

Asko Olli

SULAUTETTUJEN JÄRJESTELMIEN VÄYLÄTEKNIIKAT

Opinnäytetyö

Kevät 2010

Tekniikan yksikkö

Tietotekniikka

Sulautetut järjestelmät



SEINÄJOEN AMMATTIKORKEAKOULU

OPINNÄYTETYÖN TIIVISTELMÄ

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Tietotekniikan koulutus ohjelma

Suuntautumisvaihtoehto: Sulautettujen järjestelmien suuntautumavaihtoehto

Tekijä: Olli Asko

Työn nimi: Sulautettujen järjestelmien väylätekniikat

Ohjaaja: Palomäki Heikki

Vuosi: 2010

Sivumäärä: 67

Liitteiden lukumäärä: 0

Tässä opinnäytetyössäni esitellään tietoliikenteessä käytettävät väylät. Tietoliikenteessä on myös muitakin väyliä kuin nämä, mutta näitä väyliä käytetään yleisimmin. Tässä opinnäytetyössä tarkastellaan väyliä sulautettujen järjestelmien kannalta.

Jokaisesta väylästä on esimerkkikuvia selkeyttämään ja havainnollistamaan kerrottua asiaa. Joihinkin väyliin on lisätty koodinpätkiä, joista käy selville väylän, ja piirin yleisimmät komennot ja myös mitä piiri tarvitsee toimiakseen.

Avainsanat: Väylä, sähköliitäntä, sanomakehys, koodi, bitti

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

THESIS ABSTRACT

Faculty: School of Technology
Degree programme: Information Technology
Specialisation: Embedded Systems

Author/s: Olli Asko

Title of the thesis: Embedded System Serial Bus

Supervisor(s): Palomäki Heikki

Year: 2010 Number of pages: 67 Number of appendices: 0

The aim of this graduate study is to present the different serial buses that are being used in telecommunication. In addition to these there are also other buses in information technology, but these are the most common serial buses in use. In this study serial buses are examined from the viewpoint of embedded systems.

Each presented serial bus is accompanied by pictures for demonstration and clarification. Parts of codes are also attached to some serial buses in order to show what the buses need to be working properly.

Keywords: buses, electrical connection, message frame, code, bit

SISÄLLYSLUETTELO

OPINNÄYTETYÖN TIIVISTELMÄ.....	2
THESIS ABSTRACT	3
SISÄLLYSLUETTELO.....	4
KÄYTETYT MERKIT JA LYHENTEET	6
KUVALUETTELO	7
1 JOHDANTO	9
2 1-WIRE	10
2.1 MIKÄ ON 1-WIRE ?	10
2.2 TOIMINTAPERIAATE.....	11
2.3 SANOMAKEHYS	12
2.4 OHJELMAESIMERKKI	14
2.5 SOVELLUKSIA.....	16
3 2-WIRE	18
3.1 MIKÄ ON 2-WIRE?	18
3.2 SÄHKÖKYTKENNÄT	18
3.3 BITTIKODAAUS	20
3.4 SANOMAKEHYS	23
3.5 OHJELMAESIMERKKI	24
3.6 KÄYTTÖTAPOJA.....	26
4 SPI.....	27
4.1 MIKÄ ON SPI?	27
4.2 SÄHKÖKYTKENTÄ	28
4.3 BITTIKODAAUS	30
4.4 SANOMAKEHYS	30
4.5 OHJELMAESIMERKKI	32
4.6 KÄYTTÖTAPOJA.....	33
5 RS232.....	34
5.1 MIKÄ ON RS232?.....	34
5.2 SÄHKÖKYTKENNÄT	34
5.3 DATAN KODAAUS.....	36
5.4 OHJELMAESIMERKKI	39
5.5 KÄYTTÖTAPOJA.....	40

6	RS485	41
6.1	MIKÄ ON RS485 (EIA-485) ?	41
6.2	SÄHKÖKYTKENNÄT	42
6.3	OHJELMAESIMERKKI	45
6.3	KÄYTTÖTAPOJA.....	47
7	MODBUS	49
7.1	YLEISTÄ.....	49
7.2	SANOMAT JA RAKENNE	51
7.3	MODBUSIN KÄYTTÖ.....	52
8	PROFIBUS	53
8.1	MIKÄ ON PROFIBUS ?	53
8.2	PROFIBUSIN TEKNIikka	53
8.3	PROFIBUSSIN KÄYTTÖTAPOJA	57
9	MUUT VÄYLÄT	58
	YHTEENVETO	59
	LÄHTEET	60

KÄYTETYT MERKIT JA LYHENTEET

Rx = receive eli vastaan ottaa dataa

Tx = Transmit eli lähettää dataa

SCL = kellopulssi väylä

SDA = dataväylä

ACK = acknowledgment = kuittaus (vastaus johonkin tapahtumaan)

KUVALUETTELO

Kuva 1. DALLAS DS180	10
Kuva 2. Kaksinastainen DS1820.....	10
Kuva 3 DS9097E-piirin sisäinen kytkentä	11
Kuva 4. 1-wiren pulssimuoto	12
Kuva 5. Sanomakehys DS2341-piirissä.....	13
Kuva 6. CRC-laskenta DS2431-piirissä	14
Kuva 7. Useampi sensori 1-wire-väylässä	17
Kuva 8. 2-wiren sähkökytkentä	18
Kuva 9 Sähköinen kytkentä usealle laitteelle 2-wire-väylässä	19
Kuva 10. AT24C1024B block diagram	19
Kuva 11. Piirin SE97BTP-nastoitus.....	20
Kuva 12. 2-wire-piiri	21
Kuva 13. Aloitus- ja lopetustilat 2-wire-väylässä	22
Kuva 14. Sanomakehys BR24Lxx-w-piirissä	23
Kuva 15. Esimerkki TWI sovelluksesta	26
Kuva 16. Usean laitteen SPI-väylä.....	28
Kuva 17. SPI:n sähköinen kytkentä	29
Kuva 18. SPI-väylät sarjassa, muunnelma	31
Kuva 19. Usean datan kirjoitus SPI-väylässä.....	31
Kuva 20. Sähköinen kytkentä RS232-piirissä	35
Kuva 21. LTC1387 piiri RS232 kytkennässä.....	35
Kuva 22. RS232-differentiaali	37
Kuva 23. RS232 signaalit.....	38
Kuva 24 RS485 verkko	42
Kuva 25. RS485 Block diagram	42
Kuva 26. RS485-Signaali	44
Kuva 27. RS485 usean laitteen kytkeminen	45
Kuva 28. Baudinopeudet PCI-COM-485-4 kortissa.....	46
Kuva 29. RS485:n käyttötapa	48
Kuva 30. Modbus TCP/IP	50
Kuva 31. ModbusRTU:n sanomakehys.....	51

Kuva 32 Modbus ascii.....	52
Kuva 33 Modbusin käyttö.....	52
Kuva 34. Profibusin sanomakehys.....	54
Kuva 35. Profibussin toimintakoodit.....	56
Kuva 36. Profibusin käyttötapa	57

1 JOHDANTO

Tässä opinnäytetyössä perehdytään sulautetuissa järjestelmissä käytettäviin väyliin ja esitellään niiden toimintaperiaatteita, mitkä niiden käyttökohteita. ONE-WIRE, TWO WIRE (TWI, I²C), SPI, RS232, RS485, MODBUS ja PROFIBUS.

Jokaisesta väylästä esitetään sähköinen kytkentä eli miten virta liikkuu, sanoma-kehys eli millä tavalla tietoa liikutetaan, ja käyttötapoja eli miten kyseistä väylätekniikkaa käytetään ja mihin sitä käytetään.

Joistakin väylistä on vielä esitetty linkkiverkko eli miten laitteet linkitetään toisiinsa, jotta ne voisivat keskustella toisten laitteiden kanssa. Työssä esitellään myös muutamia koodiesimerkkejä

Lopussa on lyhyt esittely muista väylistä.

2 1-WIRE

2.1 Mikä on 1-wire ?

1-wire on väylätekniikoissa käytettävä väylä. Kyseisen väylän on suunnitellut Dallas Semiconductor corporation. 1-wire on suunniteltu ja kehitelty sellaisille laitteille, joiden pitää antaa merkkisignaalia tai joitain arvoa. (Maxim-IC DS18S20 2008)



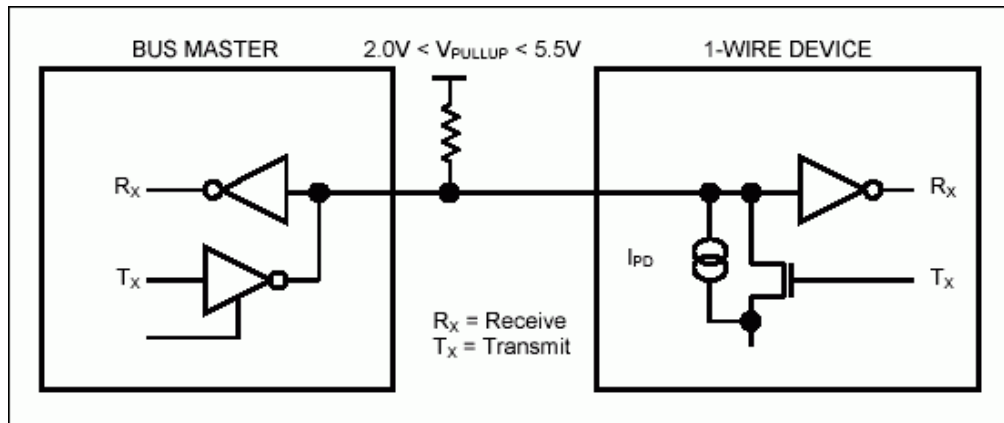
Kuva 1. DALLAS DS180
(Maxim-IC DS18S20 2008)

Kuten kuvasta (1) jo huomaa, itse laitteessa on kolme nastaa. Yleensä nastoja on vain kaksi, koska virtanasta on kytketty yhteen maanastan kanssa, jolloin siitä saadaan 2-nastainen laite. Laitteen toiseen nastaan kytketään jännite ja toiseen nastaan syötetään datat. (Maxim-IC DS18S20 2008)

Kuvassa (2) on esitetty miten 3-nastaisesta piiristä saadaan tehtyä 2-nastainen. (Maxim-IC DS18S20 2008.)



Kuva 2. Kaksinastainen DS1820
(Maxim-IC DS18S20 2008.)



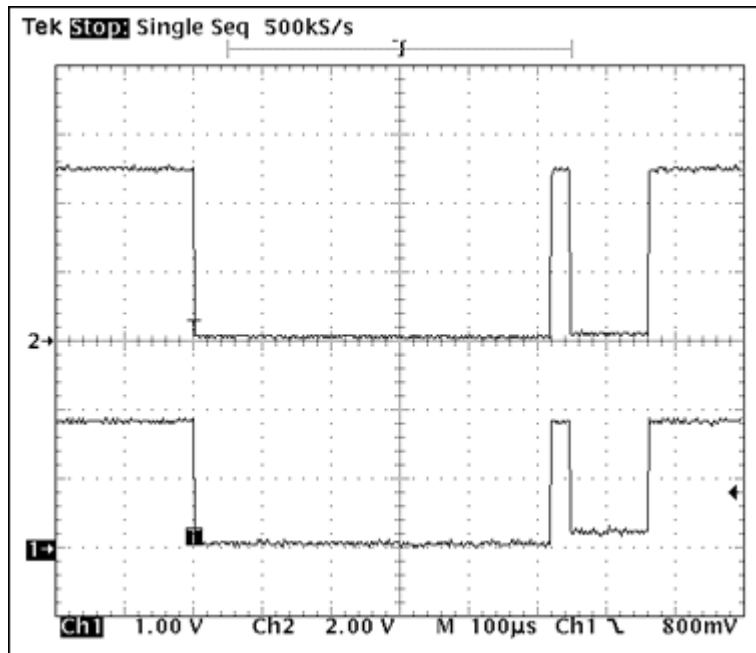
Kuva 3 DS9097E-piirin sisäinen kytkentä
(Maxim-ic AN214 2002.)

2.2 Toimintaperiaate

Kuvasta (Kuva 3) nähdään, että piiri on lepotilassa ylösvedettynä. Silloin kun piiri lähettää ja vastaanottaa tietoa (dataa), niin se menee ala-asentoon ja silloin se ottaa virran kondensaattoristaan (kyseistä kondensaattoria ei ole esitetty). (Maxim-ic AN214 2002.)

Tietokoneen ja laitteen välinen viestintä tapahtuu seuraavasti: Väylää hallitseva laite (isäntäkone) asettaa väylän alatilaa 480–960 mikrosekunnin (nollauspulssi) ajaksi ja sen jälkeen se jää odottamaan toisen laitteen vastausta. Toinen laite (orja) vastaa siihen vetämällä oman väylän alatilaa 15–60 mikrosekunnin päästä saatuaan isännältä merkin. Orja pitää väylää alatilassa 60 - 240 mikrosekuntia näin se ilmoittaa isäntäkoneelle olevansa käytettävissä. (Maxim-ic AN214 2002.)

Kuvassa (Kuva 4) on esitetty yllämainittu asia pulssimuodossa.

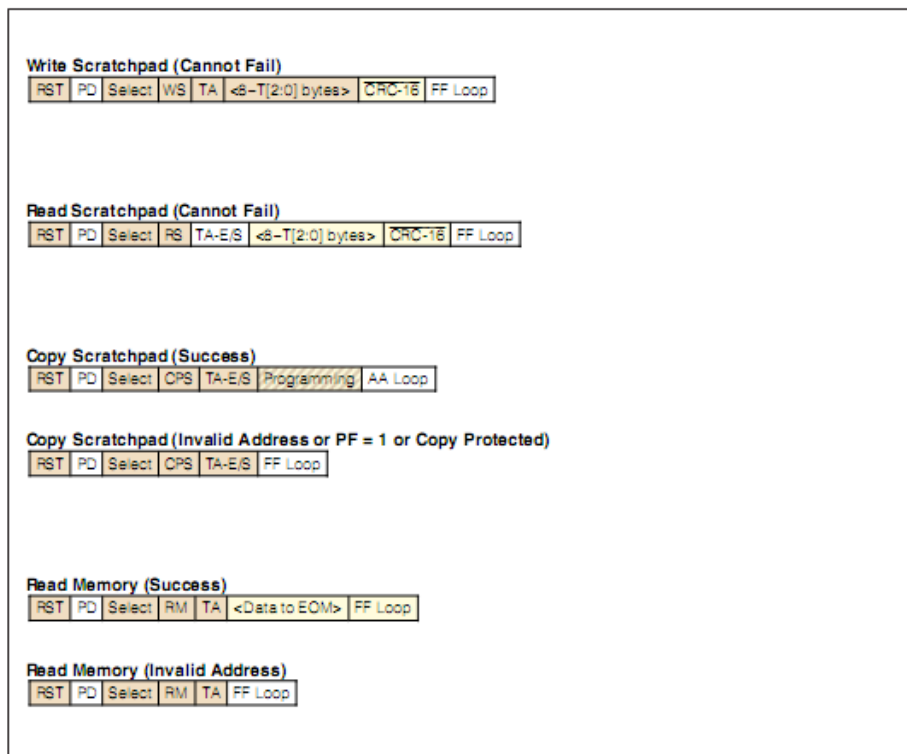


Kuva 4. 1-wiren pulssimuoto
(Maxim-IC4477)

2.3 Sanomakehys

Kuvassa (Kuva 5) on piirroksella selitetty 1-wire väylän sanoman kulku.

1-Wire Communication Examples

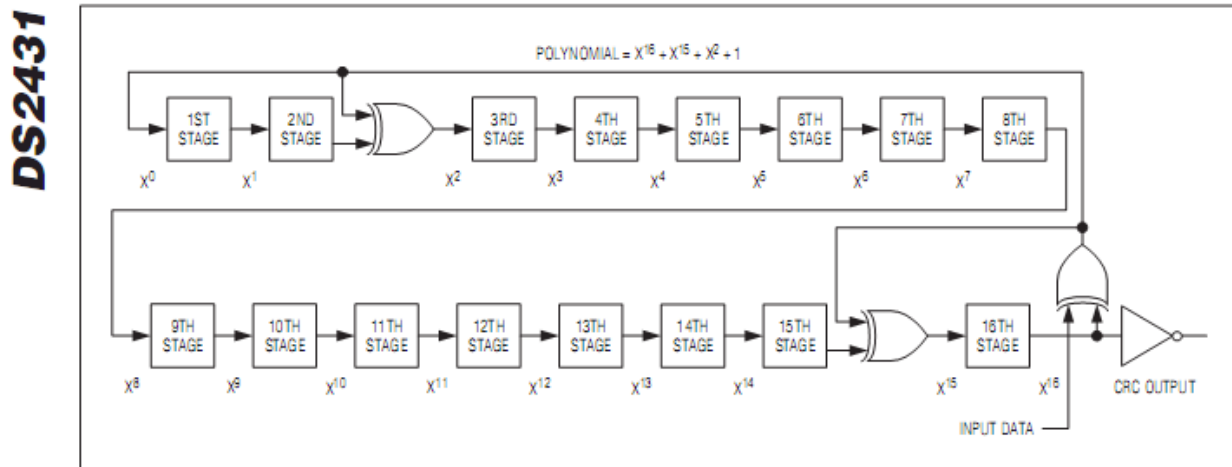


**Kuva 5. Sanomakehys DS2341-piirissä
(Maxim-IC DS2431 2009)**

1-wiren sanomakehys on siis kuvassa esitetty, esimerkkinä on muistin lukeminen (Read Memory). Lukemisen protokolla on seuraavanlainen. Ensimmäiseksi Isäntä lähettää resetoitipulssin (RST) ja orja vastaa siihen vastauspulssilla (PD). Sen jälkeen tulee valintapulssi, mitä isäntäkone haluaa tehdä (SELECT). Muistin lukukäsky tapahtuu seuraavaksi (RM) ja sen jälkeen lähetetään kohdeosoite (TA). Sen jälkeen luetaan osoitteen muisti (data to EOM). Kuvassa (Kuva 5) on vielä silmukka, joka toistaa äskeiset tapahtumat (FF LOOP). (Maxim-ic DS2431 2009.)

Virheentarkistuksia tällä piirillä on kahdenlaisia: 8-bittinen CRC ja 16-bittinen CRC. Kuvassa (Kuva 6) on esitetty 16 bittinen CRC-laskenta.

1024-Bit, 1-Wire EEPROM



**Kuva 6. CRC-laskenta DS2431-piirissä
(Maxim-IC DS2431 2009)**

CRC tarkistuksessa viestille lasketaan arvo, joka esitetään kuvassa (6). Arvo lasketaan seuraavasti $X^{16} + X^{15} + X^2 + 1$ ja siitä saadaan 16-bitin tai 8-bitin pituinen tarkistussumma. (Maxim-ic DS2431 2009.)

2.4 Ohjelmaesimerkki

Seuraavassa on esimerkkikoodi 1-wire:n käytöstä. Oheinen koodi on jaettu kolmeen osaan ensimmäisessä osassa 1-wire-väylä resetoidaan (uudelleen käynnistetään) ja toisessa vaiheessa master kirjoittaa ja kolmannessa osassa se lukee kirjoituksen. Tässä olevat koodinpätkät on tarkoitettu muun muassa seuraaville piireille DS2480B ja DS2482.

Koodissa on tärkeimmät komennot, jotka toimivat piireillä DS2480B ja DS2482. Piirissä on PORTADDRESS ja sen arvo, joka on esitetty hexana. Kyseiseen komentoon laitetaan piirin portin arvo. (Maxim-ic AN126 2006.)

Esimerkki 1. 1-wiren perustoiminnot. (Maxim-ic AN126 2006.)

```
//-----
// Generate a 1-Wire reset, return 1 if no presence detect was found,
// return 0 otherwise.
// (NOTE: Does not handle alarm presence from DS2404/DS1994)
//
int OWTouchReset(void)
{
    int result;

    tickDelay(G);
    outp(PORTADDRESS,0x00); // Drives DQ low
    tickDelay(H);
    outp(PORTADDRESS,0x01); // Releases the bus
    tickDelay(I);
    result = inp(PORTADDRESS) ^ 0x01; // Sample for presence pulse from slave
    tickDelay(J); // Complete the reset sequence recovery
    return result; // Return sample presence pulse result
}

//-----
// Send a 1-Wire write bit. Provide 10us recovery time.
void OWWriteBit(int bit)
{
    if (bit)
    {
        // Write '1' bit
        outp(PORTADDRESS,0x00); // Drives DQ low
        tickDelay(A);
        outp(PORTADDRESS,0x01); // Releases the bus
        tickDelay(B); // Complete the time slot and 10us recovery
    }
    else
    {
        // Write '0' bit
        outp(PORTADDRESS,0x00); // Drives DQ low
        tickDelay(C);
        outp(PORTADDRESS,0x01); // Releases the bus
        tickDelay(D);
    }
}

//-----
// Read a bit from the 1-Wire bus and return it. Provide 10us recovery time.
//
int OWReadBit(void)
{
    int result;

    outp(PORTADDRESS,0x00); // Drives DQ low
    tickDelay(A);
    outp(PORTADDRESS,0x01); // Releases the bus
    tickDelay(E);
    result = inp(PORTADDRESS) & 0x01; // Sample the bit value from the slave
    tickDelay(F); // Complete the time slot and 10us recovery

    return result;
}
```

Oheinen koodi on vain esimerkki kuinka kyseisellä väylällä voidaan lukea ja kirjoittaa. Tarkempaa tietoa löytyy kiinnostuneille oheisesta linkistä

<http://www.aag.com.mx/aagusa/contents/en-us/AN126.pdf>

Seuraavassa esitellään tarkemmin bitin koodausta: Viestin lähetys tapahtuu siten, että isäntäkone lähettää orjakoneelle nollauspulssin eli se vetää väylän alatilaan 480–960 mikrosekunnin ajaksi ja sen jälkeen se jää odottamaan orjakoneen kuitausta, joka tapahtuu niin, että orjakone vetää 15–50 mikrosekunnin kuluttua väylän alatilaan 60 – 240 mikrosekunnin ajaksi. Sanomakehys alkaa vähiten merkittävällä bitillä ja sanomakehykseen mahtuu 8 bittiä. (1-wire [viitattu 6.4.2010].)

Itse viestin lähettäminen tapahtuu tiloja vaihtamalla. Isäntäkoneen kirjoitus tapahtuu siten, että isäntäkone vetää väylän hetkeksi aikaa alatilaan, tästä hetkestä 15–60 mikrosekunnin päästä orjakone lukee tilan. (1-wire [viitattu 6.4.2010].)

Kirjoitustila pitää kestää vähintään 60 mikrosekuntia. Jos kyseinen tila on vielä lukuhetkelläkin alatilassa, orjakone tulkitsee sen nollaksi. Jos ylätila on päällä lukuhetkellä, niin orjakone tulkitsee sen ykköseksi. Jokaisen merkin välissä pitää olla vähintään yhden mikrosekunnin tauko. (1-wire [viitattu 6.4.2010].)

Lukutilassa isäntäkone laittaa vähintään yhden mikrosekunnin ajaksi väylän alas ja sen jälkeen se lukee orjalta tilan. Jos orjakoneessa on tila nollassa, niin silloin orja vetää välittömästi lukupulssin jälkeen tilan nollaksi. Kun taas tilassa 1 orja ei vedä tilaa alas, vaan pitää kyseisen tilan. (1-wire [viitattu 6.4.2010].)

2.5 Sovelluksia

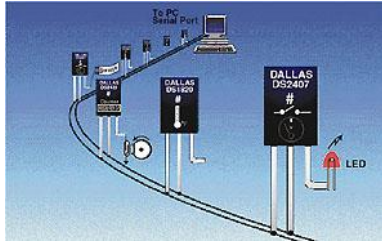
1-wire sovelluksia käytetään muun muassa seuraavasti:

- lämpötilan mittaamiseen
- akun virran näyttämiseen
- sarjanumeron tekemiseen (serial number button)

- erilaisiin anturisovelluksiin

- A/D Muuntimena. (1-wire [viitattu 6.4.2010].)

Kuvassa (Kuva 7) on esitetty usean laitteen kytkeminen 1-wire-väylään.



**Kuva 7. Useampi sensori 1-wire-väylässä
(Audon Elecktrics)**

3 2-WIRE

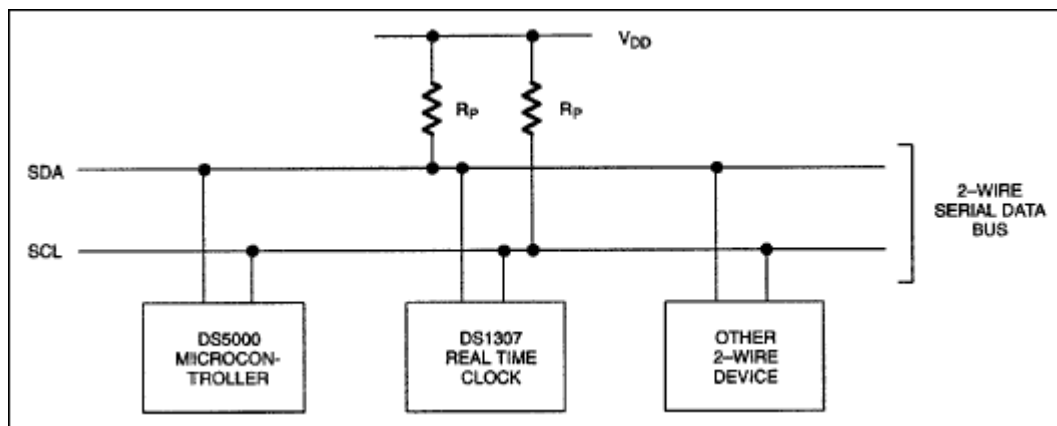
3.1 Mikä on 2-wire?

2-wire, toiselta nimeltään I²C tai TWI, on väylä, jolla laitteet pystyvät keskustelemaan toisten laitteiden kanssa. 2-wire-piirillä oli ensisijaisesti tarkoitus viestiä hitaiden oheispiirien kanssa. (2-wire 2010.)

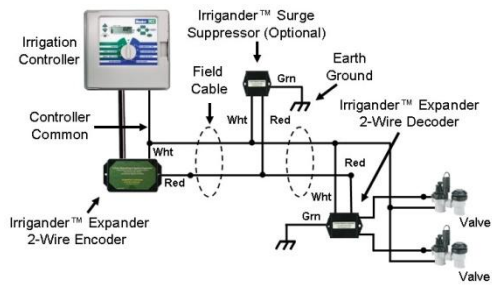
2-wire väylän kehitti 1990-luvulla Philips ja myöhemmin sitä käytti muun muassa Motorola ja Siemens AG (myöhemmin Infineon Technologies AG). (2 wire 2010.)

2-wire väylä eroaa 1-wire väylästä siinä, että 1-wire väylässä kulkee data ja virta. 2-wire-väylässä kulkee data ja kellopulssi. Isäntäkone ohjaa kellopulssia. Väylässä voi olla useampi isäntäkone ja niillä kaikilla on oma kellopulssi. (2 wire 2010.)

3.2 Sähkökytkennät

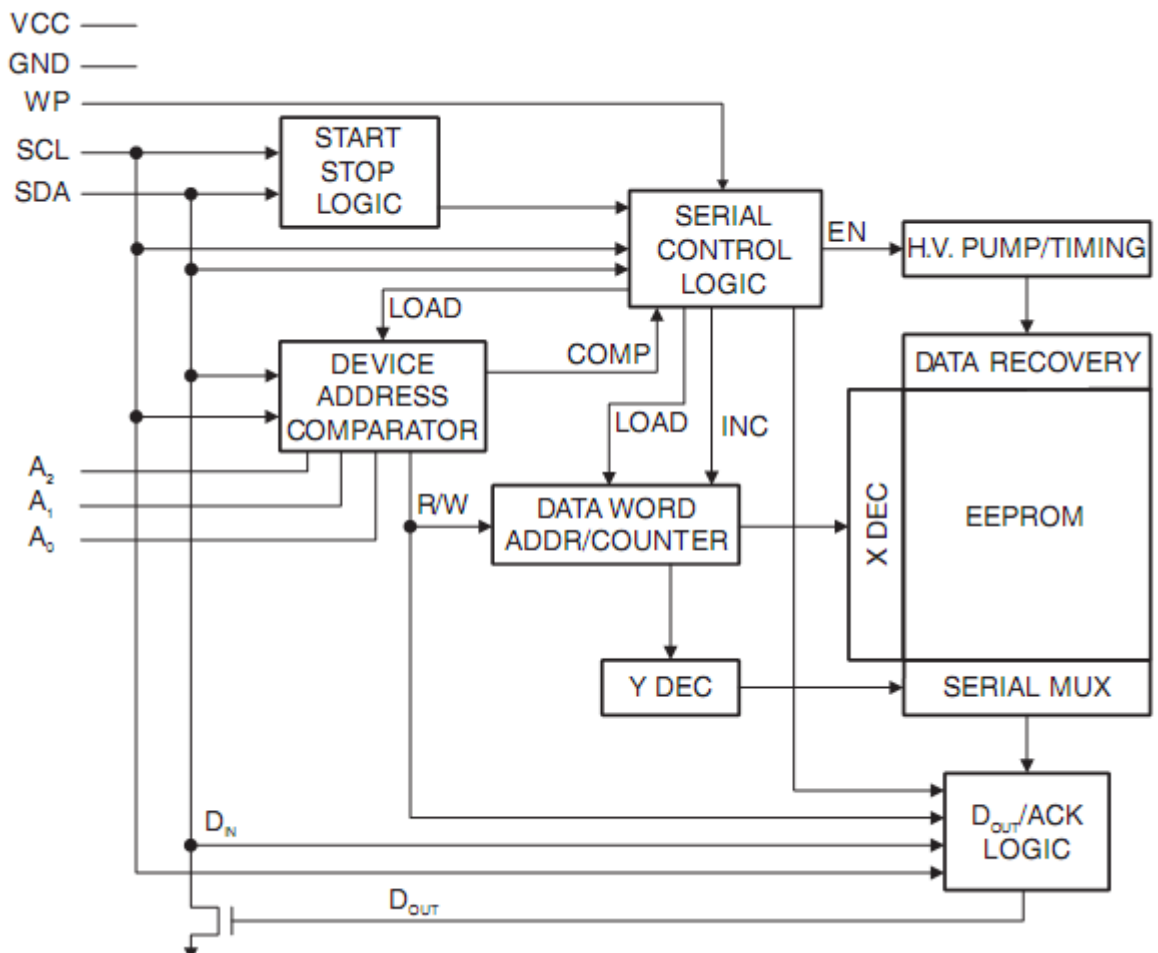


Kuva 8. 2-wiren sähkökytkentä
(Elecfans 200942111426289.gif)



**Kuva 9 Sähköinen kytkentä usealle laitteelle 2-wire-väylässä
(Irrigander 2-wire)**

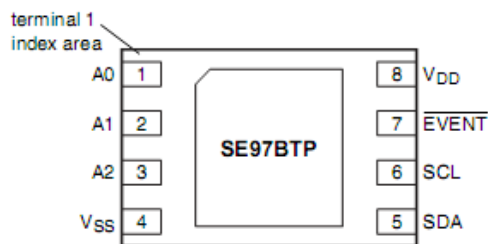
Kuvassa (8) ja kuvassa (9) on esitetty, miten useampi laite kytketään samaan väylään Kuva (10) esittää yhden piirin sähköistä kytkentää, tässä tapauksessa piiri on AT24C1024B.



**Kuva 10. AT24C1024B block diagram
(Atmel doc5194 1997)**

Kuvasta (10) nähdään, että SCL-väylä antaa kellopulssia ja SDA-väylä syöttää dataa. Muita piirin väyliä on kirjoitus suojaus (WP) väylä ja osoiteväylät (A0, A1 ja A2). (Atmel doc5194 1997.)

Kuvassa (11) on esitetty millä tavalla esimerkiksi SE97BTP-piirin voi kytkeä I²C-väylään.



**Kuva 11. Piirin SE97BTP-nastoitus
(NXP SE97B 2008)**

Piirin nastoitus on seuraava:

- Nastat 1 - 3 ovat ulkoisia osoitenastoja.
- Nasta numero 4 on piirin maanasta.
- Nasta 5 on datanasta, jolla luetaan/kirjoitetaan piirin datat.
- 6. nastaan tulee piirin kellopulssi.
- 7. nasta on tapahtuma (event) -nasta, johon tulee lämpötilaanturin hälytykset (liian korkea tai liian pieni lämpötila)
- Viimeiseen eli 8 nastaan syötetään piirin jännite. (NXP SE97B 2008.)

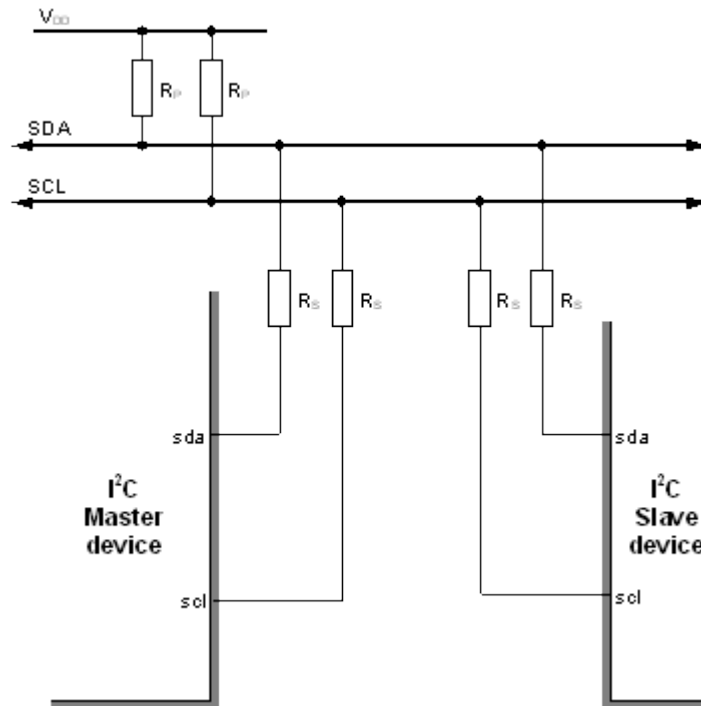
3.3 Bittikoodaus

2-wire-verkossa on neljä mahdollista tilaa, jotka ovat:

- isäntäkone lukee tietoa orjakoneelta
- isäntäkone lähettää tietoa orjakoneelle

- orjakone vastaanottaa tietoa isäntäkoneelta
- orjakone lähettää tietoa isäntäkoneelle. (Digital Core Design i2c-bus specification 2002.)

)



Kuva 12. 2-wire-piiri
(Digital Core Design i2c-bus specification 2002)

Kuvassa (12) on esitetty miten saadaan kaksi laitetta kytkettyä verkon välityksellä toisiinsa.

Viestin lähetyksessä 2-wire verkossa tapahtuu siten, että isäntäkone menee lähetystilaan ja lähettää orjakoneen osoitteen ja tiedon kirjoittaako/lukeeko se dataa. Sen jälkeen isäntäkone jää odottamaan kuittausta kyseiseltä orjakoneelta. Osoitteen pituus on 7 merkkiä ja kahdeksas bitti ilmaisee haluaako isäntäkone lukea vai kirjoittaa dataa. 0-tilassa isäntäkone haluaa lähettää ja 1-tilassa isäntäkone haluaa lukea dataa orjakoneelta. Viestin lähettäminen tapahtuu bitteinä, joita pitää aina olla 8, jonka jälkeen vastaanottava laite lähettää kuittauksen lähettäjälle. (Digital Core Design i2c-bus specification 2002.)

Kun kaikki viestit on lähetetty, niin isäntäkone laittaa itsensä nollatilaan. Kuvassa (12) on esitetty miten kaikki tapahtuvat signaalitasolla. (Digital Core Design i2c-bus specification 2002.)



**Kuva 13. Aloitus- ja lopetustilat 2-wire-väylässä
(Digital Core Design i2c-bus specification 2002)**

Kuvassa (13) on esitetty aloitus- ja lopetustila 2-wire piiriissä. Isäntäkoneen aloitustilassa signaalit ovat siten, että isäntäkone asettaa dataväylän nollatilaan ja samaan aikaan kellopulssi pysyy hetken aikaa vielä 1-tilassa (aloitustila) ja sen jälkeen menee nollatilaan. Viestin osoitteen lähettäminen tapahtuu siten, että isäntäkone vetää kellopulssin hetkeksi ylös ja sen jälkeen alkaa lähettää säännöllistä kellopulssia joka on muodoltaan kanttista. Osoite lähetetään nousevalla kellopulsilla. Osoitteen suunta on, että eniten merkitsevä bitti lähetetään ensiksi. (Digital Core Design i2c-bus specification 2002.)

Kun kaikki 8-bittiä on lähetetty, jää isäntäkone odottamaan kuittausta orjakoneelta. Jos orjakone on tunnustanut osoitteensa ja saanut tilaa koskevan tiedon, niin sen jälkeen se kuittaa viestin vetämällä väylänsä alatilaan. (Digital Core Design i2c-bus specification 2002.)

Tämän jälkeen isäntäkone ja orjakone keskustelevat niin kauan kunnes isäntäkone menee lopetustilaan. Jokainen viesti tapahtuu kahdeksalla bitillä ja aina 8 bitin jälkeen isäntäkone odottaa orjakoneelta kuittausta. Piirin voi koodata siten, että isäntäkone lähettää tietyn aikaa dataa, jos orjakone ei sen jälkeen vastaa niin isäntäkone menee lepotilaan. (Digital Core Design i2c-bus specification 2002.)

Kun isäntäkone on saanut kaiken datan lähetettyä/luettua niin se lopettaa keskustelun menemällä lopetustilaan siten, että dataväylä menee nollatilaan ja kellopulsiväylä menee ykköstitilaan hetkeksi aikaa. Sen jälkeen isäntäkone menee ykkösti-

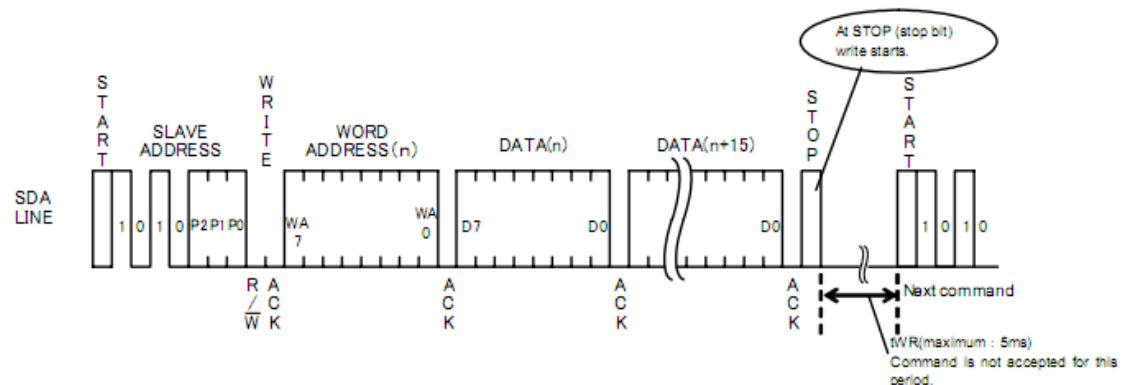
laan ja kellovähylä pitää ykköstilaa päällä. (Robot Electronics i2c-bus specification 2002);(Robot Electronics using I2C [viitattu 6.4.2010].)

3.4 Sanomakehys

Seuraavaksi esitellään kuinka kaksi laitetta tai useampi laite pystyvät keskustelemaan keskenään 2-wire-vähylässä.

Kuvassa (14) on esitetty sanomakehys, jossa on jatkuva syöttö päällä.

○ Notes on write cycle continuous input



**Kuva 14. Sanomakehys BR24Lxx-w-piirissä
(Rohm semiconductor br24101a-w-e 2008)**

Sanoma kulkee seuraavasti datavähylässä: Isäntäkone lähettää datavähylää pitkin orjalle osoitteen ja ilmoituksen siitä mitä se haluaa tehdä (lukea/kirjoittaa). Tässä esimerkissä on kirjoitustila päällä. Kun isäntäkone on lähettänyt osoitteen ja tilan niin se jää odottamaan orjakoneen kuittausta. (Rohm semiconductor br24101a-w-e 2008.)

Kuittauksen saatuaan isäntäkone lähettää osoitteen, johon kirjoitetaan data. Sen jälkeen se jää odottamaan orjakoneen kuittausta. Kun kuittaus on saatu, itse data lähetetään, jonka jälkeen isäntäkone odottaa kuittausta. (Rohm semiconductor br24101a-w-e 2008.)

Esimerkiksi 8-bittisessä piirissä "ASKO" (huom. kaikki isoilla kirjaimilla) bitit liikkuu seuraavasti:

- Ensiksi nimi pitää muuntaa 8-bitin sarjaksi.
- A = 65 joka taas on binaarisen 01000001
- S = 115 → 01110011
- K = 107 → 01101011
- O = 111 → 1101111

Kun nimi on saatu muunnettua bittisarjoiksi, niin otetaan osoite esimerkiksi 1100011. Ja päätetään, että halutaan kirjoittaa viestit.

Sanomakehyksessä oheinen tehtävä näyttäisi tällaiselta:

Ensimmäiseksi lähetettäisiin orjaosoite, joka on tässä tapauksessa 1100011 ja siihen lisättäisiin perään 0, joka ilmaisee kirjoitustilaa.

Tämän jälkeen orjakone vastaa yhdellä pulssilla (kuittauspulssi). Tämän saatua isäntäkone lähettää ensimmäisen merkin A-kirjaimen (01000001). Jonka jälkeen se jää odottamaan kuittausta. Loput merkit lähetetään samaan tapaan ja viestien loputtua sanoma lopetetaan vetämällä kellopulssiväylä ylös ja dataväylä hetken kuluttua. (ascii [viitattu 6.4.2010].)

3.5 Ohjelmaesimerkki

Esimerkkilaitteeksi on otettu ATmega16 piiri ja oheiset komennot on otettu kyseisen laitteen datalehdessä.

Piirin alustukseen käytetään seuraavia komentoja:

TWSR = 0 tällä komennolla asetetaan STATUS REGISTER nollassi

TWBR = 60 oheisella komennolla laitetaan väylän taajuus kohdalleen

TWCR = 0b10000100 laittaa twi:n päälle, jolloin väylää käytetään oikein.

Käynnistys tapahtuu komennolla TWCR= 0b10100100 HUOMAA Binaari
(Atmel Atmega16 2003)

Kirjoitustilassa pitää olla seuraavat asiat.

TWDR = data tällä komennolla piiri lukee datan rekisteriin (merkki kerrallaan)

Kontrollirekisterissä TWCR:ssä pitää olla tila enableTWI, eli koodina se on TWCR = 0b10000100

Lukutilassa pitää tehdä seuraavat asetukset:

Kontrollirekisterissä tilan pitää olla 0b11000100, eli tässä asetetaan TWI ja TWEA päälle.

Muuta huomioitavaa koodista: osoitteeksi pitää asettaa EEPROMILLE jonne, piiri kirjoittaa merkki kerrallaan dataa. Alla olevassa koodissa on näytetty yksinkertainen 2-wire-väylällä tehty, merkki kerrallaan kirjoittava koodi. (Atmel Atmega16 2003.)

Esimerkki 2 perus TWI ohjelma

```
void TWIinit(void) PIIRIN ALUSTUS
{
    TWSR = 0;
    TWBR = 60;
    TWCR = 0b10000100;           //enable TWI
}

void TWIstart(void)           TWI:N KÄYNNISTYS
{
    int to=1000;
    TWCR = 0b10100100;         // START + int nollaus
    while(((TWCR & 0x80)==0) && (to-->0)) asm("nop"); //odota kuittausta
}

void TWIstop(void)           TWI:N PYSÄYTYS
{
    TWCR = 0b10010100;         // STOP
}

void TWIwrite(unsigned char tavu) LÄHETETÄÄN TAVU EEPROMILLE KIRJOITETTAVAKSI
{
    int to=1000;
    TWDR = tavu;               //lähetä tavu
    TWCR = 0b10000100;         //enable TWI
}
```

```

while(((TWCR & 0x80)==0) && (to-->0)) asm("nop"); //odota kuittausta
}

void EEput(unsigned int adr, unsigned char dat, char typ)
{
    TWIstart();
    TWIwrite(0xA0); //piiriosite 0, write
    if(typ) TWIwrite(adr>>8); //osoitteen ylempi tavu
    TWIwrite((char)adr); //osoitteen alempi tavu
    TWIwrite(dat); //kirjoita datatavu
    TWIstop();
}

```

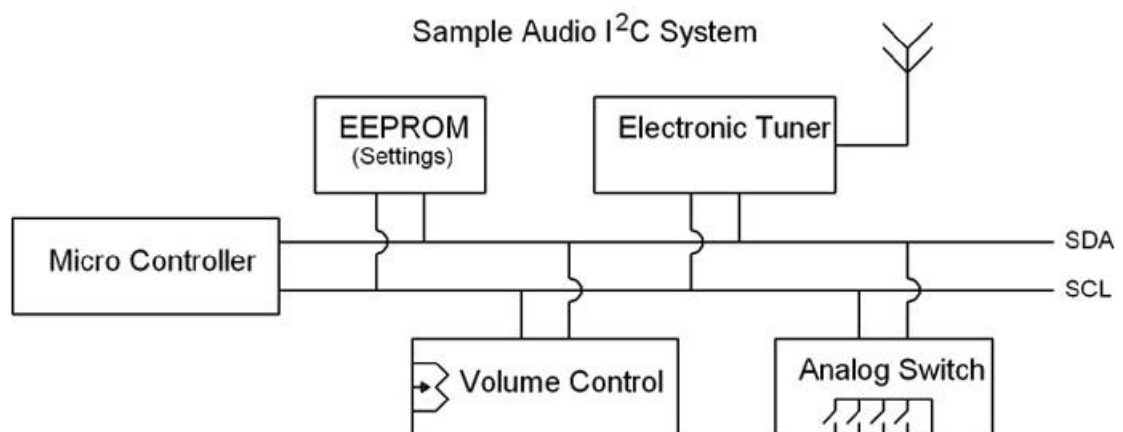
KIRJOITETAAN TAVU

3.6 Käyttötapoja

2-wire väylää käytetään muun muassa:

- lcd/led ohjauksessa
- eeprommissa
- kapasitiivisenä sensorina
- kelloissa
- antureissa (2-wire overview)

Kuvassa (15) esimerkki sovelluksesta



**Kuva 15. Esimerkki TWI sovelluksesta
(Control Resources Incorporated)**

4 SPI

4.1 Mikä on SPI?

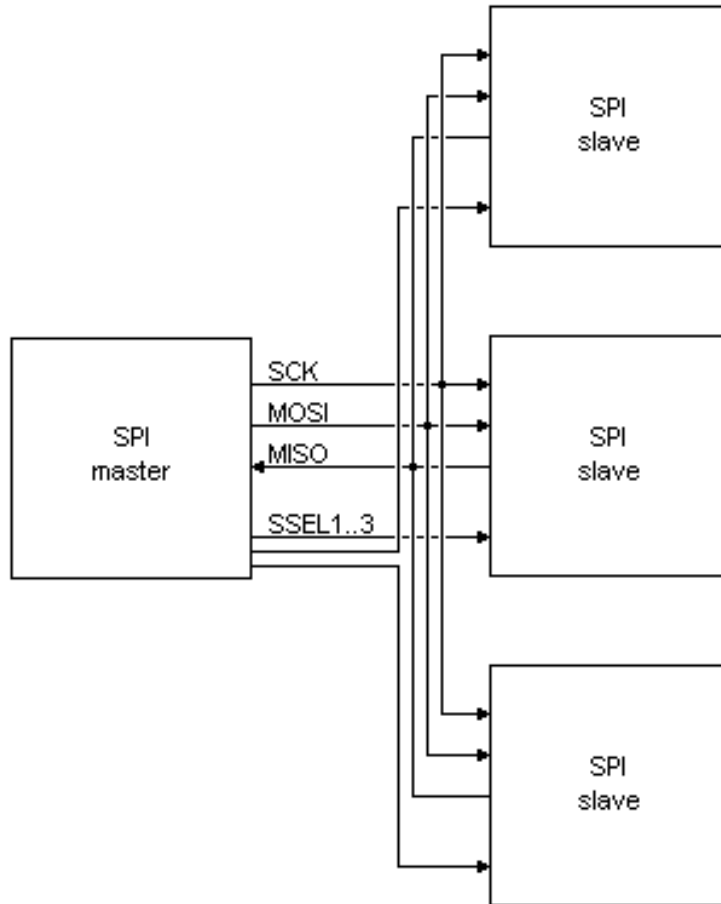
SPI (**S**erial **P**eripheral **I**nterface **B**us) on tiedonsiirrossa käytettävä laite. SPI väylä eroaa aikaisemmista väylistä siinä, että sillä on neljä eri signaalijohdinta. SPI:llä on yksi kellopulssi, jolla se antaa muulle piirille tahdin, johon pitää samaistua (synkronoitua). (SPI 2009.)

Toinen on nimeltään MOSI (**M**aster **O**utput **S**lave **I**nput), josta isäntäkone lähettää dataa orjakoneille, ja orjakone vastaanottaa dataa. Kolmas väylä on käänteinen versio edellisestä väylästä, eli isäntäkone vastaanottaa dataa ja orjakone lähettää dataa. Tämän väylän nimi on MISO (**M**aster **I**nput **S**lave **O**utput). (SPI 2009.)

Viimeinen SPI:ssä on SS (**S**lave **S**elect). SPI:ssä isäntäkoneen ja orjakoneen välissä pitää olla aina yksi SS-väylä. Tiedon siirto SPI:llä ei siis tapahdu osoitteen perusteella, kun isäntäkone haluaa lähettää/lukea dataa oheislaitteesta niin se käyttää sille menevää SS-johdinta. (SPI 2009.)

SPI käyttää FULL DUPLEX:ia eli kun isäntäkone lukee dataa tietyltä orjakoneelta, niin se samaan aikaan myös lähettää dataa orjakoneelle (tarvittaessa). (SPI 2009.)

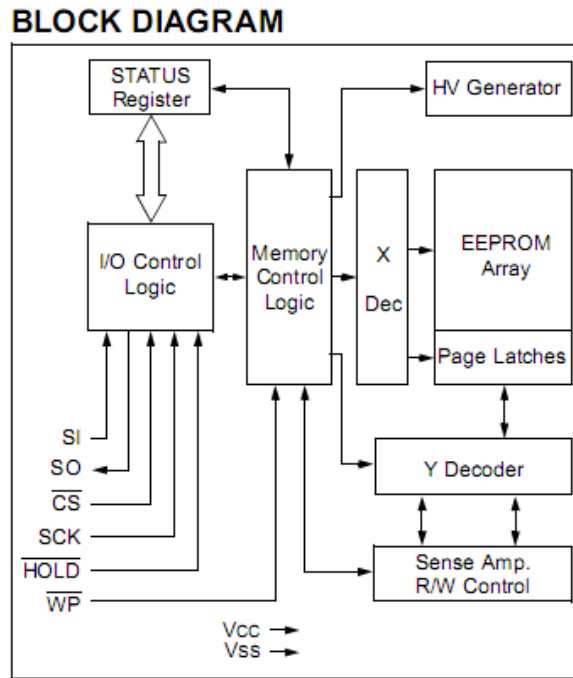
Kuvassa (16) on esitetty, miten SPI-väylä kytketään toisiin väyliin. (SPI 2009.)



Kuva 16. Usean laitteen SPI-väylä
(Fpga4fun)

4.2 Sähkökytkentä

Kuvassa (17) on esitetty 25AA010A-piirin sähköinen kaaviokuva.



**Kuva 17. SPI:n sähköinen kytkentä
(Microchip 21832C 2006)**

Kuvassa (17) on Microchipin tekemä eeprom muistipiiriin sähköinen kytkentä, jossa on käytetty SPI väylää

Kuvassa (17) on seuraavat merkinnät:

- SI tarkoittaa signaali sisään, eli se on edellä mainittu MISO-väylä.
- SO piirroksessa kuvaa MOSI-väylää.
- SCK on edellä mainittu kellopulssiväylä.
- HOLD kuvastaa sitä, jos piiri halutaan pitää tietyssä tilassa.
- WP tarkoittaa kirjoitussuojaa.
- Vcc ja Vss kuvaavat virtaa ja maadoitusta. (Microchip 21832C 2006.)

4.3 Bittikoodaus

Itse yhden bitin tiedonsiirto tapahtuu seuraavassa järjestyksessä: Ensimmäiseksi isäntäkone asettaa pulssitaajuuden orjakoneille sopivaksi. Tämän jälkeen isäntäkone päättää kenelle se haluaa tietoa lähettää tai minkä orjakoneen tietoa se haluaa lukea. Kun isäntäkone on päättänyt minkä koneen kanssa se haluaa keskustella, niin se vetää kohdekoneen Slave Select-väylän alas, jolla se ilmoittaa, että se haluaa lukea/kirjoittaa kyseisen koneen kanssa. Jokaisella kellopulssilla siirtyy yksi bitti dataa. Kellopulssin polaarisuus määrittää lähteekö data nousevalla vai laskevalla reunalla. (Motorola Inc S12SPIV3 [viitattu 6.4.2010].)

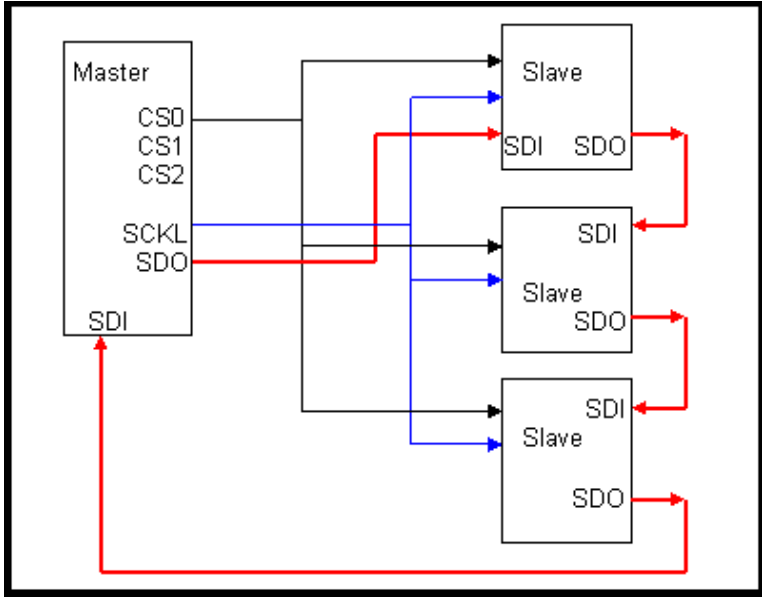
Kahdeksan bitin jälkeen Slave Select nousee ykköstitilaan ja viestintä päättyy. (Motorola Inc S12SPIV3 [viitattu 6.4.2010].)

4.4 Sanomakehys

Edellä käsitelty bittikoodaus tarkoitti yhden tavun siirtoa, eli se oli yksi osa sanomakehystä. Itse sanomakehys on seuraavanlainen: Ensimmäiset 8-bittiä on piirin ohje, jolla ilmaistaan mitä isäntäkone haluaa tehdä.

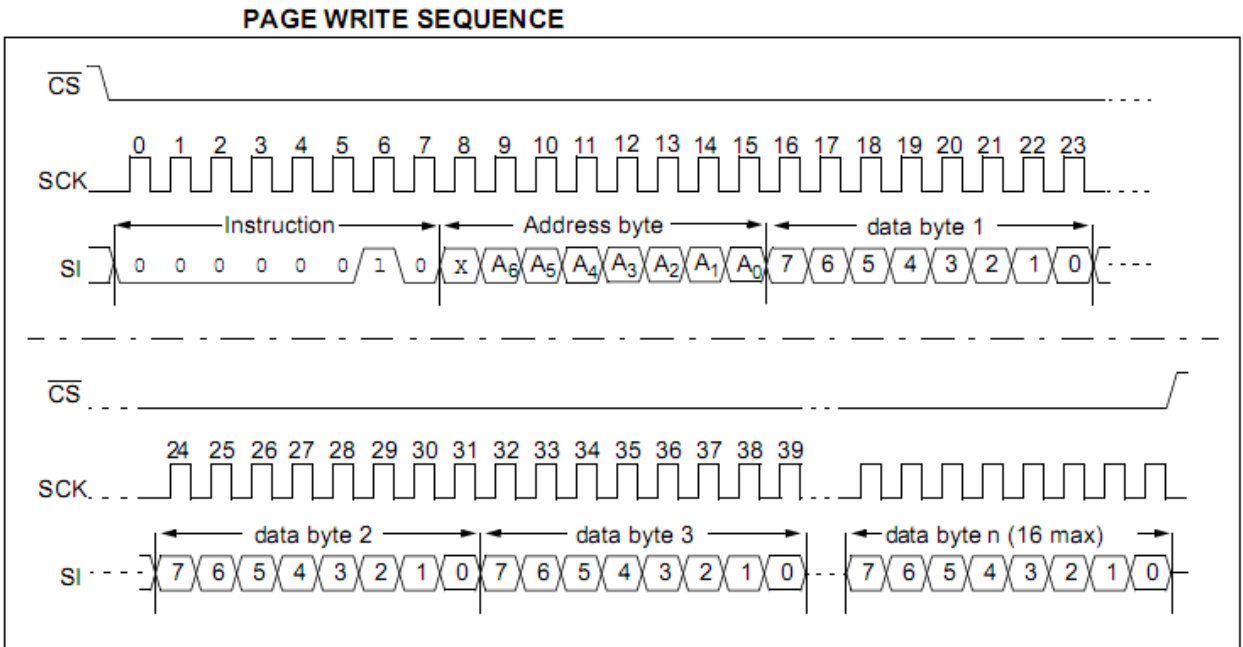
Isäntäkone voi esimerkiksi haluta kirjoittaa tiettyyn osoitteeseen dataa tai haluta lukea tilan rekisteristä. Sanomakehyksessä seuraava asia on osoitteen lähettäminen, jos monta laitetta on kytketty sarjaan pitää lähettää osoite, jotta tiedetään mille laitteelle halutaan lähettää. Kuvassa (18). Tällöin orjakone tietää kenelle data on tulossa. (Motorola Inc S12SPIV3 [viitattu 6.4.2010].)

Yleensä isäntäkoneelta on jokaiselle orjakoneelle oma johdin jolla se ilmoittaa keneltä se haluaa lukea. (Motorola Inc S12SPIV3 [viitattu 6.4.2010].)



Kuva 18. SPI-väylät sarjassa, muunnelma
(MCT Paul & Scherer [viitattu 7.4.2010].)

Kun 8-bittinen osoite on lähetetty, niin itse data lähetetään seuraavaksi ja datojen välissä ei ole mitään kuittaustapahtumaa, kuten kuvasta (19) käy ilmi.



Kuva 19. Usean datan kirjoitus SPI-väylässä
(Microchip 21832C 2006.)

Koko sanomakehysesimerkki on tehty Microchipin 25AA010A/25LC010A-piirin ohjekirjan mukaan. Jotkin asiat voivat muuttua muissa piireissä. Esimerkiksi tässä piirissä voi olla maksimissaan 16 kahdeksan-bitin merkkijonoa. (Microchip 21832C 2006.)

4.5 Ohjelmaesimerkki

Esimerkki 3 SPI-väylän koodaus

oleva koodi on vain esimerkki kuinka koodataan SPI väylää.

```
//
// SPI -väylän alustus
//
void SPInit(void)
    {
        DDRB = 0b10101000;           //SCK, MOSI, SS lähtöjä
MISO tulo        SPCR = 0b01010011; //SPI päälle, hidas nope-
us
    }
//
// SPI -slave select päälle
//
void SSON(void)
    {
        PORTB &= 0b11110111;       //SS päälle
    }
//
// SPI -slave select pois päältä
//
void SSOFF(void)
    {
        PORTB |= 0b00001000;       //SS pois päältä
    }
//
// SPI -datan siirto
//
unsigned char SPIdata(unsigned char dat)
    {
        SPDR = dat;                 //lähetä data orjalle
        while((SPSR & 0x80)==0);    //odota, että lähtee
        return SPDR;                //luetaan orjan lähettämä
data
    }
```

Koodissa on esitetty miten saadaan data kulkemaan SPI-väylässä. Seuraava teksti keskittyy selittämään yllä olevaa koodia niin, että siitä saisi hyvin selkoa. Oheiset tiedot löytyvät Atmega16- ja Atmega32-datakirjoista osoitteista:

http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf ja

<http://www.atmel.com/atmel/acrobat/doc2503.pdf>

Ensiksi piiri täytyy alustaa ja alustuksessa on käytetty seuraavia komentoja: DDRB = 0b10101000 eli oheisessa komennossa on laitettu nastat 4, 6 ja 8 ykköstilaan, mikä merkitsee sitä, että piiri antaa kellopulssia. Oheisesta koodista käy myös ilmi se, että MOSI-tila on käytössä eli isäntäkone lukee tietoa ja orjakone syöttää sitä.

Kaikkien nastojen tiedot saadaan selville datalehdestä. edellä olevassa koodissa tiedonkulun nopeus asetettiin hitaaksi komennolla SPCR = 0b01010011;

Oheinen koodi tarkoittaa sitä, että SPI on päällä. Piiri on isäntä ja piirin taajuus on $F_{osc}/64$, F_{osc} on systeemin kello. (Atmel Doc2466, Atmel doc2503 2009)

Chip Select (myös **slave select**) asetetaan päälle komennolla PORTB &= 0b11110111 ja pois päältä komennolla PORTB |= 0b00001000. Datan lähetys onnistuu komennolla SPDR = viesti, jossa viesti on haluttu datatavu. (Atmel Doc2466; Atmel doc2503 2009.)

4.6 Käyttötapoja

SPI väylää käytetään esimerkiksi seuraavissa laitteissa:

- LCD näytöissä
- kosketusnäytöissä
- muisteissa esimerkiksi (flash, eeprom, mmc ja sc)
- erilaisissa raja-antureissa. (Atmel Doc2466; Atmel doc2503 2009.)

5 RS232

5.1 Mikä on RS232?

RS232 on tietoliikenneportti kahden laitteen välillä. RS232-väylää käytetään esimerkiksi kahden tietokoneen kytkemisessä toisiinsa (nk. nollamodeemina). Tässä projektissa kuitenkin käsitellään sulautettujen järjestelmien väyliä, joten tässä käytetään komponenttia jossa on RS232. (WiseGEEk 2010.)

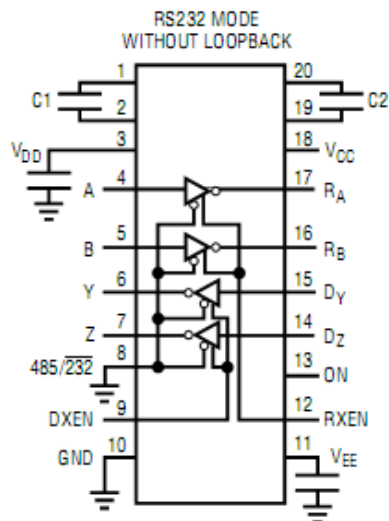
RS232 on standardisoitu tiedonsiirtomenetelmä, jossa tieto kulkee laitteelta toiselle asynkronisessa muodossa. (Kuten myös muissa edellä mainituissa väylissä). Tämäkin väylä siirtää tietoa binaarisesti. Tästä piiristä on tehty monia eri muunnoksia/parannuksia. (WiseGEEk 2010.)

RS232 on ollut myös esimerkkipiiri RS-485 piirille, jota käsitellään myöhemmin. RS232-väylää kutsutaan myös sarjaportiksi ja COM-väyläksi. (WiseGEEk 2010.)

5.2 Sähkökytkennät

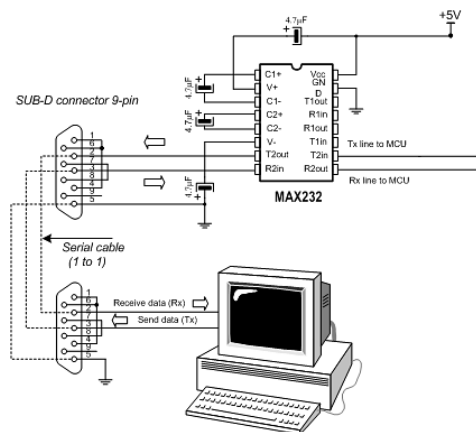
RS232-piirissä on kahdenlaisia jännitetasoja. On TTL:n taso ja RS232-jännitetaso. TTL-taso käyttää jännitettä, joka on välillä 0-5V. RS232 piiri käyttää jännitettä -10 - +10 V. Kuvassa (20) on esitetty miten virrat kulkee LTC1387-piirissä, kun siinä on RS232-väylä. Piiriin syöttönastaan tulee jännite -0,3 ja +0,3 voltin välillä ja ulostulonastassa jännite on -18 - +18 voltin välillä. (Linear technology 84296 2003.)

BLOCK DIAGRAM



**Kuva 20. Sähköinen kytkentä RS232-piirissä
(Linear technology 84296 2003)**

Kuvassa (Kuva 21) RS232 piiri on kytketty tietokoneen ja MAX232-piirin välille (liit-
timinä käytetään RJ45 mallia). Kyseisessä väylässä on käytetty 9-nastaista
RS232-kaapelia.



**Kuva 21. LTC1387 piiri RS232 kytkennässä
(Mikro Electronics RS232 [viitattu 7.4.2010].)**

Kuvassa (21) on pelkistetty miten väylä kytketään kahden portin välille. Tarvittavat
nastat ovat 2, 3 ja 5. Edellä mainitut nastat kytketään siten, että isäntäkoneen 2
nastaa tulee orjakoneen 3 nastaan. Orjakoneen kakkosnastaa tulee isäntäkoneen

kolmosnastaan. Ainoa samaan menevä johto on 5 nastojen välillä kulkeva johto. Se menee isäntäkoneen 5 nastasta orjakoneen 5 nastaan.

Piirroksessa (21) on käytetty siruna MAX232-sirua joka on monikanavainen vastaanotto/lähetyspiiri. Kaapeli kytketään vain nastoihin T2out (7. nasta) ja R2in (8. nasta). T2out-nasta on kakkosväylän ulostulo ja R2in on kakkosväylän sisääntulo (eli kuvassa (21) on piiri laitettu kakkosväylään). Piirin muut nastat ovat kytketty seuraavasti:

- 1. nasta on kytketty kondensaattorin plusnastaan.
- 2. nasta on kytketty plusjännitteeseen.
- 3. nasta on kytketty 1. nastan kondensaattorin miinusnastaan.
- 4. ja 5. nasta on kytketty yhteen kondensaattorin kanssa.
- 6. nasta on kytketty maihin kondensaattorin kanssa.
- nastat 9-14 on käytetty 1. väylän tietoliikenteeseen.
- nasta 15 on kytketty maihin.
- 16. nasta on piirin jännitenasta ja se on kytketty myös kiinni nastaan 2. (Maxim-ic Max232 2006.)

5.3 Datan koodaus

Aiemmin on esitetty kuinka merkit koodataan RS232-piirissä ja kuinka niitä liikutellaan. Seuraavassa aiheessa käsitellään kuinka kokonainen bitistö kulkee sanomakehyksessä. Bittitasolla viestintä menee siten, että isäntäkone lähettää linjaa pitkin 1-tilaa niin kauan kun se on lepotilassa. Kun isäntäkone haluaa lähettää dataa, se laittaa hetkeksi lähetysväylän 0-tilaan ja sen jälkeen tiloja vaihtamalla se lähettää tietoa eteenpäin. (ptm:RS232 [viitattu 6.4.2010].)

Datan lähetys aloitetaan vähiten merkitsevästä bitistä (LSB). Kun seitsemän bittiä on lähetetty, niin tulee lopetusbitti. Se tapahtuu siten, että lähettävä kone vetää lähetysväylän 1-tilaan. Ennen kuin yksikään bitti lähetetään, on kaikkien koneiden asetettava sama tiedonsiirtonopeus. Jokainen bitti luetaan tietyn ajan kuluessa

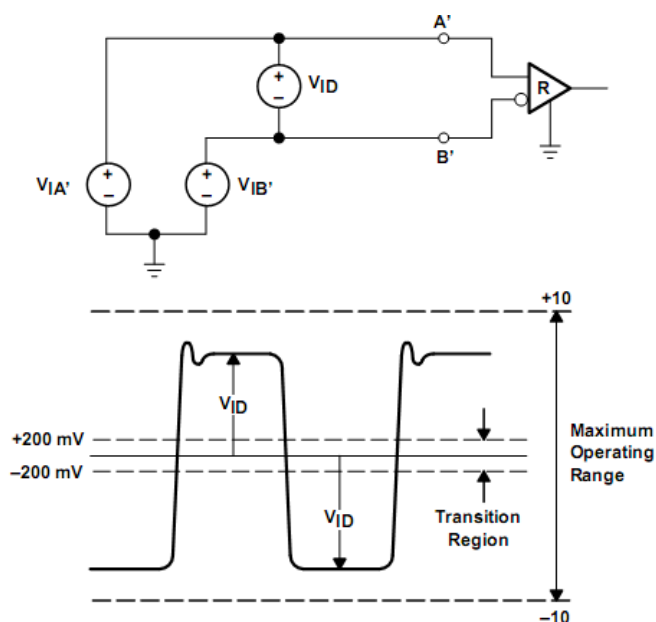
edellisestä, joten on tärkeää, että kaikilla laitteilla on sama nopeus. (Maxim-ic Max232 2006)

Seuraavassa käsitellään ohjelman koodausta. Koodissa on käytetty baudinopeutena 9600 baudia eli se lähettää bittejä nopeudella $1/9600$ joka on noin 1 millisekunti. Toinen tavallinen nopeus on 115200 baudia, joka vastaa 8,7 mikrosekunnin siirtonopeutta per bitti. (ptm:RS232 [viitattu 6.4.2010].)

Jossain piireissä piiri vaatii, että lopetusbitti on kahden bitin pituinen, eli 9600 baudilla lopetusbitin kestoksi tulee noin 0,2 millisekuntia. Piirin pitää olla lopetus tilassa $2 \cdot 0,2$ millisekuntia, jolloin piiri menee lopetustilaan (idle). (ptm:RS232 [viitattu 6.4.2010].)

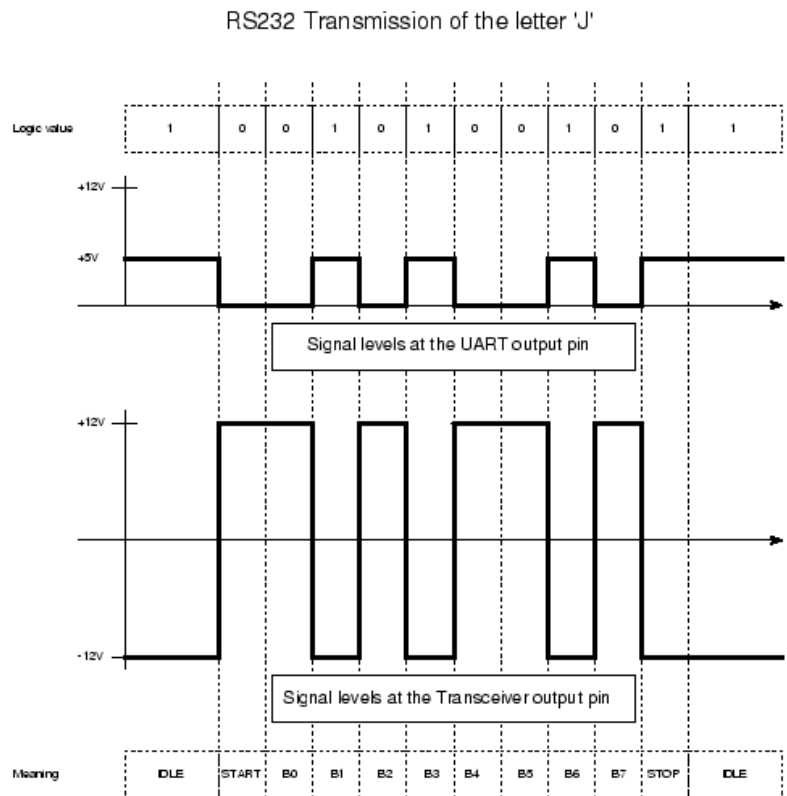
RS-232 piiri on differentiaalipiiri. Eli piirissä jännitevaihtelu ei tapahdu aina nollan ja tietyn volttimäärän rajoissa vaan jokainen piiri asettaa jännitteelle alarajan. Kun jännite tippuu tietyn arvon alapuolelle, niin piiri tulkitsee tilan nollassa. Piirissä on myös 1-tilaan oma jännitearvo. (Texas Instruments 2002.)

Jännitteiden väliin jää eräänlainen harmaa alue, jota piiri tulkitsee omalla tavallaan. Kuvassa (22) on esitetty differentiaalikytkentä RS232-tasossa.



Kuva 22. RS232-differentiaali
(Texas Instruments 2002)

Differentiaali määrää paljonko jännitteessä pitää olla väliä nollan ja ykköstilän välillä. Kuvassa 23) on esitetty niin TTL-tason signaali kuin myös RS232-tason signaali RS232-väylässä. Kuvassa (23) on otettu esimerkki, miten kirjain J koodataan TTL- ja RS232-tasolla. (Texas Instruments 2002.)



Kuva 23. RS232 signaalit

(Best-Microcontroller-projects [viitattu 7.4.2010])

RS232 väylässä ei oikeastaan ole varsinaista sanomakehystä sen kummemmin. Sanomat kulkevat seuraavanlaisesti:

Ensiksi tulee itse aloitusbitti. Sen jälkeen tulee databitit, jonka jälkeen tulee pariteettibitti (tarkistusbitti esim. CRC16-menetelmällä tehty). Tuon kaiken jälkeen tulee lopetusbitti. (Texas Instruments 2002.)

Itse viesti tulee siitä, kun bittejä laittaa monta peräkkäin. Yksi sykli vastaa yhtä merkkiä. Eli kirjain A on kahdeksan bittinen sarja. Binaarisena (sähköisenä) se on 01000001. (Texas Instruments 2002.)

5.4 Ohjelmaesimerkki

Alla on esimerkkikoodi RS232 piiristä joka on otettu osoitteesta:

(<http://www.edaboard.com/ftopic326438.html>). Alla oleva koodi on tarkoitettu Intelin 8051 piirille.

Alla olevassa koodissa on käytetty reg51.h päätiedostoa jossa on asetettu rekistereihin joitain arvoja. Tarkempia tietoja saa katsomalla tiedostoa osoitteessa:

<http://www.keil.com/dd/docs/c51/reg51.h>

Esimerkki 4. RS232

```
/* RS232 SIMPLE TRIAL */
```

```
#Include <stdio.h>
#include <reg51.h>
```

```
/*Function to initialize RS232 serial port*/
```

```
void serial_init()
{
    SCON=0X50;    //Setup for 8-bit data
    TMOD=0X20;    //Setup Timer 1 for auto-reload
    TH1=0XFD;     //Setup for 9600 baud
    TR1=1;        //Turn on Timer 1
    T1=1;         //Indicate Ready to Transmit
}
```

```
/*This func display a null-terminated string on the RS232 port*/
```

```
void send_serial(unsigned char*s)           tulostus funktio alustetaan kirjain s
{
    while(*s!=0x0)                          pyöruu niinkauan kunnes s=0
    {
        SBUF=*s;
        while(!T1)
        { }
        T1=0;
        s++;                                 kasvatetaan s:n arvoa
    }
}
```

```
/*Start of main program*/
```

Pääohjelman aloitus

```
main()
{
    unsigned char crlf[]={0x0D,0x0A,0x0};    Alustetaan crlf taulukko
    serial_init();
    for (;;)
    {
        send_serial("Another test");
        send_serial(crlf);                  kutsutaan crlf taulukkoa
    }
}
```

SCON-käskyllä piiri on asetettu 8-bittiseksi. TMOD-komennolla ajastin asetetaan haluttuun toimintoon (tässä tapauksessa ajastin latautuu aina uudestaan mentyään nolnaan). Baudinopeus on asetettu komennolla TH1 = 0XFD, jolloin nopeudeksi tulee 9600 baudia. (mikroElektronika 2010.)

Komennolla TR saadaan laitettua ajastin päälle. T1 komennolla saa laitettua piirin osoittamaan, että se on valmis lähettämään dataa. Koodin muut komennot ja lisää materiaalia löytyy esimerkiksi sivulta: <http://www.mikroe.com/en/books/8051book/>. (mikroElektronika 2010.)

Baudinopeus merkitsee sitä, montako bittiä lähetetään sekunnissa. Esimerkiksi 1000 baudia on 1000 bittiä sekunnissa eli yhden bitin siirtoon käytettävä aika on 1 millisekunti. (mikroElektronika 2010.)

5.5 Käyttötapoja

RS232-väylää käytetään tiedon siirrossa tietokoneen ja oheislaitteiden välillä, esimerkiksi tietokoneen ja hiiren välissä. Muita käyttötapoja on esimerkiksi juuri sulautetuissa järjestelmissä, jossa piirin pitää kyetä keskustelemaan esimerkiksi robotin tai muun älyllisen laitteen kanssa.

6 RS485

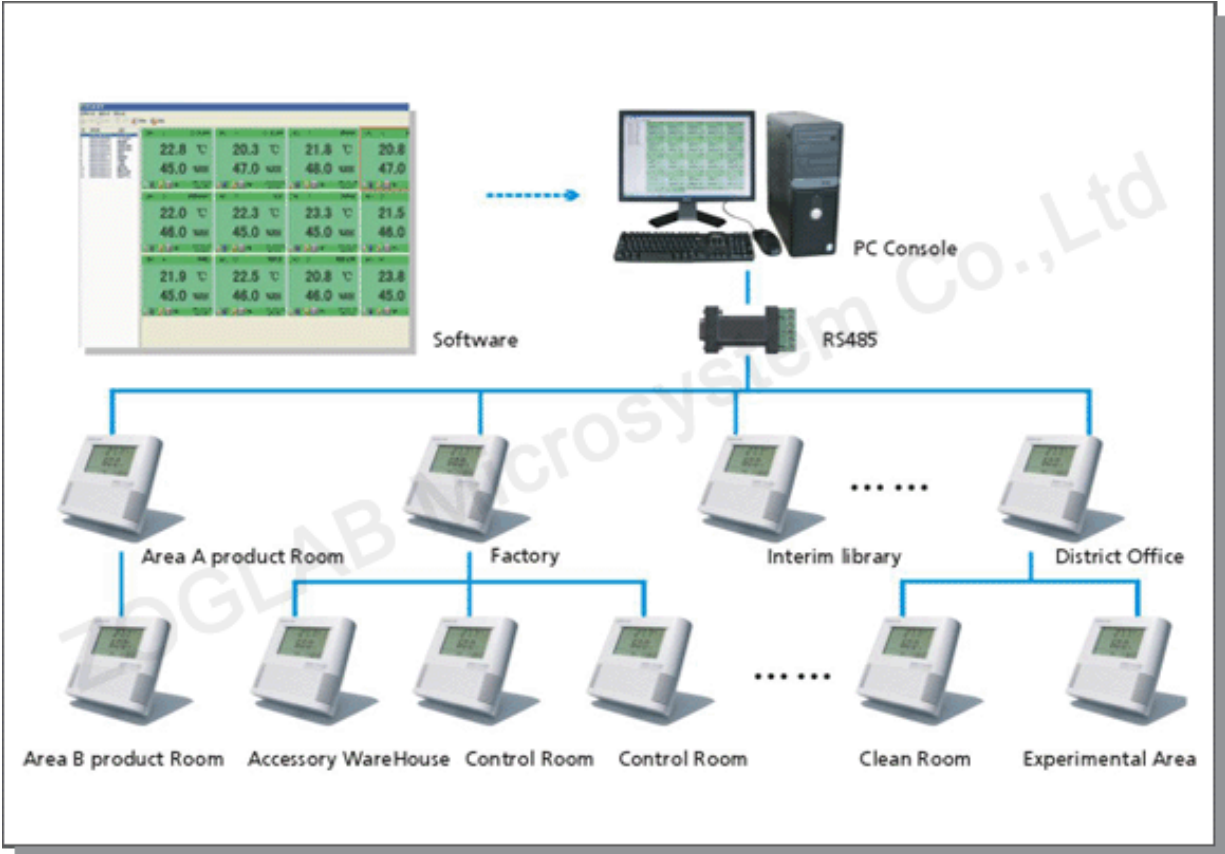
6.1 Mikä on RS485 (EIA-485) ?

RS485 on kehittyneempi versio RS232-piiristä. RS485-piirissä voi olla monta oheislaitetta. RS485 käyttää myös HALF DUPLEX- tiedonsiirtotapaa, kun RS232 käyttää FULL DUPLEX- ja HALF DUPLEX- tiedoston siirtoa. Tiedoston siirtonopeus on myös aivan eri luokkaa kuin RS232 väylässä. (Lammert Bies RS-485 2010)

Laitteiden välinen etäisyys RS485-piirissä saa olla 1200 m. Suurin ero RS232- ja RS485-väylässä on se, että RS485-väylässä on kaapelin johdot kiedottu toistensa ympärille, kun taas RS232-väylässä johdot ovat suoraan. Tämän kierron tarkoituksena on estää magneettikentissä aiheutuvat häiriöt. (Lammert Bies RS-485 2010.)

Myös tässä väylässä on differentiaaliominaisuus. Tiloista on syytä mainita, että toisessa johtimessa on tieto käännetty ja toisessa ei. (Lammert Bies RS-485 2010.)

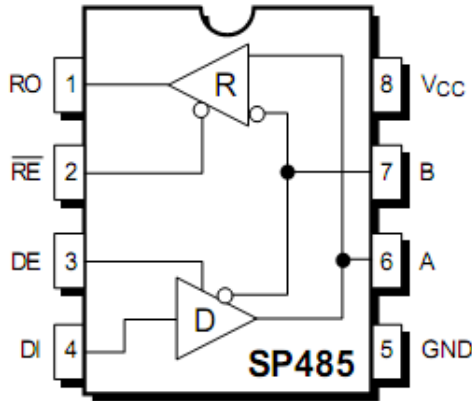
Kuvassa (24) on esitetty RS485 väylällä tehty monen laitteen kytkeminen yhteen koneeseen.



Kuva 24 RS485 verkko
(Zoglab MicroSystems co., Ltd [viitattu 7.4.2010])

6.2 Sähkökytkennät

Kuvassa (24) on esitetty RS485 väylälle tehty siru, jolla voidaan lähettää/vastaanottaa dataa.



Kuva 25. RS485 Block diagram

(Sipex SP481R/SP485R 2004.)

.

Kuvassa (25) piirin nastat ovat:

RO = Receiver output (vastaanottimen ulostulo)

RE = Receiver enable (vastaanotin päällä)

DE = Driver enable (ohjaus päällä)

DI = Driver Input (ohjauksen syöttö)

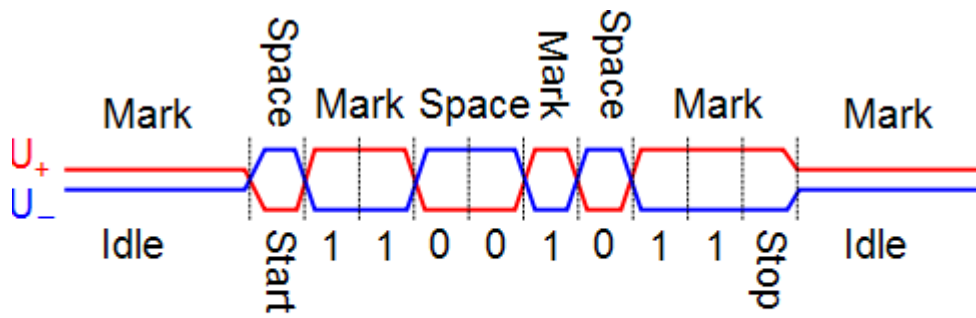
GND = maa

A = Driver Input/Output non inverting (piirin vastaanotto/lähetys ei-kääntävä)

B= Driver Input/Output inverting (piirin vastaanotto/lähetys kääntävä)

VCC= jännite. (Sipex SP481R/SP485R 2004)

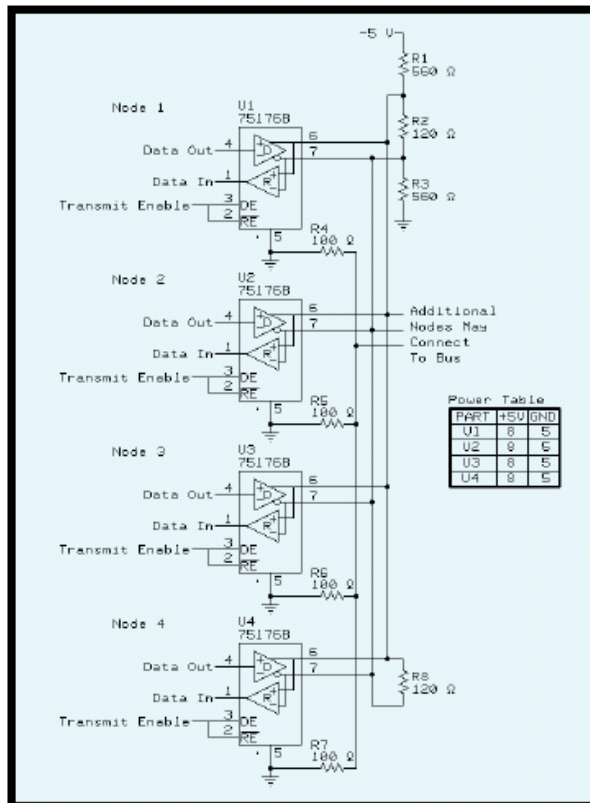
Kuvassa (26) on esitetty viestin siirto sähköisenä.



Kuva 26. RS485-Signaali
(Roy Vegard Ovesen 2007.)

Kuvassa (Kuva 26) on esitetty, miten yksi merkki saadaan lähetettyä RS485-piirissä. Tämä piiri on differentiaalipiiri. Kuvassa (26) se näkyy siten, että toinen jännite on aina käänteinen verrattuna toiseen jännitteeseen. U_+ ja U_- ovat siis jännitepari. (Chipkin Automation Systems 2007.)

Kuvassa (27) esitetään, kuinka monta laitetta kytketään sähköisesti toisiinsa.



Kuva 27. RS485 usean laitteen kytkeminen
(Electro Tech [viitattu 7.4.2010].)

6.3 Ohjelmaesimerkki

Tähän osioon on otettu vain tärkeimmät koodiesimerkit, jolla pääsee alkuun laitteen koodauksessa. Esimerkkiin on käytetty kortia PCI-COM 485/4.

Ensimmäiseksi tulee kortille kertoa baudinopeuden jakaja. Se saadaan seuraavallisesta laskukaavasta:

$$\text{HaluttuBaudinopeus} = (\text{UARTIN kellotaajuus}) / (32 * \text{JAKAJA}).$$

Kuvassa (28) on esitetty tietyille nopeuksille omat jakajat.

Baud Rate	Divisor x1	Divisor x4	Max Diff. Cable Length*
460800	-	1	550 ft
230400	-	2	1400 ft
153600	-	3	2500 ft
115200	1	4	3000 ft
57600	2	8	4000 ft
38400	3	12	4000 ft
28800	4	16	4000 ft
19200	6	24	4000 ft
14400	8	32	4000 ft
9600	12	48 – Most Common	4000 ft
4800	24	96	4000 ft
2400	48	192	4000 ft
1200	96	384	4000 ft

**Kuva 28. Baudinopeudet PCI-COM-485-4 kortissa
(Acces I/O Products, Inc. [viitattu 6.4.2010].)**

Seuraavat toimenpiteet on otettu valmistajan datakirjasta, sieltä on mahdollista nähdä komennot. Ensiksi tulee alustaa LINE CONTROL REGISTER BASE ADDRESS +3. Kyseinen alustus määrittää sanan pituuden, pysäytysbittien (stopbit) määrän, pariteetin ja DLAB:in (divisor latch access bit). (Acces I/O Products, Inc. [viitattu 6.4.2010].)

Loppubittien määrän määrää 2-bitti. Jos bitin arvo on 0, niin tällöin on 1 pysäytysbitti. Jos bitin arvo taas on 1, niin silloin pysäytysbittejä on kaksi. (Acces I/O Products, Inc. [viitattu 6.4.2010].)

Bitit 3 - 6 määrittävät pariteetin. Seitsemäs bitti on aikaisemmin mainittu DLABIN access bit. (Acces I/O Products, Inc. [viitattu 6.4.2010].)

Viimeisenä alustuksena kyseiselle kortille pitää tehdä bufferin (puskurin) tyhjennys. edellä kerrottu oli vasta piirin alustusta ja seuraavassa käsitellään, miten piirillä voidaan lähettää/vastaanottaa dataa. Helpoiten asian pystyy ymmärtämään esimerkkikoodin avulla. Seuraavassa koodissa on esimerkki datan vastaanotosta. (Acces I/O Products, Inc. [viitattu 6.4.2010].)

Esimerkki 5. RA485:n esimerkki koodeja

```
do
{
while (!(inportb(BASEADDR +5) & 1)); ← oheinen koodi odottaa kunnes data on valmis
data[i++]= inportb(BASEADDR); ← tuossa taas datataulukkoon kirjoitetaan merkki
}
while (data[i]!=13); ←lukee merkkejä kunnes tulee rivinvaihto
```

Eli oheinen koodi lukee merkki kerrallaan dataa taulukkoon niin kauan kunnes tulee 1 rivinvaihto (enter).

Seuraavassa on taas esitetty lähetysfunktio RS485 piirissä.

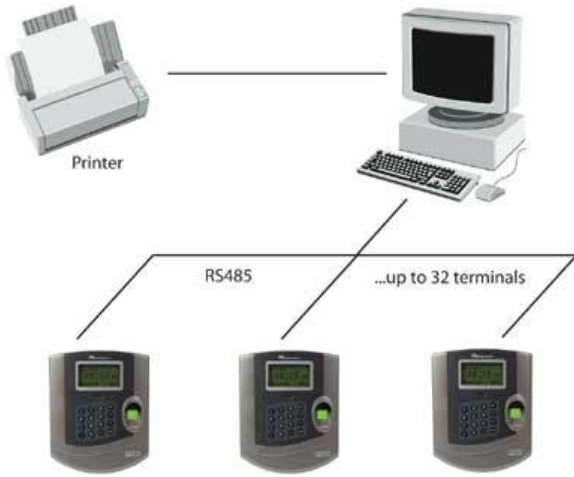
```
outportb(BASEADDR +4, inportb(BASEADDR +4)|0x02);
/*Set RTS bit without altering states of other bits*/
while(data[i]); /*While there is data to send*/
{
while(!(inportb(BASEADDR +5)&0x20)); /*Wait until transmitter is empty*/
outportb(BASEADDR,data[i]);
i++;
}
outportb(BASEADDR +4, inportb(BASEADDR +4)&0xFD);
/*Reset RTS bit without altering states of other bits*/
```

Oheisessa koodissa on pelkistetty yhden merkin lähetys. Tässä esimerkissä on käytetty atmega16-piiriä ja sen rekistereitä. Tässäkin esimerkissä kaikki on rakennettu silmukkaan, joka kestää niin kauan kunnes lähetin on tyhjä. (Atmel Atmega16 2003.)

6.3 Käyttötapoja

RS485 väylää käytetään yleisesti teollisuudessa ja muissa automaatiolaitteissa, esimerkiksi tehtaassa, jossa yhdellä koneella ohjataan muita oheislaitteita. Kyseistä väylää voidaan käyttää, kun halutaan kytkeä monta eri anturia yhteen. Kuvassa (24) on esitetty edellä mainittu kytkentä. (Lammert Bies RS-485 2010.)

Kuvassa (29) on esitetty toimistotiloissa käytettävää rs485-sovellusta.



**Kuva 29. RS485:n käyttötapa
(Global Time System [viitattu 7.4.2010])**

7 MODBUS

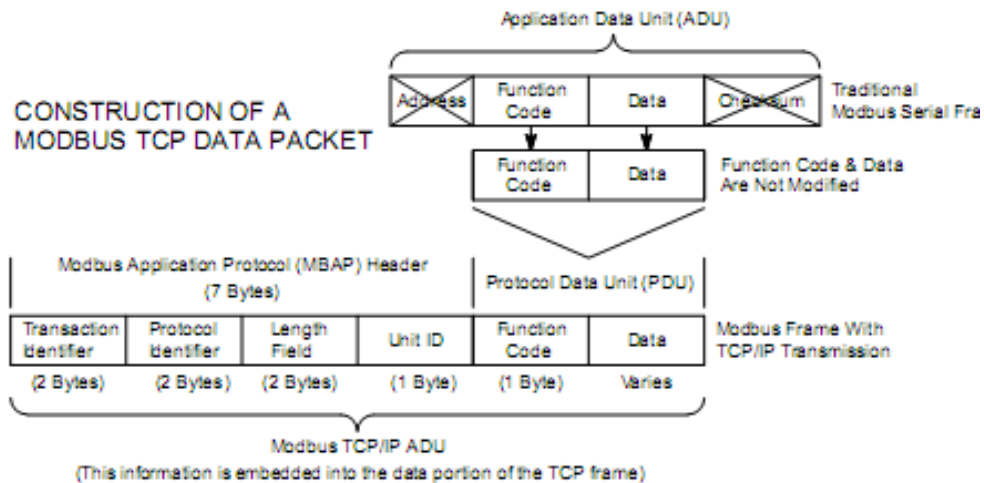
7.1 Yleistä

Modbus on tietoliikenteessä käytettävä väylä. Modbusilla pystytään laittamaan monta eri konetta kiinni toisiinsa, ja näin voidaan muodostaa verkko eri oheislaitteiden välille. Modbus väylällä pystytään myös lähettämään RS232-viestejä laitteilta toisille. (Lammert Bies Modbus 2010.)

Modbuseista on tehty monta erilaista versiota, esimerkiksi modbus RTU. Modbus RTU käyttää muihin versioihin nähden vähemmän muistia. Koska modbus RTU käyttää binaarista tiedonsiirtoa, niin se pystyy käyttämään 16-bittistä CRC-tarkistusta. (Lammert Bies Modbus 2010.)

Toinen Modbusin versio on Modbus ASCII. Tässä väylässä tiedonsiirto tapahtuu ASCII-merkkeinä. Kyseisessä versiossa virheen tarkastus on tyyppiä LRC (Longitudinal Redundancy Check). LRC-tarkistus toimii siten, että se laskee 8 bittiä yhteen ja lähettää sen summan eteenpäin. Orjakone laskee myös saamiensa bittien arvon yhteen ja vertaa sitä tarkistusbittiin. Yleensä vertailussa käytetään XOR vertailua. Orjakone vertaa saatua arvoa omaan arvoonsa bitti kerrallaan ja jos tulos on nolla, niin mikään ei ole mennyt väärin. (Lammert Bies Modbus 2010.)

Kolmas Modbusin versio on TCP/IP-versio. Sitä käytetään tietoliikenneväylissä, joissa joutuu viestimään verkon yli. TCP/IP-versiossa ei käytetä osoitetta eikä virheentarkastusta ollenkaan. Siinä käytetään vain pelkästään funktiokoodia ja itse tietoliikenne dataa. Kuvassa (30) on esitetty edellä mainittu asia. (Lammert Bies Modbus 2010.)



Kuva 30. Modbus TCP/IP
(Acromag intro_modbusTCP_765a 2005.)

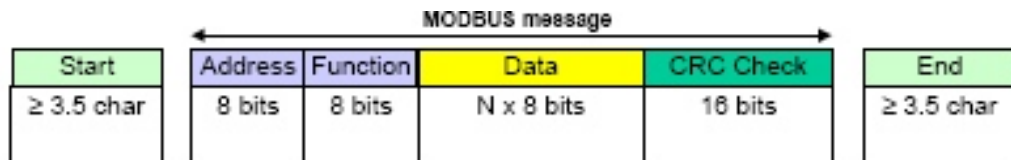
Kuvasta (30) nähdään, miten oheisessa väylässä tieto liikkuu seuraavanlaisesti: Ensimmäisen kentän tarkoitus on erotella viestit toisistaan. Esimerkiksi jos kaksi konetta lähettää viestin samaan aikaan tiettyyn osoitteeseen, niin transaction identifier ottaa ensin toisen koneen viestit vastaan kokonaan vasta sen jälkeen vastaanottaa toisen koneen viestit kokonaan. (Acromag intro_modbusTCP_765a 2005.)

Modbus-väylä käyttää AINA protocol identifier- arvoa nolla. Viestin pituus valitaan length field -kentässä. Unit Id -kentässä määritetään TCP/IP-osoite, jolle viesti lähetetään. Seuraava bitti määrittelee mitä isäntäkone haluaa tehdä. esimerkiksi lukea/kirjoittaa dataa. Viimeisenä taulukossa tulee itse data. (Acromag intro_modbusTCP_765a 2005.)

Viimeisenä versiona Modbusissa on Modbus plus –versio, joka on laajennettu versio normaalista Modbusista. Modbus plussia käytetään lähiverkoissa, koska sillä saadaan nopeita yhteyksiä koneelta toiselle, lisäksi Modbus plussia on halpa ja helppokäyttöinen. Se eroaa muista siinä, että se tarvitsee omanlaisen prosessorin käsittelemään HDLC-tyyppistä token-kiertoa. Plus versiota käytetään pääasiassa modiconissa. (Acromag intro_modbusTCP_765a 2005.)

7.2 Sanomat ja rakenne

Jokaisella Modbus versiolla on omanlaisensa protokolla. Modbus RTU:lla viestin protokolla on seuraavanlainen:



Kuva 31. ModbusRTU:n sanomakehys
(Real Time Automation RTU [viitattu 6.4.2010].)

Kuvassa (31) on esitetty Modbus RTU:n sanomakehys. Ensimmäiseksi tulee 3,5 merkkiä tai pitempi tauko viestiin ja sen jälkeen tulee 8-bittinen osoite. Seuraavaksi tulee kahdeksan bitin pituinen binaarikoodi, jolla määrätään mitä kyseisessä väylässä tapahtuu. Sen jälkeen tulee itse viesti. Viestin yksi merkki on 8-bitin pituinen ja viestissä merkkejä voi olla 255. (Real Time Automation RTU [viitattu 6.4.2010].)

Seuraavaksi tulee CRC-tarkistus, joka on 16 bittiä pitkä. Viimeisenä tulee 3,5 merkin pituinen, viive joka ilmoittaa viestin loppumisen. Real Time Automation RTU [viitattu 6.4.2010].)

Modbussin ASCII:ssa viestit kulkevat seuraavanlaisesti: Aloitus on yhden merkin kokoinen ja yleensä se on ":" . Osoite ASCII versiossa on kahden (2) merkin pituinen. Myöskin funktiokoodi on kahden merkin pituinen. Viesti on ASCII merkkeinä. Virheen tarkastuksessa käytetään kahta merkkiä, kuten edellä kerrottu. (Real Time Automation ASCII [viitattu 6.4.2010].)

ASCII-muunnelmassa viestin lopetus tapahtuu CRLF-komennolla. Real Time Automation ASCII [viitattu 6.4.2010].)

Kuvassa (32) on esitetty miten viesti kulkee ASCII tyypisenä. Aloitus merkki on ”:” ja osoite- ja funktiokomento on kahden merkin pituinen ja viestin pituus on esitetty funktiokoodissa. (Real Time Automation ASCII. [viitattu 6.4.2010].)

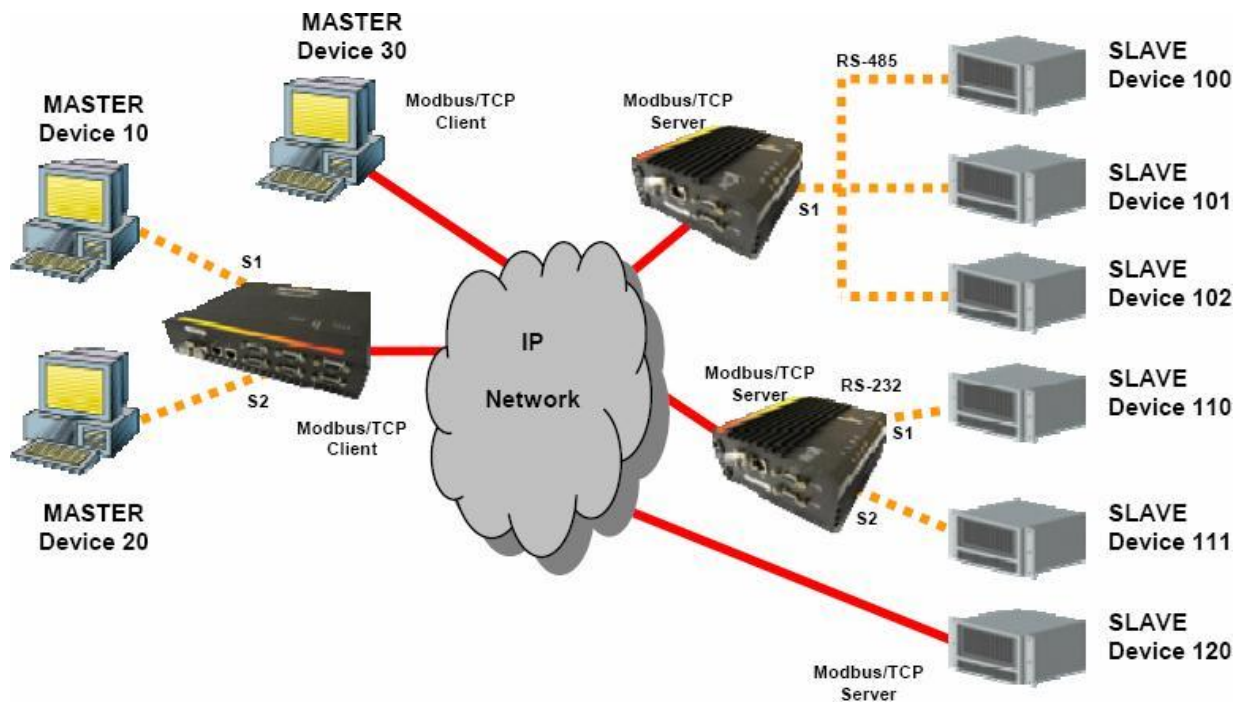
Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

Kuva 32 Modbus ascii

(Korea Imagineering [viitattu 7.4.2010].)

7.3 Modbusin käyttö

Kaikkia Modbus versioita käytetään sulautetuissa järjestelmissä. Kuvassa (33) on esitetty TCP/IP-versiolla tehty väylä, joka hyödyntää kyseistä väylää.



Kuva 33 Modbusin käyttö

(Ferret Australia's Manufacturing and Industrial Directory [7.4.2010].)

8 PROFIBUS

8.1 Mikä on profibus ?

Profibus on yksi väylätekniikan sovellusmuodoista. Profibus-väylällä voidaan ohjata (hallinnoida) montaa eri laitetta. Profibus ja profinet ovat kaksi aivan erilaista väylää, vaikka nimestä voisi päätellä toisin. Profibussista on ainakin kolme erilaista versiota. Niistä yleisesti on tällä hetkellä käytössä vain kaksi (Profibus DP ja Profibus PA). (Real Time Automation Profibus [viitattu 7.4.2010].)

Profibusin toinen versio on nimeltään Profibus DP. DP tulee sanoista Decentralized Peripherals, joka vapaasti suomennettuna tarkoittaa hajautettuja oheislaitteita. Tämä väylä on tarkoitettu esimerkiksi tehtaille, joissa yksi kone ohjaa montaa konetta, jotka on sijoitettu ympäri tehdasta. (Real Time Automation Profibus [viitattu 7.4.2010].)

Toinen Profibussin versio on nimeltään Profibus Pa. PA tulee taas sanoista process automation. Kyseistä versiota käytetään muun muassa, kun halutaan yhdellä koneella tarkkailla monen koneen anturiarvoja. Tätä versiota käytetään prosessi-automaatiossa. PA-väylä on suunniteltu räjähdysalteille ja muuten vaarallisille alueille. (Real Time Automation Profibus [viitattu 7.4.2010].)

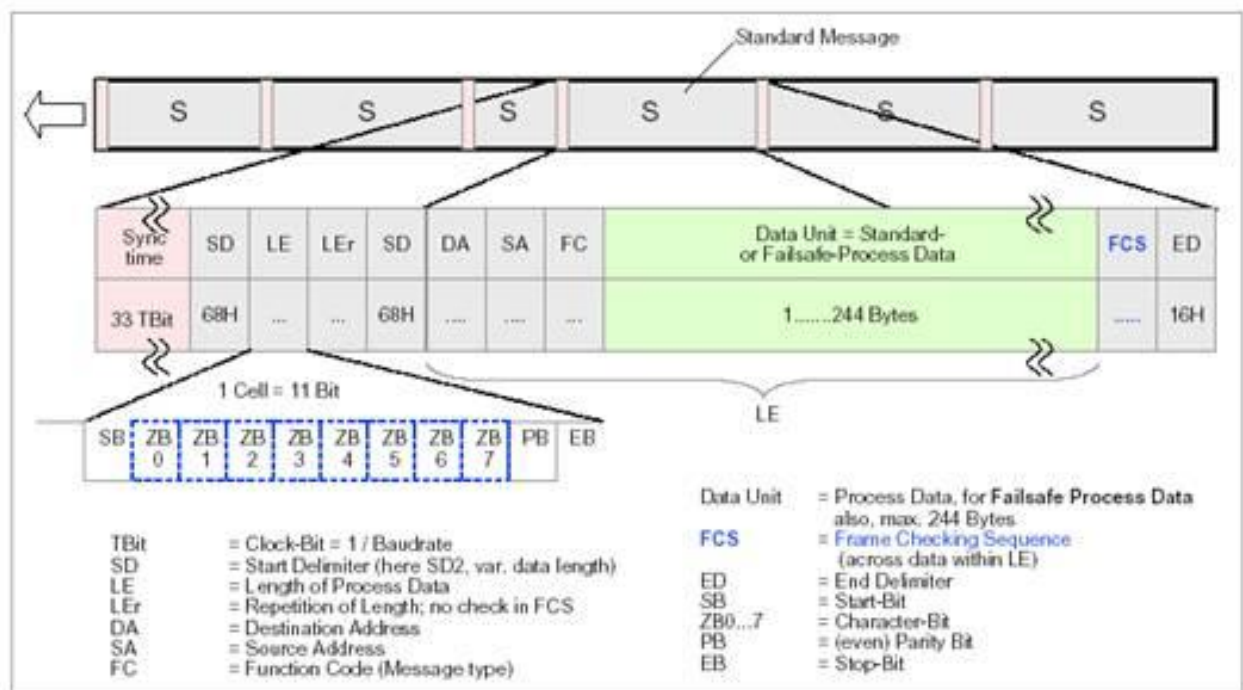
Kolmas versio profibussista on profibus fms. Kyseinen versio oli ensimmäisiä Profibus-väyliä, josta aloitettiin rakentamaan edellä mainittuja väyliä. FMS tulee sanoista fieldbus message specification. Tämä laite oli suunniteltu viestimään pc:n ja ohjelmoitavan ohjauslaitteen välillä. FMS-tekniikassa oli puutteita, joten se korvattiin edellä mainituilla sovelluksilla. (Real Time Automation Profibus [viitattu 7.4.2010].))

8.2 Profibusin tekniikka

Seuraavassa esitetään Profibusin tekniikkaa ja sitä tekniikkaa esitellään DP:n avulla. Profibusissa on 3 erilaista kerrosta. Ensimmäinen niistä on fyysinen kerros. Tässä kerroksessa kuvataan millä tavalla viesti kulkee väylässä. Profibussissa on

kolme eri vaihtoehtoa: Ensimmäinen vaihtoehto on RS-485-kaapeli. Toinen vaihtoehto on valokuitu ja kolmas mahdollisuus on käyttää MBP (Manchester Bus Powered)-väylää. Tätä kerrosta voisi myös kutsua bitinsiirtokerrokseksi, koska tässä kerroksessa itse viesti lähetetään yksi bitti kerrallaan. (Real Time Automation Profibus [viitattu 6.4.2010].)

Toinen kerros profibussin rakenteessa on datalinkki kerros eli itse kaapeli kahden laitteen välillä. Kyseinen kerros on edellisissä aiheissa kuvattu sanomakehyksenä. Kuvassa (33) on näytetty koko sanomakehys ja yhden solun koodaus. (Real Time Automation Profibus [viitattu 6.4.2010].)



Kuva 34. Profibussin sanomakehys
(SMAR Industrial Automation [viitattu 7.4.2010].)

Viestin kulku on seuraavanlainen Profibus-väylässä: Ennen kuin yhtään viestiä voidaan lähettää, pitää kaikkien piirien laitteet synkronoida keskenään samalle baudinopeudelle. Seuraavaksi viesti pitää muuntaa biteiksi. Jokainen merkki vastaa 8-bittiä. Tässä on yksi esimerkki miten merkki G muunnetaan bittikoodiksi. (Acromag Profibus 2009.)

Ensimmäiseksi merkki "G" muutetaan hexaluvuksi joka, on tässä tapauksessa 47 (oheinen arvo on otettu osoitteesta <http://asciitable.com/>). Binaarisesti se saadaan muuttamalla 4 ja 7 neljän numeron bittisarjaksi: siis 4 = 0100 ja 7 = 0111. Lopuksi nämä yhdistetään yhdeksi merkkijonoksi. G:n arvoksi binaarisena saadaan 01000111. (ASCII [viitattu 6.4.2010].)

Jokainen merkki koodataan samalla tavalla. Kun väylän nopeus on saatu aseteltua ja kaikki viestit on saatu muunnettua biteiksi, niin itse viestin lähetys voi alkaa. Aluksi kone lähettää SD:ksi nimetyn bitin, joka erottaa merkkijonot toisistaan. Kuten kuvassa (34) on esitetty. (Acromag Profibus 2009)

Seuraavaksi lähetetään viestin pituus. Viestin pituus määräytyy seuraavanlaisesti: viestin pituus + kohteen osoite + lähteen osoite (lähettäjän osoite) + toimintakoodi (mitä halutaan tehdä) + vastaanottajan COM-portin osoite + lähettäjän COM-portin osoite. Viestin pituus voi olla 1 - 244 bittiä pitkä. Muiden toimintojen arvo on yksi bitti. Viestin pituus on kuvassa (34) merkitty kirjaimin LD. (Acromag Profibus 2009)

Seuraavaksi lähetetään uudelleen viestin pituus. Tämän jälkeen lähetetään erotus merkki SD (Star Delimiter). Sen jälkeen lähetetään vastaanottajan osoite, joka on pituudeltaan yksi bitti. Lähettäjän osoite lähetetään seuraavaksi ja sekin on yhden bitin pituinen. Seuraavaksi lähetetään tieto mitä halutaan tehdä. Kuvassa (34) se on merkitty kirjaimin FC (funktion code). (Acromag Profibus 2009.)

Kuvassa (35) on esitetty mitä erilaisia toimintoja voidaan tehdä Acromag 900pb -sarjan piirillä.

Profibus DP Function Codes for Request Telegrams

FC Code	Function (The MSB in FC = 1)
4	SDN low (Send Data with No acknowledge)
6	SDN high (Send Data with No acknowledge)
7	Reserved/Request Diagnostic Data
9	Request FDL Status With Reply
12	SRD low (Send and Request Data with acknowledge)
13	SRD high (Send and Request Data with acknowledge)
14	Request ID With Reply
15	Request LSAP Status With Reply

Profibus DP Function Codes for Acknowledgement Telegrams

FC Code	Function (The MSB in FC = 0)
0	ACK Positive
1	ACK Negative (FDL/FMA1/2 user error UE, interface error)
2	ACK Negative (No resource/memory space for Send Data (RR).
3	ACK Negative (No service activated (RS), SAP not activated).
8	Response FDL/FMA 1/2 Data low and Send Data OK)
9	ACK Negative (No response FDL/FMA1/2 Data & Send Data OK).
10	Response FDL Data High and Send Data OK.
12	Response FDL Data Low, No resource for Send Data.
13	Response FDL Data High Resource For Send Data.

Kuva 35. Profibussin toimintakoodit

(Acromag Profibus 2009.)

Tämän jälkeen tulee itse data, jota voi olla kahta eri muotoa. Data voi olla standardia dataa tai se voi olla turvattua dataa (fail safe) prosessidataa. Fail safe data on tässä tapauksessa virheetöntä dataa. Eli lähettäjän ja vastaanottajan data on bitilleen samanlaista. Kun kaikki datat on lähetetty, tulee Frame Checking Sequence, eli viestin tarkistus. (Acromag Profibus 2009.)

Frame checking sequence lasketaan seuraavanlaisesti: $FCS = (DA+SA+FC+DU) \text{ mod } 256$, eli lasketaan kohteen osoite, lähettäjän osoite, toimintakoodi ja viestit yhteen binaarisella ja muutetaan se kymmenjärjestelmään. Saatu luku jaetaan 256 ja siitä saatu jakojäännös on FCS:n arvo. (Acromag Profibus 2009.)

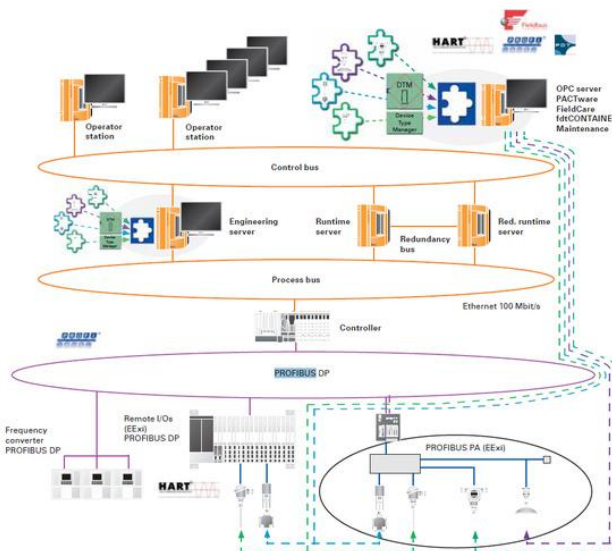
Sanomakehysten päättää ED-merkki joka on End Delimiter. Merkki on yhden bitin pituinen. On olemassa myös muita sanomakehysrakenteita riippuen siitä minkä tyyppistä sanomaa kulkee. (Acromag Profibus 2009.)

Kolmas kerros profibussin rakenteessa on application layer eli sovelluskerros. Tähän tasoon kuuluu profibussilla DP-V0, DP-01 ja DP-V2. DP-V0 väylä tukee syklistä tietojen vaihtoa, kun taas DP-V1 tukee ei-syklistä tietojen vaihtoa lisäksi ne tukee hälytyksiä (voi laittaa piirin hälyttämään kun jokin raja-arvo saavutetaan). DP-V2 tukee taas samaan aikaan lähetettävää tiedonsiirtoa orjien välillä. (Acromag Profibus 2009.)

Tähän kerrokseen myös kuuluu edellä mainittu FMS-väylä, eli Sillä ei ole ollenkaan omaa OSI-kerrosta, vaan se oli pikemminkin yksi ohjelma. (Acromag Profibus 2009.)

8.3 Profibussin käyttötapa

Profibus on maailman yleisin kenttäväylä ja sitä käytetään muun muassa prosessiautomaatiossa, joissa pitää ohjata etäältä useimpia koneita yhtä aikaa. Profibus on myös edullinen, koska siitä voidaan rakentaa yhden kaapelin väylä. Kuvassa (36) on esitetty yksi profibussin monista käyttötavoista. (Acromag Profibus 2009.)



Kuva 36. Profibussin käyttötapa
(B&R Perfection in Automation [viitattu 7.4.2010].)

9 MUUT VÄYLÄT

Tietotekniikassa ja tietoliikenteessä on myös muita väyliä ja niiden eri sovelluksia. Kuten jo yllä mainitussa SPI-väylässä oli kuva (SPI-väylät sarjassa) pelkistetystä väylästä, jossa signaali johtimet olivat kytketty laitteelta toiselle, kun normaalisti isäntä koneelta tulee joka väylälle oma johdin.

Näitä väyliä on esimerkiksi CAN-väylä, jota käytetään esimerkiksi autoissa. Toinen käytettävä väylä on microwire-väylä. Kyseistä väylää käytetään muun muassa signaalin käsittelyssä, muistipiirissä ja taajuuslaitteissa, joissa taajuutta pitää kyetä hallitsemaan.

YHTEENVETO

Yhteen vetona voisi sanoa, että kaikkia edellä mainitsemia väyliä tullaan näkemään tulevaisuudessa, Tehtaissa ja muissa isoissa toimistorakennuksissa tullaan käyttämään luultavimminkin RS485-, Modbus- tai Profibus-väylää. Muut laitteet ovat pikemminkin suunniteltu yksinkertaisimmille sovelluksille. Esimerkiksi 1-Wire-väylää tullaan käyttämään sähköisissä laskureissa.

2-wire väylää käytetään laitteissa jossa halutaan kommunikoida kahteen suuntaan helposti. Esimerkiksi antureissa ja niitä valvovissa koneissa. SPI-väylää taas tullaan käyttämään laitteissa joissa vaaditaan lukua ja kirjoitus tilaa samaan aikaan usealta laitteelta. Esimerkiksi pienissä pajoissa joissa on esimerkiksi robotteja ja kameroita.

RS485-väylä on syrjäyttänyt RS232-väylän, joten sitä ei tulla käyttämään muualla, kuin vanhoissa laitteissa joihin RS485 ei sovellu.

LÄHTEET

~Skarna. 1-wire tekniikka [www-dokumentti]. ~Skarna [viitattu 6.4.2010]

Saatavissa:

<http://www.nic.fi/~skarna/etusivu.html>

2wire. 2010. Home [www-dokumentti]. 2Wire [viitattu 6.4.2010]

saatavissa:

<http://2wire.com/> ja <http://en.wikipedia.org/wiki/2Wire>

2wire. 2010. Overview. [www-dokumentti] 2Wire [viitattu 6.4.2010]

saatavissa:

<http://www.2wire.com/index.php?p=479>

Access 2010. Model pci-com485/4 user manual [pdf-dokumentti]. Access I/O Products, Inc [viitattu 6.4.2010]

saatavissa:

<http://accessio.com/MANUALS/PCI-COM-485-4.PDF>

Agromag. 2005. ntro_modbusTCP_765a 2005. technical Reference-Modbus TCP/IP [pdf-dokumentti] Agromag The leader in Industrial I/O [viitattu 6.4.2010]

saatavissa:

http://cgmfileserv.ucdavis.edu/Data%20Acquisition%20Infrastructure/signal%20conditioning%20hardware/BusWorks%20Ethernet%20Relay%20%28982EN-4012%29/intro_modbusTCP_765a.pdf

Agromag.2009. Intoduction to Profibus DP [pdf-dokumentti] Agromag The leader in Industrial I/O [viitattu 6.4.2010]

saatavissa:

<http://www.scribd.com/doc/15560356/Profibus-Introduction-698a>

ASCII. 2010 asciiTable [www-dokumentti] ASCII [viitattu 6.4.2010]

saatavissa:

www.asciitable.com

Atmel. 2010. 8-bit Microcontroller with 16K Bytes In-System Programmable Flash [pdf-dokumentti]. Atmel Corporation. [viitattu 6.4.2010]

saatavissa:

<http://www.datasheetcatalog.org/datasheet/atmel/2466S.pdf>

Atmel. 2010. 8-bit Microcontroller with 32K Bytes In-System Programmable Flash. pdf-dokumentti]. Atmel Corporation. [viitattu 6.4.2010]

saatavissa:

www.atmel.com/atmel/acrobat/doc2503.pdf

Atmel. 1997. Two-wire Serial EEPROM 1M (131,072 x 8) [pdf-julkaisu] Atmel Corporation [viitattu 6.4.2010]

saatavissa:

atmel.com/dyn/resources/prod_documents/doc5194.pdf

Audon. 2010. 1-wire chip [www-dokumentti]. Audon Electrics. [viitattu 7.4.2010]

saatavissa:

http://www.audon.co.uk/images/1wire_chip.jpg

B&R. 2010. Profibus DP [www-dokumentti]. B&R Perfection in Automation [viitattu 7.4.2010]

saatavissa:

images_br_productcatalogue/images_content/Profibus_DP_-HART-Page127_rdx_598x520_95.jpg

Best-Microcontroller-projects.2010.how-rs232-works-tx-logic-rs232-diag [www-dokumentti]. Best-Microcontroller-projects [viitattu 7.4.2010]

saatavissa:

<http://www.best-microcontroller-projects.com/image-files/how-rs232-works-tx-logic-rs232-diag.png>

Chipkin.2007. What is RS485 EIA485. [www-dokumentti]. Chipkin Automation Systems. [viitattu 6.4.2010]

saatavissa:

<http://www.chipkin.com/articles/what-is-rs485-eia-485>

Control Resources.2010. i2c_audio [www-dokumentti]. Control Resources Incorporated. [viitattu 7.4.2010]

saatavissa:

http://www.controlres.com/images/I2C_audio.jpg

Digital Core.2002. i2c-bus specification. [pdf-dokumentti] Digital Core Design. [viitattu 6.4.2010]

saatavissa:

http://www.dcd.pl/dcdpdf/I2C_doc.pdf

Electfans 2010. 200942111426289.gif [www-dokumentti]. Electfans [viitattu 7.4.2010]

saatavissa:

<http://www.electfans.com/article/UploadPic/2009-4/200942111426289.gif>

Electro Tech.2010. [www.dokumentti]. Electro Tech [viitattu 7.4.2010]

saatavissa:

<http://www.electro-tech-online.com/attachments/micro-controllers/35978d1256492673-multidrop-communication-using-pic-rs485.png>

Irrigander. 2010. [www-dokumentti]. Irrigander Irrigation Extender [viitattu 7.4.2010]

saatavissa:

<http://www.irrigander.com/lawnsite/2-wire.jpg>

Ferret.2010. Modbus TCP is now available in MNS-DX 1.3 for DX800 and DX40 Serial Device Routers [www-dokumentti] Ferret Australia's Manufacturing and Industrial Directory. [viitattu 7.4.2010]

saatavissa:

<http://www.ferret.com.au/c/Dewar-Electronics/Modbus-TCP-is-now-available-in-MNS-DX-1-3-for-DX800-and-DX40-Serial-Device-Routers-n674455>

Fpga4fun.2010. SPI_slave3 [www-dokumentti] Fgpafun [viitattu 7.4.2010]

saatavissa:

http://www.fpga4fun.com/images/SPI_slave3.gif

Fpga4fun.2009. What is SPI [www-dokumentti] fpga4fun [viitattu 6.4.2010]

saatavissa:

<http://www.fpga4fun.com/SPI1.html>

Global Time System.2010.tqpbio-config-rs485 [www-dokumentti] Global time system [viitattu 7.4.2010]

saatavissa:

<http://www.gtstime.com/acroprint/timeqplusbio/images/tqpbio-config-rs485.jpg>

Korea Imagineering.2010. F9 [www-dokumentti] Korea imagineering [viitattu 7.4.2010]

saatavissa:

kiming.co.kr/bbs/data/cheditor/0809/f9.jpg

Lammert Bies 2010. Modbus interface tutorial [www-dokumentti] Lammert Bies [viitattu 6.4.2010]

saatavissa:

<http://www.lammertbies.nl/comm/info/modbus.html>

Lammert Bies 2010. RS485 serial information [www-dokumentti] Lammert Bies [viitattu 6.4.2010]

saatavissa:

<http://www.lammertbies.nl/comm/info/RS-485.html>

Linear Technology.2003. Single 5V RS232/RS485 [pdf-dokumentti] Linear Technology [viitattu 6.4.2010]

saatavissa:

<http://cds.linear.com/docs/Datasheet/1387fs.pdf>

Maxim-IC. 2010.4477 [www-dokumentti]. Maxim Innovation Delivered. [viitattu 7.4.2010]

saatavissa:

<http://www.maxim-ic.com/images/appnotes/4477/4477Fig02.gif>

Maxim-IC. 2010. 1-wire Communication Through Software [pdf-dokumentti]. Maxim Innovation Delivered [viitattu 6.4.2010]

saatavissa:

pdfserv.maxim-ic.com/en/an/AN126.pdf

Maxim-IC.2010. Using a UART to Implement a 1-Wire Bus Master [pdf-dokumentti] Maxim Innovation Delivered. [viitattu 6.4.2010]

saatavissa:

<http://pdfserv.maxim-ic.com/en/an/AN214.pdf>

Maxim-IC.2010. DS18S20 High-precision 1-wire Digital Thermometer [pdf-dokumentti] Maxim Innovation Delivered. [viitattu 6.4.2010]

saatavissa:

<http://datasheets.maxim-ic.com/en/ds/DS18S20.pdf>

Maxim-IC.2010.1021-bit, 1-wire eeprom [pdf-dokumentti] Maxim Innovation Delivered [viitattu 6.4.2010]

saatavissa:

<http://datasheets.maxim-ic.com/en/ds/DS2431.pdf>

Maxim-IC.2010.max232 2006. max220-max249 [pdf-dokumentti] Maxim Innovation Delivered [viitattu 6.4.2010]

saatavissa:

<http://pdfserv.maxim-ic.com/en/ds/MAX220-MAX249.pdf>

MCT Paul & Schere. 2010 SPI2 [www-dokumentti] MCT Paul & Schere [viitattu 7.4.2010]

saatavissa:

<http://www.mct.net/faq/spi2.gif>

Microchip.2006. 25AA010A/25LC010A 1K SPI Bus Serial EEPROM [pdf-dokumentti] Microchip [viitattu 6.4.2010]

saatavissa:

<http://ww1.microchip.com/downloads/en/DeviceDoc/21832C.pdf>

MikroElektronika 2010. Architecture and Programming of 8051 Microcontrollers. [www-dokumentti] MikroElektronika [viitattu 6.4.2010]

saatavissa:

<http://www.mikroe.com/en/books/8051book/>

Mikro Elektronika. 2010 7.10 serial communication [www-dokumentti] MikroElektronik [viitattu 7.4.2010]

saatavissa:

http://www.mikroe.com/en/books/picbook/7_10chapter.htm

Motorola. 2000 SPI Block Guide v03.06 [pdf-dokumentti] Motorola INC.[viitattu 6.4.2010]

saatavissa:

<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-12/DSA-222613.pdf>

NXP. 2010. SE97B DDR memory module temp sensor with integrated SPD [pdf-dokumentti]NXP SemiConductors [viitattu 6.4.2010]

saatavssa:

http://www.nxp.com/documents/data_sheet/SE97B.pdf

Ptm:RS232. Sarjaväylä eli RS232 eli COM-portti [www-dokumentti] [viitattu 6.4.2010]

saatavissa:

<http://users.utu.fi/ptmusta/rs232.shtml>

Real Time Automation. 2010. PROFIBUS Comprehensive Protocol Overview [www-dokumentti] Real Time Automation [viitattu 6.4.2010]

saatavissa:

<http://www.rtaautomation.com/profibus/>

Real Time Automation. 2010 Modbus RTU overview. [www-dokumentti] Real Time Automation [viitattu 6.4.2010]

saatavissa:

<http://www.rtaautomation.com/modbusrtu/>

Robot Electronics. 2010. Using I2C. using the I2C Bus [www-dokumentti] Robot Electronics [viitattu 6.4.2010]

saatavissa:

http://www.robot-electronics.co.uk/html/using_the_i2c_bus.htm

Rohm semiconductor. 2008. i2c bus Serial EEPROMs [pdf-dokumentti] Rohm Semiconductor [viitattu 6.4.2010]

saatavissa:

<http://www.rohm.com/products/databook/general/pdf/br24l01a-w-e.pdf>

Roy Vegard Ovesen 2007. RS485 wavefrom [www.dokumentti] Roy Vegard Ovesen [viitattu 7.4.2010]

saatavissa:

http://commons.wikimedia.org/wiki/File:RS-485_waveform.svg

Sipex.2004 SP481R/SP485R 1/10th Unit Load RS-485 Transceiver [pdf-dokumentti] Sipex Corporation [viitattu 6.4.2010]

Saatavissa:

http://www.exar.com/Files/Documents/sipex/datasheets/sp481_485r.pdf

SMAR Industrial Automation.2010. Profibus [www-dokumentti] SMAR Industrial Automation [viitattu 7.4.2010]

Saatavissa:

<http://www.smar.com/profibus.asp>

Texas Instruments 2002. 422 and 485 standards overview and system configurations [pdf-dokumentti] Texas Instruments [viitattu 6.4.2010]

saatavissa:

<http://focus.ti.com/lit/an/slla070c/slla070c.pdf>

WiseGEEK 2010. What is RS-323? [www-dokumentti] WiseGEEK [viitattu 6.4.2010]

saatavissa:

<http://www.wisegeek.com/what-is-rs-232.htm>

Zoglab MicroSystems co.,Ltd. RS485C [www-dokumentti] Zoglab MicroSystems co.,Ltd [viitattu 7.4.2010]

saatavissa:

<http://www.zoglab.cn/en/image/RS485c.gif>