

Opinnäytetyö (AMK)

Elektronikka

Elektronikkasuunnittelu

2017

Matias Sorsa

MIKROPROSESSORIOHJATTU ETÄISYYSMITTAUS ULTRAÄÄNELLÄ

Matias Sorsa

MIKROPROSESSORI OHJATTU ETÄISYYSMITTAUS ULTRAÄÄNELLÄ

Opinnäytetyön tarkoituksena oli selvittää, miten ultraäänellä pystytään mittaamaan etäisyys tiettyyn kohteeseen 200 kHz:n ultraäänisensorilla, ja toteuttaa mittausprosessi ennalta määrätyllä STM32- mikroprosessorilla. Myöskin elektroniikkasuunnittelu haluttiin toteuttaa mahdollisimman edullisesti ilman IC-piirejä, joten toteutus pyrittiin tekemään puhtaasti analogisilla komponenteilla ja vahvistinasteet transistoreilla.

Työn toteutuksen pääosat koostuivat signaalin lähetyspään ja vastaanoton suunnittelusta sekä STM32L476RG mikroprosessorin ohjelmoimisesta signaalin lähettämistä ja vastaanottamista varten sekä etäisyyden taltioimista varten. Lähetyksessä huomioitavaa oli saada prosessorilta tuotettava 3,3 V signaali vahvistettua yli 15 V huippuarvolta olevaksi ja vastaanotossa saada noin millivoltin signaali takaisin korkeaksi jännitetasoksi eli yli 2,5 V korkeaksi. Mitattu etäisyysalue oli 10cm - 1m.

Lopputuloksena saatiin kahdelle eri piirilevyille (lähetys ja vastaanotto eroteltuna) toteutettu mittauslaite, joka onnistui mittaamaan etäisyydet 10 cm:istä 1 m:iin lähettimestä vastaanottimelle. Työssä huomattiin, että suurimmat haasteet tulevat vastaanotossa tapahtuvasta häiriöiden vahvistamisesta, kun vahvistettava signaalitaso on todella alhainen.

ASIASANAT:

Ultraäänianturi
Transistori
Vahvistin
Jännitenostaja
STMicroelectronics

Matias Sorsa

MICROCONTROLLER BASED DISTANCE MEASUREMENT WITH ULTRASONIC

The goal of the thesis was to explore how to measure distance to some object with ultrasonic sensor, and to create the distance measurement device driven by STM32L476RG microprocessor. It was given that the electronical design of the signal processing should be low cost, so the electronical design of the device was to be implemented with low cost components, like transistors and analogic components.

The main parts of the designing the device were the transmitter and receiver of the signal and the programming of the microcontroller so it would send the signal and receive it in right manner. The main specification for transmitter were the boost from 3.3V signal level to over 15V peak voltage level and on the receiving end the 1-2 millivolt signal was to be amplified back to microprocessors high voltage level (over 2.5 voltage).

The outcome of the thesis was a working device with separate PCB boards for transmitter and receiver connected to the microcontroller. The device could measure the distance with >3% error margin and for 1m distance from sensor to sensor. It was seen, that when you're amplifying extremely small signal, the filtering of the signal needs to be decent to be able to get rid of the noise, because otherwise the signal is lost within the amplified noise.

KEYWORDS:

Ultrasonic transducer

Transistor

Amplifier

Boost Converter

STMicroelectronics

SISÄLTÖ

KÄYTETYT LYHENTEET	5
1 JOHDANTO	7
2 ULTRAÄÄNI JA TYÖN LÄHTÖKOHDAT	8
2.1 Ultraääni	8
2.2 Työn resurssit ja aloitus	8
3 ELEKTRONIIKKASUUNNITTELU	10
3.1 Lähetyspää	10
3.1.1 Lähetysten suunnittelu ja mitoitus	10
3.1.2 Simulointi ja mittaus	15
3.2 Lähetysten vastaanotto	17
3.2.1 Lähtökohdat	17
3.2.2 Suodatin	18
3.2.3 Vahvistin	19
3.2.4 Simulointi ja mittaukset	24
4 OHJELMOINTI	28
4.1 Lähtökohdat	28
4.2 Toteutus	28
4.2.1 200 kHz:n signaalin lähettäminen	29
4.2.2 Viiveen taltiointi	32
4.2.3 Kokonaisuus	32
5 LOPPUTESTAUS	36
5.1 Testaus	36
5.2 Kehitysideat ja ongelmat	37
5.3 Lopuksi	37
LÄHTEET	39

KÄYTETYT LYHENTEET

FET	Field-Effect Transistor, kanavatransistori
IC	Integrated Circuit, mikropiiri
STM	STMicroelectronics, puolijohdevalmistaja
Boost Converter	Jännitettä nostava hakkurityyppi
Schottky	Diodityyppi, jolla on nopea toiminta jännitteen ja virran suhteen sekä alhainen jännitehäviö
BaTiO ₃	Bariumtitaani, yksi yleisimmistä pietsosähköisten sensorien materiaaleista
SPICE	Simulation Program with Integrated Circuit Emphasis, vapaan lähdekoodin virtapiirisimulaattori
SMD	Surface Mount Technology, pintaliitoskomponentti
CE-vahvistin	Common Emitter -vahvistin, transistorin vahvistinmalli, jossa ulostulo otetaan kannalta
Aktiivitila	Transistorin tila, kun se toimii virran vahvistimena
Cut off- tila	Transistorin tila, jossa virtaa ei kulje transistorissa
Saturaatio	Transistorin tila, jossa virta on yli vahvistuskertoimen ja se käytännössä toimii kiinni olevana kytkimenä
h_{fe}	Transistorin tyypin mukaan määräytyvä virran vahvistuskertoimen
ARM	Advanced RISC Machines, 32-bittinen mikroprosessoriarkkitehtuuri
PWM	Pulse-Width Modulation, pulssinleveysmodulaatio eli modulointitapa, jolla muutetaan pulssin on/off suhdetta.
HSI	High-Seed Internal Clock, korkealla kellotaajudella toimiva mikroprosessorin sisäinen kello

GPIO	"General Purpose Input/Output" -portti, mikroprosessorin yleinen tulo / lähtöportti.
TIMER	TIMER -rekisteri, yleinen mikroprosessorirekisteri, joka on suunniteltu ajan taltioimista varten.
NVIC	Nested Vectored Interrupt controller, keskeytysten hallintarekisteri mikroprosessorilla.

1 JOHDANTO

Työn tarkoituksena oli suunnitella 200 kHz:n taajuudella toimivan ultraäänianturilla toimiva etäisyysmittauslaite, jota suoritettaisiin mikroprosessorilla. Laite haluttiin myös toteuttaa alhaisilla komponenttikustannuksilla, joten käytännön suunnittelussa ei käytettäisiin ollenkaan FET:ejä tai operaatiovahvistimia. Myös mittauslaitteen haluttiin olevan itsenäinen, paristolta virran saava laite.

Puhtaasti kustannustehokkaasta näkökulmasta ja 200 kHz:illä toimivaa mittauksia ei opinnäytetyönä ole aikaisemmin tutkittu, vaikka toki yleisesti ultraäänellä toteutettavasta etäisyysmittauksesta löytyy esimerkiksi Matti Koramon tekemä opinnäytetyö vuodelta 2009, mutta tämäkin työ keskittyy alhaisemmalla taajuudella (50 kHz) toimivaan mittaukseen. [1]

Työn päävaiheet koostuivat STM32L476RG -mikroprosessorin ohjelmoinnista sekä vahvistimen ja lähettimen elektroniikkasuunnittelusta. Työssä esitellään aluksi ultraäänimittaukseen liittyviä periaatteita, joiden jälkeen siirrytään mikroprosessorin ohjelmoinnin toteutukseen. Tämän jälkeen käydään läpi elektroniikkasuunnittelun vaiheet sekä lähetyksen että vastaanoton osalta, minkä jälkeen lopuksi siirrytään kokonaisuuden testaamiseen sekä työssä esiintyneiden ongelmien pohdintaan ja korjauksiin.

2 ULTRAÄÄNI JA TYÖN LÄHTÖKOHDAT

2.1 Ultraääni

Ultraääni on korkeataajuinen ääniaalto, jonka taajuusalue alkaa noin 20 kHz:tä eli ihmiskorva ei sitä pysty kuulemaan. Tästä syystä sitä käytetään monissa eri tekniikan alojen laitteissa niin etäisyyden mittaamiseen kuin kohteen paikantamiseen. Myös lääketieteellisyydessä sitä käytetään ihmisen sisäisten elinten tutkimiseen. [2]

Ultraäänisensorit jakautuvat kolmeen ryhmään: ultraäänilähettimet, -vastaanottimet ja -sensorit. Ultraäänilähettimet pystyvät muuntamaan siihen syötetyn AC-signaalin ultraääneksi, vastaanotin muuttamaan vastaanotetun signaalin takaisin AC-signaaliksi ja sensori pystyy tekemään molemmat. Tässä työssä käytetään 200 kHz:n sensoreita erikseen vastaanottamaan ja lähettämään signaali, koska tämä sallii kehitystyön jatkumisen niin että käytössä olisi vain yksi sensori. [3]

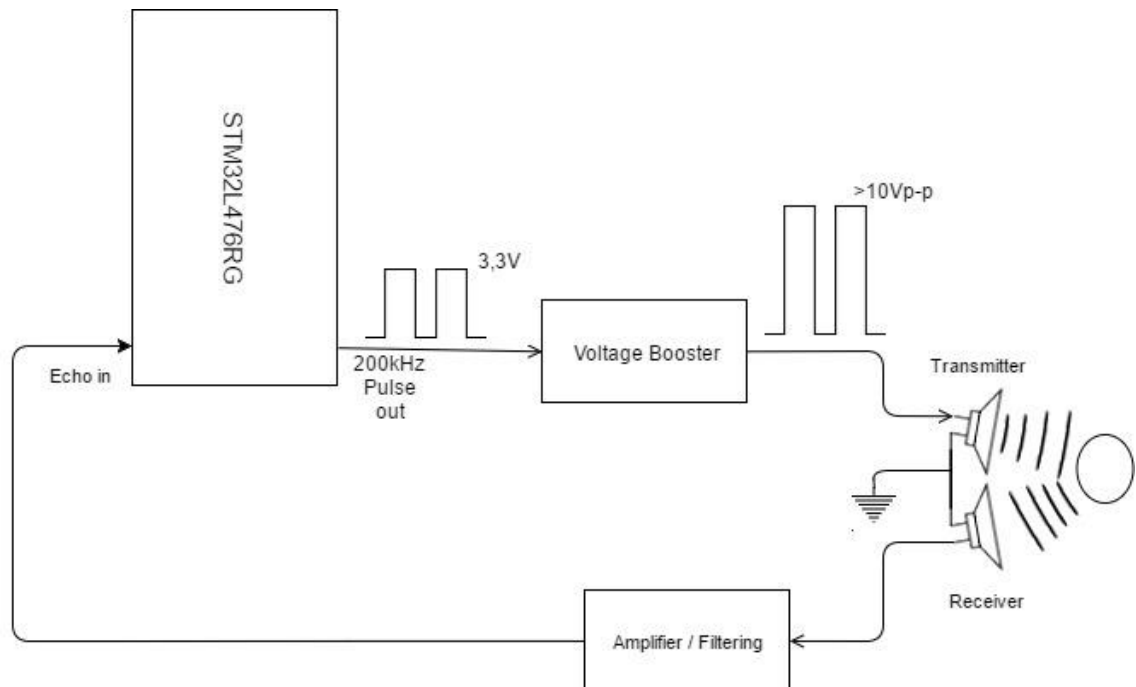
Ultraäänimittaus siis perustuu käytännön ajatukseen, että tuotetaan tässä tapauksessa 200 kHz:n AC signaali, joka syötetään ultraäänilähtimelle, joka muuntaa sen ultraääneksi. Sen jälkeen vastaanotin, vastaanottaa sen, joko suoraan kohdistettuna signaalin lähettimeen tai kun lähetetty signaali osuu kohteeseen, se kimpoaa sen pinnasta takaisin lähettimeen. Huoneen lämmössä ultraäänisignaali kulkee äänen tavoin 343 m/s, joten kun tämä tiedetään, pystytään laskemaan etäisyys kohteeseen suoraan lähetyksen ja vastaanoton välillä kuluneesta ajasta.

2.2 Työn resurssit ja aloitus

Tässä opinnäytetyössä käytettiin ta0200kb ultraäänisensoria, jolle luvataan mittausetäisyydeksi 0.1 – 2 m. Myöskin huomioitavaa on sensorin kapea vastaanottokeila ($7 \pm 2^\circ$), joka meinaa, että signaalin havaitsemiseksi on sensorien oltava melko tarkasti kohdakain tai vaihtoehtoisesti käytettävä sensoreille erillistä fyysistä "horn" -kaulaa sensorille signaalien keräämiseksi. Prosessoriksi työssä valikoitui STMicroelectronicsin

STM32L476RG, joka on 32-bittinen alhaisella virrankulutuksella ja lepo-toiminolla varustettu prosessori, joten sitä pystyy myös 3,6 V paristolla pyörittämään, mikä sallisi etäisyysmittarin käytön yksittäisenä laitteena. [4][5]

Työssä lähdettiin liikkeelle pohtimalla, mitä kaikkea etäisyysmittaukseen tarvittaisiin. Käytettävää sensoria mittaamalla selvitettiin, että lähetteen on oltava hyvin korkea (yli 10 V) amplitudiltaan ja ilman vahvistusta saatava vastaanottosignaalia on vain muutamia millivolteja, joten vahvistuksen on oltava tarpeeksi suuri, että mikrokontrolleri pystyy sen lukemaan. Näin ollen tiedetään, että signaalia täytyy vahvistaa sekä lähetyksessä että vastaanotossa. Vastaanotossa myös tarvitaan suodatus, koska vahvistettava signaali on niin pieni, että häiriötä tulee.



Kuva 1. Järjestelmän lohkokaavio.

Järjestelmän periaatteellinen lohkokaavio (kuva 1) kuvastaa työn päävaiheet. Työn elektroniikkaosuudessa pureudutaan lähetyksen jännitteen kasvattamiseen (Voltage booster) ja vastaanoton vahvistukseen ja suodatukseen (Amplifier / Filtering).

3 ELEKTRONIikkASUUNNITTELU

Työn elektroniikkasuunnittelu painottui puhtaasti lähetyksen ja vastaanoton suunnitteluun. Etäisyysmittauksen elektroniikkasuunnittelun pääpainona oli todella alhaiset komponenttikustannukset, mikä toi projektiin sopivaa suunnittelutyötä. Myös lähetyksessä ja vastaanotossa rajoittui toteutuksien määrä, kun suunnittelu päätettiin toteuttaa puhtaasti transistoreilla ja passiivisilla komponenteilla. Sekä lähetyksen, että vastaanoton piirien käyttöjännite on STM32L476RG:ltä saatava 3,3 V.

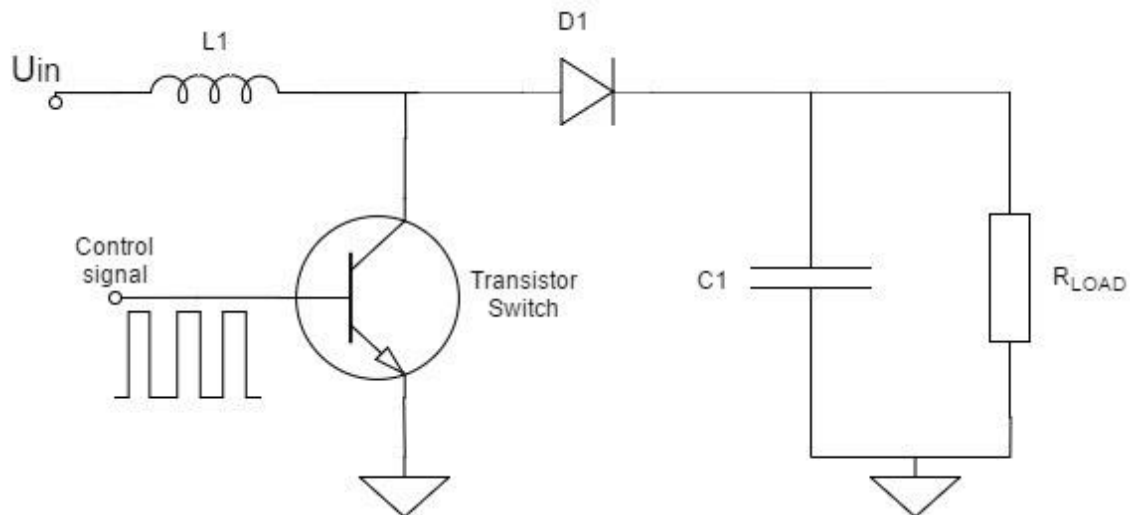
3.1 Lähetyspää

3.1.1 Lähetyksen suunnittelu ja mitoitus

Suunnittelun lähtökohdat

Sensoreita testatessa tuli selväksi, että vain 3,3 V:n signaali ei tulisi lähetysteholtaan riittämään, niin että saataisiin luettua etäisyyttä muutamaa senttiä kauempaa. Näin ollen lähdettiin periaatteella, että lähetykset vahvistettaisiin yli 15-30 V_{pp}.

Kun halutaan vahvistaa signaali yli käyttöjännitteen jännitetason, tarvitaan hakkuripiiri, missä jännitettä katkotaan joko FET:llä tai transistorilla. FET:llä katkominen olisi puhtaampaa, koska FET ei syö kantavirtaa, mutta työssä päätettiin käyttää kustannussyistä transistoreja, joten sillä edettiin.



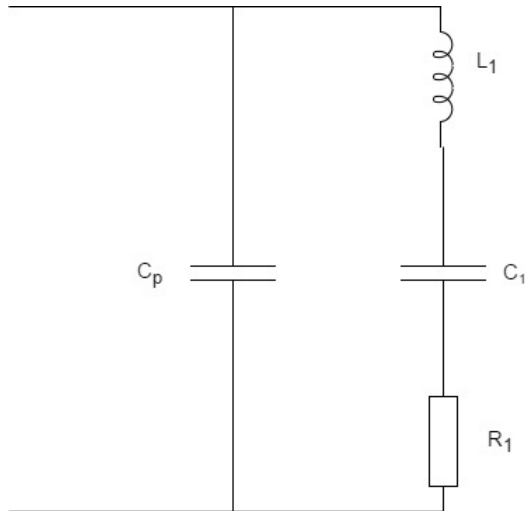
Kuva 2. "Boost Converter" – jännitenostajan rakenne.

Työssä päädyttiin kokeilemaan jännitteen kasvatusta Boost Converter (kuva 2) tyylisellä katkojapiirillä. Kytken toiminta perustuu siihen, että halutun taajuisella ohjaus signaalilla avataan ja suljetaan transistoria. Kun transistorissa kulkee virta, kulkee suurin virtapiiri transistorin läpi, koska diodista $D1$, kuormavastuksesta R_{LOAD} ja suodatuskondensaattori $C1$:stä muodostuva impedanssi on paljon suurempi. Näin ollen kelaan indusoituu virtaa eli se varastoi energiaa. Kun transistori suljetaan, kelaan indusoituu vastakkaisvaiheinen jännite, joka varastoituu suoraan ulostulon $C1$ kondensaattoriin. Näin ollen ulostulossa vaikuttaa $U_{in} + U_L - U_{D1}$ suuruinen jännite. Diodina suositaan Schottky-diodeja niiden nopean jännitteen katkaisun vuoksi ja alhaisen kynnyksjännitteen takia. [6]

Mitoitus

Käytännön kytkentää lähdettiin tekemään niin, että kytkin transistorina toimisi BC847B pintaliitostransistori, kelana koululta saatavana oleva $100 \mu\text{H}$ kela, piirilevyn tekemisen nopeuttamista varten. Diodiksi valikoitui BYS10-25 Schottky-diodi, joka virran ja jännitteenkestoltaan on tarkoitukseen sopiva, kuten myös BC847B, joka kestää 200 mA :n piikkivirran kannalla. Myöskin kytkentään lisättiin käyttöjännitteeseen suodatuskondensaattori häiriöitä suodattamaan ja transistorin kannalle lisättiin pieni suojavastus, joka ei käytännön toimintaan vaikuta, mutta on hyvä olla suojaamassa transistoria. [7][8]

Seuraavaksi tarvitsi luoda sijaiskytkentä ultraäänianturille, jotta saatiin todellinen kuva kuorman koosta mitoitusta varten. Ultraäänisensori voidaan korvata RLC- sijaiskytkennällä, joka simuloi ultraäänen toimintaa sarjaresonanssikytkennällä (kuva 3)

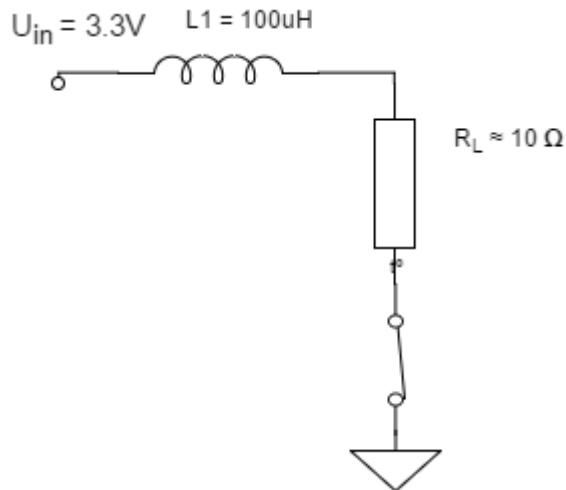


Kuva 3. Ultraäänianturin sijaiskytkentä

C_p eli staattinen kapasitanssi kuvastaa elektronista kapasitanssia, kun taas L_1, R_1, C_1 kuvastaa ultraäänianturin mekaanisia ominaisuuksia: kela L_1 kuvastaa massaa, kondensaattori C_1 jäykkyyttä ja R_1 vaimennusta. Staattiselle kapasitanssille saadaan 500 pF:n arvo datalehdessä ja mekaanisen vaimennuksen arvioinnissa voidaan käyttää minirinnakkaisresistanssin arvoa eli 250Ω . LC arvot mitoitetaan 200 kHz:n resonanssitaajuudelle RLC piirin mitoituksen mukaan niin, että otetaan huomioon sensorin materiaalin elektromeekaaninen liittämiskerroin (electromechanical coupling coefficient). Tämän arvona käytettiin 0,38, mikä on yleisimmän antureissa käytettävän materiaalin arvo ja vaikka sensorin materiaalia ei datalehdessä suoraan mainita, pystytään sen lämpötila käyttäytymisen mukaan päätellä sen olevan BaTiO_3 materiaalia. Sievennettyjen kaavojen avulla saadaan C_1 :n arvoksi 66pF ja L_1 :en arvoksi 9.55 mH. [9][10][11]

Virran latausaika

Kuten aikaisemmin mainittiin, jännitenostajalla on kaksi vaihetta: kun kytkin on auki ja kiinni. Seuraavana esitellään latausvaihe eli kun kytkin on kiinni.



Kuva 4. Jännitenostajan latausvaihe

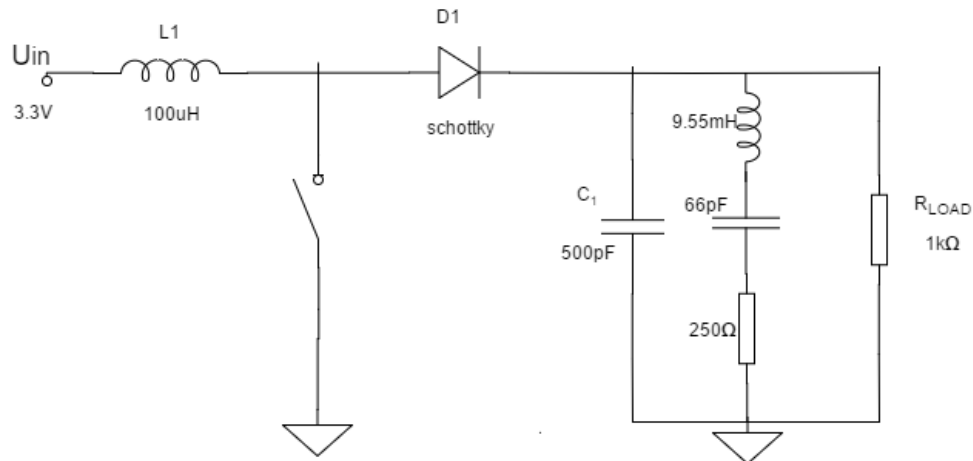
Kun kytkin suljetaan, käytännössä virta kulkee melkein kokonaan suljetun kytkimen kautta, joten diodin jälkeinen osuus voidaan jättää huomiotta (kuva 4). Kelalle käytetään laskennassa sisäistä $10\ \Omega$ sarjaresistanssia, mikä on sopiva arvio kyseiselle kelalle. Näin ollen, kun tiedetään että tämä tilanne jatkuu $2,5\ \mu\text{s}$ niin saadaan kelaan indusoituvaksi virraksi:

$$I = \frac{U}{R} \left(1 - e^{-\frac{t}{\tau}} \right) = \frac{3.3\text{V}}{10\ \Omega} \left(1 - e^{-\frac{2.5\ \mu\text{s}}{\left(\frac{100\ \mu\text{H}}{10\ \Omega} \right)}} \right) \approx 73,0\text{mA} \quad (1)$$

Tästä huomataan myös, että pulssisuhdetta muuttamalla niin, että päällä olo aika on pitempi, saadaan nopeasti indusoitua virtaa kasvatettua tarpeen tullessa.

Energian purkautuminen

Toisessa vaiheessa, kun ohjaava signaali menee alhaiseksi jännitetasoksi eli nolaksi, voidaan ajatella kytkimen aukeavan.



Kuva 5. Jännitenostajan purkautumisvaihe

Tässä vaiheessa kelaan indusoitunut virta purkautuu diodin kautta ulostuloon. Teoreettisen, suuntaa-antavan ulostulevan jännitteen laskussa voitiin jättää sensorin sijaiskytkentä pois ja laskea vain 500 pF kondensaattorin ja vastuksen avulla. Näin ollen ulostulon jännite:

$$U_{LOAD}(t) = 3,1 V \times e^{-1 \times 10^6 t} \times \cos(4,36 \times 10^6 t) + 34,2 V \times e^{-1 \times 10^6 t} \times \sin(4,36 \times 10^6 t) - 3.1 V \quad (2)$$

Kokonaisvirrasta ja ulostulevan jännitteen muodosta saadaan kokonaiskuva, millainen jännite saadaan kytkennällä ulos. Teoreettisen kaavan avulla pystyttiin laskemaan, kuinka nopeasti jännite purkaantuu, kun laskettiin jännitteentaso eri aikoina.

Taulukko 1. Jännitenostajan jännitteen purkautuminen ajan funktiona

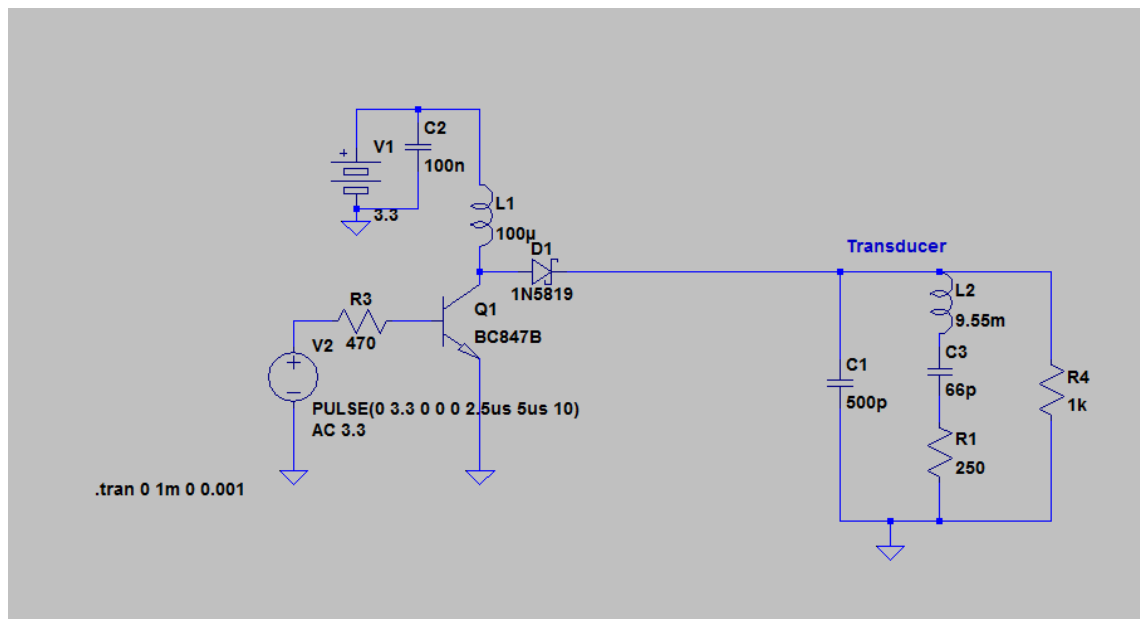
Aika[μs]	U _{LOAD} [t]
0.00	0.00
0.25	21.63
0.50	12.84
0.75	-6.61
1.00	-15.30
1.25	-9.76
1.50	-0.50
1.75	2.81
2.00	-0.41
2.25	-4.75
2.50	-5.92
2.75	-4.13
3.00	-2.13

Tärkeää oli huomata ulostulevan jännitteen lisäksi se, että kaavalla laskemalla nähdään, että jännite ehtii melkein kokonaan palautumaan 3,1 V tasolle 3 μ s aikana (Taulukko 1). Myöskin ulostuleva jännite olisi 20 – 30 V, mikä oli toivotulla tasolla.

3.1.2 Simulointi ja mittaus

Simulointi

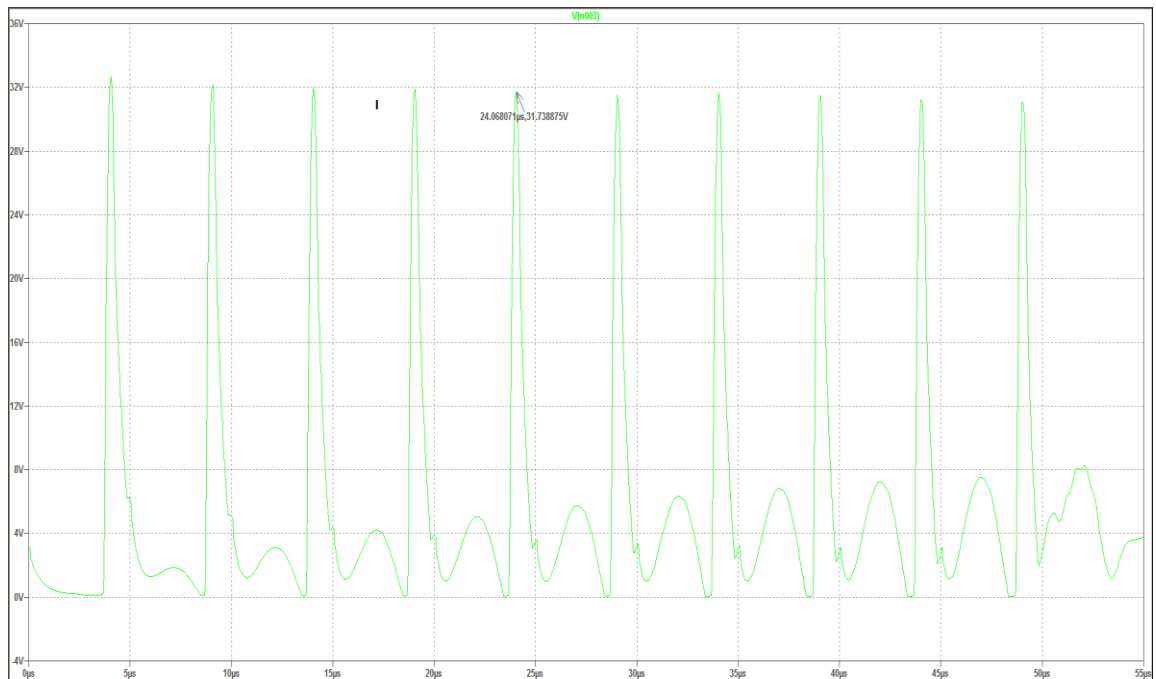
Sen jälkeen, kun oli saatu laskettua teoreettinen jännite, simuloitiin kytkennän toiminta LTSpice -simulaattorilla. Kyseinen simulaattori on Linear Technology:n tekemä SPICE -simulaattori hyvällä toteutuksella.



Kuva 6. LTSpice simulaation kytkentä.

Nähdään, että simulaattorilla pystyy myös valitsemaan oikeat komponentit ja kelalle sai myös lisättyä siihen realistisesti kuuluvan sarjaresonanssin, mikä ei kuvassa (kuva 6)

näy, mutta simulaattori sen laskee simuloitutulokseen. Kun kytKentä oli valmis, katsottiin millainen ulostulo saatiin jännitteenä ajan funktiona.

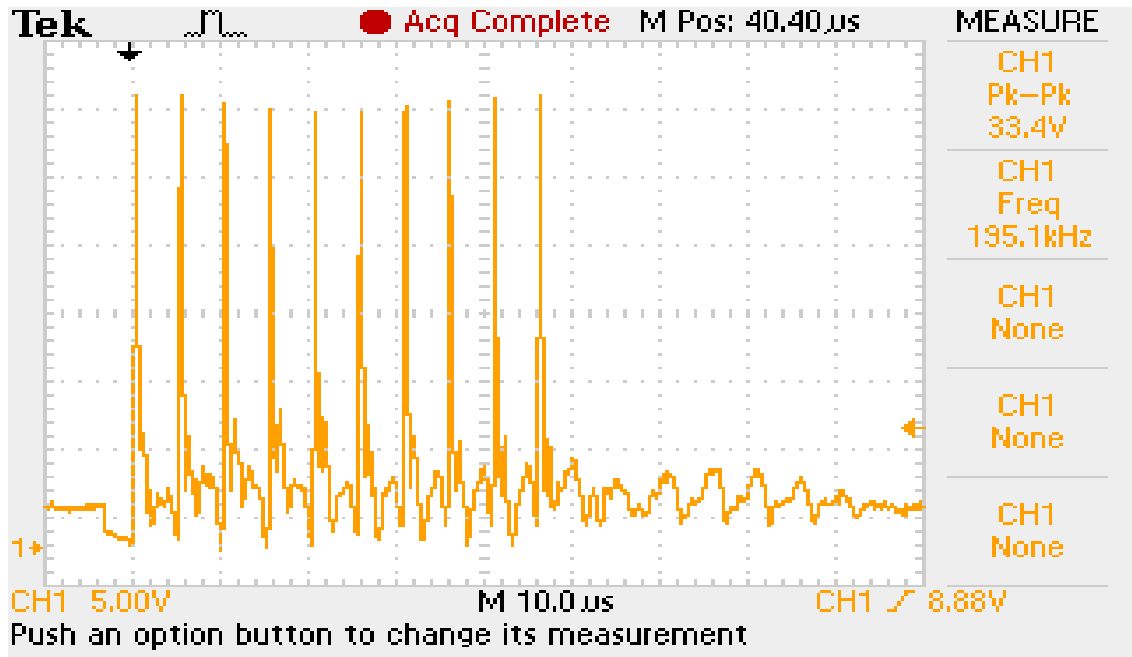


Kuva 7. LTspice ulostulo ajan funktiona, huippuarvo 32,5 V.

Kuvasta näkee (kuva 7), että jännitetaso on mitoituksen mukainen ja varsin hyvä 50 % pulssisuhteella. Simulaatiossa näkyy myös kelan virran muutosta vastustava vaikutus, minkä takia aika ei ole täysin sama mitoituksen kanssa ja tätä myöten myöskin huippuvirta oli korkeampi kuin teoreettinen, minkä takia huippujännite on korkeampi kuin teoreettisesti. Ulostulevan jännitteen muoto oli kumminkin odotetunlainen, ja jännitetaso oli toivottu. Tämän jälkeen lähdettiin tekemään levyä, kun tiedettiin että mitoitus olisi kohdillaan.

Mittaus

Piirilevy tehtiin Mentor Graphicsin PADS piirisuunnitteluohjelmalla ja se toteutettiin kokonaan SMD-komponenteilla. Tämän jälkeen mitattiin saatu jännite ulostulosta, kun ohjaussignaalina toimi 10 pulssia pitkä, 200 kHz 3,3 V:in kantiaalto.



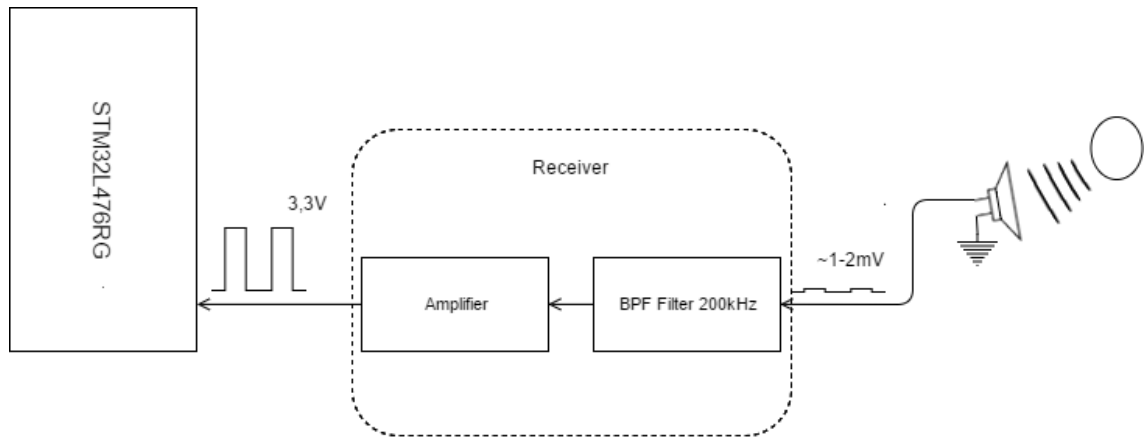
Kuva 8. Oskilloskooppi kuva lähetyspään ulostulosta

Voidaan todeta, että lähetyspään todellinen ulostulo oli jokseenkin identtinen simuloidun kanssa. Jännitenostajan alhainen jännitetaso ei ole täysin puhdasta, mutta tämä ei varsinaisesti ole ongelma, joten tämän suunnittelun kanssa pystyttiin jatkamaan eteenpäin.

3.2 Lähetysten vastaanotto

3.2.1 Lähtökohdat

Vastaanottoa lähdettiin suunnittelemaan sen jälkeen, kun oli pohjatiedoiksi saatu, että vastaanotettavan signaalin amplitudi olisi jokseenkin muutaman millivoltin luokkaa maksimietäisyyksillä. Myös häiriön suodatuksen tarvittaisiin kaistanpäästösuodatin jyrkällä vaimennuksella, koska kun signaali on vain muutamia millivolteja, se on samaa koko luokkaa erilaisten jännitteensyötön ja muiden lähteiden tuottamien häiriöiden kanssa.



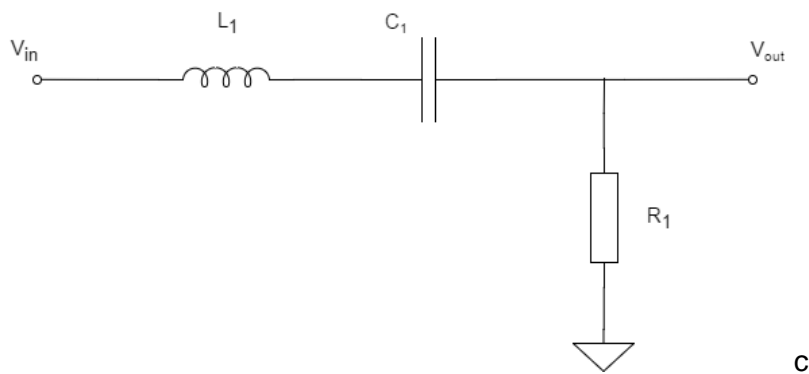
Kuva 9. Vastaanoton lohkokkaavio.

Annetuista lähtökohdista, eli analogisten komponenttien suosiminen ja transistoreilla suunnitteleminen, päädyttiin toteuttamaan suodatus ja vahvistus erikseen, niin että aluksi on kapea kaistainen kaistanpäästösuodin ja sen jälkeen tapahtuu vahvistus (kuva 9).

3.2.2 Suodatin

Kun työssä päädyttiin käyttämään kaistanpäästösuodatinta erillisenä, oli hyvin tavanomainen ratkaisu käyttää RLC-suodatinta. Suodattimen rakenne on helppo ja se käy hyvin työssä, koska sille on mahdollista saada hyvinkin kapea kaista. Suodattimen periaate on se, että ulostulo otetaan vastuksen yli, jolloin kaikki pienet taajuudet vaimenevat kollektorin vaikutuksesta, kun kollektorin induktanssi on suuri ja vastaavasti kaikki isot taajuudet kelan vaikutuksesta. Näin ollen, oikein mitoitetuna, resonanssitaajuudella kelan ja kondensaattorin induktanssi on todella pieni, jolloin koko sisäänmeno signaali saadaan ulostuloon.

RLC -suodattimen Q eli hyvyysluku määrittelee kaistanleveyden: mitä suurempi Q-luku on, niin sitä kapeampi on kaista ja mitä pienempi on Q-luku, niin sitä leveämpi on kaista. Q-lukua kasvattamalla myös aika, kuinka kauan piiri resonoi kasvaa eli sillä kestää kauemmin tasaantua signaalin mentyä läpi. [13]



Kuva 10. RLC-suodattimen rakenne

Suodattimen kelan ja kondensaattorin arvot määräytyvät siis 200 kHz:n taajuuden mukaan. Kun kelana käytettiin aikaisemminkin käytettyä 100mH:n kelaa, saatiin kondensaattorin arvo laskettua.

$$f = \frac{1}{2\pi\sqrt{L_1 C_1}} \Rightarrow 200\text{kHz} = \frac{1}{2\pi\sqrt{100\mu\text{H} \times C_1}} \quad (3)$$

Tästä (kaava 3) saadaan kondensaattorin arvoksi 66 pF eli kun käytetään E12 -sarjan komponentteja, kondensaattorin arvoksi tulee 68 pF. Tämän jälkeen, haluttiin suunnitella 50 kHz:n kaista, joka on tarkoitukseen sopiva. Tällöin vastukseksi saatiin:

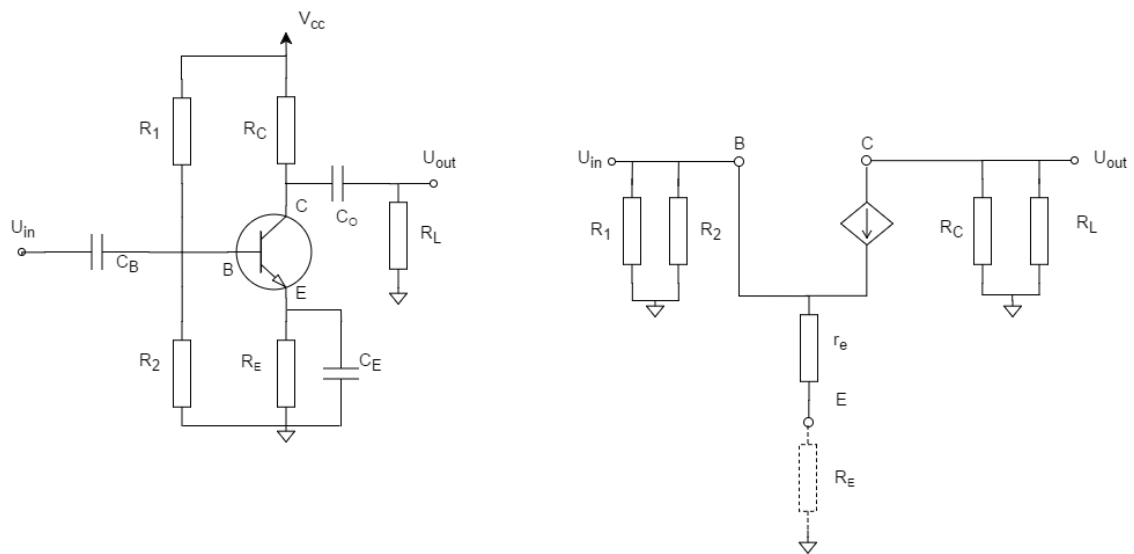
$$BW = \frac{R_1}{2\pi L_1} \Rightarrow R = 2\pi L \times BW = 31,4\Omega \quad (4)$$

Kun muistetaan, että kelalla on oma sisäinen resistanssi, mikä syö liian pienellä vastuksella ulostulevan signaalin voimakkuutta, valittiin hiukan suurempi 47 Ω vastus, mikä silti mahdollisti tarpeeksi hyvän kaistan. [14]

3.2.3 Vahvistin

Tämän jälkeen, kun suodatin oli mitoitettu, alettiin pohtia vahvistinta. Päädyttiin käyttämään 3 asteista CE-vahvistinta, jonka viimeinen aste toimisi kytkimenä 0 ja 3,3V jännitteille. CE-vahvistimeen päädyttiin, koska sen ominaisuudet ovat tarkoitukseen sopivat: sillä pystytään saavuttamaan korkea vahvistus korkeallakin taajuudella. Transistorivahvistimen valmistuskustannukset ovat myös nimelliset, niin tämä sopi lähtökohdiltaan varsin hyvin.

Yleistä CE-vahvistimesta



Kuva 11. CE -vahvistimien piirikaavio ja piensignaali ac-malli

Ulostulo saadaan transistorin kollektorilta (kuva 11) ja ulostulo erotellaan kondensaattori C_c :llä, jotta DC-jännitetaso ei tule ulostuloon. Kondensaattori C_B estää virran kulkemisen lähdön suuntaan ja C_E :llä saadaan ac-vahvistus maksimoitua, koska silloin emitterillä vaikuttaa vain piensignaali vastus r_e , koska C_E oikosulkee R_E vastuksen AC -signaalilla. Kondensaattoreilla C_B ja C_E pystytään myös tekemään alhaisten taajuuksien suodatusta, mutta tämä yksinään ei olisi meidän työssämme riittänyt, koska ensisijaisena suodattimena kondensaattorit ovat varsin heikonlainen ratkaisu, ainakin tämän työn tarkoitukseen.

Vastuksilla R_1 ja R_2 saadaan mitoitettua sopiva jännite kannalle, sekä ne vaikuttavat ratkaisevasti transistorin sisäänmeno resistanssiin. R_E vastus taas stabiloi kytkentää niin, että saadaan stabiili virta kytkennälle, joten vahvistus ei muutu lämpötilan tai muiden transistorin vahvistuskertoimeen vaikuttavien tekijöiden mukaan. Kollektorivastuksella säädelään DC -jännitetaso ulostulolle, ja yleinen periaate on asettaa se puoliväliin käyttöjännitettä, jolloin vahvistettava signaali saadaan molemmin puolin vahvistettua maksimiinsa, eikä negatiivinen tai positiivinen puolijakso ala leikkaantua ennen toista.

Active region parameters

Condition: $U_{CE} > 0.2V$, $U_{BE} > 0.7V$

$$\begin{aligned} I_E \approx I_C \approx I_B \times h_{FE} & & r_e = R_C / U_T \\ U_T \approx 25mV & & r_{out} = R_L \parallel R_C \\ A_u \approx R_C / r_e & & r_{in} = R_1 \parallel R_2 \parallel (r_e \times h_{FE}) \\ U_E = U_B + 0.7V \\ U_C = U_{CC} - (R_C \times I_C) \\ U_B \approx (R_2 / (R_1 + R_2)) \times U_{CC} \end{aligned}$$

Kuva 12. Transistorin virta ja jännitelaskut aktiivitulassa [15].

Teoreettisesti laskettuna yhden vahvistinasteen maksimivahvistus voidaan laskea käyttöjännitteen (3,3 V avulla). Tämä saavutetaan käytännössä, kun U_E on mahdollisimman alhainen eli liki 0 ja jotta transistori on aktiivitulassaan, sen U_{BE} on oltava yli 0,7V ja U_{CE} yli 0,2 V (kuva 12). Näin ollen maksimivahvistuksella U_{RC} on 3,1 V ja sievennettynä, kun U_{RC} jaetaan transistorin lämpötilakäyttäytymistä kuvaavalla U_T :lla (~25 mV), saadaan lähellä nolaa olevalla R_E (~70 - 100 Ω vastuksen arvolla) 124 -kertainen maksimivahvistus. Käytännössä kumminkin, kun halutaan luoda stabiilisti toimiva vahvistin, halutaan tasainen virta emitterille. [15]

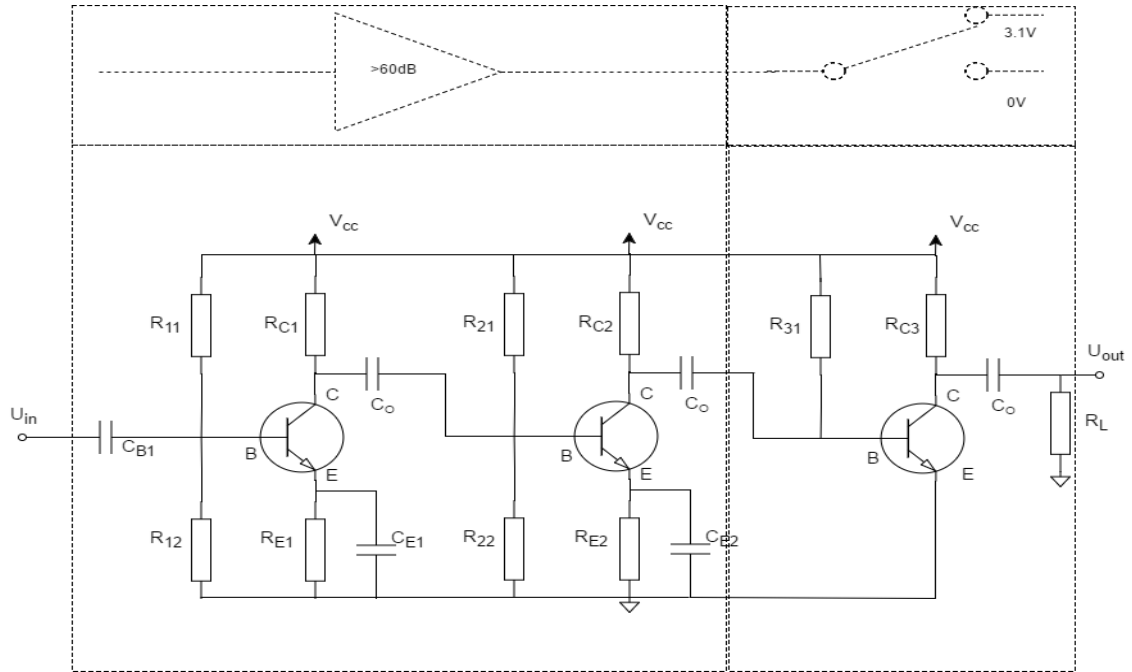
3-asteinen transistorivahvistin

Tärkeintä suunnittelua aloittaessa tietysti oli tieto siitä, kuinka vahva signaali tulisi vahvistaa. Lähtökohtaisesti tiedostettiin, että signaali voimakkuus olisi muutama mV, mutta lähettimen valmistuttua saatiin myös mitattua tilanne meidän lähetyksellä.

Taulukko 2. Vastaanotetun signaalin voimakkuus etäisyyden mukaan, 36V lähetejännitteellä

Etäisyys [cm]	5	10	20	30	40	50	60	70	80	90	100
Jännite [mV _{p-p}]	71,6	46,0	18,8	16,4	10,0	8,20	4,88	4,02	3,04	2,64	>2,00
Jännite [mV _p]	35,6	22,8	9,20	8,20	5,00	4,20	2,44	2,01	~1,60	~1,30	>1,00

Kuten taulukosta nähdään (taulukko 2), vastaanotettava amplitudi laskee aika nopeasti etäisyyden mukaan ja myös pystyttiin todeta todeksi se, että kun lähetyksellä ja vastaanotolla on metrin etäisyys, amplitudi todellakin on vain noin 1 – 2 mV. Näillä lähtökohdilla pystyttiin lähtöä vahvistinta suunnittelemaan.

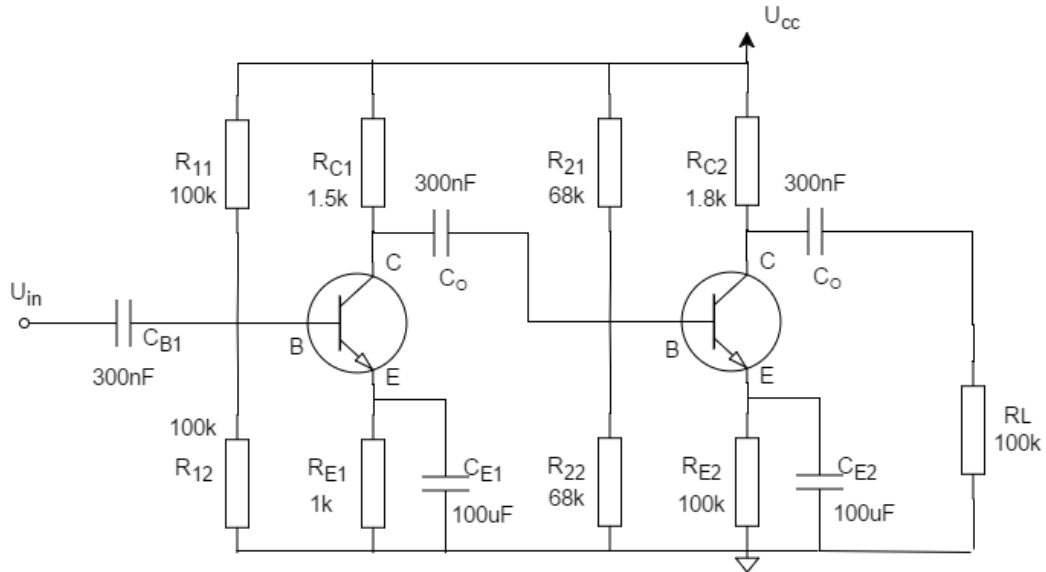


Kuva 13. Vastaanoton vahvistuksen rakenne

Päädettiin 3-transistoriseen kytkentään (kuva 13), jossa 2 ensimmäistä vaihetta toimii 2-asteisena CE-vahvistimena, joiden avulla nostetaan tuleva signaali tarpeeksi korkeaksi, että se pystyy toimimaan ohjausjännitteenä viimeiselle asteelle, joka toimii kytkimenä alhaisen ja korkean jännitetasen välillä (korkea taso hiukan alle 3,3 V käyttöjännite).

Vahvistus

Taulukosta nähtiin, että kaukana kohteesta signaalitaso on 1 mV, lähdettiin suunnittelemaan signaalille n. 60 dB vahvistusta, jolloin se ainakin riittäisi muuttamaan viimeisen asteen tilan saturaatiosta cut off -tilaan, eli laskemaan viimeisen asteen kannalle tulevan jännitteen alle 0,7 V tasolle.



Kuva 14. Vastaanottimen CE-asteiden mitoitus

Lähtökohtaisesti lähdettiin suunnittelemaan porrastetusti niin, että ensimmäiseltä asteelta saadaan n. 25 dB vahvistus ja jälkimmäiseltä reilumpi 35 dB:n vahvistus. Tämä tarkoitti, että I_E mitoitettiin molemmissa tapauksissa n. 0,9 mA:iin, koska 2. vahvistinas-teen ulostulo mitoitettiin puoliväliin käyttöjännitettä, eli 1,65 V: iin. Näin ollen, kun ensimmäisestä asteesta haluttiin n. 20- kertainen vahvistus, sen kollektori vastukseksi saatiin 560Ω ja toisella asteella, kun haluttiin 35 dB vahvistus, saatiin $1,8 \text{ k}\Omega$. Kumminkin käytännön mittauksissa huomattiin, että pienellä kollektorivastuksella häiriöt kasvoivat enemmän ensimmäisessä asteessa, joten lopulta se mitoitettiin $1,5 \text{ k}\Omega$ kollektorivastuk- sella, jolloin kokonaisvahvistus kasvoi ja U_C :n mitoitus noin puolivälissä käyttöjännitettä. Molemmissa asteissa oli myös 1 k vastus emitterillä, joilla stabiloitiin virran määrä kyt- kennöissä. Laskennallisesti yksittäisten asteiden vahvistukset olivat $A_{U1} = 48$ ja $A_{U2} = 61,6$. Kokonaisuuden vahvistuksen laskennassa piti ottaa molempien asteiden tulo ja lähtö- resistanssit huomioon (laskut kuvan 12 mukaisesti) ja kuormana toimi kytkinasteen läh- töimpedanssi, mikä oli suoraan R_{31} arvo.

$$A_{U(TOTAL)} = \left(\frac{r_{i2}}{r_{i2} + r_{o1}} \times A_{U1} \right) \times \left(\frac{R_L}{R_L + r_{o2}} \times A_{U2} \right) \Rightarrow$$

$$\left(\frac{6,96 \text{ k}\Omega}{6,96 \text{ k}\Omega + 1,5 \text{ k}\Omega} \times 48,0 \right) \times \left(\frac{100 \text{ k}\Omega}{100 \text{ k}\Omega + 1,8 \text{ k}\Omega} \times 61,6 \right) = 2389,5 = 67,56 \text{ dB}(5)$$

Näin saatiin huomattavasti yli 60 dB:n vahvistus (kaava 5), joka riitti hyvin ohjaamaan kytkintä.

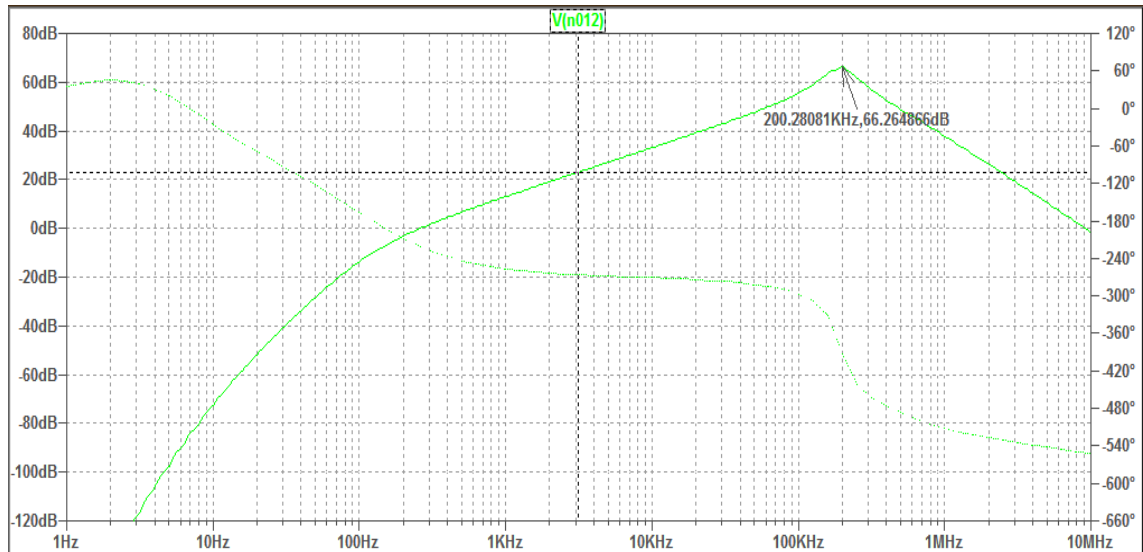
Kytkinaste

Viimeisenä toimiva transistoriasteen toiminta kytkimenä perustuu siihen, että kun emitterillä ei ole vastusta, niin emitterin virran suuruus on niin korkea, että se on normaali tilaltaan saturaatiossa, niin tämä tarkoittaa, että ulostulo on alhaisella jännitetasolla eli 0 V:ssa. Kun kannalle C_B :n kautta tulee tarpeeksi vahva jännite, joka riittää tiputtamaan U_{BE} arvon alle 0,7 V (transistorin BE-diodin kynnysjännitteen), saadaan transistori cut off -tilaan ja silloin ei kulje ollenkaan virtaa, joten ulostulossa on myös suoraan korkea jännitetaso, eli 3,3 V. U_{R3} myös mitoitettiin melko korkeaksi (100 k Ω), jotta saatiin mahdollisimman hyvin aikaisemmalta asteelta vahvistus ulos, koska viimeisen asteen sisäänmenoresistanssi on suoraan kannalla oleva vastus.

3.2.4 Simulointi ja mittaukset

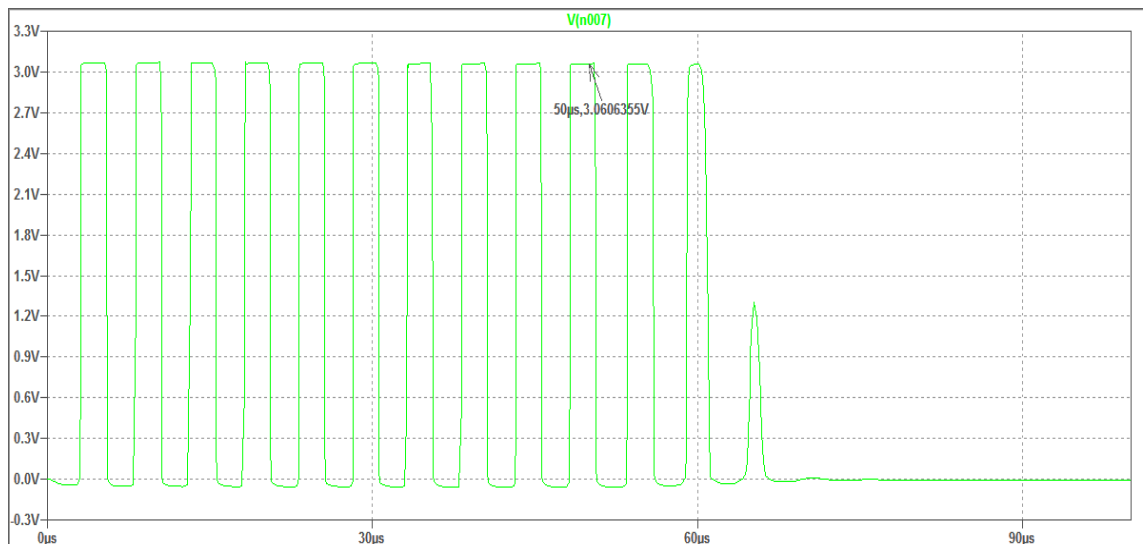
Simulointi

Teoreettisesti toimivan piirin toiminta tarkistettiin vielä LTspice -simulaattorilla, jolla pystyttiin tukea laskettuja malleja. Aluksi simuloitiin AC -analyysi suodattimen ja vahvistinasteiden jälkeen, jotta nähtiin, että vahvistus on mitoitetulla tasolla.



Kuva 15. LTspice-AC-simulaatio vahvistinasteen ja suodattimen jälkeen

LTspice -simulaatio (kuva 15) tuki mitoitettuja arvoja aika hyvin. Pienet heitot tuloksissa tietysti johtuu simulaatiomallien ja laskennallisten eroista, mutta yleisesti ottaen vahvistus oli jokseenkin odotetunlainen. Myöskin suodattimen vaimennus oli tarpeeksi hyvällä tasolla simulaation mukaan. Tämän jälkeen vielä testattiin 10 pulssin 0,001 V jännitteellä simulointijännite ajan funktiona kytkinaste mukana ollessa, jotta nähtiin, vastaako tulos odotettua.

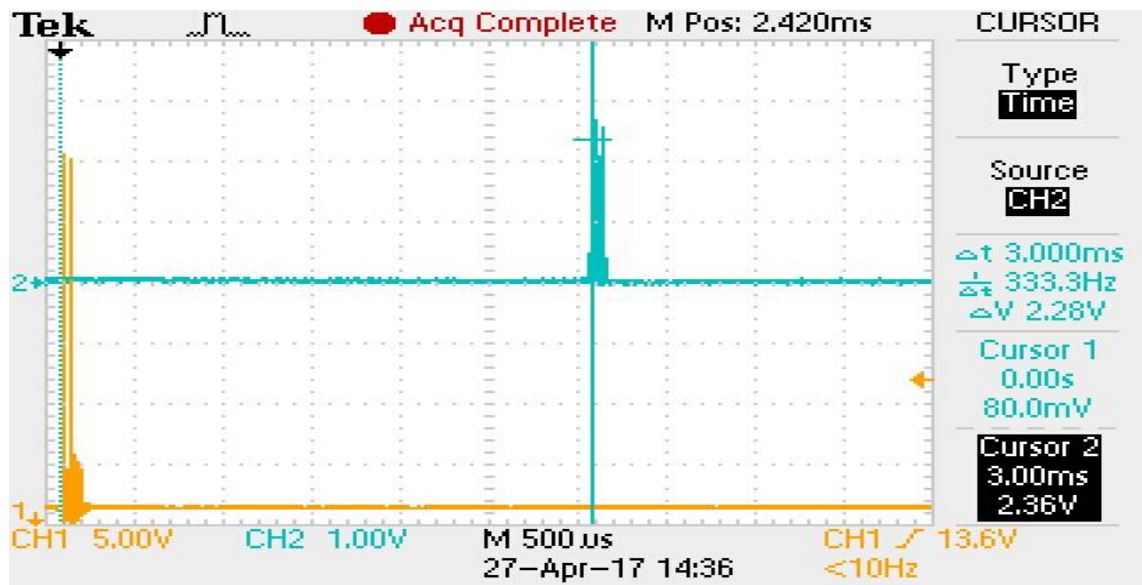


Kuva 16. Koko vastaanoton transient -simulaatio ulostulosta 1 mV signaalilla.

Transient -ajolla nähtiin, että 1 mV signaali vahvistui hyvin ja vain vähän jälkiväreilyä tuli näkyviin simulaatiotilanteessa, mikä viestii, että vastaanoton ideologia oli hyvä. Kun nähtiin simulaattorilla, että tulokset ovat toivotut, voitiin piirilevy tehdä ja toteuttaa.

Mittaukset

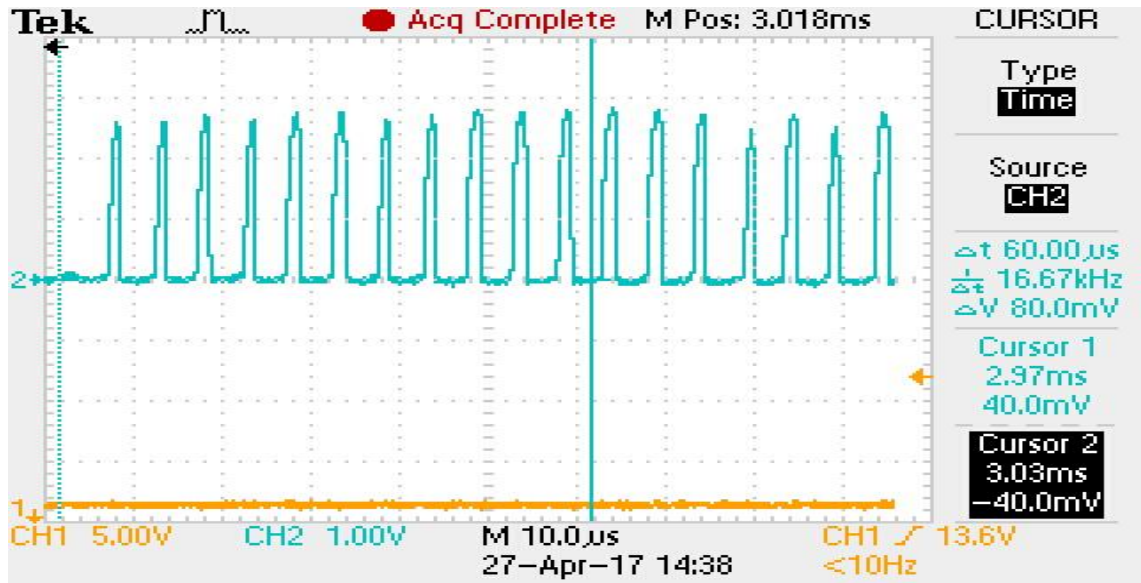
Kun piirilevy oli saatu tehtyä, tehtiin mittauksia lähettimellä ja vastaanottimella. Mittaukset tehtiin 1 m pitkällä levyllä, millä pystyttiin asettamaan lähetin ja vastaanotin kohtisuoraan aina yhteen metriin asti ja jännitelähteinä Turun Ammattikorkeakoulun jännitelähteet. Tulokset otettiin talteen oskilloskoopilla tekemällä yksittäisen näytteenoton lähetyksestä, jotta pystyi näkemään viiveen lähetyksen ja vastaanoton välillä, joten tiesi, että kyseessä oli oikea vastaanotettu signaali.



Kuva 17. Oskilloskooppi kuva 1 m etäisyydeltä

Mittauksissa huomattiin (kuva 17), että jopa 1 m päässä pulssi nähdään vielä. Pulssien välinen viive oli 3 ms, joten kun tiedetään että ääniaalto kulkee 343 m/s, voidaan kuvasta päätellä etäisyyden olevan tarkalleen 1,029 m. [16]

Vielä tarkentamalla vastaanottoon, pystyttiin näkemään millainen lähetetyn 10 pulssin signaali oli tarkalleen.



Kuva 18. Tarkennettu kuva vastaanotetusta signaalista 1 m etäisyydeltä.

Kuten tarkennuksesta nähdään, maksimietäisyydellä jännite on hiukan tippunut, mutta silti $\sim 2,5$ V:in tasolla, joten se on mikrokontrollerilla hyvin luettavissa, koska sen korkean jännitteen raja on 2,31 V. Tämän siis pitäisi riittää etäisyyden mittaamiseksi myös yhden metrin etäisyydellä, eli 50 cm päässä olevasta objektista kimmokkeen kautta. [17]

4 OHJELMOINTI

Työn toisena tärkeänä osuutena elektroniikkasuunnittelun lisäksi oli mikroprosessori STM32L476RG:n ohjelmointi. STMicroelectronics on ranskalais-italialainen puolijohdevalmistaja, joka toimii nykyään useissa maissa. Kyseisen prosessorin käyttäminen työssä oli luontevaa sen useiden ominaisuuksien vuoksi, sekä sen alhaisen virrankulutuksen ja 1,8 – 3,6 V käyttöjännitteen takia, jonka vuoksi se pystyisi toimimaan itsenäisenä laitteena 3,6 V paristolla. [18] [19]

Ohjelmointityössä auttoi STMicroelectronicsin luoma STM32CubeMX -ohjelma. Sen avulla pystyttiin alustamaan rekisterit eri porteille sen graafisen käyttöliittymän avulla tarpeen mukaan. Ohjelma pystyi kääntämään alustukset C-kielellä EWARM -ohjelmointialustalle, jolla oli itselle luontevaa ohjelmoida prosessorin toiminta mittausprosessia varten. EWARM:lla pystyi myös testaamaan STM32- mikroprosessoreitten ajoa virtuaalisesti, koska se on ARM -pohjaisille prosessoreille suunnattu alusta. Prosessorin alustana oli Nucleo64, joka mahdollisti suoran koodin lataamisen prosessorille ja ulostulojen ja sisäänmenon kytkemisen johtimilla [20] [21]

4.1 Lähtökohdat

Lähtökohtaisesti aloitettiin miettimään, mitkä ovat ohjelmoinnilta vaadittavat asiat. Prosessorin tarvitsi generoida 200kHz:n signaali, tietyn pulssimäärän verran, koska lähete ei saanut olla liian pitkä, että pystyttiin mittaamaan lyhyetkin etäisyydet. Prosessorin olisi myös pystyttävä myös laskemaan kulunut aika, sillä hetkellä, kun se vastaanottaa kaikeksignaalin. Tämän koko prosessin olisi myös tarkoitus toimia herätteenä, eli toisin sanoen pystyä herättämään prosessori lepotilasta, niin se pystyisi teoreettisesti suorittamaan ohjelman pyörityksen paristolla, ilman että pariston virta menisi nopeasti tyhjäksi.

4.2 Toteutus

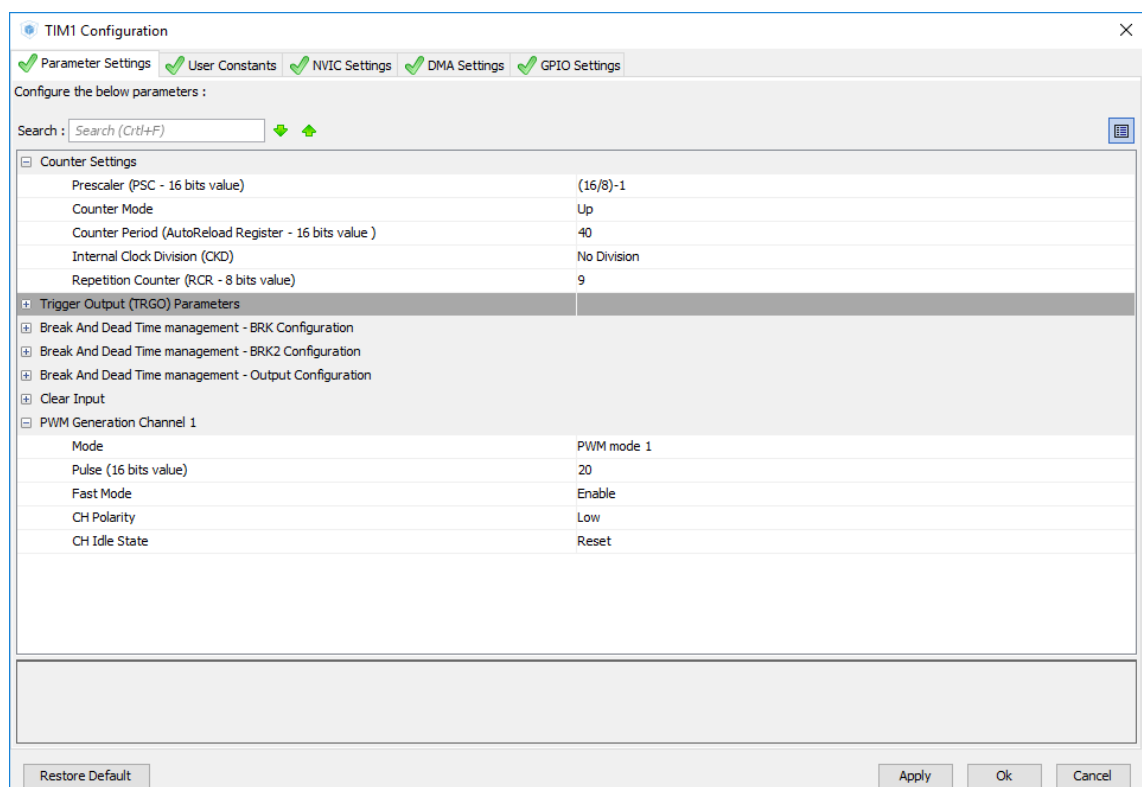
Toteutusta lähdettiin kokoamaan osissa ja sen jälkeen, kun osat oli todettu toimivaksi, alettiin siitä kokoamaan toimivaa kokonaisuutta. Seuraavaksi esitellään eri osioiden toiminta.

4.2.1 200 kHz:n signaalin lähettäminen

Signaalin tuottaminen

200 kHz:n signaalin tuottamisessa lähdettiin selvittämään, miten saataisiin tietynpituista pulssia tuotettua tietyn aikaan prosessorilta, koska tärkeää oli saada tasaista 200 kHz:n pulssia niin, että se katkeaa myöskin halutun ajan päästä, esimerkiksi 10 tai 20 pulssin päästä, jotta lähetysaika olisi pieni. Ajan mittaamista varten mikroprosessoreissa on TIMER -rekisterit, joilla pystytään prosessorin kellotaajuuden perusteella laskemaan askelia: 1 askel on aikana kellotaajuuden käänteisluku, joten askelien määrän laskemalla saadaan tiedettyä kulunut aika.

STM32L476RG:ssä oli tätä varten valmiiksi määritelty kaksi kehittyntä 16-bittistä TIMER -rekisteriä (TIMER1, TIMER8), joilla pystyttiin tuottamaan yksittäisiä pulseja halutun määrän verran PWM portin ulostulona. Tämän alustus pystyttiin tekemään STM32Cube:lla, joka nopeutti asian hoitamista. [22]



Kuva 19. TIM1- rekisterin alustaminen STM32Cube- ohjelmalla.

Kuten nähdään (kuva 19), oli rekisterin alustaminen melko suoraviivaista STM32Cubella. TIMER- rekistereillä pystytään aina antamaan esijako- arvo, millä pystytään kellotaajuutta laskemaan järjestelmän pääkellotaajuudesta. Järjestelmän kellona toimi 16 MHz: inen HSI -kello, koska sitä pystyi myös käyttämään ulkoisella jännitelähteellä, kuten paristolla, joten sen avulla ohjelma pystyisi myös tarvittaessa pyörimään pariston kautta. Tämä esijaettiin Prescaler -rekisterillä 8 MHz:iin, koska se oli tarpeeksi tarkka tähän tarkoitukseen ja sen avulla saatiin tarkat arvot rekistereihin.[23]

Ulostulevan signaalin modulointi tapahtui Counter Period- ja Pulse -rekisterien avulla. Molemmat olivat 16-bittisiä rekistereitä, joten niiden avulla pystyttiin tuottamaan 200kHz:in signaali ilman, että pääkellotaajuutta laskettiin TIMER:ia varten, mikä mahdollisti mahdollisimman tarkan kellotaajuuden. Counter Period -rekisterillä määriteltiin koko pulssin pituus ja Pulse -rekisterillä se, kuinka kauan pulssista se on HIGH- tasolla. Näillä siis pystyttiin myös tarpeen tullen muuttamaan pulssisuhdetta, mutta koska haluttiin tuottaa tasainen 50/50 pulssi, oli päällä oleva aika puolet kokonaispulssista. Rekistereihin haluttiin saada laskettujen askelten määrä 8 MHz:n taajuudella, jolla sitten muodostettiin haluttu taajuus, kun signaalin taso vaihtui halutun jaksonajan jälkeen. [24]

$$\frac{1}{8\text{MHz}} \times \text{Counter Period} = 5 \mu\text{s} \Rightarrow \text{Counter Period} = 40(6)$$

Vastaavalla tavalla (Kaava 6), lasketaan Pulse -rekisteriin arvo, niin että se on puolet ajasta eli 2,5 μs korkeana. Ulostulo tapahtui GPIO -portista A8, koska siihen pystyttiin TIMER1:n ensimmäisen kanavan ulostulo ajamaan [25].

Herätteenä lähettäminen

Kun signaalin tuottamisen alustus oli valmis, oli aika pohtia järkevää ratkaisua sen lähettämiseen käyttäjän toimesta. Tässä oli luontevaa valita niin, että se tapahtui napin painalluksesta, testaamisenkin vuoksi. Tämän oli tapahduttava keskeytyksen avulla, jotta sillä pystyisi myös prosessorin herättämään lepotoiminnosta, ja muutenkin yleisesti on suositavaa tehdä mikroprosessorilla sellainen toteutus, että varsinaisessa main -loopissa ei tapahdu mitään aktiivisia toimintoja, joten se mahdollisuuksien mukaan säästää virtaa, koska sen ei koko ajan tarvitse suorittaa komentoja.

Näin ollen tarvitsi alustaa GPIO- portti keskeytykseksi, eli prosessori lopettaisi sen hetkisen toimintansa main -ohjelmassa tai heräisi lepotoiminnosta, ja tekisi keskeytyksessä

määritetyn koodin. Näin ollen, kun portti olisi määritetty sisäänmenoportiksi, jonka keskeytys käynnistyisi jännitetason vaihdosta, sitä olisi helppo ohjata painokytkimellä, joka on kytketty käyttöjännitteeseen.

```
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin : PB10 */
    GPIO_InitStructure.Pin = GPIO_PIN_10;
    GPIO_InitStructure.Mode = GPIO_MODE_IT_RISING;
    GPIO_InitStructure.Pull = GPIO_PULLDOWN;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);

    /* EXTI interrupt init*/
    HAL_NVIC_SetPriority(EXTI15_10_IRQn, 2, 0);
    HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
}
```

Kuva 20. GPIO -portin alustus keskeytystä varten.

Portin alustus (kuva 20) tapahtui STM-EWARM-käännöksellä mikrokontrollerille tyypilliseen tyyliin, vaikkakin hiukan suoraviivaisemmin kontrollerin oman GPIO -alustusrakenteen mukaisesti: aluksi aktivoidaan kello portille, sen jälkeen valitaan konfiguroitava pinni, jonka jälkeen määritellään, halutaanko keskeytyksen tapahtuvan, kun se havaitsee muutoksen matalasta jännitteestä korkeaan ("rising edge") vai toisinpäin ("falling edge"), tai molemmissa. Sitten määritettiin GPIO:n perustila sisäisellä alas- tai ylösveto vastuksella, joko PULLUP(=HIGH) tai PULLDOWN=(LOW). Viimeisenä tarvittiin keskeytyksen toimintaa varten NVIC -rekisterin alustukset, joissa sallitaan keskeytys ja määritetään sen priorisointi muihin keskeytyksiin nähden. [26]

Kun keskeytys saatiin toimimaan painonapista, pystyi signaalia tuottavan TIMER:in käynnistyksen sisällyttää keskeytykseen, jolloin signaalin lähetys oli keskeytykseen perustuva.

4.2.2 Viiveen taltiointi

Ajan taltiointi oli ohjelmoinnin toinen tärkeä osa alue. Tämän toteutus oli loppujen lopuksi yksinkertainen, koska TIMER:iin tutuessa lähetystä varten, oli hyvin selvää, että myös ajanottoa varten oli varsinaisen valmis rakenne. Mikrokontrolleriin pystyi määrittämään TIMER -keskeytyksen, jossa TIMER:iin on yhdistetty GPIO -porttiin tulevaan keskeytykseen ja GPIO- portin havaitessa keskeytyksen käynnistävän tason, se tekee TIMER keskeytyksen. Näin ollen keskeytyksessä pystyi heti aluksi taltioimaan TIMER -rekisterin laskurin arvon, johon on taltioitu siihen mennessä kulunut aika TIMER:in käynnistyksestä. Näin ollen, kun tämäkin TIMER käynnistettiin heti lähetyksen perään, pystyttiin vain kytkemään siihen yhdistetty pinni vastaanottimen ulostuloon, jolloin vastaanottoon saapuvan signaalin saapuessa saatiin aika lähetyksestä vastaanottoon.

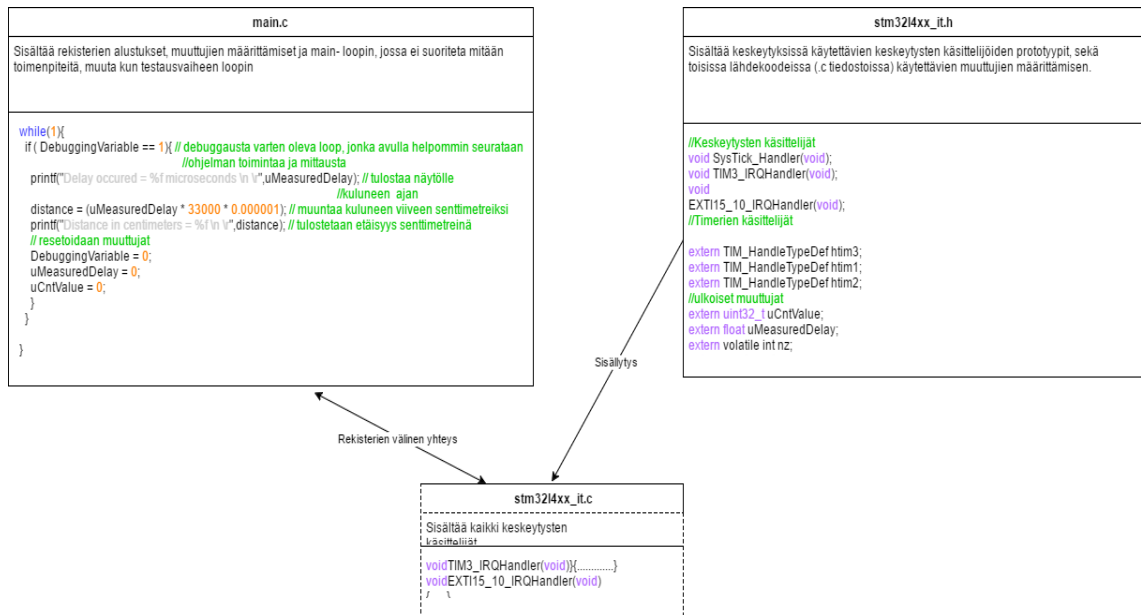
TIMER:in alustuksessa oli tärkeää tarkistaa, että sen kellotaajuus oli määritelty niin, että se pystyisi odottamaan tarpeeksi pitkän ajan, vaikka vastaanotin olisi metrin päässä. Näin ollen, kun työssä käytettiin 16-bittistä TIMER -rekisteriä, oli maksimi askelten määrä $2^{16}-1$, eli 65 535 askelta.

$$\frac{1}{f_{clock}} \times 65535 = 2,94ms \Rightarrow f_{clock,max} < 22,2MHz \quad (7)$$

Kun tiedettiin, että aika kulkee n. 340 m/s, eli 1 m matkaan ultraäänellä kuluisi ~2,94 ms, näin ollen kellotaajuuden yksittäisen askeleen eli jakson ajan ($1 / \text{taajuus}$), saatiin laskettua maksimikellotaajuus (kaava 7). Huomattiin, että näin laajalla rekisterillä ei esijakoa välttämättä tarvitsisi käyttää, kun maksimikellotaajuus olisi joka tapauksessa 16 MHz.

4.2.3 Kokonaisuus

Kokonaisuus saatiin kasaan, kun erilliset osat oli todettu toimiviksi, ja oli luontevaa alkaa yhdistellä palasia kokonaisuudeksi.



Kuva 21. Selitekaavio päätiedostoista.

Näin ollen käyttäjän editoimat päätiedostot olivat kuvan mukaiset (kuva 21). Main- tiedostossa käytännössä tehtiin kaikki alustukset, ja lisäksi pääohjelmassa oli testaamista varten testaussilmukka, jolla vain tehtiin tulosten lukeminen helpommaksi ja nollattiin niitä pitemmän testauksen vuoksi. "stm32l4xx_it.c" -lähdekoodin prototyyppi alustukset tehtiin "stm32l4xx_it.h" -headeritiedostossa. Seuraavaksi käydään läpi, mitä keskeytysten lähdekoodi sisälsi, koska siellä tapahtui varsinainen mittausprosessi.

Keskeytykset

Keskeytykset tapahtuivat kahdessa keskeytyksessä: painonapin painalluksen havaitseva keskeytyksessä ja saapuvan vastaanotetun signaalin keskeytyksessä.

```

void EXTI15_10_IRQHandler(void)
Painonapilla aktivoitavan keskeytyksen käsittelijä (portti
PB10)

int LoopVariable = 1; // muuttuja, jolla varmistetaan koko keskeytyksen hallinnan pyöriminen alusta loppuun

while( LoopVariable == 1){
HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1); // Signaalin tuottaminen TIMER1:en ensimmäisellä kanavalla PA8 porttiin(D7 nucleo)
uint32_t CounterValue = 0; // määritetään muuttuja jolla seurataan TIMER:in laskurin arvoa
nz = 1;
HAL_TIM_IC_Start_IT (&htim3, TIM_CHANNEL_2); // käynnistetään TIMER3 keskeytys ja ajanlaskenta viivettä varten
while (nz == 1)
{
see = __HAL_TIM_GET_COUNTER(&htim3);
if (100 >= CounterValue){ // jos laskurin arvo on alle 100, voidaan todeta ettei vastaanotto ole havainnut signaalia vastaanotossa
printf("no objects within 8.192 milliseconds, exiting");
nz = 0;
uMeasuredDelay = (65535 - (float) see)* 0.125; // muutetaan kulunut aika askelina mikrosekunneiksi, kertomalla askelian
//määrä yhden askeleen ajalla 1/8MHz = 0.125ms
DebuggingVariable = 1;
}
}
printf(" \n Exit interruptL_ \n \n"); // ilmoitus tulostuksella I/O moduuliin debuggauksessa keskeytyksen päättymisestä
LoopVariable = 0; // Resetoidaan loop- muuttuja jotta päästään pois while- loopista
printf("measured delay = %f\n \n",uMeasuredDelay); // debuggaus ilmoituksia
HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_10); // keskeytyksen ilmoitusten resetointi, että se pystytään havaitsemaan uudelleen.

```

Kuva 22. Painonappikeskeytyksen ohjelma.

Ohjelman varsinainen osuus käynnistyy, kun nappia painetaan, ja silloin siirrytään "EXT15_10_IRQHandler" -funktioon (kuva22). Tärkeimmät asiat painonapilla käynnistetävissä keskeytyksessä on signaalin generoinnin aloittaminen, toisen keskeytyksen käynnistäminen ja sen jälkeen vain TIMER:n laskurin ajan seuraaminen, että tiedetään koska laskuri pääsee loppuun ja näin ollen saadaan ohjelma päätettyä, jos ei vastaanotettavaa signaalia ole havaittu, eli toisinsanoen toinen keskeytys ei koskaan tapahdu.

```

void TIM3_IRQHandler(void)
Vastaanottosignaalista herätettävän keskeytyksen käsittelijä (portti
PC7)

uCntValue = HAL_TIM_ReadCapturedValue(&htim3, TIM_CHANNEL_2); // taltioidaan kulunut aika
askelina muuttuun
uMeasuredDelay = (65535 - (float) uCntValue)* 0.125; // muutetaan kulunut aika askelina
mikrosekunneiksi, kertomalla askelian määrä yhden
askeleen ajalla 1/8MHz = 0.125ms:illa
nz = 0; // nollataan muuttuja, jonka loopissa päivitetään counterin arvoa
toisessa keskeytyksessä
DebuggingVariable = 1; // muutetaan debuggausmuuttujan arvo, että päästään
keskeytyksen jälkeen siihen
HAL_TIM_IC_Stop(&htim3, TIM_CHANNEL_2); // pysäytetään timeri
__HAL_TIM_SET_COUNTER(&htim3,0xFFFF); // palautetaan timerin alotusarvo maksimiin
HAL_TIM_IRQHandler(&htim3); // keskeytyksen ilmoitusten resetointi, että se pystytään
havaitsemaan uudelleen.

```

Kuva 23. Timer -keskeytyksen ohjelma.

Kun TIMER- keskeytys tapahtuu (kuva 23) eli kun vastaanotettava signaali havaitaan, hypätään pois painonappikeskeytyksen silmukasta, otetaan etäisyys talteen, palautetaan TIMER:in laskurin arvo täyteen. Tämän jälkeen päivitetään painonappikeskeytyksen silmukan arvo, että kun TIMER- keskeytyksestä poistutaan, päästään myös toisesta keskeytyksestä pois. Molemmissa keskeytyksissä lopuksi nollataan keskeytyksen omat ilmoitusrekisterit siihen tarkoitettulla funktiolla.

Kun ohjelma oli tässä vaiheessa, päästiin testaamaan kokonaisuuden toimintaa.

5 LOPPUTESTAUS

5.1 Testaus

Lopputestaukset päästiin suorittamaan koodin valmistuttua. Mikroprosessoria käytettiin EWARM -alustan virtuaalitestauksessa, jolloin se oli yhdistettynä USB-liittimellä tietokoneeseen. Suoraviivaisten mittaustulosten saamiseksi käytössä oli 1 metrin mittainen testauskisko, jonka toisessa päässä tasolla oli lähetyksen vastaanotto lähellä mikroprosessoria ja toisella levyllä lähetyispää, jota pystyttiin siirtämään merkittyjen etäisyyksien päähän vastaanottimesta, jotta saatiin luotettavat mittaustulokset. Mittaukset suoritettiin yksittäislaukaisu- toiminnalla oskilloskoopilla samaan aikaan kun ohjelma mittasi omat arvonsa, jotta pystyttiin vertaamaan laskennallisia arvoja oikeisiin. Ohjelmassa huomattiin olevan käytetty hiukan alakanttiin olevaa ajan nopeuden arviointia (330 m/s), joten laskennalliset arvot todellisesta viiveestä signaalien välillä on laskettu tätä käyttäen asian kompensoimiseksi, koska tällä ei ole käytännön merkitystä.

Taulukko 3. Ohjelman laskemat mittausetäisyydet verrattuna oikeisiin.

Viive [ms]	Etäisyys todellisesta viiveestä[cm]	Etäisyys ohjelmalta [cm]	Virheprosentti [%]
0.32	10.69	10.95	2.36
0.63	20.72	20.92	0.94
0.92	30.36	30.67	1.01
1.21	39.93	40.24	0.77
1.49	49.17	49.94	1.54
1.80	59.40	59.81	0.69
2.09	68.97	69.22	0.36
2.37	78.21	78.51	0.38
2.62	86.46	88.80	2.64
3.00	99.00	99.21	0.21

Tästä nähdään, että virhemarginaali on kohtalaisen pieni, kun käytössä on tämän tasoinen prosessori. Virheprosentin heittelyä korkeilla etäisyyksillä selittää todennäköisesti oskilloskoopin resoluutio, kun mitattiin viiveen aikaa sen omilla kursoreilla. Minimiaskelellisella kursorilla oli 0,01 ms, joten sen vaikutus on +/- 0,3 – 1 %. Voidaan siis todeta, että realistisesti virheprosentin luokka oli 2,5 - 3,0 %. Tämän voisi todeta olevan kohtuullisen odotettava tulos, kun mittaus toimii, koska prosessorin funktioiden toteutus aiheuttaa tietysti viivettä.

5.2 Kehitysideat ja ongelmat

Ongelmia työssä koitui suurimmaksi osaksi elektroniikkasuunnittelun puolella häiriöiden suodattamisessa. Työssä huomasi vastaanottoa suunnitellessa sen, että kun vahvistetaan todella pientä signaalia, voi häiriöt olla hyvinkin suuri ongelma, mitä teoreettisessa laskennassa ja simuloitessa ei näe. Tämän kaltaisessa vastaanottosuunnittelussa ei ensimmäisen asteen vahvistus saanut olla liian korkea, tai muuten vastaanotto vahvistii liikaa häiriöitä. Työssä huomasi myös, että kunnan maadoituksen puuttuminen aiheutti häiriöitä lisää ja näiden parempi suodatus olisi tärkeää kehitystyötä. Välillä aikaa kului ehkä liikaa erilaisten ratkaisujen hakemisessa sen sijaan, että olisi määrätietoisesti jatkanut tämän ratkaisun edistämistä.

Kehitysideana olisi toteuttaa sama kytkentä yhdellä sensorilla, jolloin vastaanotto ja lähetys olisi samalla levyllä. Tämä toteutus parantaisi kohteeseen mittaamisen tarkkuutta, koska heijastuvan signaalin palauttaminen samaan pisteeseen on helpompaa, kun erillään olevilla sensoreilla. Tämä suunnittelu vaatisi jonkinasteisen puskurin vastaanoton ja lähetyksen välille, koska vahvistettu lähetyssignaali jäisi tällaisenaan soimaan liian kauaksi aikaan vastaanottoon, ja vastaanotettava signaali ei näkyisi.

5.3 Lopuksi

Työssä lähdettiin toteuttamaan etäisyysmittausta 1 m etäisyydelle ja siinä onnistuttiin. Toteutus voisi olla vielä paremmin tehty häiriöiden suodatuksen kanssa ja yhden sensorin toteutus olisi käytännön sovellutuksissa käytännöllisempi kuin kahden sensorin, mutta tässä työssä toimii ideologia etäisyysmittauksesta hyvin. Opinnäytetyössä joutui

käymään läpi molempia kohtalaisen tärkeitä elektroniikkasuunnittelijan osa-alueita: teholähteiden suunnittelua sekä vahvistinasteen mitoittamista. Tässä työssä sai myös kevyellä tasolla harjoittaa mikroprosessorin ohjelmointia.

LÄHTEET

- [1]. Matti Koramo, "Ultraäänietäisyysmittari" [www-dokumentti]
 Saatavilla: <https://www.theseus.fi/bitstream/handle/10024/12895/TTI4SMattiKo.pdf?sequence=1>
 13.04.2017
- [2] <http://searchsecurity.techtarget.com/definition/ultrasound> 13.4.2017
- [3] https://en.wikipedia.org/wiki/Ultrasonic_transducer 14.4.2017
- [4] <http://chinaultrasound.com/product/200-khz-ultrasonic-transducer-short-distance-measurement-12cm-2meters-ta0200kb/> 14.4.2017
- [5] "STM32L476xx". STMicroelectronics s1-2. 2015 (pdf-dokumentti)
- [6] <http://www.learnabout-electronics.org/PSU/psu32.php> 20.4.2017
- [7] http://harisi.com/products_pdf/BYS10-45-E3-TR3.pdf 20.4.2017
- [8] http://assets.nexperia.com/documents/data-sheet/BC847_SER.pdf 20.4.2017
- [9] <http://alejandria.ccm.itesm.mx/biblioteca/digital/liele/pcmc.pdf> 22.4.2017
- [10] <http://www.biosono.com/PrmtLgd/PrmtLgd.php?id=TrnsRlc> 22.4.2017
- [11] Introduction to Piezoelectric Actuators and Transducers [www-dokumentti]
 Saatavilla https://www.researchgate.net/publication/235110841_Introduction_to_Piezoelectric_Actuators_and_Transducers 22.4.2017
- [12] Tekniikan kaavasto. 2013. Porvoo: Bookwell Oy. 22.4.2017
- [13] <http://www.radio-electronics.com/info/formulae/q-quality-factor/basics-tutorial.php>
 23.04.2017
- [14] Robert L Boylestad, 2014, Introductory Circuit Analysis s.879-890. USA: Pearson 23.4.2017
- [15] Oppimateriaali: Vahvistinasteet, Huhtinen Ville, Turun Ammattikorkeakoulu 24.4.2017
- [16] <http://www.sengpielaudio.com/SpeedOfSoundPressure.pdf> 28.04.2017
- [17] STMicroelectronics" STM32L476xx" s142. 2015 (pdf-dokumentti) 28.4.2017
- [18] <https://en.wikipedia.org/wiki/STMicroelectronics> 28.4.2017
- [19] <http://www.st.com/en/microcontrollers/stm32l476rg.html> 28.4.2017
- [20] <http://www.st.com/en/development-tools/stm32cubemx.html> 28.4.2017
- [21] <http://www.st.com/en/development-tools/iar-ewarm.html> 28.4.2017
- [22] STMicroelectronics" STM32L476xx" s43. 2015 (pdf-dokumentti) 28.4.2017
- [23] STMicroelectronics" STM32L476xx" s21. 2015 (pdf-dokumentti) 28.4.2017
- [24] STMicroelectronics "General-purpose timer cookbook" s34-36. 2016 (pdf-dokumentti)
 28.4.2017

[25] STMicroelectronics” STM32L476xx” s68. 2017 (pdf-dokumentti) 28.4.2017

[26] STMicroelectronics “Description of STM32L4 HAL and Low-layer drivers” s53-54. 29.4.2017