

Jere Väisänen

Näyttölaitteen ja näppäimistön liittäminen sulautettuun laitteeseen

Insinöörityö
Kajaanin ammattikorkeakoulu
Tekniikan ja liikenteen ala
Tietotekniikan koulutusohjelma
Syksy 2008



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Tietotekniikan ja liikenteen ala	Koulutusohjelma Tietotekniikka
Tekijä(t) Jere Väisänen	
Työn nimi Näyttölaitteen ja näppäimistön liittäminen sulautettuun laitteeseen	
Vaihtoehtoiset ammattiopinnot Sulautetut järjestelmät	Ohjaaja(t) Jukka Heino Mika Keränen (Elektrobit) Toimeksiantaja Elektrobit, Kajaani
Aika Syksy 2008	Sivumäärä ja liitteet 37 + 3
<p>Tässä insinööriyössä suunniteltiin ja toteutettiin näyttölaitteen ja näppäimistön liittäminen sulautettuun laitteeseen. Sulautettuna laitteena työssä toimi ATmega128-mikrokontrolleriin perustuva kehitysympäristö.</p> <p>Näyttölaitteen liittämisessä suunniteltiin näytönohjain, jonka avulla ATmega128-kontrolleri pystyi piirtämään kuvaa näyttölaitteelle. Näytönohjaimen vaatimuksena oli, että se voi tuottaa värillistä grafiikkaa. Näytönohjain pyrittiin suunnittelemaan siten, että sitä voi ohjata lähes jokaisella perusmikrokontrollerilla. Näytönohjaimen ytimenä toimi Epsonin S1D13506-näytönohjainkontrolleri. Epsonin näytönohjainkontrolleri on hyvin monipuolinen toiminnoiltaan, ja se sopi hyvin työn tarkoitukseen. Piiri pystyi tuottamaan VGA-tasoisista grafiikkaa, ja siihen pystyi liittämään monitorin, television ja LCD-näytön.</p> <p>Näppäimistön liittämisessä suunniteltiin PS/2-näppäimistön ja ATmega128-kontrollerin välille signaalin sovituskortti. Sovituskortin tehtävänä on ottaa vastaan näppäimistön data ja siirtää se ATmegan mikrokontrolleriin. Signaalin sovituskortin ja ATmegan välisessä yhteydessä käytettiin sarjaporttia. Signaalin sovituskortin avulla voidaan tunnistaa ohjelmallisesti jokainen eri näppäimistön painallus ja painalluksen keskeytys. Lisäksi sillä voidaan tunnistaa useamman näppäimen samanaikainen painallus.</p> <p>Näytönohjain ja signaalin sovituskortti testattiin ATmega128:aan pohjautuvan kehitysympäristön avulla. Näytönohjaimen testauksessa monitorin kuvaruudulle piirrettiin neliö, joka koostui monesta eri väristä. Signaalin sovituskortin testauksessa tutkittiin, saako kehitysympäristö näppäimistön paikkatiedon sarjaportin kautta. Testauksissa todettiin kumpikin laite toimivaksi.</p> <p>Näyttölaitteen ja näppäimistön liittämisessä saavutettiin asetetut tavoitteet. Työssä opittiin sulautetun laitteen suunnittelua niin elektroniikan kuin ohjelmiston osalta.</p>	
Kieli	Suomi
Asiasanat	Näytönohjain, PS/2-näppäimistö
Säilytyspaikka	<input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School School of Engineering	Degree Programme Information Technology
Author(s) Jere Väisänen	
Title Connecting a Display Unit and Keyboard to Embedded Systems	
Optional Professional Studies Embedded Systems	Instructor(s) Jukka Heino Mika Keränen (Elektrobit)
	Commissioned by Elektrobit, Kajaani
Date Autumn 2008	Total Number of Pages and Appendices 37 + 3
<p>The purpose of this Bachelor's thesis was to design and construct a way of connecting a display unit and keyboard to embedded systems. The embedded system in this thesis was a development board based on the ATmega128 microcontroller.</p> <p>The designing of the display's connection method also included the designing of a display driver. The function of the driver was to draw a picture at the display unit together with the Atmel microcontroller. In the design of the display driver the aspiration was to enable connectivity with almost any microcontroller. The display driver was designed around the S1D13506 display controller which is manufactured by Epson. The Epson display controller is very versatile and it suited for the thesis. The display controller was able to produce VGA level graphics. It was compatible with monitors, televisions and LCD displays.</p> <p>The connection for the keyboard required a signal adaptation card which the keyboard and Atmel microcontroller are connected to. The adapter card receives the data from the keyboard and changes it the way so that it corresponds with the signal of the serial port. The signal adaptation card made it possible to identify if a button has been pressed and if the pressing discontinued. It also gave a means to identify if several buttons have been pressed at the same time.</p> <p>The goals in the connection of the display and keyboard were reached. The project taught how the electronics and software of embedded systems are designed.</p>	
Language of Thesis	Finnish
Keywords	Display driver, PS/2-keyboard
Deposited at	<input checked="" type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

ALKUSANAT

Tämä insinöörityö on tehty EB:n (Elektrobitin) Kajaanin yksikölle. Haluan kiittää kaikkia EB:n henkilöitä, jotka olivat mukana työssäni. Erityisesti haluan kiittää Mika Kerästä, joka toimi insinöörityöni ohjaajana.

Haluan kiittää Kajaanin ammattikorkeakoulun henkilöstöstä insinöörityöni valvojaa Jukka Heinoa hyvistä neuvoista. Lisäksi kiitokset menevät Eero Soiniselle insinöörityön kielellisestä ohjauksesta ja Kaisu Korhoselle englannin kielen abstraktin ohjauksesta.

SISÄLLYS

1 JOHDANTO	1
2 NÄYTTÖTEKNIIKAT	2
2.1 Bittikarttagrafiikka	2
2.2 Vektorigrafiikka	3
2.3 Näyttölaitteet	4
2.3.1 Kuvaputkitekniikka	4
2.3.2 LCD-tekniikka	5
3 NÄYTÖNOHJAIMEN SUUNNITTELU	7
3.1 Näytönohjainpiirin valinta	7
3.2 Elektroniikan suunnittelu	9
3.3 Piirilevyn suunnittelu	17
4 NÄYTÖNOHJAIMEN TESTAUS	20
4.1 Näytönohjaimen toiminta	20
4.2 Testausympäristö	21
4.3 Lopputestaus	23
5 SARJAMUOTOINEN LIIKENNE	24
5.1 Sarjaliikenne	24
5.2 Synkronointi	24
6 PS/2-NÄPPÄIMISTÖ	28
6.1 Elektroniikan suunnittelu	30
6.2 Piirilevyn suunnittelu	31
7 NÄPPÄIMISTÖN TESTAUS	32
8 TYÖN ANALYSOINTI	35
9 YHTEENVETO	36
LÄHTEET	37
LIITTEET	

KÄYTETYT TERMIT JA LYHENTEET

A0 – A15	Address Buss In ATmega
AB0 – AB20	System Address Buss
AB_DB0 – AB_DB7	Address and Data signal
ALE	Address Latch Enable
ANALOG_GND	Analog Ground
ASCII	American Standard Code for Information Interchange
BPS	Bir per Second
BUSCLK	System Bus Clock
CLK1	Input Clock 1
CLK2	Input Clock 2
CS#	Chip Select
CGA	Colour Graphics Adapter
CRT	Cathode Ray Tube
DAC	Digital to Analog Converter
DAC_VDD	Digital to Analog Converter Main Supply Voltage
DAC_VSS	Digital to Analog Converter Ground
DRAM	Dynamic Random Access Memory
DRDY	Display Enable Output
EDO	Extended Data Output
FPDAT	Panel Data Bus
FPFRAME	Frame Pulse
FPLINE	Line Pulse
FPM	Fast Page Mode
FPSHIFT	Shift Clock
HRTC	Horizontal Synchronization Signal
IREF	Current Reference for DAC
I/O	Input-Output Interface

LCAS#	Column Address for the Lower Byte
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LSB	Least Significant Bit/Byte
M/R#	Display Buffer and Register Signal
MA	Memory Address Buss
MD	Memory Data Bus
MDA	Monochrome Display Adapter
NRZI	Non-Return-to-Zero
PAL	Phase Alternate Line
PS/2	Personal System/2
RAS#	Row Address
RD0#	Memory Read Signal
RD1#	Read for the Upper Data Byte
RESET#	RESET
RGB	Red, Green, Blue (Colour Signal)
RS-232	Recommended Standard 232
RXD	Receive Data
SVGA	Super Video Graphics Array
TFT	Thin Film Transistor
TXD	Transmit Data
UART	Universal Asynchronous Receiver Transmitter
UCAS#	Column Address for the Upper byte
USB	Universal Serial Bus
VGA	Video Graphics Array
VRTC	Vertical Synchronization Signal
WAIT#	IO Channel Ready Signal
WE0#	Memory Write Signal
WE1#	Write Enable Signal for the Upper Data Byte

1 JOHDANTO

Tämä insinöörityö on tehty EB:n (Elektrobitin) Kajaanin yksikölle. EB on teknologia-alan yritys, joka on erikoistunut vaativiin sulautettuihin ohjelmisto- ja laitteistoratkaisuihin auto-teollisuudessa ja langattomissa ympäristöissä.

Elektrobitillä on monenlaisia kehitysympäristöjä, joita voidaan laajentaa lähes rajattomasti. Kehitysympäristöä hyödynnetään elektroniikan sekä ohjelmiston opetus- ja testauskäytössä. Työn tavoitteena oli laajentaa ATmega-kontrolleriin perustuvaa kehitysympäristöä elektroniikan osalta siten, että siihen pystyy liittämään näyttölaitteen ja näppäimistön.

Liitettävän näyttölaitteen vaatimuksena oli televisio tai monitori, eli kuvaruudultaan suuri näyttölaite. Näyttölaitteen liittäminen edellyttää näytönohjaimen suunnittelua kehitysympäristöön. Näytönohjaimen vaatimuksena oli, että se voi tuottaa värillistä grafiikkaa. Kehitysympäristön ytimenä on ATmega128-kontrolleri, joka sisältää erilaisia portteja. Näytönohjain on itsenäinen piirilevy, joka yhdistetään kehitysympäristön portteihin liitäntäkaapeleiden avulla.

Kehitysympäristöön liitettävän näppäimistön vaatimuksena oli, että siinä käytetään PC:n näppäimistöä. Tämä siksi, koska se on kustannustehokas ratkaisu ja niitä on helposti saatavilla. Elektroniikan osalta se merkitsee, että suunnitellaan PC:n näppäimistön signaalille sovitus kehitysympäristöön. Näppäimistön liitäntämenetelmän vaatimuksena oli yksinkertainen ja perinteinen tapa. Yhtenä vaihtoehtona alusta pitäen pidettiin sarjaporttia.

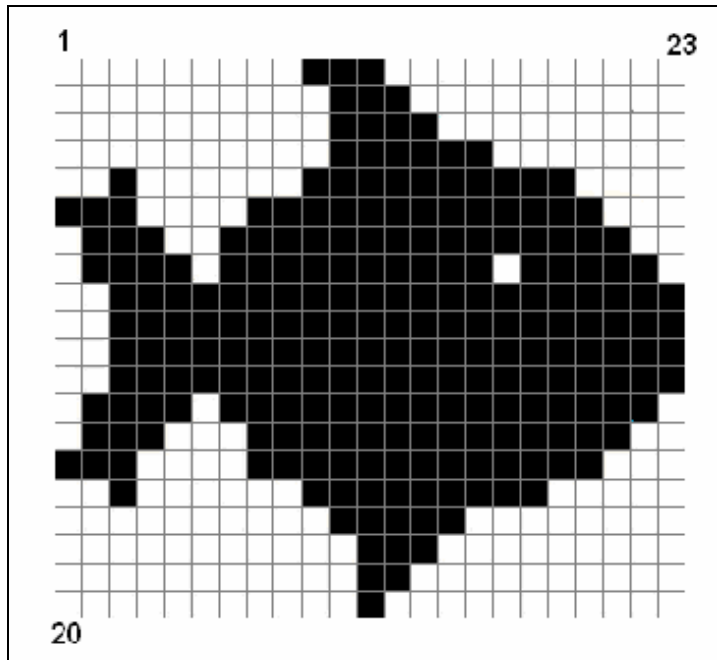
2 NÄYTTÖTEKNIIKAT

Näyttölaitteen suunnittelu aloitettiin tutkimalla eri grafiikka- ja näyttölaitemenetelmiä. Grafiikkamenetelmissä käytiin läpi perinteisimmät vaihtoehdot, kuinka kuva voidaan muodostaa. Näyttölaitemenetelmissä tutustuttiin yleisimpiin näyttötekniikoihin, joista kuvaputkitekniikka alkaa olla vanhanaikainen. Näyttötekniikoissa selvitettiin niiden toiminta, jolloin saatiin selville, kuinka niitä ohjataan ja millaisilla signaaleilla.

2.1 Bittikarttagrafiikka

Yleisin grafiikkamuoto tietotekniikassa on bittikarttagrafiikka, josta käytetään myös termejä pikseligrafiikka ja rasterigrafiikka. Bittikarttagrafiikassa kuva muodostetaan kuvapisteistä eli pikseleistä. Pikselit sijaitsevat näytössä/kuvassa aivan vierivieressä, ja niitä ohjaamalla saadaan haluttu kuva. Pikselien määrä kertoo kuvan tarkkuuden eli resoluution. Mitä suurempi resoluutio on, sen yksityiskohtaisempi kuva on. Kuvaruudun resoluutio ilmoitetaan yleensä $a \cdot b$, missä a on kuvan vaakatasossa pituussuunnassa olevan rivin pikselimäärä ja b on pystysuunnassa oleva pikselimäärä, eli kuinka monta vaakariviä kuvassa on.

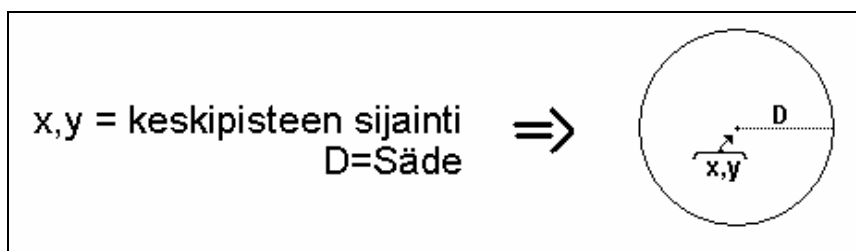
Tietotekniikassa pikseleitä ohjataan binäärisillä bittikombinaatioilla. Yksinkertaisimmillaan kuvan yhtä pikseliä voidaan ohjata yhdellä bitillä, jolloin pikseli on päällä tai pois päältä. Kuva 1 esittää yhden bitin ohjauksella saatua kuvaa. Kuvan resoluutio on 23×20 pikseliä, jossa kuvan muodostavat mustat pikselit ovat päällä ja valkeat pikselit ovat pois päältä. Yhden bitin ohjauksella ei saada aikaan kuin kaksiväritila. Jos värien määrää halutaan kasvattaa, täytyy tällöin myös ohjausbittien määrää kasvattaa. Värit muodostetaan kolmesta perusväristä punaisesta, vihreästä ja sinisestä, joita yhdistelemällä saadaan eri värejä. Ohjattavien bittien määrä kertoo saatavien värien yhdistelmien määrän korottamalla bittimäärä kakkosen potenssiin. [1.]



Kuva 1. Yhden bitin ohjauksella saatu kuva. [2.]

2.2 Vektorigrafiikka

Vektorigrafiikassa kuva esitetään matemaattisin ja geometrisin kaavoin, joita esitetään koordinaatiston avulla. Yksinkertaisimmillaan ympyrän esittämiseen ei tarvita kuin säde ja keskipisteen sijainti. Näillä tiedoilla voidaan esittää ympyrä matemaattisten kaavojen (kuva 2) avulla. Jos ympyrää halutaan tarkentaa, sille voidaan antaa lisätietoja, kuten kehän paksuus, väri, yms. Vektorigrafiikan etu bittikarttagrafiikkaan on sen erottelukyky, sillä vektorigrafiikalla piirrettyä kuvaa voidaan tarkkailla läheltä niin, että kuvan laatu ei kärsi. Toisena etuna on kuvan esittämiseen tarvitsema informaatiomäärä, joka on pienempi. [3.] Tyypillisesti vektorigrafiikkaa käytetään esimerkiksi CAD-ohjelmissa, animaatioissa ja logoissa.

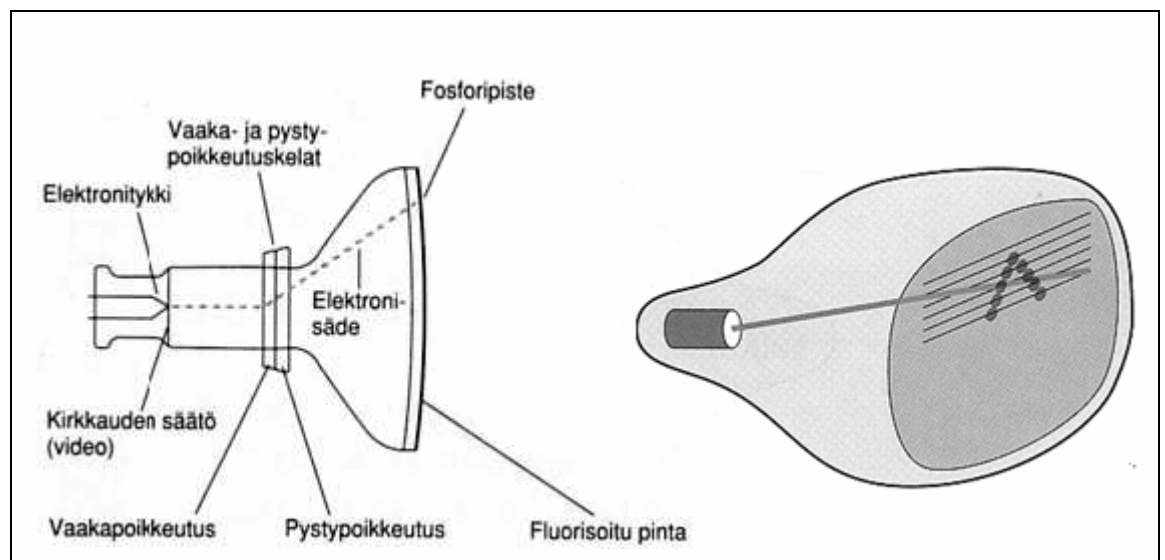


Kuva 2. Ympyrän piirtäminen vektorigrafiikalla.

2.3 Näyttölaitteet

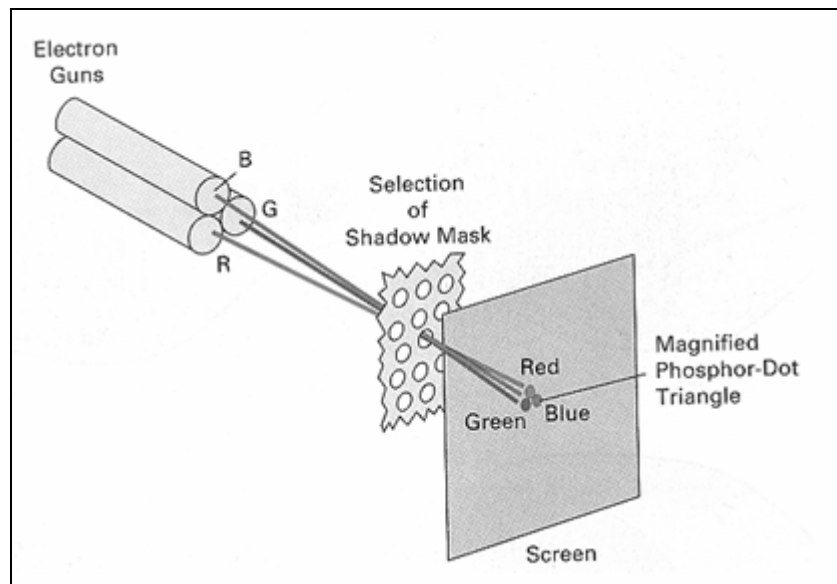
2.3.1 Kuvaputkitekniikka

CRT- eli kuvaputkinäyttölaitteet perustuvat katodisädeputkeen, joka on tyhjiöputki ja siinä on fosforoitu kuvapinta. Kuvaputken kuva muodostetaan elektrodisädejärjestelmällä, jota sanotaan myös elektronitykiksi. Elektrodisädejärjestelmä koostuu katodista ja joukosta hiloja. Katodisädeputken kaulalla ovat myös vaaka- ja pystypoikkeutuskelat (kuva 3). Katodisädeputken toisessa päässä on elektronitykki, jolla elektronisäde saadaan kuvaputken fosforipintaan suurjännitteen avulla. Tällöin elektronisuihkun kohta näkyy kirkkaana. Sädettä voidaan liikuttaa vasemmalta oikealle ja ylhäältä alas vaakapoikkeutus- ja pystypoikkeutuskelan avulla. Vaakapoikkeutus- ja pystypoikkeutuskeloja ohjataan H-sync (vaakatahdistus)- ja V-sync (pystytahdistus)-signaaleilla. Kuva muodostetaan näytölle liikuttamalla sädettä vasemmasta yläkulmasta oikeaan yläkulmaan, jolloin halutun piirtopisteen (pikselin) kohdalla säteen voimakkuutta muutetaan. Piirron jälkeen säde sammutetaan ja liikutetaan seuraavalle vaakariville. Piirrettävän kuvan muoto määritellään videosignaalin kirkkausarvoina. [4, s. 437–439.] [5.]



Kuva 3. Katodisädeputken rakenne ja kuvapisteen muodostaminen. [4, s. 437.] [5.]

CRT-putki on perustoiminnoltaan analoginen, jolloin kuvapisteen voi olla kirkkaudeltaan mitä tahansa harmaasävyä mustasta valkoiseen ja säteen piirtämä kuvapiste voi sijaita missä tahansa juovassa. Vaakasuunnan rajoitteena on kuvaputken pinnan reagoit nopeus, jolla erotetaan vierekkäiset pisteet toisistaan. Pystysuunnassa pisteitä rajoittavat juovat, sillä pisteitä voi piirtää vain juovien kohdalle. [4, s. 438.] Väriä tuottava katodisädeputki toimii hyvin pitkälti samalla tavalla kuin yllä esitetty yksiväriputki. Värit saadaan aikaan kolmella elektronitykillä ja maskimenetelmällä. Maskimenetelmässä (kuva 4) fosforipinta on korvattu fosforitäplillä, missä jokaista pikseliä kohti on kolme fosforipistettä. Fosforipisteistä jokainen lähettää eriväristä valoa (RGB). Kunkin elektronitykin säde voi osua vain tietynvärisiin fosforipisteisiin. Elektronitykin ja fosforipisteiden välissä on pieniä reikiä sisältävä levy, jonka reiät ohjaavat elektronitykkien säteet halutun pikselin vastaaviin fosforipisteisiin. Pikselin väri muodostuu fosforipisteisiin osuvien elektronisäteiden voimakkuuden mukaan. [5.] [6.]



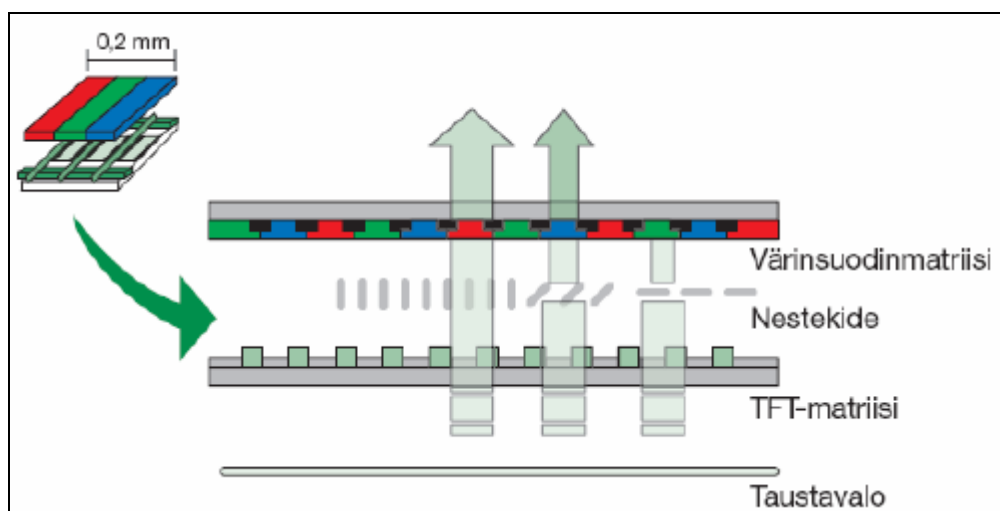
Kuva 4. Maskilevy rakenne. [5.]

2.3.2 LCD-tekniikka

LCD-näytöt perustuvat nesteessä oleviin kiteisiin, joiden valon polarisaatiota käännetään sähkökentän avulla. [6.] Kaksivärisiä LCD-näyttöjä voidaan ajatella hyvin pitkälti bittigrafiikan tapaan, missä näytön kuva muodostetaan pikseleistä. LCD-näytössä kiteet muodostavat pikseleitä, joilla on kaksi tilaa, päällä tai pois päältä. Väriillisessä TFT LCD -näytössä

pikseli on jaettu kolmeen osaan, joiden takana on värisuodatin, joka suodattaa punaista, vihreätä ja sinistä väriä. [7.] Jokaisen värin edessä on myös kide. Kiteitä ohjataan matriisi-periaatteella, missä jokaista kidettä ohjataan omalla transistorilla ja kondensaattorilla. Transistorilla ja kondensaattorilla saadaan tarkasti määritettyä kiteeseen kohdistuva sähkökenttä, jolloin kidettä voidaan kääntää halutun verran. Tällöin jokaisen kolmen perusvärin kirkkautta voidaan säätää, jolloin kolmella värillä saadaan aikaan erilaisia värejä. [6.]

Passiivimatriisinäytöissä jokaisella pikselillä ei ole omaa transistoria, vaan transistorit sijaitsevat matriisin jokaisella vaaka- ja pystyrivillä. LCD-näytön valaistus hoidetaan kiteiden takana olevalla taustavalolla tai valoa heijastavalla materiaalilla. Valoa heijastavien näyttöjen pikselien takana on peilimäinen materiaali, joka heijastaa ulkopuolisen valon takaisin päin. Heijastustekniikalla ei saada kovin suurta kontrastia, mutta tehon kulutus on pienempi. Tämän takia heijastavia näyttöjä käytetään lähinnä taskulaskimen kaltaisissa laitteissa. Kuva 5 esittää värillisen nestekidenäytön rakennetta.



Kuva 5. Värillisen LCD-näytön rakenne. [7.]

3 NÄYTÖNOHJAIMEN SUUNNITTELU

Vuonna 1981 IBM julkaisi näyttöstandardin MDA (Monochrome Display Adapter), joka oli ensimmäinen näyttöstandardi PC-mikroissa. [9.] MDA on mustavalko-ohjain, joka sisältää yksivärisen tekstiilan, eikä lainkaan grafiikkaominaisuuksia. Jokainen merkki on tallennettu ROM-piirille 8-bittisenä binäärikoodina. Jos merkkiin halutaan lisätä muuta informaatiota, esim. vilkuntaa tai väliviivaa, lisätään muistiin tälle informaatiolle 8 bitin binäärikoodi, jota kutsutaan attribuuttikoodiksi. MDA-ohjain tuottaa kuvaruudulle 720 pistettä sisältävän vaakarasterin ja 350 pistettä sisältävän pystyrasterin, eli sen resoluutio on 720 x 350 pikseliä. [4, s. 401–402.]

IBM julkaisi vuonna 1981 myös värejä tukevan näyttöstandardin CGA (Color Graphics Adapter). CGA on samankaltainen kuin MDA-ohjain, mutta poikkeaa lisäominaisuuksien puolesta edeltäjästään. CGA-ohjaimessa on suuresoluutiotila, missä kaikki pikselit ovat osoitettavissa. Lisäksi CGA tukee pienresoluutiotila, joka on alfanumeerinen. Tekstimuodossa voidaan käyttää 16:ta edustusväriä ja 8:aa taustaväriä. Kaksivärisellä grafiikkatilalla saadaan 640 x 200-resoluutio. CGA:n neliväri grafiikkatilassa saadaan resoluutio 320 x 200 ja kuuden-toista värin grafiikkatilassa resoluutio 160 x 200. [4, s. 405–406.] [9.]

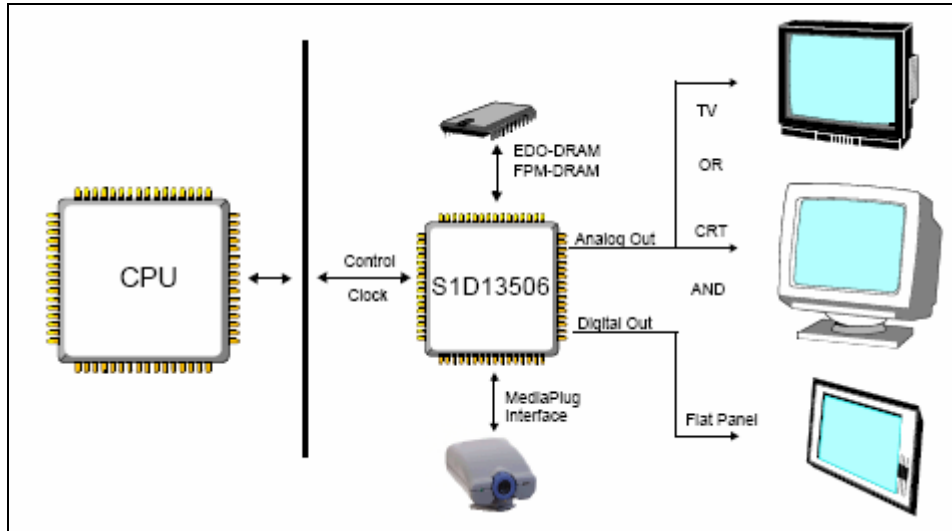
VGA (Video Graphics Array) -näyttöstandardin IBM julkaisi vuonna 1987 [9]. VGA on muistiltaan suurempi kuin edeltäjänsä, ja muistin ohjausrakenne on kehittynyt aiempiin näyttönohjausmenetelmiin verrattuna. VGA kykenee tuottamaan 256 värillä 320 x 200 resoluution ja 16 värillä 640 x 480 resoluution. [4, s. 411–412.]

SVGA (Super Video Graphics Array) sai alkunsa, kun näyttönohjainvalmistajat rupesivat parantamaan VGA-normia. [9.] SVGA-resoluutio oli aluksi 800 x 600 pikseliä, mutta näyttötekniikka kehittyi hyvin nopeasti. [4, s. 414.]

3.1 Näyttönohjainpiirin valinta

Näyttönohjainpiiriksi valittiin Epsonin S1D13506-kontrolleri. Epsonin piirin valintaan vaikutti sen nykyaikaisuus ja monipuolisuus. Toinen vaikuttava tekijä oli näyttönohjaimen kuvan päivittämisen aiheuttama kuormitus. Piirillä voidaan ohjata seuraavia näyttölaitteita: TV:tä,

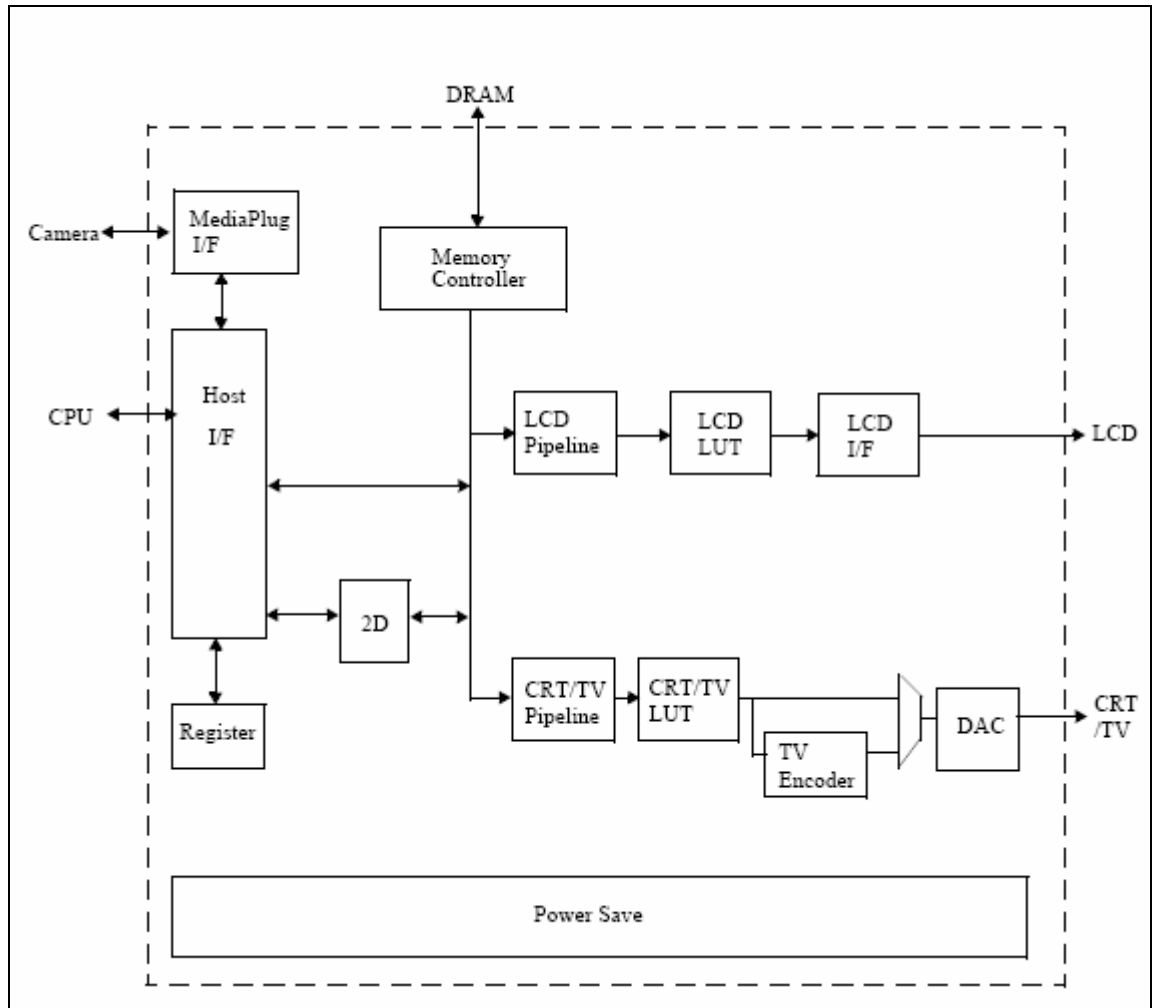
CRT-monitoria ja LCD-näyttöä. Piiri kykenee tuottamaan 800 x 600 resoluution kuvan ja 65 tuhatta väriä, kun käytössä on 16 bittiä/pikseli ohjaus. Kuva 6 esittää S1D13506-piirin mahdollistamaa järjestelmäkaaviota. [8.]



Kuva 6. S1D13506-piirin järjestelmäkaavio. [8.]

Piiriä ohjataan 21-bittisellä osoiteväylällä ja 16-bittisellä dataväylällä. Piiri sisältää oman muistikontrollerin, joka mahdollistaa muistin liittämisen suoraan piiriin. Muistikontrollerin ansiosta ATmegaan tarvitsee antaa videoinformaatiota vain näytönohjainpiirille, joka hoitaa muistin ohjaukset ja tahdistukset. Tämä pienentää kuvan päivityksen aiheuttamaa kuormitusta verrattuna siihen, että muisti olisi kytkettynä ATmegaan.

Piirillä on myös omat lähtönsä digitaaliselle ja analogiselle näyttölaitteelle, missä kummallakin on väripalettitaulukko. Analogipuolella on TV-enkooderi, joka mahdollistaa analogilähtöön TV:n liittämisen. Analogipuolella oleva DAC muuttaa digitaalisen kuva-informaation analogiseksi. DAC:n muodostama analoginen signaali on RGB, joka koostuu punaisesta, vihreästä ja sinisestä värisignaalista. RGB-signaali mahdollistaa muidenkin signaalien muodostamisen. Kuva 7 esittää S1D13506-piirin sisäistä lohkoakaaviota.

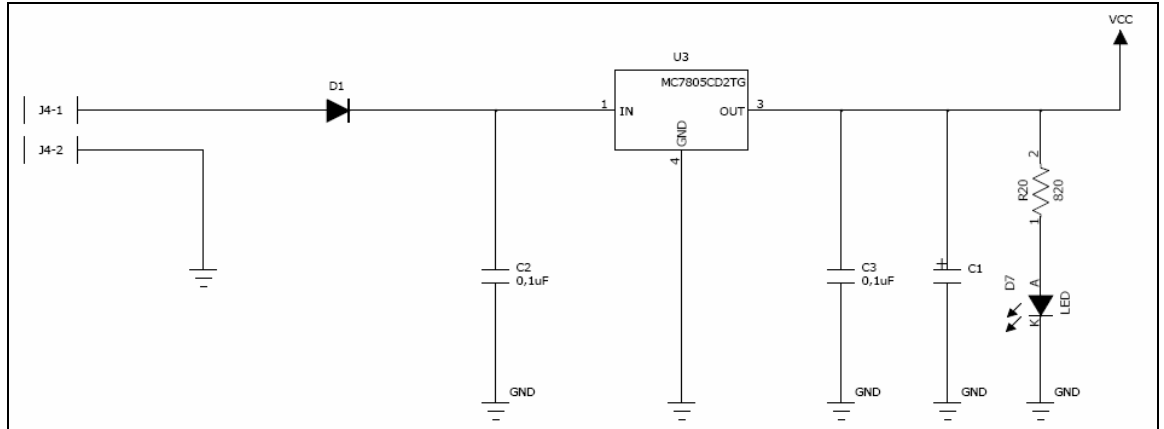


Kuva 7. S1D13506-piirin sisäinen lohkokaavio. [8.]

3.2 Elektroniikan suunnittelu

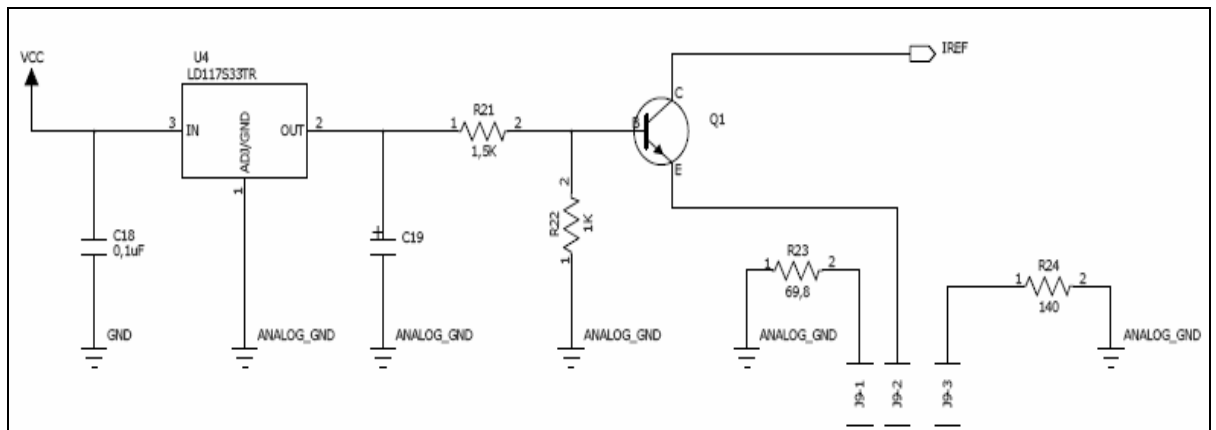
Jännitelähde

Laitteistoon kytkettävä käyttöjännite on 8–18 V, joka tuodaan 2,1 mm:n pyöreän DC-liittimen kautta. Varsinainen piirien toimintajännite on 5 V, mikä toteutetaan jännite-regulaattorin avulla (kuva 8). Regulaattorin tulopuolelle suunniteltiin diodikytkentä, jolla suojataan laitteen rikkoutuminen, jos käyttöjännite kytketään napaisuudeltaan väärin. Regulaattorin lähtöpuolella on LED, jolla nähdään laitteen olevan päällä. Käyttöjännitelinjooilla olevien piirien viereen sijoitettiin by-pass-kondensaattorit, joilla suodatettiin mahdollinen häiriö käyttöjännitteestä pois.



Kuva 8. Käyttöjännitekytkentä.

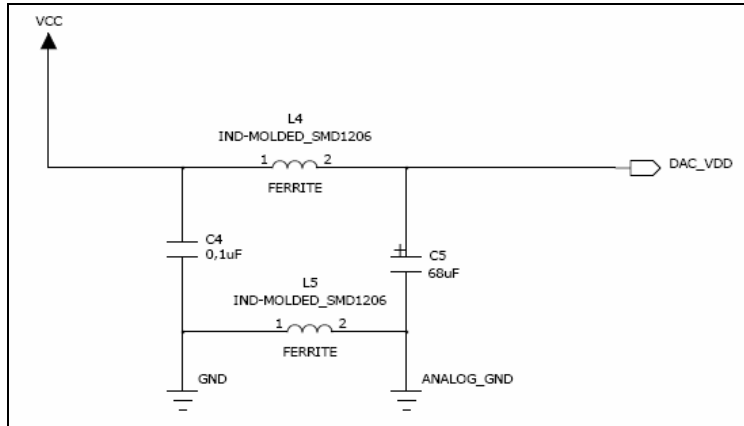
Piirikortilla on lisäksi toinen regulaattori, jolla toteutetaan 3,3 V:n jännite. Tätä jännitettä käytetään vakiovirtageneraattorin IREF-virran muodostamiseen (kuva 9). IREF-virralla muodostetaan näytönohjainpiirin sisällä olevalle DAC:lle referenssivirta. IREF:n tulee olla 9,2 mA, jos käytössä on TV-lähtö. Käyttäessään ainoastaan monitoria tulee IREF:n olla 4,6 mA. Tämän vuoksi regulaattoriksi valittiin STMicroelectronicsin LD117-sarjan regulaattori. IREF-virta valitaan liittimen J9 avulla, missä oikosulkemalla pinnit J9-1 ja J9-2 valitaan TV-lähtö käyttöön. Kun käyttöön halutaan ottaa monitori, oikosuljetaan liittimet J9-2 ja J9-3. IREF-regulointijännitteen jälkeen kaikki jännitetasot on tarkoitettu analogisille videoulostuloille, jonka vuoksi niillä on oma maapotentiaali ANALOG_GND.



Kuva 9. IREF-virran muodostaminen.

Näytönohjainpiirillä on myös DAC_VDD-jännitetuloja, jotka ovat näytönohjainpiirin analogisille signaaleille. DAC_VDD-signaaleille näytönohjainpiirissä on omat maatasot DAC_VSS, jotka on kytketty ANALOG_GND:n kanssa samaan potentiaaliin. Analogisen

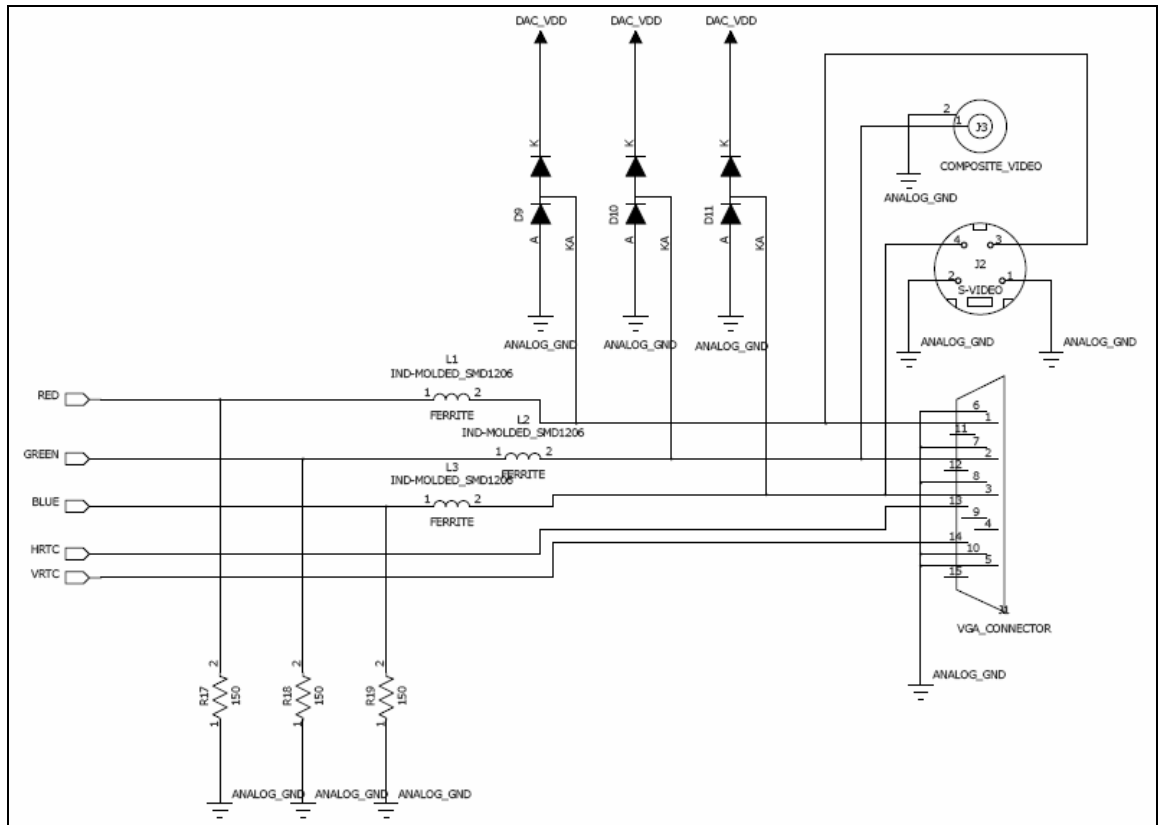
puolen jännitetasot otettiin VCC-käyttäjännitelinjasta. VCC- ja DAC_VDD-jännitteen välille suunniteltiin eromuotoisten häiriöiden varalta suodatin. Eromuotoinen häiriösuodatin (kuva 10) pystyy suodattamaan kummastakin suunnasta syntyneet häiriöt pois.



Kuva 10. Eromuotoisten häiriöiden suodatin.

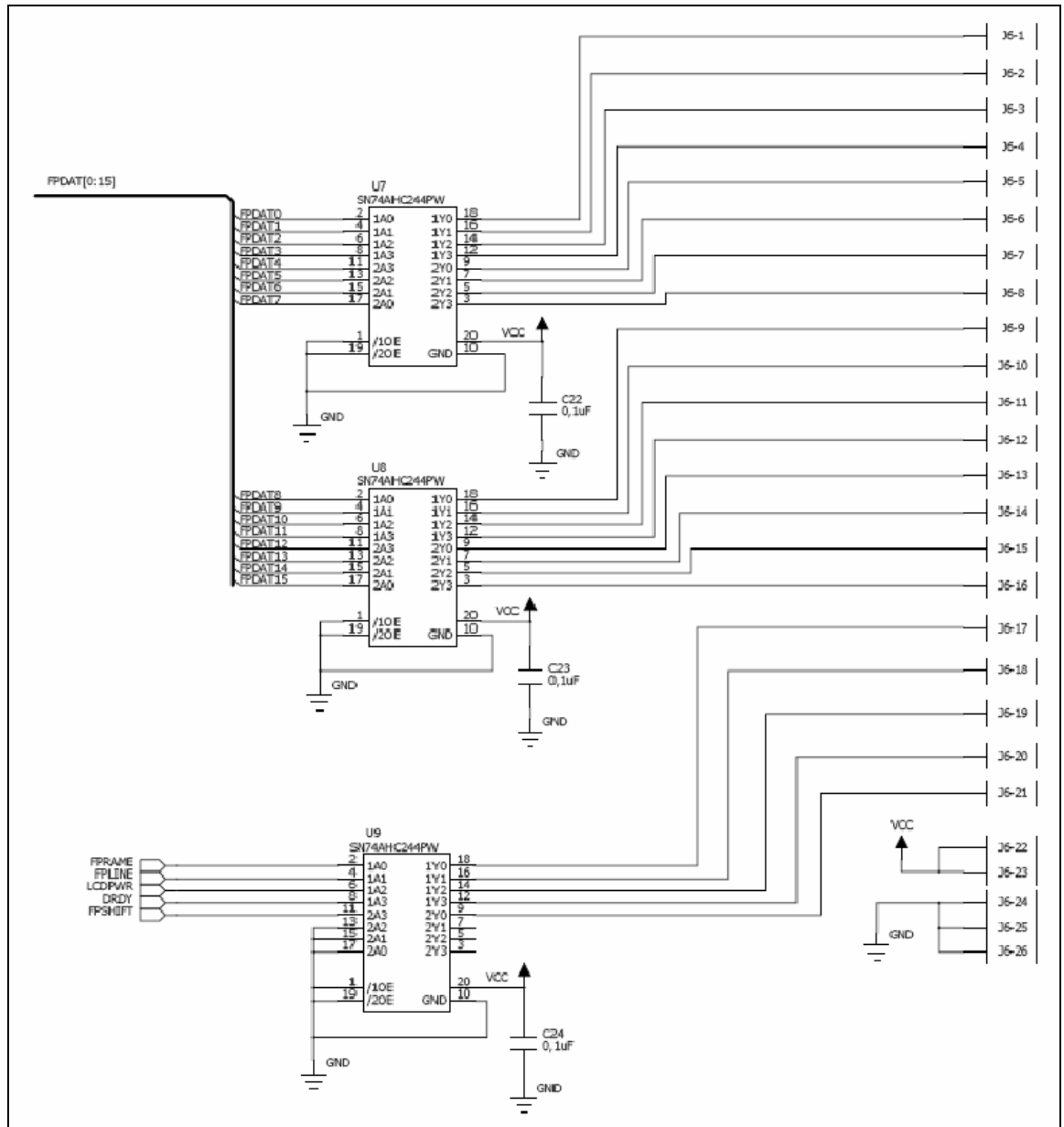
Näyttölaitteiden liitännät

Analogisia näyttölaitteita varten suunniteltiin kuvan 11 mukainen kytkentä. Kytkenässä RED, GREEN ja BLUE ovat näytönohjainpiiriltä tuleva RGB-videosignaali. RGB-signaalilinjassa olevat ferriitit toimivat häiriösuodattimina ja linjalla olevat suojadiodit toimivat mahdollisten virhejännitteiden suojoina. Kytkenässä olevat alasetovastukset R17–R19 toimivat RGB-signaalilinjan sovitustavastuksina. VGA-liitäntä koostuu RGB-, HRTC- ja VRTC-signaaleista, missä HRTC-signaali toimii kuvaruudun vaakatahdistussignaalinä ja VRTC pystytahdistussignaalinä. Komposiittisignaali muodostetaan RGB:n R-signaalista, missä on yhdistetty video- ja tahdistussignaalit yhteen. S-video-liitäntä muodostuu RGB:n R- ja G-signaalista, missä R-signaaliin on yhdistetty kirkkaus- ja tahdistussignaali. Video-signaali saadaan G-signaalista. Näytönohjainpiiriin sisällä oleva TV-enkooderi mahdollistaa RGB-linjasta signaalien yhdistämisen komposiitti- ja S-video-liitäntöille. RGB-linjalla kaikki maatasot kytkettiin analogipuolen maatasoon.



Kuva 11. Analogisten näyttölaitteiden signaaliliitännät.

LCD-näytön liittäminen on kuvattu kuvassa 12. Kytkenässä FPDAT[0:15] on dataväylä, jolla siirretään LCD-näytölle kuvainformaatio. FPSHIFT-signaalilla tahdistetaan dataväylän siirto LCD-näytölle. DRDY-signaali toimii TFT/D-TFD-näyttötyypeille sallintasiignaalina ja passiivinäyttötyypeille toisena dataväylän tahdistuskellona. FPLINE-signaali on linjapulssi, jolla ohjataan piirto seuraavalle riville. FPFRAME-signaalilla ohjataan kuvan piirtonopeus eli yhden kuvan piirtoaika. Kuvassa 12 piirit U7–U9 ovat puskuripiirejä, joilla puskuroidaan LCD-näytölle menevät signaalit. Piirien ansiota LCD-näyttö ei kuormita näyttönohjainpiiriä. Kondensaattorit C22–C24 toimivat by-pass-kondensaattoreina.



Kuva 12. LCD-näytön liittäminen.

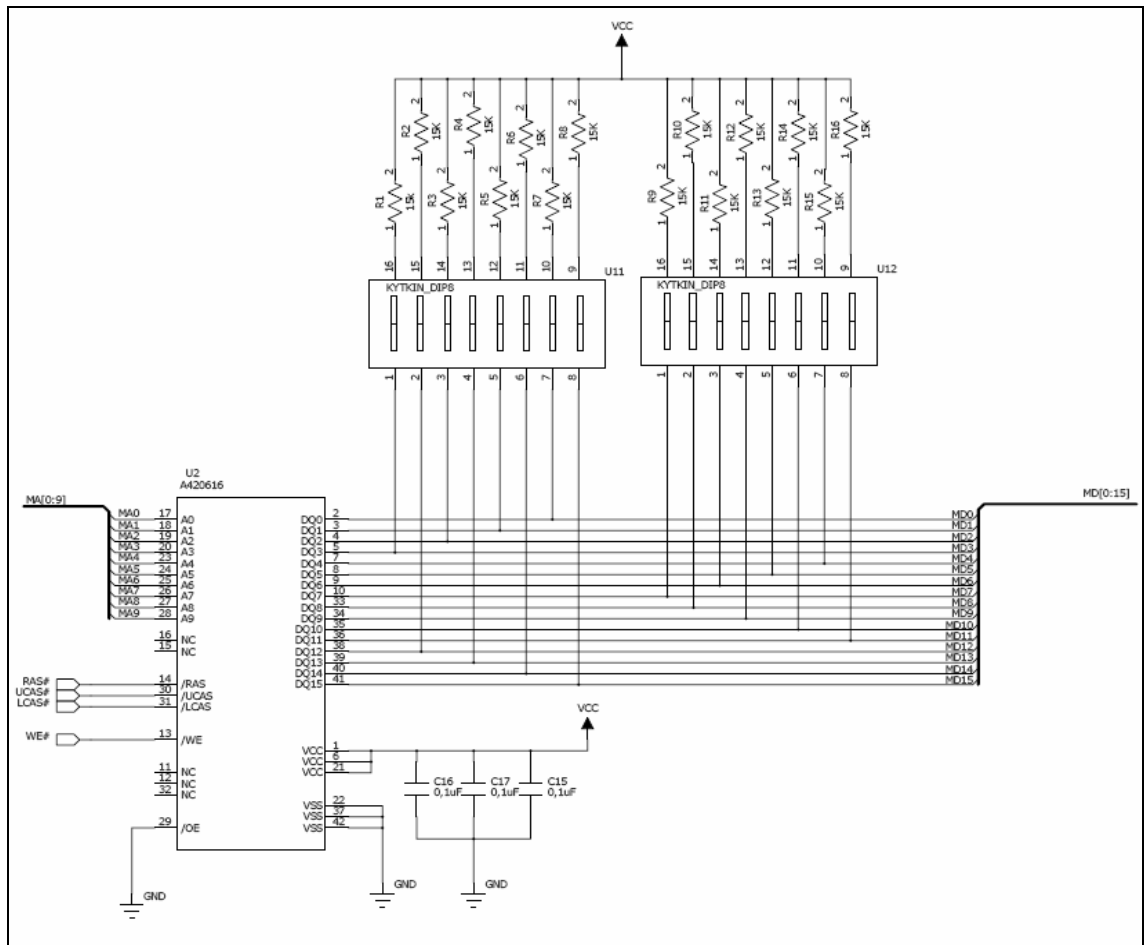
Muistin liittäminen

Näytönohjainpiiri tukee FPM- (Fast Page Mode) ja EDO- (Extended Data Output) DRAM-muisteja. Muistien datan lukeminen tapahtuu antamalla riviosoite, josta sarakeosoittimen avulla haluttu data poimitaan riviltä muistin lähtöön. Datan ollessa samalla rivillä ei riviosoitetta tarvitse kirjoittaa uudestaan, vaan data poimitaan suoraan sarakeosoittimen avulla. FPM- ja EDO-DRAM-muistit eroavat toisistaan muistin lähdön puskuroinnissa.

EDO-DRAM-muisti voi pitää datan lähdössään etsiessään uutta dataa, kun FPM-DRAM-muisti menettää datan lähdöstä aloittaessaan uuden datan etsinnän. [4, s. 263.]

Muistiksi valittiin 1M x 16-kokoinen EDO-DRAM-muisti. Kyseisen muistin haku aika oli 60 ns, jolloin näytönohjainpiirin muistikello voi olla maksimissaan 33 MHz. Muistissa MD[0:15] toimii dataväylänä ja MA[0:9] osoiteväylänä. RAS#-signaali (#-merkki tarkoittaa inverttoitua) kertoo, milloin osoiteväylältä saadaan muistipaikan riviosoitetieto. LCAS#- ja UCAS#-signaaleilla ohjataan osoiteväylältä muistipaikan sarakeosoitetieto, missä LCAS# kertoo alemman tavun ja UCAS# ylemmän tavun.

Kuva 13 esittää suunniteltua muistin kytkentää. Kytkennässä olevat vastukset toimivat ylös-
vetovastuksina. Ylösvetovastuksen ja muistin dataväylän välissä on DIP-kytkimiä, joilla valitaan ylösvetovastus toimintatilaan. DIP-kytkimillä määritetään näytönohjainpiirin toimintatiloja käynnistyksen yhteydessä. Liitteessä 1 on kuva DIP-kytkimien kytkennästä ja taulukko toimintatilojen valitsemisesta.



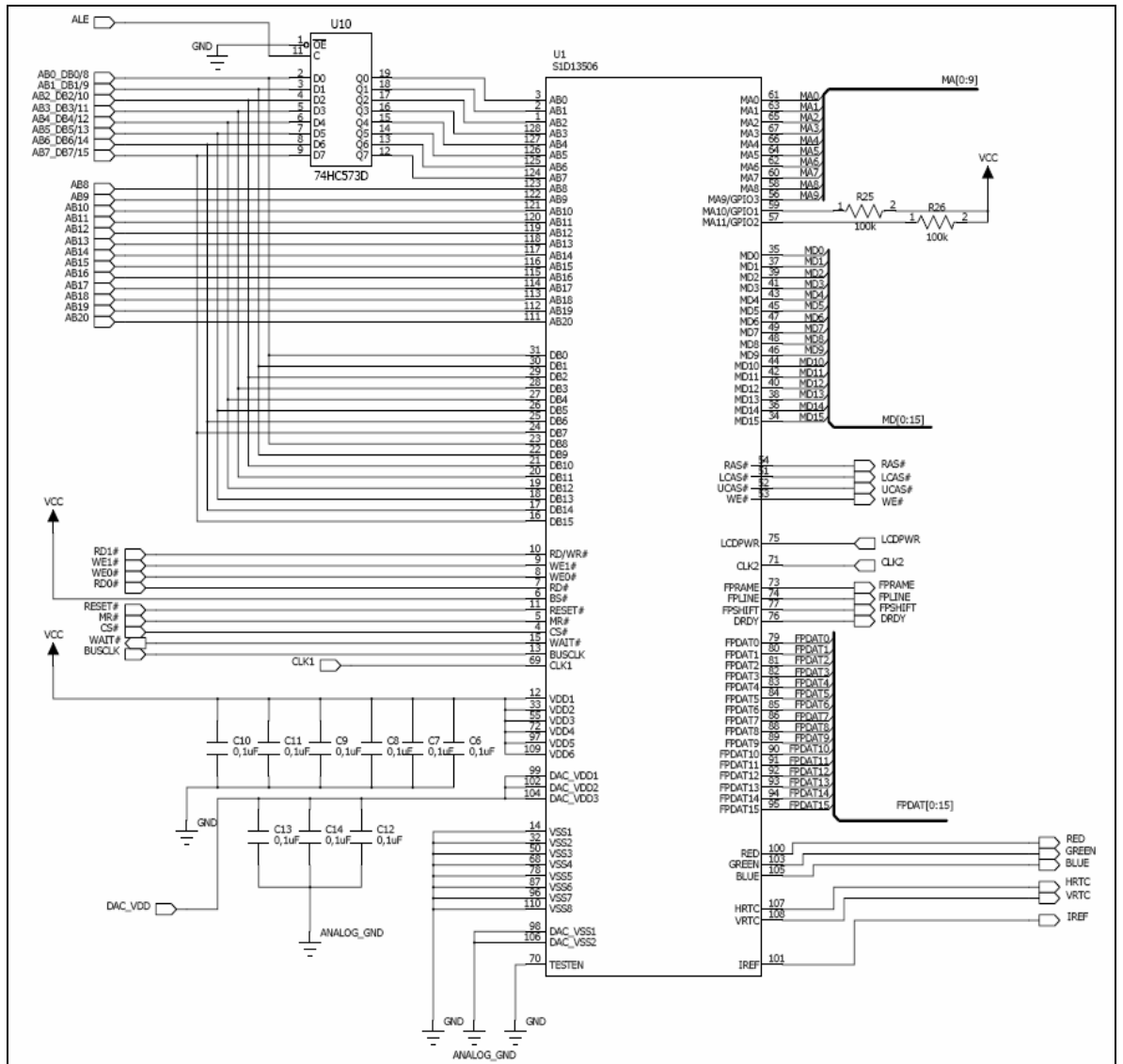
Kuva 13. Muistin liittäminen.

Näytönohjainpiirin ja ATmega128:n liittäminen

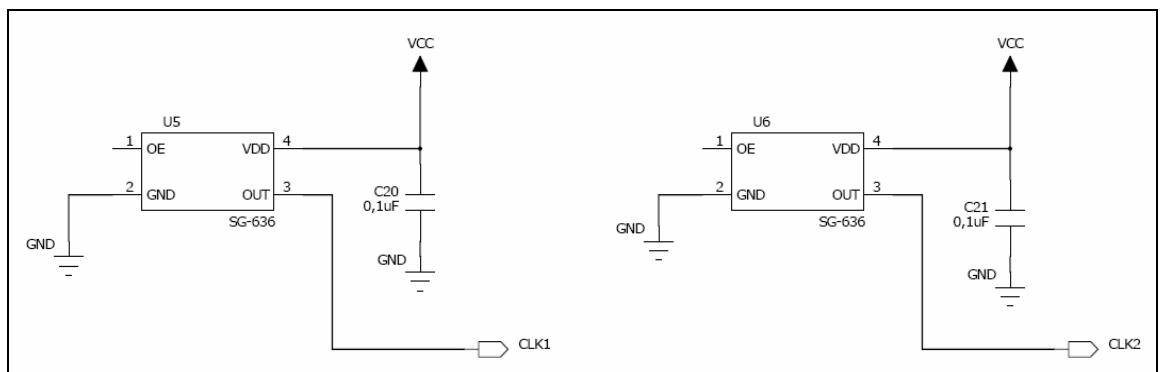
Kehitysympäristön ja näytönohjaimen liittäminen tapahtuu liitântäkaapeleiden avulla. Kuvassa 14 on esitetty Epsonin näytönohjainkontrollerin liitântärajapintaa. Kontrollerissa on 21-bittinen osoiteväylä (AB0–AB20) ja 16-bittinen dataväylä. Osoiteväylän ensimmäisistä kahdeksasta bitistä on lukittu (latchatty) dataväylä piirin 74HC573 avulla. Dataväylä on 8-bittinen, mutta näytönohjainpiiri voi käsitellä sitä 16-bittisenä. Tämä onnistuu RD0#- ja RD1#-signaalien avulla. RD0#-signaalilla valitaan, että näytönohjainpiirin dataväylältä luetaan alimmat kahdeksan bittiä ja RD1#-signaalilla valitaan dataväylän ylimmät kahdeksan bittiä. Näytönohjainpiirin kirjoittaessa dataväylälle WE0#-signaalilla valitaan dataväylälle tulevan tiedon olevan kahdeksan alinta bittiä ja WE1#-signaalilla kahdeksan ylintä bittiä.

RESET#-signaalilla resetoidaan näytönohjainpiiri. M/R#-signaalilla valitaan, ohjataanko osoiteväylällä näytönohjainpiirin sisäisiä rekistereitä vai näytön muistia. CS#-signaalilla asetetaan piiri toimintatilaan. WAIT#-signaalilla näytönohjainpiiri ilmoittaa valmiustilansa kehitysympäristössä olevalle ATmega128:lle.

BUSCLK-signaali toimii näytönohjainpiirin ja ATmega128:n tahdistuskellona. CLK1- ja CLK2-signaaleilla tuodaan näytönohjainpiirille toimintakellotaajuudet, jotka toteutettiin oskillaattorien avulla (kuva 15). Oskillaattorien taajuuksia voidaan jakaa ja ohjata ohjelmallisesti. CLK1-signalin oskillaattori toimii näytönohjainpiirin muistin tahdistuskellona sekä pikselikellona, ja sen toimintataajuus on 33 MHz. CLK2-signaalin taajuus on 17,734 MHz, jolla muodostetaan värikantaalto. Värikantaalto määräytyy käytettävän näyttölaitteen normin mukaan. Suomessa käytetään videokuvassa koodaus- ja värijärjestelmämenetelmänä PAL (Phase Alternate Line) menetelmää, jolloin värikantaalto on 4,43 MHz, joka saadaan jakamalla CLK2-signaali neljällä.

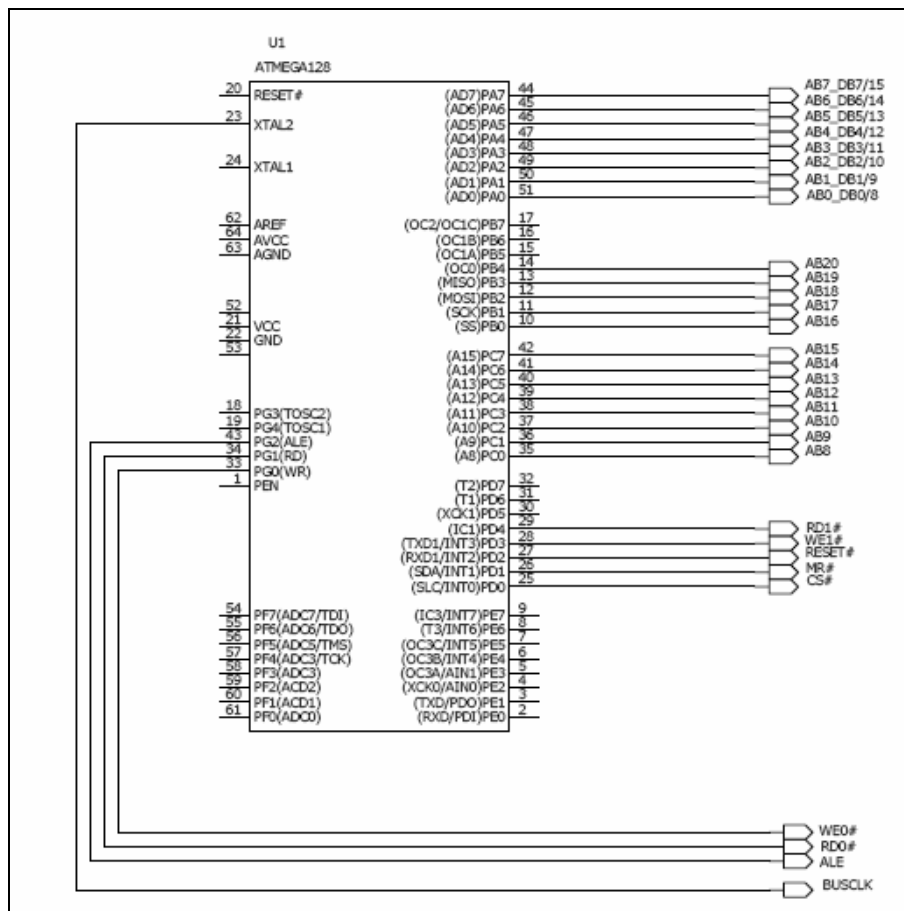


Kuva 14. Näytönohjainpiirin liitântärajapinta.



Kuva 15. Näytönohjainpiirin oskillaattorit.

Kehitysympäristö on erillinen piirikortti, jonka ytimenä toimii ATmega128-kontrolleri. ATmega128:n porteille kehitysympäristössä ovat omat liittimet, joiden avulla näytönohjainpiiri saadaan kytkettyä ATmegaan. Kuvassa 16 on ATmega128-puolen tarvittavat pinnit, johon kytketyminen tapahtuu liitântakaapeleiden avulla. Kehitysympäristön puolelta käyttöön otettiin portti A, joka on dataväylä ja osoiteväylän kahdeksan alinta bittiiä. Portit C ja B ovat osoiteväylän loppuosa. Portit G ja D antavat näytönohjainpiirille osoite- ja dataväylän ohjauskäskyt.

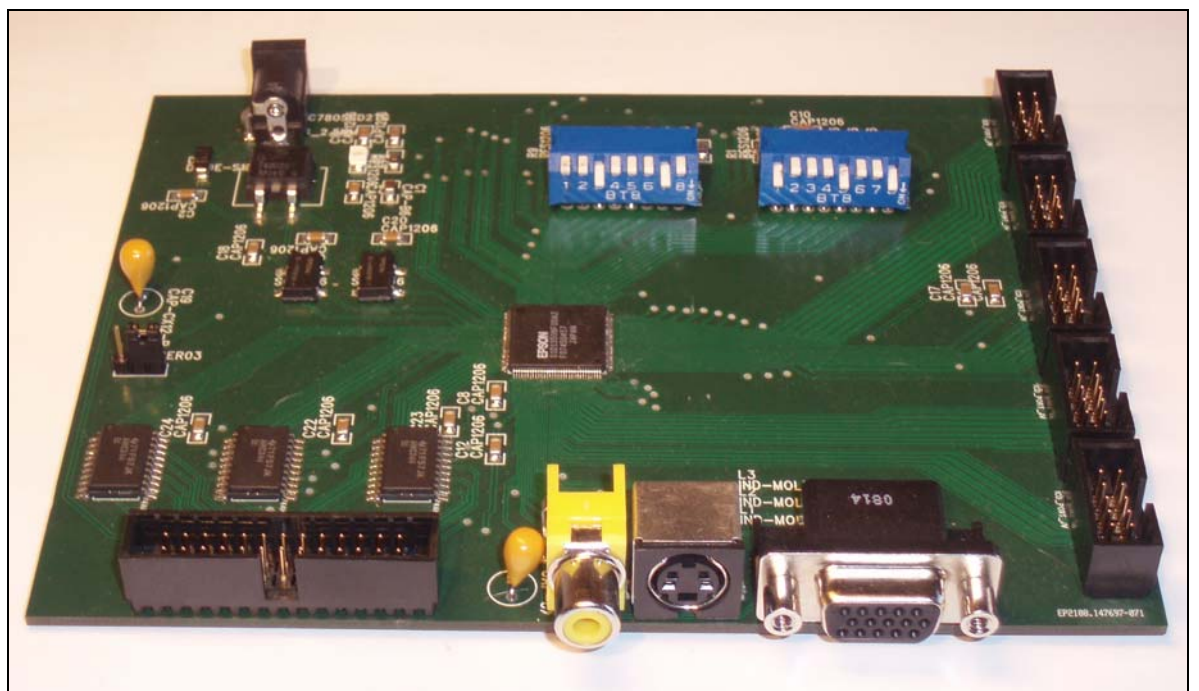


Kuva 16. ATmega128:sta tarvittavat pinnit.

3.3 Piirilevyn suunnittelu

Piirilevyn suunnittelussa käytettiin hyödyksi PADS-ohjelmaa, jolla toteutettiin komponenttien sijoittelu ja niiden reititys. Piirilevyn suunnittelussa päätettiin käyttää pääosin pintaliitoskomponentteja. Piirilevystä suunniteltiin kaksipuolinen (kuva 17), jossa komponentteja

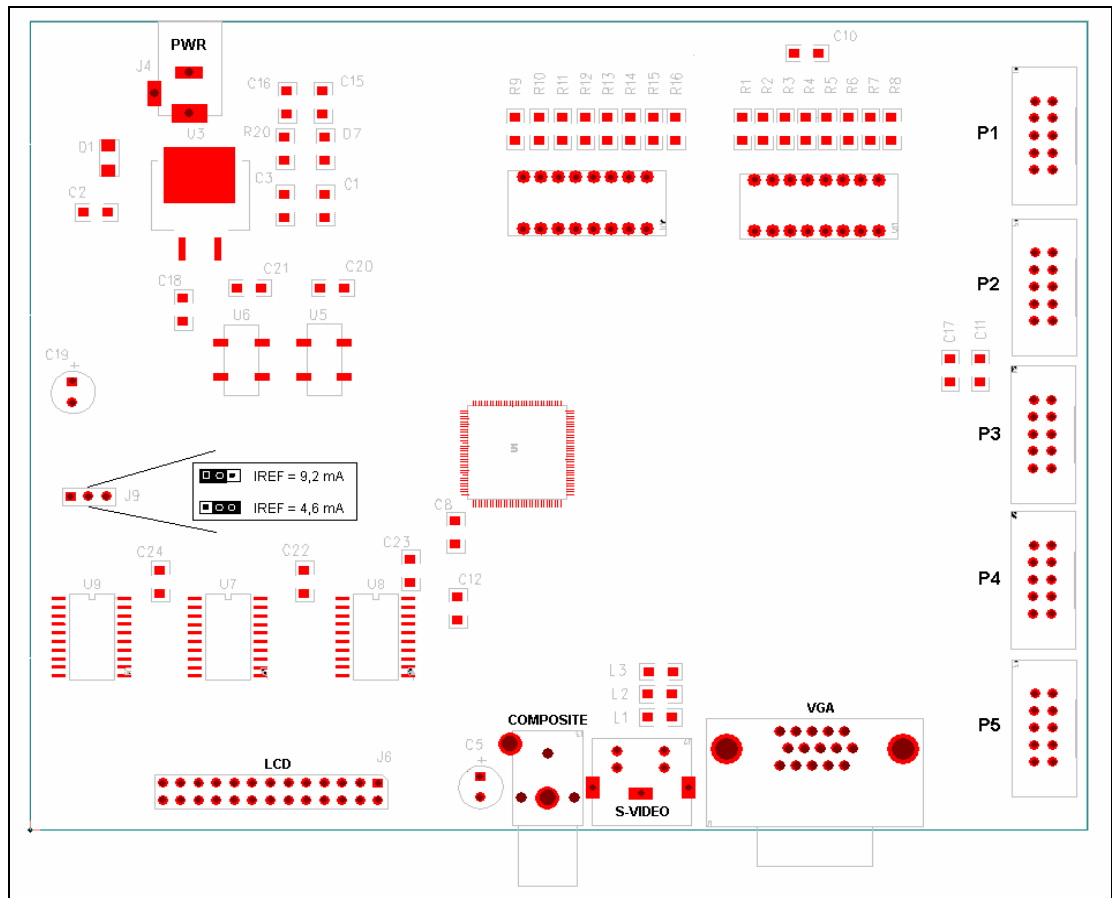
sijoitettiin molemmille puolille. Komponentit pyrittiin kuitenkin pääosin sijoittamaan piirilevyn yläpuolelle. Liitännät suunniteltiin piirilevylle mahdollisimman selkeästi, missä signaalitulot ovat omalla reunallaan ja lähdöt omalla reunalla. Komponenttien sijoittelulla oli suuri merkitys, sillä näytönohjainpiirin jalan leveys oli 0,16 mm ja jalkojen väli 0,4 mm (keskeltä keskelle). Tämän vuoksi osasta johdinvedoista tuli ohuita, jolloin johdinvedolla piti olla suora yhteys määränpäähänsä. Jalkojen fyysiset mitat antoivat valmistuksen kannalta myös haasteita, sillä ne vaikeuttivat paljon piirilevyn valmistusta ja komponenttien juotosta. Piirilevyn valmistus käytettävissä olevilla laitteilla ilmeni niin hankalaksi, että se päätettiin teettää. Piirilevyn layout-kuva on liitteessä 2.



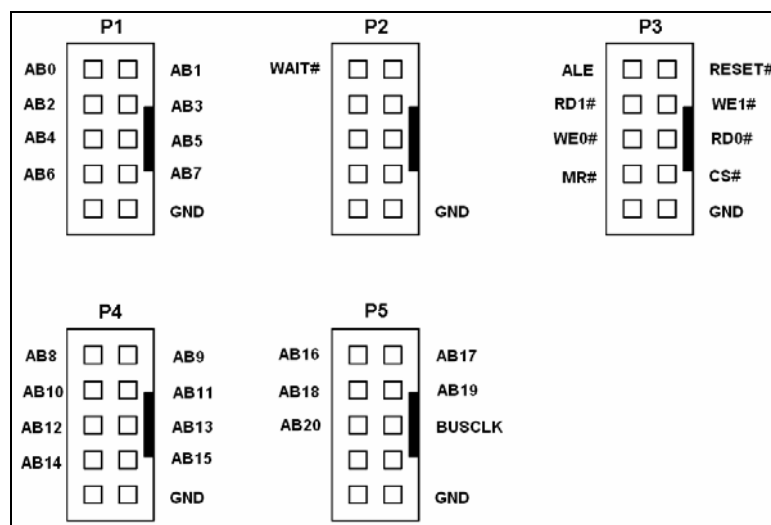
Kuva 17. Suunniteltu näytönohjin.

Maatasoja piirilevyllä on kaksi, GND ja ANALOG_GND. Piirilevyn maatasoissa käytettiin kuparointia, jossa piirilevyn ylimääräinen kuparointi toimi maadoituksena. Kuparointi toteutettiin siten, että piirilevyn yläpuoli oli GND ja alapuoli ANALOG_GND. Kuparoinnilla on hyvin suuri merkitys piirilevyn maatasoihin, sillä sen ansiosta komponenttien maapotentiaali on sama. Näytönohjin ja kehitysympäristö ovat kaksi eri piirikorttia, jonka vuoksi niiden maadoitus on hyvin tärkeä. Näytönohjaimen jokaiseen tuloliittimeen laitettiin maadoituspinni, jolloin kehitysympäristöltä voi tuoda maadoituksen näytönohjaimelle. Tämä on hyvin tärkeä tekijä häiriöiden kannalta, sillä jos kehitysympäristö ja näytönohjainkortti eivät ole samassa maapotentiaalissa, niiden 1-tilat ovat eriarvoiset. Kuvassa 18 on suunnitellun

piirilevyn komponenttien sijoittelu, josta käy ilmi liittimien nimet/sijainti, ja kuvassa 19 on selitetty tuloliittimien pinnijärjestys.



Kuva 18. Piirilevyn komponenttien sijoittelu ja liittimien sijainti.

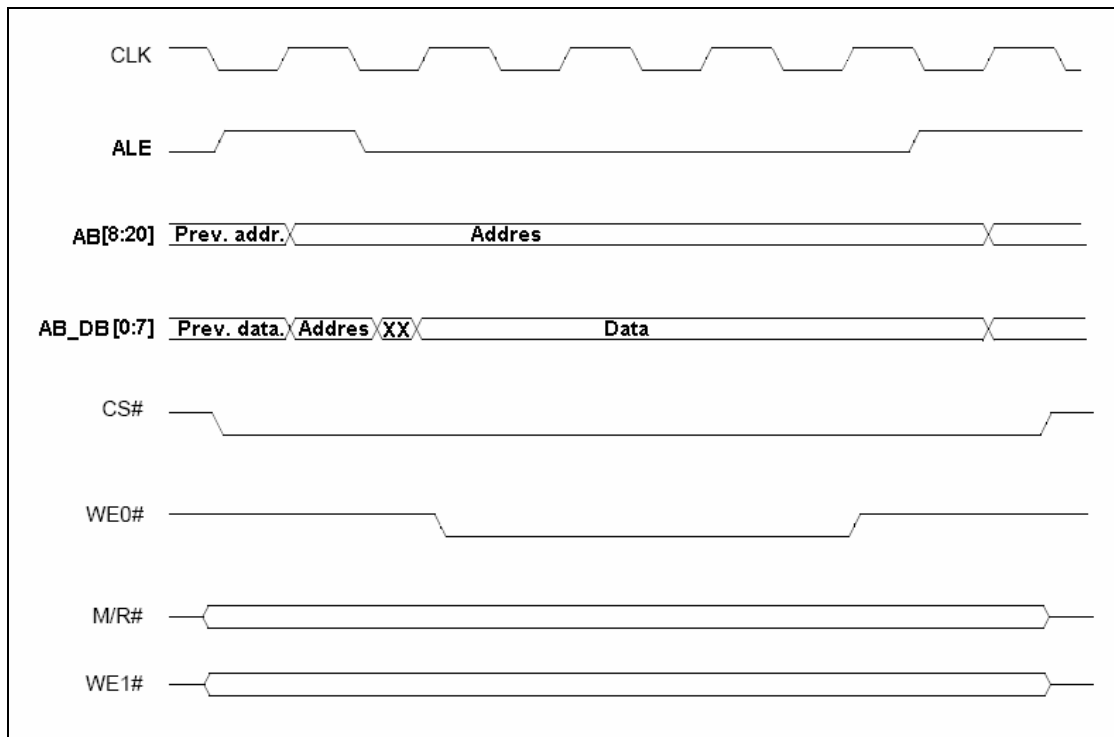


Kuva 19. Tuloliittimien pinnijärjestys.

4 NÄYTÖNOHJAIMEN TESTAUS

4.1 Näytönohjaimen toiminta

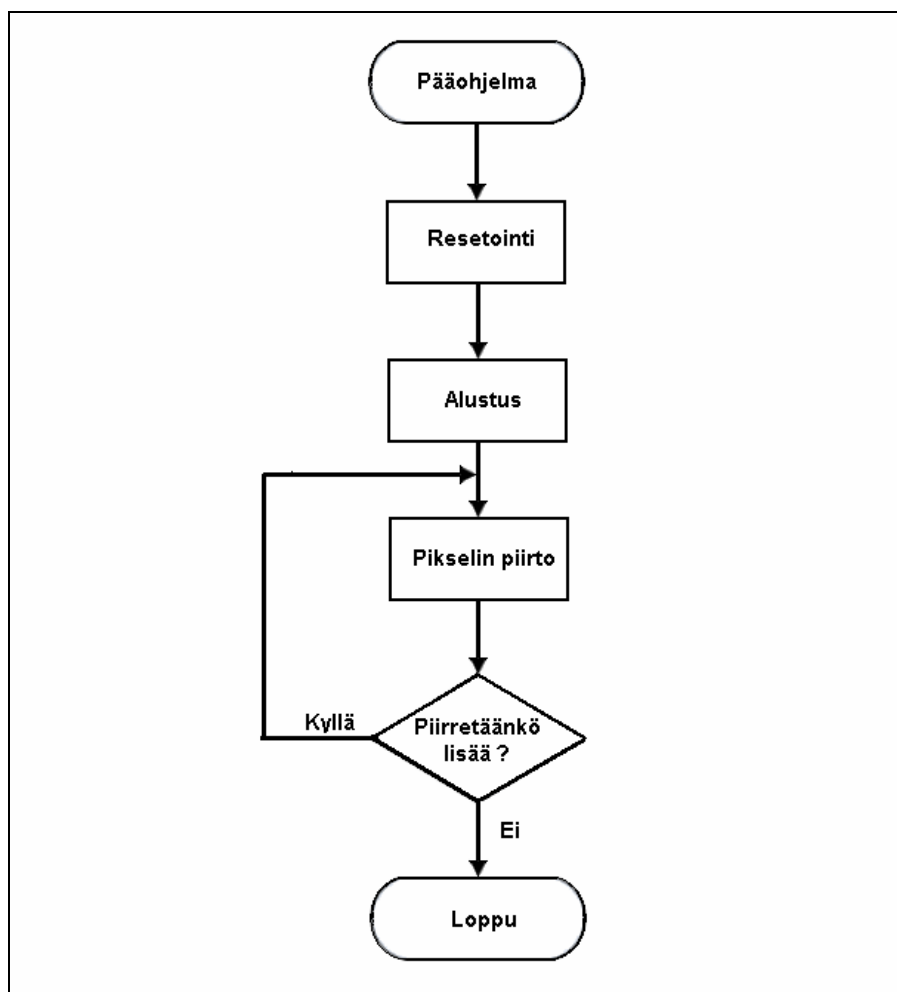
Näytönohjaimen toimintoja ohjataan hyvin pitkälti osoite- ja dataväylän kautta. Näytönohjainpiiri vaatii toimiakseen resetoinnin ja alustuksen. Näytönohjaimen resetointi tapahtuu käyttämällä RESET#-signaalia noltilassa. Tämän jälkeen näytönohjainpiiri voidaan alustaa. Alustuksessa määritetään, mitä toimintoja piiristä otetaan käyttöön. Tämä tapahtuu osoite- ja dataväylän kautta, missä osoiteväylällä kerrotaan, mihin rekisteriin osoitetaan ja dataväylällä annetaan rekisterille arvo. Rekistereihin kirjoittaessa täytyy M/R#-signaali asettaa 0-tilaan, jolloin näytönohjain tietää, että osoitetaan sen rekistereitä. Muistiin kirjoittaminen tapahtuu samalla tavalla kuin rekisteriin kirjoittaminen, mutta M/R#-signaali täytyy asettaa 1-tilaan. Kuvassa 20 on esitetty ohjaussignaalien tilat kirjoittaessa näytönohjainpiirille. Kuvassa WE1#-signaali asetetaan noltilaan, kun kirjoitetaan ylempi datatavu. Kaikki muut ohjaussignaalit, joita ei ole esitetty kuvassa, ovat ykköstilassa.



Kuva 20. Ohjaussignaalien tilat kirjoittaessa näytönohjainpiirille.

4.2 Testausympäristö

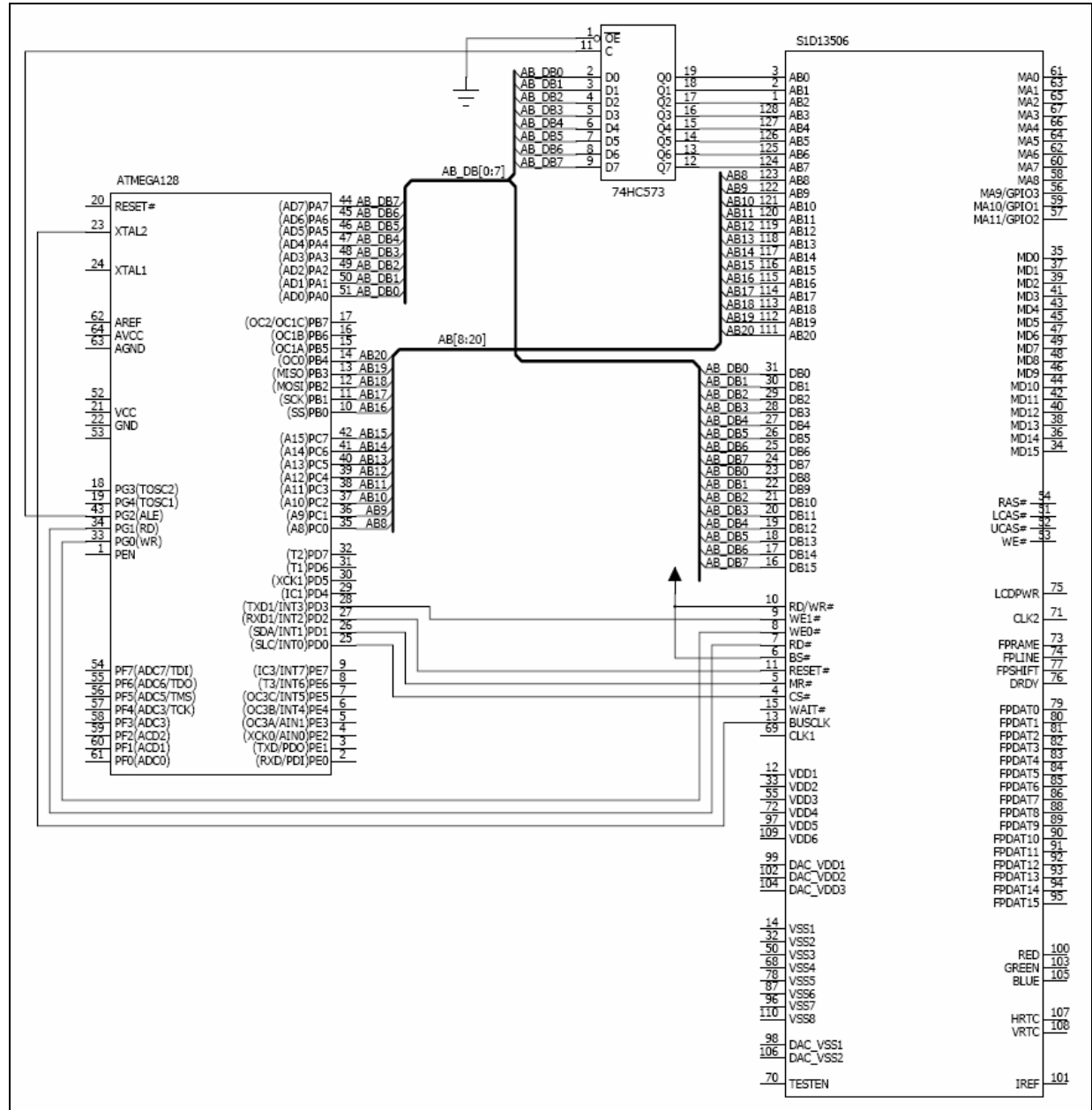
Testauksessa käytettiin Atmelin STK501-kehitysympäristöä, jossa oli ATmega128-kontrolleri. Testiohjelman suunnittelu aloitettiin tekemällä kuvan 21 mukainen vuokaavio, jonka pohjalta testiohjelma tehtiin.



Kuva 21. Pääohjelman vuokaavio.

Testiohjelmassa ATmega128 alustettiin käyttämään ulkoista muistia, missä muistin sijaan ATmega128:n liitettiin näyttöohjain. Ulkoisessa muistinohjauksessa ATmega128:n portti A on yhdistetty 8-bittinen osoite- ja dataväylä. Portti C toimii ylempään tavun (A8-A15) osoiteväylänä. Hyötynä tässä on näyttöohjaimen ohjauksen helppous, koska ATmega128 luulee näyttöohjainta muistiksi ja ohjaa osoite-/dataväylän lukko-piiriä automaattisesti ALE-signaalin avulla. Näyttöohjaimen osoiteväylä on kuitenkin 21-bittinen, joka ei suoraan vastaa liitettävyydeltään ATmega128:n 16-bittistä osoiteväylää. Tämän vuoksi näyttöohjaimen viisi

ylintä osoitebittii ohjataan ATmega128:n portti B:stä. Näytönohjaimen dataväylä on 16-bittinen, jossa ylin tavu luetaan WE1#-signaalin avulla. WE1#-signaali asetetaan nolllaksi, kun näytönohjaimen osoiteväylän alin bitti (AB0) on tilassa ”1”. Tällöin joka toinen osoite tulkitaan yleemmäksi tavuksi. Kuva 22 esittää testauksessa käytettyä kytkentää.



Kuva 22. Testauksessa käytetty kytkentä.

4.3 Lopputestaus

Näytönohjaimen lopputestauksessa näytönohjain alustettiin 640 x 480 resoluutiolle, 8 bittiä/pikseli-ohjauksella, jolloin värien määrä oli 256. Testauksessa käytettiin kuvaputkitekniikkaan perustuvaa monitoria, jossa liitännänä tapahtui VGA-liittimen kautta. Monitorin ruudulle piirrettiin 400 x 400 pikselin neliö (kuva 23).

Neliön piirto kuvaruudulle aloitettiin vaakariviltä 40 ja 120:n pikselin kohdalta. Tällöin näytönohjaimen muistipaikka oli heksadesimaaleina 6478h. Näytönohjaimen muistipaikka saadaan kaavalla $Mem = y \cdot \text{vaakaresoluutio} + x = 40 \cdot 640 + 120 = 6478h$. Osoittaessa näytönohjaimen muistiin tulee huomioida ohjaavan kontrollerin ulkoinen muistiavaruus. ATmega128:ssa ulkoinen muistiavaruus alkaa 1100h:n kohdalta, jolloin se pitää lisätä laskettuun näytönohjaimen muistipaikkaan, eli ATmega128:ssa osoitetaan muistipaikkaan $6478h + 1100h = 7578h$. ATmega128:n ulkoinen muistiavaruus on kaksi tavua, jolloin ylimmät osoitebitit ohjataan portti B:stä.

Kuvaruudun ulkoreunoille piirrettiin valkoinen kehys, jolloin nähtiin, oliko resoluutio oikea. Neliön joka kolmas vaakarivi piirrettiin eri värillä. Värillisten vaakarivien välissä olevat kaksi vaakariviä piirrettiin mustalla värillä. Tällöin neliö koostui 134 väristä. Mustien vaakarivien ideana oli saada värien välille kontrastia, jotta eriväriset viivat erottuisivat selkeämmin toisistaan.



Kuva 23. Näytönohjaimella piirretty testikuva.

5 SARJAMUOTOINEN LIIKENNE

5.1 Sarjaliikenne

Hyvin usein tiedonsiirto tapahtuu tavuina tai sanoina. Tavu tai sana on kooltaan vähintään neljä bittiä, jotka muodostavat binaariarvollaan informaation sisällön. Informaatiota siirretään rinnakkaismuodossa tai sarjamuodossa. Rinnakkaismuotoinen tiedonsiirto ei aina ole mahdollista, koska jokaista bittiä kohden pitäisi olla oma johdin tai johdinpari. Toinen vaikuttava tekijä on siirtolinjan ajoitus, sillä rinnakkaismuotoinen tiedonsiirto pitkillä matkoilla on ongelmallista. Tämän ongelman aiheuttaa siirtolinjan johtimien ideallisuus, sillä pitkillä matkoilla siirtolinjan johtimien tulee olla sähköisesti samanmittaisia. Jos siirrettävän informaation matka on pitkä ja siirtolinja ei ole ideallinen, niin siirrettävä informaatio saapuu vastaanottopään johtimiin eri aikaan, jolloin tahdistus ei onnistu. Tällöin vastaanottaja ei saa samaa informaatiota, joka sille on lähetetty. Tämän vuoksi pitkissä matkoissa käytetään hyvin usein sarjamuotoista tiedonsiirtoa. Sarjamuotoisessa tiedonsiirrossa lähetettävä tavu kaadetaan yhdeksi, jolloin tavun bittikombinaatio on peräkkäin. Näin tavu voidaan lähettää yhtä johdinta pitkin bitti kerrallaan. Vastaanottopäässä bitit nostetaan vuorostaan pystyyn, jolloin informaatiota voidaan taas käsitellä rinnakkaismuodossa. Sarjamuotoisessa tiedonsiirrossa bitit siirretään yleensä vähiten merkitsevä bitti (LSB) ensin. Lähetysjärjestyksen määrää vastaanottajan vaatimus. [10, s. 86–87.] Perinteisin sarjaliikenteen signalointitapa on RS-232, jota käytetään myös PC-tietokoneissa.

5.2 Synkronointi

Sarjamuotoisessa tiedonsiirrossa informaation muodostavat bitit lähetetään peräkkäin. Tällöin vastaanottajan täytyy tietää tarkalleen, mitkä bitit kuuluvat informaatioon. Jotta biteistä voidaan muodostaa oikea informaatio, täytyy tiedonsiirto synkronoida. Synkronointimenetelmät voidaan jakaa merkkipohjaisiin, bittipohjaisiin ja kehyspohjaisiin. Merkkipohjainen synkronointi jakautuu vielä synkroniseen ja asynkroniseen. Kehys- ja bittipohjaiset ovat aina synkronisia. [10, s. 110.]

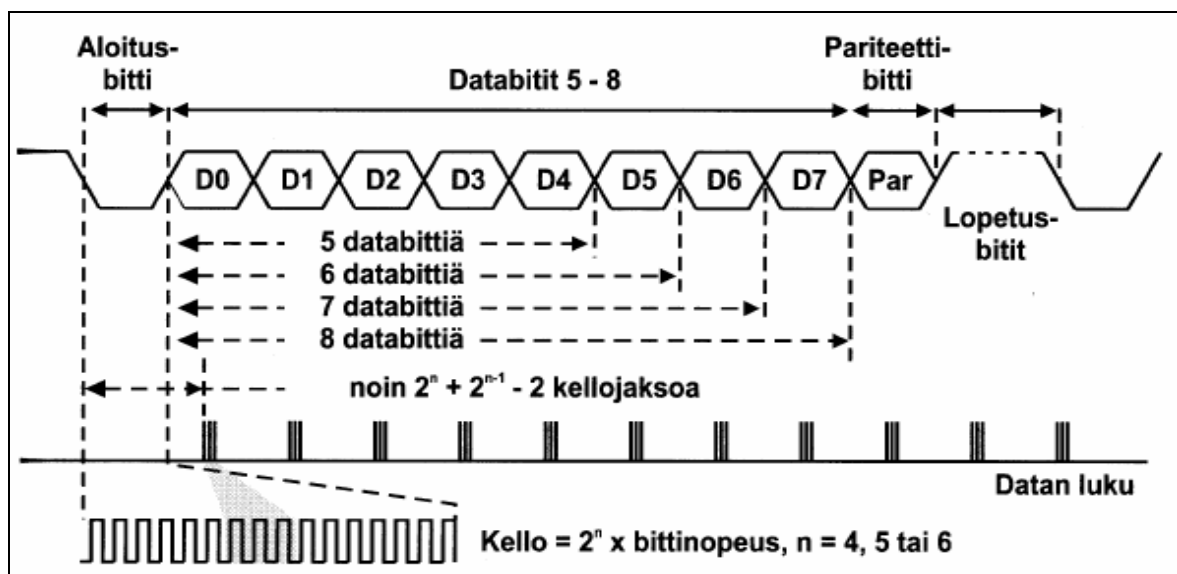
Asynkroninen tiedonsiirto

Asynkronisessa tiedonsiirrossa informaatio siirretään yleensä merkki kerrallaan. Merkki on sijoitettu kehykseen, jonka avulla se voidaan tunnistaa oikein. Kehys koostuu yksinkertaisimmillaan aloitusbitistä, databiteistä ja lopetusbitistä. Tällöin yhden merkin siirtämiseen tarvitaan merkin lisäksi kaksi ylimääräistä bittiä, aloitus- ja lopetusbitti. Aloitus- ja lopetusbitit eivät poikkea millään tavalla databiteistä. Synkronointisääntö on selkeä, aina ensimmäisenä aloitusbitti ja viimeisenä lopetusbitti. Aloitusbitti on aina 0-bitti (space), ja lopetusbitti on 1-bitti (mark). Yleensä merkki muodostuu 5, 6, 7 tai 8 databitistä. Kehys voi sisältää myös yhden tarkistusbitin (pariteettibitti). [10, s. 110–111.]

Pariteettibitti perustuu parilliseen (even) tai parittomaan (odd) 1-bittien määrään, missä vastaanottaja tietää, kumpaa käytetään. Käyttäessä parillista pariteettibittiä on 1-bittien määrä aina parillinen. Mikäli lähetettävä merkki sisältää 1-bittejä parittoman määrän, lisätään merkin viimeiseksi bitiksi yksi 1-bitti. Mikäli merkin 1-bittien määrä on parillinen, lisätään viimeiseksi 0-bitti. Vastaanottaja tarkistaa 1-bittien parillisuuden, jolloin se tietää, onko siirrossa tullut virheitä. Käytettäessä paritonta pariteettia toimintaperiaate on sama, mutta tällöin 0-bittien määrä on parillinen. Pariteettibitti-virheentarkistuksen heikkoutena on parillisten virheiden määrä, jolloin virhettä ei havaita. [11.]

Kuva 24 esittää asynkronisesti siirrettävän merkin rakennetta, josta käy hyvin ilmi, että aloitusbitti on mahdollinen vain, jos sitä edeltävä linjan tila on mark. Kuvasta näkyy myös, kun on luettu annettu määrä tietobittejä ja mahdollinen pariteettibitti, on merkin päätyttävä lopetusbittiin, jotta linjalla vastaa mark-tila.

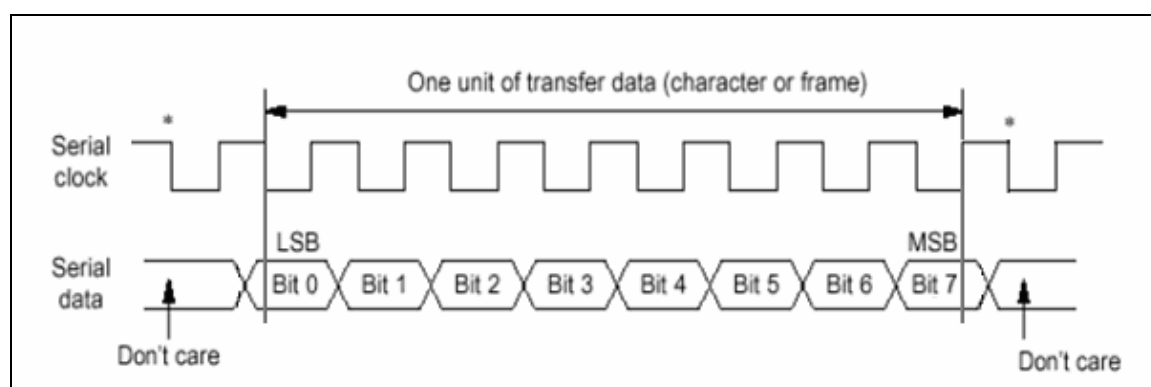
Lopetusbitin poisjättämisellä linja voi jäädä space-tilaan, jolloin uutta aloitusbittiä ei välttämättä havaita. [10, s. 111.]



Kuva 24. Asynkronisesti siirrettävän merkin rakenne. [10, s. 111.]

Synkroninen tiedonsiirto

Synkronisessa tiedonsiirrossa (kuva 25) ei tarvita aloitus- tai lopetusbittejä. Tämän vuoksi synkroninen tiedonsiirto on tehokkaampaa kuin asynkroninen. Synkronointi edellyttää kuitenkin lähettäjän ja vastaanottajan tarkan tahdistuksen. Tahdistus voidaan toteuttaa erillisellä kellosignaalinalla, tai tahdistus on sisälletty dataan. [12.] RS-232-signaalointia käyttäessä tahdistuskellosignaalinalla toimivat XTC-, TC- ja RC-signaalit. Tahdistus tapahtuu lähettäjän osalta nousevalla reunalla ja vastaanottopäässä laskevalla reunalla. [13.]



Kuva 25. Synkronisesti siirrettävän merkin rakenne. [12.]

UART

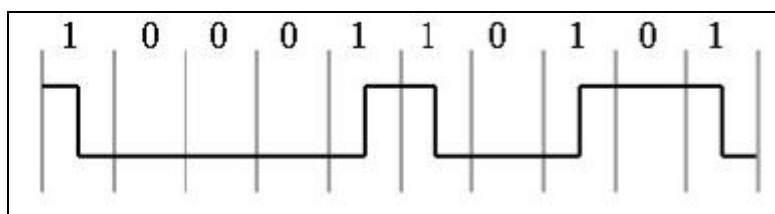
UART on hyvin suosittu sarjaliikenne tiedonsiirrossa, ja se onkin usein yhdistetty mikrokontrollerin yhdeksi osaksi. UART:n avulla rinnakkaismuotoinen data voidaan muuttaa sarjamuotoiseksi ja sarjamuotoinen rinnakkaismuotoiseksi. UART sisältää usein oman kello-
taajuuden, jolla sarjaliikenteen tahdistus hoidetaan. UART:ssa on lähtevälle (TXD) ja vastaanotettavalle (RXD) datalle omat rekisterinsä, joista data lähetetään tai vastaanotetaan bitti kerrallaan. UART:ssa on lisäksi paljon muitakin rekistereitä, joilla ohjataan UART:n toimintaa, kuten tiedonsiirtonopeutta. RS-232-liitäntää käyttäessä joudutaan UART:n lisäksi käyttämään erillistä jännitteen ohjauspiiriä, sillä RS-232-liitännässä jännitetasot ovat luokkaa ± 15 V. [14.]

6 PS/2-NÄPPÄIMISTÖ

Näppäimistorakenne perustuu matriisiin, jossa kahden johtimen risteämästä muodostuu piste eli näppäimistön paikka. Näppäimistön sisällä oleva elektroniikka tunnistaa näppäimen painamisen ja lähettää sen sarjamuotoisena datana. [4, s. 269–270.] Data on 11-bittinen, ja se perustuu näppäinkoodi -menetelmään, joka tunnetaan paremmin nimellä scan code. Siinä jokaiselle näppäimeen paikalle on oma heksadesimaaliluku. Tämä on sen takia, koska monessa maassa on oma näppäinmerkistö, jolloin suoraan ASCII-merkistön mukainen datan lähetys ei ole järkevää. Näppäimistön datan lähetyksessä kehys on seuraava: aloitusbitti, 8 databittiä, pariteettibitti ja lopetusbitti. Näppäimistön lähettämässä datassa ensimmäisenä tulee LSB-bitti.

Käynnistyksen yhteydessä suorittaa näppäimistö alustuksen. Alustuksessa näppäimistö suorittaa itsensä testauksen ja lähettää AAh-luvun, jos testaus onnistuu. Painaessa jotakin näppäintä lähettää näppäimistö kyseisen näppäimen scan code -luvun. Kun painettu näppäin vapautetaan, lähettää näppäimistö F0h ja uudestaan painetun scan code -luvun. Tällä lähetyksen menetelmällä voidaan selvittää, onko samanaikaisesti painettu toista näppäintä, joilla saadaan aikaiseksi jokin erikoismerkki. Näppäimistölle voidaan myös lähettää dataa, esim. ohjata näppäimistön LED:itä päälle.

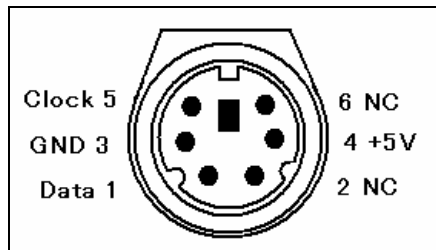
Yleisimmät näppäimistöjen liitännät ovat USB- ja PS/2-liitäntä. USB-liitännässä näppäimistölle välitetään neljä johdinta, joista kahdella johtimella välitetään näppäimistön käyttöjännite 5 V ja kahdella johtimella data. Datasignaalit ovat differentiaalisignaaleja, missä data ja kello on NRZI-koodattu yhdeksi signaaliksi. [4, s. 285–286.] NRZI-koodauksessa tilan ollessa ”1” dataväylän tila vaihtuu ja tilan ollessa ”0” dataväylän tila pysyy samana (kuva 26). [7.]



Kuva 26. NRZI-koodauksen toimintaperiaate. [7.]

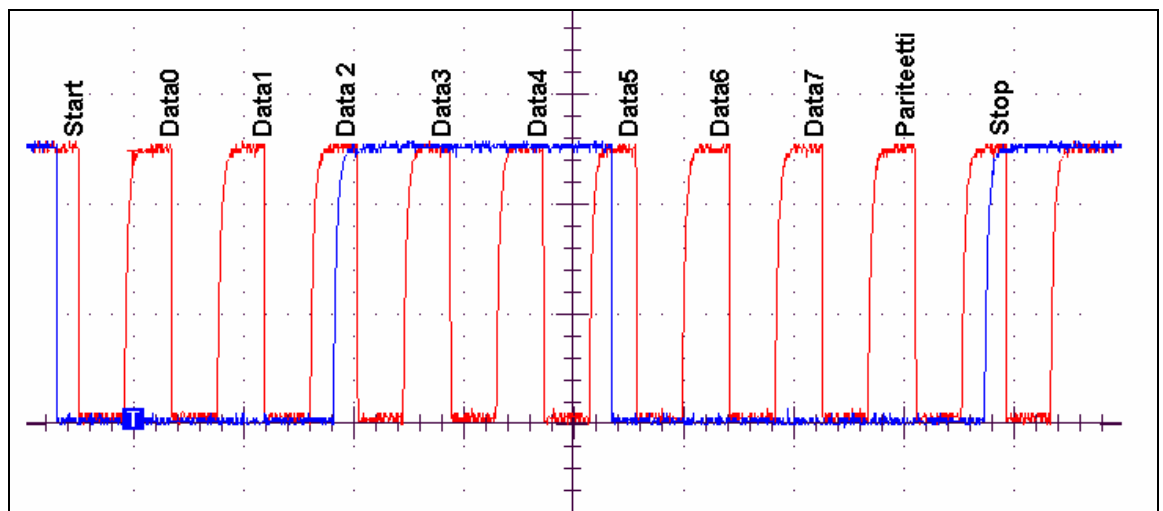
PS/2-liitäntä

PS/2-näppäimistöissä on kuvan 27 mukainen liitin. Näppäimistön saadessa käyttöjännitteen nousevat kello- ja datasiignaali ylös (5 V) ja ovat aina ylhäällä, jos dataa ei lähetetä. Näppäintä painaessa kello-signaali käy alhaalla ja rupeaa kellottamaan datasiignaalia.



Kuva 27. PS/2-näppäimistön liitin. [15.]

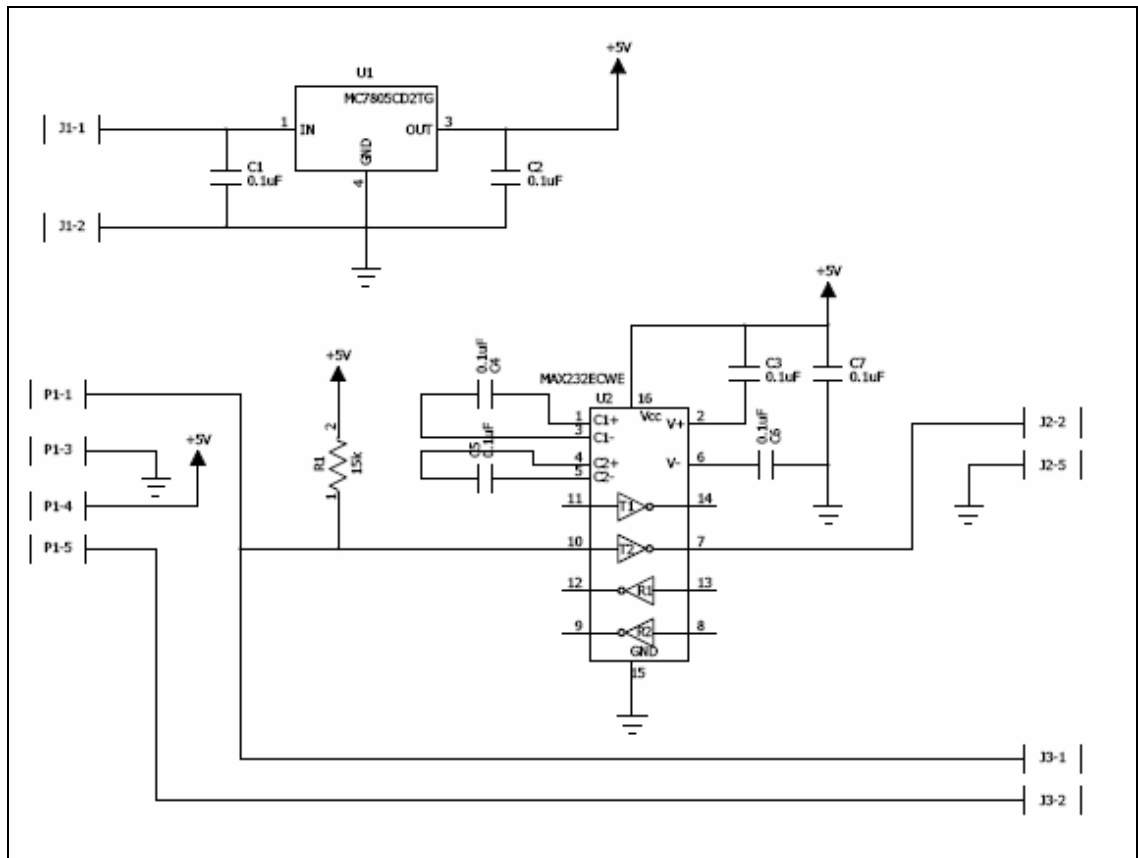
Kuva 28 esittää oskilloskooppimittausta näppäimistön data- (sininen) ja kello-signaalista (punainen), kun painetaan a-näppäintä. Oskilloskooppi on säädetty siten, että yksi ruutu on 2 V ja 100 μ s. Kuvasta 28 saadaan bittikombinaatio 0 0011 1000 01, kun dataa luetaan kellon laskevalla reunalla. Bittikombinaatiossa ensimmäinen on aloitusbitti ja kaksi viimeistä on pariteetti- ja lopetusbitti. Kun bittikombinaatiosta otetaan edellä mainitut bitit pois, saadaan seuraava bittikombinaatio 0011 1000. Tämän jälkeen huomioidaan, että näppäimistö lähettää LSB-bitin ensimmäisenä, jolloin näppäimistö dataksi saadaan 0001 1100, joka vastaa scan code -taulukossa a-näppäintä. Mikäli dataa lähetetään näppäimistölle, tapahtuu kirjoitus kellon nousevalla reunalla.



Kuva 28. PS/2-näppäimistön data- ja kello-signaali.

6.1 Elektroniikan suunnittelu

Liitettäväksi näppäimistöksi valittiin PS/2-näppäimistö, koska niitä on helposti saatavilla ja se on edullinen. Näppäimistön liittämiseksi suunniteltiin kaksi menetelmää, sarjaportin kautta ja kahden I/O-linjan kautta. Näppäimistön liittämiseksi suunniteltiin kuvan 29 mukainen kytkentä.



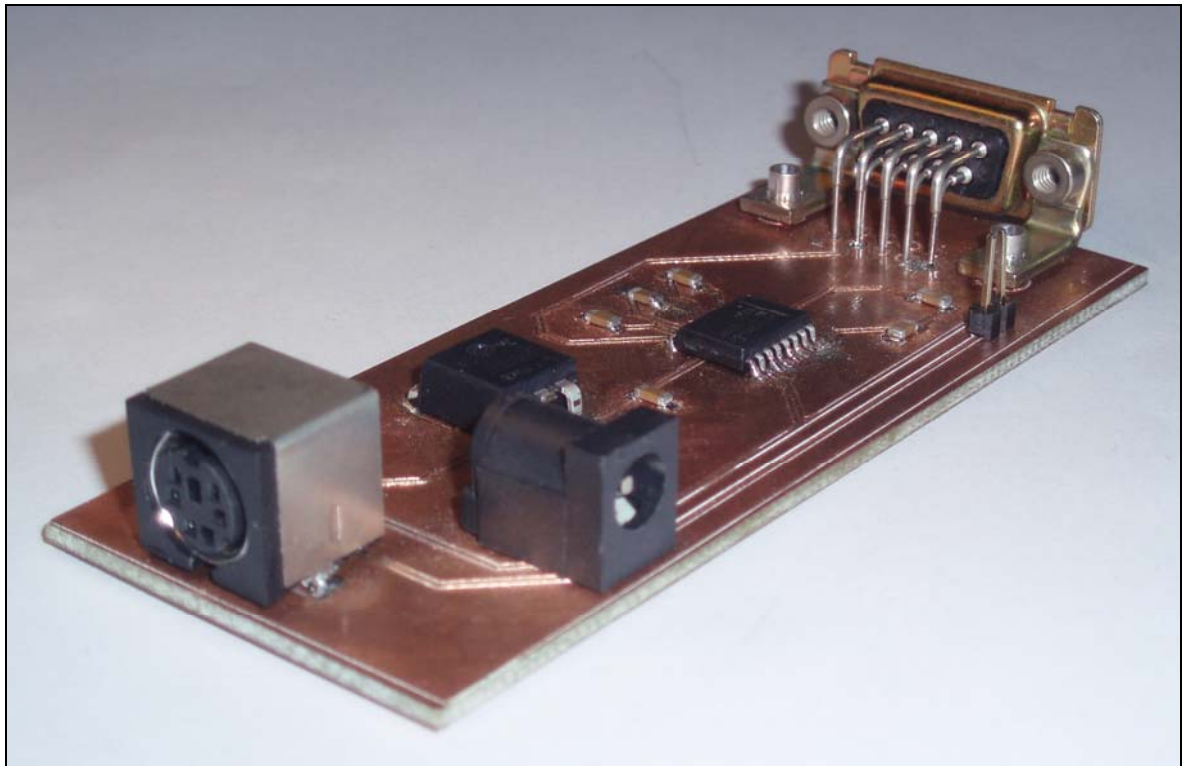
Kuva 29. Näppäimistön liittämiseen suunniteltu kytkentä.

Laitteistoon kytkettävä käyttöjännite on 8–18 V, joka tuodaan 2,1 mm:n pyöreän DC-liittimen kautta. Varsinainen piirien toimintajännite on 5 V, joka toteutetaan jännite-regulaattorin avulla (U1). Kytkenässä P1 liittimiin kytketään PS/2-näppäimistö. Liitin on 6-napainen mini-din-liitin, johon näppäimistö käy suoraan. Kytkenässä oleva vastus R1 toimii ylösvetovastuksena, jolla varmistetaan 1-tilan säilyminen käyttäessä pitkää liitännäjohtoa. Liitin J3 on ns. piikkirima, josta saadaan näppäimistön data- ja kello-signaali. Tätä liittintä voi käyttää I/O-linjoilla tapahtuvaan näppäimistön lukemiseen. Liitin J2 on sarjaporttiliitäntä, joka toimii yksisuuntaisesti. Sarjaportin kautta tapahtuva liitäntä perustuu asynkroniseen tiedonsiirtoon, missä tiedonsiirtoa ei tahdisteta erillisellä kello-signaalilla. Sarjaportin

signalointiin käytettiin MAX232-sarjaliikennepiiriä (U2). I/O-linjojen kautta tapahtuvassa liitännässä näppäimistö antaa kellon laskevalla reunalla keskeytyksen kehitysympäristössä olevalle ATmega:lle. Saadessaan keskeytyksen ATmega lukee toisen I/O-linjan tilan, jolloin ATmega saa näppäimistön datan bitti kerrallaan.

6.2 Piirilevyn suunnittelu

Piirilevyn suunnittelu toteutettiin PADS-ohjelmalla. Piirilevystä (kuva 30) suunniteltiin yksi-puoleinen, mikä toteutettiin pääosin pintaliitoskomponenteilla. Piirikortin toiminta perustuu signaalin sovittamiseen, jonka vuoksi piirikortti suunniteltiin siten, että toisesta päästä tulee näppäimistön signaali sisään ja toisesta päästä muutettu signaali lähtee kehitysympäristöön. Maatasot piirikortilla toteutettiin kuparointimenetelmällä.

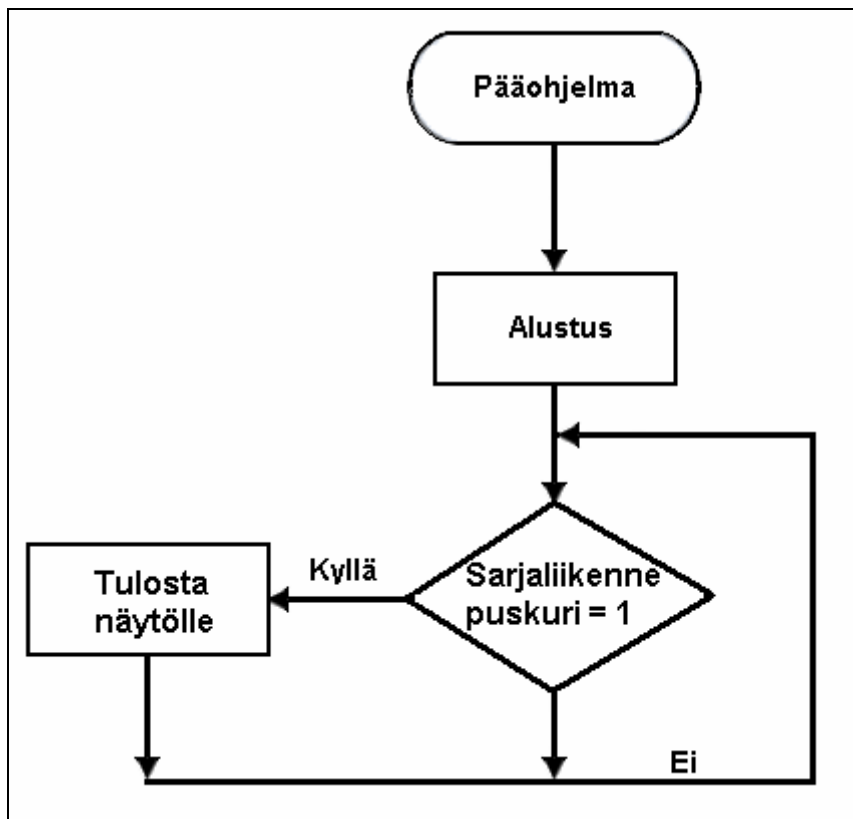


Kuva 30. Suunniteltu signaalin sovituskortti.

7 NÄPPÄIMISTÖN TESTAUS

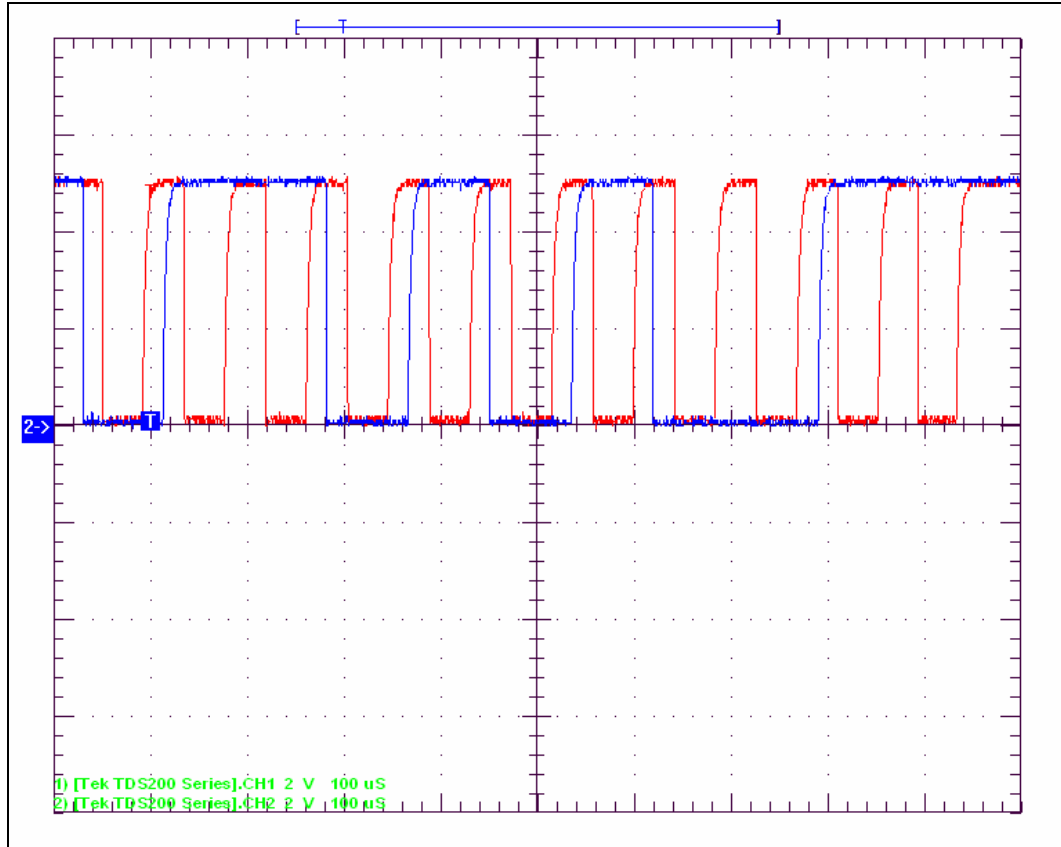
Näppäimistön sovituskortin suunnittelun alkuvaiheessa käytettiin koekytkentälevyjä, missä suunniteltu kytkentä rakennettiin siihen. Tällä menettelyllä saatiin kytkentöjen toimivuus selville jo suunnitteluvaiheessa.

Testausympäristö perustui Microsalon PK-M128-kehitysympäristöön, jossa oli ATmega128-mikrokontrolleri. Testiohjelman suunnittelu aloitettiin tekemällä kuvan 31 mukainen vuokaavio, jonka pohjalta testiohjelma toteutettiin.



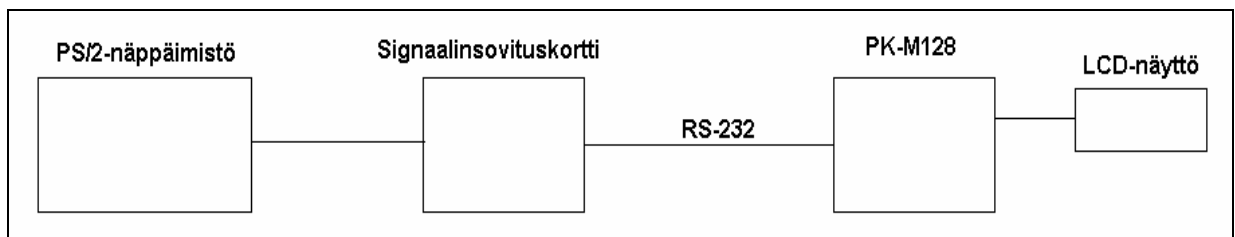
Kuva 31. Testiohjelman vuokaavio.

Testiohjelma toteutettiin C-kielellä, jossa ATmega128 alustettiin sarjaporttikäyttöön ja asynkroniseksi. Ohjelma luki sarjaportin sisääntuloa ja siirsi siitä datan suoraan LCD-näytölle. Sarjaliikennenopeus määritettiin oskilloskoopin avulla (kuva 32), joka oli 11850 bit/s. Oskilloskooppimittauksessa painettiin näppäimistön f-näppäintä ja mitattiin data (sininen)- ja kellosignaalit (punainen) piirilevyn J3-liittimestä. Mittauksessa oskilloskoopin asetukset olivat 2 V ja 100 μ s / ruutu.



Kuva 32. F-näppäimen mittaus.

Testauksessa kehitysympäristöön liitettiin LCD-näyttö ja suunniteltu piirikortti, johon kytkettiin näppäimistö kuvan 33 mukaisesti. Kehitysympäristön ja signaalisovituskortin liitännänä käytettiin sarjaporttia.



Kuva 33. Testauksessa käytetyn kytkennän rakenne.

Painaessa näppäimistöltä f-näppäintä tulostui LCD-näytölle luvut 2Bh, F0h ja 2Bh, joita verrattiin scan code -taulukkoon. Saatu heksaluku vastasi scan code -taulukkoa, jolloin saatu tulos oli oikein. Tuloksessa ensimmäinen 2Bh on näppäimistön painalluksen aiheuttama paikkakoodi. Luvut F0h ja 2Bh tulevat, kun näppäimen painallus lopetetaan.

Seuraavaksi painettiin shift- ja f-näppäintä yhtä aikaan samalla tavalla kuin kirjoittaisi ison kirjaimen. Tulokseksi saatiin 59h, 2Bh, F0h ja 59h. Tuloksesta saadaan hyvin selville, että ensimmäisenä on painettu shiftiä ja f-näppäintä, joka on vapautettu ennen shiftiä.

Testauksessa myös kokeiltiin painaa f-näppäintä pitkään pohjassa, jolloin tulokseksi saatiin 2Bh ja ei ollenkaan F0h-lukua.

Testauksia kokeiltiin muillakin näppäimillä, ja tuloksien logiikka piti paikkansa. Edellä saatu- jen tuloksien pohjalta voidaan todeta, että näppäimistöltä saaduista datoista voidaan ohjel- mallisesti tunnistaa jokainen eri näppäimen painallus, josta voidaan erotella, onko näppäimiä painettu yhtä aikaa tai onko näppäin painettupohjaan.

8 TYÖN ANALYSOINTI

Työn tavoitteena oli suunnitella ja rakentaa näyttölaitteen ja PS/2-näppäimistön liittäminen sulautettuun laitteeseen. Laitteistoista tehtiin mahdollisimman monipuolinen liitettävyyden suhteen, jolloin sitä voi käyttää monenlaisissa sovelluksissa.

Näyttölaitteen liittämässä suunniteltu näytönohjain oli työn haastavin osa. Näytönohjaimen vaatimuksena oli liitettävyyden monitoriin tai televisioon. Vaatimuksissa onnistuttiin hyvin, sillä näytönohjaimen voi liittää television, monitorin ja LCD-näytön. Työn ongelmakohtina olivat piirilevyn valmistus ja EDO-DRAM-muistien saatavuus.

Piirilevyn valmistuksessa hankaluutta aiheuttivat kapeat johdinvedot, jonka vuoksi piirilevy jouduttiin valmistuttamaan ulkopuolisella. Tästä seurasi se, että kytkentöjen toimivuutta ei voitu testata koekytkentälevyjen avulla. Tämän edellytti suurta huolellisuutta suunnittelulta, jotta piirilevyjä ei tarvitsisi tilata useampaan kertaan, sillä se toisi turhia kustannuksia ja toimitusaikoja.

Muistin saatavuudessa huomattiin, kuinka tietotekniikka kehittyi nopeasti, sillä EDO-DRAM-muistien saatavuus oli huono. Muistin EDO-tyypin lisäksi toisena vaikeuttajana oli muistin käyttöjännite. Piirilevyn käyttöjännitteenä oli 5 V, ja lähes kaikki DRAM-muistit ovat nykypäivänä alle tuon jännitteen, jolloin tehonkulutus on pienempi. 5 V:n käyttöjännitteellä oleva muisti kuitenkin löytyi pitkän etsinnän jälkeen OKI SEMICONDUCTOR -valikoimasta.

Näppäimistön liittämässä suunniteltiin PS/2-näppäimistön ja ATmega128:n väliin sarjaliikennesovitus RS-232-signaaloinnille. Tämän tekeminen ei tuottanut ongelmia, vaan työn pystyi suorittamaan perustietämyksellä. Työvaiheessa eniten aikaa meni PS/2-näppäimistön toimintaperiaatteen tutkimisessa, missä selvitettiin näppäimistön datan lähetysperiaate ja kuinka data oli koodattu.

Kokonaisuudessaan työ oli melko laaja ja siinä opittiin uusia asioita. Epsonin näytönohjainpiiri oli toiminnoltaan niin monipuolinen, että sen kaikkia toimintoja ei selvitetty työn aikana. Piirin toiminnot ovat ohjelmistopohjaisia, joista voisi tehdä oman insinööriyön.

9 YHTEENVETO

Työssä tutustuttiin näyttölaitteiden ja PS/2-näppäimistön toimintaan sekä suunniteltiin kummallekin liitännämenetelmä sulautettuun laitteeseen. Sulautettu laite on varsin laaja käsite, ja sen roolia työssä hoiti ATmega128-mikrokontrolleriin perustuva kehitysympäristö. Työssä suunniteltuja piirikortteja voidaan ajatella kehitysympäristön lisäosina, jotka laajentavat kehitysympäristön monipuolisuutta.

Näyttölaitteen liitännämenetelmässä suunniteltiin näytönohjain. Näytönohjain sai kuvainformaation kehitysympäristöltä. Kuvainformaatio sisältää pikseleiden paikat ja niiden värit. Näytönohjain muutti kehitysympäristöltä saadun kuvainformaation videosaaliksi, jonka se lähetti näyttölaitteelle. Näytönohjaimen vaatimuksena oli, että se pystyy tuottamaan värillistä grafiikkaa ja näyttölaitteena pystyy toimimaan suuren näytön omaava laite. Vaatimuksissa onnistuttiin hyvin, sillä näytönohjain kykeni tuottamaan VGA-tasoista grafiikkaa ja näyttölaitteena pystyi toimimaan monitori, televisio ja LCD-näyttö.

Näppäimistön liitännämenetelmässä suunniteltiin signaalin sovitin PS/2-näppäimistön ja sulautetun laitteen välille. Näppäimistön sovituksessa käytettiin RS-232-signaalointimenetelmää, joka on monipuolinen ja yksinkertainen menetelmä. Suunnitellussa piirikortissa oli liitäntä PS/2-näppäimistölle, mitä kautta piirikortti otti vastaan näppäimistön datan. Saatu näppäimistön data muutettiin RS-232-signaalin mukaiseksi ja lähetettiin sarjaportin kautta kehitysympäristön sarjaportille. Kehitysympäristössä saatu data voidaan muuttaa ohjelmallisesti scan code -taulukon avulla ASCII merkiksi, jolloin näppäininformaation käsittely on paljon luonnollisempaa.

LÄHTEET

1. Wikipedia, Bittikarttagrafiikka, Luettu 4.3.2008, [WWW-dokumentti]. <http://fi.wikipedia.org/wiki/Bittikarttagrafiikka>
2. Wikimedia commos, raster graphic fish 20x23 squares sdtv-example.jpg, Luettu 4.3.2008, http://commons.wikimedia.org/wiki/Image:Raster_graphic_fish_20x23squares_sdtv-example.jpg
3. Oulun yliopisto, Digitaalisen median perusteet, Luettu 4.3.2008, [WWW-dokumentti]. http://www.tol.oulu.fi/kurssit/dmp/jonne_animaatio/vektorit_teoria.html
4. Olavi Lähteine, Ville Pietikäinen, Harri Kosonen. Uusi PC-tekniikan käsikirja. viides painos. Helsinki Media Erikoislehdet, 1997 – 2000. ISBN 951-832-015-9
5. Lappeenrannan teknillinen yliopisto, Näyttölaitteet, Luettu 4.3.2008, [WWW-dokumentti]. <http://www.it.lut.fi/kurssit/98-99/1765/lectures/02/node2.html>
6. Matti Eteläperä, Risto Hyypiö, Näyttölaitetekniikat, Luettu 4.3.2008, [WWW-dokumentti]. <http://www.electronics.oulu.fi/Opetus/OE/16.pdf>
7. Harri Honkanen, Paneelinäyttötekniikat, Luettu 5.4.2008, [WWW-dokumentti]. http://gallia.kajak.fi/opmateriaalit/yleinen/HonHar/ma/TV-Monitor_PANEELINÄYTTÖTEKNIIKAT.pdf
8. Epson Semiconductor, S1D13506 datasheet,
9. Reima Flyktman, Näyttöstandardit, Luettu 14.3.2008, [WWW-dokumentti]. <http://www.laitetekniikka.com/laitteet/naytto/nayttostandardit.htm>
10. Kaj Granlund. Tietoliikenne. Ensimmäinen painos. Docendo Finland Oy, 2003. ISBN 951-846-133-3
11. Helsingin teknillinen korkeakoulu, Digitaalitekniikan perusteet, Luettu 27.3.2008, [WWW-dokumentti]. <http://wooster.hut.fi/digis/luento3/pariteetti.html>
12. Jyväskylän yliopisto, Mikrokontrollerin sisäinen arkkitehtuuri, Luettu 27.3.2008, [WWW-dokumentti]. <http://users.jyu.fi/~jupeihal/Kontrollerit2.pdf>
13. Tomi Engdahl, RS-232C, Luettu 27.3.2008, [WWW-dokumentti]. <http://www.tkk.fi/Misc/Electronics/then/mytexts/rs-232c.html>
14. Julius Luukko, Sulautetut prosessorijärjestelmät, Luettu 27.3.2008, [WWW-dokumentti]. <http://www.ee.lut.fi/courses/080560000/pruju/sulautetutruutuversio.pdf>
15. John Szybowski, Making an IBM PS/2 keyboard cable, Luettu 17.3.2008, [WWW-dokumentti]. <http://www.geocities.com/jszybowski/keyboard/MakeCable.htm>

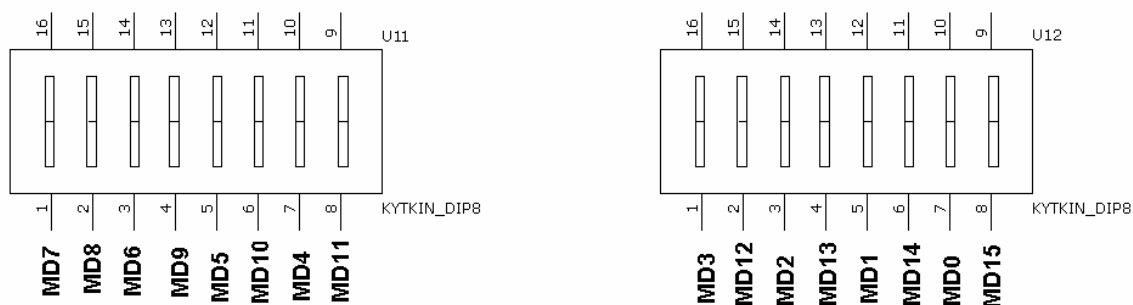
LIITTEIDEN LUETTELO

Liite 1: DIP-KYTKIMIEN SIGNALOINTI KUVA JA VALINTA TAULUKKO

Liite 2: NÄYTÖNOHJAIMEN LAYOUT-KUVA

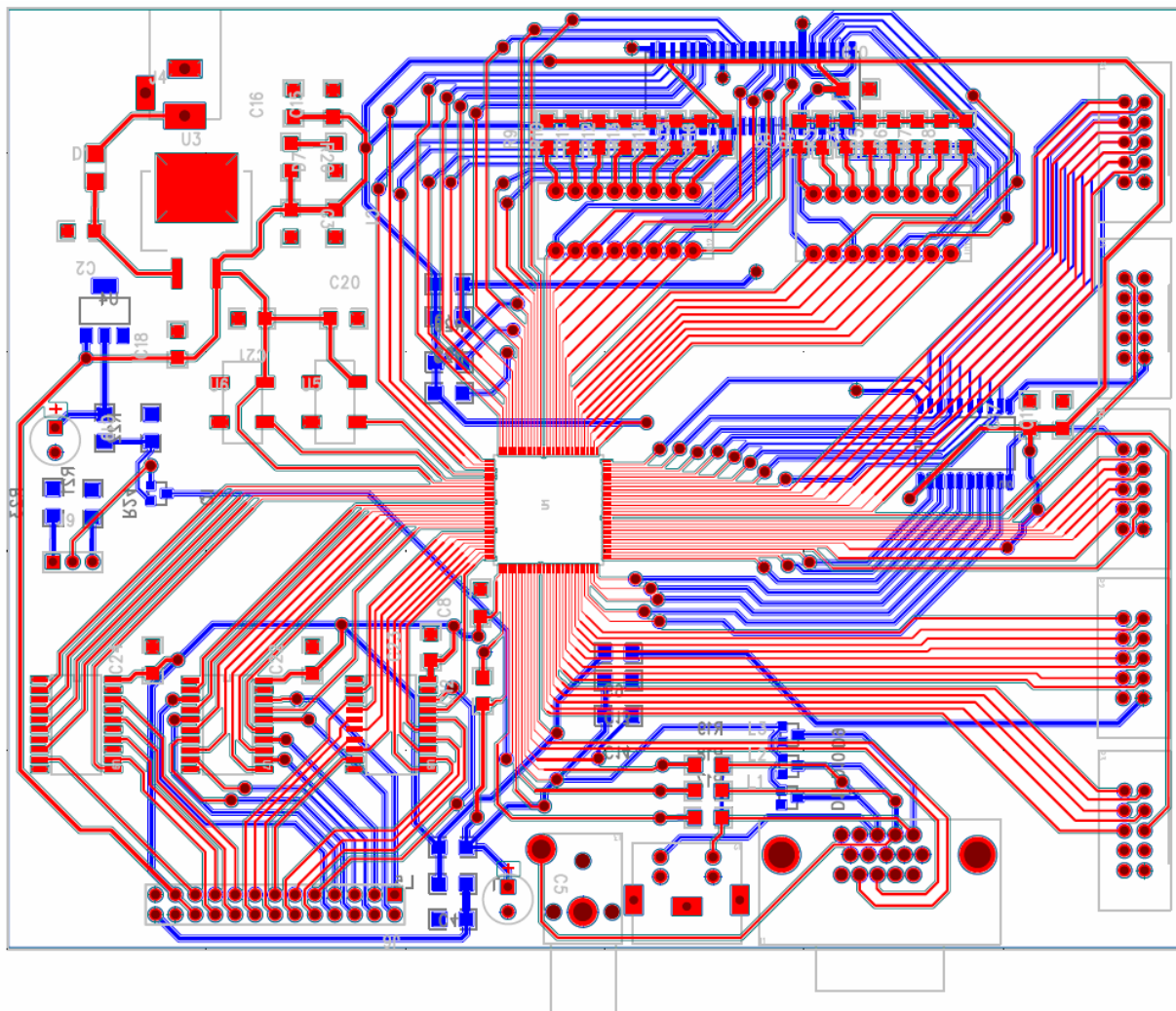
Liite 3: SCAN CODE -TAULUKKO

DIP-KYTKIMIEN SIGNALOINTI KUVA JA VALINTA TAULUKKO



Pin Name	value of this pin at rising edge of RESET# is used to configure:(1/0)																																																			
	1	0																																																		
MD0	Not used, value of this pin at rising edge of RESET# can be read at REG[00Ch] bit 0																																																			
MD11,MD[3:1]	Select Host Bus Interface as follows: <table border="1"> <thead> <tr> <th>MD11</th> <th>MD3</th> <th>MD2</th> <th>MD1</th> <th>Host Bus</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>SH-4/SH-3 Bus interface</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>MC68K Bus 1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>MC68K Bus 2</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>Generic</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>MIPS/ISA</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>PowerPC</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>PC Card (PCMCIA)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>Philips PR31500/PR31700 / Toshiba TX3912</td> </tr> </tbody> </table>		MD11	MD3	MD2	MD1	Host Bus	0	0	0	0	SH-4/SH-3 Bus interface	0	0	0	1	MC68K Bus 1	0	0	1	0	MC68K Bus 2	0	0	1	1	Generic	0	1	0	0	Reserved	0	1	0	1	MIPS/ISA	0	1	1	0	PowerPC	0	1	1	1	PC Card (PCMCIA)	1	1	1	1	Philips PR31500/PR31700 / Toshiba TX3912
MD11	MD3	MD2	MD1	Host Bus																																																
0	0	0	0	SH-4/SH-3 Bus interface																																																
0	0	0	1	MC68K Bus 1																																																
0	0	1	0	MC68K Bus 2																																																
0	0	1	1	Generic																																																
0	1	0	0	Reserved																																																
0	1	0	1	MIPS/ISA																																																
0	1	1	0	PowerPC																																																
0	1	1	1	PC Card (PCMCIA)																																																
1	1	1	1	Philips PR31500/PR31700 / Toshiba TX3912																																																
MD4	Little Endian	Big Endian																																																		
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)																																																		
MD[7:6]	Memory Address/GPIO configuration: (See Table 5-10: "MA11, MA10, MA9, and DRDY Pin Mapping," on page 35) 00 = symmetrical 256K×16 DRAM. MA[8:0] = DRAM address. MA[11:9] can be used as GPIO2,1,3 pins. 01 = symmetrical 1M×16 DRAM. MA[9:0] = DRAM address. MA[11:10] can be used as GPIO2,1 pins. 10 = asymmetrical 256K×16 DRAM. MA[9:0] = DRAM address. MA[11:10] can be used as GPIO2,1 pins. 11 = asymmetrical 1M×16 DRAM. MA[11:0] = DRAM address.																																																			
MD8	Not used, value of this pin at rising edge of RESET# can be read at REG[00Dh] bit 0																																																			
MD9	Not used, value of this pin at rising edge of RESET# can be read at REG[00Dh] bit 1																																																			
MD10	Not Used, value of this pin at rising edge of RESET# can be read at REG[00Dh] bit 2																																																			
MD12	BUSCLK input divided by 2	BUSCLK input not divided																																																		
MD13	Configure FPDAT[15:8] for MediaPlug I/F. External latches required to support 16-bit passive panels.	Support 16-bit passive panels directly																																																		
MD14	DRDY or MA11 is configured as MediaPlug power down pin (VMPEPWR). (See Table 5-10: "MA11, MA10, MA9, and DRDY Pin Mapping," on page 35)	DRDY is configured as a normal LCD I/F output pin. MA11 is configured as either a memory address or GPIO2. (See Table 5-10: "MA11, MA10, MA9, and DRDY Pin Mapping," on page 35)																																																		
MD15	WAIT# is always driven	WAIT# is tristated when the chip is not accessed by the host																																																		

NÄYTÖNOHJAIMEN LAYOUT-KUVA



SCAN CODE -TAULUKKO

KEY	MAKE	BREAK	-----	KEY	MAKE	BREAK	-----	KEY	MAKE	BREAK
A	1C	F0,1C		9	46	F0,46		[54	F0,54
B	32	F0,32		`	0E	F0,0E		INSERT	E0,70	E0,F0,70
C	21	F0,21		-	4E	F0,4E		HOME	E0,6C	E0,F0,6C
D	23	F0,23		=	55	F0,55		PG UP	E0,7D	E0,F0,7D
E	24	F0,24		\	5D	F0,5D		DELETE	E0,71	E0,F0,71
F	2B	F0,2B		BKSP	66	F0,66		END	E0,69	E0,F0,69
G	34	F0,34		SPACE	29	F0,29		PG DN	E0,7A	E0,F0,7A
H	33	F0,33		TAB	0D	F0,0D		U ARROW	E0,75	E0,F0,75
I	43	F0,43		CAPS	58	F0,58		L ARROW	E0,6B	E0,F0,6B
J	3B	F0,3B		L SHFT	12	F0,12		D ARROW	E0,72	E0,F0,72
K	42	F0,42		L CTRL	14	F0,14		R ARROW	E0,74	E0,F0,74
L	4B	F0,4B		L GUI	E0,1F	E0,F0,1F		NUM	77	F0,77
M	3A	F0,3A		L ALT	11	F0,11		KP /	E0,4A	E0,F0,4A
N	31	F0,31		R SHFT	59	F0,59		KP *	7C	F0,7C
O	44	F0,44		R CTRL	E0,14	E0,F0,14		KP -	7B	F0,7B
P	4D	F0,4D		R GUI	E0,27	E0,F0,27		KP +	79	F0,79
Q	15	F0,15		R ALT	E0,11	E0,F0,11		KP EN	E0,5A	E0,F0,5A
R	2D	F0,2D		APPS	E0,2F	E0,F0,2F		KP .	71	F0,71
S	1B	F0,1B		ENTER	5A	F0,5A		KP 0	70	F0,70
T	2C	F0,2C		ESC	76	F0,76		KP 1	69	F0,69
U	3C	F0,3C		F1	05	F0,05		KP 2	72	F0,72
V	2A	F0,2A		F2	06	F0,06		KP 3	7A	F0,7A
W	1D	F0,1D		F3	04	F0,04		KP 4	6B	F0,6B
X	22	F0,22		F4	0C	F0,0C		KP 5	73	F0,73
Y	35	F0,35		F5	03	F0,03		KP 6	74	F0,74
Z	1A	F0,1A		F6	0B	F0,0B		KP 7	6C	F0,6C
0	45	F0,45		F7	83	F0,83		KP 8	75	F0,75
1	16	F0,16		F8	0A	F0,0A		KP 9	7D	F0,7D
2	1E	F0,1E		F9	01	F0,01]	5B	F0,5B
3	26	F0,26		F10	09	F0,09		;	4C	F0,4C
4	25	F0,25		F11	78	F0,78		'	52	F0,52
5	2E	F0,2E		F12	07	F0,07		,	41	F0,41
6	36	F0,36		PRNT SCRN	E0,12, E0,7C	E0,F0, 7C,E0, F0,12		.	49	F0,49
7	3D	F0,3D		SCROLL	7E	F0,7E		/	4A	F0,4A
8	3E	F0,3E		PAUSE	E1,14,77, E1,F0,14, F0,77	-NONE-				