



# **KUUROJEN HERÄTYSKELLO**

Bluetooth-ohjattu relekytkentä

Klaus Tamminen

Panu Puurtinen

Opinnäytetyö  
Toukokuu 2013  
Tietotekniikka  
Tietoliikennetekniikka ja  
tietoverkot  
Ohjelmistotekniikka

TAMPEREEN AMMATTIKORKEAKOULU

Tampere University of Applied Sciences

## ALKUSANAT

Tämän opinnäytetyön ovat tehneet kaksi opiskelijaa tietoliikennekoulutusohjelman eri suuntautumisvaihtoehdoista. Aihe oli Klaus Tammisen keksimä laite, jolle voisi mahdollisesti olla eri käyttötarkoituksia esimerkiksi sokeiden herätyskellona tai näyttelyta-  
loissa älypuhelimella ohjattuna kytkimenä.

Eri suuntautumisten puolesta työ jaettiin pääasiallisesti kahteen osaan. Panu Puurtinen vastasi ohjelmiston suunnittelusta ja toteutuksesta ja Klausin vastuualue oli laitteiden liittäminen ja niiden keskinäinen keskustelu. Työn rakentaminen toteutettiin kuitenkin yhdessä ja siten, että molemmat osaisivat ja tajuaisivat kummatkin opinnäytetyön osa-  
alueet.

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Tietoliikennetekniikan ja tietoverkkojen suuntautumisvaihtoehto (Klaus Tamminen)  
Ohjelmistotekniikan suuntautumisvaihtoehto (Panu Puurtinen)

KLAUS TAMMINEN & PANU PUURTINEN:

Kuurojen herätyskello  
Bluetooth-ohjattu relekytkentä

Opinnäytetyö 28 sivua, joista liitteitä 2 sivua  
Toukokuu 2013

---

Laitteemme toiminta perustuu releeseen, joka toimii kytkimen lailla. Sitä ohjataan Bluetooth-vastaanottimella ja tekemällämme Android-sovelluksella, joka tässä tapauksessa on herätyskello.

Kuurojen herätyskelloissa on yleensä kaksi osaa, joista toinen vilkuttaa valoja ja toinen on tärinähälytín. Laitteemme tarkoitus on sopia mihin tahansa kytkimeen, esimerkiksi pöytälamppuun tai kattovalaisimeen, ja tärinähälyttimenä käytämme ohjaavan kännykän tärinähälytystä.

Opinnäytetyömme on jaettu kahteen pääalueeseen: tietoliikenne- sekä ohjelmistopuoleen. Tietoliikenneosassa käsitellään eri komponentit ja niiden toiminta keskenään. Ohjelmistopuolella käsittelemme tekemämme Android-sovelluksen sekä muita pienempiä ohjelmia, joita laitteen käyttö tarvitsee.

---

Asiasanat: bluetooth, rele, android, herätyskello, kuurot

## ABSTRACT

Tampere University of Applied Sciences  
Information Technology Degree Programme  
Telecommunications Engineering and Networks Orientation (Klaus Tamminen)  
Software Engineering Orientation (Panu Puurtinen)

KLAUS TAMMINEN & PANU PUURTINEN:

Alarm clock for deaf

Bluetooth controlled relay

Bachelor's thesis 28 pages, appendices 2 pages

May 2013

---

Our equipment is based on the relay, which acts as a switch. It is operated via a Bluetooth receiver and Android application which we made, which in this case is an alarm clock.

Alarm clocks for deaf people usually contain two parts: flashing lights and a vibrating alarm. Our switch equipment is designed to fit inside existing switch, for example on a table lamp or ceiling lamp and the mobile phone is acting as a vibrating alarm.

The thesis is divided into two main areas: telecommunications, as well as the software side. The telecommunication part explains the various components and how they work with each other. Software part explains our Android app, as well as other smaller programs which are needed to operate the device.

---

Key words: bluetooth, relay, android, alarm clock, deaf

## SISÄLLYS

1	JOHDANTO.....	7
2	LAITTEISTO .....	8
2.1	Bluetooth.....	8
2.1.1	Bluetooth-tekniikka lyhyesti .....	10
2.1.2	Bluetoothin ohjelmointi .....	10
2.2	Arduino Nano .....	11
2.2.1	Arduinon ohjelmointi .....	12
2.2.2	Arduinon ohjelmisto .....	13
2.3	Rele .....	14
2.4	Älypuhelin .....	15
3	RAKENTAMINEN.....	16
3.1	Ohjaussignaali.....	17
3.2	Käyttöjännitteet.....	18
3.3	Kytkinosa .....	18
4	ANDROID-OHJELMISTO.....	20
4.1	Ohjelmiston toiminta .....	20
4.2	Käyttöliittymä .....	23
5	TULEVAISUUS JA SUUNNITELMAT.....	24
5.1	Laitteen kehitys.....	24
5.2	Markkinointi .....	25
6	POHDINTA.....	26
	LÄHTEET.....	27
	LIITTEET .....	28

**LYHENTEET JA TERMIT**

AND	vertailuoperaattori
Android	Googlen kehittämä käyttöjärjestelmä mobiililaitteille
API	ohjelmointirajapinta
AT -komennot	Bluetoothin ohjelmoinnissa käytetyt komennot
bps	bittinä sekunnissa
D8-liitäntä	digitaalinen liitäntä numero 8
Eclipse	ohjelmointiympäristö
GFSK	taajuusmodulaatio
IF	ehtolause
IrDA	infrapunaan perustuva langaton tiedonsiirtojärjestelmä
Java	ohjelmointikieli
LED	valodiodi (käytetään usein myös lamppuna)
Linux	käyttöjärjestelmäydin
MB/s	tiedonsiirtonopeus megabittinä sekunnissa
MHz	taajuuden yksikkö megahertsi
RX	vastaanotto tietoliikenteessä
TTL	transistori-transistori -kytkentä
TX	lähetys tietoliikenteessä
USB	lisälaitteen liitäntä
WLAN	internet-verkkoon perustuva langaton tiedonsiirtojärjestelmä

## 1 JOHDANTO

Ihmiset kaipaavat aina vain uusia keksintöjä, ja usein ne keksinnöt, jotka helpottavat elämää, ovat suosituimpia. Vaikkakin keksintömme ratkaisee lähes olemattoman ongelman, näemme sillä silti käyttötarkoituksia.

Opinnäytetyöksemme suunnittelimme ja rakensimme kuurojen herätyskellon, joka toimii langattomasti ja älypuhelimien avulla. Opinnäytetyö koostuu rakenteellisesta prototyypistä, ohjelmistosta, seminaarista sekä kypsyyskokeesta. Näistä aiheista tässä raportissa käsitellään prototyyppiä sekä ohjelmistoa.

Kuurojen herätyskellot ovat kalliita, ja niiden saatavuus on hyvinkin rajallinen. Verrattumme muihin markkinoilla oleviin tuotteisiin uskomme pystyvämme valmistamaan tässä opinnäytetyössä esittelemämme laitteemme noin 20 € halvemmalla kappalehinnalla. Laitettamme voisi myös käyttää minkä tahansa kytkimen ohjaamiseen älypuhelimella, joten laite itsessään avaa monia eri käyttötarkoituksia ja käyttökohteita.

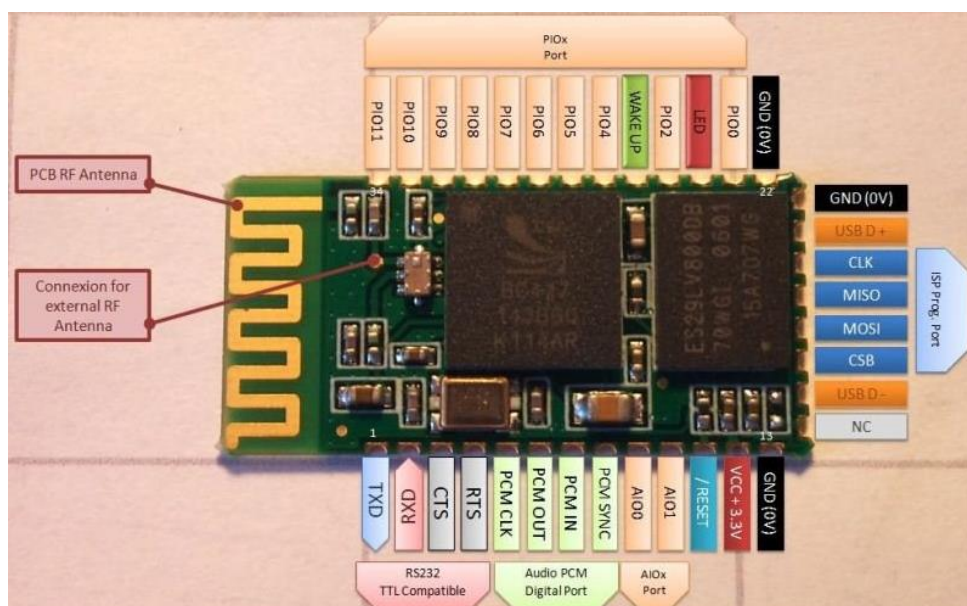
## 2 LAITTEISTO

Tässä luvussa esittelemme eri osa-alueet, joita Bluetooth-ohjattu relekytkentä tarvitsee toimiakseen. Lisäksi perustelemme myös muutamia ratkaisujamme rakentamassamme prototyypissä, jolla oli tarkoitus lähinnä testata ideamme toimintaa ja tekemäämme Android-sovellusta.

### 2.1 Bluetooth

Jotta laitteemme saisi ohjauksensa langattomasti, täytyi meidän valita jokin langattoman tekniikan järjestelmä. Sen avulla yhdistäisimme relekytkennän ja sitä ohjaavan älypuhelin. Tutustuttuamme eri vaihtoehtoihin totesimme Bluetoothin sopivan tarpeisiimme hyvin hintansa, kokonsa ja ”alueellisuutensa” kannalta.

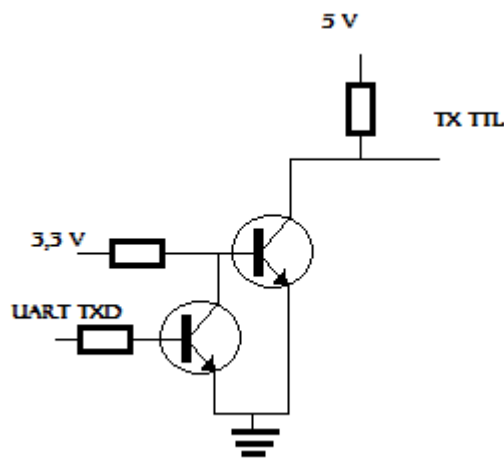
Vaihtoehtoina olisi ollut esimerkiksi IrDA tai WLAN. IrDA:n totesimme liian hankalaksi, koska se vaatii jatkuvan näköyhteyden lähettimen ja vastaanottimen välille. WLAN-tuotteet sen sijaan ovat turhan kalliita, ja tahdoimme laitteemme toimivan myös ilman langattoman verkon tukiasemaa. Langaton verkko mahdollistaa myös väärinkäytöksiä ja muita riskejä, kuten kesämökin saunan päälle laittaminen pitkän matkan päästä, mikä on erittäin suuri paloturvallisuusriski. Tämän takia halusimme järjestelmän, jota ohjataan samasta huoneesta, mutta kuitenkin joustavasti.



KUVA 1. Bluetooth-vastaanotin (Alibaba: Bluetooth-vastaanotin 2013)



Prototyypiä rakentaessamme emme kiinnittäneet huomiota Bluetoothin version valintaan vaan tyydyimme siihen, mitä Tampereen ammattikorkeakoulun tietoliikennepajalta saimme (Kuva 1.). Bluetooth-vastaanottimemme käyttää Bluetooth-versio 2:ta, jonka keskimääräinen tiedonsiirtonopeus on 2,1 MB/s. Tämä riittää meille helposti, koska emme siirrä suuria määriä dataa vaan mikroprosessori Arduinolla määrää releen toimintaa lähettämämme pulssin mukaisesti. Prototyypimme kantoalue on noin kymmenen metriä, mikä tässä kohden vielä riittää, koska laitteemme on tarkoituskin olla alueellinen. Luvussa ”Tulevaisuus ja suunnitelmat” käymme läpi myös Bluetooth-versioita ja niiden sopivuutta käyttötarkoituksiimme.



KUVA 2. Bluetoothin lähetys

Kuvassa 2 on tarkoitus hieman valottaa Bluetoothin lähetyspuolen rakennetta. Lähetyspuoli on laitteemme tärkeimpiä osa-alueita Bluetoothin osalta, sillä emme käytännössä tarvitse kuin jatkuvan käyttöjännitteen Bluetoothille, jotta se mainostaa itseään ja pysyy päällä, sekä antennin ja lähetyksen, joka ohjaa seuraavaa laitetta.

Bluetooth-moduulin älypuhelimelta vastaanottama lähetys täytyy saada kulkemaan Bluetoothilta Arduinolle siten, että Arduino sen ymmärtää. Bluetooth itsessään toimii sarjamuotoisena, ja useimmat laitteet siirtävät ja ymmärtävät sarjamuotoista dataa TTL:n avulla. Kahden transistorin avulla voidaan muodostaa AND-operaatio, joka ohjaa 5 voltin kulkua ulos lähetyksestä. Vastaanotin, meidän laitteessamme Arduino, osaa sitten tästä ”pätkitystä” jännitteestä kerätä lähettämämme sarjamuotoisen datan ja toimia sen mukaisesti.

### 2.1.1 Bluetooth-tekniikka lyhyesti

Bluetoothin taajuusalue on 2,4000 – 2,4835 GHz, mille mahtuu 79 kanavaa, ja yhden kanavan taajuussiirto on 1 MHz. Useat kanavat mahdollistavat myös paikallisesti usean laitteen samanaikaisen käytön, mutta rajoitetun taajuusalueen takia alueella on mahdollista käyttää vain kahdeksaa laitetta master - slave -tyyppisesti. Modulaationa kyseisessä tekniikassa on GFSK-modulaatio, joka on taajuuden muutoksiin pohjautuvaa koodausta. GFSK-modulaatiossa binäärinen 1 ja 0 sisällytetään kantoaallon pituuteen poikkeuttamalla kantoaallon perustaajuutta. (Wikipedia: Bluetooth 2013.)

### 2.1.2 Bluetoothin ohjelmointi

Käyttämäämme kuvan 1. kaltaista Bluetooth-vastaanotinta voidaan ohjelmoida sarjavyölyn avulla. Vastaanottimessa on sisäistä muistia, minne voidaan tallentaa esimerkiksi siirtonopeus, laitteen nimi ja salasana.

Bluetooth-vastaanottinta ohjelmoitaessa pitää se liittää tietokoneeseen. Vastaanottimesamme (Kuva 1.) oikean laidan liitännät ovat USB-johdolle, eli USB-johto voidaan suoraan juottaa tai telakoida liittimiin.

Moduulin liittämisen jälkeen ohjelmointi sinänsä on hyvin yksinkertaista. Esimerkiksi RS232 Hex Com Tool (Hex Com Tool, 2006) -ohjelmistolla saadaan terminaalilyhteys Bluetoothiin, minkä jälkeen annetaan AT-komennot. AT-komentoja ovat esimerkiksi:

- AT+BAUD4, joka määrittää siirtonopeuden. Numero 4 tulee valmiista taulukosta ja tarkoittaa 9600 bps
- AT+NAMEnimi, jolla asetetaan laitteelle nimi, jota se mainostaa master-laitteille. Esimerkiksi komennolla AT+NAMELAMPPU, näkyisi laite nimellä LAMPPU
- AT+PIN1234, PIN-komento asettaa laitteelle salasanan, joka tässä tapauksessa on 1234. (M0XDC: Bluetoothin ohjelmointi 2011.)

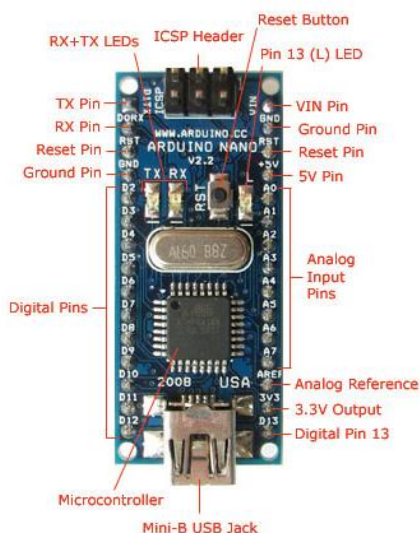
AT-komentoja voi myös käyttää signaalin vahvuuden tarkastamiseen ja muiden ominaisuuksien tarkasteluun, jotta voidaan määrittää laitteen toimivuus tietyssä tilassa. Esimerkiksi komento AT+CSQ kertoo signaalin voimakkuuden.

Prototyyppeä ohjelmoidessamme emme ohjelmoineet Bluetooth-moduulia, koska koimme tehdasasetusten riittävän mainiosti tarkoituksiimme. Tehdasasetuksilla Bluetooth mainosti itseään nimellä Linvor ja sen pin-koodi oli 1234. Myöskin siirtonopeus oli asetettu 9 600 bps:ksi, joka suurimmassa osassa laitteita on vakiosiirtonopeus.

Ajattelimme kuitenkin, että mikäli laitteita pitäisi rakentaa useampia, olisi jonkinlainen telakointiasema näppärä niiden ohjelmoimiseen. Mikäli tekisimme telakointiaseman, jossa olisi valmiina liitoskohdat kuvan 1 oikean laidan liitännöille, jotka on kuvassa nimetty ISP prog -portiksi, voisimme nopeasti ohjelmoida uusia moduuleja erilaisilla nimillä ja salasanoilla juottamatta yhtäkään johtoa kiinni.

## 2.2 Arduino Nano

Arduino Nano on ohjelmoitava mikrokontrolleripiiri, jonka valitsimme pienen koon ja työhön sopivuuden pohjalta. Esimerkiksi käyttöjännitteet käyttämillemme laitteille saadaan Arduinosta todella mainiosti: Bluetooth-vastaanottimemme tarvitsee 3,3 voltia ja releemme tarvitsee 5 voltia käyttöjännitettä. Nämä molemmat saadaan suoraan Arduinon pinneistä, kuten kuvasta 3 voidaan havaita. Oman käyttöjännitteensä Arduino saa USB:stä, ja nykyisin markkinoilla on erittäin pienikokoisia muuntajia, jotka toimivat verkkovirrassa, koska älypuhelinmarkkinat ovat kasvaneet merkittävästi ja ne käyttävät lataukseensa USB:tä.



KUVA 3. Arduino Nano (Arduino Nano 2013)

Rakentamisen alussa pelkäsimme, ettei Arduino kykene pitämään yllä tarpeeksi suurta virtaa ulkoisille laitteillemme, mutta testattuamme kytkentäämme totesimme pelkomme aiheettomiksi.

Kuvissa 1 ja 3 näkyy useita liitäntöjä, ja osa niistä on sellaisia, joita käytämmekin laitteessamme, mutta palaamme niihin vielä kytkentäosiossa. Arduinossa on myös LED-valot lähetykselle ja vastaanotolle. Niiden avulla pystyimme nopeasti tarkastamaan, tuleeko Arduinolle komentoja Bluetoothin yli ollenkaan vai katoavatko ne matkalla.

### **2.2.1 Arduinon ohjelmointi**

Arduinoa voi ohjelmoida myös suoraan USB-kaapelilla, joten sen käyttäminen on vaivatonta ja nopeaa. Myös ohjelmisto Arduinon ohjelmoimiseen löytyi valmiina TAMK:n koneilta (Arduino software 2013).

Ohjelma avattiin kuvakkeestaan kuten kaikki Windows-ohjelmat. Avautuneessa ikkunassa valittiin ensiksi liitin, jonka kautta ohjelmoitava laite löytyisi. Meidän projektissamme se oli tietenkin USB-portti. Tämän jälkeen valitsimme tiedonsiirtonopeuden ja ohjelmoitavan Arduinon mallin. Kun ohjelma oli kirjoitettu, ajettiin se Arduinon muistiin. Tämän jälkeen Arduino voitiin irrottaa tietokoneesta ja kytkeä osaksi relekytkentäämme.

## 2.2.2 Arduinon ohjelmisto

Älykkäinkään mikrokontrolleripiiri ei toimi ilman ohjelmaa, joten teimme prototyyp-  
piämme varten lyhyehkön ohjelmapätkän, jonka tarkoitus on alkaa räpsyttää valoa,  
kun älypuhelimien herätyskello alkaa soida. Kyseinen ohjelmapätkä on seuraava:

```
int val = 0;           // variable for received data
int pin = 8;          // variable for used pin (D8)

void setup() {
  pinMode(pin, OUTPUT);           // sets pin as OUTPUT
  Serial.begin(9600);             // start serial communication at 9600 bps
}

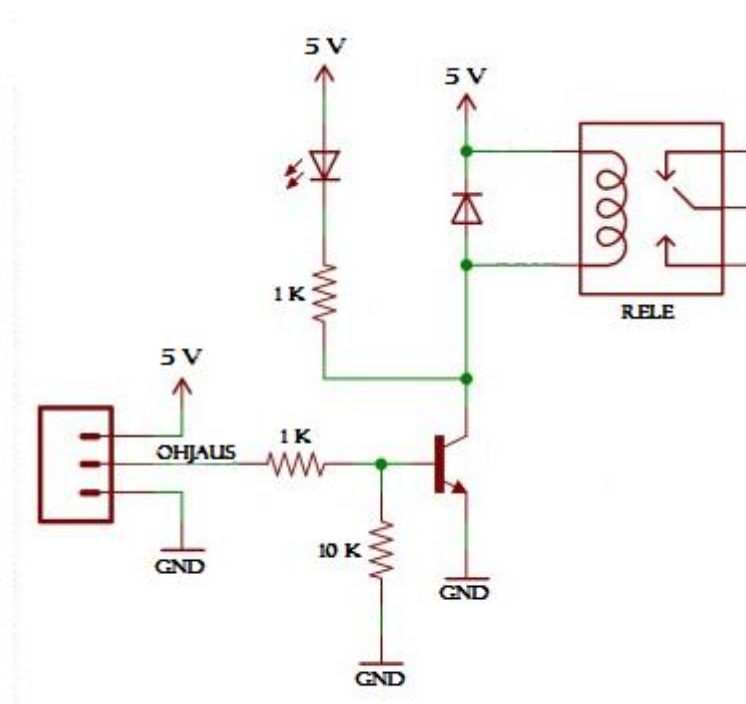
void loop() {
  if( Serial.available() )        // if data is available to read
  {
    val = Serial.read();          // read data and store it in val
  }
  if( val == 1 )                  // if 1 is received, starts flashing lights
  {
    digitalWrite(pin, HIGH);      // turns lights on
    delay(300);
    digitalWrite(pin,LOW);        // turns lights off
    delay(200);
  } else if( val == 0){           // if 0 is received turn lights off
    digitalWrite(pin, LOW);
  }
  delay(100);                     // wait 100 ms for next reading
}
```

Yllä esitetyn Arduinon-ohjelman alussa määritellään muuttujat ja ohjausliitäntä. Sen jälkeen ohjelma siirtyy seuraamaan vastaanottoliitäntäänsä, kuvassa 3 näkyvää vastaanottoliitintään (RX-Pin), ohjauksen varalta. Vastaanotetun viestin ollessa ”1” ohjelma

alkaa käydä läpi if-lausekettaan, joka kytkee valon päälle ja pois nopeassa syklissä. Vilkkutuksen voi lopettaa lähettämällä laitteelle komennon ”0”, jolloin arvo muuttujassa val vaihtuu ja ohjelma lopettaa if-lauseen ja siirtyy else if-lausekkeeseen, jossa ohjelma sammuttaa valon.

### 2.3 Rele

Rele on käyttämistämme laitteista yksinkertaisin. Siinä on verkkovirtapuoli ja ohjauspuoli. Ohjauspuolella on 5 voltin käyttöjännite, jonka avulla rele voi odottaa tulevaa ohjausta. Ja kun ohjaus lopulta tulee Arduinolta, kääntää rele verkkovirtapuolella kytkintä niin, että sähkö alkaa kulkea kytkimen läpi esimerkiksi lamppuun.



KUVA 4. Releen kytkentä

Yllä on esitetty kytkentä, josta voidaan myös ”lukea” releen toiminta. Vasemmassa laidassa on ohjauspuolen liitin, jonne tuodaan Arduinolta 5 voltin käyttöjännite, ohjaussignaali ja maa. Ohjaussignaalin saapuessa transistorin kannalle alkaa kuvan ylälaidasta LED:in ja releen kelan läpi kulkea sähköä. Tällöin LED kertoo releen olevan ”päällä” ja kela kääntää verkkovirtapuolen kytkimen toiseen asentoon.

## 2.4 Älypuhelin

Älypuhelimia löytyy nykyisin melkein jokaiselta ihmiseltä. Ne ovat käytännössä lähes aina kosketusnäytöllisiä minitietokoneita, joilla voi soittaa puheluita. Älypuhelimet siinänsä toimivat keskenään lähes yhteneväisesti, mutta käyttöjärjestelmät määrittelevät, millaista koodia ja ohjelmistoja kyseinen puhelin ymmärtää. Valitsimme laitteemme testaamiseen ja käyttämiseen Android-käyttöjärjestelmällä toimivan älypuhelimien, koska Android on yleisin käyttöjärjestelmä älypuhelimilla yli 68 %:n markkinaosuudella (IDC – Press Release 2013) ja ryhmällämme oli kokemusta Android-ohjelmoinnista.

Älypuhelimien valinta ohjaavaksi laitteeksi oli lähtökohta projektillämme, sillä ohjelmistoalustana ne tarjoavat laajimman kehitysympäristön ja mahdollistavat erilaisten ominaisuuksien lisäämisen. Lisäksi ohjelmiston jakaminen älypuhelimille on todella helppoa ja lähes ilmaista erilaisten ohjelmistokauppojen, kuten Google Playn, avulla.

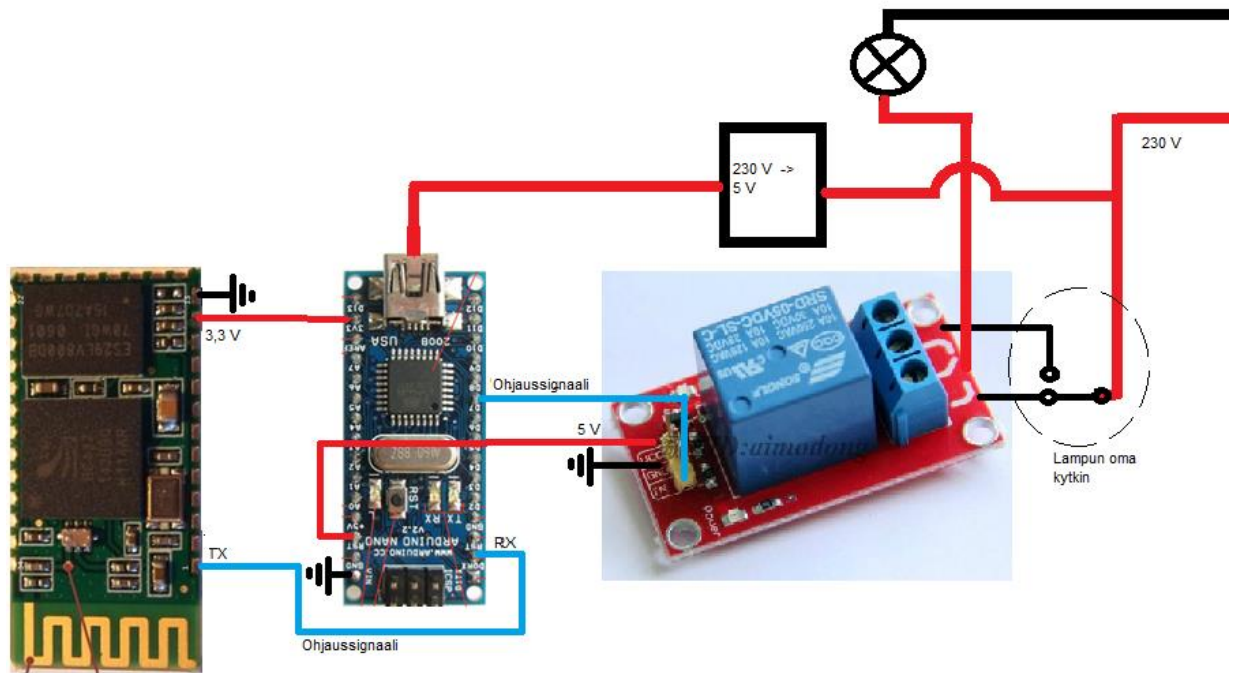
### 3 RAKENTAMINEN

Relekytkentää suunnitellessamme rakensimme prototyypin (kuva 5), jonka avulla voimme esittää laitteemme ja omien ohjelmistojemme toimintaa ja sitä, miten ne reagoivat kyseiseen kytkentään. Kytkentämme on mielestämme helpoin selittää kolmessa osassa: Ohjaussignaalin kulku, käyttöjännitteet ja kytkinosa. Eri laitteiden toimintaa ja merkitystä selitimme jo edellisessä luvussa, joten tässä keskitymme kytkentöihin ja niihin liittyviin asioihin.



KUVA 5. Rakentamamme prototyyppi pöytälampan sisällä.





KUVA 6. Bluetooth-ohjattu relekytkentä

Kuvassa 6 on esitetty relekytkentämme fyysinen osuus kokonaisuudessaan. Eri väreillä on pyritty erottamaan eri osiot kytkennästä, joista seuraavissa luvuissa kerrotaan hieman tarkemmin.

### 3.1 Ohjaussignaali

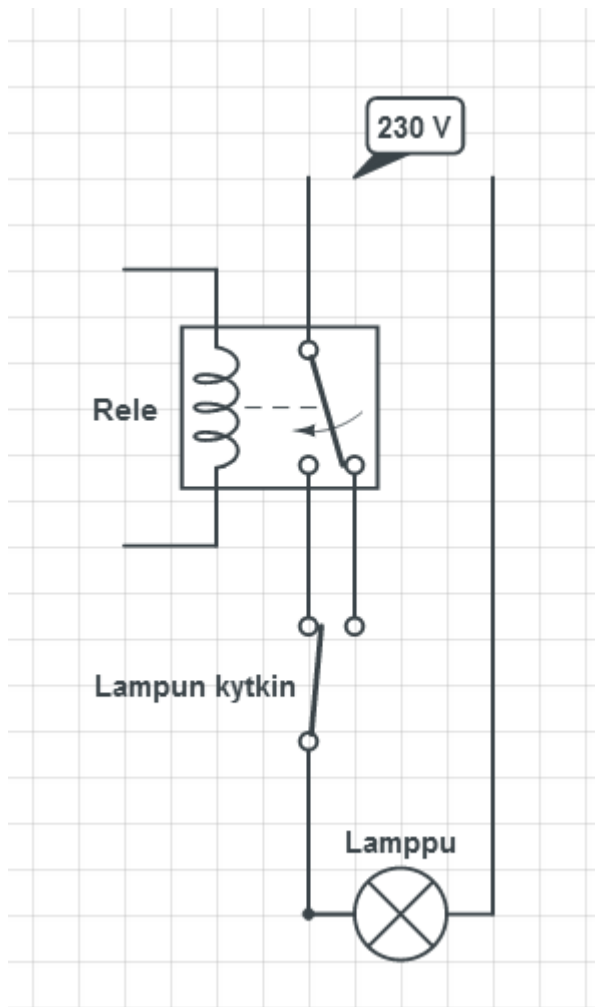
Ohjaussignaali näkyy sinisellä kuvassa 6. Se lähtee vasemman laidan Bluetooth-vastaanottimen lähetyksiinnästä saadessaan älypuhelimelta databitin. Arduino Nano keskellä ottaa Bluetoothilta tulleen bittijonon vastaan RX-liitäntäänsä, minkä jälkeen se tarkastaa oman ohjelmistonsa mukaisesti, onko ohjaus oikeanlainen, jotta se voisi ohjata omaa liitäntäänsä D8:aa. Oikeanlaisen ohjauksen tullessa lähtee ohjaus releen IN-liittimeen. IN-liittimeen tuleva ohjaus muuttaa releen tilaa ja sen oikeassa laidassa 230 V:n puolella keskimäinen kytkimen osa muuttaa tilaansa, jolloin lamppu alkaa vilkkua ohjauksen mukaisesti.

### 3.2 Käyttöjännitteet

Käyttöjännitteet eri laitteille, kuvassa 6 punaisella merkityt yhteydet, saadaan Arduino Nanolta, mutta Arduino itse saa käyttäjännitteensä oman USB-liittimensä kautta. Käytimme muuntajaa, joka antaa meille 5 voltia verkkovirrasta. Sen kytkimme verkkovirtakaapeliin ennen kytkimiämme, jotta käyttöjännite on jatkuva, eikä se katkeilisi kytkimen mukana. Kun Arduino saa oman käyttöjännitteensä, on sillä kaksi liitäntää (Kuva 3.), joista tulee jännitettä. Molemmat näistä jännitelähdöistä sopivat laitteillemme erinomaisesti. Kuvan 5 mukaisesti kytkimme 3,3 voltin jännitteen Bluetoothille ja 5 voltin jännite liitettiin releen käyttöjännitteeksi.

### 3.3 Kytkinosa

Kytkinosa käsittää lampun oman kytkimen ja relekytkimen liitännän eli sen, kuinka verkkovirran kulkua lampulle rajoitetaan. Prototyypissämme kyseinen osa toteutettiin pelkästään releen kytkimellä, koska emme testikäytössä tarvinneet lampun omaa kytkintä ollenkaan.



KUVA 7. Kytkinosa

Kytkinosassa on kaksi kytkintä liitetty toisiinsa kuvan 7 mukaisesti. Kytkennän ansiosta voidaan ohjaus antaa kummasta tahansa kytkimestä, lampun omasta tai releen kytkimestä. Näin varmistetaan, että lamppu toimii normaalisti myös ilman mobiiliyhteyttä.

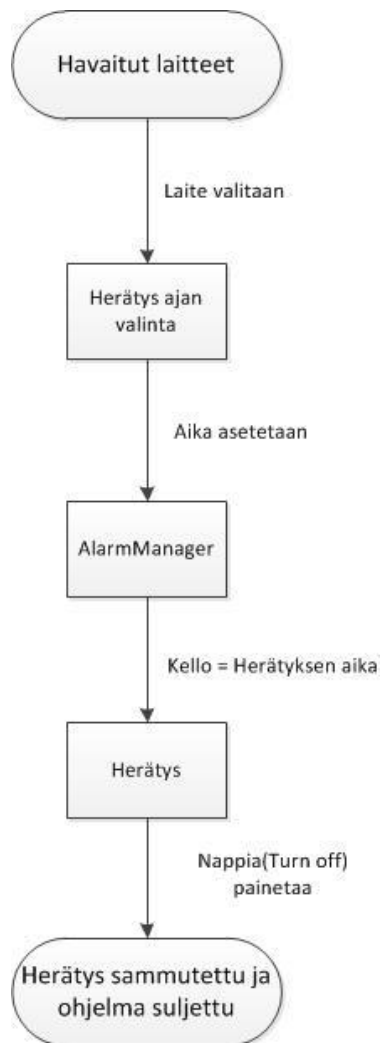
## 4 ANDROID-OHJELMISTO

Android on Googlen mobiililaitteille kehittämä avoimen lähdekoodin käyttöjärjestelmä, joka pyörii Linux-ytimellä. Itse ohjelmistokehitys pyörii Java-kirjastojen päällä, ja näin ollen ohjelmointi tapahtuu Java-kieltä käyttäen.

Android-sovelluksia kehitetään Eclipse-ohjelmointiympäristössä. Kehityksen alussa on valittava kehitettävän ohjelman minimi-API-taso. API-taso määrittää, millaisia ominaisuuksia ohjelmalla on käytettävissä. Yli 90 % Android-laitteista tukee API 10 -tasoa (Android Platform Versions 2013), joten päätimme, että sovelluksemme minimitasoksi tulee API 10, joka vastaa Android-versiota 2.3.3.

### 4.1 Ohjelmiston toiminta

Ohjelmiston toiminta on alla olevan kuvan 8 mukainen. Ohjelmiston alkunäkymässä ovat listattuna kaikki älypuhelimien havaitsemat Bluetooth-laitteet. Käyttäjän valittua haluamansa laitteen avautuu näkymä, jossa käyttäjä pääsee asettamaan herätysajan. Ajan asetuksen jälkeen herätyksen hallinta siirtyy Androidin AlarmManager-olion kontolle, jonka vastuulla on herätysaktiviteetin käynnistys.



KUVA 8. Tilakaavio

Herätysaktiviteetti yrittää muodostaa yhteyden käyttäjän aikaisemmin valitsemaan Bluetooth-laitteeseen. Yhteyden avauduttua ohjelma lähettää alla olevan ohjelmistopätkän mukaisesti ohjausbitin Bluetooth-vastaanottimelle, joka käynnistää valojen vilkkumisen lampussa. Tarvittava ohjelmapätkä on seuraava:

```

Handler mHandler = new Handler(){
    @Override
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        switch(msg.what){
            case SUCCESS_CONNECT:
// Connection successful, trying to send message b to that device
                mSocket = (BluetoothSocket)msg.obj;
                connectedThread =
                new ConnectedThread(mSocket);
                Byte b = 1;
                connectedThread.write(b);
                break;
        }
    }
};

public void write(byte bytes) {
    try {
        mmOutputStream.write(bytes);
    } catch (IOException e) { }
}

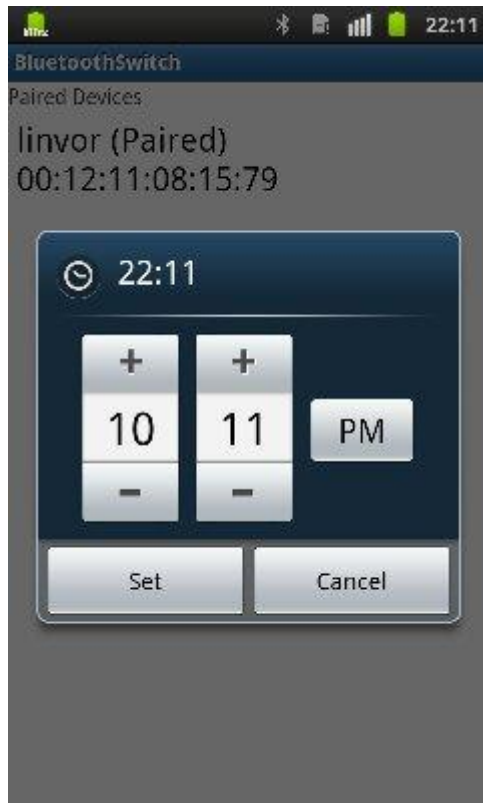
```

Herätys pysyy aktiivisena, kunnes käyttäjä kuittaa herätyksen. Tämän jälkeen sovellus sulkee Bluetooth-yhteyden ja poistuu ohjelmasta. Puhelimen ja Bluetooth-vastaanottimen välinen paritus säilyy, koska paritus hoidetaan ohjelman ulkopuolella. Tämä mahdollistaa ohjelman uudelleenkäynnistämisen niin, ettei laitteita tarvitse joka kerta parittaa uudelleen.

Ohjelmiston toiminta on esitetty tilakaaviossa (kuva 7.) askel askeleelta tapahtuvien siirtymien avulla. Tilakaaviot on suunniteltu esittämään monimutkaisempia kokonaisuuksia yksinkertaistetulla, toimintaan pohjautuvalla kuvaajalla, jolloin toimintaan tutustuvat saavat nopeasti käsityksen järjestelmän toiminnasta.

## 4.2 Käyttöliittymä

Käyttöliittymän suhteen pyrittiin mahdollisimman selkeään ja helppokäyttöiseen toteutukseen, jotta sovellus sopisi kaikenikäisille ja -tasoisille käyttäjille. Kaikki turhat napulat ja ominaisuudet jätettiin pois, ja ensimmäinen prototyyppi onkin hyvin yksinkertainen herätyskello-sovellus.



KUVA 9. Käyttöliittymä

Kuvassa 9 käyttäjä on juuri valinnut laitteen, jolle hän haluaa asettaa herätyksen. Tämän jälkeen avautuu ponnahdusikkuna, josta käyttäjä valitsee kellonajan, milloin hän haluaa herätyksen tapahtuvan. Käyttäjän painettua kuvassa 9 näkyvää set-näppäintä hälytys on asetettu eikä käyttäjältä vaadita muita toimenpiteitä.

## 5 TULEVAISUUS JA SUUNNITELMAT

Laitetta suunnitellessamme pohdimme myös laitteen tulevaisuutta ja sitä millaisia muutoksia tai parannuksia siihen tulisi tehdä. Useiden eri käyttötarkoitusten ansiosta erilaisia kehityskohteita onkin useita ja mahdollisuuksia rajoittaa vain mielikuvituksemme. Laite ja ohjelmisto itsessään kaipaavat vielä vähän kehitystä, mutta järjestelmä alkaa olla jo niin sanotussa beetavaiheessa.

### 5.1 Laitteen kehitys

Laitetta suunnitellessamme oli tarkoituksemme tehdä mahdollisimman pienikokoinen ja huomaamaton laite, jonka voisi kätkeä normaalin seinäkytkimen sisällä olevaan laatikkoon. Prototyypimme ei sinne mahdu, mutta parannusideoita on jo ehtinyt muodostua. Esimerkiksi sellainen olisi Bluetooth-vastaanotin, joka osaa suoraan käsitellä saamaansa dataa ja muodostaa oman ohjaussignaalin, jolla voisimme ohjata relettä. Tämä pienentäisi kytkentäämme jo huomattavasti, sillä voisimme poistaa Arduinon kytkennästäamme kokonaan.

Itse Bluetooth kannattaisi toteuttaa uudella versioon 3 kuuluvalla low energy -mallilla, jonka kuluttama teho on todella vähäinen ja se soveltuisi näin ollen paremmin myös kulutuksestaan tarkkoihin mallitaloihin asuntomessuille. Tietenkin on myös mahdollista kohteesta riippuen valita käytettävä Bluetooth-versio, jotta saataisiin mahdollisimman sopiva kantama verrattuna käytettyyn tehoon.

Kuten jo Bluetooth-luvussa mainitsin, mikäli alamme tuotteistaa ja tuottaa laitettamme, tulisi meidän myös rakentaa telakointiasema Bluetooth-moduulia varten, jotta moduulien ohjelmoiminen olisi nopeampaa ja paljon yksinkertaisempaa.

Rele on isokokoinen laite, mutta sen pienentäminen lienee haastavaa. Kun olemme tutkineet markkinoilla olevia vastaavia laitteita, niin emme pidä todennäköisenä, ainakaan tällä hetkellä, että korvaavaa parempaa laitetta löytyisi sopuhinnalla. Suuren koon selittää eri jännitealueiden 5 voltin ja 230 voltin puolet, eikä niitä oikein pysty vaihtamaan mitenkään.



Ohjaava opettajamme mainitsi mahdollisesti työhön sopivista releistä, jotka olisivat pienikokoisempia ja näin ollen sopivampia työhömmе. Mutta emme ainakaan vielä ole löytäneet jännitealueelle ja käyttötarkoitukseemme sopivaa kappaletta. Tosin nyt kun tiedämme mahdollisuuksia olevan, osaamme seurata myös sitä, alkaako pienempikokoisempiakin releitä löytyä halvalla.

Sovelluspuolella uusia ominaisuuksia voi lisätä lähes rajattomasti, ja sovelluskehitys onkin koko ajan työn alla. Seuraavana kehityksenä relekytkennälle on ihan vain peruskytkin (päälle – pois), joka pysyy automaattisesti yhteydessä laitteeseemme aina sen ollessa kantamalla.

Olemme myös miettineet eri käyttötarkoituksia tällaiselle päälle – pois –tyyppiselle kytkimelle. Helppoja kohteita olisivat esimerkiksi autotallin ovet, joiden käyttöön taloyhtiöt nykyisinkin myyvät erillisiä ohjaimia. Ohjaimet poistamalla saisimme käyttäjämukavuutta ja yksinkertaistettuja laitteistoja. Myöskään järjestelmämme hinta ei ole käyttäjälle tai taloyhtiölle paha, joten tuotteelle voisi olla menekkiäkin.

## **5.2 Markkinointi**

Tällä hetkellä kartoitamme vielä tarvetta laitteellemme. Olemmekin jo ottaneet yhteyttä Kuurojen keskusliittoon, ja tarkoituksena olisi vielä lähestyä useampia eri asuntomessujen talojen rakentamisesta vastaavia yrityksiä. Klaus Tammisen veli on messutoiminnassa mukana, ja toivommekin saavamme häneltä avustusta muun muassa yhteystietojen muodossa.

Kartoittaaksemme käyttäjämääriä ja tarpeellisuutta järjestämme myös pienimuotoisen kyselyn kuurojen tuttaviemme keskuudessa. Sen toivomme antavan rehellistä palautetta siitä, onko laitteelle tarvetta tai markkinoita.

Mikäli päätämme viedä laitteen markkinoille tutkimusten jälkeen, niin mielestämme loogisin askel olisi mainostaa laitetta ja asennuspalveluja Kuurojen keskusliiton oman julkaisun avulla ja asuntomessujen välityksellä.

## 6 POHDINTA

Työmme oli mielenkiintoinen, ja käytännönläheisyys oli erityisen hyvä lisä kaikelle teoriapohjaiselle opetukselle eri osa-alueista. Ihmettelimmekin, miksei tietoliikennetekniikkapuolen opetukseen kuulu vastaavan laitteen rakennusta tai testausta, jossa esimerkiksi Bluetoothilla ohjataan mikrokontrolleria.

Prototyypin rakentaminen sujui hyvin muutamia vastoinkäymisiä lukuunottamatta. Kokemattomuutemme vuoksi meillä oli muutamia ”töyssyjä” rakennuksessa. Esimerkiksi emme tiedäneet, että Arduino Nano ohjausliitäntää valittaessa ohjattavan liitännän numero ei olekaan se, monesko se on pinnijärjestyksessä, vaan sen nimi. Yritimme useamman päivän ohjata liitäntää numero 11, jonka nimi oli D8. Oikeasti ohjelmistomme kuului laittaa ohjattavaksi liitännäksi 8, joten tällaiset pienet vastoinkäymiset vähän viivyttivät laitteemme valmistumista, mutta mitään suurempia ongelmia ei rakentamisessa ilmennyt.

Android-sovelluksen tekeminen alkoi melko kivuttomasti, koska verkosta löytyi hyvät tutorialit Bluetooth-yhteyksien hallintaan. Suurimmat ongelmat tulivat ponnahdusikkunoiden kanssa, koska vanhat API 10 -tason funktiot, joilla ponnahdusikkunat luotiin, olivat poistuneet käytöstä. Sen seurauksena jouduimme käyttämään tukikirjastoja, joilla ponnahdusikkunat saatiin luotua.

## LÄHTEET

Alibaba: Bluetooth-vastaanotin, 2013. Elektriikan verkkokauppa. Luettu 25.3.2013.  
[http://www.alibaba.com/product-gs/567157895/Hot\\_selling\\_new\\_and\\_original\\_bluetooth.html](http://www.alibaba.com/product-gs/567157895/Hot_selling_new_and_original_bluetooth.html)

Android Platform Versions, 2013. Current Distribution, maaliskuu, 2013. Luettu 2.4.2013.  
<http://developer.android.com/about/dashboards/index.html>

Arduino Nano, 2013. Valmistajan sivu. Luettu 25.3.2013.  
<http://arduino.cc/en/Main/ArduinoBoardNano>

Arduino Nanon kytkennät, 2008. Kytkentäkuvaaja. Luettu 2.4.2013.  
[http://zhpan.com.cn/showpic/?t=1&u=http://www.dfrobot.com.cn/image/data/dfr0010/arduino\\_nano\\_schematic.png](http://zhpan.com.cn/showpic/?t=1&u=http://www.dfrobot.com.cn/image/data/dfr0010/arduino_nano_schematic.png)

Arduino software, 2013. Arduino piirien ohjelmointiin käytettävä ohjelma. Viimeisin versio: 1.0.4. Luettu 27.3.2013. <http://arduino.cc/en/main/software>

Hex Com Tool, 2006. Bluetooth-vastaanottimen ohjelmointi. Luettu 27.3.2013.  
[http://download.cnet.com/RS232-Hex-Com-Tool/3000-2383\\_4-10064973.html](http://download.cnet.com/RS232-Hex-Com-Tool/3000-2383_4-10064973.html)

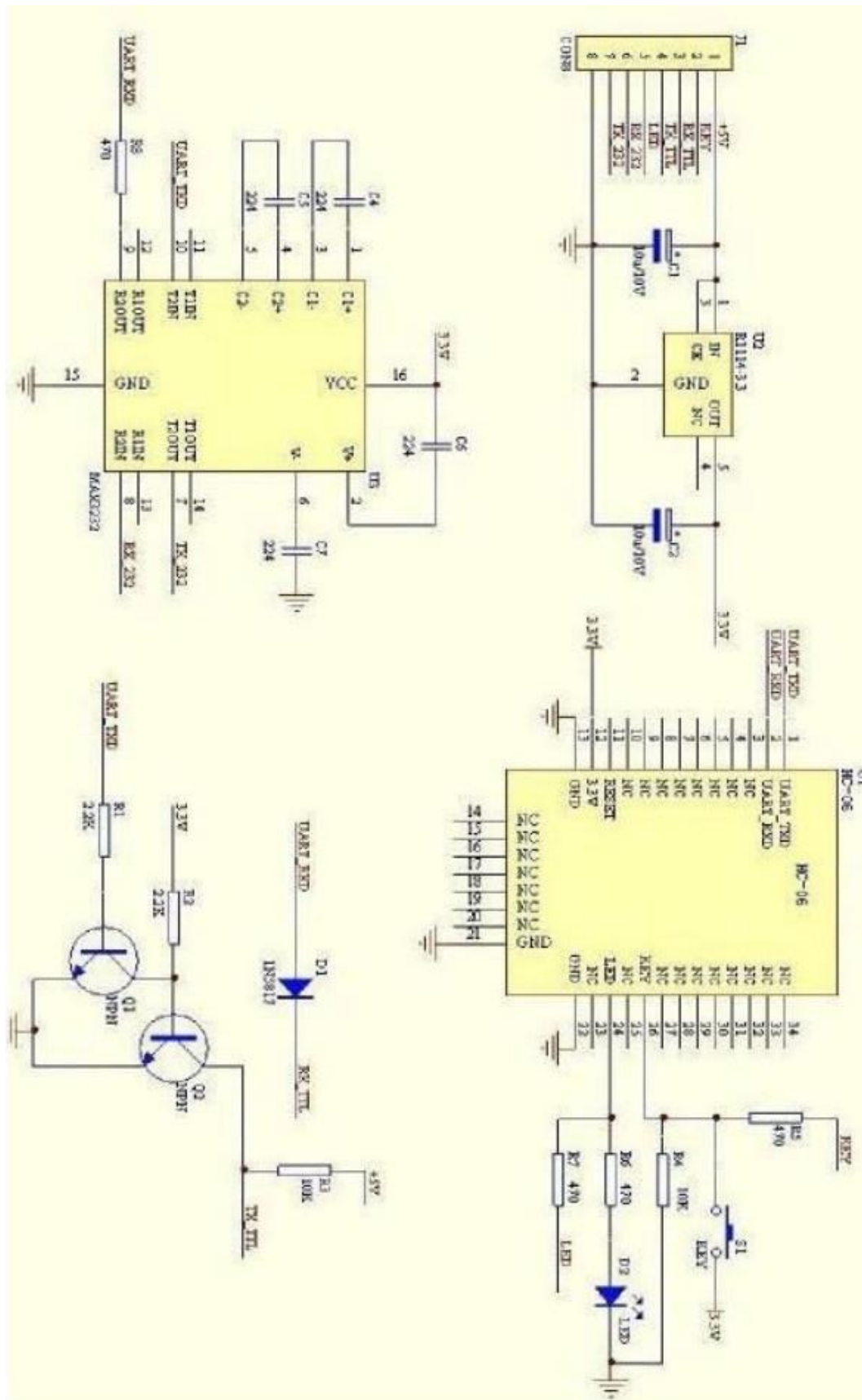
IDC - Press Release, 2013. IDC Worldwide Mobile Phone Tracker, joulukuu 3, 2012. Luettu 2.4.2013.  
<http://www.idc.com/getdoc.jsp?containerId=prUS23818212#.UVq4yTCnCGa>

MOXDC: Bluetoothin ohjelmointi, 2011. Nimimerkin MOXDC:n kirjoittama blogi bluetoothin ohjelmoinnista. Luettu 25.3.2013.  
<http://lefars.org.uk/WPblog/2011/11/30/how-to-program-a-serial-bluetooth-module/>

Wikipedia: Bluetooth, 2013. Wikipedia artikkeli Bluetoothista. Luettu 25.3.2013.  
<http://fi.wikipedia.org/wiki/Bluetooth>

**LIITTEET**

Liite 1. Bluetooth-kytkentä (Bluetooth-vastaanotin 2013)



Liite 2. Arduino-kytkentä (Arduino Nanon kytkennät 2008)

