



# **AUTOMAATTINEN TUNNISTUS BLUETOOTH-YHTEYDELLÄ**

Jussi Paukku

Opinnäytetyö  
Joulukuu 2013  
Tietotekniikka  
Elektroniikka ja sulautetut  
järjestelmät

TAMPEREEN AMMATTIKORKEAKOULU  
Tampere University of Applied Sciences

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikka  
Elektroniikka ja sulautetut järjestelmät

Jussi Paukku:

Automaattinen tunnistus Bluetooth-yhteydellä  
Opinnäytetyö 38 sivua, joista liitteitä 13 sivua  
Joulukuu 2013

---

Bluetooth-yhteys tarjoaa helpon keinon työaseman käyttäjän automaattiseen tunnistamiseen. Tässä opinnäytetyössä esitellään ratkaisu, joka eroaa yleisesti saatavilla olevista järjestelmistä suunnittelulähtökohdista lähtien. Suurin ero olemassa oleviin järjestelmiin on se, että tunnuksena toimivana laitteena (ID-laite myöhemmin) voidaan käyttää lähes mitä tahansa saatavilla olevaa Bluetooth-yhteydellä varustettua laitetta, rajoituksella että Bluetooth on oltava kytkettävissä jatkuvasti näkyvissä olevaksi. Laite voidaan pienin muutoksin laittaa toimimaan millä tahansa käyttöjärjestelmällä. Merkittävänä poikkeuksena on myös se, ettei työasemakoneella tarvitse olla tunnistuksen kannalta välttämätöntä ohjelmaa jatkuvasti käynnissä. Työasemakoneen käynnistyksen yhteydessä ajetaan pieni ohjelma joka kertoo tunnistuslaitteelle käytetyn salasanan, joka tallennetaan käyttömuistiin. Käyttömuistissa olevat tiedot pyyhkiytyvät, jos tunnistuslaitteen virta katkaistaan, mistä on se etu, että jos tunnistuslaite varastetaan, ei sen muistista voida kaivaa työaseman salasanaa. Laite kytketään työasemakoneen vapaaseen USB-liitäntään, josta se saa virtansa, ja toimii näppäimistönä. Näppäimistökomeinoilla voidaan työasemalta kirjautua ulos ja takaisin sisään.

Kaikessa yksinkertaisuudessaan tunnistuslaite etsii lähistöltä tiettyä Bluetooth-laitetta, jonka se tunnistaa MAC-osoitteen perusteella. Laitteella on kaksi tilaa kirjauduttu ja ei-kirjauduttu. Kirjauduttu-tilassa pysytään niin kauan, kunnes ID-laite ei enää löydy haussa, jolloin kirjaudutaan ulos. Ei-kirjauduttu tilassa pysytään vastaavasti niin kauan, kun ID-laitetta ei löydy skannauksessa. Laitteen löytyessä kirjaudutaan sisään. Laite siis reagoi hakutuloksien muutoksiin ID-laitteen osalta.

Järjestelmää voidaan helposti soveltaa muihinkin käyttökohteisiin kuten: auton ajon-eston ohjaus, keskuslukituksen ohjaus, kodin murtohälytysjärjestelmän ohjaus. Kaikissa käyttökohteissa järjestelmä voidaan laittaa nykyisen tunnistusjärjestelmän rinnalle, tai sarjaan. Rinnalla ollessaan se voi toimia lisävarmistuksena kytkemässä lukitukset ja hälytykset päälle, mikäli käyttäjä unohtaa ne kytkeä. Tai erillisenä tunnistautumisvaihtoehtona, jolloin sitä käyttämällä voidaan ohittaa normaali tunnistautuminen. Sarjaan kytkettynä laite toimii toisena varmennuslenkkinä, esimerkiksi estämällä normaalin tunnistautumisen kokonaan, toisen tunnistautumisen puuttuessa.

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
ICT Engineering  
Embedded Systems and Electronics

Jussi Pauku:  
Automatic identification by Bluetooth connection

Bachelor's thesis 38 pages, appendices 13 pages  
December 2013

---

Bluetooth offers simple way to implement automatic identification on workstations. In this thesis is elaborated how it can be done with several restrictions, which separate this solution from already widely available solutions. First and most important, almost any Bluetooth capable device can be used as identification device (ID-device later), granted that it can be made visible to all. It also follows that there is no required app installation on the device that is used as identifier. Second this solution can be made to work on any operating system. Third there is no application running all the time on the workstation in question. Simple software is run at operating system startup to transmit password to the identifying device, which then stores it in data memory. When identifying device loses power the password is wiped from its memory, so possible theft isn't going to make your password available to the culprit. Identifying device is plugged in free USB-port on the workstation and functions as keyboard to control automatic logging in and out.

In all simplicity the device is set up to scan specific Bluetooth device, which it identifies by MAC-address. In operation, if current state is <logged in> and ID-device isn't found anymore workstation is commanded to log out; if state is <logged out> and ID-device is found again, workstation is commanded to log in. So identifying device reacts to changes to presence of the ID-device.

This solution can be adapted to various other applications such as car immobilizer, power door locks, and anti-burglar alarm system. In all these applications and the workstation application, system can be either parallel or sequential with existing measures. When operating in parallel, there is two possible ways of operation, as added safeguard against human error, or as way to bypass conventional operation. As added safeguard device works only as locking device, for example in case the user forgets to lock his car or log out from workstation. In sequential operation system works as required second ID, for example by deactivating conventional operation if second ID isn't present.

---

Key words: bluetooth, automatic, identification, security

## SISÄLLYS

1	JOHDANTO.....	6
2	PERUSTIETOA .....	7
2.1	Tunnistimen tarkoitus .....	7
2.1.1	Laitteelle asetettuja vaatimuksia .....	7
2.2	Suunniteltu laitteisto .....	8
2.2.1	Bluetooth-moduuli .....	9
2.2.2	Mikrokontrolleri.....	10
2.2.3	ID-laite .....	10
3	MIKROKONTROLLERI.....	11
3.1	AVR.....	11
3.2	Arduino .....	12
3.2.1	Arduinon edut kehitysalustana.....	12
4	TOIMINTA .....	14
4.1	Asetukset.....	14
4.1.1	Bluetooth-yhteystiedot .....	14
4.2	Alustus rutiini käynnistettäessä .....	16
4.3	Käyttäjän tunnistus .....	17
5	TIETOTURVA.....	18
5.1	Riskikartoitus .....	18
5.1.1	Riskit.....	19
5.2	Vielä ratkaisemattomat ongelmat .....	20
5.3	Kehitysversiot .....	20
5.3.1	Ensimmäinen prototyyppi.....	21
5.3.2	Toinen prototyyppi.....	21
6	TULOKSET .....	22
7	JATKOKEHITYS .....	23
7.1	Mahdolliset käyttökohteet.....	24
	LÄHTEET.....	25
	LIITTEET .....	26
	Liite 1. Teensy sovelluksen lähdekoodi .....	26
	Liite 2. Windows sovelluksen lähdekoodi .....	35

**LYHENTEET JA TERMIT**

Bluetooth	Langaton yhteys joka hyödyntää 2,4 GHz radiotaajuutta
Mikrokontrolleri	Yksinkertainen mikroprosessori
RSSI	Signaalivoimakkuus
MAC	Laitteen yksilöivä osoite
Arduino	Avoimen lähdekoodin kehitysalusta
USB	Universal Serial Bus, yleinen väylä lisälaitteiden kytkemisessä
HID	Human Interface Device, laite jolla ihminen voi hallita tietokonetta

## 1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on esitellä Bluetooth-yhteyttä hyödyntävän automaattisen tunnistusjärjestelmän kehitys ideasta toimivaksi laitteeksi. Suunnittelun lähtökohdat tekevät toteutetusta laitteesta uniikin tällä hetkellä saatavilla oleviin ratkaisuihin nähden.

Opinnäytetyön tavoitteena on tuoda ilmi tietoturvallisuuden kannalta huomioon otettavia asioita, ja esitellä kuinka tietoturvasäikeet on toteutuksessa otettu huomioon. Antaa näyte mikrokontrollerisovellusten mahdollisuuksista.

Tunnistusjärjestelmä koostuu kolmesta osasta: ID-laitteesta, tunnistimesta ja työasemasta. Tunnistimen käyttöönottoon tarvittavat asetukset ovat minimaaliset.

## 2 PERUSTIETOA

Tilaajana on Tampereen TKI-osuuskunta Verstas.

### 2.1 Tunnistimen tarkoitus

Suunnitellun laitteen tehtävänä on tunnistaa automaattisesti käyttäjä ilman että hänen tarvitsee tehdä itse mitään. Käyttäjän lähtiessä työpisteeltä, tietokone lukitaan automaattisesti. Toimii pois lähtiessä lisä varmennuksena, mikäli käyttäjä unohtaa itse lukita työasemansa.

#### 2.1.1 Laitteelle asetettuja vaatimuksia

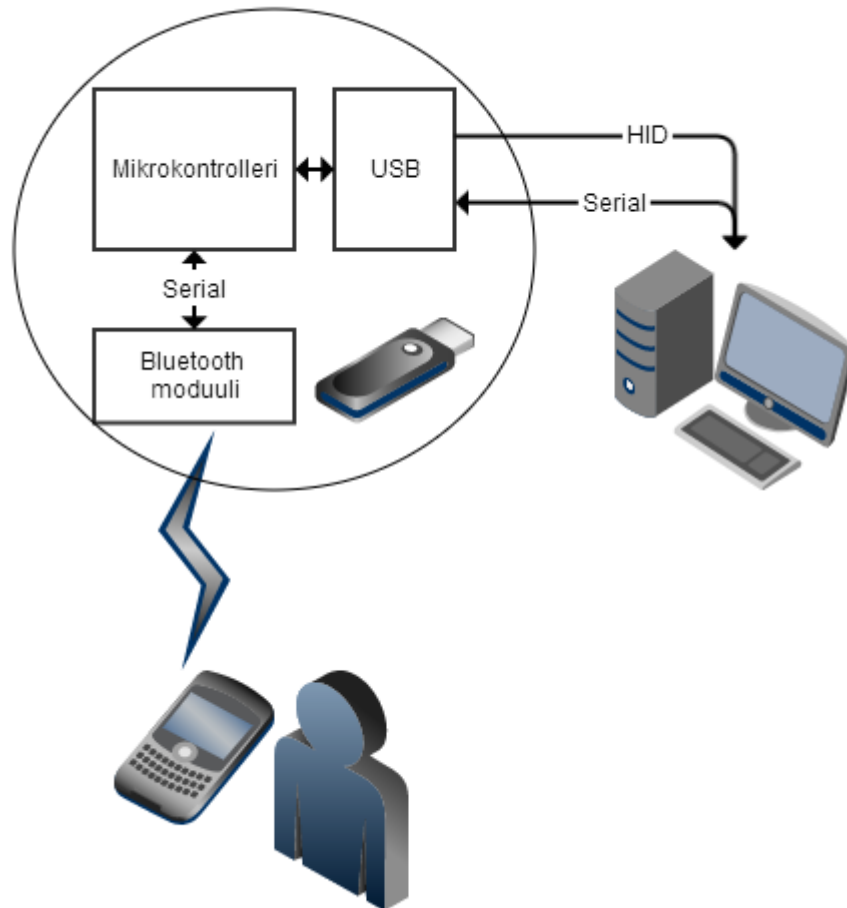
Laitteen tilaajan vaatimukset laitteelle olivat seuraavat:

- Laite on toteutettava siten, ettei tietokoneella ole jatkuvasti käynnissä toiminnan edellyttämää ohjelmaa
- ID-laitteelle ei tarvitse asentaa mitään erillistä ohjelmistoa
- Laitteesta ei saa koitua ylimääräisiä tietoturvariskejä

Koska työasemakoneella ei saa olla ohjelmia jatkuvasti päällä valittiin kehitysalusta jolla voidaan simuloida näppäimien painalluksia. Näppäimistökomennolla pystytään ohjaamaan tietokone lukitsemaan itsensä, sekä kirjautumaan takaisin sisään. Koska ID-laitteelle ei voida asettaa ohjelmisto vaatimuksia, sen asetuksissa on oltava kytkettynä tila, jossa se näkyy muille Bluetooth-laitteille. Suunnittelussa on otettu huomioon mahdollisimman paljon erilaisia sudenkuoppia. Tietoturvakysymys on käsitelty tarkemmin kappaleessa 5

## 2.2 Suunniteltu laitteisto

Tunnistamisen hoitava laite ottaa tarvitsemansa virran tietokoneen USB-väylästä. Kuvassa 1 on esiteltyä lohkokaavio laitteen toiminnan kannalta keskeisistä osista.



KUVA 1. Lohkokaavio laitteistosta.

Ympyrän sisällä olevat lohkot ovat kaikki kuvan USB-tikun sisässä.

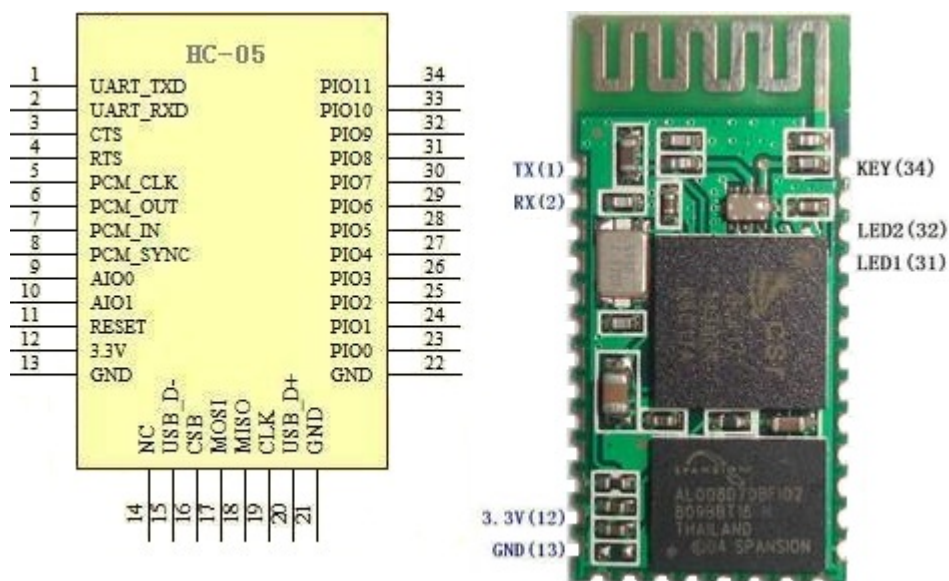


## 2.2.1 Bluetooth-moduuli

Hc-05 on yksinkertainen moduuli sarjaviestintään Bluetooth-yhteyttä hyödyntäen. Radio-sarjaviestintä toimintoa ei käytetä. Moduulia ohjataan AT-komennoilla. Mikrokontrolleri lähettää AT-komennot moduulille sarjaportin kautta, kuvassa 2 sarjaportti on pinneissä 1 ja 2.

Taulukko 1 Esimerkki AT-komennosta.

Komento	Vastaus
AT+INQ	+INQ: <addr>,<class>,<rssi>

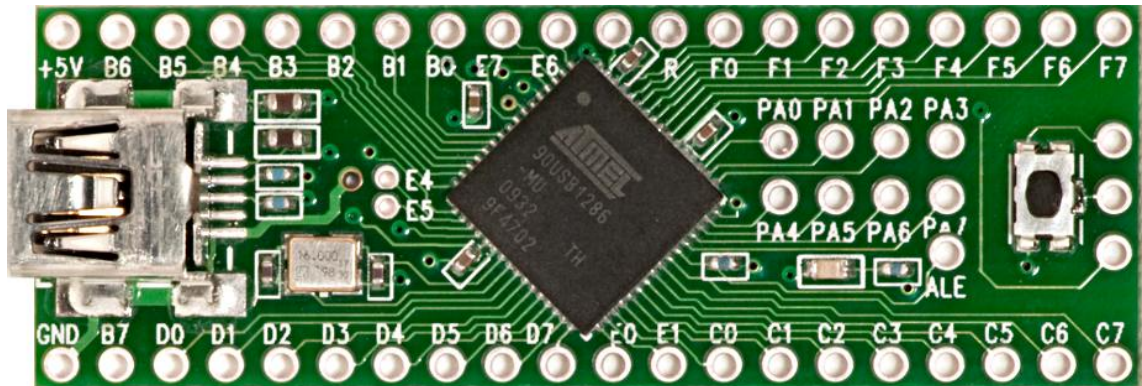


KUVA 2. HC-05 bluetoothmoduuli (<http://mbed.org/users/edodm85/notebook/HC-05-bluetooth/>)

UART\_TXD ja UART\_RXD ovat käytetyt sarjaliitännät joilla mikrokontrolleri kommunikoi Bluetooth-moduulin kanssa. Pinni 34 on kytkettynä käyttöjännitteeseen, jotta moduuli on komentotilassa.

### 2.2.2 Mikrokontrolleri

Mikrokontrolleriksi valittiin Teensy 2.0++ kehitysalusta, jossa on AT90USB1286, AVR arkkitehtuurin 8 bittinen mikroprosessori 16 MHz kellotaajuudella.



KUVA 3. Teensy 2.0++ mikrokontrolleri (<http://www.pjrc.com/store/teensypp.html>)

Tämä mikrokontrolleri valittiin koska siinä on sisäänrakennettuna mahdollisuus simuloida näppäimistökomentoja, ja sitä kautta ohjata tietokone lukitsemaan itsensä, ja kirjautumaan sisään. Käyttökohteisiin joissa näppäinkomennot tietokoneelle eivät ole tarpeellisia, voidaan käyttää lähes mitä tahansa kehitysalustaa.

Teensy on Arduino-yhteensopiva, joka tarkoittaa sitä että sen ohjelmointiin voidaan käyttää Arduino-ohjelmointiympäristöä.

### 2.2.3 ID-laite

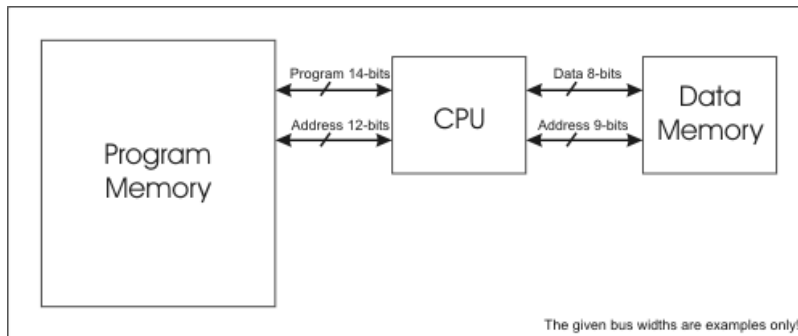
Voidaan käyttää mitä tahansa Bluetooth-laitetta. Mitä paremmin laite pysyy päällä ja mukana, sitä luotettavammin järjestelmä toimii. Ainoa vaatimus tunnistukseen käytettävälle laitteelle on se että se löytyy skannaamalla.

Tarkoitukseen soveltuu lähes kaikki uudemmat matkapuhelimet, erikseen tarkoitusta varten rakennettu avaimenperä, Bluetooth-handsfree, jne. Ainoa vaatimus mitä järjestelmä asettaa, on se että laitteen on työpisteellä olon aikana oltava kokoajan löydettävissä skannaamalla.

### 3 MIKROKONTROLLERI

#### 3.1 AVR

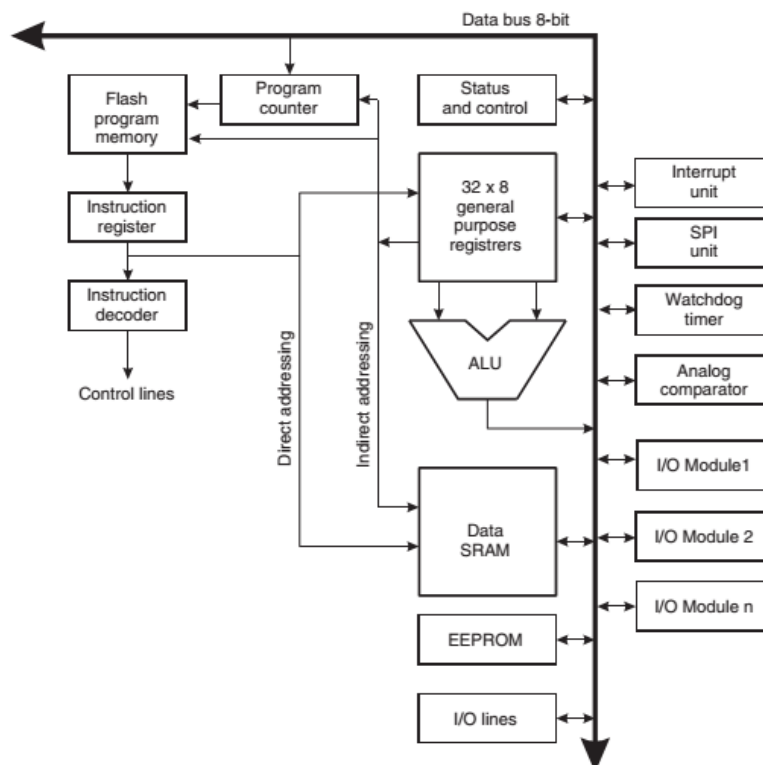
AVR on modifioitu Harvard arkkitehtuuri. Harvard arkkitehtuurissa on suoritin jossa on erillisillä väylillä yhteys ohjelmamuistiin ja käyttömuistiin. Tällä ratkaisulla voidaan rekisteriin hakea ohjelmamuistista seuraava komento sillä aikaa, kun edellistä käskyä vielä suoritetaan.



KUVA 4. Harvard

ri <http://www.elec.canterbury.ac.nz/PublicArea/Staff/hof/p10-embed/p10-tutorial/harvard.gif>

Ainoa ero puhtaaseen Harvard arkkitehtuuriin on mahdollisuus käyttää ohjelmamuistia vakionmuuttujien tallentamiseen, joita lukea ajon aikana kuin se olisi käyttömuistissa.



KUVA 5. AVR arkkitehtuuri (AT90USB1286 datalehti s.11)

## 3.2 Arduino

Arduino on avoimen lähdekoodin kehitysympäristö, johon on saatavilla valmiita kirjastoja kaikenlaisia lisäosia, antureita, tiedonsiirtoa, ym. varten. Ideasta prototyypiksi kuluva aika on Arduino alustalla hyvin lyhyt, laajasta kirjastotarjonnasta ja helposti omaksuttavasta syntaksista johtuen. Arduino yhteensopivien mikrokontrollerialustojen hinnat alkavat noin kahdeksasta eurosta. Johtuen alhaisista alkuhankinta kustannuksista idean ei tarvitse olla kummoinen, että siitä kannattaa ainakin tehdä prototyyppi, jonka perusteella voi päätellä, onko idean laitteelle oikeasti käyttöä, vai ei, missä tapauksessa alustan voi kierrättää seuraavaan projektiin.

Arduino yhteensopivuus kattaa suuren määrän eri ominaisuuksin varusteltuja mikrokontrolleri alustoja, mikäli alustan erikoisominaisuuksia ei käytetä, voidaan sama ohjelma ajaa kaikille arduino-yhteensopiville mikrokontrolleri alustoille. Tämä helpottaa laitteiden jatkekehitystä, kun voidaan hyödyntää jo toimivaa ohjelmaa, ja lisätä siihen ominaisuuksia joita toinen mikrokontrollerialusta tarjoaa.

### 3.2.1 Arduinon edut kehitysalustana

Seuraavassa luetelmassa on esiteltyä Arduino-kehitysympäristön etuja muihin kehitysympäristöihin nähden.

1. Käyttönoton helppous

Käyttönottoa helpottavat saatavilla olevat valmiit kirjastot erilaisten lisäkomponenttien hallintaan ja käyttöön.

2. Ideasta prototyypiksi

Toimivan prototyypin valmistamiseen ei tarvitse tehdä erillisiä piirilevyjä, eikä välttämättä edes tinata. Kaikki voidaan hoitaa protolevyllä ja hyppylangoilla. Kehitysvaiheessa voidaan käyttää mitä tahansa käsillä olevaa arduino kehitysalustaa, ja lisätä erikoisominaisuudet myöhemmin, kun vaatimukset ovat selvillä.

3. Skaalautuvuus

Arduino yhteensopivia kehitysalustoja on fyysisesti erikokoisilla piirilevyillä. Mikrokontrollereita on vaihtelevalla määrällä kytkentänastoja, ohjelmamuistia, erikois-ominaisuuksia ja suorituskykyä.

#### 4. Shieldit

Koska Arduino kehitysalustat ovat standardoidulla piirilevyllä, on niille saatavilla lisäkortteja jotka avaavat lisämahdollisuuksia kuten: Lähiverkko yhteys, langaton lähiverkko, gsm, GPS, erilaiset näytöt, moottorinohjaus, relekytkennät.

## 4 TOIMINTA

Tietokoneeseen kytkettynä laite näkyy käyttöjärjestelmälle HID laitteena, käytännössä näppäimistönä. Laite ohjaa tietokonetta näppäimistökomentojen avulla.

### 4.1 Asetukset

Laite tarvitsee toimiakseen tunnistettavan laitteen MAC-osoitteen, nimen, sekä työaseman salasanan. Osoite ja nimi tallennetaan mikrokontrollerin pysyvään muistiin, jolloin ne säilyvät virran katkaisun jälkeenkin, eikä niitä täten tarvitse kertoa laitteelle. Asetusohjelmalle syötetään tunnistukseen käytettävän laitteen nimi, jolloin laite itse selvittää tämän nimisen laitteen osoitteen ja tallentaa molemmat pysyvään muistiin. Tunnistettavaa laitetta ei tarvitse parittaa kirjautumislaitteen kanssa. Tunnistettavaa laitetta asetettaessa kirjautumislaitte kuitenkin tekee tunnistettavalle laitteelle parituspyynnön, varmistuksena käyttäjälle, että oikea laite on asetettu.

Salasana tallennetaan laitteen käyttömuistiin, joten tämä tieto on annettava aina, laite kytketään päälle. Tämä on käytännössä toteutettu niin, että työaseman käynnistyksen yhteydessä ajetaan rutiini, joka ottaa laitteeseen yhteyden, ja kertoo tälle salasanan. Yhteys hoidetaan USB väylän kautta sarjamuotoisena.

#### 4.1.1 Bluetooth-yhteystiedot

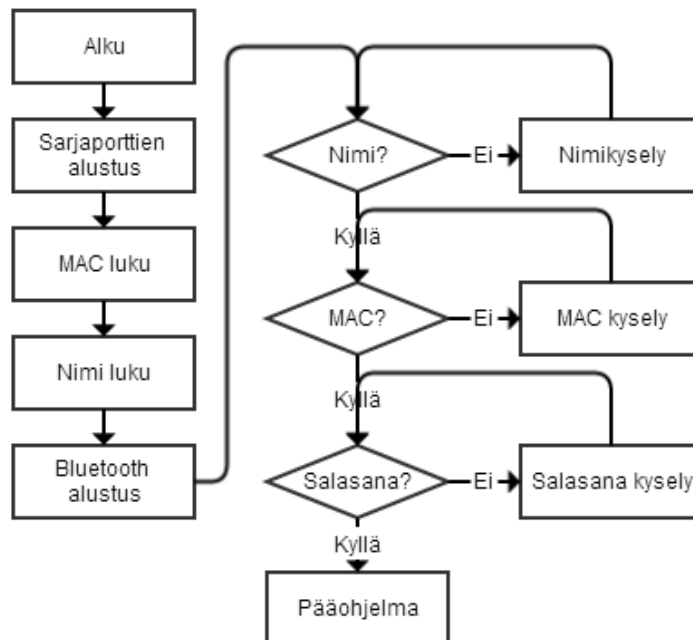
MAC eli Media Access Control on tiedonsiirtoyhteys protokolla joka on TCP/IP viitemallin peruskerroksen alikerros. Bluetooth-viestinnässä sitä käytetään erottamaan laitteet toisistaan, jotta voidaan lähettää viestejä halutulle laitteelle, joka sitten tunnistaa viestin olevan sille. MAC-osoite on kuusi tavua pitkä, yleisesti muotoa 12:34:56:78:9A:BC, ja Bluetooth-viestinnässä NAP:UAP:LAP muodossa 1234:56:789ABC. NAP ja UAP ovat valmistajan osa osoitteesta jotka löytyvät IEEE standardista, LAP on laitteen identifioiva osa joka suurella todennäköisyydellä on käyttäjän ympäristössä uniikki. Kaikkiin laitteisiin on tehtaalla asetettu joku MAC-osoite, mutta useimmissa laitteissa se voidaan vaihtelevalla vaikeusasteella muuttaa. Esimerkiksi tietokoneen Bluetooth-moduulin MAC-osoitteen pystyy muuttamaan ajurin

asetuksista. Matkapuhelinten MAC-osoitteiden muuttaminen on vaikeampaa, ja usein ilman takuun menetystä mahdotonta.

Bluetooth-laitteiden tunnisteenä on myös laiteluokka, joka koostuu kolmesta tavusta. Ensimmäinen tavu määrittelee suuren palvelu luokan. Toisen tavun viisi vähiten merkitsevää bittiä määrittelevät suuren laiteluokan. Viimeinen tavu määrittelee laitteen alaluokan, joka vaihtelee suuren laiteluokan perusteella. Esimerkiksi matkapuhelinten laiteluokka on 0x400204 ja älypuhelinten 0x40020C.

## 4.2 Alustus rutiini käynnistettäessä

Tunnistuslaite suorittaa alustusohjelman joka käynnistyksellä, kuvassa 6 on esiteltyinä toimenpiteet jotka suoritetaan, ennen kuin laite on toimintavalmis.



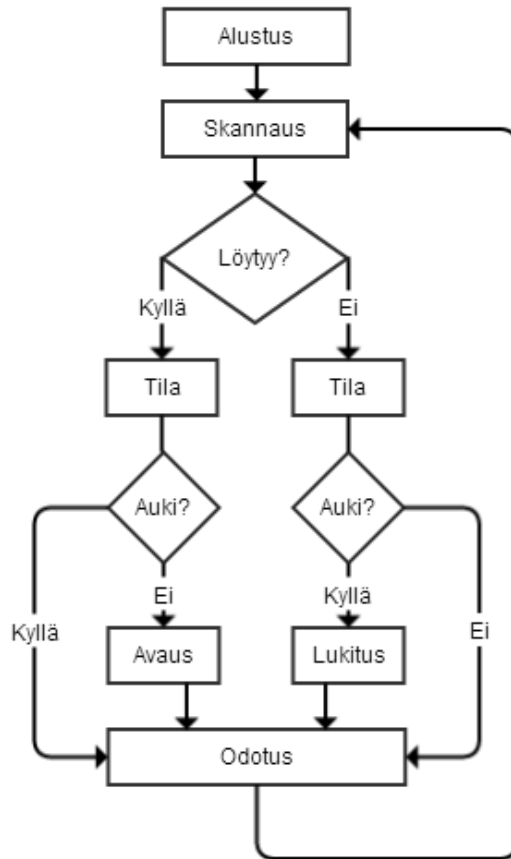
KUVA 6. Alustus prosessin lohkokaavio

Sarjaporttien alustuksessa käynnistetään sarjaviestintä komponentit, jotta voidaan kommunikoida tietokoneen ja Bluetooth-moduulin kanssa. MAC luku funktiossa luetaan pysyvältä EEPROM-muistilta asetettu ID-laitteen MAC-osoite. Nimi luku funktiossa luetaan ID-laitteen nimi EEPROM-muistilta. Bluetooth alustus funktiossa varmistetaan että Bluetooth-moduuli on käyttökunnossa. Sitten tarkistetaan että käyttömuistissa on tarvittavat tiedot. Laite odottaa kunnes saa järjestelmältä salasanan.



### 4.3 Käyttäjän tunnistus

Pääohjelasilmukka pyörii jatkuvasti, mutta skannaus suoritetaan silti määrätyn väliajoin. Kuvassa 7 on esiteltyä pääohjelma tunnistus toiminnan kannalta.



KUVA 7. Pääohjelma silmukan lohkokaavio

Uusimmassa versiossa odotus on avauksen jälkeen 2 sekuntia ja lukituksen jälkeen 10 sekuntia. Skannaus on määritelty AT-komennolla `AT+INQM=1,100,5` missä ensimmäinen osa on RSSI lippu, toinen argumentti on enimmäismäärä tuloksille ja kolmas on skannauksen kesto. RSSI kytkentä muuttaa skannaustoimintaa siten että sama laite näkyy skannauksessa useammin kuin vain kerran, mutta sen signaalivoimakkuus mitataan, tämä on vielä kokeellisella asteella toiminnassa. Koska kaikki laitteet voivat näkyä useammin kuin kerran on maksimimäärää kasvatettu sataan. Skannauksen kesto lasketaan kaavalla  $\text{argumentti} \times \text{kertaa} \times 1,28$  sekuntia, tässä tapauksessa  $6,4$  sekuntia.

## 5 TIETOTURVA

Tietoturvan maksimointi on keskeisenä suunnittelulähtökohtana, ja erilaisia vaihtoehtoja ja mahdollisia skenaarioita on tullut mietittyä useita. Suureen osaan mahdollisia tietoturvaongelmia ratkaisut ovat olleet varsin intuitiivisia.

### 5.1 Riskikartoitus

Riskien määrittelemisessä sovelletaan taulukkoa 2. Todellinen riski määräytyy tapahtuman todennäköisyyden ja seurausten vakavuuden perusteella neljään kategoriaan. Ensimmäinen vakavuus-aste koostuu harvinaisista ja harmittomista tapauksista. Toinen vakavuus-aste pitää sisällään epämiellyttävän usein tapahtuvat, mutta harmittomat tapaukset, sekä harvinaiset, mutta vakavammat tapaukset. Kolmanteen kategoriaan sijoittuu käytännössä varmat harmittomat ongelmat ja hyvin harvinaiset katastrofit. Neljännessä kategoriassa on sietämättömät riskit.

Taulukko 2. Riskianalyysi

Riskianalyysi							
tapahtuman todennäköisyys	Seuraus					Selite	
	1	2	3	4	5	Seuraus	Riski
> 90 %						1 mitätön	1 pieni
50...90 %						2 pieni	2 kohtalainen
10...50 %						3 kohtalainen	3 suuri
3...10 %						4 merkittävä	4 äärimmäinen
< 3 %						5 katastrofaalinen	

### 5.1.1 Riskit

Tässä kappaleessa esitellään laitteen käytöstä johtuvia ja järjestelmään liittyvät riskit ja niihin varautuminen ja hallinta. Riski listaukseen ei ole eritelty analyysi taulukosta arvoja, mutta seuraukset on pyritty minimoimaan tapauksesta riippumatta.

Riskejä:

Varkaus

- Laite varastetaan
- ID-laite varastetaan
- Työasemakone varastetaan

Salakuuntelu

Murtotekniikat

- Brute-force hyökkäys

Laitteen käytöstä ei varkaustapauksessa seuraa ylimääräisiä tietoturvariskejä. Käyttö lisää turvallisuutta, koska se pienentää aikaikkunaa, jolla työaseman voi anastaa käyttövalmiina.

Laitetta salakuuntelemalla ei ole mahdollista saada mitään yksityistä tietoa. Koska tunnistuslaitteen ja ID-laitteen välillä ei ole tiedonsiirtoyhteyttä, ainoa tieto mitä salakuuntelemalla saa selville, on tilasta löytyvät Bluetooth-laitteet, jotka on asetettu näkyviin kaikille.

Bluetooth MAC-spoof on ainoa bruteforce hyökkäys mitä laitetta vastaan voidaan hyödyntää. MAC-spooffaus tarkoittaa sitä, että toisen laitteen MAC-osoite vaihdetaan samaksi kuin käytössä olevan ID-laitteen MAC-osoite. Bruteforce menetelmässä ei yritetä arvata oikeaa MAC-osoitetta, vaan kokeillaan järjestyksessä kaikki läpi.

Esimerkkilaskelmassa käytetään skannausasetuksena AT+INQM=0,20,2 Laite skannaa 2,56 sekuntia tai 20 laitetta, 10sek välein työpisteen ollessa lukittuna. Hyökkäys nopeus on kuristettu siis osoitetta per 12,56 sekuntia, ja koska on mahdotonta tietää, koska laite oikeasti skannaa, hyökkäyksen teho laskee huomattavasti. Optimaalinen hyökkäysnopeus 20 osoitetta 2,56 sekunnissa kymmenen sekunnin välein eli 96

osoitetta minuutissa. Bluetooth laitteiden MAC-osoite on 6 tavua pitkä 0:0:0:0:0:0...255:255:255:255:255:255, mahdollisia osoitteita on  $2^{48}$  eli  $248,5 \cdot 10^{12}$ .

$$\frac{2^{48}}{96 \frac{\text{osoitetta}}{\text{min}} \cdot 60 \frac{\text{min}}{\text{h}} \cdot 24 \frac{\text{h}}{\text{d}} \cdot 365 \frac{\text{d}}{\text{a}} \cdot 2} = 2,79 \cdot 10^6 \text{a}$$

2,79 miljoonaa vuotta osoiteavaruuden puolessavälissä olevalle osoitteelle, täydellisesti ajoitettuna, ilman häiriöitä tai keskeytyksiä. Lisävaatimuksena on näköyhteys tietokoneeseen, että tiedetään milloin hyökkäys on viimein onnistunut. Koska ei voida tietää milloin laite suorittaa skannausta, yrityksiä mene helposti hukkaan, ja oikea osoite saattaa tulla kokeiluksi silloin, kun etsintä ei ole käynnissä.

## 5.2 Vielä ratkaisemattomat ongelmat

Koska ID-laitteen on oltava löydettävissä, kuka tahansa muukin voi halutessaan saada selville sen MAC-osoitteen, toki tämä edellyttää samaan tilaan pääsemistä, ja tietoa että käytössä on Bluetoothiin perustuva järjestelmä. Koska tunnistuslaite ei varsinaisesti viesti mitään, on järjestelmän olemassaolosta vaikea päästä selville.

## 5.3 Kehitysversiot

Seuraavaksi käsitellään hieman eri kehitysversioiden välillä tapahtuneita muutoksia. Muutosten takana on ollut tietoturvan polttopisteen muuttaminen ja tarkennus. Ensimmäisen kehitysversion ideana oli toimia siten, ettei ID-laitteen tarvitse olla näkyvissä skannaukselle, koska siinä on mahdollinen riski käytettäessä älypuhelinta ID-laitteena. Riski on että käyttäjä vahingossa hyväksyy jonkin tuntemattoman lähteen pyynnön lähettää dataa laitteeseen. Toisessa kehitysversiossa ID-laite on asetettu toissijaiseksi, koska ensimmäisen version menetelmä heikensi työasemakoneen turvallisuutta merkittävästi enemmän, kuin toisen version toiminta heikentää ID-laitteen turvallisuutta.

### **5.3.1 Ensimmäinen prototyyppi**

Ensimmäisessä kehitysversiossa oli huomattavasti parantamisen varaa. Skannauksen sijaan laite kysyi Bluetoothin yli onko tällä osoitteella laite lähistöllä. Johon laite vastaa omalla nimellään. Menettely on erittäin huono, sillä yksinkertaisella Bluetooth-salakuuntelulla saadaan selville kaikki mitä tarvitaan työasemalle pääsemiseen. Ensimmäisessä versiossa salasana oli myös tallennettuna mikrokontrollerin pysyvään muistiin.

### **5.3.2 Toinen prototyyppi**

Toisessa kehitysversiossa ensimmäisen version ongelmat on ratkaistu. Kysely suoritetaan skannaamalla, jolloin laite ei lähetä mitään, ainoastaan kuuntelee, mitä laitteita lähistöllä on. Laitteet jotka ovat asetettu näkymään, kertovat jollain tiheydellä olevansa paikalla. Avaimena toimiva laite voi siis olla mikä tahansa löytyvistä laitteista. Toisessa versiossa salasana saadaan tietokoneen käynnistyksen yhteydessä ajettavalla pienellä ohjelmalla, eikä laite toimi, ennen kuin sillä on salasana. Salasana on tietokoneella selväkielisenä tekstitiedostossa, tämän ei pitäisi olla ongelma, koska yrityskoneissa usein on verkkolevyjen salasanoja samalla tavalla säilöttynä, ja kiintolevyt ovat poikkeuksetta salakirjoitettuja. Tällä menettelyllä päästään myös eroon siitä mahdollisuudesta että salasana voisi joutua väärin tahojen haltuun, mikäli laite varastettaisiin. Salasana on laitteen muistissa ainoastaan muuttujana, joten se pyyhkiytyy virrankatkaisun yhteydessä.

## 6 TULOKSET

Molemmat esiteltyt kehitysversiot saatiin toimimaan halutulla tavalla. RSSI-ominaisuuden käyttöä tuli testailtua toista prototyyppiä rakennettaessa, mutta toiminnon käyttöönotto ei ollutkaan ihan yksinkertaista. Muutokseen skannauksen toiminnassa ei osattu varautua, ja toiminnon käyttöönotto olisi vaatinut suuria muutoksia skannaustulosten käsittelyssä, joten ominaisuus jätettiin pois toisesta prototyypistä.

Erilaisia skannausintervalleja testattiin, kirjautuneessa tilassa laite voisi skannata vaikka jatkuvasti, jolloin reaktioaika käyttäjän poistumiseen paranee. Ei-kirjautuneessa tilassa skannausintervallin pienentäminen pienentää reaktioaikaa myös, mutta helpottaa myös brute-force hyökkäystä. 10 sekuntia valittiin lopulta käytettäväksi väliajaksi, koska voidaan olettaa käyttäjältä kestävän 10 sekuntia kulkea kuuluvuusalueen reunalta työasemalle. Huonoimmassa tapauksessa käyttäjä saapuessa kuuluvuusalueelle heti skannauksen päätyttyä, seuraava skannaus tapahtuu hänen saapuessa työasemalle, jolloin hänen tarvitsee odottaa kolmisen sekuntia että työasema on käyttövalmiina. Suurimmalla todennäköisyydellä käyttäjän saadessa näköyhteyden työasemaan, se on juuri kirjautumassa sisään.

## 7 JATKOKEHITYS

Seuraavaan kehitysversioon olisi tarkoitus ottaa käyttöön RSSI tunnistus, ja siihen järkevä metodi jolla voidaan rajoittaa bruteforce-hyökkäyksen tehoa. Kaksi vaihtoehtoa toteutukseen on että etsittävä MAC-osoite tulee skannauksen aikana löytyä riittävän monta kertaa riittävällä signaalivoimakkuudella, jolloin ainoa haitta on tilanteessa jossa käyttäjä sattuu tulemaan työaseman läheisyyteen skannauksen ihan lopuksi, jolloin ID-laite ei rekisteröidy tarpeeksi monta kertaa työaseman avausta varten. Toinen vaihtoehto on tutkia ID-laitteen RSSI-arvon muutosta skannauksen aikana työaseman ollessa lukittuna. RSSI-arvon muutoksen tarkkailulla voidaan taata, että havaittu ID-laite on oikeasti jokin laite joka lähestyy työasemaa. Kun etäältä suunta-antennilla suoritettua bruteforce-hyökkäyksessä yritetyt MAC-osoitteet ilmestyvät suurella RSSI:llä tyhjästä, ja katoavat.

Toinen mahdollisesti käyttöön otettava menetelmä on laiteluokka rajaus. Asetettaessa ID-laitetta tämän laiteluokka määritellään ainoaksi haetuksi laiteluokaksi. Laiteluokka asetuksen AT-komento on AT+CLASS=, joka määriteltäisiin nolaksi kun ID-laitteen asetus ohjelma ajetaan. Skannaus AT-komento AT+INQ palauttaa kaikkien löytyneiden laitteiden laiteluokan, joten periaatteessa voidaan Bluetooth-moduulin luokka-asetus pitää nollassa, ja tallentaa muistiin asetetun ID-laitteen luokka, jonka on myös vastattava asetettua, jotta skannaus katsotaan onnistuneeksi. Tällä menetelmällä ohjelman rakennettakin saataisiin yksinkertaisemmaksi, kun saatua skannaustulosta ei tarvitse käsitellä niin paljoa.

## 7.1 Mahdolliset käyttökohteet

Auton käynnistykseneston ja varashälyttimen poiskytkentä, sekä keskuslukituksen ohjaus voidaan hoitaa Bluetooth-laitteella pienin muutoksin, lisäksi vaatimuslistalta putoaa pois näppäimistökomennot, joten tarkoitukseen voidaan käyttää muitakin mikrokontrollerialustoja.

Talon ulko-oveen voidaan asentaa sähkölukko, jonka ohjaus voidaan toteuttaa Bluetooth järjestelmällä, suurempana muutoksena on toteuttaa useampi hyväksyty tunnistus. Toteutus mahdollisuuksia on joko turvallisuuden parantaminen kaksinkertaisella varmennuksella, tai rinnakkaisena avaimen kanssa, siten että kumpi tahansa avaa lukon.



## LÄHTEET

Bluetooth määrittelyt. <http://www.bluetooth.org/en-us/specification/>  
AVR datalehti. <http://www.atmel.com/Images/doc7593.pdf>

**LIIKTEET**

1(9)

**Liite 1. Teensy sovelluksen lähdekoodi**

```

//#include <Bounce.h> //käytetty testauksessa
#include <EEPROM.h>

//globaaleiden muuttujien esittely

/* käytetty testauksessa, painonappien värähtelyn poisto
Bounce button0 = Bounce(0, 10);
Bounce button1 = Bounce(1, 10);
*/

String inputString = "";          // a string to hold incoming data
String btString = "";
boolean stringComplete = false;  // whether the string is complete
String salasana = "";

boolean kylla = false;
boolean logged = true;
unsigned long time = millis();
int limit = 10000;
int ledPin = 6;
boolean ledstate = LOW;

//                               12345678901234
//String btaddr = "8FC:88:6F8C4E";
//String btname = "sipa (Galaxy S4)";
String btaddr = "";
String btname = "";
//merkkijono taulukko struct
struct stringA {
    String value;
};

//alustusfunktio
void setup(){

    Serial.begin(38400);
    Serial1.begin(38400);
    delay(500);
    //Serial.println("yhteydet on");
    /*
    pinMode(0, INPUT_PULLUP);
    pinMode(1, INPUT_PULLUP);
*/
    inputString.reserve(200);

    btluku();
    nimiluku();

    btinit();

    while(salasana == ""){
        pwinit();
    }
    time = millis();

    while(btname == ""){

```

```

        nimiinit();
    }
    while(btaddr == ""){
        btQuery();
    }
}

//pääohjelma
void loop(){
    /*
    button0.update();
    button1.update();
    */
    if (stringComplete){

        if (inputString.indexOf("!!!!!!")){
            nimitys();
        }
        if (inputString.indexOf("aaaaaa")){
            bttallennus();
            nimitallennus();
        }
        inputString = "";
        stringComplete = false;
    }

    if (Serial.available()){
        sarjaluku();
    }

    if(millis()-time >= limit){
        digitalWrite(ledPin, HIGH);
        bttest();
        time = millis();
        digitalWrite(ledPin, LOW);
    }
    /*
    if (button0.fallingEdge()){
        Keyboard.set_modifier(MODIFIERKEY_RIGHT_GUI); //Tell Teensy to
press Windows key
        Keyboard.set_key1(KEY_L); //Tell Teensy to press L
        Keyboard.send_now(); //Press "Windows key + L"
        delay(500); //Wait for half second
        Keyboard.set_modifier(0); //Tell Teensy to release Windows key
        Keyboard.set_key1(0); //Tell Teensy to release L
        Keyboard.send_now(); //Release "Windows key + L"
        logged = false;
    }
    if (button1.fallingEdge()){
        Keyboard.println(salasana);
        logged = true;
    }
    */
}

//bluetooth moduulin toiminnan varmistus
void btinit(){

    String test= "";
    int laskuri =0;
    while(!test.indexOf("OK")){
        test = "";
        Serial1.println("AT");
        delay(200);
    }
}

```

```

    if (Serial1.available()){
        while (Serial1.available()){
            int inByte = Serial1.read();
            test += (char)inByte;
        }
    }
    laskuri++;
    if(laskuri>5){
        digitalWrite(ledPin,ledstate);
        ledstate = ~ledstate;
        laskuri = 0;
    }
}
while(Serial1.available()){
    char inChar = Serial1.read();
}
//Serial.println("yhkeys on");
Serial1.println("AT+INIT");
delay(100);

digitalWrite(ledPin,HIGH);
Serial1.println("AT+INQM=1,100,5");
delay(100);
while(Serial1.available()){
    char inChar = Serial1.read();
}
}

//salasanana saanti sarjaportista
void pwinit(){
    setpw();

    stringComplete = false;
}

void nimiinit(){
    nimitys();

    stringComplete = false;
}

//nimen asetus
void nimitys(){
    boolean nimikk = false;
    while(!nimikk){
        //Serial.println("nimee");
        while(!Serial.available()){
            sarjaluku();

            btname = inputString;
            Serial.println(btname);

            while(!Serial.available()){
                sarjaluku();
            }
            if (inputString.indexOf("#!#!")){
                nimikk = true;
                nimitallennus();
                btQuery();
            }
        }
        else{
            nimikk = false;
        }
    }
}

```

```

    }
    stringComplete = false;
}

//salasan aset
void setpw(){
    //Serial.println("salasana");
    while(!Serial.available()){
        digitalWrite(ledPin,ledstate);
        ledstate = ~ledstate;
        delay(200);
    }
    sarjaluku();
    salasana = inputString;
    digitalWrite(ledPin,LOW);
}

//bluetooth mac-osoitteen etsintä nimen perusteella
void btQuery(){
    boolean ok = false;
    int lines = 10;
    stringA input[lines];

    Serial1.println("AT+INQM=0,10,5");
    delay(100);
    while(!Serial1.available()){
        btsarjaluku();

        Serial1.println("AT+INQ");
        //Serial.println("query: AT+INQ");
        delay(100);
        while(!Serial1.available()){
            btsarjaluku();
            for(int i=0; i<lines;i++){
                int alku = btString.indexOf(":")+1;
                int loppu = btString.indexOf(",");
                input[i].value = btString.substring(alku,loppu);
                input[i].value.replace(":", ",");
                loppu = input[i].value.length();
                btString.substring(loppu);
                //Serial.print(i+1);
                //Serial.println(". value: " + input[i].value);
            }

            for(int i=0; i< lines; i++){
                Serial1.println("AT+RNAME?" + input[i].value);
                //Serial.println("AT+RNAME?" + input[i].value);
                while(!Serial1.available()){
                    // odotellaan vastaus
                    btsarjaluku();

                    if(btString.indexOf(btname)){ //vertailu
                        input[i].value.replace(",","");
                        btaddr = input[i].value;
                        btconfirm();
                        break;
                    }
                }
            }
        }
    }

    void btconfirm(){
        Serial1.println("AT+BIND=" + btaddr); //bindaus bt moduulille
        //Serial.println("AT+BIND=" + btaddr);
    }
}

```

```

        delay(100);
        Serial1.print("AT+PAIR=" + btaddr); //pairaus yritys eli 5(9)
        vahvistus käyttäjälle
        //Serial1.print("AT+PAIR=" + btaddr);
        Serial1.println(",10");
        //Serial1.println(",10");

        bttallennus();

        Serial1.println("AT+INQM=1,100,5");
        //Serial1.println("AT+INQM=1,100,5");
        delay(500);
        Serial1.flush();

    }
    /* //testaus huutelumenetelmällä
    void bttest(){

        String btString = "";
        int found = -1;
        Serial1.println("AT+RNAME?" + btaddr);
        Serial.print(">> ");
        Serial.println("AT+RNAME?" + btaddr);
        while(!Serial1.available()){
            if (Serial1.available()){
                while(Serial1.available()){
                    char btByte = Serial1.read();
                    btString += btByte;
                }
                Serial.println("<< " + btString);

                if(btString.indexOf("FAIL") > -1){
                    logout();
                }
                if(btString.indexOf(btname) > -1){
                    logout();
                    delay(200);
                    login();
                }
            }
        }
        btString = "";

    }
    */

    //testaus skannausmenetelmällä
    void bttest(){
        Serial1.println("AT+INQ");
        //Serial1.println("test: AT+INQ");

        //+INQ:25:48:21AD1A,5A020C,7FFF
        //+INQ:addr          ,class ,rssi

        btsarjaluku();
        String rivi = "";
        int found = btString.indexOf(btaddr);
        int rssi = 0;
        int hits = 0;
        int alku = found;

        while(alku){ //lasketaan montako kertaa oikea laite löytyi
            hits++;
            alku = (btString.indexOf(btaddr), alku+1);
        }
    }

```

6(9)

```

    alku = found;
    for (int i = 0; i < hits; i++){ // katsotaan kaikkien löytökertojen
rssi
        rssi = 0;

        //+INQ:25:48:21AD1A,5A020C,7FFF
        //+INQ:addr          ,class ,rssi

        alku = btString.indexOf(",", alku); // asetetaan alku 1. pil-
kun kohdalle
        alku = btString.indexOf(",", alku+1); // asetetaan alku 2.
pilkun kohdalle
        rivi = btString.substring(alku+1,alku+5); // otetaan toisen
pilkun jälkeen 4 merkkiä
        alku = (btString.indexOf(btaddr), alku+1); // asetetaan alku
seuraavan löytökerran alkuun

        for(int i = 0; i < 4; i++){ // RSSI to int
            rssi = rssi << 4; //shiftataan yksi byte vasemmalle.
            //merkityksellinen vasta ensimmäisen kierroksen jälkeen
            byte temp = rivi.charAt(i); // otetaan ensimmäinen merkki
tarkasteluun
            // temp = 48...57 or 64...70
            //      '0'...'9' 'A'...'F'
            // ->   48...57 or 58...63
            if (temp > '9') // jos (temp > 57) niin sen täytyy olla
joku hexakirjaimista
                temp -= '7'; // temp = temp- 55 <=> 'A' -> 10
            else
                temp -= '0'; // ei ollut kirjain, on siis numero

            // temp = 0...15 0x0...0xF
            // temp = 0B0000...0B1111

            rssi += (int)temp; /rssi = rssi + saatu numero
        }
        if (rssi <= 0x7FBF){ // jos rssi on kerrankin alle tämän vielä
mielivaltaisen arvon alla
            rssi = 0; // skannaus katsotaan tuloksettomaksi
            break;
        }
        else
            rssi = 1;
    }

    if (rssi && !logged){ // jos rssi ei pompsahtanut eikä olla kir-
jauduttu

        logout();
        delay(500);
        login();
        limit = 2000; // asetetaan skannausväli 2sek
    }
    else if (logged){ // jokin pumpsahti, ollaanko kirjauduttu

        logout();
        limit = 10000; // asetetaan skannausväli 10sek
    }
}

//kirjautumis näppäinkomennot

```

```

void login(){
    Keyboard.println(salasana);
    logged = true;
}

//lukitsemis näppäinkomennot
void logout(){
    logged = false;
    Keyboard.set_modifier(MODIFIERKEY_RIGHT_GUI); //Tell Teensy to
press Windows key
    Keyboard.set_key1(KEY_L); //Tell Teensy to press L
    Keyboard.send_now(); //Press "Windows key + L"
    delay(500); //Wait for half second
    Keyboard.set_modifier(0); //Tell Teensy to release Windows key
    Keyboard.set_key1(0); //Tell Teensy to release L
    Keyboard.send_now(); //Release "Windows key + L"
}
/*
switch (platform) {
    case OSX:
        Keyboard.press(KEY_LEFT_GUI);
        // Shift-Q logs out:
        Keyboard.press(KEY_LEFT_SHIFT);
        Keyboard.press('Q');
        delay(100);
        Keyboard.releaseAll();
        // enter:
        Keyboard.write(KEY_RETURN);
        break;
    case WINDOWS:
        // CTRL-ALT-DEL:
        Keyboard.press(KEY_LEFT_CTRL);
        Keyboard.press(KEY_LEFT_ALT);
        Keyboard.press(KEY_DELETE);
        delay(100);
        Keyboard.releaseAll();
        //ALT-s:
        delay(2000);
        Keyboard.press(KEY_LEFT_ALT);
        Keyboard.press('l');
        Keyboard.releaseAll();
        break;
    case UBUNTU:
        // CTRL-ALT-DEL:
        Keyboard.press(KEY_LEFT_CTRL);
        Keyboard.press(KEY_LEFT_ALT);
        Keyboard.press(KEY_DELETE);
        delay(1000);
        Keyboard.releaseAll();
        // Enter to confirm logout:
        Keyboard.write(KEY_RETURN);
        break;
}
*/

//tietokoneelta tuleva sarjadata
void sarjaluku(){
    inputString = "";
    while (Serial.available()){

        char inChar = (char)Serial.read(); // get the new byte
        inputString += inChar;
    }
}

```

7(9)



8(9)

```

        if((int)inChar == '\n'){
            stringComplete = true;
            //Serial.println(inputString);
            break;
            return;
        }
    }
}

//btmoduulilta tuleva sarjadata
void btsarjaluku(){
    btString = "";
    while (btString.indexOf("OK")<0 && btString.indexOf("FAIL")<0){
        if(Serial1.available())
            btString += (char)Serial1.read(); // get the new byte
    }
    //Serial.println(btString);
}

//bt-osoitteen lukeminen eepromilta
void btluku(){
    int alku = 100;
    int pituus = EEPROM.read(alku);
    alku++;
    if (pituus && pituus < 15){
        btaddr = "";
        for (int i = 0; i < pituus; i++){
            btaddr += (char)EEPROM.read(i+alku);
        }
    }
    btaddr = btaddr.substring(0, btaddr.indexOf('\n'));
    //Serial.println("mac: " + btaddr);
}

//nimen lukeminen eepromilta
void nimiluku(){
    int alku = 150;
    int pituus = EEPROM.read(alku);
    alku++;
    if (pituus && pituus < 48){
        btname = "";
        for (int i = 0; i < pituus; i++){
            btname += (char)EEPROM.read(i+alku);
        }
    }
    btname = btname.substring(0, btname.indexOf('\n'));
    //Serial.println("nimi: " + btname);
}

//bt-osoitteen tallenus eepromille
void bttallennus(){
    int alku = 100;
    btaddr = btaddr.substring(0, btaddr.indexOf('\n'));
    //Serial.println("MAC: " + btaddr);
    EEPROM.write(alku, btaddr.length());
    alku++;
    for (int i=0; i<=(btaddr.length());i++){
        EEPROM.write(i + alku, (int)btaddr.charAt(i));
    }
}

//nimen tallennus eepromille
void nimitallennus(){
    int alku = 150;

```

```
btname = bname.substring(0,bname.indexOf('\n'));
//Serial.println("nimi: " + bname);
EEPROM.write(alku, bname.length());
alku++;
for (int i=0; i<=(bname.length());i++){
    EEPROM.write(i + alku, (int)bname.charAt(i));
}
}
```

9(9)



```

line = fileStr;

found = line.find("USBSER000");
if ( found > 0)
valid = 1;

found = 0;
if(valid)
{
    eol = line.rfind('"');
    found= line.rfind('"',eol-1);
    if (found>0 && eol >0)
    {
        for (int i = found+1; i < eol; i++)
        {
            vali = line.at(i);
            comportti += vali;
        }
    }
}

myfile.close();
}

else cout << "Unable to open file";

cout << "Muodostetaan yhteys" << endl;

char commi[10] = "";
for (int i = 0; i<10; i++){
    commi[i] = '\\0';
}

//          1 23 456789
//"\\\\\\\\.\\COM99"

for (int i = 0; i < comportti.length(); i++)
{
    commi[i] = comportti.at(i);
}
cout << commi << endl;

Serial* SP = new Serial(commi);    // adjust as needed

char incomingData[256] = "";           // don't forget
to pre-allocate memory
char sendData[256] = "";
string test = "";
int dataLength = 256;
int readResult = 0;
int bytes = 0;
bool lippu =0;
bool sent = 0;
char kylla = '\\0';

while(SP->IsConnected())
{

```

```

cout << "Yhdistetty" << endl;
if (mode)
{
    int sizeis = 0;
    char merkki = argv[2][0];
    while (merkki != '\0')
    {
        sizeis++;
        merkki= argv[2][sizeis];
    }
    argv[2][sizeis] = '\n';
    sizeis++;
    cout << argv[2] << " " << sizeis << " merkkiaie" << endl;
    SP->WriteData(argv[2],sizeis);
    cout << "OK" << endl;
    system("PAUSE");
    return 0;
}
if (!mode)
{
    int goat= SP->WriteData(setname,sizeof(setname));
    if(!sent)
    {
        cout << "anna bluetooth nimi: ";
        getline(cin, line);
        //line += '\n';
        for (int i = 0; i < line.length(); i++)
        {
            sendData[i] = (char)line.at(i);
        }
        sendData[line.length()] = '\n';
        sent = SP->WriteData(sendData,line.length()+1);

        Sleep(500);
    }

    while(!lippu && sent){
        Sleep(1000);
        readResult = SP->ReadData(incomingData,dataLength);

        std::string test(incomingData);

        cout << "bluetooth nimi on nyt: " ;

        printf("%s",incomingData);
        test = "";

        while(kylla != 'e' && kylla != 'E' && kylla != 'k' &&
kylla != 'K')
        {
            lippu = 1;
            cout << "oikein? (k/e) ";
            cin >> kylla;

            if(kylla == 'e' || kylla == 'E')
            {
                lippu = 0;
                sent = 0;
                bytes = 0;
                for(int i = 0; i< line.length(); i++)
                {
                    sendData[i] = '\0';
                }
            }
        }
    }
}

```

4(2)

```
        line = "";
    }
    }
    kylla = '\\0';
}

if (lippu)
{
    cout << "bluetooth nimi muutettu" << endl;
    sent = SP->WriteData(nameok, sizeof(nameok));

    system("PAUSE");
    return 0;
}

}

    system("PAUSE");
    return 0;
}
system("PAUSE");
return 0;
}
```