



**LAHDEN AMMATTIKORKEAKOULU**  
*Lahti University of Applied Sciences*

## **HYBRIDILÄMMITYKSEN OHJAUS**

**AmiHybridi-käyttöliittymän suunnittelu opetuskäyttöön**

**LAHDEN  
AMMATTIKORKEAKOULU**  
Tekniikan ala  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikka  
Opinnäytetyö  
Kevät 2012  
Jorma Suomäki

Lahden ammattikorkeakoulu  
Tietotekniikan koulutusohjelma

SUOMÄKI, JORMA: Hybridilämmityksen ohjaus  
AmiHybridi-käyttöliittymän suunnittelu opetuskäyttöön

Ohjelmistotekniikan opinnäytetyö, 44 sivua, 5 liitesivua

Kevät 2012

## TIIVISTELMÄ

---

Opinnäytetyön tavoitteena on suunnitella ja toteuttaa hybridilämmityksen käyttöliittymä Helsingissä toimivan ammatillista aikuiskoulutusta järjestävän Amiedun koulutustiloissa olevien lämmitysjärjestelmien ohjaukseen. Lähtökohtana oli saada aikaan käyttöliittymäohjelma, joka mahdollistaa uusiutuvien energiamuotojen koulutuksessa tuotetun lämpöenergian hyödyntämisen kaukolämmityksessä. Lisäarvona työssä on energian säästäminen ja opetustoiminnan kehittäminen.

Opinnäytetyön ensimmäisessä osassa perehdytään LabJack Companyn USB/Ethernet-pohjaiseen tiedonkeruu- ja ohjauslaitteen toimintaan ja sen hyödyntämiseen ohjelmointirajapintana käyttöliittymän ja fyysisen laitteiston välillä. Samoin perehdytään niihin peruseräotteisiin, joiden mukaan kiinteistöautomaatiojärjestelmien käyttöliittymät tulisi yleisellä tasolla suunnitella ja tuottaa.

Toisessa osassa on lähtökohtana oppimisympäristökäyttöön soveltuvan käyttöliittymän suunnittelu. Suunnittelu sisältää toimintakaavion suunnittelun, toiminnallisuuden määrittämisen ja itse käyttöliittymän suunnittelemisen. Näiden suunnitelmien pohjalta toteutetaan varsinainen käyttöliittymä ja oppimisympäristö.

Opinnäytetyön aikana esiin tulleet kehitysajat koskivat pääosin kiinteistövalvoimosta yleisesti puuttuvia elementtejä. Tällaisia koulutuksen erityisvaatimuksia toteuttavia elementtejä ovat esimerkiksi mahdollisuus palauttaa käyttöliittymä perusasetuksiin. Samoin erilaisten vikatilanteiden helposti tehtävä simulointimahdollisuus on koulutuskäytössä olennaisesti opetusta monipuolistava ja kehittävä ominaisuus.

Avainsanat: hybridilämmitys, käyttöliittymä, LabJack, C#

Lahti University of Applied Sciences  
Degree Programme in Information Technology

SUOMÄKI, JORMA: Control of hybrid heating  
AmiHybridi interface design for educational use

Bachelor's Thesis in Software Engineering, 44 pages, 5 appendices

Spring 2012

ABSTRACT

---

The purpose of this thesis was to design and implement a user interface for a hybrid heating system. The study was made for the training facilities of Amiedu Helsinki. The starting point was to provide a user interface program that can be used both in energy production and in education. The added value is energy saving and development of teaching.

The first part of the thesis focuses on the operation of LabJack Company's USB/Ethernet-based data acquisition and control device and its utilization as an application programming interface to the basic principles according to which the interfaces of the building automation systems should be designed and developed.

The second part of the thesis consists the design of a user interface suitable for a learning environment. The design includes the determination of the functional system, as well as designing the user interface itself. Based on these designs, the actual learning environment was implemented.

In this thesis the problems encountered in developing the ideas mainly concerned the elements that are usually lacking in real-life control rooms. Since the interface will be used in education, there are some special requirements that it has to meet. For example there must be a possibility to the system easily back to basic settings, or to previous settings in educational situations, a feature that will enrich and improve teaching and learning.

Keywords: hybrid heating system, user interface, LabJack, C#

## SISÄLLYS

1	JOHDANTO	1
2	LÄHTÖTILANTEESTA HYBRIDIOHJAUKSEEN	3
2.1	Lähtötilanne	3
2.2	Lähtöteoria	4
3	GRAAFINEN KÄYTTÖLIITYMÄ LVI-OHJAUKSESSA	6
3.1	Käyttöliittymän monimuotoisuus	7
3.2	Tekniset tavoitteet	7
3.3	Trendiseuranta	8
3.4	Hälytystiedot ja niiden käsittely	8
3.5	Hälytysten seuranta	9
3.6	Tiedon reaaliaikaisuus ja käytettävyys	9
3.7	Käyttöliittymän yhteys tietokantaan	10
4	KÄYTTÖLIITYMÄN LAITTEISTOT JA OHJELMAT	11
4.1	LabJack U3 tiedonkeruu- ja ohjauslaite	11
4.1.1	Lämpötilamittaus	13
4.1.2	Pumppujen ja venttiilien ohjaukset	14
4.2	Aurinkolämmitysjärjestelmät	15
4.3	Lämpöpumppulaitteistot	16
4.4	Kaukolämmitys ja hybridivaraaja	17
4.5	Käyttöliittymäohjelman ohjelmointi	18
4.5.1	Linkkikirjasto LJUDDotNet	19
4.5.2	Tärkeimmät tiedonkeruujärjestelmän ohjausfunktiot	20
5	KÄYTTÖLIITYMÄN SUUNNITTELU	23
5.1	Hybridijärjestelmän käyttöliittymän grafiikka	23
5.2	Hybridijärjestelmän toimintaselostus	24
5.2.1	Lämpöpumppuvaraajan lämmönsiirto	25
5.2.2	Aurinkolämpövaraajien vuorottelu	26
5.2.3	Aurinkolämmitysten lämmönsiirto	27
5.2.4	Sähkölämmityksen ohjaukset	29
5.2.5	Siirretyn lämmitystehon laskenta ja seuraaminen	29
5.3	Asetukset ja asetustiedostot	30
5.4	Tietokanta	30

6	KÄYTTÖLIITTYMÄN TOTEUTUS	32
6.1	Käyttöliittymän pohjagrafiikka ja komponenttien ominaisuudet	32
6.2	AmiHybridi-ohjelman käyttövalikko	34
6.3	Trendiseuranta	37
6.4	Hälytykset	40
7	TESTAUS	42
8	YHTEENVETO JA JOHTOPÄÄTÖKSET	43

## 1 JOHDANTO

Kiinteistöjen sähkön- ja lämmönkulutukset aiheuttavat ison osan rakennusten elinkaaren aikaisista ympäristöpäästöistä ja energiakustannuksista. Kasvavan energian kulutuksen ja ympäristötietoisuuden myötä on lämmitysjärjestelmien ympäristökuormitus noussut energian hinnan ohella merkittävään asemaan lämmitysmuotojen valintapäätöksiä tehtäessä. Nykytilanteessa rakennuksen lämmitysjärjestelmän valinta on muuttunut entistä haastavammaksi myös vaihtoehtojen lisääntymisen myötä. Kasvavassa määrin rakennuksien lämmityksessä käytetään yhden lämmitysmuodon sijasta kahta tai useampaa lämmitysmuotoa, eli niin sanottua hybridilämmitystä. Hybridilämmitys on ollut jossain muodoissa käytössä jo useita vuosia vesikiertoisissa järjestelmissä, joissa esimerkiksi öljylämmityksen varmistuksena on käytetty osittaista sähkölämmitystä. Näiden kaksoislämmitysmuotojen tavoitteena on ollut ensisijaisesti varmistaa lämmityksen riittävyys ongelmatilanteissa, eikä niinkään energian säästö, mikä puolestaan on olennaista hybridilämmitykselle. Nykyisin hybridilämmityksestä puhuttaessa käsitteen katsotaan sisältävän aurinkolämmityksen, lämpöpumppulämmityksen, pellettilämmityksen ja sähkölämmityksen.

Se, että hybridilämmitys on selkeästi kasvava lämmitysmuoto, ei ole kuitenkaan saanut säätöjärjestelmiä suunnitteleva ja toteuttavia tahoja kehittämään merkittävästi hybridilämmitykseen soveltuvia säätöjärjestelmiä. Yhä edelleen hybridilämmityksissä käytetään eri lämmitysjärjestelmien ohjaukseen erillisiä yksikösäätimiä ja eri lämmitysmuotojen keskinäinen ohjaus on jäänyt karkeaksi ja usein vaille hienosäätöä. Eri lämmöntuottojärjestelmien käytönaikainen vuorottelu jää asianomaisen lämmityslaitteiston hoitajan tehtäväksi. Tämä vaatii laitteistonhoitajalta perehtyneisyyttä lämmitysjärjestelmien ominaisuuksiin ja viitseliäisyyttä käyttää kulloinkin tarkoituksenmukaisinta lämmitysmuotoa.

Helsingissä sijaitseva Amiedu on johtava ammatilliseen aikuiskoulutukseen keskittynyt oppilaitos, jossa on viime vuosina panostettu runsaasti uusiutuvien energiamuotojen opetukseen ja käyttöön. Opetuskäytössä on tällä hetkellä kaksi aurinkolämmitysjärjestelmää sekä lukuisia maa-, ilma- ja ilmasta veteen- lämpöpumppuja. Lisäksi opetuskäytössä on täydellinen kaukolämmityksen alajakokeskus ja

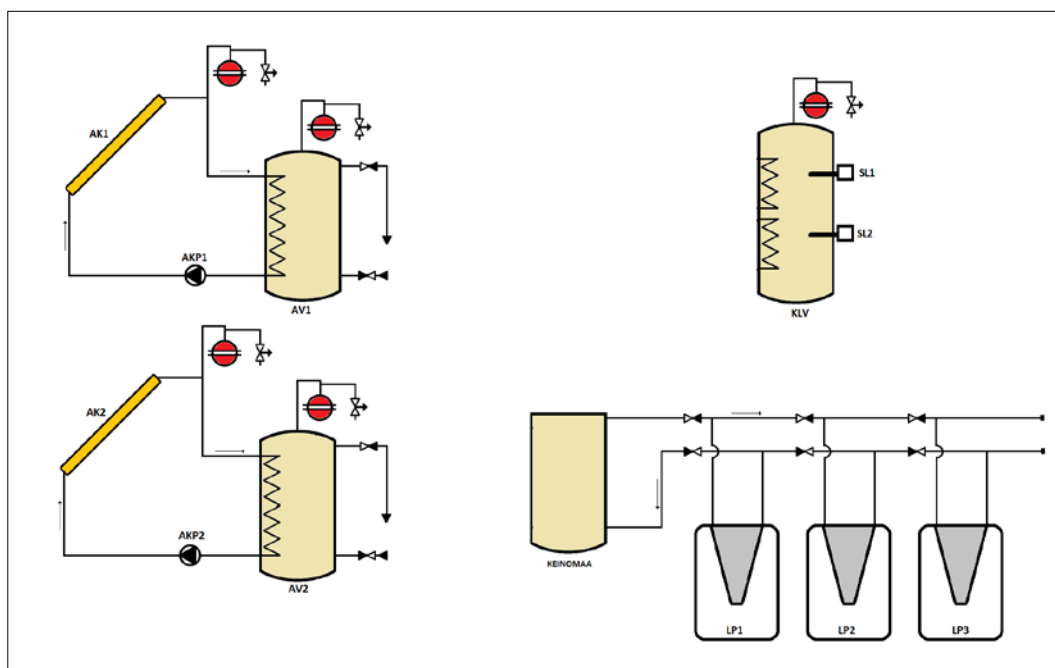
siihen liitetty kaukolämmitysvaraaja. Kaukolämmitysvaraajaan energia on tähän asti tuotettu yksinomaan sähkövastuksilla. Päästäkseen paremmin hyödyntämään muiden energiamuotojen opetuksessa syntyvää hukkaenergiaa on suunnitelmissa ollut siirtää näiden järjestelmien lämpöenergiat hybridiohjausjärjestelmän avulla kaukolämmitysvaraajaan.

Tämän opinnäytetyön tavoitteena on suunnitella ja kehittää LabJack U3-HV tiedonkeruu- ja ohjauslaitetta hyödyntävä käyttöliittymäohjelma. Lisätavoitteena on mahdollistaa käyttöliittymäohjelman keräämien tietojen käyttö opetustilanteissa. Tavoitteiden saavuttamista varten tutkitaan ensin, miten LVI-käyttöliittymät tulisi suunnitella. Sen jälkeen luodaan Windows-työpöytäsovelluksena toimiva lämmönsiirtoon erikoistunut käyttöliittymäohjelma.

## 2 LÄHTÖTILANTEESTA HYBRIDIOHJAUKSEEN

### 2.1 Lähtötilanne

Lähtötilanteessa eri lämmitysmuodot toimivat kuvion 1 mukaisesti täysin erillisinä yksikköinä. Tästä aiheutuu se, että aurinkolämmityksen varaajasäiliöihin (AV1 ja AV2) kertyy aurinkokerääjien (AK1 ja AK2) kautta lämmitysvettä, jolla ei ole varsinaisesti käyttöä kyseisen järjestelmän koulutuksessa. Samoin käy lämpöpumppujen (LP1 – LP3) osalta. Lämpöpumput ovat lämmityskokonaisuuksia ja sisältävät aina laitekohtaisen käyttövesi- ja lämmitysvaraajan. Osa lämpöpumppujen tuottamasta lämpöenergiasta voidaan siirtää keinomaahan, joka toimii lämpöpumppujen lämmönkeruupiirissä maahan asetettavan putkiston kuvaajana.

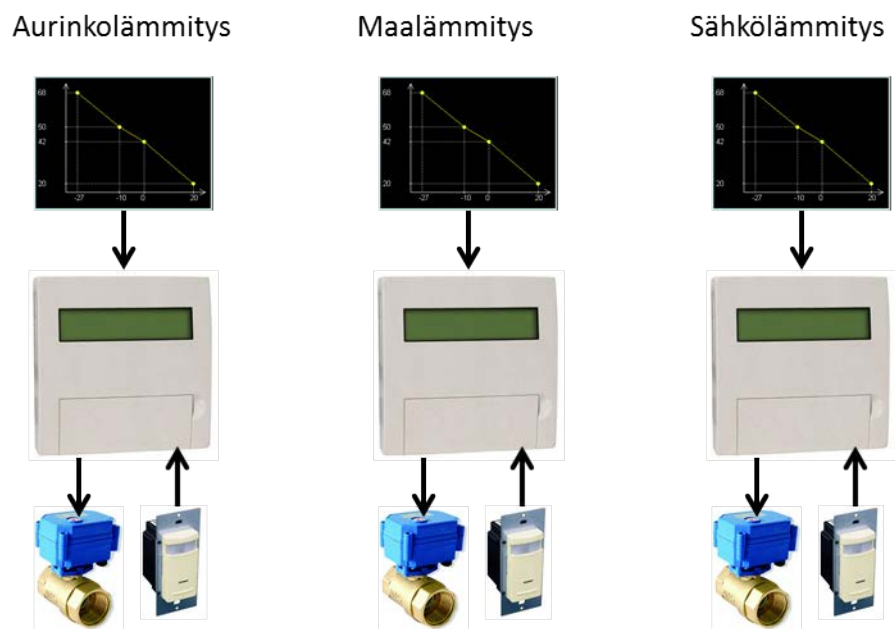


KUVIO 1. Lähtötilanne

Saadakseen eri laitteiden varaajiin luonnollista kulutusta muistuttavan veden viileneminen on varaajista poistettava sinne kertynyttä lämpöenergiaa. Lähtötilanteessa ainoa mahdollisuus on laskea lämmitetty vesi viemäriin ja samanaikaisesti tuoda tilalle kylmää vettä. Kun kaukolämmitysvaraajaa (KLV) lämmitetään näiden toimintojen kanssa samanaikaisesti sähkövastuksilla (S1 ja S2), on tilanne kokonaisuuden ja energiatalouden kannalta erittäin huono.



Kaikkia nykyisiä lämmitysmuotoja ja niiden laitteistoja ohjataan kuvion 2 mukaisesti erillisillä yksikkösäätimillä. Tämä ei ole energiataloudellisesti paras ratkaisu usean eri lämmitysmuodon yhteisohjaukseen. Energiatehokkaampaan tulokseen päästään ohjaamalla eri lämmitysmuotoja keskitetysti hybridiohjaukseen tarkoitetulla säätöjärjestelmällä. Tässä tapauksessa oli kuitenkin perusteltua jättää yksikköohjaukset ennalleen, koska kyseessä ovat koulutuskäyttöön tarkoitetut lämmityslaitteistot ja niitä tulee edelleen pystyä käyttämään myös erillisinä. Lämpöpumppulaitteistojen yhteyteen lisättiin vielä erillinen varastovaraaja, johon kerätään lämpöpumppujen tuottama lämmin vesi. Varastovaraajasta lämmitysvesi siirretään kaukolämmitysvaraajaan erillisellä kiertovesipumpulla.



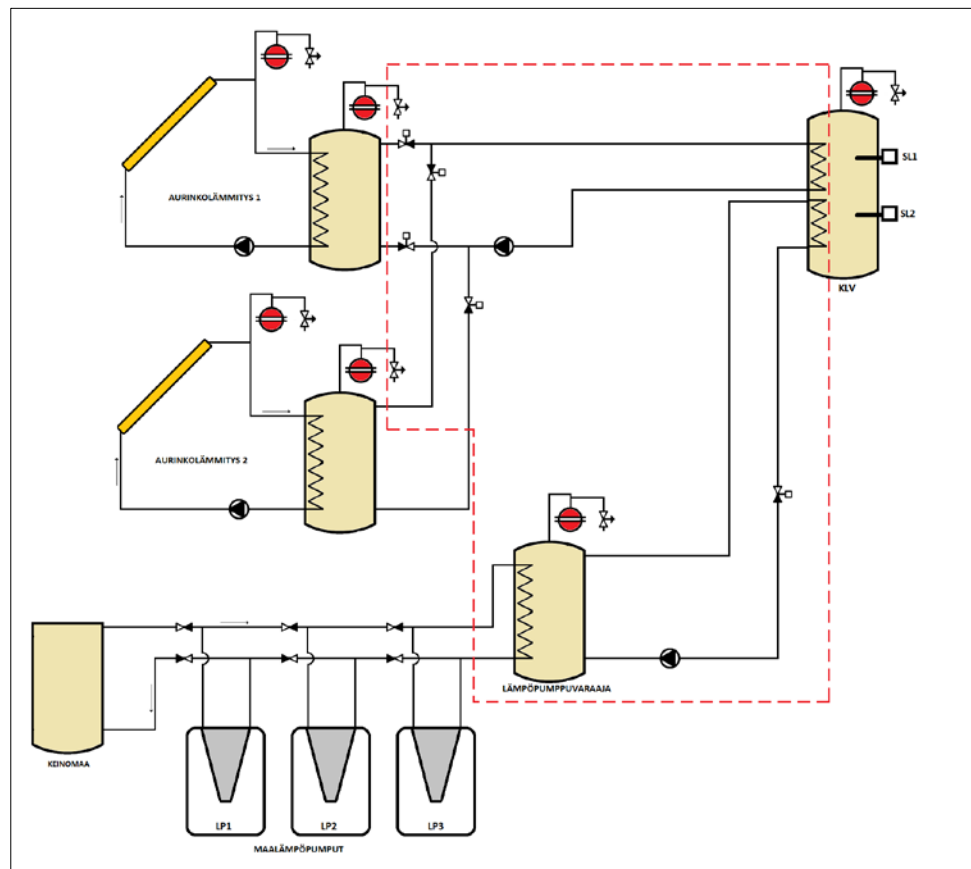
KUVIO 2. Erillisjärjestelmien ohjaaminen yksikkösäätimillä

## 2.2 Lähtöteoria

Lämmityslaitetta mitoittaessa on otettava huomioon kaksi tärkeää mitoitettavaa asiaa: lämmönlähteen lämpötilataso ja se energiamäärä, joka siitä saadaan. Aurinkolämmöllä samoin kuin palavaan prosessiin perustuvissa järjestelmissä lämpötilataso on suunnilleen sama riippumatta laitteiston tehosta. Kyseinen lämpötilataso on lisäksi aina paljon suurempi kuin lämmitysjärjestelmät yleensä tarvitsevat. Lämpöpumpulla sen sijaan lämpötilatasot ovat hyvin matalat. Lämpöpumpun tuottaman lämpöenergian hyötysuhde on lisäksi voimakkaasti riippuvainen sen

tuottaman lämpöenergian lämpötilasta. Keskimäärin voi laskea, että lämpötilan noustessa 1 °C:lla lämpöpumpun hyötysuhde laskee 5 %. Siten lämpöpumpun tuottaman lämpöenergian lämpötilan noustessa 30:stä 50 °C:seen, eli 20 °C:ta hyötysuhde tippuu noin 100 %, eli lämpöpumppu kuluttaa kaksi kertaa enemmän sähköä 30 °C:seen verrattuna. Kytettäessä lämpöpumppu ulkoisen lämmönlähteen yhteyteen, kuten aurinkokeräimeen, olisi hyötysuhteen kannalta edullisempaa lämmitellä lämmitysvesi ensin lämpöpumpulla ja lisätä siihen sen jälkeen aurinkolämmöstä saatava korkeampi lämpöenergia.

Lähtöteoriaan perustuva hybridilämmitysjärjestelmän toimintakaavio on esitetty kuviossa 2. Kuviossa on rajattu katkoviivoituksella järjestelmien yhdistämiseksi rakennettavat putkistot, toimilaitteet ja kytkennät. Lämpöpumppulaitteistoissa tuotettu lämpöenergia siirretään erilliseen lämpöpumppuvaraajaan ja johdetaan sieltä tarvittaessa kaukolämmitysvaraajan (KLV) esilämmitysosaan. Vastaavasti aurinkolämmityslaitteistojen lämmitysvesivaraajien energiat siirretään kaukolämmitysvaraajan jälkilämmitysosaan.



KUVIO 3. Lähtöteorian mukainen hybridilämmityksen toimintakaavio

### 3 GRAAFINEN KÄYTTÖLIITYMÄ LVI-OHJAUKSESSA

Käyttöliittymän ominaisuudet määräävät sen, kuinka hyvin käyttäjä pystyy hyödyntämään sen ohjaamaa järjestelmää. Hyvälle käyttöliittymälle on luonteenomaista, että järjestelmän käytön oppiminen on nopeaa, tehtävien suorittaminen käyttöliittymän avulla vaivatonta ja operointivirheiden määrä pystytään pitämään minimaalisena. Käyttöliittymän tulee myös olla sellainen, että toimintojen mieleen palauttaminen pitkänkin käyttökatkon jälkeen on nopeaa. Edellä kuvatuilla perusominaisuuksilla varustettu käyttöliittymä toimii satunnaisenkin käyttäjän tukena niin, että tämä voi keskittyä olennaiseen eli käsillä olevan tilanteen hallintaan. Hyvän käyttöliittymän tunnusmerkkeinä voidaan pitää myös sitä, että käyttäjälle ei jää tulkinnanvaraa käyttöliittymän esittämän tiedon sisällöstä tai merkityksestä, vaan kaikki käyttöliittymässä esitetty tieto on havainnollisesti esitettyä ja yksiselitteistä. (ST 22 2008, 23 - 31.)

Hyvä käyttöliittymä on

- selkeä, mahdollisimman yksinkertainen ja havainnollinen
- helposti käsiteltävä nopeasti opittavissa ja helposti mieleen palautettavissa
- käyttäjää opastava, avustava ja käyttöä ohjaava
- monimuotoinen tarjoten useita tapoja lopputuloksen aikaansaamiseksi
- aidosti ja luonnollisella tavalla vuorovaikutteinen
- yksiselitteinen tarjoten asioille vain yhden tulkintamahdollisuuden
- kriittisissä toiminnoissa käyttäjän toimenpiteet varmistava
- selkeän peruuttamismahdollisuuden tarjoava
- suoritetuista toimenpiteistä välittömän ja selkeän palautteen antava
- toiminnoiltaan ja rakenteeltaan yhdenmukainen, ristiriidaton ja looginen
- visuaalisesti selkeä ja miellyttävä käyttää
- väreiltään hallittu ja harkittu huomioiden myös rajoiteteiset käyttäjät
- käyttäjän huomion olennaiseen kiinnittävä (värit, äänet, animaatio)
- toteutettu käyttäjän ymmärtämällä kielellä

(ST 22 2008, 23 - 31).

### 3.1 Käyttöliittymän monimuotoisuus

Kiinteistövalvomojärjestelmien käyttöliittymät on usein toteutettu sekä graafisena, kaavioihin ja kuvasymboleihin perustuvana, että valikkopohjaisena tekstiin perustuvana, luettelomuotoisena hierarkiana.

*”Näistä käyttölogiikan vaihtoehdoista on käytetty yleistermejä "Insinööripolku" ja "Talonmiespolku". "Talonmiespolku" pohjautuu järjestelmien tai laitteiden sijaintiin ja vaikutusalueeseen perustuvaan logiikkaan ja "insinööripolku" kiinteistön järjestelmien tekniseen jaotteluun. Nämä käyttöliittymät on yleensä linkitetty toisiinsa siten, että tekniseltä polulta on pääsy sijaintiin perustuvaan näkymään ja päinvastoin. Tällä kahden käyttölogiikan menettelyllä saavutetaan huomattava käytön joustavuus ja tehostetaan tiedon saantia etenkin ongelmatilanteissa.” (ST 22 2008, 23 - 31.)*

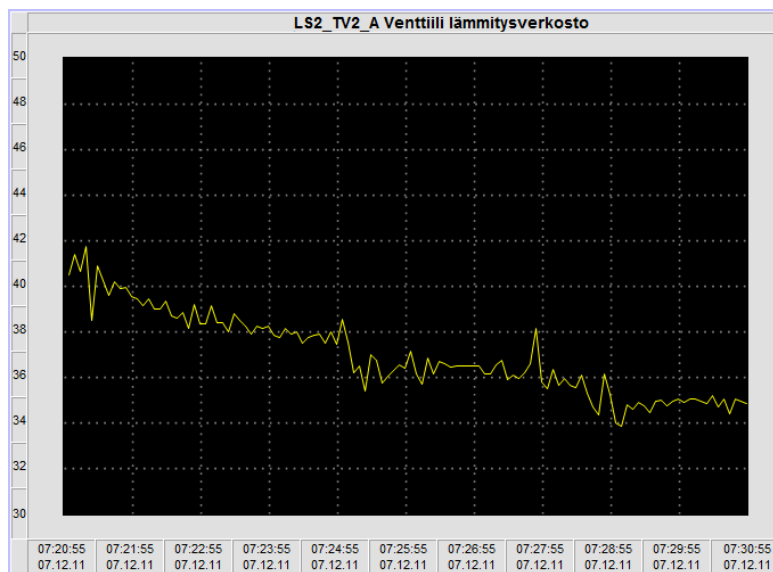
Tietojen esittämiseen ja syöttämiseen käytetään graafisissa käyttöliittymissä erilaisia visuaalisia komponentteja. Käytettävien komponenttien ja grafiikkasymbolien tulisi kuvata toimintoa tai laitetta niin, että käyttäjä, joka on perehtynyt järjestelmän toimintaan tai toimintakaavioihin, osaa helposti määritellä kunkin komponentin toiminnon. Lisäksi on otettava mahdollisuuksien mukaan huomioon kansalliset standardit ja alalla vakiintuneet esitystavat. Pelkän graafisen ohjauksen lisäksi tulisi käyttöliittymässä olla mahdollisuus toimintojen suorittamiseen myös valikkoja käyttäen. Näin saavutetaan tietyissä usein toistuvissa ohjaus- ja käyttötilanteissa mahdollisuus oikaista syvemmälle järjestelmään pitkien kuvallisten valintapolkujen sijaan. (ST 22 2008, 23 - 31.)

### 3.2 Tekniset tavoitteet

Rakennusautomaatiojärjestelmien tekniset tavoitteet edellyttävät ohjattavan prosessin suunnittelua ennen käyttöjärjestelmän ja -liittymän valintaa. Käyttövarmuus on määräävä tekijä ja järjestelmän on oltava rakenteeltaan selkeä sekä helppokäyttöinen. Lisäksi on mietittävä tarkasti, mitä reaaliaikaista tietoa halutaan saada ja miten se esitetään. (ST 17 2001, 30 - 31.)

### 3.3 Trendiseuranta

Käsitteellä trendi ymmärretään kiinteistöautomaation yhteydessä yhden tai usean mittauksen arvon jatkuvaa seuranta ja seurantagraafin tulostusta. Kuviossa 4 on esitettyä trendikäyrästä lämmitysverkoston venttiilin avautumistila suhteessa aikaan. Kuvasta on helppo seurata miten venttiilin toimii eri tilanteissa. Trendikäyrästä on yleisesti saatavissa myös taulukkomuotoisena numeerisena tietona. Taulukkomuotoisen seurannan etuna on tarkempi käsitys siitä, millaisia arvoja järjestelmästä saadaan, kun trendikäyrästä on parhaimmillaan järjestelmän suhteellisen käyttäytymisen analysointiin. Dynaamisen trendin avulla voidaan seurata reaaliaikaisesti mittaus- ja ohjausarvoja, mikä helpottaa järjestelmän reaaliaikaista ”virittämistä”. Historiatrendit puolestaan kertovat miten järjestelmä on käyttäytynyt aiemmin ja miten sen käyttäytyminen on ajan saatossa muuttunut. (ST 22 2008, 23 - 31.)



KUVIO 4. Trendikäyrästä

### 3.4 Hälytystiedot ja niiden käsittely

Kiinteistöautomaatiojärjestelmään liitetään myös järjestelmien toiminnasta aiheutuvat hälytykset. Niitä voidaan yleensä tarkastella reaaliaikaisesti ja tarvittaessa hälytyshistoriaa apuna käyttäen. Hälytysten tulee tulostua automaattisesti näytölle, erilliseen hälytysikkunaan tai taulukkoon. Hälytystiedoille tulee suunnitella erillinen historiatiedosto, jolloin takautuva hälytystietojen käsittely on mahdollista.

Yleisimmät hälytystiedot ovat järjestelmän toiminnan kannalta kriittiset hälytykset, joiden huomioimatta jättäminen saattaa aiheuttaa vaaratilanteen joko järjestelmälle tai käyttäjille. Tällaisten hälytysten yhteyteen tulee kytkeä pakkokytkenä toiminto, joka estää vahingollisen toiminnan jatkumisen, mikäli käyttäjä ei huomaa hälytystä. Toiseksi yleisimmät hälytykset ovat niin sanottuja ristiriitahälytyksiä. Ristiriitahälytys voi syntyä esimerkiksi pumpun pysähtymisestä, vaikka järjestelmässä sen statukseksi on merkitty ”käy”. Lisäksi järjestelmissä on erilaisia mittaushälytyksiä ylä- ja alarajojen ylittyessä tai alittuessa, sekä mittausanturin vikaantuessa. (ST 22 2008, 51 - 62.)

### 3.5 Hälytysten seuranta

Hälytyksiä tulee voida listata näytölle käyttämällä mahdollisesti erilaisia suodatuksia. Hälytyksien listauksessa on syytä erotella aktiiviset käynnissä olevat hälytykset, kuitatut hälytykset ja järjestelmästä riippuen erikseen vielä korjatut hälytykset. Kiinteistöautomaatiojärjestelmissä on useimmiten käytössä vain aktiivisten ja kuitattujen hälytysten listat. Kuitattu hälytys katsotaan samalla myös laitetasolla korjatuksi. Järjestelmien kannalta erittäin kriittisiin hälytyksiin voidaan pakottaa tehtäväksi erillinen kuittaus laitetasolla. Tällaisia ovat esimerkiksi ilmanvaihtokoneen jäätymissuojahälytykset, joita ei voi kuitata valvomosta käsin, vaan ne on käytävä kuittaamassa ilmanvaihtolaitteen alakeskukselta tai ohjauskaapista. (ST 22 2008, 51 - 62.)

### 3.6 Tiedon reaaliaikaisuus ja käytettävyys

Olennaista käyttöliittymien käytettävyyttä arvioitaessa on esitettävien tietojen ajantasaisuus. Kiinteistöautomaation yhteydessä tiedon reaaliaikaisuusvaatimus on hyvin väljästi määriteltävissä. Kaiken käyttöliittymässä esitettävän dynaamisen tiedon reaaliaikaisuuteen pyrkiminen ei ole tarpeellista, eikä edes mahdollista. Tiedon reaaliaikaisuuden tarve riippuu tiedon sisällöstä ja sen vaikutuksesta järjestelmään ja sen toimintaan. Lämmitysjärjestelmien kohdalla täydelliseen reaaliaikaisuuteen pyrkiminen on turhaa, koska säätöjen ja mittausten vaihtelu on yleensä hyvin loivaa ja järjestelmä sallii melko suuren toleranssin mittausten ja

ohjausten välillä. Lämmitysjärjestelmissä säätölaitteiden ajoajat ovat yleisestikin useita kymmeniä sekunteja ääripäästä toiseen ja mittausten viive riippuu huomattavasti mittauksen fyysisestä sijoittumisesta järjestelmään. (ST 22 2008, 23 - 31.)

### 3.7 Käyttöliittymän yhteys tietokantaan

Kiinteistöautomaatiojärjestelmän antamaa tietoa kerätään järjestelmästä riippuen hyvin erilaisilla toimintatavoilla. Suuremmissa järjestelmissä tieto voidaan kerätä ja säilyttää ala-asemilla tai niiden tieto voidaan tallentaa suoraan keskusyksikön kautta tietokantaan. Ala-asemalle kerätty tieto voidaan vaihtoehtoisesti lähettää tietokantaan vain silloin, kun tietosisältö muuttuu olennaisesti tai määrävälein. Peruseriaatteena voidaan todeta, että käyttöliittymässä esitettävän tiedon tulee olla linkitettyä tietokantoihin siten, että käyttöliittymässä esitettävä tieto saadaan kerättyä talteen historia- ja trendiseurantaa varten. Tietoja voidaan tallentaa automaattisesti tietokantaan tarkasteluhetken todellista tilannetta vastaaviksi tai ne voidaan kerätä keskusmuistiin ja lähettää määrävälein tietokantaan. Jälkimmäisessä vaihtoehdossa on mahdollista kadottaa tietoa mahdollisen muistiin vaikuttavan laiterikon johdosta, mutta riittävän tiheällä päivitystaajuudella tämäkin mahdollinen haitta jää hyvin vähäiseksi. (ST 22 2008, 23 - 31.)

## 4 KÄYTTÖLIITTYMÄN LAITTEISTOT JA OHJELMAT

Tässä työssä käytettävä laitteisto koostuu kahdesta aurinkolämmitysjärjestelmästä, useista maalämpöpumppulaitteistoista, lämmitysvaraajista sekä tämän kehitystyön aikana toteutetusta käyttöliittymä-/ohjausjärjestelmästä. Varsinainen fyysinen hybridijärjestelmän rakentaminen putkitus-, moottori- ja venttiiliasennuksineen tullaan suorittamaan myöhemmin tämän suunnitelman pohjalta.

### 4.1 LabJack U3 tiedonkeruu- ja ohjauslaite

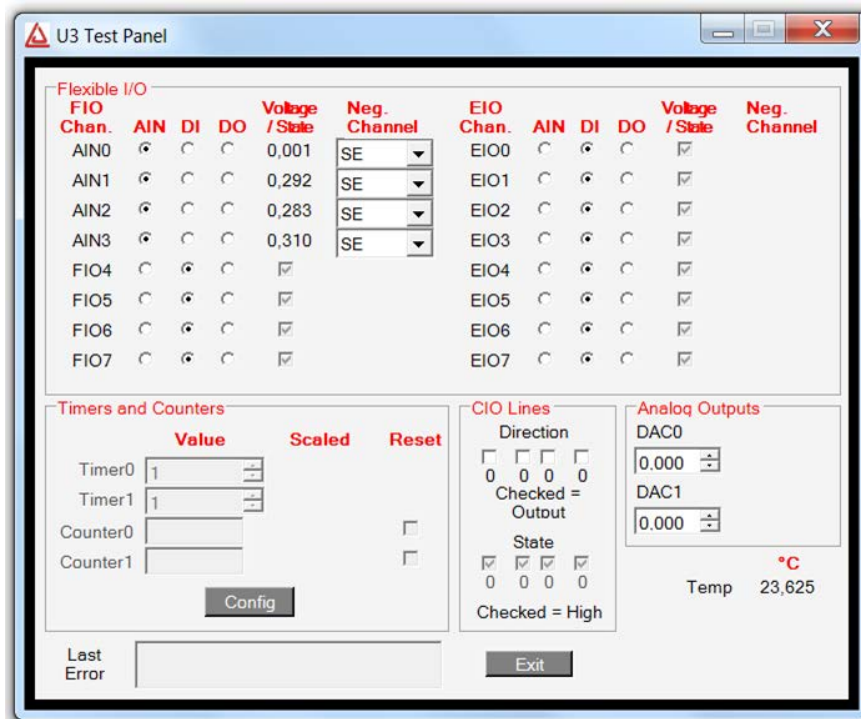
Ohjausjärjestelmän ytimeksi oli määritelty kuvion 5 mukainen LabJack U3-HV -tiedonkeruu- ja ohjauslaite. LabJack on edullinen, hieman yli 100 euron hintainen, USB- / Ethernet-pohjainen mittaus- ja automaatiolaitteiden alusta, joka mahdollistaa useiden sekä analogisten että digitaalisten tulo- ja lähtöpisteiden hyödyntämisen mittaukseen ja ohjaukseen. LabJack tarjoaa helposti muokattavan rajapinnan tietokoneen ja varsinaisen fyysisen laitteiston välille. Lisäksi se tarjoaa monille ohjelmointikielille valmiit luokkakirjastot, joiden avulla voidaan rakentaa kulloinkin tarvittavat ohjaukset ja liittymät järjestelmään halutulla ohjelmointikielellä ja alustalla. LabJack -laitteiston laajentaminen on mahdollista lisäämällä alustaan valmiina saatavia lisäkomponentteja tarpeen mukaan. (LabJack Co. 2012.)



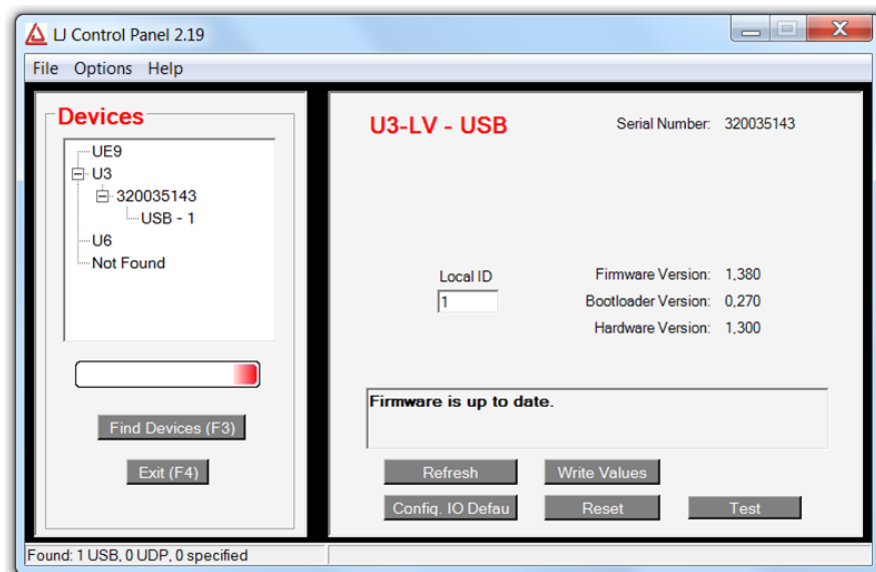
KUVIO 5. LabJack U3-HV -tiedonkeruu- ja ohjauslaite

LabJack -tiedonkeruu- ja ohjauslaitteistoa voidaan testata tietokoneessa liittämällä laitteisto USB-porttiin ja käynnistämällä kuvioissa 4 ja 5 esiintyvät ohjaus- ja testausohjelmat LJ Control Panel ja U3 Test Panel. (LabJack Co. 2012.)





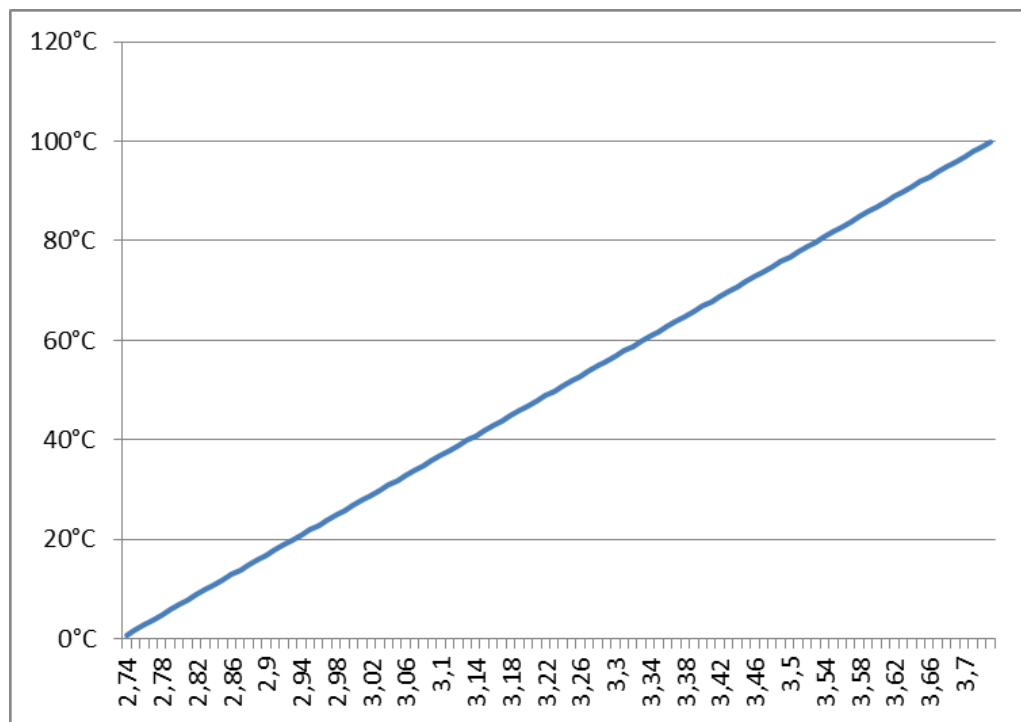
KUVIO 6. U3 Test Panel



KUVIO 7. LJ Control Panel 2.19

#### 4.1.1 Lämpötilamittaus

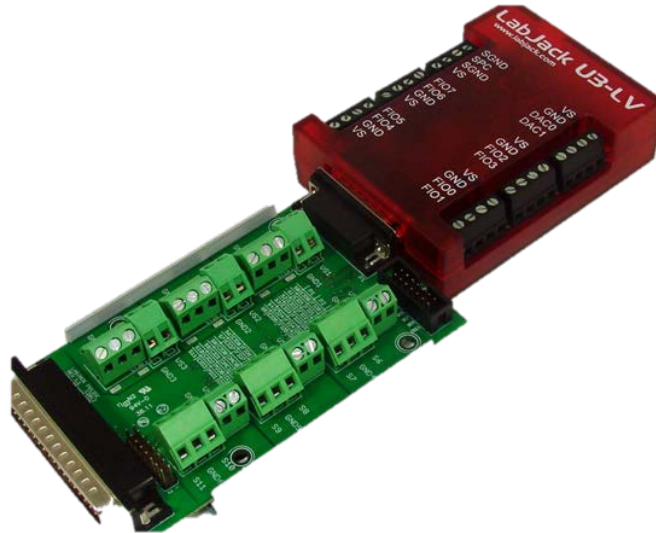
LabJack U3 -dokumentaatiossa on esimerkit lämpötila-antureiden ja relekytkentöjen suorittamiseen. Lämpötilamittausten suorittamiseksi käyttöön valittiin lineaarinen National Semiconductorin valmistama EI-1022 -lämpötila-anturi, joka koostuu muoviputkesta ja lämpötilan mukaan säätävästä vastuksesta. Anturin mittausalue on kiinteistöautomaatiolle sopiva  $-40 - 100\text{ }^{\circ}\text{C}$  ja lämpötilamuutos  $\pm 1\text{ }^{\circ}\text{C}$  vastaa  $\pm 10\text{ mV}$ :n jännitemuutosta. Jännitemittauksen muuttaminen celsiusasteiksi tapahtuu kaavalla  $^{\circ}\text{C} = 100 * \text{mitattu jännite} - 273.15$ . Kuviossa 8 on esitettyä EI-1022:n lämpötilan muutoksen vastaavuus mittausanturin ulostulon (output) antamaan jännitearvoon. (LabJack Co. 2012.)



KUVIO 8. EI-1022:n lämpötilavastaavuus

#### 4.1.2 Pumppujen ja venttiilien ohjaukset

Releohjaukset magneettiventtiileille ja kiertovesipumpuille voidaan tehdä usealla tavalla. Yksi vaihtoehto on käyttää kuviossa 9 esitettyä LabJack PS12DC -relealustaa.



KUVIO 9. LabJack PS12DC -relealusta kytkettynä

Alustalla on mahdollista ohjata 12:ta digitaalista lähtöä. Samalla alusta sallii vaihtelevat virtalähteiden valinnat 5 ja 28 V DC:n välillä. Kiinteistöautomaatiojärjestelmissä on yleisesti käytössä 24 V:n jännitettä käyttävät toimilaitteet ja järjestelmäkomponentit, joten laitteisto sopii hyvin käytettäväksi näiden kanssa. Relealusta voidaan kiinnittää LabJack U3:n DB15 -liittimen kautta. (LabJack Co. 2012.)

## 4.2 Aurinkolämmitysjärjestelmät



KUVIO 10. Ekowatti -aurinkolämpövaraaja ja aurinkokeräin

Aurinkolämmityksessä käytetään kuvissa 10 esitettyä Ekowatti- ja kuviossa 11 esitettyä Vitocelli -aurinkolämpövaraajia ja aurinkokeräimiä. Molemmat laitteistot ovat itsenäisesti toimivia aurinkolämmitysjärjestelmiä, joita ohjataan niiden omilla yksikkösäätimillä. Aurinkokeräimiä ei ole voitu koulutuksellisista syistä sijoittaa kiinteistön katolle, minne ne normaalisti asennetaan. Tämän vuoksi auringon lämpövaikutus on korvattu erillisillä lämpösäteilijöillä. Lämpösäteilijöiden käyttö mahdollistaa lämpösäteilyn tarkan annostelun ja aurinkolämmityksen käytön myös pimeänä talviaikana.



KUVIO 11. Vitocelli -aurinkolämpövaraaja ja aurinkokeräin

#### 4.3 Lämpöpumppulaitteistot

Lämpöpumppulaitteistot on asennettu kuvion 12 mukaisesti keinotekoiseen maapiiriin, joka mahdollistaa maalämpöpumppujen testaamisen ja käyttämisen ilman maahan tai porakaivoon asennettua keruupiiriä. Keinotekoinen maapiiri mahdollistaa myös kaikille maalämpöpumpuille yhtäläisen keruupuolen lämpötason, mikä edesauttaa maalämpöpumppujen testaamista ja vertailemista. Käytössä on myös ilmasta veteen lämpöpumppuja ja perinteisiä ilmalämpöpumppuja, joista jälkimmäiset eivät tekniikkansa puolesta ole tässä työssä mukana.



KUVIO 12. Maalämpöpumppuja ja niiden keruupiiri

Lämpöpumppulaboratoriossa on kaikille laitteille yhteinen lämmön tuottopiiri, joka palauttaa osan tuotetusta lämpöenergiasta takaisin maapiiriin. Järjestelmään tullaan asentamaan myöhemmin fyysisen laiteasennuksen yhteydessä aiemmin kuviossa 2 esitetyn toimintakaavion mukainen erillinen varastovaraaja, josta lämpö jaetaan kaukolämmitysvaraajalle. Erillinen varastovaraaja mahdollistaa maalämpöpumppujen käyttämisen ja lämmön keräämisen silloinkin, kun kaukolämpövaraajalle ei tarvitse tuottaa energiaa, ja vastaavasti kaukolämpövaraajalle voidaan siirtää lämpöenergiaa ilman, että maalämpöpumput ovat toiminnassa.

#### 4.4 Kaukolämmitys ja hybridivaraaja

Kuviossa 13 on kaukolämmityksen, ilmanvaihtotekniikan ja kiinteistöautomaation opetukseen käytettävä Danfoss LPM -kaukolämpöpaketti, joka saa kaukolämpöenergian kuvassa taustalla olevasta erillisestä kaukolämmitysvaraajasta. Laitteiston tarvitsema kuuma vesi, 70 – 90 °C, tuotetaan kaukolämpövaraajaan kahdella



sähkövastuksella, joiden toiminta-aika on asetettu koulutuskäytössä olevassa Fidelix-kiinteistöautomaatiojärjestelmässä. Sähkövastukset huolehtivat itsenäisesti lämmitystarpeesta sisäänrakennetun lämpötilamittauksen ja lämmitysohjauksen mukaisesti. Kaukolämpövaraajana käytetään AKVA SOLAR PLUS -hybridivaraajaa. Varaajassa on kolme kierukkaa, joihin voi kytkeä käyttöveden lämmityksen, lämmityksen ja sen lisäksi aurinkokeräimiltä tulevan lämmityksen. Kuten aiemmin lähtöteoriassa todettiin (sivu 4 - 5) on maalämpöpumppujen tuottaman lämmitysenergian lämpötila tehokkaimmillaan alhaisissa lämpötiloissa ja vastaavasti aurinkolämmityksen tuottama lämpötila korkeissa lämpötiloissa. Tämän teorian mukaisesti maalämpöpumppujen tuottama energia johdetaan hybridivaraajan esilämmitysosaan ja aurinkolämmityksen tuottama energia vastaavasti aurinkolämmitysenergia hybridivaraajan jälkilämmitysosaan.



KUVIO 13. Kaukolämpövaraaja, kaukolämpöpaketti ja ilmanvaihtokoje

#### 4.5 Käyttöliittymäohjelman ohjelmointi

Käyttöliittymä päätettiin toteuttaa Windows-työpöytäsovelluksena .NET-ympäristössä. Ohjelmointikieleksi valittiin C#.

Ohjelmointikielenä käytetty C# (Sharp) perustuu C++-kieleen ja sisältää Java-kielen piirteitä. .NET on alunperin Windows-käyttöjärjestelmien kehitysympäristöksi luotu sovelluskehys. Se mielletään turvalliseksi, koska erillinen virtuaaliko- ne huolehtii eri ohjelmointimoduulien ajamisesta riippumatta käytettävän laitteis- ton järjestelmätasosta. Tämä puolestaa mahdollistaa ohjelmamoduulien kirjoitta- misen eri ohjelmointikielillä. Näiden lisäksi .NET-sovelluskehitys on suunniteltu toimivaksi hajautetuissa ympäristöissä, joissa tietokoneet ja laitteet ovat yhteydes- sä Internetin kautta. (Wille 2001, 6 – 11.)

#### 4.5.1 Linkkikirjasto LJUDDotNet

LabJackin toimittajalta on saatavissa ohjelmakirjastot usealle eri ohjelmointikie- lelle. Tämän työn suorittamiseksi asennettiin ohjelmointiympäristöön C#-kielelle suunniteltu LJUDDotNet -aliohjelmakirjasto. Dynaaminen linkkikirjasto (LJUD- DotNet.dll) asennettiin LABJACK \ drivers-kansioon. LabJackUD kirjaston ja USB-porttiin liitetyn tiedonkeruulaitteen alustuksen esimerkkilistaus on esitetty kuviossa 14.

Kuvion 14 rivillä 70 suoritetaan USB-porttiin liitetyn LabJack-laitteiston avaami- nen. Muuttujaan u3 luetaan osoite, jonka avulla laite tunnistetaan ohjelman suori- tusvaiheessa.

Rivillä 73 palautetaan ohjauslaite tehdasasetuksiin. Tehdasasetuksissa kaikki ajas- timet ja laskurit nollautuvat ja muutettavat kanavat 0 – 15 asetetaan digitaalisiksi.

Rivillä 76 konfiguroidaan kanavat 0-3 analogisiksi ja muut kanavat 4-15 digitaali- siksi. Tämä tapahtuu asettamalla PUT\_ANALOG\_ENABLE\_PORT-muuttujan jälkeinen parametri nolllaksi (0), mikä kertoo sen olevan analoginen. Vastaava digitaalinen portti olisi arvoltaan yksi (1). Seuraavaksi annetaan luku 15, joka muodostuu bittien 0000 0000 0000 1111 tuloksesta, joka on 15. Viimeinen para- metri määrittelee, että kaikki 16 bittiä ovat asetuksessa mukana.



```

40 using System;
41 using System.Drawing;
42 using System.Collections;
43 using System.ComponentModel;
44 using System.Windows.Forms;
45 using System.Data;
46 using LabJack.LabJackUD; //otetaan labJackUD aliohjelmakirjasto käyttöön
47
48 namespace AmiHybridi
49 {
50     public AmiHybridi()
51     {
52         InitializeComponent();
53     }
54
55     // virheen käsittely ja tulostus
56     public void ShowErrorMessage(LabJackUDEException exc)
57     {
58         errorDisplay.Text = exc.ToString();
59     }
60
61     private void kaynnista_ohjelma(object sender, EventArgs e)
62     {
63         double dblDriverVersion;
64         LJUD.IO ioType = 0;
65         LJUD.CHANNEL channel = 0;
66
67         try
68         {
69             //Avataan ensimmäinen löydetty LabJack U3.
70             u3 = new U3(LJUD.CONNECTION.USB, "0", true); // Yhteys USB:n kautta
71
72             //Resetoidaan LabJack tehdasasetuksiin.
73             LJUD.ePut(u3.ljhandle, LJUD.IO.PIN_CONFIGURATION_RESET, 0, 0, 0);
74
75             //Konfiguroidaan pinnit 0-3 analogisiksi.
76             LJUD.ePut(u3.ljhandle, LJUD.IO.PUT_ANALOG_ENABLE_PORT, 0, 15, 16);
77
78         }
79         catch (LabJackUDEException exc)
80         {
81             ShowErrorMessage(exc);
82             return;
83         }
84     }
85 }
86
87

```

KUVIO 14 LabJackin alustus ja käyttöönotto ohjelmassa

#### 4.5.2 Tärkeimmät tiedonkeruujärjestelmän ohjausfunktiot

LabJackin funktiokirjastossa on useita valmiita luku- ja ohjausfunktioita, joilla voidaan kerätä tietoa tai ohjata LabJack U3-HV -tiedonkeruu- ja ohjauslaitteesta. AmiHybridi-käyttöliittymän kannalta olennaisimmat ovat lämpötilojen lukemiseen ja releiden ohjaukseen tarvittavat ohjausfunktiot ja erilaiset ajastinfunktiot. Liitteessä 1 on laajempi aliohjelmakirjaston mukana tuleva ohjelmafunktioista.

LabJack U3-HV:n testausvaiheen lämpötilojen mittausta varten kirjastosta valittiin käyttöön LJUD.eAIN()-funktio, joka saa parametreikseen ohjauslaitteen osoit-

teen, positiivisen kanavan, negatiivisen kanavan, muuttujan nimen sekä erilaisia digitaaliseen muotoon ja esittämistapaan vaikuttavia arvoja. Analogiatulot ovat normaalisti noin 0 - 2.44 voltia, mutta olemassa on myös erityinen 0 - 3,6 voltin alue, sekä mahdollisuus myös negatiivisiin alueisiin, jolloin jännitealue on -2,44 voltista +2,44 volttiin. Käytettäessä AIN()-funktioita asetetaan negatiivinen kanava arvoon 31, jolloin käytetään vain positiivista jännitealuetta.

```

300
301
302
303 //Luetaan portti 15 ja tulostetaan sen palauttama arvo.
304 binary = 0;
305 LJUD.eAIN(u3.ljhandle, 15, 31, ref dblValue15, -1, -1, -1, binary);
306 ain15Display.Text = String.Format("{0:0.###}", dblValue15);
307
308
309
310

```

KUVIO 15. Mittaustiedon luku ja tulostus

Funktio palauttaa kuviossa 15 kuvatun esimerkin tapauksessa mittaustuloksen annettuun muuttujaan (dblValue15). Ohjelmarivi 305 hakee tiedonkeruulaitteen kanavasta 15 jännitteen ja tulostaa sen rivillä 306 ain15Display.Text -kenttään halutulla muotoilulla.

Vastaavasti ohjauskäskyt releille, joiden avulla voidaan ohjata venttiilimoottoreiden tai vastaavien toimilaitteiden toimintaa, voidaan asettaa esimerkiksi kuvion 16 mukaisesti. Rivillä 489 käytetään eDO()-funktioita, joka saa parametreikseen ohjauslaitteen osoitteen, ohjattavan laitepisteen ja ohjaustilan tiedon.

```

483
484
485 //Asetetaan eDO 9 ohjaamaan relelähtöä ja ilmoitetaan dio9Display kentällä tämä
486 // tieto käyttäjälle.
487 ChannelInt = 9;
488 StateInt = 1;
489 LJUD.eDO(u3.ljhandle, ChannelInt, StateInt);
490 dio9Display.Text = "KÄY";
491
492
493

```

KUVIO 16. Jännitteellinen ohjaustieto

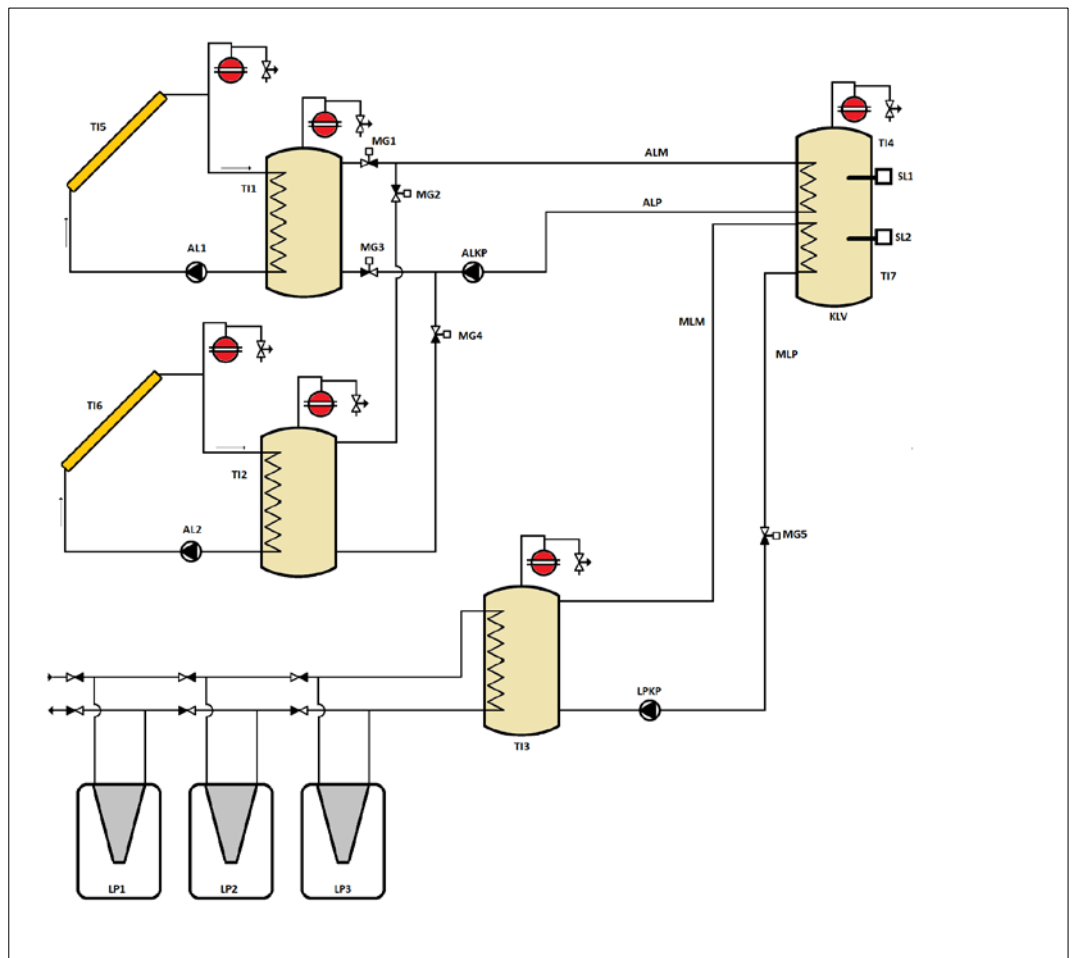
LabJackin digitaalinen input/output-tieto voidaan yksilöllisesti konfiguroida. Konfiguraatiolla voidaan määritellä tilat luku (input), korkea lähtö (output high) tai

matala lähtö (output low). Digitaalilähdössä ohjaustilan ollessa korkea lähtö tuottaa jännitteen (yleensä noin 5 - 3,3 voltia). Kun ohjaustila on määritelty matalaksi, se ei tuota jännitettä, vaan on kytkettynä maapotentiaaliin.

## 5 KÄYTTÖLIITTYMÄN SUUNNITTELU

### 5.1 Hybridijärjestelmän käyttöliittymän grafiikka

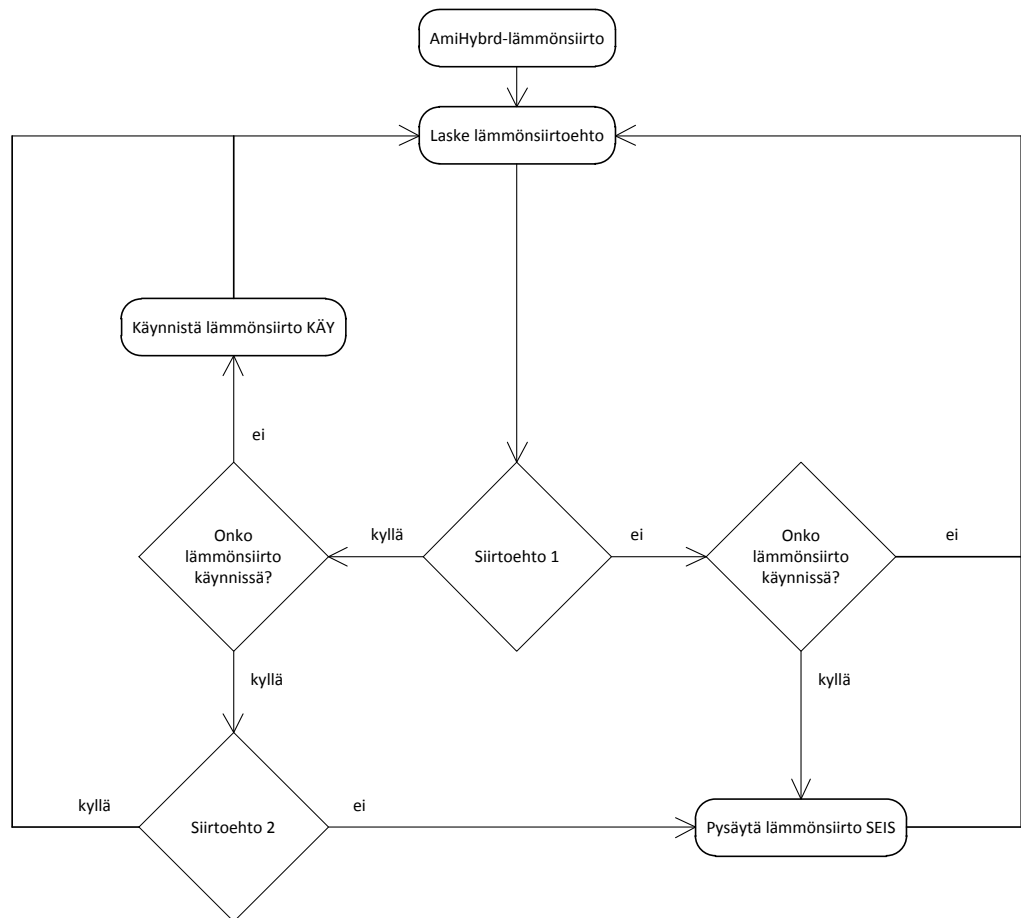
Hybridijärjestelmän käyttöliittymäsuunnittelun lähtökohdaksi valittiin järjestelmän toimintaa mahdollisimman hyvin kuvaava toimintakaavio ja sen pohjalta laadittu pohjakuva. Kuviossa 17 on esitettyä toimintakaavio, jonka perusteella käyttäjällä on mahdollisuus yhdellä silmäyksellä selvittää, missä tilassa hybridilämmitysjärjestelmän eri osat kulloinkin ovat. Toimintakaavioon on merkittyä mittaus- ja ohjauspisteet, joiden perusteella tehtiin myöhemmin käyttöliittymän vastaavat tietokentät.



KUVIO 17. AmiHybridi -toimintakaavio

## 5.2 Hybridijärjestelmän toimintaselostus

Käyttöliittymäohjelman toiminta perustuu kuvion 18 mukaiseen lämmönsiirtoehtojen vertailuun. Siirtoehtojen toteutuessa lämmönsiirto on käytössä. Sekä lämpöpumppuvaraajasta että aurinkolämpövaraajasta tehtävälle lämmönsiirrolle käytetään erillisiä lämmönsiirtovertailuja, joten molemmat lämmönsiirrot voivat toimia täysin itsenäisesti. Itsenäisen ja mahdollisesti yhtäaikaisen lämmönsiirron mahdollistaa kaukolämmitysvaraajan kaksi erillistä lämmönvaihdinpiiriä.



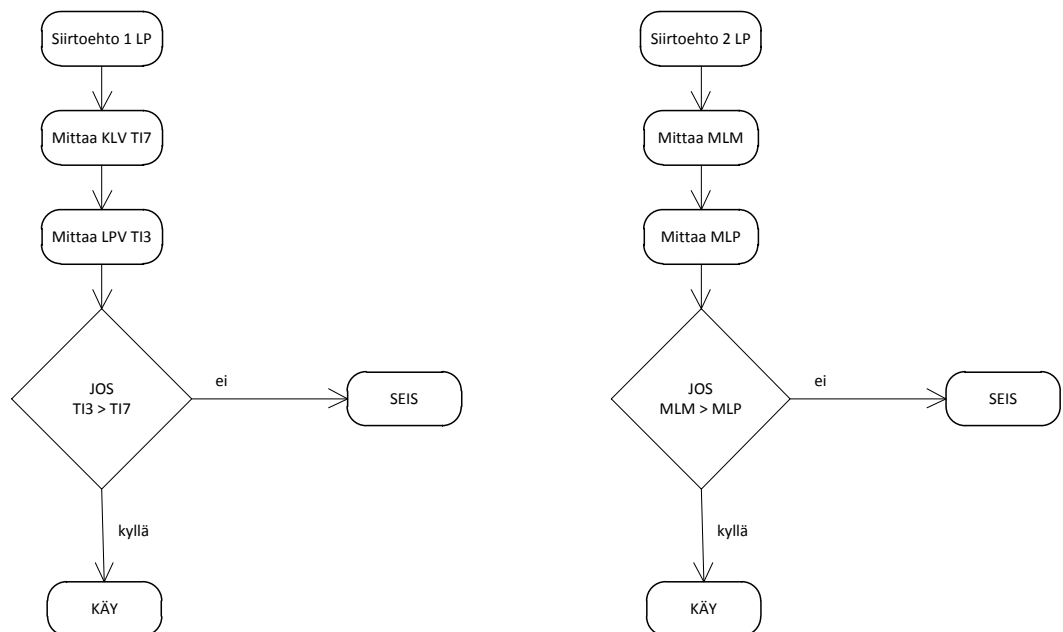
KUVIO 18. AmiHybridi-järjestelmän lämmönsiirto

Kuviossa 18 siirtoehto yksi vertaa lämpöpumppu- tai aurinkolämpövaraajan lämpötiloja suhteessa kaukolämmitysvaraajan lämpötilaan. Mikäli kaukolämmitysvaraajan lämpötila on alhaisempi, ehto toteutuu ja siirto on mahdollinen. Seuraavaksi tarkistetaan, onko lämmönsiirto jo mahdollisesti käynnissä. Siirtoehto 2 vertaa kaukolämmitysvaraajalle menevän veden ja sieltä palaavan veden lämpötilaa. Mi-

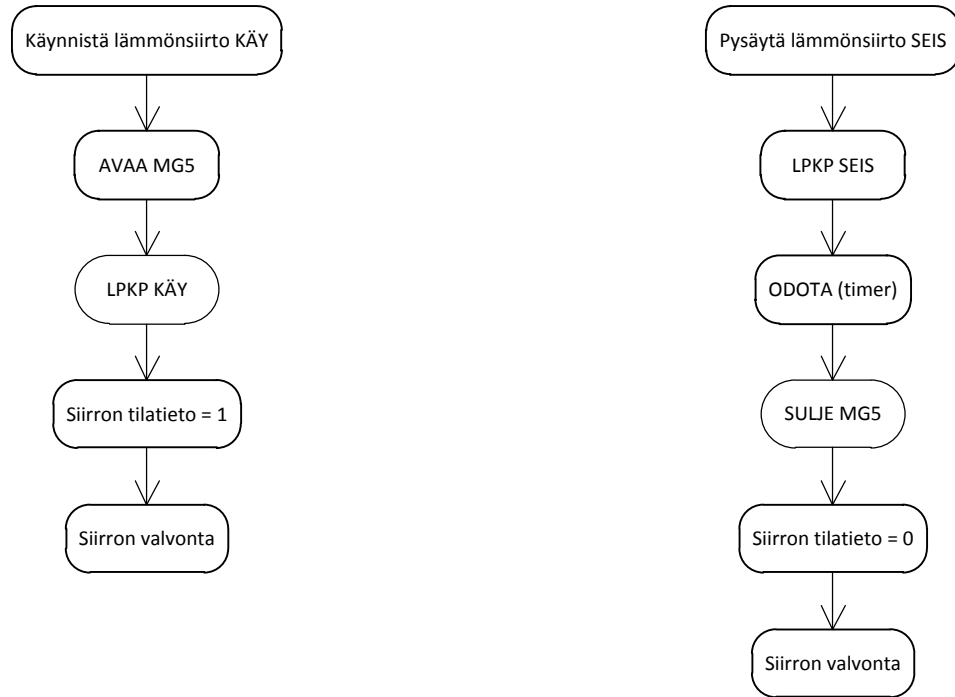
käli menovesi on lämpimämpää kuin paluuvesi, ehto toteutuu ja siirto voi jatkua. Kahden siirtoehdon käyttö mahdollistaa tarkemman siirtovalvonnan ja ehkäisee samalla tarpeettoman siirron syntymisen siinä vaiheessa, kun lämpötilat varaajissa tai siirtoputkistoissa ovat hyvin lähellä toisiaan.

### 5.2.1 Lämpöpumppuvaraajan lämmönsiirto

Lämpöpumppuvaraajan lämmönsiirron ohjaus järjestetään kuviossa 19 ja kuviossa 20 esitettyjen toimintakaavioiden mukaisesti. Käyttöliittymäohjelma mittaa kaukolämmitysvaraajan esilämmitysosan lämpötilaa mittausturinin (TI7) avulla. Saatua arvoa verrataan lämpöpumppuvaraajan lämpötila-anturin (TI3) mittausturinin arvoon. Mikäli kaukolämpövaraajan lämpötila on alhaisempi kuin lämpöpumppuvaraajan lämpötila, avaa järjestelmä magneettiventtiilin (MG5) ja käynnistää lämpöpumppuveden kiertovesipumpun (LPKP).



KUVIO 19. Lämpöpumppuvaraajan siirtoehdot



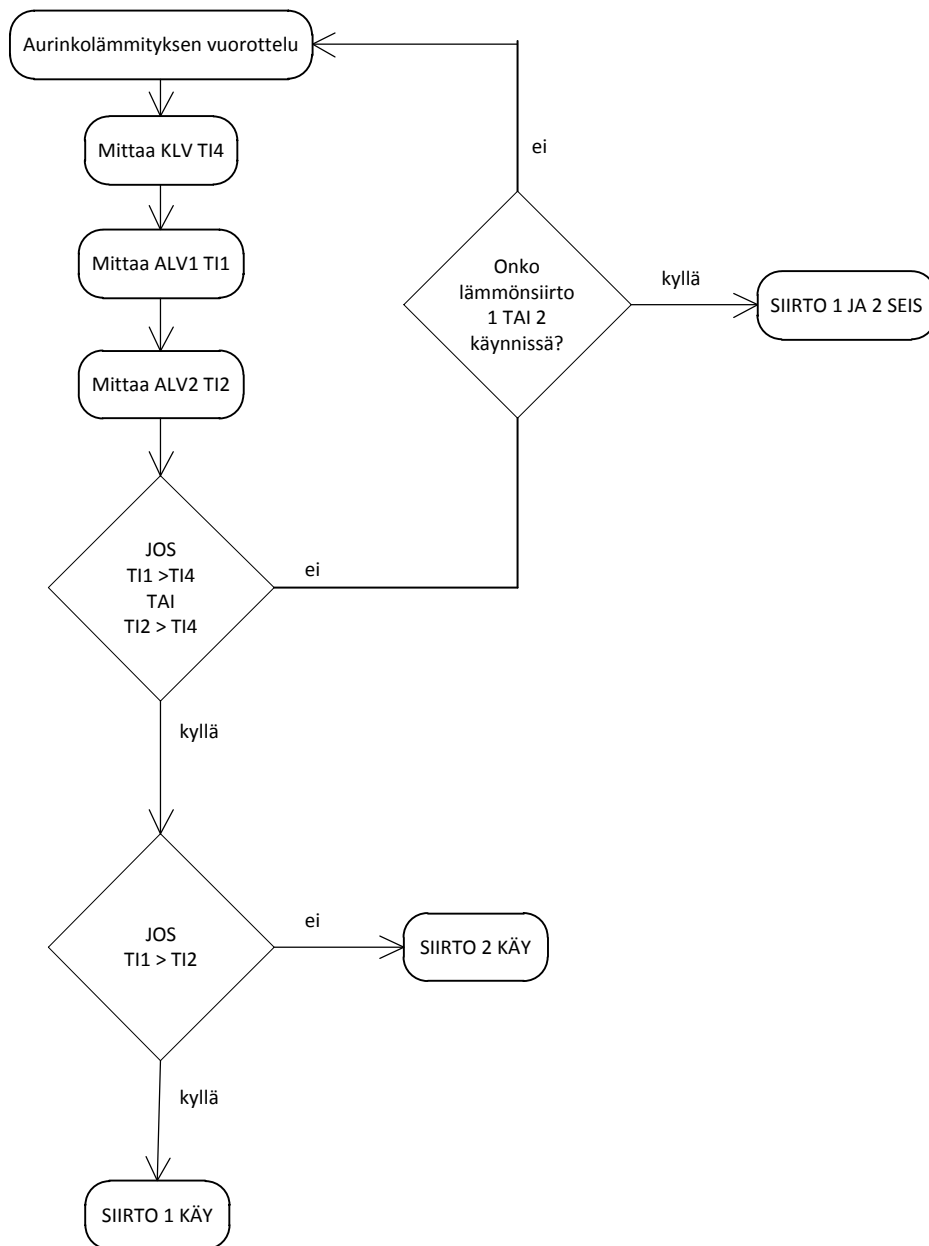
KUVIO 20. Lämpöpumppusiirron käyntiohjaukset

Lämmitysveden siirtoehto on voimassa niin kauan kun kaukolämpövaraajan esilämmitysosan lämpötila (TI7) on pienempi kuin lämpöpumppuvaraajan (TI3) lämpötila. Siirtoehdon poistuessa pysäytetään lämpöpumppuveden kierto-vesipumppu. Tämän jälkeen odotetaan asetuksissa määritellyn sulkuviiveen ajan, ennen kuin magneettiventtiili suljetaan. Sulkuviiveen tavoitteena on saada pumpun putkistoon aiheuttama paine tasaantumaan ennen magneettiventtiilin sulkeamista.

### 5.2.2 Aurinkolämpövaraajien vuorottelu

Aurinkolämpövaraajien vuorottelussa vertaillaan ensin varaajien (TI1 ja TI2) lämpötiloja kaukolämpövaraajan (TI4) lämpötilaan. Mikäli toinen tai molemmat aurinkolämmitysvaraajat ovat lämpötiloiltaan suuremmat kuin kaukolämmitysvaraajan jälkilämmitysosa, aloitetaan siirto. Siirtoehdon toteutuessa valitaan aina enemmän lämpöenergiaa sisältävä aurinkolämpövaraaja siirron kohteeksi. Siirron vuorottelu toteutetaan kuvion 21 mukaisesti. Aurinkolämmityksen siirtovaiheessa huolehditaan, ettei yhtäaikainen siirto ole voimassa molemmissa aurinkolämmitys-

piireissä. Aurinkolämmityksen vuorottelu huolehtii myös lämmönsiirron katkaisemisesta, mikäli lämmönsiirtoehto ei toteudu.



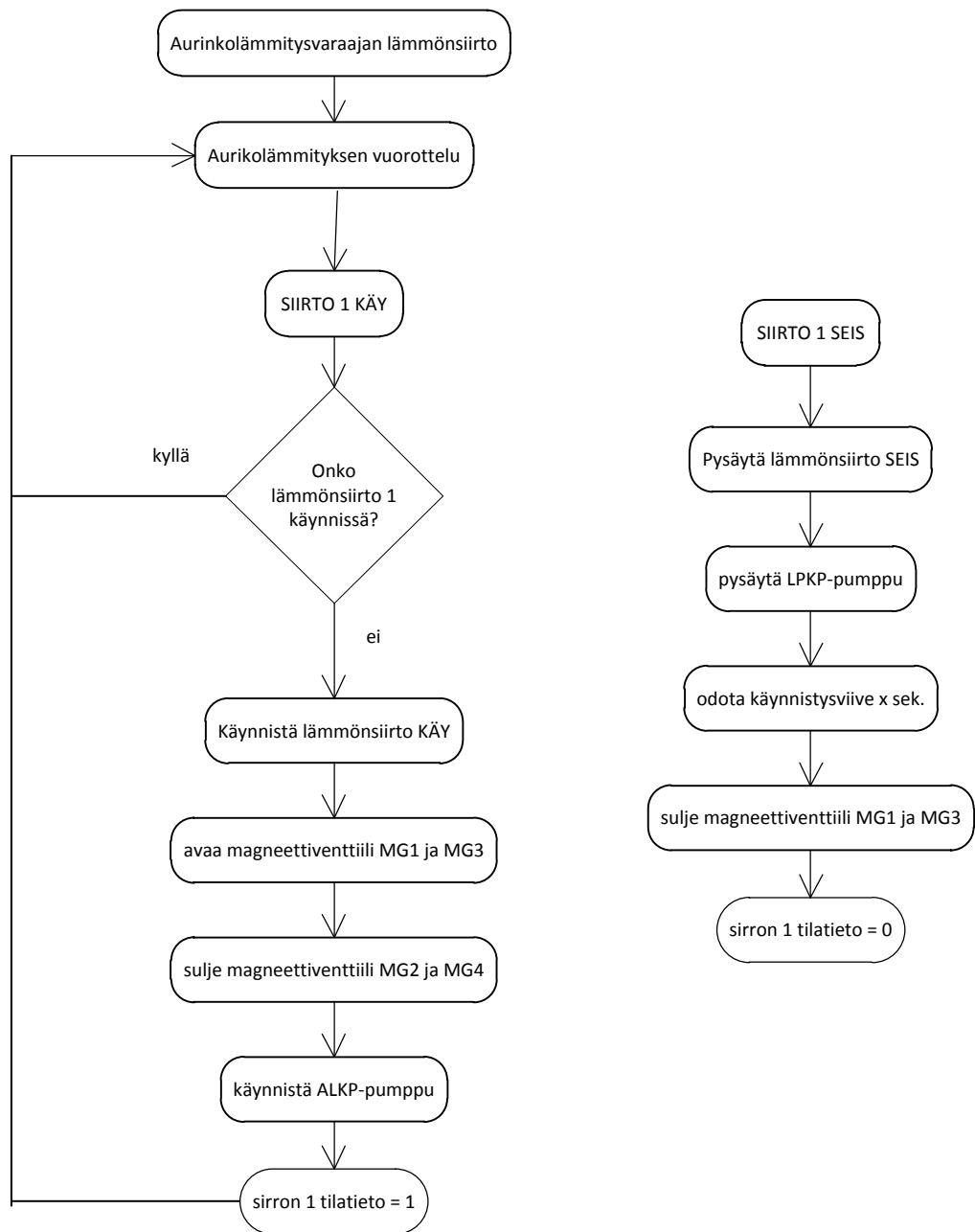
KUVIO 21. Aurinkolämmityksen vuorottelu

### 5.2.3 Aurinkolämmitysten lämmönsiirto

Kuviossa 22 on esitettyä vuorottelun jälkeiset aurinkolämmityksen (1) siirtopahtumat. Mikäli aurinkolämmityksen vuorottelu sallii siirron, avataan magneetti-



venttiilit (MG1 ja MG3), suljetaan magneettiventtiilit (MG2 ja MG4) ja käynnistetään aurinkolämpöveden kiertovesipumppu (ALKP).



KUVIO 22. Aurinkolämmityksen lämmönsiirto 1

Lämmitysveden siirtoehtoa valvotaan edellä mainitun ohjelman lisäksi kuvioden 18 ja 19 mukaisesti, menoveden ja paluueden lämpötilaohjauksella. Siirtoehto on voimassa niin kauan, kun aurinkolämmityksen menoveden lämpötilamittaus (ALM) on suurempi kuin kaukolämmitysvaraajasta palaavan lämpötilamittauksen

(ALP) arvo ja kaukolämpövaraajan jälkilämmitysosan lämpötila (TI4) on pienempi kuin aurinkolämpövaraajan (TI1 tai TI2) lämpötila. Siirtoehdon poistuessa suljetaan magneettiventtiilit ja samalla viiveen jälkeen pysäytetään kiertovesipumppu (APKP). Tavoitteena on estää lämmön siirtyminen kaukolämpövaraajasta (KLV) takaisin aurinkolämpövaraajaan.

#### 5.2.4 Sähkölämmityksen ohjaukset

Kaukolämpövaraajan sähkölämmitykset (SL1 ja SL2) kytkeytyvät päälle ajastimen mukaan, mikäli varaajaan ei siirretä aurinkolämpöä tai lämpöpumppulämpöä. Sähkölämmitysvastukset on varustettu lämpötermostaateilla, jotka huolehtivat lämmityksen katkaisemisesta ja päälle kytkennästä niihin asetetun asetusarvon mukaisesti.

#### 5.2.5 Siirretyn lämmitystehon laskenta ja seuraaminen

Siirretyn lämmitystehon laskenta suoritetaan ohjelmassa aina, kun siirtoa tapahtuu. Laskentaan käytetään kaavaa  $(4,19 \text{ kJ}/\Delta\text{K} * m)$ , missä  $4,19 \text{ kJ}$  = veden ominaislämpökapasiteetti,  $\Delta\text{K}$  = lämpötilaero ja  $m$  = siirretyn veden kokonaismassa. Laskutoimitus antaa tulokseksi kilojoulea (kJ), joka muunnetaan kWh:ksi jakamalla tulos 3600 sekunnilla. (Hautala, Peltonen & Lahden Teho-Opetus Oy 2002, 160 -161.)

*Esimerkki:  $(4,19\text{kJ}/\text{Km} * 350 \text{ kg(litraa)} * 5 \text{ }^\circ\text{C}) / 3600 \text{ s} = 2,44 \text{ kWh}$*

Laskutoimitusta varten ohjelma laskee siirtoon käytetyn ajan ja perustiedoissa annetun veden virtaaman perusteella veden kokonaismassan. Lämpötilan muutos lasketaan kaukolämpövaraajalle menevän menoveden ja sieltä palaavan paluuveden lämpötilaerona. Lämpötilan muutos on keskiarvo meno- ja paluuveden mitatuista lämpötiloista. Lämpötilaeroa ei voida laskea varaajien lämpötila muutoksina, koska varaajilla saattaa olla lämpötilaa kohottavaa tai sitä alentavaa kulutusta ja tuottoa samanaikaisesti. Laskennassa vesimäärä saadaan mittaamalla putkiston virtaama litraa/sekunnissa. Seuraavassa esimerkissä on laskettu 0,3 litraa sekun-

nissa virtaavalle vedelle energiamäärä, kun lämpötila alenee putkistossa 5 °C. Saatu arvo lisätään kumulatiiviseen lämmitysenergian muuttujaan yhteismäärän laskentaa varten.

*Esimerkki:  $(4,19\text{kJ/Km} * 0,3\text{ kg(litraa)} * 5\text{ }^{\circ}\text{C}) / 3600\text{ s} = 0,00174\text{ kWh}$*

### 5.3 Asetukset ja asetustiedostot

Käyttöliittymän asetukset tallennetaan omaan tekstitiedostoon, joka sijoitetaan käyttöliittymän asennushakemiston juureen. Asetustiedostoista kopioidaan jokaisen tehdyn tallennuksen lisäksi aikaleimalla varustettu historiatiedosto samaan kansioon. Näin on mahdollista myöhemmin tarkastella tehtyjen asetusmuutoksien vaikutuksia ohjelman toimintaa ja energian siirtomuutoksiin ja palauttaa järjestelmä aikaisemmin käytössä olleeseen tilaan tai perusasetuksiin.

Asetustiedosto sisältää:

- viimeisen muutoshetken aikaleiman
- lämpötilamittauksien raja-arvot, hälytysrajat, ala- ja ylärajat
- magneettiventtiilien käyttöviiveen
- siirtopumppujen käyttöviiveen
- käyttöliittymän virkistystaajuuden
- trendiseurannan tallennustaajuuden
- kaukolämpövaraajan viikoittaiset toiminta-ajat ja poikkeuspäivät.

### 5.4 Tietokanta

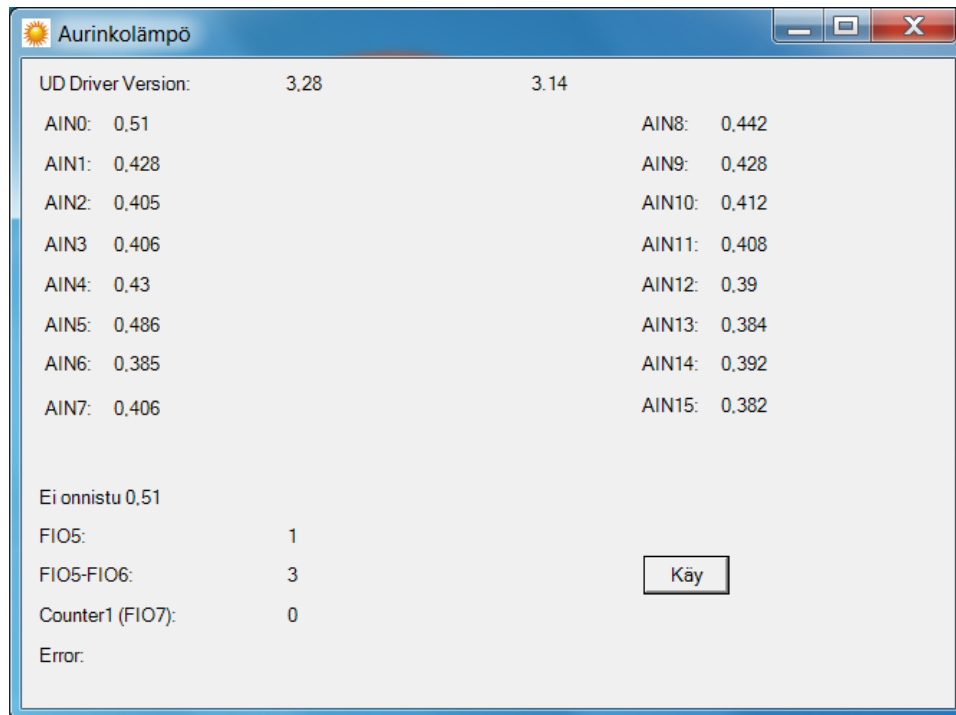
Järjestelmä tarvitsee tietojen tallennukseen tietokantaa, mistä tiedot on helposti noudettavissa myöhemmin käytettäväksi. Tietokannan sijaan tietojen tallennusmuodoksi valittiin XML-tiedostomuoto (eXtensible Markup Language). Näin tiedostojen käsittelyyn ei tarvita tietokannan hallintajärjestelmää. Valintaan vaikutti ensisijaisesti mahdollisuus lukea ja purkaa tiedoston sisältö millä tahansa XML-muotoista tekstiä tunnistavalla ohjelmalla opetustilanteessa ja se, ettei ohjelmassa nyky muodossaan tarvita käyttöoikeuksien valvontaa tai rajoituksia. Tal-

lennettujen historiatietojen seurannan mahdollisuus ilman tietokannan hallintaohjelmaa on erilaisissa käytännön opetustilanteissa lähes välttämättömyys.

XML-tiedostojen käyttöön .NET-sovelluskehitys tarjoaa XML-rajapinnan, jonka avulla voidaan kätevästi hakea, lisätä, poistaa ja muokata tietoja. XML-rajapinta sisältää runsaasti ohjelmointia helpottavia aliohjelmiä. Esimerkiksi XML-skeema, joka sisältää tiedot sarakkeiden nimistä ja tietotyypeistä, samoin kuin taulukoiden sisällöistä, voidaan kirjoittaa yhdellä kertaa WriteXML()-metodin avulla. Samoin voidaan taulukoissa oleva tieto lukea sarakkeiden nimineen ja sisältöineen tietojoukkoon ReadXML-metodilla. (Moghadampour 2011, 464.)

## 6 KÄYTTÖLIITTYMÄN TOTEUTUS

Käyttöliittymän toteutus aloitettiin testaamalla LabJack -ohjaus- ja tiedonkeruulaitteen toimivuus ja yhteyden muodostus. Testausta varten laadittiin yksinkertainen kuvan 23 mukainen käyttöliittymä, joka tehtävänä oli hakea kaikki LabJackin mittaus- ja ohjauspisteet ja tulostaa niiden jännitetiedot näytölle.

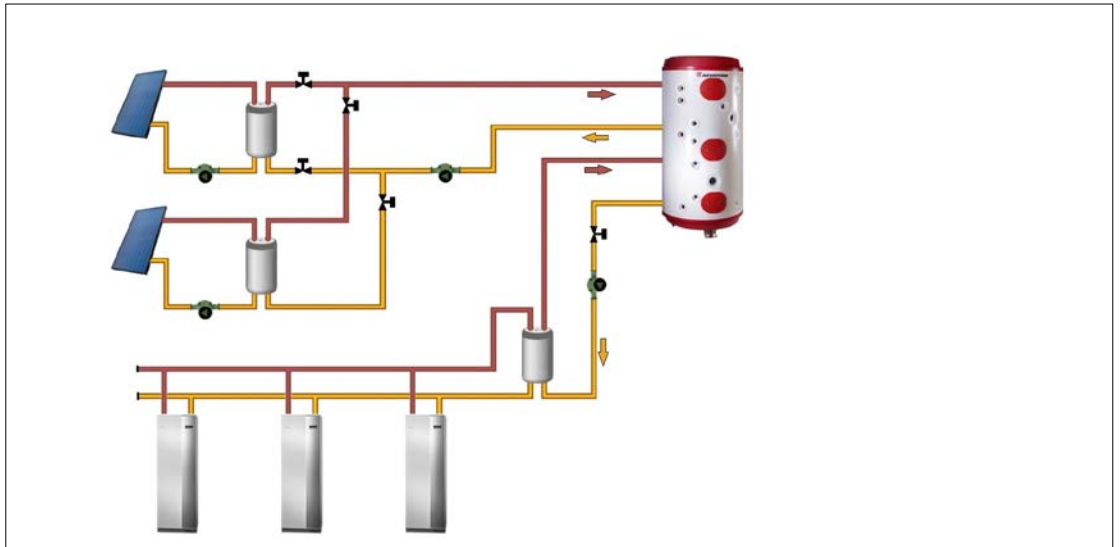


KUVIO 23. Käyttöliittymän testaus ja LabJackin pisteiden luku

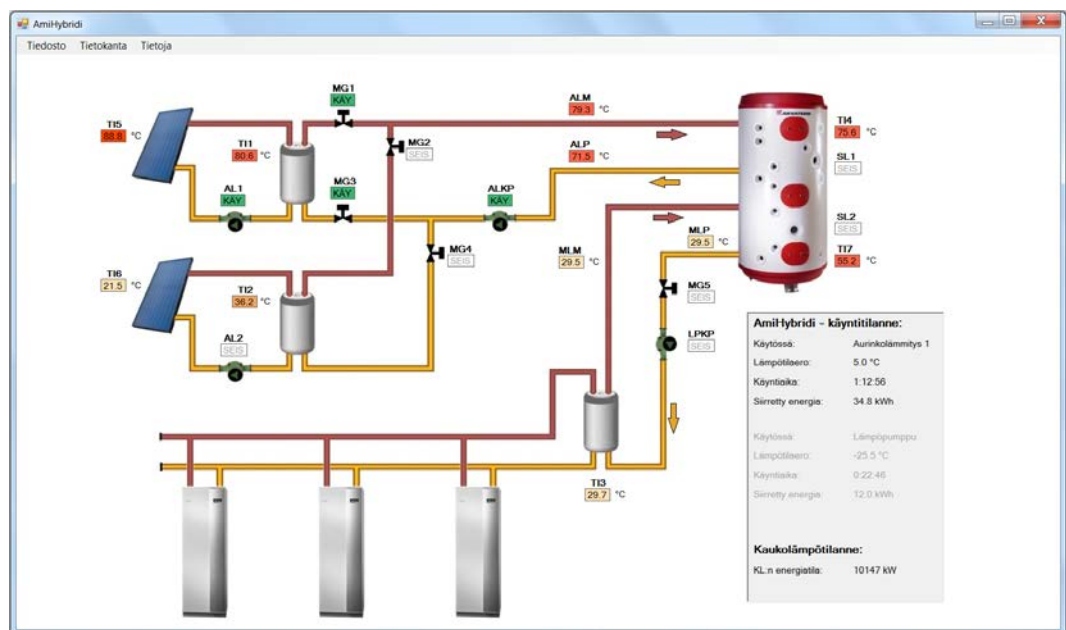
### 6.1 Käyttöliittymän pohjagrafiikka ja komponenttien ominaisuudet

Käyttöliittymän pohjaksi ladattiin aiemmin luodun AmiHybridi-toimintakaavion mukainen pohjakuva (kuvio 24). Pohjakuvaan valittiin esitettäväksi vain sellaiset komponentit, jotka ovat käyttöliittymäohjauksen kannalta olennaisia ja toiminnan ymmärtämisen kannalta välttämättömiä. Kiinteistöautomaatiojärjestelmien käyttöliittymät voidaan jakaa niissä esitettävien pohjakuvien perusteella kahteen eri toteutusmalliin. Prosessiautomaatioon perustuvissa käyttöliittymissä on käyttäjällä mahdollisuus lisätä ja poistaa näytössä esitettäviä komponentteja tai sen osia melko vapaasti. Sen sijaan kiinteällä pohjakuvalla toteutetuissa käyttöliittymissä, joudutaan muutostilanteessa käyttöliittymäohjelman ohjelmakoodi kirjoittamaan osittain uudelleen. AmiHybridi-käyttöliittymäohjelman tapauksessa kiinteä pohja-

kuva katsottiin helpommin toteutettavaksi ja riittäväksi, koska järjestelmän perustoiminta tulee pysymään jatkossakin toimintakaavion mukaisena. Pohjakuvan päälle sijoiteltiin pääkkunan tarvittavat komponentit, tekstikentät ja selostustekstit (kuvio 25). Komponentit sijoiteltiin alkuperäisen toimintakaaviosuunnitelman mukaisesti.



KUVIO 24. Toimintakaavion mukainen käyttöliittymän pohjakuva



KUVIO 25. Käyttöliittymän etusivu komponenttisijoittelun jälkeen

Lämpötilamittausten ja ohjaustilojen tekstilaatikot toteutettiin siten, että niiden värikoodi antavat suoraan informaatiota katsojalle komponentin tilasta lämpötilan

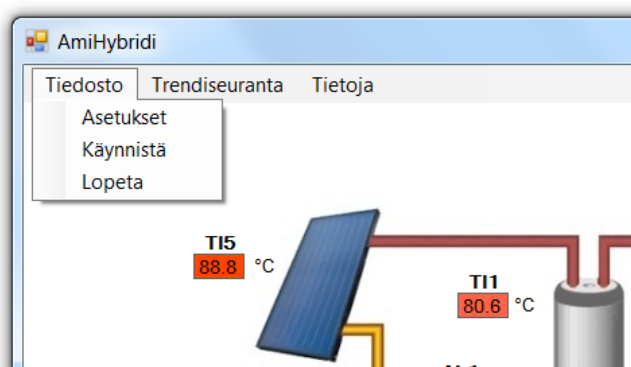
tai käyntitilan osoittamisen lisäksi. Lämpömittausten pohjaväriyty muuttuu lämpötilan mukaan viileästä keltaisesta kuumaan punaiseen. Pumppujen ja magneetti-venttiilien pohjaväriyty on käyntitilassa vihreä ja seistilan osoittaminen tehdään muuttamalla komponentin näkyvyys ”Enabled = False” tilaan. ”Värikoodaus” auttaa katsojaa hahmottamaan järjestelmän tilannetiedon nopeasti.

Päänäytössä on lisäksi AmiHybridi käyntitilanne-tietolaatikko, mistä selviää järjestelmän kokonaistilanne tarkasteluhetkellä. Tilatieto kertoo

- mikä lämmitysmuoto on käytössä
- mikä on siirrettävän menolämmön ja paluulämmön lämpötilaero
- kuinka kauan kumulatiivisesti siirto on ollut käytössä
- kuinka paljon kumulatiivisesti on siirretty lämpöenergiaa
- mikä on kaukolämpövaraajan lämpöenergiamäärä.

## 6.2 AmiHybridi-ohjelman käyttövalikko

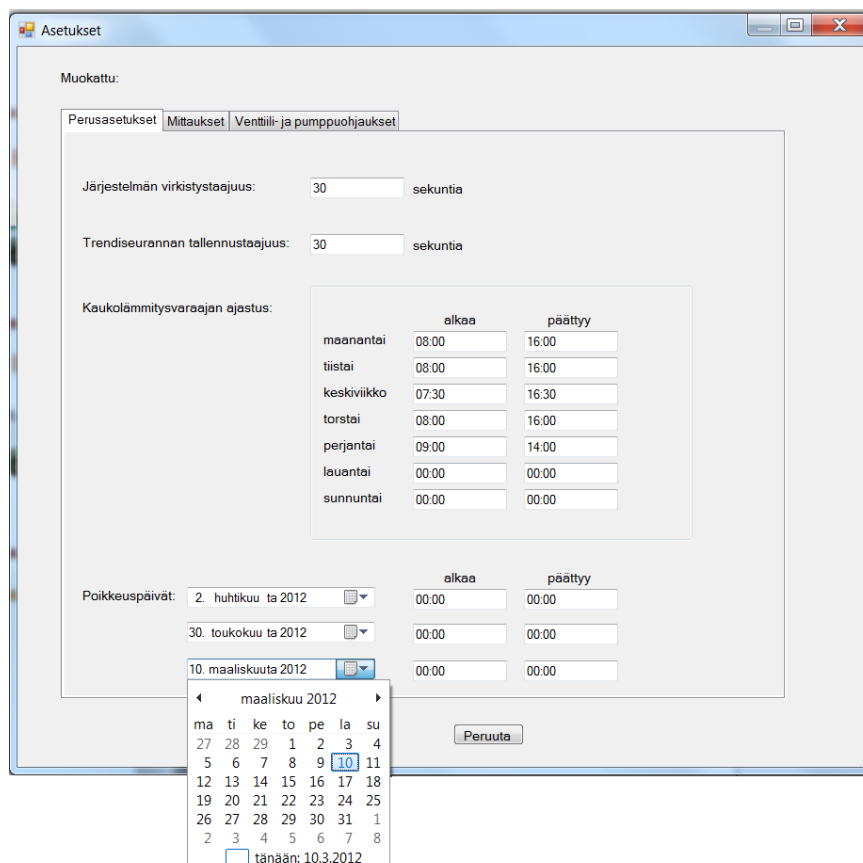
Järjestelmä on normaalisti käytössä jatkuvasti valvoen mahdollisten lämpöenergioiden siirtoja. Ohjelma käynnistetään kuvion 26 mukaisesti tiedostovalikosta käynnistä-painikkeella ja pysäytetään tarvittaessa lopeta-painikkeella.



KUVIO 26. Tiedostovalikon sisältö

Tiedostovalikon asetukset valinnalla saadaan esiin kuvion 27 mukainen kolmella alilehdellä varustettu ikkuna. Perusasetuksissa voidaan asettaa suunnitteluvaiheessa esitetyt virkistys- ja raja-arvot sekä kaukolämmityksen viikkoaikataulukko poikkeuspäivineen. Perusasetuksissa määritellään myös järjestelmän virkistystaa-

juus ja trenditiedostojen tallennustaajuus. Järjestelmän virkistystaajuus on se taajuus, millä aikavälillä mittaustietoja haetaan mittalaitteilta ja millä aikataululla mahdolliset ohjaustoimet tehdään. Yleisesti ottaen voi kiinteistöautomaatiojärjestelmissä virkistystaajuus olla useita kymmeniä sekunteja, mikäli järjestelmässä ei ole nopeasti muuttuvia osia.



KUVIO 27. Ohjelman perusasetussivu

Välilehdillä järjestelmän komponentit on valittavissa alavetovalikoista, mikä helpottaa käyttäjän valintaa ja varmistaa oikean nimisen komponentin valinnan. Samoin on perusasetusten välilehdellä käytetty Date Time Picker -komponenttia päivämäärien valinnassa.

Mittausvälilehdellä (kuviokuva 28) voidaan asettaa järjestelmän lämpötilamittauksille halutut hälytys- ja ohjausraajat sekä tarvittaessa se tavoitelämpötila, johon järjestelmän tulisi ohjauksella päästä. Mittausvalikolla tyhjä kenttä edustaa NULL-arvoa ja voidaan näin ohjelmassa havaita ja halutessa ohittaa.



	Hälytys alaraja:	Hälytys yläraja:	Ohjausalaraja:	Ohjausyläraja:	Tavoitearvo:
ALM		90		98	
ALP		90		98	
MLM		90		98	
MLP		90		98	
T11					
T12					
T13					
T14					
T15					
T16					
T17					
T18					
valitse					
valitse					

KUVIO 28. Mittausasetukset

Kolmannella välilehdellä (kuvio 29) annetaan käynnistys- ja sulkeutumisviiveet magneettiventtiileille ja pumppuille. Näiden viiveaikojen avulla voidaan estää järjestelmää sulkemasta magneettiventtiiliä, ennen kuin pumppu, joka on samassa piirissä, on pysähtynyt ja paine venttiilin kohdalla on alentunut normaaliksi.

	Käynnistysviive:	Sulkeutumisviive:	
AL1	5	5	sekuntia
AL2	5	5	sekuntia
ALKP	10	10	sekuntia
LPKP	7	8	sekuntia
MG1	5	5	sekuntia
MG2	5	5	sekuntia
MG3	5	5	sekuntia
MG4	5	5	sekuntia
MG5	5	5	sekuntia
valitse			sekuntia
AL1			sekuntia
AL2			sekuntia
ALKP			sekuntia
LPKP			sekuntia
MG1			sekuntia
MG2			sekuntia
MG3			sekuntia
MG4			sekuntia
MG5			sekuntia

KUVIO 29. Venttiili- ja pumppuohjaukset

### 6.3 Trendiseuranta

Trendiseuranta toteutettiin XML-tiedoston avulla. Seurantaan tallennetaan aina uusi tilannetieto (trendi) kaikista järjestelmän mittauksista ja ohjauksista perusasetuksissa käyttöliittymäohjelman asetuksissa valitun aikataajuuden mukaisesti. Tallennuksen kohdetiedosto ja tiedostonimi on valittavissa vapaasti, joten seuranta-tiedot voidaan halutessa tallentaa esimerkiksi oppilaitoksen levyasemalle, josta se on opiskelijoiden saatavissa. Uuden tiedoston valinnan jälkeen luodaan aluksi tyhjä XML-tiedosto ja XML-skeema, joka on kuvion 13 mukainen. XML-skeema sisältää tiedot käyttöliittymän mittaussarakkeiden nimistä ja tietotyypeistä.

```

1 <?xml version="1.0" encoding="utf-8"?><?xml version="1.0" standalone="yes"?>
2 <TRENDIDataSet xmlns="http://tempuri.org/TRENDIDataSet.xsd">
3   <xs:schema id="TRENDIDataSet" targetNamespace="http://tempuri.org/TRENDIDataSet.xsd" xmlns:msdtns
   "http://tempuri.org/TRENDIDataSet.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="u
   elementFormDefault="qualified">
4     <xs:element name="TRENDIDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
5       <xs:complexType>
6         <xs:choice minOccurs="0" maxOccurs="unbounded">
7           <xs:element name="Kaikki">
8             <xs:complexType>
9               <xs:sequence>
10                <xs:element name="PVM" type="xs:dateTime" minOccurs="0" />
11                <xs:element name="ALM" type="xs:string" minOccurs="0" />
12                <xs:element name="ALP" type="xs:string" minOccurs="0" />
13                <xs:element name="MLM" type="xs:string" minOccurs="0" />
14                <xs:element name="MLP" type="xs:string" minOccurs="0" />
15                <xs:element name="TI1" type="xs:string" minOccurs="0" />
16                <xs:element name="TI2" type="xs:string" minOccurs="0" />
17                <xs:element name="TI3" type="xs:string" minOccurs="0" />
18                <xs:element name="TI4" type="xs:string" minOccurs="0" />
19                <xs:element name="TI5" type="xs:string" minOccurs="0" />
20                <xs:element name="TI6" type="xs:string" minOccurs="0" />
21                <xs:element name="TI7" type="xs:string" minOccurs="0" />
22                <xs:element name="ALKP" type="xs:string" minOccurs="0" />
23                <xs:element name="MLKP" type="xs:string" minOccurs="0" />
24                <xs:element name="SL1" type="xs:string" minOccurs="0" />
25                <xs:element name="SL2" type="xs:string" minOccurs="0" />
26                <xs:element name="MG1" type="xs:string" minOccurs="0" />
27                <xs:element name="MG2" type="xs:string" minOccurs="0" />
28                <xs:element name="MG3" type="xs:string" minOccurs="0" />
29                <xs:element name="MG4" type="xs:string" minOccurs="0" />
30                <xs:element name="MG5" type="xs:string" minOccurs="0" />
31                <xs:element name="AL1" type="xs:string" minOccurs="0" />
32                <xs:element name="AL2" type="xs:string" minOccurs="0" />
33              </xs:sequence>
34            </xs:complexType>
35          </xs:element>
36        </xs:choice>
37      </xs:complexType>
38    </xs:element>
39  </xs:schema>
40 </TRENDIDataSet>
41

```

KUVIO 30. Trendiseurannan XML-tiedoston skeema

XML-skeeman mukaisesti tallennettua tietoa testausvaiheen lämpötilamittauksista ja toimilaitteiden käyntitiloista on esitettyä kuviossa 31. Tiedoista voidaan lukea minä päivänä ja millä ajanhetkellä arvot on mitattu. Esimerkissä on käytetty trendiseurannan taajuutena yhtä minuuttia. Lisäksi voidaan havaita, että trenditietoa kerätään kaikista mittaus- ja ohjauspisteistä. Tämä on mahdollista, koska tallen-

nettavaa tietoa on melko vähän. Suuremmissa LVI-automaatiojärjestelmissä, joissa mittaus- ja laitepisteitä on useita satoja, voi olla myös trendiseurannan tietojen keräyspisteitä syytä rajoittaa.

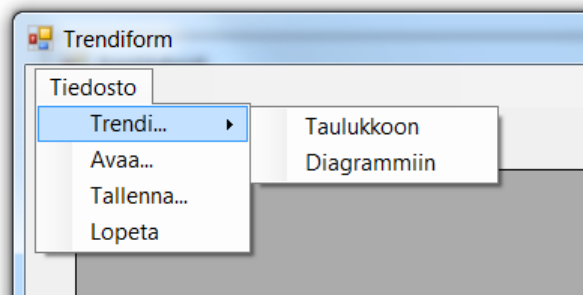
```

40 <Kaikki>
41 <PVM>2012-12-11T08:12:15+02:00</PVM>
42 <ALM>88.8</ALM>
43 <ALP>82.5</ALP>
44 <MLM>21.0</MLM>
45 <MLP>21.0</MLP>
46 <TI1>89.2</TI1>
47 <TI2>21.0</TI2>
48 <TI3>33.5</TI3>
49 <TI4>77.4</TI4>
50 <TI5>90.2</TI5>
51 <TI6>21.0</TI6>
52 <TI7>44.2</TI7>
53 <ALKP>KÄY</ALKP>
54 <MLKP>SEIS</MLKP>
55 <SL1>SEIS</SL1>
56 <SL2>SEIS</SL2>
57 <MG1>KÄY</MG1>
58 <MG2>SEIS</MG2>
59 <MG3>KÄY</MG3>
60 <MG4>SEIS</MG4>
61 <MG5>SEIS</MG5>
62 <AL1>KÄY</AL1>
63 <AL2>SEIS</AL2>
64 </Kaikki>
65 <Kaikki>
66 <PVM>2012-12-11T08:13:00+02:00</PVM>
67 <ALM>87.8</ALM>
68 <ALP>81.6</ALP>
69 <MLM>21.0</MLM>
70 <MLP>21.0</MLP>
71 <TI1>88.8</TI1>
72 <TI2>21.0</TI2>
73 <TI3>33.3</TI3>
74 <TI4>78.6</TI4>
75 <TI5>50.5</TI5>
76 <TI6>21.0</TI6>
77 <TI7>45.4</TI7>
78 <ALKP>KÄY</ALKP>
79 <MLKP>SEIS</MLKP>
80 <SL1>SEIS</SL1>
81 <SL2>SEIS</SL2>
82 <MG1>KÄY</MG1>
83 <MG2>SEIS</MG2>
84 <MG3>KÄY</MG3>
85 <MG4>SEIS</MG4>
86 <MG5>SEIS</MG5>

```

KUVIO 31. Trendiseurannan keräämiä mittaustietoja XML-tiedostomuodossa

Trendiseurantatiedot voidaan käyttöliittymässä hakea mistä tahansa aiemmin tallennetusta trenditiedostosta ohjelmaan ja tulostaa näytölle taulukkomuodossa tai viivadiagrammina. Trendiseuranta voidaan avata kuvion 32 mukaisesti tiedostovalikosta.



KUVIO 32. Trendiseurantavalikko

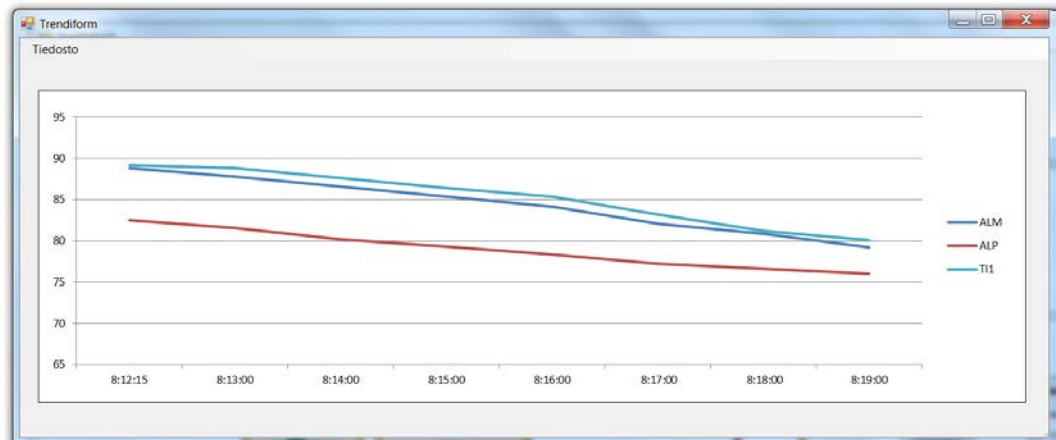
Avattavaksi valitun trenditiedoston taulukko- tai diagramminäytössä voidaan valita, esitetäänkö kaikki muuttuvat arvot samanaikaisesti vai rajataanko valinta vain tiettyihin arvoihin. Kuviossa 33 on listaus ohjelman testausvaiheen trenditaulukosta, jossa on tulostettuna kaikki mittauspisteet ja niiden lämpötilat tietyllä ajanhetkellä. Taulukkomuotoinen trendiseuranta mahdollistaa hyvin tarkan analyysin käytössä olevan järjestelmän tilasta.

PVM	ALM	ALP	MLM	MLP	T11	T12	T13	T14	T15	T16	T17
11.12.2012 ...	88.8	82.5	21.0	21.0	89.2	21.0	33.5	77.4	90.2	21.0	44.2
11.12.2012 ...	87.8	81.6	21.0	21.0	88.8	21.0	33.3	78.6	50.5	21.0	45.4
11.12.2012 ...	86.6	80.2	21.0	21.0	87.6	21.0	33.3	78.6	40.2	21.0	45.4
11.12.2012 ...	85.4	79.3	21.0	21.0	86.4	21.0	33.3	78.6	30.5	21.0	45.4
11.12.2012 ...	84.2	78.4	21.0	21.0	85.4	21.0	33.3	78.4	22.2	21.0	45.4
11.12.2012 ...	82.1	77.2	21.0	21.0	83.2	21.0	33.3	78.6	21.0	21.0	45.4
11.12.2012 ...	80.9	76.6	21.0	21.0	81.2	21.0	33.3	78.5	21.0	21.0	45.4
11.12.2012 ...	79.2	76.0	21.0	21.0	80.1	21.0	33.3	78.6	21.0	21.0	45.4

KUVIO 33. Trenditiedosto taulukkomuodossa

Kuviossa 34 on trendiseuranta tulostettuna muutaman valitun mittauspisteen mukaisena trendikäyrästä. Käyrästä on hyvin nopeasti luettavissa eri lämpötilamittausten muuttuminen ajan kuluessa. Kuvasta näkyy taulukkoa selkeämmin, miten aurinkolämmityksen meno- (ALM) ja paluuedet (ALP), samoin kuin läm-

pötila-anturin TI1 mittaama aurinkolämpövaraajan vesi lauhtuvat vähitellen siirron aikana.

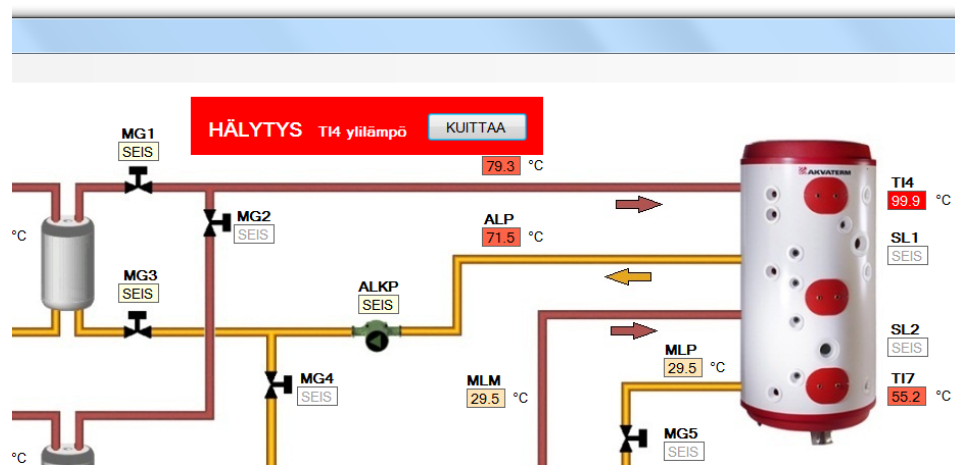


KUVIO 34. Trenditiedosto diagrammina

Valinnan rajaaminen vain tiettyihin tietoihin kerrallaan selkeyttää varsinkin diagrammi-näytössä esitettävää kuvaa ja helpottaa eri mittaus- ja ohjaustietojen havainnointia verrattuna toisiinsa.

#### 6.4 Hälytykset

Hälytysehdon toteutuessa käyttöliittymäohjelma sulkee kaikki lämmitysveden siirrot ja antaa hälytyksestä tiedon käyttöliittymän etusivulle (kuvio 35).



KUVIO 35. Hälytystieto ja kuitauspainike

Hälytystiedossa on määritelty se piste, mistä hälytys on aiheutunut ja onko kyseessä yli- vai alilämpötilanne. Samassa hälytystietopaneelissa on hälytyksen kuitauspainike, josta hälytystieto saadaan kuitattuun tilaan. Hälytyksen kuittaaminen ei automaattisesti käynnistä lämmityksen siirtoa uudelleen, vaan se on tehtävä erikseen tiedostovalikon käynnistä valinnalla.

Kuvion 35 hälytystoiminto on aiheutunut kaukolämmitysvaraajan liian korkeasta lämpötilasta (TI4). Käyttöliittymän hälytys on aikaansaanut kiertovesipumpun pysähtymisen ja magneettiventtiilien sulkeutumisen vahinkojen estämiseksi.

## 7 TESTAUS

AmiHybridi-käyttöliittymäohjelman testaus suoritettiin ohjelmallisena tapaustestauksena. Ohjelmallisena testaus jouduttiin suorittamaan siitä syystä, ettei varsinainen tiloissa tapahtuva fyysinen laitteistojen ja putkistojen asennus ollut valmiina käyttöliittymäohjelmaa tehdessä. Laitteistojen fyysinen rakentaminen suoritetaan tämän opinnäytetyön pohjalta myöhemmin oppilastyönä.

Ohjelmallisen testauksen suorittamiseksi laadittiin pienimuotoinen aliohjelma, joka kasvatti tai vähensi lähtöarvona annettua mittaustietoa käyttöliittymän virkistystaajuuden ajastamana. Ohjelmallisessa testauksessa päästään harvoin todellisen tilanteen mukaisiin monimuotoisiin mittaustuloksiin, joten testauksen pääpainona olikin havainnoida käyttöliittymäohjelman toimintaa ja muutoksia yksittäisten mittausmuutosten perusteella.

Testauksen mukaan käyttöliittymä toimii sille asetettujen toimintamallien mukaisesti ja toteuttaa suunnitteluvaiheessa sille asetetut vaatimukset. Lopullinen testaus ja käyttöliittymäohjelman viritys suoritetaan fyysisen järjestelmän valmistuttua.

## 8 YHTEENVETO JA JOHTOPÄÄTÖKSET

Työn tavoitteena oli suunnitella ja toteuttaa hybridilämmityksen käyttöliittymä. Käyttöliittymäohjelman tuli mahdollistaa lämpöä tuottavien lämmitysjärjestelmäkoulutuksien aikana tuotetun lämpöenergian hyödyntäminen lämmitysjärjestelmäkoulutuksissa. Lisätavoitteena oli energian säästäminen ja opetustoiminnan kehittäminen.

Tavoite saavutettiin suunnittelun ja käyttöliittymän toteutuksen osalta. Varsinainen hybridijärjestelmän fyysinen toteutus tullaan tekemään tämän työn pohjalta, joten lopputestaus aidossa käyttöympäristössä jouduttiin jättämään myöhempään ajankohtaan.

Työn edetessä hahmottui mittaus- ja ohjauslaitteistona käytetyn LabJack U3:n käytännöllisyys fyysisen alustan ja ohjelman rajapintana. Laitteisto on helposti käyttöönotettavissa ja laajennettavissa tarpeiden mukaan. Laitteen ohjelmakirjasto mahdollistaa laitteiston monipuolisen ohjelmoinnin.

Useiden vuosien kokemukseni kiinteistöautomaatiojärjestelmien käyttämisestä ja opettamisesta oli suureksi avuksi työn aikana. Varsinkin hybridijärjestelmän suunnitteluvaiheessa tuli selkeästi esiin se, että vaihtoehtoisia toteutusmalleja on olemassa runsaasti. Valinta näiden vaihtoehtojen välillä on usein hyvin hankalaa ja tehdyn valinnan vaikutukset ulottuvat pitkälle järjestelmän muihin osiin. Suunnitteluvaiheessa tuleekin asettaa painopisteeksi selkeä toiminnallisuus ja toimintojen mahdollisimman yksinkertainen toteutus. Tämän työn osalta työtä helpotti se, että käyttöliittymä oli suunniteltu käytettäväksi vain tämän projektin osana. Käyttöliittymäohjelman kehitystä jatketaan tulevaisuudessa, lisäämällä siihen mahdollisesti käyttöön tulevia hake- ja pellettilämmittimiä tai tuuli- ja aurinkosähköenergiaa tuottavia laitteistoja. Samoin kehittämiskohteeksi tulee etävalvonnan mahdollistaminen Internetin yli Web-sovelluksilla sekä tablet- ja matkapuhelinsovelluksilla. Ensisijaisesti käyttöliittymäohjelmaa tullaan kuitenkin muokkaamaan siitä saatavan käyttäjäpalautteen pohjalta.



## LÄHTEET

Hautala M., Peltonen H. & Lahden Teho-Opetus Oy. 2002. Insinöörin (AMK) Fysiikkaa. Jyväskylä: Gummerus

LabJack Co. 2012. U3 User's Guide. [viitattu 1.3.2012]. Saatavissa:  
<http://labjack.com/support/u3/users-guide>

Moghadampour, G. 2011. C# Windows- ja tietokantaohjelmointi. Jyväskylä: WSOYpro Oy

ST 17 2001. Rakennusautomaatiojärjestelmät ST-käsikirja 17. Sähköinfo Oy. Espoo.

ST 22 2008. Kiinteistöjen valvomojärjestelmät ST-käsikirja 22. Sähköinfo Oy. Espoo.

Wille, C. 2001. C# Toolkit. Jyväskylä: Dosendo Finland

## LIITTE 1/1. LJUDDotNet Driver kirjaston esittely

LJUDDotNet Driver Usage, August, 1st 2007

This file will give more information on the specifics of the LJUDDotNet.dll file. The driver implements a namespace called LabJack.LabJackUD under which everything exists. The main object inside that namespace is the LJUD object. That object basically copies the functionality of the LabJackUD.dll driver. It contains the following enumerated types to act as constants:

LJUDERROR  
DEVICE  
CONNECTION  
IO  
CHANNEL  
RANGES  
TIMERMODE  
TIMERCLOCKS  
STREAMWAITMODES

Each of these work similarly to that which is found in the header file for the LabJackUD.dll file. Where you would use LJ\_dtUE9 for the LabJackUD.dll, you would now use LabJackUD.DEVICE.UE9. LJ\_ioANALOG\_INPUT would be LabJackUD.IO.ANALOG\_INPUT and so on.

The function then provides static wrappers to the LabJackUD.dll function with the following definitions.

- public static LJUDERROR OpenLabJack(DEVICE deviceType, CONNECTION connectionType, String address, bool firstFound, ref int handle)
- public static LJUDERROR OpenLabJackS(String deviceType, String connectionType, String address, bool firstFound, ref int handle)
- public static LJUDERROR AddRequest(int handle, IO IOType, CHANNEL channel, double val, int x1, double userData)

## LIITTE 1/2.

- public static LJUDERROR AddRequestS(int handle, String IOType, CHANNEL channel, double val, int x1, double userData)
- public static LJUDERROR AddRequestSS(int handle, String IOType, String channel, double val, int x1, double userData)
- public static LJUDERROR AddRequest(int handle, IO IOType, int channel, double val, int x1, double userData)
- public static LJUDERROR AddRequestS(int handle, String IOType, int channel, double val, int x1, double userData)
- public static LJUDERROR AddRequest(int handle , IO IOType , CHANNEL channel , double val , byte[] x1 , double userData)
- public static LJUDERROR AddRequest(int handle , IO IOType , CHANNEL channel , double val , double[] x1 , double userData)
- public static LJUDERROR Go()
- public static LJUDERROR GoOne(int handle)
- public static LJUDERROR eGet(int handle, IO IOType, CHANNEL channel, ref double val, double[] x1)
- public static LJUDERROR eGet(int handle, IO IOType, CHANNEL channel, ref double val, int x1)
- public static LJUDERROR eGetS(int handle,String IOType, CHANNEL channel, ref double val, int x1)
- public static LJUDERROR eGetSS(int handle,String IOType, String channel, ref double val, int x1)
- public static LJUDERROR ePut(int handle, IO IOType, CHANNEL channel, double val, int x1)
- public static LJUDERROR ePutS(int handle, String IOType, CHANNEL channel, double val, int x1)
- public static LJUDERROR ePutSS(int handle, String IOType, String channel, double val, int x1)
- public static LJUDERROR GetResult(int handle, IO IOType, CHANNEL channel, ref double val)

LIITTE 1/3.

- `.public static LJUDERROR GetResults(int handle, String IOType, CHANNEL channel, ref double val)`
- `public static LJUDERROR GetResultSS(int handle, String IOType, String channel, ref double val)`
- `public static LJUDERROR GetFirstResult(int handle, ref IO IOType, ref CHANNEL channel, ref double val, ref int px1, ref double userData)`
- `public static LJUDERROR GetNextResult(int handle, ref IO IOType, ref CHANNEL channel, ref double val, ref int px1, ref double userData)`
- `public static LJUDERROR ResetLabJack(int handle)`
- `public static LJUDERROR Close()`
- `public static LJUDERROR DoubleToStringAddress(double number, char[] str, int hexDot)`
- `public static LJUDERROR StringToDoubleAddress(String str, ref double number, int hexDot)`
- `public static int StringToConstant(String str)`
- `public static void ErrorToString(ERROR errorCode, char[] str)`
- `public static double GetDriverVersion()`
- `public static LJUDERROR ListAll(DEVICE deviceType, CONNECTION connectionType, ref int numFound, int[] serialNumbers, int[] IDs, double[] addresses)`
- `public static LJUDERROR ListAllS(String deviceType, String connectionType, ref int numFound, int[] serialNumbers, int[]`

The parameters to these functions all work the same as outlined in the LabJackUD users guide. Some functions have more than one definition to provide support for more than one calling type. This is needed in .NET because of the strict type system, but not in C++. One example is

- `public static ERROR eGet(int handle, IO IOType, CHANNEL channel, ref double val, double[] x1)`
- `public static ERROR eGet(int handle, IO IOType, CHANNEL channel, ref double val, int x1)`

#### LIITTE 1/4.

The first eGet function is used for returning an array of doubles, the second is for returning a single value. If you pass the data type you are expecting, the compiler should be able to identify which function you are calling and handle it appropriately.

There is also an object in the LabJack.LabJackUD namespace called LabJackUDEException. This object is an Exception object which provides an easier way of doing error handling. The object has a public LJUD.LJUDERROR error field which can be used to retrieve the error, and a ToString() function that calls LJUD.ErrorToString(error, s) to resolve the value. Thus, you can do the following to detect errors:

```
try
{
// Add requests
LJUD.AddRequest(ue9.ljhandle, UE9.IO.ANALOG_INPUT,(UE9.CHANNEL)
0,0,0,0);

// Execute
LJUD.GoOne(ue9.ljhandle);

// Get result
LJUD.GetResult(ue9.ljhandle, UE9.IO.ANALOG_INPUT, (UE9.CHANNEL) 0,
ref AI0);

}
catch (LabJackUDEException e)
{
// If error occurred print a message box showing which
MessageBox.Show(this, e.ToString(), "LabjackUD Error #" + (int)e.error);
}
```

LITTE 1/5.

There is also an UE9 object created in the LabJack.LabJackUD namespace that provides a few extra features to make working with the UE9 easier. The UE9 object inherits from the LJUD object, thus it contains the same static functions you can call. It also supports the following functions:

```
controlConfig()  
commConfig()  
locID()  
GetHashCode()  
Equals(Object o)
```

These functions allow you to use the object as a representation as a UE9. Thus you can use it as an Object in a collection data type or for other purposes.

The UE9 object has the following public fields defined:

```
public double localid;  
public double powlev;  
public double ip;  
public double gateway;
```

(LabJack Co.)