

# libhvac ja hvacServer

Kirjasto sisältää kiinteistöautomaatiossa paljon käytettyjä toimintoja, kuten PID-säädin, viikkokello- ja kalenteriohjauksen, viiveajastimet ja säätökäyrät.

Kirjasto käynnistää hvacServer prosessin, joka suorittaa pistetietokannassa operaatiot kirjaston luomille pistetyypeille.

## hvacPIDController

Tämä pistetyyppi toimii PID säätimenä. Säätimeen voi määritellä haluamansa määrän säätöportaita, ja jokaiselle voi määrittää haluamansa suhdealueen, sekä lähdön minimin ja maksimin, sekä määrittää sen joko säätösuunnaltaan suoraksi tai käännettyksi (jolloin minimi ja maksimi toimivat nurinpäin).

Jos säätimessä on monia portaita käytössä, ja varsinkin tilanteessa jossa osa portaista on invertoitu, pitää ohjelmoijan muistaa asettaa säätimen **bias** kenttä sopivaan arvoon. Kyseinen kenttä määrittää pisteen, jossa säätimen portaiden lähdöt ovat käynnistyshetkellä. Haluttu **bias**-piste voidaan hakea laskemalla portaiden lähtöjen summa haluttuun käynnistyspisteeseen.

Jos säätimessä on kaksi porrasta, ja niiden lähdöt ovat alueella 0% – 100%. Säätimen ensimmäisen portaan saa puoliksi auki antamalla bias arvoksi 50, tai täysin auki antamalla bias-arvoksi 100. Jos myös seuraava porras halutaan vaikkapa 30 % auki, lasketaan pohjalle 1. portaan täysi säätövara (100%) ja sen jälkeen 2. portaan haluttu ohjaus eli 30 %. Tuloksena saadaan bias-arvo 130 (%).

Sama logiikka toimii useampienkin portaiden kanssa. Portaan vaikutus bias-arvoa laskettaessa on minimi-asennon ja maksimi-asennon erotus.

Koska säädin lähtee normaalisti käyntiin tilanteesta jossa kaikki portaavat ovat minimissä (pois lukien P-termin vaikutus), ja koska portaan **invertointi** kääntää portaan lähdön toiminnan, ovat invertoidut portaan tästä johtuen säätimen käynnistyessä usein lähellä maksimiarvoa. Tämä on useimmiten ei toivottu ilmiö, ja se saadaan korjattua laskemalla sopiva bias-arvo, jonka avulla säädin käynnistyykin esimerkiksi ensimmäisen säätöportaan jälkeen, jolloin bias arvoksi tulee yksinkertaisesti ensimmäisen säätöportaan lähdön maksimi – tarkemmin sanottuna ensimmäisen portaan maksimin ja minimin erotus.

Pisteen kentät:

**bias** Säätimen toimintapiste käynnityksessä.

**const\_I** Integrointiaika sekunteina (viritysparametri).

**const\_D** Derivoinnin vahvistus (viritysparametri)

<b>deadZone</b>	Integroinnin kuollut alue.
<b>input</b>	Säätimen takaisinkytkentätieto, eli mittausarvo.
<b>inputId</b>	Tietokantapiste, josta säätimen mittaustieto luetaan. Haluttaessa voidaan jättää tyhjäksi, ja kirjoittaa input kenttä vaikkapa ohjelmassa.
<b>enabled</b>	Säätimen käyntilupa. Arvossa 1 säädin algorimia suoritetaan, ja arvossa 0 säätimen lähdöt asetetaan default arvoihin.
<b>enabledId</b>	Tietokantapiste josta säätimen käyntilupa haetaan. Voidaan jättää tyhjäksi, ja kirjoittaa enabled kenttää esimerkiksi ohjelmassa.
<b>setpoint</b>	Säätimen asetusarvo, johon input arvoa koetetaan säätää ohjaamalla säätöportaita.
<b>stages</b>	Säätöportaat (taulukko)
<b>default</b>	Portaan oletusarvo, joka asetetaan portaan output kenttään kun säädin ei ole käynnissä.
<b>direction</b>	Säätöportaan suunta. Direct tarkoittaa että säätöportaan arvoa kasvatetaan kun mittausarvo on pienempi kuin asetusarvo. Inverted porras toimii päinvastoin.
<b>name</b>	Säätöportaan nimi (tiedoksi ihmisille)
<b>outMax</b>	Säätöportaan maksimiarvo.
<b>outMin</b>	Säätöportaan minimiarvo.
<b>output</b>	Säätöportaan tämän hetkinen oloarvo.
<b>outputId</b>	Tietokantapiste johon output arvo kirjoitetaan (voi jättää tyhjäksi ja lukea esimerkiksi ohjelmassa).
<b>pBand</b>	Säätöportaan suhdealue.

**hvacSchedule** Aikaohjelma, eli viikkokello johon voidaan liittää poikkeuspäiväkalenteri.

Viikkokellolle on ensiksi lähdön mahdolliset tilat. Tämä tehdään antamalla niille nimet **stateTexts** kentässä. Ne voivat olla esimerkiksi "pois" ja "päällä". Taulukon järjestyksessä ensimmäinen tila vastaa lähdön tilaa 0, seuraava rivi tilaa 1 jne. Sen jälkeen **weeklyEvents** kentän kautta voidaan lisätä tapahtumia viikon eri päiville vapaavalintainen määrä, joiden perusteella aikaohjelman **pv** kenttä vaihtaa arvoaan. Pisteiden **calendar** kenttään voidaan määrittää poikkeuspäiväkalenteri, jonka tapahtumat ajavat ylitse viikkokellon omista tapahtumista.

## hvacCalendar

Kalenteriohjelma, jota kutsutaan aikaohjelmien yhteydessä poikkeuspäiväkalenteriksi. Se on toisaalta tarkoitettu käytettäväksi aikaohjelmien yhteydessä, mutta sitä voidaan käyttää aivan hyvin myös ilman aikaohjelmaa lua ohjelmista käsin.

**stateText** kenttään tulisi ensiksi syöttää nimet kaikille kalenteriohjelman lähdön arvoille. sein riittää kaksi arvoa esim. pois ja päällä, tai seis ja käy. Sitten **dateList** kentän kautta voidaan lisätä ensiksi haluttuja ajanjaksoja kalenteriin, joiden mukaan kalenterin pv kentän arvoa vaihdetaan.

Tapahtuma kalenteri:

Jokaisella **dateList** taulukkoon lisätyllä jaksolla on seuraavat arvot.

**startDate** tarkoittaa mistä päivämäärästä jakso alkaa.

**duration** tarkoittaa montako päivää tätä jaksoa seurataan.

**event** on kyseisellä jaksolla noudatettavat päivittäiset tapahtumat (mihin aikaa lähtö menee päälle, ja milloin taas pois).

## hvacCurve

Vapaasti määriteltävä säätökäyrä, jonka **input** kenttään voidaan kirjoittaa lukuarvo, ja lukea sitä vastaava **points** kentässä annetun käyrän arvo **pv** kentästä. **y\_title** ja **x\_title** kenttiin voidaan antaa käyrän vastaavien akseleiden nimet, joka näytetään puolestaan käyttöliittymässä. Pistettä on mahdollista käyttää n.s. käsin lua-ohjelmassa, tai antaa **inputId** kenttään haluttu pistetunnus, jonka arvo viedään käyrän kautta hvacCurve-pisteen **pv** kenttään.

## hvacTimer

Viiveajastin, jota voidaan käyttää joko lua-ohjelmassa, tai suoraan pistetietokannassa.

Halutus viiveet määritellään pisteen onDelay ja offDelay kenttiin. Viiveiden yksikkö on millisekunti, joten arvo 1000 vastaa 1 sekuntia.

Tämän jälkeen, kun pisteen input kenttään kirjoitetaan jokin arvo, se siirtyy ajastimen output kenttään annettujen viiveiden jälkeen. Jos input arvossa tapahtuu pudotus (laskeva reuna) käytetään offDelay arvoa, ja jos input arvo kasvaa (nouseva reuna) käytetään onDelay arvoa. Pisteen **pv** kentästä voi lukea ajastimen kulloisenkin tilan: 0 tarkoittaa, että mikään viive ei ole kesken. 1 tarkoittaa että **offDelay** on kesken, ja arvo 2 tarkoittaa että **onDelay** juoksee.

Lisätoimintoja voidaan antaa **stepSize** parametri. Sen avulla määrittää kuinka suurissa askeleissa output kentän arvo muuttuu kohti **input** kentän arvoa jokaisen toteutuneen viivejakson jälkeen.

Jos `stepSize` kenttään laittaa vaikkapa 0.2 tai 0.5 ja `offDelay` ja `onDelay` arvoiksi vaikkapa 1, saadaan ajastimella toteutettua liukumia kahden arvon välillä, koska `output` arvo muuttuu enintään `stepSize` kentän arvon verran yhdessä noin yhden sekunnin aikana.

---

Revision #1

Created 25 May 2022 11:39:32 by Severi Hiltunen

Updated 10 May 2023 10:25:34 by Severi Hiltunen