

TALLINNA POLÜTEHNIKUM

Päevane osakond

Viktor Reinok

AA - 06

EC – 12M KÜTTEVEE REGULAATORI UUENDAMINE

SOOJUSPROTSESSIDE AUTOMATISEERIMINE

ÕPPEAINE KURSusetöö

Juhendajad: Kalle Lindvere

Tallinn 2009

SISUKORD

1. SEADME OTSTARVE JA KASUTUSALAD	4
1.1 Seade	4
1.2 Seadme tööpõhimõte.....	4
2. SEADME EHITUS JA PAIGALDUS.....	5
2.1 Seadme muutmine: Mis jääb ... Mis muutub ?.....	5
2.2 Seadme ventiiliajam (jõuosa)	5
2.3 Seadme paigaldamine süsteemi (lihtsustatud)	6
2.4 Regulaatori EC-12M kasutamine koos õli-, gaasi-või elekterküttel töötava keskküttekatalaga	7
2.5 Regulaatori EC-12M kasutamine koos tahkel kütusel töötava keskküttekatalaga	8
2.6 Regulaatori EC-12M kasutamine kaugkütte võrku ühendatud majades.....	10
2.7 Regulaatori EC-12M kasutamine kaugkütte võrku (koos soojusvahetiga) ühendatud majades.	11
2.8 Regulaatori EC-12M kasutamine koos pörandaküttega.....	12
3. SEADME TEHNILISED PARAMEETRID.....	13
3.1 Jõuosa tehnilised andmed	13
3.2 Jõuosa mõõdud ja kujundus.....	13
4. EC – 12 REGULAATORI KÜTTEGRAAFIK	14
4.1 Mis see on ?.....	14
4.2 Milleks on vaja küttegraafikut ?	14
5. EC – 12 KÜTTEVEE REGULAATORI KÜTTEGRAAFIKU PARAMEETRID	15
5.1 Graafiku valik	15
5.2 Graafiku nihe	16
5.3 Piiramine.....	17
5.4 Küttevee temperatuuri tõus 0 c kraadi juures	18
6. KÜTTEGRAAFIKU PROGRAMMIKOOD KOOS PARAMEETRITEGA	18
6.1 Graafiku funktsiooni programmikood	19
6.1.1 Programmikoodi graafiline kujundus	21
6.2 Programmikoodi väljund	22
6.2.1 Programmikoodi väljundi graafiline kujutus nr. 1	23
6.2.2 Programmikoodi väljundi graafiline kujutus nr. 2	24
7. REGULEERIMIS PROGRAMMI BLOKKSKEEM	25
7.1 Blokk skeem.....	25
7.1 Blokk skeemi selgitus.....	26
7.2 Skeemi eesmärk	26
8. EC – 12 KÜTTESÜSTEEMI TOA- JA VÄLISTEMPERATUURI ANDUR	27

8.1 Arvutused	28
8.2 PIC mikrokontroller-mooduli analoog - digitaal muundur (ADC)	29
8.3 Voolugeneraator anduri ja PIC-i vahel	29
9. PIC18 SEERIA PROGRAMEERIMISTARKVARA CSS 4.093 MEETRIKUD.....	30
9.1 Kuidas kasutada täisarvu murdarvuna ?	30
9.2 Arvtüüpide kokkuvõtte.....	31
9.3 Näidis programmikood	31
9.4 Programmikoodi seletus ja kirjeldus	32
9.5 Programmikoodi näite tulemus.....	32

1. SEADME OTSTARVE JA KASUTUSALAD

1.1 Seade

Seadmeks on kütteeve regulaatori EC - 12M uuendatud (uuendamisel) mudel. See tähendab seda, et seadmel oleksid kõik vana seadme parimad omadused ning, mõned uued juba seadmesse integreeritud funktsioonid nagu programmkella funktsioon, arvutada ühenduvuse olemasolu ja võimalike rikete või vigade kuvamise võimalus.

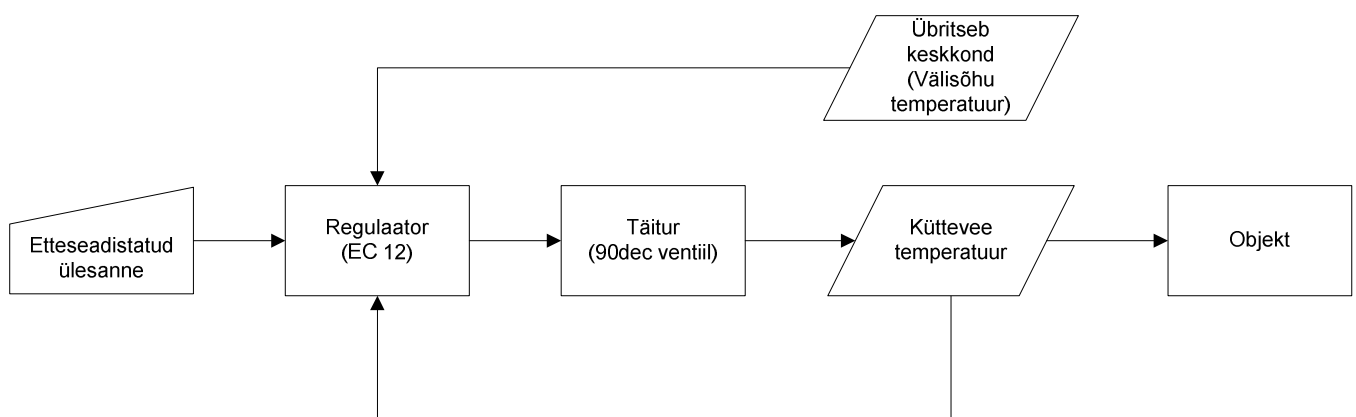
1.2 Seadme tööpõhimõte

Seade, keskkütte süsteemi kütteeve regulaator EC – 12M, eesmärk on toatemperatuuri hoidmine etteantud väärtusel sõltumatta välistemperatuuri muutustest.

Seade reguleerib objekti soojusvahetite süsteemi sissetuleva vee hulka reguleerides ventiili asendit muutes. Ehk siis seade on paigaldatud enne tsirkulatsioonipumpa ja kas segamistoru külge või ette, selle asendi eelisega saab reguleerida sissevoolva vee soojushulka.

Seade muudab ventiili asendit võrreldes küttegaafiku tahetavat kütteeve temperatuuri reaalse kütteeve temperatuuriga andurilt. Kuna seade reguleerib inertseid süsteeme siis tulemuse hilistuse kompenseerime toimub PID reguleerimis meetodi kasutusel.

Kasutuselana võib olla hoonde keskkütte süsteem. Eesmärk on teha toatemperatuur stabiilseks ja kütmine efektiivsemaks.



Illustratsioon nr. 1.2 (süsteemi tööpõhimõte)

2. SEADME EHITUS JA PAIGALDUS

2.1 Seadme muutmine: Mis jääb ... Mis muutub ?

Kuna tegu on vana seadme uuendamisega siis alles jääb:

- Seadme korpus
- Ventili asendi muutmise ajam (jõuosa)
- Jääb vana anduri tüüp

Muutub seadme juhtskeem ja kasutusliides. See asendub:

- PIC mikrokontrolleriga
- Keeratavate nuppude (potensiomeetrite) asememe tulvad nupud ja täht LCD
- Integreeritud reaajaja kell ja kalender (varem oli varaint see väliselt lisada)

2.2 Seadme ventiiliajam (jõuosa)



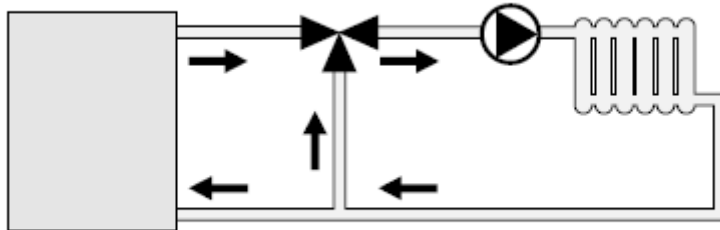
Illustratsioon (pilt) nr. 1.1

Pildid on kujutatud EC - 12M küttevee regulaator. See seade on juba valmis toodangu pilt, see on väga sarnane BELIMO tootega SM240, sest seda on kasutatud kui jõuosa. BELIMO SM240 on sisuliselt kolmejuhtmeline moriseeritud ventiili asendi muutja. EC -12M juhul seadme alus on seadme jõuosa mille sees on jõu ülekande süsteem. Selle aluse peal on plastmassis korpus. Korpuse sees on juhtskeem ja mootor, mis on ühendatud jõu ülekande süsteemiga.

2.3 Seadme paigaldamine süsteemi (lihtsustatud)

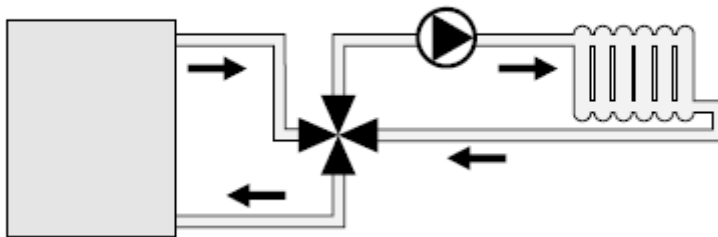
Seadme võib paigalda nii kolme kui ka nelja käigulisele ventiilile järgnevalt:

Kolmekäigulisele ventiiliga süsteem



Illustratsioon nr. 2.1

Neljakäigulise ventiiliga süsteem

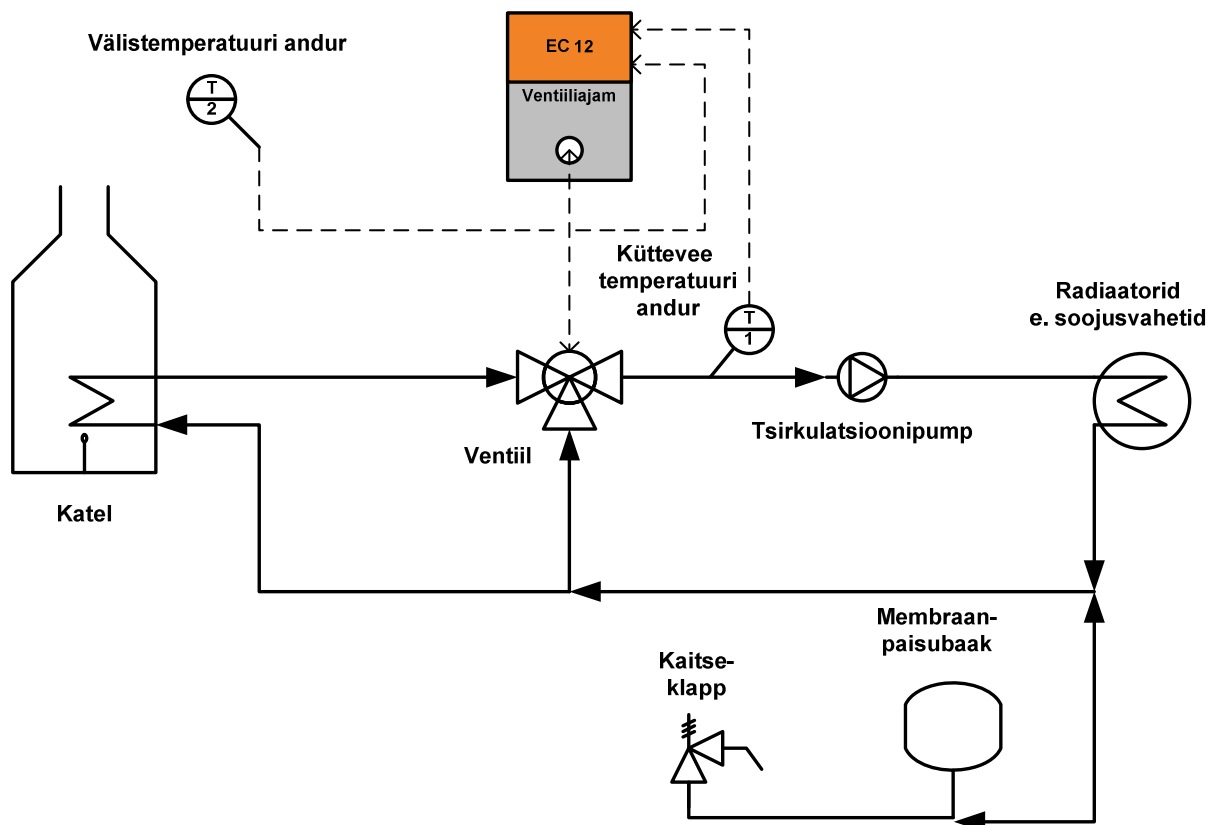


Illustratsioon nr. 2.2

Illustratsioonidel nr. 2.1 ja 2.2 on kujutatud lihtsustatud keskkütte süsteem. Hall ritkülik on katel või küttekeha, keskel on kujutatud ventiili (kas 3 või 4 käiguline ventiil), kolmnurk ringi sees on tsirkulatsiooni pump ja radiaatorid ehk soojusvahetid. Ainus asi mida seade muudab on ventiili asend.

2.4 Regulaatori EC-12M kasutamine koos õli-, gaasi-või elekterküttel töötava keskküttekatalaga

Paigaldamist kirjeldav joonis:



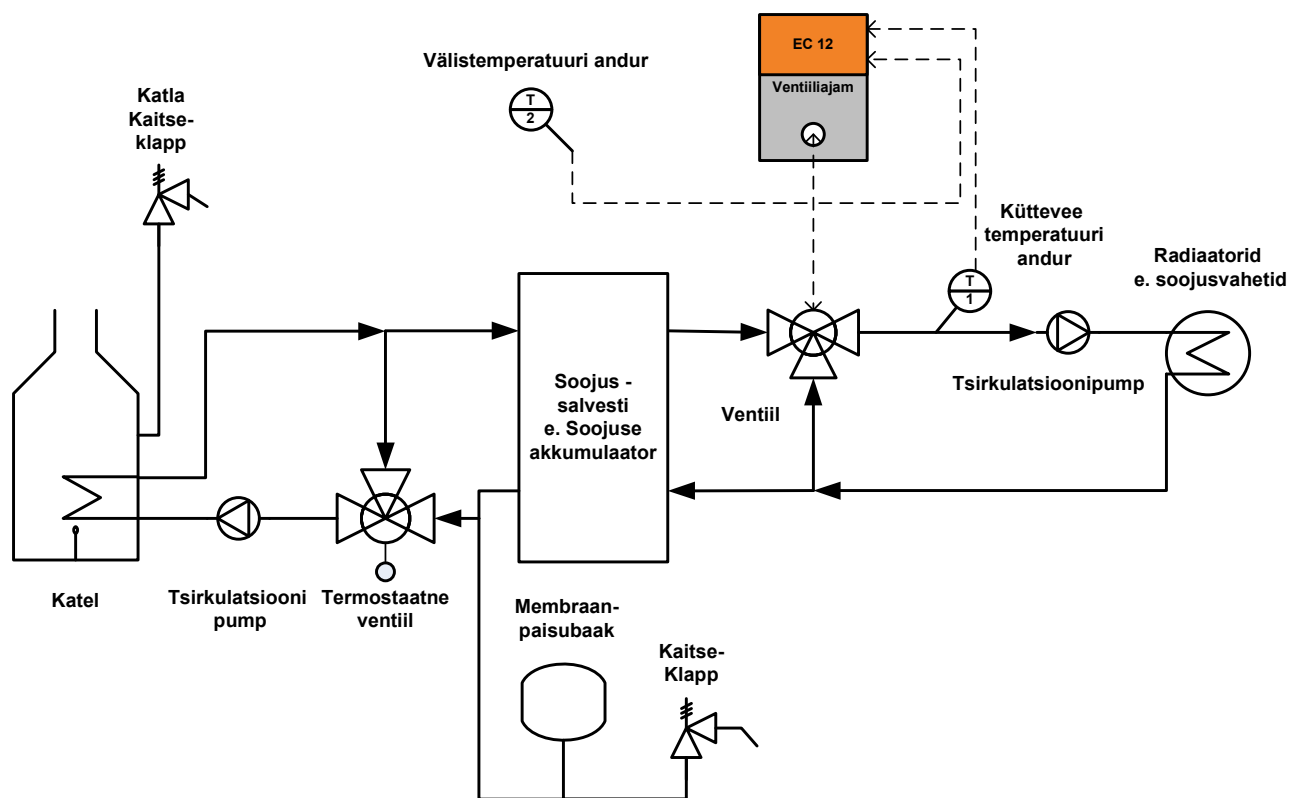
Vedel- või gaasikütte puhul on soovitatav kasutada automaatpõletit, elekterküttega katla puhul termostaatjuhtimisega küttekehi see selleks, et katla töö oleks stabiilsem ning katel suudaks koormuse muudatustele reageerida.

Kütteaine põlemiseks vajalik õhk sisaldab alati rohkem või vähem aurustunud olekus vett. Vedelal või gaasiküttel töötavatel kateldel tuleb katlavee temperatuuri hoida kastepunktist kõrgemal, et katla veesärgi välispindadel ei tekiks kondensaati. Soovitatav katlavee minimaalne temperatuur on +60 °C

Membraanpaisubaak kooskaitsekalpiga paigaldatakse rõhu madalamas punktis, mis antud joonise järgi on peale joojusvaheteid ehk radioaatoreid.

2.5 Regulaatori EC-12M kasutamine koos tahkel kütusel töötava keskküttekatalaga

Regulaatori paigaldamise joonis



Tahkel, (aga ka vedelal) kütusel töötavate katelde puhul on soovitatav kasutada soojussalvestit.

Soojussalvestiga keskkütte süsteem võimaldab maksimaalselt kasutada katelde võimsusi, saavutades sellega katelde töö kõrgeimat kasutegurit. Kütte-ja sooja tarbevee vajaduste järsu kõikumiste korral aitab soojussalvesti hoida keskkütte toitesüsteemi tööd sujuvana. Tahkel kütusel töötavate, eriti kütteainest gaasi välja põletamise põhimõttel töötavate kõrge kasuteguriga katelde puhul on soojussalvesti kasutamine vältimatu. Eiekterkütte puhul võimaldab soojussalvesti kasutada öist, odavamat elektrit.

Katelt on vaja kütta ainult siis, kui vee temperatuur soojussalvestis langeb vajalikust allapoole.

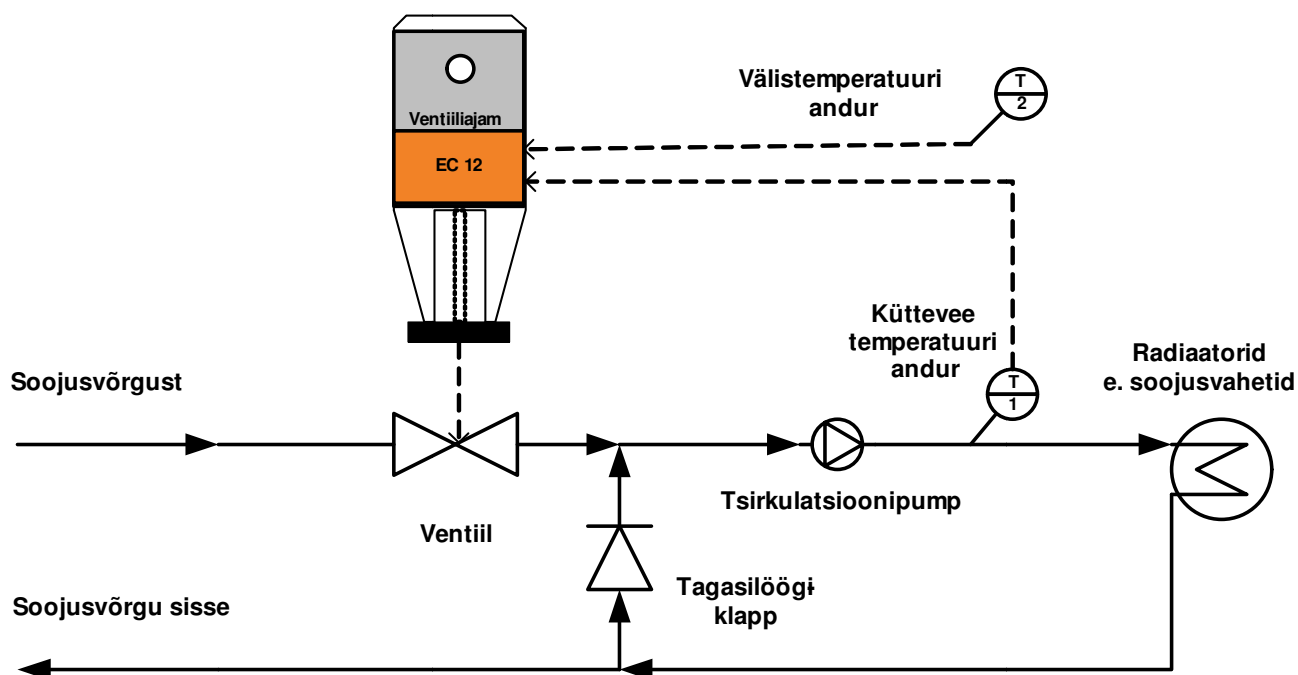
Termostaatiline ventiil takistab katla veesärgi liigset jahtumist katla kudemise ajal. Sooja tarbevee spiraalid võimaldavad saada sooja vett vahetult soojussalvestist. Katel-salvesti ringluspump peab töötama ainult siis, kui katel köeb. Salvesti katta soojusisolatsioonimaterjaliga.

Tahkel kütusel töötava katla kasutamise eeliseks on katla enada, kui ühekordse investeeringu odavam hind võrreldes õli- ja vedel-küttekateldega. Samuti on tahke küte ehk puuküte kõige odavam kütmise viis.

Tahke kütusel töötava katla kasutamisel on eelkõige puuduseks kasutajamugavus, kuna võrreldes õli- ja vedelküttekateldega tuleb tahket kütus käia töötavasse katlase lisamas.

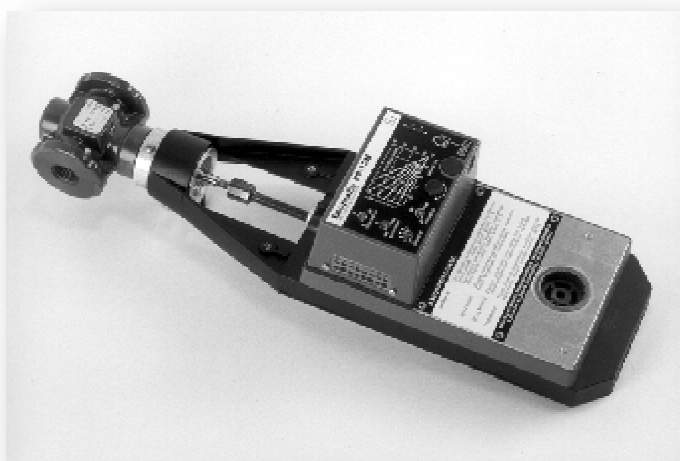
2.6 Regulaatori EC-12M kasutamine kaugkütte võrku ühendatud majades

Regulaatori paigaldamise joonis



Regulaator EC-12M koos ringluspumba ja tagasilöögiklapiga asendab elevaatorit.

Regulaator ühendatakse 2-käigulise ventiili külge spetsiaalse konsooli abil, et kasutada 2 käigulist surveventiili.

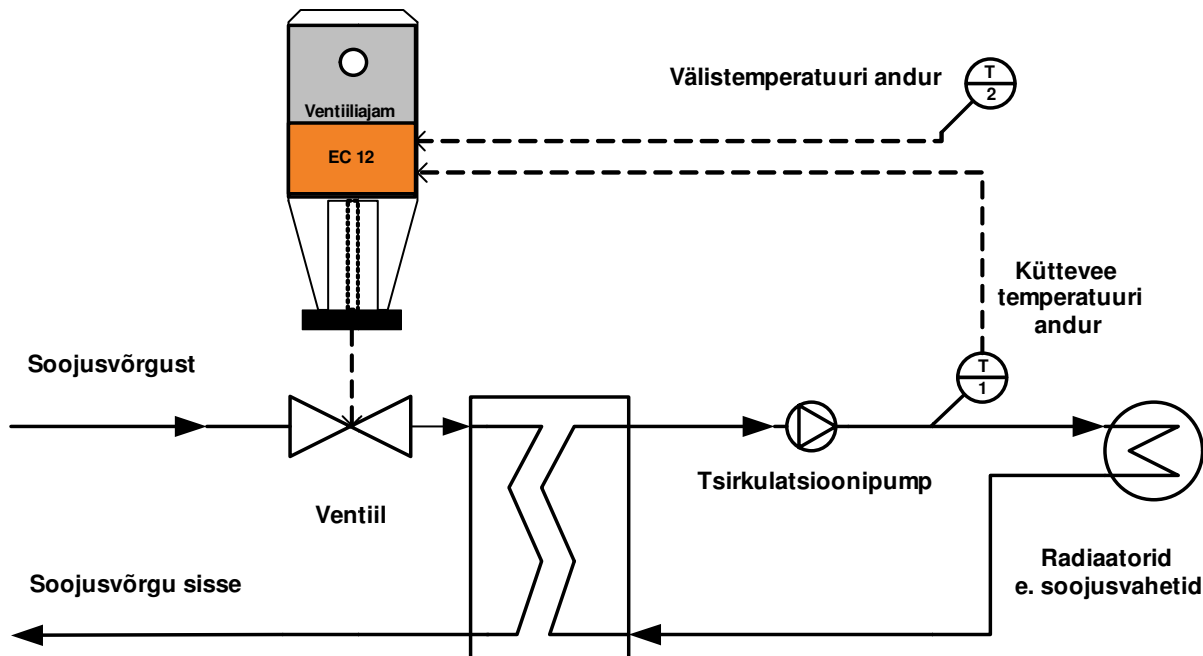


Pildid on konsool mille abil saab kasutada 2 käigulist ventiili

Enne regulaatori kasutamist kaugkütte võrku ühendatud majades tuleb konsulteerida regulaatori valmistajaga või müüjaga ning soojusvõrkude valdajaga, kuna regulaatori paigaldamine võib mõjutada soojusvõrgu töökäiku.

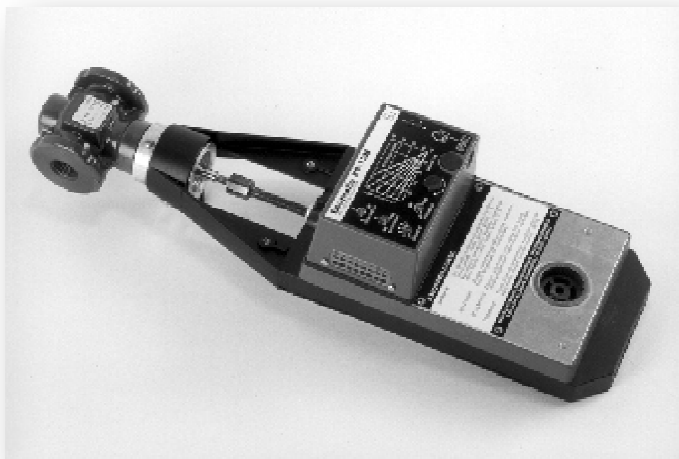
2.7 Regulaatori EC-12M kasutamine kaugkütte võrku (koos soojusvahetiga) ühendatud majades

Regulaatori paigaldamise joonis



Regulaator EC-12M koos ringluspumba ja soojusvahetiga asendab elevaatorit.

Regulaator ühendatakse 2-käigulise ventiili külge spetsiaalse konsooli abil, et kasutada 2 käigulist surveventiili.



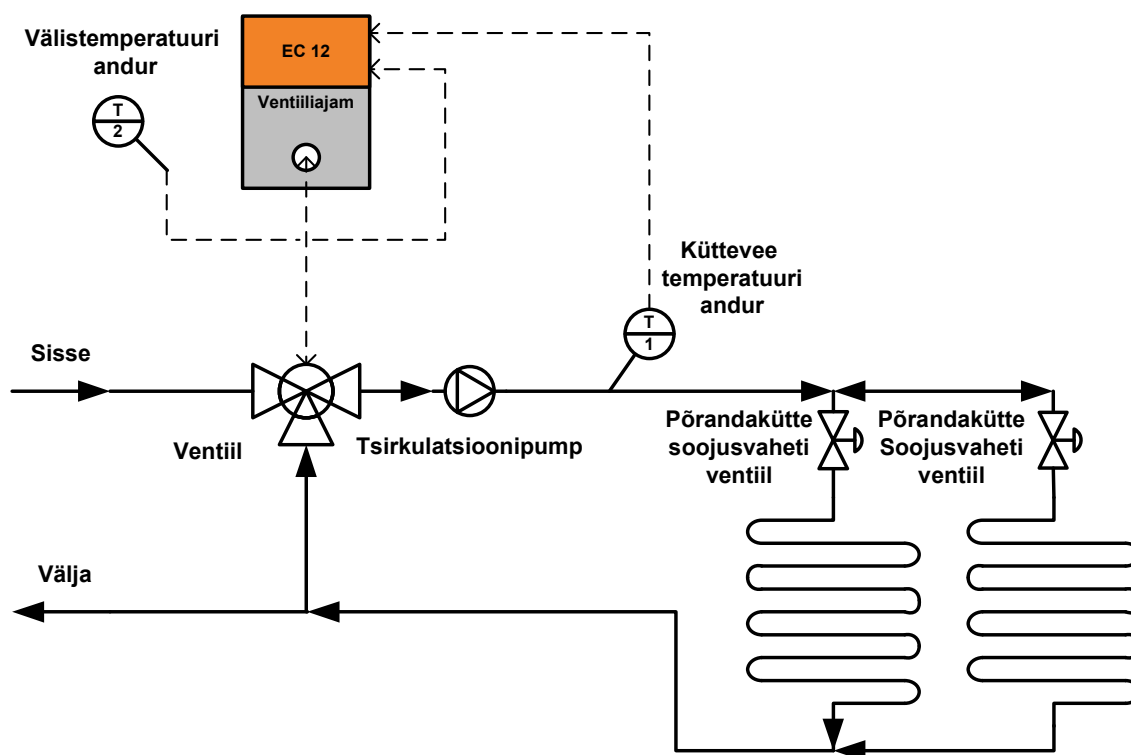
Pildid on konsool mille abil saab ka sutada 2 käigulist ventiili

Enne regulaatori kasutamist kaugkütte võrku ühendatud majades tuleb konsulteerida regulaatori valmistajaga või müüjaga ning soojusvõrkude valdajaga, kuna regulaatori paigaldamine võib mõjutada soojusvõrgu töökäiku.

Soojusvaheti kasutamine aitab tõsta süsteemi efektiivsust, juhul kui soojusvõrku tagasi suubuv torustik ei ole hästi soojustatud. Soojusvaheti aitab ära kasutada süsteemist väljuvat soojust, selle tõttu on soojus võrku tagasi mineva vee temperatuur madalam ja soojuse kasutus efektiivsem.

2.8 Regulaatori EC-12M kasutamine koos põrandaküttega

Regulaatori paigaldamise skeem



Joonisel on kujundatud põrandakütte süsteem kolmekäigulise ventiiliga ja tsirkulatsioonipumbaga. Iga põrandakütte toru (soojusvaheti) ees on ventiil millega on võimalik reguleerida sooja vee peale tulemise suhet põrandakütte soojusvahetite vahel.

Selle lahenduse puhul on nagunii vaja regulaatorit, sest suured temperatuuri kõikumised põrandakütte korral ei ole eriti soovitatavad (paisumine) ning maksimaalne temperatuur peab ka olema kindlates piirides. Samuti saab väga hästi kasutada regulaatori EC 12M programmeeritava kella funktsiooni. Just sellepärast on regulaator EC 12M väga hea lahendus põrandakütte juhtimiseks.

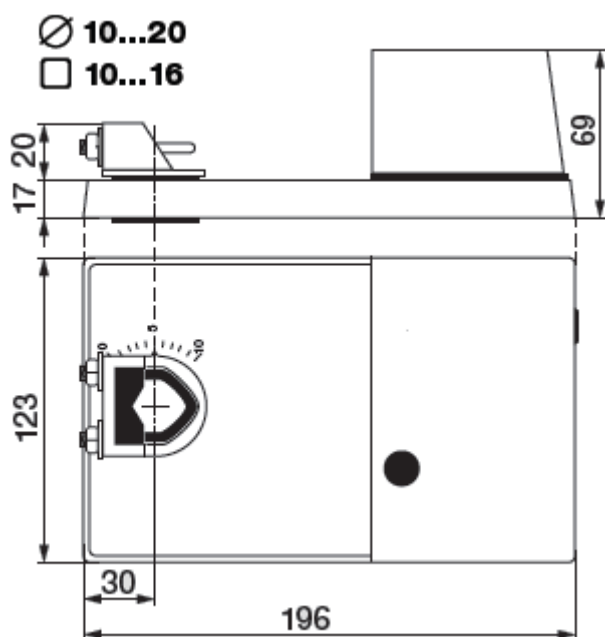
3. SEADME TEHNILISED PARAMEETRID

3.1 Jõuosa tehnilised andmed

Ventiili asendit muutev ajam jääb samaks (BELIMA SM240). Seetõttu selle parameetrid on kõige tähtsamad.

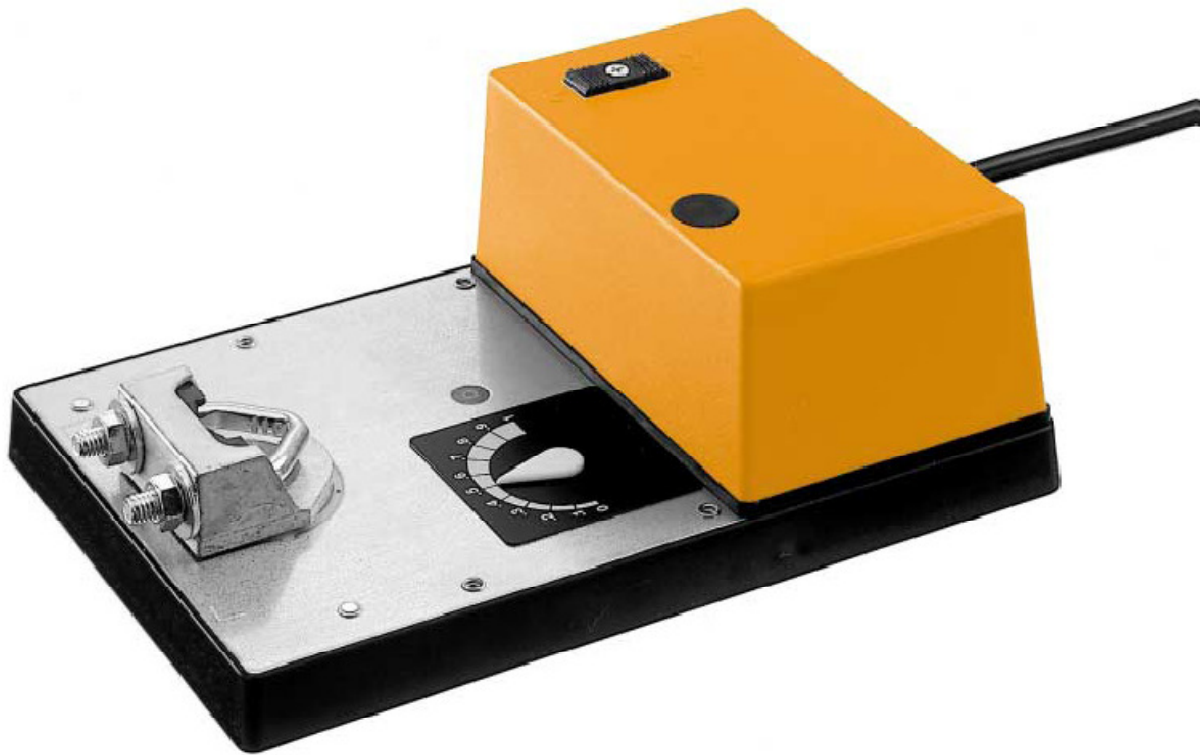
- Moment 15Nm
- Mootori nominaalpinge 24V
- Mootori võimsus 1.8W
- Mehhaniliselt limiteeritud töösektor 95°
- Ventiili kinnitus ava 10 – 20 mm
- Bloki suurus 200 x 123 x 75 mm
- Kaal 1,4 Kg
- Töösektori teostusaeg nominaalpingel ja normaalkoormusel 90...150 s (0...15 Nm)
- Manuaalse operatsiooni nupu võimalus
- Ventiili asendit kirjeldava näidiku võimalus
- Ümbritseva temp. vahemik – 30...+50°C
- Hoidmise temp. – 40...+80°C
- Niiskuse test EN 60335-1
- Kaitse aste IP 54

3.2 Jõuosa mõõdud ja kujundus



Illustratsioon (joonis) nr. 3.3

Illustratsioon nr. 2.3 on kujutatud BELIMO firma toodangu SM240 välised mõõdud ja kujundus.



Pilt nr. 3.2.1 (Ajami Belimo SM 240 pilt)

4. EC – 12 REGULAATORI KÜTTEGRAAFIK

4.1 Mis see on ?

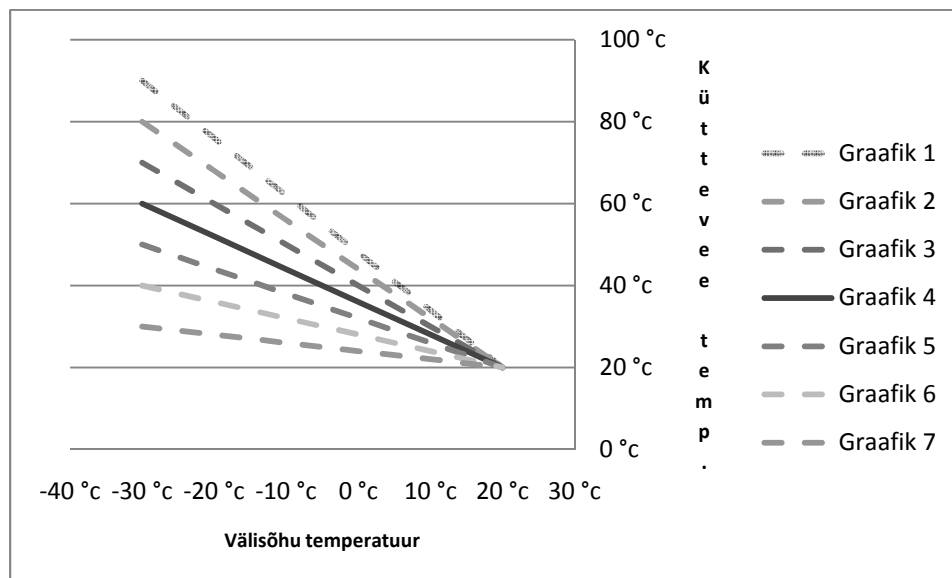
See on temperatuuri regulaatori tööpõhimõtte graafiline kujutus ja samuti ka kasutaliidese graafiline kujundus kui abivahend kasutajale. Sellel graafikul on näha välistemperatuuri ja küttevee temperatuuri suhe. Teisisõnu olevenvalt graafiku valikust muutub muutub küttevee temperatuur seoses välisõhu temperatuuri muutusega.

4.2 Milleks on vaja küttegraafikut ?

Seda on vaja selleks ,et temperatuuri regulaatori kasutajale oleks lihtsam regulaatorit seadistada. See tähedab seda ,et regulaatori tööpõhimõte oleks kasutjale paremini arusaadav ning sellest lähtudes regulaatori seadistus paremeetrid oleksind kasutajale paremini arusaadavamad. Näiteks mida teeb seadistusparameeter ja mida see konkreetene paremeeter muudab.

5. EC – 12 KÜTTEVEE REGULAATORI KÜTTEGRAAFIKU PARAMEETRID

5.1 Graafiku valik



Illustratsioon nr. 5.1

See on antud temperatuuri regulaatoris üks olulisemaid paremeetreid. Küttegaafiku valikuga muutub suhe välisõhu temperatuuri ja küttevee temperatuuri vahel. Selle valimine sõltub maja soojuspidevusest ja soojusvahetite (radiaatorite) arvust ja suurusest. Et seda parameetrit optimaalselt seadistada, ei pead olema teadmisi maja soojuspidevusest või soojusvahetite mõjust. Selle parameetri seadistamine on kõige lihtsam kasutades katse – eksituse meetodit.

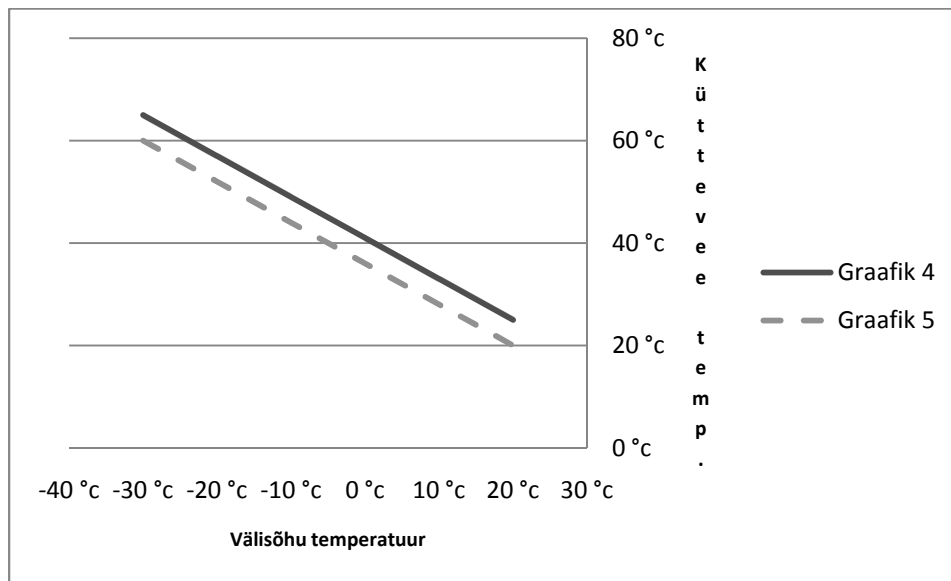
Selle meedoti kasutamine toimub järgnevalt:

Kui maja on piisavalt soojuspidev, tuleb valida laugem graafik (vaata illustratsioon nr. 5.1), kui soojuspidevus on väiksem või on radiaatorid väiksemad, tuleb valida järsem graafik. Tesisõnu tuleb jälgida toatemperatuuri muutusi (see vägagi seotud küttevee temperatuuriga) sõltuvalt välisõhu temperatuuri muutustest. Kui välisõhu külmenedes läheb toas külmemaks, on valitud liiga lauge graafik. Tuleks valida järsem graafik. Kui välisõhu temperatuuri langusega läheb toas soojemaks, on valitud liiga järsk graafik.

Õige graafik on valitud siis, kui toatemperatuur püsib stabiilsena vaatamatta välisõhu temperatuuri muutustele.

Regulaator saavutab stabiilse, graafikukohase radiaatorivee temperatuuri umbes ühe tunni jooksul pärast pingestamist. Ligikaudse temperatuuri saavutamiseks kulub umbes 10 minutit.

5.2 Graafiku nihe



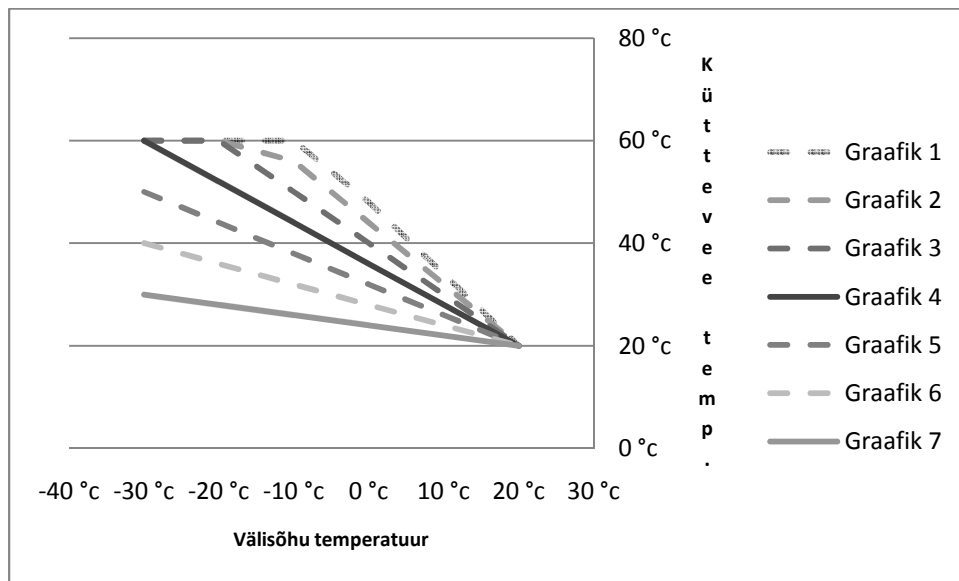
Illustratsioon nr. 5.2

Selle parameetriga saab valida toatemperatuuri, kui see on stabiilne, kuid liiga madal või liiga kõrge. Selle parameetri muutmisega saab nihutada graafikut ülesse või all (Y teljel) ilma tõusunurka muutmata. Teisisõnu, kui muuta seda parameetrit (Graafiku nihe), siis muutub küttevee temperatuur võrdeliselt parameetriga graafiku nihe.

Antud näite korral (illustratsioon nr. 5.2) on tõstetud parameetrit graafiku nihe 5 c kraadi võrra. Sellega saavutasime graafiku asendi muutuse. Kui varem oli välisõhu temperatuuril 0 c kraadi küttevee temperatuur 40 c kraadi siis seda parameetrit muutes on tõusnud küttevee temperatuur 45 c kraadile.

Seda parameetrit on vaja muuta siis, kui graafiku valikuga on saavutatud stabiilne toatemperatuur (peaaegu et ei muutu, kui muutub välisõhu temperatuur)

5.3 Piiramine



Illustratsioon nr. 5.3

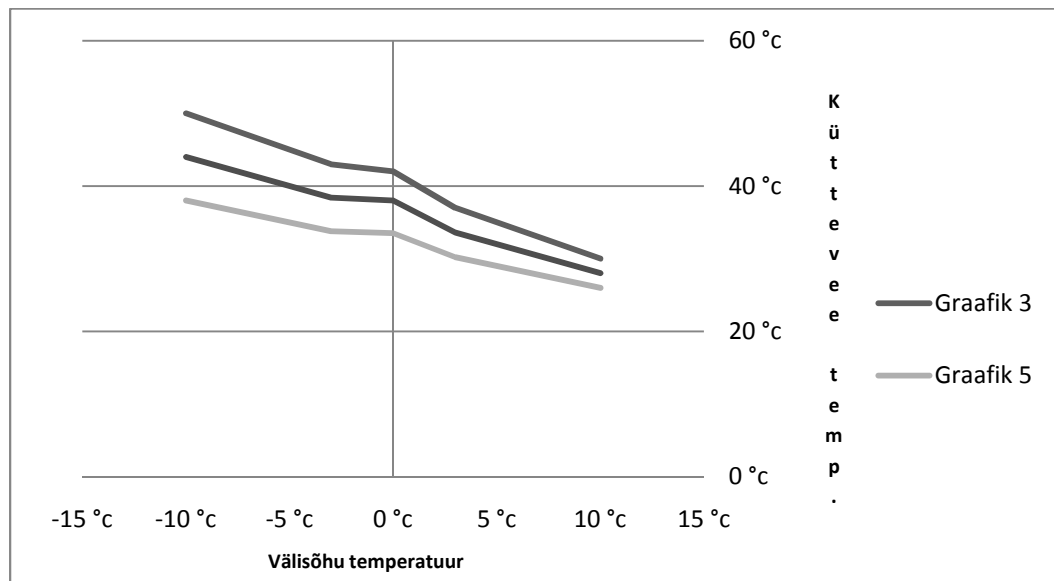
Selle parameetri muutmisega saab piirata maksimaalset küttevee temperatuuri. Teisõnu, kui on see parameeter (piiramine) seadistatud mingi c kraadi väärtuseni siis küttevee temperatuur ei ületa seda väärtust.

Illustratsioon nr. 5.3 korral on parameetri (piiramine) väärtus 60 c kraadi, graafikul on näha, et küttesesi ei ületa seda parameetrit.

Seda parameetrit võib vaadata kui maksimaalne küttevee temperatuur.

Seda parameetrit võib vaja minna juhul kui liiga tuline küttesesi võib põhjustada kahjustusi või ebamugavusi. Näiteks selleks juhtumiks võib olla põrandaküte

5.4 Küttevee temperatuuri tõus 0 c kraadi juures



Illustratsioon nr. 5.4

See on prameeter mida ei saa kasutajal muuta, seda on vaja selleks, et tagada väikese temperatuuri tõusu 0 kraadi läheduses. Seda väikest temperatuuri tõusu on vaja selleks, et parandada regulaatori reguleerimisomadusi. Täpselt veel ei tea kuidas peab parameeter toimima, tõenäoliselt hakkab selle parameetri tulemust mõjutama parameeter graafiku valik ehk siis graafiku nurk.

6. KÜTTEGRAAFIKU PROGRAMMIKOOD KOOS PARAMEETRITEGA

Programmikood on kirjutatud ANSI C programmeerimiskeeles CCS PCWHD komaraatoriga. Sisenditeks on emuleeritavad temperatuuri andmed ja constand. Väljundiks on tablei tegemiseks vajalikud vormistatud andmed graafiku kujundamiseks. Programmikood ei ole kasutatud ujuvkoma arvtüüpe, et hoida kokku programmimälu ja suurendada programmi teostuskiirust.

Oluliste ja mõjuvamate programmikäskude järel on kommentaarid mis alati algavad sümboliga „//“.

6.1 Graafiku funktsiooni programmikood

```
signed int16 graafiku_funktsioon()
{
    //Deklareerin muutujad
    signed int16 graafik_tegur, // M6jutab graafiku nurka
                graafik_null,  // Tegur mis m6jutab v2ikest temp. t6usu 0 c
    kraadi juures
                graafik_y,      // Graafiku asend y skaalal 20c v2lis6hu temp
    juures. Nom on 20 (nagu v2lis6hk-ki)
                graafik_max,    // Grrafiku max v22rtus. Graafiku v2ljus ei saa
    iialgi olla suurem kui see v22rtus.
                kyttevesi,      //(Funktsiooni v2ljund)
                v2lis6hk;       //Andmed andurilt (need tuleb natuke muundada
    et n2ev v2lja arusaadavamalt)
                                //See toimub funkts ADC_temp_conv

    //Muud tegurid vajalkud funktsiooni koostamiseks.
    signed int16 kyttevesi_ilma_null, //kyttevee tulemus ennem kui ta l2bib
    nn. t6usu 0C juures programmikoodi.
                eelmine_tulemus, //Muutuja, et saada teada kyttevee
    muutus 1C kraadi v2lis6hu muutuse kohta.
                eelmise_praeguse_vahe, //Muutuja, et saada teada kyttevee
    muutus 1C kraadi v2lis6hu muutuse kohta.
                vahe_null_l2hedal;    //Muutuja et saada teada vahe t6usu 0C
    kraadi juures.

    int16        loendur,
                loendur_nr2;

    //Paned v22rtused muutujatesse.
    //2 viimast arvu on arvud peale koma.
    graafik_tegur = -1600;
    graafik_null = -458;
    graafik_y = 255;
    graafik_max = 8550;

    //Programmikood {} vahel teostub 5 korda alustades 0-ist.
    for(loendur_nr2 = 0; loendur_nr2 < 5; loendur_nr2++) {

        if(loendur_nr2 != 0) {

            graafik_tegur += 300;

        }
        v2lis6hk = -3500;

        printf("\n\rV2lis6hk Graafik_nr_%Lu", loendur_nr2 + 1);

        //Programmikood {} vahel teostub 11 korda alustades 0-ist.
        for(loendur = 0; loendur < 11; loendur++) {

            v2lis6hk += 500;

            eelmine_tulemus = kyttevesi;
            //P6hiline valem.
            kyttevesi = (signed int16)((signed int32)(v2lis6hk - 2000) *
```

```

    graafik_tegur)/1000)+ (2000 + graafik_y);
    //kyttevee tulemus  enne kui ta l2bib nn. t6usu 0C juures
    programmikoodi.
    kyttevesi_ilma_null = kyttevesi;

    //T6usu 0C juures programmikood.
    if((v2lis6hk <= 500) && (v2lis6hk >= -500)) {
        if((v2lis6hk <= 500) && (v2lis6hk >= 0)) {
            //T6usu 0C juures programmikood valem.
            kyttevesi += ((signed int32)(500 - v2lis6hk) * graafik_null *
    graafik_tegur)/1000000;
        }
        else {
            //T6usu 0C juures programmikood valem.
            kyttevesi += ((signed int32)(500 + v2lis6hk) * graafik_null *
    graafik_tegur)/1000000;
        }
    }
    //Kood et saada teada kyttevee muutus 1C kraadi v2lis6hu muutuse
    kohta.
    if(v2lis6hk == 1000) {
        //Tulemus 10C kraadi juures
        eelmine_tulemus = kyttevesi;
    }
    if(v2lis6hk == 1500) {
        //Tulemus 15C kraadi juures jagatud 5 et saada muutus 1C kraadi
    kohta.
        eelmise_praeguse_vahe = (eelmine_tulemus - kyttevesi)/5;
    }

    //Programmikood, et saada teada vahe t6usu 0C kraadi juures.
    if(v2lis6hk == 0) {
        vahe_null_l2hedal = kyttevesi - kyttevesi_ilma_null;
    }

    //V2ljundi maksimaalse v22rtuse piiramine.
    if (kyttevesi >= graafik_max) {
        kyttevesi = graafik_max;
    }
    //N2itan tulemusi RS232 emulaatori kaudu.
    printf("\n\r%3.2Lw %3.2Lw",v2lis6hk,kyttevesi);
}
//N2itan tulemusi RS232 emulaatori kaudu.

printf("\n\rGraafik nr. %Lu",loendur_nr2 + 1);
printf("\n\rKV muutus 1C kraadi V6 muutuse kohta: %3.2Lw"
,eelmise_praeguse_vahe);
printf("\n\rGraafiku nurga tegur: %3.2Lw",graafik_tegur);
printf("\n\rGraafiku t6usu tegur 0C kraadi juures:
%3.2Lw",graafik_null);
printf("\n\rGraafiku t6us 0C kraadi juures: %3.2Lw",vahe_null_l2hedal);
printf("\n\rGraafiku asukoht Y teljel: 20.00 + %3.2Lw",graafik_y);

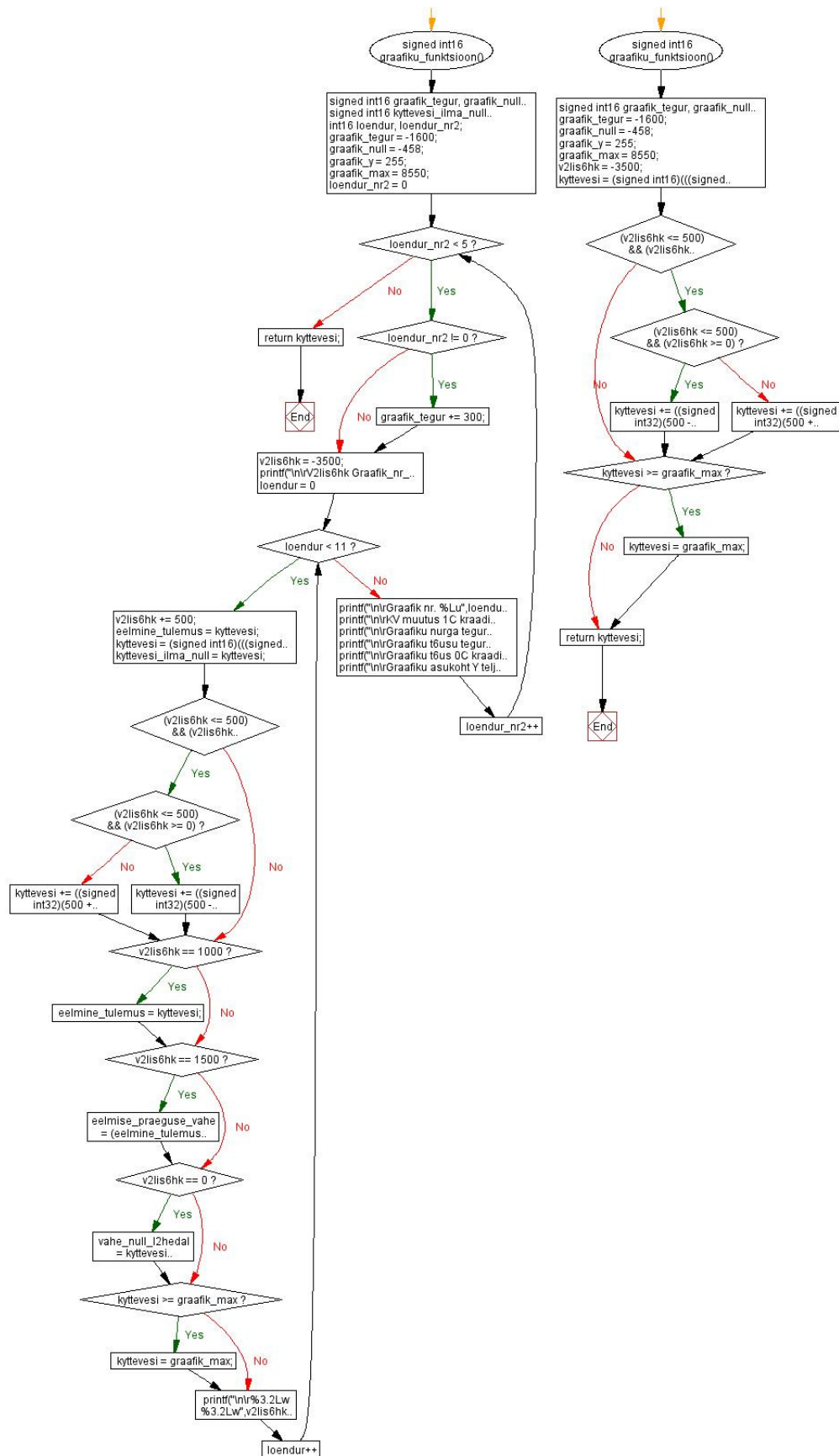
}

//Funktsiooni v2ljund muutuja.
return kyttevesi;

}

```

6.1.1 Programmikoodi graafiline kujundus



6.2 Programmikoodi väljund

Programmikoodi väljundi sain arvuti COM pordi kaudu mis oli ühendatud mikrokontrolleri üksuse UART (universal asynchronous receiver/transmitter) liidesega. Programmikoodist formistati informatsioon printf funktsiooniga, mis oli häälestatud programeerimis tarkvara poolt UART liigesega ühendusse.



V2lis6hk Graafik_nr_1	5.00 38.00	-15.00 48.00
-30.00 90.00	10.00 32.00	-10.00 44.00
-25.00 83.00	15.00 26.00	-5.00 40.00
-20.00 76.00	20.00 20.00	0.00 38.53
-15.00 69.00	V2lis6hk Graafik_nr_3	5.00 32.00
-10.00 62.00	-30.00 70.00	10.00 28.00
-5.00 55.00	-25.00 65.00	15.00 24.00
0.00 52.43	-20.00 60.00	20.00 20.00
5.00 41.00	-15.00 55.00	V2lis6hk Graafik_nr_5
10.00 34.00	-10.00 50.00	-30.00 50.00
15.00 27.00	-5.00 45.00	-25.00 47.00
20.00 20.00	0.00 43.16	-20.00 44.00
V2lis6hk Graafik_nr_2	5.00 35.00	-15.00 41.00
-30.00 80.00	10.00 30.00	-10.00 38.00
-25.00 74.00	15.00 25.00	-5.00 35.00
-20.00 68.00	20.00 20.00	0.00 33.89
-15.00 62.00	V2lis6hk Graafik_nr_4	5.00 29.00
-10.00 56.00	-30.00 60.00	10.00 26.00
-5.00 50.00	-25.00 56.00	15.00 23.00
0.00 47.79	-20.00 52.00	20.00 20.00



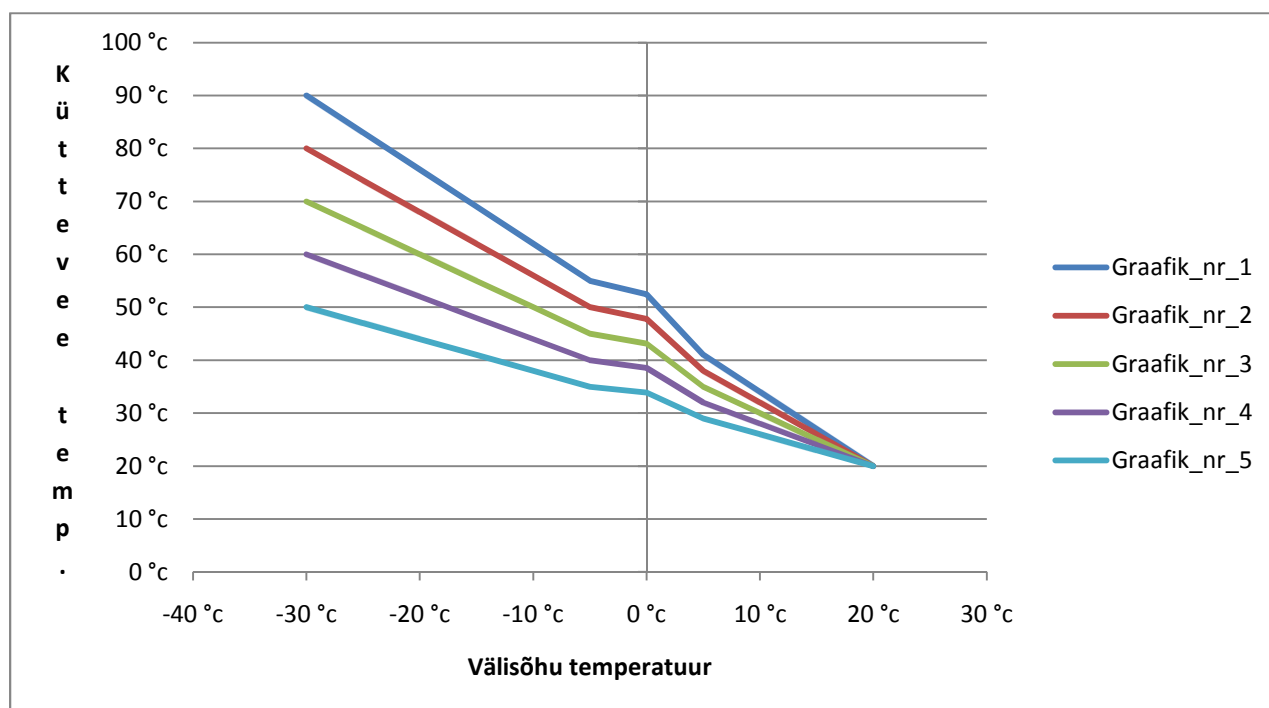
Illustratsioon nr. 6.2 (andmed command prompt- ist)

Sellise väljundi sain arvuti operatsiooni süsteemis kui olin ühendanud mikrokontrolleri COM porti. Selle peal avanes operatsiooni süsteemis automaatselt COMMAND PROMT aken ja numbrid hakkasid jooksmas.

6.2.1 Programmikoodi väljundi graafiline kujutus nr. 1

V2lis6hk	Graafik_nr_1	Graafik_nr_2	Graafik_nr_3	Graafik_nr_4	Graafik_nr_5
-30	90	80	70	60	50
-25	83	74	65	56	47
-20	76	68	60	52	44
-15	69	62	55	48	41
-10	62	56	50	44	38
-5	55	50	45	40	35
0	52,43	47,79	43,16	38,53	33,89
5	41	38	35	32	29
10	34	32	30	28	26
15	27	26	25	24	23
20	20	20	20	20	20

Tabel nr. 6.2.1



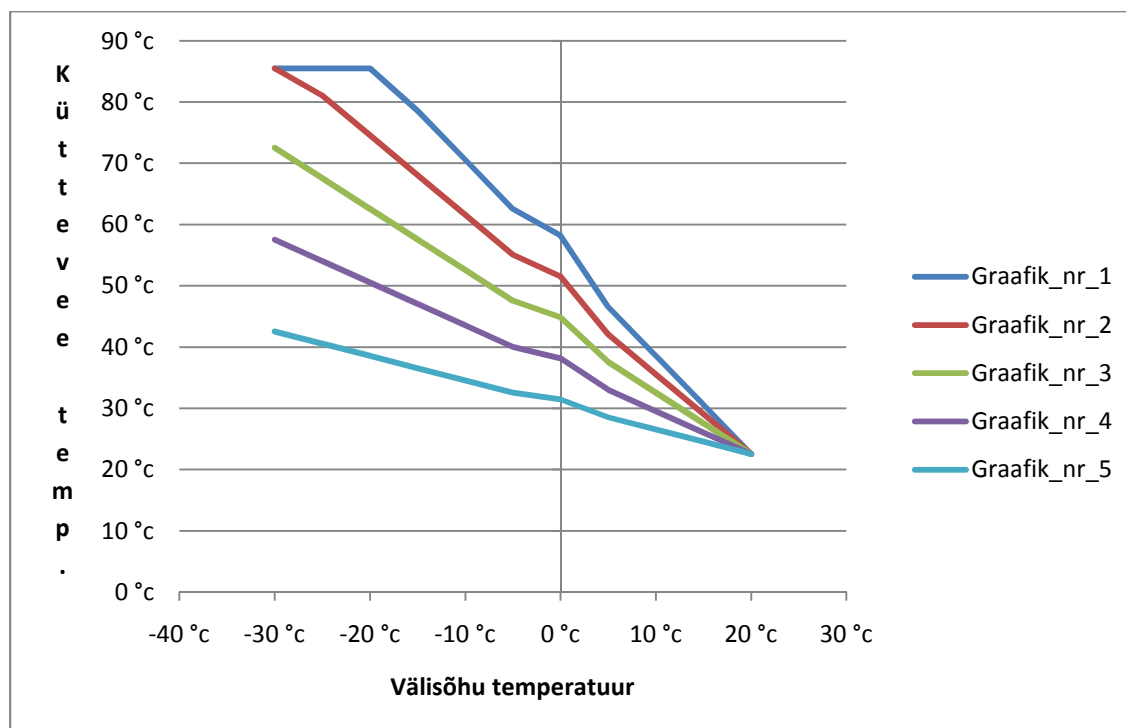
Graafik nr. 6.2.1

Muudetud on mõned konstandid, et näidata programmikoodi funktsionaalsust.

6.2.2 Programmikoodi väljundi graafiline kujutus nr. 2

V2lis6hk	Graafik_nr_1	Graafik_nr_2	Graafik_nr_3	Graafik_nr_4	Graafik_nr_5
-30	85,5	85,5	72,55	57,55	42,55
-25	85,5	81,05	67,55	54,05	40,55
-20	85,5	74,55	62,55	50,55	38,55
-15	78,55	68,05	57,55	47,05	36,55
-10	70,55	61,55	52,55	43,55	34,55
-5	62,55	55,05	47,55	40,05	32,55
0	58,21	51,52	44,84	38,15	31,46
5	46,55	42,05	37,55	33,05	28,55
10	38,55	35,55	32,55	29,55	26,55
15	30,55	29,05	27,55	26,05	24,55
20	22,55	22,55	22,55	22,55	22,55

Tabel nr. 6.2.2



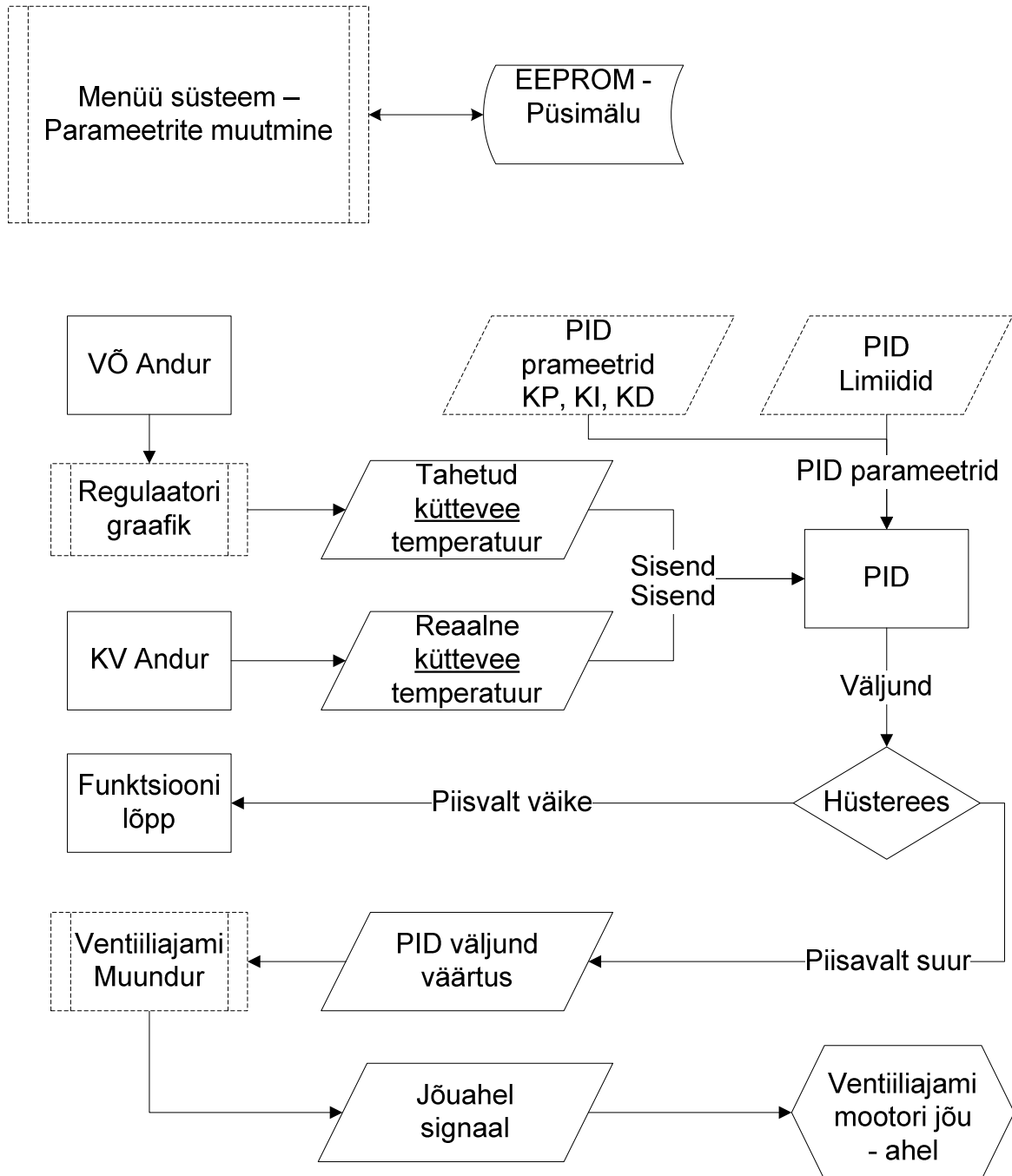
Graafik nr. 6.2.1

Muudetud on mõned konstandid, et näidata programmikoodi funktsionaalsust.

Kaasaarvatud on alandatud parameetrit „temp piiramine“.

7. REGULEERIMISPROGRAMMI BLOKKSKEEM

7.1 Blokkskeem



7.1 Blokskeemi selgitus

Sellel skeemil on kujutatud samm-sammuline tegevusjuhis küttesüsteemi küttevee regulaatori reguleerimis käigust.

Põhimõtteliselt sellel skeemil on toodud PID algoritmi ümsitsevad sisendid ja väljundid.

Sisenditeks on siis andurid. VÕ tähendab välisõhu temperatuuri ning KV tähendab küttevee temperatuuri. Samuti on näha kast „Regulaatori graafik“ mis on muundur ja millest on juttu mujal selles aruandes. See muundur on siin printsipiaal skeemis sellepärast, et see mängib tähtsat rolli regulaatori enda kui ka PID algoritmi töös. Nimelt see muundur on tähtis sellepärast, et selle muunduri väljund on arv – väärtus „Tahetud küttevee temperatuur“ , mis on sisuliselt PID algoritmi niinimetatud etteantud väärtus.

Sisendite ja PID algoritmi vahel on kaks arv – väärtust näitavat kasti:

- Tahetud küttevee temperatuur – see on PID algoritmi etteantud väärtus
- Reaalne küttevee temperatuur – see on PID algoritmi süsteemi (objekti) muutuja väärtus

Väljundiks on ventiiliajami mootor. PID algoritmi ja ventiiliajami vahele jääb ventiiliajami muundur ,et teha PID algortimist välja tulev arv sobilikuks siganaaliks ,et juhtida mootorit läbi jõuahela.

7.2 Skeemi eesmärk

Kuna programm is peab olema kirjutatud C keeles, on sellest väga raske aru saada eriti veel kui on tegemist hästi mahuka programmiga. See printsipiaal skeem näitab väga põhimõttelisel tasemel regulaatori reguleerimis osa ja töökäiku.

8. EC – 12 KÜTTESÜSTEEMI TOA- JA VÄLISTEMPERATUURI ANDUR

Antud regulaatori korral kasutatakse takistus temperatuuri andurit positiivse temperatuuri teguriga. Ehk siis mida suurem on temperatuur seda suurem on takistus. Andur omab paraboloolset temperatuuri takistus tegurit ehk siis kõrvalekalle lineaarsusest on 3 %. Seda kõrvalekannet saab elimineerida lisades anduriga kompenseerimv takisti rööbiti anduriga. Samuti tasuks mainida, et anduri täpsus on 1 %, teisiõnu anduti takistus – temperatuuri suhe võib erineda 1 %.

Järgnevas tabelis on toodud andurite parameetrid anduri tootja dokumentatsioonist, EC – 12 puhul kasutatav andur on KTY10 – 62. Tabelil on näha, et anduri takistus 25 c kraadi juures on umbes 2K oomi ning anduri komponenedi kuju on TO – 92.

Type	Marking	Ordering Code	$R_{25 \text{ min}}$	$R_{25 \text{ max}}$	Package
			(in Ω with $I_{op} = 1 \text{ mA}$)		
KTY 10-5	KTY 10-5	Q62705-K110	1950	1990	TO-92
KTY 10-6	KTY 10-6	Q62705-K132	1980	2020	TO-92
<u>KTY 10-62</u>	<u>KTY 10-62</u>	<u>Q62705-K71</u>	<u>1990</u>	<u>2010</u>	<u>TO-92</u>
KTY 10-7	KTY 10-7	Q62705-K111	2010	2050	TO-92

Tabel nr. 8.1 (Anduri andmelehel anduri valikute sektsioonist)



Illustratsioon nr. 8.2 (TO-92 korpusega andur KTY10 – 62)

8.1 Arvutused

Kuna PIC mikrokontroler moodul mõõdab pigelangu anduril siis on vaja teada vastavat takistust mingil temperatuuri väärtusel, seda saab antud valemist. See valem käib ilma kompensatsiooni takistita anduri kohta.

T =

$$\left(25 + \frac{\sqrt{\alpha^2 - 4 \times \beta + 4 \times \beta \times k_T - \alpha}}{2 \times \beta} \right) ^\circ \text{C}$$

Illustratsioon nr. 8.1 (valem)

Teistpidi:

$$R_T = R_{25} * (1 + \alpha * \Delta T + \beta * \Delta T^2) = f(T)$$

Kui: $\alpha = 7.88 \cdot 10^{-3} \text{ K}^{-1}$; $\beta = 1.937 \cdot 10^{-5} \text{ K}^{-2}$

Illustratsioon nr. 8.2 (valem)

Valemid nr. 8.1 ja 8.2 on anduri KTY10 62 mittelineaarsed valemid.

Kasutades kompensatsiooni takistit väärtusega 4K64 oomi võib kasutada lineaarseid valemid:

Temperatuur → Takistus	Takistus → Temperatuur
Takistus = 7,54 * Temperatuur + 1210,4	Temperatuur = 160,53 – (Takistus / 7,54)

Tabel nr. 8.3 (võrdlus, tabel)

Tablil nr. 8.3 on kujundatud kahte lineaarset valemit.

8.2 PIC mikrokontroller-mooduli analoog - digitaal muundur (ADC)

- Peagu, et igal PIC-il on 10 – bitine ADC. See tähendab, et mingi pinge vahemikus (max kuni 5v) on võimalik jagada 1024 võrdeseks osaks ehk siis mingi pinge väärtus on number 0 ja 1023 vahel (0- il on ka väärtus).

Näiteks 2,5 volti digitaalne väärtus on 512 ja 1,25 volit on 256 jne.

- Samuti on olemas paljudel uuematel PICidel võimalus piirata pinge vahemikku kui samal ajal jääks alles 10bit väärtus. Seda nimetatakse referents pingeks. Ref. lühidalt. On võimalik piirana nii alumist kui ka ülemist pinge vahemiku väärtust, samas jättes alles 10bit resolutsiooni.

Näiteks: Kui varem oli muundatavaks pinge vahemikuks 0 – 5V siis lisades alumise Ref. pinge 2V ja ülemise 4V siis muundatavaks pinge vahemikuks jääb 2 – 4V

8.3 Voolugeneraator anduri ja PIC-i vahel

Selleks, et teha anduri pingelang sobilikumaks PIC-i analoog – digitaalmuundurile on vaja piisavalt hea stabiilsuse ja suurusega signaali. Seda ülesannet täidab LM – 117 mis on iseeneset voolu regulaator. Lülitus on seatud nii, et 1 oomi koormuse kohta vastaks 1 mA. Teissõnud, et saada PIC – ile langev pinge peab korrutama anduri takistus 1mA – ga ehk 0,001 – ga.

Selle teadmise saab muuta anduti temperatuuri ja takistuse seose valemi temperatuuri ja pinge seose valemiks mida on kergem kasutada.

Temperatuur → Takistus	Takistus → Temperatuur
$Takistus = 7,54 * Temperatuur + 1210,4$	$Temperatuur = 160,53 - (Taksitus / 7,54)$
Temperatuur → Pinge	Pinge → Temperatuur
$Pinge = (7,54 * Temperatuur + 1210,4) * 0,001$	$Temperatuur = ((Pinge / 7,54) * 0,001) - (160,53 * 0,001)$

9. PIC18 SEERIA PROGRAMEERIMISTARKVARA CSS 4.093 MEETRIKUD

Arvutüüp	Arvtüübi suurus	Ulatus		Arvu kohti
		Unsigned	Signed	
int1	1 bit number	0 to 1	N/A	1
int8	8 bit number	0 to 255	-128 to 127	2-3
int16	16 bit number	0 to 65535	-32768 to 32767	4-5
int32	32 bit number	0 to 4294967295	-2147483648 to 2147483647	9-10
float32	32 bit float	-1.5 x 10 ⁴⁵ to 3.4 x 10 ³⁸		7-8

Tabel nr. 9.1

Tabelil nr. 9.1 on näidatud C keeles kasutatavad põhilised arvtüübid ning arv arvsüsteemid.

	8bit int8 [us]	16bit int16 [us]	32bit Int32 [us]	32 bit float [us]
+	0.3	0.4	0.6	51.3
-	0.3	0.4	0.6	52.3
*	0.4	3.2	22.2	35.8
/	11.3	32	106.6	144.9
exp()	*	*	*	510.4
ln()	*	*	*	644.8
sin()	*	*	*	698.7

Tabel nr. 9.2

Sellel tabelil nr. 9.2 on näidatud matemaatiliste operatsioonide täitmiseks kuluvat aega PIC 18 seeria mikrokontrolleriga töösagedusel 40 Mhz ehk siis 10 Miljonit instruksiooni sekundis.

Kahjuks ei ole tehtud mõõtmisi seoses arvtüüpidega operatsioonide täitmiseks vaja mineva mälukasutusega. Arvestades sellega, et ujuvkoma (floating point) arvu enda struktuur on keerulisem, see teeb keerulisemaks arvuga operatsioonide täitmise, ja selletõttu lähem mällu rohekm käske ning see tähendab suuremat programmkoodimälu -, kui ka operatiivmälu kasutus.

9.1 Kuidas kasutada täisarvu murdarvuna ?

Täisarv on näiteks 23712, et selle arvuga kirjeldada temperatuuri ja teha selle arvuga arvutusi, saab arvu jagada kahte ossa. Esimene osa on siis 3 kohaline täisarvu osa: 237 ja teine osa on 2 kohaline arv: 12. Selle puuduseks on see ,et maksimaalne täisarvuna loetav arvu osa väärtus vb olla kuni 326 mis on rohkem kui piisav, et sooritada arvutusi temperatuuri kohta. Et funktsiooni lõpus või alguses oleks arvu komakoht paigas, tuleks kasutada korrutamist või jagamist 10, 100, 1000 või 10000 -ga ehk skaleerida.

9.2 Arvtüüpide kokkuvõtte

Hea: Operatsioonid täisarvudega on kiiremini teostatavad ja vähem ressursi nõudvad.
Negatiivne külg on see, et arvulatus (ifornamtsiooni hulk) on limiteeritud.

9.3 Näidis: programmikood

```
unsigned int16 ADC2voldid()
{
    //Muutujate deklareerimine
    int16 adc_v22rtus;
    unsigned int32 voldid;
    unsigned int16 voldid_16;
    //Muutujasse "adc_v22rtus" l2heb ADC registri v22rtus
    adc_v22rtus = Adcfunkt ();
    //Valem, et teisendada ADC v22ruts --> Pingeks (V)
    voldid = (unsigned int32) (( adc_v22rtus * 500000));
    voldid_16 = (unsigned int16) ( (voldid >> 10) / 1000);
    //Funktsiooni tagatis (v22rtus)
    return voldid_16;
}
signed int16 Voldid_temp_conv()
{
    //Muutujate deklareerimine
    signed int16 Temperatuur,
    Pinge,
    Pinge2,
    a,
    b;
    // Peatan inteerruptide t66.
    disable_interrupts (int_timer3) ;
    disable_interrupts (int_RTCC) ;
    disable_interrupts (int_RB) ;
    disable_interrupts (global) ;
    // Näitab v22rtusi 1000 korda järjest
    for (a = 0; a < 1000; a++) {
        // Muutujass "Pinge" l2heb funktsiooni
        // "ADC2voldid" tagatise v22rtus
        Pinge = ADC2voldid ();
        // Valem
        b = (signed int32) (Pinge * 100000) / 754;
        Temperatuur = b - 16053;
        // LCD -le numbrite kuvamis osa
        lcd_gotoxy (1, 1) ;
        printf (lcd_putc, "Temp. % 5.2Lw ", Temperatuur);
        lcd_gotoxy (15, 1) ;
        // Kraadi märk
        lcd_putc ( (char) 223) ;
        lcd_gotoxy (16, 1) ;
        lcd_putc ( (char) 'C') ;
        lcd_gotoxy (1, 2) ;
        printf (lcd_putc, "PINGE % 3.2Lw", Pinge);
        lcd_gotoxy (16, 2) ;
        lcd_putc ('V') ;
    }
}
```

9.4 Programmikoodi seletus ja kirjeldus

Toodud on üks programmijupp mis sisaldab kahte funktsiooni:

ADC2voldid ja Voldid_temp_conv.

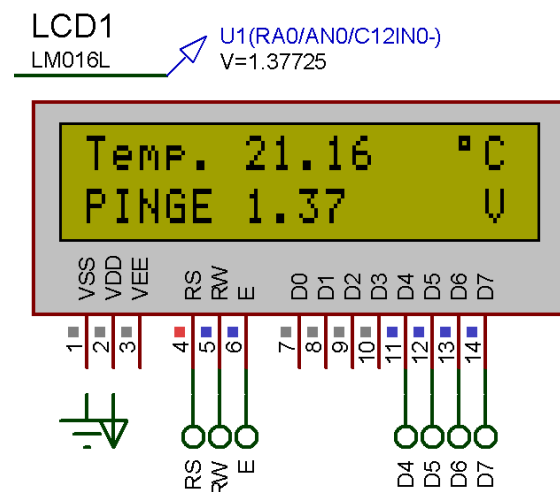
„ADC2voldid“ teeb väärtuse PIC analoog digitaal muundurist (ADC väärtuse 0 -1023) pingeks, antud juhul on pinge 0 – 5 V. Kuna see väärtus on täisarvuna siis tuleks edasiseks kasutada skaleerimist.

„Voldid_temp_conv“ funktsioon teeb aga pinge väärtuse temperatuuriks vastavalt andurile KTY10 62 lineariseeritud valemitele mille korral tuleb kasutada kompensatsiooni takistit.

Nagu näha mitmes kohas korrutatakse või jahatakse 10 – ga, et skaleerida väärtust.

Selle tulemuseks saab temperatuuri muutuja ehk temperatuuri kirjeldava arvu millega saab teha edasisis arvutusi.

9.5 Programmikoodi näite tulemus



Illustartsioon nr. 9.3 (pilt)

Pildid on kujutatud funktsioonide töökäigu tulemusena saadud muutjute kuvamine LCD ekraanil simueerimis- ja emuleerimis tarkvaraga.