ATX TUULETINOHJAIN MIKROKONTROLLERILLA

HAMK
HÄMEEN AMMATTIKORKEAKOULU

Ammattikorkeakoulun opinnäytetyö

Kone- ja tuotantotekniikka

Riihimäki, 5.11.2012

Nika Korpi

HAMK
HÄMEEN AMMATTIKORKEAKOULU

TIIVISTELMÄ

Riihimäki
Kone- ja Tuotantotekniikka
Mekatroniikka

Tekijä Nika Korpi **Vuosi** 2012

Työn nimi ATX tuuletinohjain mikrokontrollerilla

TIIVISTELMÄ

Lähdin rakentamaan itselleni tuuletinohjainta tietokoneeseen. Olin jo aiemmin miettinyt tuuletinohjaimen rakentamista ja hieman miettinyt ohjaimen ominaisuuksia etukäteen. Minulla oli tietoa ohjaimen toteutukseen etukäteen vain vähän. Ohjaimen rakentaminen oli minulle pikemminkin mikrokontrollerien, ohjelmoinnin ja elektroniikkasuunnittelun oppimisprojekti.

Työssä oli tarkoitus käyttää mikrokontrolleria jolla ohjattaisiin tuulettimien nopeutta. Tuulettimien nopeutta säädettäisiin lämpötilan perusteella tai manuaalisesti. Tavoitteena oli saada tietokoneeseen asennettava yksikkö jolla voisi ohjata tietokoneen kotelotuulettimia.

Työssä lähdin selvittämään, miten tietokoneen tuulettimia voi ohjata ja miten niiden kierrosnopeusanturia voi lukea. Valitsin mikrokontrolleriksi atmelin ATmega16. Rakensin ohjaimen ensin koekytkentäalustalle siten että ohjelmoin mikrokontrolleria sitä mukaan kuin rakensin kytkentöjä. Kun olin tyytyväinen koekytkentöihin ja ohjelmaan, aloin suunnitella ja rakentaa itse ohjaimen.

Lopputuloksena oli ohjain jossa on tuuletinpaikkoja 8, lämpöantureita 4 ja aakkosnumeerinen LCD-näyttö. Tuulettimien nopeutta säätäviä linjoja on 4 kappaletta joihin jokaiseen saa kytkettyä 2 tuuletinta. Tuuletinlinjoja pystyy säätämään manuaalisesti sekä lämpötilan perusteella. Jokaisen tuulettimen kierrosnopeutta luetaan ja se on esitetty LCD näytöllä.

Lopputulokseen olen melko tyytyväinen. Opin todella paljon uutta mikä oli työssä tarkoituskin. Vanhemman tyyppisien tuulettimien ohjaus jäi toteuttamatta, koska suunnittelemani kytkentä ei toiminutkaan toivotulla tavalla. Ohjaimen puutteisiin pääsee selville vasta kun ohjain on ollut jonkin aikaa käytössä. Suunnittelin ohjaimen siten että ohjelman pystyy päivittämään, jos ohjelmassa ilmenee puutteita.

Avainsanat Mikrokontrolleri, tuuletin, tietokone, ohjaus, PWM

Sivut 26 s. + liitteet 40 s.

HAMK
UNIVERSITY OF APPLIED SCIENCES

Riihimäki
Mechanical Engineering and Production Technology
Mechatronics

Author Nika Korpi **Year** 2012

Subject of Bachelor's thesis ATX fan controller with a microcontroller

ABSTRACT

The author of this thesis wanted to build a fan controller for a computer himself and. had already been thinking about the controller features beforehand but lacked the actual knowhow for doing it. Building the fan controller was more of a project for learning about microcontrollers, programming and electrical designing

The work was to include a microcontroller which would control the fan speed. The fan speed would be controlled by the temperature or manually. The goal was to build a unit that could be installed into a computer which would control the case fans.

The work was started by researching how computer fans can be controlled and how their speed sensors work. An Atmel's ATmega16 microcontroller was chosen. Initially, the controller was built on a breadboard, while the wiring and programing were developed simultaneously. As soon as the wiring and programing were satisfactory, then the controller was designed and built.

As a result a controller which had eight fan spaces, four temperature sensors and an alphanumerical LCD display was developed. There were four lanes for the fan control and every line was connected to two fans. The fan lanes could be adjusted manually or on the basis of temperature. The fan speed was read from all the fans and displayed on the LCD

The author learned a lot and was satisfied with the result. The older type of fan control was left out because the circuit that was designed did not function as hoped. The deficiencies of the controller could only be seen after it had been in use for some time. If deficiencies occur in the software, then the controller was designed so that it can be updated.

Keywords Microcontroller, fan, computer, controller, PWM

Pages 26 p. + appendices 40 p.

SISÄLLYS

1 JOHDANTO

Työn tavoitteena on rakentaa tietokoneeseen tuuletinohjain mikrokontrollerilla. Ohjaimella säädetään tuulettimien kierrosnopeutta joko käsisäätöisesti tai lämpötilan perusteella. Ohjaimeen myös liitetään näyttö jossa voidaan näyttää lämpötila-anturien arvot, tuulettimien kierrosnopeuden ja tuulettimille tuleva teho.

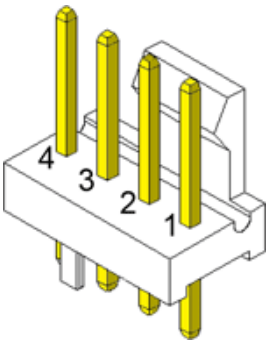
Tuuletinohjaimeen on tarkoitus tehdä useampi säädettävä lähtö tuulettimille jotta eri tuulettimia pystyy säätämään toisistaan riippumatta. Lämpötila-antureita tulee olemaan useampi, jotta pystytään seuraamaan kotelon sisältä eri lämpötiloja kuten sisääntulevan ja ulosmenevän ilman lämpötila tai asettaa antureita kiinni tai hyvin lähelle eri komponentteja kuten prosessori.

Vastaavanlaisia tuuletinohjaimia on paljonkin markkinoilla, joten en lähtenyt tekemään tuuletinohjainta tarjonnan puutteen vuoksi tai niiden puutteellisien ominaisuuksien vuoksi. Lähdin tekemään tuuletinohjainta koska olin aiemmin miettinyt tuuletinohjaimen rakentamista ja ajattelin sen toimivan hyvänä alustana päästä elektroniikkasuunnitteluun ja ohjelmointiin käsiksi. Lähtökohtaisesti minulla ei oikein ollut mitään hajua mitä lähdin tekemään joten pelkästään uuden oppiminen oli suuri motivaatio tehdä työtä.

2 ATX-TUULETIN

Tietokonetuulettimilla on kaksi erilaista liitintä, 3-johtimen ja 4-johtimen liittimet. Liittimet ovat myös yhteensopivia keskenään.

Kuva 1. 3-johtimen tuuletinliitin (3 Pin Fan Connector Pinout. 2008.)

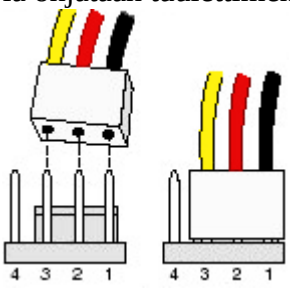


Kuva 2. 4-johtimen tuuletinliitin (4 Pin Fan Connector Pinout. 2010.)

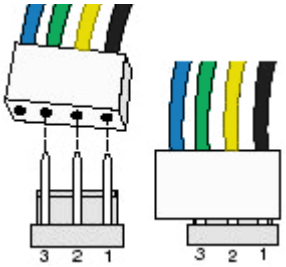
Taulukko 1. Tuuletinliittimien johtimet. (3 Pin Fan Connector Pinout. 2008)

Johdin	4 johtimen	3 johtimen
1	GND (0 V)	GND (0 V)
2	+12 V	+12 V
3	Kierrosanturi	Kierrosanturi
4	PWM	N/A

Liittimet ovat periaatteessa samoja mutta 4 johtimen liittimeen on tuotu yksi johdin lisää, johon voidaan tuoda PWM [katso kohta 2.2] signaali jolla ohjataan tuulettimen kierrosnopeutta.



Kuva 3. 3 johtimen tuuletin 4 johtimen liittimessä (4 Pin Fan Connector Pinout. 2010)



Kuva 4. 4 johtimen tuuletin 3 johtimen liittimessä (4 Pin Fan Connector Pinout. 2010)

Liittimet toimivat keskenään, mutta käyttämällä 4-johtimen tuuletinta 3-johtimen liittimessä tuulettimen nopeudensäätö ei toimi. 3-johtimen tuuletin 4-johtimen liittimessä toimii aivan kuten 3-johtimen liittimessäkin. (4 Pin Fan Connector Pinout. 2010.)

2.1 Kierrosanturi

Tuulettimien kierrosanturi on maadoittava anturi joka tuottaa kaksi pulssia kierroksella. Tuulettimen kierrosanturi toimii katkasijana jolloin kierrosanturin johdin on joko oikosulussa maajohtimen kanssa tai se on poikki

Taulukko 2. Tuulettimen kierrosanturin toiminta

Tuulettimen kierroksen osa	Kierrosanturin tila
¼	Oikosulussa
¼	Poikki
¼	Oikosulussa
¼	Poikki

Tuulettimen kierrosanturilla voidaan siis tehdä kanttiaaltoa jonka taajuuden puolikas on tuulettimen kierrosnopeus. (4-Wire Pulse Width Modulation (PWM) Controlled Fans. 2004.)

2.2 PWM ja sen käyttö tasavirtamoottorin ohjauksessa.

PWM eli pulssinleveysmodulaatio signaali on tehollisesti säädettävä signaali. Signaalin tilaa 1 (päällä) suhteessa tilaan 0 voidaan säätää. Esimerkiksi 50% työsuhteella signaali on 50% kellojaksosta tilassa 1 ja 50% tilassa 0 kun taas 10% työsuhteella signaali on 10% kellojaksosta tilassa 1 ja 90% tilassa 0. Vahvistamalla signaali DC moottorille voi moottoria tehoa säätää PWM signaalilla

PWM ohjattuun moottorin rinnalle on syytä asettaa diodi estosuuntaisesti, koska moottori tuottaa katkaisujen yhteydessä induktio piikkejä mikä saattaa hajottaa komponentteja. (How to Control Motor Speed with a PWM Circuit.)

3 MIKROKONTROLLERIN VALINTA

Vaatimuksena mikrokontrollerille olisi useampi PWM linja, useampi AD-muunnin sekä riittävä määrä I/O-portteja. Mikrokontrollerin valintaan vaikuttaa myös kontrollerin saatavuus, koska valmistajilla on hyvinkin erikoisia mikrokontrollereita, mutta niiden saatavuus on erittäin huono. Ennen kontrollerin lopullista valitsemista täytyi myös miettiä käyttöönottoa. Käyttöönotto vaatii ohjelmointilaitteen ajureineen sekä kääntäjän.

3.1 ATmega16

Mikrokontrolleriksi päädyin valitsemaan atmelin valmistaman ATmega16 mikrokontrollerin koska se oli helposti saatavilla ja sen ominaisuudet olivat alustavaan suunnitelmaan riittävät ja vähän ylikin.

Ominaisuuksia:

- 16KB Flash-muistia
- 512B EEPROM-muistia
- 1KB SRAM-muistia
- 4 PWM-linjaa (2 x 8bit ja 2 x 16bit)
- 32 I/O-porttia
- 8 AD-muunninta (10bit)
- suurin käyttötaajuus 16Mhz
- käyttöjännite 4.5V - 5.5V

ATmega16 on myös olemassa vähävirtainen ATmega16L malli, mutta sen maksimi kellotaajuus on 8Mhz. Vähävirtaisuus ei ollut työssä erityisen tärkeää, joten valitsin tehokkaamman version. ATmega16 saa kahdessa eri koteloinnissa, pintaliitos (TQFP) tai läpivienti (PDIP). Koekytkentöjä varten valitsin PDIP koteloinnin, lisäksi pintaliitoskomponenttien juottaminen piirilevylle käsin olisi melko hankalaa. (ATmega16. 2010.)

Ennen ostopäätöstä etsin alustavasti sopivia ohjelmointilaitteita ja ohjelmistoja. Ohjelmointilaiteita löytyi USB porttiin, sarjaporttiin ja rinnakkaisliittimeen. Päädyin USB ohjelmointilaitteeseen, koska tietokoneessani ei ollut sarjaporttia eikä rinnakkaisliitintä. Ohjelmistoja selailin sen verran ennen ostopäätöstä, että tarkistin, että ilmaisia ohjelmia on olemassa ja ohjeita niiden käyttöön löytyy.

4 MIKROKONTROLLERIN KÄYTTÖÖNOTTO

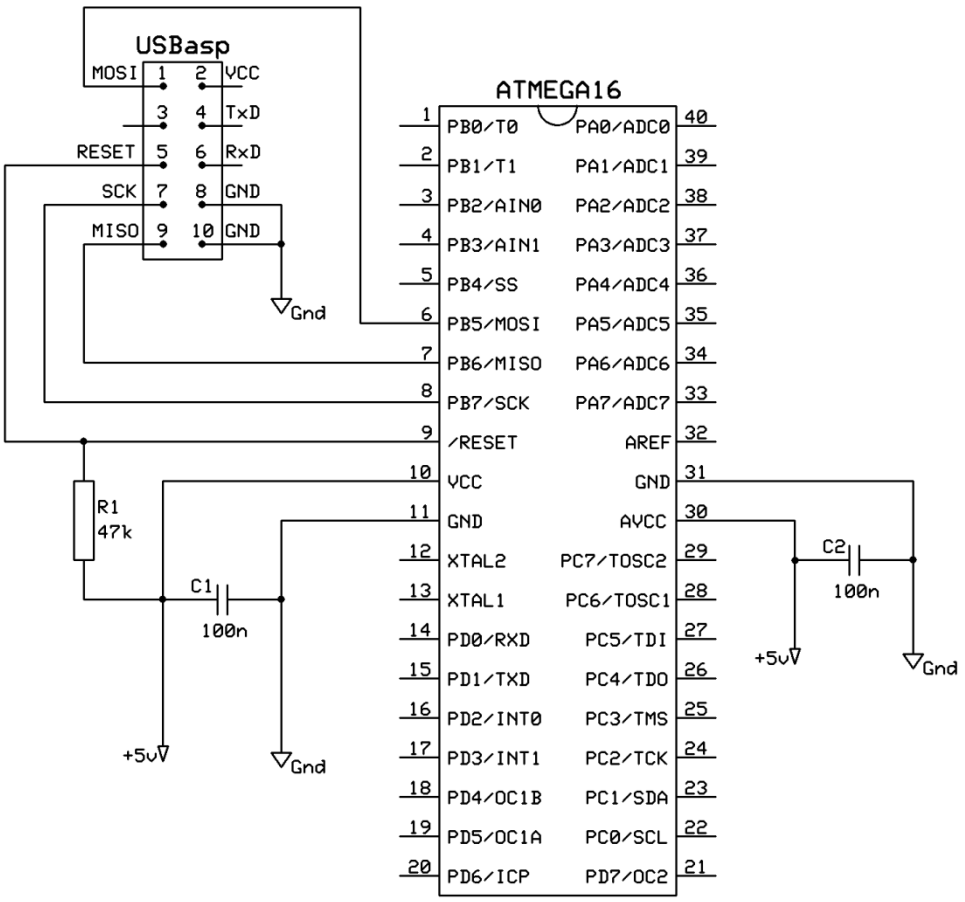
Mikrokontrollerin ohjelmointia varten tarvitsin ohjelmoijan ja päädyin USBasp ohjelmoijaan, joka on suhteellisen halpa ja harrastajien keskuudessa suosittu ohjelmoija. USBasp tarvitsi ohjelmointia varten ohjelman ja niitä oli useampi, mutta päädyin AVR eXtreme Burner ohjelmaan, koska se oli graaffinen. Ohjelmointia ja kääntämistä varten päätin käyttää AVR visual studio 5:ttä, joka on atmelin oma ohjelma.

4.1 USBasp

USBasp on harrastajien suosima USB väylään liitettävä ohjelmoija atmel avr mikrokontrollereille. USBasp on rakentelusarja, joten se täytyi omatoimisesti kasata. USBasp rakentaminen oli helppoa mukana tulleen ohjeen ja netistä löytyvän videon avulla. (Bausatz "USBasp".)

Kuva 5. USBASP (USBasp)

USBasp:ssa on kolme asetusjumpperia, JP1, JP2 ja JP3. JP1 aktivoi USBasp:n VCC pinnin, jolla voidaan tuoda virta ohjelmoitavaan mikrokontrolleriin. JP2 on itse ohjelmointi, jolloin toisella ohjelmoijalla voidaan päivittää USBasp:n firmware. JP3 asettaa hitaan ohjelmoinnin. (README file for USBasp. 2011.)



Kuva 6. Atmega16 liitäminen USBasp:hen.

Kuvasta näkee USBasp liittimen eri johtimet sekä kytkennän ATmega16 mikrokontrollerin kanssa. Johtimien RxD ja TxD käyttö ei ole välttämätöntä. RxD ja TxD johtimia käytettäessä kytkettäisi ne ATmega16:n vastaaviin pinneihin, eli RxD pinniin 14 ja TxD pinniin 15. Uutta mikrokontrolleria ohjelmoidessa pitäisi USBasp asettaa hitaan ohjelmoinnin tilaan, mikä onnistuu asettamalla jumpperi 3. (Burning hex files using USBasp and AVRdude. 2009.)

4.2 AVR eXtreme Burner

AVR eXtreme Burner on windows ohjelma USBasp:n käyttöä varten. Kääntäjän tuottama konekoodi täytyy siirtää mikrokontrollerille ja AVR eXtreme Burner toimii rajapintana kääntäjän ja USBasp:n välillä.

Ohjelma on yksinkertainen ja helppokäyttöinen. Ohjelma ei tarvitse muuta kuin yhden esiasetuksen ja se on mikrokontrollerin valinta, chip -> ATmega16. Ohjelma osaa lukea ja kirjoittaa mikrokontrollerista flash, EEPROM muistit ja asetusbitit eli sulakkeet. (eXtreme Burner – AVR. 2009.)

4.3 AVR Visual Studio 5

AVR Visual Studio 5 on atmelin sivuilta ladattava ohjelmointiympäristö AVR mikrokontrollereille (Studio Archive. 2012). Ohjelma tukee C\C++ ja assembler kieliä. Ohjelmalla voi kirjoittaa ja debugata koodia, sekä simuloida mikrokontrolleria.

Mikrokontrollerin simulointia en missään vaiheessa käyttänyt, koska ohjelman ajaminen mikrokontrollerille ja toiminnan seuraaminen oli minun käytössä riittävän tehokasta.

5 KOMPONENTIT JA KÄYTTÖÖNOTTO

Rakensin ohjainta ensin koekytkentäalustalle siten että komponentti kerrallaan tein kytkennät ja koodin. Tehtyäni kytkennät ja koodin ajoin ohjelman mikrokontrollerille ja katsoin, toimiko kytkentä ja koodi toivotulla tavalla. Muutamaan otteeseen tuli tehtyä todella pahoja kytkentävirheitä, mutta en kuitenkaan onnistunut hajottamaan mitään.

5.1 Virtalähde

Tuuletinohjain on tiekonekäyttöön joten virtalähteenä käytän tietokoneen omaa virtalähdettä. Tietokoneen molex lisälaitevirtaliitin on tuuletinohjaimeen sopiva, koska siinä on +12v ja +5v linjat. Virtaliitin itsessään on molexin valmistama 8981 liitin, mutta yleisesti tietokoneiden yhteydessä puhutaan molex virtaliittimestä. (Molex 4 pin Peripheral Power Connector Pinout.)

Molex 8981 kulmaliitin kestää 250VAC @ 6.5A. (Power connector, 4 circuits, vertical and right angle. 2010.) Tietokoneen kotelotuulettimien käyttämät virrat ovat yleensä 0.2A tai alle, mutta tehokkaampiakin löytyy. Suurin sallittu virta 4-johtimen tuulettimelle on 1A. (4-Wire Pulse Width Modulation (PWM) Controlled Fans. 2004.) 1A kuormalla on turvallista käyttää 6:tta tuuletinta. Ohjaimeen voisi kytkeä tuulettimia enemmän kuin 6, kun varmistaa ettei tuulettimien virrankulutus ylitä 6.5A.

5.2 Tuulettimet

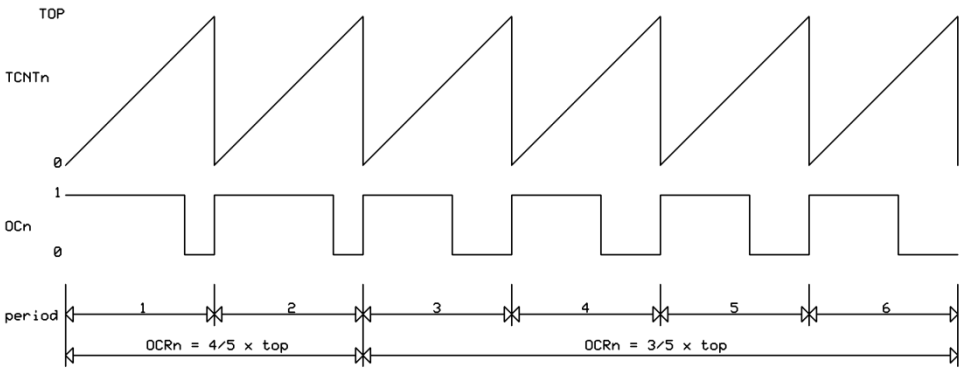
Tuulettimien lukumäärässä päädyin hyvin aikaisessa vaiheessa 8:aan siten että jokaiselle PWM-linjalle olisi 2 tuuletinta ja tuulettimien kierrosnopeutta voisi lukea esimerkiksi ATmega16:n C-porteilla.

5.2.1 Nopeuden säätö

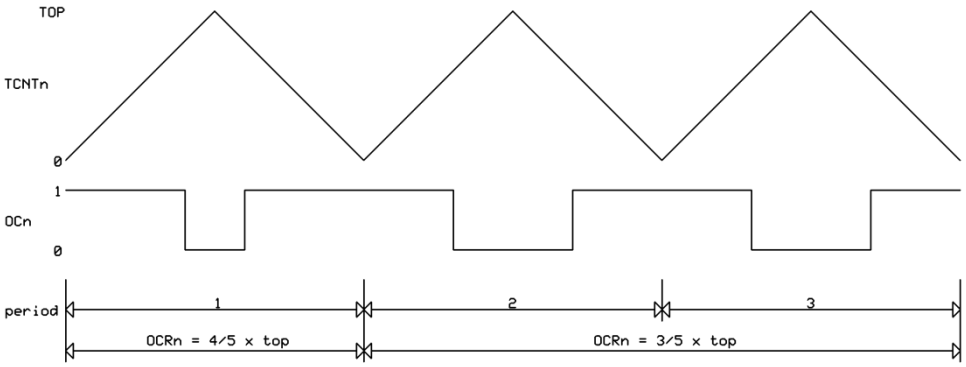
Ensimmäiset koekytkennät tein 4-johtimen tuulettimen kanssa, koska en tarvinnut erillistä laitteistoa tuulettimen ohjaamiseen. Tuuletinta ohjaava PWM-signaalin tavoitetaajuus on 25khz, mutta hyväksyttävät taajuudet ovat 21khz ja 28khz välillä. Maksimi virta PWM-linjalle on 5mA ja se on oikosulkuvirta joten etuvastuksen käyttö ei ole välttämätöntä vaikkakin it-

se käytin koekytkennöissä 1kΩ etuvastusta. (4-Wire Pulse Width Modulation (PWM) Controlled Fans. 2004.)

ATmega16 tuottaa PWM signaalin siten että TCNTn rekisterissä on juokseva luku ja OCRn rekisteriin ladataan luku. Komparaattori vertailee TCNTn rekisteriä ja OCRn rekisteriä keskenään. TCNTn:n arvon ollessa suurempi kuin OCRn:n OCn pinnin tila on nolla, jos taas TCNTn on pienempi kuin OCRn, OCn pinnin tila on yksi. OCn pinnin tila voidaan myös invertoida. TCNTn rekisterissä luku voi juosta kahdella eri tavalla, pohjasta huippuun [nopea PWM] tai pohjasta huippuun ja takaisin pohjaan [vaihekorjattu PWM].



Kuva 7. Nopea PWM



Kuva 8. Vaihekorjattu PWM

Kaava 1. PWM signaalin taajuus nopealle PWM:lle

$$f_{OCnPCPWM} = \frac{f_{clk\_I/O}}{N*256}$$

$f_{OCnPCPWM} = PWM\ signaalin\ taajuus$
$f_{clk\_I/O} = Mikrokontrollerin\ kellotaajuus$
$N = Esiajastin$

Kaava 2. Kaava vaihekorjatulle PWM:lle

$$f_{OCnPCPWM} = \frac{f_{clk\_I/O}}{N*510}$$

(ATmega16. 2010. 71-134)

Atmega16 kellotaajuutta voi nostaa ulkoisella kiteellä 16Mhz asti. Työssä tärkeää oli saada PWM signaalin taajuus oikealle alueelle [21-28Khz]. Olin ostanut Atmega16 tilauksen yhteydessä 16Mhz ja 12Mhz kiteitä. 12Mhz kidettä, vaihekorjattua PWM signaalia ja esiajastinta 1 käyttämällä pääsin hyvin lähelle 25Mhz:n tavoitetaajuutta.

[Kaava 2]

$$\frac{12Mhz}{1*510} = 23529hz$$

3-johtimen tuuletinliittimelle täytyi rakentaa oma ohjaus. Nopeutta voisi säätää pelkästään yhdellä N-kanavaisella fetillä, joka katkoisi tuulettimen maajohdinta. Ongelmaksi kuitenkin muodostuu se että maajohtoa käyttävä kierrosanturi ei toimisi. Minun alkuperäinen ohjaus 3 johtimen tuulettimelle oli tämän tyyppinen ja kierrosanturi lakkasi toimimasta. Ongelma voidaan ratkaista muun muassa siten, että PWM signaali kytketään N-kanavaisen fetin porttiin, mikä on kytketty maadoittamaan P-kanavaisen fetin porttia, mikä on kytketty katkaisemaan tuulettimen +12v linjaa.

5.2.2 Tuulettimen kierrosnopeus

Kierrosnopeusanturi toimii maadoittavana katkaisijana. Kierrosanturia ei voi lukea suoraan mikrokontrollerilla, koska se lukisi kokoajan tilaa 0. Pulssin tekeminen mikrokontrollerilla luettavaksi onnistuu siten että tuodaan kierrosanturin johtimelle +5v jännite ison etuvastuksen läpi ja kierrosanturin johtimelta mikrokontrollerille. Etuvastuksen läpi kulkeva virta pääsee kulkemaan kierrosanturin maadoittaessa ja mikrokontrolleri lukee tilan 0. Kierrosanturin ollessa poikki virta ei pääse kulkemaan joten potentiaali vastuksen jälkeen nousee +5v:iin ja mikrokontrolleri lukee tilan 1.

Tuulettimien kierrosnopeutta päädyin lukemaan ohjelmallisesti keskeytysvektorin avulla. Keskeytysvektori on aliohjelmakutsu jonka käynnistää jokin tietty tila mikrokontrollerilla, tässä tapauksessa käytin OC2 pinnin PWM signaalin huippua, jolloin näytteenottotaajuudeksi muodostuu 23529hz. Keskeytyksen aliohjelma toimii siten että joka kerta ohjelmaan siirryttäessä 16bittiseen laskuriin lisätään 1 ja luetaan tuulettimien kierrosanturien tila. Tuulettimessa tilamuutoksen tapahduttua laskurin tila tallennettaan muistiin, sitten kun toinen tilamuutos samalla tuulettimella havaitaan, vähennetään ensimmäisen tilamuutoksen tallentama laskuriarvo toisen tilamuutoksen laskuriarvosta. Lopputuloksena on arvo joka on yhden tila 1 tai tila 0 pulssin mittainen [¼ osa kierros].

Kaava 3. Laskuriarvon muuttaminen muotoon kierrosta minuutissa

$$\frac{f}{n} * \frac{60s}{4} = RPM$$

$f = Näytteenottotaajuus$
$n = Laskurin\ arvo$
$RPM = Kierrosta\ minuutissa$

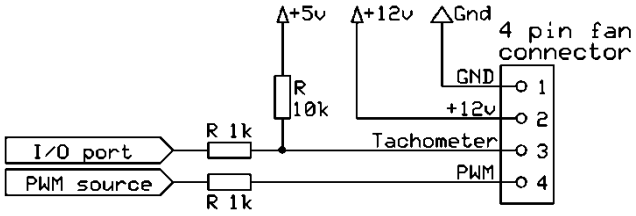
yksinkertaistamalla kaavaa voidaan se kirjoittaa muotoon

$$\frac{f * 15}{n} = RPM$$

Näytteenottotaajuus on vakio, [23529hz] joten ohjelmoinnissa käytetty lopullinen kaava on.

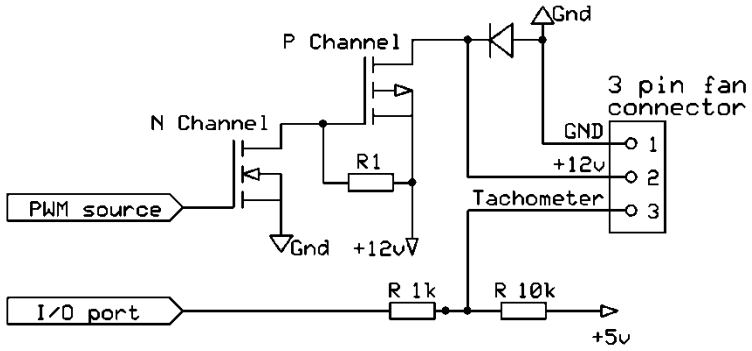
$$\frac{352941}{n} = RPM$$

5.2.3 Kytkennät



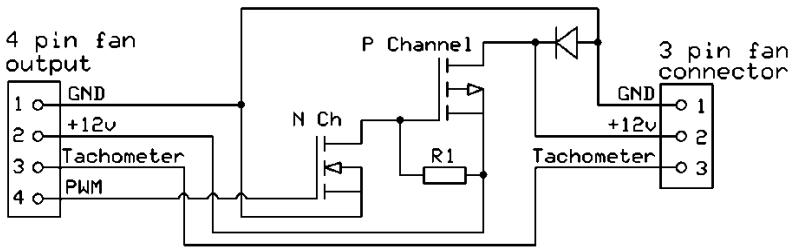
Kuva 9. 4-johtimen tuuletinliittimen kytkentä

Kuvan kytkennässä 1kohm vastuksien käyttö ei ole toiminnan kannalta välttämätöntä, mutta sillä pyritään suojaamaan mikrokontrolleria virheellisiltä kytkennöiltä yms.



Kuva 10. 3-johtimen tuuletinliittimen kytkentä

Kuvasta näkee kuinka N-kanavainen ja P-kanavainen fet tulisi kytkeä että tuulettimien ohjaaminen onnistuu ilman että kierrosanturi sekoaa.



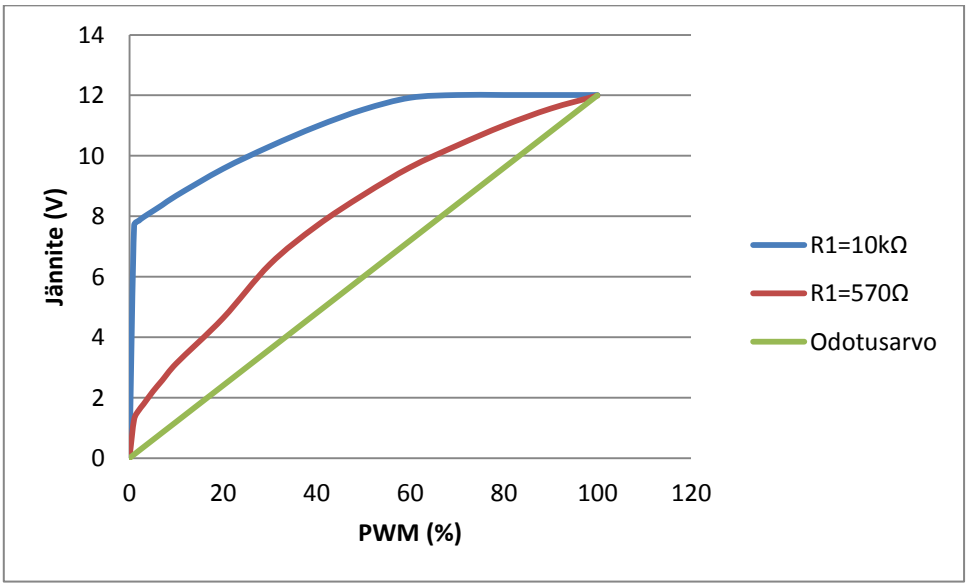
Kuva 11. 4-johtimen liittimen muuntaminen 3-johtimen liitännäksi

Kuva esittää miten pienen kytkennän avulla voidaan 4-johtimen liittimestä tehdä 3-johtimen liitin siten että ohjaus on edelleen toimiva. Ostin kytkennän testaamista varten P-kanavaiseksi fetiksi NXP BSP250 (BSP250. 1997) ja N-kanavaiseksi fetiksi NXP 2N7002BK. (2N7002BK. 2010.) Kytkennässä ilmeni kuitenkin pieni ongelma. PWM signaalia vastaavaa jännitettä ei syntynyt vaan se oli huomattavasti korkeampi.

Mittasin tuulettimen jännitettä kun ohjasin fet kytkentää PWM signaalilla. Käytin kahta erikokoista R1 vastusta 10kΩ ja 570Ω.

Taulukko 3. Taulukko esittää mitatut arvot tuuletinohjaus kytkennälle

PWM (%)	R1=10kΩ (V)	R1=570Ω (V)
0	0	0
1	7.70	1.28
2	7.85	1.58
3	7.97	1,78
5	8.17	2.20
7	8.37	2.57
10	8.68	3.13
20	9.57	4.63
30	10.31	6.41
40	10.97	7.68
50	11.53	8.71
60	11.92	9.62
70	12.01	10.34
80	12.01	11.00
90	12.01	11.56
100	12.01	11.99



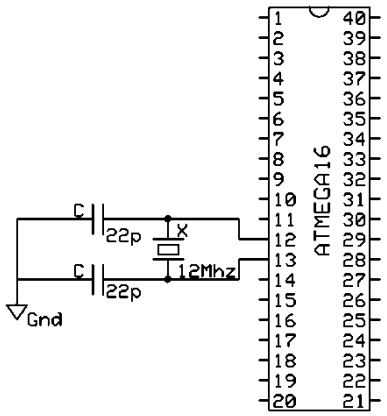
Kuva 12. Graafinen esitys edeltävästä taulukota

Tuulettimen ohjaus toimii muuten aivan hyvin mutta jännite ei ole oikea. Veikkaisin syyksi sitä, että kun N-kanavainen fet avautuu ja maadoittaa P-kanava fetin ohjausta, aukeaa P-kanavainen fet nopeasti. N-kanavaisen fetin sulkeutuessa R1 vastuksen läpi tuleva 12v jännite sulkee P-kanavainen fetin, mutta sulkeutuminen ei tapahdukaan nopeasti kytkennän pienen sisäisen kapasitanssin takia. Epäilen kapasitanssia siksi koska jännite seuraa sitä paremmin odotusarvoa mitä pienempi R1 vastus on, mutta 570Ω vastus on pienin 0.25w vastus jota voi käyttää. Harkitsin että ongelmaan voisi olla ratkaisu siinä että n-kanava fetin korvaisi operaatiovahvistimella, tai korvata molemmat fetit riittävän tehokkaalla operaatiovahvistimella joka jaksaisi ohjata tuuletinta.

5.3 Atmega16 kellotaajuuden nostaminen

Atmega16 toimii 1Mhz taajuudella sisäisellä kellolla. Ulkoisen kiteen käyttöönotto onnistuu siten että kirjoitetaan asetusbitit [sulakkeet] ulkoisen kellon käyttöön. Sulakkeiden uudelleen kirjoittaminen onnistuu suoraan AVR Extreme burner ohjelmalla Fuse bits/Settings välilehden alta.

Asetusbittejä on kaksi 8bit rekisteriä. Alemmasta rekisteristä [Low fuse] asetetaan bitit CKSEL3..1 tilaan 1 ja ylemmästä rekisteristä [High fuse] CKOPT asetetaan tilaan 0 jolloin ATmega16 käyttää ulkoista kidettä. Samalla asetin mikrokontrollerin käynnistymään hitaasti mitä varten ohjelmoin alemmasta rekisteristä CKSEL0 ja SUT1..2 bitit tilaan 1



Kuva 13. ATmega16 kiteen kytkeminen.

Kidettä käyttäessä kondensaattorien tulee olla samankokoisia. kondensaattorien tulisi olla 12-22pf väliltä. (ATmega16. 2010. 24-27)

5.4 Lämpötila-anturi

Lämpötila-antureita etsiessäni törmäsin lämpötilan suhteen lineaarisella jännitekäyrällä toimiviin lämpöantureihin, joita olisi helppo lukea AD muuntimella. Termistorin käyttö olisi ollut vaihtoehto, mutta termistorin lukeminen on huomattavasti työläämpää koska se tuottaa epälineaarisen jännitekäyrän. Termistorin käyttö olisi kuitenkin ohjaimessa täysin mahdollista pienillä muunnoksilla ja ohjelmapäivityksellä.

Päätin käyttää Analog Devices 22100KTZ anturia. 22100 KTZ anturissa on kolme johdinta VCC [4-6.5V], GND [0V] ja V_0. Anturi tekee V_0 johtimelle jännitteen joka on välillä GND-VCC. Lähdön jännitteen ja lämpötilan suhde on anturilla lineaarinen joten AD muuntimella luettu jännite on helppo muuttaa todelliseksi lämpötilaksi.

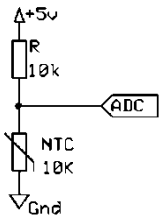
Kaava 4. AD22100KTZ anturin lähdön jännite.

$$V_{OUT} = \left(\frac{V+}{5\,V}\right) * \left(1.375\,V + 22.5\frac{mV}{°C} * T_A\right)$$

$V_{OUT} = Lähdön\ jännite$
$V+ = Jännite\ [VCC]$
$T_A = Lämpötila$

(AD22100. 2004.)

Termistorin käyttö onnistuisi siten että tekisi toisen vastuksen avulla jännitteen jaolla toimivan kytkennän.



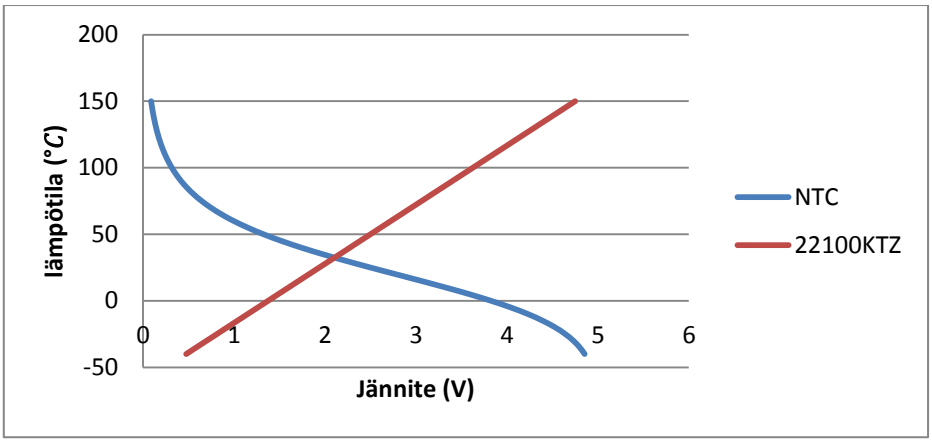
Kuva 14. Termistorin kytkentä.

Kaava 5. Termistorikytkennän ADC jännite.

$$U_{ADC} = \frac{R_{NTC}}{R + R_{NTC}} * U$$

$U_{ADC} = ADC\ jännite$
$R_{NTC} = Termistorin\ resistanssi$
$R = Etuvastuksen\ resistanssi\ (10k\Omega)$
$U = käyttöjännite\ (5V)$

Sijoittamalla kaavaan NTC termistorin resistanssin eri lämpötiloissa pystyy piirtämään jännite-lämpötilakäyrän ADC:lle.



Kuva 15. Termistorin ja AD22100 anturien tuottamat jännite-lämpötilakäyrät

vertailemalla edellä olevaa kytkentää Vishay BCcomponents NTCLE203E3103FB0 NTC termistorin kanssa ja AD22100 anturia huomaa kuinka erilaisia anturien jännitekäyrät ovat. Termistorin resistanssin eri lämpötiloilla löytyi anturin datalehdeltä. (NTC Thermistors, Radial Leaded, Accuracy Line. 2011.)

5.4.1 AD muunnin ja AD22100KTZ

Atmega16:ssa on 8kpl 10bit AD muuntimia. AD muunnin eli analogia-digitaalimuunnin käyttämällä voidaan lukea AD22100KTZ anturien antamaa jännitettä. Lämpöanturia mitatessani käytin 5v käyttöjännitettä vertailujännitteenä joka poisti pienten jännitemuutosten aiheuttaman virheen mittauksesta, koska lämpöanturi käyttää samaa jännitettä. AD muunnin muuttaa luetun jännitteen digitaaliseen muotoon. AD muunnos on lineaarinen, joten muunnos on suoraan verrannollinen mitattuun jännitteeseen.

Muunnos funktion laskeminen:

Kaava 6. AD muuntimen arvo suhteessa jännitteeseen

$$V_{in} = \frac{x}{n} * V_{ref}$$

$V_{in} = sisääntulo\ jännite$
$x = AD\ muuntimen\ antama\ arvo$
$n = AD\ muuntimen\ erottelukyky$ (1023)
$V_{ref} = Vertailujännite$ (5V)

Kaava 4 täytyy muuttaa sellaiseen muotoon että se ratkaisee lämpötilan.

$$T_A = \frac{\frac{V_{OUT}}{\left(\frac{V+}{5\,V}\right)} - 1.375\,V}{22.5\frac{mV}{°C}}$$

V_{out} tilalle voidaan sijoittaa kaava 6.

Kaava 7. AD-muuntien arvo lämpötilaksi.

$$T_A = \frac{\frac{\frac{x}{n} * V_{ref}}{\left(\frac{V+}{5\ V}\right)} - 1.375\ V}{22.5\frac{mV}{°C}}$$

$$T_A = \frac{x * V_{ref}}{n * \left(\frac{V+}{5\ V}\right) * 22.5\frac{mV}{°C}} - \frac{1.375\ V}{22.5\frac{mV}{°C}}$$

Yhtälössä ainoa tuntematon on x joten ratkaistaan kaava mahdollisimman yksinkertaiseen muotoon.

$$T_A = \frac{x * 5\ V}{1023 * \left(\frac{5\ V}{5\ V}\right) * 22.5\frac{mV}{°C}} - \frac{1.375\ V}{22.5\frac{mV}{°C}}$$

$$T_A = \frac{5}{1023 * 22.5\frac{mV}{°C}} * x - 61.111\ ...$$

$$T_A = 0.2172x - 61.111\ ...$$

Yhtälöllä voidaan nyt laskea lämpötila sijoittamalla x tilalle mitattu AD muuntimen arvo. Ohjelmassa käsittelin lämpötilaa 0.1 asteen tarkkuudella, joten muutin funktion muotoon

$$T_A = 2.172x - 611.11\ ...$$

Lisäksi binääriluvuilla laskiessa lopullinen funktio on muotoa

$$T_A = 2.171875x - 611$$

Olisin voinut käyttää suurempia tarkkuuksia, mutta jo kyseisellä tarkkuudella lämpötilat ovat todella lähellä todellista ja koska anturin sisäinen toleranssi on ±0.5 °C, on tarkkuus mielestäni riittävä

5.5 Näyttö

Näyttönä käytän Hitachin HD44780 ohjaimella varustettua aakkosnumeerista näyttöä. Aakkosnumeerisia LCD näyttöjä on olemassa myös muilla ohjaimilla, mutta hyvin monet ovat yhteensopivia HD44780:n kanssa. Näyttönä päätin käyttää 16x2 merkkistä näyttöä, [16 merkkiä rivillä ja 2 riviä] koska tarkoituksena oli saada ohjain sopimaan 5.25” lisälaiteasemapaikkaan ja suurempien näyttöjen kanssa saattaisi tulla tilaongelmia.

Näytön ohjaamiseen päätin käyttää valmista netistä löytyvää koodia, jota hieman muokkasin paremmin tarpeisiini sopivaksi. (HD44780 assembly driver.)

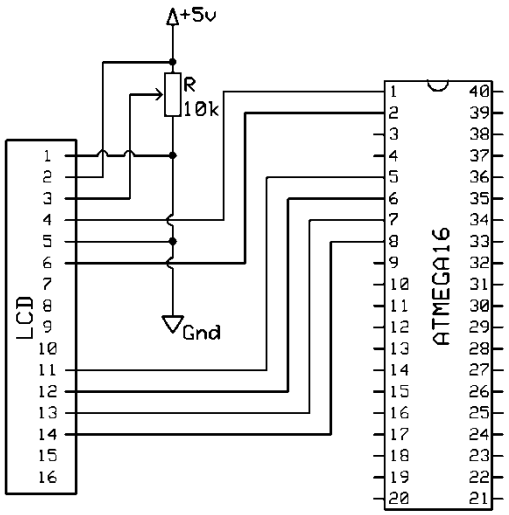
5.5.1 Kytkentä

HD44780 näytöissä on 16 johdinta [14 ilman taustavaloa]. Näyttövalmistajasta ja tyypistä riippuen liittimet voivat olla hieman toisistaan poikkeavia, mutta johtimet on numeroitu ja ne toimivat samoin.

Taulukko 4. HD44780 Johdintaulukko. (What is a Hitachi HD44780 LCD Display. 2009.)

1	Maa (0V)
2	VCC (mallista riippuen 3,3V - 5V)
3	Kontrastin säätö
4	Rekisterin valinta (RS) RS=0: Komento, RS=1: Data
5	Lue/Kirjoita (R/W) R/W=0: Kirjoita, R/W=1: Lue
6	Kello/Toteutus (Enable)
7	Bitti 0 (Ei käytössä 4-bit ohjauksessa)
8	Bitti 1 (Ei käytössä 4-bit ohjauksessa)
9	Bitti 2 (Ei käytössä 4-bit ohjauksessa)
10	Bitti 3 (Ei käytössä 4-bit ohjauksessa)
11	Bitti 4
12	Bitti 5
13	Bitti 6
14	Bitti 7
15	Taustavalon anodi (+)
16	Taustavalon katodi (-)

Päätin käyttää näyttöä 4-bit ohjauksella jolloin säästin mikrokontrollerilta 4 I/O pinniä.



Kuva 16. LCD näytön kytkentä ATMega16:lle

Koekytkennässä käytin edellä olevaa kytkentää LCD-näytön kanssa. Potentiometrilla pystyy säätämään näytön kontrastia. LCD-näytön pinni 5 [R/W] on pysyvästi maissa koska työssä ei ole tarpeen lukea dataa näytöltä.

6 OHJELMA

Ohjelmoinnin olisin voinut suorittaa joko C:llä tai assemblerilla ja päädyin käyttämään assembleria. Suoranaista syytä assemblerin valitsemiseen ei ollut, mutta pari ensimmäistä kokeiluohjelmaa tein assembler esimerkkien avulla, joten päätin ohjelmoida itse tuuletinohjaimen assemblerilla. Assemblerilla ohjelmoiminen osoittautuikin hankalaksi, mutta myös mielenkiintoiseksi, koska jokainen normaalisti lähes itsestään selvä asia piti jakaa pieniin osiin. Assembler osoittautui myös hyvin opettavaiseksi siitä miten mikrokontrolleri oikeasti toimii ja samalla avarsi hyvin paljon maailmaa tietotekniikasta.

Ohjelmia kirjoittaessa harvoin onnistuin kerralla ja pitempiä osuuksia koodista jouduin selaamaan edestakaisin kymmeniä kertoja ennen kuin sain kaikki toimimaan.

6.1 PWM säädöt

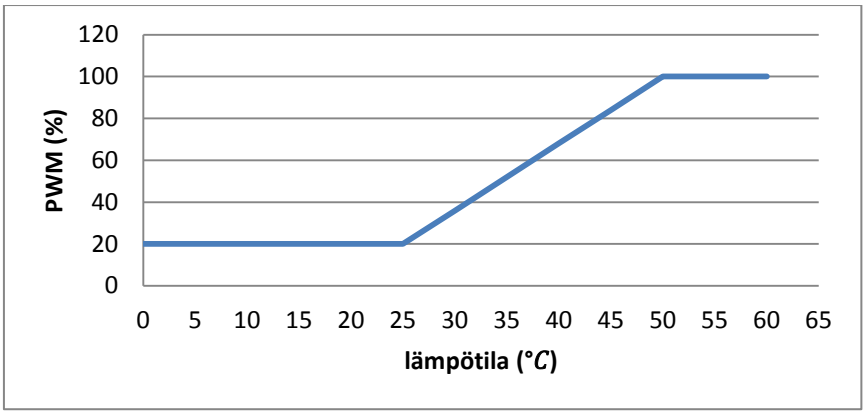
Tuulettimien ohjaamisen päätin rakentaa siten että tuulettimia voi ohjata joko manuaalisesti tai automaattisesti lämpötilan perusteella. Valikoista pystyisi valitsemaan kumpaa käyttää ja säätämään ohjauksen asetukset. Manuaalisella säädöllä voidaan asettaa minimiraja PWM:lle jolloin voidaan estää tuulettimen pysähtyminen. Automaattisessa säädössä valitaan lämpöanturi, PWM minimiraja, alalämpöraja, ylälämpöraja. PWM saa minimiarvon niin kauan kun lämpötila on alle alalämpörajan. Alalämpörajan ylitettyä PWM nousee lineaarisesti 100% kohti kunnes saavuttaa sen ylälämpörajalla. Lämpötilan ylitettyä ylälämpörajan PWM on 100%:ssa

Kaava 8. Logiikka ja funktio PWM:n laskemiseen

$$Jos\ T < T_A\ niin\ P = P_{min}\ muuten\ P = \frac{P_{min} - P_{max}}{T_A - T_Y} * (T - T_A)$$

$$jos\ P > P_{max}\ niin\ P = P_{max}$$

$T = Lämpötila$
$T_A = Alalämpöraja$
$T_Y = Ylälämpöraja$
$P_{min} = PWM\ minimiraja$
$P_{max} = PWM\ maksimi\ (100\%)$
$P = Käytetty\ PWM$



Kuva 17. Graafinen esitys miten automaattiohjaus toimii jos PWM minimiksi on asetettu 20 % alalämpörajaksi 25 °C ja ylälämpörajaksi 50 °C

Harkitsin myös toisentyyppistä automaattiohjausta jossa PWM säädettäisi kahden lämpötilan eron mukaan. Ohjauksen pystyy ohjelmoimaan ohjaimeen jälkeenpäin enkä tiennyt olisiko ominaisuus oikeasti tarpeen, joten jätin sen kehitysideaksi.

6.2 Valikot ja hallinta

Hallintalaitteita suunnitellessa päädyin viiteen nappiin, plus [plus], miinus [minus], eteen [next]. taakse [back] ja asetukset [setup]. Ihan ensimmäisessä versiossa olin harkinnut että käyttäisin kolmea nappia valikoissa navigoimiseen ja 4 potentiometriä tuulettimien manuaaliseen säätöön. Sopivien potentiometrien löytäminen koituikin ongelmaksi, joten uudelleenkirjoitin koodin toimimaan täysin napeilla. Rakensin nappien toiminnan siten että plus tai miinus nappia pitäessä pohjassa hetken päästä nappi alkaa niin sanotusti juosta, kun taas muut napit toimivat kertapainalluksella. Lisäksi jos kahta tai useampaa nappia painaa yhtä aikaa kaikki painallukset mitätöityy.

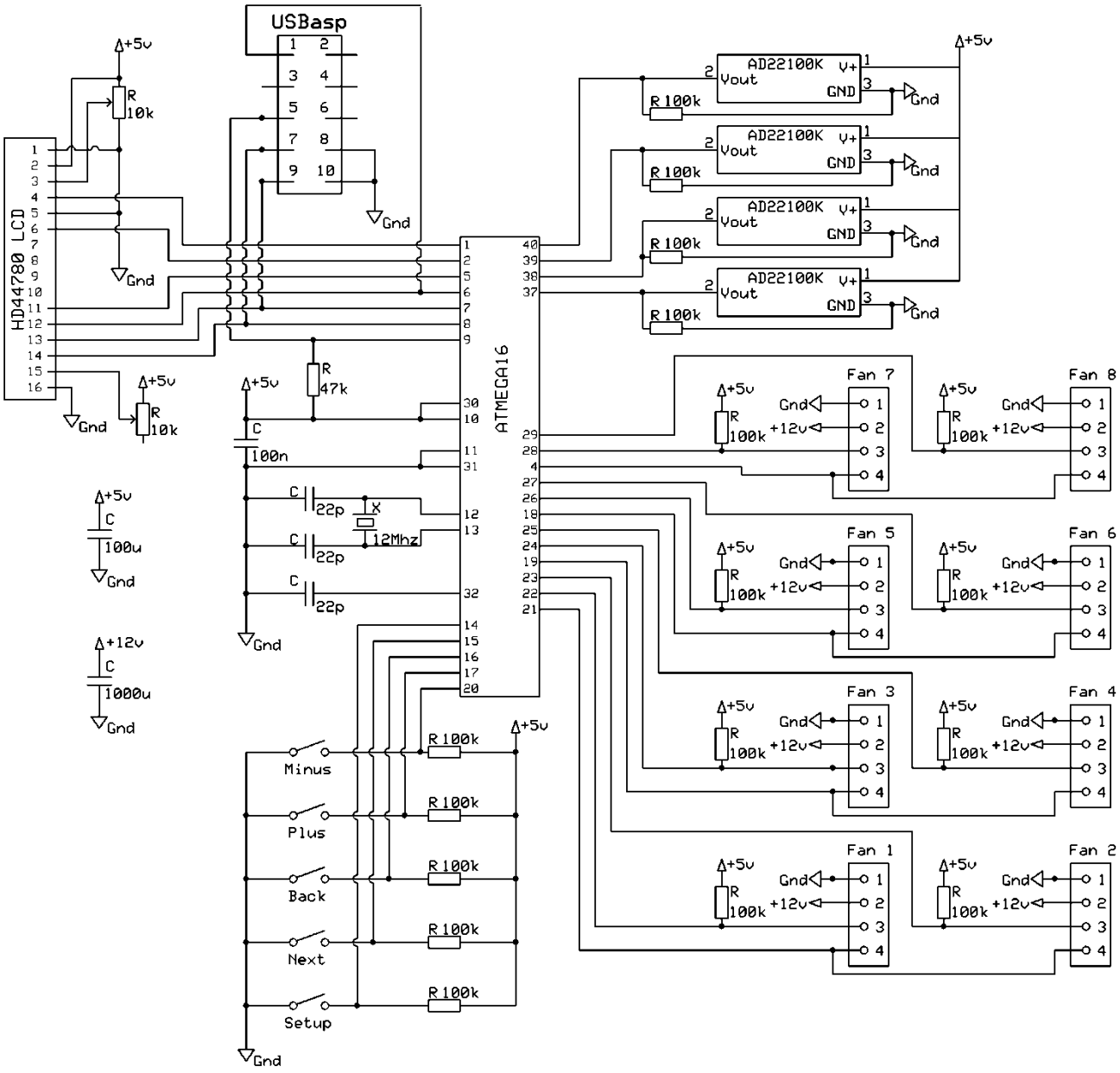
Valikot päätin rakentaa siten että päävalikossa on 6 sivua. Sivut 1-4 esittää sivunumeroa vastaavan PWM linjan tehon prosentuaalisesti, sekä kyseisen linjan ohjaamien tuulettimien kierrosnopeudet. sivuilla 5 ja 6 näytetään lämpötila-antureilta luetut lämpötilat. Painamalla asetukset nappia sivuilla 1-4 pääsee säätämään sivua vastaavan PWM:n asetuksia.

PWM asetusvalikon ensimmäinen sivu kysyy käytetäänkö automaattista vai manuaalista ohjausta. Valitsemalla manuaaliohjauksen kysytään PWM:n minimitehon. Valitsemalla automaattiohjauksen kysytään mitä lämpöanturia käytetään, PWM:n minimiarvoa, alalämpöraja ja ylälämpöraja. Jokaiselle asetettavalle arvolle on oma sivu ja sivuja voi selata eteen ja taakse napeilla. Painamalla asetukset nappia uudelleen asetukset tallennetaan ja palataan päävalikkoon. Manuaalinen säätö tapahtuu siten että päävalikon ollessa jossakin 4:stä PWM sivusta plus ja miinus napit säätävät kyseisen sivun PWM:n tehoa.

7 OHJAIMEN RAKENTAMINEN

Ohjaimen rakentamista ideoin jo paljon ennen kun rakentaminen oli ajankohtaista. Olin jo aiemmin päättänyt rakentaa ohjaimen sopimaan 5.25” asemapaikkaan. Asemapaikka vaatisi sen että kaikkien tietokoneen sisään tulevien liittimien tulisi olla kotelon takana. Kotelon taakse päätin asentaa piirilevyn pystyyn. Piirilevyyn tulisi kaikki kotelon sisäiset liittimet sekä niin paljon muita komponentteja kuin saan mahtumaan. Koteloinnin mietin mahdollisimman minimaaliseksi ja päädyin sellaiseen ratkaisuun jossa koteloon tulisi vain etulevy ja sivulevyt. Sivulevyihin tekisin reiät kierteillä, jotta ohjaimen saisi kiinnitetty ruuveilla tietokonekoteloon. Etulevyyn tulisi näyttö sekä napit.

Ohjaimen rakentamista varten täytyi miettiä lopulliset kytkennät, jotta pystyin suunnittelemaan piirilevyt, joten piirsin kytkentäkaavion.

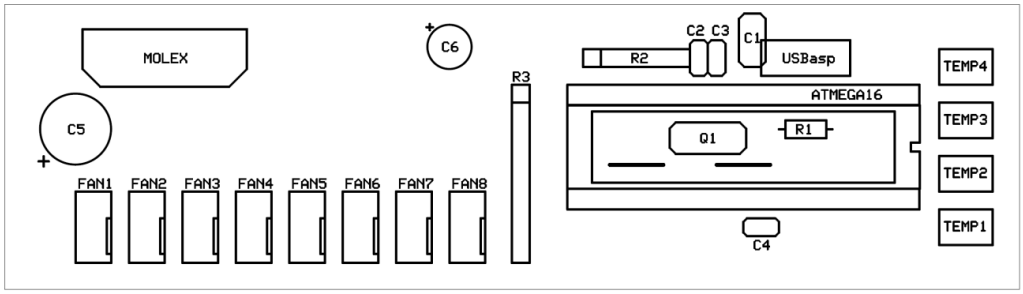


Kuva 18. Tuuletinohjaimen kytkentäkaavio

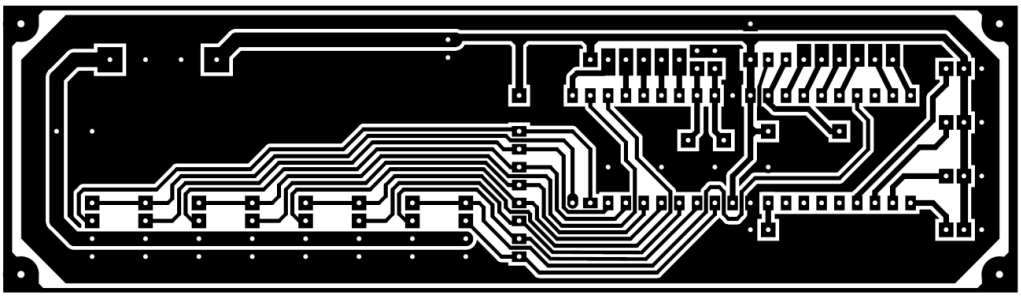
Ohjaimen kytkennät ovat yksinkertaisia, mutta useat kytkennät toistuvat monesti. Lämpöantureille lisäsin maadoittavan vastuksen jotta AD muuntimet lukevat oikeasti 0 jos lämpöantureita ei ole kytketty. Lisäksi poistin kytkennästä PWM ohjauksen ja kierrosanturien etuvastukset koska ne eivät olleet toiminnan kannalta tarpeellisia.

7.1 Piirilevyjen suunnittelu

Piirilevyt suunnittelin ExpressPCB ohjelmalla. (ExpressPCB.) Ensimmäisenä suunnittelin suuren ohjaimen takaosaan tulevan piirilevyn. Piirilevy sai olla yhtä korkea ja leveä kuin 5.25” lisäasemapaikka eli 42 x 146 mm. Asemapaikan korkeuden ja leveyden tarkistin yksinkertaisesti mittaamalla ne CD asemasta työntömitalla. Tein piirilevyn 41 x 145 mm mitoilla, jotta jäisi hieman tilaa reunoille.



Kuva 19. Komponenttien sijoittelu.



Kuva 20. Piirilevyn kytkentäkuvio.

Kytkentäkuvioon päätin tehdä 0.5 mm reiän läpivienneille, että se hieman ohjaisi suurempaa poranterää oikealle kohdalle. Suunnittelin kulmiin paikat 3 mm reille joista piirilevyn saisi koteloon kiinni.

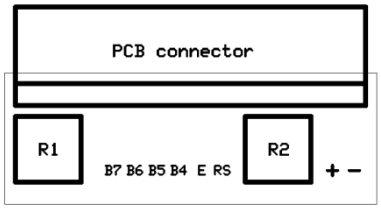
Osaluettelo:

C1 = Kondensaattori 100nF
C2, C3, C4 = kondensaattori 22pF
C5 = Kondensaattori 1000µF (ei välttämätön)
C6 = Kondensaattori 100µF (ei välttämätön)
R1 = Vastus 47kΩ
R2 = Vastusverkko 100kΩ, 6 nastainen, 5 vastusta yhteisellä nastalla.
R3 = Vastusverkko 100kΩ, 10 nastainen, 9 vastusta yhteisellä nastalla.
Q1 = Kide 12Mhz
FAN1..8 = 4 johtimen tuuletinliitin

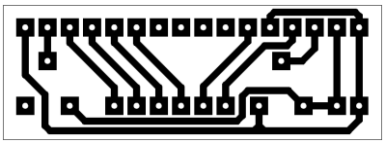
TEMP1..4 = 3 johtimen liitin lämpöanturille.
ATMEGA16 = ATmega16 ja matalaprofiilinen PDIP kanta
USBasp = 5 johtimen liitin USBasp ohjelmoijalle.
MOLEX = Molex 8981 virtaliitin.

R3 vastuksesta leikataan ensimmäisen vastuksen nasta yhteisen nastan vierestä nasta irti. Vastuksesta irrotetaan nasta, että maakytkentä kulkee vastusverkon ohi. Piirilevyyn juotetaan johtoja näytölle ja napeille kuparipuolelle joten niille on jätetty suuremmat kuparialustat.

Päätin tehdä erillisen pienen piirilevyn näytölle joka myös toimisi näytön liittimenä. Piirilevyssä on myös potentiometrit näytön kontrastin sekä taustavalonkirkkauden säätöön.



Kuva 21. Näyttöpiirilevyn komponenttiasettelu



Kuva 22. Näyttöpiirilevyn kytkentäkuvio

Osaluettelo:

PCB connector = 16 johtimen piirilevyliitin 90° kulmalla 2.54mm jaolla
R1, R2 = 10kΩ Potentiometri
B4..B7,E,RS = Näytön ohjaus (juotetaan johdot)
+ = +5v
- = GND

Napeille tulevan piirilevyn suunnittelin ja valmistin kotelon rakentamisen yhteydessä.

7.2 Piirilevyn valmistaminen

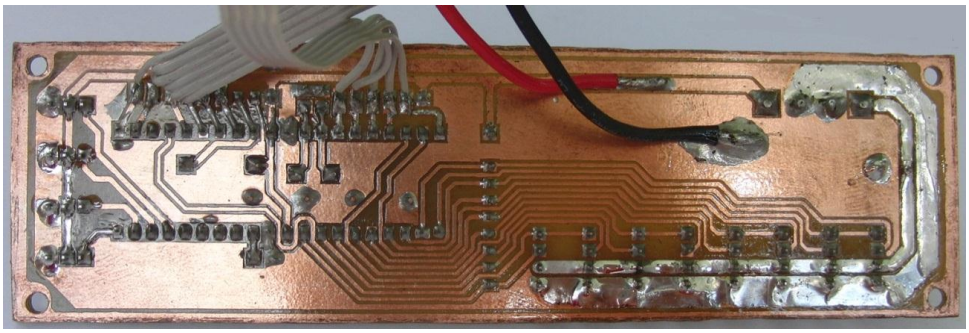
Alkuperäinen suunnitelma oli teettää koululla piirilevy tai levyt. Piirilevyjä suunnitellessa rupesin harkitsemaan piirilevyjen valmistusta itse. Alkuun harkitsin valotusmenetelmän käyttämistä, eli hankkinut valmiiksi UV reagoivalla lakalla pinnoitetta piirilevyä ja UV lampun. Myöhemmin löysin niin sanotun musteensiirtomenetelmän, jota päätin koittaa.

Prosessissa tarkoituksen on siirtää tulostetulta paperilta tai kalvolta muste toiseen pintaan. Piirilevyn valmistuksessa käyttö onnistuu siten että ensin tulostetaan kiiltopaperille piirilevyn muodot joita ei ole tarkoitus poistaa ja

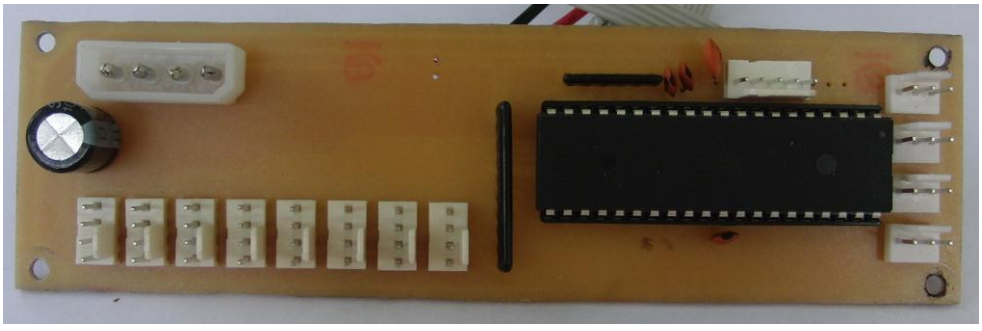
putsataan piirilevy. Tulostettu paperi asetetaan putsatun piirilevyn pintaan mustepuoli kuparipintaa vasten ja kuumalla silitysraudalla kevyesti painetaan paperia piirilevyä vasten. Silitysraudan lämpö sulattaa paperista musteen jolloin se tarttuu kuparipintaan kiinni, samalla myös paperi tarttuu piirilevyyn kiinni. Paperin saa poistettua liottamalla paperia ja piirilevyä vedessä. Vettyneen paperin saa poistettua helposti sormin ja esimerkiksi hammasharjan avulla. Paperista putsatun piirilevyn voi syövyttää, koska tulostinmuste ei liukene syövytteeseen, ainakin jos kyse on ferrikloridista. (How To Make PCB using Toner Transfer method in 7 simple steps. 2011.)

Musteensiirtoa käyttäessä on tärkeää putsata piirilevy huolellisesti ja välttää koskemista tulostetun paperin mustepintaan, etteivät lika ja rasva estä musteen tarttumista. Huomasin että silitysrauta tulisi vain asettaa paperin ja piirilevyn päälle kunnes paperi tarttuu kiinni. Jos silitysrautaa liikuttaa heti saattaa paperi liikkua mikä aiheuttaa suttuisen jäljen ja tekee siirrosta kelvottoman. Paperin tartuttua piirilevyyn voi silitysrautaa liikutella varmistaakseen että paperi ottaa kontaktia piirilevyn joka kohtaan. Paperia poistaessa musteeseen jäi pieniä määriä paperinukkaa, jota ei saanut pois poistamatta myös mustetta. Paperinukka ei tosin haitannut syövytystä vaikka ferrikloridi hieman alkuun kupli. Käyttämällä tulostettavaa kalvoa tai erilaista kiiltopaperia nukkaa ei välttämättä jäisi.

Ennen musteen poistamista piirilevyn läpiviennit on hyvä porata, koska muste toimii kuparille korroosiosuojana. Musteen pystyy helposti poistamaan esimerkiksi asetonilla. Piirilevy täytyy suojata korroosiolta ja se onnistuu esimerkiksi tinaamalla tai lakkaamalla piirilevy. Itse käytin juotosaktiivista lakkaa, joka helpotti myös komponenttien juottamista.

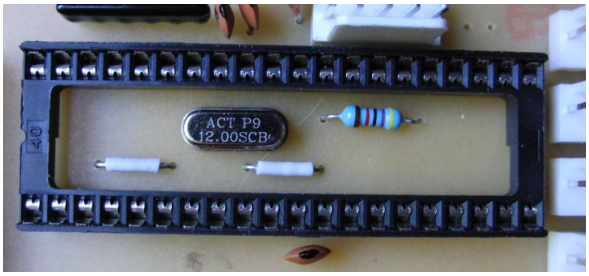


Kuva 23. Piirilevyn kuparipuoli



Kuva 24. Komponentit

Mikrokontrollerin alla on kide vastus sekä kaksi hyppy johdinta



Kuva 25. Komponentit mikrokontrollerin alla

7.3 Kotelon rakentaminen

Ensimmäisen kotelon tarkoitus oli saada kaikki osat yhteen ja saada hieman tuntumaa kotelosta, jos jokin kaipaisi muutosta. Kotelosta ei ollut tarkoitus siis saada kaunista. Tein kotelon kotona käsityökalulla (rautasaha, smirgeli, dremel ja pora). Yritin tehdä kotelosta asiallisen näköisen mutta en halunnut tuhlata kauheasti aikaa ensimmäisen kotelo version tekemiseen. Päätin koota kotelon siten että sivulevyjen päät taitettaisi 90° asteen kulmaan, jotta niihin saisi taakse tulevan piirilevyn sekä etulevyn kiinni. Pokkauksiin tuli kierteelliset reiät, jotta kotelon saisi ruuveilla kasaan. Myöhemmin ruuvit osoittautuivatkin hyväksi ideaksi koska kotelon purkaminen oli helppoa.

Parin huonosti onnistuneen yrityksen jälkeen päätin kokeilla piirilevynvalmistuksessa käytettyä musteensiirtoa metallilevyjen tekemiseen. Piirsin ExpressPCB ohjelmalla kuvion, joka oli metallilevyn muotoinen, jonka siirsin alumiinilevyn palaseen. Muotoilin alumiinilevyä smirgelillä ja dremelillä mahdollisimman tarkasti muotojen mukaan ja porasin reiät oikeille kohdille. Sivulevyistä taitoin kiinnityskorvakkeet käsin ruuvipenkissä.

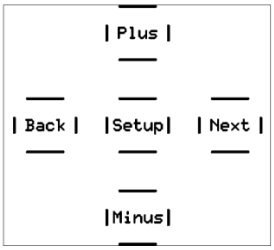
Kuva 26. Sivulevyjen kuvio

Kaikkiin kuvion osoittamiin reikiin porasin 2 mm reiän ja tein kierretyökalulla M3 kierteen. Sivuissa olevat korvakkeet taitettaisi siten että ne taittuisivat koteloon nähden sisään.

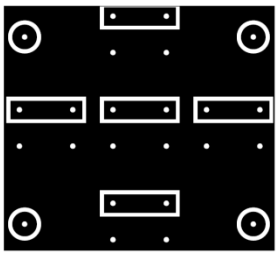
Kuva 27. Etulevyn kuvio

Etulevyyn kulmissa olevat pienet reiät osoittavat paikat joihin tulisi kasausruuvit ja niihin porattaisi 3mm reiät. Viisi ristin muodossa olevaa reikää olisi paikat napeille ja rinkulat ovat vain poraamisen kohdistamista varten. Suorakulmainen aukko on kolo näytölle.

Napit tein varastossa lojuvasta 8 mm alumiiniputkesta, josta sahasin pieniä 5mm pitkiä palasia. putken palaset täytin epoxiliimalla ja samalla liimasin pohjaan pienen alumiinilevyn. Levy esti nappien putoamisen etulevystä. Nappien taakse suunnittelin piirilevyn johon juottaisin kiinni piirilevykytkimiä. Johtimet napeille juotin suoraan piirilevyyn. Kytkiminä käytin
6 x 6 x 5 mm piirilevykytkimiä

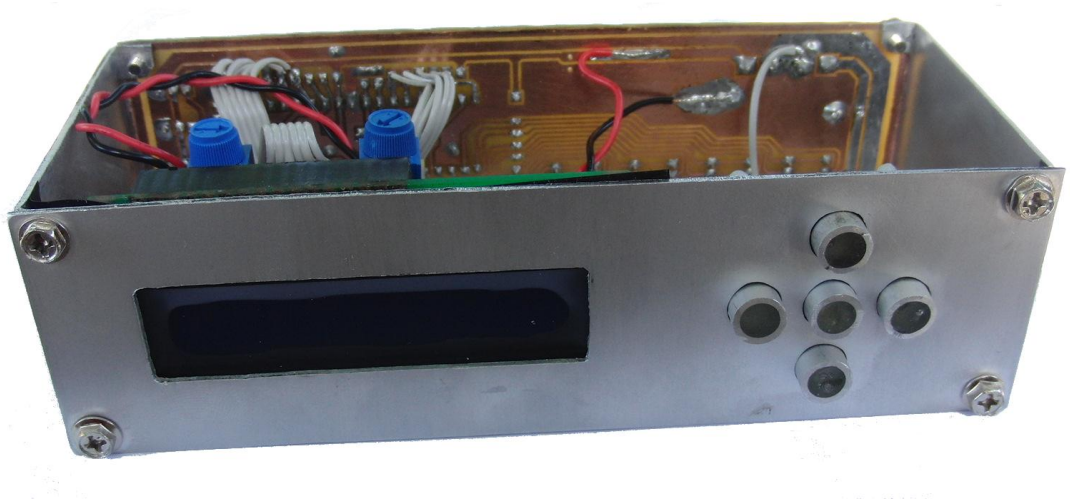


Kuva 28. Näppäinasettelu

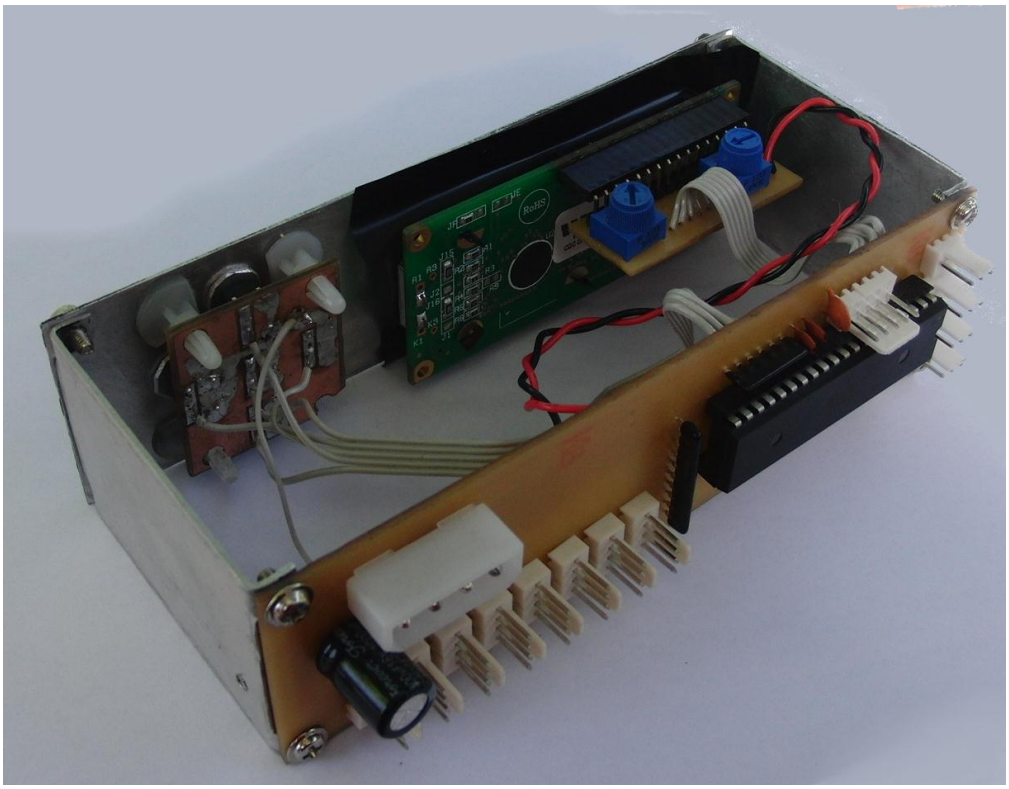


Kuva 29. Nappipiirilevyn kytkentäkuvio

Kulmissa olevat renkaat auttavat vain poraamisen kohdistamisessa. Kulmiin porasin 4 mm reiät joihin laitoin emolevyihin tarkoitetut muoviset korokepalat, joita liimasin kotelon etulevyyn. Kulmien reikiä poratessa yksi kulma murtui, mutta piirilevy pysyi riittävän hyvin kolmella kiinni. Painikkeiden väljyyttä pyrin vähentämään siten että juotin piirilevykytkimet sen verran eteen että ne melkein ottivat nappien pohjaan kiinni.



Kuva 30. Ohjain



Kuva 31. Ohjain

Ohjaimessa näyttö on vain teipillä kiinni. Näytön kiinnitys vaatii siis jatkokehitystä. Näytön kulmissa olevat 4 reikää voisivat toimia hyvinä kiinnityspisteinä, mutta reiät ovat alle 3 mm joten pitäisi käyttää M2 ruuveja tai jotain muuta pientä liitäntätapaa, jota en ole toistaiseksi keksinyt

8 YHTEENVETO

Työn tarkoituksena oli tehdä tuulettimien kierrosnopeutta säätävä ohjain tietokoneeseen. Mikrokontrollerin avulla ohjaimen tarkoituksena oli ohjata tuulettimia lämpötilan perusteella tai manuaalisesti. Ohjaimeen oli tarkoitus lisätä myös näyttö jossa näytettäisi erilaista tietoa, kuten lämpötilat kierrosnopeudet ja lisäksi se toimisi asetusvalikkona.

Perehdyin miten tietokoneessa yleisesti käytössä olevat 4 ja 3 johtimen tuulettimet toimivat ja miten PWM signaalilla pystyy ohjaamaan tasavirtamoottoria. Työssä päätin käyttää Atmel ATmega16 mikrokontrolleria. Käytin mikrokontrollerille USBasp ohjelmoijaa sekä atmelin AVR Visual Studio 5:sta ohjelmointiympäristönä. Lämpö-antureita valinnassa päädyin AD 22100KTZ anturiin. Näyttönä käytin Hitachi HD44780 ohjaimella varustettua aakkosnumeerista näyttöä. Ohjelmoinnin mikrokontrollerille tein assembler kielellä. Aluksi rakensin mikrokontrollerin koekytkentäalustalle ja rakensin ohjelmaa sitä mukaan kun rakensin kytkentöjä. Piirilevyt ohjaimeen tein musteensiirtomenetelmällä ja kotelon rakensin käsityönä alumiinilevystä.

Työssä pääsin melko hyvin tavoitteeseeni ja ohjain toimi kuten halusin. Vasta kun ohjain on ollut pitempään käytössä ja päässyt kunnolla testaamaan voin sanoa mikä vaatisi korjaamista tai muutosta. Rakensin ohjaimen siten että pystyn uudelleen ohjelmoimaan sen tarvittaessa, joten jos puutteet osoittautuvat ohjelmallisiksi pystyn helposti päivittämään ohjaimen.

Tavoitteista jäi uupumaan moduuli, joka olisi muuntanut 4-johtimen tuuletinliittimen 3-johtimen liittimeksi siten että kierrosnopeusohjaus olisi toiminut. Varsinaisen ohjaimen kannalta se ei ollut ongelma koska ohjaimessa on 4-johtimen liittimet ja moduuli käyttö olisi vapaavalintaista.

Työtä tehdessä opin todella paljon uutta. Opin hyvin paljon ohjelmoinnista ja siitä, miten ainakin Amtel AVR mikrokontrollerit toimivat. Opin paljon elektroniikkasuunnittelusta ja opin valmistamaan piirilevyjä. Työn valitsinkin pääasiassa sen takia että halusin oppia ohjelmointia sekä elektroniikkasuunnittelua ja se tavoite tuli saavutettua. Opittavaa on toki vielä paljon, mutta nyt on avattu ovi uuteen maailmaan.

LÄHTEET

2N7002BK. 2010. Datasheet. NXP
http://www.nxp.com/documents/data_sheet/2N7002BK.pdf

3 Pin Fan Connector Pinout. 2008. Motherboard (CPU). AllPinouts
http://www.allpinouts.org/index.php/Motherboard_%28CPU%29_3_Pin_Fan_Connector

4 Pin Fan Connector Pinout. 2010. Motherboard (CPU). AllPinouts
http://www.allpinouts.org/index.php/Motherboard_%28CPU%29_4_Pin_Fan

4-Wire Pulse Width Modulation (PWM) Controlled Fans. 2004. Form factors
http://www.formfactors.org/developer/specs/REV1_2_Public.pdf

AD22100. 2004. Datasheet. Analog Devices
http://www.analog.com/static/imported-files/data_sheets/AD22100.pdf

ATmega16. 2010. Datasheet. Atmel
http://www.atmel.com/Images/doc2466.pdf

Bausatz "USBasp".
http://www.fundf.net/usbasp/

BSP250. 1997. Datasheet. NXP
http://www.nxp.com/documents/data_sheet/BSP250.pdf

Burning hex files using USBasp and AVRdude. 2009.
http://elecrom.wordpress.com/2009/04/15/avrdude-tutorial-burning-hex-files-using-usbasp-and-avrdude/

ExpressPCB. 2010
http://www.expresspcb.com/expresspcbhtm/Free_cad_software.htm

eXtreme Burner – AVR. 2009. GUI Software for USBasp based USB AVR Programmers.
http://extremeelectronics.co.in/avr-tutorials/gui-software-for-usbasp-based-usb-avr-programmers/

HD44780 assembly driver.
http://en.radzio.dxp.pl/avr-mcu/hd44780-assembly-driver.html

How to Control Motor Speed with a PWM Circuit. Robot room
http://www.robotroom.com/PWM5.html

How To Make PCB using Toner Transfer method in 7 simple steps. 2011
http://www.youtube.com/watch?v=bk6WJpGyc4I

Molex 4 pin Peripheral Power Connector Pinout. Tim Fisher.
http://pcsupport.about.com/od/insidethepc/a/molex-pinout-4-pin-psu.htm

NTC Thermistors, Radial Leaded, Accuracy Line. 2011. Datasheet. Vishay bc components.
http://www.farnell.com/datasheets/1444022.pdf

Power connector, 4 circuits, vertical and right angle. 2010. molex 8981 series.
http://www.molex.com/pdm_docs/ps/PS-8981-4V.pdf

README file for USBasp. 2011
http://www.fischl.de/usbasp/Readme.txt

Studio Archive. 2012. Atmel.
http://www.atmel.com/tools/studioarchive.aspx

USBasp. USB programmer for Atmel AVR controllers.
http://www.fischl.de/usbasp/

What is a Hitachi HD44780 LCD Display. 2009. Ozirocks Projects
http://ozirock.webs.com/hitachihd44780lcd.htm

LIITE1: Lähdekoodi

```
.INCLUDE "m16def.inc"
.INCLUDE "HD44780.inc" ;Sisältää määrittelyjä näytön alustukseen.

;-------------------------------------------------------------------------------
;Rekisterimuistit
;Rekisterimuisteissa suoritetaan kaikki ohjelman toimenpiteet
;Annetaan rekistereille uusia nimiä
;-------------------------------------------------------------------------------

.def fan1h = R0
.def fan1l = R1
.def fan2h = R2
.def fan2l = R3
.def fan3h = R4
.def fan3l = R5
.def fan4h = R6
.def fan4l = R7
.def fan5h = R8
.def fan5l = R9
.def fan6h = R10
.def fan6l = R11
.def fan7h = R12
.def fan7l = R13
.def fan8h = R14
.def fan8l = R15
.def temp1 = R16
.def temp2 = R17
.def temp3 = R18
.def temp4 = R19
.def temp5 = R20
.def temp6 = R21
.def temp7 = R22
.def temp8 = R23
.def temp9 = R24
.def temp10 = R25

;-------------------------------------------------------------------------------
;SRAM muistit
;ATMEGA 16:ssa on 1KB sram ja se on nimensä mukaisesti RAM tyyppistä
;SRAM muisti toimii nopeana lisämuistina muuttujille
;Annetaan SRAM muisti osoitteille nimiä
;-------------------------------------------------------------------------------

.equ fan1rpmH = 0x0060
.equ fan1rpmL = 0x0061
.equ fan2rpmH = 0x0062
.equ fan2rpmL = 0x0063
.equ fan3rpmH = 0x0064
.equ fan3rpmL = 0x0065
.equ fan4rpmH = 0x0066
.equ fan4rpmL = 0x0067
.equ fan5rpmH = 0x0068
.equ fan5rpmL = 0x0069
.equ fan6rpmH = 0x006A
.equ fan6rpmL = 0x006B
.equ fan7rpmH = 0x006C
.equ fan7rpmL = 0x006D
.equ fan8rpmH = 0x006E
.equ fan8rpmL = 0x006F
.equ fanready = 0x0070
.equ fanstate = 0x0071
```

```
.equ Temp1h = 0x0072
.equ Temp1l = 0x0073
.equ Temp2h = 0x0074
.equ Temp2l = 0x0075
.equ Temp3h = 0x0076
.equ Temp3l = 0x0077
.equ Temp4h = 0x0078
.equ Temp4l = 0x0079
.equ Buttonstate = 0x007A
.equ Pagestate = 0x007B
.equ PWM_1 = 0x007C
.equ PWM_2 = 0x007D
.equ PWM_3 = 0x007E
.equ PWM_4 = 0x007F
.equ SetupSensor = 0x0080
.equ Setuptype = 0x0081
.equ SetupPWM = 0x0082
.equ SetupTempLH = 0x0083
.equ setuptempLL = 0x0084
.equ SetupTempHH = 0x0085
.equ SetupTempHL = 0x0086
.equ PWM_1_sensor = 0x0087
.equ PWM_1_min = 0x0088
.equ PWM_1_lowH = 0x0089
.equ PWM_1_lowL = 0x008A
.equ PWM_1_highH = 0x008B
.equ PWM_1_highL = 0x008C
.equ PWM_2_sensor = 0x008D
.equ PWM_2_min = 0x008E
.equ PWM_2_lowH = 0x008F
.equ PWM_2_lowL = 0x0090
.equ PWM_2_highH = 0x0091
.equ PWM_2_highL = 0x0092
.equ PWM_3_sensor = 0x0093
.equ PWM_3_min = 0x0094
.equ PWM_3_lowH = 0x0095
.equ PWM_3_lowL = 0x0096
.equ PWM_3_highH = 0x0097
.equ PWM_3_highL = 0x0098
.equ PWM_4_sensor = 0x009A
.equ PWM_4_min = 0x009B
.equ PWM_4_lowH = 0x009C
.equ PWM_4_lowL = 0x009D
.equ PWM_4_highH = 0x009E
.equ PWM_4_highL = 0x009F
.equ ButtonCounter = 0x00A0
.equ ButtonEffect = 0x00A1
.equ SetupPage = 0x00A2
.equ PWM_counterL = 0x00A3
.equ PWM_counterH = 0x00A4

;-------------------------------------------------------------------------------
;EEPROM osoitteita
;EEPROM:lle tallennetaan tuulettimien säätöparametrit
;EEPROM on ROM muistia joten sinne tallennetut tiedot säilyvät myös virran
;katkaisun jälkeen.
;Nimetään EEPROM osoitteita
;-------------------------------------------------------------------------------

.equ EEP_1_sensor = $0000
.equ EEP_1_min = $0001
.equ EEP_1_lowH = $0002
.equ EEP_1_lowL = $0003
.equ EEP_1_highH = $0004
```

```
.equ EEP_1_highL = $0005
.equ EEP_2_sensor = $0006
.equ EEP_2_min = $0007
.equ EEP_2_lowH = $0008
.equ EEP_2_lowL = $0009
.equ EEP_2_highH = $000A
.equ EEP_2_highL = $000B
.equ EEP_3_sensor = $000C
.equ EEP_3_min = $000D
.equ EEP_3_lowH = $000E
.equ EEP_3_lowL = $000F
.equ EEP_3_highH = $0010
.equ EEP_3_highL = $0011
.equ EEP_4_sensor = $0012
.equ EEP_4_min = $0013
.equ EEP_4_lowH = $0014
.equ EEP_4_lowL = $0015
.equ EEP_4_highH = $0016
.equ EEP_4_highL = $0017
.equ EEP_1_M = $0018
.equ EEP_2_M = $0019
.equ EEP_3_M = $001A
.equ EEP_4_M = $001B

;-----------------------------------------------------------------------------
;Nimetään I/O portteja ja pinnejä
;-----------------------------------------------------------------------------

.equ        LCD_PORT = PORTB        ;Nimetään portti johon LCD on kytketty

.equ        LCD_D4 = 4
.equ        LCD_D5 = 5
.equ        LCD_D6 = 6
.equ        LCD_D7 = 7

.equ        LCD_RS = 0
.equ        LCD_EN = 1
.equ        LCD_L1 = 0x80
.equ        LCD_L2 = 0xC0

.equ        S = 0                   ;nappien paikat (setup)
.equ        F = 1                   ;(Next)
.equ        B = 2                   ;(Back)
.equ        P = 3                   ;(Plus)
.equ        M = 6                   ;(minus)

;-----------------------------------------------------------------------------
;FLASH segmentin alku
;FLSH muisti on ohjelmamuisti joten tästä alkaa ohjelma
;Ensimmäisenä määritellään keskeytysvektorit
;-----------------------------------------------------------------------------

.CSEG
.ORG $0000
            rjmp        Setup
            jmp         Setup       ; Reset Handler
            jmp         INT_R       ; IRQ0 Handler
            jmp         INT_R       ; IRQ1 Handler
            jmp         INT_R       ; Timer2 Compare Handler
            jmp         Fan_loop    ; Timer2 Overflow Handler
            jmp         INT_R       ; Timer1 Capture Handler
            jmp         INT_R       ; Timer1 CompareA Handler
            jmp         INT_R       ; Timer1 CompareB Handler
            jmp         INT_R       ; Timer1 Overflow Handler
```

```
            jmp         INT_R       ; Timer0 Overflow Handler
            jmp         INT_R       ; SPI Transfer Complete Handler
            jmp         INT_R       ; USART RX Complete Handler
            jmp         INT_R       ; UDR Empty Handler
            jmp         INT_R       ; USART TX Complete Handler
            jmp         INT_R       ; ADC Conversion Complete Handler
            jmp         INT_R       ; EEPROM Ready Handler
            jmp         INT_R       ; Analog Comparator Handler
            jmp         INT_R       ; Two-wire Serial Interface Handler
            jmp         INT_R       ; IRQ2 Handler
            jmp         INT_R       ; Timer0 Compare Handler
            jmp         INT_R       ; Store Program Memory Ready Handler

;Paluu käyttämättömiltä keskeytysvektoreilta
INT_R:
            reti

;-----------------------------------------------------------------------------
;Määritellään mikrokontrollerin asetukset
;-----------------------------------------------------------------------------

Setup:
            cli
            ldi         temp1,high(ramend)
            out         sph,temp1
            ldi         temp1,low(ramend)
            out         spl,temp1
            ldi         temp1,0b10000000        ;otetaan Jtag pois käytöstä
            out         MCUCSR,temp1            ;mcucsr rekisteri täyty
            nop                                 ;kirjoittaa kahdesti 4
            out         MCUCSR,temp1            ;kellojaksossa
            ldi         temp1,0b10110000        ;asetetaan I/O väylien suunnat
            out         DDRD,temp1              ;1=ulos 0= sisään
            ldi         temp1,0x00
            out         DDRC,temp1
            ldi         temp1,0b11111011
            out         DDRB,temp1
            ldi         temp1,0x00
            out         DDRA,temp1
            ldi         temp1,0b01100001        ;asetetaan Timer0 PWM
            out         TCCR0,temp1
            ldi         temp1,0b10100001        ;asetetaan timer 1:sen pwm:t
            out         TCCR1A,temp1            ;PWM:t ovat käytössä 8bittisinä
            ldi         temp1,0b00000001
            out         TCCR1B,temp1
            ldi         temp1,0b01100001        ;asetetaan Timer2 PWM
            out         TCCR2,temp1
            ldi         temp1,0b01000000        ;asetetaan keskeytys Timer2:lle
            out         TIMSK,temp1
            ldi         temp1,0b01000000        ;asetetaan AD muuntimet
            out         ADMUX,temp1
            ldi         temp1,0b10000111
            out         ADCSRA,temp1
            clr         temp1                   ;tallennetaan 0 tarvittavaiin
            sts         ButtonEffect,temp1      ;sram osoitteisiin
            sts         Pagestate,temp1
            sts         PWM_counterL,temp1
            sts         PWM_counterH,temp1
            ldi         temp1,255               ;annetaan PWM:lle 100% työsuhde
            out         OCR0,temp1
            out         OCR1AL,temp1
            out         OCR1BL,temp1
            out         OCR2,temp1
            ldi         temp2,255               ;odotetaan 255ms
```

```
            rcall       Waitms

;-------------------------------------------------------------------------------
;alustetaan LCD näyttö
;Lähde: http://en.radzio.dxp.pl/avr-mcu/hd44780-assembly-driver.html
;-------------------------------------------------------------------------------

LCD_int:
            cbi         LCD_PORT, LCD_RS
            cbi         LCD_PORT, LCD_EN

            ldi         temp2, 100
            rcall       Waitms

            ldi         temp3, 3

InitLoop:
            ldi         temp1, 0x03
            rcall       LCD_WriteNibble
            ldi         temp2, 5
            rcall       Waitms
            dec         temp3
            brne        InitLoop

            ldi         temp1, 0x02
            rcall       LCD_WriteNibble

            ldi         temp2, 1
            rcall       Waitms

            ldi         temp1, HD44780_FUNCTION_SET | HD44780_FONT5x7 |
HD44780_TWO_LINE | HD44780_4_BIT
            rcall       LCD_WriteCommand

            ldi         temp1, HD44780_DISPLAY_ONOFF | HD44780_DISPLAY_OFF
            rcall       LCD_WriteCommand

            ldi         temp1, HD44780_CLEAR
            rcall       LCD_WriteCommand

            ldi         temp1, HD44780_ENTRY_MODE |HD44780_EM_SHIFT_CURSOR |
HD44780_EM_INCREMENT
            rcall       LCD_WriteCommand

            ldi         temp1, HD44780_DISPLAY_ONOFF | HD44780_DISPLAY_ON |
HD44780_CURSOR_OFF | HD44780_CURSOR_NOBLINK
            rcall       LCD_WriteCommand

;-------------------------------------------------------------------------------
;Ladataan EEPROM muistista tuuletinohjaimen asetukset
;-------------------------------------------------------------------------------

EEPROM_start:
            ldi         temp2,LOW(EEP_1_sensor)
            ldi         temp3,HIGH(EEP_1_sensor)
            call        EEPROM_Read
            sts         PWM_1_sensor,temp1

            ldi         temp2,LOW(EEP_1_min)
            ldi         temp3,HIGH(EEP_1_min)
            call        EEPROM_Read
            sts         PWM_1_min,temp1

            ldi         temp2,LOW(EEP_1_lowH)
```

```
            ldi         temp3,HIGH(EEP_1_lowH)
            call        EEPROM_Read
            sts         PWM_1_lowH,temp1

            ldi         temp2,LOW(EEP_1_lowL)
            ldi         temp3,HIGH(EEP_1_lowL)
            call        EEPROM_Read
            sts         PWM_1_lowL,temp1

            ldi         temp2,LOW(EEP_1_highH)
            ldi         temp3,HIGH(EEP_1_highH)
            call        EEPROM_Read
            sts         PWM_1_highH,temp1

            ldi         temp2,LOW(EEP_1_highL)
            ldi         temp3,HIGH(EEP_1_highL)
            call        EEPROM_Read
            sts         PWM_1_highL,temp1

            ldi         temp2,LOW(EEP_1_M)
            ldi         temp3,HIGH(EEP_1_M)
            call        EEPROM_Read
            sts         PWM_1,temp1

;PWM2
            ldi         temp2,LOW(EEP_2_sensor)
            ldi         temp3,HIGH(EEP_2_sensor)
            call        EEPROM_Read
            sts         PWM_2_sensor,temp1

            ldi         temp2,LOW(EEP_2_min)
            ldi         temp3,HIGH(EEP_2_min)
            call        EEPROM_Read
            sts         PWM_2_min,temp1

            ldi         temp2,LOW(EEP_2_lowH)
            ldi         temp3,HIGH(EEP_2_lowH)
            call        EEPROM_Read
            sts         PWM_2_lowH,temp1

            ldi         temp2,LOW(EEP_2_lowL)
            ldi         temp3,HIGH(EEP_2_lowL)
            call        EEPROM_Read
            sts         PWM_2_lowL,temp1

            ldi         temp2,LOW(EEP_2_highH)
            ldi         temp3,HIGH(EEP_2_highH)
            call        EEPROM_Read
            sts         PWM_2_highH,temp1

            ldi         temp2,LOW(EEP_2_highL)
            ldi         temp3,HIGH(EEP_2_highL)
            call        EEPROM_Read
            sts         PWM_2_highL,temp1

            ldi         temp2,LOW(EEP_2_M)
            ldi         temp3,HIGH(EEP_2_M)
            call        EEPROM_Read
            sts         PWM_2,temp1

;PWM3
            ldi         temp2,LOW(EEP_3_sensor)
            ldi         temp3,HIGH(EEP_3_sensor)
            call        EEPROM_Read
```

```
            sts         PWM_3_sensor,temp1

            ldi         temp2,LOW(EEP_3_min)
            ldi         temp3,HIGH(EEP_3_min)
            call        EEPROM_Read
            sts         PWM_3_min,temp1

            ldi         temp2,LOW(EEP_3_lowH)
            ldi         temp3,HIGH(EEP_3_lowH)
            call        EEPROM_Read
            sts         PWM_3_lowH,temp1

            ldi         temp2,LOW(EEP_3_lowL)
            ldi         temp3,HIGH(EEP_3_lowL)
            call        EEPROM_Read
            sts         PWM_3_lowL,temp1

            ldi         temp2,LOW(EEP_3_highH)
            ldi         temp3,HIGH(EEP_3_highH)
            call        EEPROM_Read
            sts         PWM_3_highH,temp1

            ldi         temp2,LOW(EEP_3_highL)
            ldi         temp3,HIGH(EEP_3_highL)
            call        EEPROM_Read
            sts         PWM_3_highL,temp1

            ldi         temp2,LOW(EEP_3_M)
            ldi         temp3,HIGH(EEP_3_M)
            call        EEPROM_Read
            sts         PWM_3,temp1

;PWM4
            ldi         temp2,LOW(EEP_4_sensor)
            ldi         temp3,HIGH(EEP_4_sensor)
            call        EEPROM_Read
            sts         PWM_4_sensor,temp1

            ldi         temp2,LOW(EEP_4_min)
            ldi         temp3,HIGH(EEP_4_min)
            call        EEPROM_Read
            sts         PWM_4_min,temp1

            ldi         temp2,LOW(EEP_4_lowH)
            ldi         temp3,HIGH(EEP_4_lowH)
            call        EEPROM_Read
            sts         PWM_4_lowH,temp1

            ldi         temp2,LOW(EEP_4_lowL)
            ldi         temp3,HIGH(EEP_4_lowL)
            call        EEPROM_Read
            sts         PWM_4_lowL,temp1

            ldi         temp2,LOW(EEP_4_highH)
            ldi         temp3,HIGH(EEP_4_highH)
            call        EEPROM_Read
            sts         PWM_4_highH,temp1

            ldi         temp2,LOW(EEP_4_highL)
            ldi         temp3,HIGH(EEP_4_highL)
            call        EEPROM_Read
            sts         PWM_4_highL,temp1

            ldi         temp2,LOW(EEP_4_M)
```

```
            ldi         temp3,HIGH(EEP_4_M)
            call        EEPROM_Read
            sts         PWM_4,temp1
            rjmp        main

;-------------------------------------------------------------------------------
;Taulukoita näytölle kirjoittamista varten
;-------------------------------------------------------------------------------

PageTable:
.db         " FAN1  FAN2 PWM1"
.db         " FAN3  FAN4 PWM2"
.db         " FAN5  FAN6 PWM3"
.db         " FAN7  FAN8 PWM4"
TempTable:
.db         " Temp1   Temp2  "
.db         " Temp3   Temp4  "

;-------------------------------------------------------------------------------
;pääohjelma
;-------------------------------------------------------------------------------

Main:
            jmp         FAN
Continue:
            call        Temperature
            call        Button
            sbrc        temp9,0
            jmp         PWM_Setup
            call        PWM
            call        Write
            rjmp        Main

;-------------------------------------------------------------------------------
;Write aliohjelmaan kuuluva osa joka tarkistaa nappien toimintaa että pitääkö
;vaihtaa sivua
;-------------------------------------------------------------------------------

Page_shift:
            sbrc        temp4,F
            inc         temp3
            sbrc        temp4,B
            dec         temp3
            cpi         temp3,6
            brsh        Page_Shift_fix
            sts         PageState,temp3
            ret
Page_Shift_fix:
            sbrc        temp4,F
            ldi         temp3,0
            sbrc        temp4,B
            ldi         temp3,5
            sts         PageState,temp3
            ret

;-------------------------------------------------------------------------------
;Write aliohjelma kirjoittaa LCD näytölle haluttavan datan
;-------------------------------------------------------------------------------

Write:
            lds         temp3,PageState
            lds         temp4,ButtonEffect
            call        Page_shift
            lds         temp10,fanready
```

```
            cpi         temp3,0
            breq        Page1
            cpi         temp3,1
            breq        Page2
            jmp         WriteA
            ret

Page1:
            ldi         temp1,LCD_L1
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(PageTable*2)
            ldi         ZL,Low(PageTable*2)
            rcall       Write_loop

            ldi         temp1,LCD_L2
            rcall       LCD_writecommand
            lds         YH,fan1rpmh
            lds         YL,fan1rpml
            sbrs        temp10,0
            call        NA
            sbrc        temp10,0
            call        write_rpm
            ldi         temp1,0x20
            rcall       LCD_Writedata

            lds         YH,fan2rpmh
            lds         YL,fan2rpml
            sbrs        temp10,1
            call        NA
            sbrc        temp10,1
            call        write_rpm

            ldi         temp1,0x20
            rcall       LCD_Writedata
            clr         YH
            lds         YL,PWM_1
            call        Write_PWM
            ret

Page2:
            ldi         temp1,LCD_L1
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(PageTable*2)
            ldi         ZL,Low(PageTable*2)
            adiw        ZL,16
            rcall       Write_loop

            ldi         temp1,LCD_L2
            rcall       LCD_writecommand
            lds         YH,fan3rpmh
            lds         YL,fan3rpml
            sbrs        temp10,2
            call        NA
            sbrc        temp10,2
            call        write_rpm
            ldi         temp1,0x20
            rcall       LCD_Writedata

            lds         YH,fan4rpmh
            lds         YL,fan4rpml
            sbrs        temp10,3
            call        NA
```

```
            sbrc        temp10,3
            call        write_rpm

            ldi         temp1,0x20
            rcall       LCD_Writedata
            clr         YH
            lds         YL,PWM_2
            call        Write_PWM
            ret

Page3:
            ldi         temp1,LCD_L1
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(PageTable*2)
            ldi         ZL,Low(PageTable*2)
            adiw        ZL,32
            rcall       Write_loop

            ldi         temp1,LCD_L2
            rcall       LCD_writecommand
            lds         YH,fan5rpmh
            lds         YL,fan5rpml
            sbrs        temp10,4
            call        NA
            sbrc        temp10,4
            call        write_rpm
            ldi         temp1,0x20
            rcall       LCD_Writedata

            lds         YH,fan6rpmh
            lds         YL,fan6rpml
            sbrs        temp10,5
            call        NA
            sbrc        temp10,5
            call        write_rpm

            ldi         temp1,0x20
            rcall       LCD_Writedata
            clr         YH
            lds         YL,PWM_3
            call        Write_PWM
            ret


WriteA:
            cpi         temp3,2
            breq        Page3
            cpi         temp3,3
            breq        Page4
            cpi         temp3,4
            breq        Page5
            rjmp        WriteB

Page4:
            ldi         temp1,LCD_L1
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(PageTable*2)
            ldi         ZL,Low(PageTable*2)
            adiw        ZL,48
            rcall       Write_loop

            ldi         temp1,LCD_L2
```

```
            rcall       LCD_writecommand
            lds         YH,fan7rpmh
            lds         YL,fan7rpml
            sbrs        temp10,6
            call        NA
            sbrc        temp10,6
            call        write_rpm
            ldi         temp1,0x20
            rcall       LCD_Writedata

            lds         YH,fan8rpmh
            lds         YL,fan8rpml
            sbrs        temp10,7
            call        NA
            sbrc        temp10,7
            call        write_rpm

            ldi         temp1,0x20
            rcall       LCD_Writedata
            clr         YH
            lds         YL,PWM_4
            call        Write_PWM
            ret
;Sivut 5 ja 6 näyttävät lämpötila-antuerien arvot
Page5:
            ldi         temp1,LCD_L1
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(TempTable*2)
            ldi         ZL,Low(TempTable*2)
            rcall       Write_loop

            ldi         temp1,LCD_L2
            rcall       LCD_writecommand
            lds         YH,Temp1h
            lds         YL,Temp1l
            call        Write_temp
            ldi         temp1,0x20
            rcall       LCD_Writedata
            rcall       LCD_Writedata
            lds         YH,Temp2h
            lds         YL,Temp2l
            call        Write_temp
            ldi         temp1,0x20
            rcall       LCD_Writedata
            rcall       LCD_Writedata
            ret

Page6:
            ldi         temp1,LCD_L1
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(TempTable*2)
            ldi         ZL,Low(TempTable*2)
            adiw        ZL,16
            rcall       Write_loop

            ldi         temp1,LCD_L2
            rcall       LCD_writecommand
            lds         YH,Temp3h
            lds         YL,Temp3l
            call        Write_temp
            ldi         temp1,0x20
            rcall       LCD_Writedata
```

```
            rcall       LCD_Writedata
            lds         YH,Temp4h
            lds         YL,Temp4l
            call        Write_temp
            ldi         temp1,0x20
            rcall       LCD_Writedata
            rcall       LCD_Writedata
            ret

WriteB:
            cpi         temp3,5
            breq        Page6
            ret

NA:
            ldi         temp1,0x20
            rcall       LCD_Writedata
            rcall       LCD_Writedata
            ldi         temp1,0x4E
            rcall       LCD_Writedata
            ldi         temp1,0x2F
            rcall       LCD_Writedata
            ldi         temp1,0x41
            rcall       LCD_Writedata
            ret


Write_rpm:
            call        numbers
            cpi         temp3,1
            brsh        rpm_jmp
            ldi         temp3,0xF0
rpm_jmp:
            mov         temp1,temp3
            subi        temp1,208
            rcall       LCD_Writedata
            mov         temp1,temp4
            subi        temp1,208
            rcall       LCD_Writedata
            mov         temp1,temp5
            subi        temp1,208
            rcall       LCD_Writedata
            mov         temp1,temp6
            subi        temp1,208
            rcall       LCD_Writedata
            ldi         temp1,0x72
            rcall       LCD_Writedata
            ret

;-------------------------------------------------------------------------------
;Ohjelma kirjoittaa LCD näytön portille puolikkaan tavun
;-------------------------------------------------------------------------------

LCD_WriteNibble:
            sbi         LCD_PORT, LCD_EN

            sbrs        temp1, 0
            cbi         LCD_PORT, LCD_D4
            sbrc        temp1, 0
            sbi         LCD_PORT, LCD_D4

            sbrs        temp1, 1
            cbi         LCD_PORT, LCD_D5
            sbrc        temp1, 1
```

```
            sbi         LCD_PORT, LCD_D5

            sbrs        temp1, 2
            cbi         LCD_PORT, LCD_D6
            sbrc        temp1, 2
            sbi         LCD_PORT, LCD_D6

            sbrs        temp1, 3
            cbi         LCD_PORT, LCD_D7
            sbrc        temp1, 3
            sbi         LCD_PORT, LCD_D7

            cbi         LCD_PORT, LCD_EN
            ret

;-------------------------------------------------------------------------------
;Ohjelma näytölle kirjoittamista varten
;-------------------------------------------------------------------------------

LCD_WriteData:
            sbi         LCD_PORT, LCD_RS
            push        temp1
            swap        temp1
            rcall       LCD_WriteNibble
            pop         temp1
            rcall       LCD_WriteNibble

            clr         YH
            ldi         YL,0b10000000
            rcall       Wait
            ret

;-------------------------------------------------------------------------------
;Ohjelma komentojen kirjoittamista varten esim. kursorin paikka, ruudun tyhjennys
;jane..
;-------------------------------------------------------------------------------

LCD_WriteCommand:
            cbi         LCD_PORT, LCD_RS
            push        temp1
            swap        temp1
            rcall       LCD_WriteNibble
            pop         temp1
            rcall       LCD_WriteNibble
            ldi         temp2,2
            rcall       Waitms
            ret

Write_loop:
            lpm         temp1,Z+
            rcall       LCD_Writedata
            dec         temp3
            cpi         temp3,1
            brsh        write_loop
            ret

;-------------------------------------------------------------------------------
;ohjelma odottamista varten
;-------------------------------------------------------------------------------

Wait:
            sbiw        YH:YL,1
            brne        Wait
            ret
```

```
;-------------------------------------------------------------------------------
;millisekunteja odottava ohjelma
;-------------------------------------------------------------------------------

Waitms:
            ldi         YH,0x0b
            ldi         YL,0xb8
Count:
            sbiw        YH:YL,1
            brne        Count
            dec         temp2
            brne        Waitms
            ret

;-------------------------------------------------------------------------------
;Aliohjelma joka jakaa 16bit luvun tuhansiin, satoihin, kymmeniin ja ykkösiin
;jotta luku voidaan kirjoittaa oikeassa muodossa lcd näytölle
;-------------------------------------------------------------------------------

Numbers:
            clr         temp3
            clr         temp4
            clr         temp5
            clr         temp6
            ldi         temp7,0b00000011
Thousand:
            cpi         YL,0b11101000
            cpc         YH,temp7
            brcs        Hundred_setup
            subi        YL,0b11101000
            sbci        YH,0b00000011
            inc         temp3
            rjmp        Thousand
Hundred_setup:
            ldi         temp7,0x00
Hundred:
            cpi         YL,0b01100100
            cpc         YH,temp7
            brcs        Ten
            subi        YL,0b01100100
            sbci        YH,0b00000000
            inc         temp4
            rjmp        Hundred
Ten:
            cpi         YL,0b00001010
            brcs        one
            subi        YL,10
            inc         temp5
            rjmp        ten
One:
            cpi         YL,0b00000001
            brcs        Return
            subi        YL,1
            inc         temp6
            rjmp        one

Return:
            ret

;-------------------------------------------------------------------------------
;Ohjelma tuulettimien kierrosnopeuden lukemiseen.
;-------------------------------------------------------------------------------
```

```
FAN:
            in          temp2,PINC   ;luetaan porttiC:n tila reksiteriin
            sts         Fanstate,temp2
            clr         temp3        ;tyhjennetään tarvittavat rekisterit
            clr         temp4
            clr         XL
            clr         XH
            sei                      ;keskeytykset aktiiviseksi
            jmp         FAN_check

;-------------------------------------------------------------------------------
;Ohjelma looppi johon keskeytysvektori tulee.
;-------------------------------------------------------------------------------

FAN_loop:
            in          temp1,PINC
            lds         temp2,fanstate
            sts         fanstate,temp1
            eor         temp1,temp2
            ldi         temp9,0x28
            ldi         temp10,0x00

            adiw        XL,1

;-------------------------------------------------------------------------------
;Ohjelmaosio joka laskee tuulettimilta tulevaa pulssinleveyttä
;-------------------------------------------------------------------------------

FAN1:
            sbrs        temp1,0
            brcc        FAN2
            sbrc        temp4,0
            rjmp        FAN1_sub
            sbr         temp4,0b00000001
            mov         fan1l,XL
            mov         fan1h,XH
FAN1_sub:
            sbrc        temp3,0
            rjmp        FAN2
            mov         temp5,XL
            mov         temp6,XH
            sub         temp5,fan1l
            sbc         temp6,fan1h
            cp          temp5,temp9
            cpc         temp6,temp10
            brlo        FAN2
            mov         fan1l,temp5
            mov         fan1h,temp6
            sbr         temp3,0b00000001


FAN2:
            sbrs        temp1,1
            brcc        FAN3
            sbrc        temp4,1
            rjmp        FAN2_sub
            sbr         temp4,0b00000010
            mov         fan2l,XL
            mov         fan2h,XH
            rjmp        FAN3
FAN2_sub:
            sbrc        temp3,1
            rjmp        FAN3
            mov         temp5,XL
```

```
            mov         temp6,XH
            sub         temp5,fan2l
            sbc         temp6,fan2h
            cp          temp5,temp9
            cpc         temp6,temp10
            brlo        FAN3
            mov         fan2l,temp5
            mov         fan2h,temp6
            sbr         temp3,0b00000010

FAN3:
            sbrs        temp1,2
            brcc        FAN4
            sbrc        temp4,2
            rjmp        FAN3_sub
            sbr         temp4,0b00000100
            mov         fan3l,XL
            mov         fan3h,XH
            rjmp        FAN4
FAN3_sub:
            sbrc        temp3,2
            rjmp        FAN4
            mov         temp5,XL
            mov         temp6,XH
            sub         temp5,fan3l
            sbc         temp6,fan3h
            cp          temp5,temp9
            cpc         temp6,temp10
            brlo        FAN4
            mov         fan3l,temp5
            mov         fan3h,temp6
            sbr         temp3,0b00000100

FAN4:
            sbrs        temp1,3
            brcc        FAN5
            sbrc        temp4,3
            rjmp        FAN4_sub
            sbr         temp4,0b00001000
            mov         fan4l,XL
            mov         fan4h,XH
            rjmp        FAN5
FAN4_sub:
            sbrc        temp3,3
            rjmp        FAN5
            mov         temp5,XL
            mov         temp6,XH
            sub         temp5,fan4l
            sbc         temp6,fan4h
            cp          temp5,temp9
            cpc         temp6,temp10
            brlo        FAN5
            mov         fan4l,temp5
            mov         fan4h,temp6
            sbr         temp3,0b00001000

FAN5:
            sbrs        temp1,4
            brcc        FAN6
            sbrc        temp4,4
            rjmp        FAN5_sub
            sbr         temp4,0b00010000
            mov         fan5l,XL
            mov         fan5h,XH
```

```
            rjmp        FAN6
FAN5_sub:
            sbrc        temp3,4
            rjmp        FAN6
            mov         temp5,XL
            mov         temp6,XH
            sub         temp5,fan5l
            sbc         temp6,fan5h
            cp          temp5,temp9
            cpc         temp6,temp10
            brlo        FAN6
            mov         fan5l,temp5
            mov         fan5h,temp6
            sbr         temp3,0b00010000

FAN6:
            sbrs        temp1,5
            brcc        FAN7
            sbrc        temp4,5
            rjmp        FAN6_sub
            sbr         temp4,0b00100000
            mov         fan6l,XL
            mov         fan6h,XH
            rjmp        FAN7
FAN6_sub:
            sbrc        temp3,5
            rjmp        FAN7
            mov         temp5,XL
            mov         temp6,XH
            sub         temp5,fan6l
            sbc         temp6,fan6h
            cp          temp5,temp9
            cpc         temp6,temp10
            brlo        FAN7
            mov         fan6l,temp5
            mov         fan6h,temp6
            sbr         temp3,0b00100000

FAN7:
            sbrs        temp1,6
            brcc        FAN8
            sbrc        temp4,6
            rjmp        FAN7_sub
            sbr         temp4,0b01000000
            mov         fan7l,XL
            mov         fan7h,XH
            rjmp        FAN8
FAN7_sub:
            sbrc        temp3,6
            rjmp        FAN8
            mov         temp5,XL
            mov         temp6,XH
            sub         temp5,fan7l
            sbc         temp6,fan7h
            cp          temp5,temp9
            cpc         temp6,temp10
            brlo        FAN8
            mov         fan7l,temp5
            mov         fan7h,temp6
            sbr         temp3,0b01000000

FAN8:
            sbrs        temp1,7
            brcc        FAN_reti
```

```
            sbrc        temp4,7
            rjmp        FAN8_sub
            sbr         temp4,0b10000000
            mov         fan8l,XL
            mov         fan8h,XH
            rjmp        FAN_reti
FAN8_sub:
            sbrc        temp3,7
            rjmp        FAN_reti
            mov         temp5,XL
            mov         temp6,XH
            sub         temp5,fan8l
            sbc         temp6,fan8h
            cp          temp5,temp9
            cpc         temp6,temp10
            brlo        FAN_reti
            mov         fan8l,temp5
            mov         fan8h,temp6
            sbr         temp3,0b10000000
FAN_reti:
            reti
;-------------------------------------------------------------------------------
;Tarkastetaan että tuuletinta on luettu riittävän kauan.
;Tuulettimien minimikierrosnopeus on 200rpm
;-------------------------------------------------------------------------------

FAN_check:
            cpi         XH,0b00001101
            breq        FAN_final
            jmp         FAN_check

;-------------------------------------------------------------------------------
;Muutetaan tuulettimien pulssinleveys muotoon RPM ja tallennetaan nopeus SRAM
;muistiin.
;-------------------------------------------------------------------------------

FAN_Final:
            cli
            sts         fanready,temp3
            sbrs        temp3,0
            rjmp        FAN2_final
            mov         temp7,fan1l
            mov         temp6,fan1h
            call        Fan_Div
            lds         temp1,fan1rpmH
            lds         temp2,fan1rpmL
            call        Sampling
            sts         fan1rpmH,temp9
            sts         fan1rpmL,temp10

FAN2_final:
            lds         temp3,fanready
            sbrs        temp3,1
            rjmp        FAN3_final
            mov         temp7,fan2l
            mov         temp6,fan2h
            call        Fan_Div
            lds         temp1,fan2rpmH
            lds         temp2,fan2rpmL
            call        Sampling
            sts         fan2rpmH,temp9
            sts         fan2rpmL,temp10

FAN3_final:
```

```
            lds         temp3,fanready
            sbrs        temp3,2
            rjmp        FAN4_final
            mov         temp7,fan3l
            mov         temp6,fan3h
            call        Fan_Div
            lds         temp1,fan3rpmH
            lds         temp2,fan3rpmL
            call        Sampling
            sts         fan3rpmH,temp9
            sts         fan3rpmL,temp10

FAN4_final:
            lds         temp3,fanready
            sbrs        temp3,3
            rjmp        FAN5_final
            mov         temp7,fan4l
            mov         temp6,fan4h
            call        Fan_Div
            lds         temp1,fan4rpmH
            lds         temp2,fan4rpmL
            call        Sampling
            sts         fan4rpmH,temp9
            sts         fan4rpmL,temp10

FAN5_final:
            lds         temp3,fanready
            sbrs        temp3,4
            rjmp        FAN6_final
            mov         temp7,fan5l
            mov         temp6,fan5h
            call        Fan_Div
            lds         temp1,fan5rpmH
            lds         temp2,fan5rpmL
            call        Sampling
            sts         fan5rpmH,temp9
            sts         fan5rpmL,temp10

FAN6_final:
            lds         temp3,fanready
            sbrs        temp3,5
            rjmp        FAN7_final
            mov         temp7,fan6l
            mov         temp6,fan6h
            call        Fan_Div
            lds         temp1,fan6rpmH
            lds         temp2,fan6rpmL
            call        Sampling
            sts         fan6rpmH,temp9
            sts         fan6rpmL,temp10

FAN7_final:
            lds         temp3,fanready
            sbrs        temp3,6
            rjmp        FAN8_final
            mov         temp7,fan7l
            mov         temp6,fan7h
            call        Fan_Div
            lds         temp1,fan7rpmH
            lds         temp2,fan7rpmL
            call        Sampling
            sts         fan7rpmH,temp9
            sts         fan7rpmL,temp10
```

```
FAN8_final:
            lds         temp3,fanready
            sbrs        temp3,7
            rjmp        FAN_ready
            mov         temp7,fan8l
            mov         temp6,fan8h
            call        Fan_Div
            lds         temp1,fan8rpmH
            lds         temp2,fan8rpmL
            call        Sampling
            sts         fan8rpmH,temp9
            sts         fan8rpmL,temp10
FAN_ready:
            jmp         Continue

;-------------------------------------------------------------------------------
;Tuulettimien pulssin leveys täytyy muuttaa muotoon RPM(kierrosta minuutissa).
;Tuulettimia luetaan 23529hz näytteenottotaajuudella ja tilamuutoksia
;(1->0 tai 0->1) tulee kierrokselta neljä.
;Muunnos yhtälö voidaan kirjoittaa mutooon 22329hz/X*(60s/4)=RPM.
;X=mitattu pulssin leveys
;hieman yksinkertaistamalla kaavaa voidaan se pyöritellä muotoon 352941/X=RPM
;lähde http://www.avr-asm-tutorial.net/avr_en/source/DIV8E.asm
;-------------------------------------------------------------------------------

Fan_Div:
            clr         temp1
            clr         temp2
            ldi         temp3,0b00000101
            ldi         temp4,0b01100010
            ldi         temp5,0b10101101
            clr         temp8
            clr         temp9
            clr         temp10
            inc         temp10

DivA:
            clc
            rol         temp5
            rol         temp4
            rol         temp3
            rol         temp2
            rol         temp1
            brcs        DivB
            cp          temp2,temp7
            cpc         temp1,temp6
            brcs        DivC
DivB:
            sub         temp2,temp7
            sbc         temp1,temp6
            sec
            rjmp        DivD

DivC:
            clc

DivD:
            rol         temp10
            rol         temp9
            rol         temp8
            brcc        DivA
            ret

;-------------------------------------------------------------------------------
```

```
;suoritetaan samplaus
;1/6 painoarvolla lisätään edelliseen tuulettimen nopeuteen uusi
;--------------------------------------------------------------------------------

Sampling:
            add         temp10,temp2
            adc         temp9,temp1
            ror         temp9
            ror         temp10
            add         temp10,temp2
            adc         temp9,temp1
            ror         temp9
            ror         temp10
            add         temp10,temp2
            adc         temp9,temp1
            ror         temp9
            ror         temp10
            ret

;--------------------------------------------------------------------------------
;Ohjelma lämpötilan mittaamiseen.
;--------------------------------------------------------------------------------

Temperature:
            ldi         temp1,0b01000000
            out         ADMUX,temp1
            sbi         ADCSRA,6
Temp1_read:
            sbic        ADCSRA,6
            rjmp        Temp1_read
            in          temp1,ADCL
            in          temp2,ADCH
            rcall       Temp_con
            sts         temp1l,temp8
            sts         temp1h,temp9

            ldi         temp1,0b01000001
            out         ADMUX,temp1
            sbi         ADCSRA,6
Temp2_read:
            sbic        ADCSRA,6
            rjmp        Temp2_read
            in          temp1,ADCL
            in          temp2,ADCH
            rcall       Temp_con
            sts         temp2l,temp8
            sts         temp2h,temp9

            ldi         temp1,0b01000010
            out         ADMUX,temp1
            sbi         ADCSRA,6
Temp3_read:
            sbic        ADCSRA,6
            rjmp        Temp3_read
            in          temp1,ADCL
            in          temp2,ADCH
            rcall       Temp_con
            sts         temp3l,temp8
            sts         temp3h,temp9

            ldi         temp1,0b01000011
            out         ADMUX,temp1
            sbi         ADCSRA,6
Temp4_read:
```

```
            sbic        ADCSRA,6
            rjmp        Temp4_read
            in          temp1,ADCL
            in          temp2,ADCH
            rcall       Temp_con
            sts         temp4l,temp8
            sts         temp4h,temp9

Temp_end:
            ret

;-----------------------------------------------------------------------------
;Muunnetaan AD muuntimelta luettu arvo lämpötilan muotoon.
;-----------------------------------------------------------------------------
Temp_con:
            ldi         temp6,0b00000010
            ldi         temp5,0b00101100
            clr         temp3
            clr         temp4
            clr         temp7
            clr         temp8
            clr         temp9
MultA:
            clc
            ror         temp6
            ror         temp5
            brcc        MultB
            add         temp7,temp1
            adc         temp8,temp2
            adc         temp9,temp3
MultB:
            clc
            rol         temp1
            rol         temp2
            rol         temp3
            cpi         temp5,0
            breq        temp_con_sub
            rjmp        multA

Temp_con_sub:
            subi        temp8,0b01100100
            sbci        temp9,0b00000010
            brpl        temp_end
            com         temp8
            com         temp9
            sbr         temp9,0x80
            ret

;-----------------------------------------------------------------------------
;Ohjelma lämpötilan kirjoittamiseen näytölle.
;-----------------------------------------------------------------------------
Write_temp:
            sbrc        YH,7
            rjmp        Negative
            call        Numbers
            cpi         temp3,1
            brsh        Write_tempA
            ldi         temp3,0xF0
            cpi         temp4,1
            brsh        Write_tempA
            ldi         temp4,0xF0
Write_tempA:
            mov         temp1,temp3
            subi        temp1,208
```

```
            rcall       LCD_Writedata
            mov         temp1,temp4
            subi        temp1,208
            rcall       LCD_Writedata
            mov         temp1,temp5
            subi        temp1,208
            rcall       LCD_Writedata
            ldi         temp1,0x2E
            rcall       LCD_Writedata
            mov         temp1,temp6
            subi        temp1,208
            rcall       LCD_Writedata
            ldi         temp1,0x43
            rcall       LCD_Writedata
            ret
;jos lämpötila on negatiivinen eteen tulostetaan - merkki
Negative:
            cbr         YH,0b10000000
            ldi         temp9,0x01
            ldi         temp10,0b10010000
            cp          YL,temp10
            cpc         YH,temp9
            brsh        jmp_na
            call        Numbers
            ldi         temp3,0xFD
            cpi         temp4,1
            brsh        Write_tempA
            ldi         temp4,0xF0
            rjmp        Write_tempA
;jos lämpötila on alle -40c tulostetaan N/A (jos anturi ei ole kytketty)
jmp_na:
            call        NA
            ldi         temp1,0x20
            rcall       LCD_Writedata
            ret

;-------------------------------------------------------------------------------
;Aliohjelma nappien lukemiseen
;-------------------------------------------------------------------------------

Button:
            in          temp7,PIND
            com         temp7
            cbr         temp7,0b10110000
            lds         temp8,Buttonstate
            clr         temp9
            cpi         temp7,64
            breq        ButtonRun
            cpi         temp7,8
            breq        ButtonRun
            cpi         temp7,4
            breq        ButtonChange
            cpi         temp7,2
            breq        ButtonChange
            cpi         temp7,1
            breq        ButtonChange
            rjmp        ButtonEnd
ButtonRun:
            cpse        temp7,temp8
            rjmp        ButtonRunB
            lds         temp10,ButtonCounter
            cpse        temp10,temp9
            rjmp        ButtonRunA
            mov         temp9,temp7
```

```
            rjmp        ButtonEnd
ButtonRunA:
            dec         temp10
            sts         ButtonCounter,temp10
            rjmp        ButtonEnd
ButtonRunB:
            ldi         temp10,5
            sts         ButtonCounter,temp10
ButtonChange:
            cpse        temp7,temp8
            mov         temp9,temp7
ButtonEnd:
            sts         ButtonEffect,temp9
            sts         Buttonstate,temp7
            ret

;-------------------------------------------------------------------------------
;aliohjelma PWM arvojen laskemiseen lämpötilasta tai manualisella säädöllä
;-------------------------------------------------------------------------------

PWM:
            lds         temp2,ButtonEffect
            sbrc        temp2,P
            rjmp        PWM_change
            sbrc        temp2,M
            rjmp        PWM_change
            rjmp        PWM_continue

PWM_change:
            lds         temp3,Pagestate
            ldi         temp8,0b00000001
            ldi         temp9,0b00000000
            sts         PWM_counterL,temp9
            sts         PWM_counterH,temp8
            cpi         temp3,0
            breq        PWM1_change
            cpi         temp3,1
            breq        PWM2_change
            cpi         temp3,2
            breq        PWM3_change
            cpi         temp3,3
            breq        PWM4_change
            rjmp        PWM_continue

PWM1_change:
            lds         temp4,PWM_1
            rcall       PWM_shift
            sts         PWM_1,temp4
            rjmp        PWM_continue

PWM2_change:
            lds         temp4,PWM_2
            rcall       PWM_shift
            sts         PWM_2,temp4
            rjmp        PWM_continue

PWM3_change:
            lds         temp4,PWM_3
            rcall       PWM_shift
            sts         PWM_3,temp4
            rjmp        PWM_continue

PWM4_change:
            lds         temp4,PWM_4
```

```
            rcall       PWM_shift
            sts         PWM_4,temp4
            rjmp        PWM_continue

PWM_shift:
            sbrc        temp2,P
            inc         temp4
            sbrc        temp2,M
            dec         temp4
            cpi         temp4,101
            brlo        PWM_shiftA
            sbrc        temp2,P
            ldi         temp4,100
            sbrc        temp2,M
            ldi         temp4,0
PWM_shiftA:
            ret

PWM_continue:
            clr         XL
            lds         YL,PWM_counterL
            lds         YH,PWM_counterH
            ldi         temp9,1
            cp          YL,temp9
            cpc         YH,XL
            breq        PWM_save
            brsh        PWM_saveA
            rjmp        PWM_continueA

PWM_save:
            ldi         XL,1

PWM_saveA:
            sbiw        YL,1
            sts         PWM_counterL,YL
            sts         PWM_counterH,YH

PWM_continueA:
            lds         temp1,PWM_1_sensor
            sbrs        temp1,7
            call        PWM1_manual
            lds         temp1,PWM_1_sensor
            sbrc        temp1,7
            call        PWM1_auto

            lds         temp1,PWM_2_sensor
            sbrs        temp1,7
            call        PWM2_manual
            lds         temp1,PWM_2_sensor
            sbrc        temp1,7
            call        PWM2_auto

            lds         temp1,PWM_3_sensor
            sbrs        temp1,7
            call        PWM3_manual
            lds         temp1,PWM_3_sensor
            sbrc        temp1,7
            call        PWM3_auto

            lds         temp1,PWM_4_sensor
            sbrs        temp1,7
            call        PWM4_manual
            lds         temp1,PWM_4_sensor
            sbrc        temp1,7
```

```
            call        PWM4_auto
            ret

;kirjoitetaan PWM rekistereihin oikeat arvot.
PWM1_manual:
            lds         temp3,PWM_1_min
            lds         temp2,PWM_1
            call        PWM_check
            sts         PWM_1,temp2

            sbrs        XL,0
            rjmp        PWM1_manualA
            push        temp2
            mov         temp1,temp2
            ldi         temp2,LOW(EEP_1_M)
            ldi         temp3,HIGH(EEP_1_M)
            rcall       EEPROM_Write
            pop         temp2

PWM1_manualA:
            call        PWM_mul
            out         OCR2,temp2
            ret

PWM1_auto:
            call        PWM_auto_sensor
            lds         temp2,PWM_1_min
            call        PWM_mul
            lds         temp5,PWM_1_lowH
            lds         temp6,PWM_1_lowL
            lds         temp7,PWM_1_HighH
            lds         temp8,PWM_1_HighL
            call        PWM_compare
            out         OCR2,temp2
            call        PWM_div
            sts         PWM_1,temp8
            ret

PWM2_manual:
            lds         temp3,PWM_2_min
            lds         temp2,PWM_2
            call        PWM_check
            sts         PWM_2,temp2

            sbrs        XL,0
            rjmp        PWM2_manualA
            push        temp2
            mov         temp1,temp2
            ldi         temp2,LOW(EEP_2_M)
            ldi         temp3,HIGH(EEP_2_M)
            rcall       EEPROM_Write
            pop         temp2

PWM2_manualA:
            call        PWM_mul
            out         OCR1BL,temp2
            ret

PWM2_auto:
            call        PWM_auto_sensor
            lds         temp2,PWM_2_min
            call        PWM_mul
            lds         temp5,PWM_2_lowH
            lds         temp6,PWM_2_lowL
```

```
            lds         temp7,PWM_2_HighH
            lds         temp8,PWM_2_HighL
            call        PWM_compare
            out         OCR1BL,temp2
            call        PWM_div
            sts         PWM_2,temp8
            ret

PWM3_manual:
            lds         temp3,PWM_3_min
            lds         temp2,PWM_3
            call        PWM_check
            sts         PWM_3,temp2

            sbrs        XL,0
            rjmp        PWM3_manualA
            push        temp2
            mov         temp1,temp2
            ldi         temp2,LOW(EEP_3_M)
            ldi         temp3,HIGH(EEP_3_M)
            rcall       EEPROM_Write
            pop         temp2

PWM3_manualA:
            call        PWM_mul
            out         OCR1AL,temp2
            ret

PWM3_auto:
            call        PWM_auto_sensor
            lds         temp2,PWM_3_min
            call        PWM_mul
            lds         temp5,PWM_3_lowH
            lds         temp6,PWM_3_lowL
            lds         temp7,PWM_3_HighH
            lds         temp8,PWM_3_HighL
            call        PWM_compare
            out         OCR1AL,temp2
            call        PWM_div
            sts         PWM_3,temp8
            ret

PWM4_manual:
            lds         temp3,PWM_4_min
            lds         temp2,PWM_4
            call        PWM_check
            sts         PWM_4,temp2

            sbrs        XL,0
            rjmp        PWM4_manualA
            push        temp2
            mov         temp1,temp2
            ldi         temp2,LOW(EEP_4_M)
            ldi         temp3,HIGH(EEP_4_M)
            rcall       EEPROM_Write
            pop         temp2

PWM4_manualA:
            call        PWM_mul
            out         OCR0,temp2
            ret

PWM4_auto:
            call        PWM_auto_sensor
```

```
            lds         temp2,PWM_4_min
            call        PWM_mul
            lds         temp5,PWM_4_lowH
            lds         temp6,PWM_4_lowL
            lds         temp7,PWM_4_HighH
            lds         temp8,PWM_4_HighL
            call        PWM_compare
            out         OCR0,temp2
            call        PWM_div
            sts         PWM_4,temp8
            ret


;-------------------------------------------------------------------------------
;PWM_mul aliohjelma kertoo tilan 0-100 siten että se toimii PWM arvona (0-255)
;-------------------------------------------------------------------------------

PWM_mul:
            clr         temp4
            ldi         temp5,0b00000010
            ldi         temp6,0b10001100
            ldi         temp7,0b11001101

            mul         temp2,temp5
            mov         temp10,R0
            mul         temp2,temp6
            mov         temp9,R0
            add         temp10,R1
            mul         temp2,temp7
            add         temp9,R1
            adc         temp10,temp4
            mov         temp2,temp10
            ret

;-------------------------------------------------------------------------------
;PWM_Div aliohjelma jakaa PWM tilan 0-255 siten että se voidaan esittää
;prosentteina eli 0-100
;-------------------------------------------------------------------------------
PWM_Div:
            clr         temp1
            clr         temp3

            ldi         temp5,0b00000010
            ldi         temp6,0b10001100
            ldi         temp7,0b11001101
            clr         temp8
            inc         temp8
            jmp         PWM_DivA

PWM_DivA:
            clc
            rol         temp3
            rol         temp2
            rol         temp1
            brcs        PWM_DivB
            cp          temp3,temp7
            cpc         temp2,temp6
            cpc         temp1,temp5
            brcs        PWM_DivC

PWM_DivB:
            sub         temp3,temp7
            sbc         temp2,temp6
            sbc         temp1,temp5
```

```
            sec

            rjmp        PWM_DivD

PWM_DivC:
            clc


PWM_DivD:
            rol         temp8
            brcc        PWM_DivA
            clc
            ror         temp5
            ror         temp6
            ror         temp7
            cp          temp3,temp7
            cpc         temp2,temp6
            cpc         temp1,temp5
            brcc        PWM_DivE
            ret

PWM_DivE:
            inc         temp8
            ret

PWM_check:
            cp          temp2,temp3
            brsh        PWM_check_ret
            mov         temp2,temp3
PWM_check_ret:
            ret

;-------------------------------------------------------------------------------
;PWM_auto_sensor tarkistaa mitä lämpötilasensoria aliohjelmaa kutsuva PWM ohjaus
;käyttää ja lataa sram muistista tallennetun lämpötilan
;-------------------------------------------------------------------------------

PWM_auto_sensor:
            cpi         temp1,0x80
            breq        sensor1
            cpi         temp1,0x81
            breq        sensor2
            cpi         temp1,0x82
            breq        sensor3
            cpi         temp1,0x83
            breq        sensor4
            ret
Sensor1:
            lds         YH,temp1h
            lds         YL,temp1l
            rjmp        PWM_Auto_sensorA
Sensor2:
            lds         YH,temp2h
            lds         YL,temp2l
            rjmp        PWM_Auto_sensorA
Sensor3:
            lds         YH,temp3h
            lds         YL,temp3l
            rjmp        PWM_Auto_sensorA
Sensor4:
            lds         YH,temp4h
            lds         YL,temp4l
            rjmp        PWM_Auto_sensorA
```

```
;-------------------------------------------------------------------------------
;PWM_Auto_sensorA tarkistaa onko luettu lämpötila-arvo negatiivinen jos on
;annetaan arvoksi 0 astetta
;-------------------------------------------------------------------------------

PWM_Auto_sensorA:
            sbrs        YH,7
            ret
            clr         YH
            clr         YL
            ret

;-------------------------------------------------------------------------------
;PWM_compare suorittaa vertailun mitatun lämpötilan sekä säädetyn lämpötilan
;välillä sekä laskee oikean PWM arvon
;-------------------------------------------------------------------------------

PWM_compare:
            cp          YL,temp6
            cpc         YH,temp5
            brsh        PWM_compare_con
            ret
PWM_compare_con:
            sub         temp8,temp6
            sbc         temp7,temp5
            mov         R10,temp2
            com         temp2
            clr         temp3
            clr         temp4
            clr         temp9
            clr         temp10
            inc         temp10

PWM_com_DivA:
            clc
            rol         temp2
            rol         temp3
            rol         temp4
            brcs        PWM_com_DivB
            cp          temp3,temp8
            cpc         temp4,temp7
            brcs        PWM_com_DivC

PWM_com_DivB:
            sub         temp3,temp8
            sbc         temp4,temp7
            sec

            rjmp        PWM_com_DivD

PWM_com_DivC:
            clc


PWM_com_DivD:
            rol         temp10
            rol         temp9
            brcc        PWM_com_DivA
            rjmp        PWM_com_add

PWM_com_add:
            sub         YL,temp6
            sbc         YH,temp5
```

```
            clr         temp5
            mul         YH,temp9
            mov         temp3,R0
            mov         temp4,R1
            mul         YL,temp10
            mov         temp1,R0
            mov         temp2,R1
            mul         YH,temp10
            add         temp2,R0
            adc         temp3,R1
            adc         temp4,temp5
            mul         YL,temp9
            add         temp2,R0
            adc         temp3,R1
            adc         temp4,temp5
            add         temp2,R10
            adc         temp3,temp5
            adc         temp4,temp5
            cpi         temp3,1
            cpc         temp4,temp5
            brsh        PWM_max
            ret
PWM_max:
            ldi         temp2,255
            ret
;-----------------------------------------------------------------------------
;Write_PWM on aliohjelma joka kirjoittaa näytölle PWM:n arvon prosentteina
;-----------------------------------------------------------------------------

Write_PWM:
            call        Numbers
            clr         temp7
            cp          temp4,temp7
            cpc         temp5,temp7
            breq        Write_PWMB
            cpi         temp4,1
            brsh        Write_PWMA
            ldi         temp4,0xF0
Write_PWMA:
            mov         temp1,temp4
            subi        temp1,208
            rcall       LCD_Writedata
            mov         temp1,temp5
            subi        temp1,208
            rcall       LCD_Writedata
            mov         temp1,temp6
            subi        temp1,208
            rcall       LCD_Writedata
            ldi         temp1,0x25
            rcall       LCD_Writedata
            ret
Write_PWMB:
            ldi         temp4,0xF0
            ldi         temp5,0xF0
            rjmp        Write_PWMA

;-----------------------------------------------------------------------------
;EEPROM komennot joko kirjoittavat tai lukevat EEPROM muistista dataa
;EEPROM_wait odottaa että muisti on valmis
;-----------------------------------------------------------------------------

EEPROM_wait:
            sbic        EECR,EEWE
            rjmp        EEPROM_wait
```

```
            ret

;-------------------------------------------------------------------------------
;EEPROM_write kirjoittaa EEPROM muistiin
;-------------------------------------------------------------------------------

EEPROM_Write:
            cli
            rcall       EEPROM_wait
            out         EEARL,temp2
            out         EEARH,temp3
            out         EEDR,temp1
            SBI         EECR,EEMWE
            SBI         EECR,EEWE
            ret
;-------------------------------------------------------------------------------
;EEPROM_Read lukee EEPROM muistista
;-------------------------------------------------------------------------------

EEPROM_Read:
            cli
            rcall       EEPROM_Wait
            out         EEARL,temp2
            out         EEARH,temp3
            sbi         EECR,EERE
            in          temp1,EEDR
            ret

;-------------------------------------------------------------------------------
;Asetusvalikkojen ohjelma.
;Ensin ladataan muistista sen hetkiset asetukset
;-------------------------------------------------------------------------------

PWM_setup:
            sts         Buttonstate,temp7
            clr         R8
            lds         temp1,Pagestate
            cpi         temp1,0
            breq        PWM1_setup
            cpi         temp1,1
            breq        PWM2_setup
            cpi         temp1,2
            breq        PWM3_setup
            cpi         temp1,3
            breq        PWM4_setup
            jmp         continue
PWM1_setup:
            lds         R2,PWM_1_sensor
            lds         R3,PWM_1_min
            lds         R4,PWM_1_lowH
            lds         R5,PWM_1_lowL
            lds         R6,PWM_1_highH
            lds         R7,PWM_1_highL
            rjmp        Setup_main
PWM2_setup:
            lds         R2,PWM_2_sensor
            lds         R3,PWM_2_min
            lds         R4,PWM_2_lowH
            lds         R5,PWM_2_lowL
            lds         R6,PWM_2_highH
            lds         R7,PWM_2_highL
            rjmp        Setup_main
PWM3_setup:
            lds         R2,PWM_3_sensor
```

```
            lds         R3,PWM_3_min
            lds         R4,PWM_3_lowH
            lds         R5,PWM_3_lowL
            lds         R6,PWM_3_highH
            lds         R7,PWM_3_highL
            rjmp        Setup_main
PWM4_setup:
            lds         R2,PWM_4_sensor
            lds         R3,PWM_4_min
            lds         R4,PWM_4_lowH
            lds         R5,PWM_4_lowL
            lds         R6,PWM_4_highH
            lds         R7,PWM_4_highL
            rjmp        setup_main


;-------------------------------------------------------------------------------
;Asetusvalikon pääohjelma looppi.
;-------------------------------------------------------------------------------

Setup_main:
            call        Button
            mov         temp3,R8
            sbrc        temp9,S
            jmp         PWM_setup_finish
            sbrc        temp9,F
            call        Setup_move
            sbrc        temp9,B
            call        Setup_move
            sbrc        temp9,P
            call        Setup_plus
            sbrc        temp9,M
            call        Setup_minus
            call        Setup_page
            ldi         temp2,80
            call        Waitms
            rjmp        Setup_main

Setup_move:
            sbrs        R2,7
            rjmp        setup_move_manual
            sbrc        temp9,F
            inc         temp3
            sbrc        temp9,B
            dec         temp3
            cpi         temp3,5
            brsh        SP_fixA
            mov         R8,temp3
            ret

Setup_move_manual:
            sbrc        temp9,F
            ldi         temp3,2
            sbrc        temp9,B
            ldi         temp3,0
            mov         R8,temp3
            ret

SP_fixA:
            sbrs        temp9,F
            ldi         temp3,0
            sbrs        temp9,B
            ldi         temp3,4
            mov         R8,temp3
```

```
            ret

PWM_type:
            ldi         temp5,0b10000000
            eor         R2,temp5
            ret

PWM_sensor:
            mov         temp5,R2
            bst         temp5,7
            sbrs        temp4,0
            dec         temp5
            sbrc        temp4,0
            inc         temp5
            cbr         temp5,0b11111100
            bld         temp5,7
            mov         R2,temp5
            ret

PWM_min:
            mov         temp5,R3
            sbrs        temp4,0
            dec         temp5
            sbrc        temp4,0
            inc         temp5
            cpi         temp5,101
            brsh        PWM_min_fix
            mov         R3,temp5
            ret

PWM_min_fix:
            sbrs        temp4,0
            ldi         temp5,0
            sbrc        temp4,0
            ldi         temp5,100
            mov         R3,temp5
            ret

Setup_plus:
            ldi         temp4,1
            rjmp        Setup_plus_minus

Setup_minus:
            ldi         temp4,0

Setup_plus_minus:
            cpi         temp3,0
            breq        PWM_type
            cpi         temp3,1
            breq        PWM_sensor
            cpi         temp3,2
            breq        PWM_min
            cpi         temp3,3
            breq        PWM_temp_low
            cpi         temp3,4
            breq        PWM_temp_high
            ret

PWM_temp_low:
            mov         YL,R5
            mov         YH,R4
            sbrc        temp4,0
            adiw        YH:YL,5
            sbrs        temp4,0
```

```
            sbiw        YH:YL,5
            ldi         temp6,0b11101001
            ldi         temp5,0b00000011
            cp          YL,temp6
            cpc         YH,temp5
            brsh        PWM_temp_low_fix
            mov         R5,YL
            mov         R4,YH
            ret
PWM_temp_low_fix:
            sbrc        temp4,0
            rjmp        PWM_temp_low_fix_up
            sbrs        temp4,0
            rjmp        PWM_temp_low_fix_down
            ret
PWM_temp_low_fix_up:
            mov         R5,temp6
            mov         R4,temp5
            dec         R5
            ret
PWM_temp_low_fix_down:
            clr         R5
            clr         R4
            ret

PWM_temp_high:
            mov         YL,R7
            mov         YH,R6
            sbrc        temp4,0
            adiw        YL,5
            sbrs        temp4,0
            sbiw        YL,5
            cp          R5,YL
            cpc         R4,YH
            brsh        PWM_temp_high_fix_low
            ldi         temp6,0b11101001
            ldi         temp5,0b00000011
            cp          R7,temp6
            cpc         R6,temp5
            brsh        PWM_temp_high_fix
            mov         R7,YL
            mov         R6,YH
            ret
PWM_temp_high_fix:
            sbrc        temp4,0
            rjmp        PWM_temp_high_fix_up
            sbrs        temp4,0
            rjmp        PWM_temp_high_fix_low
            ret
PWM_temp_high_fix_up:
            mov         R7,temp6
            mov         R6,temp5
            dec         R7
            ret
PWM_temp_high_fix_low:
            mov         YL,R5
            mov         YH,R4
            adiw        YH:YL,5
            mov         R7,YL
            mov         R6,YH
            ret

Setup_finish_temp_fix:
            mov         YL,R5
```

```
            mov         YH,R4
            adiw        YH:YL,5
            mov         R7,YL
            mov         R6,YH
            ret

;-------------------------------------------------------------------------------
;Ohjelma tallentaa asetukset EEPROM muistille
;-------------------------------------------------------------------------------


PWM_setup_finish:
            cp          R7,R5
            cpc         R6,R4
            brsh        PWM_setup_finishA
            mov         YL,R5
            mov         YH,R4
            adiw        YH:YL,5
            mov         R7,YL
            mov         R6,YH

PWM_setup_finishA:
            lds         temp1,Pagestate
            cpi         temp1,0
            breq        PWM_1_Finish
            rjmp        PWM_setup_F2

PWM_1_Finish:
            sts         PWM_1_sensor,R2
            mov         temp1,R2
            ldi         temp2,LOW(EEP_1_sensor)
            ldi         temp3,HIGH(EEP_1_sensor)
            rcall       EEPROM_Write

            sts         PWM_1_min,R3
            mov         temp1,R3
            ldi         temp2,LOW(EEP_1_min)
            ldi         temp3,HIGH(EEP_1_min)
            rcall       EEPROM_Write

            sts         PWM_1_lowL,R5
            mov         temp1,R5
            ldi         temp2,LOW(EEP_1_lowL)
            ldi         temp3,HIGH(EEP_1_lowL)
            rcall       EEPROM_Write

            sts         PWM_1_lowH,R4
            mov         temp1,R4
            ldi         temp2,LOW(EEP_1_lowH)
            ldi         temp3,HIGH(EEP_1_lowH)
            rcall       EEPROM_Write

            sts         PWM_1_HighL,R7
            mov         temp1,R7
            ldi         temp2,LOW(EEP_1_HighL)
            ldi         temp3,HIGH(EEP_1_HighL)
            rcall       EEPROM_Write

            sts         PWM_1_HighH,R6
            mov         temp1,R6
            ldi         temp2,LOW(EEP_1_HighH)
            ldi         temp3,HIGH(EEP_1_HighH)
            rcall       EEPROM_Write
            jmp         continue
```

```
PWM_setup_F2:
            cpi         temp1,1
            breq        PWM_2_Finish
            rjmp        PWM_setup_F3

PWM_2_Finish:
            sts         PWM_2_sensor,R2
            mov         temp1,R2
            ldi         temp2,LOW(EEP_2_sensor)
            ldi         temp3,HIGH(EEP_2_sensor)
            rcall       EEPROM_Write

            sts         PWM_2_min,R3
            mov         temp1,R3
            ldi         temp2,LOW(EEP_2_min)
            ldi         temp3,HIGH(EEP_2_min)
            rcall       EEPROM_Write

            sts         PWM_2_lowL,R5
            mov         temp1,R5
            ldi         temp2,LOW(EEP_2_lowL)
            ldi         temp3,HIGH(EEP_2_lowL)
            rcall       EEPROM_Write

            sts         PWM_2_lowH,R4
            mov         temp1,R4
            ldi         temp2,LOW(EEP_2_lowH)
            ldi         temp3,HIGH(EEP_2_lowH)
            rcall       EEPROM_Write

            sts         PWM_2_HighL,R7
            mov         temp1,R7
            ldi         temp2,LOW(EEP_2_HighL)
            ldi         temp3,HIGH(EEP_2_HighL)
            rcall       EEPROM_Write

            sts         PWM_2_HighH,R6
            mov         temp1,R6
            ldi         temp2,LOW(EEP_2_HighH)
            ldi         temp3,HIGH(EEP_2_HighH)
            rcall       EEPROM_Write
            jmp         continue

PWM_setup_F3:
            cpi         temp1,2
            breq        PWM_3_Finish
            rjmp        PWM_setup_F4


PWM_3_Finish:
            sts         PWM_3_sensor,R2
            mov         temp1,R2
            ldi         temp2,LOW(EEP_3_sensor)
            ldi         temp3,HIGH(EEP_3_sensor)
            rcall       EEPROM_Write

            sts         PWM_3_min,R3
            mov         temp1,R3
            ldi         temp2,LOW(EEP_3_min)
            ldi         temp3,HIGH(EEP_3_min)
            rcall       EEPROM_Write

            sts         PWM_3_lowL,R5
```

```
            mov         temp1,R5
            ldi         temp2,LOW(EEP_3_lowL)
            ldi         temp3,HIGH(EEP_3_lowL)
            rcall       EEPROM_Write

            sts         PWM_3_lowH,R4
            mov         temp1,R4
            ldi         temp2,LOW(EEP_3_lowH)
            ldi         temp3,HIGH(EEP_3_lowH)
            rcall       EEPROM_Write

            sts         PWM_3_HighL,R7
            mov         temp1,R7
            ldi         temp2,LOW(EEP_3_HighL)
            ldi         temp3,HIGH(EEP_3_HighL)
            rcall       EEPROM_Write

            sts         PWM_3_HighH,R6
            mov         temp1,R6
            ldi         temp2,LOW(EEP_3_HighH)
            ldi         temp3,HIGH(EEP_3_HighH)
            rcall       EEPROM_Write
            jmp         continue

PWM_setup_F4:
            cpi         temp1,3
            breq        PWM_4_Finish
            jmp         continue

PWM_4_Finish:
            sts         PWM_4_sensor,R2
            mov         temp1,R2
            ldi         temp2,LOW(EEP_4_sensor)
            ldi         temp3,HIGH(EEP_4_sensor)
            rcall       EEPROM_Write

            sts         PWM_4_min,R3
            mov         temp1,R3
            ldi         temp2,LOW(EEP_4_min)
            ldi         temp3,HIGH(EEP_4_min)
            rcall       EEPROM_Write

            sts         PWM_4_lowL,R5
            mov         temp1,R5
            ldi         temp2,LOW(EEP_4_lowL)
            ldi         temp3,HIGH(EEP_4_lowL)
            rcall       EEPROM_Write

            sts         PWM_4_lowH,R4
            mov         temp1,R4
            ldi         temp2,LOW(EEP_4_lowH)
            ldi         temp3,HIGH(EEP_4_lowH)
            rcall       EEPROM_Write

            sts         PWM_4_HighL,R7
            mov         temp1,R7
            ldi         temp2,LOW(EEP_4_HighL)
            ldi         temp3,HIGH(EEP_4_HighL)
            rcall       EEPROM_Write

            sts         PWM_4_HighH,R6
            mov         temp1,R6
            ldi         temp2,LOW(EEP_4_HighH)
            ldi         temp3,HIGH(EEP_4_HighH)
```

```
            rcall       EEPROM_Write
            jmp         continue

;-------------------------------------------------------------------------------
;Aliohjelmia asetusvalikkojen kirjoittamiseen näytölle
;-------------------------------------------------------------------------------

Setup_page:
            cpi         temp3,0
            breq        Setup_page1
            cpi         temp3,1
            breq        Setup_page2
            rjmp        Setup_page_continue

Setup_page1:
            ldi         temp1,LCD_L1
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(SetupTable*2)
            ldi         ZL,Low(SetupTable*2)
            call        Write_loop
            ldi         temp1,LCD_L2
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(SetupTable*2)
            ldi         ZL,Low(SetupTable*2)
            adiw        ZL,16
            call        Write_loop
            sbrc        R2,7
            rjmp        Auto
            sbrs        R2,7
            rjmp        Manual
            ret
Auto:
            ldi         temp1,0x88
            rcall       LCD_writecommand
            ldi         temp1,0x3E
            rcall       LCD_writedata
            ret
Manual:
            ldi         temp1,0xC8
            rcall       LCD_writecommand
            ldi         temp1,0x3E
            rcall       LCD_writedata
            ret

Setup_page2:
            ldi         temp1,LCD_L1
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(SetupTable*2)
            ldi         ZL,Low(SetupTable*2)
            adiw        ZL,32
            call        Write_loop
            ldi         temp1,LCD_L2
            rcall       LCD_writecommand
            ldi         temp3,16
            ldi         ZH,High(SetupTable*2)
            ldi         ZL,Low(SetupTable*2)
            adiw        ZL,48
            call        Write_loop
            mov         temp9,R2
            cbr         temp9,0b10000000
            cpi         temp9,0
```

```
            breq        temp1_arrow
            cpi         temp9,1
            breq        temp2_arrow
            cpi         temp9,2
            breq        temp3_arrow
            cpi         temp9,3
            breq        Temp4_arrow
            ret
Temp1_arrow:
            ldi         temp1,0x85
            rcall       LCD_writecommand
            ldi         temp1,0x3C
            rcall       LCD_writedata
            ret
Temp2_arrow:
            ldi         temp1,0xC5
            rcall       LCD_writecommand
            ldi         temp1,0x3C
            rcall       LCD_writedata
            ret
Temp3_arrow:
            ldi         temp1,0x8A
            rcall       LCD_writecommand
            ldi         temp1,0x3E
            rcall       LCD_writedata
            ret
Temp4_arrow:
            ldi         temp1,0xCA
            rcall       LCD_writecommand
            ldi         temp1,0x3E
            rcall       LCD_writedata
            ret

Setup_page_continue:
            cpi         temp3,2
            breq        Setup_page3
            cpi         temp3,3
            breq        Setup_page4
            cpi         temp3,4
            breq        Setup_page5
            ret

Setup_page3:
            ldi         temp1,HD44780_CLEAR
            rcall       LCD_writecommand
            ldi         temp3,12
            ldi         ZH,High(SetupTable2*2)
            ldi         ZL,Low(SetupTable2*2)
            call        Write_loop
            mov         YL,R3
            call        Write_PWM
            ret

Setup_page4:
            ldi         temp1,HD44780_CLEAR
            rcall       LCD_writecommand
            ldi         temp3,10
            ldi         ZH,High(SetupTable2*2)
            ldi         ZL,Low(SetupTable2*2)
            adiw        ZL,16
            call        Write_loop
            mov         YL,R5
            mov         YH,R4
            call        write_temp
```

```
            ret

Setup_page5:
            ldi         temp1,HD44780_CLEAR
            rcall       LCD_writecommand
            ldi         temp3,10
            ldi         ZH,High(SetupTable2*2)
            ldi         ZL,Low(SetupTable2*2)
            adiw        ZL,32
            call        Write_loop
            mov         YL,R7
            mov         YH,R6
            call        write_temp
            ret

;Asetusvalikkojen sivujen kirjoittamiseen taulukko.
SetupTable:
.db         " PWM     AUTO   "
.db         "         MANUAL "
.db         "Temp1      Temp3"
.db         "Temp2      Temp4"

SetupTable2:
.db         " PWMMIN         "
.db         " TEMPLOW        "
.db         " TEMPHIGH       "
```