

TAMPEREEN AMMATTIKORKEAKOULU

Tietokonetekniikan koulutusohjelma

Tietokonetekniikka

Tutkintotyö

Timo Kuni

**PIC16F690-POHJAINEN LÄMPÖMITTARI JA TIEDONKERÄIN**

Työn ohjaaja

Kai Poutanen

Työn teettäjä

Timo Kuni

Tampere 2006

<b>Tekijä:</b>	Timo Kuni
<b>Työn nimi:</b>	PIC16F690-POHJAINEN LÄMPÖMITTARI JA TIEDONKERÄIN
<b>Päivämäärä:</b>	19.4.2006
<b>Sivumäärä:</b>	55 sivua ja 82 liitesivua
<b>Hakusanat:</b>	PIC16F690, mikrokontrolleri
<b>Koulutusohjelma:</b>	Tietotekniikka
<b>Suuntautumisvaihtoehto:</b>	Tietokonetekniikka
<b>Työn ohjaaja:</b>	Poutanen Kai

## TIIVISTELMÄ

Lämmönmittauksessa tärkeintä on mitata lämpötila mahdollisimman tarkasti ja tallentaa se jälkitarkastelua varten. Tarkoituksena on ollut suunnitella prototyyppikytkentä, jolla voidaan mitata lämpötilaa. Lisäksi on ollut tarkoituksena laatia ohjelma mikroprosessorille, jolla voidaan tulkita mittaustietoja ja tallentaa niitä tietyin aikaväleihin muistiin. Lämpötilan seuranta on mielenkiintoinen tieteenala nykyään. Ilmaston lämpeneminen ja muiden luonnonilmiöiden tutkiminen on mahdollista pitkältikin ajalta. Kytkennän suunnittelu tapahtui Orcad-ohjelmistolla, joka sopi parhaiten tähän tarpeeseen. Lisäksi itse mikroprosessoriohjelma tehtiin MBLAB IDE -ohjelmointiympäristössä assembler-kielellä. Tämä ratkaisu valittiin sen takia, että valmistaja tarjoaa ilmaisen ohjelmointiympäristön. Lämpötilan mittauksessa lämpötilan seuranta onnistuu käyttäjältä helposti lyhyenä aikana. Lämmönmittaustuloksia voidaan hyödyntää hyvinkin monessa osa-alueessa. Kaikella aineella on lämpötilansa. Parhaiten tämä lämpömittari soveltuu normaalin ulkolämpötilan mittaukseen. Ohjelmaa kehitetään edelleen jatkuvasti uusien ominaisuuksien vuoksi. Muutamana kehityskohteena on entistä tarkempi lämpötilanmittaus ja kommunikointi tietokoneen kanssa. Toisena hyvänä kehityskohteena on muidenkin anturityyppinen liittäminen laitteeseen.

<b>Author:</b>	Timo Kuni
<b>Work label:</b>	PIC16F690 BASED TEMPERATURE METER AND DATA LOGGER
<b>Date:</b>	19.4.2006
<b>Number of pages:</b>	55 pages and 82 appendix pages
<b>Keyword:</b>	PIC16F690, microcontroller
<b>Education programme:</b>	Information technology
<b>Line:</b>	Computer engineering
<b>Work supervisor:</b> Poutanen Kai	

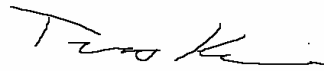
## ABSTRACT

In temperature measuring the most important thing is to measure temperature accurate and then save it for re-examination. Meaning of this job is to do prototype equipment, which can measure temperature and log it. Also compose a program for microprocessor, which can be used to decode measured data and save it in a certain time period in memory. Tracking temperature is most interesting line of science. Analyzing global warming and others natural phenomena is now possible to do with this equipment for a short period of time. Circuit designing is made with Orcad-software, which fit best to this job. The program for microprocessor is made by MBLAB IDE assembler software. This software was selected, because it was free and easy to use. Measuring temperature with this equipment is made easy and following data in a short period of time is very easy. Stored temperature data result can be abuse in many places. Every material has its own temperature. This equipment fit's best to normal temperature outdoor measuring. This program is constantly upgraded. In few development object is to make it more accurate and make it to communicate with computer.

## ALKUSANAT

Lämpötilan mittausta tapahtuu monella eri teollisuus- ja yksityis-alalla. Virike aiheeseen löytyi omasta harrastuksesta mikrokontrollereiden parissa. Työn eri vaiheet etenivät rinnakkain eri tahtia, joka hieman vaikeutti työn lopputulosta. Työskentely tiloina työssä oli Tampereen ammattikorkeakoulun ATK-luokat ja oma asunto Tampereen Hervannassa. Työn tekeminen kesti 3 kuukautta. Aikaan ei laskettu mukaan ennen itse tekemien komponenttien tekemiseen mennyttä aikaa. Työssä tärkeinä koulussa oppimissa asioissa tuli elektroniikan, digitaalitekniikan ja laitteistoläheisen ohjelmoinnin kurssit, joista suuret kiitokset ansaitsevat itse kurssien pitäjät.

Tampereella 19.4.2006



---

Timo Kuni

## SISÄLLYLUETTELO

TIIVISTELMÄ .....	2
ABSTRACT .....	3
ALKUSANAT .....	4
SISÄLLYLUETTELO .....	5
LYHENTEET JA SYMBOLIT .....	6
1 JOHDANTO .....	7
2 PIC16F690 .....	8
2.1 Ominaisuuksia .....	9
2.2 Harvard-arkkitehtuuri .....	9
2.3 PIC16F690:n ominaisuudet .....	10
2.4 I/O-liitännät .....	11
2.5 Kotelointityypit .....	12
2.6 Mikroprosessorin ohjelman siirto muistiin .....	12
2.7 W-rekisteri .....	15
2.8 BANK:it ja rekisterit .....	15
2.9 STATUS-rekisteri ja liput .....	17
2.10 PORT A-C, TRIS A-C ja ANSEL-, ANSELH-rekisterit .....	18
2.11 A/D-muunnos .....	20
2.12 Käskykanta .....	23
3 PIC16F690-OHJELMOINTI .....	24
3.1 Prosessorin alustukset .....	24
3.2 EEPROM:in luku ja kirjoitus .....	25
4 DISPLAYTEK YJ-162A SAMSUNG KS0066U LCD-MODUULI .....	27
4.1 LCD-tekniikka .....	27
4.2 KS0066U-piirin ominaisuuksien esittely .....	28
4.3 LCD:n ohjaaminen .....	31
4.4 Ajoitukset .....	37
4.5 Ohjelmointi .....	38
5 KERÄÄVÄ LÄMPÖMITTARI .....	46
5.1 Kytkenän suunnittelu .....	46
5.2 NTC .....	47
6 OHJELMA .....	50
6.1 PIC16F690:n alustus .....	50
6.2 LCD:n alustus .....	50
6.3 A/D-muunnos .....	51
6.4 EEPROM-kirjoitus ja -luku .....	51
6.5 NTC-binaariarvo ASCII-luvuksi .....	51
6.7 Menukila ja menun selaus .....	52
7 YHTEENVETO .....	53
LÄHTEET .....	54
LIITTEET .....	55

LYHENTEET JA SYMBOLIT

k	kilo, $10^3$ , 1000
M	mega, $10^6$ , 1000000
Hz	Hertsi, taajuuden perusyksikkö, 1/s
°C	celsius lämpötila aste yksikkö
U	Jännite
V	Voltti
I	Virta
A	Ampeeri
R	Resistanssi
$\Omega$	Ohmi, Omega, resistanssin perusyksikkö
A/D	Analog to Digital, Analoginen digitaaliseksi
LCD	Liquid Crystal Display, Nestekidenäyttö
ASCII	ASCII-standardi
XOR	Exclusive OR, vain TAI
DSP	Digital Signal Processing, Digitaalinen signaalin käsittely

## 1 JOHDANTO

Työn tekeminen lähti tarpeesta saada lämpötila-arvojen keräin ja mielenkiinnosta tutkia lämpötilan kehitystä pitkältä ajalta. Työn kehitys tapahtui itse hankituilla laitteilla. Laitteeksi tavoitteena oli tehdä toimiva lämpömittarin prototyyppi. Mittarilla piti olla ominaisuutena arvojen keräys tietyin aikavälein. Lisäksi mahdollisuutena piti olla talletettujen arvojen selaus laitteen käynnissä ollessa. Mikroprosessoriksi ohjamaan koko kytkentään valittiin Microchipin valmistama PIC16F690 hyvien ominaisuuksien vuoksi. PIC-mikroprosessorista on kerrottu osioissa 2 ja 3. Kytkennän näytöksi valittiin Samsungin piirillä varustettu Displaytekin YJ162 LCD-moduuli. LCD:stä kerrotaan osiossa 4. Prototyypin testauslaitteena käytettiin PICKIT 2-ohjelmointilaitetta ja sen mukana tulevaa prototyyppiä. Itse lämpömittarin toteutuksesta kerrotaan osioissa 5 ja 6.

## 2 PIC16F690

PIC16F690 on varsin uusi tulokas PIC-mikroprosessoririntamalla. Ominaisuuksia tulokkaassa on varsin paljon. Prosessorilla pystyy varsin monimutkaisiinkin toimintoihin. Prosessori ei itsessään vaadi mitään lisäkomponentteja toimiakseen. Se yksinkertaistaa kytkentää. Työhön valitsin juuri tämän mikroprosessorin helpon ohjelmoinnin ja halvan hinnan takia. Tässä osiossa kerrotaan hieman mikroprosessorin ominaisuuksista. On olemassa Microchipin valmistamia 8- tai 16-bittisiä RISC-prosessoreita, ja ne on johdettu alun perin Kennie Rennien suunnittelemasta PIC1650:sta (Microelectronic Division). Microchip ei käytä PIC-sanaa varsinaisena lyhenteenä, oikeasti tuotteen nimi on PICmicro. Lyhenne PIC-oheislaiteliitäntäohjain (Peripheral Interface Controller) on yleisesti käytetty nimitys, mutta alun perin PIC1650-mikroprosessorilla tarkoitettiin älykästä ohjelmoitavaa tietokonetta (Programmable Intelligent Computer). Alkuperäinen PIC-mikroprosessori oli varustettu 16-bittisellä CP1600-prosessoriytimellä. Ydin oli muuten hyvä prosessori, mutta kärsi huonosta I/O-suorituskyvystä. Vuonna 1975 siirryttiin käyttämään 8-bittisiä ytimiä, joilla saatiin I/O-suorituskyky kohdalleen. PIC käytti tuohon aikaan pelkkään ROM-muistia ja kykeni suorittamaan yksinkertaisia käskyjä, vaikka tuohon aikaan ei vielä sana RISC (reduced instruction set computer) ollutkaan vakiintunut. PIC käytti tätä tekniikkaan, joka suorittaa yhden käskyn yhden käskyjakson aikana. Vuonna 1985 uusi omistaja melkein lopetti kaikkien PIC-mikroprosessoreiden tuotannon, mutta kuitenkin prosessorit varustettiin myöhemmin EPROM-muistilla. Tämä mahdollisti paljon helpomman ohjelmoinnin piireille ja uudelleenohjelmoinnin. Tästä alkoi PIC-mikroprosessoreiden suosio. Nykyään PIC-mikroprosessoreita löyty melkein kaikkeen käyttöön ja monilla eri ominaisuuksilla. Ehkä yleisön suosioon PIC-piirit nousivat erilaisten MOD-piirien takia, joita käytettiin ohittamaan Playstation-konsolin suojauksia ynnä muita tämän-tyyppisiä sovelluksia. /1/



## 2.1 Ominaisuuksia

PIC-mikroprosessoriaita on saatavilla monta tyyppiä 8- tai 16-bittisillä ytimillä. Näihin tyypeihin kuuluvat myös digitaaliseen signaalinkäsittelyyn erikoistuneet dsPIC:t, joihin on lisätty käskyjä helpottamaan DSP:n sovelluksia. Yhtenä mikropiirimallina voi myös mainita rfPIC:n. rfPIC on mikroprosessorilla varustettu mikropiiri, jossa on lisäksi radiolähetin. Radiolähetin toimii 300 - 900 MHz:n alueella mallin mukaan. Tällaiset rfPIC-mikropiirit soveltuvat hyvin autojen lukkojärjestelmien ohjaamiseen. Kotelointityyppejä PIC-mikroprosessoreilla on useita pintaliitoksesta normaaliin reikäjuotettaviin. Käskyjä käskykannassa vakio PIC:ssä on 35 ja paremmissa jopa yli 70 käskyä. Ohjelmointi tapahtuu joko C-kielellä tai assemblerillä. Freeware- ja kaupallisohjelmia on myös paljon eri ohjelmointikielille. Nykyään myös ROM- ja EPROM-muistit on korvattu poikkeuksetta Flash-muistilla ja lisäksi yleensä piireissä on sisällä pieni EEPROM-muisti datan talletukseen. Vanhojakin on vielä, UV-valolla tyhjennettävät EPROM, OTP, Prom, mutta niitä täytyy erikseen tilata. PIC-mikroprosessoreista on myös klooneja, esimerkiksi Uvicom (ennen Scenix), joka tekee jopa nopeammin toimivia klooniprosessoreita. Myös opencore eli avoimen suunnittelun prosessoreina on ainakin PIC 16F84-malli. PIC-mikroprosessorien integroitua ominaisuuksia eri malleissa on kattavasti, esimerkiksi USB, CAN, AD/DA, Ethernet. Ominaisuuksia kehitellään koko ajan, joten lisää eri ominaisuuksia tulee koko ajan lisää. /2/

## 2.2 Harvard-arkkitehtuuri

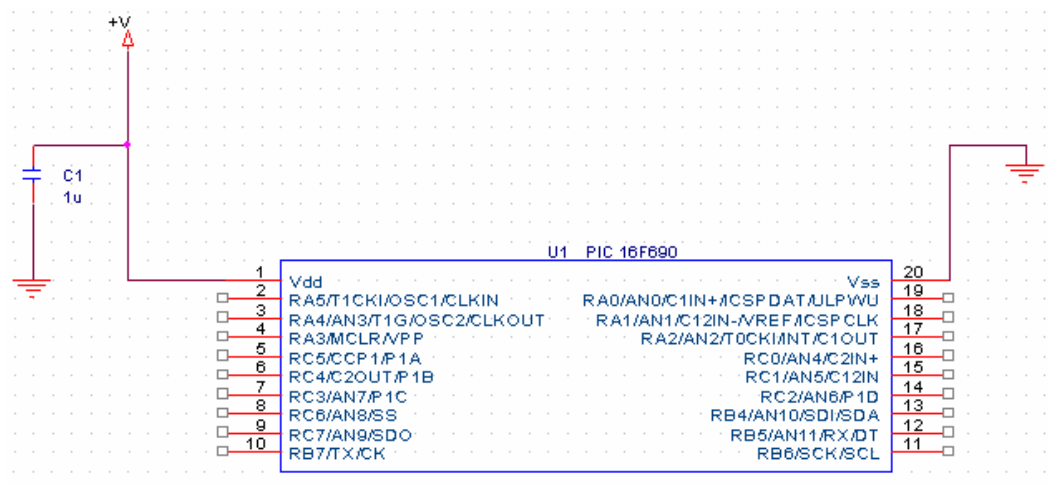
Yksi mielenkiintoinen ominaisuus PIC-prosessoreissa on Harvard-arkkitehtuurin käyttö. Harvard-arkkitehtuuri on alun perin peräisin järjestelmästä, jossa täytyi fyysisesti erottaa data- ja signaaliväylät toisistaan. Termi Harvard on peräisin Harvard Mark 1 -releohjaisesta tietokoneesta, jossa täytyi tallentaa käskyt 24-bittiselle reikänauhalle ja data 23-bittiseen releverkkoon.

Koska mikroprosessoreissa on vähän muistia käskyille ja datalle, ottaa PIC käyttöön supistetun Harvard-arkkitehtuurin (reduced instruction set). Näin tehdään

siksi, että varmistetaan yhden käskyn suoritus yhden jakson aikana. Erotetut muistit tarkoittavat sitä, että ohjelma- ja datamuistilla on eri bittikoko. PIC-mikroprosessoreilla on 12 – 16-bittisiä ohjelmamuisteja datamuistien ollessa 8-bittisiä. Tällä mahdollistetaan pitkienkin käskyjen suoritus yhden jakson aikana./3/

### 2.3 PIC16F690:n ominaisuudet

PIC16F690 on mikroprosessori 8-bittisellä Harvard-ytimellä. Ydin pystyy toimimaan jopa 20 MHz:n taajuudella ulkoisella kello-oskillaattorilla. Ytimessä on myös sisäinen ohjelmoitava kello-oskillaattori, jossa on säätövaraa 32 kilohertsisistä 8 megahertsiin, joten ulkoista kello-oskillaattoria ei välttämättä tarvita. Tämä ominaisuus tekee mikroprosessorin kytkennästä hyvin helpon, koska piirille täytyy vain syöttää syöttöjännite ja lisätä yksi tasauskondensaattori syöttöjännitteen ja maan välille. Kuvassa 1 on minimikytkentä, jonka mikroprosessori vaatii toimiakseen.



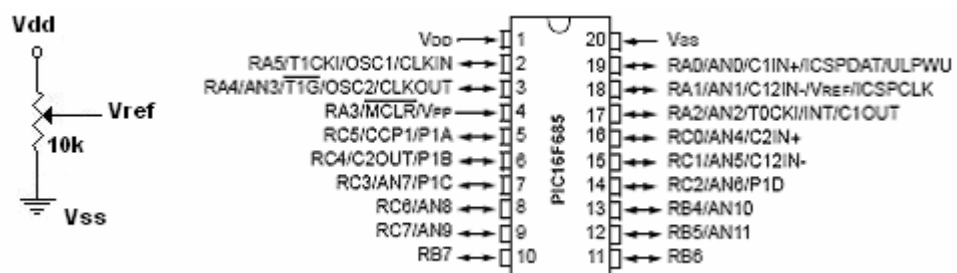
**Kuva 1** PIC16F690:n vaatima minimikytkentä

Mikroprosessorissa on Flash-tyyppistä sisäistä muistia ohjelmamuistina 4096 tavua (4kt). Harvard-arkkitehtuurin takia on käskyn pituus 14-bittinä, joten muisti on jaettu 14-bitin osiin. Ohjelmamuistia voidaan lukea myös ohjelman ajon aikana. 8-bittistä SRAM-muistia on 256 tavua. Tämä SRAM-muisti on tarkoitettu nopeaan datansiirtoon. Tätä käyttävätkin rekisterit ja itse määritellyt muuttujat

(rekisterit). Lisäksi piirillä on vielä 256 tavun 8-bitin EEPROM-muisti. Tätä muistia voidaan käyttää datan säilömiseen ja lukemiseen ohjelman ajon aikana, koska tieto pysyy EEPROM-muistissa käyttöjännitteiden poiston jälkeen, esimerkiksi jonkin tietotaulukon tallentamiseen, joka ei enää sovi Flash-muistin alueelle.

## 2.4 I/O-liitännät

I/O-liitäntöjä on mikropiirissä 18 kappaletta. Liitännät on mahdollista ohjelmoida tuloiksi tai lähdöiksi. Liitännöistä voidaan lisäksi määrätä joidenkin osalta, ovatko ne analogisia vai digitaalisia, lähinnä sen takia, jotta voitaisiin käyttää A/D-muunnosta eri nastoista. Kuvassa 2 näkyvät eri liitinnastojen käyttömahdollisuudet. AN0 - AN11:t kuvassa ovat nastoja, jotka voidaan ohjelmoida A/D-muuntamaan tulevaan jännitettä kyseisestä nastasta. Kuvassa 2 olevaa nastaa 18 voidaan käyttää myös ottamaan Vref-jännite, tällä määrätään referenssijännite A/D muunnokselle. Referenssijännite on A/D-muunnoksen maksimivertausjännite. Esimerkiksi, jos Vref-jännite on 4 V ja muunnettava jännite on 2 V, on 8-bitin muunnoksella Vref-jännite 4 V:llä bitteinä 1111 1111b. Tällöin 2 V:a on puolet tuosta eli 0111 1111b ja 0 V:tä on tietysti 0000 0000b.

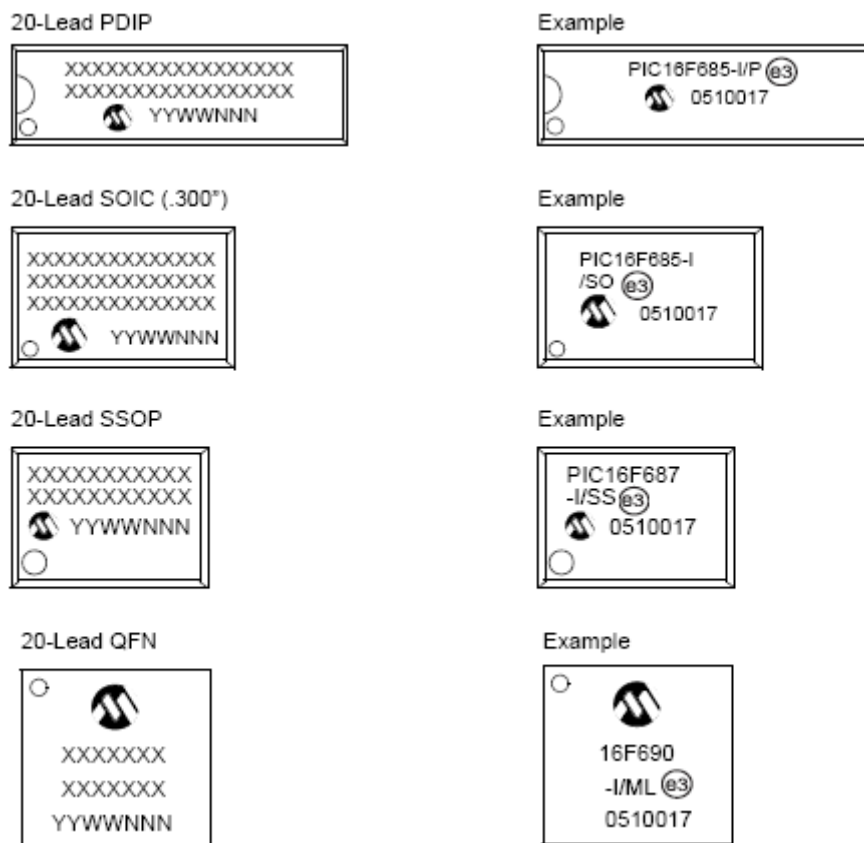


**Kuva 2** Vref-kytkentä ja PIC16F690:n liitäntänastat /4/

Kuvassa 2 olevalla Vref-jännitteen säädöllä voidaan säätövastuksella säätää jännite kohdalleen.

## 2.5 Kotelointityypit

Mikropiirin kotelointityypeinä ovat PDIP, SOIC, SSOP ja QFN. Eri kotelointityypit eroavat lähinnä toisistaan siinä, ovatko ne pintajuotettava vai eivät. Kuvassa 3 näkyvät eri kantojen erot. Eniten eroa on jalkojen osalta ensimmäisessä (PDIP) verrattuna muihin. PDIP-kotelointityyppi on reikäjuotettava, kun taas muut kolme ovat pintajuotettavia. Pintajuotettavissa erot ovat taas erilaisissa kotelojen muodoissa SSOP:n ja SOIC:n välillä. QFN-kotelo on taas kokonaan jalaton pintaliitoskomponentti. Tarkemmat tiedot kaikista kantatyypeistä löytää valmistajan datalehdestä /5/.

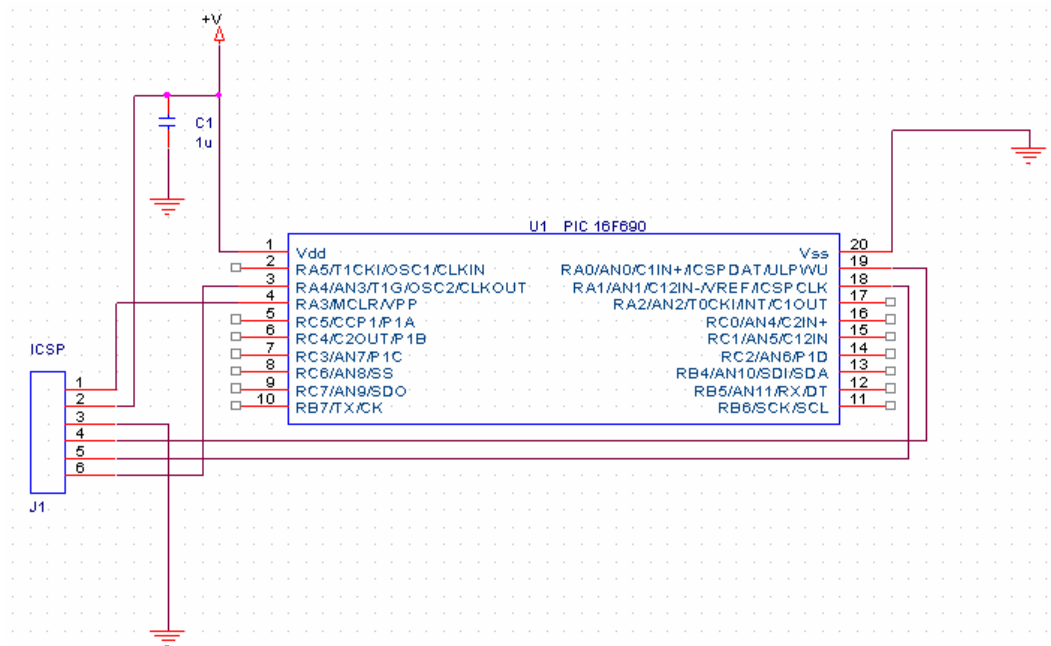


**Kuva 3** PIC16F690-piirin koteloinnit /6/

## 2.6 Mikroprosessorin ohjelman siirto muistiin

Yksi mielenkiintoinen piirin ominaisuus on ICSP (In-Circuit Serial Programming), muulla tavalla toteutettu ohjelmointi on tarpeetonta. Tämä tarkoittaa pii-

rin ohjelmoimista sarjadataalla kytkennän ollessa paikallaan. Ominaisuus on hyvä, koska esimerkiksi, jos tehdään tuote ja juotetaan osat paikoilleen, niin ohjelmointi voidaan suoraan tehdä tähän irrottamatta PIC-piiriä tai mitään muuta komponenttia. Ainoastaan kytkennän täytyy olla jännitteettömänä. Kuvassa 4 näkyy, kuinka ICSP kytketään PIC16F690-mikroprosessoriin.



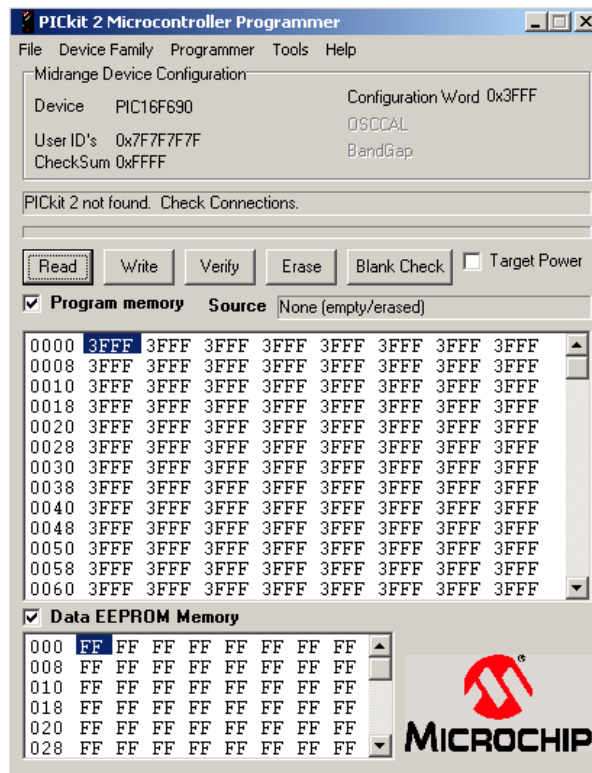
**Kuva 4** ICSP:n peruskytkentä

Ohjelmointi vaatii vain käyttöjännitteen (Vdd), maan (Vss), ohjelmointijännitteen (Vpp), ICSP-kellosignaalin (ICSPCLK) ja ICSP-datan (ICSPDAT). Ohjelmointiin on valmiina saatavissa laitteita, joilla piirin muistin ohjelmointi ja lukeminen onnistuu usealla eri piirillä. Näistä halvimpana mutta toimivana on PICKIT 2, joka toimii Windows-ympäristössä USB-liitännän kautta. Ohjelmointilaitte tukee useita eri PIC-piirejä, jotka on varustettu ICSP-väylällä. Kuvassa 5 on PICKIT 2 -laite. Laitteessa itsessäänkin on sisällä PIC-mikroprosessori (PIC18F2550), jolla hoidetaan liikennöinti USB-väylässä ja datan syöttäminen ohjelmoitavalle prosessorille.



Kuva 5 PICKIT 2 /7/

Ohjelmistona mukana tulee ohjelmointiin ja piirin muistin lukemiseen tarkoitettu ohjelma PICKIT 2 Programmer. Ohjelma, lähdekoodit ja kytkennät ovat myös saatavissa Microchipin kotisivuilta. Kuvassa 6 on ohjelmointiohjelmiston käyttöliittymä.



Kuva 6 Ohjelmointiohjelmiston käyttöliittymä

Ylävalikossa kuvassa 6 näkyy File-valikko, jolla voidaan ladata tai tallettaa tiedosto. Device Family -kohdasta saadaan valita mitä Microchipin PIC-perhettä käytetään. Ohjelma tunnistaa itse ohjelmoitavan mikroprosessorin. Kuvassa 6 näkyvät painikkeet lukemiseen (Read) ja kirjoittamiseen (Write) erikseen. Näillä voidaan kirjoittaa muistiin ja lukea piiristä. Ohjelmamuisti (Program Memory) näkyy painikerivistön alla ja piirin EEPROM-muistin sisältö (Data EEPROM Memory) näkyy alempana. Lisäksi datan perillemeno voidaan varmistaa Verify-napilla ja Erase-napilla voidaan tyhjentää piirin muistit. Tyhjennys voidaan tarkistaa Blank Check -napilla. Lisäksi muistien päällä on rastit, joista voidaan hallita, mihin muistiin painikkeiden toiminnot vaikuttavat. Kytkeä voidaan myös ajaa suoraan USB-portin virralla panemalla rasti Target Power -ruutuun. Siinä huonona puolena on, että virta on rajoitettu 300 mA:iin. Tämä kyllä riittää pienissä kytkennöissä, mutta suuremmissa kytkennöissä virtaraja saattaa tulla vastaan.

Ohjelman siirto prosessorin muistiin on siis varsin helppoa. Ensin kytketään laite kiinni ICSP-väylään. Tämän jälkeen ladataan ohjelma File-> Import File. Sitteen vain painetaan Write-painiketta, jolloin ohjelma siirtyy PIC-mikroprosessorin muistiin.

## 2.7 W-rekisteri

W-rekisteri on yleiskäyttörekieteri mikroprosessorissa. Sitä kutsutaan myös nimellä work- eli työrekisteri. W-rekisteriä käytetään välittämään dataa toisesta rekisteristä toiseen.

## 2.8 BANK:it ja rekisterit

PIC16F690-mikroprosessori sisältää neljä pankkia(BANK). Jokaisessa pankissa on tietyt tila- ja ohjausrekisterit eri ominaisuudelle. Eri pankin sisältöön ei pääse käsiksi, jos on toisessa pankissa. Kuvassa 7 PIC16F690-mikroprosessorin rekisterirakenne näkyy kokonaisuudessaan.

File Address	File Address	File Address	File Address
Indirect addr. (1) 00h	Indirect addr. (1) 80h	Indirect addr. (1) 100h	Indirect addr. (1) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	PORTA 105h	TRISA 185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	PORTC 107h	TRISC 187h
08h	88h	108h	188h
09h	89h	109h	189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDAT 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2(1) 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	18Eh
TMR1H 0Fh	OSCCON 8Fh	EEADRH 10Fh	18Fh
T1CON 10h	OSCTUNE 90h	110h	190h
TMR2 11h	91h	111h	191h
T2CON 12h	PR2 92h	112h	192h
SSPBUF 13h	SSPADD(2) 93h	113h	193h
SSPCON 14h	SSPSTAT 94h	114h	194h
CCPR1L 15h	WPUA 95h	WPUB 115h	195h
CCPR1H 16h	IOCA 96h	IOCB 116h	196h
CCP1CON 17h	WDTCON 97h	117h	197h
RCSTA 18h	TXSTA 98h	VRCON 118h	198h
TXREG 19h	SPBRG 99h	CM1CON0 119h	199h
RCREG 1Ah	SPBRGH 9Ah	CM2CON0 11Ah	19Ah
1Bh	BAUDCTL 9Bh	CM2CON1 11Bh	19Bh
PWM1CON 1Ch	9Ch	11Ch	19Ch
ECCPAS 1Dh	9Dh	11Dh	PSTRCON 19Dh
ADRESH 1Eh	ADRESL 9Eh	ANSEL 11Eh	SRCON 19Eh
ADCON0 1Fh	ADCON1 9Fh	ANSELH 11Fh	19Fh
20h	A0h	120h	1A0h
General Purpose Register	General Purpose Register	General Purpose Register	
96 Bytes	80 Bytes	80 Bytes	
7Fh	EFh	16Fh	
	accesses 70h-7Fh F0h	accesses 70h-7Fh 170h	accesses 70h-7Fh 1F0h
Bank 0	Bank 1	Bank 2	Bank 3

**Kuva 7** PIC16F690-prosessorin pankit /8/

Pankkien sisällön lukeminen voidaan tehdä epäsuorasti ja suorasti. Epäsuorasti luettuna on aina kahden pankin sisältö käytössä, mutta joudutaan käyttämään eri osoitealuetta toisen pankin lukemiseen. Esimerkiksi BANK0 ja BANK1 epäsuorasti ohjattuna löytyvät osoitealueelta 00h - FFh ja BANK2 ja BANK3 osoitealueelta 100h - 1FFh. Tällöin joudutaan esimerkiksi lukemaan BANK1- ja TRISA-rekisteriä osoitteesta 85h, kun muulloin se onnistuisi suoralla ohjauksella 05h:in alueesta. Suorassa osoituksessa valitaan aina yksi pankki käyttöön ja sen



osoitealue on aina 00h - 7Fh. Pankkien vaihtaminen tapahtuu STATUS-rekisteristä käsin.

## 2.9 STATUS-rekisteri ja liput

STATUS-rekisteri on tärkeä rekisteri. Siellä sijaitsevat suorittimen ALU, Reset-tila ja Bankin valintatiedot. Rekisteri sijaitsee kaikissa neljässä pankissa. Taulukossa 1 näkyy kokonaisuudessaan, mitä STATUS-rekisteri sisältää ja sen jälkeen bittien tilojen merkitys STATUS-rekisterissä esimerkissä 1.

**Taulukko 1** STATUS-rekisterin sisältö

7	6	5	4	3	2	1	0
IRP	RP1	RP0	/TO	/PD	Z	DC	C

Bitti 7 IRP Bank-valinta	0	BANK0, BANK1	Epäsuora
Bitti 6-5 RP1-RP0	1	BANK2, BANK3	
	00	BANK0	Suora Bank Valinta
	01	BANK1	
	10	BANK2	
	11	BANK3	
Bitti 4 /TO	0	Piirin käynnistyksen jälkeen, CLRWDT- tai SLEEP-käsky	
	1	WDT-laskuri ympäri	
Bitti 3 /PD	0	Kun Sleep-operaatio on käynnissä	
	1	Piirin käynnistyksen jälkeen tai CLRWDT- käskyn jälkeen	
Bitti 2 Z	0	Aritmeettinen operaatio on muu kuin nolla	
	1	Aritmeettinen operaatio on nolla	
Bitti 1 DC	0	Ei ylivuotoa 5 bittiin	
	1	Ylivuoto 4 alimmasta bitistä viidenteen bittiin	
Bitti 0 C	0	Jos ei ole ylivuotoa	
	1	On ylivuoto, ylivuoto ”yhdeksänteen” bittiin	

**Esimerkki 1** STATUS-rekisterin bittien tilojen merkitys

Ehkä tärkeimpinä bitteinä STATUS-rekisterissä ovat bitit 6 - 5 RP1 - RP0, joilla valitaan mitä pankkia käytetään. Näillä biteillä ohjataan, mitä eri rekisterialuetta halutaan lukea tai muokata. Toisina tärkeinä bitteinä voidaan pitää Z-, DC- ja C-bittejä. Z-bitin sisältö on aina yksi, kun aritmeettisen operaation tulos on 0.

Toinen hyödyllinen bitti on C-bitti, eli carry-bitti toiselta nimeltään ylivuoto-bitti. C-bitti on aina 1, jos esimerkiksi lukua 1111 1111b lisätään yhdellä, tällöin kaikki muut bitit menevät nolaksi, paitsi bitti 9, joka ylivuotaa ja nostaa C-bitin ylös.

DC-bittiä taas voidaan käyttää negatiivisen ja positiivisen alueen erottamiseen 8-bittisellä luvulla. Esimerkiksi, jos tulos on suurempi kuin 0Fh, DC on 1, mutta jos pienempi kuin 0Fh, niin tällöin DC on 0. Kaikki komennot eivät tietysti vaikuta näihin lippuihin. Tästä vaikutuksesta lippuihin on lisää tietoa liitteestä 1 käskyjen toiminnasta.

## 2.10 PORT A-C, TRIS A-C ja ANSEL-, ANSELH-rekisterit

TRIS A-, TRIS B-, TRIS C-rekisterit ovat olennainen osa mikroprosessorin ohjelmointia. TRIS-rekistereillä määrätään, ovatko eri portit ulostuloja vai sisääntuloja. Kuvan 2 mukaisten liitännöiden ohjelmointi tapahtuu pääosin näissä rekistereissä. TRIS A-rekisterillä ohjataan portin A tilaa, TRIS B-rekisterillä ohjataan portin B tilaa ja TRIS C-rekisterillä portin C tilaa. TRIS-rekistereissä 0 tarkoittaa ulostuloa ja 1 sisääntuloa. PORT A-, PORT B-, PORT C-rekisteristä voidaan syöttää tietoa ulostuloksi, tai voidaan lukea sisääntulotiedot. Kun portit ovat sisääntuloina, tulkitaan tilat TTL-tasoina. Taulukossa 2 ovat TRISA-C ja PORTA-C -rekisterit kokonaisuudessaan ja niiden liitännät suoraan piiriin navoissa. Kaikissa porteissa ei ole käytössä kaikkia 8-bittisiä. Taulukossa 2 nämä on määritelty tummanharmaisiin alueisiin.

**Taulukko 2** Rekisterit TRIS A - C ja rekisterit PORT A - C

Bitit/Rekisterit	7	6	5	4	3	2	1	0
TRISA			I/O RA5	I/O RA4	I/O RA3	I/O RA2	I/O RA1	I/O RA0
TRISB	I/O RB7	I/O RB6	I/O RB5	I/O RB4				
TRISC	I/O RC7	I/O RC6	I/O RC5	I/O RC4	I/O RC3	I/O RC2	I/O RC1	I/O RC0
PORTA			RA5	RA4	RA3	RA2	RA1	RA0
PORTB	RB7	RB6	RB5	RB4				
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0

Mikroprosessorissa on lisäksi ANSEL-rekisteri, jolla määrätään, onko portti analoginen vai digitaalinen. Tätä ominaisuutta tarvitaan varsinaisesti vain, jos käytetään A/D-muunnosta, jolloin täytyy portti asettaa sisääntuloksi ja analogiseen moodiin. PIC16F690-prosessori käyttää kahta ANSEL-rekisteriä, koska mikroprosessorissa on 12 kappaletta 10-bittistä A/D-kanavaa. ANSELH-rekisterillä siis ohjataan neljää ylintä kanavaa. Kun rekisterissä on 0, tarkoittaa se digitaalitilaa, ja taas kun on rekisterissä 1, tarkoittaa se analogitilaa. Kun käyttää yhtä aikaa portteja ulostulona ja analogisena, tekee prosessori virheellisiä toimintoja. Taulukossa 3 ovat ANSEL- ja ANSELH-rekisterin sisällöt.

**Taulukko 3** Rekisterit ANSEL ja ANSELH

Bitit/Rekisterit	7	6	5	4	3	2	1	0
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
ANSELH					ANS11	ANS10	ANS9	ANS8

Taulukosta 3 vertaamalla suoraan kuvan 2 merkintöihin saadaan selville mitä porttia eri ANS-merkintä tarkoittaa. Kuvassa 2 on merkitty ANS:n tilalle vain AN-merkintä. Esimerkiksi ANS7 on piirin nastassa 7 oleva A/D-portti. Esimerkissä 2 näytetään kuinka Portin A:n tilojen muuttaminen ulostuloksi tapahtuu. Esimerkissä 3 näytetään, kuinka saadaan Portti A:n tilat sisääntuloiksi.

BCF STATUS,RP0	; Mennään Bank 0
BCF STATUS,RP1	;
CLRF PORTA	; Tyhjennetään PORTA
BSF STATUS,RP1	; Mennään BANK 2
CLRF ANSEL	; Tyhjennetään ANSEL ja ANSELH, kaikki portit digitaalisiksi
CLRF ANSELH	;
BSF STATUS,RP0	; Mennään Bank 1
BCF STATUS,RP1	;
MOVLW 00h	; Talletetaan W-rekisteriin 00h
MOVWF TRISA	; Siirretään 00h TRISA-rekisteriin, kaikki portit ulostuloksi.
BCF STATUS,RP0	; Mennään Bank 0
MOVLW 0xFFh	; Pannaan FFh rekisteriin-W
MOVWF PORTA	; Siirretään FFh PORTA, kaikki portin A-bitit nousevat ykköstitilaan ja siirtyvät niitä vastaaviin nastoihin ulostulossa

**Esimerkki 2** PORTA:n asettaminen ulostuloksi ja portin tilojen muuttaminen

BCF STATUS, RP0	; Mennään Bank 0
BCF STATUS, RP1	;
CLRF PORTA.	; Tyhjennetään PORTA
BSF STATUS, RP1	; Mennään BANK 2
CLRF ANSEL	; Tyhjennetään ANSEL ja ANSELH, kaikki portit digitaaliksi
CLRF ANSELH	;
BSF STATUS, RP0	; Mennään Bank 1
BCF STATUS, RP1	;
MOVLW FFh	; Talletetaan W-rekisteriin FFh
MOVWF TRISA	; Siirretään FFh TRISA-rekisteriin, kaikki portit sisääntuloiksi
BCF STATUS, RP0	; Mennään Bank 0
MOVF PORTA, W	; Luetaan PORTA-sisääntulon sisältö W-rekisteriin

### Esimerkki 3 PORTA:n asettaminen ulostuloksi ja portista lukeminen

#### 2.11 A/D-muunnos

A/D-muunnos (Analog to Digital) on muunnos analogisesta signaalista digitaaliseksi signaaliksi. A/D-muunnos tapahtuu PIC16F690-mikroprosessorissa SigmaDelta-periaatteella. A/D-muunnoskanavia mikroprosessorissa on 12 kappaletta, jotka voidaan kytkeä ja ottaa pois käytöstä. Jokainen kanava pystyy ottamaan 10-bittisen muunnoksen eli 1024 arvoa alueesta. Pienempiäkin bittimääriä voidaan käyttää vain olemalla lukematta kahta alinta bittiä ja näin saadaan 8-bittinen muunnos. A/D-muunnoksen referenssijännite voidaan määrätä, otetaanko se suoraan käyttöjännitteestä vai itse määritetystä piirin navasta 18 Vref eli AN1 A/D-muunnoskanavasta. Referenssijännite on jännite, jota vastaan mitattua jännitettä verrataan. Se on suurin jännite arvoasteikolla. Esimerkiksi, jos referenssijännite on 3 V tämän vastaava bittiarvo muunnoksessa 10-bittisellä lukemalla on 3FFh ja 0 V:lla 000h. A/D-kanavan määrääminen tapahtuu ANSEL-, ANSELH-, ADCON0-, ADCON1- ja TRIS-rekistereissä. ANSEL- ja ANSELH-rekistereillä ohjataan, onko portti analoginen vai digitaalinen. Tässä tapauksessa se on analoginen, jos halutaan A/D-muunnosta käyttää. TRIS-rekisterillä pitää portti määrätä sisääntuloksi, jotta saadaan jännitettä luettua portista. ADCON1-rekisterillä määrätään A/D-muunnoksen muunnosnopeus, jos ei haluta tehdä mahdollisimman nopeaa muunnosta, ei tähän tarvitse koskea.

ADCON0-rekisterillä ohjataan ja määrätään, mitä ja mistä tehdään A/D-muunnos. Taulukossa 4 ADCON rekisterin sisältö on kokonaisuudessaan esiteltyä.

**Taulukko 4** ADCON0-rekisteri

Bitit/Rekisterit	7	6	5	4	3	2	1	0
ADCON0	AFDM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

AFDM-bitillä määritellään, onko muunnettu bittijono oikealle vai vasemmalle keskitetty. Taulukko 5 kuvaa, kuinka keskitys vaikuttaa tulokseen. Vaaleanharmaat ovat aktiivisia bittejä.

**Taulukko 5** AFDM:n tilat

	MSB														LSB		
Tila/Bitit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
AFDM=0	9	8	7	6	5	4	3	2	1	0							Vasen
AFDM=1							9	8	7	6	5	4	3	2	1	0	Oikea
	ADRESL														ADRESH		

Itse tuloksen lukeminen tapahtuu ADRESL- ja ADRESH-rekistereistä kuvan osoittamalla tavalla. VCFG-bitillä ohjataan, käytetäänkö referenssijännitteenä käyttöjännitettä vai jalan 18 jännitettä. Jos bitti on 0, käytetään käyttöjännitettä. Jos se on 1, käytetään 18 jalkaan tulevaa jännitettä. CHS0 - CHS3-biteillä määrätään, mitä A/D-muunnoskanavaa käytetään. Taulukossa 6 näkyy, mitä kanavaa vastaa mikäkin bittiyhdistelmä. Rivin BIN alla olevat bitit vastaavat CHS3-, CHS2-, CHS1-, CHS0-bittejä suoraan.

**Taulukko 6** CHS0- CHS3:n avulla kanavien valinta

DES	BIN	KANAVA
0	0000	AN0
1	0001	AN1
2	0010	AN2
3	0011	AN3
4	0100	AN4
5	0101	AN5
6	0110	AN6
7	0111	AN7
8	1000	AN8
9	1001	AN9
10	1010	AN10
11	1011	AN11
12	1100	Vref
13	1101	VP6
14	1110	Ei käytössä
15	1111	Ei käytössä

GO/DONE-bitillä käynnistetään A/D-muunnos ja tutkitaan, onko se valmis. Asetamalla GO-bitti alkaa A/D-muunnos. Kun A/D-muunnos on valmis, laskee bitti alas eli 0-tilaan. ADON-bitillä määritellään pelkästään, onko A/D-muunnos lohko käytössä mikroprosessorissa. 0 on pois päältä ja 1 on päällä. Esimerkissä 4 näytetään kuinka A/D-muunnoksen ottaminen tapahtuu RA0-portista.

BSF STATUS, RP0	; Mennään Bank 1
BCF STATUS, RP1	;
MOVLW B'01110000'	; A/D-muunnoksen kellokerroin
MOVWF ADCON1	; Siirretään ADCON1-rekisteriin
BSF TRISA,0	; Asetetaan RA0 ulostuloksi
BCF STATUS, RP0	; Mennään Bank 2
BSF STATUS, RP1	;
BSF ANSEL, 0	; Asetetaan RA0 analogiseksi
BCF STATUS, RP0	; Mennään Bank 0
MOVLW B'10000001'	; Oikealle keskitetty, Referenssi käyttöjänniteestä, ja portista AN0-muunnos
MOVWF ADCON0	; Siirretään ADCON0-rekisteriin.
CALL Odota5us	; Odotetaan 5 mikrosekuntia, jotta kondensaattorin tasaavat jänniteensä
BSF ADCON0, GO	; Aloitetaan käännös
BCF STATUS, RP1	; Mennään Bank 0
BTFSC ADCON0, GO	; Onko käännös valmis. GO/DONE-bitti mennyt alas
GOTO \$-1	; Ei valmis, joten hypätään rivi ylemmäs ja testaataan uudelleen
MOVF ADRESH, W	; Lue 2 ylintä bittia W-rekisteriin
MOVWF RESULTHI	; Siirrä talteen
BSF STATUS, RP0	; Mennään Bank 1
MOVF ADRESL, W	; Lue 8 alinta bittia W-rekisteriin
BCF STATUS, RP0	; Mennään Bank 0
MOVWF RESULTLO	; Siirrä talteen

**Esimerkki 4** A/D-muunnoksen ottaminen portista RA0 /9/

## 2.12 Käskykanta

PIC16F690-käskykanta sisältää kokonaisuudessaan 35-käskyä. Käskyt ovat nopeita käyttää RISC- ja Harvard-arkkitehtuurin takia. Kaikki käskyt vievät vain yhden käskyjakson suoritusaikaa, poikkeuksena hyppykäskyt. Näillä käskyillä tulee kyllä toimeen, mutta monimutkaiset laskutoimitukset on hankala toteuttaa ja varsinkin siksi, koska ne vievät paljon muistista tilaa. Liitteessä 1 on lyhyesti kaikki 35 käskyä ja jokaisesta pienen esimerkkikoodin avulla selitetty kuinka ne toimivat ja vaikuttavat rekisterien tilaan. Joidenkin käskyjen perästä löytyvät merkinnät f ja d. f tarkoittaa rekisterin osoitetta tai rekisterille esimääritettyä nimeä. d on suuntaa ohjaava ohjausbitti. Jos d on 1, käskyn tulos menee annetun rekisterin osoitteeseen. Sama toiminta saavutetaan jättämällä koko d pois käskystä, mikä tällöin tarkoittaa nolla-tilaa. Jos d on 0, käskyn tulos ohjautuu W-rekisterin sisältöön. Joillakin käskyillä ei ole d- tai f-ominaisuutta käytössään, koska ne vaikuttavat suoraan, esimerkiksi käsky CLRW. Tämä käsky vaikuttaa suoraan W-rekisteriin tyhjäämällä sen. Joissakin bittimuokkausoperaatioissa on b-merkintä käskyn perässä. Tämä tarkoittaa sitä, että bittiä tarkastellaan tai muutetaan rekisterissä. Myös merkintä k on joidenkin käskyjen perässä, tällä tarkoitetaan vain käytettävää lukua tai osoitetta. Lukuna tässä voi olla binaari-, heksa-, desimaali- tai ascii-luku. Esimerkissä 5 on esitelty, kuinka tiedot syötetään eri lukujärjestelmissä.

MOVLW d'10'	; Desimaalina 0-255
MOVLW b'00000001'	; Binaarina 00000000 - 11111111
MOVLW 0x20h	; Heksana 0x00 - 0xFF
MOVLW a'A'	; ASCII-merkinä

**Esimerkki 5** Tiedon syöttäminen eri lukujärjestelmissä

### 3 PIC16F690-OHJELMOINTI

Ohjelmointiosiossa esittelen vain työssä tarvitsemat alueet, koska muuten alustusten esittelyyn menisi iso osa koko työstä.

#### 3.1 Prosessorin alustukset

Koska prosessorissa on monia eri vaihtoehtoja, jotka täytyy määrittää ennen kuin koko ohjelmaa edes ajetaan. Tämän takia näiden ominaisuuksien määrittäminen onnistuu pelkästään ohjelmointivaiheessa. Näille on englanninkielinen nimi Configuration bits, joka suomeksi suoraan ohjausbitit. Ohjausbitit sijaitsevat eri paikassa kuin ohjelmamuisti, osoitteessa 2007h. Kuvassa 8 näkyvät kaikki ohjausbitit.

Reserved	Reserved	FCMEN	IESO	BOREN1 <sup>(1)</sup>	BOREN0 <sup>(1)</sup>	CPD <sup>(2)</sup>	CP <sup>(3)</sup>	MCLR <sup>(4)</sup>	PWRT <sup>(4)</sup>	WDTE	FOSC2	FOSC1	FOSC0
bit 13													bit 0
bit 13-12 <b>Reserved:</b> Reserved bits. Do Not Use.													
bit 11 <b>FCMEN:</b> Fail-Safe Clock Monitor Enabled bit 1 = Fail-Safe Clock Monitor is enabled 0 = Fail-Safe Clock Monitor is disabled													
bit 10 <b>IESO:</b> Internal External Switchover bit 1 = Internal External Switchover mode is enabled 0 = Internal External Switchover mode is disabled													
bit 9-8 <b>BOREN&lt;1:0&gt;:</b> Brown-out Reset Selection bits <sup>(1)</sup> 11 = BOR enabled 10 = BOR enabled during operation and disabled in Sleep 01 = BOR controlled by SBOREN bit (PCON<4>) 00 = BOR disabled													
bit 7 <b>CPD:</b> Data Code Protection bit <sup>(2)</sup> 1 = Data memory code protection is disabled 0 = Data memory code protection is enabled													
bit 6 <b>CP:</b> Code Protection bit <sup>(3)</sup> 1 = Program memory code protection is disabled 0 = Program memory code protection is enabled													
bit 5 <b>MCLR:</b> RA3/MCLR/VPP pin function select bit <sup>(4)</sup> 1 = RA3/MCLR/VPP pin function is MCLR 0 = RA3/MCLR/VPP pin function is digital input, MCLR internally tied to VDD													
bit 4 <b>PWRT:</b> Power-up Timer Enable bit 1 = PWRT disabled 0 = PWRT enabled													
bit 3 <b>WDTE:</b> Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled and can be enabled by SWDTEN bit (WDTCON<0>)													
bit 2-0 <b>FOSC&lt;2:0&gt;:</b> Oscillator Selection bits 111 = RC oscillator: CLKOUT function on RA4/AN3/T1G/OSC2/CLKOUT pin, RC on RA5/T1CKI/OSC1/CLKIN 110 = RCIO oscillator: I/O function on RA4/AN3/T1G/OSC2/CLKOUT pin, RC on RA5/T1CKI/OSC1/CLKIN 101 = INTOSC oscillator: CLKOUT function on RA4/AN3/T1G/OSC2/CLKOUT pin, I/O function on RA5/T1CKI/OSC1/CLKIN 100 = INTOSCIO oscillator: I/O function on RA4/AN3/T1G/OSC2/CLKOUT pin, I/O function on RA5/T1CKI/OSC1/CLKIN 011 = EC: I/O function on RA4/AN3/T1G/OSC2/CLKOUT pin, CLKIN on RA5/T1CKI/OSC1/CLKIN 010 = HS oscillator: High-speed crystal/resonator on RA4/AN3/T1G/OSC2/CLKOUT and RA5/T1CKI/OSC1/CLKIN 001 = XT oscillator: Crystal/resonator on RA4/AN3/T1G/OSC2/CLKOUT and RA5/T1CKI/OSC1/CLKIN 000 = LP oscillator: Low-power crystal on RA4/AN3/T1G/OSC2/CLKOUT and RA5/T1CKI/OSC1/CLKIN													
<b>Note</b> 1: Enabling Brown-out Reset does not automatically enable Power-up Timer. 2: The entire data EEPROM will be erased when the code-protect is turned off. 3: The entire program memory will be erased when non code-protect is turned off. 4: When MCLR is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.													

**Kuva 8** Ohjausbitit /10/



Ohjausbitit määritellään esimerkin 6 mukaan. Ohjelmointiympäristössä bittien määrittely on helpompaa, kuin itse ruveta bittijonoja kirjoittamaan. Esimerkissä 6 ovat yleisimmät määrittelyt selitettynä.

```
#include <p16F690.inc> ; Määritellään prossi yyppi.
__config(_INTRC_OSC_NOCLKOUT & _WDT_OFF & _PWRTE_OFF & _MCLRE_OFF & _CP_OFF &
        _CPD_OFF & _BOR_OFF & _IESO_OFF & _FCMEN_OFF); Ohjausbitit.

INTRC_OSC_NOCLKOUT ; Sisäinen kello päälle ei kellonsignaalin-ulosantia.
WDT_OFF ; WDT-ajastin pois päältä.
PWRTE_OFF ; POWERUP-laskuri pois päältä
MCLRE_OFF ; Reset-painike ei toiminnassa
CP_OFF ; Koodin-suojaus pois päältä, kopiointia ei estetty
CPD_OFF ; Data-muistin suojaus pois päältä.
BOR_OFF ; BOR pois päältä.
IESO_OFF ; Sisäinen/ulkoisen vaihto pois päältä.
FCMEN_OFF ; Vikatila -monitori pois päältä.
```

### Esimerkki 6 Mikroprosessorin-alustukset

## 3.2 EEPROM:in luku ja kirjoitus

PIC16F690-mikroprosessori sisältää 256 tavua sisäistä EEPROM-muistia. EEPROM-muisti (Electrically Erasable PROM) on sähköisesti tyhjennettävää muistia. EEPROM:in kirjoitus onnistuu ohjelman ajon aikana ja ohjelmointivaiheessa mikroprosessorille. EEPROM:in lukemiseen ja kirjoittamiseen tarvittavia rekistereitä ovat EECON1, EECON2, EEDAT, EEDATH, EEADR ja EEADRH. Joista EEDATH ja EEADRH voidaan poistaa käytöstä EEPROM:in luvun ja kirjoituksen aikana. Nämä rekisterit ovat vain ohjelmamuistin lukemiseen tarkoitettuja rekisterejä. Koska ohjelmalaskuri on 14-bittinen ja kääntäjädata on 14-bittinen, vaaditaan tällöin kaksi 8-bittistä rekisteriä niiden käyttämiseen. EEPROM:in lukeminen on melko yksinkertaista. Esimerkin 7 koodissa luetaan jostakin osoitteesta dataa EEPROM:in muistialueelta. Datan osoite välitetään ohjelmalle DATA\_EE\_ADDR-muuttujan sisältönä ja kyseisen paikan data saadaan ulos W-rekisteristä.

BSF STATUS, RP1	; Mennään Bank 2
BCF STATUS, RP0	;
MOVLW DATA_EE_ADDR	; Siirretään joku osoite DATA_EE_ADDR W-rekisteriin.
MOVWF EEADR	; Siirretään se EEADR-rekisteriin, tämä se osoite josta luetaan dataa Eepromin muistista.
BSF STATUS, RP0	; Mennään Bank 3.
BCF EECON1, EEPGD	; Asetetaan luku datamuistiin eli EEPROM:in alueelle.
BCF EECON1, RD	; Aloitetaan luku.
BCF STATUS, RP1	; Mennään Bank 2.
MOVF EEDAT, W	; Siirretään EEDAT sisältö W-rekisterin sisällöksi.
BCF STATUS, RP0	; Mennään Bank 0

**Esimerkki 7** EEPROM:in luku /11/

Kirjoittaminen on hieman monimutkaisempaa. Esimerkissä 8 näytetään, kuinka kirjoittaminen onnistuu jollakin datalla(DATA\_EE\_DATA), johonkin osoitteeseen(DATA\_EE\_ADDR).

BCF STATUS, RP0	; Mennään Bank 2
BSF STATUS, RP1	;
MOVLW DATA_EE_ADDR	; Siirretään osoite W-rekisteriin.
MOVWF EEADR	; Siirretään osoite EEADR-rekisteriin, kirjoitus osoitteeksi EEPROM:iin.
MOVLW DATA_EE_DATA	; Siirretään data W-rekisteriin.
MOVWF EEDAT	; Siirretään data EEDAT-rekisteriin kirjoitettavaksi EEPROM:iin.
BSF STATUS, RP0	; Mennään Bank 3.
BCF EECON1, EEPGD	; Kirjoitetaan data muistiin eli EEPROM:in alueelle.
BSF EECON1, WREN	; Sallitaan kirjoitus.
BCF INTCON, GIE	; Otetaan keskeytykset pois käytöstä, ettei kirjoitus häiriinny.
MOVLW 55h	; Siirretään 55h W-rekisteriin.
MOVWF EECON2	; Siirretään 55h EECON2-rekisteriin.
MOVLW AAh	; Siirretään AAh W-rekisteriin.
MOVWF EECON2	; Siirretään AAh EECON2-rekisteriin.
BSF EECON1, WR	; Asetetaan WR bitti EECON1-rekisteristä ja aloitetaan kirjoitus.
BSF INTCON, GIE	; Sallitaan keskeytykset.
SLEEP	; Odotetaan niin kauan kunnes keskeytys tulee, että kirjoitus valmis.
BCF EECON1, WREN	; Otetaan kirjoitus pois käytöstä.
BCF STATUS, RP0	; Mennään Bank 0

**Esimerkki 8** EEPROM:in kirjoitus /11/

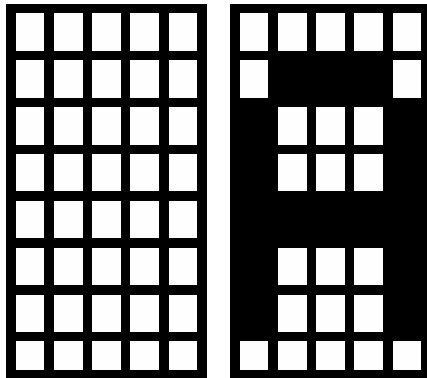
Ehkä huomion arvoiset asiat tässä rutiinissa ovat 55h ja AAh kirjoitus EECON2-rekisteriin peräkkäin. Tämä on datakirjan mukainen kirjoituksen aloitusrutiini.

#### 4 DISPLAYTEK YJ-162A SAMSUNG KS0066U LCD-MODUULI

KS0066U on Samsungin valmistama LCD-ohjainmikropiiri, jota käytetään yleisesti monessa eri elektroniikkasovelluksessa. Suosioon syynä lienee moduulien halpuus ja helppo ohjelmitavuus. Displaytekin valmistama moduuli koostuu LCD-näytöstä ja sen ohjaamiseen tarkoitettusta mikropiiristä KS0066U.

##### 4.1 LCD-tekniikka

Nestekidenäyttö LCD koostuu monesta pienestä sähköisesti aktiivisesta solusta, joihin johtamalla sähkökenttä saadaan ne näyttämään tummilta. Moduulin näyttö koostuu kahdesta tekstirivistä, joihin mahtuu jokaiseen riviin 16-merkkiä. Jokainen merkki koostuu 5 x 8 kokoisesta pistematriisista (kuva 9).



**Kuva 9** 5 x 8 pistematriisi ja A-kirjain

Jokaisen matriisin välissä on yhden pikselin levyinen väli, jotta merkit erottuvat toisistaan. On myös mahdollista käyttää 5 x 11 kokoisia merkkejä, mutta tällöin näytön rivimäärä supistuu yhteen. Näytössä valmiina on jo käytössä merkkejä 5 x 8 kokoisille matriiseille 208 kappaletta ja 5 x 11 kokoisia 32 kappaletta. Lisäksi on mahdollista ohjelmoida omia merkkejä 5 x 8 kokosiin matriiseihin 8 kappaletta. Näyttö käyttää ASCII-standardin mukaista merkkikoodausta, joten merkkitaulukko on kuvan 10 mukainen.

Char. code

	0	0	0	0	0	0	1	1	1	1	1	1		
	0	0	0	1	1	1	1	0	0	1	1	1	1	
	0	1	1	0	0	1	1	1	1	0	0	1	1	
	0	0	1	0	1	0	1	0	1	0	1	0	1	
xxxx0000			0	a	P	`	P	-	9	E	α	ρ		
xxxx0001			!	1	A	Q	a	q	。	ア	チ	4	ä	Q
xxxx0010			"	2	B	R	b	r	「	イ	ツ	×	β	θ
xxxx0011			#	3	C	S	c	s	」	ウ	フ	E	ε	ω
xxxx0100			\$	4	D	T	d	t	、	エ	ト	ト	μ	Ω
xxxx0101			%	5	E	U	e	u	・	オ	ナ	1	ε	ü
xxxx0110			&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111			'	7	G	W	g	w	フ	キ	ヲ	ラ	Q	π
xxxx1000			(	8	H	X	h	x	イ	ク	ネ	リ	」	α
xxxx1001			)	9	I	Y	i	y	ッ	ク	ル	」	U	
xxxx1010			*	:	J	Z	j	z	エ	コ	ハ	レ	i	〒
xxxx1011			+	;	K	[	k	[	オ	サ	ヒ	ロ	*	斤
xxxx1100			,	<	L	¥	l	¥	ハ	シ	フ	フ	¢	円
xxxx1101			-	=	M	]	m	]	ユ	ズ	ハ	ン	も	÷
xxxx1110			.	>	N	^	n	^	ヨ	セ	ホ	ハ	ん	
xxxx1111			/	?	O	_	o	_	ウ	ツ	マ	°	ö	■

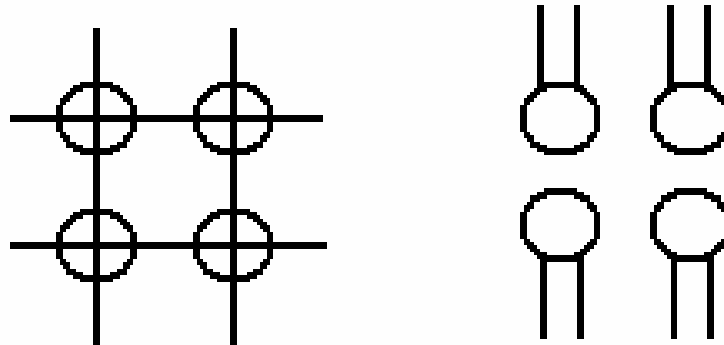
**Kuva 10** Ascii-koodattu merkkikartta, merkkikoolla 5 x 8 /12/

Esimerkiksi 41-heksaluku vastaa binäärilukuna 0100 0001, joka taas ASCII-kartasta luettuna on kirjain A. Kaiken kaikkiaan vaihtoehtoja ASCII-kartassa on siis 256 kappaletta. Karttaan on koodattu numerot, yleiset merkit, yleiset laskutoimitukset, latinalaiset aakkoset, skandinaaviset kirjaimet ja kreikkalaiset aakkoset. Lisäksi on myös japanin katakana-merkkejä, joilla saadaan japanin kielistä tekstiä esitettyä. Nämä ovat mukana myös sen takia, koska näillä japanilaiset voivat kirjoittaa vieraskielisiä erisnimiä ja lainasanoja.

#### 4.2 KS0066U-piirin ominaisuuksien esittely

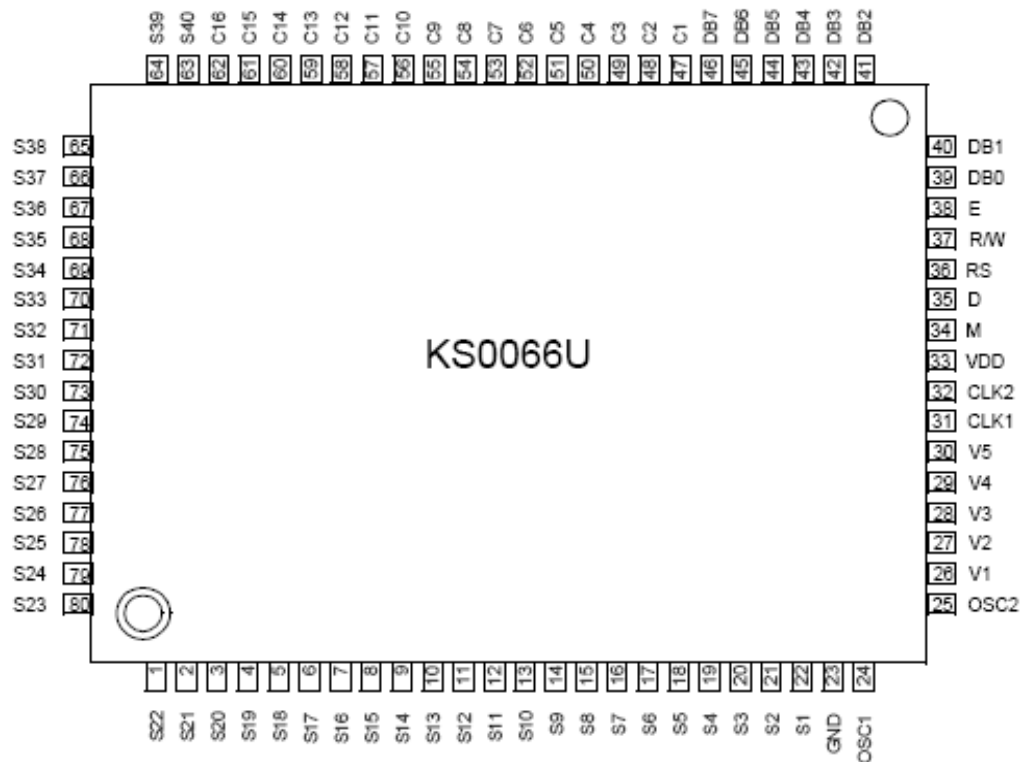
KS0066U-piiri ohjaa piirin jokaista pistettä niin sanotulla ristimatriisi kytkennällä, kuten liitteen 2 sivulta 1 kuvassa 3. Samanlaista kytkentää käytetään myös

tietokoneen näppäimistön nappien lukemiseen. Tällä tekniikalla säästetään johdotusten määrää, koska jos jokaisella pisteelle vedettäisiin oma johtonsa veisi se monta johdinta lisää. Kuvassa 11 esiteltynä hyöty.



**Kuva 11** Ristimatriisi ja normaali

Kuvassa 11 pallot kuvaavat pisteitä ja viivat pisteisiin tulevia johtimia. Kun halutaan ohjata tiettyä pikseliä, täytyy pikseliä ohjata kahdella johdolla, plussa ja miinus jännitteellä. Kun jännite vaikuttaa pikseliin muuttuu se mustaksi. Kuvasta 11 huomaamme, että ristimatriisilla toteutetulla tavalla saadaan neljä pikseliä toimimaan neljällä johtimella, kun taas normaali tekniikka vaatisi kahdeksan johdinta. KS0066U on kahdeksankymmentä jalkainen pintaliitos-mikropiiri, kuten liitteessä 2 sivulla 2.



**Kuva 12** KS0066U-jalkajärjestys (Liite 2 sivu 2)

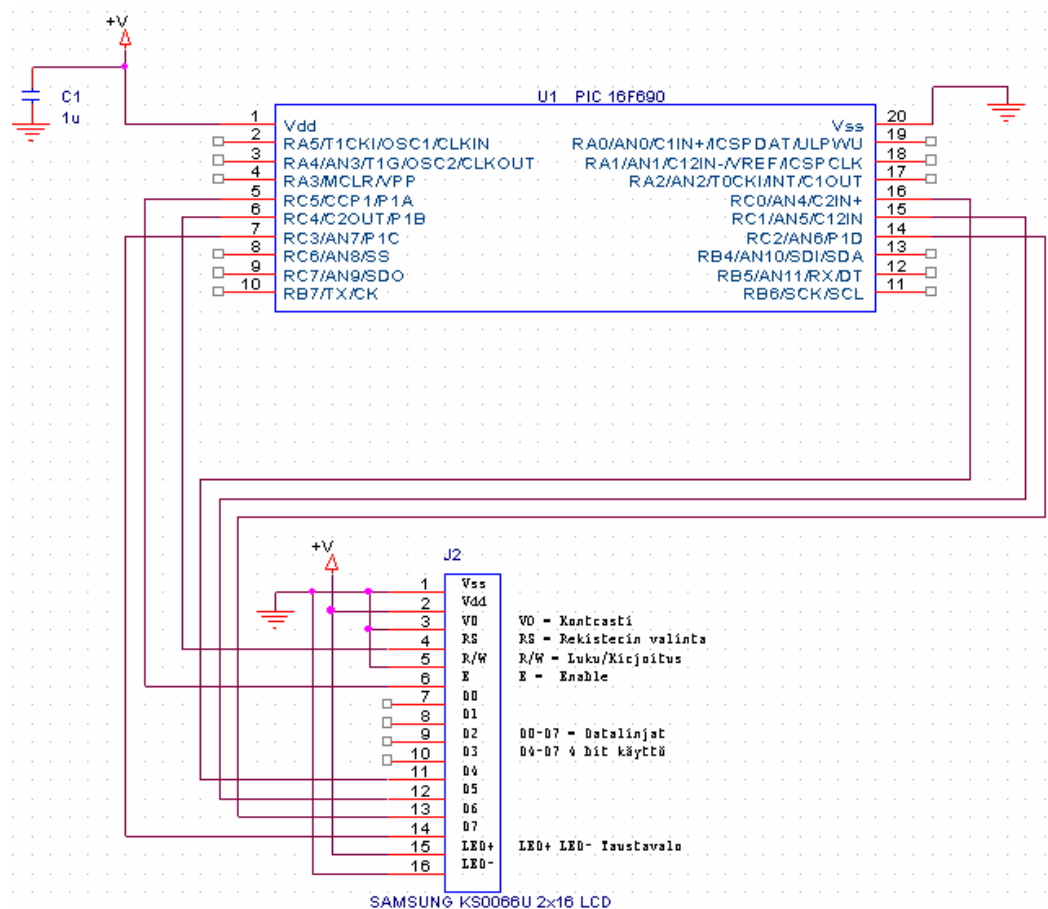
Kuvan 12 jaloista S-alkuiset (S1-S40) ovat pystyjuovia ja C-alkuiset (C1-C16) ovat vaakajuovia. Piirillä on yksistään mahdollista ohjata jopa 16x40 pikseliä, josta voidaan suoraan laskea merkkien määrä kun tiedetään, että yhden merkin koko 5 x 8 pikseliä. Joten pystyriiville mahtuu  $16/8 = 2$  eli kaksi riviä ja vaakariville  $40/5 = 8$  eli kahdeksan merkkiä. Datalinjat ohjausta varten ovat DB0 - DB7, näihin voidaan syöttää dataa joko 8- tai 4-bittisenä. E-jalalla ohjataan onko piiri aktiivinen vai ei aktiivinen. R/W-jalalla ohjataan mikropiirin muistinluku ja -kirjoitus suunta. RS-jalasta valitaan mikropiirin sisäisten rekistereiden väliltä. Kirjoitetaanko vai luetaanko käsky ja kirjoitetaanko tai luetaan näyttömuistista kirjain Lisäksi Vdd ja GND ovat käyttöjännite ja maa. CLK1 ja CLK2 antavat sisäisenkellon ulos, jonka OSC1- ja OSC2-napoihin on syötetty. V1 – V5 ovat biastaso antoja LCD-ohjaamiseen. Kokonainen esittely jalkojen merkityksestä on liitteestä 2 sivulta 3. Vaakamerkkien määrää voidaan lisätä lisäämällä KS0065B-mikropiirejä vaakariveille, tällä menetelmällä päästään jopa 80-merkkiin rivillä (liite 2 sivu 4 kuva 6). Käyttämässäni Displaytekin näyttömoduulissa on vaakarivillä 16-merkkiä ja se käyttää yhtä KS0065B-piiriä

apunaan. Jokainen KS0065B-piiri lisää rivien määrää  $40/5 = 8$  kahdeksalla kappaleella ja kahdenrivin näytöissä 16-kappaleella. Mikropiirissä on sisäinen CGROM (Character Generator ROM), joka on 10080-bitin kokoinen. CGROM on ROM-muisti mikropiirin sisällä (integroitu), joka sisältää kaikki kuvan 2 mukaiset merkit. Koska merkit ovat valmiiksi tehtyjä muistiin, se helpottaa ja nopeuttaa näytön käyttöä. Lisäksi omatekoisille merkeille on oma ohjelmoitava muisti alueensa CGRAM (Character Generator RAM), joka on 64x8-bitin kokoinen. Tämä koko vastaa kahdeksan itse ohjelmoitavan merkin viemää tilaa. Piirillä on vielä näyttötilan muisti DDRAM (Display Data Ram), tämä muisti käsittää kaiken mitä ruudulla näkyy.

#### 4.3 LCD:n ohjaaminen

KS0066U-mikropiiri on HD44780U-mikropiirin kanssa lähes täydellisesti yhteensopiva. Hitachin valmistama HD44780 on hyvin yleisesti käytetty mikropiiri ohjaamaan merkki LCD-näyttöjä. Näyttöjä on saatavissa monia eri kokoja, yhdestä rivistä neljään riviin ja yhden rivin merkkien määrä on jopa neljäkymmentä kappaletta. KS0066U-mikropiiriä voidaan ohjata 8-bittisellä väylällä ja myös 4-bittisellä väylällä. 4-bittisen väylän käyttö vähentää näin johdotusten määrää neljällä. Kun tiedetään kuinka leveää ohjaustapaa käytetään, tämän jälkeen pitää myös ohjaus signaalit RS, R/W ja E kytkeä. E-signaali(enable) signaalilla ohjataan näytön datan ottoa. Jos E-signaali ei ole aktiivinen, ei dataa voida ottaa vastaan. Jos taas E-signaali on aktiivinen, voidaan dataa ottaa vastaan. R/W-signaalilla(read/write) määrätään luetaanko mikropiirin muistista vai kirjoitetaanko mikropiirin muistiin. Jos R/W-signaali on aktiivinen, on kirjoitus käytössä ja taas jos ei ole aktiivinen luku on käytössä. RS-signaali(register select) signaalilla ohjataan laitetaanko tietoa mikropiirin näyttömuistiin vai kirjoitetaanko dataa käskymuistiin. Kuva 13 esittää esimerkki kytkentää, jolla voidaan minimissään ohjata näyttöä PIC16F690-mikroprosessorilla. Kuvasta 13 nähdään kuinka datalinjat D4 - D7, ovat kytkettyinä suoraan prosessorin RC0 - RC3 portteihin. RS-signaali on kytketty porttiin RC4 ja E-signaali porttiin RC5. Tässä kytkennässä on R/W-signaali kytketty maihin, koska silloin se on vain kirjoitus-tilassa. Tämä johtuu taas siitä, että näytönohjaus on toteutettu vakiomittaisilla

viiveillä käskyjen välissä. Jos vakiomittaisia viiveitä ei käytettäisi, jouduttaisiin käyttämään Busy Flag:in lukemista. Busy Flag ilmoittaa onko piiri suorittamassa käskyä eli onko se varattu. Busy Flag lukeminen taas onnistuu laittamalla R/W-signaali aktiiviseen tilaan, jolloin sallitaan tietojen lukeminen näytönmuistista. Ilman Busy Flag:in käyttöä voidaan ohjelmaa yksinkertaistaa ja jopa nopeuttaa hieman.

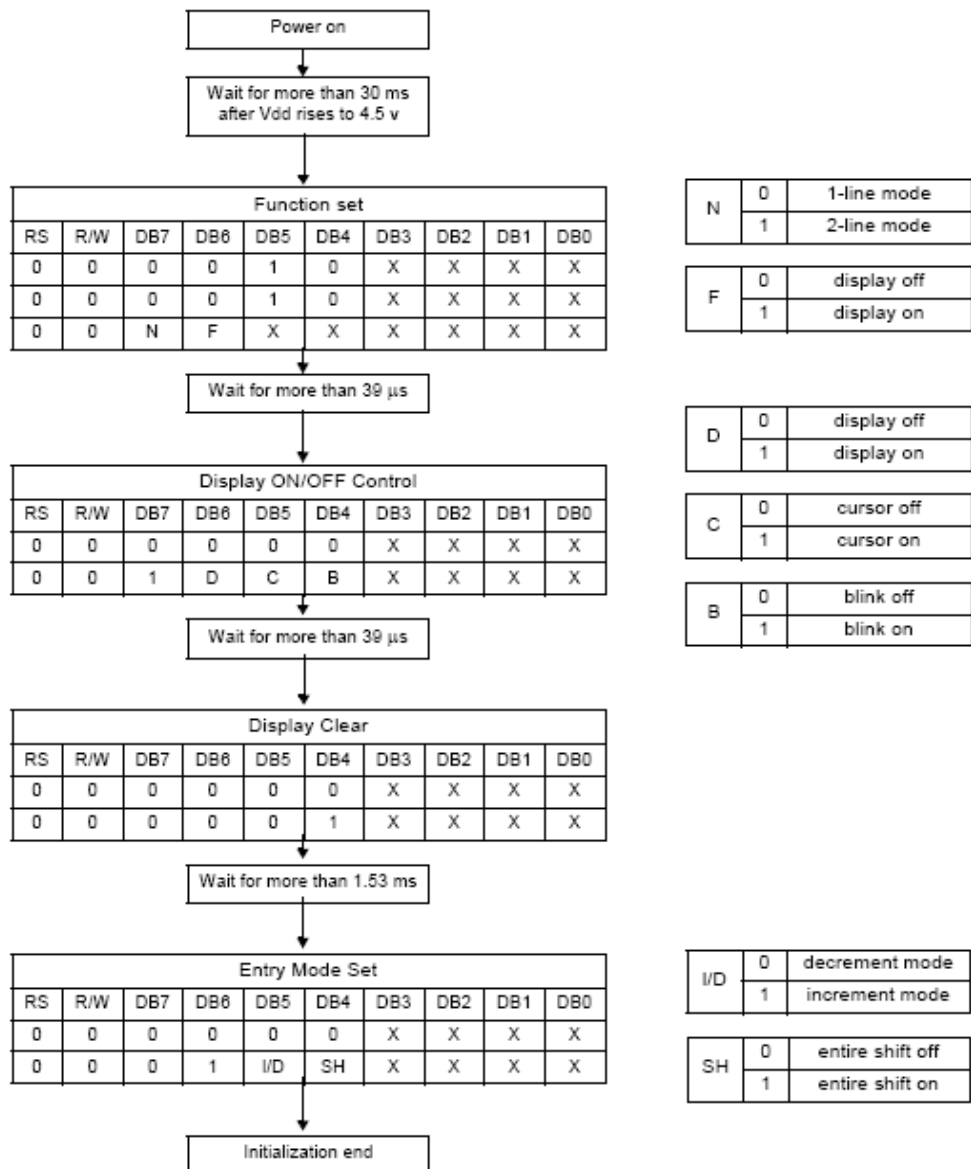


**Kuva 13** Esimerkki kytkentä PIC16F690-mikroprosessoriin

Ennen kuin voidaan ottaa näyttö käyttöön, pitää näyttö alustaa. Alustaminen tapahtuu liitteen 2 sivujen 5 ja 6 tavalla. Sivulla 5 näkyy kuinka alustus tapahtuu 8-bittisellä kytkennällä ja sivulla 6 kuinka alustus tapahtuu 4-bittisellä kytkennällä. 4-bittisessä kytkennässä täytyy ottaa huomioon, että data joudutaan välittämään kahdessa eri 4-bittisessä osassa mikroprosessorille. Tämän jälkeen kun alustus on saatu suoritettua, pitäisi näyttömoduulin ruudulle olla ilmestynyt kursori vilkuttamaan ensimmäisen rivin alkuun. Alustuksen aikana voidaan määrätä



eri alku asetukset näytölle. Kuvassa 14 näkyy kuinka LCD alustetaan 4-bittiseen tilaan. Kuvassa 14, 0 tarkoittaa loogista nolla tasoa ja 1 tarkoittaa loogista ykkös tasoa. X on taas mikä tahansa arvo (Don't Care).



**Kuva 14** 4-Bittinen KS0066U-piirin alustus (Liite 2 sivu 6)

Alussa näyttöön kytketään jännitteet ja odotetaan 30 millisekuntia, jotta jännite nousisi 4,5 volttiin. Ajat voivat olla pitempiäkin, mutta tämä taas hidastaisi näyttönohjausta. Tämän jälkeen syötetään kolme bittijonoa näytön datalinjoihin, mutta pidetään RS- ja R/W-signaalit nolla tilassa. Ensimmäinen ja toinen bittijono on aina vakio, kolmannella bittijonolla määritellään moniko rivinen näyttö

on ja kytketäänkö näyttö päälle. Näiden arvojen muuttaminen tapahtuu rivien osalta DB7(N)-bitin arvolla ja näytön päälle laitto bitillä DB6(F). Tässä vaiheessa joudutaan odottamaan 39 mikrosekuntia vähintään, tämän jälkeen syötetään kaksi bittijonoa. Näillä bittijonoilla määritellään onko näyttö päällä, onko kursoria, ja välkkyykö kursori. Nämä asetuksen määritellään tämän vaiheen toisella bittijonolla. Bitillä DB6(D) määrätään onko näyttö päällä, bitillä DB5(C) määrätään onko kursori päällä ja bitillä DB4(B) määrätään onko kursorin vilkkuminen päällä. Tässä vaiheessa odotellaan 39 mikrosekuntia. Tämän jälkeen syötetään kaksi bittijonoa, joilla tyhjennetään näyttö. Tämän jälkeen odotellaan 1,53 millisekuntia. Sitten syötetään kaksi bittijonoa, joilla määritellään onko merkisyötöä lisäävä ja siirretäänkö ruutua mukana kun kirjoitetaan. Merkisyötön lisääminen määritellään toisesta bittijonosta bitistä DB5(I/D) ja näytön siirto bitistä DB4(SH). Tämän jälkeen näytön alustus on valmis. Jokaisen bittijonon lopussa täytyy E-signaali nostaa ylös ja laskea alas, koska näyttö ottaa datan vastaan E-signaalin laskevalla reunalla. Kun taas R/W- ja RS-signaalit pysyvät vakioina koko alustuksen ajan. KS0066U-mikroprosessoriin on esiohjelmoituja käskyjä näytön ohjaamiseen, kuvassa 15 on kaikki käskyt lueteltu. Käskyt kokonaisuudessaan löytyvät liitteestä 2 sivuilta 7 - 12.

Instruction	Instruction Code											Description	Execution time (fosc=270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Clear Display	0	0	0	0	0	0	0	0	0	0	1	Write '20H' to DDRAM and set DDRAM address to '00H' from AC	1.53 ms
Return Home	0	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to '00H' from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53 ms
Entry Mode Set	0	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39 μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B		Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39 μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-		Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39 μs
Function Set	0	0	0	0	1	DL	N	F	-	-		Set interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line) and, display font type (F:5×11dots/5×8 dots)	39 μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0		Set CGRAM address in address counter.	39 μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0		Set DDRAM address in address counter.	39 μs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0		Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 μs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0		Write data into internal RAM (DDRAM/CGRAM).	43 μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0		Read data from internal RAM (DDRAM/CGRAM).	43 μs

**Kuva 15** KS0066U-mikroprosessorin käskykanta (liite 2 sivu 12)

Käskey "Clear Display" tyhjentää koko näyttömuistin, ja täten ollen tyhjentään koko näyttö.

Käskey "Return Home" palauttaa kursorin ensimmäisen rivin alkuun.

Käskeyllä "Entry Mode Set" määrätään kursorin liikkumissuunta I/D ja liikkuko näyttö kursorin mukana SH.

Käskeyllä "Display ON/OFF control" onko näyttö päällä D, onko kursori päällä C ja vilkkuuko kursori B.

Käskeyllä "Cursor or Display Shift" määrää kumpaan suuntaan kursori liikkuu S/C ja kumpaan suuntaan näyttötila siirtyy R/L.

Käskeyllä "Function Set" määritellään ohjataanko näyttöä 8 vai 4 bittisesti DL, myös määritellään kuinka monta rivinen näyttö on N ja minkä kokoisia fontteja käytetään F.

Käskeyllä "Set CGRAM Address" määrätään mihinkä muistipaikkaan CGRAM sisällä talletetaan omatekoinen merkki biteillä AC0-AC5.

Käskeyllä "Set DDRAM Address" määrätään missä kohtaan näyttömuistin kursori on biteillä AC0-AC6.

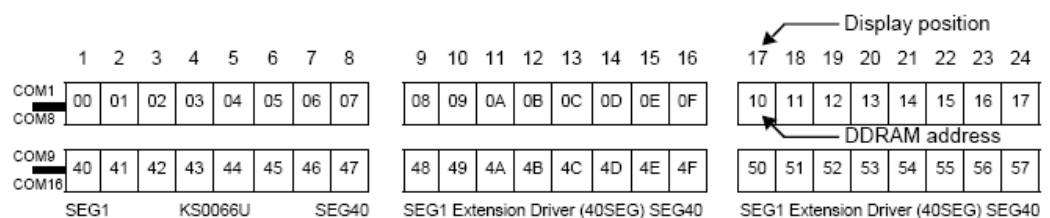
Käskeyllä "Read Busy Flag and Address" luetaan Busy Flag:in tila.

Käskeyllä "Write Data to RAM" kirjoitetaan data näyttömuistiin DDRAM tai kirjoitetaan CGRAM-muistiin oman merkin datat. Kun kirjoitetaan DDRAM-muistiin dataa, laitetaan se kuvan 2 mukaisen ASCII-taulukon merkien joukosta. Esimerkiksi jos halutaan kirjoittaa merkki 0 näytölle laitetaan dataa sisään 00110000 datalinjoihin D0-D7.

Käskeyllä "Read Data from RAM" voidaan lukea tietoa D0-D7 osoittamasta osoitteesta DDRAM- tai CGRAM-muistista.

### Kuva 15 Käskyjen selitykset

Näyttömuisti DDRAM on jaettu näkyvään näyttöön kuvan 16 mukaisiin segmentteihin. Kuvassa ylänumerot ilmaisevat pystyrivejä ja vaakarivillä olevat kaksi jonoa kuvaavat kahta vaakariviä.

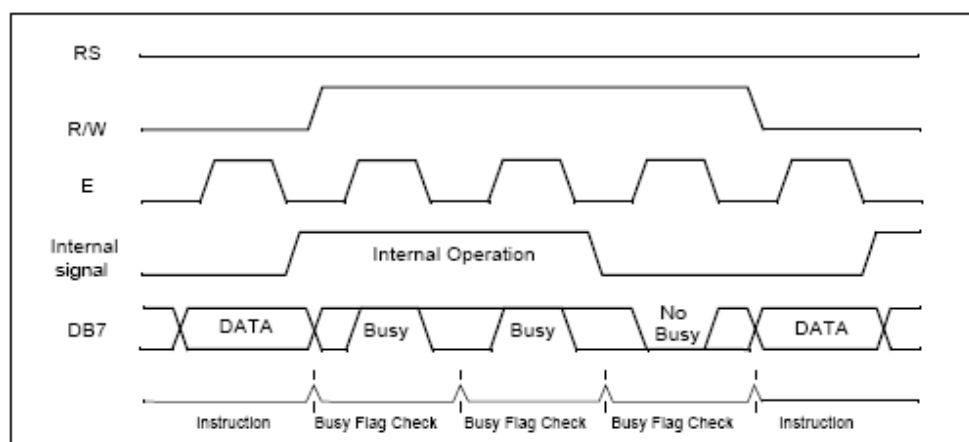


**Kuva 16** Kaksirivisen LCD-näytön DDRAM-muistiavaruus (liite 2 sivu 13 kuva 2)

Esimerkiksi jos näyttönä on kaksi rivinen ja 16-merkkinen näyttö, olisi osoitealue ensimmäiselle riville 00 - 0F. Muu osa menisi yli ja ei olisi enää näkyvissä ruudulla, jollei sitten näyttöä siirrettäisi käskeyllä. Toisen rivin osoitteen taas menisivät 40 - 4F alueella.

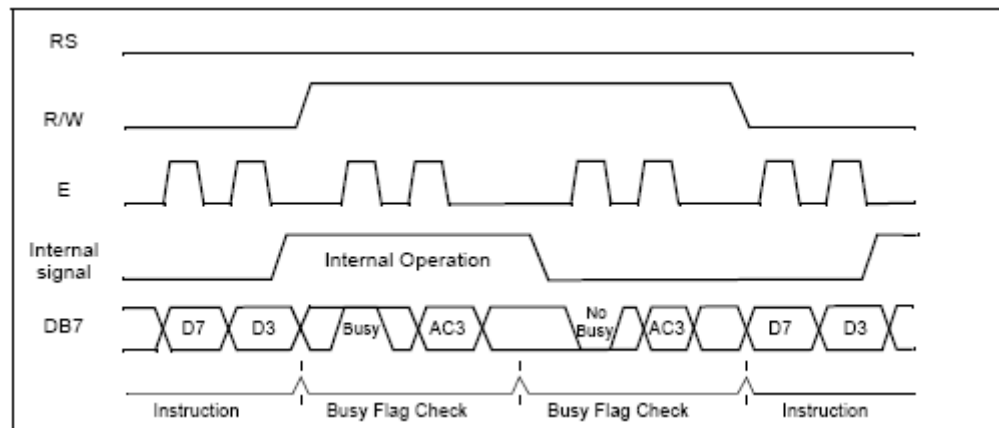
#### 4.4 Ajoitukset

Piirin ohjaaminen eroaa hieman dataväylän koon muuttumisen johdosta. 8-bittisellä väylällä ohjatulle piirille voidaan suoraan syöttää kahdeksan bittiä rinnakkaisväylään. Kun taas 4-bittisellä dataväylällä, joudutaan data jakamaan kahden osaan. Kuvassa 17 näkyy kuinka KS0066U-mikroprosessoria ohjataan 8-bittisellä kytkennällä ja Busy Flag:iä apuna käyttäen. Kuvassa 17 RS on register select, R/W on Read/Write, E on Enable ja DB7 on dataväylän bitit DB0 - DB7.



**Kuva 17** KS0066U-ajotukset 8-bittisellä dataväylällä (liite 2 sivu 14 kuva 1)

Ensin syötetään käskydata väylään DB7 eli DATA. Tämän jälkeen käytetään E-signaalia ylhäällä ja laskureunalla data luetaan muistiin. Tämän jälkeen nostetaan R/W-signaali ylös, jotta saataisiin luettua Busy Flag eli tutkia onko piiri varattu. Busy Flag:iä luetaan niin kauan, kunnes saadaan ei varattu tila eli Busy Flag ei ole aktiivinen. Jokaisen Busy Flag:in luennan aikana täytyy E-signaali käyttää ylhäällä. Kun saadaan luettua että No Busy eli ei varattu voidaan jatkaa suoritusta. Tämän jälkeen lasketaan R/W-signaali alas ja laitetaan uusi data sisään väylään. 4-bittisellä kytkennällä datan kirjoitus on hieman erilainen(kuva 18). Dataväylä on nyt 4-bittinen, kuvassa D7 tarkoittaa neljää enitenmerkitsevää bittiä ja D3 tarkoittaa neljää vähitenmerkitsevää bittiä.



**Kuva 18** KS0066U-ajoitukset 4-bittisellä dataväylällä (liite 2 sivu 14 kuva 2)

Ensin laitetaan neljä enitenmerkitsevää bittiä (MSB) D7 datalinjalle ja käytetään E-signaali ylhäällä. Tämän jälkeen syötetään neljä vähitenmerkitsevää bittiä (LSB) D3 ja käytetään E-signaali ylhäällä. Busy Flag:in luku tapahtuu samalla lailla kuin 8-bittisellä kytkennällä, mutta data vaan jaetaan kahteen osaan. Tämän jälkeen voidaan syöttää taas uusi data sisään. Esimerkiksi, merkkien kirjoittaminen näytölle tapahtuu muuten samalla lailla kuin kuvissa 17 ja 18, mutta RS-signaalia käytetään ylhäällä, jolloin data menee näyttömuistiin. Busy Flag:in käyttö ei välttämättä ole tarpeellista hitaissa sovelluksissa, koska näyttö toimii myös hitaalla käskytyksellä. Busy Flag:in poistaminen yksinkertaistaa koodia.

#### 4.5 Ohjelmointi

Ohjelmointiesimerkkinä käytän tässä PIC16F690-piirillä ohjattua Displaytek:in 2x16 LCD-näyttöä Samsungin KS0066U-piirillä. Ohjaamiseen tarkoitettu kytkentä on kuvassa 13. Ensin esitellään kuinka välitetään käsky LCD-näytölle esimerkin 9 mukaisella esimerkkikoodilla. Aliohjelmalle täytyy välittää W-rekisterissä suoritettava käsky, ennen kuin kutsutaan aliohjelmaa LCD\_CTRL4.

1	LCD_CTRL4		
2	BCF	PORTC, LCD_RS	;RS-signaali alas
3			
4	MOVWF	LCD_TEMP	;otetaan W-rekisterin sisältö talteen LCD_TEMP
5			
6	SWAPF	LCD_TEMP,W	; vaihdetaan 4-ylintä bittä neljään alimpaan bittiin
7			
8	ANDLW	0x0F	; W-rekisterin ylimmät bitit nolllaksi
9	MOVWF	PORTC	; Siirretään W-rekisterin sisältö porttiin C
10			
11	BSF	PORTC,LCD_E	; E-signaali ylös
12	BCF	PORTC,LCD_E	; E-signaali alas
13			
14	MOVWF	LCD_TEMP, W	; siirretään LCD_TEMP W rekisteriin
15	ANDLW	0x0F	; ylimmät bitit nollliksi
16	MOVWF	PORTC	;W->PORTC
17			
18	BSF	PORTC,LCD_E	; E-signaali ylös
19	BCF	PORTC,LCD_E	; E-signaali alas
20			
21	CALL	VIIVE500	; viive 500 mikrosekuntia
22			
23	RETURN		

Selitykset:

Rivi 1. Aliohjelma alkaa LCD\_CTRL4

Rivi 2. Asetetaan RS-signaali alas rivillä 2, joten kirjoitus kohdistuu käskyrekisteriin.

Rivi 4. Siirretään W-rekisterin sisältö talteen muuttujaan LCD\_TEMP

Rivi 6. Siirretään 4 ylintä bittä alimmiksi biteiksi ja 4 alinta ylimmiksi, koska LCD vaatii eniten merkitsevät bitit ensiksi siirrettäväksi datalinjaan 4 bitin ohjauksella. Lopuksi siirretään se W-rekisteriin.

Rivi 8. Nollataan W-rekisteristä ylimmät bitit nollliksi, jotta portin muu toiminta ei häiriintyisi ja jätetään siirrettävä 4 bittinen luku koskemattomaksi.

Rivi 9. Siirretään W-rekisterin sisältö ulostuloon porttiin-C.

Rivi 11-12. Pidetään LCDen E-signaalia ylhäällä 1 mikrosekunnin ajan, kun E-signaali laskee ottaa se portissa C olevan datan muistiin.

Rivi 14. Siirretään LCD\_TEMP sisältö W-rekisteriin.

Rivi 15. Nollataan ylimmät bitit, jotta saadaan puhdas 4-bittinen luku W-rekisteriin. Datan 4 alinta bittia

Rivi 16. Siirretään W-rekisterin sisältö ulostulo porttiin C.

Rivi 18 - 19. Sama kuin rivit 11 - 12.

Rivi 21. Kutsutaan viivesilmukkaa 500 mikrosekuntia, että LCD ehtii ottamaan datan vastaan ja tulostamaan sen ruudulle.

Rivi 23. Palataan aliohjelmasta takaisin kutsupaikan seuraavalle riville.

### Esimerkki 9 LCD:n ohjauskomentojen välitys näytölle

Lisäksi tarvitaan vielä näytön alustuksiin pelkän yhden neljä bittisen luvun porttiin C syöttävä ohjelma, lähdekoodi tähän on esimerkissä 10. Esimerkin 10 mukaiseen ohjelmaa täytyy välittää käskybitit W-rekisterin sisällä.

1	LCD_CTRL			; Aliohjelma alkaa
2		BCF	PORTC, LCD_RS	; RS-signaali alas
3				
4		ANDLW	0x0F	; Ylimmät bitit nolliksi
5				
6		MOVWF	PORTC	; W-rekisterin sisältö PORTC
7				
8		BSF	PORTC,LCD_E	; E-signaali ylös
9		BCF	PORTC,LCD_E	; E-signaali alas
10				
11		CALL	VIIVE500	; Kutsutaan viivesilmukkaa 500 us.
12	RETURN			; Palataan kutsupaikan seuraavalle riville.

Selitykset:

Rivi 1. Aliohjelma alkaa.

Rivi 2. Asetetaan RS-signaali alas.

Rivi 4. Nollataan ylimmät bitit W-rekisteristä, ettei turhia bittejä siirry portti C:hen.

Rivi 6. Siirretään W-rekisterin sisältö ulostuloporttiin C.

Rivit 8-9. Käytetään E-signaalia ylhäällä ja data asettuu LCD:lle E:n laskevalla reunalle.

Rivi 11. Kutsutaan viivesilmukkaan 500 mikrosekuntia.

Rivi 12. Palataan kutsupaikkaan seuraavan käskyn suoritukseen.

### Esimerkki 10 LCD\_CTRL:in lähdekoodi.

Lisäksi tarvitaan vielä viivesilmukat, jonka lähdekoodit ovat esimerkissä 11. VIIVE500 on vain 500 mikrosekuntia kestävä viive 4 MHz PIC-mikroprosessorilla ja X\_VIIVE500 on viive, joka käy 500 mikrosekunnin viiveen niin monta kertaa kuin W-rekisteriin on syötetty ohjelman kutsuhetkellä.



```

1 ;*****
2 ;Viive 500us
3 ;*****
4 VIIVE500 ; Aliohjelma alkaa
5          MOVLW      d'165' ; Laitetaan rekisteriin W 165 desimaali
6          MOVWF      VIIVE   ; Kopioidaan W:n sisältö VIIVE muuttujaan
7 VIIVE500_LOOP ; Apu hyppykohta
8          DECFSZ     VIIVE,F ; Vähennetään yhdellä kunnes VIIVE on nolla
9          GOTO       VIIVE500_LOOP ; Hypätään kohtaan VIIVE500_LOOP
10 VIIVE500_END ; Apu hyppykohta
11          RETURN     ; Paluu apuohjelmaan
12
13
14 ;*****
15 ;W-rekisteri kertaa 500us
16 ;*****
17 X_VIIVE500 ; Aliohjelma alkaa
18          MOVWF      VIIVEX  ; Kopioidaan tuotu W:n arvo VIIVEX muuttujaan
19 X_VIIVE500_LOOP ; Apu hyppykohta
20          CALL       VIIVE500 ; Kutsutaan VIIVE500 aliohjelmaa
21          DECFSZ     VIIVEX,F ; Vähennetään yhdellä kunnes VIIVEX on 0
22          GOTO       X_VIIVE500_LOOP ; Hyppää X_VIIVE500_LOOP
kohtaan
23 X_DELAY500_END ; Apu hyppykohta
24          RETURN     ; Paluu kutsupaikkaan

Selitykset:

Rivi 4. Aliohjelma alkaa.
Rivi 5. Laitetaan W-rekisteriin luku 165 desimaalisena.
Rivi 6. Kopioidaan W-rekisterin sisältö VIIVE muuttujaan.
Rivi 7. Apu hyppykohta vähennyslasku silmukalle.
Rivi 8. Vähennetään VIIVE muuttujaa kunnes nolla, kun VIIVE on nolla hypätään seuraavan käskyn yli.
Rivi 10. Apu hyppykohta, VIIVE500 lopetus.
Rivi 11. Paluu kutsupaikkaan.
Rivi 17. Aliohjelma alkaa.
Rivi 18. Siirretään tuotu W:n arvo VIIVEX muuttujaan.
Rivi 19. Apu hyppykohta vähennyslasku silmukalle.
Rivi 20. Kutsutaan aliohjelmaa VIIVE500.
Rivi 21. Vähennetään VIIVEX muuttujaa kunnes nolla, kun VIIVEX on nolla hypätään seuraavan käskyn yli.
Rivi 22. Hyppää kohtaan X_VIIVE500_LOOP.
Rivi 23. Apu hyppykohta.
Rivi 24. Paluu kutsupaikkaan.

```

### Esimerkki 11 Viivesilmukat VIIVE500 ja X\_VIIVE500

Näiden aliohjelmien avulla voidaan tehdä LCD-näytölle alustusrutiini, esimerkin 12 mukaan ja kuvan 14 osoittamalla tavalla.

```

1 LCD_ALUSTUS_4BIT
2
3     CLRFB          PORTC                ; Nollataan Portti-C
4
5     ; 30ms power viive
6     MOVLW         0x03C                ; 3Ch = 60d
7     CALL          X_VIIVE500           ; 60x500us=30ms
8
9     MOVLW         b'00000010'         ; 0x02
10    CALL          LCD_CTRL
11
12    MOVLW         b'00000010'         ; 0x02
13    CALL          LCD_CTRL
14
15    ; N ja F 0000NF00 N=1 2line F=1 display on
16    MOVLW         b'00001100'
17    CALL          LCD_CTRL
18
19    ; wait more than 39us
20    MOVLW         0x02                 ; 2x500us=1000us
21    CALL          X_VIIVE500
22
23    MOVLW         b'00000000'
24    CALL          LCD_CTRL
25
26    ; 00001DCB
27    MOVLW         b'00001111'
28    CALL          LCD_CTRL
29
30    ; wait more than 39us
31    MOVLW         0x02                 ; 2x500us=1000us
32    CALL          X_VIIVE500
33
34    ; 0
35    MOVLW         b'00000000'
36    CALL          LCD_CTRL
37
38    ; 1
39    MOVLW         b'00000001'
40    CALL          LCD_CTRL
41
42    ; wait more than 1.53ms
43    MOVLW         0x04
44    CALL          X_VIIVE500           ; 4x500us = 2000us 2ms
45
46    ; 0
47    MOVLW         b'00000000'
48    CALL          LCD_CTRL
49
50    ; 000001(I/D)(SH) ,SH=0.
51    MOVLW         b'00000110'
52    CALL          LCD_CTRL
53
54    ; wait more than 1.53ms
55    MOVLW         0x04
56    CALL          X_VIIVE500           ; 4x500us = 2000us 2ms
57    ;ALUSTUS valmis
58 RETURN

```

**Esimerkki 12** LCD:n alustus PIC16F690-prosessorille

Rivi 1. Aliohjelman kutsu.  
Rivi 3. Varmistetaan että ulostuloportti C on tyhjä nollaamalla se.  
Rivit 6-7. Tehdään 30 millisekunnin viive, että näytön jännite ehtii nousta yli 4,5 voltin.  
Rivit 9 - 13. Syötetään kahteen kertaan 0x02h arvo datalinjaan, alustusrutiinin mukaan.  
Rivit 16 - 17. Määritellään näytön koko ja laitetaanko näyttö päälle.  
Rivit 20 - 21. Odotellaan 39 mikrosekuntia minimissään, koodissa 1000 mikrosekuntia.  
Rivi 20 - 23. Syötetään 0x00h datalinjaan.  
Rivit 27 - 28. Syötetään 0x0Fh datalinjaan, määritellään onko näyttöpäällä, onko kursori käytössä ja välkkykö se.  
Rivit 31 - 32. Odotellaan 39 mikrosekuntia vähintään, koodissa 1000 mikrosekuntia.  
Rivit 35 - 36. Syötetään datalinjaan 0x00h data.  
Rivit 39 - 40. Syötetään 0x01 data datalinjaan.  
Rivit 43 - 44. Odotellaan 1,53 millisekuntia, koodissa 2 millisekuntia.  
Rivit 47 - 48. Syötetään datalinjaan 0x00h.  
Rivit 51 - 52. Syötetään datalinjaan data 0x06h, määritellään onko kursori lisäävä ja liikkuuko ruutu kursorin mukana.  
Rivit 55 - 56. Odotellaan 1.53 millisekuntia, koodissa 2 millisekuntia.  
Rivi 58. Paluu kutsupaikkaan.

### **Esimerkki 12** Rivien selitykset

LCD:n alustus on nyt valmis, tämän jälkeen ruudun pitäisi olla tyhjä. Ensimmäisellä rivillä oikealla pitäisi vilkkua kursori. Nyt LCD-näyttö on kirjoitusvalmis käyttöä varten. Merkin kirjoitus ruudulle tapahtuu esimerkin 13 mukaan, ennen aliohjelman kutsua täytyy merkin ASCII-koodi olla W-rekisterissä valmiina.

```

1 LCD_MERKKI_4BIT          ; Aliohjelma alkaa
2
3          ;BCF          PORTA, LCD_RW ; RW-signaali alas, jos käytössä
4
5          MOVWF        LCD_TEMP      ; Otetaan W-rekisterin sisältö talteen
6          SWAPF        LCD_TEMP, W   ; Vaihdetään ylimmät bitit alemmiksi 0f -> f0 -> W
7
8          ANDLW        0x0F          ; Ylimmät bitit nolliksi
9          MOVWF        PORTC         ; W->PORTC
10
11         BSF          PORTC, LCD_RS  ; RS-signaali ylös
12
13         BSF          PORTC, LCD_E   ; Käytetään E-signaalia ylhäällä
14         BCF          PORTC, LCD_E   ;
15
16         BCF          PORTC, LCD_RS  ; RS-signaali alas
17
18         MOVF         LCD_TEMP, W    ; Siirretään talteen otettu W takaisin W:n sisällöksi
19
20         ANDLW        0x0F          ; Ylimmät bitit nolliksi
21         MOVWF        PORTC         ; W->PORTC
22
23         BSF          PORTC, LCD_RS  ; RS-signaali ylös
24
25         BSF          PORTC, LCD_E   ; Käytetään E-signaalia ylhäällä
26         BCF          PORTC, LCD_E   ;
27
28         BCF          PORTC, LCD_RS  ; RS-signaali alas
29
30         CALL        VIIVE500       ; Kutsutaan 500us viivettä
31
32 RETURN                    ; Paluu

```

Selitykset:

Rivi 1. Aliohjelma alkaa.

Rivi 3. Jos R/W-signaali käytössä, täytyisi laskea R/W-signaali alas.

Rivit 5-6. Otetaan W- rekisterin sisältö talteen ja käännetään W-rekisterin sisältö väärinpäin. Eli alimmat 4-bittia ylempiä ja ylempät alemmiksi. Tämä sen takia, jotta voitaisiin ensin laittaa dataväylään eniten merkitsevät bitit näytön spesifikaatioiden mukaan.

Rivit 8-9. Nollataan ylimmät bitit W-rekisteristä, ettei tule virhe signaaleja datalinjaan.

Rivit 11–16. Nostetaan RS-signaali ylös, eli laitetaan dataa näytön näyttömuistiin, käytetään E-signaalia ylhäällä.

Lasketaan RS-signaali takaisin alas.

Rivi 18–21. Otetaan W-rekisterin talteen otettu data, nollataan siitä 4 eniten merkitsevää bittia ja syötetään se data-väylään. Tämä sen takia että saadaan nyt vähiten merkitsevät bitit datalinjaan.

Rivit 23–28. Sama kuin riveillä 11 - 16.

Rivi 30. Kutsutaan 500 mikrosekunnin viivettä, että data ehtii asettumaan näytölle.

Rivi 32. Palataan kutsuohjelmaan.

### Esimerkki 13 Merkin kirjoitus LCD:lle

Näillä viidellä esimerkikoodilla pystytään jo ohjaamaan täysin LCD-näyttöä. Muutamalla valmiiksi tehdyllä ohjelmalla voidaan näytönohjausta hieman helpottaa. Esimerkissä 14 on kuinka LCD:n rivin vaihto onnistuu ja kuinka siirrytään ensimmäiselle riville.

```
1 ;*****
2 ;vaihtaa lcd näytön rivin siirtyä osoitteeseen 40 eli rivin 2 alkuun
3 ;*****
4 TOISELLE_RIVILLE
5         ; 1AAA AAAA
6         ; 40d -> 28h -> 0010 1000 b
7         ; 1010 1000
8         ; A8
9         MOVLW      b'10101000'      ; siirrytään näytön osoitteeseen 0x28h
10        CALL      LCD_CTRL4
11
12 RETURN
13
14 ;*****
15 ;vaihtaa lcd näytön rivin siirtyä osoitteeseen 0 eli rivin 1 alkuun
16 ;*****
17 PALAA_ALKUUN
18        MOVLW      b'10000000'      ; Siirrytään näytön osoitteeseen 0x80h
19        CALL      LCD_CTRL4
20 RETURN
```

Selitykset:

Rivillä 9-10 on laitettu siirto osoitteeksi 28h, joka on näytön paikka toisen rivin alussa.

Kuvassa 16 on ilmoitettu desimaalilukuina näytön eri paikkojen osoitteet.

Syötettä lukema on kuitenkin A8h.

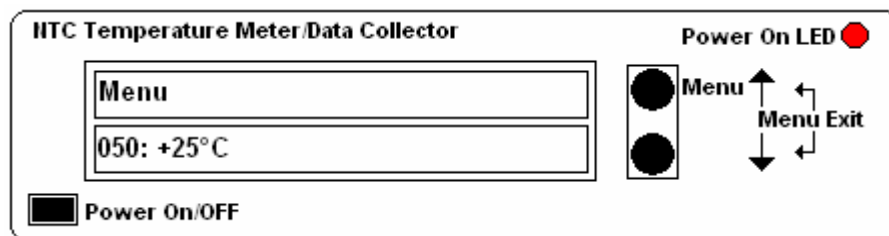
Tämä johtuu siitä koska näytön käsky ”Write Data Ram” vaatii, että ensimmäinen bitti on 1 ja näytön osoite muodostuu siis seitsemästä muusta bitistä.

Rivillä 18–19 on laitettu pelkästään 80h näytölle, ja tässä näytön osoite on 00 joten kursori siirtyy alkuun.

### Esimerkki 14 Siirtyminen ensimmäiselle ja toiselle riville

## 5 KERÄÄVÄ LÄMPÖMITTARI

Työssä tavoitteena oli saada toimiva prototyyppi lämpömittari, joka kerää arvoja tietyn aikavälin välein mikroprosessorin-data EEPROM:iin (tässä 1 sekunti). Laitte kytetään päälle Power On/Off painikkeesta, kun jännite kytketään kytkentään syttyy punainen LED. Punainen LED osoittaa laitteen olevan aktiivinen. Laitteen ollessa aktiivinen, tulostuu LCD:n yläriville valmistajan logo, joka voidaan itse määrätä ohjelmointi vaiheessa. Alemmalle LCD:n riville tulostuu tämän hetkinen lämpötila arvo. Itse laitetta ohjataan kahdella painikkeella, jolla toisella mennään Menu tilaan. Kun painetaan ylös painiketta, päästään Menu tilaan. Menu tilassa tulostuu LCD:n yläriville Menu ja alariville valitun muistipaikan numero ja kyseiseen muistipaikkaan talletetun lämpötilan astemäärä. Talletettujen lämpötilojen arvojen selaus tapahtuu samoilla kahdella painikkeella (ylös tai alas). Menusta poispääsemiseen painetaan vain ylös ja alas painiketta yhtä aikaan alhaalla. Muistipaikkoja lämpömittarissa on 255 kappaletta, näiden täyteen tullessaan aloittaa ohjelma tallettamisen alusta eli muistipaikasta 0. Kuvassa 19 on sovelluksen kuvitettu käyttöpaneeli.



**Kuva 19** Kuvitettu käyttöpaneeli

### 5.1 Kytkennän suunnittelu

Kytkenän suunnittelu tapahtui valmistajan antamien datakirjojen mukaan ja lisäksi NTC-sovitusta tapahtui laskemalla valmistajan mukaiset arvot. Liitteessä 3 näkyy Orcad ohjelmistolla piirretty kytkentäkaavio kokonaisuudessaan. Päävirta kytketään suoraan liittimeen J3, josta johdot menevät suoraan mikroprosessorin Vss ja Vdd napoihin. Päävirta ohjaamaan on tehty kytkin U4. U4 on kytkin, jolla yksinkertaisesti vain kytketään laite aktiiviseksi. Ledillä D2 ilmoitetaan, onko

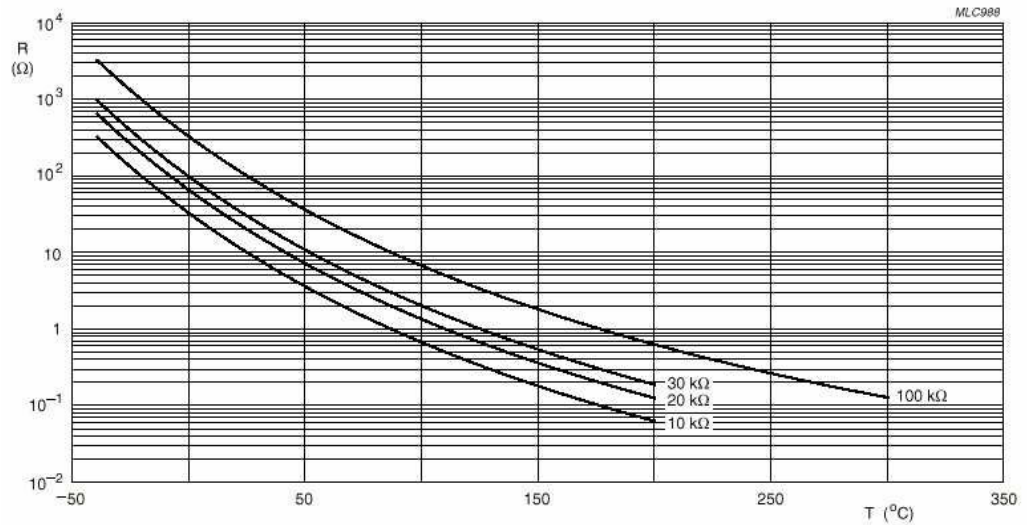
laite aktiivinen. Ledin etuvastus saadaan lasketuksi, kun tiedetään ledin käyttämä jännite 2 V ja ottama virta 20 mA. Kun tiedetään lisäksi vielä, että käyttöjännite on 5 V:a. Tästä voidaan laskea jännite, joka muodostuu vastuksen yli ( $5\text{ V} - 2\text{ V} = 3\text{ V}$ ). Tällöin resistanssi muodostuu seuraavasti.

$$R = \frac{5\text{ V} - 2\text{ V}}{20\text{ mA}} \approx 150\ \Omega$$

Painikkeet SW1 ja SW2, ovat kytketty mikroprosessorin RA3- ja RA4-portteihin. Koska piirin portteja voidaan ohjata tuloiksi ja lähdöiksi, täytyi painikkeiden SW1 ja SW2 kanssa sarjaa kytkeä 1 k $\Omega$ :n vastus (vastukset R1 ja R3). Tällä vastuksella estetään ulostulo käytössä oikosulun syntyminen portin lähdöstä suoraan maatasoon. Muulloin jos tätä vastusta ei olisi, joutuisi mikroprosessori puskemaan liian suurta virtaa ulostuloon. Tässä tapauksessa mikroprosessori hajoaisi tai menisi vikatilaan. Toinen vastus, jota tarvitaan painikkeiden SW1 ja SW2 käytössä, ovat ylösveto-vastukset R2 ja R4. Näillä vastuksilla pidetään jännitetaso porteissa viidessä voltissa, jolloin prosessori tulkitsee tilan loogiseksi ykköseksi. Jollei näitä vastuksia olisi, piirin tila hyppisi arvaamattomasti puolelta toiselle. Kun painetaan painiketta, kytkee kytkin itsensä maatasoon ja täten ollen syntyy mikroprosessorin-porttiin nolla jännite. Tämä jännite vastaa loogista nollatasoa. R2- ja R4-vastuksien koolla ei ole suurempaa merkitystä, kunhan se ei ole turhan pieni. Jos vastus on liian pieni, kuluttaa vastukset virtaa tarpeettomasti, tämän takia vastus on 10 k $\Omega$ :n kokoinen.

## 5.2 NTC

NTC (Negative Temperature Coefficient) on vastus, jolla on negatiivinen lämpötilakerroin. NTC-vastuksen resistanssi suhteessa lämpötilaan on epälineaarinen, tämä vaikeuttaa anturin käyttöä. Kuvassa 20 on NTC-vastuksen resistanssi suhteessa lämpötilaan.



**Kuva 20** NTC-vastuksen resistanssi lämpötilan suhteen /13/

NTC:n arvona yleensä annetaan resistanssi arvo + 25 °C:ssa. Kuvassa 20 on 10 kΩ:n NTC-vastus alimmaisessa käyrässä, tällä on resistanssi + 25 °C:ssa on 10 kΩ:a. Yleensä valmistajilta on valmiiksi lasketut muuntokertoimet tai resistanssit viiden celsiusasteen välein. Liitteessä 3 on käyttämäni NTC-vastuksen valmistajan muunnosarvot 1 kΩ:n vastukselle. Valmistaja on antanut arvot seuraavan kaavan 1 mukaan.

$$R_t / R_{+25^\circ\text{C}} = k \quad (1)$$

Missä  $R_{+25^\circ\text{C}}$  = arvo on resistanssi + 25 °C.

$R_t$  = on taas resistanssi tietyssä lämpötilassa.

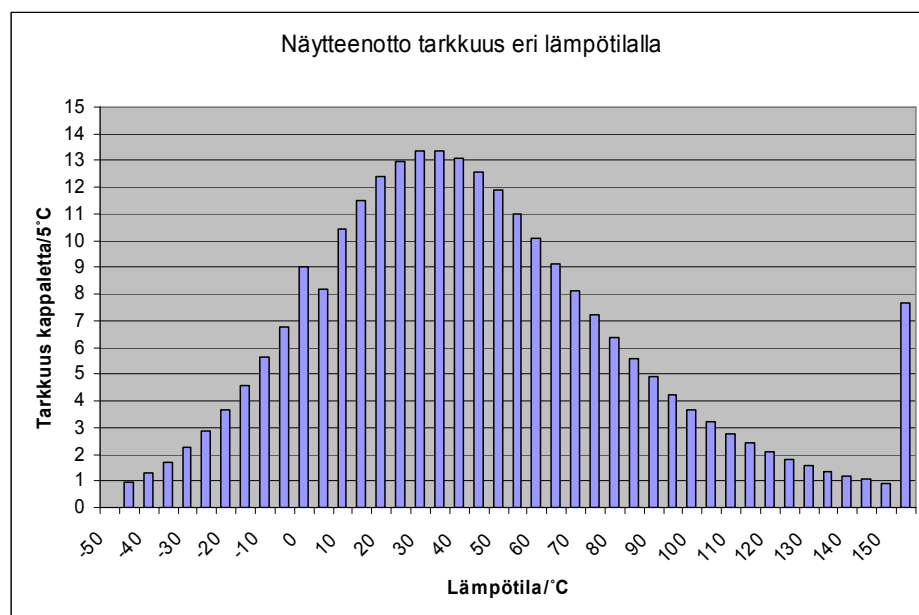
$k$  = valmistajan antama valmis muunnoskerroin(RATIO).

Tästä saadaan laskettua resistanssi tietyllä lämpötilalla ratkaisemalla kaavasta 1  $R_t$ , saadaan seuraava kaava 2.

$$R_t = R_{+25^\circ\text{C}} \cdot k \quad (2)$$



Tällä periaatteella saadaan laskettua kaikkien lämpötilojen resistanssi viiden asteen tarkkuudella. Muiden arvojen laskeminen on mahdollista keskiarvoa hyväksi käyttämällä, koska NTC:n tarkkuus ei pysty enää pitämään resistanssi tarpeeksi vakaana tätä pienemmillä arvoilla. Liitteessä 5 on valmiiksi laskemani resistanssi arvot kaikille valmistajan antamille lämpötila kertoimille. Kuvaajassa 1, on resoluutio eri lämpötiloille 5°C asteen välein. Esimerkiksi 40- ja 45-asteen välissä on 13-arvoa, täten ollen tarkkuus on  $5 / 13 = 0,38$ . Resoluutioksi maksimissaan saadaan 40- ja 45-asteen väliin on 0,38 astetta.



**Kuvaaja 1** Resoluutio eri lämpötiloille

Etuvastus R6 säädettiin 600 Ω:n kokoiseksi, koska tällä saatiin lämpötila erotte-  
lu suurimaksi 10°C - 70°C lämpötiloille. Lisäksi etuvastuksella R6 estetään vir-  
taa nousemasta liian suureksi, jos NTC-vastus olisi pieni tai oikosulussa. Tällai-  
nen tilanne voi esimerkiksi tapahtua, kun lämpötila on 150 astetta tai suurempi.  
Toiseksi vastuksella voidaan hienosäätää lämpötila arvot kohdalleen, jos laske-  
tut eivät pidä paikkaansa. Referenssijännite säädettiin R5-säätövastuksella koh-  
dalleen, jännitearvo saatiin liitteen 5 laskelmista. Kun tiedetään maksimijännite  
mitä laiteelle tulee, voitiin tämän arvon avulla laskea jännite, mitä NTC-  
vastuksen yli maksimissaan olisi. Liitteessä 5 on laskettu referenssijännite, jän-  
nite näkyy punaisella värjättynä ja on 4,026 V:a.

## 6 OHJELMA

Itse ohjelman lähdekoodi on kokonaisuudessaan liitteessä 6. Ohjelman kulkua kuvaava vuokaavio on liitteessä 7. Vuokaavio kuvaa parhaiten ohjelmankulkua ja havainnollistaa periaatteellista ohjelmankulkua.

### 6.1 PIC16F690:n alustus

PIC16F690-piirin alustus alkaa liitteen 6 sivulta 1 kohdasta ”; ALUSTUS ALKAA” ja loppuu sivun 2 ”; ALUSTUS LOPPUU”. Aluksi määritellään prosessorityypin kirjasto ja esialustukset prosessorille. Tämän jälkeen varataan tilaa muuttujille alkaen osoitteesta 0x20h (cblock 0x20 ja endc väliset). Vakiot LCD:n ohjaamiseen määritellään kohdassa ”; Vakiot”. Itse ohjelmakoodi alkaa käskyn org 0x000 jälkeen. Tässä vaiheessa alkaa mikroprosessorin porttien määrittely, mitä toimintoja mihinkin kuuluu. Portti A määritellään tuloksi ja Portti C lähdeksi. Tämän jälkeen määritellään Portin A tulot RA3 ja RA4 digitaalisiksi. Portti C muutetaan kokonaan digitaaliseksi lähdeksi. Portin määrittelyjen jälkeen määritellään A/D-muunnoksen ominaisuudet. Tehdään AN2-kanavasta A/D-kanava ja otetaan referenssijännite AN1-kanavasta. Tehdään bittimuunnoksesta vasemmalle keskitetty AFDM avulla ja asetetaan A/D-muunnoslohko aktiiviseksi. Näiden vaiheiden jälkeen nollataan ja alustetaan kaikki ohjelman tarvitsemat rekisterit alkuarvoihinsa.

### 6.2 LCD:n alustus

LCD:n alustus tapahtuu esimerkin 12 mukaan, ja liitteen 6 lähdekoodista se on sivulla 39 kohdassa LCD\_ALUSTUS\_4BIT. Tässä koodissa alustetaan LCD alkuutilaan 4 bittisellä dataväylällä.

### 6.3 A/D-muunnos

A/D-muunnos tapahtuu liitteen 6 lähdekoodin sivulla 2 kohdassa ”; Tehdään A/D muunnos” ja loppuu kohtaan ”; A/D muunnos valmis”. A/D muunnoksen toiminta tapahtuu täysin esimerkin 6 mukaan otsikon A/D-muunnos alla.

### 6.4 EEPROM-kirjoitus ja -luku

EEPROM:in kirjoitus ja luku tapahtuu liitteen 6 lähdekoodin sivun 4 kohdissa READ\_EEPROM\_W ja WRITE\_EEPROM\_W. READ\_EEPROM\_W aliohjelma lukee välitetystä W-rekisterin arvon osoittamasta paikasta EEPROM:in sisällön ja palauttaa sen takaisin W-rekisteriin. WRITE\_EEPROM\_W kirjoittaa taas välitettyyn W-rekisterin osoittamaan paikkaan EEPROM:issa A/D-muunnoksen tämän hetken 8-ylintä bittiä. Tarkemmin EEPROM:in käsittelyt ovat esiteltyinä otsikon 3.2 EEPROM:in luku ja kirjoitus alla esimerkeissä 7 ja 8.

### 6.5 NTC-binaariarvo ASCII-luvuksi

Liitteen 6 lähdekoodista sivulta 5, aliohjelmalla TULOSTA\_TEMPERATURE tehdään muunnos A/D-muunnetulle bittijonolle. Tällä muunnoksella vertailemalla sitä ennalta laskettuihin NTC:n binaarilämpöihin, saadaan tulostettua lämpötila LCD:lle ASCII-muodossa. Valmiiksi lasketut arvot löytyvät liitteestä 5 taulukon kohdasta Arvo/BIN. Vertailu toimii XOR:in ja Z-lipun avulla. A/D-muunnoksesta saatua 8-bittistä lukemaa verrataan XOR:in avulla. Jos tulos on nolla, nousee Z-lippu ylös. Jos tulos on jotain muuta, Z-lippu pysyy nollassa. Esimerkki 15 havainnollistaa XOR:in vertailun käyttöä.

	Bittiluku	Desimaaliluku
Vertailtava lukema 0000 1001	9	
XOR	0000 1011	11
Tulos	0000 0010	2
ZERO LIPPU	0	
Vertailtava lukema 0000 1001	9	
XOR	0000 1001	9
Tulos	0000 0000	0
ZERO Lippu	1	

**Esimerkki 15** XOR:in ja Z-lipun käyttö

## 6.7 Menu-tila ja menun selaus

Menu-tila ja sinne meneminen tapahtuu RA3- ja RA4-portteihin kytketyillä painikkeilla. RA3 ja RA4 ovat alkualustuksissa määritelty digitaalisiksi ja sisään-tuloiksi. Pääsilmukassa liitteessä 6 sivulla 3 kohdassa CALL MENEEN-KO\_MENU kutsutaan aliohjelmaa, joka tutkii onko painiketta RA3:ssa painettu ja RA4:ssa painike on painamatta. Tämä sen takia, että estetään hyppiminen menusta pois ja takaisin. Aliohjelma on liitteen 6 sivulta 34, kohdasta MENEEN-KO\_MENU. Jos painiketta RA3 on painettu, menee ohjelma Menu-tilaan ja tulostaa LCD:n yläriville ”Menu Up/Down 1s”. Tämän jälkeen ohjelma pysyy menu-tilassa niin kauan kunnes RA3:ssa ja RA4:ssa olevia painikkeita painetaan yhtäaikaan. Menu-tilassa liikkuminen muistipaikkojen välissä tapahtuu RA3- ja RA4-painikkeilla. RA3-painikkeella mennään ylöspäin muistissa kun taas RA4-painikkeella mennään alaspäin. Tätä varten ohjelma tutkii, onko painettu ylös vai alas liitteen 6 sivulta 36 olevalla aliohjelmalla YLOS\_ALAS. Riippuen siitä, onko painettu ylös vai alas, osoitelaskuri (BINAARI\_LASKURI) pienenee tai suurenee. Ohjelma tulostaa LCD:n alariville osoitepaikannumeron desimaalilukuna ja sen hetkisen osoitteen osoittaman muistipaikan binääriluvun arvon ruudulle muunnettuna ASCII-muotoiseksi desimaaliksi LCD:lle. Jos osoitelaskuri ylittää arvon 255 ylöspainettaessa, siirtyy osoite laskuri lukuun 0. Ja jos alaspäin painettaessa osoitelaskuri menee 255, osoitelaskuri nollaantuu.

## 7 YHTEENVETO

Suurena puutteena lämpömittarissa oli pieni tarkkuus. Tätä tarkkuutta tullaan kuitenkin parantamaan tulevaisuudessa, koska ohjelmassa on olemassa mahdollisuus siihen. Työn prototyypin kuvat löytyvät liitteestä 8. Liitteen 8 sivulla 1 käy ilmi minkälainen työn prototyypistä tuli. Sivulla 2 on esitelty prototyypin ohjaukseen tarvittavat komponentit. Sivulla 3 mittarin päänäyttö aktiivisena ja sivulla 4 mittarin menuselausvalikko. Työn toteuttaminen onnistui täysin tämän hetkisten suunnitelmien mukaan. Laitteen testaus vaiheessa käytiin läpi yleisimmät lämpötilat vertailemalla elohopeamittarin näyttämään arvoon. Tarkkuudessa laite kärsii elohopeamittarille, mutta päivitysnopeudessa laite pärjäsikin hyvin. Tulevaisuudessa kehittäminen jatkuu parempien tarkkuuksien toivossa ja lisäantureiden ohjelmoinnilla. Valoisuus anturi on seuraava kehitys kohde.

## LÄHTEET

1. Wikipedia. [www-sivu]. [viitattu 4.4.2006]  
[http://en.wikipedia.org/wiki/PIC\\_microcontroller](http://en.wikipedia.org/wiki/PIC_microcontroller)
2. Microchip. [www-sivu]. [viitattu 4.4.2006] [www.microchip.com](http://www.microchip.com)
3. Wikipedia. [www-sivu]. [viitattu 4.4.2006]  
[http://en.wikipedia.org/wiki/Harvard\\_architecture](http://en.wikipedia.org/wiki/Harvard_architecture)
4. PIC16F690 datakirja. [Sähköinen dokumentti]. [Viitattu 4.4.2006]  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41262A.pdf> s. 4  
kuva 1.
5. PIC16F690 datakirja. [Sähköinen dokumentti]. [Viitattu 4.4.2006]  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41262A.pdf> s. 242  
– 245
6. PIC16F690 datakirja. [Sähköinen dokumentti]. [Viitattu 4.4.2006]  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41262A.pdf> s. 241.
7. Microchip. [www-sivu]. [viitattu 4.4.2006]  
[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE  
&nodeId=1406&dDocName=en023805](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023805).
8. PIC16F690 datakirja. [Sähköinen dokumentti]. [Viitattu 4.4.2006]  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41262A.pdf> s. 21.
9. PIC16F690 datakirja. [Sähköinen dokumentti]. [Viitattu 4.4.2006]  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41262A.pdf> s. 101.
10. PIC16F690 datakirja. [Sähköinen dokumentti]. [Viitattu 4.4.2006]  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41262A.pdf> s. 176.
11. PIC16F690 datakirja. [Sähköinen dokumentti]. [Viitattu 4.4.2006]  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41262A.pdf> s. 110.
12. Markh.de [www-sivu]. [viitattu 4.4.2006]  
<http://markh.de/hd44780-charset.png>.
13. ifent.org [www-sivu]. [viitattu 4.4.2006]  
<http://www.ifent.org/lecciones/ntc/NTC01.jpg>.
14. Samsung KS0066U datakirja. [Sähköinen dokumentti]. [Viitattu 4.4.2006]  
[http://www.microtipsusa.com/driver\\_pdfs/samsung/KS0066u.PDF](http://www.microtipsusa.com/driver_pdfs/samsung/KS0066u.PDF)

## LIITTEET

- Liite 1. PIC16F690-Käskykanta, pituus 19 sivua
- Liite 2. Samsung KS0066U-datakirja /14/, pituus 14 sivua
- Liite 3. Kytöntäkaavio, pituus 1 sivu
- Liite 4. Valmistajan antamat arvot NTC-vastukselle, pituus 1 sivu
- Liite 5. NTC-vastukselle lasketut arvot, pituus 1 sivu
- Liite 6. Lähdekoodi prototyypilaitteelle, pituus 41 sivua
- Liite 7. Ohjelman-vuokaavio, pituus 1 sivu
- Liite 8. Laitteen prototyypikuvat, pituus 4 sivua

## SISÄLLYSLUETTELO

SISÄLLYSLUETTELO .....	1
ADDWF f, d .....	2
ANDWF f, d .....	3
CLRF f.....	3
CLRW .....	3
COMF f, d.....	4
DECF f, d.....	5
DECFSZ f, d.....	5
INCF f, d.....	6
INCFSZ f, d.....	6
IORFW f, d.....	7
MOVF f, d.....	8
MOVWF f.....	8
NOP .....	8
RLF f, d.....	9
RRF f, d.....	10
SUBWF f, d.....	11
SWAPF f, d.....	12
XORWF f, d.....	13
BCF f, b.....	13
BSF f, b.....	14
BTFSC f, b.....	14
BTFSS f, b.....	14
ADDLW k.....	15
ANDLW k.....	15
CALL k.....	16
CLRWDT .....	16
GOTO k.....	16
IORLW k.....	17
MOVLW k.....	17
RETFIE.....	17
RETLW k.....	18
RETURN .....	18
SLEEP.....	18
SUBLW k.....	19
XORLW k.....	19



ADDWF f, d

ADDWF-käskyllä lasketaan W-rekisterin ja annetun rekisterin f sisältö toisiinsa. Jos tulos on nolla nousee Z-lippu ylös. Jos tulos on positiivinen, nousee C-lippu ylös, jos taas tulos on negatiivinen on C-lippu alhaalla. Jos tulos on suurempi kuin 0000 1111b niin DC-lippu on 1. Jos tulos taas on pienempikuin 0000 1111b on DC-lippu alhaalla.

**Esimerkki**

ADDWF AX, 1 ; AX-rekisteriin lisätään W-rekisterin sisältö ja tallennetaan tulos AX-rekisteriin.

Alkuarvot:

AX = 0000 0000b

W = 0000 0001b

Käskyn jälkeen:

W = 0000 0001b ; W-rekisterin sisältö pysyy muuttumattomana.

AX = 0000 0001b ; Yhteen laskettu tulos.

ADDWF AX, 0 ; AX-rekisteriin lisätään W-rekisterin sisältö ja tallennetaan se tulos W-rekisteriin.

ADDWF AX, 0 ; Sama kuin edellinen.

Alkuarvot:

AX = 0000 0000b

W = 0000 0001b

Käskyn jälkeen:

AX = 0000 0000b ; AX-rekisterin sisältö pysyy muuttumattomana.

W = 0000 0001b ; Yhteen laskettu tulos.

ANDWF f, d

ANDWF-käskyllä tehdään looginen operaation AND eli JA, rekisterin W ja annetun rekisterin f välille. Jos tulos on 0, nousee Z-lippu ylös.

**Esimerkki**

ANDWF AX, 1 ; AX-rekisteriin sisältöön tehdään looginen AND vertailu W-rekisterin sisällön kanssa, tulos talletetaan W-rekisteriin.

Alkuarvot:

AX = 0010 0101b

W = 0100 0101b

Käskyn jälkeen:

W = 0100 0101b ; W-rekisterin sisältö pysyy muuttumattomana.

AX = 0000 0101b ; AND vertailun tulos.

ANDWF AX ; AX-rekisteriin sisältöön tehdään looginen AND vertailu W-rekisterin sisällön kanssa, tulos talletetaan W-rekisteriin.

ANDWF AX,0 ; Sama kuin edellinen.

Alkuarvot:

AX = 0010 0101b

W = 0100 0101b

Käskyn jälkeen:

AX = 0010 0101b ; AX-rekisterin sisältö pysyy muuttumattomana.

W = 0000 0101b ; Yhteen laskettu tulos.

CLRf f

CLRf-käskyllä nollataan annettu rekisteri f arvoon 00h, samalla Z-lippu nousee pystyyn.

**Esimerkki**

CLRf AX ; AX-rekisterin sisältö nollataan arvolla 00h.

CLRw

CLRw-käskyllä nollataan W-rekisterin sisältö arvoon 00h, samalla Z-lippu nousee pystyyn.

**Esimerkki**

CLRw : W-rekisterin sisältöön talletetaan 00h.

COMF f, d

COMF-käskyllä suoritetaan annetulle rekisterille f looginen NOT operaatio, eli luvun komplementti. Jos tulos on 0 nousee Z-lippu ylös.

**Esimerkki**

COMF AX,1 ; AX-rekisteriin tehdään looginen NOT operaatio ja tulos tallennetaan tulos AX-rekisteriin.

Alkuarvot:

AX = 0010 0101b

W = 0000 0000b

Käskyn jälkeen:

W = 0000 0000b ; W-rekisterin sisältä pysyy muuttumattomana.

AX = 1101 1010b ; NOT operaation tulos.

COMF AX ; AX-rekisteriin tehdään looginen NOT operaatio ja talletetaan tulos W-rekisteriin.

COMF AX,0 ; Sama kuin edellinen.

Alkuarvot:

AX = 0010 0101b

W = 0000 0000b

Käskyn jälkeen:

AX = 0010 0101b ; AX-rekisterin sisältö pysyy muuttumattomana.

W = 1101 1010b ; NOT operaation tulos.

DECF f, d

DECF-käskyllä vähennetään annetusta rekisteristä f yksi. Jos tulos menee alle 0000 0000b, niin hyppää tulos 1111 1111b eli pyörähtää ympäri. Samalla Z-lippu nousee ylös, kun tulos on 0000 0000b.

**Esimerkki**

DECF AX,1 ; AX-rekisterin arvosta poistetaan yksi ja tulos tallennetaan AX-rekisteriin.

Alkuarvot:  
AX = 0000 1111b  
W = 0000 0000b

Käskyn jälkeen:  
W = 0000 0000b ; W-rekisterin sisältö pysyy muuttumattomana.  
AX = 0000 1110b ; AX-rekisteri vähennettynä yhdellä.

DECF AX ; AX-rekisteriin arvosta vähennetään yksi ja talletetaan tulos W-rekisteriin.  
DECF AX,0 ; Sama kuin edellinen.

Alkuarvot:  
AX = 0010 1111b  
W = 0000 0000b

Käskyn jälkeen:  
AX = 0000 1111b ; AX-rekisterin sisältö pysyy muuttumattomana.  
W = 0000 1110b ; AX rekisterin sisältö vähennettynä yhdellä.

DECFSZ f, d

DECFSZ-käsky toimii muuten samalla lailla kuin DECF-käsky. Mutta jos tulos on 0, ohitetaan seuraava käsky.

**Esimerkki**

DECFSZ AX ,1 ; Vähennetään yksi rekisteristä AX ja talletetaan tulos rekisteriin AX, jos tulos on olla ohitetaan seuraava käsky.

GOTO TULOS\_MUU ; Hypätään tänne jos tulos on muuta kuin nolla.  
GOTO TULOS\_NOLLA ; Hypätään tänne jos tulos on nolla, eli ohitetaan toinen rivi.

DECFSZ AX ; Ainoana erona tälle käskylle tulee, että tulos talletetaan W-rekisteriin.

DECFSZ AX ,0 ; Sama kuin edellinen.

INCF f, d

INCF-käsky on DECF-käskyn vastakohta. Käskyllä lisätään annettuun rekisteriin f yksi. Jos tulos menee yli 1111 1111b pyörähtää rekisteri ympäri eli arvoon 0000 0000b. Samalla Z-lippu(zero) eli nolla bitti nousee ylös, kun tulos on 0000 0000b.

**Esimerkki**

INCF AX,1 ; AX-rekisterin arvoon lisätään yksi ja tulos tallennetaan AX-rekisteriin.

Alkuarvot:

AX = 0000 1111b

W = 0000 0000b

Käskyn jälkeen:

W = 0000 0000b ; W-rekisterin sisältö pysyy muuttumattomana.

AX = 0001 0000b ; AX-rekisteri lisättynä yhdellä.

INCF AX ; AX-rekisteriin lisätään yksi ja talletetaan tulos W-rekisteriin.

INCF AX,0 ; Sama kuin edellinen.

Alkuarvot:

AX = 0000 1111b

W = 0000 0000b

Käskyn jälkeen:

AX = 0000 1111b ; AX-rekisterin sisältö pysyy muuttumattomana.

W = 0001 0000b ; AX-rekisterin sisältö lisättynä yhdellä W-rekisterissä.

INCFSZ f, d

INCFSZ-käsky toimii muuten samalla lailla kuin INCFSZ-käsky. Mutta jos tulos on 0, ohitetaan seuraava käsky.

**Esimerkki**

INCFSZ AX,1 ; Lisätään AX-rekisteriin yksi ja talletetaan tulos rekisteriin AX, jos tulos on olli ohitetaan seuraava käsky.

GOTO TULOS\_MUU ; Hypätään tänne jos tulos on muuta kuin nolla.

GOTO TULOS\_NOLLA ; Hypätään tänne jos tulos on nolla, eli ohitetaan toinen rivi.

INCFSZ AX ; Ainoana erona tälle käskylle tulee, että tulos talletetaan W-rekisteriin.

INCFSZ AX,0 ; Sama kuin edellinen.

IORFW f, d

IORFW-käskyllä tehdään annettuun rekisteriin f looginen operaatio OR eli TAI. Jos tulos on 0, nousee Z-lippu ylös.

**Esimerkki**

IORFW AX,1 ; AX-rekisterin arvolle suoritetaan OR operaatio W-rekisterin kanssa ja tulos tallennetaan AX-rekisteriin.

Alkuarvot:

AX = 0000 1111b

W = 0110 0000b

Käskyn jälkeen:

W = 0110 0000b ; W-rekisterin sisältö pysyy muuttumattomana.

AX = 0110 1111b ; AX-rekisterin sisältö OR operaation jälkeen.

IORFW AX ; AX-rekisterin arvolle suoritetaan OR-operaatio W-rekisterin kanssa ja tulos talletetaan W-rekisteriin.

IORFW AX,0 ; Sama kuin edellinen.

Alkuarvot:

AX = 0000 1111b

W = 0110 0000b

Käskyn jälkeen:

AX = 0000 1111b ; AX-rekisterin sisältö pysyy muuttumattomana.

W = 0110 1111b ; W-rekisterin sisältö lisättynä yhdellä.

## MOVF f, d

MOVF-käskyllä siirretään annettu rekisterin f sisältö W-rekisteriin tai kopioidaan rekisterin sisältö itseensä. Jos kopioidaan itseensä tarkistetaan, onko tulos nolla, jos on Z-lippu nousee ylös.

**Esimerkki**

MOVF AX,1 ; AX-rekisterin sisältö kopioidaan AX-rekisterin sisällöksi.

Alkuarvot:  
AX = 0000 1111b  
W = 0000 0000b

Käskyn jälkeen:  
W = 0000 0000b ; W-rekisterin sisältönä pysyy muuttumattomana.  
AX = 0000 1111b ; AX-rekisteri on kopioituna.

MOVF AX ; AX-rekisterin sisältö siirretään W-rekisterin sisällöksi.  
MOVF AX,0 ; Sama kuin edellinen.

Alkuarvot:  
AX = 0000 1111b  
W = 0000 0000b

Käskyn jälkeen:  
AX = 0000 1111b ; AX-rekisterin sisältö pysyy muuttumattomana.  
W = 0000 1111b ; AX-rekisterin sisältö W-rekisterissä.

## MOVWF f

MOVWF-käskyllä siirretään W-rekisterin sisältö annetun rekisterin f sisällöksi.

**Esimerkki**

MOVWF AX ; W-rekisterin sisältö siirretään AX-rekisterin sisällöksi.

Alkuarvot:  
AX = 0000 0000b  
W = 0000 1111b

Käskyn jälkeen:  
W = 0000 1111b ; W-rekisterin sisältönä pysyy muuttumattomana.  
AX = 0000 1111b ; W-rekisterin sisältö siirrettynä AX-rekisteriin.

## NOP

NOP-käsky ei tee mitään. Se vain odottaa yhden käskyjakson ajan. Esimerkiksi, 4MHz taajuudella ajettuna, NOP-käskyn suorittaminen vaatii 1 mikrosekunnin ajan.

RLF f, d

RLF-käskyllä siirretään annettua rekisteriä f yhden bitin oikealle. Jos 7-bitti on yksi, kun käsky suoritetaan tapahtuu ylivuoto. Tällöin C-lippu nousee ylös. Oikealle aina lisätään nollia sitä mukaa mitä siirretään lukua.

**Esimerkki**

RLF AX,1 ; AX-rekisterin sisältöä siirretään bitti vasemmalle ja talletetaan tulos AX rekisteriin.

Alkuarvot:  
AX = 0000 1100b  
W = 0000 0000b

Käskyn jälkeen:  
W = 0000 0000b ; W-rekisterin sisältönä pysyy muuttumattomana.  
AX = 0001 1000b ; AX-rekisteri sisältö siirretty yhden bitin verran vasemmalle.

RLF AX ; AX-rekisterin sisältöä siirretään bitti vasemmalle ja talletetaan tulos W-rekisteriin.

RLF AX,0 ; Sama kuin edellinen.

Alkuarvot:  
AX = 0000 1100b  
W = 0000 0000b

Käskyn jälkeen:  
AX = 0000 1100b ; AX-rekisterin sisältö pysyy muuttumattomana.  
W = 0001 1000b ; AX-rekisterin sisältö siirretty yhden bitin vasemmalle ja tulos W-rekisterissä.



RRF f, d

RRF-käsky on muuten identtinen RLF-käskyn kanssa, mutta käsky siirtää vaan rekisteriä f yhden bitin verran oikealle. Lisäksi jos bitti-0 on 1, kun käskyä suoritetaan. Tällöin C-lippu nousee ylös ylivuodon seurauksena.

**Esimerkki**

RRF AX,1 ; AX-rekisterin sisältöä siirretään bitti oikealle ja talletetaan tulos AX-rekisteriin.

Alkuarvot:  
AX = 0000 1100b  
W = 0000 0000b

Käskyn jälkeen:  
W = 0000 0000b ; W-rekisterin sisältönä pysyy muuttumattomana.  
AX = 0000 0110b ; AX-rekisteri sisältö siirretty yhden bitin verran oikealle.

RRF AX ; AX-rekisterin sisältöä siirretään bitti oikealle ja talletetaan tulos W-rekisteriin.

RRF AX,0 ; Sama kuin edellinen.

Alkuarvot:  
AX = 0000 1100b  
W = 0000 0000b

Käskyn jälkeen:  
AX = 0000 1100b ; AX-rekisterin sisältö pysyy muuttumattomana.  
W = 0000 0110b ; AX-rekisterin sisältö siirretty yhden bitin oikealle ja tulos W-rekisterissä.

SUBWF f, d

SUBWF-käskyllä vähennetään W-rekisterin sisältö annetusta rekisteristä f. Jos tulos on nolla, nousee Z-lippu. Jos tulos on positiivinen, nousee C-lippu ylös. Jos tulos on taas negatiivinen on C-lippu alhaalla. Jos tulos on suurempi kuin 0000 1111b, niin DC-lippu on 1. Jos tulos taas on pienempi kuin 0000 1111b, on DC-lippu alhaalla.

**Esimerkki**

SUBWF AX,1 ; AX-rekisterin sisällöstä poistetaan W-rekisterin sisältö. Talletetaan tulos AX-rekisteriin.

Alkuarvot:

AX = 0000 1111b

W = 0000 0001b

Käskyn jälkeen:

W = 0000 1111b ; W-rekisterin sisältönä pysyy muuttumattomana.

AX = 0000 1110b ; AX-rekisteristä poistettuna W-rekisterin sisältö.

SUBWF AX ; AX-rekisterin sisällöstä poistetaan W-rekisterin sisältö. Talletetaan tulos W-rekisteriin.

SUBWF AX,0 ; Sama kuin edellinen.

Alkuarvot:

AX = 0000 1111b

W = 0000 0001b

Käskyn jälkeen:

AX = 0000 1111b ; AX-rekisterin sisältö pysyy muuttumattomana.

W = 0000 1110b ; AX-rekisterin sisällöstä poistettuna W-rekisterin sisältö.

SWAPF f, d

SWAPF-käskyllä vaihdetaan annetun rekisterin f neljän ensimmäisen ja viimeisten bittien paikkaa.

**Esimerkki**

SWAPF AX,1 ; AX-rekisterin bitit vaihdetaan. Talletetaan tulos AX-rekisteriin.

Alkuarvot:

AX = 0000 1111b

W = 0000 0000b

Käskyn jälkeen:

W = 0000 0000b ; W-rekisterin sisältönä pysyy muuttumattomana.

AX = 1111 0000b ; AX-rekisteristä neljät ensimmäiset ja alimmat bitit vaihtoivat paikkaansa.

SWAPF AX ; AX-rekisterin bitit vaihdetaan. Talletetaan tulos W-rekisteriin.

SWAPF AX,0 ; Sama kuin edellinen.

Alkuarvot:

AX = 0000 1111b

W = 0000 0000b

Käskyn jälkeen:

AX = 0000 1111b ; AX-rekisterin sisältö pysyy muuttumattomana.

W = 1111 0000b ; AX-rekisteristä neljät ensimmäiset ja alimmat bitit vaihtoivat paikkaansa.

Tulos W-rekisterissä.

XORWF f, d

XORWF-käskyllä tehdään looginen operaatio XOR tuodun rekisterin f ja W-rekisterin välille. Jos tulos on nolla, Z-lippu nousee ylös.

**Esimerkki**

XORWF AX,1 ; AX-rekisterin ja W-rekisterin välille tehdään XOR-operaatio. Tulos talletetaan AX-rekisteriin.

Alkuarvot:  
AX = 0000 1111b  
W = 0000 0000b

Käskyn jälkeen:  
W = 0000 0000b ; W-rekisterin sisältönä pysyy muuttumattomana.  
AX = 0000 1111b ; XOR-operaation tulos AX-rekisterissä.

XORWF AX ; AX-rekisterin ja W-rekisterin välille tehdään XOR-operaatio. Tulos talletetaan W-rekisteriin.

XORWF AX, 0 ; Sama kuin edellinen.

Alkuarvot:  
AX = 0000 1111b  
W = 0000 0000b

Käskyn jälkeen:  
AX = 0000 1111b ; AX-rekisterin sisältö pysyy muuttumattomana.  
W = 0000 1111b ; XOR-operaation tulos W-rekisterissä.

BCF f, b

BCF-käskyllä nollataan rekisterin f:n tietty bitti-b.

**Esimerkki**

BCF AX,2 ; AX-rekisterin 2-bitti nollataan.

Alkuarvot:  
AX = 1111 1111 b

Käskyn jälkeen:  
AX = 1111 1011 ; AX-rekisterin 2-bitti on nolla.

BSF f, b

BSF-käskyllä asetetaan rekisterin f tietty bitti-b.

**Esimerkki**

BSF AX,2	; AX-rekisterin 2-bitti asetetaan.
Alkuarvot: AX = 0000 0000 b	
Käskyn jälkeen: AX = 0000 0100	; AX-rekisterin 2-bitti on yksi.

BTFSC f, b

BTFSC-käskyllä tutkitaan rekisterin f bittiä b, että onko se nolla. Jos tulos on nolla, ohitetaan seuraava käskyriivi.

**Esimerkki**

BTFSC AX,2	; Jos AX-rekisterin bitti 2 on nolla ohitetaan seuraava rivi, muuten jatketaan normaalisti.
GOTO TULOS_MUU	; Hypätään tänne jos tulos on muuta kuin nolla.
GOTO TULOS_NOLLA	; Hypätään tänne jos tulos on nolla, eli ohitetaan toinen rivi.

BTFSS f, b

BTFSS-käskyllä tutkitaan rekisterin f bittiä b, että onko se yksi. Jos tulos on yksi ohitetaan seuraava käskyriivi.

**Esimerkki**

BTFSS AX,2	; Jos AX-rekisterin bitti 2 on yksi ohitetaan seuraava rivi, muuten jatketaan normaalisti.
GOTO TULOS_MUU	; Hypätään tänne jos tulos on muuta kuin yksi.
GOTO TULOS_NOLLA	; Hypätään tänne jos tulos on yksi, eli ohitetaan toinen rivi.

## ADDLW k

ADDLW-käskyllä lisätään W-rekisteriin luku k. Tulos talletetaan W-rekisteriin. Jos tulos on nolla, nousee Z-lippu ylös. Jos tulos on positiivinen, nousee C-lippu ylös, jos tulos on negatiivinen on C-lippu alhaalla. Jos tulos on suurempi kuin 0000 1111b, niin DC-lippu on 1. Jos tulos taas on pienempikuin 0000 1111b, on DC-lippu alhaalla.

**Esimerkki**

ADDLW b'00000010' ; Lisätään W-rekisteriin 0000 0010b ja talletetaan tulos W-rekisteriin.

Alkuarvot:

k = 0000 0010b

W = 0000 0001b

Käskyn jälkeen:

W = 0000 0011b ; W-rekisterin sisältöön lisätty 2, joten tulos on 3.

## ANDLW k

ANDLW-käskyllä tehdään looginen operaation AND, W-rekisterin ja luvun k välillä. Tulos talletetaan W-rekisteriin. Jos tulos on nolla, nousee Z-lippu ylös.

**Esimerkki**

ANDLW b'00001111' ; Tehdään AND-operaatio W-rekisterin sisällön ja 00001111b välille. Talletetaan tulos W-rekisteriin.

Alkuarvot:

k = 0000 1111b

W = 0000 1100b

Käskyn jälkeen:

W = 0000 1100b ; W-rekisterin sisältö AND-operaation jälkeen.

## CALL k

CALL-käskyllä kutsutaan aliohjelmaa osoitteesta k tai aliohjelman nimellä.

**Esimerkki**

MAINLOOP	; MAINLOOP ohjelman aloitus.
CALL ALIOHJELMA	; Kutsutaan aliohjelmaa ALIOHJELMA.
GOTO MAINLOOP	; Mene MAINLOOP kohtaan.
END	; Ohjelman lopetus.
ALIOHJELMA	; Aliohjelman kutsu osoite/nimi. ; Aliohjelma rutiini..
RETURN	; Paluu aliohjelma kutsupaikan seuraavalle riville.

## CLRWDT

CLRWDT-käskyllä nollataan WDT-ajastin. Ajastimen arvoksi asetetaan 00h. Esijakajan sisälöksi 00h. TO- ja PD-bitteihin asetetaan ykkös tilaan.

**Esimerkki**

CLRWDT	; Nollataan WDT-ajastin.
--------	--------------------------

## GOTO k

GOTO-käskyllä siirrytään tiettyyn osoitteeseen tai nimeen k.

**Esimerkki**

MAINLOOP	; Ohjelman aloitus ; Ohjelman koodi
GOTO MAINLOOP	; Palaa kohtaan MAINLOOP

## IORLW k

IORLW-käskyllä tehdään looginen operaatio OR, W-rekisterin ja annetun arvon k välille. Tulos talletetaan W-rekisteriin. Jos tulos on nolla, on Z-lippu 1, muuten Z-lippu on 0.

**Esimerkki**

IORLW b'00001111' ; Tehdään OR-operaatio W-rekisterin sisällön ja 00001111b välille. Talletetaan tulos W-rekisteriin.

Alkuarvot:

k = 0000 1111b

W = 0010 0000b

Käskyn jälkeen:

W = 0010 1111b ; W-rekisterin sisältö OR-operaation jälkeen.

## MOVLW k

MOVLW-käskyllä siirretään lukema k W-rekisterin sisältöön.

**Esimerkki**

MOVLW b'00001111' ; Siirretään luku 00001111b W-rekisteriin.

Alkuarvot:

k = 0000 1111b

W = 0000 0000b

Käskyn jälkeen:

W = 0000 1111b ; Luku k siirretty W-rekisteriin.

## RETFIE

RETFIE-käskyllä palataan ulkoisesta keskeytyksestä.

**Esimerkki**

ULKOINEN KESKEYTYS	; Ulkoinen keskeytys alkaa.
	; Ulkoisen keskeytyksen ohjelma.
RETFIE	; Paluu ulkoisesta keskeytyksestä.



## RETLW k

RETLW-käskyllä palataan keskeytyksestä keskeytys hetkellä talletettuun kohtaan ja talletetaan lukema k rekisteriin W.

**Esimerkki**

KESKEYTYS	; Keskeytysohjelma alkaa
	; Keskeytysohjelma
RETLW b'00000001'	; Paluu keskeytyksestä ja samalla lukema 0000 0001b W-
rekisteriin.	

## RETURN

RETURN-käskyllä palataan aliohjelmasta takaisin kutsupaikkaan.

**Esimerkki**

MAINLOOP	; Pääohjelma alkaa
CALL ALIOHJELMA	; Kutsutaan aliohjelmaa
GOTO MAINLOOP	; Mene takaisin kohtaan MAINLOOP
ALIOHJELMA	; Keskeytys ohjelma alkaa
	; Aliohjelman ohjelmakoodi
RETURN	; Palataan kutsupaikan seuraavalle riville

## SLEEP

SLEEP-käskyllä siirretään mikroprosessori lepotilaan. WDT-ajastimeen asetetaan 00h ja esijakajaan 00h. Lippu TO asetetaan 1 tilaan ja lippu PD asetetaan 0 tilaan. Lopuksi mikroprosessorin kello-oskillaattori pysäytetään.

**Esimerkki**

SLEEP	;Ohjelman pysäytys ja oskillaattori pysäytetään.
-------	--

## SUBLW k

SUBLW-käskyllä vähennetään luku k rekisteristä W ja talletetaan tulos W-rekisteriin. Jos tulos on nolla, nousee Z-lippu ylös. Jos tulos on positiivinen, nousee C-lippu ylös, jos taas negatiivinen on C-lippu alhaalla. Jos tulos on suurempi kuin 0000 1111b, niin DC-lippu on 1. Jos tulos taas on pienempikuin 0000 1111b, on DC-lippu alhaalla.

**Esimerkki**

SUBLW b'00000001' ; Poistetaan luku 00000001b W-rekisteristä. Talletetaan tulos W-rekisteriin.

Alkuarvot:

k = 0000 0001b

W = 0000 1111b

Käskyn jälkeen:

W = 0000 1110b ; W-rekisteri Vähennyslaskun jälkeen.

## XORLW k

XORLW-käskyllä tehdään looginen operaatio XOR, rekisterin W ja luvun k välillä. Jos tulos on nolla, nousee Z-lippu ylös, jos jokin muu pysyy Z-lippu alhaalla.

**Esimerkki**

XORLW b'10101010' ; Tehdään XOR-operaatio W-rekisterin ja luvun 1010 1010b välillä. Tulos talletetaan W-rekisteriin.

Alkuarvot:

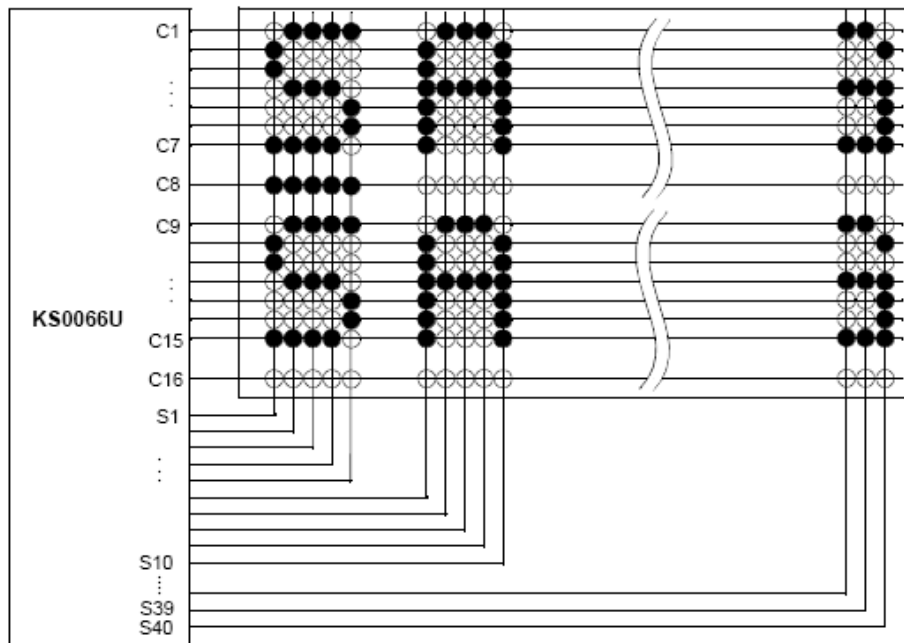
k = 1010 1011b

W = 0100 0101b

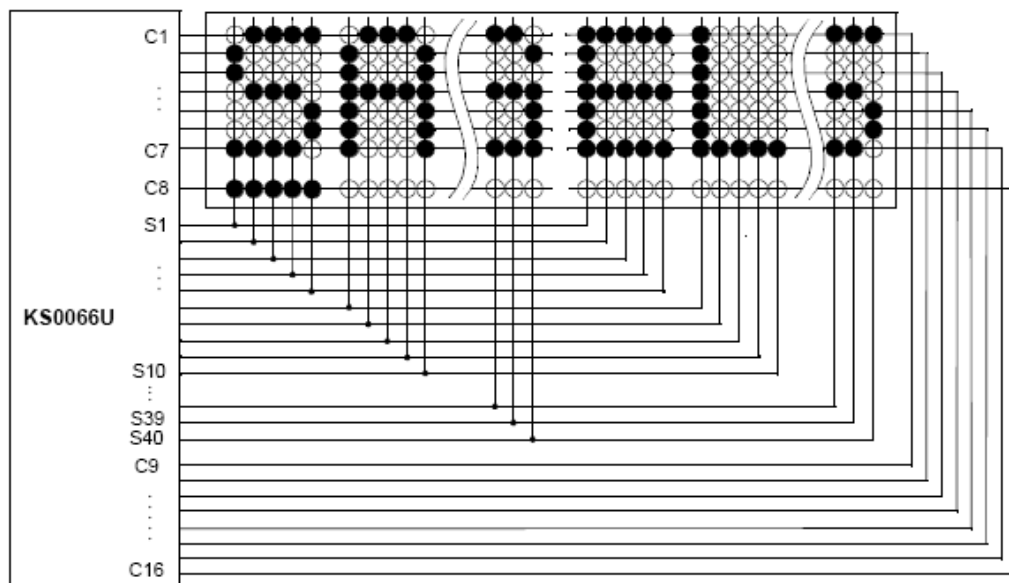
Käskyn jälkeen:

W = 1110 1110b ; W-rekisteri sisältö XOR-operaation jälkeen.

3) LCD Panel: 8 characters  $\times$  2-line format (5 $\times$ 7 dots + line, 1/5 bias, 1/16 duty)



4) LCD Panel: 16 characters  $\times$  1-line format (5 $\times$ 7 dots + 1 cursor line, 1/5 bias, 1/16 duty)





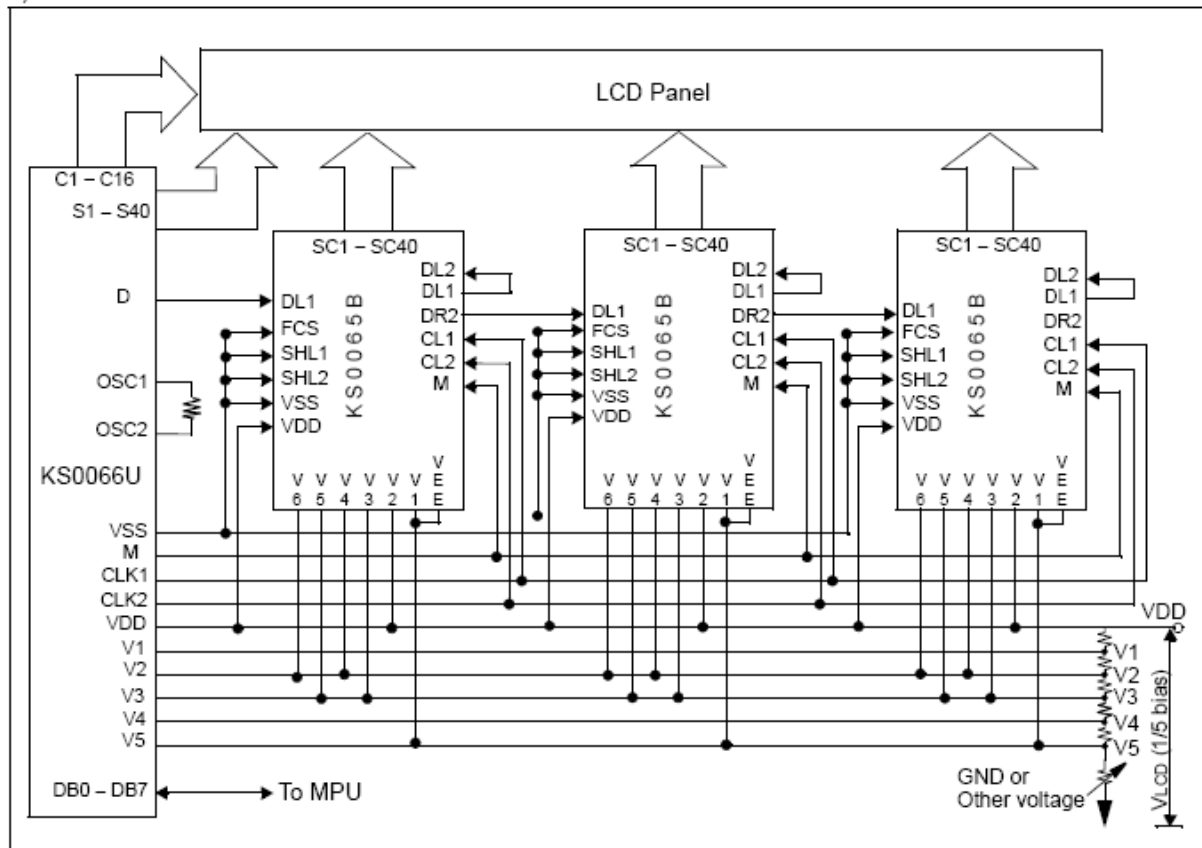
**KS0066U****16COM / 40SEG DRIVER & CONTROLLER FOR DOT MATRIX LCD****PIN DESCRIPTION****Table 2. Pin Description**

Pin	Pin No.	I/O	Name	Description	Interface
VDD	33	-	Supply Voltage	Supply Voltage for logical circuit (+3V ± 10%, +5V ± 10%)	Power Supply
GND	23			Ground (0V)	
V1-V5	26-30			Bias voltage level for LCD driving	
S1-S40	1-22, 63-80	O	Segment output	Segment signal output for LCD drive	LCD
C1-C16	47-62	O	Common output	Common signal output for LCD drive	LCD
OSC1	24	I	Oscillator	Oscillator. When using internal oscillator, connect external Rf resistor. If external clock is used, connect it to OSC1.	External resistor/oscillator (OSC1)
OSC2	25	O	Oscillator		
CLK1	31	O	Extension driver Latch clock	Extension driver latch clock	Extension driver
CLK2	32	O	Extension driver Shift clock	Extension driver shift clock	
M	34	O	Alternated signal for LCD driver output	Outputs the alternating signal to convert LCD driver waveform to AC.	Extension driver
D	35	O	Display data interface	Outputs extension driver data (the 41st dot's data)	Extension driver
RS	36	I	Register select	Used as register selection input. When RS = "High", Data register is selected. When RS = "Low", Instruction register is selected.	MPU
R/W	37	I	Read/Write	Used as read/write selection input. When RW = "High", read operation. When RW = "Low", write operation.	MPU
E	38	I	Read/Write enable	Used as read/write enable signal.	MPU
DB0-DB3	39-42	I/O	Data bus 0-7	In 8-bit bus mode, used as low order bidirectional data bus. In 4-bit bus mode, open these pins.	MPU
DB4-DB7	43-46			In 8-bit bus mode, used as high order bidirectional data bus. In 4-bit bus mode, used as both high and low order. DB7 used for Busy Flag output.	MPU

**KS0066U**

**16COM / 40SEG DRIVER & CONTROLLER FOR DOT MATRIX LCD**

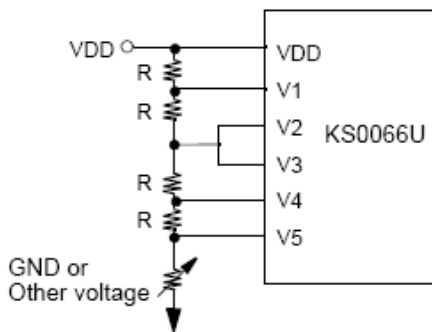
6) APPLICATION CIRCUIT



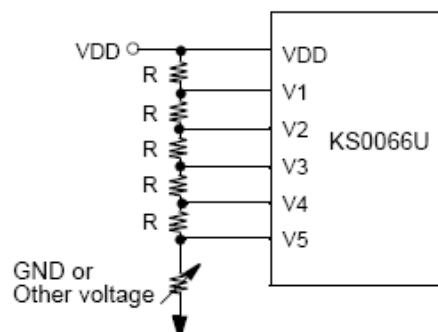
NOTE: When KS0065B is externally connected to the KS0066U, you can increase the number of display digits up to 80 characters.

**BIAS VOLTAGE DIVIDE CIRCUIT**

1) 1/4 bias, 1/8 or 1/11 duty



2) 1/5 bias, 1/16 duty

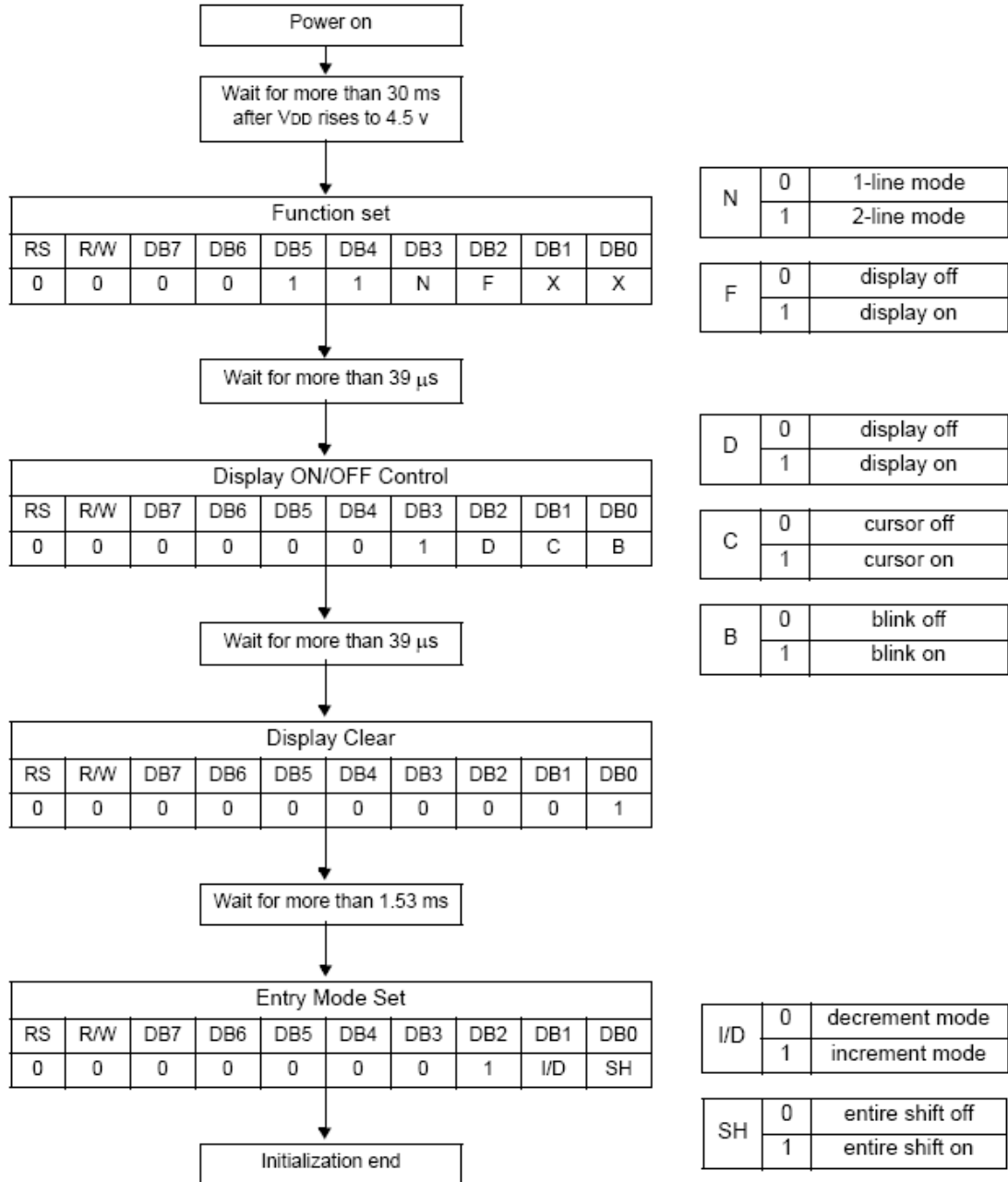


**KS0066U**

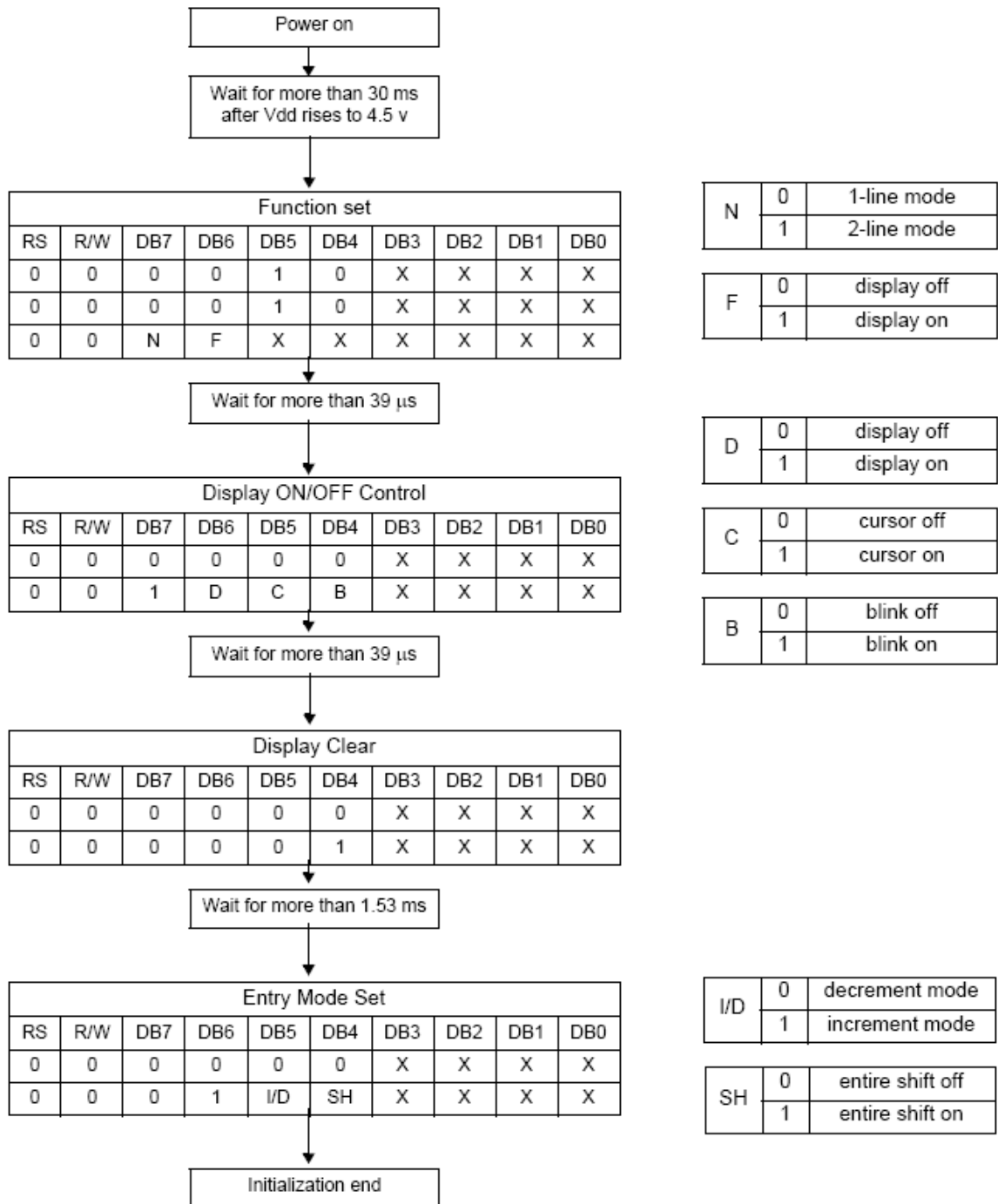
**16COM / 40SEG DRIVER & CONTROLLER FOR DOT MATRIX LCD**

**INITIALIZING BY INSTRUCTION**

1) 8-bit interface mode (Condition: fosc = 270KHZ)



2) 4-bit interface mode (Condition: fosc = 270KHZ)





## INSTRUCTION DESCRIPTION

## Outline

To overcome the speed difference between the internal clock of KS0066U and the MPU clock, KS0066U performs internal operations by storing control informations to IR or DR. The internal operation is determined according to the signal from MPU, composed of read/write and data bus (Refer to Table 7).

Instructions can be divided largely into four groups:

- 1) KS0066U function set instructions (set display methods, set data length, etc.)
- 2) address set instructions to internal RAM
- 3) data transfer instructions with internal RAM
- 4) others

The address of the internal RAM is automatically increased or decreased by 1.

Note: During internal operation, Busy Flag (DB7) is read 'High'.

Busy Flag check must be preceded by the next instruction.

When an MPU program with checking the Busy Flag (DB7) is made, it must be necessary  $1/2 f_{osc}$  for executing the next instruction by the falling edge of the 'E' signal after the Busy Flag (DB7) goes to 'Low'.

## Contents

## 1) Clear Display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Clear all the display data by writing "20H" (space code) to all DDRAM address, and set DDRAM address to "00H" into AC (address counter).

Return cursor to the original status, namely, bring the cursor to the left edge on the first line of the display. Make the entry mode increment (I/D = 'High').

## 2) Return Home

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	-

\* "-" : dont care

Return Home is cursor return home instruction.

Set DDRAM address to "00H" into the address counter.

Return cursor to its original site and return display to its original status, if shifted.

Contents of DDRAM does not change.

**KS0066U****16COM / 40SEG DRIVER & CONTROLLER FOR DOT MATRIX LCD**

## 3) Entry Mode Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	SH

Set the moving direction of cursor and display.

**I/D: Increment / decrement of DDRAM address (cursor or blink)**

When I/D = "High", cursor/blink moves to right and DDRAM address is increased by 1.

When I/D = "Low", cursor/blink moves to left and DDRAM address is decreased by 1.

\* CGRAM operates the same way as DDRAM, when reading from or writing to CGRAM.

**SH: Shift of entire display**

When DDRAM read (CGRAM read/write) operation or SH = "Low", shifting of entire display is not performed.

If SH = "High" and DDRAM write operation, shift of entire display is performed according to I/D value

(I/D = "High": shift left, I/D = "Low": shift right).

## 4) Display ON/OFF Control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Control display/cursor/blink ON/OFF 1 bit register.

**D: Display ON/OFF control bit**

When D = "High", entire display is turned on.

When D = "Low", display is turned off, but display data remains in DDRAM.

**C: Cursor ON/OFF control bit**

When C = "High", cursor is turned on.

When C = "Low", cursor is disappeared in current display, but I/D register preserves its data.

**B: Cursor Blink ON/OFF control bit**

When B = "High", cursor blink is on, which performs alternately between all the "High" data and display characters at the cursor position.

When B = "Low", blink is off.

**KS0066U****16COM / 40SEG DRIVER & CONTROLLER FOR DOT MATRIX LCD**

## 5) Cursor or Display Shift

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	-	-

Shifting of right/left cursor position or display without writing or reading of display data. This instruction is used to correct or search display data. (Refer to Table 6)  
 During 2-line mode display, cursor moves to the 2nd line after the 40th digit of the 1st line. Note that display shift is performed simultaneously in all the lines.  
 When displayed data is shifted repeatedly, each line is shifted individually.  
 When display shift is performed, the contents of the address counter are not changed.

**Table 6. Shift Patterns According to S/C and R/L Bits**

S/C	R/L	Operation
0	0	Shift cursor to the left, AC is decreased by 1
0	1	Shift cursor to the right, AC is increased by 1
1	0	Shift all the display to the left, cursor moves according to the display
1	1	Shift all the display to the right, cursor moves according to the display

## 6) Function Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	-	-

**DL: Interface data length control bit**

When DL = 'High', it means 8-bit bus mode with MPU.

When DL = 'Low', it means 4-bit bus mode with MPU. Hence, DL is a signal to select 8-bit or 4-bit bus mode.

When 4-bit bus mode, it needs to transfer 4-bit data twice.

**N: Display line number control bit**

When N = 'Low', 1-line display mode is set.

When N = 'High', 2-line display mode is set.

**F: Display font type control bit**

When F = 'Low', 5 × 8 dots format display mode is set.

When F = 'High', 5 × 11 dots format display mode.

**KS0066U****16COM / 40SEG DRIVER & CONTROLLER FOR DOT MATRIX LCD**

## 7) Set CGRAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Set CGRAM address to AC.  
This instruction makes CGRAM data available from MPU.

## 8) Set DDRAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Set DDRAM address to AC.  
This instruction makes DDRAM data available from MPU.  
When 1-line display mode (N = Low), DDRAM address is from '00H" to "4FH".  
In 2-line display mode (N = High), DDRAM address in the 1st line is from "00H" to "27H", and DDRAM address in the 2nd line is from "40H" to "67H".

## 9) Read Busy Flag &amp; Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This instruction shows whether KS0066U is in internal operation or not.  
If the resultant BF is "High", internal operation is in progress and should wait until BF is to be Low, which by then the next instruction can be performed. In this instruction you can also read the value of the address counter.

**KS0066U****16COM / 40SEG DRIVER & CONTROLLER FOR DOT MATRIX LCD**

10) Write data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write binary 8-bit data to DDRAM/CGRAM.

The selection of RAM from DDRAM, and CGRAM, is set by the previous address set instruction (DRAM address set, CGRAM address set).

RAM set instruction can also determine the AC direction to RAM.

After write operation, the address is automatically increased/decreased by 1, according to the entry mode.

11) Read data from RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

Read binary 8-bit data from DDRAM/CGRAM.

The selection of RAM is set by the previous address set instruction. If the address set instruction of RAM is not performed before this instruction, the data that has been read first is invalid, as the direction of AC is not Yet determined. If RAM data is read several times without RAM address instructions set before read operation, the correct RAM data can be obtained from the second. But the first data would be incorrect, as there is no time margin to transfer RAM data.

In case of DDRAM read operation, cursor shift instruction plays the same role as DDRAM address set instruction, it also transfers RAM data to output data register.

After read operation, address counter is automatically increased/decreased by 1 according to the entry mode.

After CGRAM read operation, display shift may not be executed correctly.

**NOTE:** In case of RAM write operation, AC is increased/decreased by 1 as in read operation.

At this time, AC indicates the next address position, but only the previous data can be read by the read instruction.

KS0066U

16COM / 40SEG DRIVER &amp; CONTROLLER FOR DOT MATRIX LCD

Table 7. Instruction Table

Instruction	Instruction Code										Description	Execution time (fosc=270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM and set DDRAM address to "00H" from AC	1.53 ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39 μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39 μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39 μs
Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line) and, display font type (F:5×11dots/5×8 dots)	39 μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 μs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 μs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 μs

\* "-": dont care

NOTE: When an MPU program with checking the Busy Flag(DB7) is made, it must be necessary 1/2Fosc is necessary for executing the next instruction by the falling edge of the 'E' signal after the Busy Flag (DB7) goes to "Low".

2) 2-line display

In case of 2-line display, the address range of DDRAM is 00H–27H and 40H–67H.

An extension driver will be used. Fig-3 shows the example with 40 segment extension driver added.

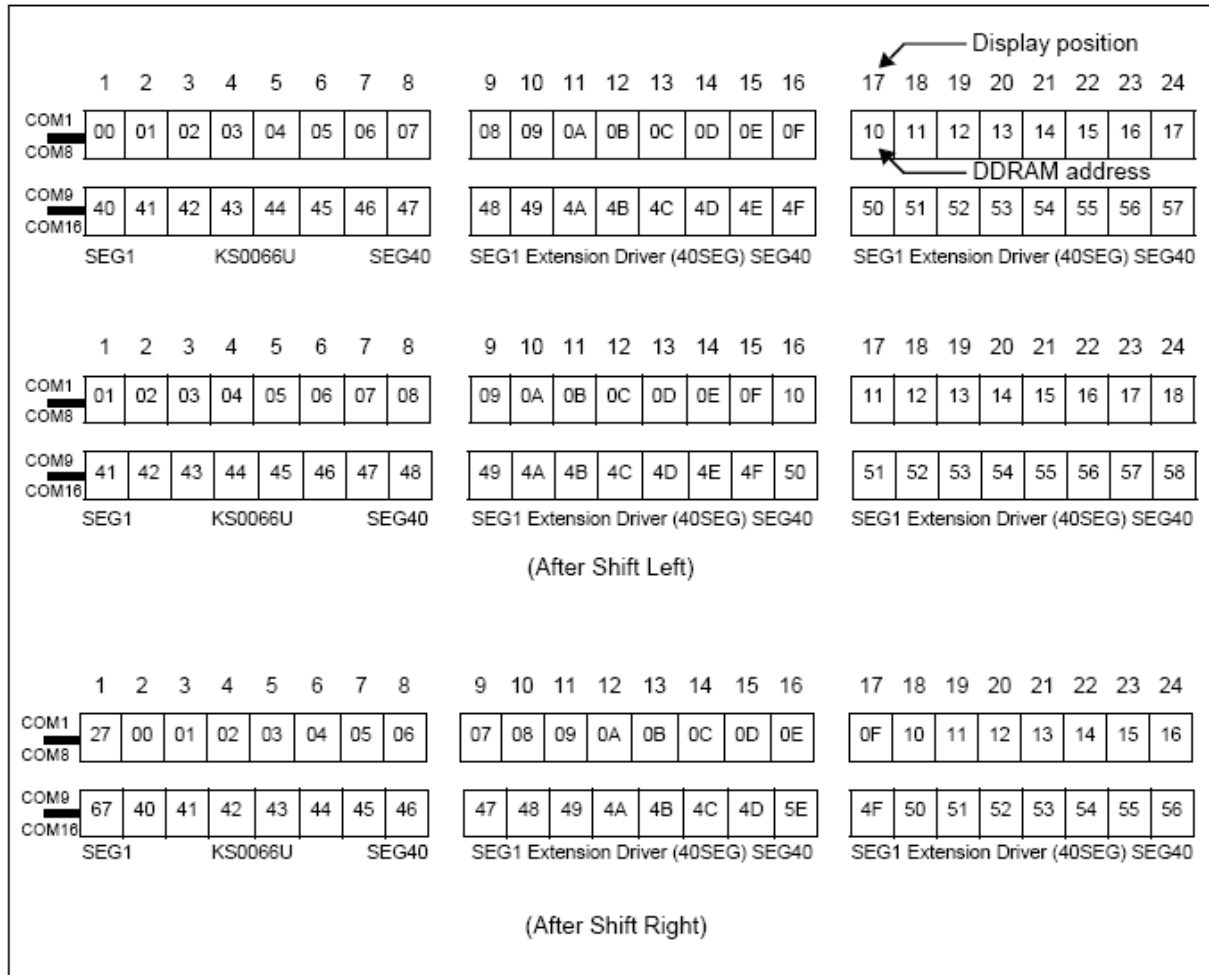
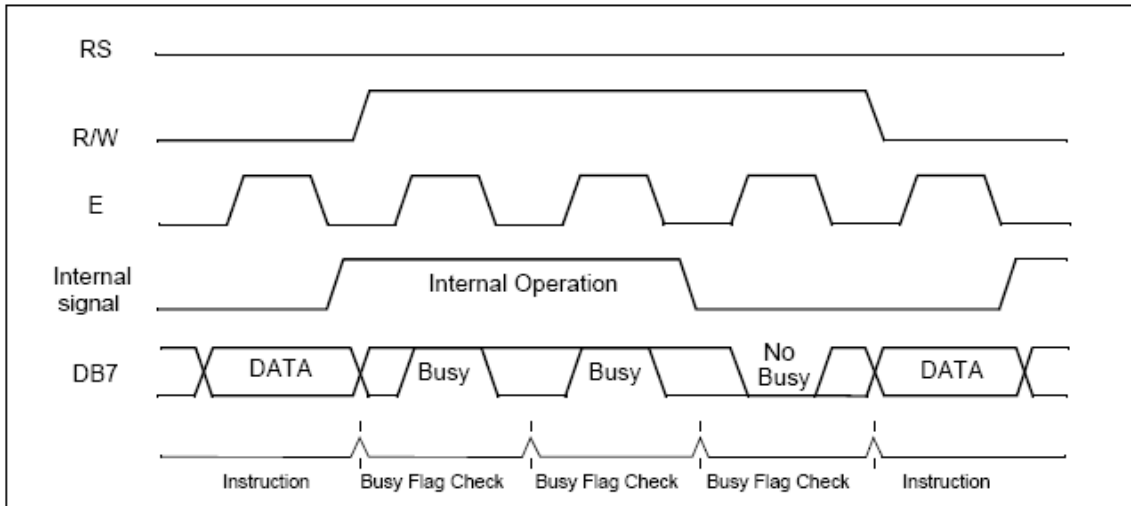


Figure 3 . 2-line x 24 char. display with 40 SEG. extension driver

**INTERFACE WITH MPU**

1) Interface with 8-bit MPU

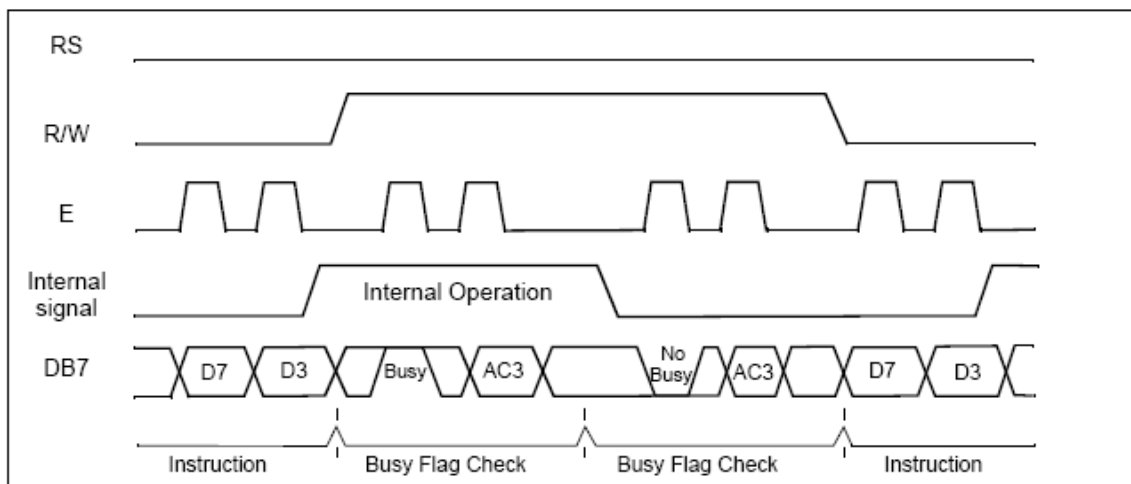
When interfacing data length are 8-bit, transfer is performed at a time through 8 ports, from DB0 to DB7. Example of timing sequence is shown below.



**Figure 4 . Example of 8-bit Bus Mode Timing Diagram**

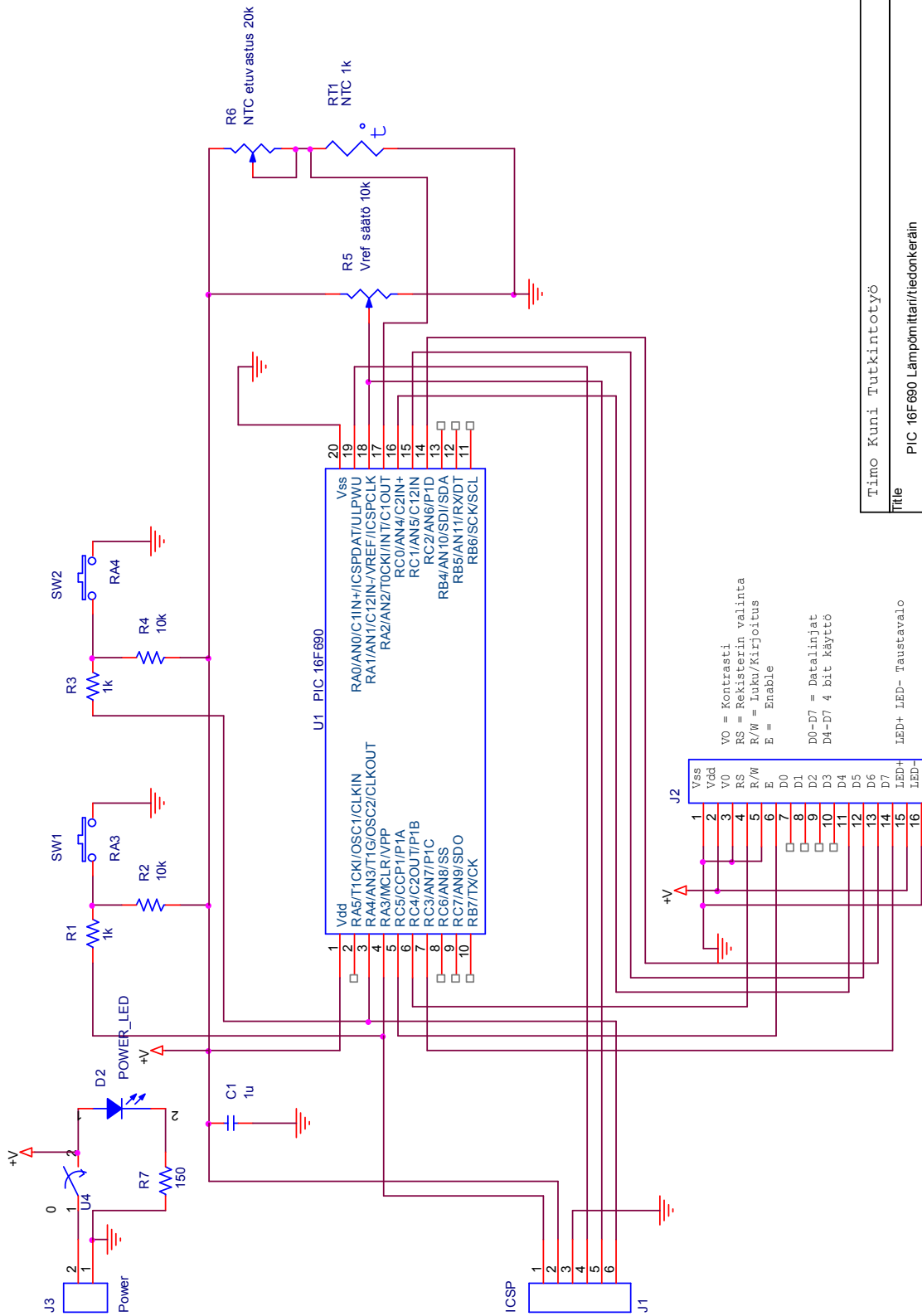
2) Interface with 4-bit MPU

When interfacing data length are 4-bit, only 4 ports, from DB4 to DB7, are used as data bus. At First, the higher 4-bit (in case of 8-bit bus mode, the contents of DB4 - DB7), and then the lower 4-bit (in case of 8-bit bus mode, the contents of DB0 - DB3) are transferred. So transfer is performed twice. Busy Flag outputs 'High" after the second transfer is ended. Example of timing sequence is shown below.



**Figure 5 . Example of 4-bit Bus Mode Timing Diagram**





VO = Kontrasti  
 RS = Rekisterin valinta  
 R/W = Luku/Kirjoitus  
 E = Enable  
 D0-D7 = Datalinjat  
 D4-D7 4 bit käyttö

SAMSUNG KS0066U 2x16 LCD

Timo Kuni Tutkintotyö	
Title PIC 16F690 Lämpömittari/teidonkeräin	
Size A	Document Number
1	Rev 1
Date: Thursday, March 30, 2006	Sheet 1 of 1

$\beta$ at 25°C/85°C		CURVE A 3975			CURVE B 3942			
Temp °C	Typical $R_{25} = 1K$ to 100K	RATIO	TOL	NTC	Typical $R_{25} = 10K$ to 100K	RATIO	TOL	NTC
		$R_1/R_{25}$ RATIO			$R_1/R_{25}$ RATIO			
-50	67.13	3.5	7.1	6.7	56.39	3.5	6.7	3.3
-45	47.26	6.9	6.9	6.5	40.56	6.5	6.5	3.2
-40	33.69	3.0	6.7	6.3	29.48	3.0	6.3	3.2
-35	24.29	6.4	6.4	6.1	21.64	6.1	6.1	3.1
-30	17.71	2.4	6.2	5.9	16.03	2.4	5.9	3.0
-25	13.05	6.0	6.0	5.7	11.99	5.7	5.7	2.9
-20	9.711	1.9	5.8	5.6	9.040	1.9	5.6	2.9
-15	7.297	5.6	5.6	5.4	6.875	5.4	5.4	2.8
-10	5.534	1.4	5.4	5.2	5.270	1.4	5.2	2.8
-5	4.234	5.3	5.3	5.1	4.071	5.1	5.1	2.7
0	5.266	1.0	5.1	4.9	3.168	1.0	4.9	2.6
5	2.540	5.0	5.0	4.8	2.483	4.8	4.8	2.6
10	1.991	0.5	4.8	4.7	1.959	0.5	4.7	2.5
15	1.572	4.7	4.7	4.5	1.556	4.5	4.5	2.5
20	1.249	0.1	4.5	4.4	1.244	0.1	4.4	2.4
25	1.000	0.0	4.4	4.3	1.000	0.0	4.3	2.4
30	0.8056	0.2	4.3	4.2	0.8088	0.2	4.2	2.4
35	0.6530	4.1	4.1	4.1	0.6579	4.1	4.1	2.3
37	0.6014	4.1	4.1	4.0	0.6066	4.0	4.0	2.3
40	0.5325	0.6	4.0	4.0	0.5380	0.6	4.0	2.3
45	0.4367	3.9	3.9	3.9	0.4423	3.9	3.9	2.3
50	0.3601	1.0	3.8	3.8	0.3654	1.0	3.8	2.3
55	0.2985	3.7	3.7	3.7	0.3034	3.7	3.7	2.3
60	0.2487	1.2	3.6	3.6	0.2531	1.2	3.6	2.3
65	0.2082	3.5	3.5	3.5	0.2121	3.5	3.5	2.3
70	0.1752	1.6	3.4	3.4	0.1785	1.6	3.4	2.3

$\beta$ at 25°C/85°C		CURVE A 3975			CURVE B 3942			
Temp °C	Typical $R_{25} = 1K$ to 100K	RATIO	TOL	NTC	Typical $R_{25} = 10K$ to 100K	RATIO	TOL	NTC
		$R_1/R_{25}$ RATIO			$R_1/R_{25}$ RATIO			
75	0.1480	2.1	3.1	3.1	0.1508	2.1	3.1	3.1
80	0.1256	1.9	3.2	3.2	0.1280	1.9	3.2	3.2
85	0.1071	2.1	3.1	3.1	0.1091	2.1	3.1	3.1
90	0.09161	2.1	3.1	3.1	0.09327	2.1	3.1	3.1
95	0.07870	3.0	3.0	3.0	0.08006	3.0	3.0	3.0
100	0.06786	2.4	2.9	2.9	0.06897	2.4	2.9	2.9
105	0.05873	2.9	2.9	2.9	0.05962	2.9	2.9	2.9
110	0.05100	2.6	2.8	2.8	0.05171	2.6	2.8	2.8
115	0.04444	2.7	2.7	2.7	0.04500	2.7	2.7	2.7
120	0.03885	2.9	2.7	2.7	0.03928	2.9	2.7	2.7
125	0.03408	3.0	2.6	2.6	0.03439	3.0	2.6	2.6
130	0.02997	3.1	2.5	2.5	0.03020	3.1	2.5	2.5
135	0.02645	2.5	2.5	2.5	0.02660	2.5	2.5	2.5
140	0.02340	3.4	2.4	2.4	0.02349	3.4	2.4	2.4
145	0.02076	2.4	2.4	2.4	0.02080	2.4	2.4	2.4
150	0.01847	3.5	2.3	2.3	0.01846	3.5	2.3	2.3

Resistance Temperature Tables: Curves A & B

For application configurations, see [Applications link](#)



TEMP/C	RATIO	R/ohm	I/A	Jännite NTC yli/U	Arvo/BIN	Arvo/DES	ERO
-50	67,13	67130	6E-05	4,026	11111111	255	0
-45	47,26	47260	8E-05	4,0111	11111110	254	1
-40	33,69	33690	0,0001	3,9909	11111100	253	1
-35	24,29	24290	0,0002	3,9641	11111011	251	2
-30	17,71	17710	0,0002	3,9289	11111000	249	2
-25	13,05	13050	0,0003	3,8835	11110101	246	3
-20	9,711	9711	0,0004	3,8256	11110010	242	4
-15	7,297	7297	0,0005	3,7534	11101101	238	5
-10	5,534	5534	0,0007	3,6647	11101000	232	6
-5	4,234	4234	0,0008	3,5578	11100001	225	7
0	3,168	3168	0,0011	3,4152	11011000	216	9
5	2,54	2540	0,0013	3,2858	11010000	208	8
10	1,991	1991	0,0016	3,1214	11000101	198	10
15	1,572	1572	0,0019	2,9399	10111010	186	11
20	1,249	1249	0,0022	2,7439	10101101	174	12
25	1	1000	0,0025	2,5388	10100000	161	13
30	0,806	805,6	0,0029	2,3281	10010011	147	13
35	0,653	653	0,0032	2,1169	10000110	134	13
40	0,533	532,5	0,0036	1,9099	01111000	121	13
45	0,437	436,7	0,0039	1,7111	01101100	108	13
50	0,36	360,1	0,0042	1,5235	01100000	96	12
55	0,299	298,5	0,0045	1,3495	01010101	85	11
60	0,249	248,7	0,0048	1,1903	01001011	75	10
65	0,208	208,2	0,005	1,0464	01000010	66	9
70	0,175	175,2	0,0052	0,918	00111010	58	8
75	0,148	148	0,0054	0,8037	00110010	51	7
80	0,126	125,6	0,0056	0,7031	00101100	45	6
85	0,107	107,1	0,0057	0,6152	00100110	39	6
90	0,092	91,61	0,0059	0,538	00100010	34	5
95	0,079	78,7	0,006	0,471	00011101	30	4
100	0,068	67,86	0,0061	0,4127	00011010	26	4
105	0,059	58,73	0,0062	0,3622	00010110	23	3
110	0,051	51	0,0062	0,3182	00010100	20	3
115	0,044	44,44	0,0063	0,2801	00010001	18	2
120	0,039	38,85	0,0064	0,247	00001111	16	2
125	0,034	34,08	0,0064	0,2183	00001101	14	2
130	0,03	29,97	0,0064	0,1932	00001100	12	2
135	0,026	26,45	0,0065	0,1715	00001010	11	1
140	0,023	23,4	0,0065	0,1525	00001001	10	1
145	0,021	20,76	0,0065	0,1358	00001000	9	1
150	0,018	18,47	0,0066	0,1213	00000111	8	1
150					00000000	0	8

skel VREF/256	0,015788235
NTC 25/ohm	1000
VREF/V	4,026
MAX jännite/V	4,062
Etuvastus NTC	600

```

; ALUSTUS ALKAA
; Prosessorin valinta.
#include <p16F690.inc>

; Prosessorin alustus.
__config (_INTRC_OSC_NOCLKOUT & _WDT_OFF & _PWRTE_OFF & _MCLRE_OFF & _CP_OFF & _CPD_OFF &
_BOR_OFF & _IESO_OFF & _FCMEN_OFF)

; muuttujat 0x20 paikasta eteenpäin
cblock 0x20

    VIIVE
    VIIVEX
    LCD_TEMP                ;lcd merkki temp

    TEST_LASKURI            ;testi laskuri
    TEST_LASKURI2

    ADH_TEMP                ;ad muunnos high temp
    ADL_TEMP                ;ad muunnos low temp

    XOR_TEMP                ;xor muuttuja temp

    NAPIT_A                 ; a portin tiedot.

    DATA_EE_ADDR           ; eeprommi osoite
    KIRJ_EE_ADDR            ; eeprommin kirjoitus osoite.
    DATA_EE_DATA           ; eeprommin luettu/kirjoitettava data
    BIN_TEMP                ; binääri luku apumuuttuja

; tehdään kahdella eri laskurilla toinen laskeen ascii digittejä toinen ihna binäärisenä
ASCII_LASKURI_1            ; 0-9                eli digit 1
ASCII_LASKURI_10           ; 10-99                eli digit 2
ASCII_LASKURI_100          ; 100-999            eli digit 3
BINAARI_LASKURI            ; 0-255                binäärilukuna sama

    W_TEMP                  ; W rekisterin TEMP tiedosto

;EEPROM_TALLENNUS_VIIVE muuttujat
VIIVE_500
VIIVE_125
VIIVE_1

endc

; Vakiot
LCD_RS EQU                4
LCD_E EQU                  5                ;LCD OHJAUS BITIT
;LCD_RW EQU                6                ; Ei käytössä, kytketty maihin.

org 0x000 ; Ohjelmakoodi alkaa

; Porttien määrittely
Start

    bsf STATUS,RP0        ; BANK1
    movlw 0xFF
    movwf TRISA            ; Kaikki A portit sisääntuloiksi
    clrf TRISC             ; Kaikki C portit ulostuloiksi
    movlw 0x10             ; A2D Kello Fosc/8
    movwf ADCON1

    bcf STATUS,RP0        ; BANK 0

    bcf STATUS,RP0        ; BANK 2
    bsf STATUS,RP1

    movlw b'00000111'      ; RA3 ja RA4 digitaaliksi

; Katso pic16f690 datakirja "sivu 98" ANSEL rekisteri

; jätetään vain RA0 RA1 RA2 analogisiksi
movwf ANSEL                ; ansel rekisteriin
clrf ANSELH                ; anselh tyhjäksi kaikki DIGITAALIKSI eli portit RC6,RC7 ja RB4,RB5 digitaaliksi

```

```

bcf          STATUS,RP0      ; BANK 0
bcf          STATUS,RP1

; jos luetaankin portin a eri an porttia
; pitää ADCON0 bitit 5-2 muuttaa 0011 = AN3 tai jos halutaan nolla napaa AN0 0000
; AN2 0010
; AD-muunnos aktiiviseksi 0 bitti 1
; movlw      b'01001001'
; Bitti 7 AFDM kummalle puolelle justivoidaan eli 0 vasen
; Bitti 6 Vref käyttöön (AN1)

movlw       b'01001001'
movwf       ADCON0          ; AN2 kanava käyttöön, Vref AN1 ja AD muunnos aktiiviseksi.

; Nollataan kaikki portit ja W rekisteri.
clrw
clrf        PORTC
clrf        PORTB
clrf        PORTA

; Ascii ja laskurin alustus
movlw       b'00110000'      ; ASCII luku nolla
movwf       ASCII_LASKURI_1  ; 0-9          eli digit 1
movwf       ASCII_LASKURI_10 ; 10-99         eli digit 2
movwf       ASCII_LASKURI_100; 100-999       eli digit 3
clrf        BINAARI_LASKURI

CALL LCD_ALUSTUS_4BIT      ; Alustetaan LCD
CALL TKL_LOGO_4BIT        ; Tulostetaan Logo 1 rville
CALL TOISELLE_RIVILLE     ; Vaihdetaan riviä

; Nollataan Eeprommin osoite ja data tiedot.
CLRF        DATA_EE_ADDR    ; eeprommi osoite
CLRF        KIRJ_EE_ADDR     ; eeprommin kirjoitus osoite.
CLRF        DATA_EE_DATA    ; eeprommin luettu/kirjoitteva data

; ALUSTUS LOPPUU

; Pääsilmukka alkaa.
MainLoop

; Tehdään A/D muunnos
;
nop         ; Odotetaan 5us A/D muunnos kondensaattorin latautuu.
nop         ; Odota 1uS
nop         ;
nop         ;
nop         ;

bsf ADCON0, GO ; Aloita A/D muunnos          GO=bit1 eli GO/_DONE bitti In portA0

          btfss ADCON0, GO      ; Odotetaan että GO bitti laskee. Jos bitti 1 on 0 eli Go/_done bitti on nolla
goto      $-1

; Luetaan Yläbitit
movf       ADRESH,0
MOVWF     ADH_TEMP

; Luetaan Alabitit
BSF STATUS,RP0      ; Bank 1
MOVF ADRESL,W       ; Luetaan 8 alinta bittiä
BCF STATUS,RP0      ; Bank 0
MOVWF ADL_TEMP
; A/D muunnos valmis.

; Katsotaan onko 1s kulunut. Jos on talletetaan arvo muistiin.
CALL EEPROM_TALLENNUS_VIIVE

; Tulostetaan Data ruudulle

```

```

CALL          TOISELLE_RIVILLE

; Tulostetaan 8 alinta bittiä LCD:n alariville.
MOVW         ADH_TEMP,W
CALL         TULOSTA_TEMPERATURE

; Estetään hyppiminen Menun ja päätilan välillä
movlw       0xff
CALL        X_VIIVE500

; tutkitaan mennäänkö menuun.

CALL        MENEKO_MENU

; Pääsilukka loppuu.
goto       MainLoop

;*****
;Laskuri Jolla määritellään, koska mennään tallentamaan arvo Eepromiin.
;*****
; Tässä tapauksessa viive on 1 sekunnin 4 MHz kiteellä.
; Jos laskuri on lopussa, talletaan ohjelma 0 paikasta lähtien dataa eepromiin.
; Joka kerta kun talletetaan, lisää aliohjelma paikkaa yhdellä.
; Jos arvo on 255 hyppää se nollaan.

;
;
; VIIVE_500          EQU          0x23          ; 500us          165
; VIIVE_125         EQU          0x24          ; 125ms          250
; VIIVE_1           EQU 0x25          ; 1s            8
;                   500us*250=125ms
;                   125ms*8=1s
; KIRJ_EE_DAT       Data kirjoitus osoite.

EEPROM_TALLENNUS_VIIVE
  DECFSZ       VIIVE_500          ; jos nolla ohitetaan seuraava
  RETURN
  CALL        EEPROM_TALLENNUS_VIIVE_LISAA125MS
  RETURN

EEPROM_TALLENNUS_VIIVE_LISAA125MS
  MOVLW       d'165'              ;laitetaan alkuarvo 500us viiveeseen BUG
  MOVWF       VIIVE_500

  DECFSZ       VIIVE_125          ;jos nolla ohitetaan seuraava käsky
  RETURN
  CALL        EEPROM_TALLENNUS_VIIVE_LISAA1S
  RETURN

EEPROM_TALLENNUS_VIIVE_LISAA1S
  MOVLW       d'250'              ;laitetaan alkuarvo 125ms viiveeseen BUG
  MOVWF       VIIVE_125

  DECFSZ       VIIVE_1            ; jos nolla ohitetaan seuraava käsky
  RETURN
  MOVLW       0xFF
  ;Tähän mitä tehdään kun 1s on kulunut

; kirjoitetaan data eepromiin
  MOVF        KIRJ_EE_ADDR, W      ; Laitetaan kirjoitusosoite W rekisteriin.
  CALL       WRITE_EEPROM_W

; Kasvatetaan Eeprommin osoitetta yhdellä.
  INCF        KIRJ_EE_ADDR

RETURN
;*****
;Luek EEPROM datan osoitteesta W rekisterissä ja palauttaa sen W rekisteriin
;*****

READ_EEPROM_W
;eepromin lukemisen ruudulle W rekisterin osoittamaan paikkaan

```

```

BSF      STATUS, RP1    ; Bank 2
BCF      STATUS, RP0    ;

MOVWF    EEADR          ; laitetaan EEADR:diin W sisältämä data
MOVLW    0x00
; Laitetaan yläosanollaksi
MOVWF    EEADRH        ; osoitteesta jottei lue väärästä paikkaa.

;Osoite luettavaksi
BSF      STATUS, RP0    ; Bank 3
BCF      EECON1, EEPGD  ; Osoitetaan DATA
;Muistiin
BCF      EECON1, RD     ; EE luku
BCF      STATUS, RP1    ; Bank 2
MOVF     EEDAT, W       ; W = EEDAT
BCF      STATUS, RP0    ; Bank 0

RETURN

;*****
; Kirjoittaa W rekisterin osoittamaan Eepro:iin paikkaan ADH_TEMP sisällön
;*****
;
WRITE_EEPROM_W

BCF      STATUS, RP0    ; Mennään Bank 2
BSF      STATUS, RP1    ;

MOVWF    EEADR          ; Siirretään osoite W rekisteristä EEADR rekisteriin, kirjoitus osoitteeksi Eeprommiin.
MOVLW    ADH_TEMP       ; Siirretään data W rekisteriin, 8 ylintä bittiä.
MOVWF    EEDAT          ; Siirretään data EEDAT rekisteriin kirjoitettavaksi Eeprommiin.
BSF      STATUS, RP0    ; Mennään Bank 3.
BCF      EECON1, EEPGD  ; Kirjoitetaan data data muistiin eli Eeprom alueella.
BSF      EECON1, WREN   ; Sallitaan kirjoitus.
BCF      INTCON, GIE    ; Otetaan keskeytykset pois käytöstä, ettei kirjoitus häiriinny.
MOVLW    55h            ; Siirretään 55h W rekisteriin.
MOVWF    EECON2         ; Siirretään 55h EECON2 rekisteriin.
MOVLW    AAh            ; Siirretään AAh W rekisteriin.
MOVWF    EECON2         ; Siirretään AAh EECON2 rekisteriin.
BSF      EECON1, WR     ; Asetetaan WR bitti EECON1 rekisteristä ja aloitetaan kirjoitus.
BSF      INTCON, GIE    ; Sallitaan keskeytykset.
SLEEP    ; Odotetaan niin kauan kunnes keskeytys tulee, että kirjoitus valmis.
BCF      EECON1, WREN   ; Otetaan kirjoitus pois käytöstä.
BCF      STATUS, RP0    ; Mennään Bank 0

RETURN

;*****
;TULOSTA_TEMPERTATURE tulostaa lämpötilan 5 asteen välein LCD:lle W tuodusta arvosta.
;*****
;STATUS rekisterissä bitti 2 on Z bitti.
;
TULOSTA_TEMPERATURE

```

```

MOVWF W_TEMP ; Otetaan W arvo talteen.

; 150 ALUE 0-8
ASTE_150
    MOVLW d'0' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    MOVLW d'1' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    MOVLW d'2' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    MOVLW d'3' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    MOVLW d'4' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    MOVLW d'5' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    MOVLW d'6' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    MOVLW d'7' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    MOVLW d'8' ; Laitetaan W:hen
    XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_150

    GOTO ASTE_145 ; Jos ei mikään näistä Seuraava alue.

;TULOSTETAAN 150.
TULOSTA_150
    MOVLW a'1'
    CALL LCD_MERKKI_4BIT
    MOVLW a'5'
    CALL LCD_MERKKI_4BIT
    MOVLW a'0'
    CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;145 ALUE 9
ASTE_145
    MOVLW d'9' ; Laitetaan W:hen

```



```

XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_145

GOTO ASTE_140 ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSETAAN 145.
TULOSTA_145
MOVLW a'1'
CALL LCD_MERKKI_4BIT
MOVLW a'4'
CALL LCD_MERKKI_4BIT
MOVLW a'5'
CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

```

;140 ALUE 10
ASTE_140

```

```

MOVLW d'10' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_140

GOTO ASTE_135 ; Jos ei mikään hypätään seuraavaan alueeseen.

```

```

;TULOSETAAN 140.

```

```

TULOSTA_140
MOVLW a'1'
CALL LCD_MERKKI_4BIT
MOVLW a'4'
CALL LCD_MERKKI_4BIT
MOVLW a'0'
CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

```

;135 ALUE 11
ASTE_135

```

```

MOVLW d'11' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_135

GOTO ASTE_130 ; Jos ei mikään hypätään seuraavaan alueeseen.

```

```

;TULOSETAAN 135.

```

```

TULOSTA_135
MOVLW a'1'
CALL LCD_MERKKI_4BIT
MOVLW a'3'
CALL LCD_MERKKI_4BIT
MOVLW a'5'
CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

```

;130 ALUE 12-13
ASTE_130

```

```

MOVLW d'12' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_130

MOVLW d'13' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_130

```

```

        GOTO        ASTE_125        ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 130.
TULOSTA_130
        MOVLW      a'1'
        CALL       LCD_MERKKI_4BIT
        MOVLW      a'3'
        CALL       LCD_MERKKI_4BIT
        MOVLW      a'0'
        CALL       LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI      ; Mennään ohjelman loppuun.

;125 ALUE 14-15
ASTE_125

        MOVLW      d'14'           ; Laitetaan W:hen
        XORWF      W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFS      STATUS, Z       ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO      TULOSTA_125

        MOVLW      d'15'           ; Laitetaan W:hen
        XORWF      W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFS      STATUS, Z       ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO      TULOSTA_125

        GOTO      ASTE_120        ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 125.
TULOSTA_125
        MOVLW      a'1'
        CALL       LCD_MERKKI_4BIT
        MOVLW      a'2'
        CALL       LCD_MERKKI_4BIT
        MOVLW      a'5'
        CALL       LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI      ; Mennään ohjelman loppuun.

;120 ALUE 16-17
ASTE_120

        MOVLW      d'16'           ; Laitetaan W:hen
        XORWF      W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFS      STATUS, Z       ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO      TULOSTA_120

        MOVLW      d'17'           ; Laitetaan W:hen
        XORWF      W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFS      STATUS, Z       ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO      TULOSTA_120

        GOTO      ASTE_115        ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 120.
TULOSTA_120
        MOVLW      a'1'
        CALL       LCD_MERKKI_4BIT
        MOVLW      a'2'
        CALL       LCD_MERKKI_4BIT
        MOVLW      a'0'
        CALL       LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI      ; Mennään ohjelman loppuun.

;115 ALUE 18-19
ASTE_115

        MOVLW      d'18'           ; Laitetaan W:hen

```

```

    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_115

    MOVLW    d'19'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_115

    GOTO     ASTE_110          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 115.
TULOSTA_115
    MOVLW    a'1'              ;
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'1'              ;
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'5'              ;
    CALL     LCD_MERKKI_4BIT
    GOTO     TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;110 ALUE 20-22
ASTE_110

    MOVLW    d'20'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_110

    MOVLW    d'21'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_110

    MOVLW    d'22'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_110

    GOTO     ASTE_105          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 110.
TULOSTA_110
    MOVLW    a'1'              ;
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'1'              ;
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'0'              ;
    CALL     LCD_MERKKI_4BIT
    GOTO     TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;105 ALUE 23-25
ASTE_105

    MOVLW    d'23'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_105

    MOVLW    d'24'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_105

    MOVLW    d'25'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.

```

```

GOTO          TULOSTA_105

GOTO          ASTE_100      ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 105.
TULOSTA_105
  MOVLW      a'1'
  CALL       LCD_MERKKI_4BIT
  MOVLW      a'0'
  CALL       LCD_MERKKI_4BIT
  MOVLW      a'5'
  CALL       LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;100 ALUE 26-29
ASTE_100

  MOVLW      d'26'          ; Laitetaan W:hen
  XORWF      W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
  BTFSC     STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
  GOTO      TULOSTA_100

  MOVLW      d'27'          ; Laitetaan W:hen
  XORWF      W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
  BTFSC     STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
  GOTO      TULOSTA_100

  MOVLW      d'28'          ; Laitetaan W:hen
  XORWF      W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
  BTFSC     STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
  GOTO      TULOSTA_100

  MOVLW      d'29'          ; Laitetaan W:hen
  XORWF      W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
  BTFSC     STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
  GOTO      TULOSTA_100

  GOTO      ASTE_95        ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 100.
TULOSTA_100
  MOVLW      a'1'
  CALL       LCD_MERKKI_4BIT
  MOVLW      a'0'
  CALL       LCD_MERKKI_4BIT
  MOVLW      a'0'
  CALL       LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;95 ALUE 30-33
ASTE_95

  MOVLW      d'30'          ; Laitetaan W:hen
  XORWF      W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
  BTFSC     STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
  GOTO      TULOSTA_95

  MOVLW      d'31'          ; Laitetaan W:hen
  XORWF      W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
  BTFSC     STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
  GOTO      TULOSTA_95

  MOVLW      d'32'          ; Laitetaan W:hen
  XORWF      W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
  BTFSC     STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.

```

```

        GOTO          TULOSTA_95

        MOVLW        d'33'          ; Laitetaan W:hen
        XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO          TULOSTA_95

        GOTO          ASTE_90        ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 95.
TULOSTA_95
        MOVLW        a'0'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'9'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'5'
        CALL         LCD_MERKKI_4BIT
        GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;90 ALUE 34-38
ASTE_90

        MOVLW        d'34'          ; Laitetaan W:hen
        XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO          TULOSTA_90

        MOVLW        d'35'          ; Laitetaan W:hen
        XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO          TULOSTA_90

        MOVLW        d'36'          ; Laitetaan W:hen
        XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO          TULOSTA_90

        MOVLW        d'37'          ; Laitetaan W:hen
        XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO          TULOSTA_90

        MOVLW        d'38'          ; Laitetaan W:hen
        XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO          TULOSTA_90

        GOTO          ASTE_85        ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 90.
TULOSTA_90
        MOVLW        a'0'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'9'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'0'
        CALL         LCD_MERKKI_4BIT
        GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;85 ALUE 39-44
ASTE_85

        MOVLW        d'39'          ; Laitetaan W:hen

```

```

    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_85

    MOVLW    d'40'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_85

    MOVLW    d'41'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_85

    MOVLW    d'42'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_85

    MOVLW    d'43'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_85

    MOVLW    d'44'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_85

    GOTO     ASTE_80           ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 85.
TULOSTA_85
    MOVLW    a'0'              ; Laitetaan W:hen
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'8'              ; Laitetaan W:hen
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'5'              ; Laitetaan W:hen
    CALL     LCD_MERKKI_4BIT
    GOTO     TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;80 ALUE 45-50
ASTE_80

    MOVLW    d'45'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_80

    MOVLW    d'46'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_80

    MOVLW    d'47'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_80

    MOVLW    d'48'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_80

    MOVLW    d'49'             ; Laitetaan W:hen

```

```

    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_80

    MOVLW   d'50'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_80

    GOTO    ASTE_75            ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 80.
TULOSTA_80
    MOVLW   a'0'
    CALL    LCD_MERKKI_4BIT
    MOVLW   a'8'
    CALL    LCD_MERKKI_4BIT
    MOVLW   a'0'
    CALL    LCD_MERKKI_4BIT
    GOTO    TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;75 ALUE 51-57
ASTE_75

    MOVLW   d'51'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_75

    MOVLW   d'52'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_75

    MOVLW   d'53'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_75

    MOVLW   d'54'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_75

    MOVLW   d'55'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_75

    MOVLW   d'56'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_75

    MOVLW   d'57'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_75

    GOTO    ASTE_70            ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 75.
TULOSTA_75
    MOVLW   a'0'
    CALL    LCD_MERKKI_4BIT
    MOVLW   a'7'

```

```

CALL      LCD_MERKKI_4BIT
MOVLW    a'5'
CALL      LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

```

;70 ALUE 58-65
ASTE_70

```

```

MOV LW    d'58' ; Laitetaan W:hen
XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFS    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO     TULOSTA_70

```

```

MOV LW    d'59' ; Laitetaan W:hen
XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFS    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO     TULOSTA_70

```

```

MOV LW    d'60' ; Laitetaan W:hen
XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFS    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO     TULOSTA_70

```

```

MOV LW    d'61' ; Laitetaan W:hen
XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFS    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO     TULOSTA_70

```

```

MOV LW    d'62' ; Laitetaan W:hen
XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFS    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO     TULOSTA_70

```

```

MOV LW    d'63' ; Laitetaan W:hen
XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFS    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO     TULOSTA_70

```

```

MOV LW    d'64' ; Laitetaan W:hen
XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFS    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO     TULOSTA_70

```

```

MOV LW    d'65' ; Laitetaan W:hen
XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFS    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO     TULOSTA_70

```

```

GOTO     ASTE_65 ; Jos ei mikään hypätään seuraavaan alueeseen.

```

```

;TULOSETAAN 70.
TULOSTA_70

```

```

MOV LW    a'0'
CALL      LCD_MERKKI_4BIT
MOV LW    a'7'
CALL      LCD_MERKKI_4BIT
MOV LW    a'0'
CALL      LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

```

;65 ALUE 66-74
ASTE_65

```

```

MOV LW    d'66' ; Laitetaan W:hen

```



```

    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    MOVLW    d'67'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    MOVLW    d'68'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    MOVLW    d'69'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    MOVLW    d'70'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    MOVLW    d'71'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    MOVLW    d'72'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    MOVLW    d'73'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    MOVLW    d'74'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_65

    GOTO     ASTE_60           ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 65.
TULOSTA_65
    MOVLW    a'0'              ;
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'6'              ;
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'5'              ;
    CALL     LCD_MERKKI_4BIT
    GOTO     TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;60 ALUE 75-84
ASTE_60

    MOVLW    d'75'             ; Laitetaan W:hen
    XORWF    W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_60

    MOVLW    d'76'             ; Laitetaan W:hen

```

```

    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    MOVLW   d'77'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    MOVLW   d'78'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    MOVLW   d'79'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    MOVLW   d'80'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    MOVLW   d'81'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    MOVLW   d'82'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    MOVLW   d'83'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    MOVLW   d'84'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_60

    GOTO    ASTE_55             ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 60.
TULOSTA_60
    MOVLW   a'0'               ;
    CALL    LCD_MERKKI_4BIT
    MOVLW   a'6'               ;
    CALL    LCD_MERKKI_4BIT
    MOVLW   a'0'               ;
    CALL    LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;55 ALUE 85-95
ASTE_55

    MOVLW   d'85'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'86'              ; Laitetaan W:hen

```

```

    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'87'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'88'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'89'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'90'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'91'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'92'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'93'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'94'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    MOVLW   d'95'              ; Laitetaan W:hen
    XORWF   W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC   STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO    TULOSTA_55

    GOTO    ASTE_50           ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 55.
TULOSTA_55
    MOVLW   a'0'
    CALL    LCD_MERKKI_4BIT
    MOVLW   a'5'
    CALL    LCD_MERKKI_4BIT
    MOVLW   a'5'
    CALL    LCD_MERKKI_4BIT
    GOTO    TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;50 ALUE 96-107
ASTE_50

    MOVLW   d'96'              ; Laitetaan W:hen

```

```

    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'97'               ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'98'               ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'99'               ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_55

    MOVLW    d'100'              ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'101'              ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'102'              ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'103'              ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'104'              ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'105'              ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'106'              ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    MOVLW    d'107'              ; Laitetaan W:hen
    XORWF    W_TEMP, 0           ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z           ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_50

    GOTO     ASTE_45             ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 50
TULOSTA_50
    MOVLW    a'0'

```

```

CALL          LCD_MERKKI_4BIT
MOVLW        a'5'
CALL          LCD_MERKKI_4BIT
MOVLW        a'0'
CALL          LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;45 ALUE 108-120
ASTE_45

    MOVLW        d'108'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'109'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'110'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'111'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'112'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'113'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'114'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'115'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'116'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'117'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

    MOVLW        d'118'          ; Laitetaan W:hen
    XORWF        W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC        STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO        TULOSTA_45

```

```

    MOVLW      d'119'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_45

    MOVLW      d'120'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_45

    GOTO      ASTE_40;      Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 45
TULOSTA_45
    MOVLW      a'0'
    CALL      LCD_MERKKI_4BIT
    MOVLW      a'4'
    CALL      LCD_MERKKI_4BIT
    MOVLW      a'5'
    CALL      LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;40 ALUE 121-133
ASTE_40

    MOVLW      d'121'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'122'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'123'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'124'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'125'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'126'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'127'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'128'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

```

```

    MOVLW      d'129'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'130'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'131'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'132'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    MOVLW      d'133'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_40

    GOTO      ASTE_35      ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 40
TULOSTA_40
    MOVLW      a'0'
    CALL      LCD_MERKKI_4BIT
    MOVLW      a'4'
    CALL      LCD_MERKKI_4BIT
    MOVLW      a'0'
    CALL      LCD_MERKKI_4BIT
    GOTO      TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;35 ALUE 134-146
ASTE_35

    MOVLW      d'134'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'135'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'136'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'137'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'138'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

```

```

    MOVLW      d'139'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'140'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'141'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'142'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'143'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'144'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'145'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    MOVLW      d'146'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_35

    GOTO      ASTE_30      ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 35
TULOSTA_35
    MOVLW      a'0'
    CALL      LCD_MERKKI_4BIT
    MOVLW      a'3'
    CALL      LCD_MERKKI_4BIT
    MOVLW      a'5'
    CALL      LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;30 ALUE 147-160
ASTE_30

    MOVLW      d'147'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_30

    MOVLW      d'148'      ; Laitetaan W:hen
    XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_30

```



```

MOV LW      d'149'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'150'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'151'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'152'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'153'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'154'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'155'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'156'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'157'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'158'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'159'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

MOV LW      d'160'      ; Laitetaan W:hen
XORWF W_TEMP, 0      ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.

BTFSC      STATUS, Z   ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO      TULOSTA_30

GOTO      ASTE_25      ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 30
TULOSTA_30

```

```

    MOVLW    a'0'
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'3'
    CALL     LCD_MERKKI_4BIT
    MOVLW    a'0'
    CALL     LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI    ; Mennään ohjelman loppuun.

```

```

;25 ALUE 161-173
ASTE_25

```

```

    MOVLW    d'161'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'162'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'163'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'164'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'165'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'166'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'167'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'168'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'169'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'170'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO     TULOSTA_25

    MOVLW    d'171'    ; Laitetaan W:hen
    XORWF    W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC    STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.

```

```

GOTO          TULOSTA_25

rekisteriin.  MOVLW          d'172'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_25

rekisteriin.  MOVLW          d'173'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_25

GOTO          ASTE_20          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 25
TULOSTA_25
MOVLW        a'0'
CALL         LCD_MERKKI_4BIT
MOVLW        a'2'
CALL         LCD_MERKKI_4BIT
MOVLW        a'5'
CALL         LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;20 ALUE 174-185
ASTE_20

rekisteriin.  MOVLW          d'174'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'175'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'176'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'177'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'178'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'179'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'180'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'181'          ; Laitetaan W:hen
XORWF        W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS        STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.

```

```

GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'182'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'183'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'184'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

rekisteriin.  MOVLW          d'185'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_20

GOTO          ASTE_15          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 20
TULOSTA_20
MOVLW          a'0'
CALL           LCD_MERKKI_4BIT
MOVLW          a'2'
CALL           LCD_MERKKI_4BIT
MOVLW          a'0'
CALL           LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;15 ALUE 186-197
ASTE_15

rekisteriin.  MOVLW          d'186'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'187'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'188'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'189'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'190'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'191'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.

```

```

GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'192'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'193'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'194'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'195'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'196'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

rekisteriin.  MOVLW          d'197'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_15

GOTO          ASTE_10          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 15
TULOSTA_15
MOVLW          a'0'
CALL          LCD_MERKKI_4BIT
MOVLW          a'1'
CALL          LCD_MERKKI_4BIT
MOVLW          a'5'
CALL          LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI          ; Mennään ohjelman loppuun.

;10 ALUE 198-207
ASTE_10

rekisteriin.  MOVLW          d'198'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'199'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'200'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'201'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.

```

```

GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'202'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'203'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'204'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'205'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'206'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

rekisteriin.  MOVLW          d'207'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_10

GOTO          ASTE_5          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 10
TULOSTA_10
MOVLW          a'0'
CALL           LCD_MERKKI_4BIT
MOVLW          a'1'
CALL           LCD_MERKKI_4BIT
MOVLW          a'0'
CALL           LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;5 ALUE 208-215
ASTE_5

rekisteriin.  MOVLW          d'208'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_5

rekisteriin.  MOVLW          d'209'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_5

rekisteriin.  MOVLW          d'210'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

BTFS          STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO          TULOSTA_5

rekisteriin.  MOVLW          d'211'          ; Laitetaan W:hen
XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

```

```

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_5

    rekisteriin.
    MOVLW     d'212'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_5

    rekisteriin.
    MOVLW     d'213'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_5

    rekisteriin.
    MOVLW     d'214'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_5

    rekisteriin.
    MOVLW     d'215'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_5

    GOTO      ASTE_0         ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 5
TULOSTA_5
    MOVLW     a'0'           ;
    CALL     LCD_MERKKI_4BIT
    MOVLW     a'0'           ;
    CALL     LCD_MERKKI_4BIT
    MOVLW     a'5'           ;
    CALL     LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;0 ALUE 216-224
ASTE_0

    rekisteriin.
    MOVLW     d'216'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    rekisteriin.
    MOVLW     d'217'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    rekisteriin.
    MOVLW     d'218'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    rekisteriin.
    MOVLW     d'219'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    rekisteriin.
    MOVLW     d'220'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    rekisteriin.
    MOVLW     d'221'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

```

```

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    rekisteriin.
    MOVLW     d'222'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    rekisteriin.
    MOVLW     d'223'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    rekisteriin.
    MOVLW     d'224'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_0

    GOTO      ASTE_N5        ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN 0
TULOSTA_0
    MOVLW     a'0'
    CALL     LCD_MERKKI_4BIT
    MOVLW     a'0'
    CALL     LCD_MERKKI_4BIT
    MOVLW     a'0'
    CALL     LCD_MERKKI_4BIT
    GOTO     TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;N5 ALUE 225-231 N=negative
ASTE_N5

    rekisteriin.
    MOVLW     d'225'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_N5

    rekisteriin.
    MOVLW     d'226'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_N5

    rekisteriin.
    MOVLW     d'227'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_N5

    rekisteriin.
    MOVLW     d'228'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_N5

    rekisteriin.
    MOVLW     d'229'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_N5

    rekisteriin.
    MOVLW     d'230'          ; Laitetaan W:hen
    XORWF    W_TEMP, 0       ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W

    BTFSC      STATUS, Z      ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO      TULOSTA_N5

    MOVLW     d'231'          ; Laitetaan W:hen

```



```

XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_N5

GOTO ASTE_N10 ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN N5
TULOSTA_N5
MOVLW a'-
CALL LCD_MERKKI_4BIT
MOVLW a'0'
CALL LCD_MERKKI_4BIT
MOVLW a'5'
CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

```

;N10 ALUE 232-237 N=negative
ASTE_N10

```

```

MOVLW d'232' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_N10

```

```

MOVLW d'233' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_N10

```

```

MOVLW d'234' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_N10

```

```

MOVLW d'235' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_N10

```

```

MOVLW d'236' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_N10

```

```

MOVLW d'237' ; Laitetaan W:hen
XORWF W_TEMP, 0 ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
BTFSC STATUS, Z ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
GOTO TULOSTA_N10

```

```

GOTO ASTE_N15 ; Jos ei mikään hypätään seuraavaan alueeseen.

```

```

;TULOSTETAAN N10
TULOSTA_N10

```

```

MOVLW a'-
CALL LCD_MERKKI_4BIT
MOVLW a'1'
CALL LCD_MERKKI_4BIT
MOVLW a'0'
CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

```

;N15 ALUE 238-241 N=negative
ASTE_N15

```

```

MOVLW d'238' ; Laitetaan W:hen

```

```

    XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_N15

    MOVLW d'239'             ; Laitetaan W:hen
    XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_N15

    MOVLW d'240'             ; Laitetaan W:hen
    XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_N15

    MOVLW d'241'             ; Laitetaan W:hen
    XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_N15

    GOTO ASTE_N20            ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSETAAN N15
TULOSTA_N15
    MOVLW a'-'               ;
    CALL LCD_MERKKI_4BIT
    MOVLW a'1'               ;
    CALL LCD_MERKKI_4BIT
    MOVLW a'5'               ;
    CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

;N20 ALUE 242-245 N=negative
ASTE_N20

    MOVLW d'242'             ; Laitetaan W:hen
    XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_N20

    MOVLW d'243'             ; Laitetaan W:hen
    XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_N20

    MOVLW d'244'             ; Laitetaan W:hen
    XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_N20

    MOVLW d'245'             ; Laitetaan W:hen
    XORWF W_TEMP, 0          ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
    BTFSC STATUS, Z          ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
    GOTO TULOSTA_N20

    GOTO ASTE_N25            ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSETAAN N20
TULOSTA_N20
    MOVLW a'-'               ;
    CALL LCD_MERKKI_4BIT
    MOVLW a'2'               ;
    CALL LCD_MERKKI_4BIT
    MOVLW a'0'               ;
    CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

;N25 ALUE 246-248 N=negative  
ASTE\_N25

```

      MOVLW      d'246'      ; Laitetaan W:hen
      XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
      BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
      GOTO      TULOSTA_N25

      MOVLW      d'247'      ; Laitetaan W:hen
      XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
      BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
      GOTO      TULOSTA_N25

      MOVLW      d'248'      ; Laitetaan W:hen
      XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
      BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
      GOTO      TULOSTA_N25

      GOTO      ASTE_N30     ; Jos ei mikään hypätään seuraavaan alueeseen.

```

;TULOSETAAN N25  
TULOSTA\_N25

```

      MOVLW      a'1'
      CALL      LCD_MERKKI_4BIT
      MOVLW      a'2'
      CALL      LCD_MERKKI_4BIT
      MOVLW      a'5'
      CALL      LCD_MERKKI_4BIT
      GOTO      TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

;N30 ALUE 249-250 N=negative  
ASTE\_N30

```

      MOVLW      d'249'      ; Laitetaan W:hen
      XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
      BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
      GOTO      TULOSTA_N30

      MOVLW      d'250'      ; Laitetaan W:hen
      XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
      BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
      GOTO      TULOSTA_N30

      GOTO      ASTE_N35     ; Jos ei mikään hypätään seuraavaan alueeseen.

```

;TULOSETAAN N30  
TULOSTA\_N30

```

      MOVLW      a'1'
      CALL      LCD_MERKKI_4BIT
      MOVLW      a'3'
      CALL      LCD_MERKKI_4BIT
      MOVLW      a'0'
      CALL      LCD_MERKKI_4BIT
      GOTO      TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

;N35 ALUE 251-252 N=negative  
ASTE\_N35

```

      MOVLW      d'251'      ; Laitetaan W:hen
      XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
      BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
      GOTO      TULOSTA_N35

      MOVLW      d'252'      ; Laitetaan W:hen
      XORWF     W_TEMP, 0    ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
      BTFSC     STATUS, Z    ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
      GOTO      TULOSTA_N35

```

```

        GOTO          ASTE_N40          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN N35
TULOSTA_N35
        MOVLW        a'-'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'3'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'5'
        CALL         LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI          ; Mennään ohjelman loppuun.

;N40 ALUE 253-253 N=negative
ASTE_N40

        MOVLW        d'253'           ; Laitetaan W:hen
        XORWF        W_TEMP, 0        ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z        ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO         TULOSTA_N40

        GOTO          ASTE_N45          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN N40
TULOSTA_N40
        MOVLW        a'-'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'4'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'0'
        CALL         LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI          ; Mennään ohjelman loppuun.

;N45 ALUE 254-254 N=negative
ASTE_N45

        MOVLW        d'254'           ; Laitetaan W:hen
        XORWF        W_TEMP, 0        ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z        ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO         TULOSTA_N45

        GOTO          ASTE_N50          ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN N45
TULOSTA_N45
        MOVLW        a'-'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'4'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'5'
        CALL         LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI          ; Mennään ohjelman loppuun.

;N50 ALUE 255-255 N=negative
ASTE_N50

        MOVLW        d'255'           ; Laitetaan W:hen
        XORWF        W_TEMP, 0        ; Vertaillaan W rekisterin sisältöä XOR operaatiolla W_TEMP sisältöön, tulos W
rekisteriin.
        BTFSC        STATUS, Z        ; Jos Z lippu nousee ylös on tulos sama kuin vertailtava. Hyppää seuraavan käskyn yli.
        GOTO         TULOSTA_N50

        GOTO          TULOSTA_TEMPERATURE_OHI ; Jos ei mikään hypätään seuraavaan alueeseen.

;TULOSTETAAN N50
TULOSTA_N50
        MOVLW        a'-'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'5'
        CALL         LCD_MERKKI_4BIT
        MOVLW        a'0'

```

```

CALL LCD_MERKKI_4BIT
GOTO TULOSTA_TEMPERATURE_OHI ; Mennään ohjelman loppuun.

```

```
TULOSTA_TEMPERATURE_OHI
```

```

;tulostetaan astemerkki ja celsius
MOVLW b'11011111' ; Astemerkki
CALL LCD_MERKKI_4BIT
MOVLW a'C'
CALL LCD_MERKKI_4BIT

```

```
; Tulostetaan 3 riviä tyhjää, saadaan menu merkinnät pois.
```

```

MOVLW a' '
CALL LCD_MERKKI_4BIT
MOVLW a' '
CALL LCD_MERKKI_4BIT
MOVLW a' '
CALL LCD_MERKKI_4BIT
MOVLW a' '
CALL LCD_MERKKI_4BIT

```

```
MOVF W_TEMP, 0 ; Talletetaan W_TEMP takaisin W rekisteriin.
```

```
RETURN
```

```

;*****
;Mennäänkö menuun ja tulostaa menun
;*****
; Tutkii onko Painettu RA3, jos on menen menuu
; Pois pääsy Napeista RA3+RA4
;

```

```
MENEEKO_MENU
```

```

;          BTFSC ohitetaan seuraava käsky jos nolla
;          BTFSS ohitetaan seuraava käsky jos yksi
BTFSC     PORTA, 3 ;Bitti 3 eli RA3 nappi muuten ohi ja takaisin main looppiin
goto      MENEEKO_MENU_EXIT

nop

; Mitä vittua muuten ei toimi jos ei noppa käskyä, tilttaa koko kone

BTFSS     PORTA, 4 ; bitti 4 täytyy olla nolla että voidaan mennä ulos - > estetään
hyppiminen ettei mene
goto      MENEEKO_MENU_EXIT ; kun painetaan menusta ulos yhtäaikaan nappeja takaisin menuun

```

```
;Palataan alkuriville ja tulostetaan "Menu Up/Down 1s"
```

```
CALL PALAA_ALKUUN
```

```

MOVLW     a'M'
CALL      LCD_MERKKI_4BIT

```

```

MOVLW     a'e'
CALL      LCD_MERKKI_4BIT

```

```

MOVLW     a'n'
CALL      LCD_MERKKI_4BIT

```

```

MOVLW     a'u'
CALL      LCD_MERKKI_4BIT

```

```

MOVLW     a' '
CALL      LCD_MERKKI_4BIT

```

```

MOVLW     a'U'
CALL      LCD_MERKKI_4BIT

```

```

MOVLW     a'p'
CALL      LCD_MERKKI_4BIT

```

```

MOVLW     a/'
CALL      LCD_MERKKI_4BIT

```

```

MOVLW     a'D'
CALL      LCD_MERKKI_4BIT

```

```

MOVLW      a'o'
CALL       LCD_MERKKI_4BIT

MOVLW      a'w'
CALL       LCD_MERKKI_4BIT

MOVLW      a'n'
CALL       LCD_MERKKI_4BIT

MOVLW      a' '
CALL       LCD_MERKKI_4BIT

MOVLW      a'l'
CALL       LCD_MERKKI_4BIT

MOVLW      a's'
CALL       LCD_MERKKI_4BIT

MOVLW      a' '
CALL       LCD_MERKKI_4BIT

MENEKO_MENU_LOOP

CALL       YLOS_ALAS                ;tutkii mennäänkö ylös vai alas ja palauttaa W rekisteriin osoitteen
eepromiin lukuun

MOV        BINAARI_LASKURI, W      ; osoite Eepromin lukuun W rekisteriin
CALL      READ_EEPROM_W           ; luetaan Eepromin data W rekisteriin
CALL      TULOSTA_TEMPERATURE     ; Tulostetaan Data ruudulle.

; pois menusta jos 3=0 ja 4=0
; jos porttia 3 on 1 hypätään seuraava yli niin jatketaan
; jos porttia 3 on 0 ja jos porttia 4 on 1 niin jatketaan
; jos porttia 3 on 0 ja jos porttia 4 on 0 niin ulos ohjelmasta
BTFS      PORTA,3                  ;ohitetaan seuraava jos 1
BTFS      PORTA,4                  ;ohitetaan seuraava jos 0
goto     MENEKO_MENU_LOOP

MENEKO_MENU_EXIT                    ; ohitus merkki

;tehostaan koodia
; ei tarvitse mainloopissa joka kierroksella tulostaa
CALL     PALAA_ALKUUN
CALL     TKL_LOGO_4BIT

RETURN

```

```

;*****
;Tutki onko painettu ylös tai alas ja sen mukaan siirtää eeprom osoitetta ja tulostaan ruudulle ascii luvun desimaaleina ja palauttaa
W rekisterissä Eeprommin osoitteen josta lukea
;*****

```

```

;RA4 ylös
;RA3 alas
;ASCII_LASKURI_1 ; 0-9 eli digit 1
;ASCII_LASKURI_10 ; 10-99 eli digit 2
;ASCII_LASKURI_100 ; 100-999 eli digit 3
;BINAARI_LASKUR ; 0-255 binäärilukuna sama

```

YLOS\_ALAS

```

; Mittari ympäri
; ; 0->255
00000000 00000000 00000000 -> 00110010 00110101 00110101
; ; 255->0
00110010 00110101 00110101 -> 00000000 00000000 00000000
; tämän näkee BINAARI_LASKURI muuttujasta kun se on 1111 1111
; jos menee 255 yli nolaa laskurin ja digitit nolliin

```

```

BTFSS BINAARI_LASKURI, 0 ;1
goto OHI_255
BTFSS BINAARI_LASKURI, 1 ;1
goto OHI_255
BTFSS BINAARI_LASKURI, 2 ;1
goto OHI_255
BTFSS BINAARI_LASKURI, 3 ;1
goto OHI_255
BTFSS BINAARI_LASKURI, 4 ;1
goto OHI_255
BTFSS BINAARI_LASKURI, 5 ;1
goto OHI_255
BTFSS BINAARI_LASKURI, 6 ;1
goto OHI_255
BTFSS BINAARI_LASKURI, 7 ;1
goto OHI_255

movlw b'00110000'
movwf ASCII_LASKURI_1
movwf ASCII_LASKURI_10
movwf ASCII_LASKURI_100
incf BINAARI_LASKURI

```

OHI\_255

```

BTFSC PORTA,3 ;alaspäin ruudulla
goto ALAS_OHI

INCF BINAARI_LASKURI

INCF ASCII_LASKURI_1

;jos on 1010 eli kymmenne lisätään seuraavaan digitiin ja nolataan 1 digitti
BTFSC ASCII_LASKURI_1, 0 ;0
goto ALAS_OHI
BTFSS ASCII_LASKURI_1, 1 ;1
goto ALAS_OHI
BTFSC ASCII_LASKURI_1, 2 ;0
goto ALAS_OHI
BTFSS ASCII_LASKURI_1, 3 ;1
goto ALAS_OHI

;lisätään 2 digittiin yksi
INCF ASCII_LASKURI_10
movlw b'00110000'
movwf ASCII_LASKURI_1

;jos on 1010 eli kymmenne lisätään 3 digitiin ja nolataan 2 digitti
BTFSC ASCII_LASKURI_10, 0 ;0
goto ALAS_OHI
BTFSS ASCII_LASKURI_10, 1 ;1
goto ALAS_OHI
BTFSC ASCII_LASKURI_10, 2 ;0
goto ALAS_OHI
BTFSS ASCII_LASKURI_10, 3 ;1

```

```

goto          ALAS_OHI

;lisätään 3 digittiin yksi
INCF          ASCII_LASKURI_100
movlw        b'00110000'
movwf        ASCII_LASKURI_10

ALAS_OHI

BTFSC        PORTA,4                ;ylöspäin ruudulla
goto         YLOS_OHI

DECF         BINAARI_LASKURI

DECF         ASCII_LASKURI_1

; eli jos lukema menee yhden alle 00110000 binääri luvun poistaa ohjelma 2 digitistä pois yhden ja laittaa 1 digittiin 9
; ja jos kakkos digitti on alle 0 niin poistaa 3 digitistä yhden ja laittaa 2 digittiin 9.

; 00110000 -0000 0001 = 00101111
;
; eli viimeiset bitit
BTFSC        ASCII_LASKURI_1, 4      ;0
goto         YLOS_OHI
BTFSS        ASCII_LASKURI_1, 5      ;1
goto         YLOS_OHI
BTFSC        ASCII_LASKURI_1, 6      ;0
goto         YLOS_OHI
BTFSC        ASCII_LASKURI_1, 7      ;0
goto         YLOS_OHI

;lisätään 2 digittiin yksi
DECF         ASCII_LASKURI_10
movlw        b'00111001'            ;9
movwf        ASCII_LASKURI_1

; viimeiset biti tarkistus
BTFSC        ASCII_LASKURI_1, 4      ;0
goto         YLOS_OHI
BTFSS        ASCII_LASKURI_1, 5      ;1
goto         YLOS_OHI
BTFSC        ASCII_LASKURI_1, 6      ;0
goto         YLOS_OHI
BTFSC        ASCII_LASKURI_1, 7      ;0
goto         YLOS_OHI

;lisätään 3 digittiin yksi
DECF         ASCII_LASKURI_100
movlw        b'00111001'            ;9
movwf        ASCII_LASKURI_10

YLOS_OHI

; Tulosta digiitit
movlw        0xFF
CALL         X_VIIVE500
;movlw        0xFF
;CALL         X_VIIVE500

CALL TOISELLE_RIVILLE

MOVF        ASCII_LASKURI_100,W
CALL        LCD_MERKKI_4BIT

MOVF        ASCII_LASKURI_10,W
CALL        LCD_MERKKI_4BIT

MOVF        ASCII_LASKURI_1,W
CALL        LCD_MERKKI_4BIT

MOVLW      A:'
CALL        LCD_MERKKI_4BIT

RETURN
;*****

```



```

;Tulostaa TKL logon 4bit LCD:lle
;*****
;numerot
;00110000 – 00111001 numerot 0-9 ASCII
;b'00100000'        väli

TKL_LOGO_4BIT
;TULOSTETAAN TKLegends 2006

    MOVLW    a'T'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'K'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'L'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'e'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'g'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'e'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'n'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'd'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a's'
    CALL     LCD_MERKKI_4BIT

    MOVLW    b'00100000'        ; Väilyönti
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'A'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'D'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'2'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'0'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'0'
    CALL     LCD_MERKKI_4BIT

    MOVLW    a'6'
    CALL     LCD_MERKKI_4BIT

RETURN

;*****

```

```

;Viive 500us
;*****
VIIVE500
    MOVLW    d'165'
    MOVWF    VIIVE
VIIVE500_LOOP
    DECFSZ   VIIVE,F
    GOTO     VIIVE500_LOOP
VIIVE500_END
    RETURN

;*****
;W-rekisteri kertaan 500us
;*****
X_VIIVE500
    MOVWF    VIIVEX
X_VIIVE500_LOOP
    CALL    VIIVE500
    DECFSZ  VIIVEX,F
    GOTO    X_VIIVE500_LOOP
X_DELAY500_END
    RETURN

;*****
;LCD:n alustus 4-bit pic 16f690 with Samsung KS0066U LCD 2x16
;*****
; PORTC d0-d3
; PORTC d4
; PORTC d5

LCD_ALUSTUS_4BIT
    CLRF    PORTC

    ; 30ms power viive
    MOVLW  0x078                ; 78 = 120
    CALL   X_VIIVE500           ; 120x500us=60ms
    ;0x3
    ;MOVLW  0x30
    MOVLW  b'00000010'
    CALL   LCD_CTRL

    ;0x3
    ;MOVLW  0x30
    MOVLW  b'00000010'
    CALL   LCD_CTRL

    ; N ja F 0000NF00 N=1 2line F=1 display on
    MOVLW  b'00001100'
    CALL   LCD_CTRL

    ; wait more than 39us
    MOVLW  0x02                ;noin 1000us
    CALL   X_VIIVE500

    MOVLW  b'00000000'
    CALL   LCD_CTRL

    ; 00001DCB
    MOVLW  b'00001111'
    CALL   LCD_CTRL

    ; wait more than 39us
    MOVLW  0x02                ;noin 1000us
    CALL   X_VIIVE500

    ; 0
    MOVLW  b'00000000'
    CALL   LCD_CTRL

    ; 1
    MOVLW  b'00000001'
    CALL   LCD_CTRL

    ; wait more than 1.53ms
    MOVLW  0x04                ;noin 2000us

```

```

CALL          X_VIIVE500      ; 4x500us = 2000us 2ms

; 1
MOVLW        b'00000000'
CALL         LCD_CTRL

; 000001(I/D)(SH)
MOVLW        b'00000110'
CALL         LCD_CTRL

; wait more than 1.53ms
MOVLW        0x20             ; noin 2000us
CALL         X_VIIVE500      ; 4x500us = 2000us 2ms

;ALUSTUS valmis
RETURN

;*****
;LCD:n ohjaus
;*****
LCD_CTRL
;BCF          PORTA, LCD_RW ;rw alas
BCF          PORTC, LCD_RS ;rs alas

ANDLW        0x0F             ; ylimmät bitit nolliksi
MOVWF        PORTC

CALL         LCD_ENABLE_DISABLE
;BSF          PORTC, LCD_E
;BCF          PORTC, LCD_E

CALL         VIIVE500

RETURN

;*****
;LCD:n ohjaus 4bit
;*****
LCD_CTRL4
;BCF          PORTA, LCD_RW ;rw alas
BCF          PORTC, LCD_RS ;rs alas

MOVWF        LCD_TEMP          ; talteen W

SWAPF        LCD_TEMP, W

ANDLW        0x0F             ; alimmat bitit nolliksi
MOVWF        PORTC            ; W->PORTC

CALL         LCD_ENABLE_DISABLE
;BSF          PORTC, LCD_E
;BCF          PORTC, LCD_E

;SWAPF        LCD_TEMP, W          ; Vaihdetaan ylimmit alemmiksi 0f -> f0 -> W
MOVF         LCD_TEMP, W
ANDLW        0x0F             ; alimmat bitit nolliksi
MOVWF        PORTC            ; W->PORTC

CALL         LCD_ENABLE_DISABLE
;BSF          PORTC, LCD_E
;BCF          PORTC, LCD_E

CALL         VIIVE500

RETURN

;*****

```

```

;LCD:n merkki 4bit
;*****
LCD_MERKKI_4BIT

        ;BCF                PORTA, LCD_RW ;rw alas

        MOVWF      LCD_TEMP          ; talteen W
        SWAPF     LCD_TEMP, W        ; Vaihetaan ylimmit alemmiksi 0f -> f0 -> W

        ANDLW     0x0F                ; ylimmät bitit nolliksi
        MOVWF     PORTC              ; W->PORTC

        BSF       PORTC, LCD_RS      ; RS ylös

        CALL      LCD_ENABLE_DISABLE
        BCF       PORTC, LCD_RS      ; RS alas

        MOVF     LCD_TEMP, W

        ANDLW     0x0F                ; ylimmät bitit nolliksi
        MOVWF     PORTC              ; W->PORTC

        BSF       PORTC, LCD_RS      ;rs ylös

        CALL      LCD_ENABLE_DISABLE

        BCF       PORTC, LCD_RS      ; RS alas

        CALL      VIIVE500

RETURN

;*****
; Vaihdaa lcd näytön rivin siirtyä osoitteeseen 40 eli rivin 2 alkuun
;*****
TOISELLE_RIVILLE
        ; 1AAA AAAA
        ; 40d -> 28h -> 0010 1000 b
        ; 1010 1000
        ; A8
        MOVLW     b'10101000'
        CALL      LCD_CTRL4

RETURN

;*****
; Vaihdaa lcd näytön rivin siirtyä osoitteeseen 0 eli rivin 1 alkuun
;*****
PALAA_ALKUUN
        MOVLW     b'10000000'
        CALL      LCD_CTRL4

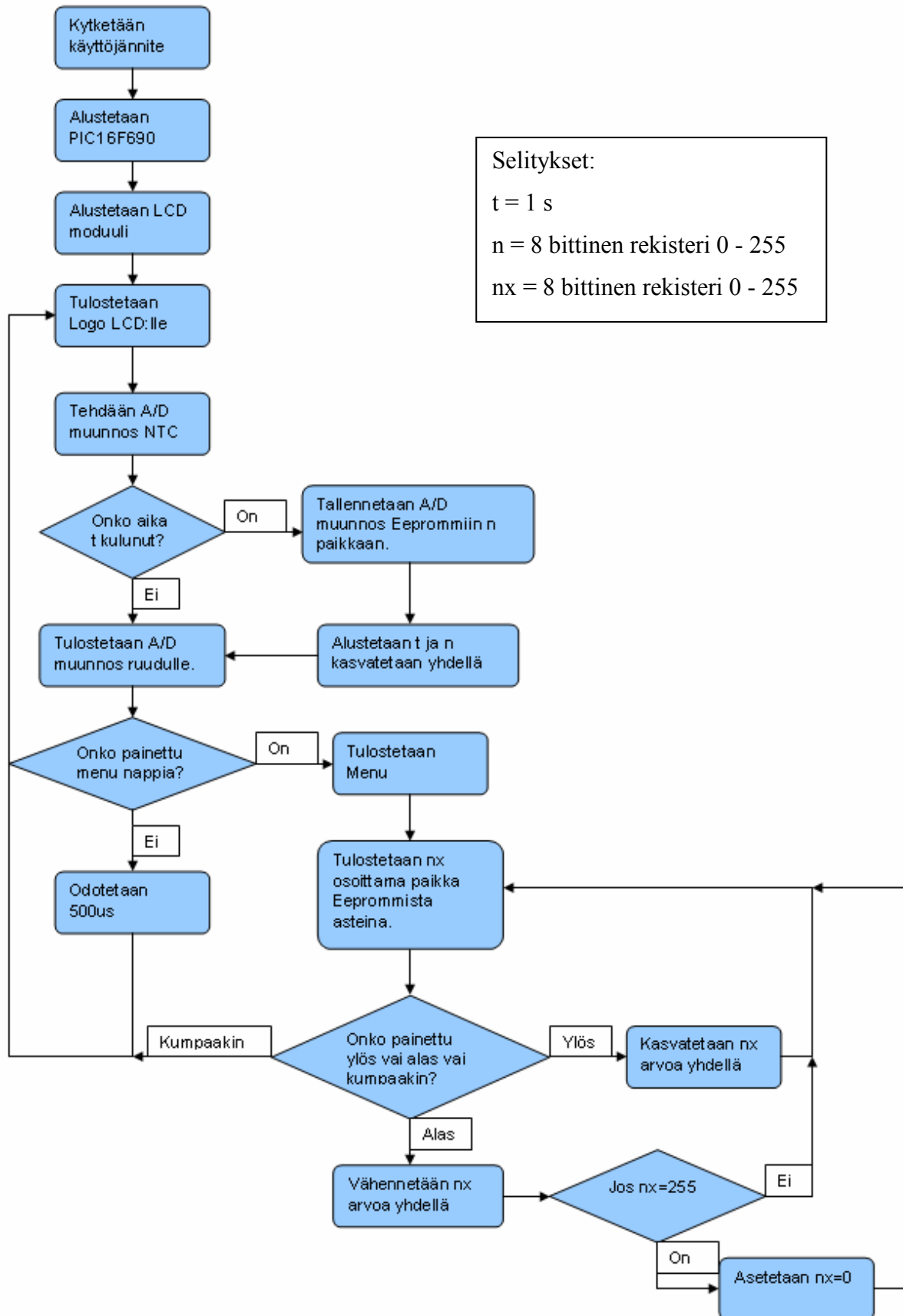
RETURN

;*****
; Käyttää E signaalia ylhäällä.
;*****
LCD_ENABLE_DISABLE
        BSF       PORTC, LCD_E
        BCF       PORTC, LCD_E

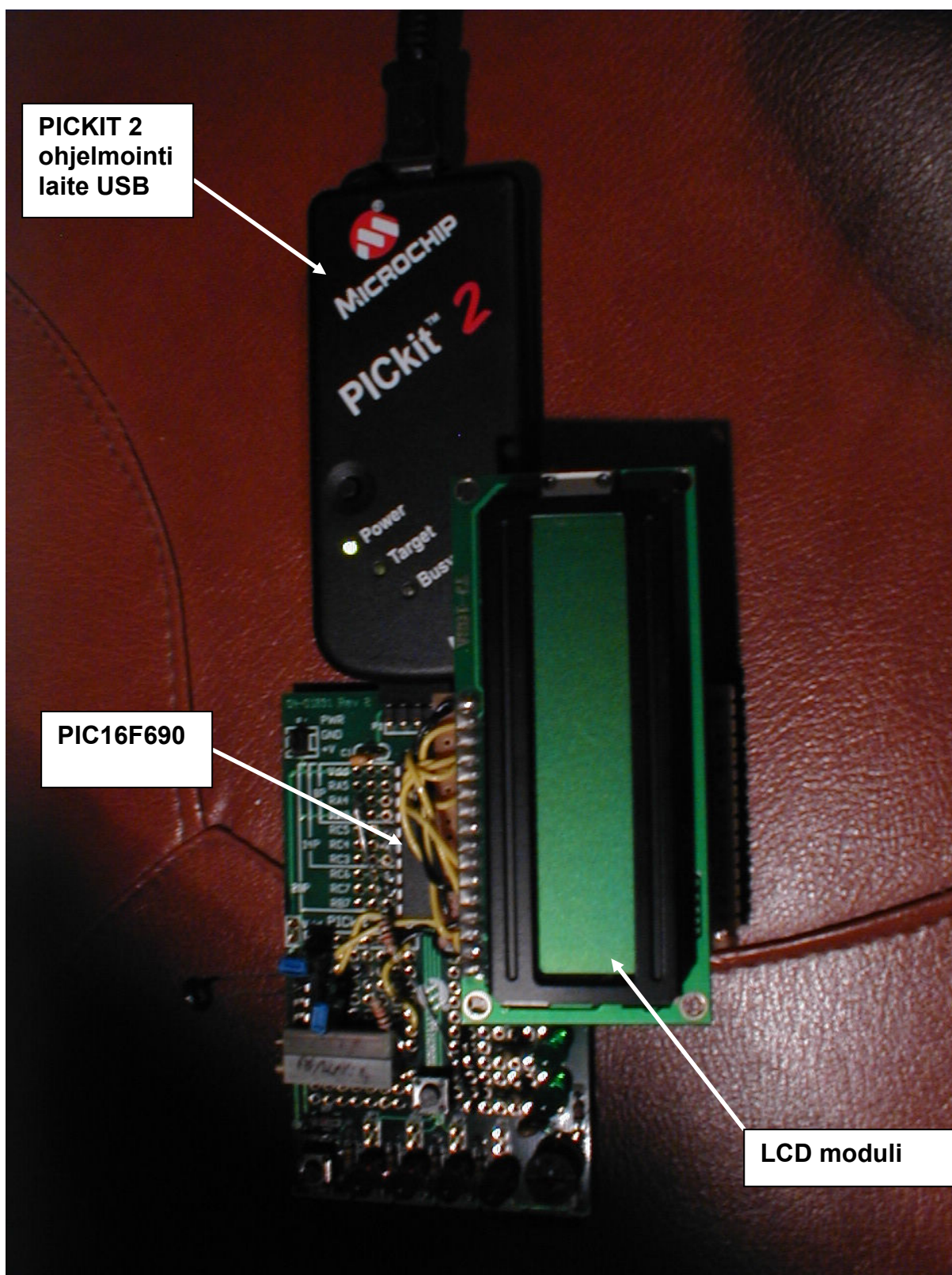
RETURN

; Koodin loppu.
END

```

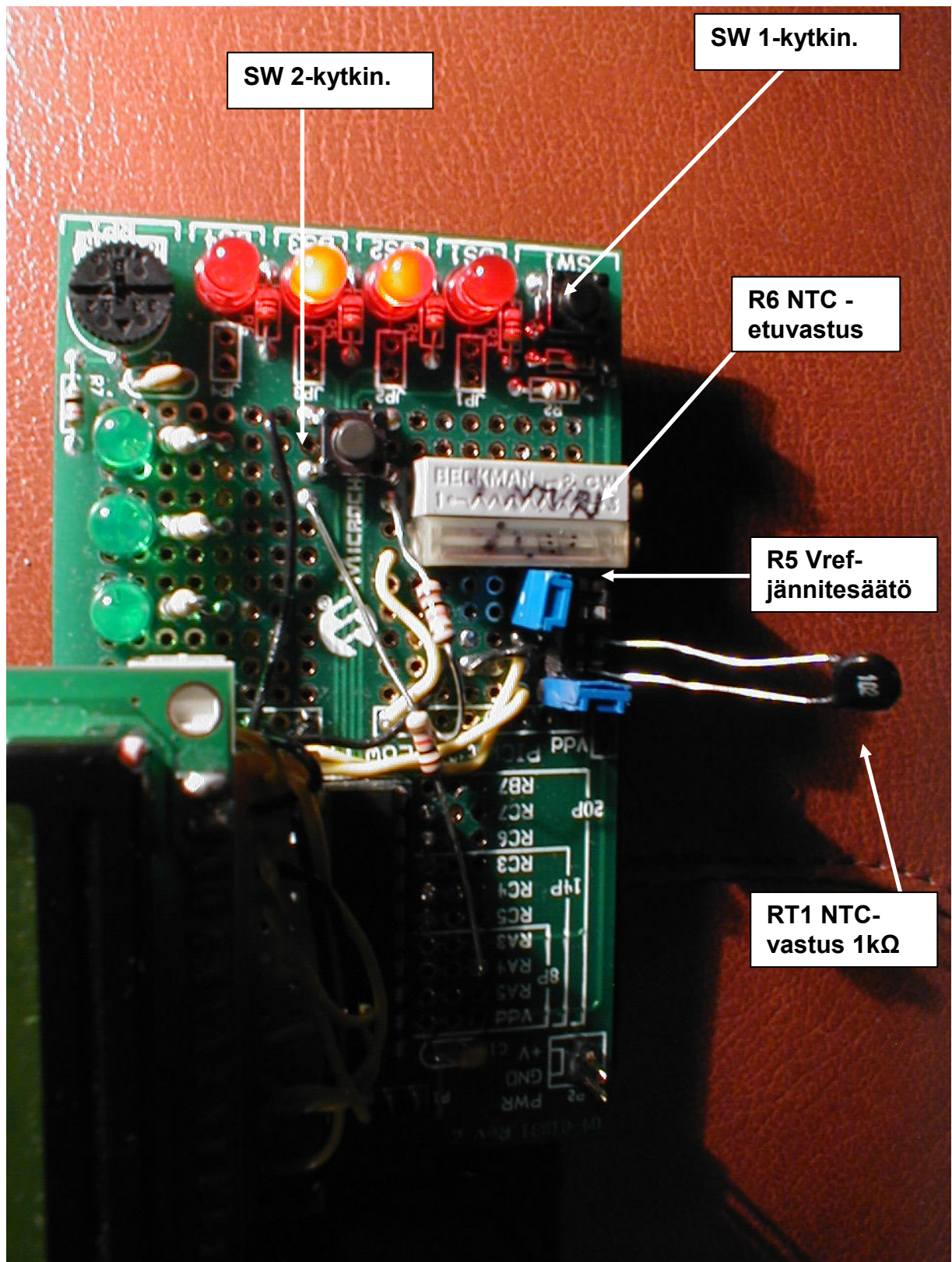


**Laite kokonaisuudessaan.**



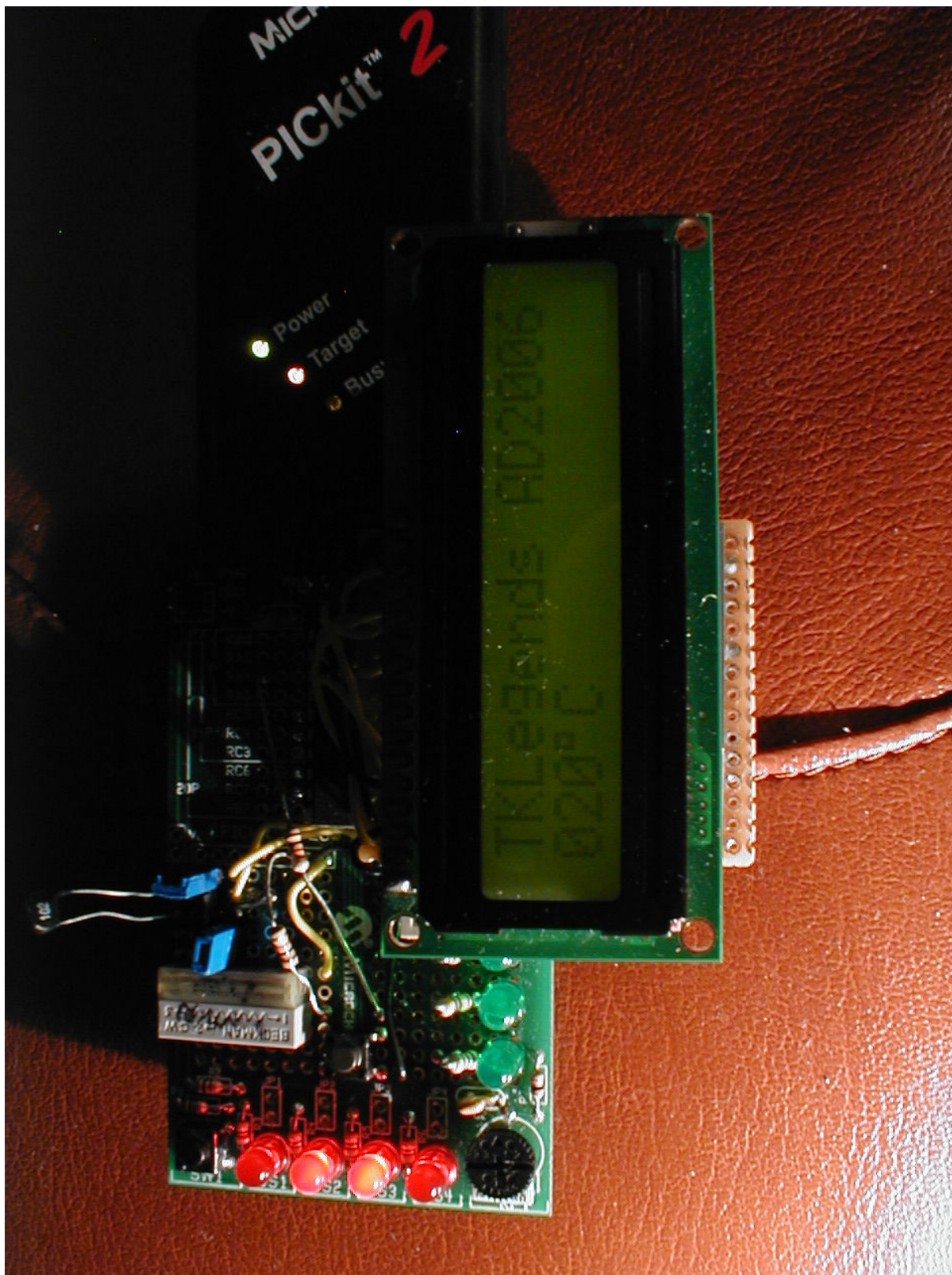


Laitteen NTC, Vref, NTC-etuvastus ja painonapit.





**Laite normaalitilassa.**





**Laite Menu tilassa.**

