

Opinnäytetyö (AMK)

Tietoliikenne ja sähköinen kauppa

Elektroniikkasuunnittelu

2011

Allan Junkkila

USB-TUNNELMAVALO



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Elektroniikka | Elektroniikkasuunnittelu

Kesä 2011 | Sivumäärä 43

Ohjaaja: Tkt Timo Tolmunen

Allan Junkkila

USB-TUNNELMAVALO

Tässä työssä suunniteltiin ja rakennettiin tietokoneeseen liitettävä USB-tunnelmavallo. Suunnittelun lähtökohtana oli edullinen, matalatehoinen ja yksinkertainen USB-laite, joka voisi vaihtaa väriä.

Tunnelmavallo toteutettiin Atmelin AT90USB162-mikrokontrollerilla ja RGB-teholedillä. Mikrokontrollerin USB-kehyksenä käytettiin avoimen lähdekoodin perustuvaa LUFA-kehystä. Sillä kehitettiin tunnelmavallo USB-laitteeksi, jolla saatiin virtuaalinen sarjaportti toteutettua.

Tunnelmavallo vaihtaa väriä tietokoneohjelmiston avulla, ja käyttäjä voi valita värin värikartan avulla. Väri valitaan hiiren avulla, jolloin ohjelmisto lukee RGB-värin ja lähettää sen sarjaportin yli tunnelmavalolle. Ohjelmisto on kehitetty Windowsin sivupalkin pienoisohjelmaksi ja käyttää sarjaporttityhteytenä työssä kehitettyä ActiveX-komponenttia.

Tunnelmavallo rakennettiin piirilevylle ja suunnitteluun käytettiin EAGLE-ohjelmistoa. Teholähteenä toimii USB-portti ja teholedin ohjaus toteutettiin PWM-modulaatiolla.

Työssä saavutettiin toimiva USB-laite, joka toimii halutulla tavalla. Valo vaihtaa väriä ja reagoi käyttäjän valintoihin ripeästi. Työssä kehitettyä ohjelmistoa voidaan myös hyödyntää muissa sovelluksissa.

ASIASANAT:

USB, AVR

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Electronics | Electronics Design

Summer 2011 | Total number of pages 43

Instructor: Timo Tolmunen, D.Sc., Principal Lecturer

Allan Junkkila

USB MOOD LAMP

This thesis is about designing an USB Mood lamp. The design criteria were focused on making an low cost, low power and easy to use USB device that could change colour.

The device was implemented with an Atmel AT90USB162 microcontroller and a RGB power led. The microcontroller software was developed with the open source USB framework LUFA. With that the device was implemented as an USB device, that acts like an serial port.

To change the colour of the mood lamp, a software was developed for the windows operating system. The user uses the mouse to pick a colour of an colour map. Then the software reads the corresponding rgb value of the map and sends the information to the mood lamp by serial port. The software was developed as a windows sidebar gadget, and uses an ActiveX-object to facilitate the serial port communication.

The mood lamp was built on a custom made circuit board. The design was made with EAGLE PCB design software. The mood lamp uses USB as its main power source. PWM modulation was used to control the power led

A fully working USB device was achieved and the mood lamp works as planned. The device changes colour and reacts to the user input quickly. Some parts of the software is reusable and can be customized for new projects.

KEYWORDS:

USB, AVR

Sisältö

SISÄLTÖ	iv
SYMBOLIT JA LYHENTEET	v
1 JOHDANTO	1
2 TUNNELMAVALON MÄÄRITTÄMINEN	2
3 USB 2.0 -SPESIFIKAATIO	4
3.1 USB-väylä	5
3.1.1 Kaapeli	6
3.1.2 Teho	7
3.2 USB-protokolla	7
3.2.1 Paketit	8
3.2.2 Loppupiste	9
3.2.3 Putki	10
3.2.4 Tiedonsiirtomenetelmät	11
3.3 USB-laitekehys	12
3.3.1 Tilat	12
3.3.2 Luettelointiprosessi	14
3.3.3 Operaatiot	15
3.3.4 Pyynnöt	15
3.3.5 Kuvaukset	16
4 SUUNNITTELU	21
4.1 LUFA	21
4.2 Komponentit	22
4.3 Suunnitteluohjeet	23
4.4 Toteutus	26
5 OHJELMA	30
5.1 Mikrokontrolleri	30
5.1.1 Kuvaukset	30
5.1.2 Alustukset	31
5.1.3 Ohjelman kulku	32
5.2 Windows	33
5.2.1 Pienoisohjelma	33
5.2.2 ActiveX	34
6 MITTAUKSET JA TULOKSET	36
7 YHTEENVETO	41
LÄHTEET	42

SYMBOLIT JA LYHENTEET

$f_{clk_{I/O}}$	paikallisoskillaattorin taajuus
I	virta
R	resistanssi
U	jännite
ADC	analogia-digitaalimuunnin (Analog-to-Digital converter)
BCD	koodausjärjestelmä, jossa luku on koodattu neljäksi bitiksi (Binary Coded Decimal)
CRC	tiivistealgoritmi (Cyclic Redundancy Check)
CSS	porrastetut tyyliarkit (Cascading Style Sheets)
DFU	laitapäivitys USB:n avulla (Device Firmware Upgrade)
DPRAM	käyttömuisti (Dual-Ported Random Access Memory)
EEPROM	haihtumaton puolijohdemuisti (Electrically Erasable Programmable Read-Only Memory)
ESD	kipinäpurkaus (Electrostatic discharge)
ISP	piiri, joka voidaan ohjelmoida asennettuna järjestelmässä (In-System Programming)
HTML	hypertekstin merkintäkieli (Hypertext Markup Language)
LUFA	kevyt USB-kehysrakenne AVR-mikrokontrollereille (Lightweight USB Framework for AVRs)
MIPS	miljoonaa käskyä sekunnissa (Million Instructions per second)
MOSFET	kanavatransistori (Metal-Oxide-Semiconductor Field-Effect Transistor)
NRZI	binäärikoodaustapa (NON Return to Zero Invert)
RISC	suoritinarkkitehtuuri (Reduced instruction set computing)
SRAM	staattinen käyttömuisti (Static Random Access Memory)
PWM	pulssinleveysmodulaatio (Pulse-Width Modulation)

1 JOHDANTO

USB-laitteet ovat tänä päivänä varsin yleisiä ja markkinoilla on monenlaisia laitteita. Eri-
laisia USB:hen kytkettäviä valoja on olemassa, mutta ei väriä vaihtavia. Työn tavoitteena
olisi tehdä USB:hen kytkettävä tunnelmavalon, joka vaihtaisi väriä. Sen tulisi olla yksinker-
tainen, ja suunnittelussa tulisi painottaa edullisia komponenttivalintoja, jotta laitteen voisi
mahdollisesti toteuttaa kaupallisesti.

Tämän työn tarkoituksena on tutustua USB 2.0 -standardiin ja siihen, miten USB-laite
toteutetaan mikrokontrollerilla. Tarkoituksena olisi myös hyödyntää ilmaisia ohjelmia ja
apuvälineitä, ja toteuttaa näillä täysin toimiva USB-tuote.

Työn tavoitteena on tehdä tietokoneeseen kytkettävä tunnelmavalon, joka on halpa, pieni-
kokoinen ja helposti ohjattavissa. Sen tulisi myös olla helppokäyttöinen ja turvallinen.
Vaatimuksena on myös, ettei erillistä virtalähdettä tarvittaisi, vaan tunnelmavalon käyttäi-
si teholähteenä tietokonetta. Helppokäyttöisyyden kannalta vaatimuksena on myös, että
tunnelmavalon kytkeytyisi tietokoneeseen ulkoisesti.

Työn ensimmäisessä vaiheessa käydään laitteen spesifointi läpi ja tutustutaan USB 2.0 -stan-
dardiin laitekehittäjän näkökulmasta. Sen jälkeen perehdytään LUFAan ja yleisiin suun-
nitteluohjeisiin AVR-mikrokontrollereille. Viimeiseksi laite suunnitellaan ja suoritetaan
valovoimakkuuden mittauksia.

2 TUNNELMAVALON MÄÄRITTÄMINEN

Tämän päivän tietokoneissa ulkoisia liitäntämahdollisuuksia tarjoaa USB, FireWire sekä sarjaportti. Näistä vaihtoehdoista sarjaportti olisi helpoiten ohjelmoitavissa, mutta kyseistä porttia ei löydä enää jokaisesta tietokoneesta. FireWire-portti tarjoaisi sähköisiltä ominaisuuksilta parhaimman vaihtoehdon. FireWire kykenee syöttämään 30 V:n jännitteellä 1,5 A:n virtaa [1]. Vastaavasti USB-portti pystyy ainoastaan syöttämään 5 V:n jännitteellä 500 mA [2, s.178]. Toisaalta tietokoneiden emolevyiltä löytyy USB-portteja lukumääräisesti enemmän kuin FireWire- tai sarjaportteja. Tästä syystä valittiin tunnelmavalon liitynnäksi USB. Tätä valintaa puoltaa myös se, että USB on muutenkin hyvin yleinen oheislaitteiden liitäntämuoto.

USB-liitäntän toteuttamiseksi tunnelmavalon tarvitsee siis USB-kontrollerin. Tähän tarkoitukseen löytyy monia eri mikropiiriratkaisuja, esimerkiksi Atmel- tai Microchip-mikrokontrollerit sekä FTDI:n USB-mikropiirit. Yksinkertaisin ratkaisu olisi valita FTDI:n tuotevalikoimasta mikropiiri, sillä niille on valmiina kattava ajurikokoelma eri käyttöjärjestelmille [3]. Nämä mikropiirit on tarkoitettu lähinnä USB-yhteyden luomiseksi tietokoneen ja mikrokontrollerin välille, joten tällä valinnalla tarvittaisiin kaksi mikropiiriä. Tästä syystä Atmelin tai Microchipin USB-mikrokontrollerit tarjoavat edullisemman vaihtoehdon. Varjopuolena on oman ajurin kehittäminen tietokoneen käyttöjärjestelmälle. Molemmilta mikrokontrollerivalmistajalta löytyvät tähän tarkoitukseen valmiit kehysrakenteet USB:lle [4, 5]. Näiden kahden valmistajan 8-bittisten mikrokontrollereiden välillä ei ole merkittävää eroa hinnassa tai suorituskyvyssä. Siksi aikaisempien tuntemusten ja kokemusten perusteella, valittiin Atmelin AVR-mikrokontrolleri tähän työhön.

Atmelin megaAVR-tuoteperheestä löytyy laaja valikoima USB 2.0 -standardin mukaisia mikrokontrollereita, joiden merkittävimpana erona on flash-muistin koko. Tähän työhön riittää pieninkin koko, 8 kt, joten mikrokontrollerin valinta rajoittuu siten vain saatavuuteen ja hintaan. Näillä perusteluilla valittiin tähän työhön AT90USB162-mikrokontrolleri.

Valolähteen valikoima on hyvin rajallinen pienen käyttöjännitteen takia. Erittäin hyvä vaihtoehto, on LED-tekniikka. Se tarjoaisi modernin ja turvallisen valoratkaisun. Pienen kokonsa ja jännitevaatimuksen takia teholedi sopisi tähän työhön erittäin hyvin. Moniväri RGB-teholedi antaisi myös mahdollisuuden värien sekoittamiselle, mikä loisi edellytykset hyvälle tunnelmavalolle. RGB-ledejä löytyy lukuisia eri malleja ja teholuokkia. Yksinkertaisuuden kannalta valittiin käytettäväksi RGB-teholedi, jotta komponenttimäärä ja

piirilevyn koko pysyisivät kohtuullisena. Tämän työn valolähteeksi on valittu Multicompin OSW-8349 RGB-teholedi.

RGB-teholedi koostuu siis kolmesta ledistä, ja värien sekoittaminen vaatii sen, että jokaista väriä on kyettävä ohjaamaan. Tämän ohjauksen toteuttamiseen voidaan käyttää esim. pulssinleveysmodulaatiota tai säädettäviä vakiovirtalähteitä. Ohjausmuodoksi valittiin kumminkin pulssinleveysmodulaation, koska Atmelin AVR-mikrokontrollerista löytyy valmiina mahdollisuus käyttää sitä. Tämä myös vähentää ulkoisten komponenttien määrää.

Tunnelmavalon käyttö tietokoneelta tulisi olla helppoa ja vaivatonta. Käyttöohjelman saisi olla kevyt eikä tuhjata tietokoneen resursseja. Se voisi myös olla ulkonäöllisesti tyylikäs. Koska tunnelmavalon ohjauksessa ei tarvita suuria määriä dataa eikä asetuksia, riittäisi pienikin ohjelma tähän tarkoitukseen. Sen takia päätettiin kehittää käyttöohjelma Windowsin sivupalkin pienoisohjelmanä. Tämä tarkoittaisi sitä, että tunnelmavalon toimisi vain uusimmissa Windowsin versioissa, koska Windowsin sivupalkki toimii ainoastaan Windows 7- ja Windows Vista -käyttöjärjestelmissä. Windowsin sivupalkin pienoisohjelma on käytännössä HTML-sivu, jossa voidaan käyttää internetsivustoille tyypillisiä tekniikoita, kuten esim. CSS:ää, Javascriptiä ja ActiveX:ää.

Tunnelmavalon aloitettiin suunnitella Atmelin AT90USB162-mikrokontrollerin ympärille. Alkuvaiheessa tiedettiin kokonaisvirran määrä sekä käyttöjännite. Laite haluttiin pienikokoiseksi, joten pintaliitoskomponentteja pyrittiin käyttämään aina, kun oli mahdollista.

Laitteelta vaadittiin seuraavat spesifikaatiot:

– kokonaisteho	< 2,5 W
– valoteho	1,5 W
– käyttöjännite	4,75 – 5,25 V
– kokonaisvirta	< 500 mA
– piirilevy	Kaksipuolinen FR4
– fetit	N-kanavainen SOT-23 -pintaliitoskotelo
– vastukset	1206-pintaliitoskotelo toleranssi $\pm 1\%$
– kondensaattorit	1206-pintaliitoskotelo toleranssi $\pm 10\%$.

3 USB 2.0 -SPESIFIKAATIO

USB 2.0 on spesifioitu teollisuusstandardiksi tietokoneiden oheislaitteiden liitännäksi. USB:n tavoitteina on mm. helppokäyttöisyys, matala hinta sekä joustavuus. Täysi yhteensopivuus aikaisempien versioiden kanssa on myös olennainen osa USB 2.0 -spesifikaatiota. [2, s.11]

USB (Universal Serial Bus) on kaapeliväylä, joka tukee datan siirtoa isäntälaitteen ja monen samanaikaisesti kytketyn oheislaitteen välillä. USB 2.0 -standardilla on kolme eri nopeusluokkaa:

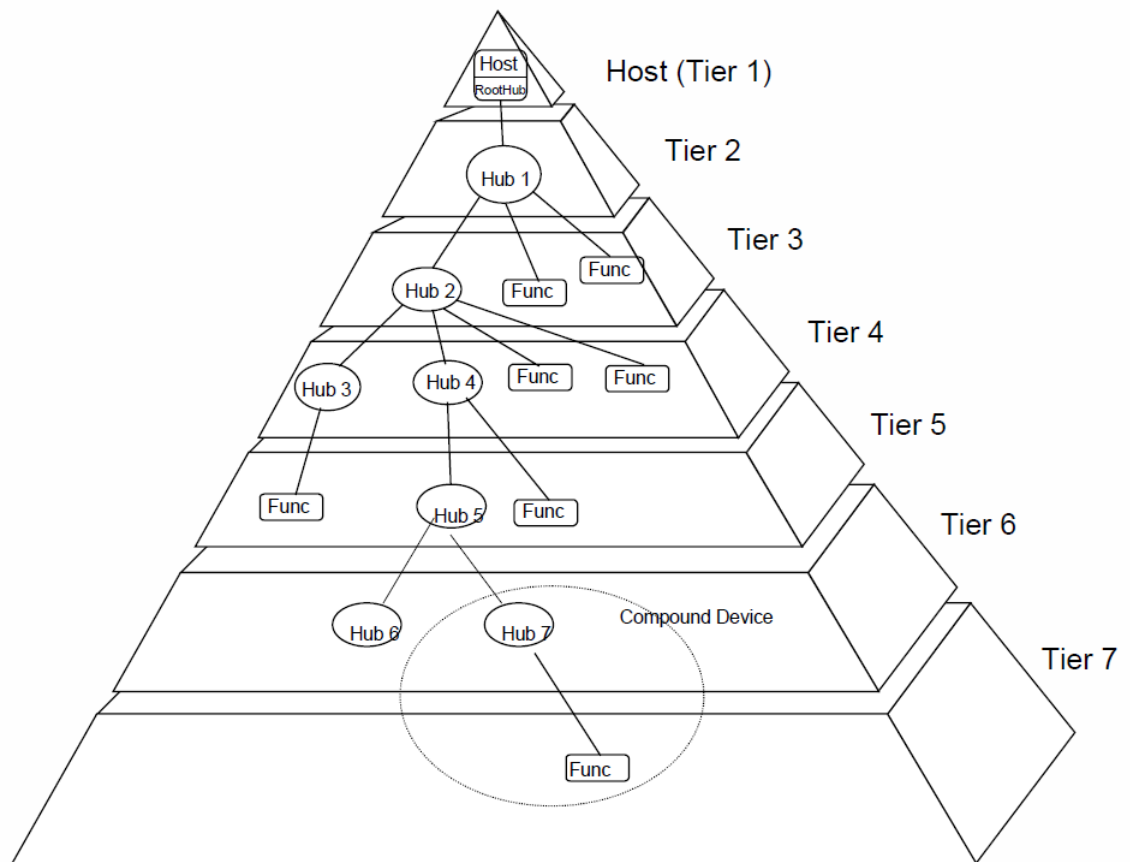
- low-speed: 10 – 100 kt/s
- full-speed: 500 kt/s – 10 Mt/s
- high-speed: 25 – 400 Mt/s.

Kiinnitetyt oheislaitteet jakavat USB-kaistan aikajakoisesti isäntälaitteen aikatauluttamana. USB-järjestelmä määrytyy kolmesta eri alueesta: USB-linkistä, USB-laitteesta ja USB-isännästä. Näistä USB-linkki määrittää sen, miten laite ja isäntä ovat yhteydessä toisiinsa. USB-linkki sisältää väylätopologian, kerroksien väliset suhteet, tietovirran mallit sekä USB-aikataulun. [2, s.12,15]

USB-laitteella on kaksi päämääräistä luokkaa: keskitin ja funktio. Keskitin koostuu kolmesta osasta: ohjaimesta, toistimesta ja tapahtumien kääntäjästä (engl. transaction translator). Keskitin tarkoittaa yksinkertaistaa USB-laitteiden kytkeytymistä isäntälaitteeseen. Keskitin muuttaa myös yhden liitinpuheen moneksi, ja liitinpuheita kutsutaan porteiksi. USB sallii myös keskitimien ketjuuttamista. Funktio on USB-laite, joka pystyy lähettämään ja vastaanottamaan dataa USB-väylän yli. Funktio voi olla joko erikseen tai yhdessä keskitimen kanssa. Mikäli samassa laitteessa on funktio ja keskitin, kutsutaan sitä yhdistelmälaitteeksi (engl. compound device). Jokainen funktio sisältää määrittämiset laitteen ominaisuuksille ja resurssien tarpeille. Ennen kuin funktio voidaan ottaa käyttöön, isäntälaitteen täytyy määrittää se luettelointiprosessissa (engl. enumerate) . [2, s.23–24]

3.1 USB-väylä

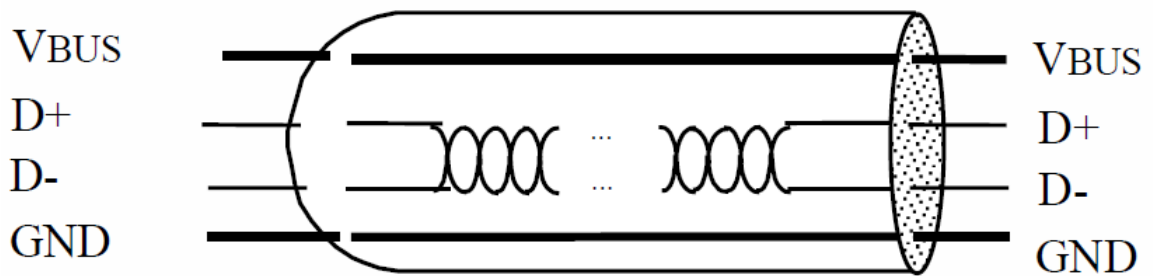
USB-laitteet kytkeytyvät väylässä porrastetussa tähtitopologiassa. Jokaisen tason (engl. tier) keskellä on aina USB-keskitin. Kuva 3.1 havainnollistaa USB-laitteiden kytkeytymistä isäntään. USB-keskittimet ja USB-laitteet kytkeytyvät toisiinsa aina suoraan. Tasojen enimmäismäärä on seitsemän, ja yhdistelmälaite vie kaksi tasoa. Mikäli yhdistelmälaite on tasolla seitsemän, sitä ei voida ottaa käyttöön. Tasolla seitsemän voidaan ainoastaan aktivoida funktioita. [2, s.16]



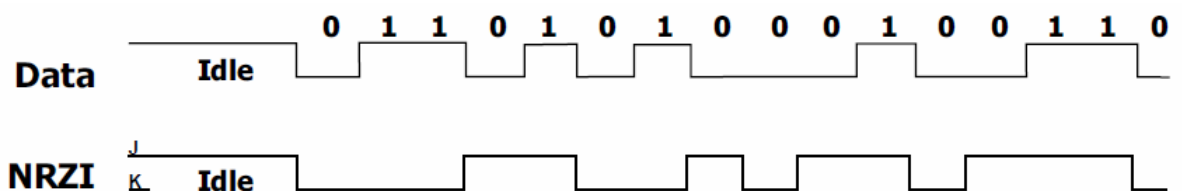
Kuva 3.1 USB-väylätopologia. Tason keskellä on keskitin ja tasojen/funktioiden välillä on suora yhteys. Tasojen enimmäismäärä on seitsemän ja yhdistelmälaiteelle on varattava kaksi tasoa [2, s.16].

3.1.1 Kaapeli

USB-kaapelissa on neljä johdinta: *VBUS*, *D+*, *D-* ja *GND*. Kuvassa 3.2 on tyypillisen kaapelin rakenne, missä *D+* ja *D-* ovat kierretty toistensa ympärille häiriöiden vähentämiseksi. *VBUS*:n nimellisedännite on +5 V maahan nähden, mutta edännitteen sallittu vaihtelualue on 4,75 – 5,25 V. Datan differentiaalijohtimet, *D+* ja *D-*, toimivat 3,6 V:n nimellisedännitteellä ja niiden impedanssi on 90 Ω . Johtimissa virta kulkee toisessa aina päinvastaiseen suuntaan ja datan siirto tapahtuu J- ja K-, tai 1- ja 0-merkkien avulla. Kun halutaan lähettää J, virta kulkee *D+*-suuntaan ja K:n tapauksessa *D-*-suuntaan. USB käyttää tiedonsiirrossa NRZI-linjakoodausta ja bittitiedennystä (engl. bit stuffing). Kuvassa 3.3 on esimerkki tiedonsiirrosta. NRZI-linjakoodauksessa 1 merkitään samalla tasolla ja 0 tason vaihtumisella. Varmistaakseen, että signaalitasot vaihtuvat riittävän usein, käyttää USB tiedonsiirrossa bittitiedennystä. Jos lähetettävässä datassa on kuusi peräkkäistä 1, lisää USB-lähetin ylimääräisen 0:n dataan ennen NRZI-linjakoodausta. Tämä varmistaa sen, että signaalitasot vaihtuvat riittävän usein. [2, s.120,157,178]



Kuva 3.2 USB-kaapelin rakenne. Kaapeli koostuu suojavaipasta, kierretystä signaaliparista *D+* ja *D-* sekä johtimista *VBUS* ja *GND* [2, s.17].



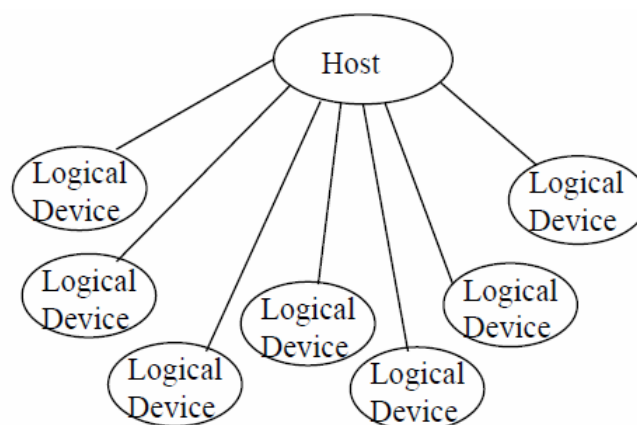
Kuva 3.3 NRZI-linjakoodaus. Lähetettävässä datassa 1 ilmaistaan saman tason avulla ja 0 tason vaihtumisella [2, s.157].

3.1.2 Teho

USB-laitteilla on kaksi eri teholuokkaa: väylätehoinen tai omatehoinen (engl. bus- and self-powered). Väylätehoiset laitteet ottavat kaiken tehonsa USB-portista ja omatehoisella laitteella on ulkoinen teholähde. Virransyöttö on määritelty 100 mA:n virtayksiköiden avulla, ja suurin sallittu määrä on 5 virtayksikköä. Tämä on absoluuttinen maksimi, eikä aikakeskiarvo. Laite voidaan määrittellä joko matalatehoiseksi (engl. low-power) tai korkeatehoiseksi (engl. high-power) riippuen siitä, kuinka monta virtayksikköä laite vie. Matalatehoinen on määritelty 1:ksi ja korkeatehoinen 5 virtayksiköksi. Kaikki USB-laitteet ovat lähtökohtaisesti matalatehoisia, ja siirtyminen korkeatehoiseksi tapahtuu ohjelmiston avulla. Omatehoiset laitteet voivat viedä korkeintaan yhden virtayksikön USB-portista. USB-laite ei saa missään tapauksessa syöttää virtaa VBUS-linjaan, eikä virrankulutus saa ylittää yhtä virtayksikköä, ennen kuin isäntälaitte on määritellyt laitteen. [2, s.171]

3.2 USB-protokolla

Vaikka USB-laitteet kytkeytyvät fyysisesti tähtitopologiassa, kommunikoi isäntälaitte jokaisen loogisen USB-laitteen kanssa ikään kuin, se olisi kytketty suoraan isäntään. Kuva 3.4 havainnollistaa loogisten laitteiden kytkeytymistä isäntään.



Kuva 3.4 Looginen USB-väylä. Isäntä näkee jokaisen loogisen laitteen suorakytkentänä, vaikka laitteet ovat fyysisesti kytketty tähtitopologiassa [2, s.30].

USB:n tiedonsiirtoprotokolla perustuu kiertokyselyyn (engl. poll) ja isäntä aloittaa jokaisen tiedonsiirtotapahtuman. Tiedonsiirto tapahtuu kehyksissä ja kehyksien pituudet riippuvat

siitä, onko kyseessä full-/low- vai high-speed laite. Full-/low-speed kehys on $1 \text{ ms} \pm 500 \text{ ns}$ ja high-speed mikrokehys on $125 \mu\text{s} \pm 62,5 \text{ ns}$ pitkä [2, s.159].

3.2.1 Paketit

Yhdessä tiedonsiirtotapahtumassa on yleensä kolme pakettia. Ensimmäinen paketti kertoo tiedonsiirtotapahtuman tyyppin sekä suunnan. Tätä pakettia kutsutaan token-paketiksi. Sen jälkeen tulee varsinainen data-paketti ja viimeisenä on vastaanottajan handshake-paketti, joka ilmaisee oliko lähetys onnistunut. [2, s.18–19]

Paketin lähetyksessä, vähiten merkitsevä bitti lähetetään ensimmäiseksi. Paketin ensimmäinen kenttä on sync-kenttä, jonka tarkoitus on synkronisoida vastaanottoiiri paikallisen kellon kanssa. Sen jälkeen tulee pakettitunnus (engl. packet identifier), PID-kenttä. Se on jaettu neljään osaan: token, data, handshake ja special. PID-kentässä selviää paketin tyyppi ja paketin suunta. [2, s.195]

Token-paketti

Taulukossa 1 on token-paketin kentät. Token-paketti sisältää PID-kentän lisäksi: address-, endpoint- ja CRC-kentän. ADDR-kentässä on funktion osoite ja ENDP-kentässä on loppupisteen numero. Token-paketti käyttää 5-bittistä CRC-tarkisteavainta. [2, s.199]

Taulukko 1 Token-paketin kentät [2, s.199].

Kenttä	PID	ADDR	ENDP	CRC5
Bittejä	8	7	4	5

Data-paketti

Data-paketti sisältää PID-, DATA- ja CRC16-kentän. Data-kentän suurin koko riippuu nopeusluokasta. Low-speed laitteella suurin datamäärä on 8 tavua, full-speed laitteella 1023 tavua ja high-speed laitteella 1024 tavua. Data-paketti käyttää 16-bittistä CRC-tarkistusavainta. [2, s.206]

Taulukko 2 Data-paketin kentät [2, s.206].

Kenttä	PID	DATA	CRC16
Bittejä	8	0–8192	16

Handshake-paketti

Handshake-paketti koostuu ainoastaan PID-kentästä. [2, s.206]

Taulukko 3 Handshake-paketin kenttä [2, s.206].

Kenttä	PID
Bittejä	8

3.2.2 Loppupiste

Looginen laite koostuu itsenäisistä loppupisteistä, ja jokainen tiedonsiirtotapahtuma on yhden loppupisteen ja isännän välillä. Yhden loppupisteen ja isännän välistä yhteyttä kutsutaan putkeksi. Jokaisen USB-laitteen on toteutettava loppupisteessä nolla ns. oletus ohjausputken (engl. default control pipe). Tämä putki on kaksisuuntainen ja on olemassa heti, kun laite kytketään USB-väylään. USB-järjestelmän ohjelmisto käyttää tätä ohjausputkea alustamaan sekä yleisesti ohjaamaan USB-laitetta. Ohjausputken lisäksi, laite voi määritellä lisäputkia. Nämä ovat pääsääntöisesti yhdensuuntaisia, joko IN (laitteelta isännälle) tai OUT (isännältä laitteelle). Low-speed laitteella voi olla loppupiste nollan lisäksi kaksi loppupistettä. Full-speed laitteella voi olla loppupiste nollan lisäksi 30 loppupistettä, 15 sisään ja 15 ulos. Loppupiste määräytyy seuraavista asioista:

- Väylän käyttötarve/latenssi vaatimukset
- Kaistanleveyden tarve
- Loppupisteen numero
- Virheenkäsittelyn tarve
- Lähettevän/vastaanottevan paketin maksimikoko
- Tiedonsiirron suunta
- Tiedonsiirtomenetelmä.

Kaikki loppupisteet, paitsi nolla, ovat tuntemattomassa tilassa eikä niitä voida käyttää, ennen kuin isäntälaitte on määritellyt ne luettelointiprosessissa. Kun USB-laite kiinnitetään

USB-väylään, saa se järjestelmältä ainutlaatuisen laiteosoitteen. Tämän ja loppupistenumeron perusteella voidaan yksilöllisesti viitata mihin tahansa loppupisteeseen. [2, s.33–34]

3.2.3 Putki

USB-putki on assosiaatio laitteen loppupisteen ja isäntälaitteen ohjelmiston välillä, ja putki edustaa ominaisuutta siirtää dataa näiden välillä. USB ei tulkitse siirtämäänsä dataa millään tavalla ja USB-putkella on kaksi tiedonsiirtotapaa, jotka ovat toisensa poissulkevia: Stream ja Message. [2, s.34]

Stream

Stream-tiedonsiirtotavan data-paketin sisällöllä ei ole USB:n mukaista rakennetta. Stream-putki on aina yhdensuuntainen ja järjestelmä olettaa, että datavirta on vuorovaikutuksessa ainoastaan yhden laitteen kanssa. Data virtaa putkessa peräkkäisessä järjestyksessä, fifo-periaatteen mukaan. Stream-putki tukee bulk-, isochronous- ja interrupt-tiedonsiirtomenetelmää. [2, s.35–36]

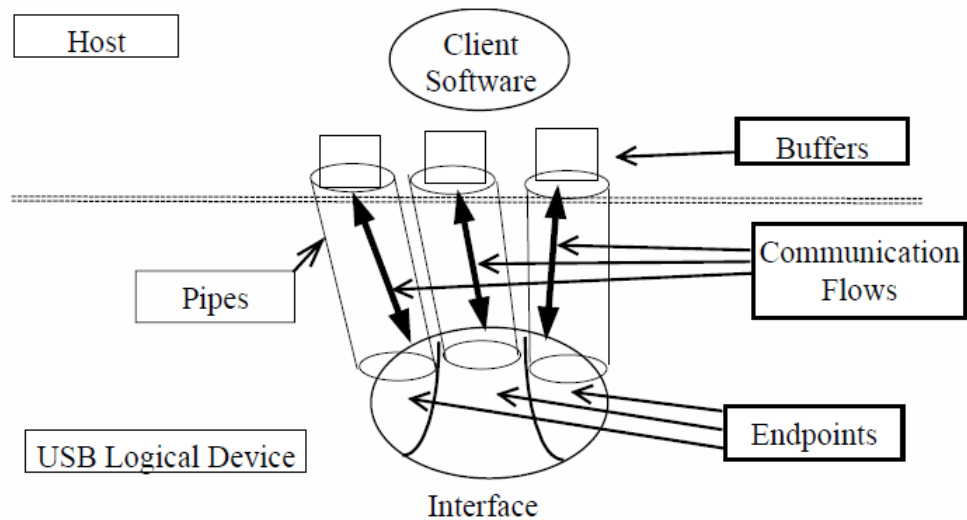
Message

Message-putki aloittaa tiedonsiirtotapahtuman lähettämällä pyynnön (token-paketti) loppupisteelle. Tämän jälkeen data-paketti lähetetään pyynnön mukaisessa suunnassa. Viimeiseksi seuraa status-vaihe. Message-putki voi olla kaksisuuntainen, mutta pääasiassa yhdensuuntainen. Loppupisteessä nolla oleva ohjausputki on message-putki. [2, s.36]

USB-järjestelmä takaa sen, että message-putkessa on vain yksi pyyntö kerrallaan. USB-laitteelta vaaditaan, että laite palvelee vain yhtä pyyntöä message-putkea kohti kerrallaan. Laite voi säädellä informaation kulkua data- ja status-vaiheen avulla. [2, s.36]

Message-putki vaatii sen, että loppupisteen suunta on määritelty sekä sisään että ulos. Message-putki ei voi olla assosioitu eri loppupisteiden kanssa kuhunkin suuntaan. Message-putki tukee vain control-tiedonsiirtomenetelmää. [2, s.36]

Kuvassa 3.5 on yhteenvedona hahmotelma, miten loppupisteet ja putket ovat osana USB-laitteen ja ohjelmiston tietovirtamalla.



Kuva 3.5 USB-tietovirran malli [2, s.33].

3.2.4 Tiedonsiirtomenetelmät

USB siirtää dataa isännän ja laitteen välillä putkien avulla. Message-putki noudattaa USB:n mukaista rakennetta, mutta itse data-paketin sisältö voi olla laitekohtainen. Datan on noudettava tiettyjä pakettikokoja, mutta paketin muotoilu ja tulkinta on vapaasti valittavissa. USB-laitteen suunnittelija voi valita loppupisteen ominaisuuksia, ja kun putki on alustettu ohjelmiston ja laitteen välillä, pysyy se kutakuinkin muuttumattomana putken eliniän aikana. [2, s.36–37]

USB määrittelee neljä siirtomenetelmää seuraavasti:

- Control-tiedonsiirtomenetelmä: Voidaan käyttää ainoastaan message-putken kanssa ja noudattaa USB:n sekä pyyntö-, data-, statusvaiheita. Tiedonsiirto on aina isännän aloittama, ei jatkuvaa ja järjestelmä voi rajoittaa väylän käyttö- ja kaistavaatimuksia. [2, s.38]
- Isochronous-tiedonsiirtomenetelmä: USB-järjestelmä takaa siirtomenetelmälle loppupisteen määrityksen mukaiset kaista- ja latenssivaatimukset. Siirtomenetelmälle luvataan myös kiinteää tiedonsiirtonopeutta, kunhan putkeen syötetään jatkuvasti dataa. Tiedonsiirtovirheiden kohdalla järjestelmä ei yritä uudelleenlähettämistä. [2, s.44]
- Interrupt-tiedonsiirtomenetelmä: Laitteille jotka lähettävät tai vastaanottavat dataa harvoin, mutta tarvitsevat säännöllisin aikavälin palvelua. USB-järjestelmä takaa

siirtomenetelmälle säännöllisen palveluvälin sekä tiedonsiirtovirheiden sattuessa pakettien uudelleenlähettämistä. [2, s.48]

- Bulk-tiedonsiirtomenetelmä: Laitteille jotka siirtävät suuria määriä dataa epäsäännöllisesti. Tiedonsiirtotapahtuma voi käyttää mitä tahansa vapaana olevaa kaistanleveyttä. USB-järjestelmä antaa siirtomenetelmälle pääsyn USB-väylälle, vapaan kaistanleveyden perusteella. USB-järjestelmä uudelleenlähettää epäonnistuneet tiedonsiirrot ja takaa sen, että data menee perille, mutta ei kaistanleveyttä tai latenssia. [2, s.52]

Bulk-tiedonsiirtomenetelmä tapahtuu vain, kun USB-väylällä on vapaana kaistaa. Siksi järjestelmässä missä on vähän kaistaa vapaana, tapahtuu tiedonsiirto pätkittäin suhteellisen pitkän ajanjakson aikana. [2, s.52]

3.3 USB-laitekehys

USB-laite voidaan jakaa kolmeen kerrokseen:

- Alin kerros on USB-väylän rajapinta, joka huolehtii pakettien lähettämisestä sekä vastaanottamisesta.
- Keskimmäinen kerros reitittää paketit rajapinnan ja loppupisteiden välillä.
- Ylin kerros on se toiminto, jonka USB-laitteen on tarkoitus antaa.

USB-laitekehys on keskimmäisen kerroksen yleiset ominaisuudet ja toiminnot, joilla ylin kerros viestittää USB-väylän avulla isännän kanssa. [2, s.239]

3.3.1 Tilat

USB-laitteella on monta toiminnallista tilaa ja nämä tilat määräytyvät attribuuteista. USB-laitteen kiinnittyessä USB-väylään, saa se Attached-attribuuttin. USB-laitteen teholähteenä on kaksi mahdollisuutta, joko USB-väylä tai sitten ulkoinen lähde. Laite saa Powered-attribuuttin, kun *VBUS*-linja on kytketty laitteeseen. Jos laite käyttää ulkoista lähdettä, niin sen ei katsota olevan powered, ennen kuin laite on kiinnitetty ja *VBUS*-linja on kytketty laitteeseen. [2, s.242]

Taulukko 4 USB-laitteen tilat. Tilat määräytyvät laitteen attribuuttien perusteella [2, s.241].

Attached	Powered	Default	Address	Configured	Suspended	Tila
Ei	-	-	-	-	-	Laite ei ole kiinnitetty. Muilla attribuuteilla ei ole väliä.
Kyllä	Ei	-	-	-	-	Laite on kiinnitetty mutta ei powered.
Kyllä	Kyllä	Ei	-	-	-	Laite on kiinnitetty mutta ei resetoitu.
Kyllä	Kyllä	Kyllä	Ei	-	-	Laite on kiinnitetty ja resetoitu, mutta ei ole saanut osoitetta. Laite vastaa oletusosoitteessa.
Kyllä	Kyllä	Kyllä	Kyllä	Ei	-	Laite on kiinnitetty, resetoitu, saanut osoitteen mutta ei ole määritelty.
Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Ei	Laite on kiinnitetty, resetoitu, saanut osoitteen ja määritelty. Isäntä voi nyt käyttää laitteen funktiota.
Kyllä	Kyllä	-	-	-	Kyllä	Laite on vähintään kiinnitetty ja powered, mutta ei ole nähnyt väylällä siirtotapahtumia 3 ms:n aikana. Laite on ehkä saanut osoitteen ja määritelty käyttöä varten, mutta isäntä ei voi käyttää laitteen funktiota.

Vaikka laite on päällä (Powered-attribuutti), se ei saa vastata väylän siirtotapahtumiin, ennen kuin väylä on resetoitunut laitteen. Sen jälkeen laite on osoitettavissa oletusosoitteessa. Laitteella on resetoinnin jälkeen Default-attribuutti ja seuraavaksi isäntä aloittaa määrittämisen. Ensimmäiseksi isäntä antaa laitteella osoitteen, jolloin laite saa Address-attribuuttin. Sen jälkeen isäntä suorittaa luettelointiprosessin loppuun, ja laite saa Configured-attribuuttin. [2, s.242–243]

USB-laite täytyy automaattisesti mennä keskeytettyyn tilaan, jos väylällä ei ole ollut siirto-tapahtumia tietyn ajanjakson aikana. Laite säilyttää kaiken määrittämisensä sekä osoitteen keskeytyksen aikana. Keskeytyksen aikana laitteella on Suspended-attribuutti. [2, s.243]

Taulukossa 4 on tiivistelmä miten attribuutit määräävät laitteen tilan.

3.3.2 Luetteloointiprosessi

Kun USB-laite kiinnitetään tai irrotetaan, tapahtuu ns. väylän luetteloointiprosessi (engl. bus enumerate). Sen tarkoitus on tunnistaa ja käsitellä laitteen tilamuutoksia. Kun laite kiinnittyy USB-väylään, tapahtuu seuraavaa:

1. USB-Keskittin välittää isännälle, että laite on kiinnittynyt porttiin. Laitteessa on nyt virta päällä ja isäntä poistaa portin pois käytöstä, luetteloinnin ajaksi.
2. Isäntä selvittää keskittimeltä tarkalleen, mitä tapahtui.
3. Isäntä tietää, mitä tapahtui ja missä portissa laite on kiinni. Isäntä odottaa vähintään 100 ms, että laite ehtii käynnistyä ja laitteen jännite vakavoituu. Sen jälkeen isäntä lähettää port enable- ja reset-käskyt.
4. Keskittin suorittaa resetoinnin ja sen jälkeen portti on taas käytössä. Nyt USB-laite on default-tilassa, eikä saa viedä enempää virtaa kuin 100 mA. USB-laite vastaa nyt oletusosoitteessa.
5. Isäntä lähettää nyt laitteelle uniikin osoitteen.
6. Isäntä lukee ohjausputken avulla mikä on putken suurin pakettikoko
7. Isäntä lukee järjestyksessä laitteen loppupisteiden kuvaukset. Tämä saattaa kestää useita millisekunteja.
8. Kuvauksien perusteella isäntä antaa laitteelle määritellyn tilan. Loppupisteet saavat kuvauksien mukaiset ominaisuudet ja laite voi nyt viedä haluamansa määrän virtayksiköitä. USB-laitteen näkökulmasta laite on valmiina käyttöä varten.

Kun USB-laite irrotetaan, lähettää keskittin siitä tiedon isännälle. Tämä poistaa portin käytöstä ja päivittää paikallisen topologiansa. [2, s.243–244]

3.3.3 Operaatiot

USB-laitteen on tuettava seuraavia monta eri toimenpidettä, jotka ovat pakollisia kaikille USB-laitteille. Sen on tuettava dynaamista kiinnittymistä ja irrottautumista. USB-laitteen on pystyttävä kiinnittämään tai irrottamaan USB-väylästä milloin tahansa. Kiinnitystapah- tumassa laitteen on vastattava oletusosoitteessa eikä laitteella saa olla määritelmiä tehtynä. Sen jälkeen, kun laite on kiinnittynyt USB-väylään, on sen tuettava osoitteen asettamista. Isäntä määrittää laitteelle osoitteen ja siirtyy sitten määrityksiensä asettamiseen, missä isäntä suorittaa luettelointiprosessin. [2, s.244–245]

Laitteen on tuettava tehonhallintaa, ja luettelointiprosessissa isäntä arvioi, onko väylällä riittävästi tehoa tarjolla. Mikäli on, laite voi viedä kuvauksen mukaisen määrän virtayksikköjä. Ennen määritysten asettamista, laitteen on rajoitettava virrankulutusta yhteen virtayksikköön. [2, s.244–245]

Laitteen on tuettava tiedonsiirtoa, ja datapaketteja on voitava välittää isännän ja loppupis- teiden välillä, jollakin neljästä siirtomenetelmästä. Tiedonsiirron lisäksi laitteelta vaaditaan pyyntöjen käsittelemistä. Laitteen täytyy käsitellä ja vastata pyyntöihin ajoissa. Osoitteen asettamiseen sekä tavallisissa laitepyynnöissä, missä ei ole data-vaihetta, laitteen on vas- tattava 50 ms:ssa. Pyyntöissä jossa on data-vaihe, niihin laitteen on vastattava 500 ms:n kuluttua. Muutoin pyyntöjen yläraja on 5 sekuntia. Mikäli laite vastaanottaa virheellisen pyynnön, tai se on sopimaton nykyisten asetusten kanssa, on laitteen vastattava lähettämällä STALL PID -pakettitunnus seuraavassa datapaketissa tai status-vaiheessa. [2, s.244–245]

3.3.4 Pyyntöt

Kaikki USB-laitteet vastaavat isännän pyyntöihin ohjausputken kautta, joka käyttää Cont- rol-tiedonsiirtomentelmää ja on assosioitu loppupiste nollan kanssa. Pyyntö ja vastaus tapahtuu setup-paketissa ja paketin koko on 8 tavua. Taulukossa 5 on setup-paketin rakenne ja taulukossa 6 on tiivistelmä pyynnöistä. [2, s.248]

Taulukko 5 Setup-paketin rakenne [2, s.248].

Offset	Kenttä	Koko	Arvo	Kuvaus
0	bmRequestType	1	Bittikartta	Kenttä kertoo tiedonsiirron suunnan sekä vastaanottajan.
1	bRequest	1	Arvo	Pyynnön tyyppi (ks. taulukko 6).
2	wValue	2	Arvo	Käytetään syöttämään parametreja pyyntöjen yhteydessä.
4	wIndex	2	Indeksi/Offset	Käytetään syöttämään parametreja pyyntöjen yhteydessä. Yleensä käytetään ilmaisemaan tiettyä loppupistettä.
6	wLength	2	Määrä	Datan määrä, jos pyynnössä siirretään dataa.

Taulukko 6 USB-laitepyynnöt [2, s.251].

Nimi	Koodi	Kuvaus
Get_Status	0	Palauttaa laitteen tilan.
Clear_Feature	1	Poistaa tietyn toiminnon (DEVICE_REMOTE_WAKEUP, ENDPOINT_HALT tai TEST_MODE).
	2	Varattu myöhempää käyttöä varten.
Set_Feature	3	Asettaa tietyn toiminnon (DEVICE_REMOTE_WAKEUP, ENDPOINT_HALT tai TEST_MODE).
	4	Varattu myöhempää käyttöä varten.
Set_Address	5	Asettaa laitteelle osoitteen.
Get_Descriptor	6	Palauttaa kuvauksen, jos se on olemassa.
Set_Descriptor	7	Tämä on valinnainen pyyntö, jolla voidaan päivittää tai lisätä kuvauksia.
Get_Configuration	8	Palauttaa laitteen määrittelyn arvon. Jos se on 0, niin laite ei ole määritelty
Set_Configuration	9	Asettaa laitteelle määrittelyn tilan.
Get_Interface	10	Palauttaa vaihtoehdoisen asetuksen liitännälle.
Set_Interface	11	Asettaa liitännälle vaihtoehdoisen asetuksen tai palauttaa oletusasetuksen.
Synch_Frame	12	Asettaa ja palauttaa loppupisteen synkronisointikehyksen.

3.3.5 Kuvaukset

USB-laite ilmoittaa ominaisuuksistaan kuvauksilla (engl. descriptors). Kuvaus on data-rakenne ja niitä on yhteensä seitsemän kappaletta. Kuvauksen ensimmäisenä kenttänä on kuvauksen koko tavuissa. Sen jälkeen tulee kenttä, missä on kuvauksen tyyppi. Nämä kaksi kenttää ovat yhteisiä kaikille kuvauksille, jonka jälkeen tulee kenttiä kuvauksen tyyppin perusteella. Joissakin kohdissa kuvaus voi sisältää merkkijonoja (engl. strings), joita

ihmisen on helppo lukea, mutta ovat valinnaisia. Jos kuvaus ei tue merkkijonoja, täytyy merkkijonokentät asettaa nolllaksi. USB-laitteella on seuraavat kuvaukset:

- Device: kuvailee yleisiä ominaisuuksia laitteesta.
- Device_Qualifier: kuvailee miten high-speed laitteen asetukset muuttuisivat, jos laite toimisi alemmalla nopeudella.
- Configuration: Kuvaile laitteen astukset tietylle määritelmälle.
- Other_Speed_Configuration: kuvailee miten high-speed laite toimisi alemalla nopeus luokilla, ja kuvauksen rakenne on identtinen configuration-kuvauksen kanssa.
- Interface: kuvailee liitännän ominaisuuksia, kykyjä ja loppupisteiden lukumäärää.
- Endpoint: kuvailee loppupisteen ominaisuuksia ja vaatimuksia.
- String: Valinnainen merkkijono, joka sisältää tekstin UNICODE-koodattuna.

Seuraavaksi esitellään tässä työssä käytettyjä kuvauksia.

Device-kuvaus

Device-kuvaus sisältää USB-laitteen yleisiä ominaisuuksia ja määrytyksiä. USB-laitteella voi olla vain yksi device-kuvaus. Kenttä bcdUSB sisältää USB-version BCD-koodattuna, ja tästä selviää laitteen nopeusluokka. High-speed laitteella versionumero on 2.0 ja full-/low-speed laitteella 1.1. Kentässä bMaxPacketSize0 on loppupiste nollan suurin pakettikoko. High-speed laitteella sen on oltava 64-tavuinen, mutta alemmilla nopeuksilla se voi olla: 8-, 16-, 32- tai 64-tavuinen. Kenttä bNumConfigurations sisältää sen, kuinka monta eri määriteltyä tilaa laitteella on kyseisellä nopeusluokalla. Taulukossa 7 on device-kuvauksen rakenne. [2, s.261–263]

Taulukko 7 Device-kuvauksen rakenne [2, s.262–263].

Offset	Kenttä	Koko	Arvo	Kuvaus
0	bLength	1	Numero	Kuvauksen koko tavuissa.
1	bDescriptorType	1	Vakio	Kuvauksen tyyppi.
2	bcdUSB	2	BCD	BCD-koodattuna USB-spesifikaation numero.
4	bDeviceClass	1	Luokka	USB-IF:n määräämä luokan koodi.
5	bDeviceSubClass	1	Aliluokka	USB-IF:n määräämä aliluokan koodi.
6	bDeviceProtocol	1	Protokolla	USB-IF:n määräämä protokollan koodi.
7	bMaxPacketSize0	1	Numero	Loppupiste nollan pakettikoko.
8	idVendor	2	ID	Vendor-tunnus.
10	idProduct	2	ID	Valmistajan tuotetunnus.
12	bcdDevice	2	BCD	Laitteen julkaisunumero BCD-koodattuna.
14	iManufacturer	1	Index	Merkkijonon indeksi.
15	iProduct	1	Index	Merkkijonon indeksi.
16	iSerialNumber	1	Index	Merkkijonon indeksi.
17	bNumConfigurations	1	Numero	Määritelmien kokonaismäärä.

Configuration-kuvaus

Kuvaus sisältää tietoa yhdestä tietyistä laitteen määrittämisestä, ja laitteella voi olla monta määrittystä. Kenttä bConfiguration sisältää numeron, joka syötettynä parametrina SetConfiguration()-pyynnön yhteydessä tekee sen, että laite ottaa configuration-kuvauksen kuvailemat määrittäykset käyttöön. Taulukossa 8 on configuration-kuvauksen rakenne. [2, s.264–265]

Taulukko 8 Configuration-kuvauksen rakenne [2, s.265–266].

Offset	Kenttä	Koko	Arvo	Kuvaus
0	bLength	1	Numero	Kuvauksen koko tavuissa.
1	bDescriptorType	1	Vakio	Kuvauksen tyyppi.
2	wTotalLength	2	Numero	Kuvauksen kokonaispituus.
4	bNumInterfaces	1	Numero	Liitäntöjen määrä.
5	bConfigurationValue	1	Numero	Määritelmän arvo
6	iConfiguration	1	Indeksi	Merkkijonon indeksi
7	bmAttributes	1	Bittikartta	Laitteen toiminnot
8	bMaxPower	1	mA	Laitteen virrankulutus 2mA:n yksiköissä (50 = 100 mA).

Interface-kuvaus

Jokaisella määrittelyllä voi olla yksi tai useampi liitäntä, ja interface-kuvaus kuvailee yhtä tiettyä liitäntää yhdessä tietyssä määrittelyssä. Yhteen liitäntään voi kuulua monta loppupistettä ja ja näillä voi olla vaihtoehtoisia määritelmiä. Liitännän oletusasetus on nolla, jota voidaan muuttaa SetInterface()-pyynnön avulla. Taulukossa 9 on interface-kuvauksen rakenne. [2, s.267–268]

Taulukko 9 Interface-kuvauksen rakenne [2, s.268–269].

Offset	Kenttä	Koko	Arvo	Kuvaus
0	bLength	1	Numero	Kuvauksen koko tavuissa.
1	bDescriptorType	1	Vakio	Kuvauksen tyyppi.
2	bInterfaceNumber	1	Numero	Liitännän numero.
3	bAlternateSettings	1	Numero	Vaihtoehtoisen asetuksen numero.
4	bNumEndpoints	1	Numero	Loppupisteiden määrä.
5	bInterfaceClass	1	Luokka	USB-IF:n määräämä liitännän luokan koodi.
6	bInterfaceSubClass	1	Aliluokka	USB-IF:n määräämä liitännän aliluokan koodi.
7	bInterfaceProtocol	1	Protokolla	USB-IF:n määräämä protokollan koodi.
8	iInterface	1	Indeksi	Merkkijonon indeksi.

Endpoint-kuvaus

Jokaisen liitännän loppupisteellä on oma kuvauksensa, missä on tieto loppupisteen ominaisuuksista. Tämän tiedon perusteella isäntä määrittää tarvittavan kaistanleveyden loppupisteelle. Taulukossa 10 on endpoint-kuvauksen rakenne. [2, s.269–271]

Taulukko 10 Endpoint-kuvauksen rakenne [2, s.270–271].

Offset	Kenttä	Koko	Arvo	Kuvaus
0	bLength	1	Numero	Kuvauksen koko tavuissa.
1	bDescriptorType	1	Vakio	Kuvauksen tyyppi.
2	bEndpointAddress	1	Loppupiste	Loppupisteen osoite.
3	bmAttributes	1	Bittikartta	Loppupisteen ominaisuudet.
4	wMaxPacketSize	2	Numero	Suurin pakettikoko.
6	bInterval	1	Numero	Kiertokyselyn aikaväli.

String-kuvaus

String-kuvaukset ovat valinnaisia, ja jos laite ei tue merkkijonoja täytyy kaikki merkkijonoindeksit muissa kuvauksissa asettaa 0:ksi. Taulukossa 11 on string-kuvauksen rakenne. [2, s.272–273]

Taulukko 11 String-kuvauksen rakenne [2, s.274].

Offset	Kenttä	Koko	Arvo	Kuvaus
0	bLength	1	Numero	Kuvauksen koko tavuissa.
1	bDescriptorType	1	Vakio	Kuvauksen tyyppi.
2	bString	N	Numero	Merkkijono UNICODE-koodattuna.

4 SUUNNITTELU

4.1 LUFA

LUFA (Lightweight USB Framework for AVR) on kevyt USB-laitekehys, joka on avoimeen lähdekoodin perustuva. Sitä voidaan käyttää ilmaiseksi, mutta se on lisensoitu MIT-lisenssin alle. Tämä tarkoittaa sitä, että mikäli on käyttänyt LUFA:n lähdekoodia, täytyy siitä olla merkintä laitteen dokumentaatiossa. Tästä velvoitteesta voidaan vetäytyä vastuusta maksamalla kertaussumma 1500 \$. [6, 7]

LUFA on tehty Atmelin AVR-mikrokontrollereille, joilla on USB-valmius. Se on kirjoitettu C-ohjelmointikielellä, ja se on suunnattu ilmaiselle AVR-GCC kääntäjälle. LUFA sisältää kattavat demonstraatiot eri USB-oheislaitteesta, esim (hiiri, massamuisti, tulostin). [6]

LUFA:lla on monta etua verrattuna Atmelin USB-pinoon:

- Lisenssi: LUFA:n MIT-lisenssi on hyvin salliva, kun taas Atmelin lisenssillä on rajoituksia miten ja missä sitä voidaan käyttää. LUFA:n lisenssi on sopiva sekä kaupalliseen että ei-kaupalliseen käyttöön.
- Demot ja projektit: LUFA sisältää kattavan valikoiman valmiita demoja sekä projekteja. Tämä nopeuttaa kehitystyötä ja oppimista.
- Laitteiston tuki: LUFA on kehitetty siten, että se tukee mahdollisimman hyvin eri AVR-mikrokontrollereita. Tämä mahdollistaa sen, että mikrokontrollerin vaihtaminen vaatii hyvin vähän toimenpiteitä. Atmelin USB-pino vaatii kirjastoja AVR-mikrokontrollerisarjan perusteella, mikä hankaloittaa siirtymistä toiseen sarjaan.
- Ohjelmakoodin koko: LUFA on kirjoitettu alhaalta ylöspäin niin, käyttäen hyvää ja selkeää koodia, että se kääntyisi optimaalisesti konekielelle.

LUFA:lla on myös verkossa laaja käyttäjäkunta, joilta saa kehitysapua tarvittaessa. LUFA on suosittu USB-laitekehys, ja mm Arduino Uno, joka on kaupallinen AVR-kehitysalusta, käyttää sitä. [8]

LUFA sisältää kattavan kirjaston USB-laitekehystä varten. AVR-projekti tehdään LUFA:n ympärille, ja siinä on kehitetty kaikki tarvittavat funktion USB yhteyden toteuttamiseksi.

LUFA:ssa on valmiina suuri määrä USB-kuvauksia, joita muokkaamalla voidaan soveltaa uusissa USB-laitteissa. Tässä työssä on käytetty LUFA 101122 -versiota.

4.2 Komponentit

Työ keskittyy kahden komponentin ympärille: AT90USB162-mikrokontrollerin ja OSW-8349-teholedin.

AT90USB162 on 8-bittinen AVR-mikrokontrolleri, joka perustuu RISC-arkkitehtuuriin. Se on koteloitu TQFP-32-pintaliitoskoteloon, ja sillä on yleisiä ominaisuuksia:

- 125 käskyä, joista moni suoriutuu yhden kellonjakson aikana.
- 16 MIPS 16 MHz:n kellotaajuudella.
- 16 kt Flash-ohjelmamuistia.
- 512 t SRAM-muistia.
- 512 t EEPROM-muistia.
- 16-bittinen ajastin ja yksi 8-bittinen ajastin.

16-bittinen ajastin on oleellinen osa PWM-modulaatiota, sillä ajastimella on kolme PWM-kanavaa. PWM:n resoluutio voi olla joko 8-, 10- tai 12-bittinen. AT90USB162 on ohjelmoitavissa sekä ISP:n että DFU:n avulla, ja Flash-muistin luvataan kestävän 10 000 kirjoitus ja tyhjennys-kierrosta. [9]

AT90USB162 on täysin USB 2.0 -yhteensopiva, ja sillä on seuraavia tärkeitä USB-ominaisuuksia:

- Full-speed toiminta: tiedonsiirtonopeus 12 Mbit/s.
- 176 t USB DPRAM -muistia loppupisteiden käyttöä varten.
- Loppupiste nollan Control-tiedonsiirtomenetelmää varten. Tukee 8 – 64 tavuisia pakettikokoja.

- 4 ohjelmoitavaa loppupistettä, niiden pakettikoko voi olla 8 – 64 t ja suuntana IN tai OUT.

AT90USB162:lla on myös sisäinen 3,3 V:n regulaattori USB-kaapelin $D+$ - ja $D-$ -diferentiaalijohtimia varten. Tätä regulaattoria voidaan myös käyttää mikrokontrollerin teholähteenä. [9]

OSW-8349 on moniväri RGB-teholedi, joka koostuu kolmesta ledistä. Niitä voidaan ohjata erikseen ja niiden allonpituudet ovat seuraavia:

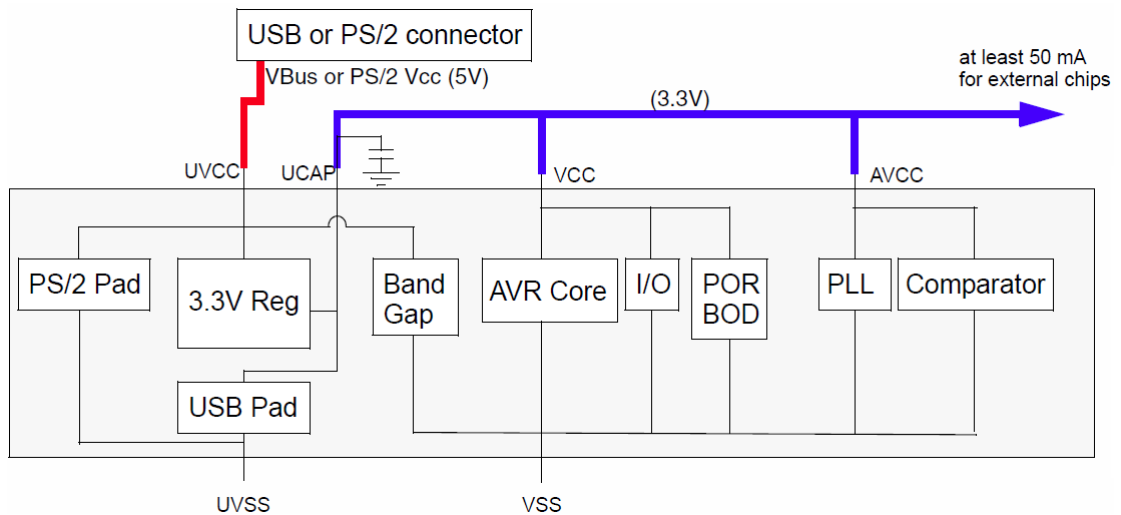
- punainen: 620 – 630 nm
- vihreä: 520– 535 nm
- sininen: 460 – 475 nm.

Ledit kestävät 350 mA jatkuvaa dc-virtaa ja pulsittain 1000 mA. Pulssin jakso täytyy olla alle 100 μ s ja pulssisuhde 0,005. [10]

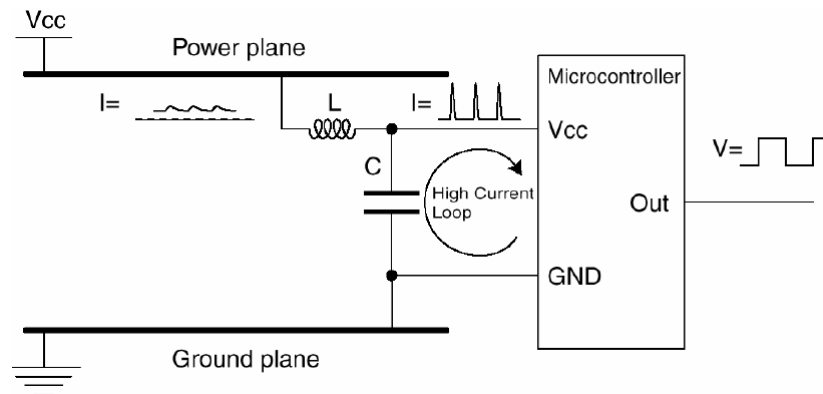
4.3 Suunnitteluohjeet

Teholähden kytkentä

AT90USB162:n sisäistä regulaattoria, joka on tarkoitettu $D+$ - ja $D-$ -nastoille, voidaan myös käyttää mikrokontrollerin teholähteenä. Regulaattori vaatii kondensaattorin, ja datalehden suositusarvo on $1 \mu\text{F} \pm 10 \%$. Kuvassa 4.1 on sisäisen regulaattorin esimerkkikytkentä ja kuvassa 4.2 on suositeltu ohituskondensaattoreiden kytkentätapa. Ohituskondensaattori tulisi kytkeä mahdollisimman lähelle käyttöjännitenastaa, ja mikäli nastoja on enemmän, tulisi jokaisella olla oma ohituskondensaattorinsa. Sen arvoksi suositellaan 100 nF. [9, 11]



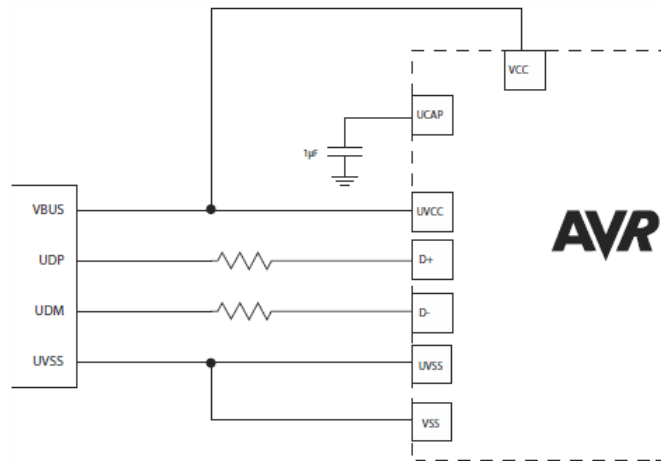
Kuva 4.1 Sisäisen regulaattorin esimerkkikytkentä [9].



Kuva 4.2 Ohituskondensaattorin esimerkkikytkentä [11].

USB:n esimerkkikytkentä

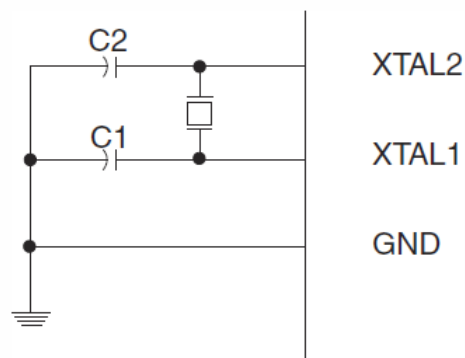
USB:n kytkeminen AVR-mikrokontrollerille on varsin yksinkertaista. Kuvassa 4.3 on datalehden esimerkkikytkentä. Differentialijohtimet vaativat sarjavastuksen, jonka arvo on $22 \Omega \pm 5 \%$. Datalehti suosittelee, että johdinvedot piirilevyllä olisivat mahdollisimman lähellä toisiaan, mahdollisimman lyhyitä ja välttäisivät läpivientejä. VBUS-linjaan suositellaan myös $10 \mu\text{F}$:n kondensaattoria. [9]



Kuva 4.3 USB:n esimerkkikytkentä [9].

Kiteen kytkentä

AT90USB162 tarvitsee kiteen toimiakseen USB-kontrollerina. Kuvassa 4.4 on datalehden esimerkkikytkentä kiteen kytkemiselle. Kide tarvitsee rinnalleen kondensaattoreita, ja niiden arvo on datalehden suosituksen mukaan oltava 12 – 22 pF. Kondensaattorit olisi hyvä pitää samanarvoisina, ja niiden optimaalinen arvo riippuu kiteestä, hajakapasitanssin määrästä ja elektromagneettisesta kohinasta ympäristössä. [9]

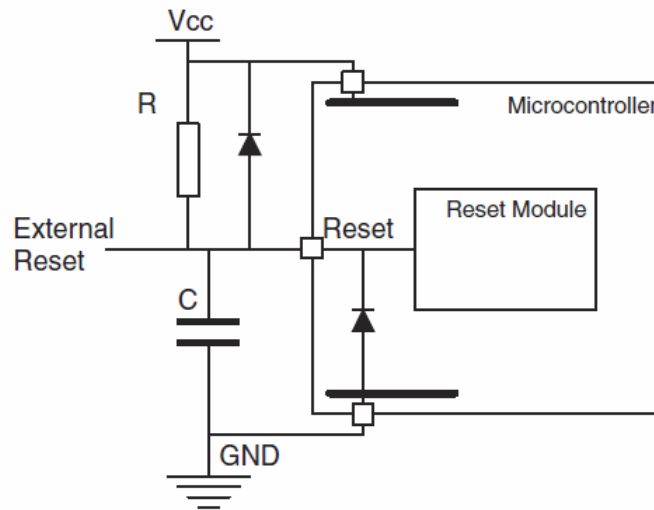


Kuva 4.4 Kiteen esimerkkikytkentä [9].

Reset-nastan suojaaminen

Kuvassa 4.5 on suositukset Reset-nastan suojaamiseen. Reset-nastaan pitäisi kytkeä vastus sekä kondensaattori hyvän ulkoisen reset-toiminnan saavuttamiseksi. Tyypillisiä arvoja

ovat $10\text{ k}\Omega$ sekä $4,7\text{ nF}$. Jos halutaan korkea ESD-suojaus, voidaan kytkeä diodi Reset-nastasta käyttöjännitteeseen. [12]



Kuva 4.5 Reset-nastan suojaaminen [12].

4.4 Toteutus

Tunnelmavaloa ruvettiin suunnittelemaan edellä olevien ohjeiden ja spesifioinnin perusteella. Piirilevyksi tulisi kaksipuoleinen FR4-kuparilevy.

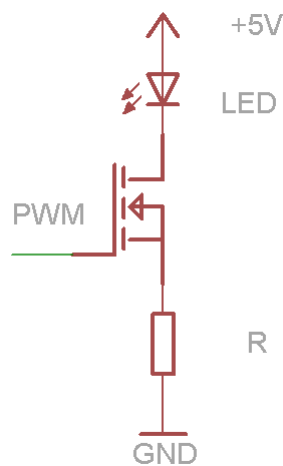
Laitteelle valittiin B-mallinen USB-liitin USB-standardin mukaan [2, s.85]. Liitin on läpiladottava, ja se sijoitettiin piirilevyn reunaan. VBUS-linjaan kytkettiin $10\ \mu\text{F}$:n elektrolyyttikondensaattori suositusten mukaan. Siihen kytkettiin myös sarjaan pintaliitossulake. Mikrokontrollerin UCAP-nastan kytkettiin suositusten mukaan $1\ \mu\text{F}$:n elektrolyyttikondensaattori. USB-liittimen ja mikrokontrollerin $D+$ - ja $D-$ -nastojen väliin kytkettiin $22\ \Omega$:n pintaliitosvastukset sarjaan kuvan 4.3 mukaan.

Mikrokontrollerille valittiin rinnakkaisresonanssiin tarkoitettu 16 MHz :n HC49/4H kvartsikide, joka on pintaliitosmallinen. Kide kytkettiin mikrokontrollerin XTAL1- ja XTAL2-nastoihin kuvan 4.4 osoittaman tavan mukaan. Kiteelle valittiin myös 18 pF :n pintaliitoskondensaattorit.

Seuraavaksi suunniteltiin reset-piiri kuvan 4.5 mukaan. Ylösvetovastukseksi valittiin suositusten mukaan $10\text{ k}\Omega$:n pintaliitosvastus. Kondensaattoriksi valittiin $4,7\ \mu\text{F}$:n pintalii-

toskondensaattori. Reset-nastasta kytkettiin käyttöjännitteeseen diodikaksikko, joka oli SOT-23 pintaliitoskotelossa.

Teholedin yksittäisen värikanavan ohjaus toteutettiin MOSFET-transistorilla ja vastuksella. Kuvassa 4.6 on suunniteltu piiri ledin-ohjaamiseen. Ledin läpi kulkeva virta riippu siten vastuksen R suuruudesta ja kytkinfetin sisäisestä $R_{DS(ON)}$ resistanssista. Vastus sijoitettiin maan ja fetin lähteen väliin, jotta virran mittaaminen vastuksen jännitteen avulla olisi helppoa. Tällöin fetin kynnyksjännite muodostui tärkeäksi kriteeriksi. Kytkinfetiksi valittiin N-kanavainen FDV303N-fetti. Sillä on hyvin matala kynnyksjännite, $U_{GS(th)} < 1,5 \text{ V}$, ja $R_{DS(ON)}$ on riittävän pieni [13].



Kuva 4.6 Suunniteltu piiri ledin ohjaamiseen.

Vastuksen R :n arvon laskeminen ei ole ihan suoraviivaista. Koska ledin kynnyksjännite U_{th} sekä fetin $R_{DS(ON)}$ muuttuvat molemmat virran funktiona, eikä datalehdistä löydy näille tarkkaa arvoa. Vastuksen jännite U_R on kumminkin lausuttavissa kaavan (1) mukaan.

$$U_R = U - U_{th} - U_{DS} \quad (1)$$

$$U_{DS} = I_D \cdot R_{DS(ON)} \quad (2)$$

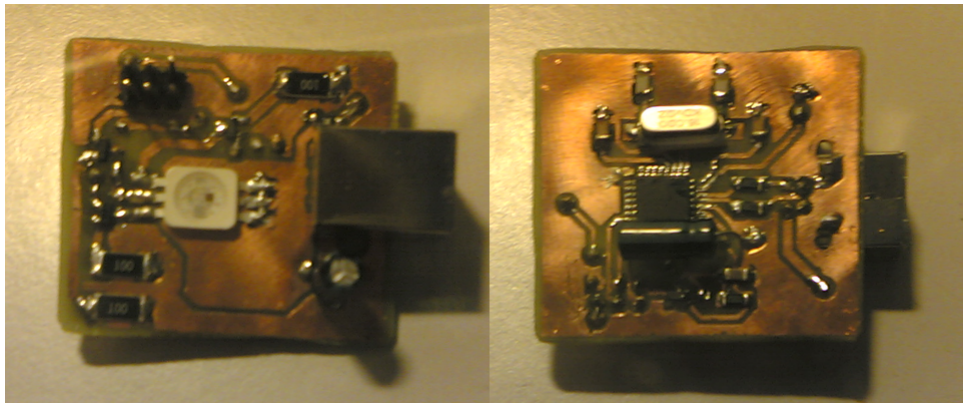
Sijoittamalla kaava (1) Ohmin lakiin (3) saadaan vastukselle R likimääräinen arvo (4).

$$U_R = I_D \cdot R \quad (3)$$

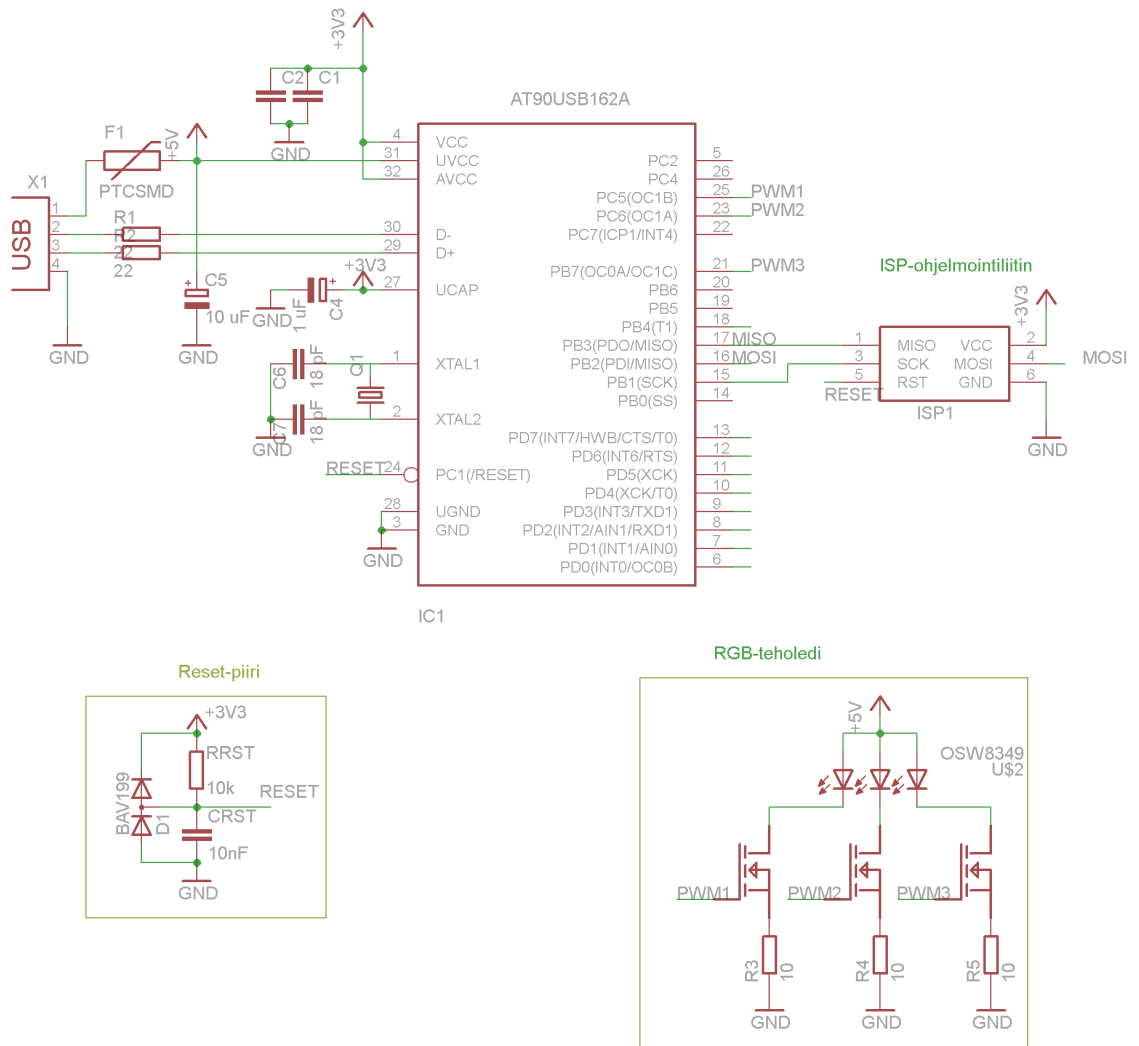
$$R = \frac{U - U_{th} - U_{DS}}{I_D} \quad (4)$$

Olettamalla että ledien kynnyksjännite on 3 – 4 V ja $R_{DS(ON)}$ noin 2 Ω , kun $I_D = 150\text{mA}$, voidaan arvioida R :n suuruudeksi noin 5 – 11 Ω . Vastukseksi R valittiin 2512-pintaliitoskotelossa oleva 10 Ω :n vastus, jonka nimellisteho on 1 W.

Piirilevyn layout tehtiin siten, että alapuolelle sijoitettiin mikrontrolleri ja siihen liittyvät oheiskomponentit. Yläpuolelle sijoitettiin ISP-ohjelmointiliitin, teholedi ja PWM-ohjausfetit. Kuvassa 4.7 on valmis piirilevy, ja kuvassa 4.8 on lopullinen kytkentä.



Kuva 4.7 Rakennettu tunnelmavalo. Vasemalla yläpuoli ja oikealla alapuoli.



Kuva 4.8 USB-tunnelmavalon kytkentä.

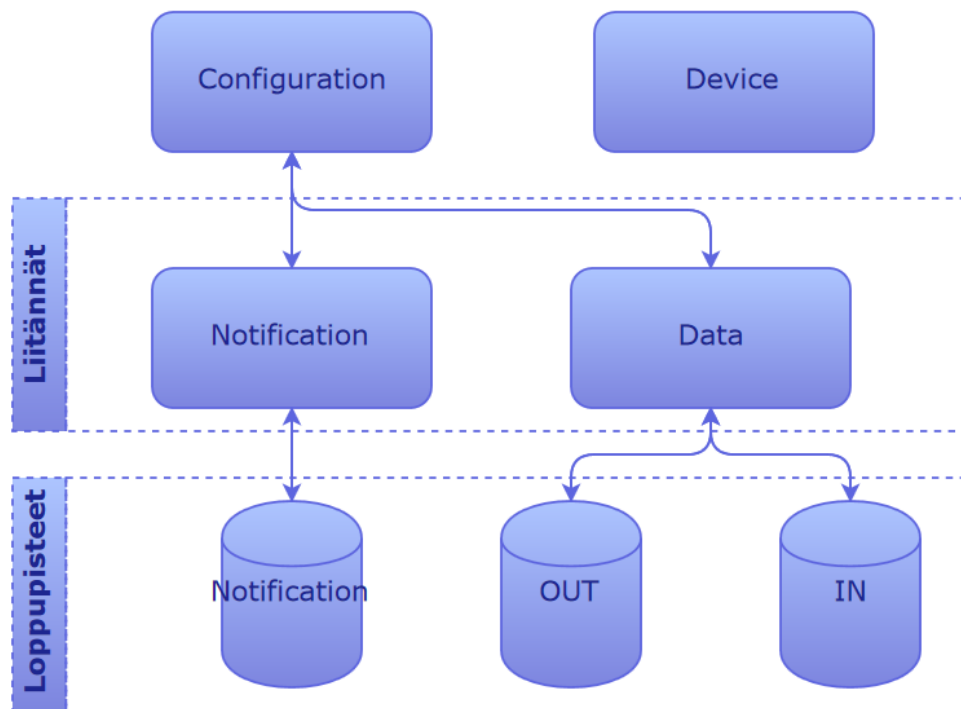
5 OHJELMA

USB-tunnelmavalon ohjelma koostuu kahdesta osasta: mikrokontrolleri- ja Windows-osasta.

5.1 Mikrokontrolleri

5.1.1 Kuvaukset

Tämä osa koostuu suurimmaksi osasta LUFA:sta ja erillisestä PWM-moduulista. Ohjelmaa lähdettiin tekemään LUFA:n VirtualSerial-demosta. Siinä mikrokontrolleri määrittää USB-laitteeksi ja sille tehdään 2 liitännää ja 3 loppupistettä. Kuvassa 5.1 on hahmotelma kuvauksista.



Kuva 5.1 USB-tunnelmavalon liitännät ja loppupisteet.

Kuvaukset vaativat hyvin vähän muokkaamista ja seuraavaksi käydään läpi oleelliset osat device- ja configuration-kuvauksista. Kuvauksien koko ja tyyppi tulevat LUFAssa automaattisesti makrojen avulla, joten niistä ei tarvitse välittää.

Device-kuvaus

Device-kuvauksen `bcdUSB` asetetaan versioksi 1.10 ja `bDeviceClass` asetetaan 02h:ksi. Tämä on USB-IF:n CDC (Communication Device Class) luokka eikä sillä ole aliluokka- tai protokolla-koodia. Vendor-tunnukseksi asetetaan Atmelin tunnus, 03EBh, ja julkaisunumeroksi 0.1. Viimeiseksi määritellään kaksi merkkijonoindeksiä: 01h ja 02h. Nämä ovat `iManufacturer` sekä `iProduct`. Näihin String-kuvauksiin tulee tekstit 'Allan Junkkila' ja 'Tunnelmavallo'. LUFA täyttää muut kentät automaattisesti kääntäjälle annetuilla lipuilla.

Configuration-kuvaus

LUFA:ssa configuration-kuvaus on määritelty struktuuriksi, joka sisältää kaikki liitännät ja loppupisteet kuvan 5.1 mukaan. Aluksi määritellään struktuurin configuration-kuvaukseen laite väylätehoiseksi `bmAttributes`-kenttään, ja virtayksiköiden määräksi 5 `bMaxPower`-kenttään.

Notification-liitännällä on `bInterfaceNumber`-kentässä sekä `bAlternateSettings`-kentässä 0. Liitännän numero on siis 0 ja eikä sillä ole vaihtoehtoisia asetuksia. Sillä on yksi loppupiste ja seuraavat tunnukset: luokkakoodi 02h, aliluokka 07h ja protokolla 01h. Notification-loppupisteen tiedonsiirtomenetelmäksi määritellään Interrupt, kiertokyselyn aikaväliksi 256 ms ja paketin kooksi 8 t.

Laitteen toinen liitäntä on tiedonsiirtoa varten. Sillä on kaksi loppupistettä, joilla on samat asetukset: Bulk-tiedonsiirtomentelmä, kiertokyselyn aikaväli 1 ms ja paketin koko 16 t. Liitännällä ei ole vaihtoehtoisia asetuksia ja liitännän numero on 1.

5.1.2 Alustukset

Ennen kuin USB:ta tai PWM-modulaatiota voi käyttää, täytyy ne alustaa. USB:n alustus tapahtuu LUFA:n toimesta ja siinä asetetaan mikrokontrollerin rekisterit ja loppupisteet kuvauksien mukaan. Myös USB:n toimivuudelle on tärkeää, että mikrokontrollerin USB-kellon vaihelukko on asetettu oikein. Alustuksen yhteydessä poistetaan käytöstä Watchdog-ajastin.

AT90USB162:lla on monta eri PWM-moodia, jotka soveltuvat eri käyttötarkoituksiin. Tähän sovellukseen valittiin PCPWM (Phase Correct PWM), suuren resoluution takia. Re-

soluution voi laskea kaavalla (5) ja taajuus kaavalla (6), joissa TOP on vapaasti valittavissa ja N on esiskaalaajan arvo [9].

$$R_{PCPWM} = \frac{\log(TOP + 1)}{\log(2)} \quad (5)$$

$$f_{PCPWM} = \frac{f_{clk_I/O}}{2 \cdot N \cdot TOP} \quad (6)$$

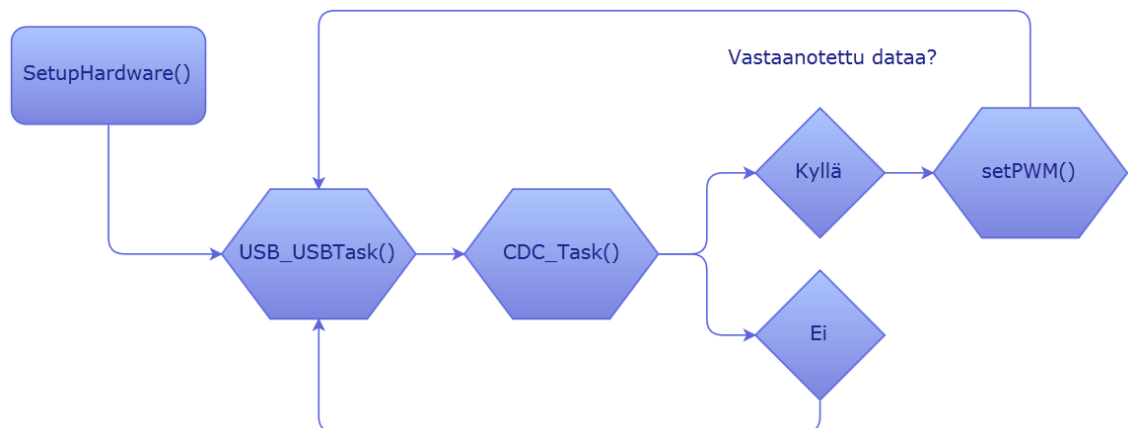
PCPWM:n resoluutioksi valittiin 14-bittiä, jolloin taajuus on 244 Hz ilman esiskaalausta. Tämä on riittävän iso taajuus, jottei ihmisen silmä näe välkkymistä.

Data-liitännän linjakoodaus asetettiin seuraavasti:

- Baud 9600 bps
- Parity None
- DataBits 8
- StopBits one.

5.1.3 Ohjelman kulku

Kuvassa 5.2 on kaaviokuva mikrokontrollerin ohjelmasta. Kun laitteeseen kytketään virta, alkaa ohjelma SetupHardware()-funktiolla. Siinä mikrokontrolleri alustetaan USB:a ja PWM-modulaatiota varten.



Kuva 5.2 USB-tunnelmavalon rutiini.

USB_USBTask()-funktio on LUFA kirjastosta, joka huolehtii USB:n toimivuudesta (mm. kiinnitykset, luettelointi). CDC_Task()-funktio on myös LUFA:sta. Siinä luetaan vastaan-

tettu data, data-liitäntään OUT-loppupisteestä. Funktiota on muokattu siten, että jos laite on vastaanottanut dataa, siirtyy se sitten setPWM()-funktioon.

PWM-modulaatio asetetaan setPWM()-funktiolla. Siinä vastaanotetusta RGB-väriarvosta asetetaan 16-bittisen ajastimen PWM-lähdöt väriarvon mukaan. Lopuksi funktio lähettää takasin joko 'ok' tai 'fail' riippuen siitä, onnistuiko PWM-muutos.

5.2 Windows

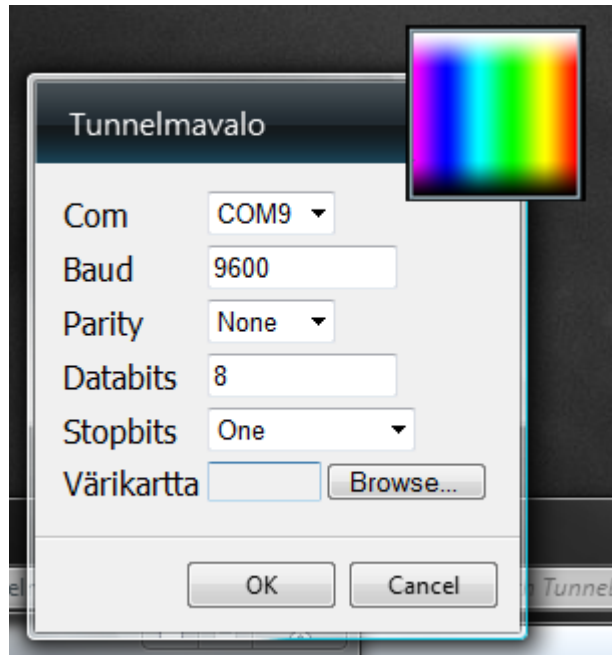
Tunnelmavalon käyttöohjelmaa lähdettiin suunnittelemaan alustavien vaatimusten perusteella Windowsin sivupalkin pienoishjelmaksi. Se on käytännössä HTML-sivusto, jolla voidaan käyttää Jscript-ohjelmointikieltä sekä CSS-tyylikieltä, mutta jonka ikkunakoko on pieni. Heti alussa kuitenkin osoittautui, että Jscript-ohjelmointikielellä ei ole pääsyä tietokoneen I/O-portteihin. Tämä ongelma paikattiin kehittämällä pienoishjelmalle ActiveX-komponentti, jolla on oikeus I/O-portteihin. ActiveX-komponenttia käytettiin sitten välittämään tietoa sarjaportin yli tunnelmavalolle.

5.2.1 Pienoishjelma

Tunnelmavalo-pienoishjelmaa lähdettiin kehittämään Microsoftin ohjeistuksen avulla [14]. Pienoishjelmalla on kolme eri tilaa: Dock, Settings ja Flyout, joista viimeistä ei kehitetty ollenkaan.

Settings-tilassa (kuva 5.3) ohjelmaan saa asettaa sarjaportin asetuksia sekä värikartan. Sen jälkeen kun käyttäjä on syöttänyt asetukset, tallentaa pienoishjelma ne tekstitiedostoon.

Ohjelma aloittaa Dock-tilassa (kuva 5.4), ja tarkistaa se ensimmäiseksi, onko tarvittava ActiveX-komponentti rekisteröity käyttöjärjestelmässä. Ellei ole, pienoishjelma yrittää rekisteröidä sen käyttämällä Microsoftin WScript.Shell -komponenttia. Jos tämä epäonnistuu, joutuu käyttäjä rekisteröimään komponentin itse. ActiveX-komponentin löydettyä, syöttää pienoishjelma komponentille käyttäjän asetuksista värikartan. Sen jälkeen pienoishjelma lukee sarjaportin asetukset tekstitiedostosta, jos semmoinen on olemassa, ja yrittää avata yhteyden tunnelmavaloon. Jos yhteyden avaus onnistui, on pienoishjelma valmiina käyttöä varten.



Kuva 5.3 Tunnelmavalo-pienoisohjelman Settings-tilassa.

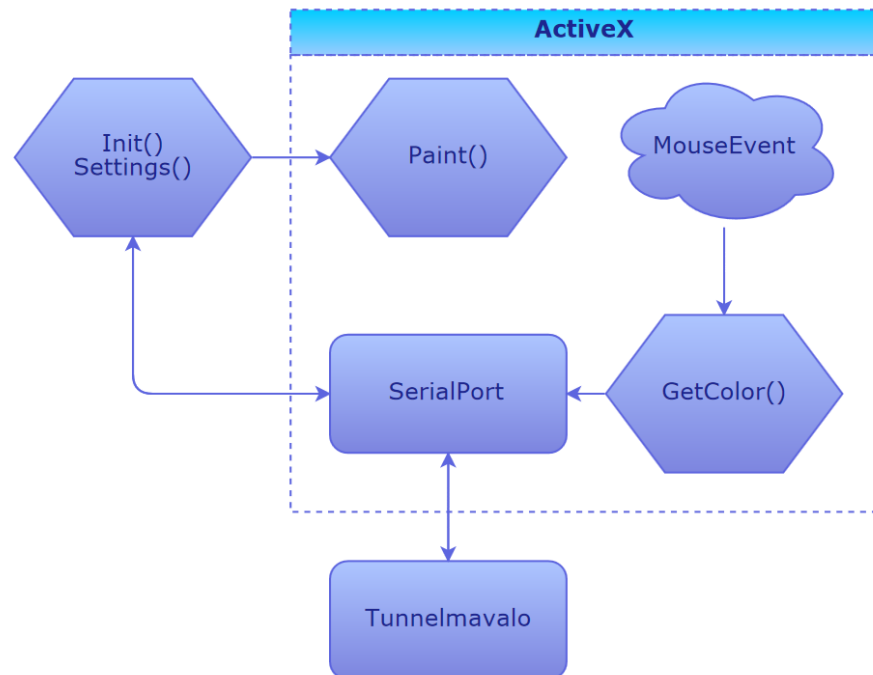


Kuva 5.4 Tunnelmavalo-pienoisohjelma Dock-tilassa.

5.2.2 ActiveX

ActiveX-komponentti kehitettiin .NET Framework 3.5 -versiolla ja pohjana käytettiin internetistä löydettyä ohjetta [15]. Komponentille tehtiin objekti SerialPort-luokasta, joka on .NET Frameworkin luokka sarjaportille. Luokka sisältää valmiina kaikki tarvittavat funktiot sarjaporttiyhteyden toteuttamiseksi. Joten tarvittiin ainoastaan kehittää komponentti yhteensopivaksi pienoisohjelman kanssa, ja välittää SerialPort-luokan funktiokutsut pienoisohjelmalle.

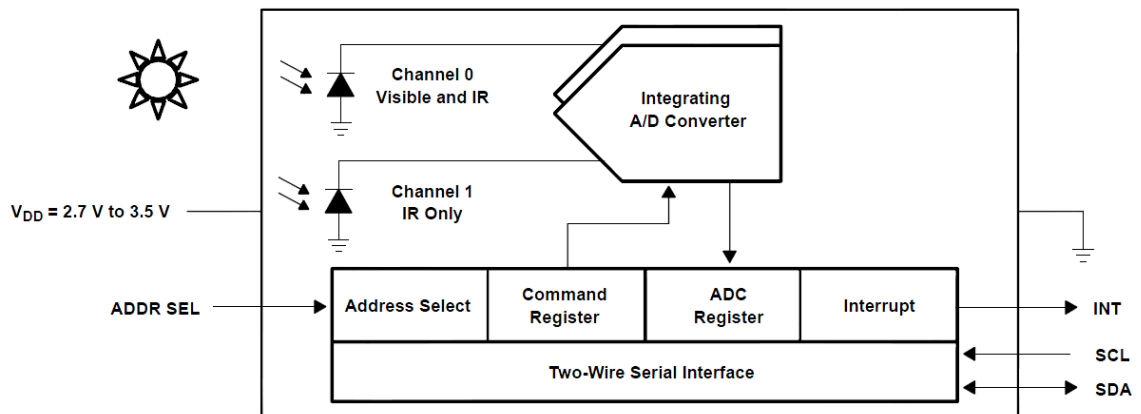
Kuvassa 5.5 on Tunnelmavallo-käyttöohjelman lohkokaavio. Pienoisohjelma toimii säiliönä ActiveX-komponentille ja huolehtii asetuksista. Kun ohjelma on käynnissä ja sarjaporttityhteys on avattu, toimii ActiveX-komponentti pääosassa ja MouseEvent-tapahtumien laukaisijana. Kun käyttäjä painaa hiirellä värikarttaan, aktivoituu GetColor()-funktio. Siinä selvitetään hiiren tarkka sijainti, ja sen avulla luetaan värikartasta RGB-väriarvo. Koska väriarvo on 8-bittinen, tapahtuu skaalaus asetetulla PWM-resoluutiolla. Sen jälkeen välitetään RGB-väriarvo tunnelmavalolle, joka vuorostaan asettaa värin PWM-lähtöihin.



Kuva 5.5 Tunnelmavallo-pienoisohjelman lohkokaavio.

6 MITTAUKSET JA TULOKSET

Tunnelmavalon mittaamisen tehtiin pieni oheispiiri, jossa on TAOS:n valmistama TSL2651 (kuva 6.1) Light-To-Digital valoanturi. Siinä on kaksi fotodiodia, joista toinen on herkkä näkyvälle- sekä infrapunavalolle, ja toinen ainoastaan infrapunavalolle. Fotodiodeilla on oma integroiva ADC-muunnin, ja integrointi tapahtuu molemmilla ADC-muuntimilla samanaikaisesti. Piiri keskustelee I2C-väylän avulla, eikä piiri ole erityisen herkkä kohinalle, koska se muuttaa analogisen fotodiodin signaalin digitaalseksi. Tästä syystä piiriä on helppo käyttää mikrokontrollerin kanssa, ja mittaamaan valomäärää käytettiin Arduino AVR-kehitysalustaa. [16]

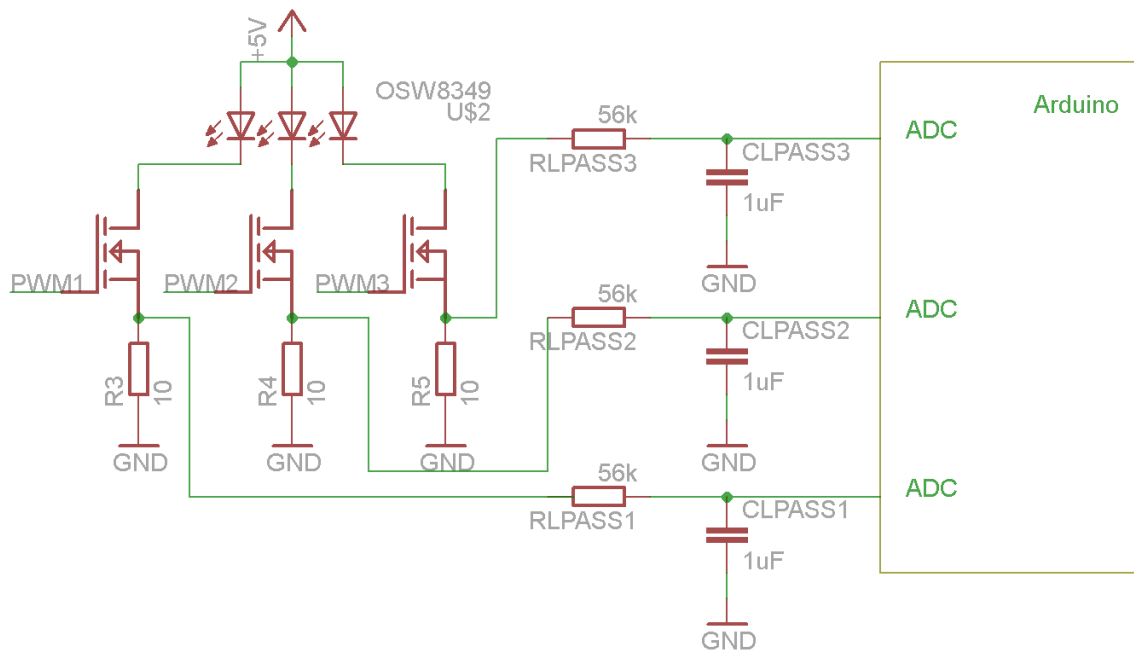


Kuva 6.1 TSL2651:n lohkokaavio [16].

Oheispiirille tehtiin myös kolme alipäästösuodatinta, joihin tuotiin tunnelmavalolta kunkin PWM-lähdön jännite $10\ \Omega$:n vastuksen yli kuvan 6.2 mukaisesti. Vastuksen arvoksi valittiin $56\ \text{k}\Omega$ ja kondensaattorin arvoksi $1\ \mu\text{F}$, jolloin suodattimen rajataajuus on $2,8\ \text{Hz}$. Kondensaattorin jännite mitattiin Arduinin ADC-muuntimen avulla, ja yhteen arvoon otettiin 10 peräkkäistä lukemaa ja keskiarvoistettiin. Jännite keskiarvoistettiin, koska alipäästösuodattimen jännite purkaantuu ja latautuu kokoajan, PWM-lähdön tahdissa. Keskiarvoistus kertoisi tarkemmin dc-virran suuruuden.

Arduinolla on 10-bittinen ADC-muunnin, joten jännitemittauksen tarkkuudeksi tulee siten $5\ \text{V}$:n referenssitasolla $4,88\ \text{mV/taso}$. Referenssitasoa voidaan kuitenkin muuttaa, jolloin saavutetaan tarkempi tulos. Referenssitasoksi asetettiin $2,25\ \text{V}$, jolloin tarkkuus on $2,2\ \text{mV/taso}$. Tämä tarkoittaisi sitä, että virran tarkkuudeksi tulisi noin $220\ \mu\text{A} \pm 5\ %$.

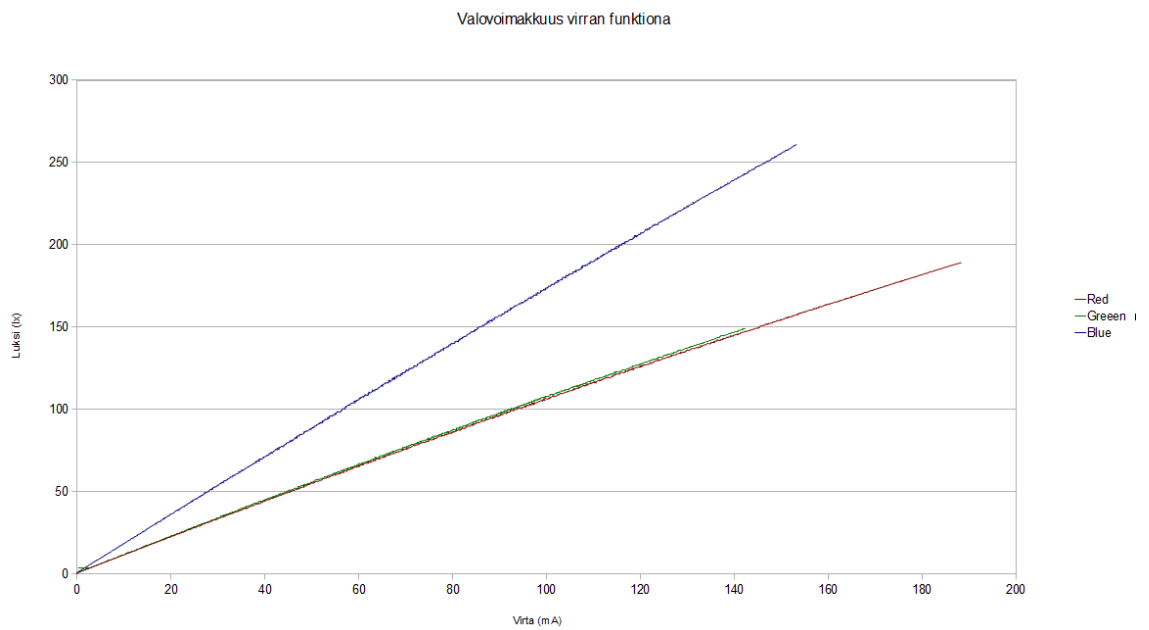
TSL2561:n vaste näkyvälle valolle on likimain ihmisen silmän kaltainen [16], joten oheispiirillä voidaan mitata valovoimakkuuden suhde dc-virtaan hyvin suurella alueella. Koska TSL2651:llä menee yhden valomittauksen integrointiin 400 ms, rajoittaa tämä mittausten määrää. Mikäli tunnelmavaloa askeltaisi koko 14-bitin alueella ($2^{14} = 16384$), kestäisi yhden värin mittaaminen noin 3 h. Tunnelmavaloa päätettiin askeltaa 10-bitin resoluutiolla, koska Arduinin ADC-muunnin on 10-bittinen, joten suurempi resoluutio ei toisi mitään etuja virtamittauksen kannalta.



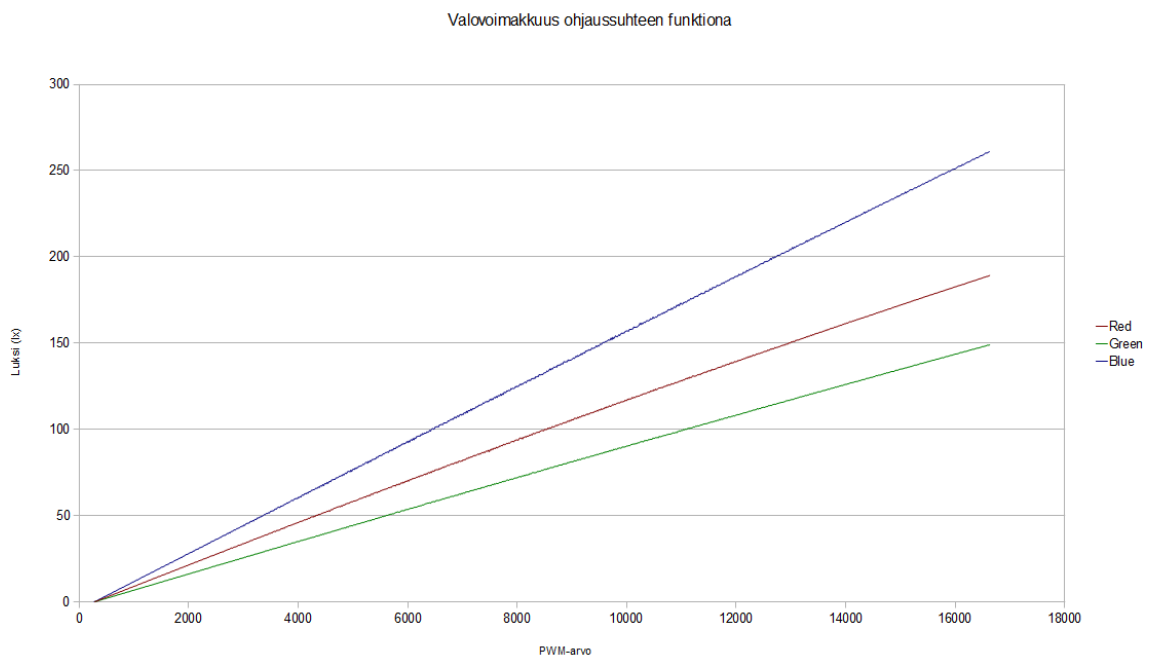
Kuva 6.2 Tunnelmavalon dc-virtojen mittaus.

Mittauksia varten tehtiin pieni sivuohjelma tunnelmavalon ActiveX-komponentin avulla, jossa komponentti keskusteli samanaikaisesti Arduinin sekä tunnelmavalon kanssa. Mittaukset tehtiin siten, että pimeässä huoneessa askellettiin yksittäisen PWM-lähdön arvoa, ja kirjattiin virta sekä valomäärä. Valoanturi pidettiin noin 10 cm:n etäisyydellä tunnelmavalosta. 16-bittisen TSL2651:n valomäärä muutettiin tietokoneella lukseihin (lx) datalehdien ohjeiden mukaisesti. Mittaukset tehtiin askeltamalla yhtä väriä 1024 kertaa koko alueella (0 – 16384), ja mittauksen ajaksi sammutettiin muut värit. Näin saatiin selville yhden värin valovoimakkuus virran ja ohjaussuhteen funktiona.

Ensimmäisen mittauksen tulokset näkyvät kuvassa 6.3 ja 6.4. Sininen on selvästi valovoimaisempi, kuin punainen tai vihreä. Virrat pysyvät sinisen ja vihreän värin osalta 150 mA:n tuntumassa, mutta punaisen virta on melkein 200 mA.



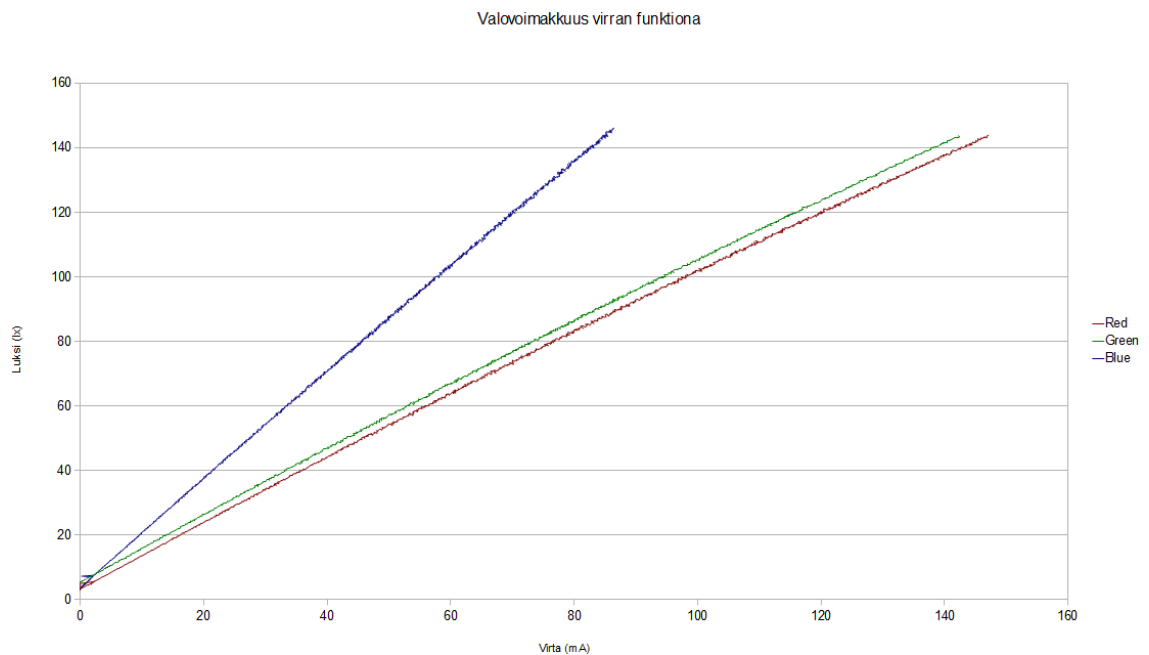
Kuva 6.3 Tunnelmavalon valovoimakkuuden suhde virtaan.



Kuva 6.4 Tunnelmavalon valovoimakkuuden suhde PWM-ohjaussuhteeseen.

Värien valovoimakkuudet muuttuvat, PWM-ohjaussuhteen mukaan, lineaarisesti hieman erilaisilla kulmakertoimilla. Tähän voidaan ohjelmallisesti vaikuttaa siten, että värit muuttuisivat samalla lailla. Kuvasta 6.4 nähdään, että vihreällä värillä on huonoin valovoimakkuus PWM-ohjaussuhteeseen nähden. Joten luonnollisesti skaalataan sinistä ja punaista

siten, että täydellä ohjaussuhteella ne saavuttaisivat saman valovoimakkuuden, kuin vihreä. Tämä saavutetaan skaalaamalla sinistä kertoimella 0,58 ja punaista kertoimella 0,78.

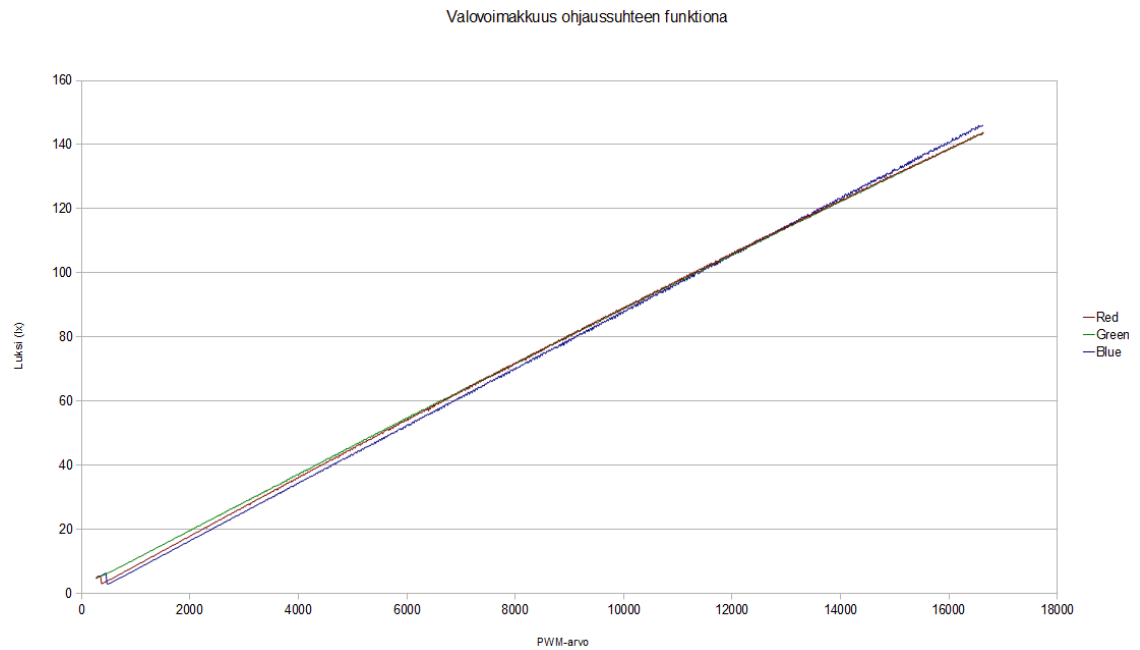


Kuva 6.5 Tunnelmavalon valovoimakkuuden suhde virtaan skaalatulla PWM-ohjaussuhteella.

Skaalauksen jälkeen mittaus suoritettiin uudestaan. Nyt sinisen ja punaisen virrat ovat pienentyneet (kuva 6.5), mutta vihreä on pysynyt samanlaisena. Sinisen huippuvirta on nyt noin 80 mA ja punaisen noin 140 mA. Merkittävin muutos on tapahtunut valovoimakkuuden suhteessa PWM-ohjaussuhteeseen. Kuvasta 6.6 nähdään nyt, että värit muuttuvat suurin piirtein samalla kulmakertoimella.

Tunnelmavalon ohjaaminen ActiveX-komponentilla onnistui erittäin hyvin. Suurena hyötynä komponentista on se, että sitä voidaan soveltaa muihinkin mikrokontrollereiden ohjaamiseen, kuten Arduinin tapauksessa. Tunnelmavalon pienoisohjelma toimii myös hyvin, ja ohjelma osaa lukea kuvasta RGB-väriarvon, joten siihen on mahdollista laittaa minkäläisen kuvan tahansa värikartaksi.

Tietokoneohjelmiston suunnitteluun ja kehittämiseen kului runsaasti aikaa lähinnä siksi, että kaikki oli tuntematonta projektin alussa. ActiveX-komponentti tuotti paljon vaivaa, eikä Microsoftin sivustoilta tahtonut löytyä ohjeita ActiveX-komponentin rekisteröimiseen. Väriarvon selvittäminen värikartasta osoittautui haasteelliseksi, koska Jscript-ohjelmointikieli ei tue kuvien käsittelyä kovinkaan laajasti. Tähän olisi ollut helppo ratkaisu, HTML5-Canvas elementti, mutta valitettavasti HTML5-tuki puuttui Windowsin sivupalkista.



Kuva 6.6 Tunnelmavalon valovoimakkuuden suhde skaalattuun PWM-ohjaussuhteeseen.

ActiveX-komponentti onnistui erittäin hyvin ja sitä on mahdollista käyttää muissakin projekteissa. LUFA:n toimivuus ja helppokäyttöisyys antaa näin ollen ActiveX-komponentin kanssa mahdollisuuden kehittää nopeasti uusia USB-laitteita. Tämä on ehdottomasti tämän työn tärkein saavutus.

7 YHTEENVETO

Tässä työssä suunniteltiin ja rakennettiin USB-tunnelmavalon piirilevyille, ja sille kehitettiin käyttöohjelma. Tunnelmavalon suunnittelussa painotettiin komponenttien määrää ja tavoitteena oli saada edullinen laite.

Työn ensimmäinen vaihe oli tutustua USB 2.0 -standardiin sekä LUFA-kehysrakenteeseen. Seuraava vaihe oli laitteen suunnitteleminen sekä komponenttien hankkimista. Komponenttivalinnoissa jouduttiin tyytymään osiin, jotka on yleisesti saatavilla elektroniikka-harrastelijoille. Tunnelmavalon tehtiin Atmelin AT90USB162 AVR-mikrokontrollerilla ja 3 W:n RGB-teholedillä. Työssä käytettiin suunnittelussa apuna yleisiä suunnitteluohjeita AVR-mikrokontrollerisovelluksen tekemiseen.

Tunnelmavalon pystyy vaihtamaan väriä, ja valolähteenä on RGB-teholedi. Erilaisia värejä saadaan aikaseksi ohjaamalla teholedin PWM-modulaatiolla. Värit valitaan tietokoneohjelmalla, joka on toteutettu Windowsin sivupalkin pienoisohjelmalla. Ohjelmaan jouduttiin myös kehittämään erillinen ActiveX-komponentti sarjaporttityhteyden toteuttamiseksi. Tunnelmavalon pienoisohjelmalla on toteutettu värikartta, josta hiiren avulla painamalla valitaan väri. Ohjelma lukee sitten värikartasta RGB-väriarvon, joka välittyy sitten ActiveX-objektin avulla sarjaportin yli tunnelmavalolle. Tunnelmavalon tulkitsee sitten vastaanotetun tiedon ja asettaa sopivan pulssinleveyden erikseen jokaiselle värikanavalle. Tunnelmavalon ei tarvitse erillistä teholähdettä vaan käyttää USB-porttia teholähteenä. Teho on siten rajoittunut 2,5 W:iin.

Tunnelmavalon valomäärää ja dc-virrat mitattiin erillisellä oheispiirillä ja Arduino AVR-kehitysalustalla. Teholedin valovoimakkuuden suhde PWM-ohjaussuhteeseen osoittautui hyvin lineaarisiksi. Teholedin värien valovoimakkuudet olivat odotetusti hieman poikkeavia, mutta ne korjattiin ohjelmallisesti skaalaamalla PWM-ohjaussuhdetta skalaarilla.

Tunnelmavalolle voisi jatkossa suunnitella ja kehittää jonkinlainen diffuusiokupu tai kotelointi. Se voisi mahdollisesti olla laavalampun muotoinen, tai sitten jokin yksinkertaisempi muoto, kuten esimerkiksi pallo. Pienoisohjelman Flyout-tilaa voisi jatkossa kehittää vaikka siten, että siinä olisi mahdollista tehdä yksinkertaisesti värisilmukoita. Silmukan värit ja ajastuksen voisi asettaa mielivaltaisesti, ja näin voisi saada aikaiseksi mielenkiintoisia syklejä. Värien asettaminen FFT:n avulla tietokoneen musiikin tahdissa, olisi myös mieluinen lisä.

Lähteet

- [1] Wikipedia, 'IEEE 1394 interface' [www-dokumentti]. Saatavilla: http://en.wikipedia.org/wiki/IEEE_1394_interface (luettu 20.4.2011).
- [2] Universal Serial Bus Specification Revision 2.0 (2000). Saatavilla: <http://www.usb.org/developers/docs/> (luettu 2.5.2011).
- [3] FTDI Chip, 'FTDI Drivers' [www-dokumentti]. Saatavilla: <http://www.ftdichip.com/FTDrivers.htm> (luettu 20.4.2011).
- [4] Microchip, 'USB Framework for PIC18, PIC24 & PIC32' [www-dokumentti]. Saatavilla: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en537044 (luettu 20.4.2011).
- [5] Atmel, 'AVR USB Series2 Software Packages' [www-dokumentti]. Saatavilla: http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4440&category_id=163&family_id=607&subfamily_id=760 (luettu 20.4.2011).
- [6] Four Walled Cubicle, 'LUFA (2011)' [www-dokumentti]. Saatavilla: <http://www.fourwalledcubicle.com/LUFA.php> (luettu 3.5.2011).
- [7] LUFA Library, 'Source Code License' [www-dokumentti]. Saatavilla: http://www.fourwalledcubicle.com/files/LUFA/Doc/110528/html/_page__license_info.html (luettu 3.5.2011).
- [8] LUFA Library, 'LUFA Library Documentation' [www-dokumentti]. Saatavilla: <http://www.fourwalledcubicle.com/files/LUFA/Doc/110528/html/> (luettu 3.5.2011).
- [9] "AT90USB162 Data Sheet," Atmel Corporation, San Jose.
- [10] "OSW-8349 Data Sheet," Multicomp.
- [11] AVR042: AVR Hardware Design Considerations. [pdf-dokumentti] Saatavilla: http://www.atmel.com/dyn/resources/prod_documents/doc2521.pdf (luettu 5.5.2011).
- [12] AVR040: EMC Design Considerations. [pdf-dokumentti] Saatavilla: http://www.atmel.com/dyn/resources/prod_documents/doc1619.pdf (luettu 5.5.2011).
- [13] "FDV303N Data Sheet," Fairchild Semiconductors, San Jose.

- [14] MSDN, 'Developing a Gadget for Windows Sidebar Part 1: The Basics' [www-dokumentti]. Saatavilla:
<http://msdn.microsoft.com/en-us/library/bb456468%28v=vs.85%29.aspx> (luettu 11.5.2011).
- [15] Jaime Rodrigues, 'writing a .net activex control for your sidebar gadget..' [www-dokumentti]. Saatavilla: http://blogs.msdn.com/b/jaimer/archive/2006/10/02/writing-a-.net-activex-control-for-your-sidebar-gadget_2e002e00_.aspx.
- [16] "TSL2560, TSL2561 Light-To-Digital Converter Data Sheet," Texas Advanced Optoelectronic Solutions Inc., Plano Texas.