

Teppo Kauppinen

Helppokäyttöinen kaukosäädin vanhuksille

Insinööriö
Kajaanin ammattikorkeakoulu
Tekniikka ja liikenne
Tietotekniikka
24.11.2008



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Tekniikka ja liikenne	Koulutusohjelma Tietotekniikan koulutusohjelma
Tekijä(t) Teppo Kauppinen	
Työn nimi Helppokäyttöinen kaukosäädin vanhuksille	
Vaihtoehtoiset ammattiopinnot Sulautetut järjestelmät	Ohjaaja(t) Tuomo Rantala Toimeksiantaja Ismo Talus
Aika	Sivumäärä ja liitteet 31+6
<p>Insinööriyön tarkoituksena oli suunnitella ja toteuttaa vanhuksille suunnatun kaukosäätimen prototyyppi MediaPortal -ohjelmistolla toteutetulle multimediatekeskukselle. Kaukosäätimen toiminnot on rajattu hyvin suppeaan perustoimintojen joukkoon, joka sisältää välttämättömimmät toiminnot järjestelmän ohjaukseen. Näitä toimintoja ovat laitteen virran kytkentä ja sammuttaminen, valikkorakenteiden selailu sekä äänenvoimakkuuden säätö.</p> <p>Kaukosäädin koostuu mikrokontrollerista, näppäinmatriisista ja infrapunalinkin lähetinosiosta. Mikrokontrollerina käytetään Atmel ATTiny2313 -mikrokontrolleria, johon on liitetty 3x3 -näppäinmatriisi sekä infrapunälähetin. IR-lähetin koostuu IR-ledistä, virranrajoitusvastuksista sekä kahdesta transistorista. Mikrokontrolleri huolehtii transistorien ohjauksesta ja siten lähetettävän signaalin generoinnista.</p> <p>Kaukosäädin käyttää IR-linkissä Philipsin alun perin kehittämää RC-5-protokollaa, joka on yksinkertaisuutensa vuoksi hyvin suosittu tämän tyyppisissä laitteissa. Kyseinen koodaustapa käyttää Manchester-koodausta lähetettävän datan häiriönsiedon parantamiseksi.</p> <p>Prototyyppilaitteen testaus suoritettiin perustason tietokoneelle asennetussa MediaPortal-ympäristössä. Testauksen tuloksena oli, että suunniteltu laite on toimiva ja siitä voidaan jatkokehittää lopullinen tuote.</p>	
Kieli	Suomi
Asiasanat	
Säilytyspaikka	<input type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School School of Engineering	Degree Programme Information Technology
Author(s) Teppo Kauppinen	
Title Simple Remote Control for Elderly People	
Optional Professional Studies Embedded Systems	Instructor(s) Tuomo Rantala
	Commissioned by Ismo Talus
Date 26.11.2008	Total Number of Pages and Appendices 31+6
<p>The purpose of this Bachelor's thesis was to build a universal infrared remote control for a MediaPortal-based multimedia center that is utilized by elderly people. The functions built into the remote control were kept in minimum and therefore only the most important functions were implemented. These functions include powering the system on and off, moving around in the menu structures and controlling the system volume.</p> <p>The remote control is based on the Atmel ATtiny2313 microcontroller. The other parts of the hardware include 3x3 matrix keyboard and an infrared transmitter circuit. The transmitter circuit consists of an infrared photodiode, two transistors and three current limiting resistors.</p> <p>On the infrared link the remote control utilizes the RC-5 protocol originally developed by Philips. The RC-5 protocol is very simple to implement and therefore it is widely used in projects like this thesis.</p> <p>The resulting prototype of the designed remote control was tested in a basic computer environment which had the MediaPortal software installed. The testing concluded that the prototype works as intended and it can be used as a basis to a full-featured final product.</p>	
Language of Thesis Finnish	
Keywords	
Deposited at	<input type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input type="checkbox"/> Library of Kajaani University of Applied Sciences

ALKUSANAT

Haluan kiittää Tuomo Rantalaa insinööriyön ohjauksesta sekä Ismo Talusta tarvittavien välineiden järjestämisestä.

Kajaanissa 24.11.2008

Teppo Kauppinen

LYHENTEET

BJT	Bipolar Junction Transistor
ESD	Electrostatic discharge
LED	Light-Emitting Diode
IO	Input/Output
IR	Infrared, infrapuna
IRDA	Infrared Data
PWM	Pulse Width Modulation, pulssinleveysmodulaatio
WLAN	Wireless Local Area Network

SISÄLLYS

1 JOHDANTO	1
2 KAUKOSÄÄTIMEN TOIMINTAPERIAATE JA VAATIMUKSET	3
2.1 Infrapunalinkki	4
2.2 Toteutustapoja	4
2.3 Philips RC5 -protokolla	5
2.4 Pulssileveysmodulaatio IR-linkin toteutuksessa	6
3 KAUKOSÄÄTIMEN ELEKTRONIIKAN SUUNNITTELU	8
3.1 Suunnittelu	8
3.2 Mikrokontrolleri	10
3.3 Näppäimistö	11
4 KAUKOSÄÄTIMEN OHJELMISTON SUUNNITTELU	13
4.1 AVR Studio-kehitysympäristö ja WinAVR	13
4.2 Ohjelmiston toimintaperiaate	14
5 KAUKOSÄÄTIMEN LIITTÄMINEN PC-YMPÄRISTÖÖN	16
5.1 MediaPortal	16
5.2 Vastaanotin	17
6 KAUKOSÄÄTIMEN ULKOASU	19
6.1 Prototyypin ominaisuudet	19
6.2 Lopullisen tuotteen malli	21
7 LAITTEEN TESTAUS	23
8 TULOSTEN ANALYSOINTI JA JATKOKEHITYSIDEAT	28
9 YHTEENVETO	29
LÄHTEET	30
LIITTEET	

1 JOHDANTO

Yhtenä nykypäivän tiedonsiirron peruspilareista voidaan pitää sähkömagneettista säteilyä. Sähkömagneettisten aaltojen spektri on hyvin laaja. Niitä on havaittu ainakin taajuusalueella $10^{-3} \dots 10^{24}$ Hz. Tämä alue jaetaan lähinnä aaltojen syntyvän perusteella eri osa-alueisiin, jotka yhdessä muodostavat sähkömagneettisten aaltojen spektrin. Näitä alueita ovat mm. radioaallot, mikroaallot, infrapuna-aallot sekä näkyvä valo. Näistä erityisesti radioaaltojen käyttö langattomassa tiedonsiirrossa on hyvin yleistä ja niitä käytetäänkin nimensä mukaan radioliikenteessä. Lisäksi radioaaltoja käytetään televisiosignaalien lähettämisessä ja esimerkiksi langattomissa tietoliikenneverkoissa ja tiedonsiirtoratkaisuissa [1], joita ovat mm. WLAN ja Bluetooth. Nykyään radioaalloilla toteutetut tiedonsiirtoratkaisut ovat kovaa vauhtia korvaamassa infrapuna-aalloilla toteutettuja langattomia tiedonsiirtoratkaisuja.

Saksalainen muusikko ja tähtitieteilijä Frederick Herschel havaitsi infrapunasäteilyn (IR-säteily) olemassaolon vuonna 1800. IR-säteilyn taajuusalue on noin $10^{11} \dots 4,3 \cdot 10^{14}$ Hz ja aallonpituusalue noin 1 mm:stä 700 nm:ään. Pitkäaaltoisen IR-säteilyn voimme tuntea iholla lämpönä, jonka vuoksi sitä sanotaan usein lämpösäteilyksi [1.]

IR-säteilyä hyödynnetään mm. lämpökameroissa, joita käytetään etsimään ekstreemien ihmisten etsimiseen ja rakennusten lämpövuotojen paikantamiseen. Sen avulla voidaan myös mitata pintalämpötiloja ilman anturin kosketusta. Lisäksi sovelluksia löytyy myös puunjalostusteollisuudesta, jossa lämpösäteilyä käytetään materiaalien kuivattamiseen. Langattomassa tiedonsiirrossa IR-säteilyä käytetään mm. langattomissa lähiverkoissa ja erilaisissa kaukosäätimissä [1], joista tässä työssä käsitellään yksinkertaista IR-linkkiä käyttävää kaukosäädintä. On kuitenkin huomattava, että IR-linkin käyttö rajoittuu vain tapauksiin joissa lähettimen ja vastaanottimen välille on mahdollista muodostaa näköyhteys ja niiden välinen etäisyys on muutamia metrejä [2, s. 58.]

Tämän insinööriyön aiheena on suunnitella ja toteuttaa helppokäyttöinen kaukosäädin vanhuksille erityisesti heille räätälöidyn järjestelmän ohjaukseen, joka pohjautuu MediaPortal -multimediakeskusohjelmistoon. Toimeksianto työhön tuli Kajaanin Ammattikorkeakoulun laboratorioinsinööri Ismo Talukselta ja työ on toteutettu annettujen speksien mukaan. Työtä suunnitellessa speksejä muutettiin vielä hieman, jotta toteutuksesta tulisi mielekäs ja turhat ominaisuudet karsittiin pois. Työssä otettiin kuitenkin huomioon mahdollisuus lisätä kau-

kosäättimeen tässä pois jätettyjä ominaisuuksia, kuten esimerkiksi digitaalisella potentiometrillä toteutettava rullasäädin.

2 KAUKOSÄÄTIMEN TOIMINTAPERIAATE JA VAATIMUKSET

Kaukosäätimen toimintaperiaate perustuu lähettimeen, joka lähettää näppäinkomentoa vastaavan koodin kohteena olevan laitteen vastaanottimelle. Tämä lähetin/vastaanotinpari voidaan toteuttaa monella eri tavalla, joista nykyisin pääasiallisesti käytössä ovat infrapunalinkki ja radiotaajuuslinkki. Ennen muoin kaukosäädin on toiminut myös kiinteällä johdolla liitetynä, mikä ei ole kovin käyttäjäystävällinen vaihtoehto johdon mahdollisen sotkeentumisen takia. Kaukosäädin lukee näppäinkomennot yleensä näppäinmatriisista ja tutkii mitä näppäintä on painettu joko diskreettien logiikkapiirien tai mikrokontrollerin avulla, sen jälkeen se koostaa näppäimen painallusta vastaavan sanoman lähetettäväksi laitteelle valitun liitännämenetelmän kautta. Jokaisella valmistajalla on omanlaisensa protokolla laitteiden väliseen kommunikointiin, ja tämän vuoksi eri valmistajien kaukosäätimet eivät voi ohjata samoja laitteita. On kuitenkin yleismallisia kaukosäätimiä joille voidaan opettaa eri laitteiden käyttämiä koodistoja esimerkiksi ohjelmoimalla niitä tietokoneen avustuksella.

Suunnittelun lähtökohtana oli tehdä mahdollisimman yksinkertainen ja selkeä mutta silti monipuolinen yleiskaukosäädin, joten ulkoiseen olemukseen tuli kiinnittää erityistä huomiota. Näppäimiä ei saanut olla liian montaa, ja niiden tuli olla riittävän isokokoisia ja selkeitä. Toiminnallisuuden osalta päädyttiin valitsemaan laitteeseen vain välttämättömimmät perustoiminnot MediaPortal-järjestelmän [3] ohjaukseen, mikä yksinkertaisti suunnittelua huomattavasti. MediaPortal-järjestelmästä on kerrottu enemmän luvussa 5.1.

Vaatimuksiksi muotoutuivat seuraavat asiat:

- normaalin kaukosäätimen kokoinen, n. 15cm x 7cm x 2cm
- painikkeet virran kytkemiseen, kanavien vaihtamiseen, äänenvoimakkuuden säätämiseen, valikoissa liikkumiseen ja näytön zoomaukseen
- selkeät ja riittävän suuret painikkeet
- toteutus soveltuvalla infrapunalinkillä
- pieni virrankulutus, kun kaukosäädin ei ole käytössä
- yhteensopiva MediaPortal -ohjelmiston kanssa.

Tässä työssä kuvattu kaukosäädin käyttää näppäimistöissä matriisiperiaatetta, jota mikrokontrolleri tarkkailee. Näppäintä painettaessa kontrolleri tutkii, mitä näppäintä on painettu ja lähettää sille määritellyn koodin vastaanottimelle. Mikrokontrollerin ohjelmisto on kuvattu tarkemmin osassa 4.2 Ohjelmiston toimintaperiaate.

2.1 Infrapunalinkki

Yleisesti eri laitteiden kaukosäätimet toimivat infrapunalinkkiä käyttäen siten, että kaukosäätimessä on lähetin ja ohjattavassa laitteessa vastaanotin. Linkki voi olla myös kaksisuuntainen, jolloin monipuolisemmat kaukosäätimet voivat näyttää omalla näytöllään tietoja laitteesta, esimerkiksi äänenvoimakkuuden tai muuta vastaavaa tietoa.

Infrapunalinkin lähetin koostuu infrapunavaloa lähettävästä valodiodista (IR-led) jota ohjataan tietyllä vastaanottimen kanssa yhteensopivalla kantoaallolla. Vastaanotin muodostuu infrapunavalolle herkstä transistorista, joka kytkee, kun se saa tietyn aallon pituista ja riittävän voimakasta infrapunavaloa. Vastaanottimia saa valmiiksi oheiselektronikalla varustettuihin paketteihin, jolloin täytyy huolehtia vain siitä, että lähetin lähettää vastaanottimelle soveliaista kantoaaltoa, johon on moduloitu oikeanlainen data käyttäen valittua menetelmää.

Joissakin tietokoneissa on valmiina infrapunavastaanotin, jota voidaan käyttää myös kaukosäätimen vastaanottimena. Useimmiten nämä vastaanottimet on kuitenkin tarkoitettu IrDA eli Infrared Data -linkin käyttöön eivätkä siksi ole normaalien kaukosäätimien optimaaliselle toiminta-alueelle viritettyjä, mikä näkyy kantaman heikentymisenä. Tämän takia tietokoneeseen on suositeltavaa asentaa juuri kyseessä olevalle kaukosäätimelle suunniteltu vastaanotin.

2.2 Toteutustapoja

Infrapunalinkki (myöhemmin IR-linkki) voidaan toteuttaa käyttäen valmiita lähetin- ja vastaanotinpiirejä, kuten esimerkiksi datasiirrossa suosittuja IrDA-tyyppisiä ratkaisuja. Nämä piirit eivät kuitenkaan toimi yleisesti kaukosäätimien käyttämällä 36 - 38 kHz kantoaallolla parhaimmalla mahdollisella tavalla, vaan häviävät kantamassa varsinaisille kaukosäädinkäyttöön tehdyille piireille.

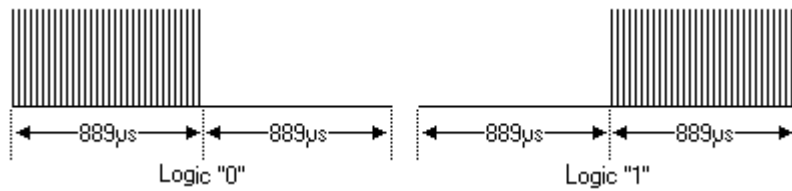
Yleisesti kodin viihde-elektronikassa kaukosäätimen ja laitteen välinen IR-linkki on toteutettu erillisellä vastaanotinpiirillä, joka purkaa saamansa Manchester -koodatun datan normaalisti sarjamuotoiseksi binääridataksi. Lähettimenä tällaisessa järjestelmässä toimii yksi tai kaksi IR-lediä, joita ohjataan transistorikytkimen kautta erillisellä mikrokontrollerilla. Mikrokontrolleri huolehtii tällaisessa tapauksessa tarvittavan kantoaallon generoinnin ja datan liittämisen siihen taajuusmodulaatiota hyväksi käyttäen. Lisäksi mikrokontrolleri koodaa datan sopivaan muotoon lähetystä varten, yleensä käyttäen Manchester-koodausta IR-ledin ohjauksessa.

2.3 Philips RC-5 -protokolla

Philipsin alun perin kehittämä RC-5 -koodaustapa on eniten käytetty IR-linkin koodaus harrastajien keskuudessa, koska sitä käyttäviä halpoja kaukosäätimiä on paljon saatavilla. Sen ominaisuuksiin kuuluu viisibittinen osoitekenttä, kuusibittinen komentosana, ja se on Manchester-koodattu. Alkuperäisen määritelmän mukaan kantoaallon taajuus on 36 kHz, jolloin kaikki bitit ovat ajallisesti 1,778 ms pitkiä. Tässä työssä käytetään kuitenkin 38 kHz:n kantoaaltoa, jotta yhteensopivuus vastaanottimen kanssa olisi parempi.

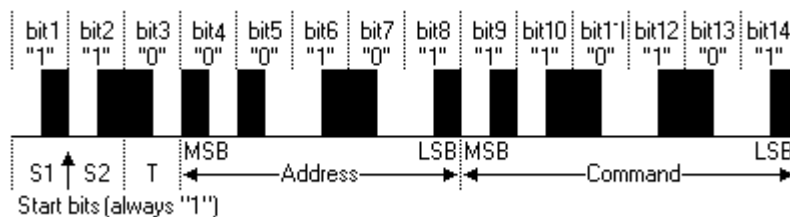
Granlund kertoo kirjassaan [2, s. 99], että Manchester-koodauksessa yhden bitin siirtämiseen tarvitaan enimmillään kaksi tilanmuutosta linjalla. Tästä johtuen yhden bitin siirtämiseen tarvittava aikajakso on jaettu kahteen osaan ja siirrettävän bitin tila saadaan tutkimalla mihin suuntaan tilanmuutos tapahtuu kyseisen aikajakson puolessavälissä. Jos lähetettävä bitti on 1, tilanmuutos on nouseva (alhaalta ylös), ja jos lähetettävä bitti on 0, tilanmuutos on laskeva (ylhäältä alas) [2, s. 98.]

RC5-protokollan ollessa kyseessä tilanmuutosta tutkitaan kantoaallon olemassaolosta. Kyseisessä tapauksessa aikajakson toinen puolikas on täytetty kantoaallolla ja toinen puoli on tyhjä. Jos kantoaalto on ajanjakson ensimmäisellä puolikkaalla, tulkitaan siirrettävä bitti loogiseksi nollaksi (laskeva tilanmuutos). Looginen ykkönen on taas päinvastoin alkupuoliskoltaan tyhjä ja loppupuoliskoltaan kantoaaltoa (nouseva tilanmuutos). Tämä on havainnollistettu kuvassa 1. [4][5.]



Kuva 1. RC5-koodin logiikkatasojen modulointi havainnollistettuna. [4]

Varsinainen protokolla on määritelty siten, että viesti on 14 bittiä pitkä ja se alkaa kahdella aloitusbitillä, jotka ovat molemmat loogisia ykkösiä. Viestin kolmas bitti on ns. toggle-bitti, joka vaihtaa tilaa aina kun kaukosäätimen näppäin päästetään ylös ja painetaan uudelleen. Vastaanotin käyttää tätä tunnistamaan pohjassa pidetyn näppäimen. Seuraavat viisi bittiä kertovat ohjattavan laitteen osoitteen. Osoite lähetetään aina eniten merkitsevä bitti (MSB) ensin. Osoitteen jälkeen tulee kuusibittinen komentosana, joka myös lähetetään MSB ensin. Viestin lähety aika on kokonaisuudessaan 25 ms. Mikäli näppäintä pidetään pohjassa, viestin lähety toistetaan 114 ms:n välein. Esimerkki protokollan mukaisesta viestistä on esitetty kuvassa 2 [4][5.]



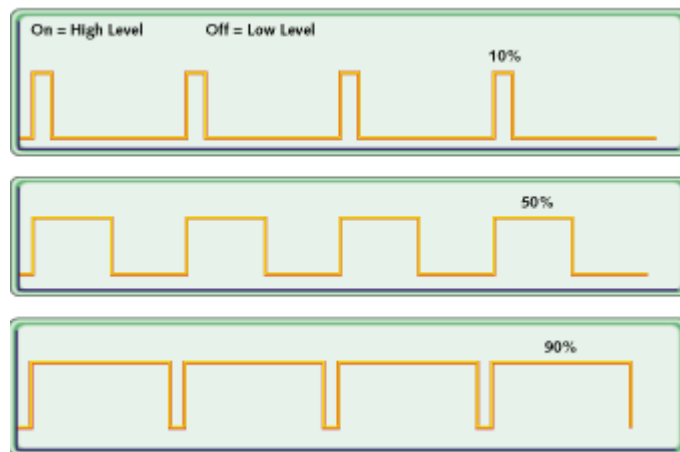
Kuva 2. Esimerkki RC5-koodatusta viestistä [4.]

2.4 Pulssileveysmodulaatio IR-linkin toteutuksessa

Kanta-aallon luominen mikrokontrollerin avulla on ohjelmallisesti hyvin epätarkkaa, mutta kun valitaan sopiva mikrokontrolleri, on kanta-aallon luomiseen mahdollista käyttää pulssileveysmodulaatiota (PWM) ja siten saada säädettävä kanta-aallon taajuus hyvin tarkasti haluttuun arvoon. Tämä on mahdollista siksi, että pulssileveysmodulaatiolla generoidun signaalin taajuus ei riipu mikrokontrollerissa ajettavasta ohjelmistosta muutoin kuin käynnistyksen ja pysäytyksen osalta. Pulssileveysmodulaatio on sidottu ainoastaan mikrokontrollerin kelloaaju-

teen, ja siten esimerkiksi 4 MHz:n kellolla toimivalla mikrokontrollerilla voidaan generoida hyvin tarkka 38 kHz:n kantiaalto käytettäväksi infrapunalinkissä.

Pulssileveysmodulaation käyttö mahdollistaa lisäksi pulssisuhteen säädön, jolloin on mahdollista muuttaa lähetettävän pulssin muotoa säätämällä pulssin logiikkatasojen suhdetta. Esimerkkejä tästä on nähtävissä kuvassa 1. PWM:n käyttö IR-linkin kantaallon generointiin on perusteltua juuri tarkasti säädettävän taajuuden osalta. Käytettävä pulssisuhde on 50 % ON - 50 % OFF jotta signaalista saadaan symmetrinen [6.] Periaatteessa olisi mahdollista käyttää ulkoista signaalilähdettä, esimerkiksi RC-piiriä, kantaallon luomiseen, mutta PWM-toteutus mahdollistaa kantaallon helpon muuttamisen ohjelmakoodia muuttamalla tarpeen niin vaatiessa. Kuvassa 3 on esitetty muutamia esimerkkejä erilaisista PWM-signaaleista.



Kuva 3. Erilaisilla päälläoloajoilla varustettuja PWM-generoituja signaaleja [6.]

3 KAUKOSÄÄTIMEN ELEKTRONIIKAN SUUNNITTELU

Työn elektroniikka on suunniteltu käyttämällä Spectrumin Micro-Cap 8- [7] ja Mentor Graphicsin Pads Logic [8] -ohjelmistoja. Lisäksi prototyypin piirilevyt on suunniteltu käyttäen Mentor Graphicsin Pads Layout- ja Pads Router -ohjelmia [8]. Elektroniikan suunnittelussa on pyritty huomioimaan vallitsevat määräykset pienelektronikkalaitteille niin turvallisuuden kuin häiriönsietokyvyn ja häiriöitä aiheuttavien tekijöiden osalta. Varsinaisia EMC-testejä prototyypille ei kuitenkaan ole tehty. Prototyypin kytkentäkaavio ja piirilevykuvat löytyvät liitteestä 1.

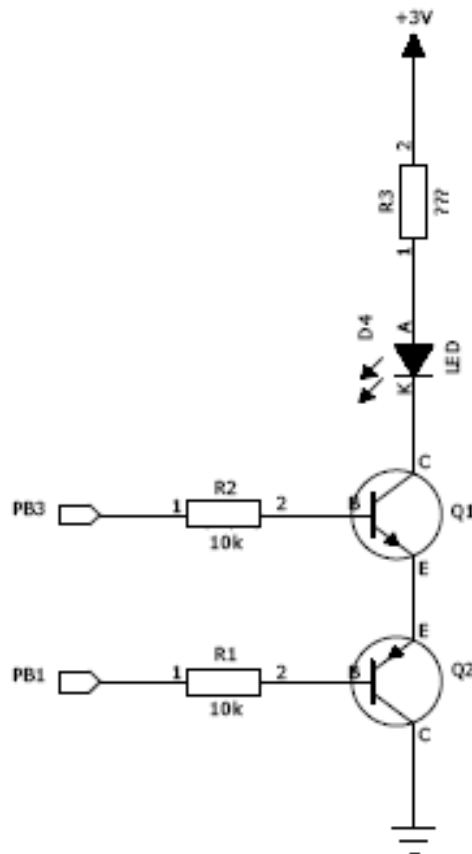
3.1 Suunnittelu

Elektroniikan suunnittelu aloitettiin valitsemalla sopivat toteutustavat laitteiston eri osille. Koska laitteiston pääasiallinen toimintaperiaate on muuntaa näppäinkomennot infrapunalinkin ymmärtämäksi dataksi, laitteiston tärkeimmät osat ovat näppäimistö ja sitä lukeva mikrokontrolleri, joka hoitaa myös infrapunalinkin ohjauksen lähetyssä. Mikrokontrolleri aiheuttaa tiettyjä rajoituksia näppäimistön ja infrapunalinkin toteutuksen suhteen, joten loogisinta oli valita ensin mikrokontrolleri. Tämän jälkeen muun elektroniikan suunnittelu on helpompaa, sillä mikrokontrollerin pinnimäärä rajoittaa näppäimistön kokoa ja sen virranantokyky vaikuttaa infrapunadiodin ohjaukseen tarvittavan elektroniikan määrää.

Mikrokontrolleriksi valittiin Atmel ATTiny2313, joka käyttää samaa käskykanta kuin esimerkiksi isompi ATmega128, jota käytetään hyvin paljon esimerkiksi sulautettujen järjestelmien laboraatioissa Kajaanin ammattikorkeakoulussa. AVR-perheen mikrokontrollerit ovat yleisesti ottaen helposti lähestyttäviä niin elektroniikkasuunnittelun kuin ohjelmistosuunnittelunkin näkökulmasta, ja siksi kyseisen perheen kontrolleri oli luonteva valinta tämän työn toteuttamiseen.

Infrapunadataa lähettävän diodin ohjaus vaatii suurehkoa hetkellistä virtaa, mihin mikrokontrollerin virranantokyky ei riitä. Tämän vuoksi diodin ohjaukseen joudutaan käyttämään transistorikytkintä, joka toteutetaan tässä työssä käyttäen kahta BJT-tyyppistä transistoria. Datan yhdistäminen kantoaaltoon tapahtuu myös tässä transistorikytkimessä siten, että toisen, NPN-tyyppisen transistorin kannalle tulee kantoaalto mikrokontrollerilta ja toisen,

PNP-tyyppisen transistorin kannalle data. Transistorien emitterit kytketään yhteen ja NPN-transistorin kollektori kytketään ledin kautta käyttöjännitteeseen. PNP-transistorin kollektori kytketään maapotentiaaliin. Toteutettu kytkentä on esitetty kuvassa 4.



Kuva 4. IR-Ledin ohjauskytkentä

Transistoreiksi voidaan valita melkeinpä mitkä tahansa yleiskäyttöiset transistorit, sillä taajuusalue, jolla toimitaan, ei ole kovinkaan suuri. Suurin rajoite on transistorien virrankesto, jonka tulee olla vähintään 100 mA, mutta suuremmasta virrankestoista ei ole haittaa. Tässä työssä käytetään BD137- ja BD138-transistoreja, jotka ovat toistensa komplementtiparit. Nämä transistorit kykenevät toimimaan vielä 10 MHz:n taajuudella, joten 38 kHz:n kantaalto ei ole niille ongelma. Lisäksi ne ovat virrankestoiltaan jopa liiankin järeitä, sillä maksimi jatkuva virta, jota ne kestävät on 1,5 A ja hetkellisen virran osalta 2 A [9.]

Transistorien suuri virrankesto mahdollistaa IR-ledin korvaamisen tarvittaessa tehokkaammalla mallilla ilman, että kytkentään tarvitsee tehdä muutoksia muutoin kuin ledin etuvastuksen osalta. Jotta kytkentä toimisi oikein, täytyy kytkentään mitoittaa kolme vastusta. Kuvassa 4 olevat vastukset R1 ja R2 valitaan siten, että transistorit vievät virtaa mikrokontrollerilta vain nimellisesti. Hyviä arvoja ovat esimerkiksi 4,7 k Ω ja 10 k Ω , joista jälkimmäinen valittiin

tässä toteutettavaan kytkentään. Lisäksi IR-lediä varten täytyy mitoittaa etuvastus R_3 , jonka koko riippuu käyttöjännitteestä, ledin yli haluttavasta virrasta sekä ledin kynnsjännitteestä. Vastuksen resistanssi voidaan laskea kaavalla

$$R = \frac{U - U_{led}}{I}.$$

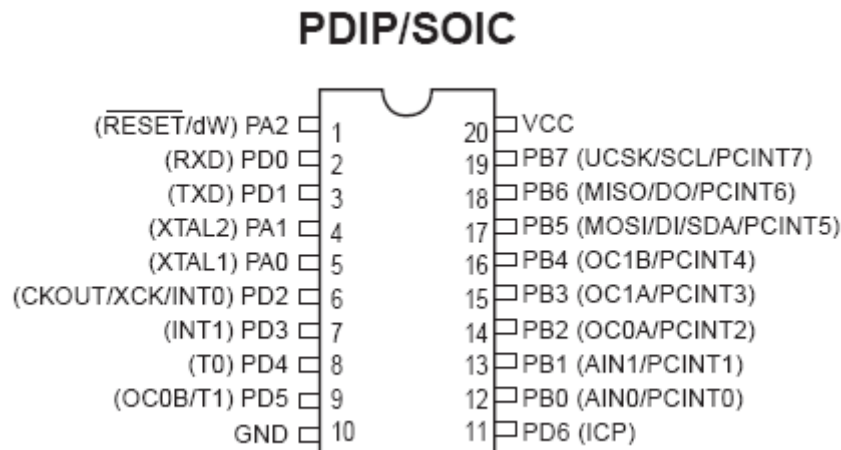
Kun tähän kaavaan sijoitetaan käyttöjännite $U = 3 \text{ V}$ ja IR-ledin kynnsjännite $U_{led} = 1,5 \text{ V}$ sekä ledille sallittu maksimivirta $I = 100 \text{ mA}$ saadaan kaava muotoon

$$R = \frac{3 \text{ V} - 1,5 \text{ V}}{100 \text{ mA}} = 15 \text{ } \Omega.$$

Ledin etuvastukseksi tulee asettaa siis vähintään $15 \text{ } \Omega$:n suuruinen vastus, jotta ledille ei johdeta liian suurta hetkellistä virtaa. Koska ledin ohjaus tapahtuu pulssitettuna, voidaan huolehtia valita ledin etuvastukseksi edellä laskettu $15 \text{ } \Omega$:n vastus. Valodiodi kestäisi hetkellisesti vieläkin suurempia virtoja, koska päälläoloaika on suhteellisen pieni, mutta virran kasvattaminen lyhentäisi ledin elinikää ja siten vaikuttaisi kaukosäätimen käyttöikään negatiivisesti.

3.2 Mikrokontrolleri

Työhön valittu mikrokontrolleri, Atmel ATTiny2313, on 8-bittinen AVR-käskykantaan käytävä kontrolleri. Kuvassa 5 näkyy tässä työssä käytetty 20-pinninen versio kyseisestä mikrokontrollerista. Kyseisessä versiossa on 3 IO-porttia, joista portti A on 3-bittinen, portti B 8-bittinen ja portti D 7-bittinen. Porteilla on myös vaihtoehtoisia toimintoja, kuten esimerkiksi portissa B ulkoiset keskeytykset ja pulssileveysmodulaatioiden lähdöt. Mikrokontrolleri voi toimia jopa 20 MHz:n kellolla, mutta tässä työssä käytetään mikrokontrollerin sisäistä 12 MHz:n kelloa, jotta kaikki porttipinnit ovat käytettävissä laitteelle. Ulkoinen kello nimittäin veisi kaksi pinniä portista D, mikä vaikeuttaisi näppäinmatriisin liittämistä mikrokontrolleriin [10.]

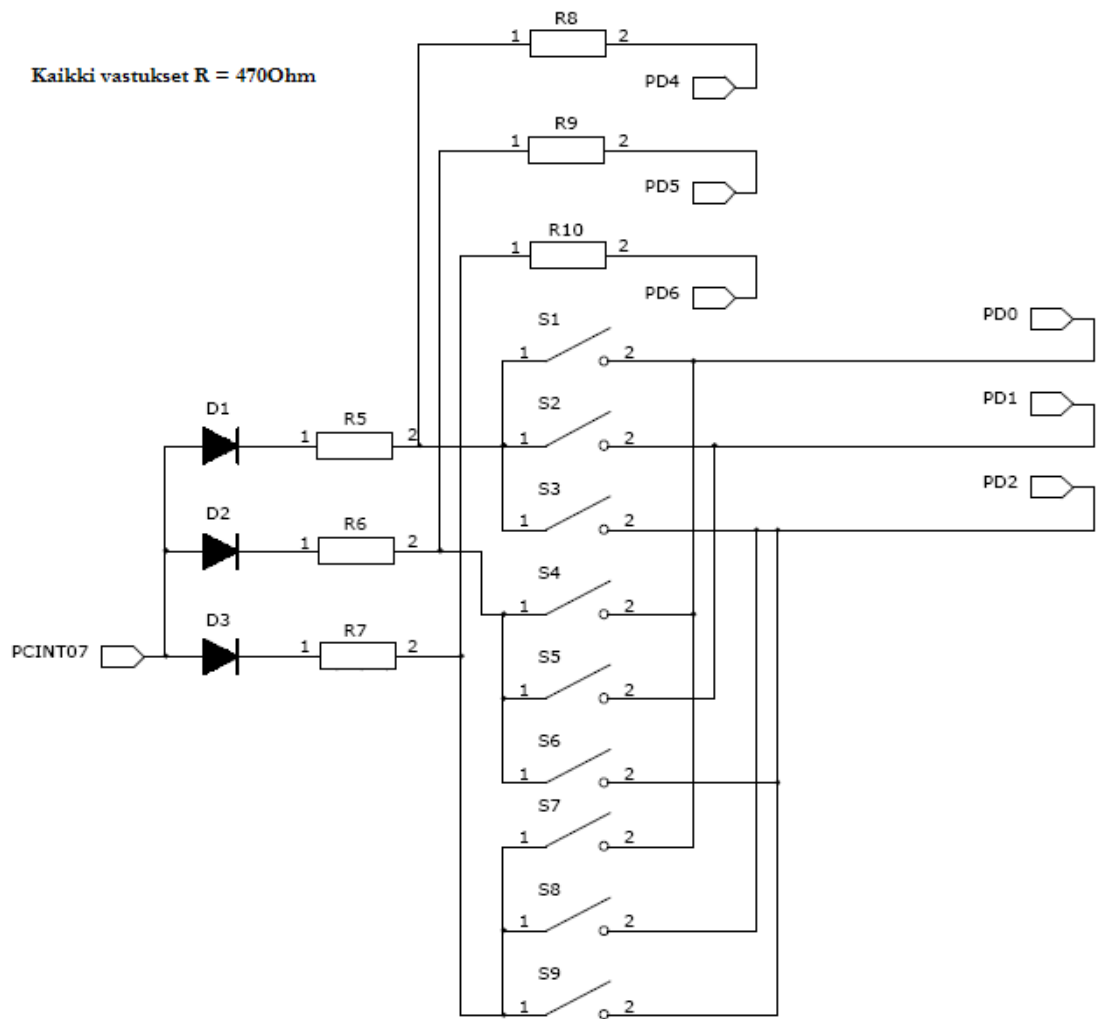


Kuva 5. PDIP/SOIC-koteloidun ATtiny2313-mikrokontrollerin pinnijärjestys [10]

Mikrokontrollerista jäi käyttämättä muutamia pinnejä, joten tulevaisuudessa kaukosäädintä on mahdollista laajentaa esimerkiksi muuttamalla näppäinmatriisi 4x4-matriisiksi tai vaikkapa lisäämällä johdannossa mainittu rullatyypinen säädin.

3.3 Näppäimistö

Kaukosäätimen näppäimistö toteutetaan 3x3-matriisina, joka mahdollistaa yhteensä yhdeksän näppäimen liittämisen mikrokontrolleriin käyttäen vain kuutta mikrokontrollerin pinniä yhdeksän sijaan ja siten säästää mikrokontrollerin liitäntöjä muille toiminnoille. Lisäksi matriisista kytketään diodien kautta keskeytyslinja mikrokontrollerille, jotta kontrollerin virransäästöominaisuuksia voidaan käyttää siten, että näppäintä painettaessa kontrolleri saa herätteen ulkoisesta keskeytyksestä. Kaavio toteutettavasta kytkennästä on kuvassa 6.

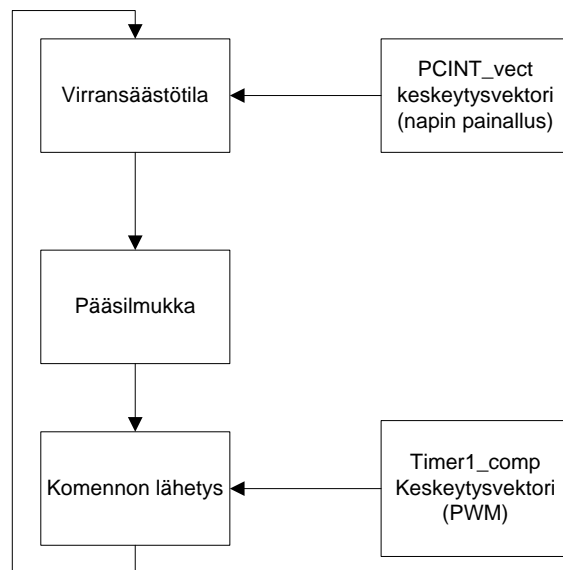


Kuva 6. Näppäinmatriisin kytkentäkaavio

Kytkenä on muunnelmä Atmelin tarjoamasta esimerkistä [11], josta saadaan suoraan tarvittavat komponenttiarvot. Kaikki vastukset ovat arvoltaan $470\ \Omega$, ja diodit ovat tyypiltään 1N4148. Vastukset rajoittavat kytkennän ottamaa virtaa ja samalla suojaavat mikrokontrollereita näppäimistön mahdollisilta ESD-piikeiltä. Kun jotain näppäintä painetaan, vetää jokin diodeista nastan PCINT07 logiikkatason nolnaan, mikä aiheuttaa mikrokontrollerille keskeytyksen ja saa sen heräämään lepotilasta.

4 KAUKOSÄÄTIMEN OHJELMISTON SUUNNITTELU

Tässä osiossa kuvataan kaukosäätimen ohjelmistoon käytetyt kehitystyökalut sekä ohjelmiston toimintaperiaate. Kaukosäätimelle tuleva ohjelmisto on kirjoitettu kokonaisuudessaan C-ohjelmointikielellä ja se on optimoitu erittäin tehokkaaksi. Ohjelmiston lohkokaavio on esitetty kuvassa 7.



Kuva 7. Ohjelmiston lohkokaavio

4.1 AVR Studio-kehitysympäristö ja WinAVR

AVR Studio on Atmelin tarjoama ilmainen kehitysympäristö AVR-perheen mikrokontrollereille. Se mahdollistaa ohjelmien kirjoittamisen ja debuggauksen AVR-mikrokontrollereille Windows 98/ME/NT/2000/XP- ja Vista -käyttöjärjestelmissä. AVR Studio sisältää projektihallintatyökalut, lähdekoodieditorin ja emulaattorin. Varsinaista kääntäjää pakettiin ei kuulu, mutta AVR Studio toimii hyvin yhteen ilmaisen GCC-pohjaisen WinAVR-kääntäjän kanssa. Kehitysympäristöstä löytyy myös mahdollisuus käyttää erilaisia in-circuit-emulaattoreita yhdessä AVR Studion kanssa, joten valmiiden laitteiden kehitys, debuggaus ja päivitys onnistuvat tämän kehitysympäristön kautta helposti [12.]

WinAVR on kokoelma pieniä työkaluja, jotka mahdollistavat ohjelmistokehityksen Atmel AVR -perheen mikrokontrollereille Windows-käyttöjärjestelmissä. Siihen sisältyy GNU GCC -kääntäjä C- ja C++ -lähdekoodeille. Jakelupaketista löytyy myös kaikki muut tarvittavat työkalut ohjelmien kehittämiseen projektinhallintatyökaluista ja lähdekoodieditoreista alkaen aina mikrokontrollerin ohjelmointiin tarvittaviin työkaluihin saakka. WinAVR:ää voidaan siis käyttää myös yksinään ilman AVR Studiota, mutta käytettävyydeltään AVR Studio on huomattavasti parempi ja yksinkertaisempi käyttää. [13.]

Tässä työssä käytettiin AVR Studion versiota 4.14, WinAVR-kääntäjää sekä Atmel STK-500-kehitysalustaa mikrokontrollerin ohjelmointiin ja ohjelmiston debuggaukseen Atmelin tarjoaman debugWire-liityntän kautta käyttäen Atmelin JTAG ICE -liityntämoduulia.

4.2 Ohjelmiston toimintaperiaate

Mikrokontrollerille tuleva ohjelmisto on suhteellisen yksinkertainen ja lyhyt, sillä sen ei tarvitse huolehtia muusta kuin näppäinmatriisin lukemisesta ja IR-lediä ohjaavan pulssinleveysmodulaation (PWM) toiminnasta. Ohjelmistoon on määritelty kiinteästi eri näppäimien painalluksia vastaavat koodit Philipsin RC-5-standardin mukaisesti, ja PWM:n taajuudeksi on määritelty kiinteästi 38 kHz. Ohjelmistoon on lisätty myös ominaisuus, joka asettaa mikrokontrollerin virransäästötilaan, mikäli mitään näppäintä ei paineta. Ohjelmistoon määritellyt RC-5-koodit on esitelty taulukossa 1.

Taulukko 1. Ohjelmistoon määritellyt RC-5-koodit

Indeksi	Start-bitit	Toggle	Osoite	Komento
0	11	0	01011	100000
1	11	0	01011	100001
2	11	0	01011	010000
3	11	0	01011	010001
4	11	0	01011	001100
5	11	0	01011	110010
6	11	0	01011	110110
7	11	0	01011	110100
8	11	0	01011	110111

Ohjelmiston pääasiallinen toiminta on jaettu kahteen osaan. Ensimmäinen osa on ohjelmiston varsinainen pääsilmutka, joka huolehtii virransäätötilasta ja näppäinmatriisin lukemisesta. Näppäintä painettaessa mikrokontrolleri saa keskeytyksen PCINT-vektorilta, joka herättää ohjelman tutkimaan, mitä näppäintä on painettu. Näppäinmatriisin skannaus tapahtuu vaakarivi kerrallaan, ja kun painettu näppäin on saatu tunnistettua, ohjelma kutsuu sendCommandRC5-funktiota. Tämä funktio on ohjelmiston toinen osa, joka ohjaa PWM-generaattoria ja IR-lediä. Lähetettävä komentosana on 14 bittiä, ja yksi bitti on 64 PWM:n kellojaksoa pitkä, joten funktion toiminta pohjautuu näihin tiedettyihin vakioarvoihin. Tämä funktio suorittaa myös vaaditun Manchester-koodauksen. Kun komentosana on lähetetty, ohjelmisto palaa pääsilmutkaan, joka asettaa mikrokontrollerin virransäätötilaan. Ohjelmiston lähdekoodi löytyy liitteestä 2.

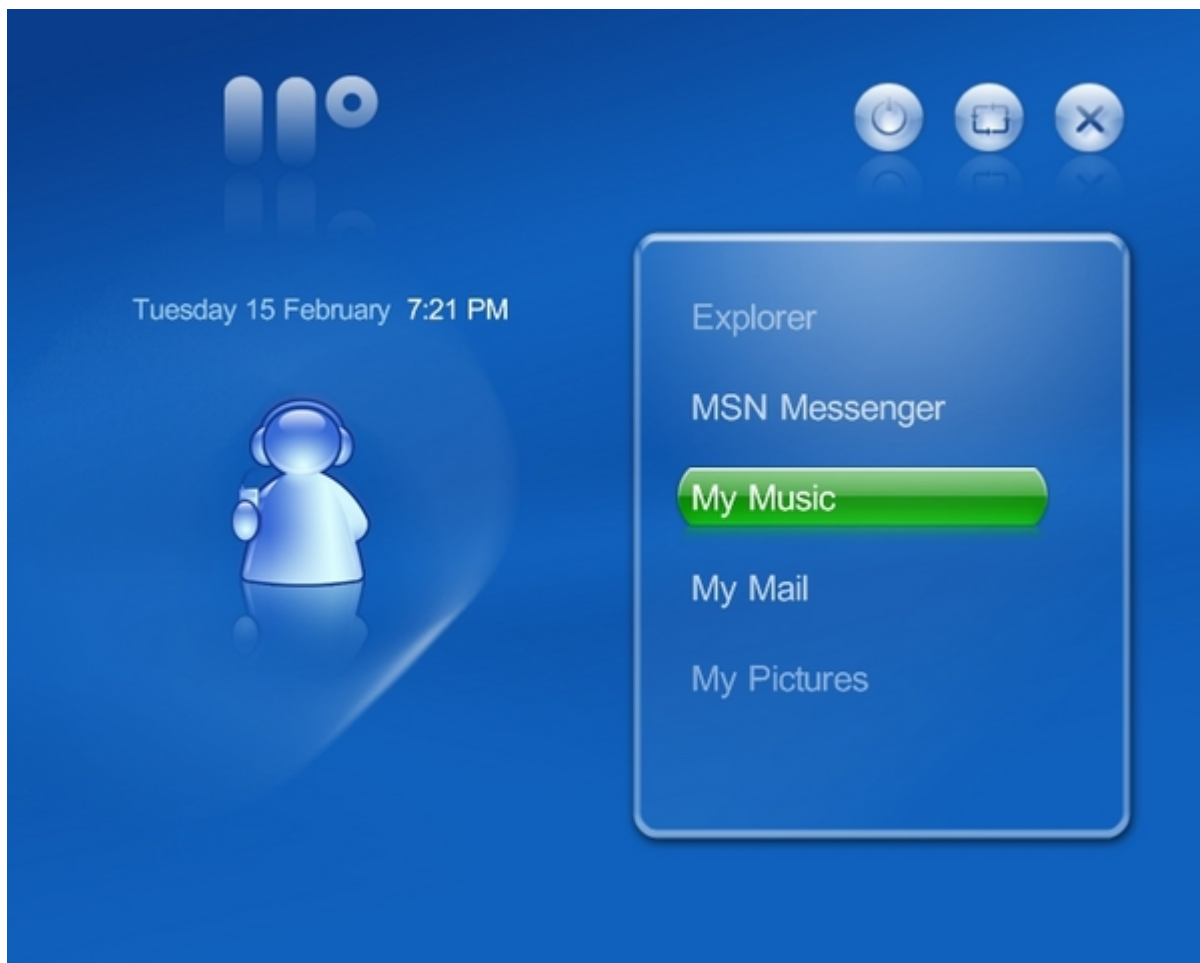
5 KAUKOSÄÄTIMEN LIITTÄMINEN PC-YMPÄRISTÖÖN

Kaukosäätimen liittämiseksi tietokoneeseen tarvitaan vastaanotin sekä sille soveltuva ohjelmisto. Tämä ei kuitenkaan riitä, vaan lisäksi tarvitaan tapa, jolla kaukosäätimen komentoja vastaanottava ohjelmisto välittää komennot ohjattavalle ohjelmalle. Koska tässä työssä kuvattu kaukosäädin on suunniteltu toimimaan yleisesti käytössä olevan LIRC-järjestelmän kanssa, voidaan se liittää helposti WinLIRC-liitännäisen avulla MediaPortal-ympäristöön. MediaPortal-ohjelmiston lisäksi tietokoneeseen tarvitaan siis WinLIRC-ohjelmisto ja MediaPortaliin lisäosa vastaanottamaan komentoja WinLIRC-ohjelmistolta. Tämä lisäosa on löydettävissä MediaPortalin kotisivuilta [3.]

5.1 MediaPortal

MediaPortal on avoimen lähdekoodin ohjelmisto, jolla on mahdollista muuttaa PC-tietokone monipuoliseksi mediakeskukseksi. Se mahdollistaa muun muassa radion kuuntelun ja television katselun tietokoneella sekä niiden nauhoittamisen aivan kuten tallentavalla digiboksilla. Koska MediaPortal on julkaistu avoimena, käyttäjät voivat kehittää sitä eteenpäin tarpeidensa mukaan. Samasta syystä se on myös täysin ilmainen ohjelmisto. MediaPortal sai alkunsa, kun sen kehittäjä Erwin Beckers lähti Xbox Media Center -projektista ja alkoi kehittää omaa, Microsoft Windows -käyttöjärjestelmän päälle tulevaa mediakeskusohjelmistoaan. Nykyään Mediaportalia kehittävässä yhteisössä on noin 10000 rekisteröitynyttä käyttäjää ja lisää jäseniä liittyy koko ajan [3.]

MediaPortal on myös helposti laajennettavissa oleva järjestelmä, ja siihen on saatavilla runsaasti erilaisia liitännäisiä, jotka mahdollistavat esimerkiksi ulkoisten ohjelmien, kuten pelien, suorittamisen ja vaikkapa tietokoneen lämpötilojen seurannan. Esimerkkikuva MediaPortalin käyttöliittymästä on kuvassa 8.



Kuva 8. Esimerkkikuva MediaPortalin käyttöliittymästä [3]

5.2 Vastaanotin

Alun alkaen tässä työssä oli tarkoitus käyttää omavalmisteista vastaanotinta infrapunasignaalille, mutta koska työn tilaajalta löytyi hyllystä soveltuva laite, päädyttiin käyttämään sitä. Laite on nähtävissä kuvassa 9. Vastaanotin on tyypiltään Extended Systems JetEye PC Serial Infra Red Magic Eye, ja se on suunniteltu erityisesti kaukosäädinten liittämiseen tietokoneisiin. Kyseinen laite toimii parhaiten 38 kHz:n kantaaltaa käyttävien kaukosäätimien kanssa. Vastaanotin kytketään tietokoneen sarjaporttiin, jolloin varsinaisia laiteajureita ei tarvita. Vastaanotinta luetaan suoraan sarjaportista käyttäen esimerkiksi LIRC (Linux Infra Red Control)-, WinLIRC-, uICE- tai Girder-ohjelmistoja. MediaPortal sisältää tuen kaikille LIRC:n tukemille kaukosäätimille, joten yhteensovittaminen on helppoa, sillä koko työn lähtökohtana on ollut suunnitella LIRC:n kanssa yhteensopiva kaukosäädin.



Kuva 9. Työssä käytettävä IR-vastaanotin

6 KAUKOSÄÄTIMEN ULKOASU

Lähtökohtana laitteen ulkoasun suunnitteluun oli yksinkertaisuus ja helppokäyttöisyys. Ulkoasun suunnittelu aloitettiin kyselemällä tuotteen kohderyhmään kuuluvilta henkilöiltä, miten he parantaisivat nykyisiä televisioiden ja digisovittimien kaukosäätimiä. Päälimmäisiksi toiveiksi nousivat suuret näppäimet ja helppo käytettävyys siten, että esimerkiksi kanavien vaihdossa ei vahingossa syötä numeroa 11, kun tarkoitus on syöttää numero 1. Kanavien vaihdon ongelma ratkaistiin siten, että päädyttiin tarjoamaan vain kanavaa ylös- tai alaspäin vaihtavat näppäimet. Näppäinten osalta päädyttiin siihen, että lopullisessa tuotteessa jokaisen painikkeen tulisi olla muodoltaan ja väriltään erilainen. Kustannussyistä prototyypissä kaikki näppäimet ovat samanlaisia.

6.1 Prototyypin ominaisuudet

Rakennettu prototyyppi on suunniteltu standardiin yleiskoteloon, ja siihen on sijoitettu yhdeksän painonappia. Kuvassa 10 on nähtävissä rakennettu laite.

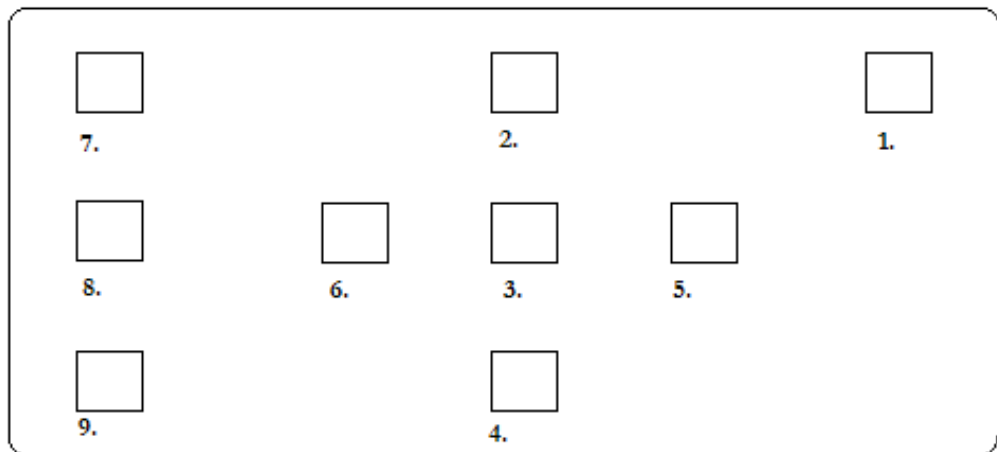


Kuva 10. Rakennettu kaukosäätimen prototyyppi

Piirilevy on toteutettu kaksipuoleisena, ja se on valmistettu jyrsimällä Kajaanin ammattikorkeakoulun tuotantotekniikan laboratoriossa. Painikkeet on liitetty levyyn johdoilla, jotta niiden sijoittelu olisi helpompaa ja laite voidaan säilyttää avattavana pariston vaihtamiseksi ja mikrokontrollerin irrottamiseksi uudelleenohjelmointia varten. Kuvassa 11 on nähtävissä painikkeiden sijoittelu, ja ne on numeroitu taulukossa 2 olevien painikkeiden toimintokuvusten mukaan.

Taulukko 2. Painikkeille määritellyt toiminnot ja niiden RC-5-koodit

Numero	Toiminto	RC-5 koodi
1	Virta päälle / pois	101100b
2	Valikko vasemmalle	110010b
3	Enter / Hyväksy valinta	110110b
4	Valikko oikealle	110100b
5	Valikko ylös	100000b
6	Valikko alas	100001b
7	Äänenvoimakkuus -	010001b
8	Esc / peruuta	110111b
9	Äänenvoimakkuus +	010000b



Kuva 11. Painikkeiden sijoitteluesimerkki

6.2 Lopullisen tuotteen malli

Lopullinen tuote voisi olla esimerkiksi kuvassa 12 esitetyn mallin kaltainen. Kotelon muuttaminen toki vaatisi uuden, juuri kyseiseen koteloon sopivan piirilevyn. Lopullisessa tuotteessa tulee ottaa huomioon esteettiset seikat, ja ulkonäön suunnittelu olisikin parasta jättää teollisen muotoilijan tehtäväksi.



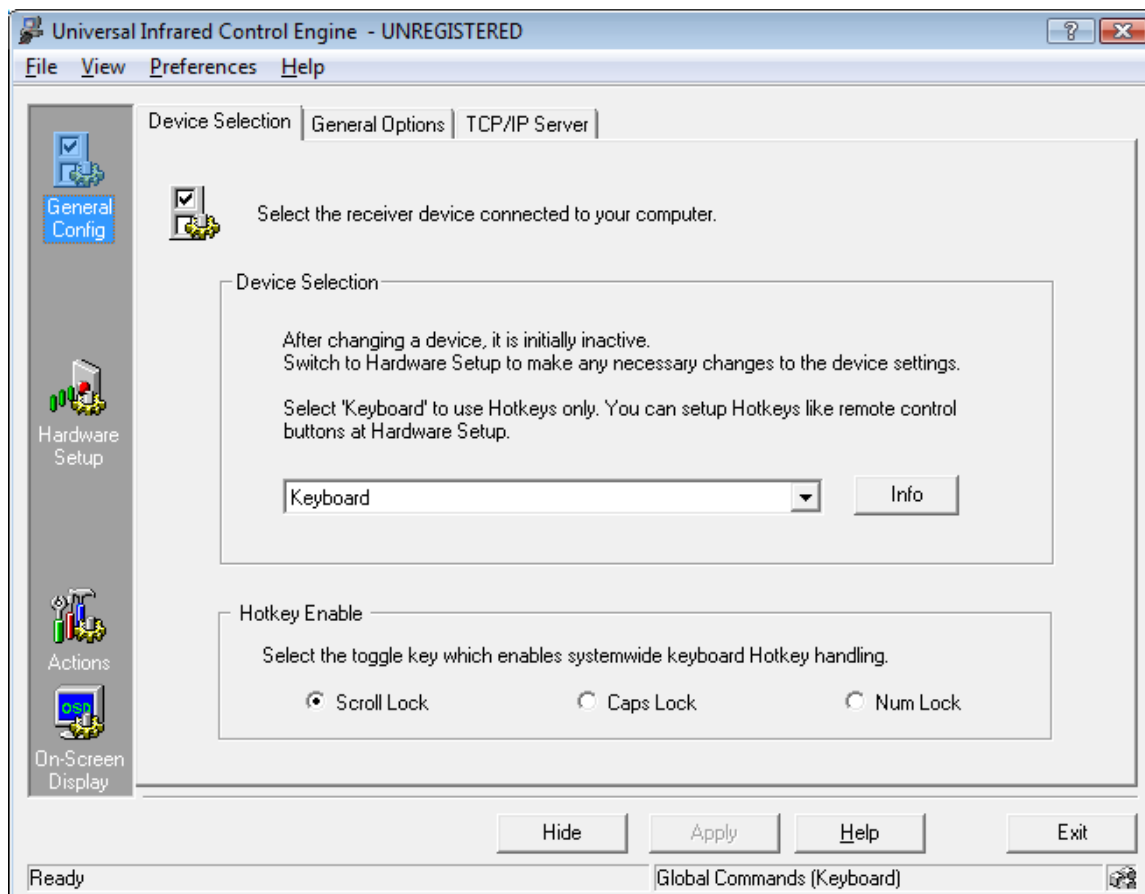
Kuva 12. Taiteilijan näkemys lopullisen tuotteen ulkoasusta

Kuvassa 12 esitetty kotelon prototyyppi olisi periaatteessa mahdollista valmistaa Kajaanin ammattikorkeakoulun tuotantotekniikan laboratoriossa sijaitsevalla 3D-tulostimella. Kuitenkin mallia piirrettäessä mittakaavaan ja paristojen sijoitteluun ei kiinnitetty juurikaan huomiota. Siksi mallia olisi muutettava jonkin verran, jotta siitä tulisi käyttökelpoinen varsinaisena kotelona. Lisäksi se tulisi piirtää uudestaan varsinaisella 3D-mallinnusohjelmalla.

7 LAITTEEN TESTAUS

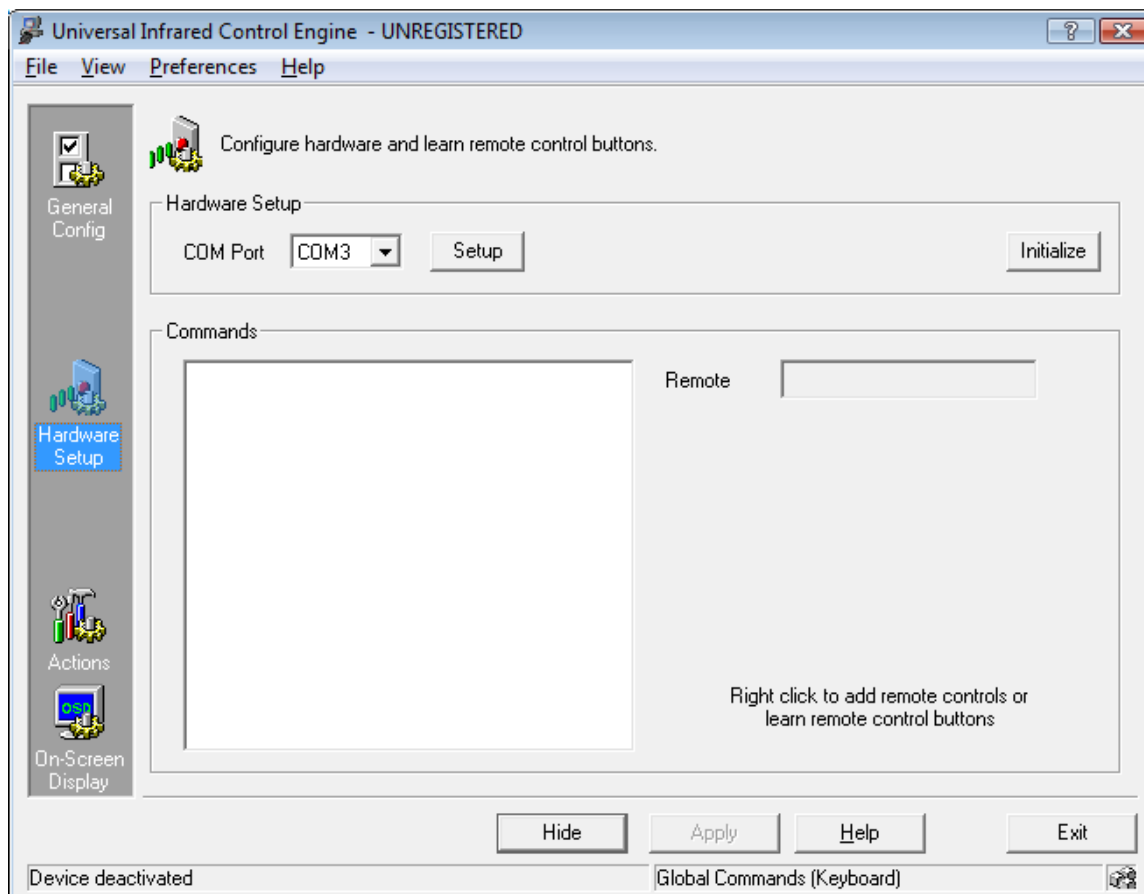
Laitteen testausympäristön alkuperäinen kokoonpano koostui perustason tietokoneesta, johon on asennettu Windows XP -käyttöjärjestelmä, MediaPortal RC2 -ohjelmisto, WinLIRC-ohjelmisto sekä tarvittavat ajurit. Referenssilaitteistona käytettiin tehokasta kotiteatterikäyttöön rakennettua tietokonetta, johon on asennettu Windows Vista -käyttöjärjestelmä ja MediaPortal RC2 -ohjelmisto. Referenssilaitteiston kaukosäätimenä toimii kaupallinen iMon Pad -kaukosäädin, jonka käyttämä iMon-hallintaohjelma on suoraan yhteensopiva MediaPortalin kanssa.

Testausympäristöä pystytettäessä ensimmäinen ongelma oli WinLIRC-ohjelmiston toimintaan saamisessa. Viimeisin versio WinLIRC-ohjelmasta on vuodelta 2003, ja sen yhteensopivuus Windows XP -käyttöjärjestelmän kanssa on puutteellinen. Yhteensopivuuden lisäämiseksi testikokoonpanoon asennettiin vielä MediaTexxin uICE [14](Universal Infrared Control Engine) -ohjelmisto, joka mahdollistaa kaukosäätimen toimintojen ohjelmoinnin vastamaan esimerkiksi normaaliin tietokoneen näppäimistöön määriteltyjä pikanäppäimiä. Kuvassa 13 on nähtävillä uICE-ohjelmiston käyttöliittymä ensimmäisen käynnistyksen yhteydessä.



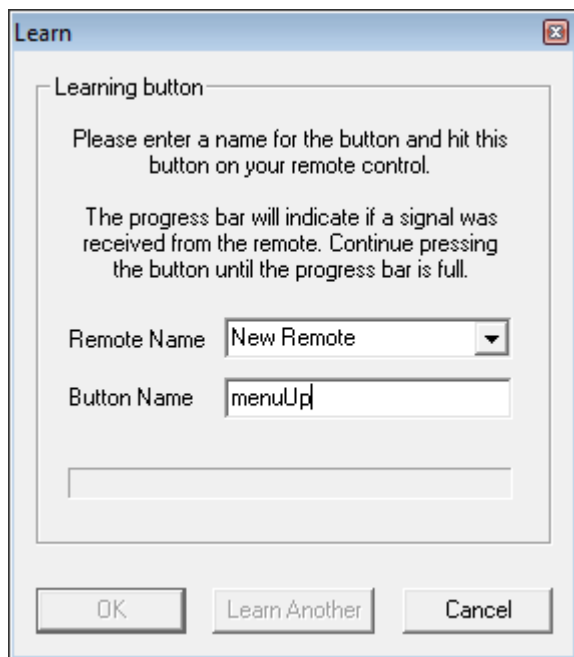
Kuva 13. Universal Infrared Control Engine -ohjelmiston ensimmäinen käynnistys

Koska käytössä oleva vastaanotin on yleismallinen sarjaporttiin tuleva IR-vastaanotin, voidaan uICE-ohjelmisto asettaa lukemaan sitä suoraan ilman WinLIRC-ohjelmistoa. Tämä tehdään siten, että uICE:n Device Selection -kohdassa valitaan laitteeksi *Generic Serial IR Receiver*. Tämän jälkeen voidaan siirtyä *Hardware Setup* -välilehdelle, joka on kuvan 14 kaltainen. Kyseisessä näkymässä säädetään vastaanottimen sarjaliikenteen parametrit sekä opetetaan kaukosäätimen painikkeiden koodit ohjelmistolle. Vastaanottimen sarjaliikenteen parametrit ovat laitekohtaisia, ja ne tulee selvittää aina kyseisen vastaanottimen käyttöohjeista, datakirjasta tai suoraan valmistajalta.



Kuva 14. uICE-ohjelmiston Hardware Setup -välilehti.

Näppäinkomennot opetetaan ohjelmistolle kuvassa 15 esitetyn näkymän kautta. Tässä näkymässä kirjoitetaan painikkeen toiminnolle nimi, jonka jälkeen painetaan kaukosäätimen näppäintä useaan kertaan, jotta ohjelmisto tunnistaa lähetetyn koodin oikein. Kun ohjelmisto on tunnistanut näppäimen, voidaan ohjelmalle opettaa seuraava näppäin painamalla *Learn Another* -painiketta. Kun kaikki painikkeet on syötetty, poistutaan valikosta painamalla *OK*-painiketta. Kyseinen näkymä on myös hyvä tapa testata kaukosäätimen ja vastaanottimen toiminta, sillä mikäli ohjelmisto ei tunnista painikkeita tai saa samaa näppäintä painettaessa joka kerta eri tuloksen, voidaan päätellä, että kaukosäädin ja vastaanotin eivät toimi oikein.



Kuva 15. Kaukosäätimen painikkeen opetus ohjelmistolle tapahtuu tämän näkymän kautta.

Kun kaukosäätimen painikkeille on määritelty vastaavat näppäinkomennot, voidaan avata MediaPortal ja testata kaukosäätimen toimivuus. MediaPortalin valikoissa liikutaan käyttäen nuolinäppäimiä sekä enter- ja esc-näppäimiä, jotka siis edellä määriteltiin kaukosäätimen painikkeisiin. Määritykset ovat nähtävissä taulukossa 3. Järjestelmä toimi odotetulla tavalla, ja kaukosäätimellä päästiin liikkumaan valikoissa. Koko laitteiston virranhallintaan joudutaan käyttämään globaalia pikanäppäintä, joka täytyy määritellä erikseen ohjelmistoon. Tässä työssä sitä ei määritelty.

Taulukko 3. Kaukosäätimen painikkeiden määrittelyt MediaPortalin näppäinkomentoihin

Numero	Toiminto	RC-5 koodi	Mediaportalin pikanäppäin
1	Virra päälle / pois	101100b	
2	Valikko vasemmalle	110010b	Vasen nuoli
3	Enter / Hyväksy valinta	110110b	Enter
4	Valikko oikealle	110100b	Oikea nuoli
5	Valikko ylös	100000b	Ylänuoli
6	Valikko alas	100001b	Alanuoli
7	Äänvoimakkuus -	010001b	-
8	Esc / peruuta	110111b	Esc
9	Äänvoimakkuus +	010000b	=

Verrattaessa referenssinä toimineeseen laitteistoon todettiin, että tässä työssä rakennettu prototyyppi toimii vähintään yhtä hyvin, ellei jopa paremmin, kuin vastaava kaupallinen tuote.

Varsinaisen kohderyhmän eli vanhusten kanssa kyseistä järjestelmää ei ole päästy testaamaan tätä kirjoitettaessa, mutta prototyyppilaitteen nähneet henkilöt ovat olleet pääasiassa sitä mieltä, että laite on riittävän yksinkertainen, jotta sitä oppisi käyttämään.

8 TULOSTEN ANALYSOINTI JA JATKOKEHITYSIDEAT

Prototyypilaitte onnistui odotusten mukaisesti, ja sen toimivuus saatiin todennettua. Varsinaiselta kohderyhmältä saatu palaute laitteen ulkoasusta oli positiivista, vaikka varsinaista kohderyhmälle suunnattua versiota järjestelmästä ei ollut testaushetkellä käytettävissä. Järjestelmän konfigurointi on suhteellisen monimutkainen tapahtuma, jonka suorittamiseen tarvitaan aiheeseen perehtynyt henkilö. Kun laitteisto on saatu toimintakuntoon, sen käyttäminen on erittäin helppoa ja onnistuu keneltä tahansa.

Kaukosäätimen jatkokehityksessä olisi hyvä siirtyä käyttämään radiotaajuksia komentojen siirrossa kaukosäätimeltä tietokoneelle. Tämä mahdollistaisi sen, että kaukosäädintä ei tarvitsisi osoittaa vastaanotinta kohti aina näppäintä painettaessa ja vastaanotin olisi mahdollista sijoittaa piiloon. Lisäksi kaukosäädin olisi tällöin mahdollista integroida esimerkiksi tuolin käsinojaan.

Tuotteen pariston kestoja olisi mahdollista parantaa vaihtamalla mikrokontrolleriksi vähemmän virtaa kuluttava mikrokontrolleri, esimerkiksi jokin Microchipin PIC-perheen mikrokontrolleri. Tämä kuitenkin vaatisi uuden piirikortin suunnittelun ja ohjelmiston muokkaamisen PIC-perheen käskykannalle sopivaksi.

9 YHTEENVETO

Insinööriyössä suunniteltiin ja toteutettiin MediaPortal-ympäristöön vanhuksille suunnattu yleiskäyttöinen kaukosäädin. Suunnittelu oli suhteellisen yksinkertainen ja nopea prosessi aiemman aiheeseen perehtyneisyyden ansiosta. Kaukosäätimen ohjelmiston suunnittelu oli aikaa vievää lähinnä käytetyn pulssinleveysmodulaation toteuttamisen osalta, koska yhtä selkeää ohjeistusta kyseisen toiminnallisuuden toteuttamiseen ei ollut saatavilla, vaan datakirjasta saatavia tietoja jouduttiin soveltamaan runsaasti.

Prototyypin rakentaminen kesti yllättävän pitkään ja varsinaista prototyypilaitteen ulkoasua ja osia jouduttiin muuttamaan työn teon aikana, koska alun perin valittuja osia ei ollut saatavilla. Muutettuja kohtia olivat kotelointi ja virransyöttö. Kotelon tyyppin muutos johti kaukosäätimen virransyötön muuttamiseen siten, että uuteen koteloon ei saatu mahdutettua ensiksi suunniteltuja AAA-tyypin alkaaliparistoja. Lopulliseen prototyyppiin tuli litteä 3 V:n litiumparisto.

Varsinaisen testausympäristön pystyttäminen oli myös kohtuullisen vaivaton prosessi, mutta valitettavasti varsinaista vanhuksille suunnattua versiota MediaPortal-järjestelmästä ei ollut saatavilla. Prototyypin perustoiminnallisuus saatiin todennettua, ja lopputulos oli että tässä työssä esitellyn prototyypin pohjalta on mahdollista jatkokehittää lopullinen tuote kyseiseen ympäristöön.

Koko insinööriyön prosessi viivästyí komponenttien, kuten kotelon ja paristoliittimien, saatavuussyistä noin puolella vuodella. Prototyypin peruskomponentit kuitenkin olivat hyvin saatavilla, joten ensimmäiset testaukset päästiin suorittamaan hyvissä ajoin. Laitetta ei päästy testaamaan lopullisessa käyttöympäristössä ja kohteena olleella käyttäjäryhmällä.

LÄHTEET

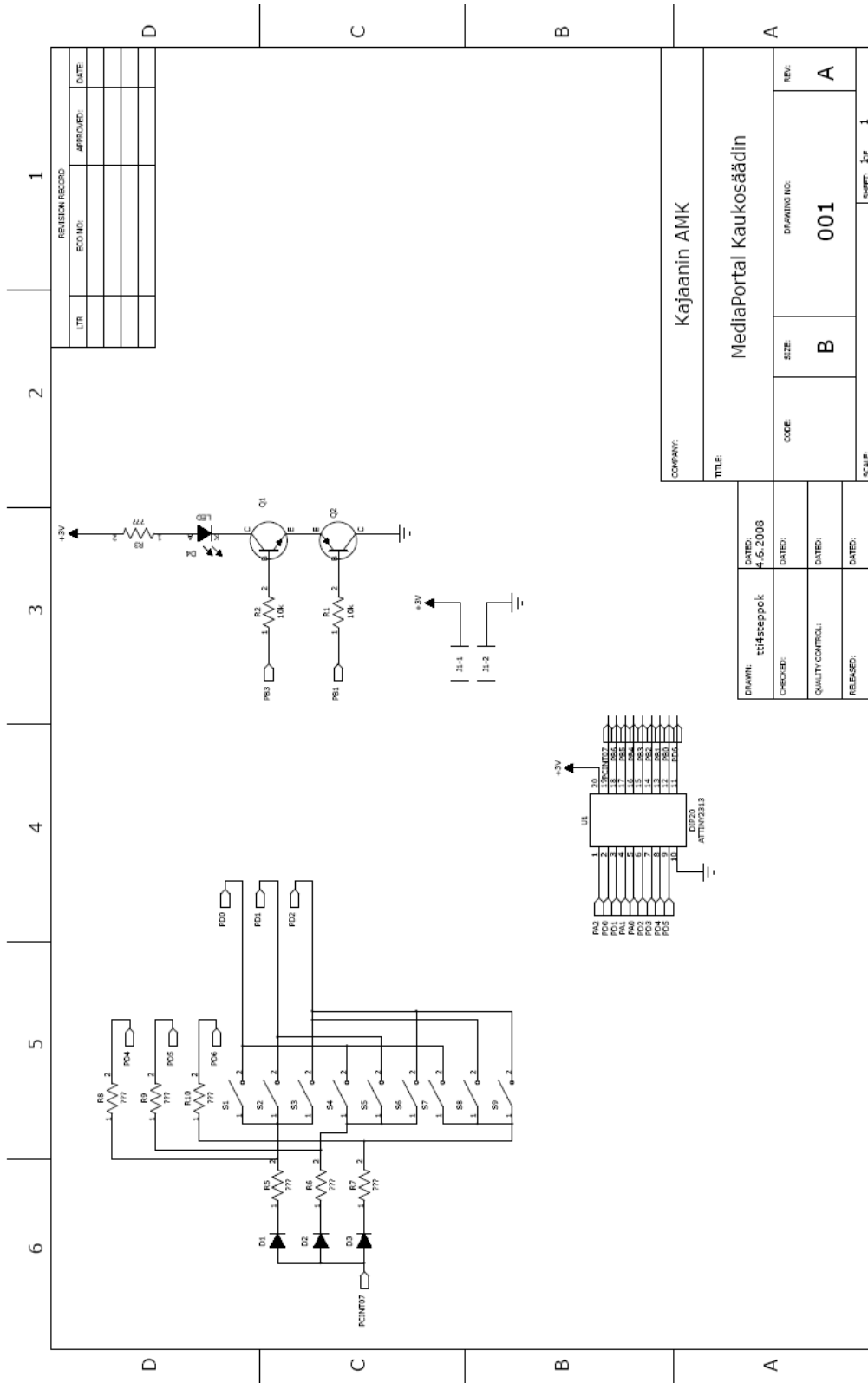
- 1 Inkinen, Manninen, Tuohi: Momentti 2 Insinöörifysiikka, ISBN 951-1-18457-1 s. 329-330
- 2 Granlund, Kaj: Tietoliikenne, Docendo, ISBN 951-846-133-3
- 3 MediaPortal - Free Media Center [WWW-Dokumentti] <<http://www.team-mediaportal.com/>> (Luettu 11.4.2008)
- 4 SB-Projects: IR remote control: Philips RC-5 [WWW-Dokumentti] <<http://www.sbprojects.com/knowledge/ir/rc5.htm>> (Päivitetty viimeksi 23.4.2008)
- 5 Innotech Systems Inc.: A Primer on Remote Control Technology [WWW-Dokumentti] <<http://www.innotechsystems.com/primer1.pdf>> (Luettu 30.4.2008)
- 6 Barr, Michael: Introduction to Pulse Width Modulation (PWM) [WWW-Dokumentti] <<http://www.netrino.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>> (Luettu 29.10.2008)
- 7 Haiko, Timo: Elektroniikka Simulaattori, WSOY, ISBN 951-0-25672-2
- 8 Tikkanen, Hannu: Piirilevysuunnitteluopas, ISBN 952-90-9423-1
- 9 BD135; BD137; BD139 NPN Power Transistors Product Specification Datasheet [WWW-Dokumentti] <<http://www.datasheetcatalog.org/datasheet/philips/BD139-16.pdf>> (Viimeksi muutettu 12.5.1999, luettu 24.4.2008)
- 10 Atmel ATTiny2313/V Preliminary [WWW-Dokumentti] <http://www.atmel.com/dyn/resources/prod_documents/DOC2543.PDF> (Luettu 20.4.2008)
- 11 Atmel application note AVR240: 4x4 Keypad - Wake-up on Keypress [WWW-Dokumentti] <http://www.atmel.com/dyn/resources/prod_documents/doc1232.pdf> (Luettu 23.4.2008)
- 12 Atmel AVR Studio Integrated Development Environment -tuote-esite [WWW-Dokumentti] <http://www.atmel.com/dyn/resources/prod_documents/doc2510.pdf> Luettu 30.10.2008
- 13 WinAVR: AVR-GCC for Windows [WWW-Dokumentti] <http://winavr.sourceforge.net/> Luettu 30.10.2008

14 MediaTexx Software Solutions [WWW-Dokumentti]
<<http://www.mediatexx.com/>> Luettu 10.11.2008

LIITTEIDEN LUETTELO

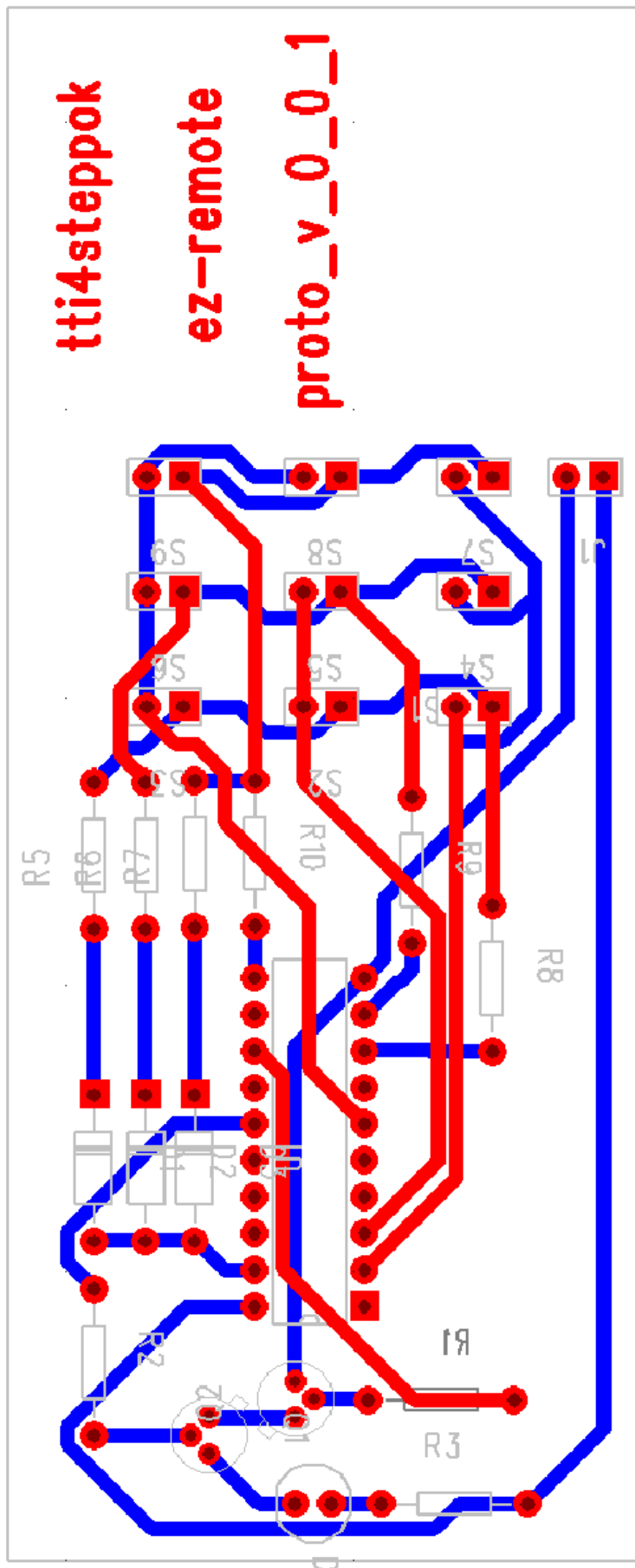
- Liite 1. Kytkäntäkaavio ja piirilevykuvat
- Liite 2. Kaukosäätimen ohjelmiston lähdekoodi

Kytöntäkaavio ja piirilevykuvat



REVISION RECORD		
LT#	ECO NO.	DATE

COMPANY: Kajaanin AMK	
TITLE: MediaPortal kaukosäädin	
DRAWN: tti4steppek	DATE: 4.6.2008
CHECKED:	DATE:
QUALITY CONTROL:	DATE:
RELEASED:	DATE:
CODE: B	DRAWING NO: 001
SCALE: 1	SHEET: 1 of 1



MCU Ohjelmakoodi

```

/*****
 * Helppokäyttöinen kaukosäädin MediaPortal-ympäristöön      *
 * Insinööriyö, Teppo Kauppinen TTI4S                        *
 * Kajaanin Ammattikorkeakoulu                               *
 *                                                            *
 * 15.5.2008 v. 0.0.6                                        *
 *****/

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>

#define CMDLENGTH 14

unsigned int countteri;
// Philips RC5 -komentosarjat      S S T ADDRESS-- C O M M A N D
static char chanUp[CMDLENGTH] = {1,1,0,0,1,0,1,1,1,0,0,0,0,0};
static char chanDwn[CMDLENGTH] = {1,1,0,0,1,0,1,1,1,0,0,0,0,1};
static char volUp[CMDLENGTH] = {1,1,0,0,1,0,1,1,0,1,0,0,0,0};
static char volDwn[CMDLENGTH] = {1,1,0,0,1,0,1,1,0,1,0,0,0,1};
static char standBy[CMDLENGTH] = {1,1,0,0,1,0,1,1,0,0,1,1,0,0};
static char enter[CMDLENGTH] = {1,1,0,0,1,0,1,1,1,1,0,1,1,0};
static char esc[CMDLENGTH] = {1,1,0,0,1,0,1,1,1,1,0,1,1,1};
static char menuL[CMDLENGTH] = {1,1,0,0,1,0,1,1,1,1,0,0,1,0};
static char menuR[CMDLENGTH] = {1,1,0,0,1,0,1,1,1,1,0,1,0,0};

SIGNAL(SIG_OUTPUT_COMPARE1A)
{
    countteri++;
}

SIGNAL(PCINT_vect)
{
    sleep_disable();
}

void viive(int ms)
{
    int i,j,k;

    for (i = 0; i < ms; i++)
    {
        for (j=0; j<20; j++)
            for(k=0; k<10000; k++);
    }
}

void sendCommandRC5(char *cCommandString)
{
    unsigned char cBitsSent = 0;
    while ( cBitsSent < CMDLENGTH )
    {
        if ( countteri == 64 )
        {
            cBitsSent++;
            countteri = 0;
        }
        if (countteri < 64)
        {
            if (cCommandString[cBitsSent] == 1)
            {
                if (countteri < 32)
                {

```

```

        PORTB |= (1<<PB1);
    }
    else
    {
        PORTB &= (0<<PB1);
    }
}
else
{
    if (countteri < 32)
    {
        PORTB &= (0<<PB1);
    }
    else
    {
        PORTB |= (1<<PB1);
    }
}
TCCR1B |= (1<<CS11);
}
}
TCCR1B &= (0<<CS11);
PORTB &= (0<<PB1)&(0<<PB3);
}

int main(void)
{
    DDRA=0xFF;
    PORTD=0xFF;
    PORTD=(0<<PD0)&(0<<PD1)&(0<<PD2);
    DDRD=(1<<PD0)|(1<<PD1)|(1<<PD2);
    PORTB=0x00;
    // OC1A, OC1B outputs
    DDRB = (1<<PB4)|(1<<PB3)|(1<<PB1);

    // TOP, set for 38kHz
    ICR1 = 26;
    // Center outputs
    OCR1A = 13;
    OCR1B = 13;
    // Timer 1 fast PWM mode 14
    // Clear on compare, set at TOP
    // /8 prescaler
    TCCR1A = (1<<COM1A1)|(1<<COM1B1)|(1<<WGM11);
    TCCR1B = (1<<WGM13)|(1<<WGM12)|(0<<CS11);
    TIMSK = (1<<OCIE1A);
    GIMSK = (1<<PCIE);
    PCMSK = (1<<PCINT7)|(1<<PCINT0)|(1<<PCINT3);
    countteri = 0;
    sei();
    while (1)
    {
        PORTD= ( (0<<PD0)&(0<<PD1)&(0<<PD2)
) |(1<<PD6)|(1<<PD5)|(1<<PD4);
        sleep_enable();
        sei();
        sleep_cpu();
        PORTD|= (1<<PD1)|(1<<PD2);
        if ((PIND&PD6) == 0)
        {
            sendCommandRC5(chanUp);
        }
        else if ((PIND&PD5) == 0)
        {
            sendCommandRC5(volUp);
        }
        else if ((PIND&PD4) == 0)

```

```
        {
            sendCommandRC5(menuL);
        }
PORTD&= (0<<PD1);
PORTD|= (1<<PD0) | (1<<PD2);
        if ((PIND&PD6) == 0)
        {
            sendCommandRC5(chanDwn);
        }
        else if ((PIND&PD5) == 0);
        {
            sendCommandRC5(volDwn);
        }
        else if ((PIND&PD4) == 0);
        {
            sendCommandRC5(menuR);
        }
PORTD&= (0<<PD2);
PORTD|= (1<<PD0) | (1<<PD1);
        if ((PIND&PD6) == 0)
        {
            sendCommandRC5(standBy);
            viive(113);
        }
        else if ((PIND&PD5) == 0);
        {
            sendCommandRC5(enter);
        }
        else if ((PIND&PD4) == 0);
        {
            sendCommandRC5(esc);
        }
    }
    return 0;
}
```