

Jarkko-Pekka Kilpeläinen

SULAUTETUN LAITTEEN OHJAUS MOBIILILAITTEELLA

Insinööriö
Kajaanin ammattikorkeakoulu
Tekniikan ala
Tietotekniikka
Kevät 2014



Koulutusala Tekniikan ala	Koulutusohjelma Tietotekniikka
Tekijä(t) Jarkko-Pekka Kilpeläinen	
Työn nimi Sulautetun laitteen ohjaus mobiililaitteella	
Vaihtoehtoiset ammattiopinnot	Toimeksiantaja Markku Karppinen, Kajaanin ammattikorkeakoulu
Aika Kevät 2014	Sivumäärä ja liitteet 47
<p>Tässä insinöörityössä oli tarkoituksena toteuttaa Kajaanin ammattikorkeakoululle toimiva esimerkki mobiililaitteen käyttöliittymäsovelluksen ja sulautetun laitteen välisestä kommunikoinnista. Työssä käytetty käyttöliittymäsovellus on Kajapro Oy:n kehittämä, ja sitä käytetään Android-pohjaisella tablet-tietokoneella. Sovelluksen pääasiallinen käyttötarkoitus on toimia eräässä koulun projektissa älykkään perävaunun käyttöliittymänä, mutta sitä voidaan käyttää myös koulun muissa opiskelijaprojekteissa.</p> <p>Työssä toteutettiin sulautetun laitteen mikrokontrollerille ohjelmakoodi. Ohjelmakoodi toimii mikrokontrolleriin esimerkin vuoksi liitettyjen laitteiden ohjaamisen lisäksi tulkkina mobiililaitteelta tuleville käskyille. Sulautettuun laitteeseen liitettävät laitteet ovat kolme dc-moottoria ja kiihtyvyyssanturi. Mobiililaitteella voidaan ohjata moottoreita ja antureita voidaan lukea tietoa, joka tulostetaan mobiililaitteen näytölle. Näin saadaan esimerkki ohjauksen toiminnasta molempiin suuntiin. Tiedonsiirto sulautetun laitteen ja mobiililaitteen välillä tapahtuu langattomasti Bluetooth-yhteyden yli.</p> <p>Työn teoriaosuudessa käsitellään suurimmaksi osaksi Atmel Atmega 128L -mikrokontrollerin ominaisuuksia ja Bluetooth-teknologiaa. Mikrokontrollerin ominaisuuksia käsitellään vain työn käytännön osuudessa käytettyjen ominaisuuksien osalta. Bluetooth-osiossa perehdytään mm. standardin perusteisiin ja protokollakerroksiin, sekä tarkastellaan tietoliikennettä mm. datakehysien osalta.</p> <p>Työn lopputuloksena on ohjelmakoodi, joka toimii viestinkäsittelijänä mikrokontrollerin ja mobiililaitteen välillä. Ohjelmakoodissa otetaan myös käyttöön mikrokontrollerin eri ominaisuuksia ja hyödynnetään niitä. Työn lopputulos saavutti projektin alussa määritellyt ominaisuusvaatimukset.</p>	
Kieli	Suomi
Asiasanat	Atmel, Atmega128L, Bluetooth, C-kieli, Langaton
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Engineering	Degree Programme Information Technology
Author(s) Jarkko-Pekka Kilpeläinen	
Title Controlling Embedded Devices by a Mobile Device	
Optional Professional Studies	Commissioned by Markku Karppinen, Kajaani University of Applied Sciences
Date Spring 2014	Total Number of Pages and Appendices 47
<p>The aim of this thesis was to implement a working example of communication between a mobile device's user interface application and an embedded device. The task was commissioned by Kajaani University of Applied Sciences. The application was developed by Kajapro Oy and is used in an Android-based tablet computer. The main purpose for the application was to use it as a GUI (Graphical User Interface) in the intelligent trailer project. However, the application can also serve in other student projects of the university.</p> <p>The main task in this thesis was to create a program code for the microcontroller. That code would control the devices connected to the microcontroller and also act as a message parser between the embedded device and the mobile device. The embedded device is connected to motors and sensors to get some demonstration of the connection. The motors can be controlled by the mobile device and the data of sensors can be read on the screen of the mobile device. In this way, it is possible to show that data transmission functions in both directions. The data between embedded and mobile devices is carried wirelessly via Bluetooth.</p> <p>The theoretical part of this thesis is mainly focused on the Bluetooth technology and Atmel Atmega 128L, the microcontroller of the embedded device. Only the features of the microcontroller used in the practical part of the thesis are taken under review. In the Bluetooth section, the basic knowledge, the protocol layers and data transmission are discussed.</p> <p>The end result is a program code which takes care of the communication between the microcontroller and the mobile device. In addition, the program code also controls the various features of the microcontroller. Therefore, the result meets the requirements.</p>	
Language of Thesis	Finnish
Keywords	Atmel, Atmega128L, Bluetooth, C-language, Wireless
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input type="checkbox"/> Library of Kajaani University of Applied Sciences

ALKUSANAT

Haluan kiittää Kajapro Oy:tä, joka antoi minulle mahdollisuuden tehdä insinöörityön heidän tuotteeseensa liittyen. Kiitokset myös kaikille työn ohjauksessa auttaneille sekä koulun että Kajapro Oy:n päässä.

SISÄLLYS

1 JOHDANTO	1
2 ATMEL ATMEGA128L -MIKROKONTROLLERI	3
2.1 Tekniset ominaisuudet	3
2.2 Sarjaportti	5
2.3 Keskeytykset	7
2.3.1 Ulkoinen keskeytys	8
2.3.2 Ajastimet	9
2.3.3 PWM-pulssinmuodostus	10
2.4 AD-muunnin	12
2.5 Virransäästö	14
3 BLUETOOTH	17
3.1 Versiot	17
3.2 Topologiat	18
3.3 Taajuusalueet	19
3.4 Protokollakerrokset	20
3.5 Datakehukset	23
3.6 Laiteosoitteet	25
4 KÄYTTÖLIITTYMÄSOVELLUS	26
5 KEHITYSYMPÄRISTÖ	27
6 TYÖN SUORITUS	30
6.1 Kehitysympäristön asennukset	30
6.2 Sarjaliikenteen käyttöönotto	31
6.3 Ohjausviestien käsittely	31
6.4 Bluetooth-yhteyden muodostaminen	32
6.4.1 Bluetooth-moduulin kytkentä	32
6.4.2 Yhdistäminen	35
6.4.3 Moduulin alustaminen C-kielellä	36
6.4.4 Komennot	38
6.4.5 Simply Blue Commander	38

6.5 Mikrokontrollerin muiden ominaisuuksien käyttöönotto	40
6.6 Esimerkkikäyttöliittymän luonti	41
7 TULOKSET	42
8 TULOKSIEN ANALYSOINTI	44
9 YHTEENVETO	45
LÄHTEET	46

SYMBOLILUETTELO

ACL	Asynchronous Connectionless Link
AD	Analog to Digital
ADC	AD Converter
ADCSR	ADC Control and Status Register
ADEN	ADC Enable
ADLAR	ADC Left Adjust Result
ADPS	ADC Prescaler Select
ADSC	ADC Start Conversion
AFH	Active Frequency Hopping
AMP	Alternative MAC/PHY
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
CTC	Clear Timer on Compare
EDR	Enhanced Data Rate
EEPROM	Electrically Erasable Programmable Read-Only Memory
EICR	External Interrupt Control Register (-A & -B)
EIMSK	External Interrupt Mask Register
FIFO	First In First Out
FM	Frequency Modulation
GIAC	General Inquiry Access Code
GUI	Graphical User Interface
HS	High Speed (Core Configuration)
I/O	Input/Output
IEEE	The Institute of Electrical and Electronics Engineers
INT	External Interrupt
ISC	Interrupt Sense Control
ISM	Industrial, Scientific and Medical (radio bands)
JTAG	Joint Test Action Group
L2CAP	Logical Link Control and Adaption Protocol (L2 = LL)
LAP	Lower Address Part
LSB	Least Significant Bit
MAC	Media Access Control

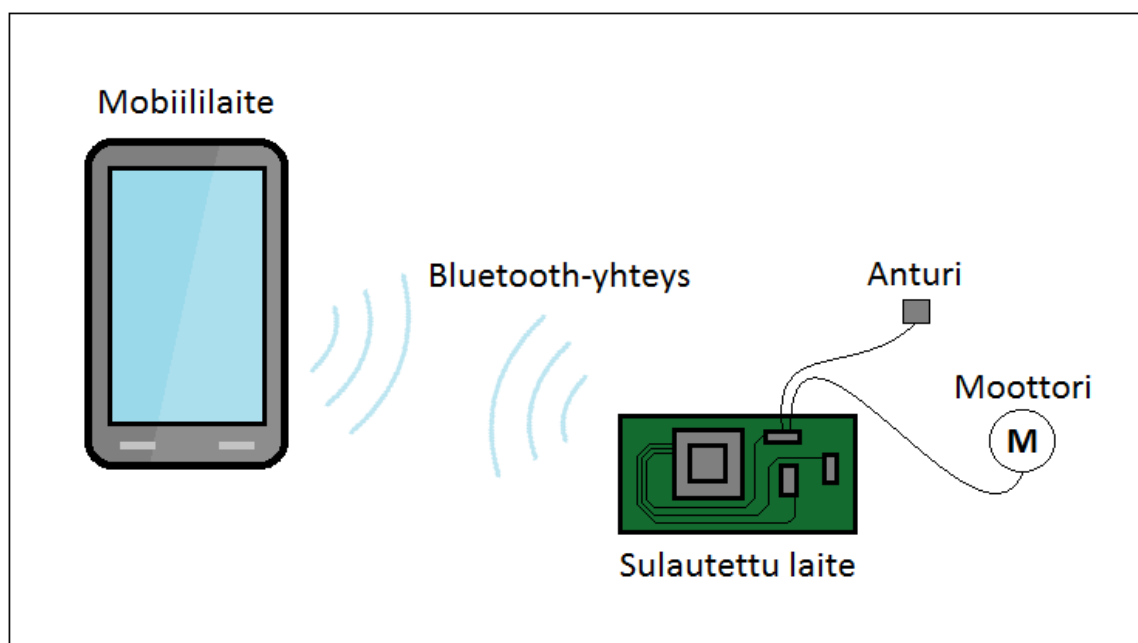
MCUCR	Microcontroller Unit Control Register
MSB	Most Significant Bit
MUX	Multiplexer
NAP	No-significant Address Part
OC	Output Compare
OCR	Output Compare Register
OSI	Open Systems Interconnection (-model)
PC	Personal Computer
PIN	Personal Identification Number
PHY	Physical layer of the OSI-model
PWM	Pulse Width Modulation
REFS	Reference Selection
RFCOMM	Radio Frequency Communication
RISC	Reduced Instruction Set Computer
RX	Receive
SCO	Synchronous Connection-Oriented (link)
SDP	Service Discovery Protocol
SE	Sleep Enable
SM	Sleep Mode
SPI	Serial Peripheral Interface
SPP	Serial Port Profile
SRAM	Static Random Access Memory
TCNT	Timer/Counter
TDMA	Time Division Multiple Access
TQFP	Thin Quad Flat Pack
TWI	Two Wire Interface
TX	Transmit
UAP	Upper Address Part
UART	Universal Asynchronous Receiver and Transmitter
UBRR	USART Baud Rate Register (-L = Low, -H = High)
UCSR	USART Control and Status Register
UDR	USART Data Register
USART	Universal Synchronous and Asynchronous Receiver and Transmitter
USB	Universal Serial Bus

1 JOHDANTO

Kajaanin ammattikorkeakoulu sai eräältä yritykseltä opiskelijaprojektin, jossa yrityksen tuotteena olevaan perävaunuun suunnitellaan ja toteutetaan mikrokontrollerilla ohjattuja älykkäitä ominaisuuksia. Näitä ominaisuuksia ovat mm. perävaunuun lavan aktiivinen suorassa pitäminen ja perävaunun lastin massan mittaaminen.

Keväällä 2013 ollessani Kajapro Oy:ssä suorittamassa harjoittelujaksoa saatiin tieto, että perävaunuprojektiin tarvittaisiin käyttöliittymäratkaisu, jolla mikrokontrolleria voitaisiin ohjata. Kajapro Oy on kajaanilainen ohjelmistoihin suuntautunut yritys, joka tarjosi käyttöliittymäohjelmistollaan tähän ongelmaan ratkaisua. Ohjelmisto täytyi vain enää saada toimimaan sulautetun laitteen kanssa. Aihe insinööriytyölle oli näin syntynyt.

Työn tarkoituksena on saada mobiililaitteen ja sulautetun laitteen välille luotua kaksisuuntainen tietoliikenneyhteys. Tätä yhteyttä pitkin voidaan vastaanottaa anturidataa ja lähettää ohjaustietoa moottorille. Mobiililaitteen yhteyden muodostamiseen käytettävä sovellus on Kajapro Oy:n kehittämä käyttöliittymäohjelmisto. Tietoliikenneyhteys on sarjamuotoinen, ja se muodostetaan Bluetoothin välityksellä. Kuvassa 1 on esitetty yksinkertaistetettu periaatekuva työn tavoitteesta.



Kuva 1. Periaatekuva työn tavoitteesta

Työn lopputulos on tarkoitettu Kajaanin ammattikorkeakoululle esimerkiksi, kuinka Kajapro Oy:n ohjelmistoa käytetään. Työn havainnollistamista varten on toteutettu myös pienimuotoinen käyttöliittymä, jonka avulla voidaan muun muassa ohjata moottorialustaa ja piirtää vastaanotetusta anturitiedosta graafista käyrää. Esimerkin myötä seuraavat opiskelijat pystyvät rakentamaan todennäköisesti myöhemmin ilmenevien vaatimusten mukaisen käyttöliittymän perävaunuprojektiin.

2 ATMEL ATMEGA128L -MIKROKONTROLLERI

Työssä käytettäväksi sulautetuksi laitteeksi valittiin Atmel Atmega 128L -mikrokontrolleri ja sen liitäntäalustaksi STK600-alusta. Kyseinen mikrokontrolleri valittiin, koska sitä oli käytetty myös aiemmin koulun projekteissa. Perehtymiseen ei menisi näin kovin kauan, ja aikaa jäisi enemmän itse työn toteutukseen. Tässä luvussa tarkastellaan mikrokontrollerin perustietoja ja ominaisuuksia, joita työssä on käytetty.

2.1 Tekniset ominaisuudet

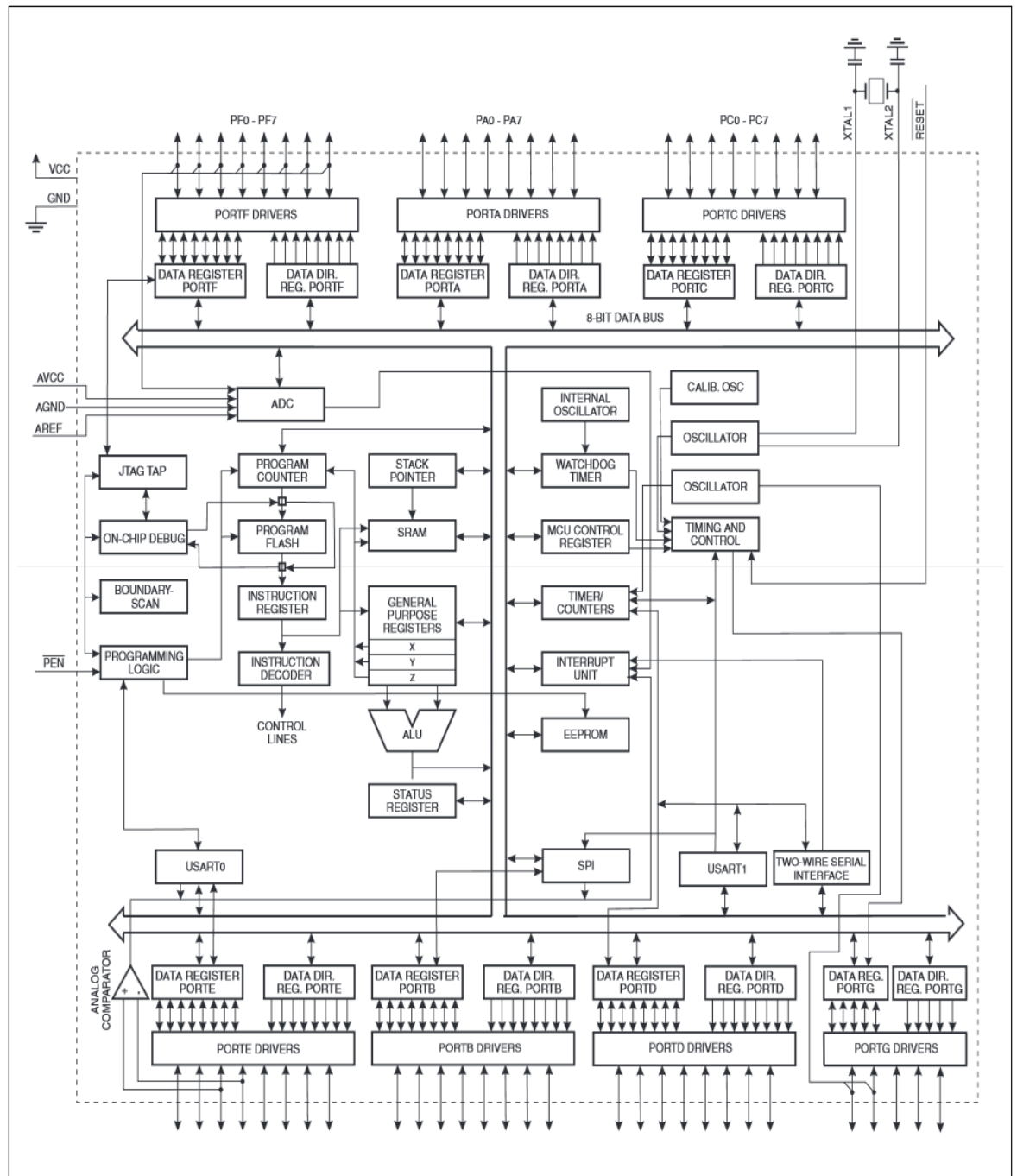
Atmel Atmega 128L on 8-bittinen RISC-arkkitehtuurityyppinen mikrokontrolleri, jonka kellotaajuus on säädettävissä ohjelmallisesti välille 1–8 MHz [1, s. 1]. RISC-arkkitehtuurityyppiset prosessorit käyttävät yksinkertaistettuja konekielen käskyjä monimutkaisempien käskyjen sijaan. Näille prosessoreille on tyypillistä suuri rekisterien määrä. Tällä pyritään pitämään prosessorin ja keskusmuistin välinen tarpeeton liikenne mahdollisimman vähäisenä. Lisäksi yhdellä käskyllä pyritään saamaan mahdollisimman paljon aikaiseksi. Edellämainitut toimenpiteet tehdään, koska tyypillisesti muistin määrä RISC-prosessorin omaavissa laitteissa on pieni. RISC-prosessorit hyödyntävät myös pipelining-tekniikkaa, joka tarjoaa mahdollisuuden eri toimintojen samanaikaiseen suoritukseen. [2.]

Kontrolleri sisältää neljän kilotavun kokoiset EEPROM- ja SRAM-muistit. SRAM-muistiin tallennetaan oletuksena kaikki muuttujat ja taulukot. Muistin vähäinen koko aiheuttaa ongelman etenkin suurien taulukoiden kanssa. Kuitenkin käytettävissä on edellä mainittujen muistien lisäksi 128 kilotavua ohjelmoitavaa flash-muistia, jonne suuret taulukot on mahdollista tallentaa. Datan lataaminen flash-muistista on ohjelmassa hieman työläämpää kuin SRAM-muistista, ja onkin suositeltavaa, että ”tavallisia” muuttujia sinne ei tallenneta. [1, s. 1.]

Kontrollerissa on JTAG-rajapinta debuggaamista varten. Rajapintaan voidaan kytkeytyä esimerkiksi AVR JTAG ICE -debuggerilla, joka liitetään edelleen USB:llä PC:hen kiinni. [1, s. 1.]

Ajastimia kontrollerissa on yhteensä neljä: kaksi 8-bittistä ja kaksi 16-bittistä. PWM-kanavia on kahdeksan, joista kaksi on 8-bittisiä ja loput kuusi voidaan ohjelmoida toimimaan 2–16-bittisenä. Kontrolleri sisältää myös 10-bittisen AD-muuntimen. Tietoliikenneväyliä kontrollerista löytyy kahden USART-väylän sekä TWI- ja SPI-väylien muodossa. [1, s. 1.]

Mikrokontrollerin lohkokaavio on nähtävissä kuvassa 2. Lohkokaaviosta selviää piirin sisäinen toiminta.



Kuva 2. Atmel Atmega 128L:n lohkokaavio [1, s. 3]

2.2 Sarjaportti

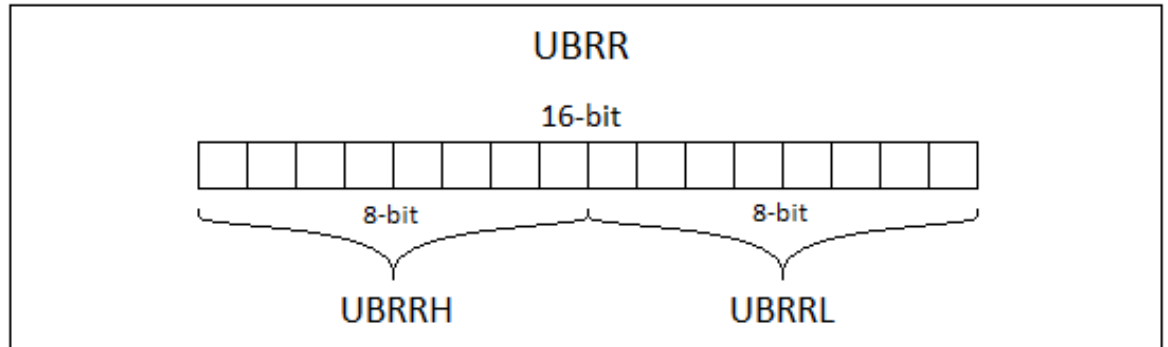
Atmel Atmega128L sisältää kaksi sarjaliikenteen vastaanotto- ja lähetyspiiriä (USART). Nämä piirit ovat nimeltään USART0 ja USART1 [1, s. 170]. USART0:n vastaanotto- ja lähetyspinnit RXD0 ja TXD0 sijaitsevat pinneissä PE0 ja PE1. USART1:n vastaanotto- ja lähetyspinnit RXD1 ja TXD1 taas löytyvät pinneistä PD2 ja PD3. [1, s. 2.]

Ennen käyttöönottoa sarjaportti täytyy alustaa. Tähän toimenpiteeseen kuuluu olennaisimpana osana UBRR-rekisterin (USART Baud Rate Register) alustaminen. Rekisteriin sijoitettavalla arvolla päätetään, käytetäänkö asynkronista vai synkronista toimintamoodia. Asynkronisessa toimintamoodissa tiedonsiirto ei ole tahdistettu, kuten on synkronisessa toimintamoodissa. Asynkroniseen tiedonsiirtoon kuuluu olennaisesti myös start- ja stop-bittien käyttö, joilla ilmaistaan datakehysten loppuminen ja alkaminen [3]. Asynkronisia toimintamoodeja on Atmel Atmega128L:ssä kaksi, joista toinen on ns. normaalinopeuksinen ja toinen kaksinkertaisella nopeudella operoiva. Alustus tapahtuu sijoittamalla rekisteriin taulukon 1 kaavoista saatu luku. [1, s. 173.]

Taulukko 1. UBRR-arvon laskukaavat eri toimintamoodeille [1, s. 173]

Toimintamoodi	UBRR-arvon laskukaava
Asynkroninen (normal mode)	$UBRR = \frac{f_{osc}}{16 * BAUD} - 1$
Asynkroninen (double speed)	$UBRR = \frac{f_{osc}}{8 * BAUD} - 1$
Synkroninen	$UBRR = \frac{f_{osc}}{2 * BAUD} - 1$

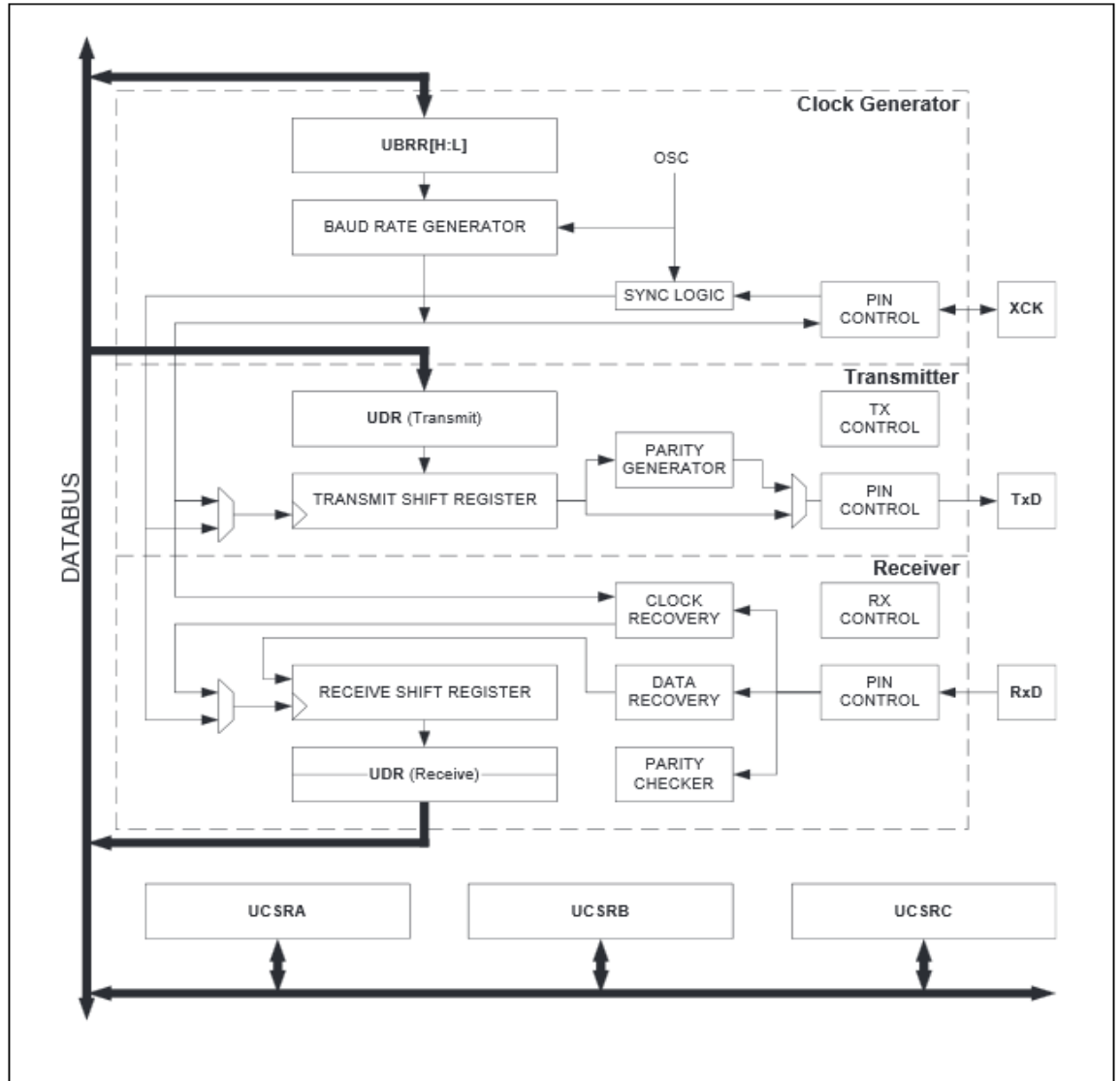
Taulukon 1 muuttuja f_{osc} on kontrollerin oskillaattorin kellotaajuus (esimerkiksi 8 MHz) ja BAUD valittu baudinopeus (esimerkiksi 9600 Bd). UBRR-rekisteri koostuu kahdesta rekisterin osasta, jotka ovat UBRRH ja UBRRL (kuva 3). Kaavasta laskettu UBRR-luku voidaan siis sijoittaa näihin rekisterin osiin myös erikseen. Jos esimerkiksi arvoksi saadaan 8-bittinen luku, voidaan se sijoittaa pelkästään UBRRH-rekisteriin. Suositeltavaa on kuitenkin luvun sijoittaminen suoraan UBRR:lle, jolloin sijoitus menee automaattisesti oikein. [1, s. 173.]



Kuva 3. UBRR-rekisterin muodostuminen.

Jos halutaan liikennöidä dataa molempiin suuntiin, eli lähetys- ja vastaanotto-ominaisuudet vaaditaan sarjaportilta, on nämä aktivoitava erikseen. Toimenpide suoritetaan asettamalla TXEN- (TX-enable, eli lähetys) ja RXEN-bitit (RX-enable, eli vastaanotto) 1-tilaan UCSRB-rekisteristä (USART Control and Status Register B). [1, s. 176.]

UCSR-rekisteristä voidaan myös asettaa sarjaportti aiheuttamaan keskeytyksiä, jolloin tietoliikennettä on helpompi hallita. Näitä ovat esimerkiksi datan lähetyksen (TXCIE-bitti) tai vastaanoton (RXCIE-bitti) aiheuttamat keskeytykset. UDR-rekisteri (USART Data Register) toimii datarekisterinä, jonne sijoitetaan lähetettävä data ja josta otetaan esimerkiksi taulukkoon talteen vastaanotettava data. UDR-rekisteri on FIFO-mallinen ympyräpuskuri, jonka vuoksi puskurissa oleva data on mahdollista lukea ainoastaan kerran. Kuvassa 4 on esitetty USART-piirin lohkokaavio, josta piirin toiminta on nähtävissä. [1, s. 179–182.]



Kuva 4. USART-piirin lohkokkaavio [1, s. 171]

2.3 Keskeytykset

Keskeytykset ovat toimintoja, jotka aiheuttavat mikrokontrollerin senhetkisen toiminnan suorituksen keskeytymisen, jotta on mahdollista suorittaa keskeytyspyynnön vaatimat toimenpiteet. Pyynnöt tulevat tyypillisesti keskeytyssignaalina ulkoiselta laitteelta tai ajastimelta laskurin arvon osuessa ennalta annettuun vertailuarvoon. Myös tietoliikenneporttien, kuten sarjaportin ja TWI:n (Two Wire Interface), datan vastaanoton tai lähetyksen voi ohjelmoida aiheuttamaan keskeytyksen mikrokontrollerilla. [1, s. 59–60.]

Jos keskeytyksiä otetaan Atmel Atmega128L:ssä käyttöön, on ne ohjelmakoodin puolella muistettava yleisesti sallia. Tämä suoritetaan asettamalla AVR Status -rekisterin (SREG) Global Interrupt Enabled -bitti 1-tilaan. [1, s. 10–11.]

2.3.1 Ulkoinen keskeytys

Atmel Atmega128L:ssa on ulkoisille keskeytyksille olemassa kahdeksan keskeytyspinniä. Keskeytyspinnit sijaitsevat porteissa D (INT0–INT3 pinneissä PD0–PD3) ja E (INT4–INT7 pinneissä PE0–PE3). Käytettäessä ulkoisia keskeytyksiä täytyy valittua keskeytystä vastaava bitti kirjoittaa 1-tilaan External Interrupt Mask -rekisteristä (EIMSK). Kuvassa 5 on esitetty EIMSK-rekisterin bittien nimet ja numeroinnit. [1, s. 91.]

bitti nro.							
7	6	5	4	3	2	1	0
INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0

Kuva 5. EIMSK-rekisteri [1, s. 91]

Keskeytykset ovat ohjelmoitavissa aktivoitumaan eri tavoilla. Signaalin nousevalle reunalle ohjelmoitu keskeytys aiheutuu ainoastaan, kun ulkoisen keskeytyksen tulopinnin tila vaihtuu $0 \rightarrow 1$ (jännitetason muutos $0\text{ V} \rightarrow 3,3\text{ V}$ tai 5 V). Laskevalle reunalle ohjelmoitu keskeytys toimii päinvastoin, eli pinnin tilan vaihtuessa $1 \rightarrow 0$ keskeytys aktivoituu. Low-level-keskeytys aiheuttaa jatkuvasti keskeytyksiä, kun pinnin tila on 0-tasossa. Ohjelmakoodissa keskeytyksen asetukset määritetään External Interrupt Control -rekistereissä EICRA ja EICRB (A- ja B-kontrollirekisterit). EICRA-rekisterissä määritellään keskeytyksen aktivoitumisasetukset keskeytyksille INT0–INT3 ja EICRB-rekisterissä keskeytyksille INT4–INT7. Kuvassa 6 on esitetty EICRA-rekisterin bittien nimet ja numeroinnit. [1, s. 89–90.]

bitti nro.							
7	6	5	4	3	2	1	0
ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00

Kuva 6. EICRA-rekisteri [1, s. 89]

Rekisterissä on varattu kutakin keskeytystä kohti kaksi asetusbittiä. Esimerkiksi ulkoista keskeytystä INT2 ohjaavat EICRA-rekisterissä ISC21- ja ISC20-bitit (bitit 5 ja 4). EICRB-

rekisterissä ohjausbitit ovat samalla logiikalla keskeytyksille INT4–INT7. Taulukossa 2 on esitetty ISC-bittien tilat ja niistä aiheutuvat toiminnot.

Taulukko 2. Taulukossa esiintyvä n-muuttuja korvataan konfiguroinnin kohteena olevan keskeytyksen numerolla (esim. INT3 → n = 3) [1, s. 90].

Toiminto	ISCn1	ISCn0
0-tila aiheuttaa keskeytyksen (jatkuva).	0	0
EICRA = ei toimintoa, tila on varattu tulevaan käyttöön. EICRB = sekä laskeva että nouseva reuna aiheuttaa keskeytyksen.	0	1
Laskeva reuna aiheuttaa keskeytyksen.	1	0
Nouseva reuna aiheuttaa keskeytyksen.	1	1

Kuten taulukosta 2 nähdään, keskeytyspinnit INT0–INT3 ja INT4–INT7 omaavat hieman toisistaan poikkeavat ominaisuudet. Tämä on syytä huomioida keskeytyksen aktivoinnin vaatimuksia mietittäessä.

2.3.2 Ajastimet

Ajastimia käytetään, kun mikrokontrollerilla halutaan esimerkiksi jonkin funktion tapahtuvan aina tietyin aikavälein. Atmel Atmega 128L:ssä on neljä ajastinta, joita voidaan käyttää tätä tarkoitusta varten. Ajastimet löytyvät kontrollerista nimillä TIMER_COUNTER0–TIMER_COUNTER3. [1, s. 92 & 111.]

TCCR-rekisteri (Timer/Counter Control Register) on ajastimen kontrollirekisteri. Jokaisella ajastimella on oma kontrollirekisterinsä, joskin 16-bittisissä ajastimissa, TIMER_COUNTER1:ssä ja TIMER_COUNTER3:ssä, tämä rekisteri on jaettu kolmeen osaan (A-, B- ja C-rekisterit). Kontrollirekisteristä voidaan muuttaa mm. laskurin käyttämää aaltomuotoa (Waveform Generation Mode), käytettävän vertailuarvon asetuksia (Compare Match) ja kellopulssin jakajan (Clock Select) asetuksia. Kuvassa 7 on esitetty TIMER_COUNTER2:n asetuksia ohjaava TCCR2-rekisteri. [1, s. 156.]

bitti nro.							
7	6	5	4	3	2	1	0
FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20

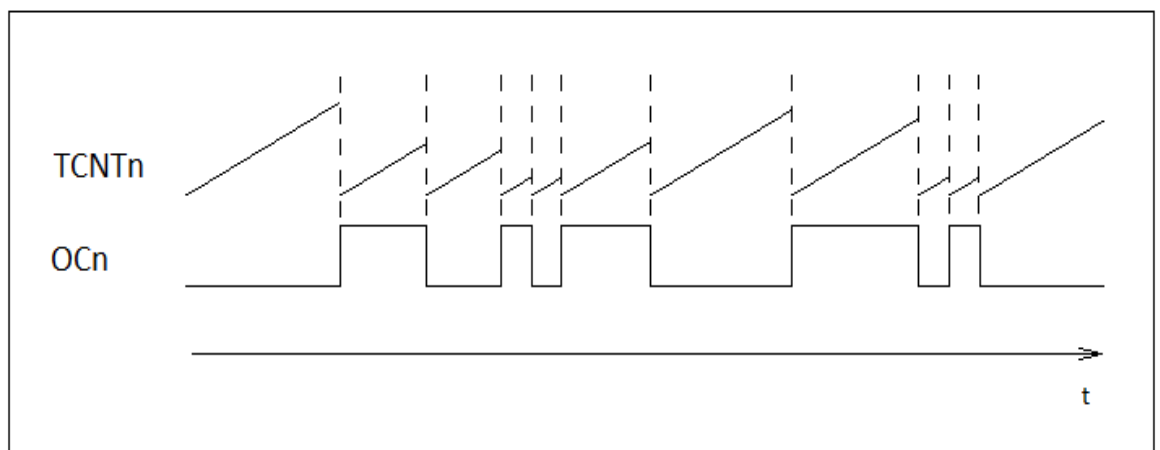
Kuva 7. TIMER_COUNTER2:n TCCR2-rekisteri [1, s. 156]

TIMSK-rekistereistä voidaan säätää ajastimen keskeytysasetukset. Jokaisella ajastimella on oma keskeytysasetusten rekisterinsä [1, s. 158].

2.3.3 PWM-pulssinmuodostus

Ajastimia tarvitaan myös silloin, kun tarvitaan säätää tarkasti pulssin leveyttä esimerkiksi moottorinohjauksessa. Signaalia täytyy tällöin moduloida, ja tätä modulointitapaa kutsutaan PWM-moduloinniksi. PWM-toimintamoodeja (aaltomuotoja) on olemassa erilaisia, ja ne valitaan käyttöön WGM-biteillä. Näitä tiloja käsitellään seuraavaksi. [1, s. 120.]

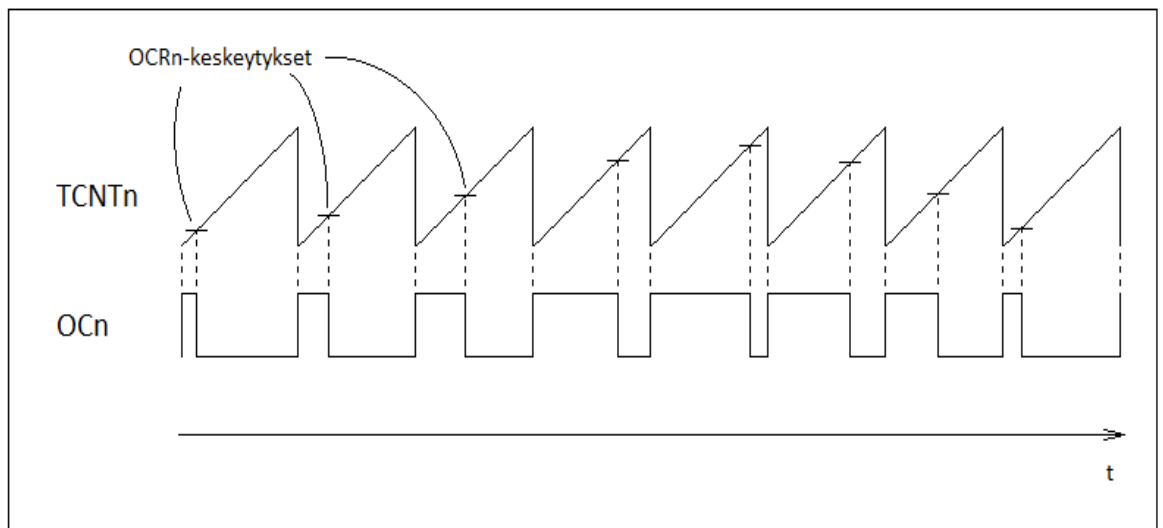
CTC-tilassa (Clear Timer on Compare Match) manipuloidaan itse laskurin arvoa. Laskurin arvo nollataan aina, kun laskuri saavuttaa asetetun vertailuarvon. Samalla ohjattavan lähtöpinnan tila muuttuu. CTC-tila ei ole varsinainen PWM-toimintamoodi, mutta sen avulla voidaan kuitenkin kontrolloida lähtöpinnan taajuutta hyvin tarkasti. [1, s. 150.]



Kuva 8. Esimerkki pulssin muodostumisesta CTC-tilassa. [1, s. 153]

Kuvan 8 tapauksessa TCNT-laskurin (Timer/Counter) arvo alkaa nolasta ja nousee haluttuun raja-arvoon saakka. Kun raja-arvo on saavutettu, laskuri asetetaan jälleen arvoon nolla ja se jatkaa samalla kaavalla uudelleen. Laskurin saavuttaessa raja-arvon myös ohjattavan pinnan tila vaihtuu. Raja-arvon suuruus siis määrää, kuinka leveä pulssijakso on.

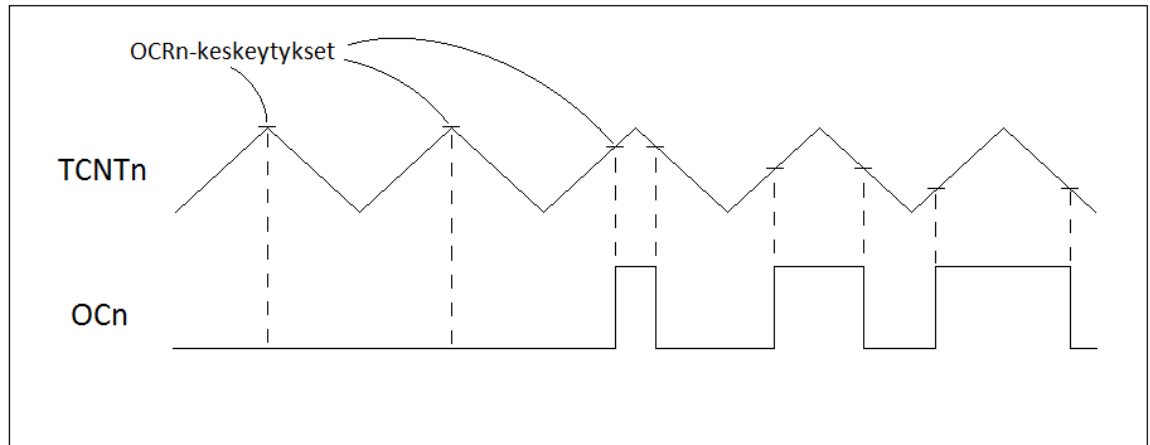
Varsinaisista PWM-toimintamoodeista Fast PWM -toimintamoodilla saadaan aikaiseksi korkeataajuuksista PWM-pulsseja. Taajudet ovat kaksi kertaa suurempia kuin esimerkiksi Phase Correct PWM -tilassa, johtuen laskurin single slope -aaltomuodosta. Tämän lisäksi Fast PWM -moodi soveltuu hyvin myös tehon säätelyyn. Koska tässä työssä käytetyt moottorit ovat kokoluokaltaan pieniä, on Fast PWM -moodia käytetty niiden ohjaamiseen tarjoamansa korkean taajuuden vuoksi. Fast PWM -moodin periaate on esitetty kuvassa 9. [1, s. 151.]



Kuva 9. Esimerkki tässä työssä moottorinohjaukseen käytetystä Fast PWM -pulsin muodostamisesta. [1, s. 151]

Kuvassa 9 TCNT-laskuri on asetettu nousemaan maksimiarvoonsa. Pinni (OC) on 1-tilassa, kunnes laskuri saavuttaa OCR:lle asetetun arvon, jolloin tila vaihtuu nolaksi. Tätä arvoa muuttamalla voidaan siis säätää pinnin lähdön antamaa keskiarvoista tehoa.

Toisin kuin Fast PWM -moodissa, Phase Correct PWM -moodissa laskuri aluksi nousee ja saavuttaessaan huippuarvonsa alkaa laskea kohti pohja-arvoaan. Laskuria ei siis nolleta missään vaiheessa. Näin ollen OCR:lle asetettu arvo saavutetaan sekä laskurin noustessa että laskiessa. Tämän vuoksi Phase Correct PWM -moodi ei voi tarjota niin korkeita taajuuksia kuin Fast PWM -moodi. Kuitenkin tämä ominaisuus tarjoaa tarkemman ohjauksen, ja se onkin yleisesti käytetty moottoreiden ohjauksessa. Phase Correct PWM -pulsseista on nähtävissä visuaalinen esitys kuvassa 10. [1, s. 152.]



Kuva 10. Phase Correct PWM -toimintamoodin havainnekuva [1, s. 152]

2.4 AD-muunnin

AD-muunnin muuntaa analogisen signaalin, esimerkiksi anturilta tulevan tulojännitteen, digitaaliseksi arvoksi tietyllä resoluutiolla. Atmel Atmega 128L sisältää 10-bittisen AD-muuntimen. Muunnin on kytketty 8-kanavaiseen multiplekseriin, jolla valitaan muunnoksessa aktiivisena oleva tulojännitekanava. Jokaiseen kanavaan on mahdollista kytkeä oma tulojännite. Muunnokseen käytettävän kanavan valinta tehdään ohjelmallisesti. Muunnin on 8-kanavainen, ja tulot sijaitsevat portissa F (pinnit PF0–PF7) [1, s. 230.]

Ohjelmallisessa toteutuksessa on AD-muunnin ensin alustettava. Muuntimen käyttämä referenssijännite määritetään ADMUX-rekisterin biteillä REFS1 ja REFS0. Koska AD-muunnos on 10-bittinen, tarvitaan muunnokselle kaksi rekisteriä, jotka ovat 8-bittiset ADCL- (low) ja ADCH- (high) rekisterit. Toinen rekistereistä täyttyy kokonaan, ja toisesta rekisteristä käytetään ainoastaan kaksi bittiä. ADLAR-bitillä määritetään, kummasta rekisteristä alkaen ”täyttö” aloitetaan. MUX-biteillä valitaan käytettävä tulojännitekanava, kanavan differentiaalisuus ja vahvistus. Jos vahvistusta tai differentiaalista tuloa ei tarvita, riittää kun MUX4- ja MUX3- bitit pidetään 0-tilassa ja valitaan käytettävä tulojännitekanava (ADC7–ADC0) MUX2- MUX1- ja MUX0-biteillä MUX2-bitin ollessa MSB ja MUX0 LSB. ADMUX-rekisteri on esitetty kuvassa 11. [1, s. 242–243.]

bitti nro.							
7	6	5	4	3	2	1	0
REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0

Kuva 11. ADMUX-rekisteri [1, s. 242]

ADCSRA-rekisteristä täytyy muunnoksen toteuttamiseksi sallia muunnokset asettamalla ADEN-bitti 1-tilaan. Itse muunnos käynnistetään asettamalla ADSC-bitti 1-tilaan. [1, s. 242.] ADCSRA-rekisteri on esitetty kuvassa 12.

bitti nro.							
7	6	5	4	3	2	1	0
ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Kuva 12. ADCSRA-rekisteri [1, s. 244]

Muunnoksen valmistumisnopeuteen voidaan vaikuttaa prescaler-biteillä (ADPS2–ADPS0). Nämä bitit määrittävät jakajan CPU:n kellotaajuudelle. Jakamalla CPU:n kellotaajuus prescaler-biteillä määritellyllä jakajalla, saadaan lopputulokseksi kellopulssi, jota käytetään AD-muuntimen kellopulssina. Jakajan arvo määräytyy taulukon 3 mukaisesti. AD-muuntimen kellopulssin taajuus on pyrittävä pitämään 50–200 kHz:n välillä, jotta muunnoksen tarkkuus ei kärsisi. Esimerkiksi CPU:n kellotaajuuden ollessa 8 MHz pienin ja suurin käytettävä jakaja lasketaan seuraavasti:

$$div_{MIN} = \frac{clk_{CPU}}{clk_{ADCMAX}} = \frac{8 \text{ MHz}}{200 \text{ kHz}} = 40 \quad (1)$$

$$div_{MAX} = \frac{clk_{CPU}}{clk_{ADCMIN}} = \frac{8 \text{ MHz}}{50 \text{ kHz}} = 160. \quad (2)$$

Taulukossa 3 on esitetty, kuinka ADPS-bittien tilat vaikuttavat CPU:n kellopulssin jakajaan. Tässä esimerkissä jakajaksi voidaan siis valita joko 64 tai 128, jotta AD-muuntimen kellopulssi saadaan oikealle taajuusvälille. [1, s. 233.]

Taulukko 3. Bittiasetukset mikrokontrollerin kellotaajuuden jakajan määräytymiselle [1, s. 245].

ADPS2	ADPS1	ADPS0	Jakaja
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

2.5 Virransäästö

Virransäästötiloja käytetään nimensä mukaisesti säästämään virtaa sammuttamalla tarpeettomia ominaisuuksia mikrokontrollerilta. Atmel Atmega128L:ssä on kuusi virransäästötilaa. Käytettävää virransäästötilaa valittaessa täytyy ottaa huomioon, millä herätteellä virransäästötilasta pitää voida palata takaisin normaaliin tilaan. [1, s. 44.]

Virransäästötiloja kontrolloidaan MCUCR-rekisteristä. Rekisterin SE-bitillä (Sleep enable) kontrolloidaan sitä, onko virransäästöominaisuus käytössä. Jos virransäästölle on tarvetta, täytyy tämä bitti olla asetettu 1-tilaan. SM0-, SM1- ja SM2-biteillä (Sleep mode) valitaan käytettävä virransäästötila. MCUCR-rekisteri on esitetty kuvassa 13, josta nähdään SE- ja SM-bittien paikat. SM-bittien paikassa ei ole virhettä, vaan järjestys rekisterissä on 1, 0 ja 2. Taulukossa 4 on puolestaan esitetty SM-bittien kaikki tilat ja niiden aiheuttamat virransäästötilat. [1, s. 44.]

bitti nro.							
7	6	5	4	3	2	1	0
SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE

Kuva 13. MCUCR-rekisterin bitit [1, s. 30]

Taulukko 4. Virransäästötilat [1, s. 44]

SM2	SM1	SM0	Virransäästötila
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Varattu myöhempää käyttöä varten
1	0	1	Varattu myöhempää käyttöä varten
1	1	0	Standby
1	1	1	Extended standby

Taulukon 4 mukaan idle-tila saadaan aikaiseksi, kun kaikki SM-bitit ovat 0-tilassa. Idle-tilan virransäästö syntyy pysäyttämällä clk_{CPU} ja clk_{FLASH} (CPU:n ja flashin kellopulssit). Idle-tilasta herätään takaisin sisäisen (esim. ajastimet, tietoliikenneportit) tai ulkoisen keskeytyksen avulla. [1, s. 45.]

ADC:n kohinanvaimennustilassa (ADC noise reduction) kohinan ja häiriön määrä on pienennetty optimaaliseksi AD-muuntimen käyttöä varten. Tämä tila vaikuttaa siis SNR-arvoon (Signal to Noise Ratio), joka on desibeliarvo ja kuvaa signaalin tehon suhdetta kohinan tehoon [4, s. 836]. Tämä on aikaansaatu pysäyttämällä clk_{CPU} :n ja clk_{FLASH} :n lisäksi $clk_{I/O}$ (I/O-kellopulssi). Tästä seuraa se, että I/O-kellopulssia käyttävät ominaisuudet, kuten esimerkiksi SPI ja USART, eivät enää herätä mikrokontrolleria. Kun SM-bitit on asetettu taulukon 4 mukaisesti tähän tilaan, käynnistyy AD-muunnos automaattisesti ja ADC Conversion Complete -keskeytyksen herättää mikrokontrollerin. [1, s. 45.]

Power-down-tila on virransäästötila, jossa mikrokontrolleri herää ainoastaan keskeytyksiin, jotka havaitaan asynkronisesti. Asynkronisesti havaittavia keskeytyksiä ovat mm. ulkoisten

keskeytysten low-level-keskeytykset sekä reset-käskyt, kuten watchdog reset, brown-out reset ja external reset. [1, s. 45.]

Power-save-tila on muuten samanlainen kuin power-down-tila, mutta Timer/Counter0-ajastin on toiminnassa, jos ASSR-rekisterin AS0-bitti on asetettu 1-tilaan (kuva 14). Mikrokontrolleri voi herätä tästä tilasta Timer/Counter0-ajastimen ylivuotoon tai ajastimen vertailuarvoon osuessaan. [1, s. 45.]

bitti nro.							
7	6	5	4	3	2	1	0
-	-	-	-	AS0	TCNOUB	OCROUB	TCROUB

Kuva 14. Asynchronous status -rekisteri ASSR [1, s. 44]

Mikrokontrolleri menee standby-tilaan, jos ulkoinen oskillaattori on otettu käyttöön (tuottaa kellopulssin) ja SM-bitit on asetettu taulukon 4 mukaisesti. Standby-tila on samanlainen kuin power-down-tila, mutta tässä tilassa ulkoinen oskillaattori ei pysähdy. [1, s. 46.]

Kuten stand-by-tilassa, myös extended standby -tilan aktivoinnissa täytyy olla ulkoinen oskillaattori otettu käyttöön. Extended standby -tila on samanlainen kuin power-save-tila, mutta tässä tilassa ulkoinen oskillaattori ei pysähdy. [1, s. 46.]

Atmel Atmega128L:ssä on virransäästötilan valinnan jälkeen lisäksi mahdollista sammuttaa tiettyjä yksittäisiä ominaisuuksia. Näitä ominaisuuksia ovat mm. AD-muunnin, analoginen komparaattori ja JTAG-rajapinta. Tällä toiminnalla pyritään entisestään laskemaan mikrokontrollerin virrankulutusta. [1, s. 47–48.]

3 BLUETOOTH

Bluetooth on vuonna 1998 julkaistu standardi lyhyen kantaman radioteknologialle. Sen kehittivät yhdessä IBM, Intel, Nokia, Ericsson ja Toshiba. Nykyään kehittämiseen osallistuu yli 20000 yritystä ja ne muodostavat yhdessä Bluetooth Special Interest Groupin [6]. Bluetooth kehitettiin alun perin yksinkertaiseksi ja luotettavaksi langattomaksi ratkaisuksi toimimaan esimerkiksi RS-232-kaapelin tilalla. [7, s. 325.]

Tässä luvussa käsitellään Bluetooth-protokollaa lähinnä varhaisimpien versioiden ja protokollan perusidean osalta. Luvussa ei ole siis suuremmin käsitelty esimerkiksi uudempien versioiden mukana tulleita tekniikoita ja protokollakerroksia, jotka yleisesti liittyvät lähinnä tiedonsiirron nopeutukseen tai virransäästöön.

3.1 Versiot

Bluetoothin ensimmäiset versiot olivat vuonna 1998 julkaistut 0.8 ja 0.9. Varsinainen standardi 1.0 julkaistiin vuonna 1999, ja vuonna 2002 julkaistiin päivitetty versio 1.1, jonka myös IEEE standardoi nimikkeellä IEEE 802.15.1. Bluetooth-tekniikkaan tehtiin versiossa 1.2 parannuksia, joista mainittavin oli AFH (Adaptive Frequency Hopping), jota käytetään vaihtamaan radiotaajuutta. Hyppelyn taajuus on jopa 1600 hyppyä sekunnissa [5, s. 20]. Tällä tavoin voidaan välttää yhteisten radiotaajuuksien käyttö. [7, s. 325.]

Vuonna 2004 Bluetooth-protokollasta julkaistiin 2.0-versio [8, s. 47]. Version mukana tullut EDR-tekniikka toi mukanaan merkittävän parannuksen tiedonsiirtonopeuteen, joka saavutti nyt jopa nopeuden 3 Mb/s [7, s. 180].

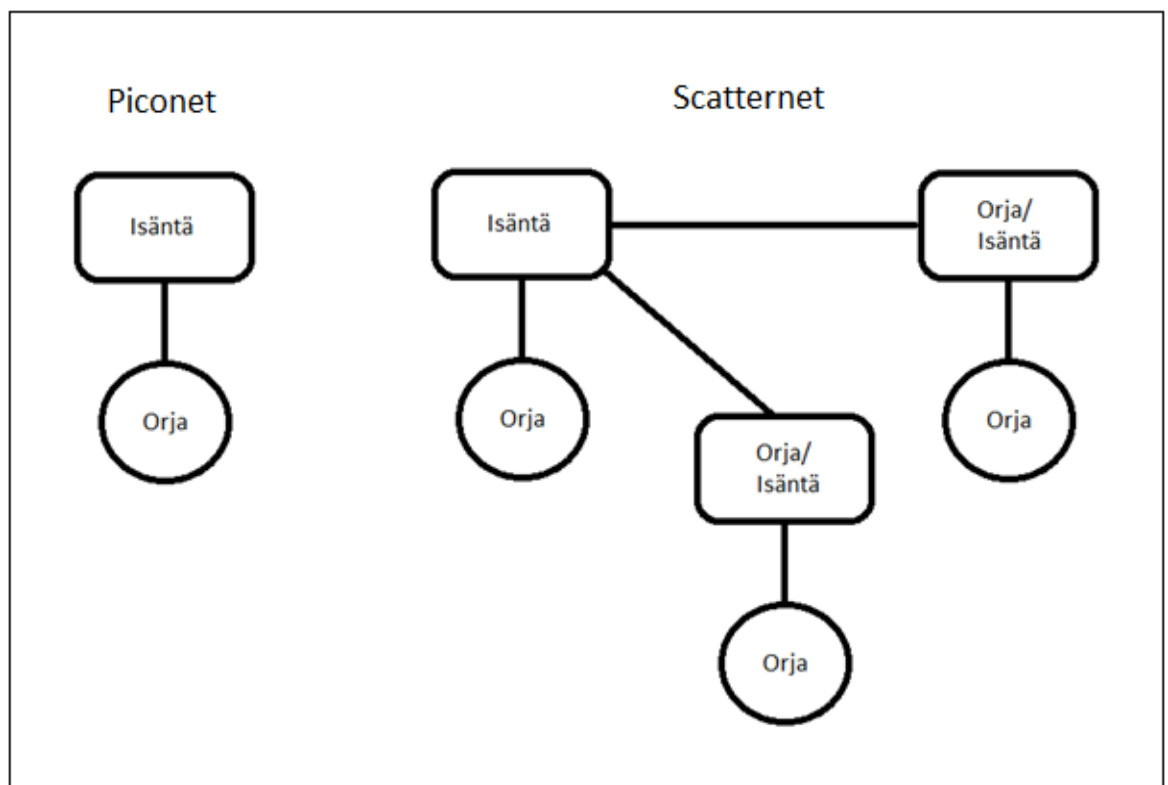
Bluetooth 3.0 -versio julkaistiin vuonna 2009 [8, s. 47]. Sen mukana tulivat AMP- ja HS-tekniikat, ja tämän vuoksi 802.11-sovituserros liitettiin mukaan protokollaan [8, s. 173]. Tiedonsiirtonopeuksiin saatiin jälleen huomattava parannus, ja nyt nopeuksissa päästiinkin jopa useisiin kymmeniin megabitteihin sekunnissa [9].

Vuonna 2010 julkaistussa Bluetooth 4.0 -versiossa uutta oli lähinnä BLE-tekniikan mukaantulo, jonka myötä virrankulutus saatiin pieneneään [10, s. 61]. Protokollan uusin

versio on 4.1, joka on julkaistu vuoden 2013 joulukuussa. Tämä versio sisälsi lähinnä päivityksiä 4.0-versioon, eikä niinkään uusia ominaisuuksia. [11, s. 65.]

3.2 Topologiat

Bluetooth-protokollassa yksi laite toimii isäntänä (master) ja tällä voi olla yksi tai useampi laite orjana (slave). Yksinkertaisimmillaan Bluetooth-verkko sisältää siis ainoastaan isäntä- ja orjalaitteen tai -laitteita. Tällaista verkkoa, jossa päätelaitteita on maksimissaan kahdeksan (isäntä + 7 * orja), kutsutaan pikoverkoksi (piconet) [12, s. 496]. Toisaalta myös orjalaitteilla voi olla omia orjalaitteita. Tämä mahdollistaa monimutkaisemmat verkkorakenteet. Tällaista rakennetta, jossa isäntälaitteella on orjalaitteita, jotka toimivat edelleen orjalaitteiden isäntinä, kutsutaan hajaverkoksi (scatternet). Kuvassa 15 on annettu visuaalinen selvitys edellä kuvatuista topologioista. [7, s. 325–326.]



Kuva 15. Bluetooth-verkkotyypit [7, s. 326]

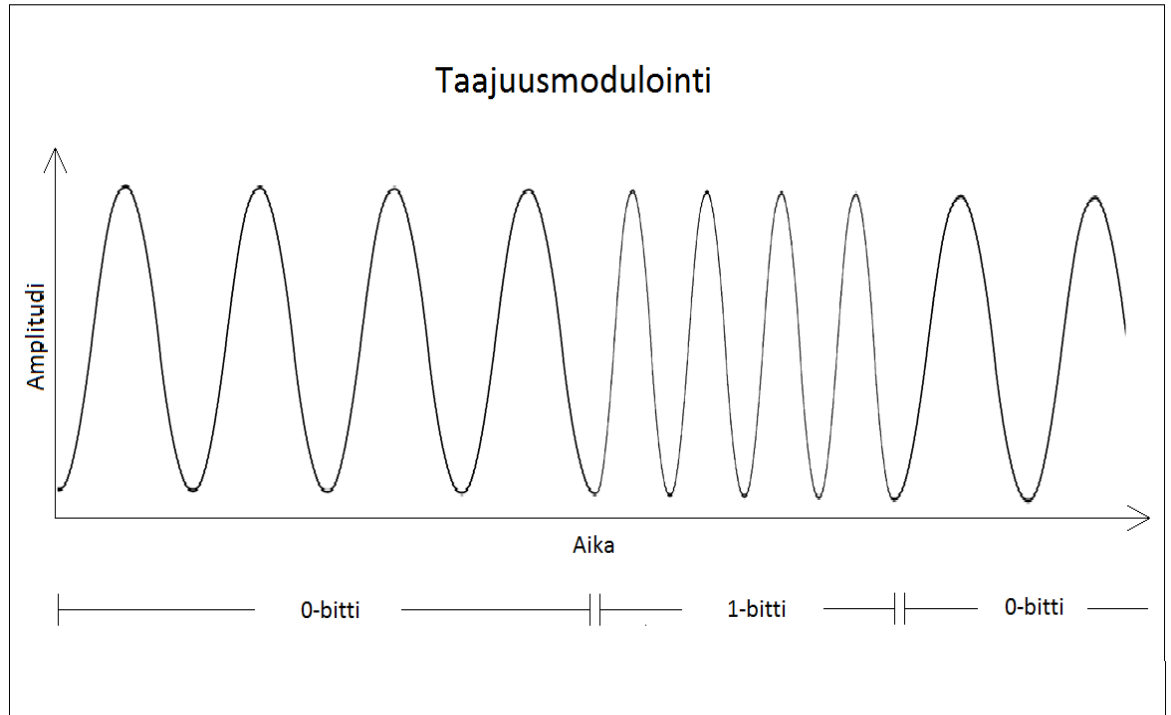
Kuvan 15 vasemmanpuoleinen verkkoratkaisu on nimeltään pikoverkko (piconet). Tämän tyyppisessä verkossa laitteita on ainoastaan kahdella ”tasolla”. Oikealla puolella on

kuvattu ns. hajaverkoksi (scatternet) kutsuttua verkkoratkaisua, jossa on taas useampi kuin kaksi tasoa. [7, s. 326.]

3.3 Taajuusalueet

Bluetooth on standardoitu toimimaan taajuusalueella 2400–2483,5 MHz. Alue kuuluu ISM-taajuusalueeseen, joka on tarkoitettu ja suojattu teollista (industrial), tieteellistä (scientific) ja lääketieteellistä (medical) käyttöä varten. Bluetooth-protokollan taajuuskaista on leveydeltään 83,5 MHz (2483,5 MHz–2400 MHz = 83,5 MHz), ja se on jaettu 1 MHz leveyksisiin kanaviin, joita on yhteensä 79 kappaletta. Bluetooth-verkon ensimmäinen kanava on taajuudella 2402 MHz. [7, s. 328.] Bluetooth-protokollan taajuuskaistan muista protokollista erottava suoja-alue on näin ollen alapäässä leveydeltään 2 MHz ja yläpäässä 3,5 MHz [8, s. 225].

Bluetoothin välityksellä siirrettävän datan 1- ja 0-bitit tunnistetaan taajuusmodulaation (FM) avulla. Taajuusmodulointi tarkoittaa sitä, että 1- ja 0-biteillä on eri taajuusalueet, joiden perusteella erottelu tapahtuu. Bluetooth-protokollassa bittien taajuusalueet on standardoitu siten, että 1-bitti havaitaan, kun kantoaallon taajuus on 140–175 kHz kanavan keskitaajuutta korkeampi. 0-bitti taas havaitaan vastaavasti 140–175 kHz kanavan keskitaajuutta matalammalta taajuusväliltä. Esimerkiksi Bluetooth-verkon ensimmäisessä kanavassa keskitaajuuden ollessa 2402 MHz on 0-bitin havaitsemisalue 2401,825–2401,860 MHz (alarajataajuus = 2402 MHz - 0,175 MHz, ylärajataajuus = 2402 MHz - 0,140 MHz) ja 1-bitin havaitsemisalue 2402,140–2401,175 MHz (alarajataajuus = 2402 MHz + 0,140 MHz, ylärajataajuus = 2402 MHz + 0,175 MHz). 1- ja 0-bittien taajuuksien väliin jää siis hieman ”harmaata aluetta” erottamaan taajuudet toisistaan. Kuvassa 16 on havainnollistettu taajuusmoduloitu signaali. [7, s. 328.]

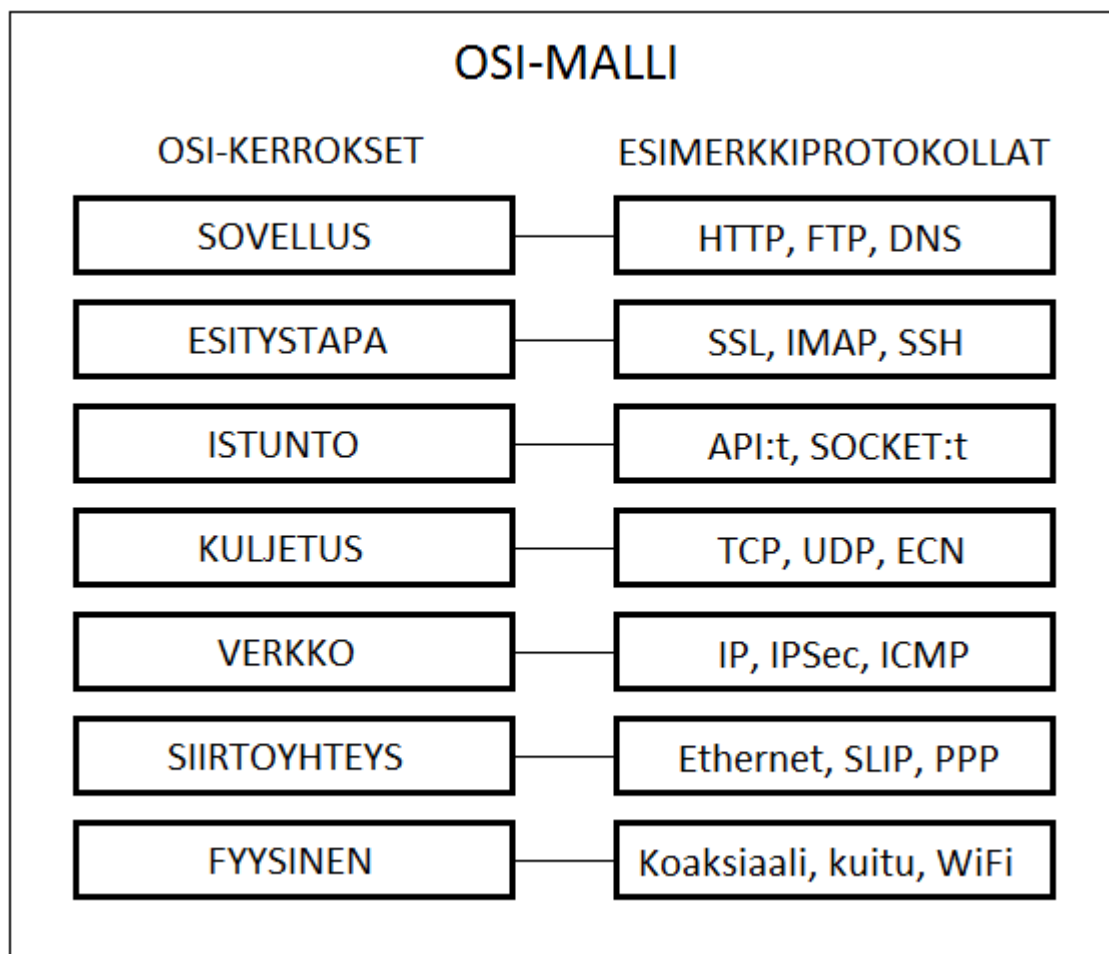


Kuva 16. Periaatekuva taajuusmoduloidusta signaalista.

Kuten kuvasta 16 voidaan todeta, bittien 1- ja 0-tason tunnistus perustuu taajuuden muutoksiin. Bluetooth-standardissa 0-bittien taajuus on määritelty matalammaksi kuin 1-bittien taajuus.

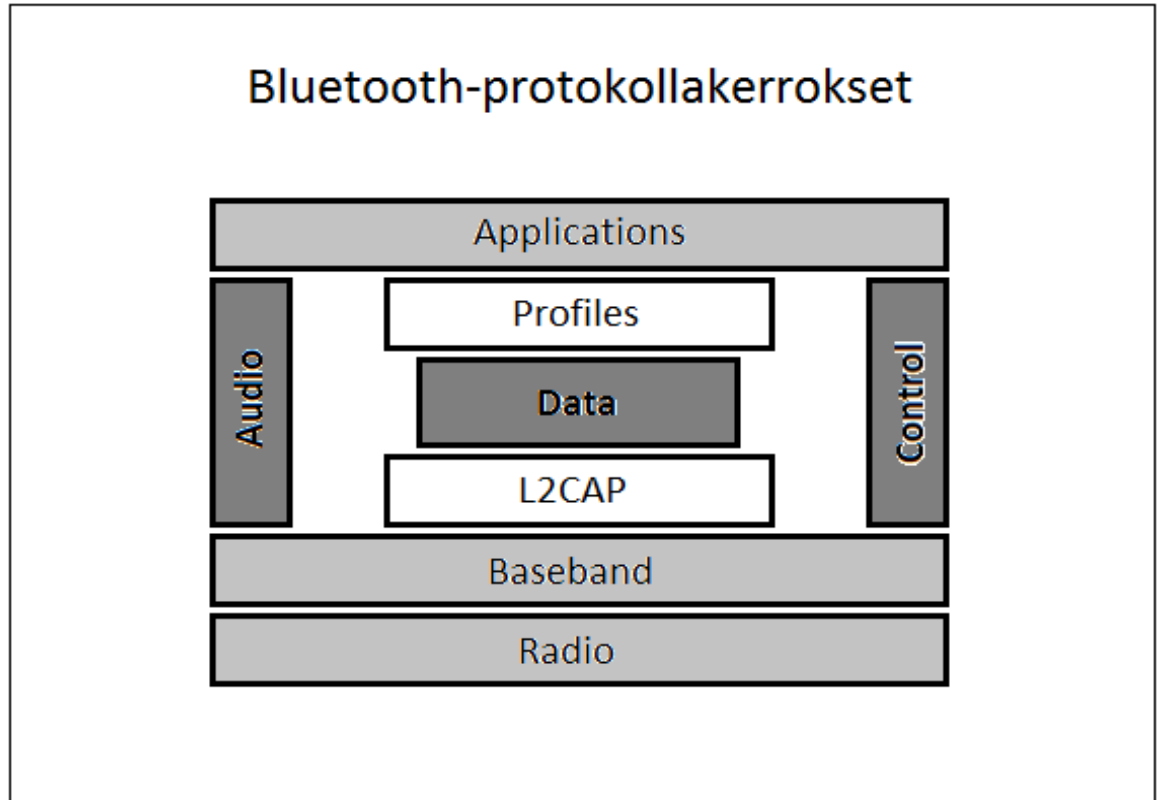
3.4 Protokollakerrokset

Bluetooth-protokollaa voidaan kuvata esimerkiksi OSI-mallin (kuva 17) tapaisen kerrostyyppisen protokollamallin avulla. Bluetooth-protokollan kerrosmalli ei kuitenkaan vastaa aivan yksi yhteen perinteistä OSI-mallia. Esimerkiksi Bluetooth-protokollan ensimmäiset kaksi kerrosta, eli radio- ja baseband-kerrokset, sisältyvät periaatteessa OSI-mallin ensimmäiseen kerrokseen [7, s. 326]. Tässä luvussa käsitellään lähinnä alimpia protokollakerroksia.



Kuva 17. Perinteinen OSI-malli ja eri kerrosten esimerkkiprotokollat [13]

Kuvassa 18 on esitetty yksinkertaistettu esimerkki Bluetooth-protokollakerroksista. OSI-mallin ja Bluetooth-protokollan mallin välillä on protokollakerroksissa nähtävissä tiettyjä yhtäläisyyksiä.



Kuva 18. Bluetooth-protokollan kerrosmalli [12, s. 497]

Radiokerros toimii protokollan ”fyysisenä kerroksena”. Tämä kerros kuvastaa siis käytännössä Bluetooth-laitteiden välistä langatonta tai langallista siirtotietä. [12, s. 501.]

Baseband- eli kantataajuuskerros huolehtii tiedon vastaanoton ja lähetyksen synkronoinnista. Bluetooth-protokollassa lähetyksen ja vastaanotto tapahtuu eri aikaan standardimittaisissa aikajaksoissa. Yhden aikajakson pituus on 625 μ s. Datan lähetystä tästä ajasta on 366 μ s ja loppuaikaa ($625 \mu\text{s} - 366 \mu\text{s} = 259 \mu\text{s}$) kutsutaan suoja-ajaksi, jolloin esimerkiksi asetetaan radiolaitteet tarvittaessa oikealle taajuudelleen [7, s. 328]. Kahden laitteen (isäntä+orja) verkossa lähetyksen tapahtuu vuoronperään, eli ensin isäntälaitte lähettää orjalaitteelle dataa 625 μ s:n ajan ja tämän jälkeen orjalaitte vastaa isäntälaitteelle. Kolmen laitteen (isäntä+orja+orja) verkossa isäntälaitte lähettää ensin ensimmäiselle orjalaitteelle, jonka jälkeen orjalaitte vastaa isännälle. Tämän jälkeen isäntälaitte suorittaa lähetyksen toiselle orjalaitteelle ja orjalaitte jälleen vastaa. Vastaavalla logiikalla toimitaan myös kolmen ja useamman orjalaitteen tapauksessa. [12, s. 498–499.]

L2CAP-kerros (Logical Link Control and Adaption Protocol) toimii datanvälityskerroksena ns. sovelluskerroksien (applications) ja baseband-kerroksen välissä. L2CAP-kerros purkaa baseband-kerrokselta vastaanotettavan tiedon ja ohjaa sen sille osoitetulle kerrokselle.

Ylemmiltä kerroksilta tuleva tieto taas pakataan ja lähetetään baseband-kerrokselle. Jos tämä ylemmiltä kerroksilta tuleva paketti ylittää baseband-kerroksen maksimipakettikoon, pilkotaan data pienempiin osiin. [12, s. 497–498.]

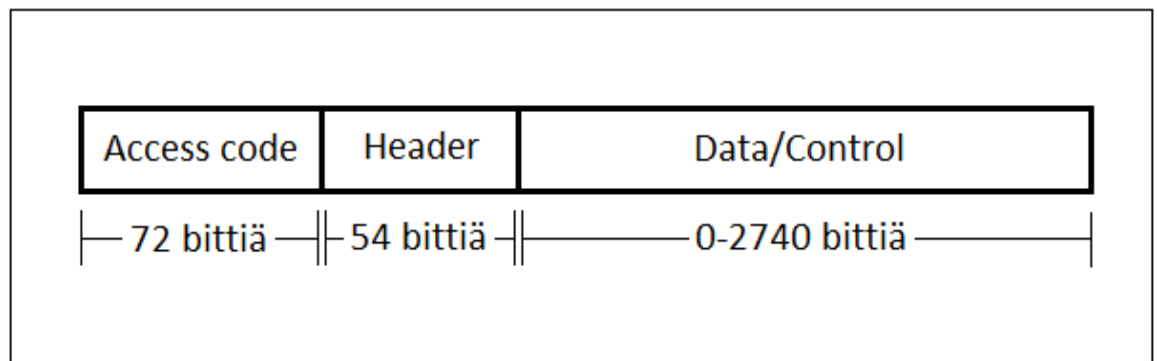
L2CAP-kerroksen yläpuolisia palveluita ovat mm. RFCOMM, joka käytännössä mallintaa sarjaporttia (RS-232), sekä SDP, joka tunnistaa ja etsii uusia palveluja [14, s. 97–98].

3.5 Datakehukset

Baseband-kerroksen datakehys voi olla yhden, kolmen tai viiden aikajakson pituinen. Tämän siirtojakson lopussa on jälleen jo aiemmin mainittu suoja-aika. Varsinaisen siirtojakson pituus lasketaan siis kaavalla 3:

$$t_{\text{siirtojakso}} = 625 \mu\text{s} * n - 259 \mu\text{s}, \quad (3)$$

jossa 625 μs kuvastaa yhden aikajakson pituutta, n kokonaislukua (1, 3 tai 5) ja 259 μs suoja-aikaa. Kuvassa 19 on esitetty baseband-kerroksen datakehys. [12, s. 500.]

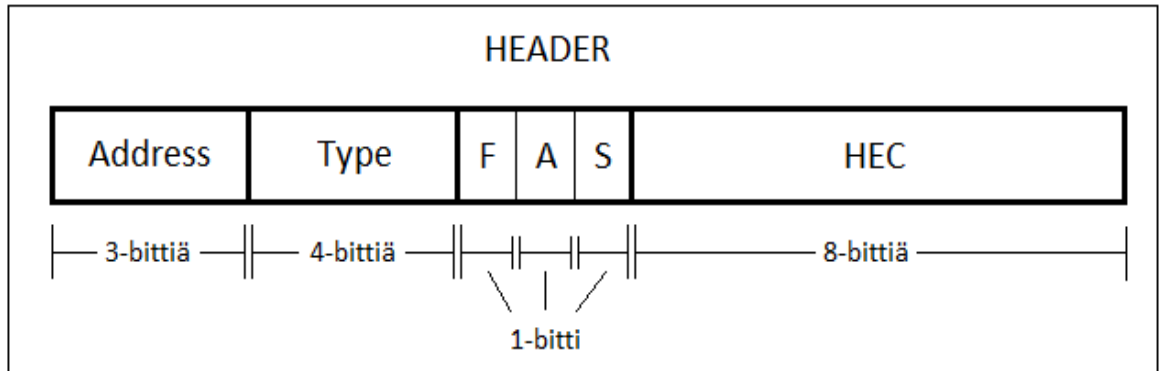


Kuva 19. Baseband-kerroksen datakehys [12, s. 500.]

Ensimmäinen segmentti on 72 bittiä pitkä Access code -niminen kenttä, joka sisältää alkutahdistus- ja synkronointibittejä sekä pikoverkkojen erottamiseen tarvittavan tunnisteiden. [12, s. 500.]

Toinen segmentti on header. Header on 18 bittiä pitkä kenttä, joka sisältää omia sisäkenttiä. Header-kenttä toistetaan kolme kertaa, jolloin lopulliseksi pituudeksi saadaan 54 bittiä. Kuvassa 20 on esitetty header-kentän sisältö. [12, s. 500.]

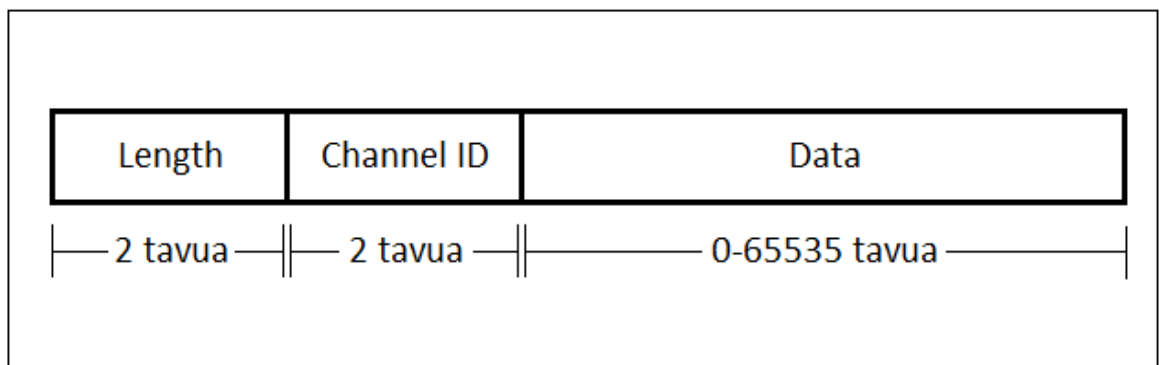
Kolmas segmentti on Data/Control-kenttä, joka sisältää hyötykuorman. Tämän kentän pituus riippuu siitä, kuinka monen aikajakson pituinen kehys on. [12, s. 501.]



Kuva 20. Baseband-kerroksen datakehysen Header-segmentti [12, s. 500]

Header-kentän ensimmäinen sisäkenttä on address eli 3-bittinen orjalaitteen osoite. Lähetettäessä dataa isäntälaitteelta orjalaitteelle ja osoitteen arvon ollessa 1–7 (001–111 B) lähetys on osoitettu yksittäiselle orjalaitteelle. Arvon ollessa 0 (000 B) on kyseessä lähetys, joka on tarkoitettu kaikille (broadcast). Arvoa 0 käytetään myös aina lähetettäessä dataa orjalaitteelta isännälle [7, s. 334]. Type-sisäkentässä on määritetty, minkätyyppistä dataa ylemmiltä kerroksilta tulee. F-kenttä on varattu flowcontrol-bitille, A acknowledgement-bitille ja S sekvenssinumerolle. HEC-kenttä on varattu tarkistussummalle. [12, s. 501.]

L2CAP-kerroksella on oma datakehyksensä. Datakehys koostuu kolmesta segmentistä, joista ensimmäisessä kerrotaan kehyksen dataosion pituus. Toisessa segmentissä ilmoitetaan virtuaalisen kanavan ID, jolle data on tarkoitettu. Kolmas segmentti sisältää kehyksen tietosion. Kuvassa 21 on esitetty L2CAP-kerroksen datakehys. [12, s. 497.]

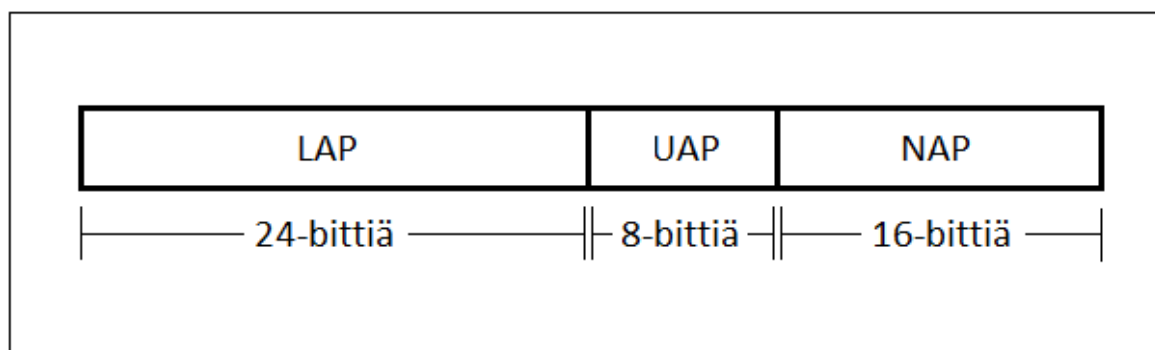


Kuva 21. L2CAP-kerroksen lähettämä datakehys [12, s. 497]

3.6 Laitesoitteet

Bluetooth-laitteet tunnistetaan yksilöllisen laiteosoitteen perusteella. Laitesoite (BD_ADDR) on kovakoodattu laitteeseen, ja se ei ole muutettavissa. Osoitteet esitetään yleensä heksadesimaalisessa muodossa, ja ne ovat pituudeltaan 48-bittisiä. Osoitteen esitysmuoto on IEEE:n kehittämä, ja se perustuu MAC-osoitteiden esitysmuotoon. [15, s. 148.]

Laitesoite koostuu 24-bittisestä LAP-osasta, 8-bittisestä UAP-osasta ja 16-bittisestä NAP-osasta. UAP- ja NAP-osiot kertovat Bluetooth-laitteen valmistajan, eli nämä muodostavat yhdessä yritys-ID:n. LAP-osio on laitteen valmistajan määrittelemä tunnistusmerkintä. Kuvassa 20 on esitetty Bluetooth-laitesoitteen muodostuminen ja osiot. [15, s. 148.]



Kuva 22. Bluetooth-laitesoite [15, s. 148]

4 KÄYTTÖLIITTYMÄSOVELLUS

Työssä käytetty mobiililaitteelle asennettava käyttöliittymäsovellus on Kajapro Oy:n kehittämä ohjelmistotuote. Sovellusta käytetään tässä projektissa tablettitietokoneessa, joka on varustettu Android 4.0.3 -käyttöjärjestelmällä. Kajaanin ammattikorkeakoulun käyttöön luovutettu sovellusversio on kyseiselle käyttöjärjestelmäversiolle testattu toimivaksi. Testaus on suoritettu projektin vaatimusmääritelmässä mainittujen ominaisuuksien toiminnallisuuden osalta. [16, s. 8.]

Sovellus on kehitetty toimimaan myös muilla Android-versioilla ja Windows Phone 8 -alustalla. Mobiiliversioiden lisäksi sovelluksesta on olemassa Windows- ja selainpohjaiset versio, joten käyttö normaalilla PC:llä on myös mahdollista. Tämän sovellusversion vaatimusmäärittelyn ominaisuuksien testausta kyseisillä alustoilla ei ole tehty. [17, s. 4.]

Käyttöliittymäsovellus tukee erilaisia käyttöliittymän peruskomponentteja. Tätä projektia varten sovellukseen on kehitetty ”joystick”-tyyppinen komponentti. Tämä komponentti oli listattu vaatimusmäärittelyyn, ja sen tehtävänä on toimia moottorinohjauksessa moottorin tehon säätimenä. [17, s. 7.]

Käyttöliittymäsovellusta käytetään tässä projektissa Bluetooth-yhteyden välityksellä. Sovellus mahdollistaa kuitenkin myös käytön IP:n ja perinteisen sarjaportin välityksellä. Käyttäjä voi valita sovelluksen asetuksista käytettäväksi kieleksi englannin tai suomen. [17, s. 4.]

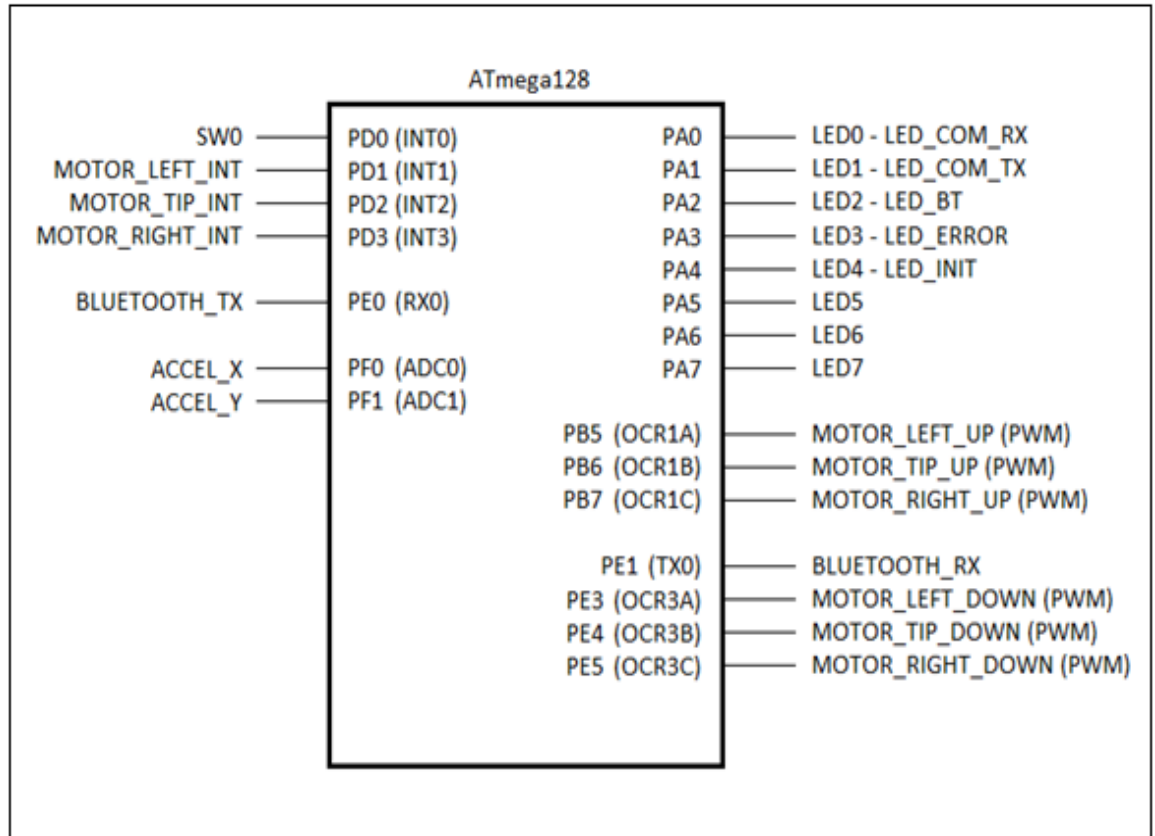
5 KEHITYSYMPÄRISTÖ

Sulautetun laitteen (Atmel Atmega 128L + STK600) ohjattavana laitteena oli tässä projektissa Kajaanin ammattikorkeakoululta saatu kolmimoottorinen dc-moottorialusta. Alustan moottorien ja sulautetun laitteen väliin on kytketty H-silta, jonka avulla saadaan moottoreille syötettävä teho riittävän suureksi liikuttamista varten. Alustassa oli lisäksi moottorien ylä- ja ala-asennon tunnistavat rajakytkimet. Rajakytkimiä voitiin käyttää aiheuttamaan sulautetulle laitteelle ulkoisen keskeytyksen, ja näin ollen pysäyttämään moottorien liikkeen. Alustan päälle oli sijoitettu kiihtyvyyssanturi, jonka x- ja y-akseleiden kallistustiedon perusteella saatiin alustan asento selville.

Mikrokontrollerin ohjelmointi tapahtui AVR Studio 4.18 -ohjelmistoa käyttäen. Ohjelmistoon on asennettu kyseisen ohjelmistoversion korjauspaketit (service pack) 1, 2 & 3. Käyttöjärjestelmänä PC:ssä toimi Windows 8 Pro 64-bit. Ohjelmointikielenä käytettiin C-kieltä, ja käytetyt ohjelmakirjastot olivat suurimmaksi osaksi AVR:n omia kirjastoja.

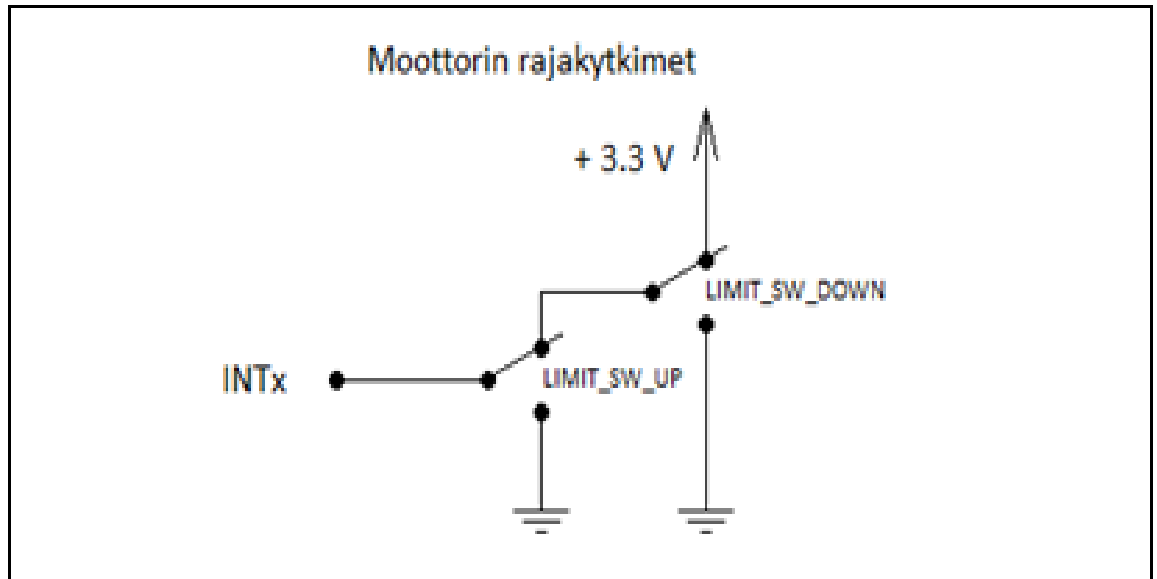
Ohjelmakoodin debuggaamisessa käytettiin AVR JTAG ICE mk2 -debuggeria (lyhyemmin JTAG ICE), joka liitetään mikrokontrolleriin JTAG-rajapintaan ja PC:ssä USB-porttiin kiinni. JTAG ICE vaatii PC:lle asennettavan laitteen omat ajurit, jotka löytyvät valmistajan (AVR) sivuilta.

Kuvassa 23 on esitetty työssä mikrokontrolleriin liitettävien laitteiden kytkentäkaavio.



Kuva 23. Mikrokontrollerin kytkentäkaavio.

Portti A on varattu ledeille, joista kolme on käyttämättömiä. Ledit informoivat bluetooth-moduulin toiminnasta, sarjaportin lähetyksestä ja vastaanotosta, virhetilanteista ja mikrokontrollerin alustusten valmistumisesta. Kiihtyvyyssanturi on kytketty AD-muuntimelle kahteen tulopinniin PF0 ja PF1. Moottorit on kytketty PWM-pinneihin, jotka sijaitsevat porteissa B ja E. Bluetooth-moduulin reset-kytkin (SW0) on liitetty keskeytykseen INT0. Kytkin on nähtävissä kuvassa 25. Keskeytykset INT1–INT3 on liitetty moottoreiden rajakytkimiin. Yhtä moottoria kohden on kaksi rajakytkintä eli ylä- ja alarajakytkimet. Rajakytkinten kytkentäkaavio on nähtävissä kuvassa 24.



Kuva 24. Moottorien rajakytkimien kytkentäkaavio. Jos kumpi tahansa kytkin aktivoituu, aiheutuu tällöin keskeytys.

6 TYÖN SUORITUS

Työn suoritus aloitettiin tekemällä projektista vaatimusmäärittely. Vaatimusmäärittelyyn kirjattiin ominaisuudet, joita projektin haluttiin antavan projektin tilaajalle eli Kajaanin ammattikorkeakoululle. Kaikki dokumenttiin kirjatut yksittäiset vaatimukset nojasivat siihen lopulliseen päämäärään, jossa mobiililaitteen näytöltä voidaan ohjata sulautettua laitetta ja sulautettu laite voi toisaalta myös lähettää tietoa mobiililaitteelle. Vaatimusmäärittelyn laatimisen jälkeen se lähetettiin tilauksen vastuuhenkilölle hyväksyttäväksi. Tämän jälkeen työn suorittamisesta laadittiin projektisuunnitelma, johon kirjattiin aikataulu, jonka mukaan projekti suoritettaisiin. Koko projektin toteutus aloitettiin kesäkuun 2014 alussa, ja projektin oli tarkoitus valmistua elokuun 2014 loppuun mennessä.

6.1 Kehitysympäristön asennukset

Ennen varsinaisen ohjelmointityön aloitusta täytyi asentaa kehitysympäristön vaatimat ohjelmat ja ajurit. Asennuksiin ei liittynyt isompia ongelmia muuten, kuin debuggerina käytetyn JTAG ICE:n ajureiden asennus. Ongelma johtui siitä, kun JTAG ICE:n ajurit eivät ole digitaalisesti allekirjoitettuja ja Windows 8 ei anna niitä asentaa sellaisenaan. Näin ollen käyttöjärjestelmän asetuksista täytyy tämä ehto käydä kumoamassa. Asetus vaihdetaan seuraavasti:

- Vie hiiri näytön oikeaan alakulmaan ja valitse asetukset.
- Valitse ”Muuta tietokoneen asetuksia”.
- Valitse ”Yleiset” -> ”Käynnistyksen lisäasetukset” ja ”Käynnistä uudelleen nyt”.
- Valitse ”Vianmääritys” (jos ”Vianmääritys”-painiketta ei ole, tällöin valitse ”Jatka” → ”Vianmääritys”).
- Valitse ”Lisäasetukset”, ”Käynnistyksen asetukset” ja ”Käynnistä uudelleen”.
- Valitse funktionäppäimellä (F1-F9) valikosta ohjaimen allekirjoituksen tarkistus pois käytöstä.

Asetuksen vaihdon jälkeen ajurit voidaan asentaa normaalisti. Asennuksen aikana käyttöjärjestelmä kysyy vielä varmistusta kolmannen osapuolen valmistamien ajureiden asentamiseksi, jonka jälkeen ajurit asentuvat tietokoneelle.

6.2 Sarjaliikenteen käyttöönotto

Sulautetun laitteen ja mobiililaitteen käyttöliittymäsovelluksen välinen tietoliikenne päätettiin toteuttaa sarjamuotoisena ja langattomasti Bluetoothin välityksellä. Sarjaportin käyttöönotto suoritettiin alustamalla ohjelmallisesti portin asetukset ja kirjoittamalla lähetys- ja vastaanottofunktiot ohjelmakoodiin. Projektissa haluttiin edetä askel kerrallaan, jotta mahdolliset virheet ja ongelmat voidaan havaita hyvissä ajoin. Tästä johtuen ennen Bluetooth-yhteyden muodostamista sarjaliikenne haluttiin saada toimimaan RS-232-kaapelin välityksellä.

Tässä vaiheessa työtä yhteyden testaamiseen käytettiin PC:tä, jossa oli RS-232-liitäntä. Sarjaliikennekaapeli liitettiin PC:n ja sulautetun laitteen välille ja yhteyttä testattiin käyttäen PC:llä Hyper Terminal -ohjelmaa. Yhteyden testaus suoritettiin lähettämällä ja vastaanottamalla yksittäisiä merkkejä ja merkkijonoja sulautetun laitteen ja PC:n välillä. Samalla varmistettiin, että data tulee myös samassa muodossa perille, kuin missä se oli lähetetty.

6.3 Ohjausviestien käsittely

Sarjaliikenteen ohjelmoimisen jälkeen ryhdyttiin tekemään viestinkäsittelijää käyttöliittymäohjelmiston ja sulautetun laitteen väliselle tietoliikenteelle. Tietoliikenneprotokolla määräytyi käyttöliittymäohjelmiston käyttämän protokollan mukaan. Viestinkäsittelijä ohjelmoitiin käyttämään samoja protokollan mukaisia käskyjä ja merkkejä niin viestin vastaanotossa kuin lähetyksessäänkin.

Viestinkäsittelijän toimintaa testattiin jälleen samalla testuslaitteistolla kuin sarjaportin toimintaa testatessa. Erona edelliseen järjestelyyn oli PC:llä käytettävä testausohjelma, joka tässä tapauksessa oli Kajapron käyttöliittymäohjelmiston Windows-pohjainen versio.

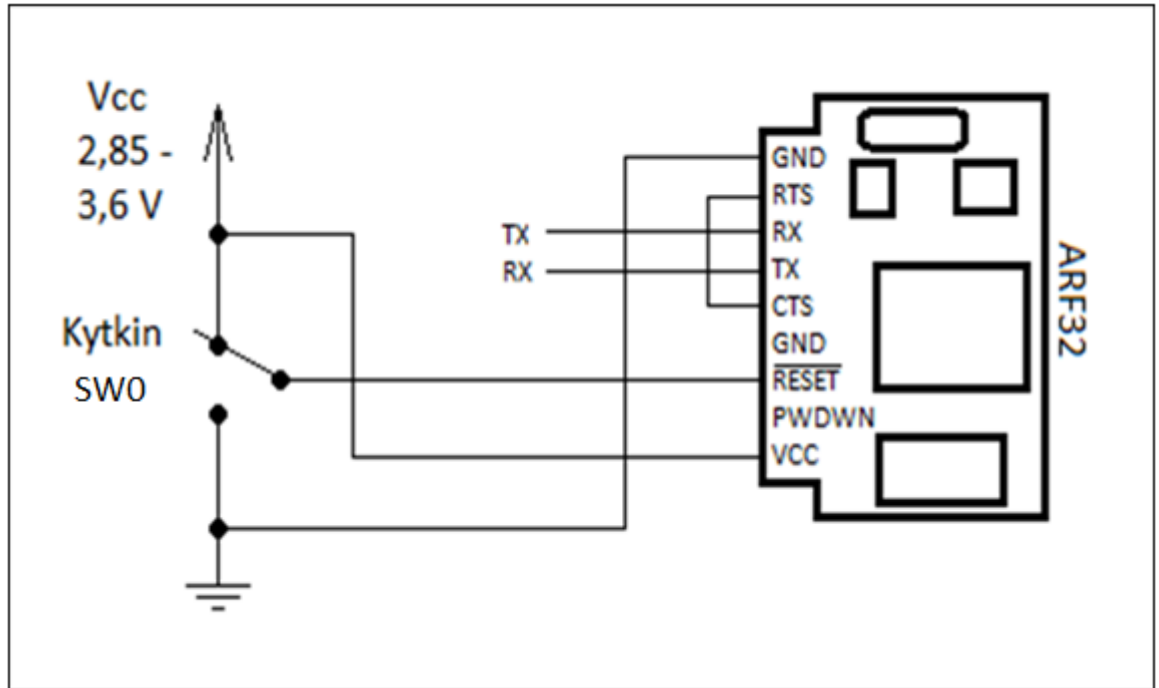
Sulautetun laitteen ja käyttöliittymäohjelmiston välinen viestiliikenne testattiin ja todettiin molempiin suuntiin toimivaksi.

6.4 Bluetooth-yhteyden muodostaminen

Tässä työssä sulautettuun laitteeseen liitettävänä Bluetooth-moduulina käytettiin Adeunis RF:n ARF32-moduulia. Moduuli valittiin sen suhteellisen edullisen hinnan ja riittävien ominaisuuksiensa vuoksi. Moduulin tilaus hoidettiin Kajaanin ammattikorkeakoulun kautta.

6.4.1 Bluetooth-moduulin kytkentä

Bluetooth-moduulin lähetyspinni (TX) kytketään host-laitteen (esim. sulautettu laite) RX-pinniin ja vastaanottopinni (RX) host-laitteen TX-pinniin. Esimerkiksi Atmel Atmega128L:lla nämä pinnit ovat PE0 ja PE1 (RX0 & TX0 UART0:lla) tai PD2 ja PD3 (RX1 & TX1 UART1:lla). RTS- (Request to send) ja CTS- (Clear to send) pinnit kytketään yhteen, koska ne ovat tarpeellisia ainoastaan *flow-control*-tilassa. Moduulin reset-pinni on 0-aktiivinen. Reset aktivoidaan, jos esimerkiksi toiset Bluetooth-laitteet eivät saa yhteyttä moduuliin. Tämän vuoksi on kätevää liittää kytkin reset-pinnin, maan ja käyttöjännitteen välille. Kuvassa 25 on esitetty edellä mainitut kytkennät.



Kuva 25. Bluetooth-moduulin kytkentäkaavio [18, s. 6]

Moduulin käyttöjännitealue (V_{CC}) on 2,85–3,6 V. Looginen 1-tila on välillä $0,7 * V_{CC} - V_{CC} + 0,5 \text{ V}$ ja 0-tila välillä $-0,5 \text{ V} - 0,3 * V_{CC}$. Esimerkiksi 3,3 V käyttöjännitteellä 0-tila aktivoituu välillä $-0,5 \text{ V} - 0,99 \text{ V}$ ($0,3 * 3,3 \text{ V}$) ja 1-tila välillä $2,31 \text{ V}$ ($0,7 * 3,3 \text{ V}$) – $3,8 \text{ V}$ ($3,3 \text{ V} + 0,5 \text{ V}$). [18, s. 7.]

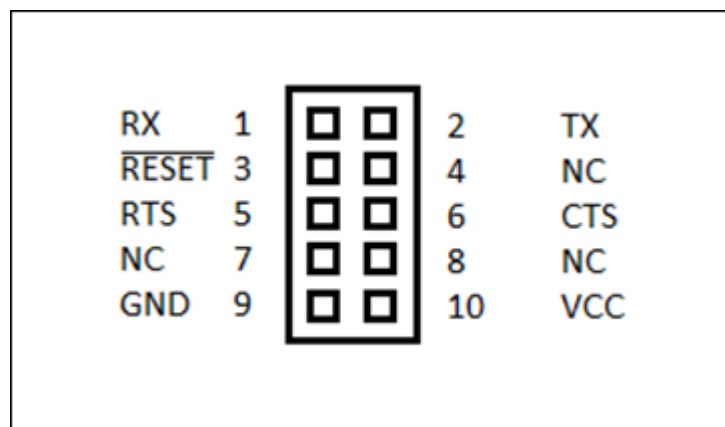
Huomionarvoisaa on, että moduulia ei tule missään tapauksessa yrittää kytkeä suoraan sarjaporttiliittimen (RS-232) pinneihin. RS-232-liittimelle tulevat jännitetasot ovat suuruudeltaan $\pm 12 \text{ V}$, ja tämä voi rikkoa moduulin.

Moduulissa ei ollut valmiina mitään liittimiä. Liittimellinen malli olisi ollut myös saatavilla, mutta sen hinta oli liian korkea. Tämän johdosta moduulin pinnit jouduttiin juottamaan itse kiinni haluttuun liittimeen kuvassa 26 esitetyllä tavalla. Kotelointi suunniteltiin ja toteutettiin myös itse.



Kuva 26. ARF32-moduuli on juotettu kiinni liittimeen ja asennettu suojakoteloon.

Tässä työssä moduulin liittimestä tehtiin kuvan 27 kaltainen. Syy tähän on se, että Atmegalta käyttöön valittiin sarjaportiksi UART0, jonka lähetys- ja vastaanottopinnit sijaitsevat STK600-alustalla E-portin (PORTE) 1- ja 2-pinnissä. Lisäksi maa- ja käyttöjännitepinnit sijaitsevat joka portissa pinneissä 9 ja 10.



Kuva 27. Esimerkki moduulin liittimestä.

Alla vielä kuvassa 28 on esitelty valmis Bluetooth-moduuli.



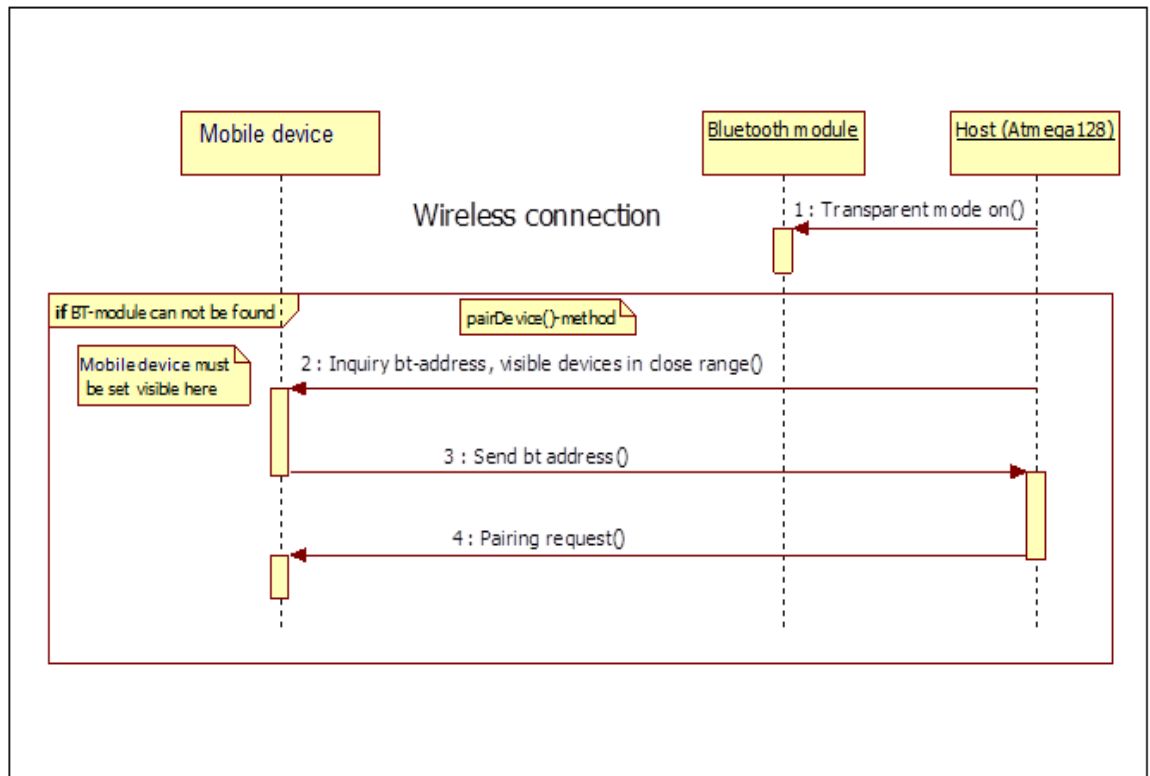
Kuva 28. Valmis liittimellinen Bluetooth-moduuli

6.4.2 Yhdistäminen

Kun moduulin käyttöjännite ja maa on kytketty, se näkyy muille Bluetooth-laitteille suoraan. Moduuli löydetään mobiililaitteen Bluetooth-laitteiden etsinnällä, ja se täytyy parittaa. Parittamisen yhteydessä pyydetään PIN-koodia (vakiona 0000). Bluetooth-moduuli on lisäksi asetettava datan lähetyksen ja vastaanoton sallivaan tilaan syöttämällä sille ”Enter transparent mode” -käsky (ks. luku 6.3.4, Komennot). Jos moduuli ei kuitenkaan näy toisessa Bluetooth-laitteessa, täytyy yhteyspyyntö muodostaa moduulilta päin. [18, s. 17.]

Moduuli tarvitsee yhteyspyynnön lähetykseen vastaanottavan laitteen Bluetooth-osoitteen (BD_ADDR). Osoite saadaan selville, kun asetetaan vastaanottava laite näkyvään tilaan ja lähetetään moduulilla ympäristöön GIAC-pyyntö (General Inquiry Access Code), jolloin kaikki näkyvät laitteet lähettävät moduulille osoitteensa. Osoite otetaan talteen ja lähetetään moduulilla ”SPP connect request” -yhdistämisspyyntö osoitteeseen. Vastaanottavaan

laitteeseen tulee näkyville ilmoitus yhteyspyynnöstä ja PIN-koodikysely. PIN-koodi syötetään kenttään, ja tämän jälkeen laitteet ovat paritettuja. Kuvassa 29 on sekvenssikaavio Bluetooth-moduulin ja -laitteen välisestä yhdistämisestä [18, s. 17.]



Kuva 29. Sekvenssikaavio Bluetooth-yhteyden muodostamisesta mobiililaitteelle

6.4.3 Moduulin alustaminen C-kielellä

Jos vastaanottava Bluetooth-laite näkee moduulin, tarvitsee moduuli asettaa ainoastaan datan vastaanottamisen ja lähettämisen sallivaan tilaan lähettämällä sille ”Enter transparent mode” -käsky. Seuraavassa koodiesimerkissä on esitetty, kuinka tämä toteutetaan C-kielellä.

```

char bt_transp_on[] = {0x02, 0x52, 0x11, 0x01, 0x00, 0x64, 0x01, 0x03}; // "Enter
transparent mode"

for(uint8_t i = 0; i < sizeof(bt_transp_on); i++)
{
    putchar(bt_transp_on[i]);
}
  
```

Käskyn lähettämisessä pitää käyttää metodia, joka ei terminoi lähetystä lähetettäessä 0x00-merkki (NULL). Tähän käy esimerkiksi putchar-komento. Bt_transp_on-taulukko sisältää merkkijonomuotoisena aiemmin mainitun ”Enter transparent mode” -käskyn. Moduuli vastaa tähän komentoon lähettämällä merkkijonon ”0x02 0x43 0x11 0x02 0x00 0x56 0x00 0x01 0x03”. Tämä on hyvä varmistaa isäntälaitteen vastaanottopuskurista, jotta tiedetään moduulin toimineen halutulla tavalla.

Jos vastaanottava Bluetooth-laite ei näe moduulia, ”Enter transparent mode” -käskyn jälkeen lähetetään osoitteen kysely (GIAC) vastaanottavalle laitteelle.

```
char bt_giac[] = {0x02, 0x52, 0x00, 0x03, 0x00, 0x55, 0x0A, 0x00, 0x00, 0x03}; //
“General Inquiry Access Code”
```

```
for(uint8_t i = 0; i < sizeof(bt_giac); i++)
{
    putchar(bt_giac[i]);
}
_delay_ms(2000);
```

Kyselyn lähettämisen jälkeen odotetaan vastausta mobiililaitteelta. Viiveen on hyvä olla vähintään kahden sekunnin mittainen, mutta suositeltavaa on käyttää jopa neljän sekunnin mittaista viivettä. Pitempi viive vastauksessa voi syntyä, jos esimerkiksi laitteiden välinen kantama on pidempi ja yhteyden laatu näin ollen heikompi. Tämä asia selvitettiin digitaalisen muistioskilloskoopin avulla.

Laite vastaa osoitekyselyyn merkkijonolla ”0x02 0x69 0x01 0x09 0x00 0x73 0x34 0xBE 0x1F 0x17 0x00 0x08 0x00 0x00 0x00 0x03”, jossa tässä tapauksessa laitteen osoite on ”0x34 0xBE 0x1F 0x17 0x00 0x08”, eli osoite on heti merkin 0x73 jälkeiset kuusi merkkiä.

Osoitteella korvataan ”SPP connect request” -komennon 0xFF-merkit ja lähetetään tämän jälkeen komento vastaanottavalle laitteelle. Osoite sijoitetaan lähetettävään komentoon alla olevan esimerkin mukaisesti.

```
char bt_pair_spp[] =
{0x02, 0x52, 0x0A, 0x08, 0x00, 0x64, 0x01, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x01,
0x03}; // “SPP connect request”
```

6.4.4 Komennot

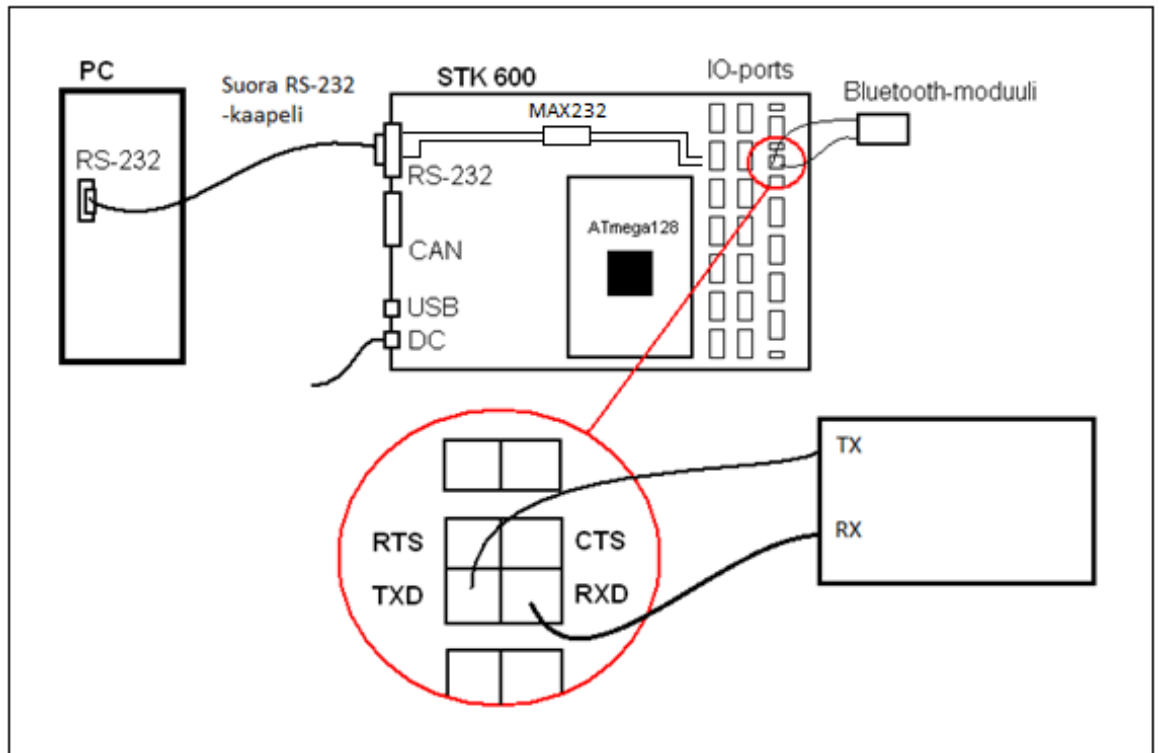
Bluetooth-moduulin asetuksia konfiguroidaan syöttämällä moduulin datan vastaanottopinniin (RX) merkkijonoja esimerkiksi C-kielisesti. Taulukossa 5 on esitetty moduulin asetusten tärkeimmät komennot.

Taulukko 5. Bluetooth-moduulin peruskomennot [18, s. 17–20]

	Komento
Osoitteen kysely, "GIAC"	0x02 0x52 0x00 0x03 0x00 0x55 0x0A 0x00 0x00 0x03
"Transparent"-tila (data liikkuu)	0x02 0x52 0x11 0x01 0x00 0x64 0x01 0x03
SPP-yhdistämispyyntö (0xFF-merkit korvataan vastaanottavan laitteen osoitteella)	0x02 0x52 0x0A 0x08 0x00 0x64 0x01 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x01 0x03
Tehdasasetusten palautus	0x02 0x52 0x1A 0x00 0x00 0x6C 0x03

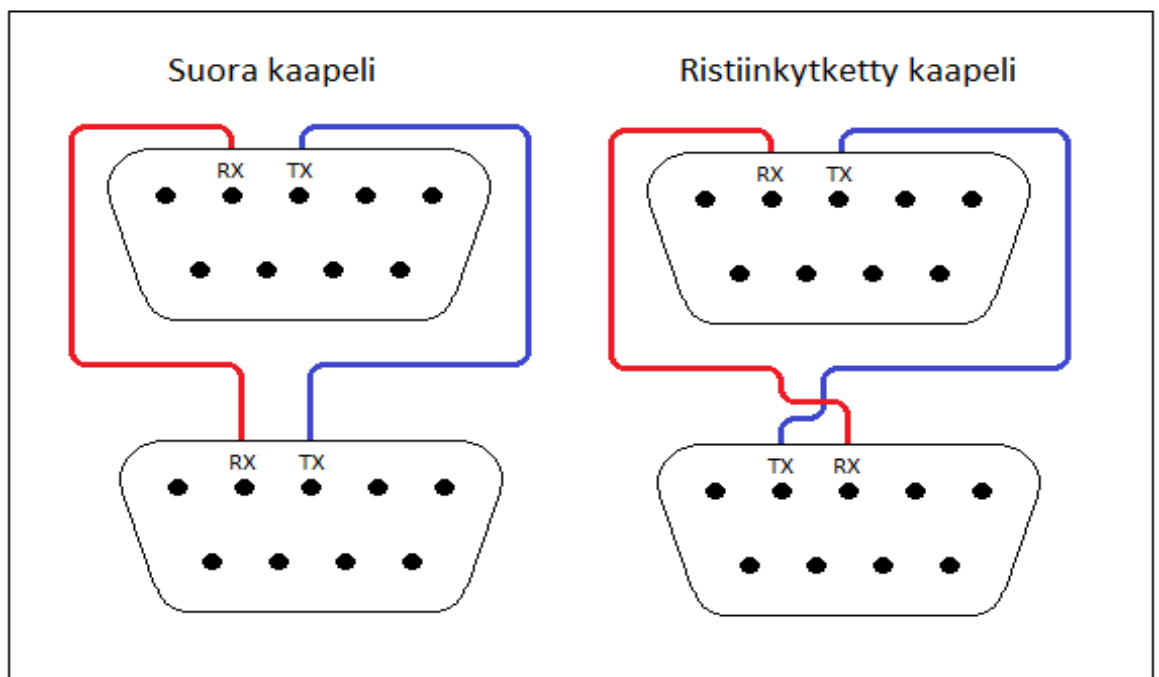
6.4.5 Simply Blue Commander

Simply Blue Commander on ohjelmisto, jolla voidaan lähettää merkkijonomuotoisia Bluetooth-käskyjä esimerkiksi tässä työssä käytetylle ARF32-moduulin Bluetooth-piirille, LMX9830:lle. Ohjelmalla voidaan kätevästi vaihtaa esimerkiksi moduulin baudinopeuksia ja muita asetuksia. Ohjelma vaatii PC:ltä sarjaliikenneportin (RS-232). Kuvassa 29 on esitetty esimerkikykentä ohjelman käyttöä varten. Esimerkissä käytössä on Atmel Atmega128L-suorittimella oleva RS-232-liittimellinen STK600-alusta.



Kuva 29. Simply Blue Commanderille tehty kytkentä.

PC kytketään kuvan 29 mukaisesti suoralla sarjaporttikaapelilla STK600-alustaan kiinni. Suoralla sarjaporttikaapelilla tarkoitetaan sitä, että kaapelin lähetys- ja vastaanottopinnit liittimien molemmissa päissä ovat samalla kohdalla. Jos kohdat ovat päinvastaisesti toisessa liittimessä, on kyseessä ristiinkytketty kaapeli (kuva 30).



Kuva 30. Sarjaporttikaapeleiden rakenteelliset erot

Koska sarjaportin jännitetasot eivät sovellu suoraan moduulille, on ne muunnettava kuvan 29 kytkennän avulla. STK600:lla on sarjaporttiliittimen sekä TXD- ja RXD-pinnien välissä MAX232-piiri, joka hoitaa tämän jännitetason muunnoksen.

ARF32-moduuli käyttää oletusarvoisesti baudinopeutena 9600 Bd. Pariteettibittiä ei ole käytössä, ja stop-bittejä on yksi kappale. Tehdasasetusten palautus -komennolla saa nämä arvot jälleen aktiivisiksi. Tehdasasetusten palautus on tarpeen, jos esimerkiksi baudinopeuden nosto epäonnistuu ja moduuli lakkaa vastaamasta mihinkään käskyyn. Komennot voidaan syöttää joko ohjelmakoodin kautta tai vaihtoehtoisesti Hyperterminal-tai Simply Blue Commander -ohjelmaa käyttämällä. Moduuli vastaa aina oikeassa muodossa syötettyihin komentoihin. Moduulin ja tämän isäntälaitteen (eli laitteen, jossa moduuli on kiinni) välisessä viestiliikenteessä 0x02 on aloitusmerkki ja 0x03 lopetusmerkki.

6.5 Mikrokontrollerin muiden ominaisuuksien käyttöönotto

Mikrokontrollerilla otettiin käyttöön edellä toteutetun Bluetooth-yhteyden lisäksi erilaisia ominaisuuksia. Tämä tehtiin sen vuoksi, jotta saadaan konkreettista referenssiä siihen, kuinka käyttöliittymää ja Bluetooth-yhteyttä voidaan hyödyntää.

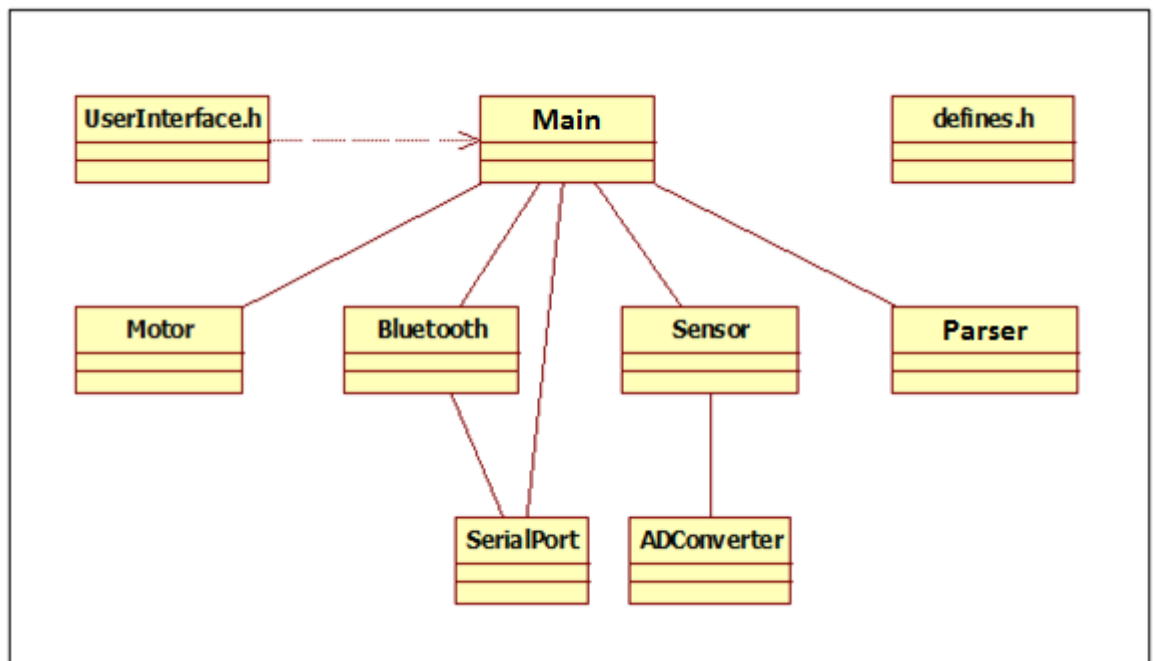
Anturidatan lähetyksessä täytyi ottaa käyttöön AD-muunnin, joka mittaa anturilta tulevan jännitteen ja muuntaa sen digitaaliseksi signaaliksi. Uusien analogisten antureiden liitäntä sulautettuun laitteeseen pyrittiin tekemään mahdollisimman helpoksi ohjelmakoodissa. Ajastinta tarvittiin AD-muuntimen lukemiseen anturidatan lähettämiseen käyttöliittymälle tietyin väliajoin. Toinen käyttötarkoitus ajastimelle oli moottorinohjaus PWM-tekniikkaa hyödyntäen. Ulkoisia keskeytyksiä tarvittiin moottorin ylä- ja alarajakytkinten antamien ”rajahälytyksien” vuoksi. Myös pienimuotoista automaattista ohjausta pyrittiin saamaan aikaiseksi. Tässä käytettiin hyödyksi kiihtyvyyssanturin antamaa kulmatietoa. Jos esimerkiksi anturin lukema poikkesi vaakatasoksi määritellyn toleranssin rajoista, ryhdyttiin moottorilla oikaisemaan alustaa jälleen tasaiseksi. Myös automaattisia ohjauksia pystyttiin kytkemään päälle ja pois päältä mobiililaitteelta käsin. Edellä mainittujen ominaisuuksien lisäksi mikrokontrollerilla otettiin käyttöön erilaisia virransäästöominaisuuksia.

6.6 Esimerkkikäyttöliittymän luonti

Työn lopuksi sulautetun laitteen ohjelmakoodiin sisällytettiin merkkijonopohjainen esimerkkimerkkikäyttöliittymä, johon koottiin painikkeita ja ohjaimia, jotka hyödynsivät sulautetun laitteen käyttämiä laitteita ja ominaisuuksia. Käyttöliittymään tehtiin useampi ”ikkuna” ja paneeli, jonka kautta ikkunoiden välillä voitiin navigoida. Käyttöliittymään sisällytettiin myös graafinen pürtoikkuna, johon pystyttiin piirtämään käyrää anturidatan mukaisesti. Käyttöliittymästä tehtiin erikseen ”kännykkäversio” ja ”tablettiversio”, jotta käytettävyys pieni- ja isonäyttöisellä laitteella ei merkittävästi muuttuisi objektien koon vaihtelun vuoksi.

7 TULOKSET

Työn tuloksena saatiin ohjelma, jonka tärkein tehtävä on hoitaa sulautetun laitteen ja mobiililaitteen välinen kommunikointi. Tämän lisäksi ohjelma sisältää sulautettuun laitteeseen liitettyjen laitteiden kontrolloinnin. Ohjelman rakenne ja siihen kuuluvat osiot ovat nähtävissä kuvassa 31.



Kuva 31. Työn tuloksena tehdyn ohjelman rakenne.

Main-osio on ohjelman pääosio. Osio sisältää while-loopin, jossa ohjelman suoritukset tehdään. Loopista poistutaan käyttäjän käyttöliittymäohjelmiston kautta aiheuttamien tai muuten aiheutuneiden keskeytysten avulla. Yksittäisen moottorin ohjaukseen liittyvät toiminnallisuudet sijaitsevat Motor-osioissa. Bluetooth-laitteen asetus- ja yhdistysfunktiot sijaitsevat Bluetooth-osiossa. Itse tiedonsiirto sulautetun laitteen ja käyttöliittymäohjelmiston välillä on hoidettu Serialport-osiossa. Sarjaportin kautta vastaanotettujen käyttöliittymäohjelmistolta saatujen viestien parsinta hoidetaan Parser-osiossa. Adconverter-osiossa alustetaan AD-muunnin ja valitaan aktiivisena oleva anturi. Sensor-osiossa muunnetaan anturin mittaustiedot haluttuun muotoon. Näiden lisäksi ovat omat osiot käyttöliittymäasettelulle (Userinterface.h) ja ohjelman käyttämille määrittelyille (defines.h). Käyttöliittymäasettelun osioon käyttäjä pystyy luomaan omaan projektiinsa soveltuvan käyttöliittymän merkkijonomuodossa.

Työssä tehdyn ohjelman lisäksi tulokseksi voidaan laskea työn yhteydessä tilattujen Bluetooth-moduulien toimintakuntoon laitto. Kyseisiä moduuleja on näin ollen mahdollista käyttää myös eri projekteissa esimerkiksi ”langattomana sarjaporttina”. Työn ohessa kirjoitettiin myös Bluetooth-moduulien ja mobiililaitteen ohjelmiston käyttöönottoon liittyvät ohjeet.

8 TULOKSIEN ANALYSOINTI

Työn suoritus onnistui hyvin, sillä vaatimusmäärittelyssä asetetut vaatimukset saatiin projektissa toteutettua. Projektin alussa laaditun projektisuunnitelman aikataulussa onnistuttiin pysymään. Työn vaiheet saatiin suoritettua myös kohtuullisen tarkasti suunnitelmaan kirjatussa järjestyksessä.

Projektin tuloksena tehdyn ohjelman avulla voidaan Kajapron käyttöliittymäohjelmiston sisältävällä mobiililaitteella, esimerkiksi tabletilla, muodostaa Bluetooth-yhteys sulautettuun laitteeseen. Yhteyden muodostamisen jälkeen mobiililaitteella voidaan ohjata esimerkiksi sulautettuun laitteeseen liitettyjä moottoreita ja saada näytölle anturitietoja. Käyttöliittymäsovelluksen asennuspaketti on toimitettu Kajaanin ammattikorkeakoululle, joten se on mahdollista asentaa projektissa käytetyn tabletin lisäksi myös muihin koulun mobiililaitteisiin.

Ohjelmakoodi on pyritty pilkkomaan osioihin, jotta koodin suoritus olisi selkeästi nähtävillä. Esimerkiksi anturitiedon käsittely on laitettu omaan osioon ja käyttöliittymäohjelmiston viestien parsiminen niin ikään omaan osioon. Ohjelmakoodiin on tehty myös esimerkkejä käyttöliittymäsovelluksen käyttöön liittyen. Eri komponenttien käytöstä on tehty esimerkkitapaukset, kuinka komponentteja ohjelmallisesti käsitellään ja otetaan käyttöön. Tämän lisäksi ohjelmakoodiin on tehty sulautetun laitteen ohjelman kehitystä helpottavia funktioita. Työstä tehdyistä dokumenteista ilmenevät myös tarkemmin käyttöliittymäsovelluksen käskymuodot ja koko tietoliikenneprotokolla.

Ohjelmallisessa toteutuksessa on pyritty ottamaan mahdollisimman laajasti sulautetun laitteen eri ominaisuuksia käyttöön. Työ on suoritettu kuitenkin tietyn aikarajan puitteissa, ja tästä johtuen kaikkia sulautetun laitteen ominaisuuksia ei keretty saamaan toimintaan. Yksi tärkeimmistä ominaisuuksista, joita ohjelmakoodiin olisi tulevaisuudessa hyvä ottaa käyttöön, on toisen tiedonsiirtoväylän, esimerkiksi SPI:n käyttöönotto. Väylän käyttöönotto mahdollistaisi digitaalisten antureiden käytön.

9 YHTEENVETO

Tässä projektissa oli tarkoitus saada Kajapro Oy:n valmistama käyttöliittymäsovellus toimimaan sulautetun laitteen ohjaimena. Sovellusta käytetään mobiililaitteella, joka tässä projektissa oli tabletti. Sulautetussa laitteessa oli esimerkin vuoksi kiinni kolmimoottorinen alusta ja kiihtyvyyssanturi. Tällä kokoonpanolla voitiin demonstroida tietoliikennettä molempiin suuntiin; moottoria voidaan ohjata tabletilta ja anturi lähettää kulmatietoa tabletille.

Työn tekemisessä ei ylitsepääsemättömiä vaikeuksia ollut. Suurimmat ongelmat olivat Bluetooth-yhteyden muodostamisessa. Kajaanin ammattikorkeakoululta alun perin saatu Bluetooth-moduuli ei ollut tähän työhön soveltuva. Tämän vuoksi työtä varten täytyi tilata siihen soveltuva Bluetooth-moduuli. Moduulin tilaus hoidettiin Kajaanin ammattikorkeakoulun kautta ja moduuleja tilattiin lopulta kaksi kappaletta. Tilattu moduuli ei aluksi halunnut reagoida mihinkään, mutta digitaalisen oskilloskoopin avulla laitteen toimintalogiikka saatiin selville. Bluetooth-yhteyden muodostamisen jälkeen työn suorituksessa ei tullut enää pidemmän miettimisen vaatimia ongelmia eteen.

Työn lopputulokseksi saatu ohjelma voi toimia tulevaisuudessa Kajaanin ammattikorkeakoululla projekteissa, joissa tarvitaan langatonta käyttöliittymäratkaisua sulautetun laitteen ohjaamiseen. Käyttöliittymäsovelluksen käytöstä ja sen toiminnasta on myös toimitettu Kajaanin ammattikorkeakoululle dokumentit.

LÄHTEET

1. 8-bit Atmel Microcontroller with 128 Kbytes In-System Programmable Flash, Atmega128/Atmega128L, [PDF-dokumentti]
<<http://www.atmel.com/Images/doc2467.pdf>> (Päivitetty 6/2011)
2. What is RISC?, [WWW-dokumentti]
<<http://www-cs-faculty.stanford.edu/~eroberts/courses/soco/projects/2000-01/risc/whatis/>> (Luettu 30.1.2014)
3. Asynchronous Transmission Overview, [WWW-dokumentti]
<<http://www.techfest.com/networking/wan/async.htm>> (Luettu 10.4.2014)
4. Hambley Allan R., Electronics, Upper Saddle River NJ: Prentice Hall, 2000, 2. painos, ISBN 0-13-691982-0.
5. Fette Bruce A., Cognitive Radio Technology, Burlington MA: Newnes/Elsevier, 2006, 1. painos, ISBN 978-0-7506-7952-7.
6. About the Bluetooth SIG, [WWW-dokumentti]
<<https://www.bluetooth.org/en-us/members/about-sig>> (Luettu 17.3.2014)
7. Granlund Kaj, Tietoliikenne, Jyväskylä: Docendo, 2007, 1. painos, ISBN 978-951-0-32821-7.
8. Specifications of the Bluetooth System: 3.0 + HS, [PDF-dokumentti]
<https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=174214> (Päivitetty 4/2009)
9. Quick Guide to IEEE 802.11 Activities, [WWW-dokumentti]
<http://www.ieee802.org/11/QuickGuide_IEEE_802_WG_and_Activities.htm> (Luettu 17.3.2014)
10. Specifications of the Bluetooth System: 4.0, [PDF-dokumentti]
<https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737> (Päivitetty 6/2010)
11. Specifications of the Bluetooth System: 4.1, [PDF-dokumentti]
<https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=282159> (Päivitetty 12/2013)
12. Behrouz A. Forouzan & Firouz Mosharraf, Computer Networks: A Top-Down Approach, New York NY: McGraw-Hill, 2012, 1. painos, ISBN 978-0-07-352326-2.
13. Data Transfer Across the Internet, [WWW-dokumentti]
<<http://essayweb.net/miscellany/datatransmission.shtml>> (Luettu 25.3.2014)
14. Sikora Axel, Wireless Personal and Local Area Networks, Chichester West Sussex England: Wiley, 2003, 1. painos, ISBN 0-470-85110-4.

15. Gratton Dean A., Developing practical wireless applications, Burlington MA: Elsevier, 2006, 1. painos, ISBN 978-1-55558-310-1.
16. Projektin vaatimusmäärittely, Sisäinen lähde
17. Käyttöliittymäsovelluksen ohjedokumentti Kajaanin ammattikorkeakoululle, Sisäinen lähde
18. ARF32 Bluetooth Audio Modules User Guide, [PDF-dokumentti]
<<http://www.farnell.com/datasheets/1675095.pdf>> (Päivitetty 8/2004)

