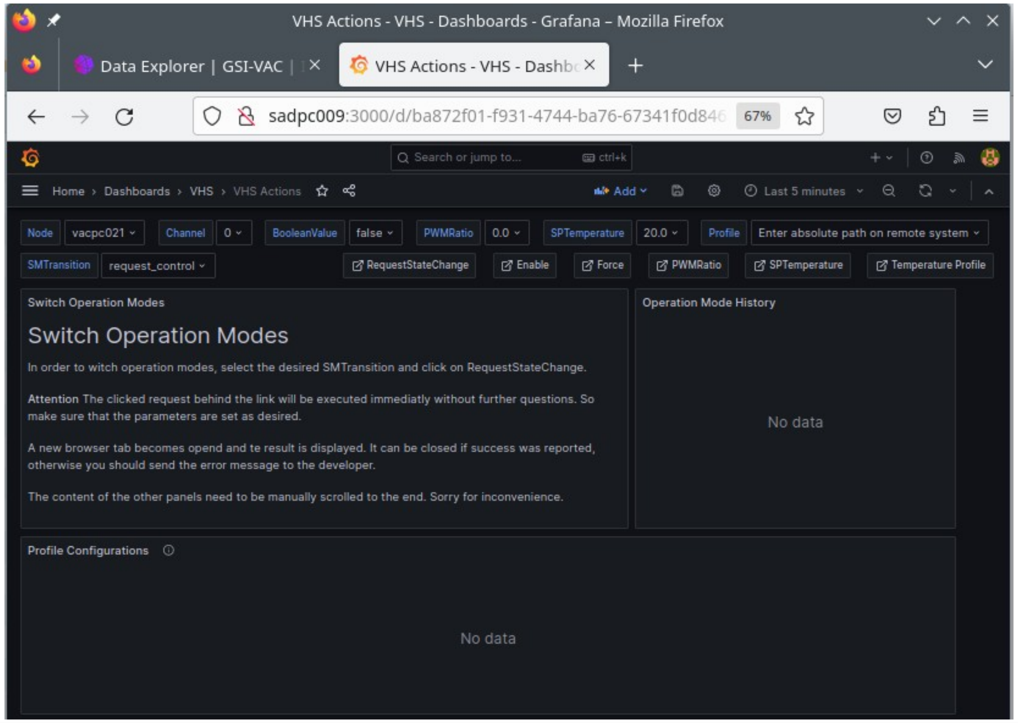
Thresholds
- 🟥 280
- 🟩 5
- 🟥 Base

Thresholds mode (Percentage means thresholds relative to min & max): Absolute

Grafana
Actions

- HTTP Request handled by
  - vhs_http_server.py
    - from http.server import BaseHTTPRequestHandler, HTTPServer
    - import json
    - import paho.mqtt.publish as publish

1) Parse json parameter string
2) Publish to MQTT Topic



Switch Operation Modes

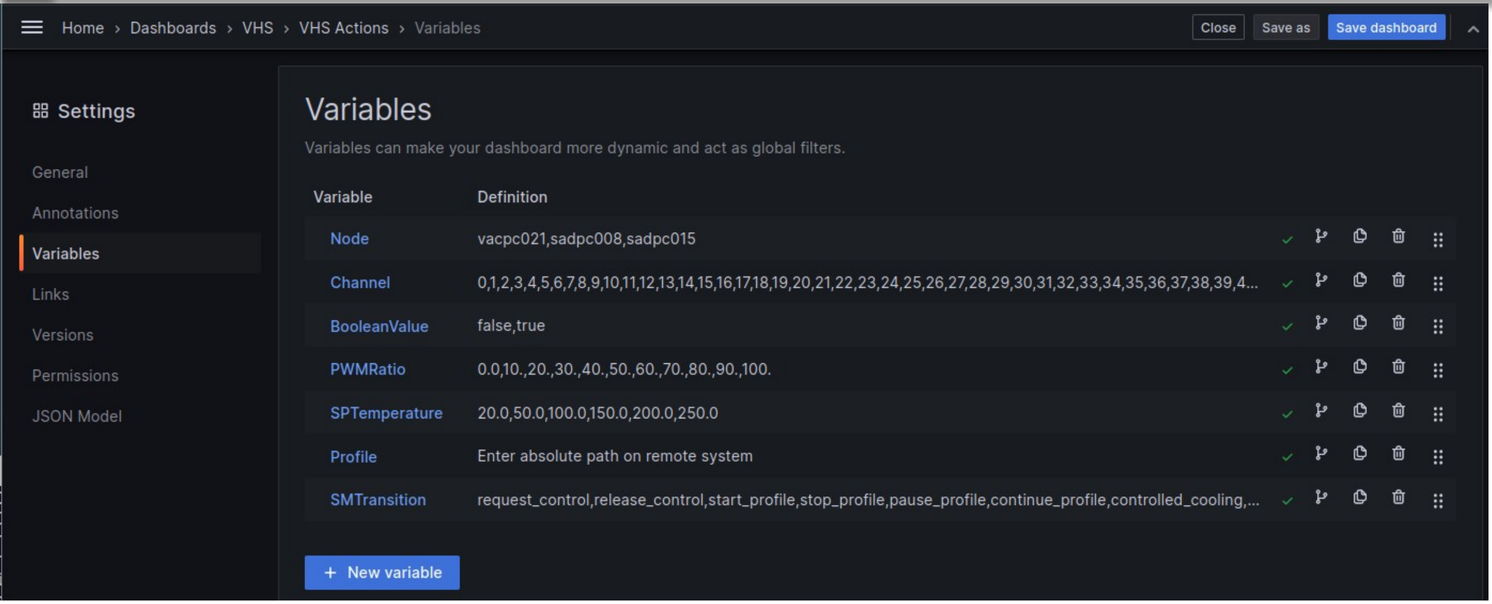
In order to witch operation modes, select the desired SMTransition and click on RequestStateChange.

Attention The clicked request behind the link will be executed immediatly without further questions. So make sure that the parameters are set as desired.

A new browser tab becomes opend and te result is displayed. It can be closed if success was reported, otherwise you should send the error message to the developer.

The content of the other panels need to be manually scrolled to the end. Sorry for inconvenience.

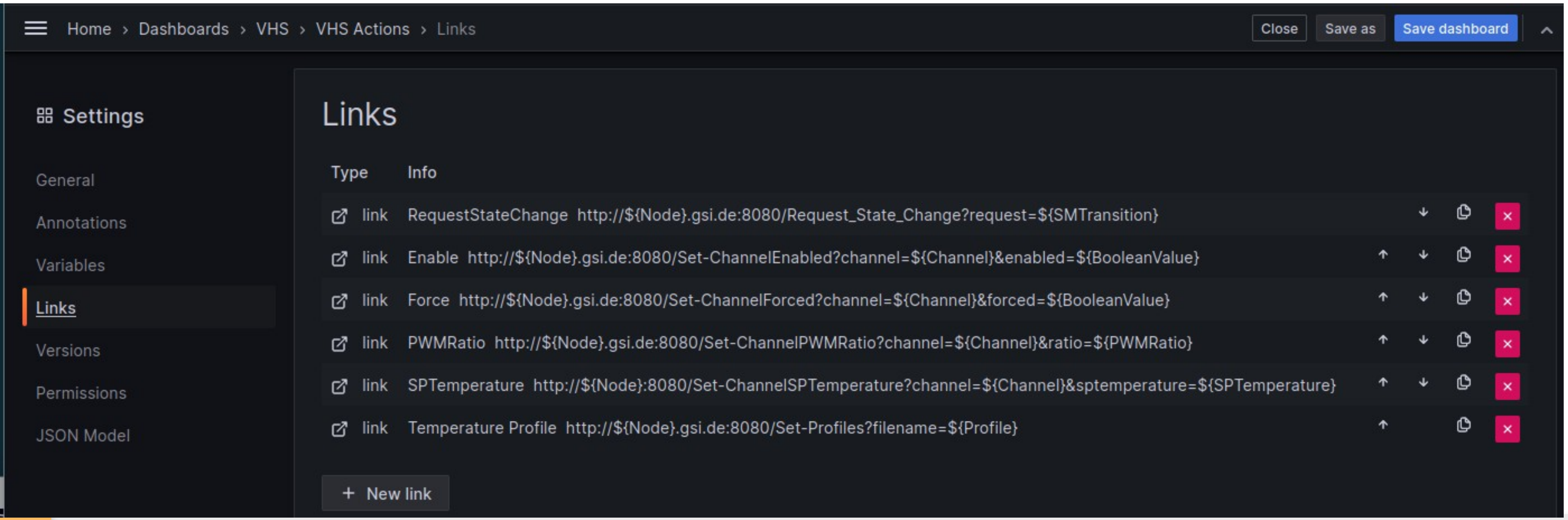
Grafana
Actions



Variables

Variables can make your dashboard more dynamic and act as global filters.

Variable	Definition
Node	vacpc021,sadpc008,sadpc015
Channel	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,4...
BooleanValue	false,true
PWMRatio	0.0,10.,20.,30.,40.,50.,60.,70.,80.,90.,100.
SPTemperature	20.0,50.0,100.0,150.0,200.0,250.0
Profile	Enter absolute path on remote system
SMTransition	request_control,release_control,start_profile,stop_profile,pause_profile,continue_profile,controlled_cooling,...



Links

Type	Info
link	RequestStateChange http://${Node}.gsi.de:8080/Request_State_Change?request=${SMTransition}
link	Enable http://${Node}.gsi.de:8080/Set-ChannelEnabled?channel=${Channel}&enabled=${BooleanValue}
link	Force http://${Node}.gsi.de:8080/Set-ChannelForced?channel=${Channel}&forced=${BooleanValue}
link	PWMRatio http://${Node}.gsi.de:8080/Set-ChannelPWMRatio?channel=${Channel}&ratio=${PWMRatio}
link	SPTemperature http://${Node}:8080/Set-ChannelSPTemperature?channel=${Channel}&sptemperature=${SPTemperature}
link	Temperature Profile http://${Node}.gsi.de:8080/Set-Profiles?filename=${Profile}

Mosquitto MQTT Broker
https://mosquitto.org/

- Eclipse Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers.
- The MQTT protocol provides a lightweight method of carrying out messaging using a publish/subscribe model. This makes it suitable for Internet of Things messaging such as with low power sensors or mobile devices such as phones, embedded computers or microcontrollers.
- The Mosquitto project also provides a C library for implementing MQTT clients, and the very popular mosquitto_pub and mosquitto_sub command line MQTT clients.
- Python: https://pypi.org/project/paho-mqtt/
- Support for Linux, Raspian, Windows

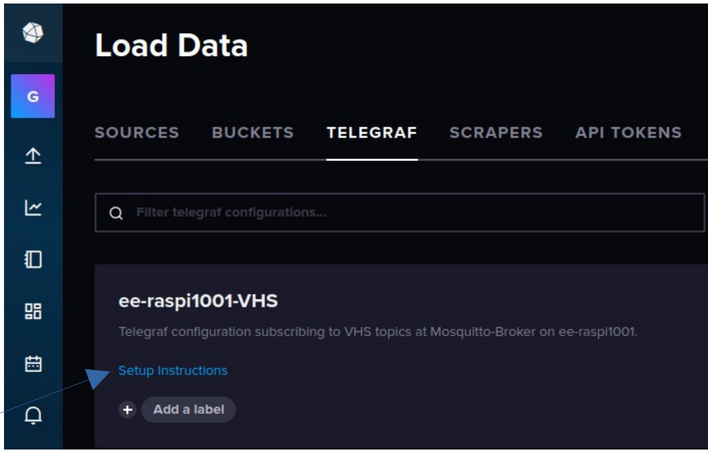
Mosquitto MQTT Broker
Installation auf Linux

- apt/zypper search mosquitto
- sudo apt/zypper install mosquitto
- Edit */etc/mosquitto/mosquitto.conf*
  - listener 1883 0.0.0.0
  - allow_anonymous true ==Needs to be configured for user/password instead.==
- systemctl (re)start mosquitto.service
- Start at reboot
  - systemctl enable mosquitto.service

- MQTT-Explorer
  - Download at: http://mqtt-explorer.com/

InfluxDB
Installation on openSuse Tumbleweed

- OpenSuse: zypper ar
  - https://download.opensuse.org/repositories/server:/database/openSUSE_Tumbleweed/ influxdb2
- sudo apt/zypper install influxdb2
- systemctl (re)start influxdb.service
- systemctl enable influxdb.service
  - Browser URL: `localhost:8086
  - Set password on first call
- Set Organisation
  - Add more users by using the admin-token
  - influx user create -n *username* -p *password* -o *organization*
- Telegraf plugin
  - Follow Setup Instructions
  - You need to set a valid token for authentication, explict or as environment variable.



ee-raspi1001-VHS

Telegraf configuration subscribing to VHS topics at Mosquitto-Broker on ee-raspi1001.

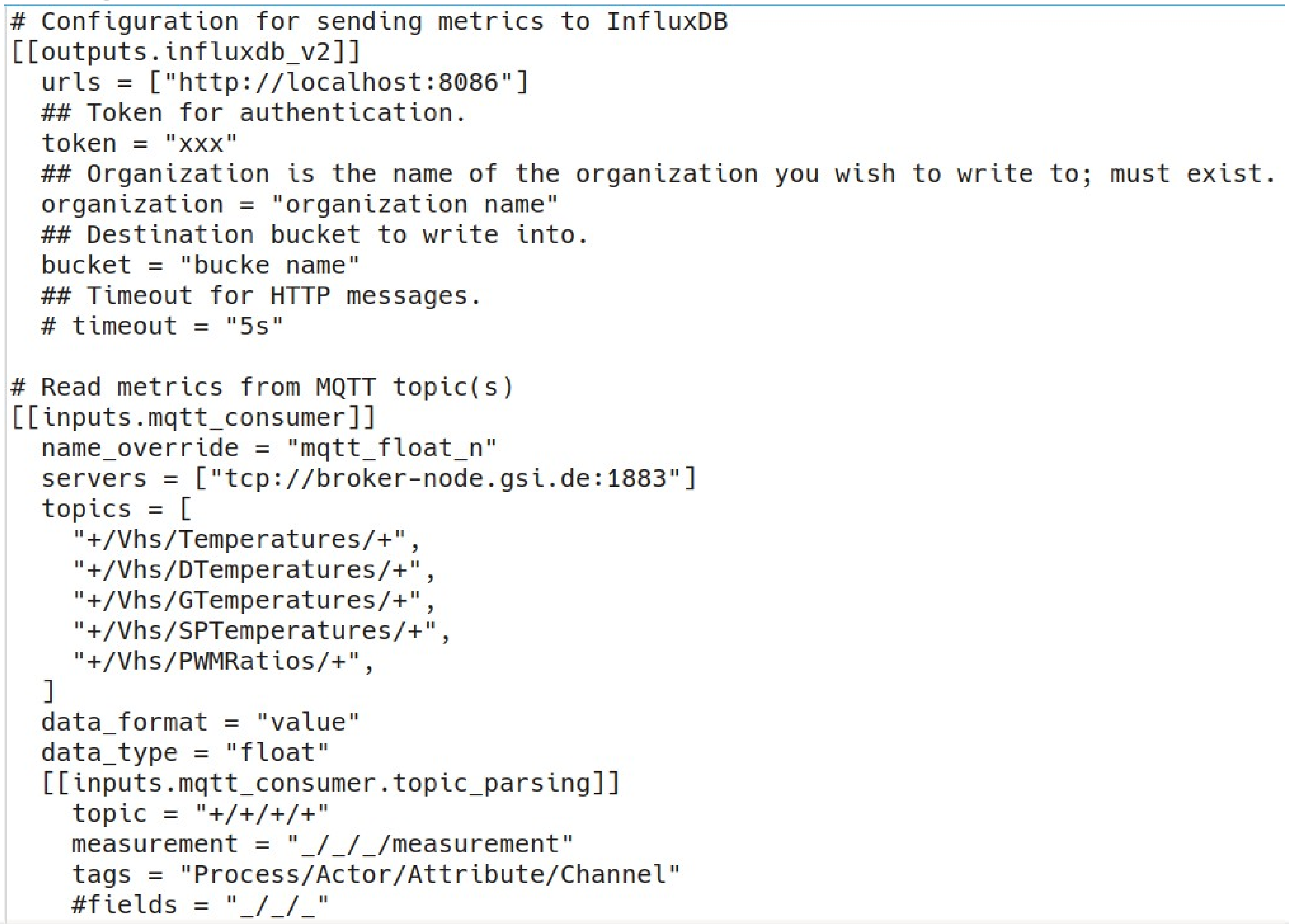
Telegraf
https://www.influxdata.com/time-series-platform/telegraf/

- Telegraf is the open source server agent to help you collect metrics from your stacks, sensors, and systems.

- Installation
  - apt/zypper search telegraf
  - sudo apt/zypper install telegraf
  - Start Telegraf with:
    - 1. Pflege in InfluxDB.: telegraf --config http://localhost:8086/api/v2/telegrafs/xxx
    - 2. Pflege in einem User-Verzeichnis: telegraf --config /home/brand/Downloads/telegraf.conf
    - 3. Pflege als admin: telegraf --config /etc/telegraf/telegraf.conf
    - 4. Start as service:
      - 1. sudo cp telegraf.conf /etc/telegraf
      - 2. sudo cp /usr/lib/systemd/system/telegraf.service /usr/lib/systemd/system/telegraf.service.org
      - 3. Edit telegraf.service and replace config file to be used.
        - ExecStart=/usr/bin/telegraf -config /etc/telegraf/telegraf.conf -config-directory /etc/telegraf/telegraf.d $TELEGRAF_OPTS
      - 4. systemctl restart telegraf.service

Telegraf
Configuration

- telegraf.conf

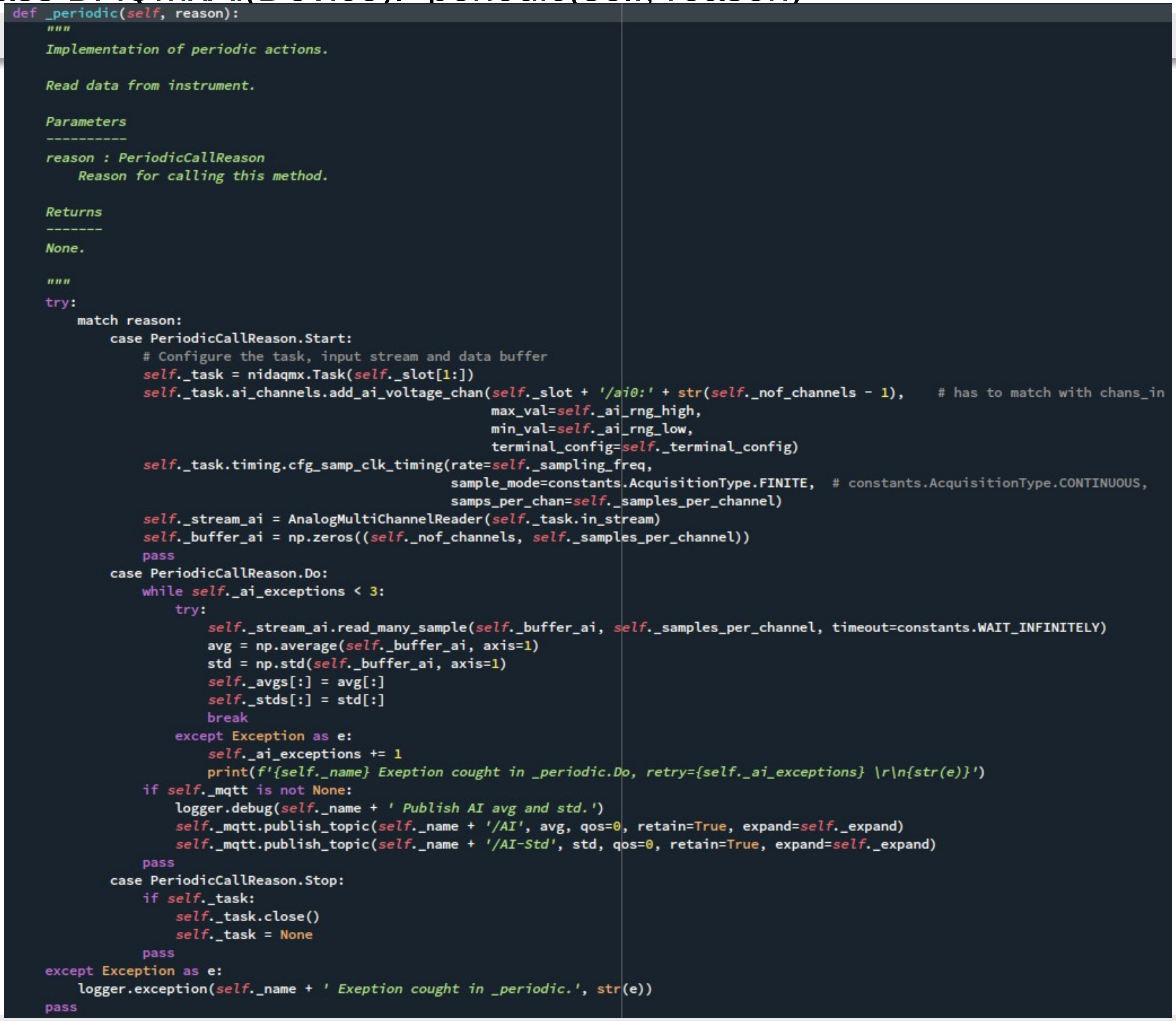


```
# Configuration for sending metrics to InfluxDB
[[outputs.influxdb_v2]]
  urls = ["http://localhost:8086"]
  ## Token for authentication.
  token = "xxx"
  ## Organization is the name of the organization you wish to write to; must exist.
  organization = "organization name"
  ## Destination bucket to write into.
  bucket = "bucke name"
  ## Timeout for HTTP messages.
  # timeout = "5s"

# Read metrics from MQTT topic(s)
[[inputs.mqtt_consumer]]
  name_override = "mqtt_float_n"
  servers = ["tcp://broker-node.gsi.de:1883"]
  topics = [
    "+/Vhs/Temperatures/+",
    "+/Vhs/DTemperatures/+",
    "+/Vhs/GTemperatures/+",
    "+/Vhs/SPTemperatures/+",
    "+/Vhs/PWMRatios/+",
  ]
  data_format = "value"
  data_type = "float"
  [[inputs.mqtt_consumer.topic_parsing]]
    topic = "+/+/+/+"
    measurement = "_/_/_/measurement"
    tags = "Process/Actor/Attribute/Channel"
    #fields = "_/_/_"
```

Grafana
https://grafana.com/

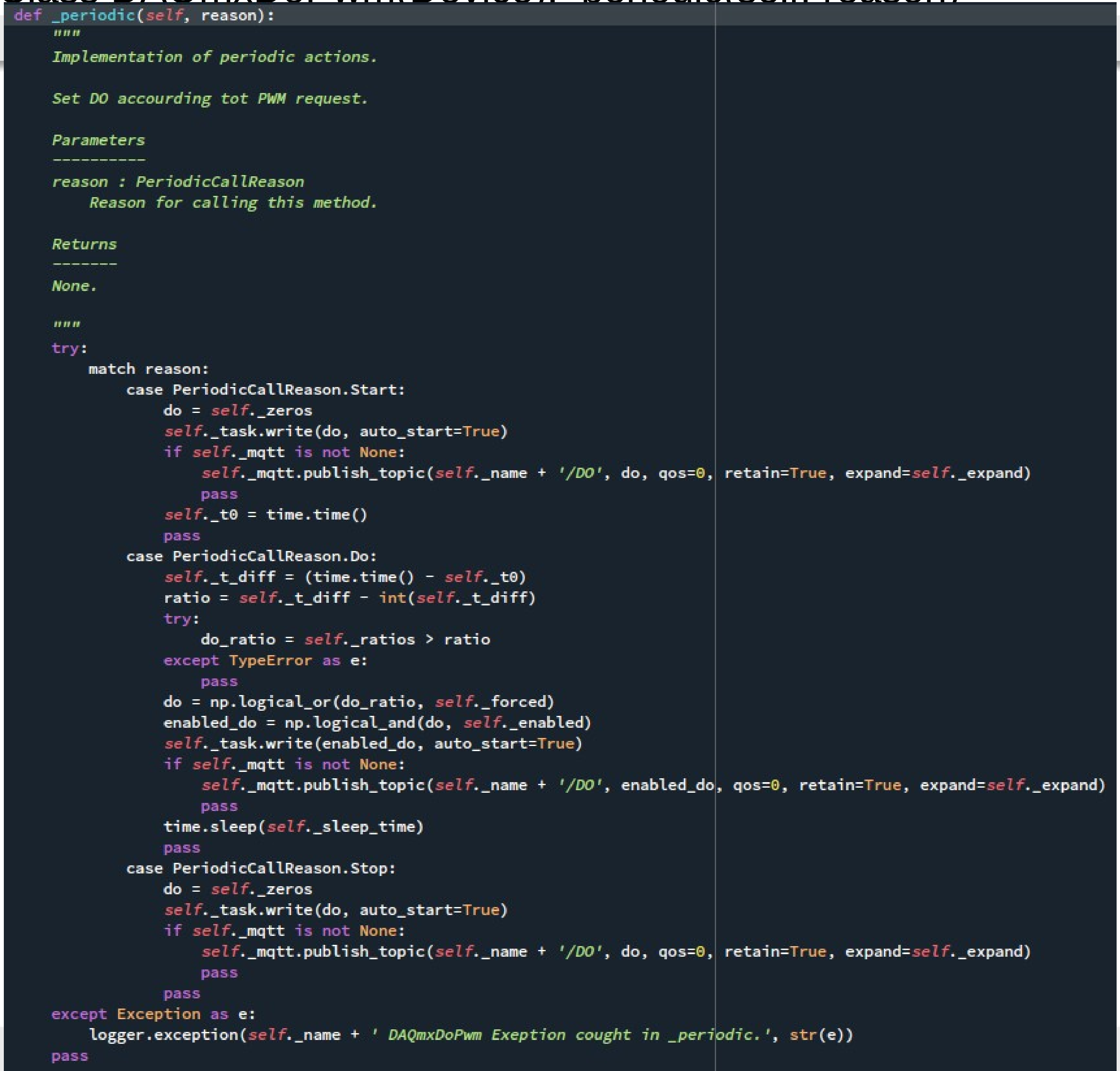
- Installation
  - apt/zypper search grafana
  - sudo zypper install grafana
  - systemctl (re)start grafana.service
  - Start at reboot
    - systemctl enable grafana.service
  - Browser URL: http://localhost:3000
    - Set password on first call.
    - Add more Users
  - Add data source: InfluxDB
    - Query Language: Flux
    - HTTP URL: http://localhost:8086
    - Auth: Basic auth
      - *User/Password*
    - InfluxDB Details
      - Organization: *organization*
      - Token: *InfluxDB generated API token*
    - Default: Bucket: *bucket name*

PyAcdaq
Class DAQmxAi(Device)._periodic(self, reason)



```python
def _periodic(self, reason):
    """
    Implementation of periodic actions.

    Read data from instrument.

    Parameters
    ----------
    reason : PeriodicCallReason
        Reason for calling this method.

    Returns
    -------
    None.

    """
    try:
        match reason:
            case PeriodicCallReason.Start:
                # Configure the task, input stream and data buffer
                self._task = nidaqmx.Task(self._slot[1:])
                self._task.ai_channels.add_ai_voltage_chan(self._slot + '/ai0:' + str(self._nof_channels - 1),    # has to match with chans_in
                                                           max_val=self._ai_rng_high,
                                                           min_val=self._ai_rng_low,
                                                           terminal_config=self._terminal_config)
                self._task.timing.cfg_samp_clk_timing(rate=self._sampling_freq,
                                                      sample_mode=constants.AcquisitionType.FINITE,  # constants.AcquisitionType.CONTINUOUS,
                                                      samps_per_chan=self._samples_per_channel)
                self._stream_ai = AnalogMultiChannelReader(self._task.in_stream)
                self._buffer_ai = np.zeros((self._nof_channels, self._samples_per_channel))
                pass
            case PeriodicCallReason.Do:
                while self._ai_exceptions < 3:
                    try:
                        self._stream_ai.read_many_sample(self._buffer_ai, self._samples_per_channel, timeout=constants.WAIT_INFINITELY)
                        avg = np.average(self._buffer_ai, axis=1)
                        std = np.std(self._buffer_ai, axis=1)
                        self._avgs[:] = avg[:]
                        self._stds[:] = std[:]
                        break
                    except Exception as e:
                        self._ai_exceptions += 1
                        print(f'{self._name} Exeption cought in _periodic.Do, retry={self._ai_exceptions} \r\n{str(e)}')
                if self._mqtt is not None:
                    logger.debug(self._name + ' Publish AI avg and std.')
                    self._mqtt.publish_topic(self._name + '/AI', avg, qos=0, retain=True, expand=self._expand)
                    self._mqtt.publish_topic(self._name + '/AI-Std', std, qos=0, retain=True, expand=self._expand)
                pass
            case PeriodicCallReason.Stop:
                if self._task:
                    self._task.close()
                    self._task = None
                pass
    except Exception as e:
        logger.exception(self._name + ' Exeption cought in _periodic.', str(e))
    pass
```

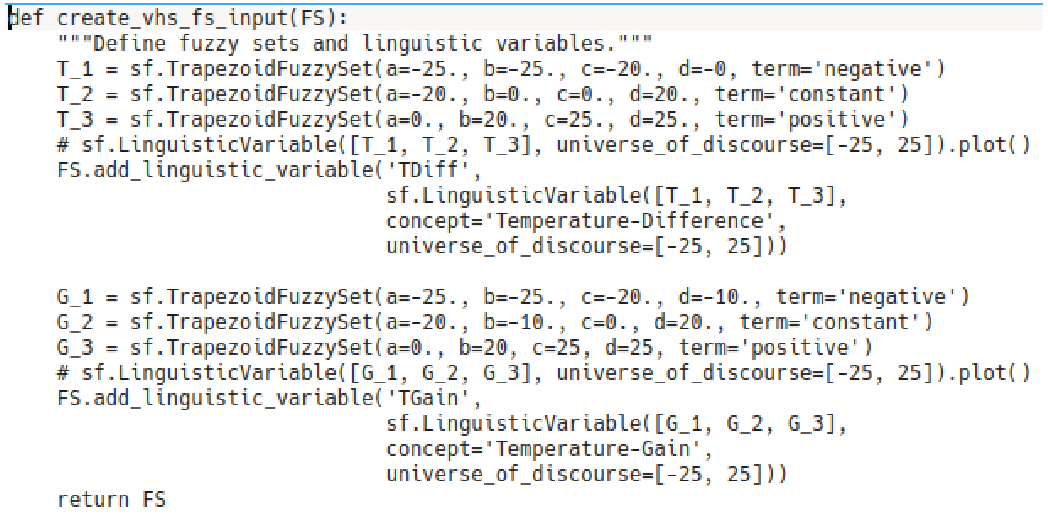
PyAcdaq
Class DAQmxDoPwm(Device), _periodic(self, reason)



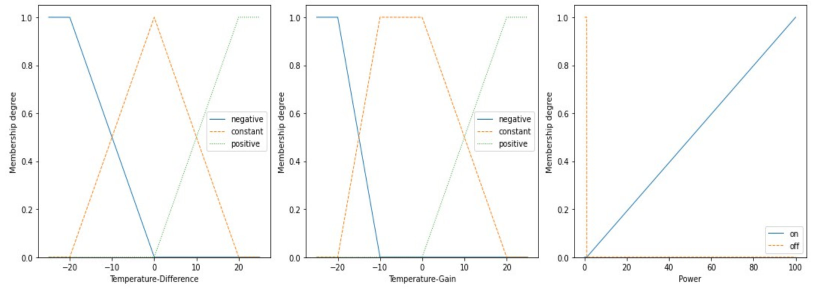
```
def _periodic(self, reason):
    """
    Implementation of periodic actions.

    Set DO accourding tot PWM request.

    Parameters
    ----------
    reason : PeriodicCallReason
        Reason for calling this method.

    Returns
    -------
    None.

    """
    try:
        match reason:
            case PeriodicCallReason.Start:
                do = self._zeros
                self._task.write(do, auto_start=True)
                if self._mqtt is not None:
                    self._mqtt.publish_topic(self._name + '/DO', do, qos=0, retain=True, expand=self._expand)
                    pass
                self._t0 = time.time()
                pass
            case PeriodicCallReason.Do:
                self._t_diff = (time.time() - self._t0)
                ratio = self._t_diff - int(self._t_diff)
                try:
                    do_ratio = self._ratios > ratio
                except TypeError as e:
                    pass
                do = np.logical_or(do_ratio, self._forced)
                enabled_do = np.logical_and(do, self._enabled)
                self._task.write(enabled_do, auto_start=True)
                if self._mqtt is not None:
                    self._mqtt.publish_topic(self._name + '/DO', enabled_do, qos=0, retain=True, expand=self._expand)
                    pass
                time.sleep(self._sleep_time)
                pass
            case PeriodicCallReason.Stop:
                do = self._zeros
                self._task.write(do, auto_start=True)
                if self._mqtt is not None:
                    self._mqtt.publish_topic(self._name + '/DO', do, qos=0, retain=True, expand=self._expand)
                    pass
                pass
    except Exception as e:
        logger.exception(self._name + ' DAQmxDoPwm Exeption cought in _periodic.', str(e))
    pass
```

Fuzzy System with Simpful
https://pypi.org/project/simpful/

Linguistic input variables and ranges:



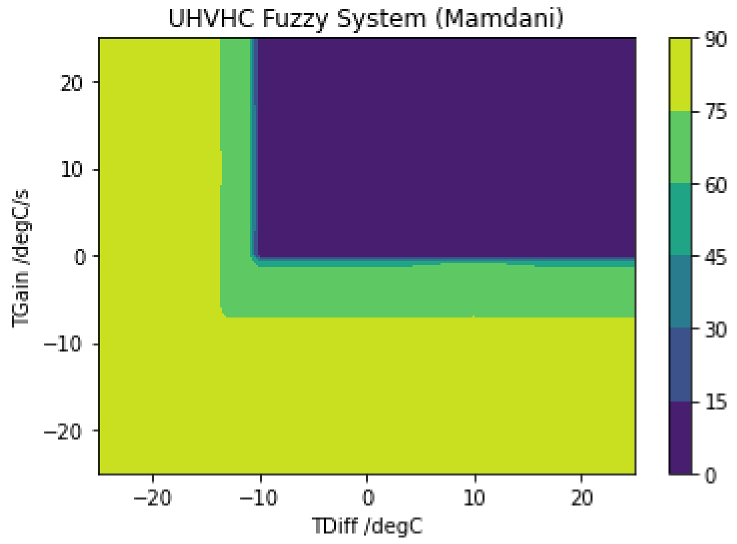
```
def create_vhs_fs_input(FS):
    """Define fuzzy sets and linguistic variables."""
    T_1 = sf.TrapezoidFuzzySet(a=-25., b=-25., c=-20., d=-0, term='negative')
    T_2 = sf.TrapezoidFuzzySet(a=-20., b=0., c=0., d=20., term='constant')
    T_3 = sf.TrapezoidFuzzySet(a=0., b=20., c=25., d=25., term='positive')
    # sf.LinguisticVariable([T_1, T_2, T_3], universe_of_discourse=[-25, 25]).plot()
    FS.add_linguistic_variable('TDiff',
                               sf.LinguisticVariable([T_1, T_2, T_3],
                               concept='Temperature-Difference',
                               universe_of_discourse=[-25, 25]))

    G_1 = sf.TrapezoidFuzzySet(a=-25., b=-25., c=-20., d=-10., term='negative')
    G_2 = sf.TrapezoidFuzzySet(a=-20., b=-10., c=0., d=20., term='constant')
    G_3 = sf.TrapezoidFuzzySet(a=0., b=20, c=25, d=25, term='positive')
    # sf.LinguisticVariable([G_1, G_2, G_3], universe_of_discourse=[-25, 25]).plot()
    FS.add_linguistic_variable('TGain',
                               sf.LinguisticVariable([G_1, G_2, G_3],
                               concept='Temperature-Gain',
                               universe_of_discourse=[-25, 25]))

    return FS
```



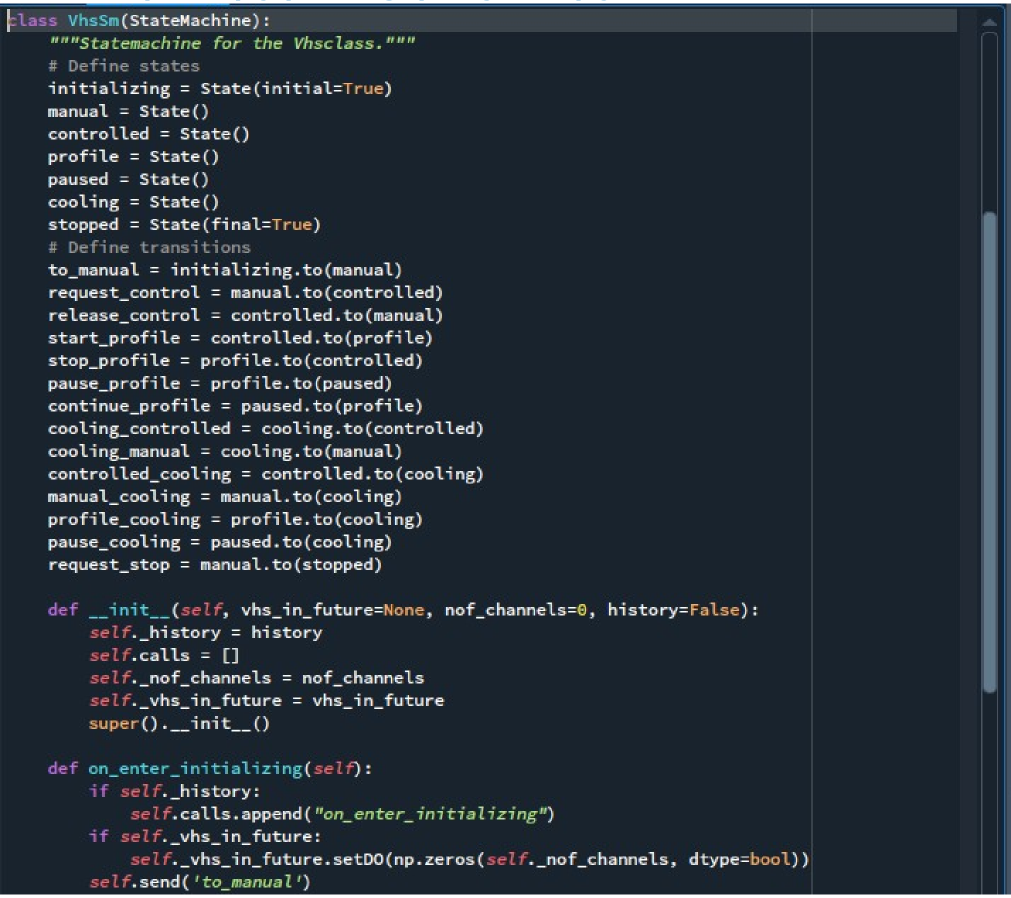
Set of Rules

```
R0 = 'IF (TDiff IS constant) AND (TGain IS negative) THEN (Power IS on) WEIGHT 1.0'
R1 = 'IF (TDiff IS constant) AND (TGain IS constant) THEN (Power IS off WEIGHT 1.0'
R2 = 'IF (TDiff IS constant) AND (TGain IS positive) THEN (Power IS off) WEIGHT 1.0'
R3 = 'IF (TDiff IS positive) AND (TGain IS negative) THEN (Power IS on) WEIGHT 1.0'
R4 = 'IF (TDiff IS positive) AND (TGain IS constant) THEN (Power IS off) WEIGHT 1.0'
R5 = 'IF (TDiff IS positive) AND (TGain IS positive) THEN (Power IS off) WEIGHT 1.0'
R6 = 'IF (TDiff IS negative) AND (TGain IS negative) THEN (Power IS on) WEIGHT 1.0'
R7 = 'IF (TDiff IS negative) AND (TGain IS constant) THEN (Power IS on) WEIGHT 1.0'
R8 = 'IF (TDiff IS negative) AND (TGain IS positive) THEN (Power IS on) WEIGHT 1.0'
R = [R0, R1, R2, R3, R4, R5, R6, R7, R8]
```

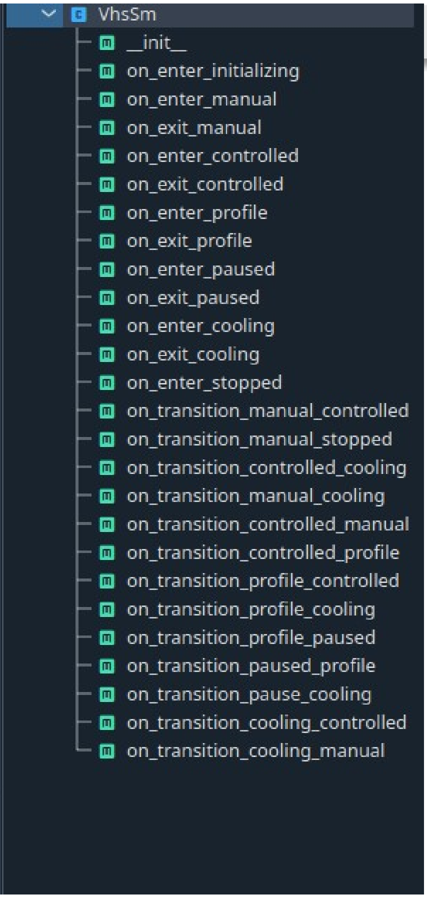
```
The resulting power is scaled
case 'mamdani':
    p = self._fs.Mamdani_inference(["Power"])["Power"]
    if p < 0.:
        p = 0.
    else:
        p *= 1.3
    pass
```



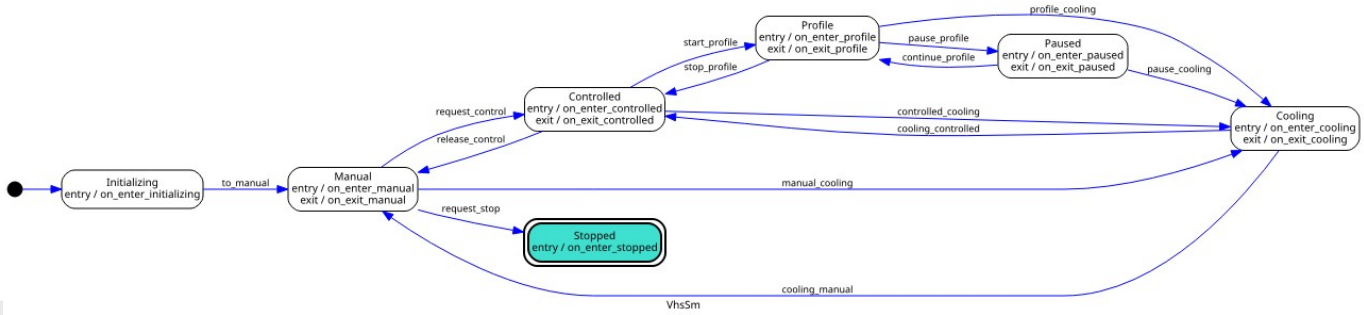
VHS StateMachine
https://pypi.org/project/python-statemachine/



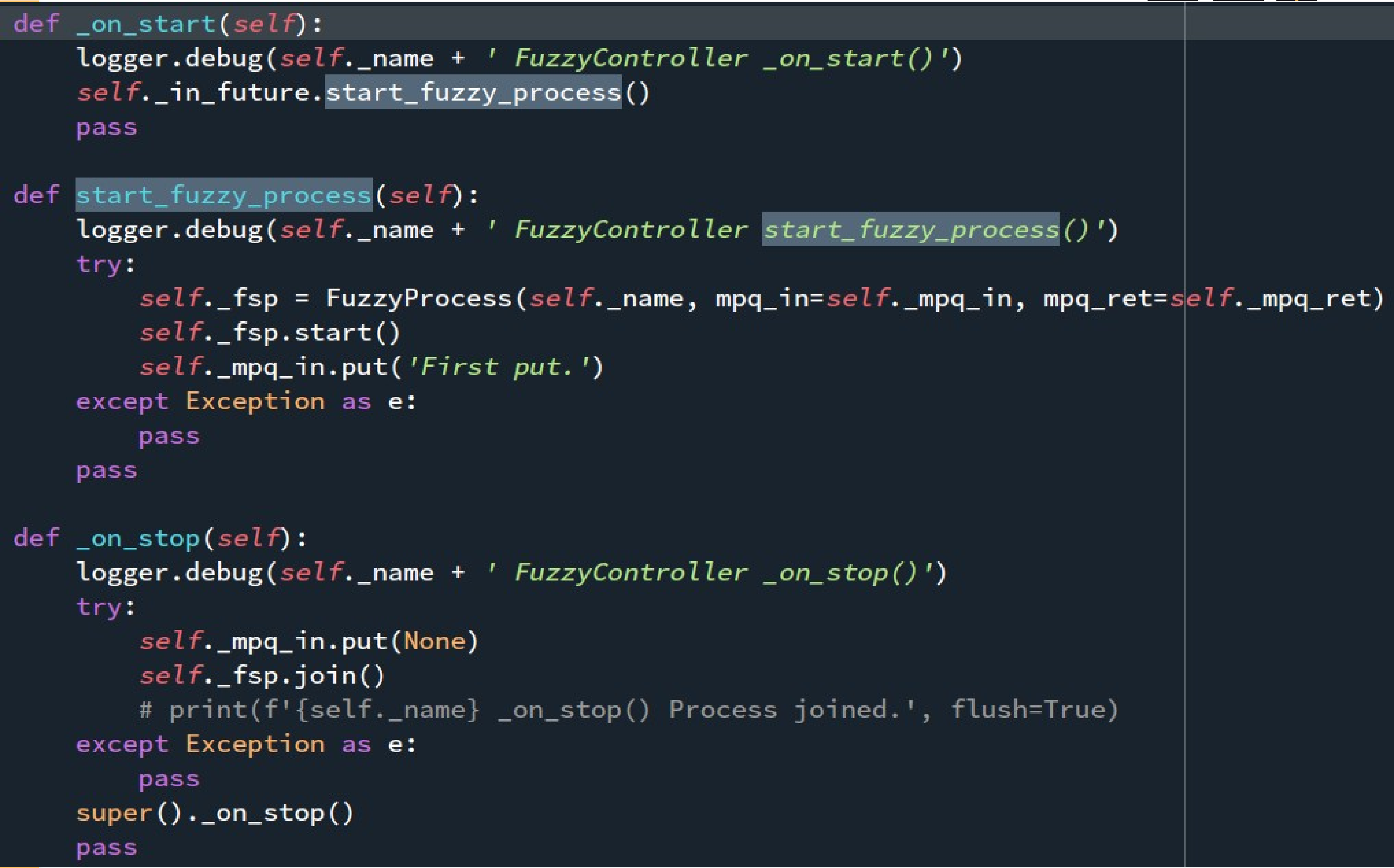
```
class VhsSm(StateMachine):
    """Statemachine for the Vhsclass."""
    # Define states
    initializing = State(initial=True)
    manual = State()
    controlled = State()
    profile = State()
    paused = State()
    cooling = State()
    stopped = State(final=True)
    # Define transitions
    to_manual = initializing.to(manual)
    request_control = manual.to(controlled)
    release_control = controlled.to(manual)
    start_profile = controlled.to(profile)
    stop_profile = profile.to(controlled)
    pause_profile = profile.to(paused)
    continue_profile = paused.to(profile)
    cooling_controlled = cooling.to(controlled)
    cooling_manual = cooling.to(manual)
    controlled_cooling = controlled.to(cooling)
    manual_cooling = manual.to(cooling)
    profile_cooling = profile.to(cooling)
    pause_cooling = paused.to(cooling)
    request_stop = manual.to(stopped)

    def __init__(self, vhs_in_future=None, nof_channels=0, history=False):
        self._history = history
        self.calls = []
        self._nof_channels = nof_channels
        self._vhs_in_future = vhs_in_future
        super().__init__()

    def on_enter_initializing(self):
        if self._history:
            self.calls.append("on_enter_initializing")
        if self._vhs_in_future:
            self._vhs_in_future.setDO(np.zeros(self._nof_channels, dtype=bool))
        self.send('to_manual')
```



```
VhsSm
    __init__
    on_enter_initializing
    on_enter_manual
    on_exit_manual
    on_enter_controlled
    on_exit_controlled
    on_enter_profile
    on_exit_profile
    on_enter_paused
    on_exit_paused
    on_enter_cooling
    on_exit_cooling
    on_enter_stopped
    on_transition_manual_controlled
    on_transition_manual_stopped
    on_transition_controlled_cooling
    on_transition_manual_cooling
    on_transition_controlled_manual
    on_transition_controlled_profile
    on_transition_profile_controlled
    on_transition_profile_cooling
    on_transition_profile_paused
    on_transition_paused_profile
    on_transition_pause_cooling
    on_transition_cooling_controlled
    on_transition_cooling_manual
```

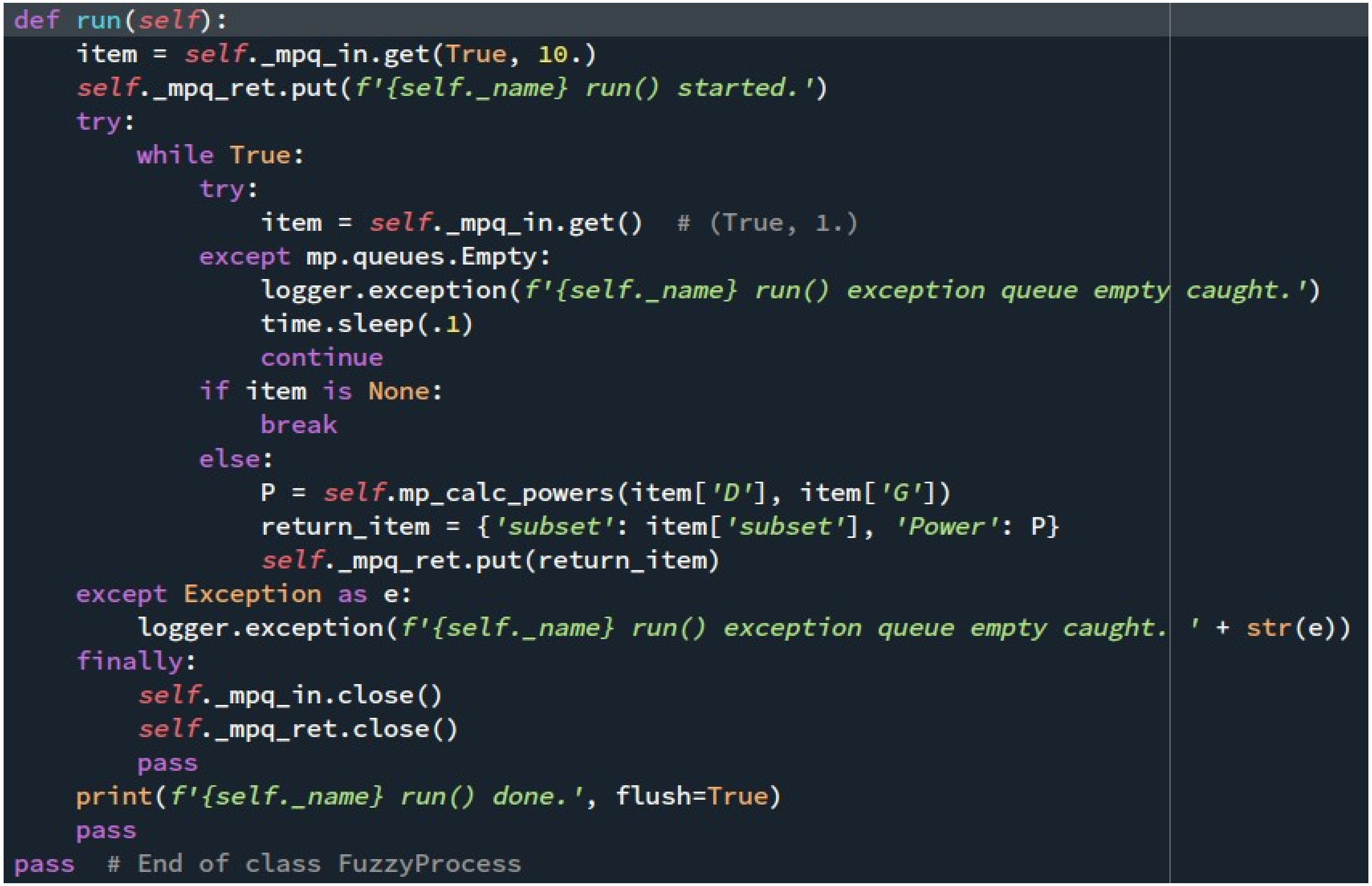


Class FuzzyController using Multi-Processing
Start FuzzyProcess



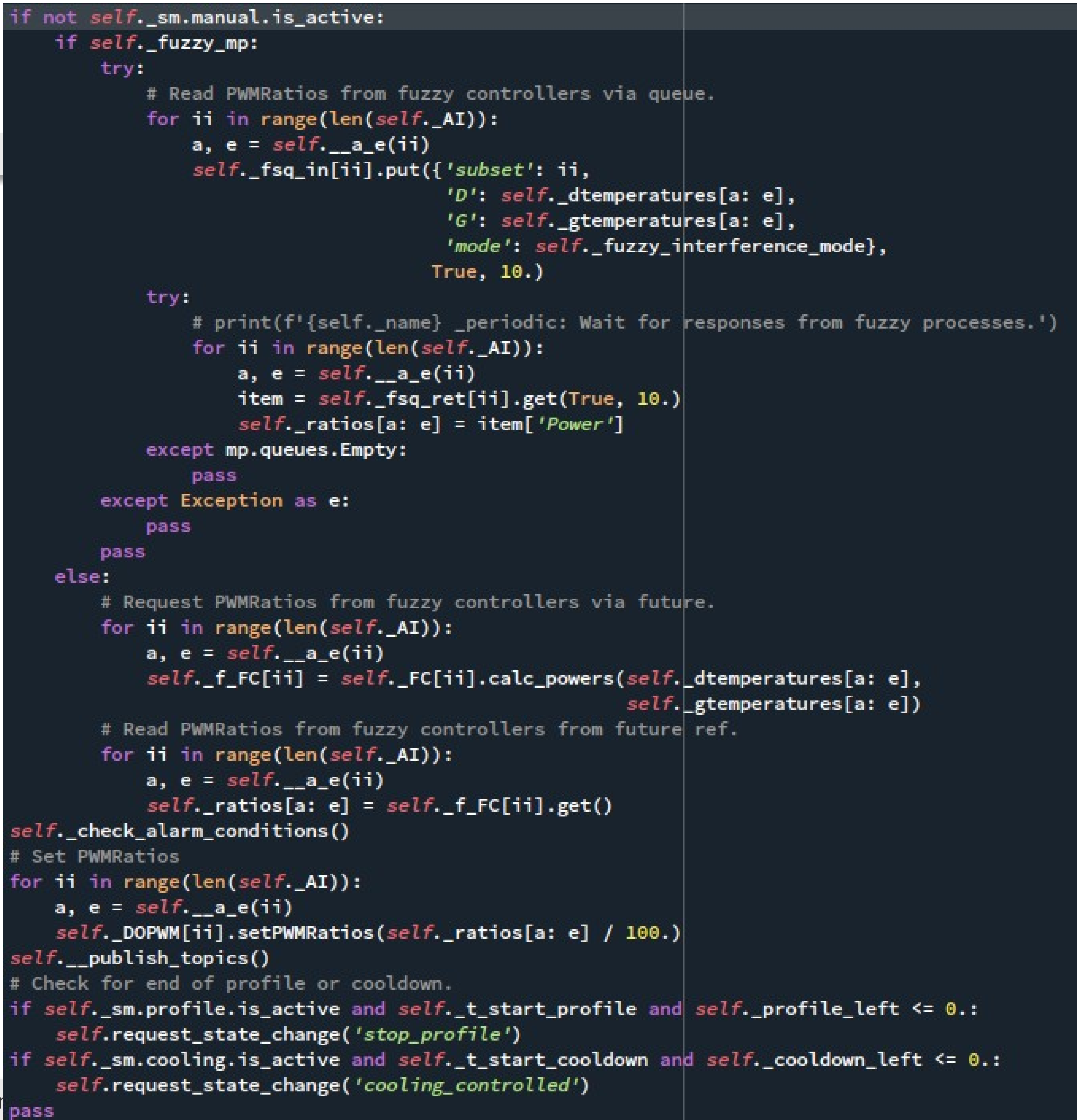
```
def _on_start(self):
    logger.debug(self._name + ' FuzzyController _on_start()')
    self._in_future.start_fuzzy_process()
    pass

def start_fuzzy_process(self):
    logger.debug(self._name + ' FuzzyController start_fuzzy_process()')
    try:
        self._fsp = FuzzyProcess(self._name, mpq_in=self._mpq_in, mpq_ret=self._mpq_ret)
        self._fsp.start()
        self._mpq_in.put('First put.')
    except Exception as e:
        pass
    pass

def _on_stop(self):
    logger.debug(self._name + ' FuzzyController _on_stop()')
    try:
        self._mpq_in.put(None)
        self._fsp.join()
        # print(f'{self._name} _on_stop() Process joined.', flush=True)
    except Exception as e:
        pass
    super()._on_stop()
    pass
```

Class FuzzyProcess(mp.Process)



```
def run(self):
    item = self._mpq_in.get(True, 10.)
    self._mpq_ret.put(f'{self._name} run() started.')
    try:
        while True:
            try:
                item = self._mpq_in.get()  # (True, 1.)
            except mp.queues.Empty:
                logger.exception(f'{self._name} run() exception queue empty caught.')
                time.sleep(.1)
                continue
            if item is None:
                break
            else:
                P = self.mp_calc_powers(item['D'], item['G'])
                return_item = {'subset': item['subset'], 'Power': P}
                self._mpq_ret.put(return_item)
    except Exception as e:
        logger.exception(f'{self._name} run() exception queue empty caught. ' + str(e))
    finally:
        self._mpq_in.close()
        self._mpq_ret.close()
        pass
    print(f'{self._name} run() done.', flush=True)
    pass
pass  # End of class FuzzyProcess
```

Class FuzzyController periodic(reason='Do')



```
if not self._sm.manual.is_active:
    if self._fuzzy_mp:
        try:
            # Read PWMRatios from fuzzy controllers via queue.
            for ii in range(len(self._AI)):
                a, e = self.__a_e(ii)
                self._fsq_in[ii].put({'subset': ii,
                                      'D': self._dtemperatures[a: e],
                                      'G': self._gtemperatures[a: e],
                                      'mode': self._fuzzy_interference_mode},
                                     True, 10.)
            try:
                # print(f'{self._name} _periodic: Wait for responses from fuzzy processes.')
                for ii in range(len(self._AI)):
                    a, e = self.__a_e(ii)
                    item = self._fsq_ret[ii].get(True, 10.)
                    self._ratios[a: e] = item['Power']
            except mp.queues.Empty:
                pass
        except Exception as e:
            pass
        pass
    else:
        # Request PWMRatios from fuzzy controllers via future.
        for ii in range(len(self._AI)):
            a, e = self.__a_e(ii)
            self._f_FC[ii] = self._FC[ii].calc_powers(self._dtemperatures[a: e],
                                                      self._gtemperatures[a: e])
        # Read PWMRatios from fuzzy controllers from future ref.
        for ii in range(len(self._AI)):
            a, e = self.__a_e(ii)
            self._ratios[a: e] = self._f_FC[ii].get()
self._check_alarm_conditions()
# Set PWMRatios
for ii in range(len(self._AI)):
    a, e = self.__a_e(ii)
    self._DOPWM[ii].setPWMRatios(self._ratios[a: e] / 100.)
self.__publish_topics()
# Check for end of profile or cooldown.
if self._sm.profile.is_active and self._t_start_profile and self._profile_left <= 0.:
    self.request_state_change('stop_profile')
if self._sm.cooling.is_active and self._t_start_cooldown and self._cooldown_left <= 0.:
    self.request_state_change('cooling_controlled')
pass
```

References

- CS Framework
  - https://wiki.gsi.de/CSframework/WebHome
- Actor Framework
  - https://forums.ni.com/t5/Actor-Framework-Documents/tkb-p/7301
- **CS++**
  - Scientific Report 2014 GSI Report 2015-1, 459 p. (2015) http://repository.gsi.de/record/184173/files/FG-GENERAL-41.pdf
  - A.N. State, ..., H. Brand, ..., “The slow control system of the FRS Ion Catcher”, Nuclear Inst. and Methods in Physics Research, A 1034 (2022) 166772
- python.org
  - Documentation https://docs.python.org/3/
  - Tutorials https://docs.python.org/3/tutorial/index.html
  - Simpful
    - Spolaor S., Fuchs C., Cazzaniga P., Kaymak U., Besozzi D., Nobile M.S.: Simpful: a user-friendly Python library for fuzzy logic, International Journal of Computational Intelligence Systems, 13(1):1687–1698, 2020 DOI:10.2991/ijcis.d.201012.002
  - Asynchronous / Prarallel Prgamming
    - Multi-threading: https://docs.python.org/3/library/threading.html
    - Multi-processing: https://docs.python.org/3/library/multiprocessing.html
    - Asyncio: https://docs.python.org/3/library/asyncio.html
    - Advertising: https://superfastpython.com/
      - Develop Faster Python Programs with Threading, Multiprocessing, and AsyncIO

Diskussion

- Dank an alle Kollegen, die mich unterstützen.
  - Python, MQTT, Telegraf, InfluxDB, Grafana, Linux
  - https://mattermost.gsi.de →EPICS →Python (GSI Web-Lgin required.)
  - Alle Notizen sind in meinem Joplin-Notizbuch zu finden.
    - Ich gebe es auf Anfrage gern frei, auch zum Mitmachen.
- Dank für Eure Aufmerksamkeit
- Fragerunde und Diskussion ist eröffnet.
  - Programmdetails
  - Konfigurationen
  - Architekturen
  - Frameworks
  - ...