

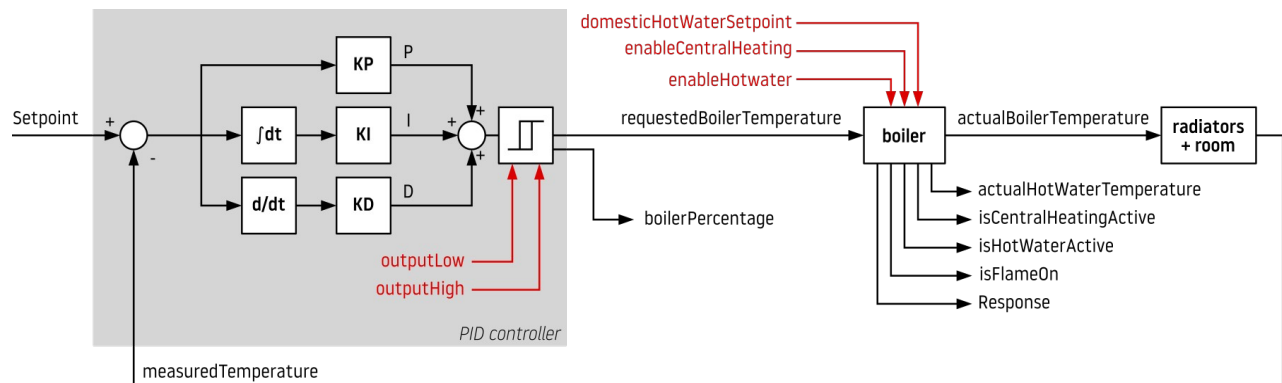
# Hoe het werkt

De *measuredTemperature* is de temperatuur die door de thermometer in de kamer wordt gemeten.

Het *Setpoint* is de temperatuur die het in de kamer moet zijn.

- Overdag staat deze meestal op 21°C.
- 's Nachts is deze 10°C
- Je kunt hem ook overrulen waardoor hij bijvoorbeeld ook op 20°C of 20.5°C kan staan

De temperatuurregelaar ziet er schematisch als volgt uit:



De rode waarden / pijltjes staan wel in de software maar worden niet naar de logfile geëxporteerd. Ze zitten dus niet in de dataset.

Er is gebruik gemaakt van een PID regelaar (<https://nl.wikipedia.org/wiki/PID-regelaar>) die de watertemperatuur *requestedBoilerTemperature* van de ketel bestuurt op basis van de gemeten kamertemperatuur *measuredTemperature* en de gewenste kamertemperatuur *Setpoint*.

Deze PID regelaar heeft intern de volgende waarden:

$$KP = 10 [-]$$

$$KI = 0.02 [1/s]$$

$$KD = 0 [s]$$

Omdat de laatste waarde 0 is wordt de D-actie dus eigenlijk niet gebruikt (ik twijfel eigenlijk of ik die wel goed heb geïmplementeerd in de software als ik hem aan zou zetten).

Er is een anti-windup maatregel ([https://en.wikipedia.org/wiki/Integral\\_windup](https://en.wikipedia.org/wiki/Integral_windup)) op de I-actie gezet om te voorkomen dat deze heel hoog wordt als het setpoint veel hoger of lager is dan de gemeten temperatuur. Deze anti-windup betekent dat de I-actie niet verder mag oplopen als de uitgestuurde waarde al aan het einde van het regelbereik zit.

In de logfile zie je ook kolommen P, I en D. Zoals je in het schema kunt zien kun je de P, I en D waarden bij elkaar optellen om de *requestedBoilerTemperature* uit te rekenen. Dit is de

gevraagde temperatuur van het water dat door de radiatoren stroomt. Het blokje wat er nog tussen zit zorgt ervoor dat de waarde niet beneden *outputLow* (=10°C) kan komen, en ook niet boven *outputHigh* (=80°C).

Omdat ik zeker wilde weten of de verwarming 's nachts wel echt uit gaat heb ik ook geïmplementeerd dat *enableCentralHeating* op 'false' wordt gezet als de *Setpoint* temperatuur minimaal 2°C lager is dan de *measuredTemperature*. Met andere woorden: als je 's avonds naar bed gaat, gaat het setpoint naar 10°C maar is het in de kamer nog 21°C. Het verschil is dan groter dan 2°C waardoor *enableCentralHeating* uit gaat. Dit had ik de eerste dagen nog niet geïmplementeerd dus het kan zijn dat dit de laatste twee dagen ander gedrag oplevert.

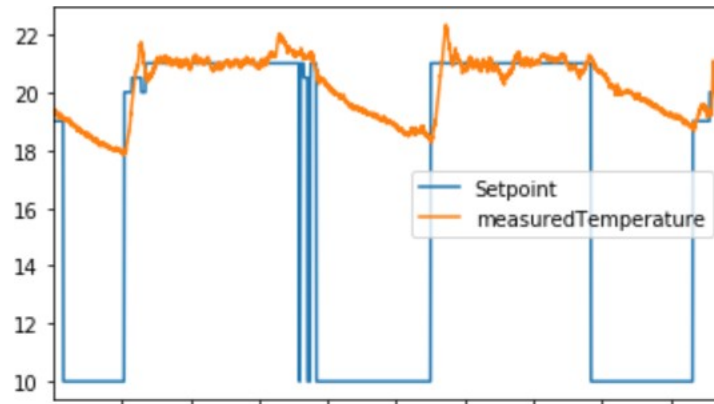
Met *flameOn* rapporteert de verwarmingsketel terug of de vlam van de brander wel of niet brandt. Verder meldt de ketel ook de echte temperatuur *actualBoilerTemperature* van het water terug.

Het warme water van de douche is apart geregeld. Ik stuur *enableHotwater* naar de ketel om aan te geven dat hij ook de douche van warm water mag voorzien (deze waarde is altijd *true*). Verder is er *domesticHotWaterSetpoint*. Dit is de gewenste waarde voor de watertemperatuur voor de douche. Deze stond in de eerste versie op 40°C, maar omdat Eliza toen een koud bad had heb ik die waarde verhoogd naar 55°C. De ketel rapporteert terug of *isHotWaterActive* wordt gebruikt (Y/N) en wat de werkelijke temperatuur van het warme water *actualHotWaterTemperature* is.

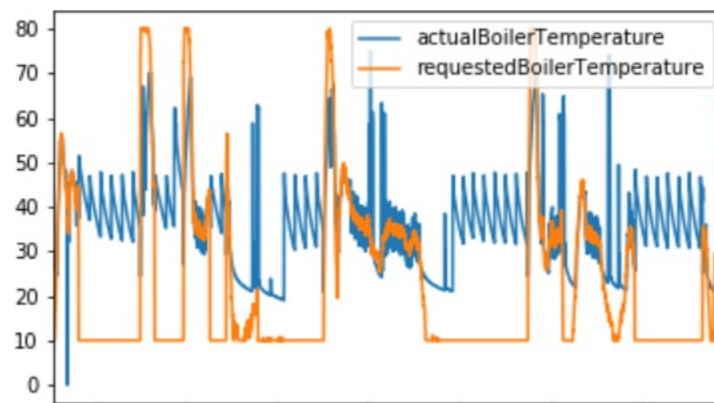
# Dingen waar ik benieuwd naar ben

## *Gaat de kachel 's nachts wel uit?*

Het verbaast me een beetje dat de *measuredTemperature* in de woonkamer 's nachts slechts tot 18°C lijkt af te koelen. Ik had verwacht dat het veel kouder zou worden. Staat de kachel echt uit?



Hoewel ik 's nachts *enableCentralHeating* uit zet (in de eerste dagen niet, heb ik later veranderd, maar ik weet niet of je de invloed daarvan op de dataset kan zien omdat *enableCentralHeating* zelf niet wordt gelogd) en *requestedBoilerTemperature* ook op 10°C staat, schommelt de *actualBoilerTemperature* tussen 30°C en 50°C. Gaat de ketel dan toch aan 's nachts?



Hier zie je bijvoorbeeld dat de watertemperatuur midden in de nacht in ene 17°C opwarmt terwijl *isFlameOn* N blijft:

Time	requestedBo	actualBoilerT	actualHotWa	isCentralHea	isHotWaterA	isFlameOn
03:17:34	10	31.098	0	N	N	N
03:18:35	10	31	0	N	N	N
03:19:37	10	31	0	N	N	N
03:20:38	10	30.797	0	N	N	N
03:21:39	10	30.699	0	N	N	N
03:22:40	10	47.559	0	N	N	N
03:23:41	10	46.199	0	N	N	N
03:24:42	10	45.5	0	N	N	N
03:25:43	10	45.098	0	N	N	N
03:26:43	10	44.5	0	N	N	N
03:27:45	10	44.199	0	N	N	N
03:28:46	10	43.898	0	N	N	N
03:29:47	10	43.598	0	N	N	N
03:30:49	10	43.398	0	N	N	N
03:31:49	10	43	0	N	N	N

Misschien gaat de vlam maar heel kort branden als de temperatuur onder de 30°C zakt maar is de vlam weer uit als de volgende regel in de dataset wordt toegevoegd. Mogelijk is die 30°C een instelling op de ketel zelf waar ik me niet van bewust ben. Het bijzondere is dan wel dat de *actualBoilerTemperature* soms wel naar 20°C daalt.

Ik heb de handleiding van de Remeha Avanti in de readonly directory gezet maar kon zo snel niets vinden.

### *Kunnen we aan de dataset zien wat ons gasverbruik is?*

Ik heb een tweede dataset *gasverbruik.txt* toegevoegd met ons gasverbruik in m<sup>3</sup> per dag. Dit heb ik overgetypt uit de engie app.

Het lijkt er op dat ons gasverbruik de laatste tijd niet omhoog is gegaan, dus dat is goed nieuws.

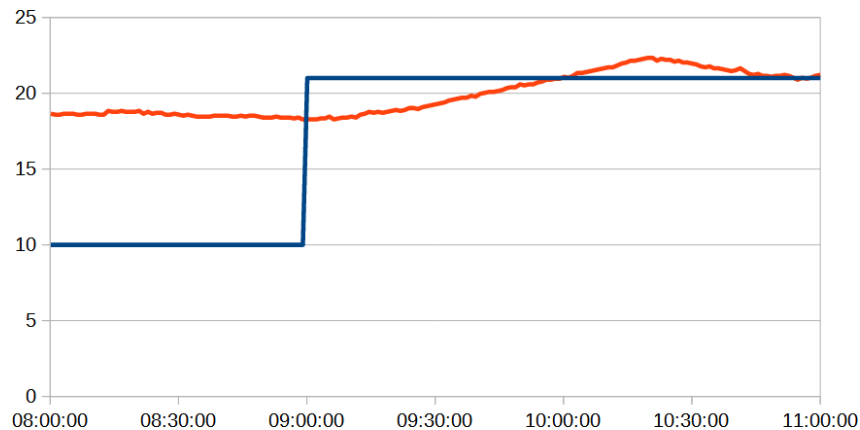
Ik kan me voorstellen dat dit correleert met:

- de integraal van het positieve deel van  $d(\text{actualBoilerTemperature})/dt$  (de ketel stookt alleen als dit een positief getal is, je levert geen gas terug als de temperatuur daalt)
- het aantal minuten per dag dat *isHotWaterActive* aan staat

Het is misschien wel lastig om met zo weinig data en correlatie vast te stellen omdat het gasverbruik per dag niet zo heel sterk varieert.

## Is de regelaar optimaal?

Als ik naar dit stukje grafiek kijk:



Zie je:

- dat *measuredTemperature* pas na een uur de juiste waarde heeft bereikt
- er vervolgens 1.3°C overshoot is
- de temperatuur verder heel nauwkeurig wordt geregeld

Dit kan vast beter op basis van de beschikbare meetwaarden. Ik denk aan:

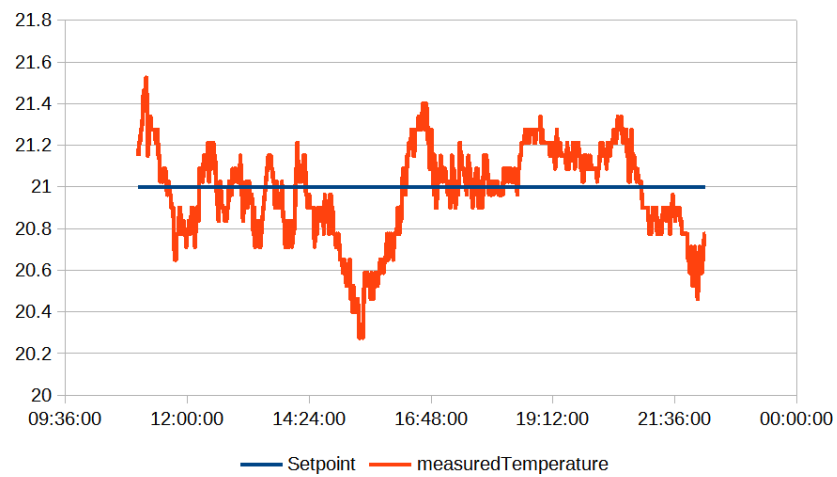
- betere instellingen van P, I en D
- misschien een ander soort regelaar (machinelearning in plaats van PID?)
- een lerend systeem die bijvoorbeeld bijhoudt dat de verwarming alvast een uur van tevoren moet beginnen

Bij al die dingen is het wel belangrijk om te voorkomen dat het gasverbruik omhoog gaat. Ik wil ook oppassen dat we geen verkeerde waarden naar de ketel sturen waardoor de levensduur achteruit zou kunnen gaan.

Een verkeerd ingestelde PID regelaar kan gaan opslingeren waardoor het minder aangenaam in hui wordt en het gasverbruik zeker omhoog zal gaan.

## Kan het gasverbruik omlaag?

Ik weet ook niet zeker of de regelaar gedurende de dag te agressief is afgesteld. Over de dag heen wijkt de temperatuur maximaal ongeveer een halve graad af van het setpoint.



Stel dat we de regelaar iets soepeler afstellen, zou dat invloed hebben op het gasverbruik?

Iets anders is of we *measuredTemperature* eigenlijk nog willen filteren. Nu zit er wat ruis op dit signaal, ik weet eigenlijk niet of dit erg is.

### *Bijlage: werkende import van de logdata*

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime

df = pd.read_csv('Logdata.txt', sep='\t', parse_dates=[['Date', 'Time']],
index_col='Date_Time')
```