

Timo Nurmesniemi  
TAAJUUSDISTRIBUUTORI

Insinööri  
Kajaanin ammattikorkeakoulu  
Tekniikan ja liikenteen ala  
Tietotekniikan koulutusohjelma  
Kevät 2004



**Kajaanin  
ammattikorkeakoulu**

**INSINÖÖRITYÖ  
TIIVISTELMÄ**

Osasto Tekniikka	Koulutusohjelma Tietotekniikka
Tekijä(t) Timo Nurmesniemi	
Työn nimi Taajuusdistribuuttori	
Vaihtoehtoiset ammattiopinnot Langaton tiedonsiirto	Ohjaaja(t) Heino Jukka
Aika 01.04.2004	Sivumäärä 31 + 31
Tiivistelmä <p>Insinööriyön aiheena oli suunnitella Kajaanin ammattikorkeakoululle radiolaboratorion taajuusstandardin taajuusosa. Laitteen tuli jakaa kideoskillaattorista saatua taajuutta halutuille laitteille. Lähdöt tuli olla dekadeittain 10 MHz – 1 Hz. Työssä tuli toteuttaa myös radiolaboratorioon tulevan taajuusstandardin valvontalaite, jolla tarkkaillaan kokonaisuuden toimintaa. Valvonnan voi toteuttaa ajannäyttölaitteella, jolle taajuusstandardista otetaan sekuntipulssi. Ajannäyttölaitteesta voidaan tarkkailla kelloaikaa ja päivämäärää. Kun järjestelmä toimii, pysyy kello oikeassa ajassa. Kelloaika ja päivämäärä asetetaan oikeaksi sarjaportin kautta ASCII-tekstiä tukevalla ohjelmistolla.</p> <p>Laitteet toteutettiin erikseen. Taajuusdistribuuttori toteutettiin veroboard-kortille käyttämällä pääasiassa synkronisia binäärilaskureita 74HC163. Taajuusdistribuuttorista saatavat neliöaallot ovat symmetrisiä ja tahdistetut samalla kellosignaalilla. Kellolaite näyttölaitteineen toteutettiin mikrokontrolleri MCI8051:n avulla kytkentäkortille. Ohjelmisto kellolaitteistolle toteutettiin C-kielellä ja näyttönä toimii yksirivinen LCD-näyttö.</p>	
Luottamuksellinen Kyllä Ei    X	
Hakusanat Taajuusdistribuuttori, jakaja, laskuri	
Säilytyspaikka Kajaanin ammattikorkeakoulun kirjasto	



**Kajaanin  
ammattikorkeakoulu**

**ABSTRACT  
FINAL YEAR PROJECT**

Faculty Faculty of Engineering	Degree programme Information Technology
Author(s) Timo Nurmesniemi	
Title A Frequency Distributor	
Optional professional studies Wireless Data Transmission	Instructor(s) / Supervisor(s) Heino Jukka
Date 01.04.2004	Total number of pages 31 + 31
Abstract  <p>The purpose of this final year project was to plan a part of the radio laboratory's frequency standard for Kajaani Polytechnic. The device was for dividing frequency which is from the crystal oscillator. The desired outputs are in the decades 10 MHz – 1 Hz. In addition the project included making a clock with a display. The time can be switched via RS232 which has the speed of 2400 bits / sec.</p> <p>The mainly devices were implemented individually. The frequency distributor was built on the veroboard card by using synchronic binary counters 74HC163. The square waves received from the frequency distributor are symmetrical and they are synchronized by the clock signal. The clock with its monitor was realized with a microcontroller MCI8051 on the connection card. The program code for the clock was carried out by the C language. The screen type is LCD.</p>	
Confidential Yes No <input checked="" type="checkbox"/>	
Keywords Frequency distributor, counter	
Deposited at Kajaani Polytechnic library	

## SISÄLLYSLUETTELO

1 JOHDANTO .....	5
2 TAAJUUSDISTRIBUUTTORI.....	6
2.1 Esiselvitys.....	6
2.1.1 Bistabiilit kiikkupiirit .....	6
2.1.2 Fanout.....	8
2.1.3 Laskuripiirin valinta .....	11
2.2 Dekadi-jakajan suunnittelu .....	13
2.3 Dekadi-jakajakytkennän toiminta.....	14
3 KELLO NÄYTTÖLAITTEINEEN.....	17
3.1 Yleistä.....	17
3.2 Kytkennän suunnittelu .....	19
3.2.1 Osoitekoodaus .....	19
3.2.1 Testaamiseen 7-segmentinäyttö.....	20
3.2.2 LCD-näyttö.....	21
3.2.3 Sarjaportti .....	22
4 OHJELMAKOODI .....	23
5 KÄYTTÖOHJE .....	28
6 YHTEENVETO.....	29
LÄHDELUETTELO .....	30
LIITTEET	

## 1 JOHDANTO

Työn aihe on saatu Kajaanin ammattikorkeakoululta. Työn taustalla on tarkoitus rakentaa Kajaanin ammattikorkeakoulun radiolaboratorioon taajuusstandardi. Taajuusstandardikokonaisuus tulee muodostumaan useasta osasta, jotka on jaettu useaan insinööriyön aiheeseen. Taajuus tullaan ottamaan todennäköisesti kideoskillaattorista, joka stabiloidaan ja synkronoidaan GPS-satelliittivastaanottimesta saatavan sekuntipulssin avulla. Tarkan taajuuden ympärille rakennetaan muita ominaisuuksia ja tämän insinööriyön aiheena on tehdä osa tästä radiolaboration taajuusstandardista.

Taajuusdistribuutori on laitteisto, jolla kideoskillaattorista saatua taajuutta sekä sekuntimerkkiä voidaan jakaa halutuille laitteille. Lähdöt ovat dekadeittain 10 MHz – 1 Hz. Distribuutori on oleellisesti ottaen vahvistin, jolla vastaanottavaa laitetta pystytään ohjaamaan yli 200 m pitkällä koaksikaapelilla.

Työssä tuli toteuttaa myös radiolaboratorioon tulevan taajuusstandardin valvontalaite, jolla tarkkaillaan kokonaisuuden toimintaa. Valvonnan voi toteuttaa ajannäyttölaitteella, jolle taajuusstandardista otetaan sekuntipulssi. LCD-näytölle toteutettavasta ajannäyttölaitteesta voidaan tarkkailla kelloaikaa ja päivämäärää. Jos järjestelmä toimii, pysyy kello oikeassa ajassa. Kelloaika ja päivämäärä asetetaan oikeaksi sarjaportin kautta ASCII-tekstiä tukevalla ohjelmistolla. Tällainen ohjelma on esimerkiksi Hyper Terminal.

## 2 TAAJUUSDISTRIBUUTORI

### 2.1 Esiselvitys

Tehtävän aloittamiseksi tuli selvittää, miten pystyy jakamaan tarkasti taajuuksia dekaideittain. Tämän vuoksi kannatti tutkia peruskiikkujen toimintaa ja edetä laskureihin. Dekadi-jakaja olisi mahdollista tehdä myös ohjelmallisesti, mutta tässä työssä päädyttiin tehtävän suorittamiseen mikropiireillä.

#### 2.1.1 Bistabiilit kiikkupiirit

Bistabiilien kiikkupiirien lähteenä käytettiin Vesa Volotisen kirjaa Digitaalitekniikka [1].

Yhden bitin muistipiireinä voidaan käyttää bistabiileja kiikkuja eli bistabiileja multivibraattoreita (Bistable flip-flop). Näiden avulla voidaan tulosignaalien ohjauksella asettaa kiikku tilaan 1 ja kiikku pitää tilan, kunnes se nollataan uudella ohjauksella. Bistabiilit kiikut jaetaan synkronisiin (Synchronous) ja asynkronisiin (Asynchronous) kiikkupiireihin. Synkroniset kiikkupiirit sisältävät kellotulon, jonka signaalin tahdissa kiikku toimii. Asynkroniset kiikut toimivat pelkästään ohjaustulojen mukaan ja niitä kutsutaankin salpapiireiksi (Latch). Bistabiilit kiikut jaetaan ohjausominaisuuksien mukaan neljään perustyyppiin: RS-, D-, JK- ja T-kiikku.

#### RS-kiikku

RS-kiikku (RS flip-flop) on kaikkien bistabiilien kiikkupiirien asynkroninen peruskiikku. RS-kiikulla on kaksi staattista pakkotuloa, R (Reset) ja S (Set), sekä lähtöinä Q ja sama invertoituna eli  $\bar{Q}$ . RS-kiikku muistaa, kumpi tulo on ollut viimeksi aktiivisessa tilassa. Kun alkutilanteessa kiikku nollataan R-tulolla, on S ja  $\bar{Q}$  0-tilassa sekä Q tilassa 1. Jos kumpikin R- sekä S-tulo laitetaan aktiiviseksi eli 1-tilaan, tulee mahdoton tilanne, koska tällöin Q ja  $\bar{Q}$  pyrkii nousemaan 1-tilaan. Tällöin kiikku joutuu metastabiiliin tilaan, jossa kiikun seuraavaa tilaa ei voida ennustaa. Tätä kutsutaan kielletyksi tilaksi. RS-kiikkuja

ei yleensä ole saatavana valmiina piirinä, vaan sen toiminta toteutetaan esimerkiksi porttipiirien avulla.

### D-kiikku

D-kiikkuä käytetään mm. linjalähtettiminä ja -vastaanottimina. D-kiikulla on valmistetulo  $D$  ja kellotulo  $CP$ . Lisäksi esimerkiksi D-kiikulla 74LS74 on asynkroniset pakkotulot  $S_D$  ja  $C_D$ , joilla voidaan kiikku asettaa ja nollata kello-tulosta riippumatta, eli kiikku toimii asynkronisena RS-kiikkuna. Synkronisesti toimivan D-kiikun lähtö asettuu kellotulon nousevalla, tai kiikusta riippuen laskevallareunalla D-tuloa vastaavaan tilaan.

### JK-kiikku

JK-kiikku on parannettu versio RS-kiikusta. JK-kiikuilla ei ole kiellettyä tilaa, joten ne ovat hyvin monipuolisia ja paljon käytettyjä kiikkutyyppejä. JK-kiikuilla on staattiset valmiste-tulot  $J$  ja  $K$  sekä dynaaminen kellotulo  $CP$ . Lisäksi yleensä on kaksi pakkotuloa, kuten D-kiikussa. JK-kiikku voi toimia synkronisesti tai pakkotulojen avulla asynkronisesti. Sekvenssiipiirien ns. herätetaulukoissa käytetään JK-kiikun lähdöstä merkintää  $Q_N$ , joka tarkoittaa lähdön tilaa ennen kellopulssin tuloa ja  $Q_{N+1}$  joka tarkoittaa kellopulssin jälkeen olevaa lähdön tilaa. JK-kiikkuja on useita sarjoja, esimerkiksi mikropiiri 74HC107 sisältää kaksi reunaliiipaistavaa (Edge-triggered) JK-kiikkuä.

### T-kiikku

T-kiikku vaihtaa lähdön tilaa jokaisella kellotulon  $CP$  aktiivisella muutoksella. T-kiikku saadaan JK-kiikusta kytkemällä  $J$ - ja  $K$ -tulot aktiiviseksi eli 1 tilaan. Käytännössä T-kiikku jakaa kellotulon taajuuden kahdella.

Taulukossa 1 on peruskiikkutyyppien totuustaulut. Taulukossa olevat x merkit tarkoittavat kiellettyä tilaa.

Taulukko 1. Peruskiikkutyypin totuustaulut

RS-kiikku				JK-kiikku				D-kiikku			T-kiikku				
tulot		lähdöt		tulot		lähdöt		tulot		lähdöt		tulot		lähdöt	
R	S	Q	$\bar{Q}$	J	K	Q	$\bar{Q}$	D	Q	$\bar{Q}$	T	Q	$\bar{Q}$		
0	0	Q	$\bar{Q}$	0	0	Q	$\bar{Q}$	0	0	1	0	Q	$\bar{Q}$		
0	1	1	0	0	1	0	1	1	1	0	1	$\bar{Q}$	Q		
1	0	0	1	1	0	1	0								
1	1	x	x	1	1	$\bar{Q}$	Q								

### Laskurit

Laskurirekisterit eli laskurit ovat yleisiä digitaalitekniikan sekvenssiipiirejä. Laskurit rakennetaan yhteen kytketyistä kiikkupiireistä. Kaikille laskureille on yleistä, että yleensä laskenta tapahtuu binaariluvuilla. Laskuria ohjataan kello- tai muiden tulopulssien avulla. Kun kello- tai tulopulssi kasvattaa lukemaa, on kyseessä ylöspäin laskeva laskuri (Up counter), ja tulopulssin vähentäessä lukemaa, on kyseessä alaspäin laskeva laskuri (Down counter). Kun laskurissa on merkintä Modulo-N-laskuri, laskee kyseinen laskuri tulevia pulsseja lukuun N saakka, jolloin laskuri nollautuu ja aloittaa laskemisen alusta. [1.]

### 2.1.2 Fanout

#### Staattinen fanout

Koska kellosignaali joudutaan jakamaan useille piireille, on syytä laskea, jaksako signaalia syöttävä mikropiiri 74HC04 syöttää signaalin kaikille tarvitseville piireille. Jos kaikki mikropiirit kuuluisivat samaan piiriperheeseen, voitaisiin laskea staattinen fanout-luku, joka kuvaa mikropiirin lähdonkykyä imeä virtaa 0-tilassa ja syöttää virtaa 1-tilassa. Fanout-luku, jolla on myös nimitys antoisuus, voidaan laskea seuraavasti:

$$0\text{-tila: fanout} = \frac{I_{OL}}{I_{IL}},$$

jossa  $I_{OL}$  on syöttävän piirin syöttämä virta ja  $I_{IL}$  on vastaanottavan piirin ottama virta.



$$1\text{-tila: fanout} = \frac{I_{OH}}{I_{IH}},$$

jossa  $I_{OH}$  on syöttävän piirin syöttämä virta ja  $I_{IH}$  on vastaanottavan piirin ottama virta. Fanout-luvuista valitaan pienempi.

Käytettävän piirin datakirjasta löytyy tarvittavat tiedot, joilla fanout-luku voidaan laskea. Saatavista fanout-luvuista pienempi tarkoittaa suurinta määrää samaan piiriperheeseen kuuluvia mikropiirejä, joita piiri jaksaa syöttää. Käytännössä HCMOS-tekniikkaan perustuvissa HC-, HCT-, HCU-, AC- ja ACT-piiriperheissä ei tarvitse laskea fanout-lukua, koska esimerkiksi piirille 74HC04 datakirjan [2] mukaan piirin kuluttama virta on symmetrinen  $\pm 1 \mu\text{A}$  ja piirin syöttämä virta  $\pm 4 \text{ mA}$ . Fanout-luvuksi saadaan laskettua 4000. Tämä tarkoittaa, että samaan 74HC-piiriperheeseen kuuluvia piirejä voi syöttää yhdellä 74HC04-piirillä 4000 kpl.

Työssä käytetään myös 74LS244-piiriä, jonka suurin kuluttama virta on 0-tilassa 0,2 mA ja 1-tilassa 0,1 mA. Tästä voidaan päätellä, että piiri 74HC04 jaksaa syöttää koko kytkennän kellotuloille riittävästi virtaa. [3, 4 s 53-54]

### Dynaaminen fanout

Todellisuudessa piirien väliset johtimet aiheuttavat kapasitanssia arviolta 50...100 pF/10 cm. Käytettävästä jännitteestä ja mikropiirin ottamasta virrasta syntyy Ohmin lain mukaan vastus R. Vastus R ja kapasitanssi C muodostavat RC-piirin, joka vaikuttaa pulssin nousu-aikaan.

Pulssin nousuajan likiarvoon on approksimaatiokaava  $2,2 \cdot RC$ . Tämä on lähellä oikeaa arvoa, joka saadaan laskemalla RC-piirin aiheuttama hidastuminen nousu-aikaan seuraavasti. Ratkaistaan RC-piirin aika t,

$$U = U_0(1 - e^{-t/\tau}),$$

jossa aikavakio  $\tau = RC$

ja josta

$$t = \ln \frac{U}{U_0} * RC \quad (1)$$

Johtimien aiheuttamiksi kapasitansseiksi arvioitiin 200 pF. Vastus R saadaan Ohmin lain mukaan siten, että 74HC04:n lähtövirta on 4 mA ja tästä Ri on 1,25 kΩ. Kun kaavaan 1 sijoitetaan 90 % jännite ja tästä vähennetään kaavaan 2 sijoitettu 10 % jännite, saadaan pulssin nousuaika seuraavasti:

$$t = \left( \ln \frac{4,5 \text{ V}}{5 \text{ V}} - \ln \frac{0,5 \text{ V}}{5 \text{ V}} \right) * 1250 \Omega * 200 \text{ pF} = 0,55 \mu\text{s} \quad (2)$$

Laskettua signaalin nousunopeutta verrataan piirin 74HC163 datakirjasta löytyvään suurimpaan kellosignaalin nopeuteen  $T_{r_{max}}$ . Kaavalla saatu aika tulee olla hitaampi kuin, mitä piirin 74HC163 aika  $T_{r_{max}}$ . Datakirjaan [6] ei ollut merkitty aikaa  $T_{r_{max}}$ , vaan aika jouduttiin laskemaan jakamalla  $f_{max}$  kolmella ja muuttamalla taajuus ajaksi. Taajuus jaetaan kolmella, koska taajuus lasketaan pulssista seuraavan pulssin samaan kohtaan. Staattinen fanout tarkastelee vain yhtä nousuaikaa ja taajuudessa nousuaikoja on kolme. Piirin-74HC163 datakirjan arvoista laskettu nousuaika on 1,3 μs. Tämä on aika, jonka piirin-kellosignaali tarvitsee vähintään toimiakseen. Kaavalla 2 laskettu signaalin nopeus on 0,55 μs, joten voidaan todeta että piireille syötettävä kellosignaalin nopeus on riittävä.

### 2.1.3 Laskuripiirin valinta

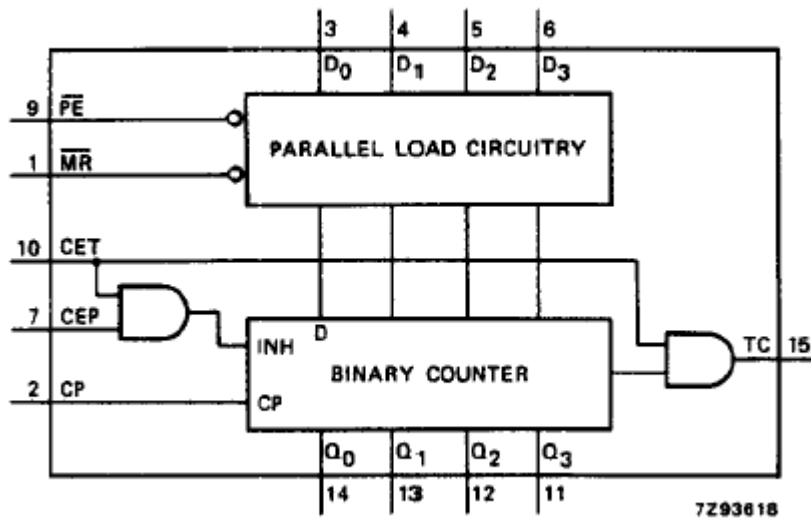
Taulukossa 2 on laskuripiirin valinnan kannalta joitakin tärkeimpiä ominaisuuksia 74-sarjan 4-bittisistä synkronisista laskuripiireistä.

*Taulukko 2. Laskuripiirien ominaisuuksia [5 s 117]*

74...	Sekvenssi	Suunta	Nollaus	Lataus	Muuta
160	BCD	ylös	asynkr.	synkr.	
161	Binaari	ylös	asynkr.	synkr.	
162	BCD	ylös	synkr.	synkr.	
163	Binaari	ylös	synkr.	synkr.	
190	BCD	ylös/alas	ei ole	asynkr.	yksi kellotulo, suunta valitaan
191	Binaari	ylös/alas	ei ole	asynkr.	$\bar{U}$ /D-ohjaustuloilla
192	BCD	ylös/alas	asynkr.	asynkr.	kaksi kellotuloa, kummallekin
193	Binaari	ylös/alas	asynkr.	asynkr.	laskentasuunnalle omansa

Taulukon 1 perusteella sopivin ratkaisu tehtävään olisi kymmeneen laskeva BCD-laskuri 74xx162, mutta koska piiriä ei ollut koulun valikoimassa, päädyttiin binaarilaskuriin 74HC163.

Kuvassa 1 on 4-bittisen 74HC163 sisäinen toimintaperiaate. Laskuri laskee 0...15 ( $0000_2 \dots 1111_2$ ), jolloin lähdöstä TC tulee pulssi ja laskuri aloittaa laskemisen alusta. Laskuri toimii synkronisesti täysin kellotulon mukaan. Seuraava tapahtuma tapahtuu aina seuraavalla kellosignaalin nousevalla reunalla. Tämän avulla voidaan koko kytkennässä käyttää samaa kellotusta eikä asynkronisessa laskurissa olevaa peräkkäisistä piireistä syntyvää viivettä esiinny.



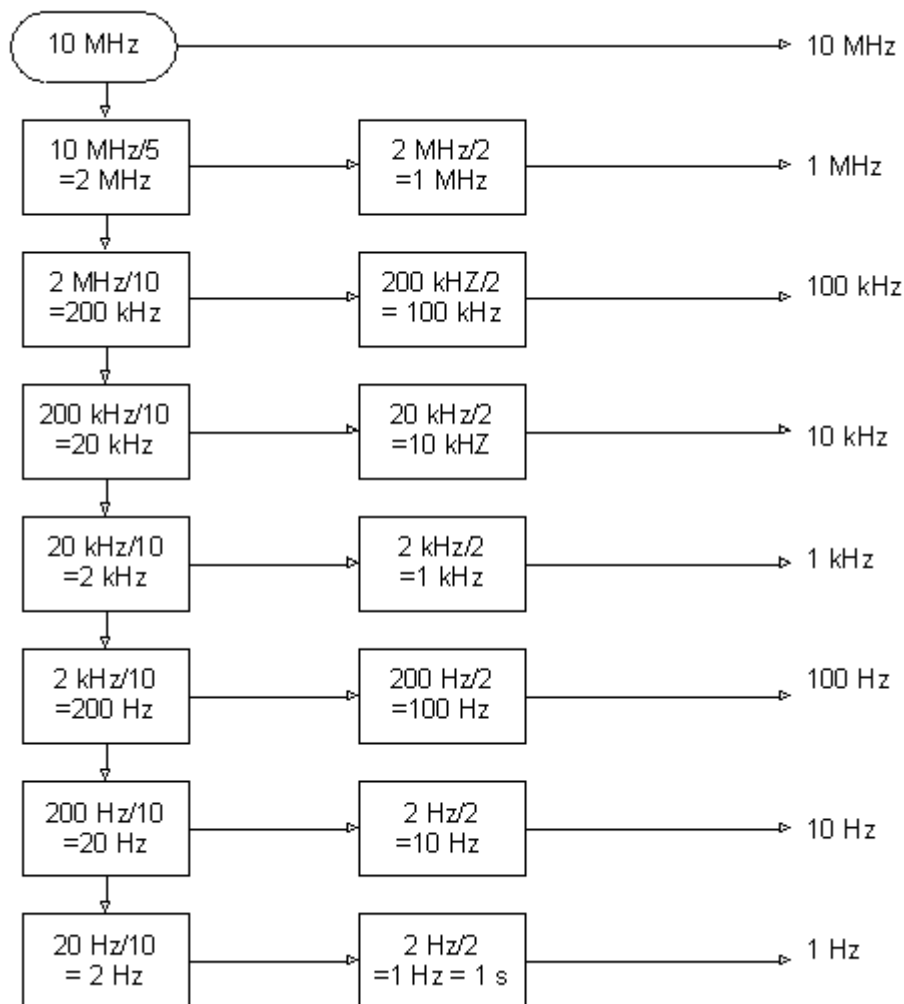
Kuva 1. Binaarilaskurin 74HC163 sisäinen toiminta [6]

Laskuri tulkitaan myös taajuuden jakajana. Lähtö  $Q_0$  jakaa kellotuloon syötetyn taajuuden kahdella, lähtö  $Q_1$  neljällä, lähtö  $Q_2$  kahdeksalla ja lähtö  $Q_3$  jakaa kellotuloon syötetyn taajuuden 16. Kun 16 jakajan tiloja vähennetään, ei lähdöistä  $Q_0 \dots Q_3$  saa enää kanttiaaltoja joka on yhtä kauan sekä H että L tilassa kuin osassa tilanteista. Liitteessä A on esitetty aikakaaviona lähtöjen  $Q_0 \dots Q_3$  -tilat kun käytössä normaalit 16 eri tilaa ja muutettuna kymmenen eri tilaa sisältäväksi.

Laskurin nollaus ja uuden luvun asettaminen voidaan tehdä esimerkiksi lisäämällä NAND-piiri tutkimaan lähdön tiloja, ehdon täytyessä NAND-piiri antaa pulssin laskurin nollaukseen, sekä uuden luvun asettamiseen. Koska laskuripiiri 74xx163 antaa ylivuotobitin eli lyhyen pulssin laskurin pyörähtäessä ympäri, kannattaa tätä käyttää tehtävässä hyödyksi. Tätä lyhyttä pulssia pystyy käyttämään invertoituna uuden luvun asettamiseen laskurissa. Luku kytketään tuloihin  $D_0 \dots D_4$  kytkemällä kukin tulo haluttuun tilaan. Binaariluku asetetaan laskuriin  $\overline{PE}$ -tuloon syötetyllä nollapulssilla, jolloin laskuri aloittaa laskemisen tuloihin  $D_0 \dots D_4$  kytketystä binaarisesta luvusta. CET- ja CEP-tulot toimivat sallintalinjoina. Kumpikin CET ja CEP tulee olla 1-tilassa, jotta laskuri aloittaa laskemisen alusta. Näiden avulla voidaan kytkeä useita laskureita peräkkäin.

## 2.2 Dekadi-jakajan suunnittelu

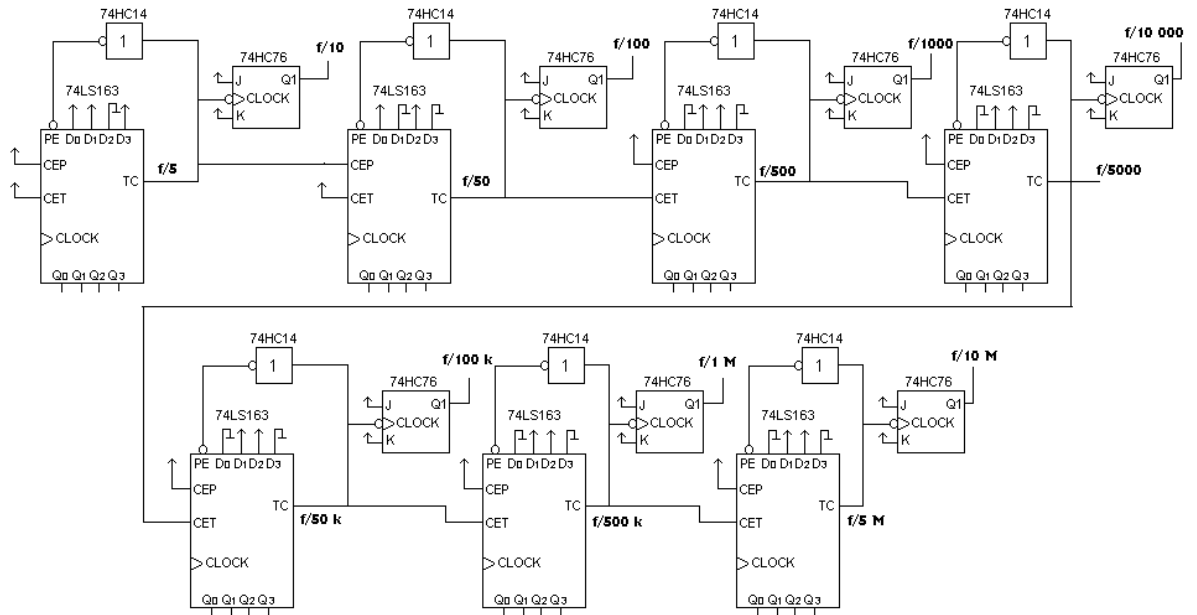
Koska tavoitteena oli saada kanttiaallosta symmetristä, niin jouduttiin kytkentä suunnittelemaan siten, että piirin 74HC163 lähdöistä  $Q_0...Q_3$  saatavia valmiita pulsseja ei käytetty. Ratkaisu ongelmaan löytyi tutkimalla kiikkupiirien toimintaa ja havaittiin, että T-kiikku vaihtaa lähdön tilaa jokaisella kellotulon CP aktiivisella muutoksella. Tällöin kiikku jakaa kellotulotaajuuden kahdella. T-kiikku saadaan JK-kiikusta kytkemällä J- ja K-tulot aktiiviseksi. Kuvassa 2 on näiden tietojen pohjalta suunniteltu vuokaavio taajuuden jakamisen periaatteesta.



Kuva 2. Vuokaaviokuvaus jakajakytkennän periaatteesta

### 2.3 Dekadi-jakajakytkennän toiminta

Kuvassa 3 on kuvan 2 vuokaavion pohjalta suunniteltu taajuuden jakamiseen käytettävä kytkentä. Tarkempi kytkentäkaavio on liitteessä B.



Kuva 3. Suunniteltu dekadijakajakytkentä

Kytkentään tuodaan 10 MHz sini- tai kantti- eli neliöaalto-signaali. Signaali tuodaan kahden invertterin läpi, jolloin syntyy 10 MHz kanttiaalto, joka on samassa vaiheessa, kuin kytkentään tuotava signaali. Käytännössä ensimmäisen invertterin tehtävänä on myös toimia suojana kytkennälle.

Ensimmäisenä invertterinä voisi käyttää 74HC14 Schmitt-trigger-invertteriä, jolloin saadaan soudatettua häiriösignaalien aiheuttamat häiriöpiikit pois. Tehtävässä käytetään kuitenkin 74HC04-invertteriä. Ensimmäinen invertterin yli laitetaan 22 kΩ vastus. Vastus muuttaa invertterin toimintapistettä, jolloin kytkentään tulevan signaalin muutokset voivat olla heikompia kuin muuten. Invertteristä tulee tavallaan pieni vahvistin.

Toisen invertterin antamaa signaalia käytetään kellosignaalinä muille piireille sekä 10 MHz lähtösignaalina. Laskuripiirit 74HC163 kytketään jakamaan signaalia kuvan 2 kaavion mukaisesti. Kuvassa 10 MHz taajuuden jakaminen

toteutetaan aluksi jakamalla taajuus viidellä. Jakaminen toteutetaan piirillä 74HC163. Piirin TC-lähdöstä saatava lyhyt pulssi kytketään invertoituna takaisin  $\overline{PE}$ -tuloon. Lisäksi TC-lähtö kytketään suoraan T-kiikun kellosignaaliaksi ja seuraavan jakajan CEP tai CET valintalinjaan. T-kiikku vaihtaa antaman signaalin tilaa jokaisella TC-lähdön antamalla lyhyellä pulssilla, jolloin taajuus jaetaan kahdella ja T-kiikusta saadaan ulos symmetristä kanttiaaltoa.

Valintalinjoihin CEP tai CET kytketty signaali sallii laskurin edetä vain edellisen laskurin TC-lähdöstä saatavan signaalin tahdissa. Lisäksi edellistä signaalia nopeampi kellosignaali antaa tahdin nousevilla reunoilla kaikille laskuripiirin tapahtumille. Vastaavasti seuraavat laskurit kytketään ketjuksi, jolloin saadaan signaali jaettua oikein ja T-kiikkujen avulla saadaan ulos symmetrinen signaali dekaideittain.

Koska muodostettujen signaalien oli tavoitteena kulkea yli 200 m koaksiaali-kaapelissa, tuli kytkentään saada piiri, joka vahvistaisi riittävästi lähetettävää signaalia. Työn ohjaaja ehdotti tähän ratkaisuksi linjaohjaimeksi tarkoitettua piiriä 74LS244, jota tehtävässä myös käytettiin.

Häiriön poistoon kytkennässä käytettiin jokaisen piirin käyttöjännitteen ja maanvälille keraamisia bypass-kondensaattoreita. Lisäksi häiriöiltä pyrittiin suojautumaan tantaalikondensaattorin avulla ja kytkemällä kaikki vapaana oleva piirilevyn pinta-ala maapotentiaaliin. Näistä toimenpiteistä huolimatta kytkentään jäi runsaasti pientä häiriösignaalia, joka näkyy saatavien signaalien pienenä epätasaisuutena 0- tai 1-tilassa. Häiriö on kuitenkin niin pieni, että siitä ei ole haittaa toiminnalle.

### 2.3 Taajuusdistribuuttorin toteutus

Piirin 74HC163 kytkemisessä havaittiin ongelma erilaisten merkintätapojen vuoksi. Taulukossa 3 on eri merkintäjärjestelmien vastaavuudet.

*Taulukko 3. Merkintöjen vastaavuudet laskurin datakirjoissa*

Eurooppalainen merkintä	Amerikkalainen merkintä
CEP	EN P
CET	EN T
CP	CK
TC	RCO
PE	LD
MR	CLR
Q0...Q3	QA..QD
D0...D3	A...D

KytKentä rakennettiin aluksi osittain koekytKentäalustalle, jossa toimivuutta testattiin. Kun kytkennän periaatteellinen toiminta oli selvillä, juotettiin piirilevyille vastaavanlainen kytkentä. Valmiin kytkennän kuva on liitteessä C.

KytKentä alkaa toimia, kun käyttöjännitteeksi laitetaan 5 V ja kytkentään tulevan 10 MHz taajuuden amplitudi ja offset jännite asetetaan sopivaksi. Offset-jännite kokeilussa oli 1,3 V ja amplitudi 650 mVPP. Taulukossa 4 on kytkennän liittimen johdinjärjestys.

*Taulukko 4. Liittimen johdinjärjestys*

Liitin	KytKetty	Liitin	KytKetty	Liitin	KytKetty	Liitin	KytKetty
1	+5 V	9	GND	17	GND	25	GND
2		10		18	1 MHz	26	100 Hz
3	GND	11	GND	19	GND	27	GND
4	Reset	12	$f_{in}(10 \text{ MHz})$	20	100 kHz	28	10 Hz
5	GND	13	GND	21	GND	29	GND
6		14		22	10 kHz	30	1 Hz
7	GND	15	GND	23	GND	31	GND
8		16	10 MHz	24	1 kHz	32	GND



## 3 KELLO NÄYTTÖLAITTEINEEN

### 3.1 Yleistä

Proessorikortin sisältämän mikrokontrollerin ohjelmointi on sulautettujen järjestelmien kehitysympäristö. Mikrokontrolleri on yleensä yksinkertainen pienoistietokone, joka sisältää kaikki tarvittavat apupiirit ja komponentit integroituna yhdelle piirille.

MCS 8051 on alun perin Intelin suunnitteleman mikrokontrolleriperheen peruspiiri. Se sisältää mm. 8-bittisen keskusyksikön, 64 kt:n muistiavaruuden ohjelmamuistille, sekä 64 kt:n muistiavaruuden datamuistille. Liikenne kortin ulkopuolelle tapahtuu 32 kaksisuuntaisen ja yksittäin osoitettavan I/O-linjojen avulla, jotka on ryhmitelty neljäksi 8 bitin porteiksi (P0, P1, P2 ja P3), sekä kaksisuuntaisen sarjaportin avulla. [7]

#### Osoitedekoodaus

Proessori voi keskustella vain yhden dataväylään liitetyn piirin kanssa kerrallaan. Tämän vuoksi jokaiseen dataväylään liitetulle piirille on tehtävä oma piirinvalintasiignaali, joka on aktiivinen omassa osoitevälissä. Piirinvalintasiignaali voidaan toteuttaa esim. 74HC138-piirin avulla. Piirin ylimmillä (MSB) osoitelinjoilla tehdään piirien valinta, ja alimmilla (LSB) osoitelinjoilla valitaan muistipaikka valitun piirin sisältä. Proessorin osoiteväylän leveys määrää, kuinka monta erilaista osoitetta se pystyy osoittamaan. [8]

Lineaarista piirinvalintaa käytetään yksinkertaisissa sovelluksissa, koska sen komponenttikytkentä on hyvin yksinkertainen. Linearisessa piirinvalinnassa osoiteväylän ylimpiä linjoja käytetään suoraan piirin valintalinjoina ja alimmat osoitelinjat voidaan viedä suoraan tai invertoituina valittavien piirien  $\overline{CS}$ -nastoihin. Tästä seuraa, että muistiavaruus saadaan huonosti käytettyä, koska suurella osalla osoitteista tulee monta piiriä samanaikaisesti valittua. Invertterien avulla lineaarisella piirinvalinnalla saadaan yksi liitettävistä piireistä sijoitettua haluttuun paikkaan muistiavaruudessa, mutta muiden piirien osoitteet muistiavaruudessa menevät mihin sattuu.

Dekoodatussa piirinvalinnassa dekodataan varsinaiset piirinvalintalinjat, jolloin muistialueet saadaan peräkkäin eli prosessorin koko muistiavaruus saadaan hyödynnettyä. Piirillä 74HC138 voidaan muistialue jakaa maksimissaan kahdeksaan yhtä suureen osaan, ohjelmoinnissa käytetään yleensä pienintä osoitetta missä piiri näkyy eli kantaosoitetta. [8]

### 7-segmentinäyttö

Seitsemällä LED:llä toteutettu 7-segmenttinäyttö on yleinen näyttötyyppi digitaalitekniikan sovelluksissa. Niitä voidaan ohjata piireillä, jotka muuttavat binaarisen BCD-luvun näytöllä näkyväksi numeroksi. Eräs suosittu 7-segmentnäytön ohjainpiiri on CMOS- ja HC-sarjan 4511, jossa aktivoitaessa  $\overline{LT}$ -tulo, kaikki segmentit syttyvät,  $\overline{BI}$ -tulolla näyttö voidaan pimentää ja LE-tulolla näytölle voidaan lukita sen hetkinen tila.

### LCD-näyttö

LCD (Liquid Crystal Display) eli nestekidenäyttö on ohut sähköisesti ohjattu näyttölaite. Näyttö koostuu valoa polaroivasta nesteestä, joka on suljettu soluihin kahden läpinäkyvän polarisoivan levyn väliin. Ilman sähkökenttää solut ovat lähes läpinäkyviä. Jokaiseen soluun voidaan johtaa sähköä, jolloin solun sisältämään nesteeseen muodostuu sähkökenttä. Tästä seuraa solun sisältämien molekyylien kääntyminen sähkökentän suuntaiseksi. Tällöin valo osuu omaa polarisaatiotasoaan vasten kohtisuorasti polarisoivaan levyyn, jolloin valo absorboituu täysin ja solu näyttää tummalta. [9]

### Sarjaportti

Sarjaliikenneportti RS-323 on yleisesti käytössä mm. PC:ssä ja monenlaisissa laitteissa joiden toimintaan voidaan ulkopuolelta vaikuttaa, kuten aktiiviset verkonosat. Se on vanha ja yleisin käytössä oleva liitännästandardi, joka on ollut käytössä jo ennen nykyistä PC:tä. Sarjaliitimenä on yleensä 9- tai 25-napainen D-liitin. Sarjaliikenne vaatii vähintään kolme johtoa: lähetys, vastaanotto ja maa.

Sarjaliikenteessä tiedonsiirto tapahtuu yhtä linjaa pitkin eli se onnistuu käyttämällä yhtä johtoa sekä maajohtoa. Yleensä varsinaisen tiedonsiirto johdon lisäksi käytetään muutamaa johtoa tietoa siirtävien laitteiden väliseen kommunikointiin. Näitä kutsutaan kättelylinjoiksi. Kättelylinjojen avulla laitteet ovat tietoisia toistensa olemassaolosta ja valmiudesta tiedonsiirtoon. Sarjaporttiliitännästä on olemassa monta toisistaan poikkeavaa toteutusta. Synkronisessa tiedonsiirrosta ajoitus hoidetaan erillisen kellosignaalin avulla. PC:n sarjaliitännää käytetään yleensä asynkroniseen tiedonsiirtoon, jossa merkkien ajallinen kesto on kiinteä ja tarkasti määritelty, mutta merkkien välistä aikaa voidaan muuttaa.

Sarjaliikenteessä RS-232C mukaan loogista nollaa vastaa (+3...+25)V jännite ja loogista ykköstä vastaa (-3...-25)V jännite. Vastaavat jännitteet PC:ssä on +12 V ja -12 V. Osa sarjaliitännöistä ymmärtää myös TTL-tasoisia signaaleja 0 V ja +5 V, mutta tieto tulee käännteisenä. Helppo standardiratkaisu on käyttää TTL-tasoisien ja RS-232 tasoisien signaalien sovittamiseen esimerkiksi MAX232-driver-piiriä. Tyypillinen sarjaliikenteessä lähetettävä merkki sisältää 8 databittiä sekä aloitus- ja lopetusbitin. [8, 10]

### 3.2 Kytkenän suunnittelu

Tehtävänä oli suunnitella laite, joka näyttää LCD-näytöllä kelloajan sekä päivämäärän. Kelloaika sekä päivämäärä tuli pystyä muuttamaan sarjaportin kautta nopeudella 2400 bit/s. Kytkenän sai rakentaa esimerkiksi 8051-mikrokontrollerin avulla.

#### 3.2.1 Osoitekoodaus

Proessori voi keskustella vain yhden dataväylään liitetyn piirin kanssa kerrallaan. Tämän vuoksi jokaiseen dataväylään liitetulle piirille on tehtävä oma piirivalintasignaali. Työssä päätettiin käyttää dekodattua piirivalintaa, jossa dekodataan varsinaiset piirivalintalinjat. Tämä onnistuu piirillä 74HC138, jolla muistialue voidaan jakaa enintään kahdeksaan yhtä suureen osaan. Työssä tarvittiin oma osoite LCD-näytölle, ulkoiselle EPROM-muistille ja testauksessa

käytettävälle 7-segmentnäytölle. Työssä käytettiin valmista kytkentäkorttia, jossa oli kytketty valmiiksi piiri 74HC138. Tehtäväksi jäi tutkia kytkennän perusteella osoiteväli jonka dekodauspiiri jakoi.

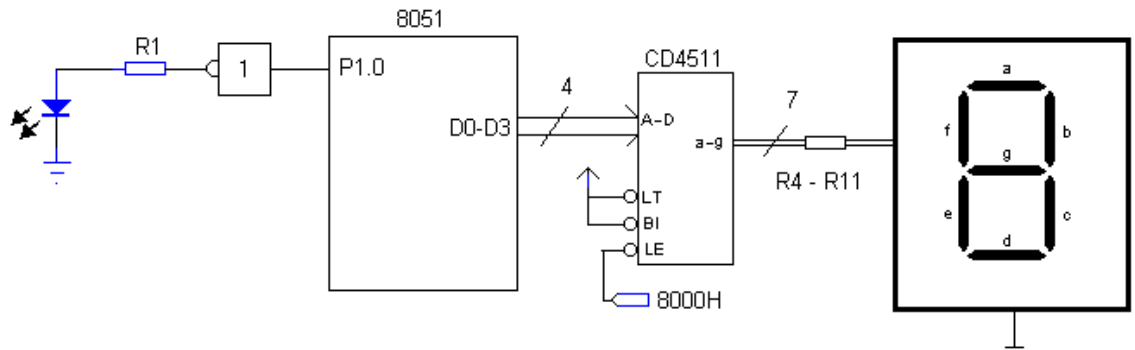
Taulukossa 5 on saadut osoitevälit. Osoiteväli saatiin jakamalla bittikartta neljään neljän bitin ryhmään. MSB-ryhmä saatiin tuloista  $A_{15}$ - $A_{12}$  ja LSB-bitti ryhmä saatiin tuloista  $A_3$ - $A_0$ . Alkuosoitteessa x eli don't care termi on nolla ja loppuosoitteessa x on yksi. Näistä muodostettiin neljä neljän bitin ryhmää, joista jokainen ryhmä muutettiin heksaluvuksi.

*Taulukko 5. Piirivalinnan osoiteavaruus*

Dekooderi	$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$		$A_{11}...A_0$	Osoiteväli
Liitin 0 ( $\overline{Y0}$ )	1	0	0	0		X...x	8000H - 8FFFH
Liitin 1 ( $\overline{Y1}$ )	1	0	0	1		X...x	9000H - 9FFFH
Liitin 2 ( $\overline{Y2}$ )	1	0	1	0		x...x	A000H - AFFFH
Liitin 3 ( $\overline{Y3}$ )	1	0	1	1		X...x	B000H - BFFFH
Liitin 4 ( $\overline{Y4}$ )	1	1	0	0		x...x	C000H - CFFFH
Liitin 5 ( $\overline{Y5}$ )	1	1	0	1		X...x	D000H - DFFFH
Liitin 6 ( $\overline{Y6}$ )	1	1	1	0		x...x	E000H - EFFFH
Liitin 7 ( $\overline{Y7}$ )	1	1	1	1		X...x	F000H - FFFFH

### 3.2.1 Testaamiseen 7-segmentnäyttö

Ohjelmistoa tehdessä pyrin aluksi suorittamaan ohjelmiston testausta 8051 porttiin P1.0 kytketyn LED:n avulla. Tämä osoittautui kuitenkin riittämättömäksi, joten jouduin lisäämään kytkentään 7-segmentnäytön. Kuvassa 4 on periaate LED:n ja 7-segmentnäytön lisäämisestä kytkentään.

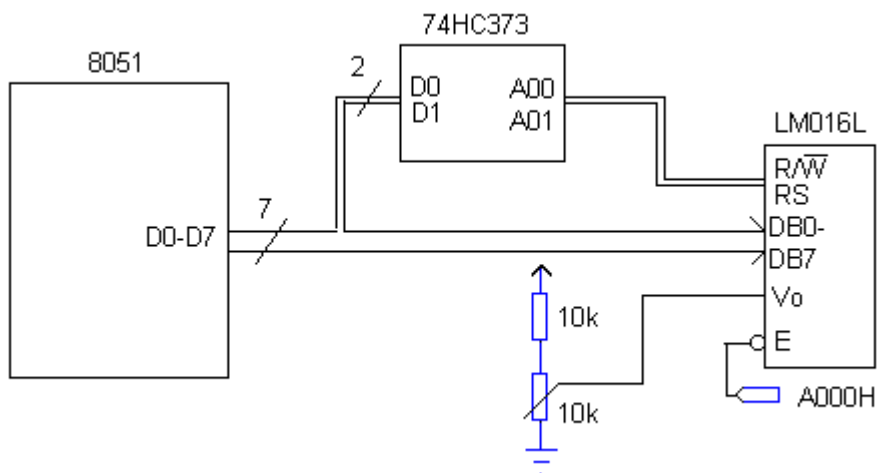


Kuva 4. Periaate 7-segmenttinäytön ja LED:n lisäämisestä kytkentään

Piiri CD4511 ohjaa 7-segmenttinäyttöä. Piirin tuloihin A-D kytetään prosessorikortin datalinjat D0-D3 ja piirinvalintasignaalksi valittiin kantaosoite 8000H.

### 3.2.2 LCD-näyttö

Tehtävän tarkoituksena oli näyttää kellonaika ja päivämäärä LCD-näytöllä. Kuvassa 5 on periaate LCD-näytön kytkemisestä prosessorikorttiin. Näyttönä oli yksirivinen LCD-näyttö, jossa pystyi näyttämään 16 merkkiä kerrallaan.



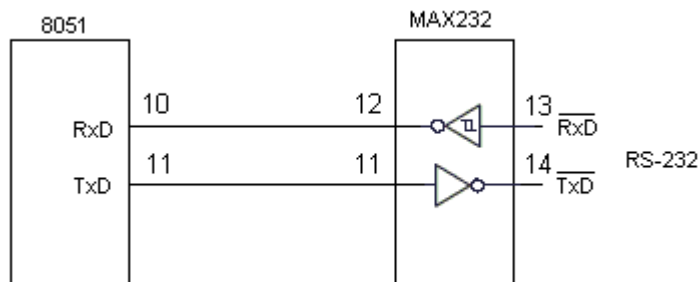
Kuva 5. LCD-näytön liittäminen prosessorikorttiin

LCD-näyttö sisälsi näytönohjainpiirin LM016L, jonka tuloihin DB<sub>0</sub>-DB<sub>7</sub> kytkettiin prosessorikortilta datalinjat D<sub>0</sub>-D<sub>7</sub>. Tuloon R/ $\bar{W}$  kytkettiin A<sub>00</sub> ja tuloon RS kytkettiin A<sub>01</sub>. Kytkentäkortilla oli valmiiksi kytketty lukkopiiri 74HC373 eli latch, jossa lähdöt A<sub>00</sub> ja A<sub>01</sub> sijaitsivat. Näytönohjainpiirin tulolla Vo säädetään näytön

kontrastia ja tähän tuli tehdä jännitteenjako. Näytön teksti näkyi noin 1 V jännitteellä. Enable-tuloon kytketään sallintasignaali osoitetekooderilta. Osoitteeksi valittiin A000H.

### 3.2.3 Sarjaportti

Kelloajan ja päivämäärän muuttaminen tuli toteuttaa RS-232-liitännän kautta. Koska jännitetasot prosessorikortissa ja PC:n sarjaportissa ovat erilaiset, joudutaan väliin laittamaan piiri, joka muuttaa jännitteet sopiviksi molempiin suuntiin. Sarjaliitettä voisi toimia ilman erillistä piiriäkin, mutta bitit olisivat aina käänteisiä. Tässä työssä sovitukseen käytetään piiriä MAX232. Kuvassa 6 on periaate sarjaportin liittämistä.



Kuva 6. Sarjaportin liittäminen prosessorikorttiin

Yhteys PC:ltä sarjaportin kautta kellolaitteeseen muodostetaan ristiin kytketyllä sarjakaapelilla. Sarjaportin alustukseen annettava luku lasketaan kaavalla:

$$\text{baudinopeus} = \frac{2^{\text{SMOD}}}{32} * \frac{\text{kellotaajuus}}{12[256 - \text{TH1}]}$$

josta tehtävän ominaisuuksin saadaan:

$$2400 = \frac{2^0}{32} * \frac{11,059\text{MHz}}{12[256 - \text{TH1}]} \Rightarrow \text{TH1} = 244 = \text{F4H}$$

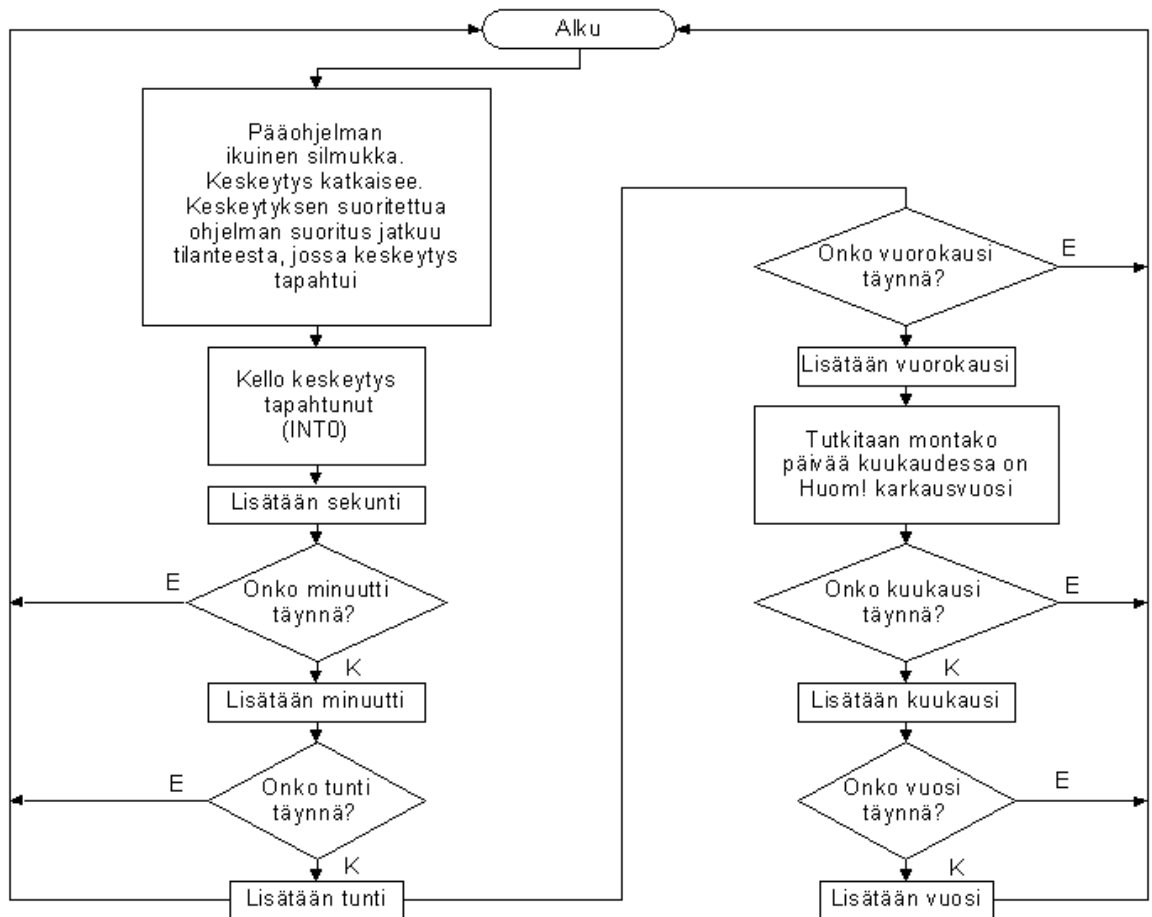
Saadulla luvulla F4H alustetaan koodissa sarjaportin laskuri, jolloin nopeudella 2400 bit/s tapahtuva sarjaliikenne tapahtuu oikein ilman merkkivirheitä.

Taajuusstandardin valmiin valvontakortin kuva on liitteessä D.

#### 4 OHJELMAKOODI

Ohjelmakoodin suunnittelu tuli aloittaa miettimällä tarpeita, joita kokonaisuuden tulisi toteuttaa. Koska tarkoituksena oli toteuttaa ajannäyttölaite, joka kertoo käyttäjälle laitekokonaisuuden toimivan oikein, tuli kellosta tehdä tarkka ja luotettava.

Jotta ajan lisääminen toimisi oikein, tuli ajan lisäämiseen käytettävän signaalin päivittää sekuntia riippumatta muista tekijöistä. Tämän vuoksi sekuntisignaali päätettiin liittää keskeytykseen INT0, koska tällä keskeytyksellä on oletuksena suurin prioriteetti 8051:ssä. Muita keskeytyksiä 8051:ssä on toinen ulkoinen keskeytys INT1, ajastin/laskuri 0 ja 1 sekä sarjaportti. Keskeytyksen INT0 suorittamaan ohjelmalohkoon päätettiin laittaa sekuntipulssista aiheutuva ajanlisäys siten, että sekunti päivittää minuuttilaskuria, minuuttilaskuri päivittää tuntilaskuria ja edeten edelleen laskureina kymmeneen vuosiin. Ongelmana tässä oli eripituiset kuukaudet, joten jokaiselle kuukaudelle täytyi tehdä oma vertailu. Lisäksi helmikuussa on joka neljäs vuosi karkauspäivä ja kelloajassa on normaaliaika ja kesäaika. Kelloa käännetään tunnilla eteenpäin maaliskuun viimeinen sunnuntai klo 4.00 ja normaaliaikaan palataan muuttamalla kelloaikaa lokakuun viimeinen sunnuntai klo 4.00. Kesä- ja normaaliajan muutos jätettiin tässä ohjelmassa käyttäjälle. Kuvassa 7 on vuokaavioesitys tehdystä ajanlisäysratkaisusta.



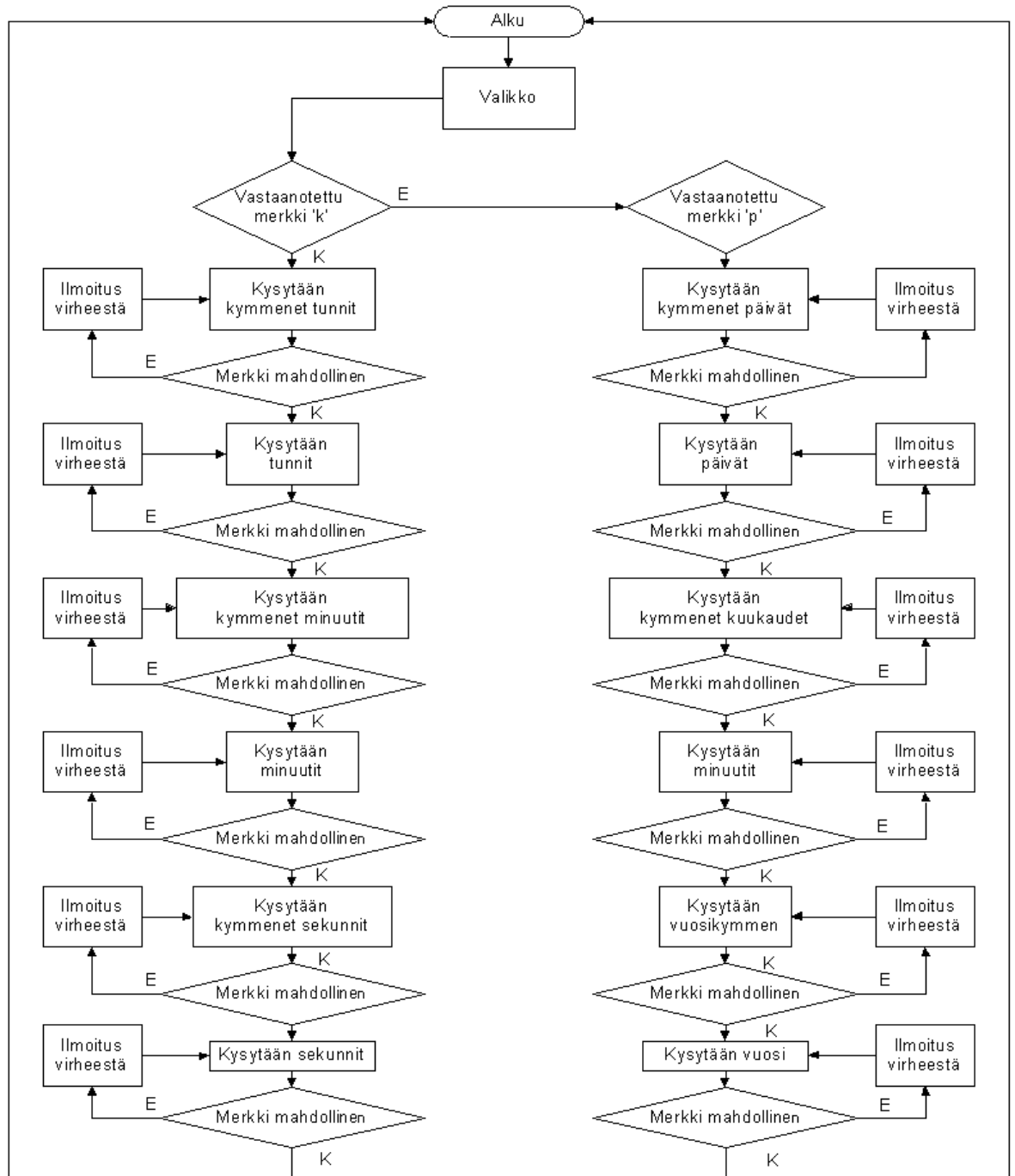
Kuva 7. Vuokaavioesitys kellopulssia seuraavasta ajan lisäämisestä

Keskeytyksen antavaa laskuri 0:aa päätettiin käyttää ohjelmassa viiveen tekemiseen. Viivettä tarvitaan LCD-näytölle lähetettävien merkkien väliin. Viiveen voi tehdä myös esim. kahdella for-silmukalla. Toinen keskeytyksen antava laskuri 1 päätettiin käyttää sarjaliikenteen tahdistamiseen. Väärin tahdistettu sarjaporttiliikenne aiheuttaa merkkivirheitä.

Sarjaportin keskeytyksellä on pienin prioriteetti keskeytyksistä. Sarjaportin keskeytyksen ohjelmalohkoon päätettiin laittaa vain tarkastus tiedosta, joka sarjaportilla vastaanotetaan. Tarkastus vaikuttaa vain, kun ohjelma lähettää sarjaportin avulla vastaanottavaan ohjelmaan aloitusvalikon. Ajan tai päivämäärän muuttaminen aloitetaan, kun aloitusvalikon jälkeen syötetään PC:ltä aloitusvalikossa oleva kirjain. Tämän jälkeen tulevaa sarjaliikennettä käsitellään pääohjelman kautta pyörivällä silmukalla, joka lähettää kysymyksen ja odottaa



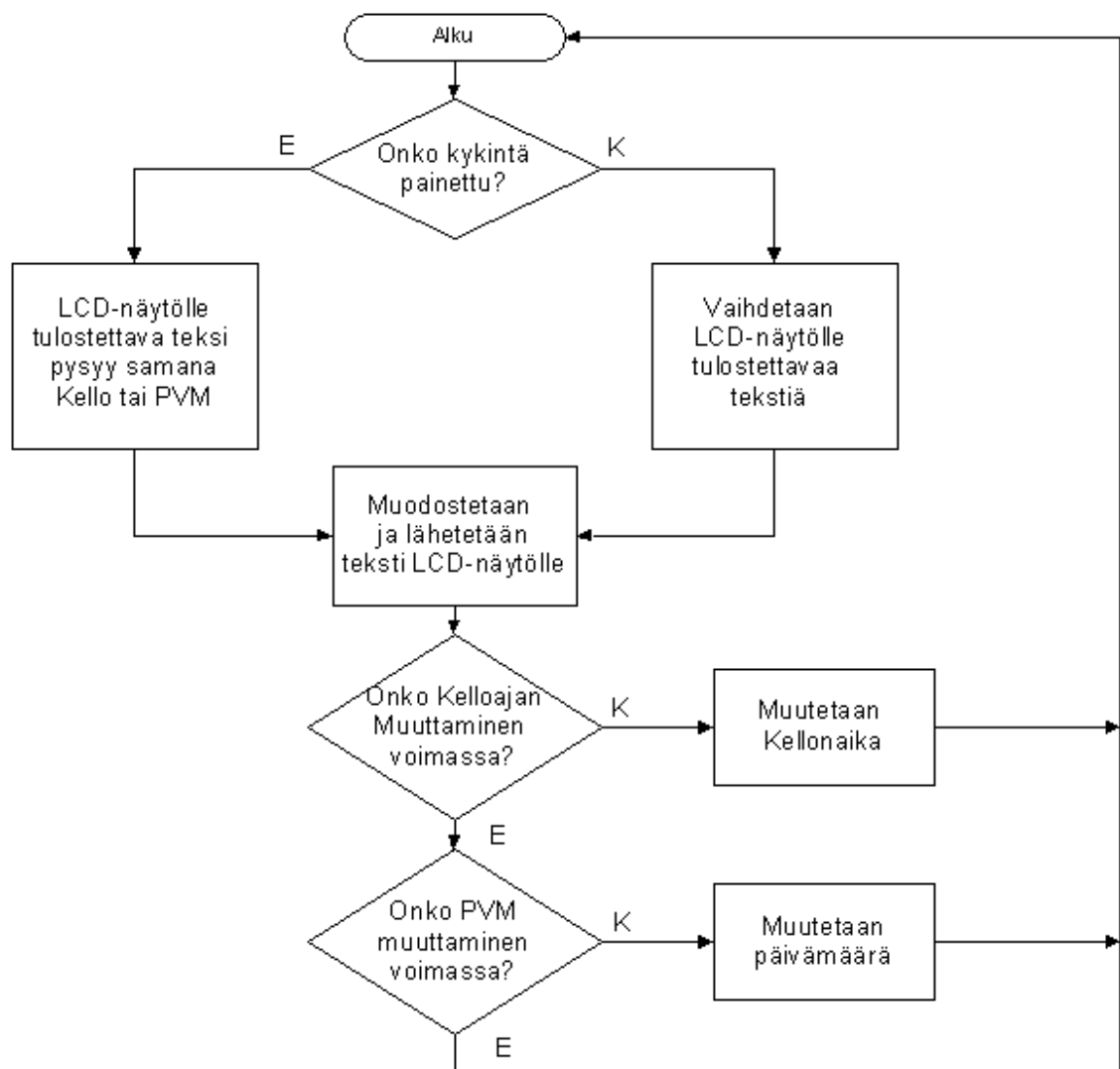
tähän hyväksyttävän vastauksen. Kuvassa 8 on vuokaavioesitys periaatteesta, jolla ajan tai päivämäärän muuttaminen tapahtuu.



Kuva 8. Vuokaavioesitys kelloajan sekä päivämäärän muuttamisesta

Ohjelma toimii lähettämällä aloitusvalikon. Kun valikossa valitaan oikea merkki, edetään kyseistä polkua. Sarjaporttia pitkin lähetetään kysymys, ja kun tähän vastataan oikein, lähetetään seuraava kysymys ja palataan pääohjelman

silmukkaan, joka jää odottamaan seuraavaa sarjaportilta tulevaa keskeytystä. Keskeytyksen jälkeen saapunutta merkkiä verrataan seuraavaan kysymykseen. Kun merkki on sopiva, etenee ohjelman suoritus edelleen. Tämä silmukka etenee, kunnes tarvittaviin tietoihin on saatu vastaus. Tämän jälkeen aloitusvalikko lähetetään uudelleen ja jäädään odottamaan seuraavaa kelloajan tai päivämäärän muutosta. Pääohjelman toiminta on kuvan 9 vuokaavioesityksessä.



Kuva 9. Vuokaavioesitys pääohjelman ikuisesta silmukasta

Kokonaisuutena ohjelma pyörii koko ajan pääohjelman sisällä suorittaen kyselyä, ollaanko muuttamassa kellonaikaa tai päivämäärää, sekä lähettää LCD-näytölle uuden tekstin. Kaikki keskeytykset keskeyttävät pääohjelman toiminnan ja suuremmalla prioriteetilla oleva keskeytys keskeyttää pienemmällä

prioriteetilla olevan keskeytyksen tapahtumat. Kun edelle mennyt tehtävä on suoritettu, palataan aina edellisen keskeytyskohtaan ja jatketaan sen suorittamista loppuun. Ohjelman on lähdekoodi liitteessä E.

Yleensä LCD-näytölle lähetettävä teksti vaihtuu alle sekunnin välein, mutta annettaessa uusia tietoja saattaa LCD-näytön päivitys keskeytyä hetkeksi. Tämä ei vaikuta kellonaikaan, vaan kelloaika päivittyy joka tapauksessa aina tulevan sekuntisignaalin mukaan.

## 5 KÄYTTÖOHJE

Kellolaitteiston aikaa voidaan muokata kytkemällä laite ristiinkytketyllä sarjakaapelilla PC:n sarjaporttiin. Tämän jälkeen käynnistetään ASCII-koodia tukeva ohjelma, kuten Hyper Terminal. Ohjelmasta valitaan käytettävä sarjaportti. Tiedonsiirtomäärityksiin määritellään nopeudeksi 2400 bit/s, 8 databittiä, ei pariteettia, 1 stop-bitti ja ei kättelyä.

Koska ohjelma lähettää aloitusvalikon käynnistyksessä, kannattaa kortilta katkaista virta tai käynnistää uudelleen painonapista. Käynnistyksessä ohjelma lähettää aloitusvalikon sarjaporttia pitkin. Vaihtoehtoisesti voi kokeilla ilman uudelleenkäynnistämistä lähettämällä 'k' tai 'p' -kirjaimen, jolloin ohjelma kysyy uutta tietoa tai lähettää virheilmoituksen väärästä merkistä.

Kysymyksiin vastataan painamalla merkkiä kerran ja odotetaan, kunnes tulee vastaus ok, jolloin tieto on muutettu ja tämän jälkeen lähetetään seuraava kysymys. Aikaa yhden tiedon hyväksymiseen kuluu noin sekunti. Jos kysymykseen annetaan mahdoton vastaus kahdesti peräkkäin, tulee virheilmoitus ja kysymys uudelleen.

Laitteistossa olevalla toisella painonapilla voidaan vaihtaa LCD-näytön tekstiä päivämäärän ja kelloajan välillä. Painonappia täytyy pitää pohjassa kunnes näytöntila vaihtuu.

## 6 YHTEENVETO

Työssä toteutettiin osa Kajaanin ammattikorkeakoulun radiolaboratorioon tulevasta taajuusstandardista. Tässä työssä tehtiin taajuudenjakolaite, joka jakaa taajuuden dekaideittain 10 MHz:stä 1 Hz:iin saakka. Taajuudenjakolaiteessa päädyttiin käyttämään pääasiassa synkronisia binaarilaskureita 74HC163. Näiden pohjalle suunniteltiin kytkentä, joka juotettiin piirikortille. Ongelmia oli saada kytkentä toimimaan oikein. Suurimman ongelman tuottivat 74HC163-piirit, jotka yleensä toimivat oikein, mutta välillä piirit antoivat TC-lähdöstä yhden lyhyen pulssin sijasta kaksi pulssia. Tästä johtuen T-kiikun lähdöstä saatava signaali ei ollut symmetristä kanttiaaltoja, vaan signaali vain kävi nopeasti 1-tilassa. Vikaa etsiessä havaittiin, että osa piireistä toimi oikein ja osalla piireistä tuli toiminta virheitä 5 V:n käyttöjännitteellä, mutta toimivat täysin vähän yli 4 V:n käyttöjännitteellä. Ongelmasta päästiin kokeilemalla noin 20 piiriä, joista suurimmassa osassa oli sama vika. Testatuilla piireillä oli kaksi valmistajaa, joista huomattiin, että vialliset piirit olivat pääasiassa samalta valmistajalta. Nämä piirit saattoivat olla vanhempia ja olleet kauemmin käytössä, joten ne saattoivat olla ajan mittaa kärsineet kytkentävirheistä tai staattisesta sähköstä. Kytkentään jäi edelleen pientä ylikuulumista, jota tällä valmistusmenetelmällä ei saatu täysin poistumaan. Häiriötä pyrittiin vähentämään bypass-kondensaattoreilla ja yhdistämällä käyttämätön piirilevy maapotentiaaliin.

Työssä toteutettiin myös radiolaboratorioon tulevan taajuusstandardin valvontalaite, jolla tarkkaillaan kokonaisuuden toimintaa. Tämä toteutettiin rakentamalla prosessorikortille ajannäyttölaite, jolle taajuusstandardista tullaan ottamaan sekuntipulssia. LCD-näytölle toteutetusta ajannäyttölaitteesta voidaan tarkkailla kelloaikaa ja päivämäärää. Jos järjestelmä toimii, pysyy kello oikeassa ajassa. Kelloaika ja päivämäärä asetetaan oikeaksi sarjaportinkautta ASCII-tekstiä tukevalla ohjelmistolla. Tällainen ohjelma on esimerkiksi Hyper Terminal. Pahimmat ongelmat tuli lopuksi. Kellolaitteisto toimi oikein 8051:n emulaattorilla, mutta itsenäisesti EPROM-muistin avulla ohjelma ei käynnistynyt. Viaksi selvisi liian pieni RAM-muisti.

## LÄHDELUETTELO

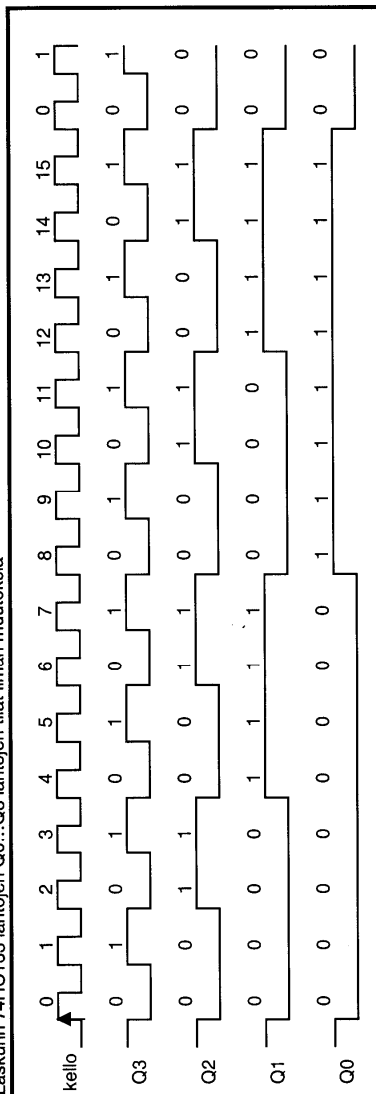
- [1] Volotinen Vesa. Digitaalitekniikka, perusteet ja sovellukset. Toinen painos. Porvoo: WSOY 1997. 290 s. ISBN 951-0-21150-8
- [2] Mikropiirin 74HC04 datakirja luettu 1.3.2004. [www dokumentti]  
[http://www.philipslogic.com/products/hc/pdf/74hc\\_hct04.pdf](http://www.philipslogic.com/products/hc/pdf/74hc_hct04.pdf)
- [3] Harri Honkanen. Elektroniikan kurssi- ja laboratoriotyön materiaali 2002
- [4] Pekka Rantala. Digitaalitekniikka osa A. Nide A. 1997 ISBN 951-559-213-5
- [5] Pekka Rantala. Digitaalitekniikka osa B. Nide B. 1997 ISBN 951-559-214-3
- [6] Mikropiirin 74HC163 datakirja, luettu 1.3.2004 [www dokumentti]  
[http://www.semiconductors.philips.com/acrobat/datasheets/74HC\\_HCT163\\_CNV\\_2.pdf](http://www.semiconductors.philips.com/acrobat/datasheets/74HC_HCT163_CNV_2.pdf)
- [7] Mikrokontrolleri MCS-51-perheen tietoja. luettu 3.3.2004. [www dokumentti]  
<http://www.tietokotka.planet.fi/tiedostot/mcs51.pdf>
- [8] Airaksinen Risto. Digitaalitekniikan kurssi- ja laboratoriotyön materiaali 2002
- [9] Wikipedia vapaa tietosanakirja. muutettu 18.2.2004 [www dokumentti]  
<http://fi.wikipedia.org/wiki/LCD>
- [10] Tomi Engdahl. RS-232-sarjaliitäntä. päivitetty 20.07.1993 [www dokumentti]  
<http://www.hut.fi/~then/mytexts/rs-232c.html#xi>

## LIITTEET

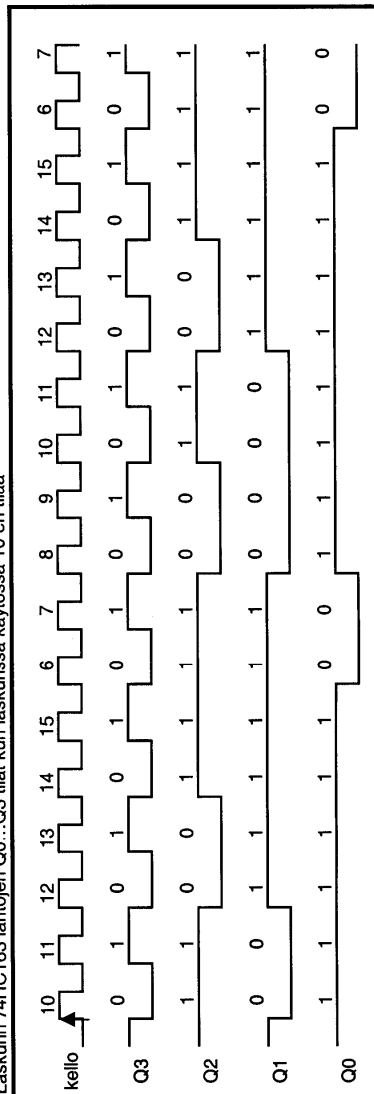
- A Laskurin 74HC163 aikakaavio 16:sta ja kymmenen eritilaisena.
- B Taajuusdistribuuttorin kytkentäkaavio.
- C Toteutettu taajuusdistribuuttori.
- D Toteutettu valvontakortti.
- E Valvontakortin ohjelmakoodi

LIITE A

Laskurin 74HC163 lähtöjen Q0...Q3 lähtöjen tilat ilman muutoksia



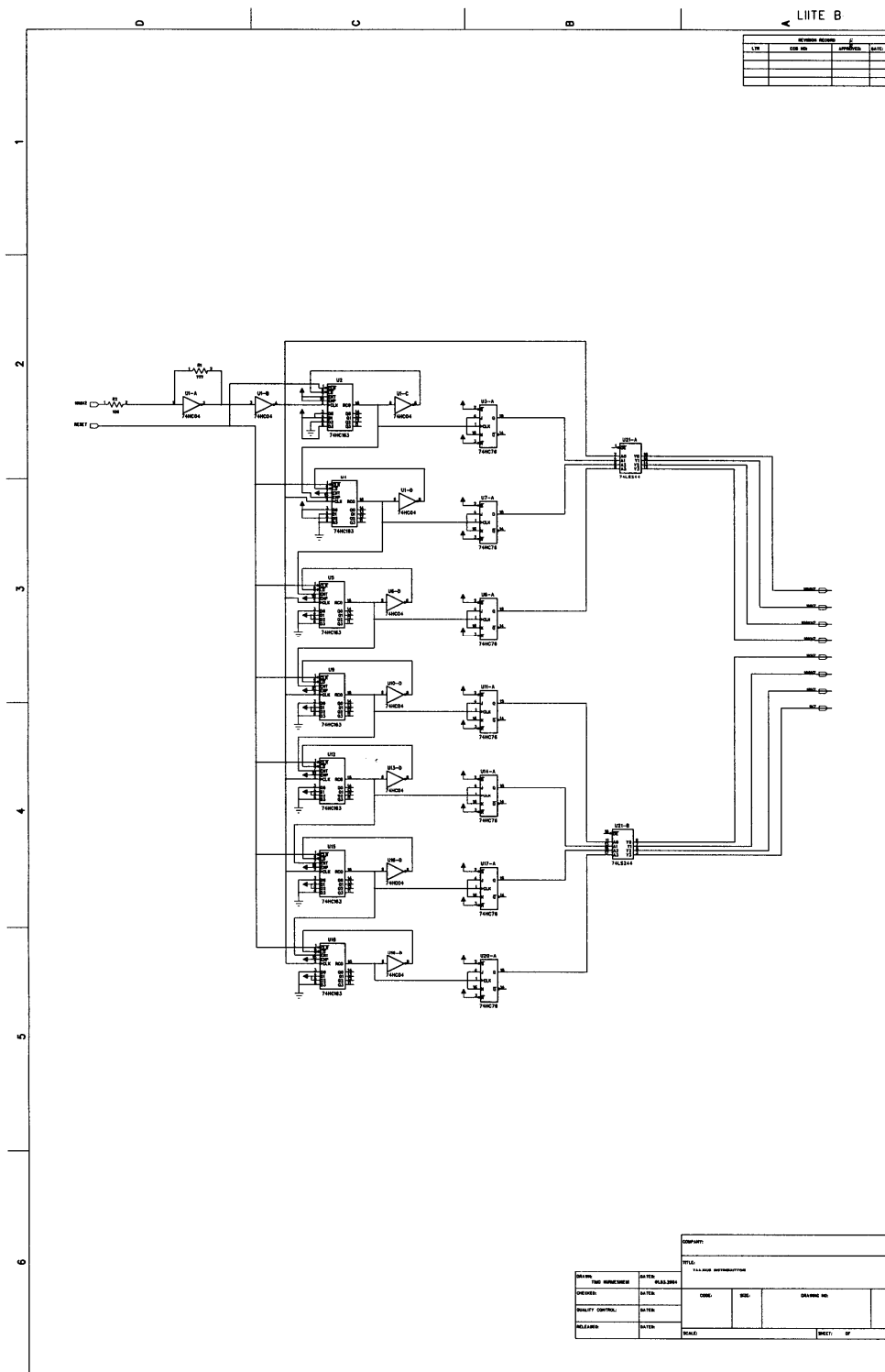
Laskurin 74HC163 lähtöjen Q0...Q3 tilat kun laskurissa käytössä 10 eri tilaa





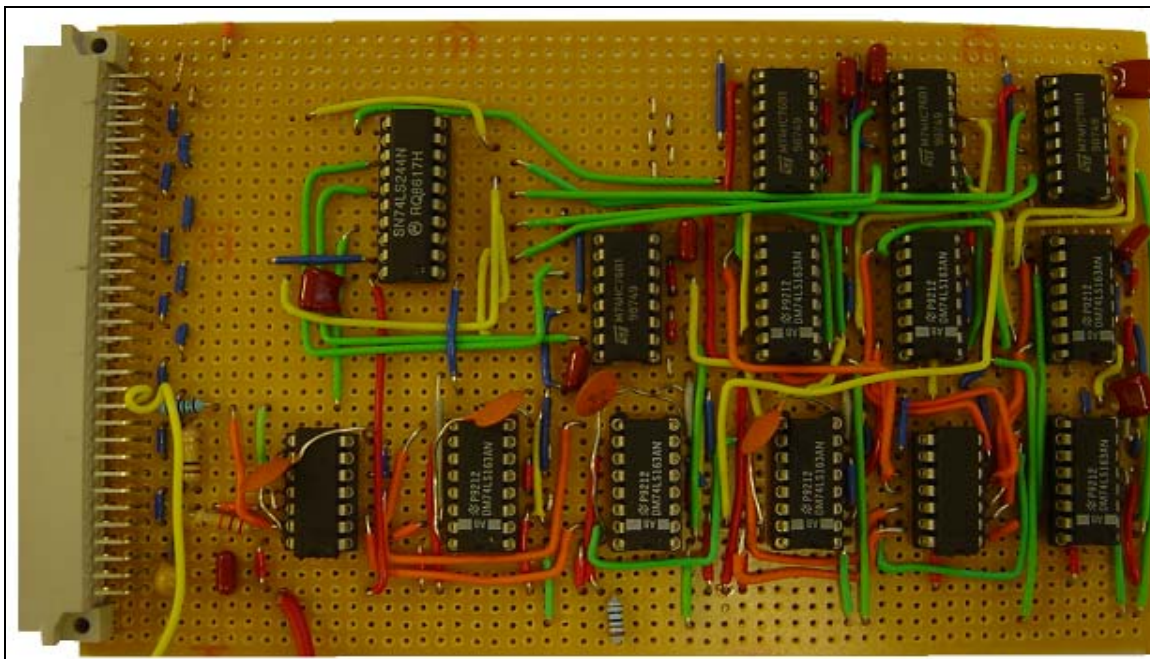
LIITE B

REVISIONI MUUTOS		P	
N:o	MIKÄ MÄÄ	SIJOITUS	ALUE

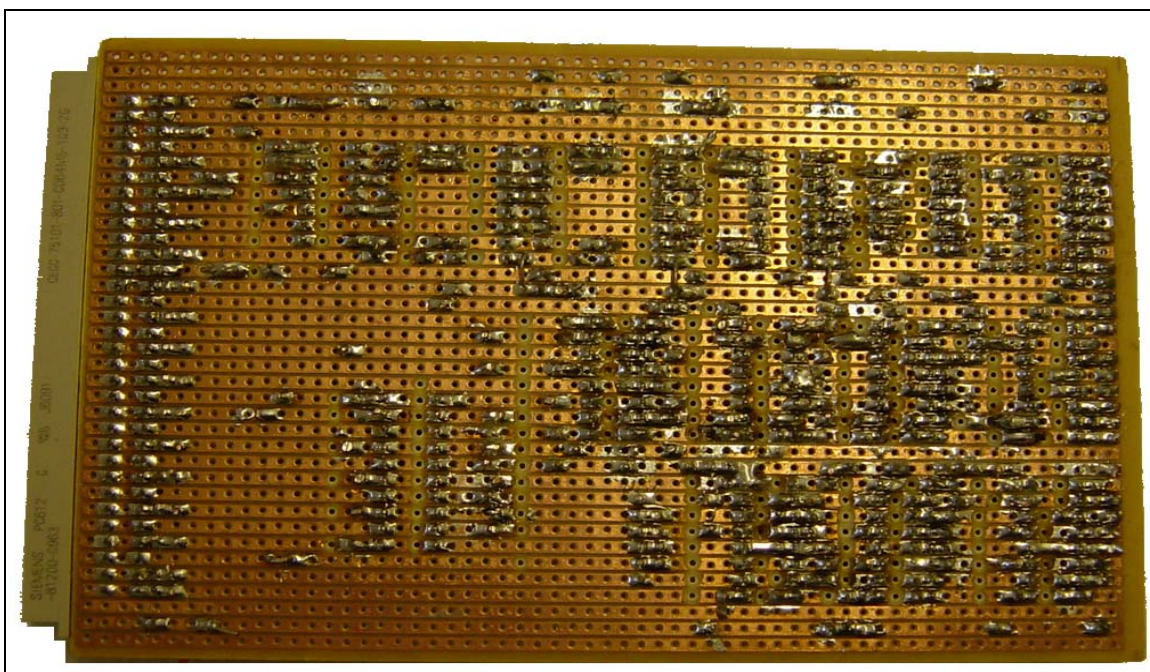


DATE:		DRAWN BY:		CHECKED BY:		DATE:	
DESIGN:		DATE:		SCALE:		SHEET: OF	

Taajuusdistribuuttorin toteutus piirilevyllle.

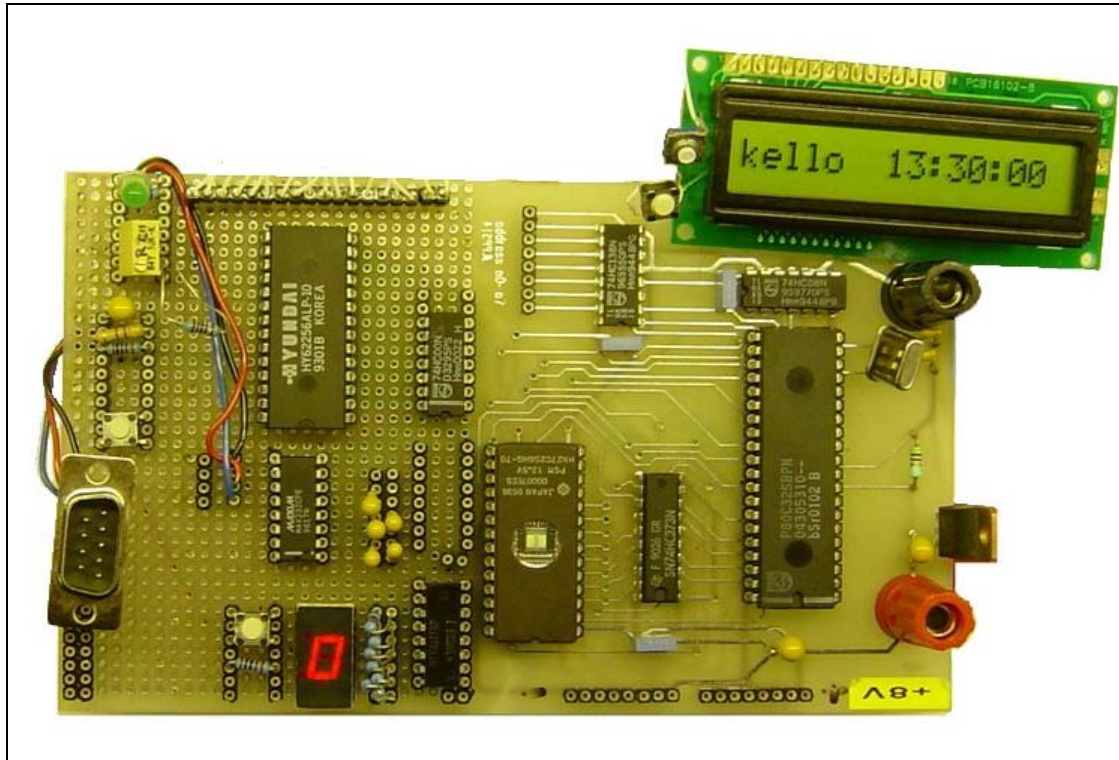


Taajuusdistribuuttorin juotokset.

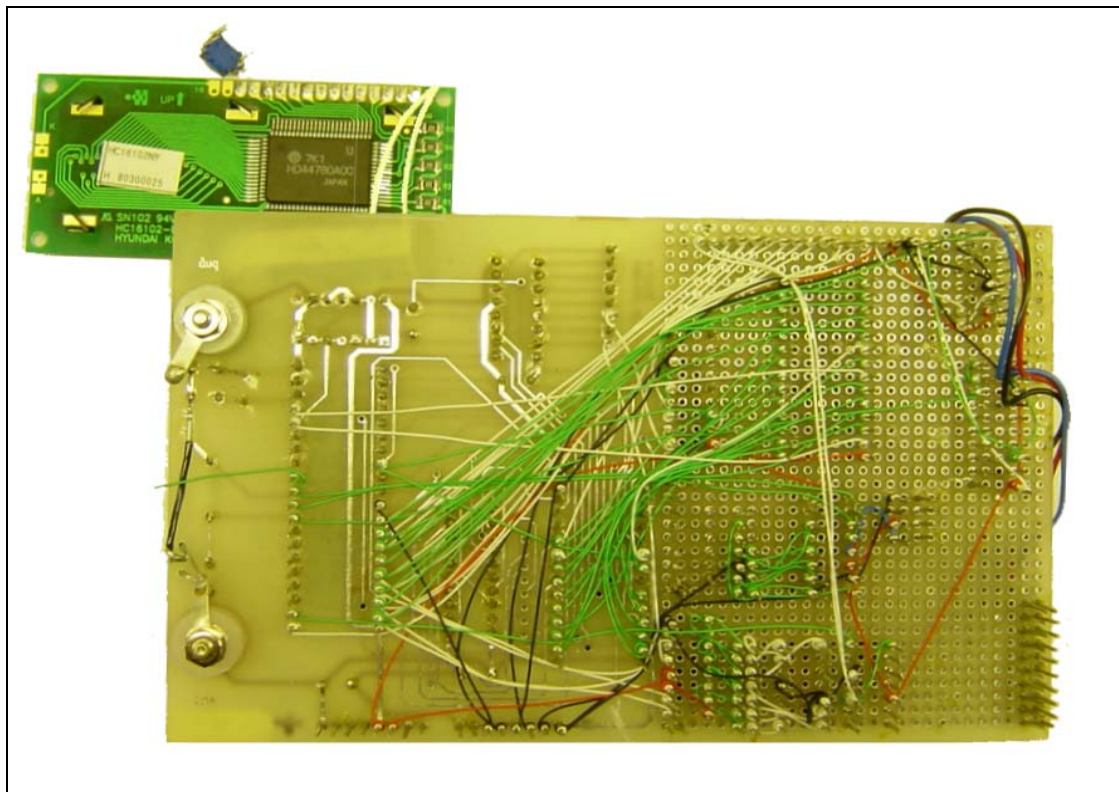




Taajuusstandardin toiminnan valvontakortti.



Valvontakortin kytkentä alapuolelta.



## LIITE E/1

/\*\*\*\*\*\*

Timo Nurmesniemi 23.03.2004  
 Insinööriyöhön liittyvän kellon ohjelmisto  
 Ohjelma toimii seuraavasti:

Sarjaporttiin kytkettävä ristiin kytketty kaapeli!!!  
 LCD-näytölle ilmestyy päivämäärä ja kellonaika. Piirikortilla olevasta painonapista painettaessa LCD-näytölle ilmestyy pelkkä kellonaika tai Päivämäärä. Kellonaikaa voi muuttaa kytkemällä sarjakaapeli PC:hen ja avaamalla Hyper Terminal ohjelma. Tiedonsiirron nopeus 2400 bit/s, 8 databit, ei pariteettia, 1 stop bitti ja ei kättelyä.  
 Kun korttiin kytkee jännitteen 8 V, ilmestyy Hyper Terminaaliin valikko, josta voi valita kelloajan tai päivämäärän muuttamisen. Annettavat uudet arvot tulevat voimaan heti ja kello toimii kokoajan taustalla vaikka LCD-näytölle uusi aika ei heti päivytyisikään.

\*\*\*\*\*/

/\* OTSIKKOTIETOJEN MÄÄRITYS \*/

```
#include<io51.h>
#include <stdio.h> /* sprintf vaatii*/
#include <string.h> /* strlen vaatii*/
#define LCD_ALUSTUS (*(char *)0x01A000) /* LCD-näytön control-rekisterin
                                         osoite */
#define LCD_DATA (*(char *)0x01A002) /* LCD-näytön data-rekisterin osoite
                                         */
#define LED P1.0 /* LED testaamista varten */
#define S1 P1.1 /* Painonapin määrittys LCD-näytön
näyttämän vaihtamista varten */
#define SEVSEG (*(char *)0x018000) /* 7-segment näytön osoitteen
määrittys */

#define INIT_TMOD 0x01 /* otetaan laskuri käyttöön */
#define INIT_TL0 0x20 /* lsb laskurille alkuarvo */
#define INIT_TH0 0xFC /* msb laskurille alkuarvo jolloin laskuri alkaa
laskemaan luvusta 18FCH*/
```

/\* ALIOHJELMIEN JA MUUTTUJIEN MÄÄRITYS \*/

```
void tulostusI(char siirto [16]);
void delay(int aika);
void LCD_n_alustus();
void laskurin_alustus();
void sarjaportin_alustus();
void aikalasku();
void tekstimuodostus();
void paivays();
void RS_lahetys(char hyper[98]);
void valikkovalinta();
```

## LIITE E/2

```

void valikkoH();                               /* Kelloajan muutokseen käytettävä
                                                valikkorakenne */

    void RS_kello();
    void val_ktunti();
        int vikailmkt=0;
        int valm=0;
    void val_tunti();
        int vikailmt=0;
    void vall_kmin();
        int vikailmkmin=0;
    void vall_min();
        int vikailmmin=0;
    void vall_ksek();
        int vikailmksek=0;
    void vall_sek();
        int vikailmsek=0;

void valikkoP();                               /* Päivämäärän muutokseen
                                                käytettävä valikkorakenne */

    void RS_date();
    void va2_kymkuu();
        int val2_kkuu;
        int vikailmkkuu=0;
        char Pmuuttuja[40]="\r\nValitsit paivamaaran muutoksen\n\n";
    void va2_kuu();
        int vikailmkuu=0;
    void va2_kvuo();
        int vikailmkvuo=0;
    void va2_vuosi();
        int vikailmvuo=0;
        char val2_svirhe[22]="\r\nHUOM! PVM muutettu";
    void va2_kpaivat();
        int vikailmkpai=0;
    void va2_paivat();
        int vikailmpai=0;
        int va2m=0;
        int paivmtest=0;

char muuttuja[6]="\nOK\n\r";                   /* Yleisiä muuttujia */
char test;
char hm;                                       /* Hyper terminal ohjelman lähettämä merkki */
int hk;                                       /* Sarjaportin vastaanoton
                                                keskeytyksen tila */
int ktaso=0;                                  /* Sisältää kellon muutoksen
                                                kohdetiedon switch-case rakenteessa */
int ptaso=4;                                  /* Sisältää päivämäärän muutoksen
                                                kohdetiedon switch-case rakenteessa */

int aika=20;
int valinta=1;                                /* LCD-näytön näyttämän valinnan muuttuja */
int i=0,apu, pulssi=0;
int sek=0, ksek,min=5,kmin=3,tun=6, ktun=1;  /* Kelloajan muuttujat */
int pv=5,kpv=1,kk=3, kkk=0,kuukausi=3;      /* Päivämäärän muuttujat */
int vuosi=4, kvuosi=0, kv=0, karkaus=0;      /* HUOM! Laita kk ja kuukausi
                                                aina samaksi! */

int merkkilaskuri=0,riv=0;                   /* Sarjaportin muuttujat */
char merkkijono[16];                         /* LCD-näytölle tulostettavat tekstin taulukko */
char merkkijonol[40]="\r\n\r\nTerve tuloa muuttamaan aikaa!!!\n\n\r";

```

## LIITE E/3

```

/*****/
main()
{
    TCON=0x85;          /* Keskeytys /int0 ja /INT1toimimaanlaskevalla reunalla */
    IE=0x91;           /* Sallitaan keskeytys /INT0 ja sarjaportti */

    LCD_n_alustus();   /* Suorittaa LCD-näytön alustuksen aliohjelman */
    laskurin_alustus(); /* Suorittaa sarjaportissa käytettävän laskurin
                          alustuksen aliohjelman */
    sarjaportin_alustus(); /* Suorittaa sarjaportin alustuksen aliohjelman */
    i=0;

    RS_lahetys(merkkijono1); /* Suorittaa aliohjelman, joka lähettää
                              sarjaporttia pitkin tervehdyksen */
    valikkovalinta();        /* Suorittaa aliohjelman, joka lähettää
                              sarjaporttia pitkin alkuvalikon */

    for(;;)                /* Ikuinen silmukka alkaa */
    {                       /* Silmukan suoritus keskeytyy aina keskeytyksen
                              tapahtuessa, jolloin kyseisen keskeytyksen ohjelma
                              suoritetaan loppuun ja ohjelman suoritus jatkuu siitä
                              missä se oli ennen keskeytystä */

        SEVSEG=hk;
        if(S1==0)          /* Tarkistaa onko painonappia painettu */

        {
            if(valinta==0) {valinta=1;}
            else {valinta=0;}
        }

        if(hk==1)          /* Tarkistaa onko sarjaportilla on vastaanotettu*/
        {
            SEVSEG=hk;     /* Koodin testaamiseen */
            if(test=='h')  /* Tarkistaa onko kelloajan muutos valittuna */
            {
                valikkoH(); /* Jos kelloajan muutos on valittuna siirrytään
                              kelloajan muutosvalikkoon */
            }
            if(test=='p')  /* Testaa onko päivämäärän muutos valittuna */
            {
                valikkoP(); /* Jos päivämäärän muutos on valittuna siirrytään
                              päivämäärän muutosvalikkoon */
            }
            hk=0;          /* Nollaa muuttujan joka kertoo sarjaportin
                              vastaanoton keskeytyksestä */
        }

        tekstimuodostus(); /* Suorittaa aliohjelman joka muodostaa
                              LCD-näytölle lähetettävän tekstin */
        /* HUOM! Älä laita tämän ikuisen silmukan sisälle viivettä */
    }
}
/*****/

```

## LIITE E/4

```

/*****/
void valikkoP()      /* Tämä aliohjelma tutkii mikä päivämäärän muuttamisessa on
                    vuorossa ja suorittaa kyseisen aliohjelman */
{
switch(ptaso)
{
case 0:             /*Suorittaa aliohjelman, joka muuttaa kymmenet kuukaudet */
    {
        va2_kymkuu();
        break;
    }

case 1:             /* Suorittaa aliohjelman, joka muuttaa kuukaudet */
    {
        va2_kuu();
        break;
    }

case 2:             /* Suorittaa aliohjelman, joka muuttaa kymmenet vuodet */
    {
        va2_kvuosi();
        break;
    }

case 3:             /* Suorittaa aliohjelman, joka muuttaa vuodet */
    {
        va2_vuosi();
        break;
    }

case 4:             /* Suorittaa aliohjelman, joka muuttaa kymmenet päivät */
    {
        va2_kpaivat();
        break;
    }

case 5:             /* Suorittaa aliohjelman, joka muuttaa päivät */
    {
        va2_paivat();
        break;
    }

case 6:             /* Kun päivämäärän muutokset on suoritettu nollataan */
    {
        /* muuttujia sekä näytetään uudelleen alkuvalikko */
        test='t';          /* Poistutaan päivämäärän muutos luupista */
        ptaso=4;
        vikailmvuo=2;     /* Muuttuja nollataan, jotta seuraavan kerran
                        kysytään päivämäärää samassa paikassa */
        RS_lahetys(merkkijonol); /* Näytetään alkutervehdys */
        valikkovalinta();    /* Näytetään Alkuvalikko */
        break;
    }
}
}
/*****/

```

## LIITE E/5

```

/*****/
void va2_kymkuu()      /* Tämä aliohjelma suorittaa kymmenien kuukausien
                        muuttamisen */
{
    char val2_kku[28]="\n\rAnna kymmenet kuukaudet>";
    char val2_kkuuv[35]="\r\nKirjoita oikein !!! 0 tai 1 >";
    char val2_kuuEKA[17]="\r\nAnna kuukausi>";

    paivmtest=0;      /* Nollaa muuttujan(päivän muuttaminen) jolla
                        testataan kuukauden pituutta */

    val2_kkuu=SBUF-0x30; /* Sarjaportista tulee ASCII merkkejä. ASCII
                        taulukosta näkee muunnoksen, jonka mukaan pitää ASCII
                        numeroa vähentää 30H jotta saadaan kokonaisluku */

    if(val2_kkuu<2&&val2_kkuu>=0) /* Testataan onko annettu luku järkevä ja ei
                                    esim. kirjain */
    {
        kkk=val2_kkuu; /* Laitetaan saatu kymmenet kuukaudet käyttöön */
        RS_lahetys(muuttuja); /* Lähettää sarjaportilla "OK" */
        TI=0; /* Nollaa sarjaportin TI lipun */
        ptaso=1; /* Siirtyy seuraavasta sarjaportilla lähetetystä
                    merkistä kuukausien kyselyyn */
        val2_kkuu=77; /* Muuttujaan jokin ei käytössä olevaluku, jotta
                    ei sotke seuraavaa kkk kyselyä */
        RS_lahetys(val2_kuuEKA); /* Lähettää sarjaportilla seuraavan eli
                    kuukauden kyselyn */
        while(TI==0); /* Odottaa että kaikki merkit on lähetetty */
    }

    else /* Näyttää virheilmoituksen, jos syötetty */
    { /* merkki ei ole sopiva */

        if(vikailmkkuu==2) /* if-lauseella estetään virheilmoituksen
                            näyttämisen ensimmäisellä kerralla */
        {
            RS_lahetys(val2_kkuuv); /* Lähettää virheilmoituksen sarjaportilla */
            while(TI==0);
            RS_lahetys(val2_kku); /* Kysyy kymmenet kuukaudet uudelleen */
            while(TI==0);
        }
    }
    vikailmkkuu=2;
}

/*****/

```



## LIITE E/6

```

/*****/
void va2_kuu()          /* Tämä aliohjelma suorittaa kuukauden muuttamisen */
{
    char val2_ku[28]="\n\rAnna kuukausi>";
    char val2_kuuv[35]="\r\nKirjoita oikein !!! >";
    char val2_kvuoEKA[22]="\r\nAnna vuosikymmen>";
    int val2_kuu;

    vikailmkkuu=0;
    va2m=0;           /* Seuraavalla kerralla kysytään
                      kymmenet vuodet */
    val2_kuu=SBUF-0x30; /* Sarjaportista tulee ASCII merkkejä.
                      ASCII taulukosta näkee muunnoksen,
                      jonka mukaan ASCII numeroa pitää
                      vähentää 30H jotta saadaan kokonaisluku */

    if(val2_kuu<10&&val2_kuu>=0&&kkk+val2_kuu>0&&kkk*10+val2_kuu<13)
        /* Testataan onko annettu luku järkevä ja ei esim. kirjain */
    {
        kk=val2_kuu;          /* Laitetaan saadut kuukaudet käyttöön */
        kuukausi=val2_kuu;   /* Sama kuukausitieto käyttöön switch-case
                              rakenteeseen, jossa kuukausitietoa
                              muutetaan */

        RS_lahetys(muuttuja); /* Lähettää sarjaportilla "OK" */
        TI=0;                 /* Nollaa sarjaportin TI lipun */
        ptaso=2;              /* Siirtyy seuraavasta sarjaportilla
                              lähetetystä merkistä kymmenien vuosien kyselyyn */

        val2_kkuu=77;
        RS_lahetys(val2_kvuoEKA); /* Lähettää sarjaportilla seuraavan eli
                              vuosikymmenen kyselyn */

        while(TI==0);
    }
    else
    {
        if(vikailmkkuu==2)    /* Näyttää virheilmoituksen, jos syötetty
                              merkki ei ole sopiva */

        {
            RS_lahetys(val2_kuuv); /* Lähettää virhe ilmoituksen sarjaportilla */
            while(TI==0);          /* Odottaa että sarjaporttiin lähetys on
                              päättynyt */

            RS_lahetys(val2_ku);  /* Kysyy kuukauden uudelleen */
            while(TI==0);
        }
    }
    vikailmkkuu=2;
}

/*****/

```

## LIITE E/7

```

/*****/
void va2_kvuosi()          /* Kymmenien vuosien muuttaminen */
{
    char val2_kvuo[28]="\n\rAnna vuosikymmen>";
    char val2_kvuoov[35]="\r\nKirjoita oikein !!!>";
    char val2_vuoEKA[22]="\r\nAnna vuosi>";
    int val2_kvuos;

    vikailmkuu=0;        /* Estetään kuukauden virheilmoitus seuraavalla kerralla */

    val2_kvuos=SBUF-0x30; /*Sarjaportista tulee ASCII merkkejä. ASCII
                           taulukosta näkee muunnoksen */

    if(val2_kvuos<10&&val2_kvuos>=0) /* Testataan onko annettu luku järkevä ja
                                       ei esim. kirjain */
    {

        kvuosi=val2_kvuos;          /* otetaan saatu kymmenet vuodet käyttöön */
        RS_lahetys(muuttuja);      /* Lähettää sarjaportilla "OK" */
        TI=0;
        ptaso=3;                   /* Siirtyy seuraavasta sarjaportilla
                                       lähetetystä merkistä vuosien kyselyyn */

        val2_kvuos=77;
        RS_lahetys(val2_vuoEKA);   /* Kysyy vuoden */
        while(TI==0);
    }

    else
    {
        /* Näyttää virheilmoituksen, jos syötetty merkki ei ole sopiva */
        if(vikailmkvuo==2)
        {
            RS_lahetys(val2_kvuoov); /* Lähettää virheilmoituksen sarjaportilla */
            while(TI==0);
            RS_lahetys(val2_kvuo);   /* Kysyy kymmenetvuodet uudelleen */
            while(TI==0);
        }
    }
    vikailmkvuo=2;
}

/*****/

```

## LIITE E/8

```

/*****/
void va2_vuosi()                /* Vuosien muuttaminen */
{
    char val2_vuo[18]="\n\rAnna vuosi>";
    char val2_vuov[30]="\r\nKirjoita oikein !!!>";
    char val2_loppu[40]="\r\nPaina jotain näppäintä jatkaaksesi>";
    int val2_vuos,karv;

    vikailmkvuo=0;             /* Estetään vuoden virheilmoitus seuraavalla kerralla */
    val2_vuos=SBUF-0x30;      /*Sarjaportista tulee ASCII merkkejä. ASCII taulukosta
                               näkee muunnoksen */

    if(val2_vuos<10&&val2_vuos>=0)        /* Testataan onko annettu luku järkevä
                                           ja ei esim. kirjain */
    {
        vuosi=val2_vuos;                 /* Laitetaan saatu vuodet käyttöön */
        RS_lahetys(muuttuja);           /* Lähettää sarjaportilla "OK" */
        TI=0;
        ptaso=6;                         /* Siirtyy aloitus valikkoon */
        val2_vuos=77;

        /* Testataan onko helmikuu ja karkaus laitettu oikein */
        if(kkk==0&&kk==2&&kpv*10+pv>28)
        {
            kpv=2;
            pv=8;
            SEVSEG=7;

            karv=kvuosi*10+vuosi;
            if(karv%4==0 && karv%100!=0 && karv!=4860 || karv%400==0)
            {
                kpv=2;
                pv=9;
            }
            RS_lahetys(val2_svirhe);
        }
        RS_date();                       /* Näyttää voimassaolevan päivämäärän */
        RS_lahetys(val2_loppu);          /* Pyytää painamaan jotain näppäintä
                                           jatkaaksesi */
    }
    else /* Näyttää virheilmoituksen, jos syötetty merkki ei ole sopiva */
    {
        if(vikailmvuo==2)
        {
            RS_lahetys(val2_vuov);       /* Lähettää virheilmoituksen
                                           sarjaportilla */

            while(TI==0);
            RS_lahetys(val2_vuo);        /* Kysyy vuoden uudelleen */
            while(TI==0);
        }
        vikailmvuo=2;
    }
}
/*****/

```

## LIITE E/9

```

/*****/
void va2_kpaivat() /* Kymmenien Päivien muuttaminen */
{
    char val2_kpai[25]="\n\rAnna kymmenet paivat>";
    char val2_kpaiv[30]="\r\nKirjoita oikein !!!>";
    char val2_paiEKA[20]="\r\nAnna paivat>";
    int val2_kpa;

    if(va2m==0) /* Näytetään tekstit vain ensimmäisellä kerralla */
    {
        RS_lahetys(Pmuuttuja);
        while(TI==0);
        RS_lahetys(val2_kpai);
        while(TI==0);
        va2m=1;
    }

    val2_kpa=SBUF-0x30; /* Sarjaportista tulee ASCII merkkejä. ASCII
                        taulukosta näkee muunnoksen, jonka mukaan ASCII
                        numeroa pitää vähentää 30H jotta saadaan kokonaisluku */

    if(val2_kpa<4&&val2_kpa>=0) /* Testataan onko annettu luku järkevä ja ei
                                esim. kirjain */
    {
        kpv=val2_kpa; /* Laitetaan saatu kymmenet päivät käyttöön */
        RS_lahetys(muuttuja); /* Lähettää sarjaportilla "OK" */
        TI=0; /* Nollaa sarjaportin TI lipun */
        ptaso=5; /* Siirtyy seuraavasta sarjaportilla
                 lähetetystä merkistä päivän kyselyyn */
        val2_kpa=77; /* Muuttujaan jokin ei käytössä olevaluku,
                    jotta ei sotke seuraavaa kkk kyselyä */
        RS_lahetys(val2_paiEKA); /* Lähettää sarjaportilla seuraavan eli päivän
                                   kyselyn */
        while(TI==0); /* Odottaa että kaikki merkit on lähetetty */
    }

    else
    {
        /* Näyttää virheilmoituksen, jos syötetty
           merkki ei ole sopiva */
        if(vikailmkpai==2) /* if-lauseella estetään virheilmoituksen
                            näyttämisen ensimmäisellä kerralla */
        {
            RS_lahetys(val2_kpaiv);
            while(TI==0);
            RS_lahetys(val2_kpai); /* Lähettää sarjaportilla
                                   virheilmoituksen*/
            while(TI==0);
        }
    }
    vikailmkpai=2;
}

/*****/

```

## LIITE E/10

```

/*****
void va2_paivat()          /* Kymmenien Päivien muuttaminen */
{
    char val2_pai[16]="\n\rAnna paivat>";
    char val2_paiv[24]="\r\nKirjoita oikein !!!>";
    char val2_kvuoEKA[28]="\r\nAnna kymmenet kuukaudet>";
    int val2_pa, karkausidea;

    va2m=0; /* Nollaksi, jotta seuraavalla kerralla kysytään kymmenet päivät */
    val2_pa=SBUF-0x30; /* Sarjaportista tulee ASCII merkkejä. ASCII
                       taulukosta näkee muunnoksen, jonka mukaan ASCII
                       numeroa pitää vähentää 30H jotta saadaan kokonaisluku */

    /* Testataan mikä kuukausi menossa, jotta tietää kuukauden pituuden */

    if(kuukausi==1||kuukausi==3||kuukausi==5||kuukausi==7||kuukausi==8||
        kuukausi==10||kuukausi==12)
    {
        /* Testataan kuukaudet joiden pituus on 31 päivää */
        if(val2_pa<10&&val2_pa>=0&&kpv*10+val2_pa<32)
        {
            paivmtest=2;
        }
    }

    if(kuukausi==4||kuukausi==6||kuukausi==9||kuukausi==11)

    {
        /* Testataan kuukaudet joiden pituus on 30 päivää */
        if(val2_pa<10&&val2_pa>=0&&kpv*10+val2_pa<31)
        {
            paivmtest=2;
        }
    }

    if(kuukausi==2) /* Testataan helmikuu ja karkauspäivä */
    {
        karkausidea=kvuosi*10+vuosi;
        if(karkausidea%4==0 && karkausidea%100!=0 && karkausidea!=4860 ||
            karkausidea%400==0)
        {
            /* Karkausvuoden määritelmä lainattu osoitteesta: */
            /* http://cs.stadia.fi/~lehtonen/Opetus/TiTe/C-OPASX.DOC*/

            if(val2_pa<10&&val2_pa>=0&&kpv*10+val2_pa<30)
            {
                paivmtest=2;
            }
        }
    }

    else
    {
        if(val2_pa<10&&val2_pa>=0&&kpv*10+val2_pa<29)
        {

```

## LIITE E/11

```

    paivmtest=2;      /* jos karkauspäivä ei ole voimassa, on helmikuun
    }                  pituus 28 päivää */
                      }

    if(val2_pa<10&&val2_pa>=0&&kpv*10+val2_pa>28&&kpv*10+val2_pa<32)
    {
        paivmtest=2; /* Koska kuukaudet ja vuodet kysytään myöhemmin */
    }                /* Annetaan mahdollisuus, ja muutos tehdään myöhemmin */
}

if(paivmtest==2) /* Testataan onko annettu luku järkevä ja ei esim. kirjain */
{
    pv=val2_pa; /* Laitetaan saatu päivät käyttöön */
    RS_lahetys(muuttuja); /* Lähettää sarjaportilla "OK" */
    TI=0; /* Nollaa sarjapotin TI lipun */
    ptaso=0; /* Siirtyy seuraavasta sarjaportilla lähetetystä
    merkistä kymmenien kuukausien kyselyyn */

    val2_pa=77; /* Muuttujaan jokin ei käytössä olevaluku, jotta
    ei sotke seuraavaa pv kyselyä */

    RS_lahetys(val2_kvuoEKA); /* Lähettää sarjaportilla seuraavan eli kymmenien
    kuukausien kyselyn */

    while(TI==0); /* Odottaa että kaikki merkit on lähetetty */
    paivmtest=0;
}

else
{
    /* Näyttää virheilmoituksen, jos syötetty merkki ei ole sopiva */
    if(vikailmpai==2) /* if-lauseella estetään virheilmoituksen
    näyttämisen ensimmäisellä kerralla */

    {
        RS_lahetys(val2_paiv); /* Lähettää virheilmoituksen sarjaportilla */
        while(TI==0);
        RS_lahetys(val2_pai);
        while(TI==0);
    }
}
vikailmpai=2;
}
/*****/

```



```

/*****/
void val_ktunti()          /* Kymmenien tuntien muuttaminen */
{

    char vall_ktu[25]="\n\rAnna kymmenet tunnit>";
    char vall_ktuv[35]="\r\nKirjoita oikein !!! (1 tai 2) >";
    char vall_tuEKA[16]="\r\nAnna tunnit>";
    int vall_ktun;

    if(valm==0)           /* Näytetään tekstit vain ensimmäisellä kerralla */
    {
        SEVSEG=5;         /* Testaukseen */
        RS_kello();       /* Lähettää sarjaporttiin kelloajan */
        RS_lahetys(vall_ktu); /* Kysyy kymmenet tunnit */
        while(TI==0);
        valm=1;
    }

    vall_ktun=SBUF-0x30;   /*Sarjaportista tulee ASCII merkkejä. ASCII
                           taulukosta näkee muunnoksen */

    if(vall_ktun<3&&vall_ktun>=0) /* Testataan onko annettu luku järkevä ja
                                     ei esim. kirjain */
    {
        SEVSEG=vall_ktun; /* Testaukseen */
        ktun=vall_ktun;   /* Laitetaan saadut kymmenet tunnit käyttöön */
        RS_lahetys(muuttuja); /* Lähettää sarjaportilla "OK" */
        TI=0;
        ktaso=1;         /* Siirtyy seuraavasta sarjaportilla
                           lähetetystä merkistä tunnin kyselyyn */

        vall_ktun=77;
        RS_lahetys(vall_tuEKA); /* Kysyy tunnit */
        while(TI==0);
    }

    else
    {
        if(vikailmkt==2)
        {
            RS_lahetys(vall_ktuv);
            while(TI==0);
        }
    }
    vikailmkt=2;
}

/*****/

```



## LIITE E/14

```

/*****/
void val_tunti()                               /* Tunnin muuttaminen */
{

    char vall_tu[15]="\r\nAnna tunnit>";
    char vall_tuv[42]=" Valitse oikein !!!\n\r";
    char vall_kmiEKA[25]="\n\rAnna kymmenet minuutit>";
    int vall_tun;

    vikailmkt=0;                               /* Estetään kymmenien tuntien vikailmoitus seuraavalla
                                                kerralla */
    valm=0;                                     /* Kymmenien tuntien kysymys näytetään myös seuraavalla
                                                kerralla */
    vall_tun=SBUF-0x30; /* Sarjaportista tulee ASCII merkkejä. ASCII
                        taulukosta näkee muunnoksen */

    if(vall_tun<10&&vall_tun>=0&&vall_tun+ktun*10<25)
        /* Testataan onko annettu luku järkevä ja ei esim. kirjain */
        {
            SEVSEG=vall_tun;                   /* Testausta */
            tun=vall_tun;                       /* Tunnit käyttöön */
            RS_lahetys(muuttuja);              /* Lähettää sarjaportilla "OK" */
            TI=0;
            ktaso=2;                            /* Siirtyy seuraavasta sarjaportilla lähetetystä
                                                merkistä kymmenien minuuttien kyselyyn */
            vall_tun=77;                       /* Jokin muu kuin ehdon täyttävä luku */
            RS_lahetys(vall_kmiEKA); /* Kysyy kymmenet minuutit */
            while(TI==0);
        }
    else
        { /* Näyttää virheilmoituksen, jos syötetty merkki ei ole sopiva */

            if(vikailmt==2)                    /* Poistetaan ensimmäisellä kerralla tuleva
                                                vikailmoitus */
                {
                    RS_lahetys(vall_tuv);      /* Lähettää virheilmoituksen */
                    while(TI==0);
                    RS_lahetys(vall_tu);      /* Kysyy tunnit uudelleen */
                    while(TI==0);
                }
            vikailmt=2;
        }
}

/*****/

```

## LIITE E/15

```

/*****/
void vall_kmin() /* Kymmenien minuuttien muuttaminen */
{

char vall_kmi[25]="\n\rAnna kymmenet minuutit>";
char vall_kmiv[52]="Annoit huonon valinnan. kokeile uudestaan";
char vall_miEKA[25]="\n\rAnna minuutit>";
int vall_kmin;

vikailmt=0; /* Estetään tuntien vikailmoitus seuraavalla kerralla*/
vall_kmin=SBUF-0x30; /* Sarjaportista tulee ASCII merkkejä. ASCII taulukosta
näkee muunnoksen */

if(vall_kmin<6&&vall_kmin>=0) /* Testataan onko annettu luku järkevä ja
ei esim. kirjain */
{
kmin=vall_kmin;
RS_lahetys(muuttuja); /* Lähettää sarjaportilla "OK" */
TI=0;
ktaso=3; /* Siirtyy seuraavasta sarjaportilla
lähetetystä merkistä minuutin kyselyyn */
vall_kmin=77; /* Jokin muu kuin ehdon täyttävä luku */
RS_lahetys(vall_miEKA);
while(TI==0);
}

else
{
if(vikailmkmin==2) /* Näyttää virheilmoituksen, jos
syötetty merkki ei ole sopiva */
{
RS_lahetys(vall_kmiv); /* Lähettää virheilmoituksen */
while(TI==0);
RS_lahetys(vall_kmi); /* Kysyy kymmenet minuutit uudelleen */
while(TI==0);
}
}
vikailmkmin==2;
}
/*****/

```

## LIITE E/16

```

/*****/
void vall_min()          /* Minuuttien muuttaminen */
{

char vall_mi[25]="\n\rAnna minuutit>";
char vall_miv[52]="Annoit huonon valinnan. kokeile uudestaan";
char vall_kseEKA[25]="\n\rAnna kymmenet sekunnit>";
int vall_min;

vikailmkmin=0;
vall_min=SBUF-0x30; /*Sarjaportista tulee ASCII merkkejä. ASCII taulukosta
näkee muunnoksen */

if(vall_min<10&&vall_min>=0)      /* Testataan onko annettu luku järkevä ja ei
esim. kirjain */
{
min=vall_min;                    /* Otetaan minuutit käyttöön */
RS_lahetys(muuttuja);           /* Lähettää sarjaportilla "OK" */
TI=0;
ktaso=4;                         /* Siirtyy seuraavasta sarjaportilla lähetetystä
merkistä kymmenien sekuntien kyselyyn */
vall_min=77;                    /* Jokin muu kuin ehdon täyttävä luku */
RS_lahetys(vall_kseEKA);       /* Kysyy kymmenet sekunnit */
while(TI==0);
}

else
{
/* Näyttää virheilmoituksen, jos syötetty merkki
ei ole sopiva */
if(vikailmmin==2)
{
RS_lahetys(vall_miv);          /* Lähettää virheilmoituksen */
while(TI==0);
RS_lahetys(vall_mi);          /* Kysyy minuutit uudelleen */
while(TI==0);
}
}
vikailmmin=2;
}

/*****/

```

```

/*****/

void vall_ksek()          /* Kymmenien sekuntien muuttaminen */
{

    char vall_kse[25]="\n\rAnna kymmenet sekunnit>";
    char vall_ksev[52]="Annoit huonon valinnan. kokeile uudestaan";
    char vall_seEKA[25]="\n\rAnna sekunnit>";
    int vall_ksek;

    vikailmmin=0;        /* Estetään minuuttien vikailmoitus seuraavalla
                           kerralla */
    vall_ksek=SBUF-0x30; /* Sarjaportista tulee ASCII merkkejä. ASCII
                           taulukosta näkee muunnoksen */

    if(vall_ksek<6&&vall_ksek>=0) /* Testataan onko annettu luku järkevä ja ei
                                       esim. kirjain */
    {
        ksek=vall_ksek;          /* Kymmenet sekunnit käyttöön */
        RS_lahetys(muuttuja);    /* Lähettää sarjaportilla "OK" */
        TI=0;
        ktaso=5;                /* Siirtyy seuraavasta sarjaportilla lähetetystä
                                   merkistä sekunnin kyselyyn */
        vall_ksek=77;           /* Jokin muu kuin ehdon täyttävä luku */
        RS_lahetys(vall_seEKA);  /* Kysyy sekunnin */
        while(TI==0);
    }

    else /* Näyttää virheilmoituksen, jos syötetty merkki ei ole sopiva */
    {
        if(vikailmksek==2)
        {
            RS_lahetys(vall_ksev); /* Lähettää virheilmoituksen */
            while(TI==0);
            RS_lahetys(vall_kse);  /* Kysyy kymmenet sekunnit uudelleen */
            while(TI==0);
        }
        vikailmksek=2;
    }

}

/*****/

```

```

/*****/
void vall_sek()          /* Sekuntien muuttaminen */
{

    char vall_se[18]="\n\rAnna sekunnit>";
    char vall_sev[42]="Annoit huonon valinnan. kokeile uudestaan";
    char vall_loppu[31]="\r\nPaina näppäintä jatkaaksesi\r\n";
    int vall_sek;

    vikailmksek=0;      /* Estetään kymmenien sekuntien vikailmoitus
                        seuraavalla kerralla */
    vall_sek=SBUF-0x30; /*Sarjaportista tulee ASCII merkkejä. ASCII taulukosta
                        näkee muunnoksen */

    if(vall_sek<10&&vall_sek>=0) /* Testataan onko annettu luku järkevä ja
                                    ei esim. kirjain */
    {
        sek=vall_sek;
        RS_lahetys(muuttuja); /* Lähettää sarjaportilla "OK" */
        RS_kello();
        TI=0;
        ktaso=6;           /* Siirtyy seuraavasta sarjaportilla
                            lähetetystä merkistä sekunnin kyselyyn */
        vall_sek=77;      /* Jokin muu kuin ehdon täyttävä luku */
        RS_lahetys(vall_loppu); /* Pyytää painamaan näppäintä jatkaaksesi */
    }

    else
    {
        /* Näyttää virheilmoituksen, jos syötetty merkki ei
           ole sopiva */
        if(vikailmsek==2)
        {
            RS_lahetys(vall_sev); /* Lähettää virheilmoituksen */
            while(TI==0);
            RS_lahetys(vall_se); /* Kysyy sekunnin uudelleen */
            while(TI==0);
        }
    }
    vikailmsek=2;
}

/*****/

```

## LIITE E/19

```

/*****/
void valikkovalinta()      /* Lähettää sarjaporttia pitkin valikkotekstit */
{
    char Kvlikko[30]="\n\rValitse:\n\n\r[k] KELLO"; /*Valikon tekstit */
    char Pvlikko[30]="\r\n[p] PAIVAMAARA\n\n\n\r";

    RS_lahetys(Kvlikko);
    while(TI==0);
    RS_lahetys(Pvlikko);
    while(TI==0);
}

/*****/

/*****/
void RS_lahetys(char hyper[98])      /* Tällä lähetetään merkit
                                      sarjaportinkautta eteenpäin */
{
    /* Tulevat merkit käydään läpi
       merkkikerrallaan */

    for(i=0;i<strlen(hyper);i++)      /* siirtämässä sarjaportin SBUF
                                       rekisteriin */
    {
        TI=0;
        SBUF=hyper[i];
        while(TI==0);
    }
}

/*****/

```

## LIITE E/20

```

/*****/

void valikkoH()          /* Tämä aliohjelma vertaa switch-case menetelmällä */
{                       /* mitä kelloaikaa ollaan kysymässä */

    switch(ktaso)
    {
        case 0:          /* Muutetaan kymmenet tunnit */
            {
                val_ktunti();
                break;
            }

        case 1:          /* Muutetaan tunnit */
            {
                val_tunti();
                break;
            }

        case 2:          /* Muutetaan kymmenet minuutit */
            {
                vall_kmin();
                break;
            }

        case 3:
            {
                vall_min();          /* Muutetaan minuutit */
                break;
            }

        case 4:
            {
                vall_ksek();          /* Muutetaan kymmenet sekunnit */
                break;
            }

        case 5:
            {
                vall_sek();          /* Muutetaan sekunnit */
                break;
            }

        case 6:
            {
                test='t';          /* Poistutaan Kelloajan muutos luupista */
                ktaso=0;          /* Ajanmuutosvalikko lähtee seuraavan
                                   kerran ktunneista liikkeelle */
                vikailmse=0;      /* Estetään sekuntien virheilmoitus
                                   seuraavalla kerralla */
                RS_lahetys(merkkijono1);
                valikkovalinta();          /* Näytetään Alkuvalikko */
                break;
            }
    }
}

/*****/

```

## LIITE E/21

```

/*****/
interrupt void EX0_int()          /* Keskeytys INTO */
{
    /* Keskeytys INTO, menee aina kaiken
       edelle suoritusjärjestyksessä */
    pulssi=1;                    /* Pulssi muuttuja asetetaan 1, kun keskeytys tapahtuu */
    aikalasku();                /* Suoritetaan aikalasku */
}
/*****/

/*****/
void tekstimuodostus()          /* Aliohjelmassa testataan onko kytkintä painettu
                               ja tämän mukaan muodostetaan haluttu merkkijono,
                               joka lähetetään tulostus 1 aliohjelmaan */

{
    int b;

    if(valinta==0)              /* Näyttää Päivämäärän ja kelloajan */
    {
        for(b=0;b<15;b++)
        {
            sprintf(merkkijono, " PVM%d%d.%d%d.20%d%d", kpv, pv, kkk, kk, kvuosi, vuosi);

            tulostusI(merkkijono);          /* Lähetä AINA 16 merkkiä*/
            /* Lähettää merkkijono taulukon
               tulostettavaksi LCD-näytölle */
        }
    }

    if(valinta==1)              /* Näyttää kelloajan */
    {
        for(b=0;b<15;b++)
        {
            sprintf(merkkijono, " kello %d%d:%d%d:%d%d ", ktun, tun, kmin, min, ksek, sek);

            tulostusI(merkkijono);          /* Lähetä AINA 16 merkkiä*/
        }
    }
}
/*****/

```



## LIITE E/22

```

/*****/
void aikalasku()          /* Laskee kelloajan saadusta pulssista. */
{                          /* Pulssi lisätään sekunneiksi, sekunti
                           kymmeniksi sekunneiksi */

    if(pulssi==1)         /* kymmenet sekunnit minuteiksi... */
    {
        sek=sek+1;
        pulssi=0;
        if(sek==10)
        {
            ksek=ksek+1;
            sek=0;
        }

        if(ksek==6)
        {
            min=min+1;
            ksek=0;
        }

        if(min==10)
        {
            kmin=kmin+1;
            min=0;
        }

        if(kmin==6)
        {
            tun=tun+1;
            kmin=0;
        }

        if(tun==10)
        {
            ktun=ktun+1;
            tun=0;
        }

        if(ktun==2&tun==4)
        {
            ktun=0;
            tun=0;
            paivays();
        }

    }

    pulssi=0;
    LED=0;
}

/*****/

```

```

/*****
interrupt void SCON_int (void) /*Sarjaportin keskeytys*/
{
    if(RI==1) /*onko puskurissa uudet 8-bittiä*/
    {
        if(RI==1)
        {
            if(SBUF==0x4b) /*jos merkki on 'K' */
            {
                test='h';
            }

            if(SBUF==0x6b) /*jos merkki on 'k' */
            {
                test='h';
            }

            if(SBUF==0x50) /*jos merkki on 'P' */
            {
                test='p';
            }

            if(SBUF==0x70) /*jos merkki on 'p' */
            {
                test='p';
            }

            hk=1; /* Jos sarjaportin keskeytys johtuu saapuvasta datasta on hk
                muuttuja 1 */
        }

        TI=0; /* sarjaportin lähetyksen keskeytyksen nollaus */
        RI=0; /* sarjaportin vastaanoton keskeytyksen nollaus */
    }
}

*****/
```

## LIITE E/24

```
/******  
void paivays()          /* Tämä aliohjelma tutkii mikä kuukausi on menossa */  
{                    /* ja lisää päivää kunnes kuukausi vaihtuu */  
  
    int kar;  
  
    switch(kuukausi)  
    {  
        case 1: /* Tammikuu */  
            {  
                if(kpv==3&pv==1)  
                {pv=0; pv=0; kuukausi=2;}  
                break;  
            }  
  
        case 2: /* Helmikuu */  
            {  
                kar=kvuosi*10+vuosi;  
                if(kar%4==0 && kar%100!=0 && kar!=4860 || kar%400==0)  
                    /* Karkausvuoden määritelmä lainattu osoitteesta: */  
                    /* http://cs.stadia.fi/~lehtonen/Opetus/TiTe/C-OPASX.DOC*/  
                    {  
                        if(kpv==2&pv==9)  
                            {kpv=0; pv=0; kuukausi=3; kk=3; kkk=0;}  
                    }  
                else  
                {  
                    if(kpv==2&pv==8)  
                        {kpv=0; pv=0; kuukausi=3; kk=3; kkk=0;}  
                }  
                break;  
            }  
  
        case 3: /* Maaliskuu */  
            {  
                if(kpv==3&pv==1)  
                    {kpv=0; pv=0; kuukausi=4; kk=4; kkk=0;}  
                break;  
            }  
  
        case 4: /* Huhtikuu */  
            {  
                if(kpv==3&pv==0)  
                    {kpv=0; pv=0; kuukausi=5; kk=5; kkk=0;}  
                break;  
            }  
  
        case 5: /* Toukokuu */  
            {  
                if(kpv==3&pv==1)  
                    {kpv=0; pv=0; kuukausi=6; kk=6; kkk=0;}  
                break;  
            }  
  
        case 6: /* Kesäkuu */  
            {  
                if(kpv==3&pv==0)  
                    {kpv=0; pv=0; kuukausi=7; kk=7; kkk=0;}  
                break;  
            }  
  
        case 7: /* Heinäkuu */
```

```

    {
        if(kpv==3&pv==1)
            {kpv=0; pv=0; kuukausi=8; kk=8; kkk=0;}
        break;
    }
case 8: /* Elokuu */
    {
        if(kpv==3&pv==1)
            {kpv=0; pv=0; kuukausi=9; kk=9; kkk=0;}
        break;
    }
case 9: /* Syyskuu */
    {
        if(kpv==3&pv==0)
            {kpv=0; pv=0; kuukausi=10; kk=0; kkk=1;}
        break;
    }
case 10: /* Lokakuu */
    {
        if(kpv==3&pv==1)
            {kpv=0; pv=0; kuukausi=11; kk=1; kkk=1;}
        break;
    }
case 11: /* Marraskuu */
    {
        if(kpv==3&pv==0)
            {kpv=0; pv=0; kuukausi=12; kk=2; kkk=1;}
        break;
    }
case 12: /* Joulukuu */
    {
        if(kpv==3&pv==1)
            {kpv=0; pv=0; kuukausi=1; kk=1; kkk=0; vuosi=vuosi+1;}
        break;
    }
}

pv=pv+1; /* Päivän lisäys */
if(pv==10)
    {
        pv=0;
        kpv=kpv+1;
    }

if(vuosi==10)
    {
        vuosi=0; kv=kv+1;
    }
if(kv==10)
    {
        kvuosi=kvuosi+1;
    }
}

/*****/

```

## LIITE E/26

```

/*****/
void tulostusI(char siirto [16])      /* Tämä aliohjelma suorittaa LCD-näytölle
                                     tulostuksen ja näytön rivin vaihdot */
{
    int j,ap=0;

    for(j=0;j<strlen(siirto);j++)
    {
        if(merkkilaskuri==8)        /* Näyttö on yksirivinen joka on jaettu
                                     kahteen 8 merkin osaan */
        {
            /* Tämän vuoksi täytyy tehdä ensimmäinen
               "rivinvaihto" 8 merkin kohdalla */

            LCD_ALUSTUS=0xc0;
            /* delay(1); */
            riv=riv+1;
        }
        if(merkkilaskuri==16)        /* 16 merkin kohdalla tyhjennetään
                                     koko näyttö */
        {
            /* LCD_ALUSTUS=0xc0;
               delay(1); */
            merkkilaskuri=0;
            riv=riv+1;
        }

        if(riv==2)
        {
            riv=0;
            LCD_ALUSTUS=0x02;        /* Kirjoitus aloitetaan alusta*/
            delay(1);                /* edellisen päälle. (ei välky) */
        }
    }

    merkkilaskuri=merkkilaskuri+1;

    if(siirto[j]==0xC4)
    {
        LCD_DATA=0x41;
        delay(1);
        ap=1;
    }

    if(siirto[j]==0xE4)             /*ä kirjoitetaan oikein*/
    {
        LCD_DATA=0xE1;
        delay(1);
        ap=1;
    }

    if(ap==0)                       /*itse merkin tulostus*/
    {
        LCD_DATA=siirto[j];
        delay(1);
    }
}
/*****/

```

```

/*****/
void LCD_n_alustus()          /* Tämä aliohjelma alustaa LCD-näytön */
{
    char alustus[11]={0x30,0x30,0x30,0x38,0x08,0x01,0x07,0x01,0x0F,0x06,0x01};
                                /*LCD näytön alustus data*/
    for(i=0; i<11; i++)        /* LCD näytön alustus data syötetään
                                kontrollirekisteriin tavu kerrallaan */
    {
        LCD_ALUSTUS =alustus[i];
        delay(10);
    }
}
/*****/
/*****/

void laskurin_alustus()      /* Tämä aliohjelma toimii sarjaportin kellona */
{
    PCON=0x00; /*SMOD-bitti on 0 */
    TMOD=0x21; /*laskurin TR1 (8 bit)toimintamuoto 2 ja laskuri TR0 (16
                bit) toimintamuoto 1 */
    TH1=0xf4; /*nopeus 2400 */
    TR1=1; /*timer 1:n on/off-bitti timer 1 käyntiin */
}
/*****/
/*****/

void sarjaportin_alustus()   /* Tämä aliohjelma alustaa sarjaportin */
{
    SM0=0; /*käytössä toimintamuoto 1 */
    SM1=1; /* eli 10-bitin kehys. START+8bit+STOP */
    REN=1; /* vastaanotto sallittu */
    RI=0;
    SBUF=0;
}

/*****/
/*****/

void delay(int aika)         /* Tämä aliohjelma toimii viiveenä */
{
    int i;
    for(i=0; i<aika; i++)
    {
        TLO = INIT_TLO; /* laskurin 0 alustus */
        TH0 = INIT_TH0; /* arvot voidaan asettaa alkumäärittelyssä */
        TR0 = 1; /* laskurin 0 käynnistys */

        while(TF0==0); /* tutkitaan laskurin 0 ylivuotolippua
                        milloin se asettuu ykköseksi */
        TR0 = 0; /* laskurin 0 pysäytys */
        TF0 =0; /* keskeytyslipun nollaus */
    }
}

/*****/
/*****/
LOPPU
/*****/

```