Config und Logging

Config

Die config.py ist eine Zusammenfassung aller wichtigen Funktionsvariablen der verwendeten Programme im HAL.

Dabei handelt es sich vor allem um die Updateintervalle der Wrapper und den Messtoleranzen der CurrentCost Geräte.

```
# -*- coding: utf-8 -*-

import logging
import logging.config

import yaml

config = yaml.load(open('logging.conf', 'r'))
logging.config.dictConfig(config)

hwcontrol_settings = {
       }

arduino_settings = {
       'serial': {
           'port': '/dev/ttyACM0',
           'timeout': 6
           },
       'update_interval': 5,
       'wait': 0.5,
       'timeout': 30
       }

currentcost_settings = {
       'serial': {
           'port': '/dev/ttyUSB0',
           'baudrate': 57600,
           'timeout': 0
           },
       'update_interval': 5,
       'wait': 0.5,
       'timeout': 30
       }

googleweather_settings = {
       'language': 'de',
       'update_interval': 5
       }

homematic_settings = {
       'server': 'http://192.168.0.30:2001',
       'update_interval': 15,
       # False -> schneller
       'check_ops': False
       }

twitter_settings = {
       'update_interval': 120
       }

# Klassennamen als Strings
```

```
used_wrappers = ['HomeMatic', 'CurrentCost'] #'Arduino'

# Format: dev__subdev__tag
tolerance = {
       # Arduino
       'Arduino__Arduino__temp': 0.2,
       # CurrentCost
       'CurrentCost__tmpr__temp' : 0.2,
       # sensor 0
       'CurrentCost__0_tmpr__temp': 0.2,
       'CurrentCost__0_ch1__power': 2,
       'CurrentCost__0_ch2__power': 2,
       'CurrentCost__0_ch3__power': 2,
       'CurrentCost__0_ch4__power': 2,
       'CurrentCost__0_ch5__power': 2,
       'CurrentCost__0_ch5__power': 2,
       'CurrentCost__0_ch6__power': 2,
       'CurrentCost__0_ch7__power': 2,
       'CurrentCost__0_ch8__power': 2,
       'CurrentCost__0_ch9__power': 2,
       # sensor 1
       'CurrentCost__1_tmpr__temp': 0.2,
       'CurrentCost__1_ch1__power': 2,
       'CurrentCost__1_ch2__power': 2,
       'CurrentCost__1_ch3__power': 2,
       'CurrentCost__1_ch4__power': 2,
       'CurrentCost__1_ch5__power': 2,
       'CurrentCost__1_ch5__power': 2,
       'CurrentCost__1_ch6__power': 2,
       'CurrentCost__1_ch7__power': 2,
       'CurrentCost__1_ch8__power': 2,
       'CurrentCost__1_ch9__power': 2,
       # sensor 2
       'CurrentCost__2_tmpr__temp': 0.2,
       'CurrentCost__2_ch1__power': 2,
       'CurrentCost__2_ch2__power': 2,
       'CurrentCost__2_ch3__power': 2,
       'CurrentCost__2_ch4__power': 2,
       'CurrentCost__2_ch5__power': 2,
       'CurrentCost__2_ch5__power': 2,
       'CurrentCost__2_ch6__power': 2,
       'CurrentCost__2_ch7__power': 2,
       'CurrentCost__2_ch8__power': 2,
       'CurrentCost__2_ch9__power': 2,
       # sensor 3
       'CurrentCost__3_tmpr__temp': 0.2,
       'CurrentCost__3_ch1__power': 2,
       'CurrentCost__3_ch2__power': 2,
       'CurrentCost__3_ch3__power': 2,


       'CurrentCost__3_ch4__power': 2,
       'CurrentCost__3_ch5__power': 2,
       'CurrentCost__3_ch5__power': 2,
       'CurrentCost__3_ch6__power': 2,
       'CurrentCost__3_ch7__power': 2,
       'CurrentCost__3_ch8__power': 2,
       'CurrentCost__3_ch9__power': 2,
       # sensor 4
       'CurrentCost__4_tmpr__temp': 0.2,
       'CurrentCost__4_ch1__power': 2,
       'CurrentCost__4_ch2__power': 2,
       'CurrentCost__4_ch3__power': 2,
       'CurrentCost__4_ch4__power': 2,
       'CurrentCost__4_ch5__power': 2,
       'CurrentCost__4_ch5__power': 2,
       'CurrentCost__4_ch6__power': 2,
       'CurrentCost__4_ch7__power': 2,
       'CurrentCost__4_ch8__power': 2,
       'CurrentCost__4_ch9__power': 2,
       # GoogleWeather
       #'GoogleWeather_GoogleWeather_condition': 0,
       'GoogleWeather__GoogleWeather__temp_c': 0.2,
       #'GoogleWeather__GoogleWeather__wind_condition': 5,
       'GoogleWeather__GoogleWeather__humidity': 10
       # HomeMatic
       #'HomeMatic__IEQ00070561__state': 0
}
db_daemon_settings = {
       'sleep_durotation' : 0.5,
       'CC_count' : 1,
       'default_count' : 5,
       'max_count' : 10,
}

# Settings für den rules_daemon
rules_daemon_settings = {
       'sleep_durotation' : 10,
}

# vim: set sts=4 sw=4 et:
```

Logging

Zum Protokollieren des Kontrollflusses wird das standardmäßige Modul logging verwendet.

Logs werden in folgenden Kategorien anhand des Loglevels gespeichert:

- DEBUG: Variablendaten welche beim Debugging hilfreich sind.
- INFO: Zwischennachrichten, bei planmäßiger Funktion.
- ERROR: Exception wurde geworfen, massiver Funktionsfehler

Lognachrichtformat: Datum (Jahr-Monat-Tag) - Uhrzeit - Datei:Zeile - Loglevel - Nachricht
Loglevel:ERROR

Das Logging wird dabei in eine Log-Datei “logging.log” gespeichert und beinhaltet.

Das gewählte Format ist dabei YAML.

```
version: 1
formatters:
 standard:
   format: '%(asctime)s - %(name)s:%(lineno)d - %(levelname)s - %(message)s'
root:
 class: logging.FileHandler
 level: ERROR
 formatter: standard
 filename: logging.log
handlers:
 console:
   class: logging.StreamHandler
   level: ERROR
   formatter: standard
   stream: ext://sys.stdout
 file:
   class: logging.FileHandler
   filename: logging.log
   level: ERROR
   formatter: standard
 file_wrapper:
   class: logging.FileHandler
   filename: logging.log
   level: ERROR
   formatter: standard
loggers:
 hwcontrol:
   handlers: [file]
   propagation: no
 utils:
   handlers: [file]
   propagation: no
 hwcontrol_dummy:
   handlers: [file]
   propagation: no
 wrapper.arduino:
   handlers: [file_wrapper]
   propagation: no
 wrapper.currentcost:
   handlers: [file_wrapper]
   propagation: no
 wrapper.googleweather:
   handlers: [file_wrapper]
   propagation: no
 wrapper.homematic:
   handlers: [file_wrapper]
   propagation: no
 wrapper.twitter:
   handlers: [file_wrapper]
   propagation: no
 wrapper.facebook:
   handlers: [file_wrapper]
   propagation: no
```