

Jaakko Heinonen

Puutavarakuivaamon ohjaus Arduino-kehitysalustalla

Automaatiotekniikan koulutusohjelma

2015

Puutavarakuivaamon ohjaus Arduino-kehitysalustalla

Heinonen, Jaakko
Satakunnan ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma
Toukokuu 2015
Ohjaaja: Asmala, Hannu
Sivumäärä: 49
Liitteitä: 9

Asiasanat: käyttöliittymä, mikrokontrolleri, anturi

Tässä opinnäytetyössä tehtiin uusi ohjaus vanhalle puutavarakuivaamolle. Ohjaus uusittiin, koska laitteisto oli mekaanisesti hyvässä kunnossa, mutta ohjaus tarvitsi päivityksen. Työssä testattiin, miten mikrokontrollerilla toteutettu ohjaus ja selainpohjainen käyttöliittymä soveltuvat teollisuusympäristöön. Kehitysalustaksi valittiin Arduino Yún, koska se täytti kaikki ne kriteerit mitä tämä projekti vaatii. Arduino on liitetty tehtaan langattomaan verkkoon. Selainpohjainen käyttöliittymän mahdollistaa kuivaamon ohjauksen tietokoneella tai mobiililaitteella.

Lopputuloksena saatiin toimiva ohjaus ja etäkäyttöliittymä kuivaamolle. Käyttöliittymästä voi seurata kuivaamon mittauksen oloarvoja ja I/O-tietoja. Salanasuojatulta linkkisivulta voi säätää ohjauksen asetusarvoja.

Arduino controlled timber drying system

Heinonen, Jaakko

Satakunta University of Applied Sciences

Degree Programme in Automation

May 2015

Supervisor: Asmala, Hannu

Number of pages: 49

Appendices: 9

Keywords: interface, microcontroller, sensor

In this thesis a new control of timber drying system was designed and implemented. Control was renewed because of the system is mechanically still in good shape, but the control needed to upgrade. One of the goals was to test, how the microcontroller and the browser-based user interface are suitable for industrial environments. Arduino Yun was chosen because it met all the criteria of what this project requires. Arduino is connected to the factory wireless network. Browser-based interface allows operation via a PC or mobile device.

As a result complete control of the timber drying system was implemented. Remote user interface can monitor the actual value of the measurements and the I/O data. Set values are password protected.

SISÄLLYS

1	JOHDANTO	6
2	PUUTAVARANKUIVAUS	7
2.1	Lauhde- eli kondenssikuivaus	7
2.2	Kuivain	8
2.3	Kylmäkoneen toimintaperiaate	11
2.4	Kompressorit	12
2.5	Lauhduttimet	12
2.6	Kylmäkoneen ohjaus	13
3	MIKROKONTROLLERIT	13
3.1	Historia	13
3.2	Mikrokontrollerin osat	14
3.2.1	Proessori	14
3.2.2	Käyttömuisti (SRAM)	14
3.2.3	Ohjelmamuisti (flash)	14
3.2.4	EEPROM-muisti	14
3.2.5	Tulo- ja lähtöpinnit	15
3.2.6	AD-muunnin	15
3.2.7	PWM-lähtö	15
3.2.8	Watchdog	15
3.2.9	Ajastin	15
4	ARDUINO	17
4.1	Yleistä	17
4.2	Arduino Yún	17
4.3	Liitynnät	18
4.4	Arduino Yún tekniset tiedot	19
4.5	WLAN-verkon konfigurointi	19
4.6	Yún muistin laajennus	21
5	ANTURIT	22
5.1	Lämpötila-anturit	22
5.1.1	DS18B20-lämpötila-anturi	23
5.1.2	1-wire väylä	23
5.2	Kosteusanturi	25
5.2.1	Resistiivinen kosteusanturi	25
5.2.2	Kapasitiivinen kosteus anturi	25
5.2.3	Kuivaamokaapin kosteusanturi	27
5.3	Virran mittaus	28

5.3.1 Hall-ilmio.....	28
5.3.2 Sähkömagneettinen induktio	29
5.3.3 Kompressorin virranmittaus	30
6 OHJAUSKESKUS	31
6.1 Piirikaavio.....	31
6.2 Komponenttien valinta	32
6.2.1 Pääkytkin	32
6.2.2 Moottorinsuojakytkimet	32
6.2.3 Releet.....	33
6.2.4 Kontaktorit.....	34
6.2.5 Johdotus.....	34
7 OHJELMAT	36
7.1 Yleistä.....	36
7.2 Autobahn	37
7.3 WebSocket.....	37
7.4 Kontrollerin (ATmega32u4) ohjelma.....	39
7.5 Prosessorin(Atheros AR9331) ohjelma.....	42
7.6 Käyttöliittymän suunnittelu	44
7.6.1 HTML5.....	44
7.6.2 Uudet ominaisuudet	44
7.6.3 Selaimenohjelma.....	45
8 LIITYNTÄ TEHTAAN VERKKOON	47
9 YHTEENVETO	49
LÄHTEET	50
LIITTEET	

1 JOHDANTO

Tässä opinnäytetyössä tehdään uusi ohjaus puutavarakuivaamolle. Tammiston Puu Oy:llä vanha puutavarakuivaamon ohjaus tarvitsi päivityksen. Kuivaamo on tyypiltään lauhde- eli kondenssikuivain. Kuivaamon ohjauksessa on tarkoitus testata, miten mikrokontrolleri soveltuu teollisuussovelluksen ohjaukseen. Aluksi on tarkoitus ottaa selvää millaisia mikrokontrollereita on markkinoilla ja millainen alusta on tähän projektiin sopivin. Käyttöliittymä toteutetaan internetselaimella. Käyttöliittymä on tarkoitus tehdä niin, että asiakaspäässä ei tarvita erillisiä ohjelmia käyttöliittymän pyörittämiseen. Tässä raportissa käydään läpi kuivaamon ohjauksen suunnittelu ja työvaiheet siinä järjestyksessä, jossa ne etenivät. Raportissa käydään läpi kuivaamon ja eri komponenttien toimintaperiaatteet ja miten ne liittyvät tähän laitteistoon.

2 PUUTAVARANKUIVAUS

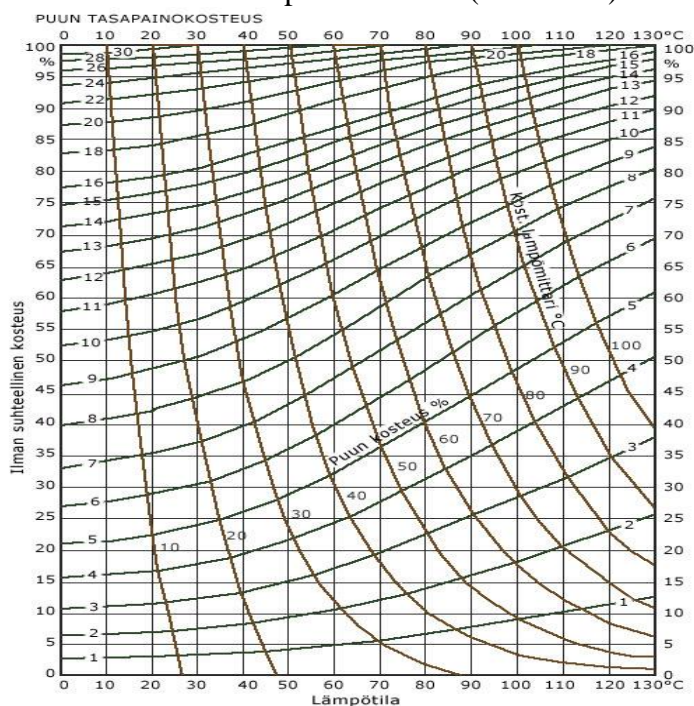
2.1 Lauhde- eli kondenssikuivaus

Lauhdekuivaamon toimintaperiaate on hyvin samankaltainen kuin kamarikuivaamon, mutta kosteuden poisto kuivaamosta tehdään tiivistämällä vesihöyry vedeksi, kun kamarikuivaamossa se poistetaan ilmanvaihdolla. Lauhdekuivaamon energiakustannukset ovat alhaiset, koska puussa olevan veden höyrystämiseen kuluva energia saadaan talteen, kun lämmin vesihöyry jäähtyy ja tiivistyy höyrystimen kylmässä pinnassa vedeksi. Lauhdekuivaamon lauhduttimeen pumpattu lämpöenergia siirretään puhaltimilla takaisin kuivaamoon.

Lauhdekuivaamon rakennuskustannukset ovat edullisemmat muihin kuivaamotyypeihin verrattuna, koska ei tarvita välttämättä erillistä lämpökeskusta. Lauhdekuivauksen etuja on: kuivauslaatu, puutavaran väri ei muutu ja edulliset käyttö- ja investointikustannukset. Huonot puolet: pitkät kuivausajat, vaatii paljon kuivaustilaa ja sinistymis- ja homehtumisvaara, joka johtuu alhaisesta kuivauslämpötilasta.

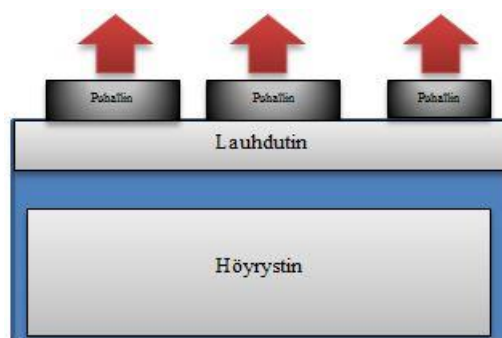
(PuuProffa)

Taulukko 1. Puun tasapainokosteus. (PuuProffa)

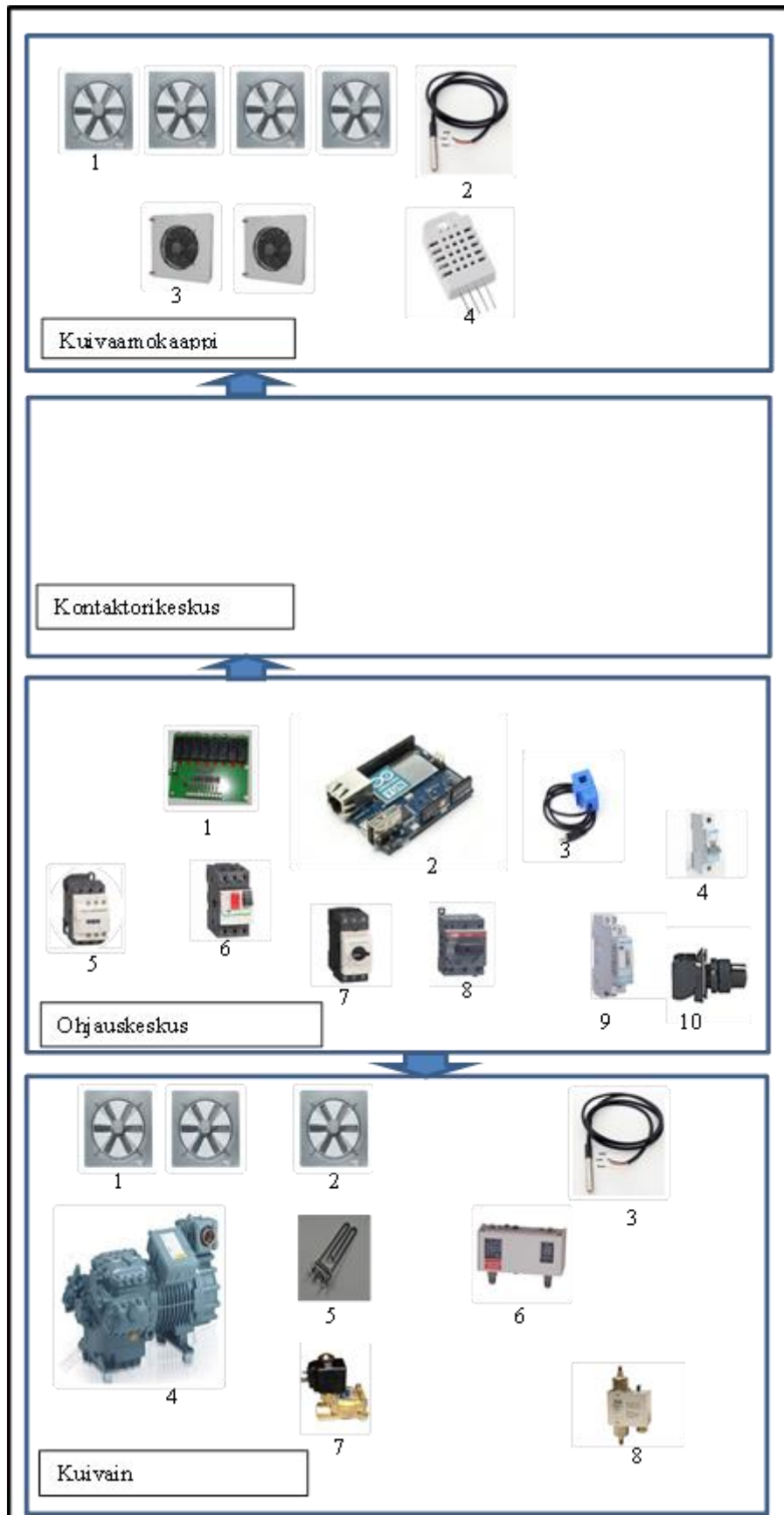


2.2 Kuivain

Tässä kuivaimessa on Copelandin puolihhermeettinen mäntäkompressori. Kuivaimen pääperiaate: Lauhdutin on ilmajäähdytteinen, jossa on kaksi puhallinta, jotka kierrättävät kuivaamokaapin ilmaa höyrystimen läpi lauhduttimelle. Kuivaamon lauhdutinpuhaltimet ohjaavat kuivaamon lämpimän ilman höyrystimen läpi, jossa ilma jäähtyy ja kosteus tiivistyy vedeksi. Ylikuumenemista varten on lisäpuhallin joka kierrättää tarvittaessa ulkoilmaa lauhduttimen läpi.



Kuva 1. Lauhdekuivain



Kuva 2. Kuivaamon pääkomponenttien sijainti

Kuivaamon pääkomponentit.

Kuivaamokaappi:

1. Kiertoilmapuhallin
2. Lämpöanturi(DS18B20)
3. Lämpöpuhallin
4. Kosteusanturi(AM2302)

Ohjauskeskus:

1. Relekortti(8pcrelays)
2. Arduino Yún
3. Virtamuunnin(SCT-013-000)
4. Johdonsuoja(NRN104)
5. Kontaktori(LC1D09P7)
6. Moottorisuoja(GV2ME)
7. Moottorisuoja(GV3P)
8. Pääkytkin(OT80F3)
9. Rele(ERC216)
10. Kiertopainike(XB5AD25)

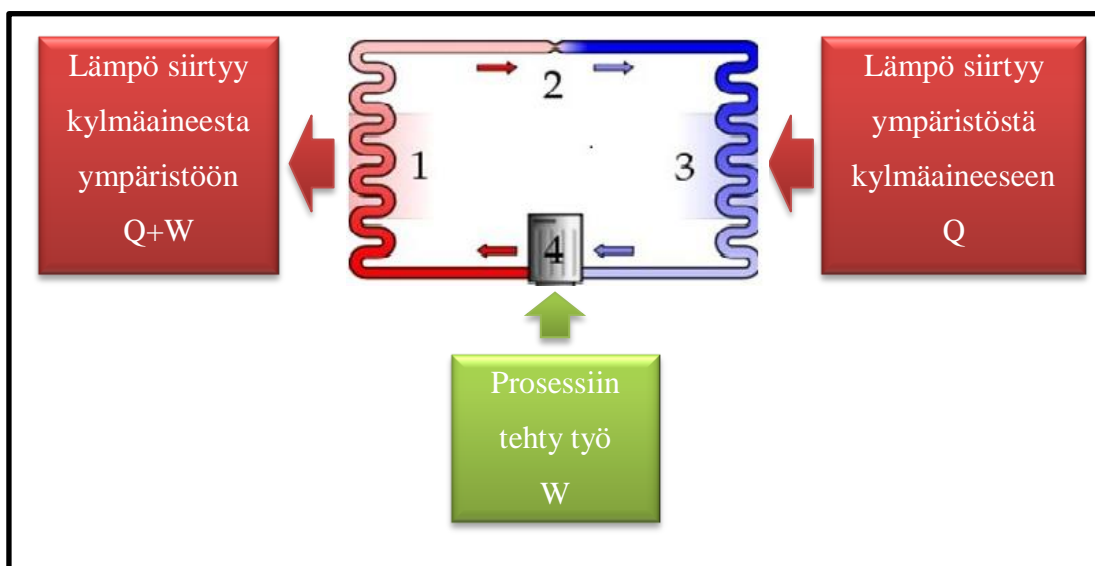
Kuivain:

1. Lauhdepuhallin
2. Jäähdytyspuhallin
3. Lämpöanturi(DS18B20)
4. Kompessorin (Copeland D4RH1-2000-EWM)
5. Kompessorin lämpövastus
6. Kaksoispressostaatti(KP15)
7. Magneettiventtiili(BF230AS)
8. Öljyn eropainekytkin (MP 54)

2.3 Kylmäkoneen toimintaperiaate

Kylmäteknikkaan liittyy kaksi termodynamiikan pääsääntöä oleellisesti. Termodynamiikan 1. pääsäännön mukaan energia ei voi kadota se voi vain muuttua muotoaan. Termodynamiikan 2. pääsäännön mukaan lämpö siirtyy aina luonnostaan lämpimästä kylmään. Jos halutaan lämmön siirtyvän matalammasta korkeampaan lämpötilaan, on jonkin tehtävä aina työ. Kylmälaitteessa kompressorin sähkömoottori eli sähköenergia tekee työn.

Kompressori muuttaa sähköenergian lämpöenergiaksi eli energia muuttuu muotoaan. Kompressori pumppaa höyrystyneen kylmäaineen lauhduttimeen, jolloin se nesteytyy ja kuumenee. Lauhdutinpuhaltimet siirtävät lämmön ympäristöön. Lauhduttimesta kylmäaine siirtyy höyrystimelle. Paisuntaventtiilin jälkeen kylmäaine purkautuu höyrystimeen ja jäähtyy, näin lämpö siirtyy ympäristöstä kylmäaineeseen. (Kaappola, 2011)



Kuva3. Kylmäkoneen toimintaperiaate

Kylmäkoneen pääkomponentit ovat.

1. Lauhdutin
2. Paisuntaventtiili
3. Höyrystin
4. Kompressori

2.4 Kompressorit

Kompressorit jaetaan sen rakenteen perusteella hermeettisiin, puolihermeettisiin ja avokompressoreihin. Hermeettiset kompressorin sähkömoottori ja kompressorin on rakennettu suljettuun hitsattuun metallikuoreen. Puolihermeettisen kompressorin sähkömoottori ja kompressorin ovat myös saman kuoren sisällä, mutta se on avattavissa. Avokompressorissa itse kompressorin on erillään sähkömoottorista ja on kytketty kompressorin hihnalla tai akselikytkimellä. Kompressoreita voidaan jakaa myös niiden pumppaustekniikan perusteella mäntäkompressoreihin, ruuvikompressoreihin ja scroll- eli kierukkakompressoreihin. (Kaappola, 2011)

2.5 Lauhduttimet

Lauhduttimet voidaan jakaa karkeasti kahteen eri tyyppiin ilmajäähdytteisiin ja nestejäähdytteisiin. Ilmajäähdytteiset lauhduttimet on valmistettu yleensä kupariputkista, jotka kiertävät alumiinilamellien sisällä. Lauhdutustehoa säädetään puhaltimien määrällä tai säädetään puhaltimien pyörintänopeutta esim. taajuusmuuttajalla. Ilmajäähdytteisen lauhduttimen hyviä puolia ovat edullisuus ja helppo huollettavuus. Huono puoli on sen suuri koko. Nestejäähdytteisiä lauhduttimia on rakenteeltaan moniputki-lauhduksia, levylämmönsiirtimiä ja koaksiaalilauhduksia. Nestejäähdytteiset lauhduttimen yleinen toimintaperiaate on, että jäähdytysneste ja kylmäaine siirretään lämmönvaihtimeen, jossa ne menevät vierekkäisissä lamelleissa, ja näin lämpö siirtyy jäähdytysnesteeseen. Jäähdytysneste voidaan jäähdyttää erillisellä ilmajäähdytteisellä kennolla tai siirtää lämmitysjärjestelmään. Nestejäähdytteisen lauhduttimen hyviä puolia ovat jäähdytysteho ja lämmön siirrettävyys kauemmas kylmäkoneesta. (Kaappola, 2011)

2.6 Kylmäkoneen ohjaus

Lauhdekuivaamossa kompressoria ohjataan kuivaamon kosteusanturin (AM2302) ohjaukselle antaman tiedon perusteella. Kompressoria ohjataan niin sanotulla Pum Down -kytkennällä, kun ohjaus saa pysäytystiedon kosteusanturilta ohjaus sulkee magneettiventtiilin. Kompressori käy niin kauan, että imupuolen paine laskee matalapainekeytkimelle asetetun arvon alapuolelle, jolloin se antaa ohjaukselle pysäytystiedon ja pysäyttää kompressorin. Lauhdepuhaltimet käyvät koko kuivausajan. Lauhduttimen jäähdytinpuhallin käynnistyy, jos lauhdepuhaltimien jäähdytysteho ei riitä. Lämpötilaa mitataan lauhduttimessa olevalla lämpöanturilla (DS18B20). Tämän anturin mittaustiedolla pysäytetään myös kompressori, jos lämpö nousee yli hälytysrajan. Kuivaimen varolaitteet ovat: moottorisuojakytkimet, kompressorin öljynpainekeytkin, moottoritermostori ja korkeapainekeytkin.

3 MIKROKONTROLLERIT

3.1 Historia

Ensimmäisen mikrokontrollerin kehitti Amerikkalainen yritys Texas Instruments 1970-luvun alkupuolella. Mikrokontrollerien ja prosessorien kehitys kulkee käsi kädessä. Ensimmäinen mikroprosessorin Intel 4004 esitteli puolijohdevalmistaja Intel vuonna 1971, samoihin aikoihin kehitti myös Texas Instruments oman 4-bittisen TMS 1000 mikroprosessorin, joita se käytti aluksi vain omissa taskulaskimissa. Prosessorit olivat vaatimattomia tiedonkäsittelykapasiteetiltaan nykyisiin prosessoreihin verrattuna. Intel 4004 käsitteli dataa neljän bitin paketteina, ja transistoreita siinä oli 2300, uusimmissa niitä on miljardeja, esim. Intelin 15-core Xeon Ivybridge-EX:ssä niitä on 4,3 miljardia. Transistoreita oikein ohjaamalla luodaan haluttu logiikka. (Nikkilä)

Mikrokontrolleri on sulautetun laitteen aivot, joka sisältää aina prosessorin. Sulautettuja laitteita on kaikkialla esimerkiksi taskulaskin, rannekello, henkilöauto ja sykemittari. Mikrokontrolleri on pieni tietokone, joka on integroitu pienelle piipalalle.

3.2 Mikrokontrollerin osat

Mikrokontrollerissa yleensä ainakin seuraavat osat.

3.2.1 Prosessori

Prosessoria sanotaan mikrokontrollerin sydämeksi. Se suorittaa annetun käskyn ohjelman mukaan ja lopuksi tallentaa tiedon muistiin tai siirtää käskyn I/O-portille, joka ohjaa esim. releitä.

3.2.2 Käyttömuisti (SRAM)

Käyttömuistista häviää tiedot käytösähkön katkettua. Käyttömuistiin tallennetaan käsiteltävä ja käsitelty tieto. SRAM-muisti on hyvin nopeaa, muistisolun luku/kirjoitusaika on vain 5-10ns.

3.2.3 Ohjelmamuisti(flash)

Ohjelmamuistin on säilytettävä tietonsa vaikka sähkötkä katkeavat. Ohjelmamuistina käytetään pääsääntöisesti flash-muistia. Ohjelma muistiin talletetaan esim. ohjelma koodi.

3.2.4 EEPROM-muisti

EEPROM-muistia käytetään käyttömuistina sellaiselle tiedolle, joka ei saa haihtua sähkön katkettua.

3.2.5 Tulo- ja lähtöpinnit

I/O-liitäntä mahdollistaa prosessorin ulkopuolisiin laitteisiin liittymisen. Voidaan ottaa mittaustieto vastaan esim. antureilta tai ohjata moottoria. I/O-tieto voi olla analogista, digitaalista, rinnakkais- tai sarjamuotoista.

3.2.6 AD-muunnin

AD-muuntimen tehtävä on muuttaa analogiaviesti prosessorille ymmärrettävään digitaaliseen muotoon.

3.2.7 PWM-lähtö

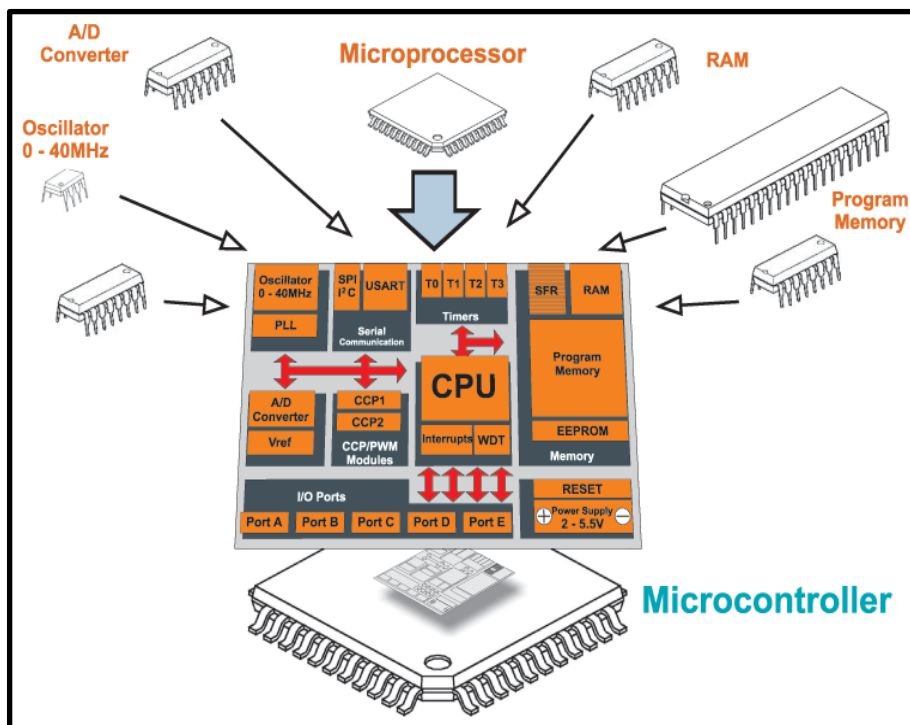
PWM- eli pulssinleveysmodulaatio. Jännitteen säätö perustuu jännitepulssin leveyteen, amplitudi pysyy samana. Säädetään aikaa kuinka kauan pulssi on ylhäällä eli pulssin leveyttä. PWM-lähdöllä voidaan säätää esimerkiksi DC-sähkömoottorin pyörintä nopeutta.

3.2.8 Watchdog

Vahtiajastin tarkistaa ohjelman toiminnan määräajoin, jos ohjelmassa tapahtuu virhe ja ohjelma jumittuu käynnistää vahtiajastin ohjelman uudelleen

3.2.9 Ajastin

Mikrokontrollerin ajastimia käytetään ohjauksen aikaa mittaavissa toimenpiteissä. (Vahtera, 2015)



Kuva. 4 Mikrokontrollerin sisältämät osat. (MikroElektroniikka)

Mikrokontrolleri ohjaa laitteita I/O-pinneihin liitettyjen tulojen avulla ja käsittelee tulodataa ja ohjaa lähtöjä kontrolleriin ladattavan ohjelmakoodin mukaan. Mikrokontrollerien ohjelma kirjoitetaan yleensä tietokoneella olevalla ohjelmointiohjelmalla, joka kääntää ohjelman koneielelle ennen ohjelman siirtoa kontrollerille. Mikrokontrollereiden yleisimpiä ohjelmointikieliä on C-kieli.

4 ARDUINO

4.1 Yleistä

Arduino on avoimeen lähdekoodiin perustuva kehitysalusta. Arduinolla on oma Wiring-ohjelmointikieli, joka perustuu C-kieleen. Arduino on nykyään hyvin suosittu kehitysalusta. Arduinoon löytyy todella hyvin ohjeita ja paljon esimerkkiohjelmia ja projekteja netistä, mihin suosio todennäköisesti perustuu. Arduinon kontrollereita on tarjolla monia eri vaihtoehtoja. Arduino Yún on uusimpia tulokkaita. Tällä hetkellä on tulossa kaksi uutta versiota Arduino Tre ja Arduino Zero. Tre on samankaltainen kuin Yún, siinä on sama Atmel ATmega32u4 mikrokontrolleri ja erillinen Linux prosessori kuten Yúnissa.

4.2 Arduino Yún

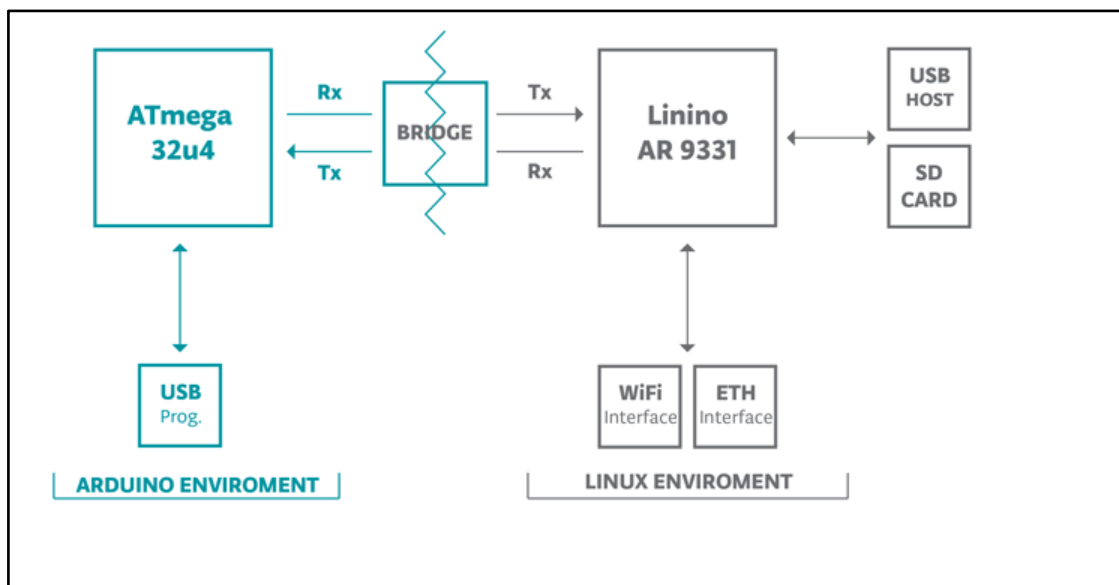
Arduino Yún on hyvin samankaltainen kuin Leonardo, ja siinä on sama prosessori Atmel ATmega 32U4 ja Micro-B USB. Arduino Yúnissa on lisäksi Atheros AR9331-posessori, joka tukee linux jakeluja. Uusia käteviä ominaisuuksia mitä muissa Arduino-kehitysalustoissa ei ole: kortille sisään rakennettu Ethernet, WLAN ja SD-korttipaikka. (Arduino)



Kuva 5. Arduino Yún (Arduino)

4.3 Liitynnät

Micro-B USB kautta voit liittää Yúnin tietokoneeseen, tätä kautta se saa virran ja samalla voi siirtää dataa. Ethernet, WLAN, USB A ja SD-muistikortin liitynnät ei ole fyysisesti yhteydessä 32U4 prosessoriin, mutta Bridge-kirjastot mahdollistavat kommunikoinnin AR9331- ja 32U4-prosessorin välillä. WLAN-verkon kautta voi langattomasti ohjelmoida Arduinoa, kun se on ensin liitetty omaan WLAN-verkkoon. (Arduino)



Kuva 6. Liitynnät. (Arduino)

4.4 Arduino Yún tekniset tiedot. (Arduino)

AVR Mikrokontrolleri	
Mikrokontrolleri	ATmega32u4
Käyttö jännite	5V
Sisääntulo jännite	5V
Digitaaliset tulot	20
PWM kanava	7
Analogia tulot	12
DC virta /I/O Pin	40 mA
DC virta / 3.3V Pin	50 mA
Flash muisti	32 KB
SRAM	2.5 KB
EEPROM	1 KB
Kellotaajuus	16 MHz
Linux Mikroprosessori	
Proessori	Atheros AR9331
Arkkitehtuuri	MIPS @400MHz
Käyttöjännite	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0 Host
Kortti paikka	Micro-SD
RAM	64 MB DDR2
Flash muisti	16 MB

4.5 WLAN-verkon konfigurointi

Yúnin on mahdollisuus liittää salaamattomaan- tai verkkoon jotka tukevat WEP, WPA tai WPA2 salausta. Kun Yúniin kytkee ensimmäisen kerran virrat, se luo oman WLAN-verkon nimeltään ArduinoYún. Liitytään tietokoneella tähän verkkoon, avataan selain ja kirjoitetaan `http://arduino local` tai `192.168.240.1`.

Valintaikkunassa kysytään salasanaa, joka on oletuksena ”Arduino” ja painetaan ”Log in ja Configure” nappia. Konfigurointi sivulla valitaan uusi käyttäjänimi ja sa-

lasana. Seuraavaksi valitaan oikea aikavyöhyke, se on suositeltavaa, koska tämä auttaa yhdistämisessä paikallisiin verkkoihin. Valitaan verkko mihin halutaan liittyä, verkon salaustyyppi ja salasana. Painetaan ”Configure & Restart” painiketta ja odotetaan, että konfigurointi on valmis ja kytketään tämän jälkeen virta pois.

The image shows two configuration windows for an Arduino-based device. The first window, titled "YÚN BOARD CONFIGURATION", contains the following fields: "YÚN NAME *" with the value "MyYun", "PASSWORD" (masked with dots), "CONFIRM PASSWORD" (masked with dots), and "TIMEZONE *" with a dropdown menu set to "America/New York". The second window, titled "WIRELESS PARAMETERS", has a checked checkbox for "CONFIGURE A WIRELESS NETWORK", a "WIRELESS NAME *" field with the value "AccessPoint", a "SECURITY" dropdown menu set to "WPA2", and a "PASSWORD *" field (masked with dots). At the bottom of the second window are two buttons: "DISCARD" and "CONFIGURE & RESTART".

Kuva 7. Konfigurointi ikkuna. (Arduino)

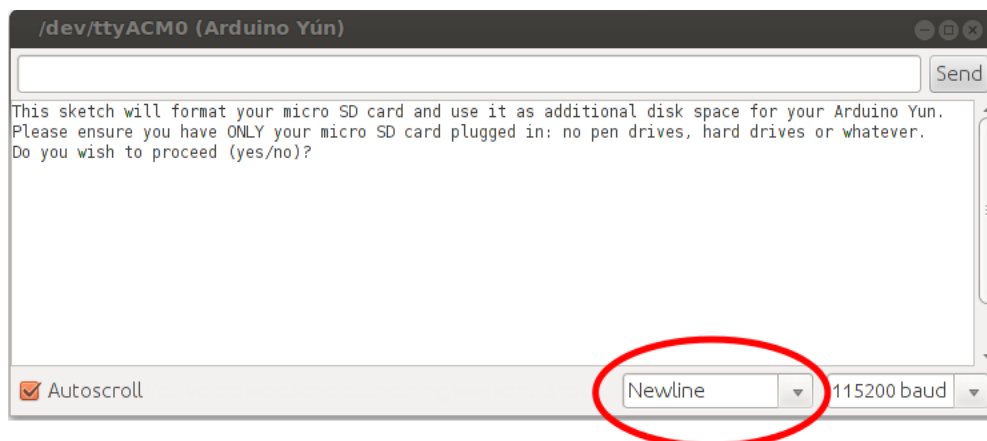
Tietokoneeseen pitää asentaa Arduino IDE, jolla ohjelmat kirjoitetaan. Yúnia tukee vain versio 1.5.4 ja uudemmat, tässä projektissa on käytössä versio 1.5.8. Kun IDE on asennettu, käynnistetään ohjelma ja valitaan työkaluvalikosta oikea kortti ja portti ja tämän jälkeen konfigurointi on valmis. (Arduino)

4.6 Yún muistin laajennus

Linux OpenWRT-Yún -käyttöjärjestelmä on asennettu valmiiksi Yúniin, joka vie kortissa olevasta 16MB:n flash-muistista 9MB. Tässä projektissa tämä muistitila ei todennäköisesti riitä, joten on laajennettava muistia.

Muistitilan laajennus SD-kortilla: Arduinon sivulla on konfigurointiin valmis sketsi, joka käyttää Bridge-kirjastoja SD-kortin liittämiseksi Yúniin. Sketsi ohjaa askel kerrallaan mitä tehdään. Sketsi jakaa SD-kortin kahteen osaan. Ensimmäinen osa on FAT32, tätä osaa voidaan käyttää tiedostojen jakamiseen tietokoneen ja Yúnin välillä. Toinen osa sisältää Linux tiedostojärjestelmiä, tämä ei ole käytettävissä tietokoneella, joka käyttää Mac- tai Windows-käyttöjärjestelmää. Tästä lähtien kaikki mitä tehdään Linux-puolella, tallentuu SD-kortille.

Asennuksen alussa ladataan sketsi Arduinon nettisivulta ja puretaan se Arduinon IDE sketsi kansioon. Avataan IDE ja ladataan sketsi. Seuraavaksi irrotetaan kaikki ylimääräiset laitteet ja kytketään vain SD-kortti ja avataan Serial Monitor.



Kuva 8. Serial Monitor ikkuna. (Arduino)

Tarkistetaan että pudotusvalikossa lukee ”Newline”. Monitori ohjaa irrottamaan ylimääräiset laitteet irti, kirjoitetaan ”Yes” ja painetaan ”Enter”. Ensimmäisessä askeleessa ohjelma hakee lisä ohjelmia netistä jonka jälkeen se alustaa SD-kortin. Seuraavaksi kysytään kuinka paljon muistia halutaan antaa yleiseen käyttöön. Loppu SD-kortin muistista on omistettu Linuxille. Tässä projektissa muisti jaettiin 0,9/1,1GB. Lopuksi monitori ilmoittaa onnistuiko konfigurointi ja näyttää miten SD-kortti on jaettu. (Arduino)

5 ANTURIT

Tässä projektissa tarvitaan antureita neljän eri suureen lämmön, kosteuden, paineen ja virran mittaukseen. Lämpöantureita on kaksi, toinen mittaa kuivaamokaapin lämpötilaa, jolla ohjataan lämpöpuhaltimien käyntiä. Lauhdutinkennossa on lämpöanturi, joka mittaa lauhduttimen ylälämpöä, tällä ohjataan jäähdytinpuhallinta ja se sammuttaa tarvittaessa kompressorin, lämpötilan noustessa yli hälytysrajan. Kosteusanturi mittaa kuivaamo kaapin kosteutta, joka ohjaa kuivaimen käyntiä magneettiventtiilin avulla. Matalapainekytin ohjaa kompressorin käyntiä, kompressori sammuu, kun imupuolen paine laskee säädetyn rajan alle. Virtamittari mittaa kompressorin ottamaa virtaa verkosta.

5.1 Lämpötila-anturit

Lämpötilan mittaamiseen on kehitetty monia eri menetelmiä, seuraavassa muutama esimerkki.

- Platinasta valmistetut PT100 ja PT1000, näiden toiminta perustuu materiaalin resistanssin riippuvuuteen lämpötilasta.
- Nestepohjainen lämpömittari ja kaksoismetallilämpömittari, näiden toiminta perustuu materiaalien lämpölaajenemiseen.
- Termopari, jonka toiminta perustuu kahden eri metallin liitoksessa syntyvään jännitteeseen, joka on riippuvainen lämpötilan muutoksesta.
- Infrapunakamera, joka mittaa mitattavan kohteen lähettämää lämpösäteilyä.

Lämpötila-anturin mittaustieto voidaan lähettää ohjaukselle analogisena tai digitaalisena viestinä. Yleensä mittauslähetin muuttaa analogisen kenttäsuureen numeeriseksi binäärikoodiksi, jos käytetään digitaalista viestintää. Tässä projektissa käytettävä DS18B20-lämpötila-anturi muuttaa lämpötilan suoraan digitaaliseen muotoon. DS18B20-lämpötila-anturi on 1-wire väylään liitettävä, johon voi liittää 124 anturia ja väylän pituus voi olla max. 300 metriä. Lämpötila-anturin valinta tähän projektiin perustui pitkälti väylään liitettävyydellä. Yúnissa I/O-määrä oli aika tarkka tähän projektiin ja tämä vaatii vain yhden tulon.

5.1.1 DS18B20-lämpötila-anturi

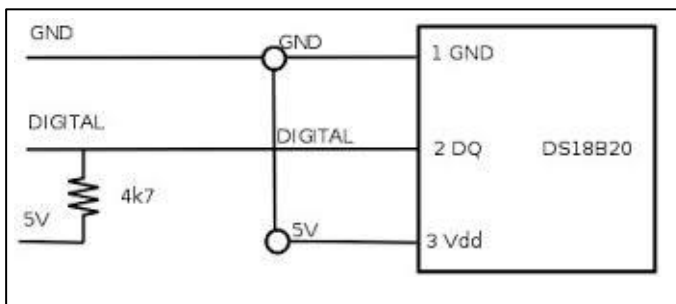
Dallas DS18B20 lämpöanturin tiedonsiirto perustuu 1-wire bus tekniikkaan. 1-wire väylän on kehittänyt Dallas puolijohde valmistaja.

Dallas DS18B20-lämpötila-anturi muuttaa lämpötilan suoraan digitaaliseen muotoon. Anturin toiminta perustuu kahteen oskillaattoriin (värähtelijään), voimakkaasti lämpötilasta riippuvaisesta oskillaattorista, jonka värähtelytaajuus muuttuu lämpötilan mukaan lähes lineaarisesti ja lähes lämpötilariippumattomasta oskillaattorista. Anturin elektroniikka laskee kuinka monta pulssia lämpötilariippumaton oskillaattori tuottaa voimakkaasti lämpötilariippuvaisen oskillaattorin määräämänä aikana. Lämpötilan kohotessa lämpötilariippuvaisen oskillaattorin jaksonaika suurenee, ja näin myös mittausjaksolle mahtuvien pulssien määrä kasvaa. Anturin elektroniikka laskee oskillaattorien tuottamien pulssien erotuksen, josta se pystyy laskemaan vastaavan lämpötilan. (Pulli)

5.1.2 1-wire väylä

1-wire väylä on master-slave -tyyppinen eli isäntä aloittaa aina tiedonvaihdon, isäntä kysyy ja orja vastaa. Väylä on lepotilassa aina ylätilassa, näin anturi saa käyttöjännitteensä lepotilassa. Alatilassa väylä on hyvin pieniä aikoja ja silloin käyttöjännitteensä anturi saa sen sisäisestä kondensaattorista. Tämä mahdollistaa kaksijohdin kytkennän. Jokaisella laitteella 1-wire väylässä on yksilöllinen ja muuttumaton 64-bittinen

laite-ID, joka on asetettu tehtaalla. Laite-ID sisältää 8-bittisen perhekodein, 48-bittisen yksilöllinen koodin ja 8-bittisen tarkistussumman. (Nic)



Kuva 9. 2-johdin kytkentä (HQEW)



Kuva 10. 3-johdin kytkentä (HQEW)

Eri tilat erotetaan toisistaan pulssinpituuden perusteella

Taulukko3. Pulssipituudet 1-wire väylässä (Dallas DS18S20). (Honkanen)

AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}=3.0V$ to 5.5V)

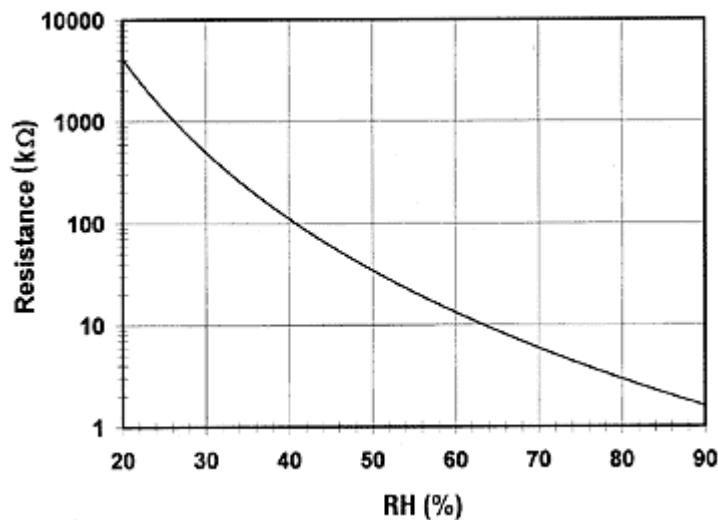
PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	t_{CONV}				750	ms	1
Time to Strong Pullup On	t_{SPON}	Start Convert T Command Issued			10	μs	
Time Slot	t_{SLOT}		60		120	μs	1
Recovery Time	t_{REC}		1			μs	1
Write 0 Low Time	t_{LOW0}		60		120	μs	1
Write 1 Low Time	t_{LOW1}		1		15	μs	1
Read Data Valid	t_{RDV}				15	μs	1
Reset Time High	t_{RSTH}		480			μs	1
Reset Time Low	t_{RSTL}		480			μs	1,2
Presence Detect High	t_{PDHIGH}		15		60	μs	1
Presence Detect Low	t_{PDLow}		60		240	μs	1
Capacitance	$C_{IN/OUT}$				25	pF	

5.2 Kosteusanturi

Ilman suhteellista kosteutta mitataan monella eri toimintaperiaatteella, seuraavassa kaksi esimerkkiä.

5.2.1 Resistiivinen kosteusanturi

Anturin toiminta perustuu resistanssin muutokseen suhteessa mitattavan kaasun kosteuteen. Siinä on yleensä jalometalliset elektrodiparit, joiden välissä on suolalla tai polymeeriseoksella käsitelty substraatti. Elektrodien päistä mitattava resistanssi on kääntäen verrannollinen mitattavan kaasun suhteelliseen kosteuteen. Vasteaika resistiivisillä antureilla on yleensä 10-30s 63% askelmuutoksesta. Resistiivisen anturin resistanssi on epälineaarinen kosteuden- ja lämpötilan suhteen, joten se on kompensoitava anturissa elektroniikalla. (Sensors online)

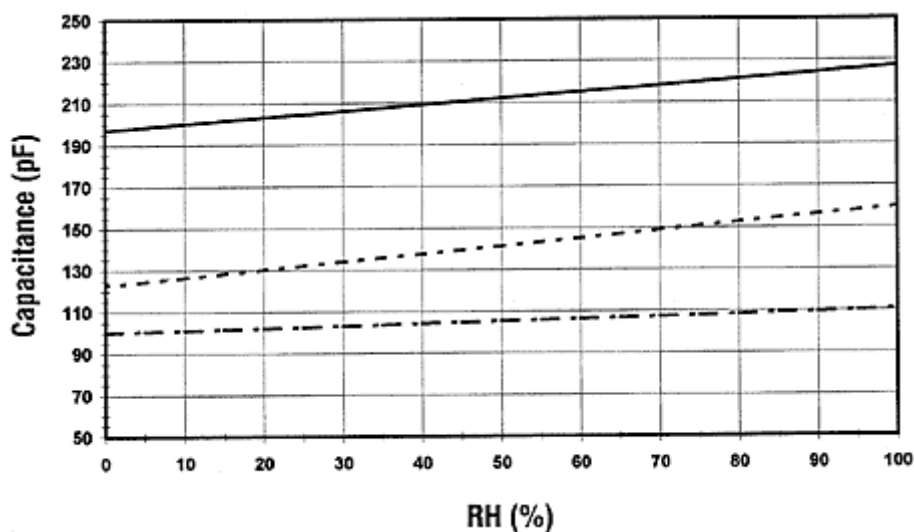


Kuva 11. Esimerkki resistiivisen kosteusanturin resistanssin muutoksesta RH(%) muutokseen. (Sensors online)

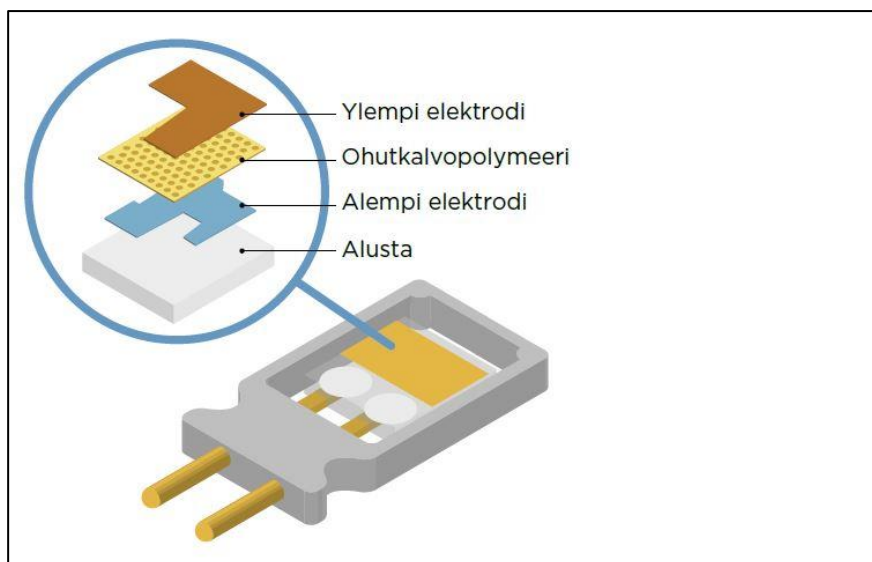
5.2.2 Kapasitiivinen kosteusanturi

Anturin toiminta perustuu kapasitanssin muutokseen suhteessa mitattavan kaasun kosteuteen. Anturin rakenne on seuraavanlainen: siinä on ohut polymeerikalvo tai metallioksidia, joka päällystetty kahdella huokoisella metallielektrodilla, jotka suo-

jaavat tunnustuskalvoa epäpuhtauksilta. Anturin väliaineen permittiivisyys muuttuu mitattavan kaasun kosteuden mukaan ja samalla elektrodien päistä mitattava kapasitanssi muuttuu kaavan $C = (\epsilon * A)/s$ mukaan. Anturin kapasitanssi muuttuu lähes lineaarisesti kosteuden suhteen ja lämpötilakerroin on hyvin alhainen, joten lämpötilan muutos ei vaikuta kovin paljon mittaustulokseen ja toimii hyvin korkeissa lämpötiloissa. Vasteaika kapasitiivisilla antureilla on yleensä 30-60s 63% askelmuutoksesta. (Sensors online)



Kuva 12. Esimerkki kapasitiivisen kosteusanturin kapasitanssin muutos RH(%) muutokseen. (Sensors online)



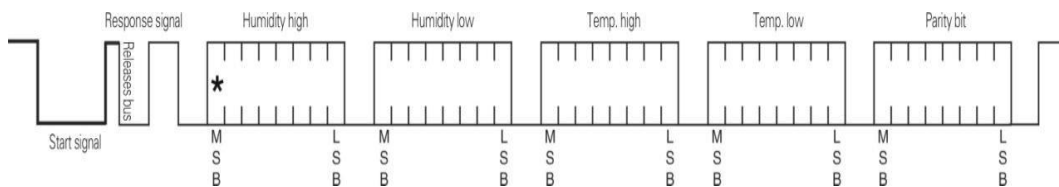
Kuva13. Kapasitiivisen ohutkalvopolymeerianturin rakenne (Vaisala)

5.2.3 Kuivaamokaapin kosteusanturi

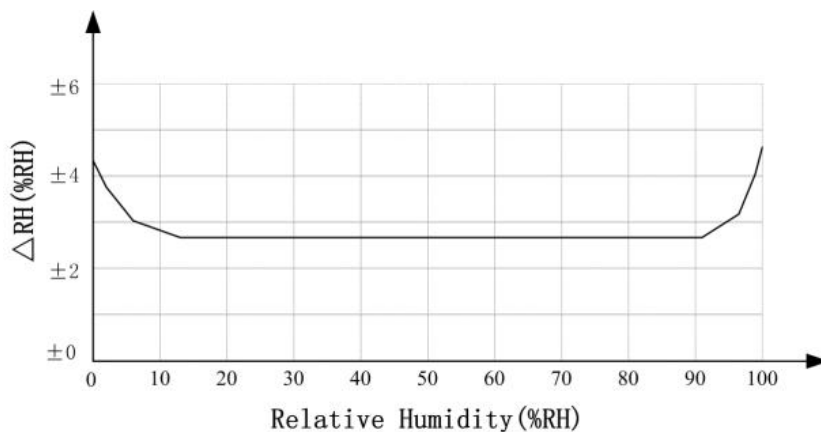
Tähän projektiin valittiin digitaalisella ulostulolla oleva anturi (AM2302), jossa on kapasitiivinen kosteusanturi ja lämpöanturi samassa. Anturi käyttää tiedonsiirtoon yksinkertaistettua yhden väylän tiedonsiirtoa, jossa tieto siirretään yhtä datajohdinta pitkin kontrollerille. Data siirretään neljäkymmenen bitin paketeissa. Kontrollerilta lähetetään ensin aloitus signaali anturille, johon anturi vastaa vetämällä väylän $80\mu\text{s}$:ksi ensin alas ja sitten ylös, jonka jälkeen anturi lähettää mittausdatan. Kuusi-toista ensimmäistä bittiä on kosteuden mittausarvo ja seuraavat kuusitoista bittiä lämpötilan mittausarvo ja viimeiset kahdeksan on pariteettibitit. Tässä käytetään paritonta pariteettibittitarkistusta, joka on jaettu neljän bitin ryhmiin. Lämpötiladatan viidestoista bitti osoittaa onko tulos negatiivinen tai positiivinen, jos tulos negatiivinen bitti on 1. (Aosong Electronics)

Data	0000 011+1001 0010 +1000 0001+0001 0101= 1110 0001
Kosteus	0000 011+1001 0010=392H=3*256+9*16+2=660/10=66.0%
Lämpötila	1000 0001+001 0101=104H=256+16+4=276/10=-27.6°C

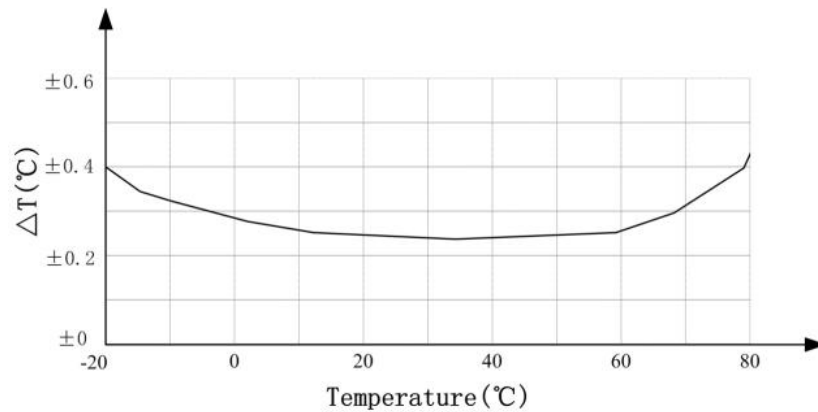
Kuva 14. Laskuesimerkki mittauksesta.



Kuva 13. AM2301 Single-bus kommunikointi protokolla. (Aosong Electronics)



Kuva 15. AM2301 suhteellisen kosteuden mittaus virhe 25°C lämpötilassa. (Aosong Electronics)



Kuva 16. AM2301 lämpötilan maksimi mittausvirhe eri lämpötiloissa. (Aosong Electronics)

5.3 Virran mittaus

Virran mittaus ilman galvaanista kosketusta mitattavaan suureeseen.

5.3.1 Hall-ilmio

Hall-ilmion käyttö virran mittauksessa. Voimaa joka aiheuttaa Hall-ilmion kutsutaan Lorentzin voimaksi, voima vaikuttaa kaavan $F = q(E + v * B)$ mukaan.

F = Lorentzin voima

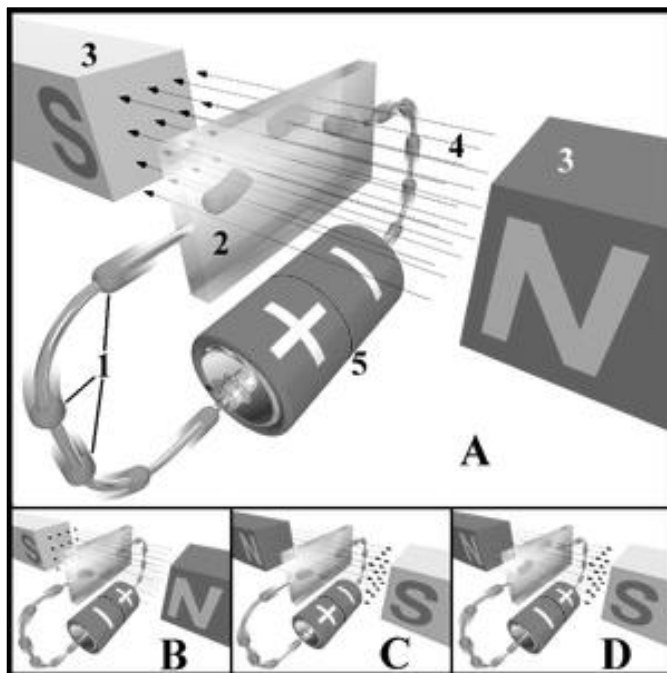
q = Alkeisvaraus

E = Sähkökentän voimakkuus

v = Elektronin nopeus

B = Magneettikentän voimakkuus

Mitataan Hall-jännitettä esim. puolijohteesta kohtisuoraan virran ja magneettikentän suuntaan tasaisessa magneettikentässä, kun virta kasvaa Hall-jännite nousee. Voidaan mitata AC- ja DC -virtoja



Kuva 17. Hall-anturin toiminta. (Wikimedia commons)

1. Elektroni(Mitattava virta)
2. Hall-anturi
3. Magneettikenttä
4. Magneettikenttä
5. Virtalähde

5.3.2 Sähkömagneettinen induktio

Faradayn lain mukaan muuttuvassa magneettikentässä olevaan virtapiiriin indusoituu

lähdejännite kaavan $E = -N \frac{d\Phi}{dt}$ mukaan.

E = Lähdejännite, sähkömotorinen voima

N = Johdinkierrosten määrä

$\frac{d\Phi}{dt}$ = Vuon muutosnopeus

Jännitteen voimakkuuteen vaikuttavat magneettikentän muutos nopeus. Amperen lain mukaan johtimessa kulkeva virta muodostaa ympärilleen magneettikentän, jonka voimakkuus kasvaa kun virta kasvaa. AC-pihtimittarin toiminta perustuu tähän ilmiöön, kun virtapihtien sisälle laitetaan johdin, jossa kulkee vaihtovirta, muodostaa se ympärilleen muuttuvan magneettikentän, jonka muutosnopeuteen vaikutta virran

voimakkuus ja vaihtovirran taajuus. Muuttuva magneettikenttä indusoi ympärillä olevaan käämiin lähdejännitteen. Magneettikenttä muuttuu, koska virran suunta muuttuu. Magneettikentän voimakkuus on suoraan verrannollinen virran voimakkuuteen.

5.3.3 Kompressorin virranmittaus

Kompressorin virtamittauksella on kaksi tehtävää: energiankulutuksen seuranta ja kylmäaineen vuotojen seuranta. Virtamittaukseen valittiin Dechang Electronic Ltd:n valmistama (SCT-013-000) virtamuunnin. Muuntimen muuntosuhde on 100/0.05A. Muuntimen ja Arduinon väliin on lisättävä kuormavastus ja jännitteenjakovastukset. Seuraavassa kuormitusvastuksen laskukaava. (Openenergymonitor)

$$(U_a/2)/(I_{max} * \sqrt{2}/N) = R_k$$

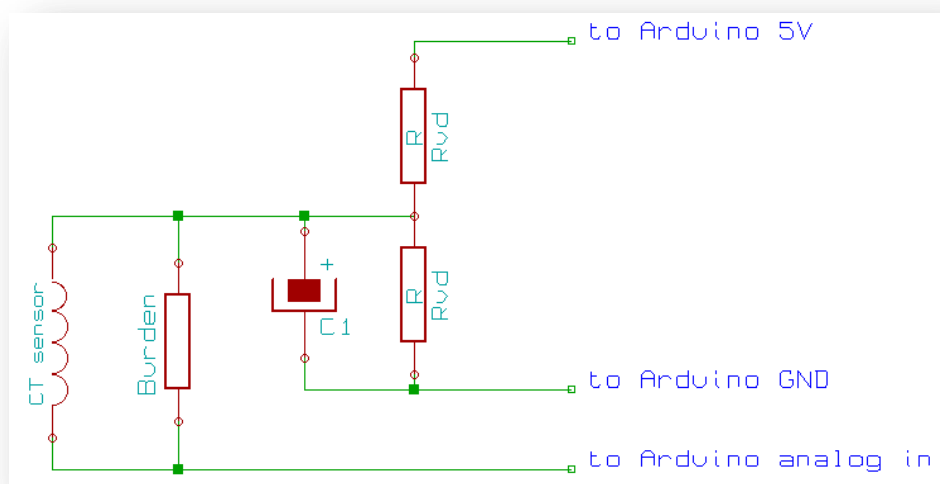
$$(5V/2)/(100A * \sqrt{2}/2000) = 34,7\Omega$$

R_k = Kuormavastuksen koko

U_a = Arduinon input jännite alue

N = Käämin kierrosten lukumäärä

I_{max} = Ensiöpuolen maksimi virta



Kuva 18. Esimerkki virtamuuntimen kytkennästä Arduinoon. (Openenergymonitor)

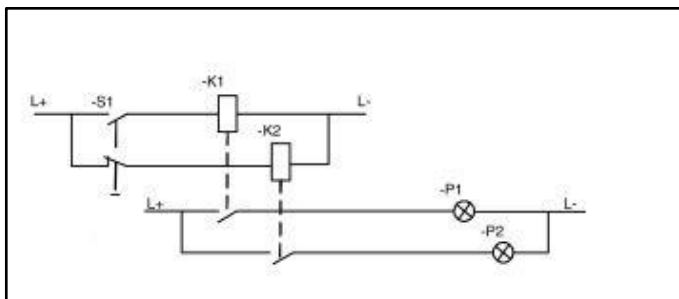
6 OHJAUSKESKUS

Tässä kappaleessa kuvataan piirikaavion suunnittelu, ohjauskeskuksen komponenttien valinta ja johdotukseen liittyvät standardit.

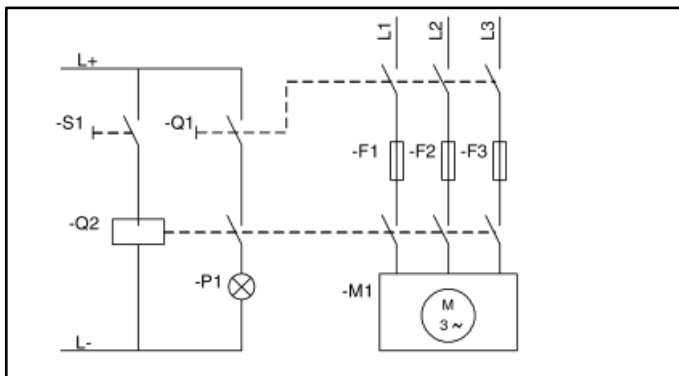
6.1 Piirikaavio

Piirikaaviot ovat sähkölaitteiston ohjauksen toiminnan ymmärtämistä varten. Standardi (SFS EN 61082-1) määrittelee piirtämissääntöjä. Piirikaavio esittää laitteen sähkötekniset yksityiskohdat, mutta ei määrittele komponenttien todellista kokoa muotoa tai sijaintia. Piirikaavio sisältää seuraavia elementtejä piirrosmerkkejä (komponentteja), liitännäviivoja, jolla yhdistetään komponentit, komponenttien yksilölliset tunnuksot (viitetunnuksot), liitintunnuksot, signaalitunnuksot ja sijaintiviitteet ja lisäinformaatiota, jos on tarpeen. Piirikaavion esitystapoja on sidottu-, koottu- ja vapaa esitystapa. Yleisin näistä on vapaa esitystapa. Tämän laitteen piirikaavion suunnittelu tehdään vapaalla esitystavalla CADS suunnitteluohjelmalla.

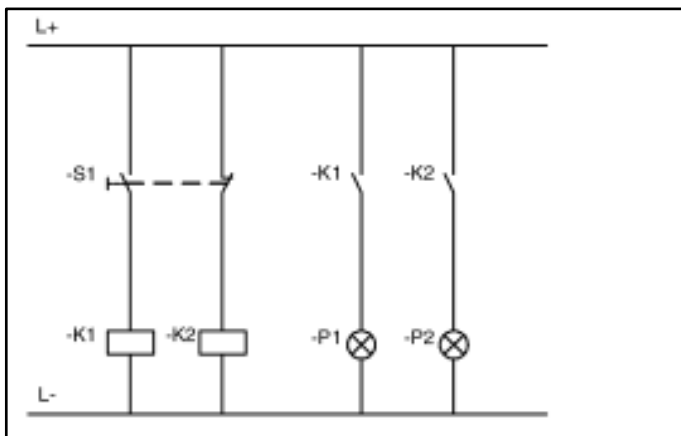
(SFS, SFS EN 61082-1, 2006)



Kuva 19. Sidottu esitystapa.



Kuva 20. Koottu esitystapa.



Kuva 21. Vapaa esitystapa.

6.2 Komponenttien valinta

6.2.1 Pääkytkin

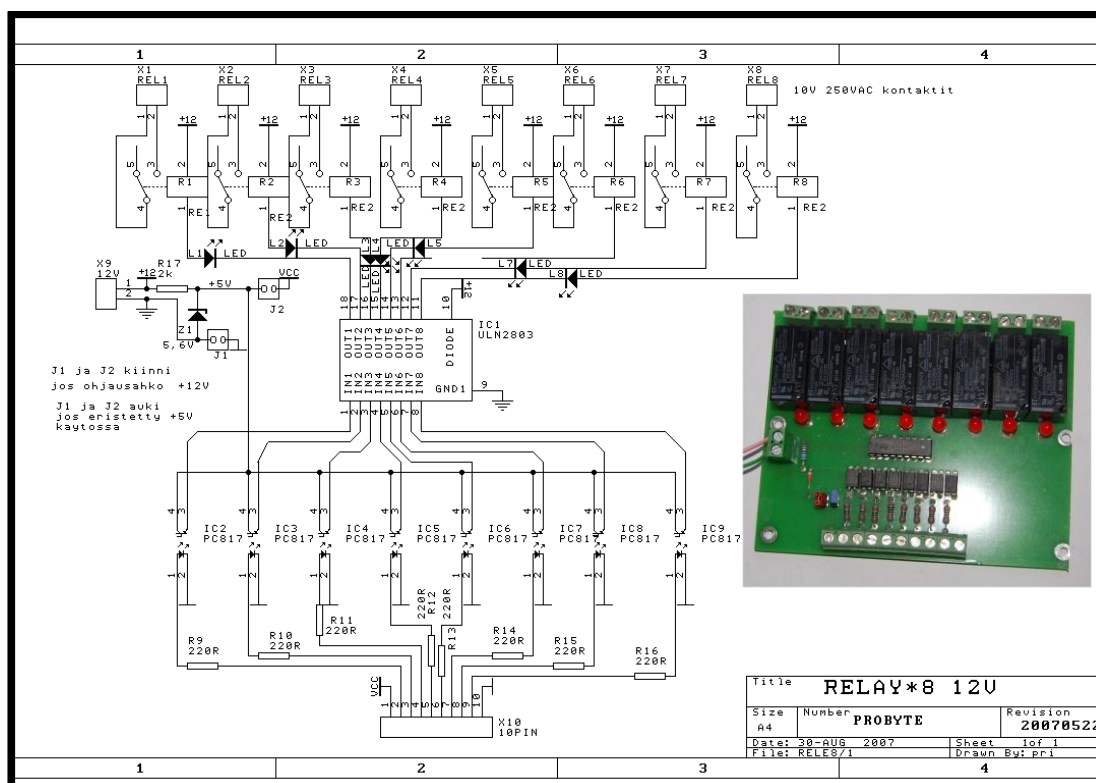
Pääkytkimen valintaan vaikuttavat seikat ovat: keskuksen ohjattavien laitteiden otama nimellisvirta, syötön sulakekoko, oikosulkuvirta ja asennuspaikka keskuksessa.

6.2.2 Moottorinsuojakytkimet

Moottorinsuojakytkin toimii sekä oikosulkusuojana, että ylikuormitussuojana. Moottorinsuojan bi-metallielementti valvoo moottorin ylikuormitusta ja sähkömagneettinen laukaisu valvoo piirin oikosulkua. Moottorinsuojan valitaan moottorin nimellisvirran perusteella ja on huomioitava, että sillä on riittävä oikosulun katkaisukyky. (Hietalahti, 2013)

6.2.3 Releet

Arduinon I/O-portti on hyvin herkkä, ja sen ohjausvirta on maksimissaan 40 mA, tästä johtuen kontaktoria ei voida ohjata suoraan. Relettä valittaessa on otettava huomioon mitä releellä ohjataan. Puolijohdereleellä voi tulla ongelmia, jos kuorma on kokonaan induktiivista. Tähän projektiin valittiin välireleeksi optoerotettu relekortti, jossa on kahdeksan tavallista 12VDC:n ohjausjännitteellä olevaa relettä, johon syötetään erillinen 12V:n ohjausjännite. Arduinon ulostulo signaali vahvistetaan relekortissa olevalla ULN2803-piirillä, kun ULN2803-piiriin kytketään 5V:n ohjausjännite tulopuolelle, kytkeytyy lähtöpuoli 0-pisteeseen.



Kuva 22. Relekortti

6.2.4 Kontaktorit

Kontaktorin valintaan vaikuttavat ohjattavan laitteen kuorma, jännite ja käyttöluokka. Oikosulkumoottorien ohjauksessa käytettävän kontaktorin käyttöluokka on AC-3 tai AC-4. AC-3 käyttöluokassa kontaktorien pitää pystyä kytkemään käynnistysvirran ja katkaisemaan nimellisvirran vähintään miljoonaa käyttökertaa. AC-4 käyttöluokassa kontaktorin on pystyttävä kytkemään ja katkaisemaan käynnistysvirta ja toimintakertojen mitoitus määritetään tapauskohtaisesti. (Hietalahti, 2013)

6.2.5 Johdotus

Johtimien tunnistamisesta standardin (SFS_EN 60204-1) mukaan jokainen johdin pitää pystyä tunnistamaan liitoksessa teknisten piirustusten mukaan. Keskuksen johtimien tunnistamiseen voi käyttää numeroita, kirjaimia tai värejä. Suojajohtimen väri on aina oltava keltavihreä. Nollajohtimen väriksi suositellaan vaaleansinistä väreä, jos tunnistamiseen käytetään värejä, nollajohtimen tunnistamiseen voi käyttää myös muita menetelmiä esim. kirjaimia tai numeroita. Muut suositellut johdinvärit, jos tunnistamiseen käytetään värejä. (SFS, SFS_EN 60204-1)

Taulukko 5. Johdinvärit

Musta	AC- ja DC päävirtapiirit
Punainen	AC ohjausvirtapiirit
Sininen	DC ohjausvirtapiirit
Oranssi	Ulkopuolinen ohjaus

Taulukko6. Keskuksen sisäisen johdotuksen ohjearvot. (SFS, SFS käsikirja 154 Jakokeskukset.)

Johdotus yhdellä johtimella	
Johtimenpoikkipinta kupari mm ²	Kuormittettavuus A
1,5	14
2,5	20
4	26
6	33
10	62
16	82
25	107
35	135
50	160
70	200
95	245
120	280
150	320
185	365
240	425

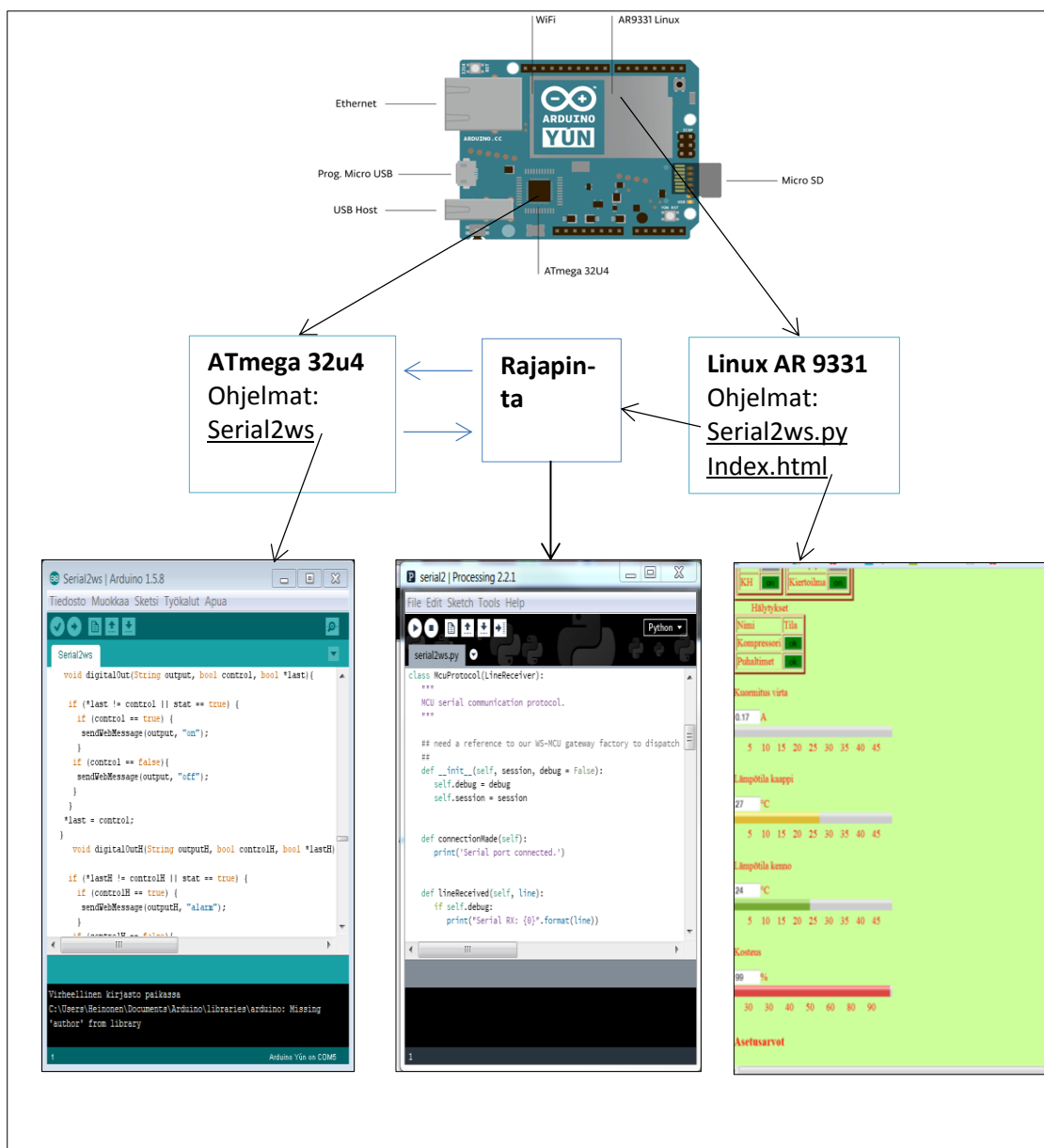


Kuva 23. Ohjauskeskus

7 OHJELMAT

7.1 Yleistä

Tässä sovelluksessa on kolme eri ohjelmaa. Kontrollerin (ATmega 32u4) ohjelma ohjaa I/O:ta ja lähettää ja vastaan ottaa tietoa prosessorilta (Linux AR 9331). Prosessorin ohjelma välittää tietoa kontrollerilta selaimelle ja toisinpäin. Selaimen ohjelma välittää tiedon käyttäjälle ja käyttäjän käskyt prosessorille.



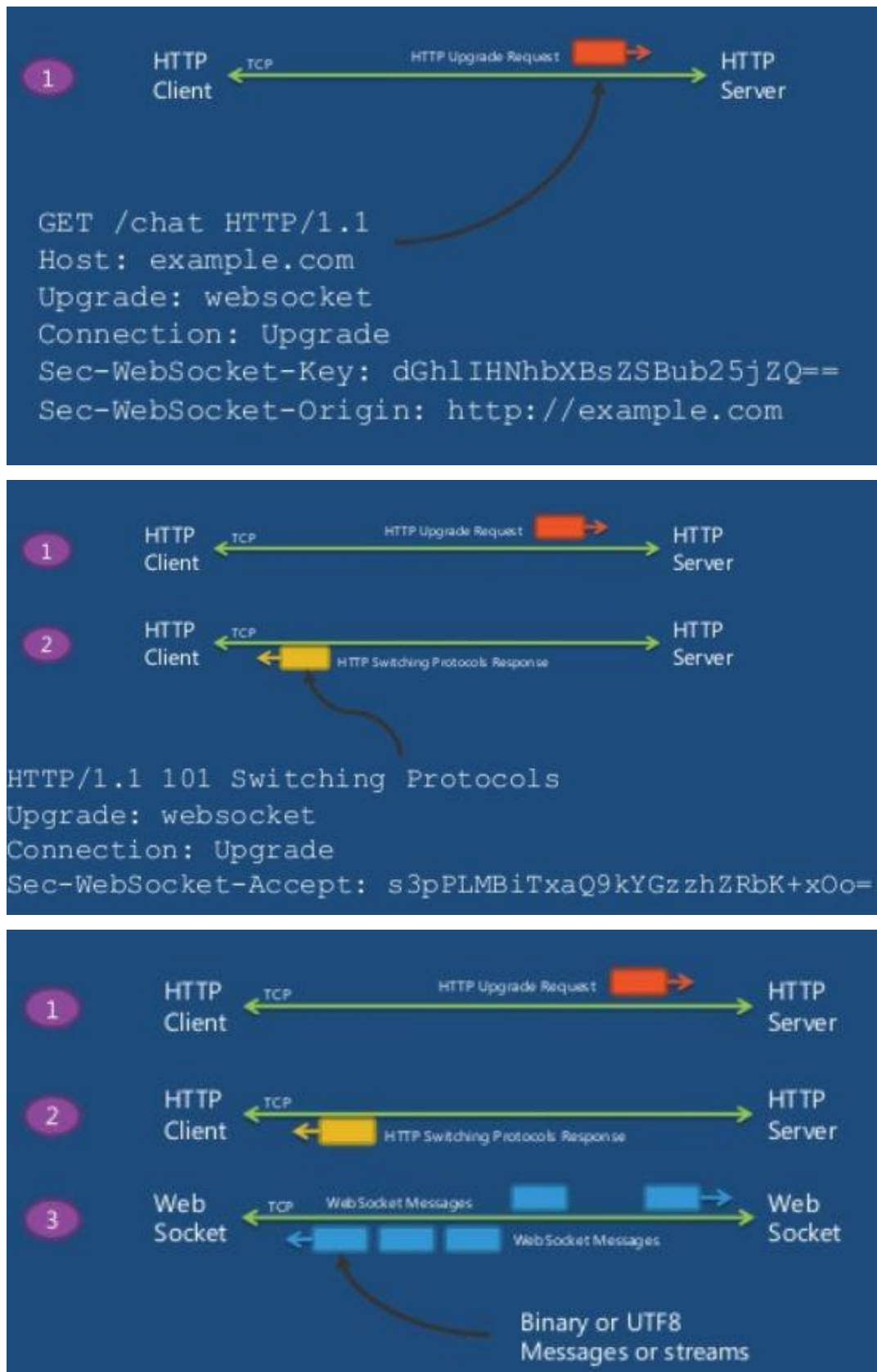
Kuva 24. Ohjelmien fyysinen hierarkia

7.2 Autobahn

Tämän projektin malli on otettu “Getting started with Arduino Yún and Autobahn” opetusprojektista. Autobahn käyttää avoimeen lähdekoodiin perustuvia WebSocket- ja WAMP -protokollia. WebSocket mahdollistaa kaksisuuntaisen reaaliaikaisen viestinnän Webissä. (Tavendo, 2014)

7.3 WebSocket

WebSocket-protokolla on standardoitu vuonna 2011. WebSocket-protokolla tarjoaa jatkuvan kaksisuuntaisen yhteyden palvelinohjelman ja asiakasohjelman välillä yhden TCP-verkkopistokkeen yli. Yhteyden avaa aina asiakasohjelman, se lähettää kätelypyynnön palvelimelle ja samalla varmistetaan, että molemmat tukevat WebSocket-protokollaa. Kun yhteys on avattu, kumpikin voi lähettää viestin koska tahansa. Tämä tekee viestinnästä lähes reaaliaikaista palvelimen ja asiakasohjelman välillä. Yhteyden sulkemisviestin voi lähettää, joko asiakas- tai palvelinohjelma. WebSocket yhteyden yli lähetetty data voi olla UTF-8-koodatussa tekstimuodossa tai binäärimuodossa. WebSocket-protokolla toimii TCP-protokollan päällä, se käyttää TCP-protokollaa kuljetustasossa. WebSocket-protokolla jakaa datan erillisiin paketteihin. Se yhdistää kuljetus tasossa UDT- ja TCP -protokollia. Yhteensopivuus verkkoinfrastruktuurin kanssa on huomioitu siten, että Websocket-yhteys aloitetaan HTTP-kyselynä ja sen oletusportit ovat samat kuin HTTP-protokollalla 80 ja 443. Tämä hyödyttää WebSocket-protokollan käyttöönottoa, koska näillä porteilla on erikoisasema nykyisissä käyttöjärjestelmissä ja palomuuureissa. Säästytään näin ylimääräisten porttien avaamiselta verkkojen palomuuureihin WebSocket-protokollalle. (Tampereen yliopisto, 2014)



Kuva 25. WebSocket yhteyden aloitus. (Slideshare)

7.4 Kontrollerin (ATmega32u4) ohjelma

Arduinon ohjelma kirjoitetaan netistä ladattavalla ohjelmointiohjelmalla, tässä työssä versio on Arduino 1.5.8. Tiedonsiirtoon prosessorilta kontrollerille ja kontrollerilta prosessorille on oma funktio. Lukufunktio lukee aina, kun tieto on saatavilla prosessorilta. Lähetyksfunktio lähettää tietoa prosessorille aina, kun sitä kutsutaan. Mittaus-tulos lähetetään aina, kun anturilta tuleva mittaustulos muuttuu tai kymmenen sekun-nin välein, jos muutosta ei tule.

Lämpöanturille(DS20B18), kosteusanturille(AM2301) ja virtamuuntimelle(SCT-013-000) löytyy ohjelmaesimerkkejä netistä, joista muokkaamalla saatiin ne sovel-lukseen sopiviksi. Ohjelmiin on ladattava mukaan myös kirjastot, jotka tallennetaan ohjelmointiohjelman Libraries-kansioon.



```

Dallas | Arduino 1.5.8
Tiedosto Muokkaa Sketsi Työkalut Apua
Dallas
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 2

// Setup a oneWire instance to communicate with any OneWire devices
// (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup(void)
{
  // start serial port
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");

  // Start up the library
  sensors.begin();
}

void loop(void)
{
  // call sensors.requestTemperatures() to issue a global temperature
  // request to all devices on the bus
  Serial.print(" Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");

  Serial.print("Temperature for Device 1 is: ");
  Serial.print(sensors.getTempCByIndex(0)); // Why "byIndex"?
  // You can have more than one IC on the same bus.
  // 0 refers to the first IC on the wire
}
36 Arduino Yun on 87.92.184.110

```

Kuva 26. Arduinon ohjelmointiympäristö.

Tämä funktio lukee prosessorilta tulevat tiedot merkki kerrallaan ja erottaa ne ”=” merkin kohdalta ja tallettaa ne id ja value tauluun.

```

    }
    // Tietojen lukeminen prosessorilta.

bool readWebMessage(String *idStr, String *valueStr){
    if (port->available()) {
        int index;
        char character, id[40], value[30];
        bool idRead=false;

        index=0;
        while (port->available() > 0){
            char character = port->read();
            delay(2);
            if (idRead == true){
                value[index++] = character;
                value[index] = 0;
            }
            if (character == '='){
                idRead = true;
                index = 0;
            }
            if (idRead == false){
                id[index++] = character;
                id[index] = 0;
            }
        }
        *idStr = String(id);
        *valueStr = String(value);
        return true;
    }
    return false;
}

```

Tämä funktio lähettää tiedot prosessorille.

```

//Tietojen lähetys prosessorille.
void sendWebMessage(String id, String data){
    port->print(id);
    port->print('\t');
    port->print(data);
    port->println();

}

```


Funktio mgValve ohjaa magneettiventtiiliä ja samalla myös kompressoria. Magneettiventtiilin sulkeuduttua paine laskee kylmäkoneen imupuolella, kun se on laskenut säädetyin arvon alapuolelle, matalapainekeytkin katkaisee ohjauksen kompressorilta.

```

// Venttiilin ohjaus
bool mgValve(){
  bool viive;
  int inA = dryer;
  int inB = pressLH;

  // Yläraja lämpö
  if (templVal >= templSetHigh){
    high = false;
  }
  if (templVal <= (templSetHigh - templHyster)){
    high = true;
  }
  // Alaraja lämpö
  if (templVal <= templSetLow){
    low = false;
  }
  if (templVal >= (templSetLow + templHyster)){
    low = true;
  }
  // Kosteus raja
  if (humidityVal >= humiditySet){
    humidA = true;
  }
  if (humidityVal <= humiditySet - humidityHyster){
    humidA = false;
  }

  if (inA == HIGH && high == true && low == true && humidA == true &&
  alarmKomp() == false && inB == HIGH){
    digitalWrite(o_mgValve, HIGH);
    lastTimeMg = millis();
    viive = true;
  }
  else{
    digitalWrite(o_mgValve, LOW);
    if ((millis() - lastTimeMg) > delayTimeMg){
      viive = false;
    }
  }
  return viive;
}

```

7.5 Prosessorin(Atheros AR9331) ohjelma

Prosessoriin ohjelma on kirjoitettu Python ohjelmointikielellä. Sovelluksen on tarkoitus käyttää sarjaliikenneyhteyttä kontrollerin ja prosessorin välillä. Oletuksena yhteys on Linuxin käytössä, joten tämä oletus on poistettava. Muutetaan tämä käsky kommentiksi(# ttyATH0::askfirst:/bin/ash --login) tiedostosta /etc/inittab. Tämän jälkeen emme voi enää käyttää valmiita bridge-kirjastoja Arduinossa. Tällä käskyllä käynnistetään tämän sovelluksen oma bridge (python serial2ws.py --port/dev /ttyATH0), kun tämä käsky kirjoitetaan tiedostoon /etc/rc.local (cd/root/usr/ bin/python/serial2ws.py --port /dev/ttyATHO), se käynnistää automaattisesti portin Arduinoon, kun asiakas avaa yhteyden palvelimeen. Ohjelma välittää tilatietoja kontrollerilta selaimelle ja selaimelta käyttäjän käskyjä kontrollerille, lähinnä asetusarvoja. Seuraavassa ohjelman kaksi funktiota, jotka välittävät tietoa.

Tämä funktio välittää tietoa selaimelta kontrollerille.

```
def controlFromClient(self, id, value):
    """
    This method is exported as RPC and can be called by connected clients
    """
    msg = "%s=%s" % (id, value)
    """
    Tässä selaimelta tuleva tieto talletetaan muuttujaan msg.
    Id ja value erotetaan "-" merkillä.
    """

    if self.debug:
        print("Serial TX: {0}".format(msg))
    self.transport.write(str(msg))
```

Tämä funktio välittää tietoa kontrollerilta selaimelle.

```
def lineReceived(self, line):
    if self.debug:
        print("Serial RX: {}".format(line))

    try:
        ## parse data received from MCU
        ##
        data = [x for x in line.split('\t')]

        """
        Tässä kontrollerilta tuleva tieto sijoitetaan
        taulukkoon data. Id ja value erotetaan '\t' kohdalta.
        """

    except ValueError:
        print('Unable to parse value {}'.format(line))
    else:
        ## create payload for WAMP event
        ##
        payload = {'id': data[0], 'value': data[1]}
        print(data[0])
        print(data[1])
        ## publish WAMP event to all subscribers on topic
        ##
        self.session.publish("com.myapp.mcu.onvaluechange", payload)
```

7.6 Käyttöliittymän suunnittelu

7.6.1 HTML5

HTML-kielen uusin versio on HTML5. Ensimmäinen HTML5-luonnos julkaistiin vuonna 2008. Yksi tärkeimmistä ominaisuuksista on alustariippumattomuus. Uudessa versiossa on monia uusia ominaisuuksia, esimerkiksi piirtäminen webissä, joka käytännössä puuttuu edellisestä versiosta. Seuraavassa luvussa on muutama esimerkki HTML5-kielen uusista ominaisuuksista (Korpela, 2011).

7.6.2 Uudet ominaisuudet

Canvas-elementti on yksi isoista uudistuksista HTML5:ssä, se tarjoaa piirtoalustan graafisille esityksille. HTML5 sivun määrittely on lyhentynyt vanhasta merkittävästi, seuraavassa esimerkki. (Korpela, 2011)

```

1 <!DOCTYPE html PUBLIC
2 "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
7 <title>Pikku HTML4-sivu</title>
8 </head>
9 <body>
10 <p>Heippa!</p>
11 </body>
12 </html>






1 <!DOCTYPE html>
2 <html lang="fi">
3 <head>
4 <meta charset="utf-8" />
5 <title>Pikku HTML5-sivu</title>
6 </head>
7 <body>
8 <p>Heippa!</p>
9 </body>
10 </html>

```

Kuva 27. HTML sivun määrittelyt vanha ja uusi.

7.6.3 Selaimenohjelma

Selain ohjelmaa kirjoittaessa täytyy huomioida, että selaimet eivät tue vielä kaikkia HTML5 tageja. Tämän sovelluksen kosteusmittari ja lämpömittarit on toteutettu meter tagilla, jota Explorer11 ei tue toistaiseksi, siksi mittaustulos näytetään myös erillisellä input ikkunalla. W3schools.com sivuilta löytyy, mitä tageja eri selaimet tukevat tällä hetkellä. (W3Schools)

Element					
<meter>	8.0	Not supported	6.0	6.0	11.0

Kuva 28. Meter-tagin tuki eri selaimilla. (W3Schools)

Selain ohjelmaan tehtiin salasanasuojattu linkki asetusarvojen säätösivulle. Salasana lähetetään selaimelta kontrolleriin, jossa tehdään vertailu ja se lähettää takaisin selaimelle hyväksytty tai hylätty koodin. Selainohjelmassa on JavaScript funktio, joka avaa sivun, kun se saa hyväksytty koodin.



Kuva 29. Käyttöliittymä.

Tämä funktio ottaa vastaan tietoa selaimelle.

```
function onValue(args) {
var payload = args[0];
var element = document.getElementById(payload.id);
if (element !== null) {
element.innerHTML = payload.value;
if (payload.id == "status") {
status.innerHTML = payload.value;
}
}
```

Tämä funktio lähettää tietoa selaimelta.

```
function toArduino(id, value) {
session.call("com.myapp.mcu.controlfromclient", [ id, value
]);
}
```

Input lämpötila näyttö.

```
<p>Lämpötila kaappi</p>
<input id="temp1" size="5" type="text">°C
```

Meter-tagilla tehty lämpömittari.

```
<meter style="width: 300px" min="0" max="50" low="25"
high="45"
optimum="10"
<span id="temp1">-</span>
</meter>
<ul id="scal">
<li style="width: 5%"><span></span></li>
<li><span id="scal">5</span></li>
<li><span id="scal">10</span></li>
<li><span id="scal">15</span></li>
<li><span id="scal">20</span></li>
<li><span id="scal">25</span></li>
<li><span id="scal">30</span></li>
<li><span id="scal">35</span></li>
<li><span id="scal">40</span></li>
<li><span id="scal">45</span></li>
<li style="width: 5%"><span id="scale"></span></li>
</ul>
```

8 LIITYNTÄ TEHTAAN VERKKOON

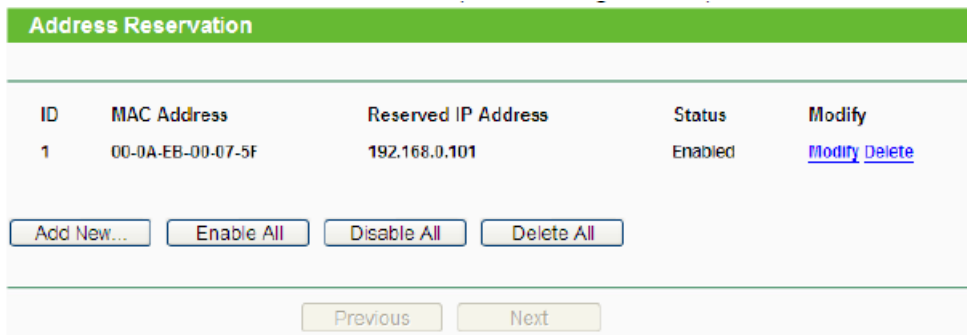
Tehtaan langatonta verkkoa jouduttiin laajentamaan. Välimatka ei ole kovin pitkä, mutta tuotantohalli ja kuivaamo on rakennettu peltielementeistä, jotka heikentävät signaalia. Signaalin vahvistus tehtiin kahdella TP-LINK tukiasemalla toinen tuotantohallin katolle, joka on samassa rakennuksessa toimiston kanssa ja toinen kuivaamoon, joka on kahdenkymmenen metrin päässä hallista. Välikatolla oleva reititin yhdistettiin toimiston reitittimeen Cat6 kaapelilla, josta jatkettiin langattomasti kuivaamon reitittimelle. Välikaton TP-LINK asetettiin ”Access point” toimintatilaan, tässä tilassa reititin muuttaa toimistossa olevan kiinteän verkon langattomaksi verkoksi. Kuivaamon TP-LINK asetettiin ”Repeater” tilaan jolloin se vain vahvistaa signaalia.

Reitittimellä varataan Arduinolle kiinteä IP-osoite, kun tämä pyytää IP-osoitetta reititin antaa aina saman osoitteen. Arduinolla on oltava kiinteä IP-osoite. Reitittimen palomuriin on avattava portit, jotka linkitämme Arduinoon, jotta voidaan ottaa yhteys sisäverkon ulkopuolelta.

The screenshot shows the 'NAT -- Virtual Servers Setup' page on a TP-LINK 108M Wireless ADSL2+ Router (Model No. TD-W8920G). The page includes a sidebar with navigation options: Device Info, Quick Setup, Advanced Setup, WAN, LAN, MAC Clone, NAT (selected), Virtual Servers, Port Triggering, DMZ Host, ALG, Security, Routing, DNS, DSL, Port Mapping, Wireless, Diagnostics, and Management. The main content area has a title 'NAT -- Virtual Servers Setup' and a description: 'Virtual Server allows you to direct incoming traffic from WAN side (identified by Protocol and External port) to the Internal server with private IP address on the LAN side. The Internal port is required only if the external port needs to be converted to a different port number used by the server on the LAN side. A maximum 32 entries can be configured.' Below the description are 'Add' and 'Remove' buttons. A table lists three virtual server entries:

Server Name	External Port Start	External Port End	Protocol	Internal Port Start	Internal Port End	Server IP Address	Remove
			TCP/UDP	8000	8000	192.168.100.45	<input type="checkbox"/>
			TCP/UDP	8000	8000	192.168.100.45	<input type="checkbox"/>
			TCP/UDP	8080	8080	192.168.100.45	<input type="checkbox"/>

Kuva 30. Porttien linkitys Arduinoon.



ID	MAC Address	Reserved IP Address	Status	Modify
1	00-0A-EB-00-07-5F	192.168.0.101	Enabled	Modify Delete

Buttons: Add New..., Enable All, Disable All, Delete All

Navigation: Previous, Next

Kuva 30. Kiinteän IP-osoitteen varaus tietylle MAC-osoitteelle reitittimessä.

Tehtaan langattoman verkon liittymässä on vaihtuva IP-osoite, tämä vaatii toimenpiteitä, jotta voimme löytää Arduinon sisäverkon ulkopuolelta. Yksi vaihtoehtoa on vaihtaa liittymän IP-osoite kiinteäksi, mutta tämä nostaa liittymän hintaa. Halvempi ratkaisu tähän on ilmainen dynaaminen DNS palvelu, tähän valittiin suomalainen dy.fi palvelu. Palvelu antaa käyttöön domain-nimen, joka osoitetaan dynaamiseen IP-osoitteeseen. Palvelun idea on siinä, että kun koneellasi käydään osoittamassa voimassa oleva IP-osoite palveluntarjoajan palvelimelle, niin se pystyy ohjaamaan sen kautta tulevan tiedon oikeaan osoitteeseen. Päivityksen voi tehdä dy.fi sivuilla manuaalisesti tai tehdä automaattisen ohjelman jollekin koneelle, joka on tässä verkossa. Esimerkiohjelma löytyy dy.fi sivulta.

9 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli suunnitella ohjaus puutavarakuivaamoon. Lähtökohtana oli, että se toteutetaan mikrokontrollerilla ja siihen liitetään selainpohjainen käyttöliittymä ja selvittää miten tämä sovellus sopii teollisuusympäristöön.

Ensimmäinen tavoite oli ohjata mikrokontrollerilla kuivaamon laitteita. Mikrokontrolleri sopeutui suhteellisen hyvin tähän sovellukseen. Ohjauksen suunnittelussa täytyi ottaa huomioon se, että Arduino on sähköisesti hyvin herkkä.

Toinen tavoite oli selainpohjainen käyttöliittymä ohjaukselle. Käyttöliittymästä tuli toimiva, selaimesta voi seurata lämpötilan, kosteuden ja virran mittauksen oloarvoja, sekä I/O-tietoja. Salanasuojatulta linkkisivulta voi säätää ohjauksen asetusarvoja.

Kuivaamossa on kuivattu nyt kaksi kaapillista puutavaraa. Ensimmäisen kuivauserän aikana tuli esille pieniä ongelmia. Kuivauksen aikana oli sähkökatkoja ja nämä aiheuttivat kontrollerin jumittumisen. Tämä ongelma korjattiin pienellä UPS:lla. Toinen ongelma oli kosteusanturin rikkoontuminen. Anturin rikkoutumisen syy oli todennäköisesti valmistusvika. Anturin vaihdon jälkeen, ongelmia ei ole enää ollut.

Sovelluksen kehitysmahdollisuudet kuivaamon ohjauksessa: mittaustietoja voisi tallentaa prosessorille ja kuivaamon ohjaukseen voisi tehdä reseptejä eri puutavaralajien kuivaukseen.

Tätä sovellusta on mahdollisuus pienillä muokkauksilla käyttää moneen eri käyttötarkoitukseen. Yksi tämän sovelluksen hyvä puoli on, että se käyttää WebSocket-protokollaa, joka tekee tiedonsiirrossa lähes reaaliaikaista. Voidaan ohjata nopeutta vaativia sovelluksia.

LÄHTEET

Aosong Electronics. (ei pvm). Haettu 4. 6 2015 osoitteesta
<http://akizukidenshi.com/download/ds/aosong/AM2302.pdf>

Arduino. (ei pvm). Haettu 21. 2 2015 osoitteesta
<http://arduino.cc/en/Main/ArduinoBoardYun?from=Products.ArduinoYUN>

Hietalahti. (2013). *Teollisuuden sähkökäytöt*. Vantaa: Hansaprint Oy.

Honkanen. (ei pvm). *Gallia.kajak*. Haettu 12. 2 2015 osoitteesta
http://gallia.kajak.fi/opmateriaalit/yleinen/honHar/ma/ELE_Piirien_v%C3%A4liset_ohjausv%C3%A4ly%C3%A4t.pdf

HQEW. (ei pvm). Haettu 4. 6 2015 osoitteesta http://circuit-diagram.hqew.net/Arduino-and-DS18B20---1-wire-digital-thermometer-_12639.html

Kaappola, H. J. (2011). *Kylmätekniikan perusteet*. Helsinki: Juvenesprint Oy.

Korpela. (2011). *HTML uudet ominaisuudet*. Porvoo: BookWell Oy.

MikroElektroniikka. (ei pvm). Haettu 7. 2 2015 osoitteesta
<http://www.mikroe.com/chapters/view/74/pic-basic-book-chapter-1-world-of-microcontrollers/>

Nic. (ei pvm). Haettu 12. 2 2015 osoitteesta <http://www.nic.fi/~skarna/etusivu.html>

Nikkilä, V. (ei pvm). *Noppa.aalto*. Haettu 15. 3 2015 osoitteesta
https://www.google.fi/?gws_rd=ssl#q=AS-74.3135+Seminaariesitys

Openenergymonitor. (ei pvm). Haettu 27. 2 2015 osoitteesta
<http://openenergymonitor.org/emon/buildingblocks/report-yhdc-sct-013-000-current-transformer>

Pulli, T. (ei pvm). */noppa.aalto.fi/noppa/kurssi*. Haettu 14. 2 2015 osoitteesta
https://noppa.aalto.fi/noppa/kurssi/elec.../ELEC-A4010_anturiharjoitus.pdf

PuuProffa. (ei pvm). *PuuProffa*. Haettu 14. 2 2015 osoitteesta
<http://www.puuproffa.fi/>

Sensors online. (ei pvm). Haettu 14. 2 2015 osoitteesta
<http://www.sensormag.com/sensors/humidity-moisture/choosing-a-humidity-sensor-a-review-three-technologies-840>

SFS, S. S. (27. 11 2006). *SFS EN 61082-1*. Haettu 28. 2 2015

SFS, S. S. (ei pvm). *SFS käsikirja 154 Jakokeskukset*. Haettu 1. 3 2015

SFS, S. S. (ei pvm). *SFS_EN 60204-1*. Haettu 1. 3 2015

Slideshare. (ei pvm). Haettu 20. 2 2015 osoitteesta
<http://www.slideshare.net/Sovelto/palvelujen-ohjelmointi>

Tampereen yliopisto. (6 2014). Haettu 9. 2 2015 osoitteesta
<https://tampub.uta.fi/bitstream/handle/10024/95854/gradu07209.pdf?sequence=3>

Tavendo. (10. 10 2014). Haettu 8. 2 2015 osoitteesta
<http://tavendo.com/blog/post/arduino-yun-with-autobahn/>

Teknillinen, L. t. (ei pvm). *Doria*. Haettu 12. 11 2013 osoitteesta
<http://www.doria.fi/bitstream/handle/10024/70843/nbnfife201109132386.pdf?sequence=3>

Vahtera, P. (2015). *Mikro-ohjaimen ohjelmointi*. Helsinki: WS Bookwell Oy.

Vaisala. (ei pvm). *www.vaisala.fi*. Haettu 15. 2 2015 osoitteesta
<http://www.vaisala.fi/Vaisala%20Documents/Technology%20Descriptions/HUMICAP-Technology-description-B210781FI-C.pdf>

W3Schools. (ei pvm). Haettu 23. 2 2015 osoitteesta
http://www.w3schools.com/tags/tag_meter.asp

Wikimedia commons. (ei pvm). Haettu 18. 5 2015 osoitteesta
http://commons.wikimedia.org/wiki/File:Hall_effect.png

Wikipedia. (ei pvm). Haettu 14. 2 2015 osoitteesta
<http://en.wikipedia.org/wiki/Humidity>

LIITTEET

- LIITE 1 Lämpötila anturin (DS18B20) esimerkki ohjelma
- LIITE 2 Virtamuuntimen(SCT-013-000) esimerkki ohjelma.
- LIITE 3 Kosteusanturin(AM2301) esimerkki ohjelma.
- LIITE 4 Piirikaavio(OK_KV2)

Lämpötila anturin (DS18B20) esimerkki ohjelma.

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS 2

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup(void)
{
  // start serial port
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");

  // Start up the library
  sensors.begin();
}

void loop(void)
{
  // call sensors.requestTemperatures() to issue a global temperature
  // request to all devices on the bus
  Serial.print("Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");

  Serial.print("Temperature for the device 1 (index 0) is: ");
  Serial.println(sensors.getTempCByIndex(0));
}
```

Virtamuuntimen(SCT-013-000) esimerkki ohjelma.

```
// EmonLibrary examples openenergymonitor.org, Licence GNU GPL V3

#include "EmonLib.h"           // Include Emon Library
EnergyMonitor emonl;         // Create an instance

void setup()
{
  Serial.begin(9600);

  emonl.current(A1, 60.2);    // Current: input pin, calibration.
}

void loop()
{
  double Irms = emonl.calcIrms(1480); // Calculate Irms only

  Serial.print(Irms*230.0/1000);
  Serial.println("Kw");      // Apparent power
  Serial.print(" ");
  Serial.println(Irms);      // Irms
  delay(3);
}
```

Kosteusanturin(AM2301) esimerkki ohjelma.

```

#include "DHT.h"
#define DHTPIN 2      // what pin we're connected to

// #define DHTTYPE DHT11    // DHT 11
#define DHTTYPE DHT22    // DHT 22  (AM2302)
// #define DHTTYPE DHT21    // DHT 21  (AM2301)
// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  dht.begin();
}
void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Compute heat index
  // Must send in temp in Fahrenheit!
  float hi = dht.computeHeatIndex(f, h);

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");

```
