Smart Virtual Thermostat for Domoticz

Version 0.1 Jan 18 2017

Developed by logread (pm user “logread” on the Domoticz forum at http://www.domoticz.com/forum).

This script is based on the Vera plugin from Antor (see http://www.antor.fr/apps/smart-virtual-thermostat-eng-2/?lang=en), but significantly rewritten due to Domoticz peculiarities.

Purpose:

This scripts allows to run on Domoticz a virtual thermostat that will switch on/off heating device(s) based on a proportional time logic rather than more conventional setpoint +/- hysteresis thermostats, allowing a much finer temperature control over time, once the initial learning phase is over (up to 3 days). Please read http://www.antor.fr/apps/smart-virtual-thermostat-eng-2/?lang=en for more details.

Installation:

Before implementing the script in Domoticz, please read carefully the below instructions and follow the exact sequence.

Step 1: Make note of the names of your Domoticz devices that will be used by the script:

1. Heating device(s) – typically switch(es) with on/off status – MANDATORY
2. Inside temperature sensor(s) - MANDATORY
3. Outside temperature sensor(s) – OPTIONAL (but heating control will work much better if used)
4. Contact sensor(s) – that will pause the thermostat if a door or window stays open for a certain period of time OPTIONAL

You will need the names of these devices later in the process (step 3.3)

Step 2: Run the SVT_Setup.lua script to download from the https://github.com/999LV/SmartVirtualThermostat GitHub repository to automatically create the Domoticz virtual devices and user variables that the script MANDATES. The following devices will be

1. "Thermostat" -- virtual, type switch
2. "Forced_Heating" -- virtual, type switch
3. "Temp_Thermostat" -- virtual, type temperature
4. "Day_SetPoint" -- virtual type thermostat setpoint
5. "Night_SetPoint" -- virtual type thermostat setpoint

The device names can be changed from the above, but you will then need to adjust some script variables in step 3.3)

Note that you cannot run the SVT_Setup.lua script from Domoticz as it requires external modules (socket.http and dkjson) that are not available from within the Domoticz environment. You therefore need to install Lua and the LuaRocks modules on your system (if you do not know if it is installed or how to install it, Google is your friend… not me), copy/download the script to some place on your system and run it from the command line shell.

Step 3: Create and edit the script code:

1. Create a new Lua time script in Domoticz (menu Setup / More Options / Events and see help if you are not familiar with Domoticz scripting).

2. Paste the code from the SVT_timescript.lua script from the https://github.com/999LV/SmartVirtualThermostat GitHub repository, replacing the sample code that Domoticz always places in new scripts

3. EDIT the devices naming section of the code to match your setup (as per step 1):
```
local thermostat = "Thermostat"
local forcedheating = "Forced_Heating"
local tempthermostat = "Temp_Thermostat"
local setpoints = {"Day_SetPoint", "Night_SetPoint"}
local heaters = {"Heater"}
local sensorsin = {"Temp_Inside"}
local sensorsout = {"Temp_Outside"}
local sensorspause = {"Door"}
```
Note that the heaters, sensorsin, sensorsout and sensorspause are Lua tables of device names allowing multiple sensors and heaters device(s). For instance if you have two contact sensors that you want to use to trigger a pause in heating on a door (“Door”) and a window (“Window”), the correct code edit will be `local sensorspause = {"Door", “Window”}`
Also note that when multiple temperature sensors are provided, the script will perform an averaging calculation (displayed in the “Temp_Thermostat” virtual device for inside temperature). Sensors not updated in the last hour (adjustable in the “variables and constants” section of the code) are presumed dead by the script, possibly causing the thermostat to switch off by security to avoid overheating (it will then write an error message in the Domoticz log).
If you do not plan on using any outside temperature sensor or pause sensor, you can just specify an empty table (`e.g. local sensorpause = {}`)

4. Select “Event active” box and save the script. The name of the script can be of your choice, BUT it needs to be of type “Lua” and a “time” event.

Step 4: Edit as needed the Domoticz “User Variables” that have been automatically created by the SVT_Setup.lua script upon its run in Step 2 above:

SVT_DayStartHour___________time when Day SetPoint is active (default 07:00),
SVT_NightStartHour__________time when Night SetPoint is active (default 22:00),
SVT_CalcInterval_____________seconds between consecutive thermostat cycles (default is 1200 that is 20 minutes). Please see “Configuration” section below on how to set this critical parameter.
SVT_ForcedHeatingDuration___seconds of heating when forced mode is active (default is 3600 that is 1 hour)

SVT_PauseDelay_____________seconds of waiting after a pausing sensor is either becoming open or closed (or on/off) before thermostat heats again or pauses (default 60 that is 1 minute). This avoids switching heating on/off repeatedly when a door or window is briefly opened.

Please DO NOT edit or change the “SVT_Internals” and “SVT_AutoLearning” user variables as these are needed by the code to save its variables between runs of the script.

Configuration:

The critical variable you need to adjust first in your setup is the “cycle time” of the thermostat script (“SVT_CalcInterval” user variable), that is the time between two heating calculations take place.

If this variable is set too low, then the thermostat will not “learn” about your physical home/room parameters and will just keep oscillating like a regular thermostat… It will work but not in an optimized way.

If this variable is set too high, then the thermostat will not be able to compensate adequately for the actual thermal inertia or insulation coefficient of your home/room and you will notice significant “cold” spikes at the end of each cycle.

The higher the thermal inertia and the better the insulation of your home/room are, the higher the calculation interval needs to be. As a guidance, a low inertia setup (simple room electric heaters for instance) should probably use a calculation interval between 900 and 1800 seconds (1/4 to 1/2 hour) while a high inertia setup (well insulated house with whole house circulating hot water boiler) should probably use 3600 seconds (1 hour).

Once you start the thermostat, it will start learning and auto-calibrate itself over time. After two or three days, you can then see how the temperature actually behaves (log of the “Temp_Thermostat” device) and then can decide to change the “SVT_CalcInterval” user variable.

There are certain parameters you can use to monitor and tune the thermostat operation:
“SVT_AutoLearning” user variable:

It is stored as a comma separated list of variables as follows: X, Y1, Y2, T1, T2, P.

- X is the state parameter: 0 = uninitialized, 1 = initialized, 2 = disabled.
- Y1 and Y2 correspond to number of learnings for the constants C (inside temps) and T (outside temps). when Y reaches 50 first learning is complete, but self-calibration continues.
- T1 and T2 are temperatures inside and outside at time of last calculation power.
- P is the percentage of power from the last calculation.

“SVT_Internals” user variable:

It is stored as a comma separated list of variables as follows: X, Y, PH, H, FH, FHT, C, T

- X is the timestamp of the last calculation.
- Y is the timestamp of the end of the current heating period when heating is on.
- PH is the pause state of the thermostat (0 = no pause, 1 = pause) based on the pause sensors input.
- H is the heating state of the thermostat (0 = no heat, 1 = heat on)
- FH is the forced heating state of the thermostat ((0 = automatic mode, 1 = forced heating)

- FHT is the timestamp of the forced heating, if active, has been set
- C is the heating constant for internal temperature, used in the heating power calculation. It depends on the size of your room(s) and the heating capacity of your heater(s). It is set by default to 60 and subsequently adjusted by the auto-learning process.
- T is the heating constant for outside temperature, used in the heating power calculation. It depends primarily on the level of insulation of your premises. It is set by default to 1 and subsequently adjusted by the auto-learning process.

Note that timestamps are in Unix epoch format for easier internal calculations (use a converter such as www.epochconverter.com if you want to see the data in human readable format).

There are also a few configuration constants embedded into the script code, that you may want to change:

```
-- scripts constants - can be modified by user
local dirtydata = 3600 -- number of seconds since last update, used to
determine if a given sensor is alive or presumed dead
local powermin = 0 -- minimum heating at each calculation cycle (0-100)
local deltamax = 0.2 -- allowed temp excess over setpoint temperature
local debug = false – set to true to turn on verbose logging for
debugging purposes
```