GROUPE ATLANTIC

Thermal comfort for all

Thermal comfort is life

atlantic ideal HEATING Thermor Jauter ACV Austria Email Hamworthy GLEDHILL YGNIS Triangle Tube erensan lazzarini Keston BY IDEAL HEATING edesa ORCON TEPLOLUXE INNOVERT HAUTEC Wärmepumpen THERMIC ENERGY Regenerative Energie

GROUPE ATLANTIC

OUR MISSION

To transform prevailing energies into lasting well-being

12 000
EMPLOYEES

€2.7bn
NET TURNOVER

4
COMMERCIAL OPERATIONS ACROSS 4 CONTINENTS

31
INDUSTRIAL SITES

10M
APPLIANCES AND PRODUCTS PRODUCED PER YEAR

INTERNATIONAL **DIVISION**

UNITED KINGDOM **DIVISION**
REPUBLIC OF IRELAND
NORTH AMERICA

RADIATOR & SANITARY HOT WATER **DIVISION**
AIR CONDITIONING AND AIR TREATMENT **DIVISION**
HEAT PUMP & BOILER **DIVISION**
CUSTOMER SERVICE **DIVISION**

GROUPE ATLANTIC

OUR

CORE BUSINESS

HVAC: Heating Air and Water, cooling and air treatment.

GROUPE ATLANTIC designs, manufactures and markets **energy-efficient solutions and systems, accessible to all and suited to individual needs.**

1. Project Overview

Project overview and main goals

- Model a full Air Handling Unit (AHU) with GT-Suite (1D model)
- Couple M172 Programmable Logic Controller (PLC) to 1D AHU model to be able to test the regulation software (same than serial one)
- Perform several typical user cases to test software robustness (scenarios)
- Improve the model with as many options as possible to represent the full complexity of AHU configurations

Project overview and main goals

Main Goal: Improve our capacity to test software robustness

Yesterday: Only manual test

Test case with potentiometers (manual)

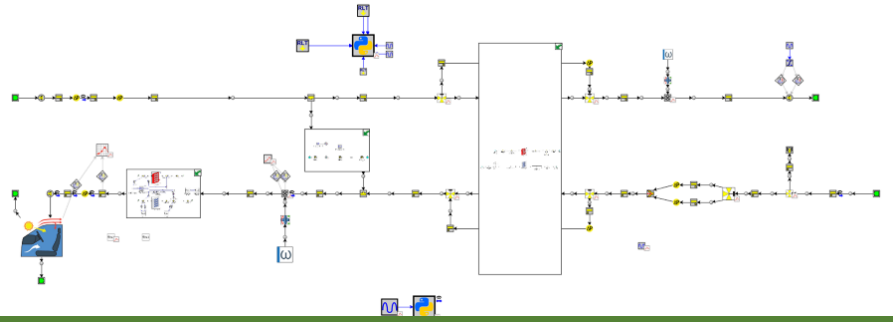
Tests on real AHU (or Field tests)

Project overview and main goals

Main Goal: Improve our capacity to test software robustness

Tomorrow: PLC connected to a virtual AHU

Manual tests on case with potentiometers



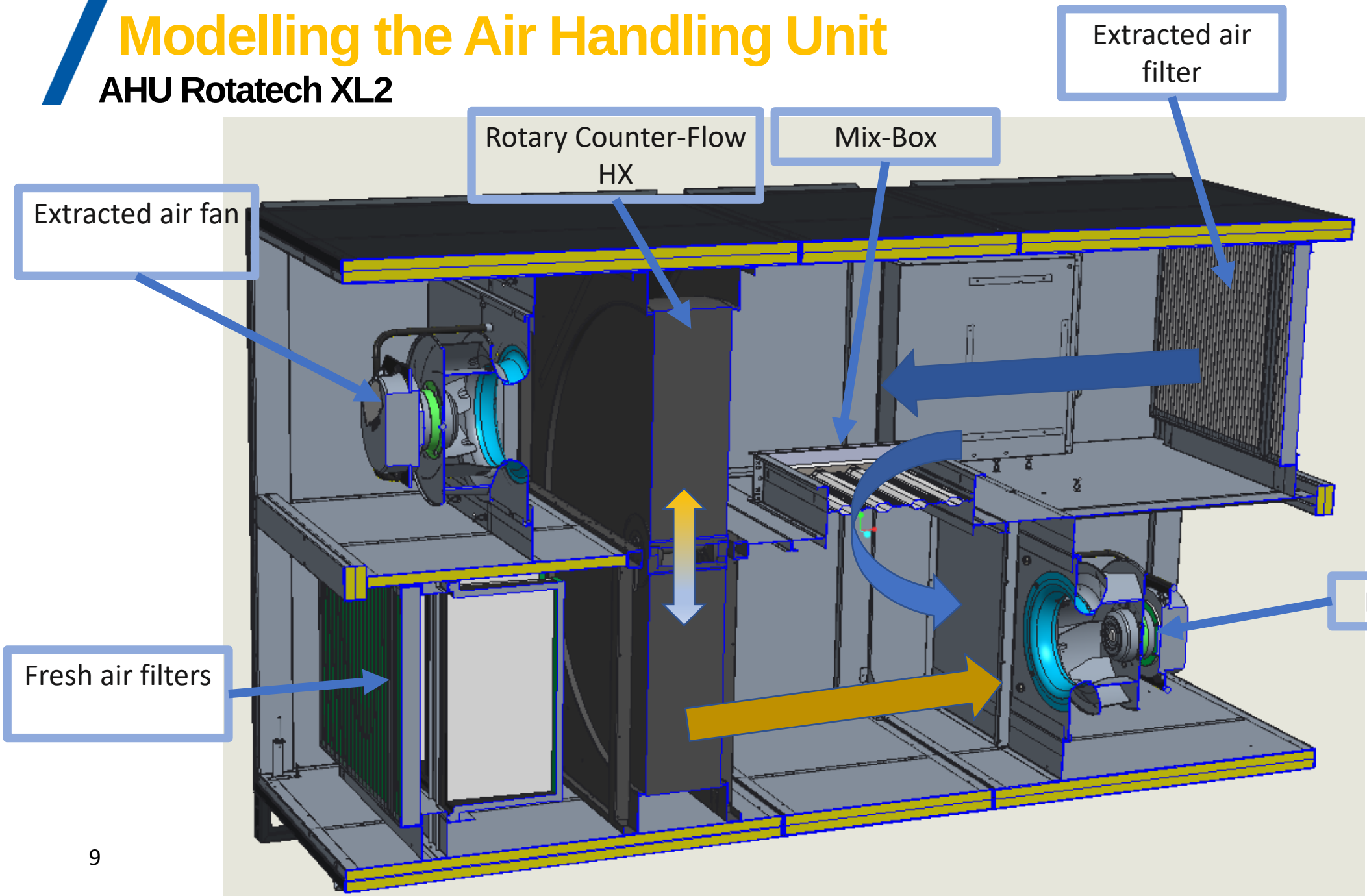
Automatic validation of the software with the model (24/7)

Final tests on real AHU

2. Air Handling Unit model

Modelling the Air Handling Unit

AHU Rotatech XL2

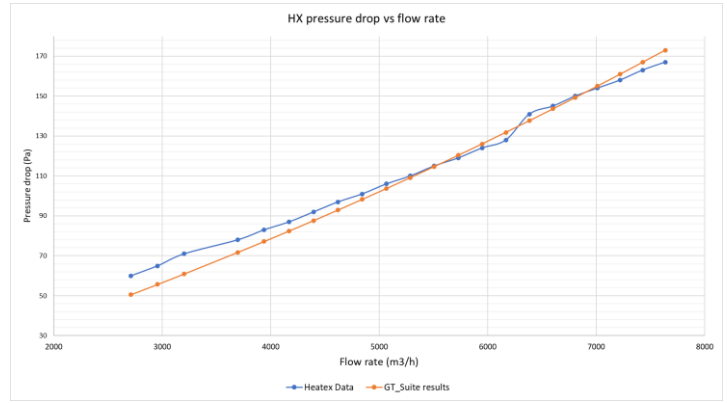


Optional:

- Hydraulic coils
- Dampers (not represented here)
- Electric Heater
- Additional filters
- Sensors
- Etc…

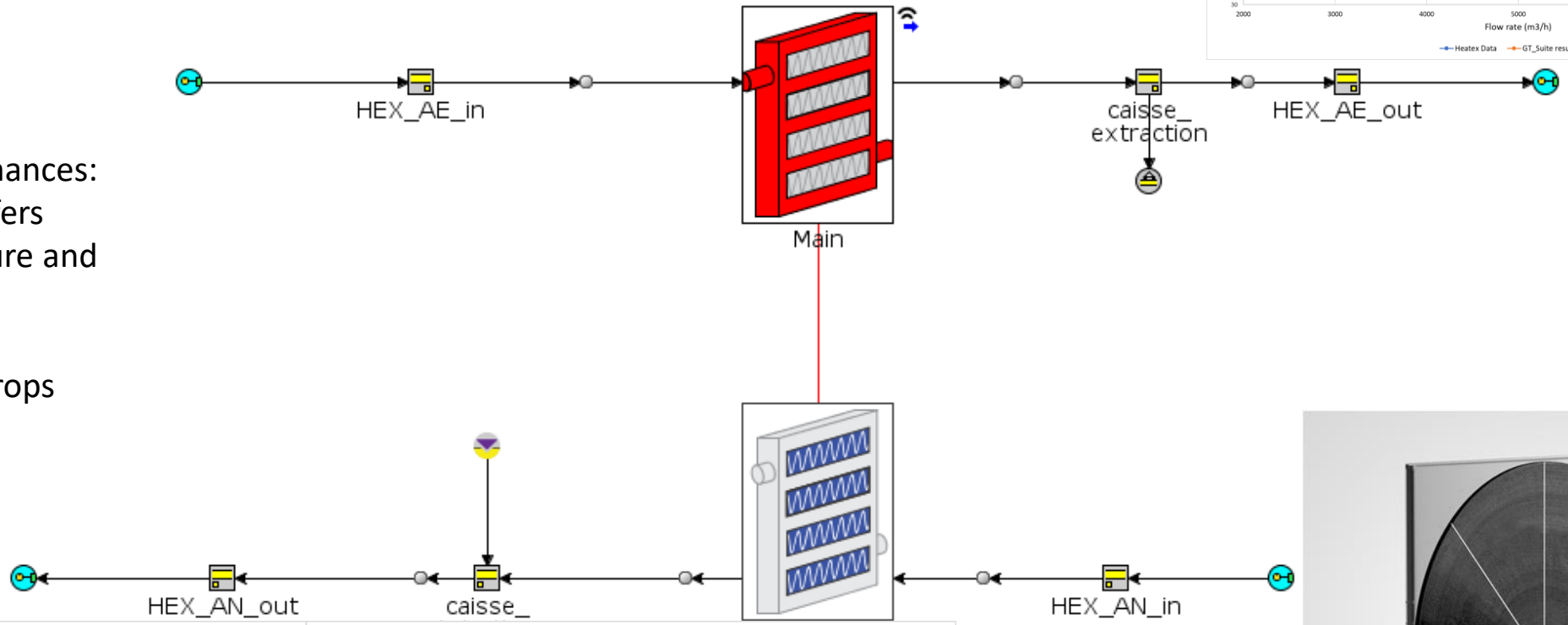
Modelling the Air Handling Unit

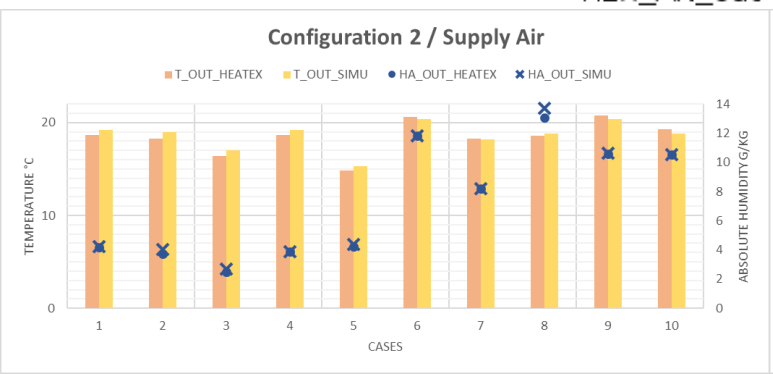
Rotary Heat exchanger

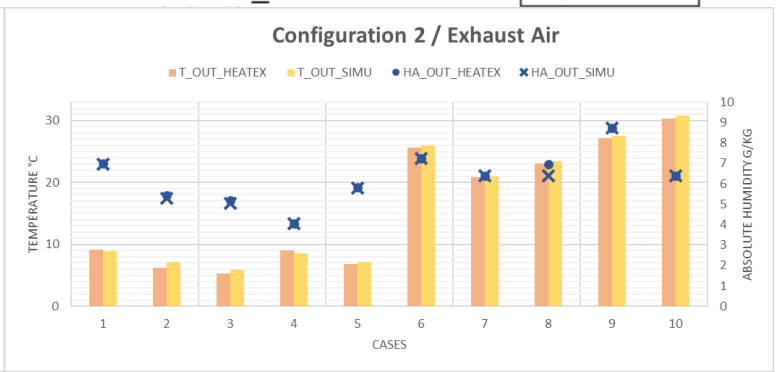


Needs:

- Supplier performances:
  - Heat transfers (temperature and humidity)
  - Geometry
  - Pressure drops

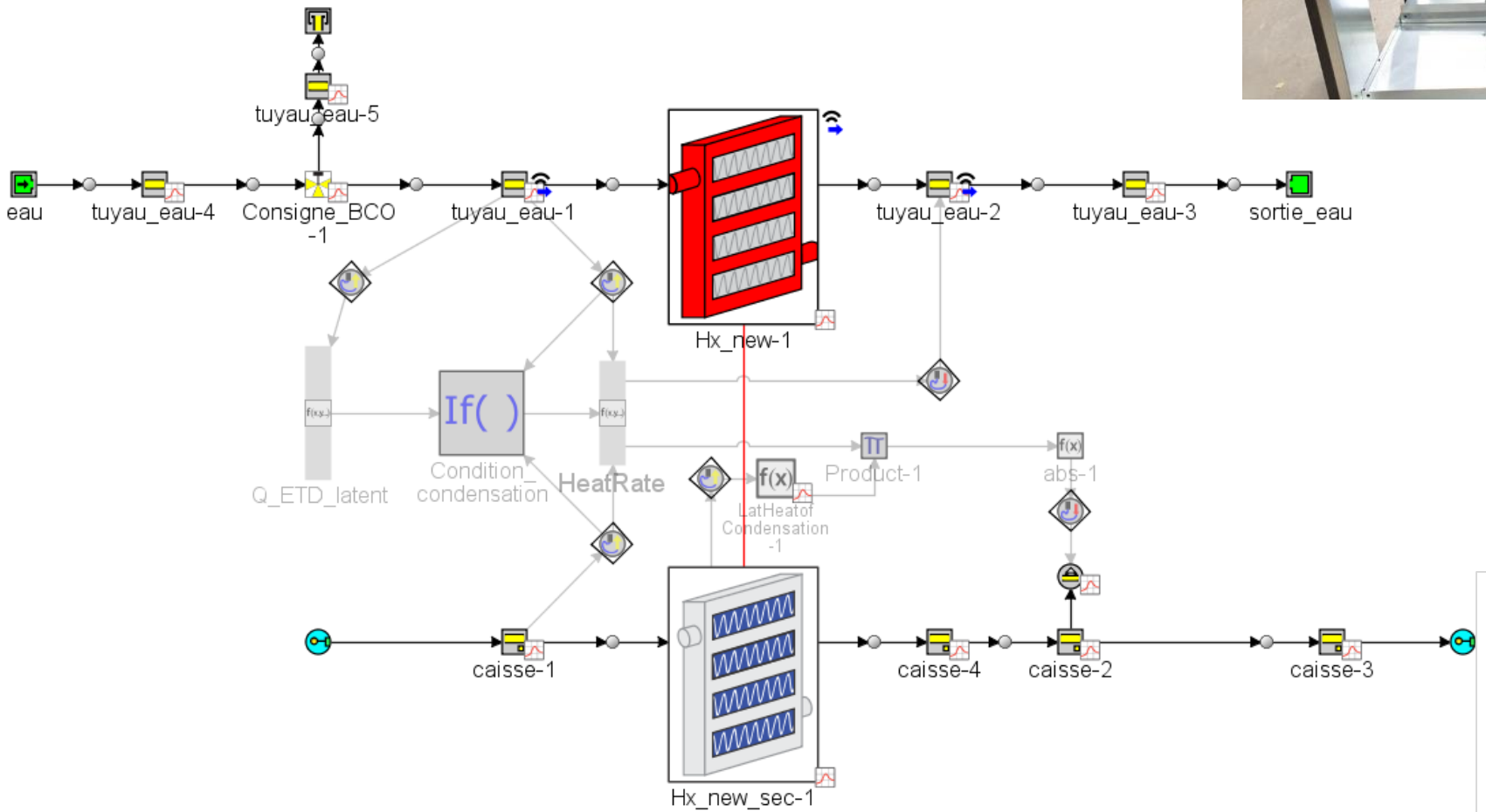






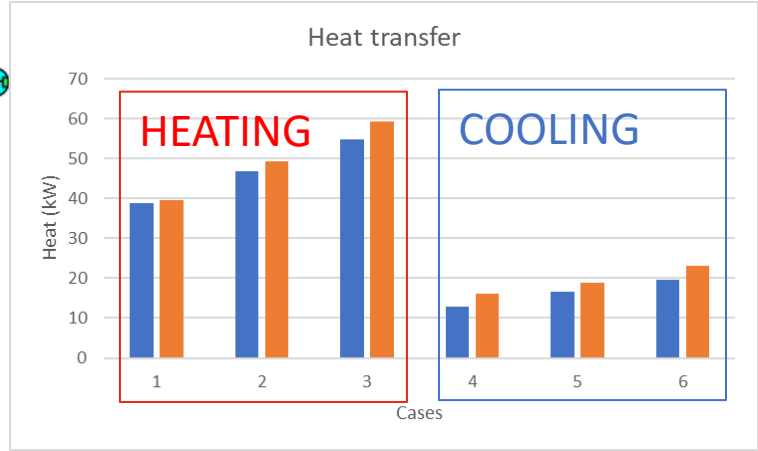
Modelling the Air Handling Unit

Hydraulic Coils (cooling and heating)



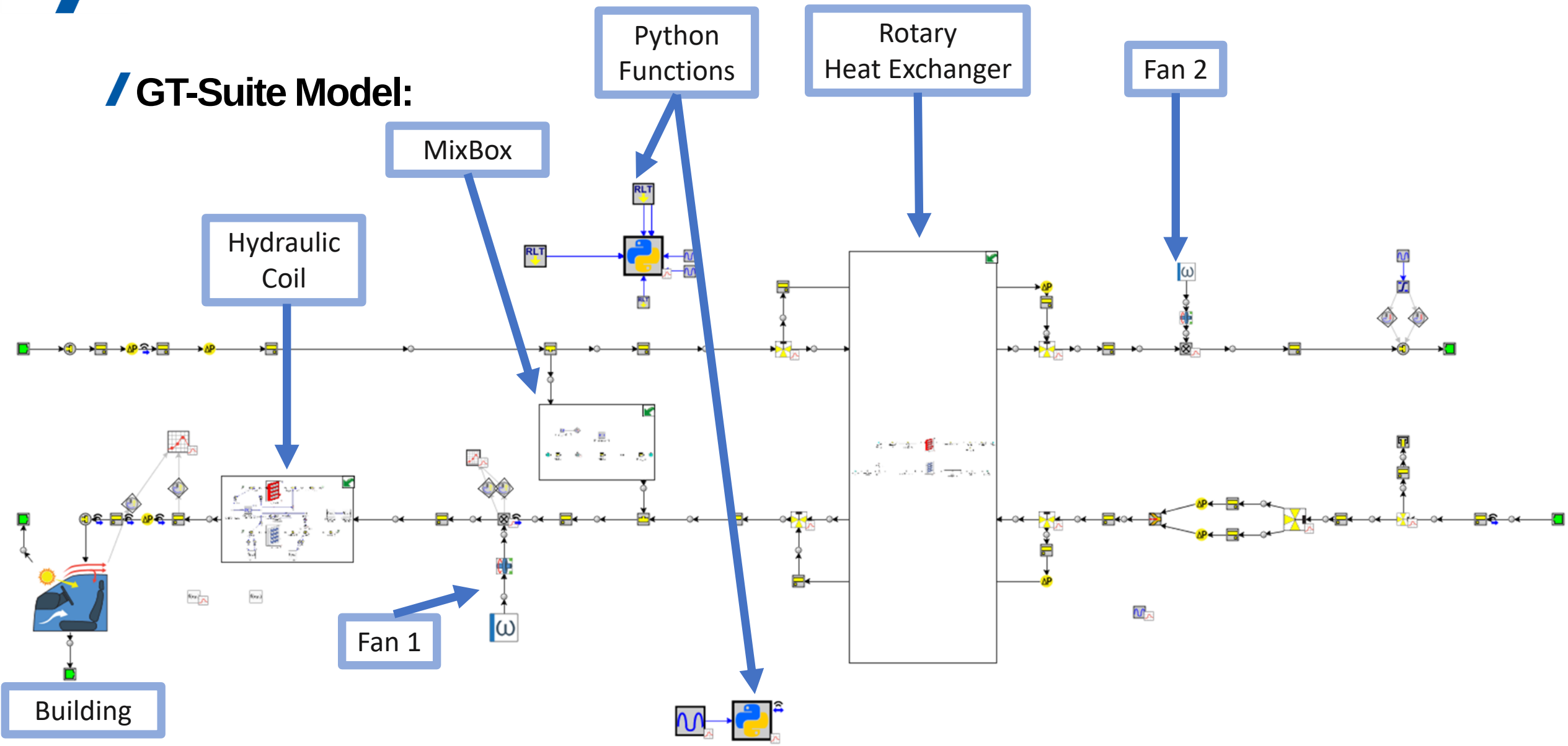
Needs:

- Thermal performances
- Pressure drops
- Geometry



Modelling the Air Handling Unit

GT-Suite Model:

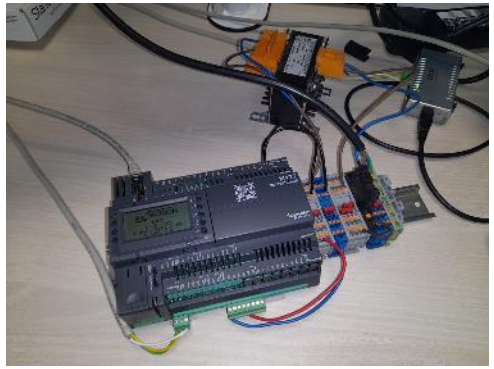


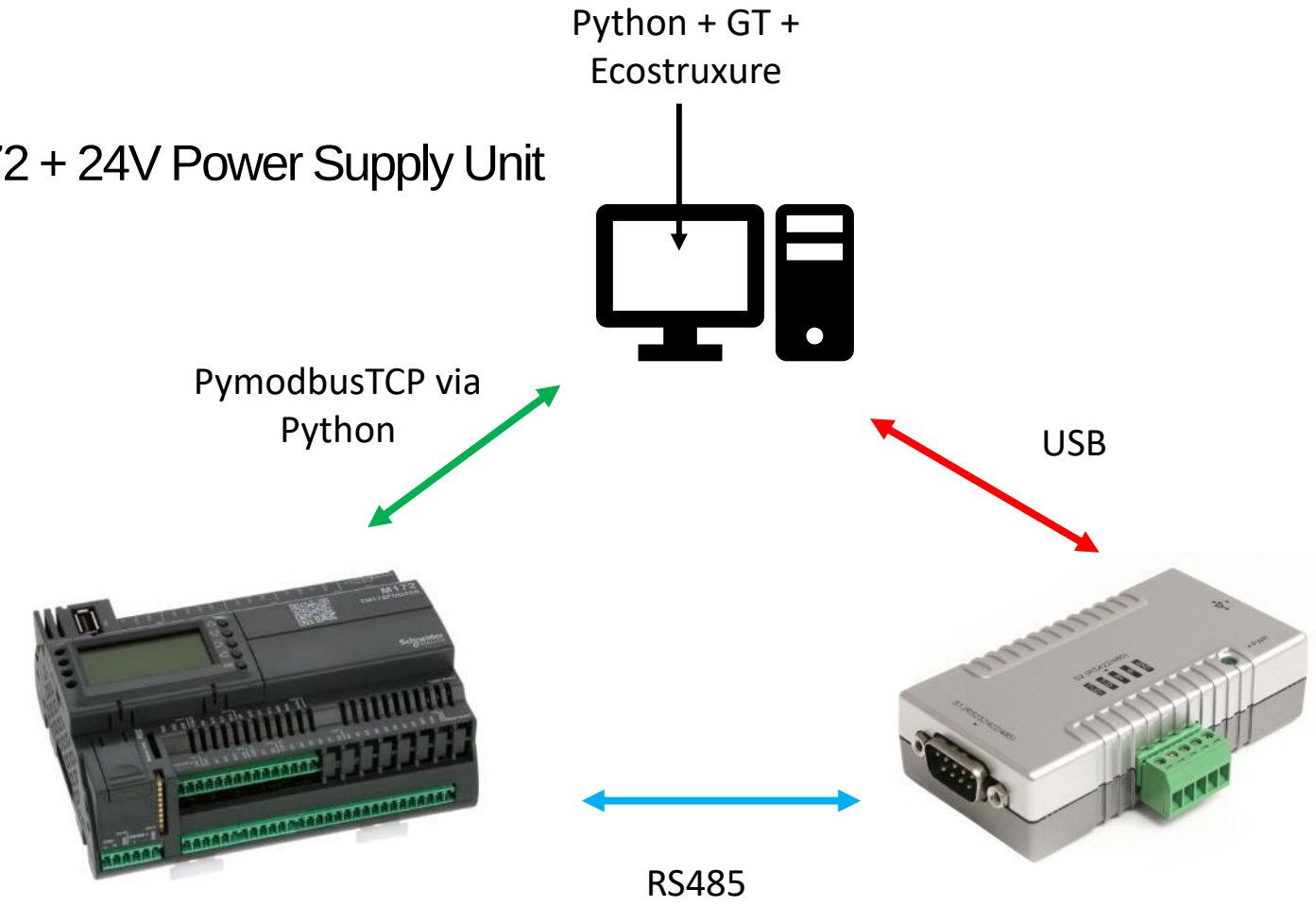
3. PLC coupling

Coupling with the M172 (Schneider Electric)

Required equipment

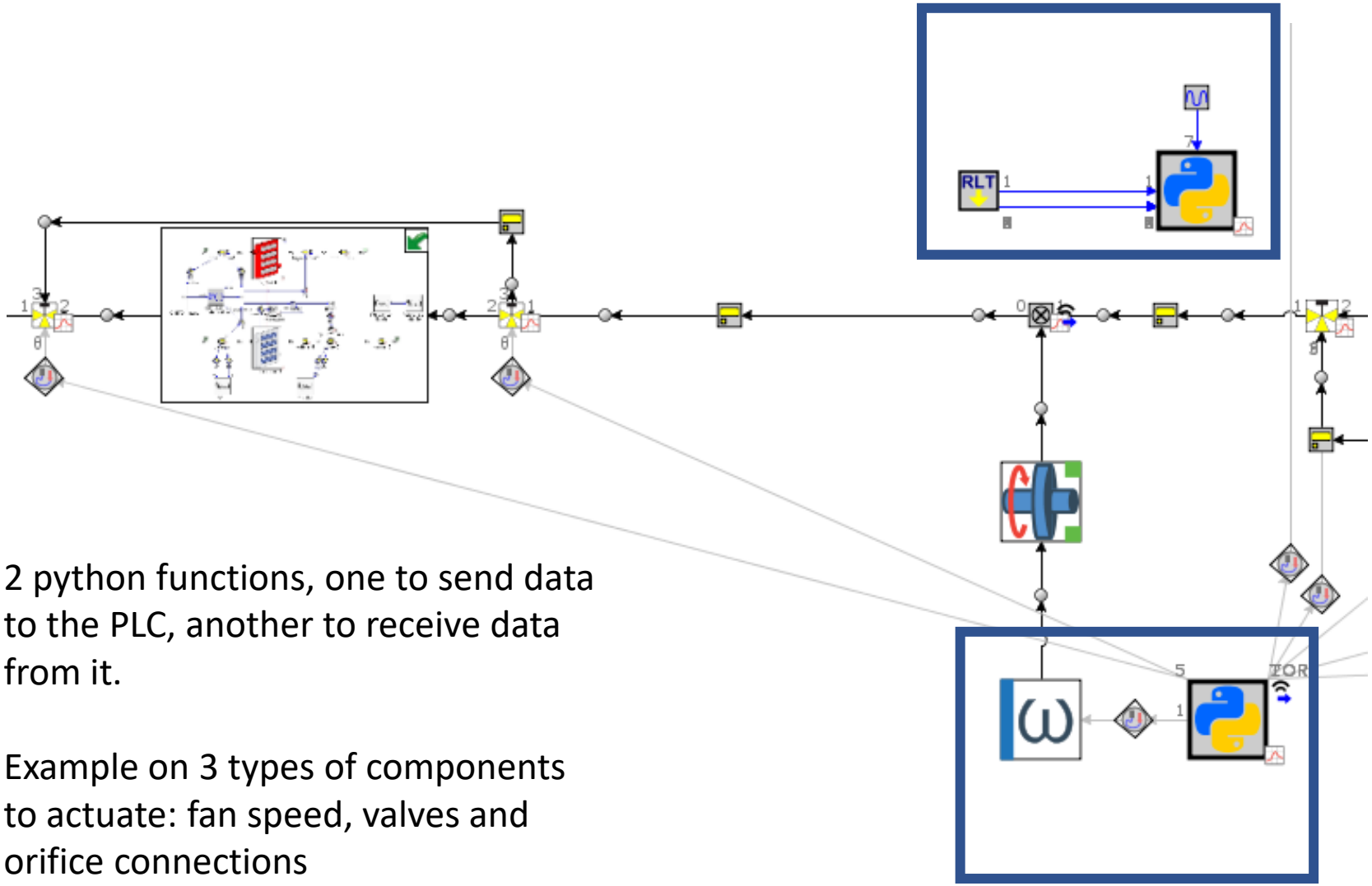
- Programmable Logic Controller M172 + 24V Power Supply Unit
- USB/RS485 converter
- GT-Suite license
- Any computer with Python and 16gb of RAM
- Optional: Ecostruxure software (Schneider one) to spy on the PLC during the run





Coupling with the M172

General concept



- 2 python functions, one to send data to the PLC, another to receive data from it.
- Example on 3 types of components to actuate: fan speed, valves and orifice connections

Coupling with the M172

General concept

RLT | Options | Plots

Attribute	Output Signal #1	Output Signal #2	Output Signal #3	Output Signal #4	Output Signal #5
RLT Variable	t_air_neuf+	tavr0:caisse-...	tavl0:caisse-...	fan-toa0:ven...	fmr-flrt:ventil...
Initial Output	0.0	0.0	0.0	0.0	0.0
Multiplier to Output Signal	def	def	def	def	def

GT new time step with the new values

```
def Receive(time: float, inputs: dict, outputs: dict) -> int:

#Réception vitesse fan AN

    inputs[1] = [0]
    inputs[1] = client.read_holding_registers(10014)
    decoder = BinaryPayloadDecoder.fromRegisters(inputs[1], Endian.Big)
    vAN = inputs[1][0]
    outputs[1] = vAN
```

Values retrieval and injection into GT with the 'Receive' function

Reading data from GT and writing those in the pymodbus registers with the 'Send' function

```
def Send(time: float, inputs: dict, outputs: dict) -> int:

    EAN = (int(inputs[1]-273)*10)
    outputs[1] = client.write_single_register(9049, EAN)

    SAN = int(inputs[2]*10)
    outputs[2] = client.write_single_register(9050, SAN)
```

PLC calculation

Commissioning Watch

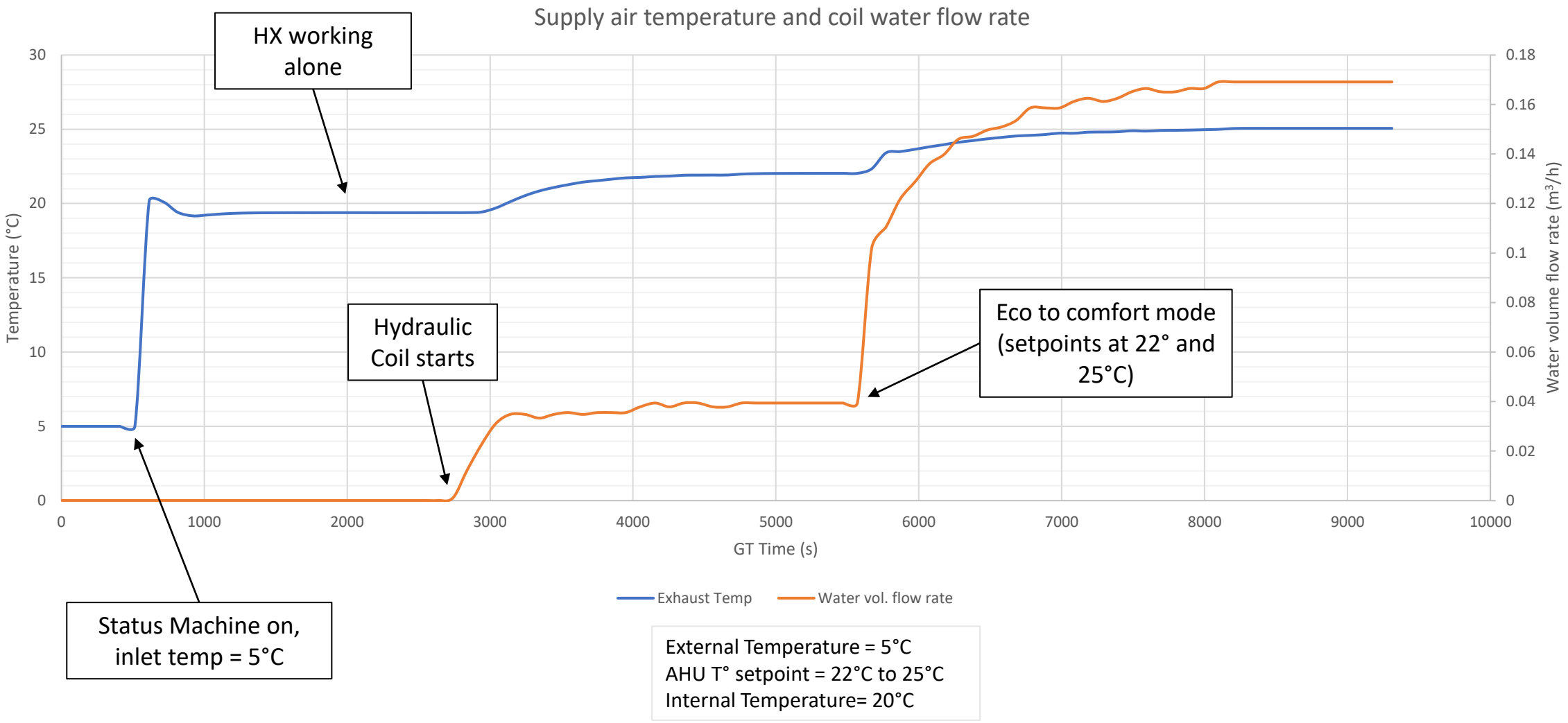
Device	Name	Value	SubIdx	Um
M172 28-42 I/Os	G_iSupplyFanCmd_433	574	-	
M172 28-42 I/Os	G_uiSupplyActualSpeed_3186	1380	-	Tr/min
M172 28-42 I/Os	G_iReturnFanCmd_431	342	-	
M172 28-42 I/Os	G_uiReturnActualSpeed_3193	823	-	Tr/min

Even if GT-Suite model is around 5* faster than real time the PLC works at real time

4. Some examples

Results

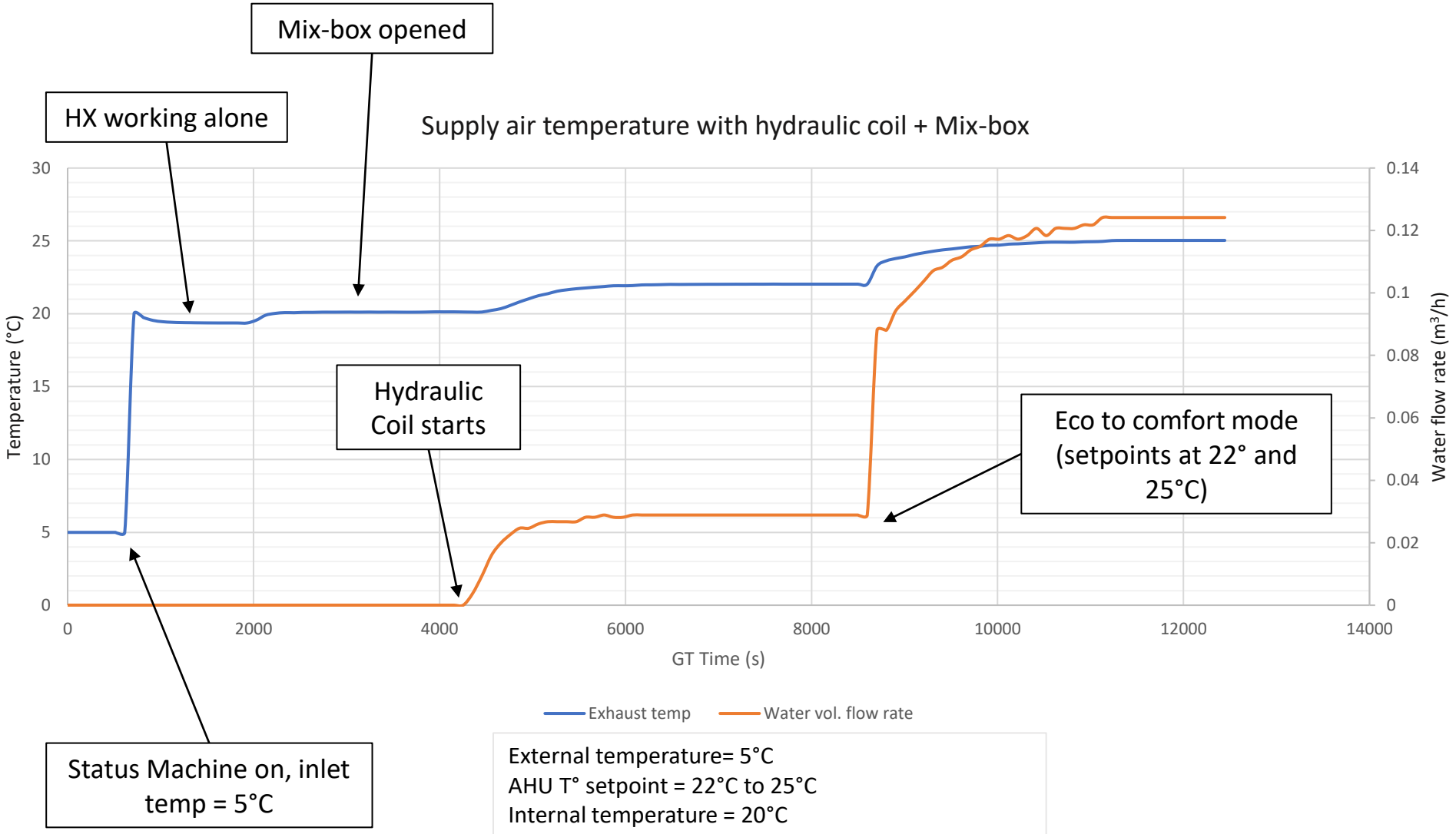
Single Zone, constant volume flow rate



Verify Thermal Regulation over a single sequence (HX + Hydraulic Coil)

Results

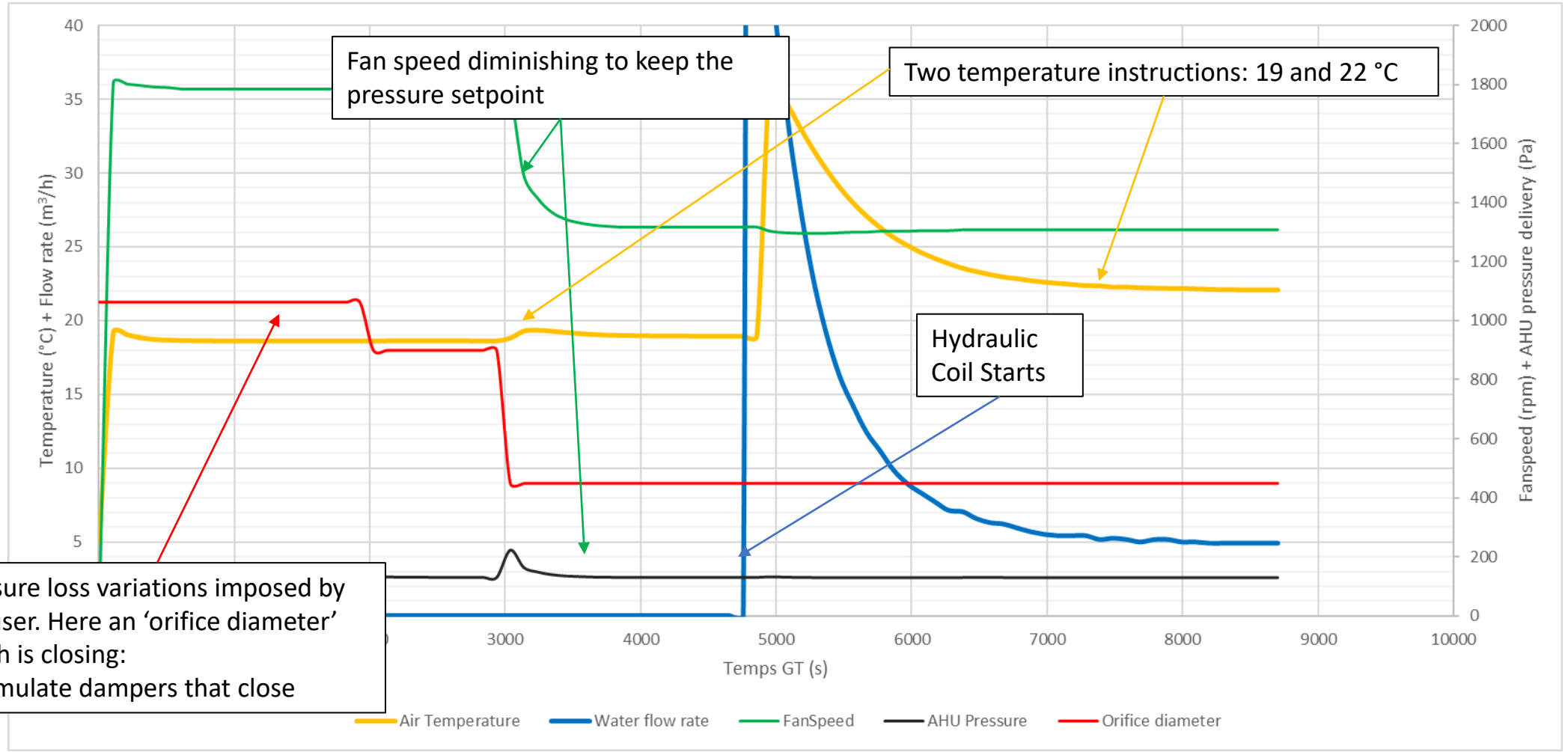
Single zone, constant volume flow rate



Verify overall Thermal Regulation (HX + Mix-Box + Hydraulic Coil)

Results

Multi-Zone, Constant pressure to the building with hydraulic coil



Verify good fan behavior and Hydraulic Coil responsiveness

5. Conclusions

Conclusions

- **Good behavior of the virtual AHU model (steady state and transient)**
- **The connection with the PLC was quite easy to set (Python function)**
- **The hardest points were the Air/Air Heat exchanger and the hydraulic coil for cooling mode due to air condensation (latent heat)**

- **Use cases:**
  - Non regression test for new software versions
  - New function pre-validation

- **Next Step:**
  - Create a 24/7 test bench to detect as early as possible bugs/improvements on the control software

THANK YOU