

Sammendrag

Denne oppgaven tar for seg mulighetene for å overføre EKG-prosessdata via Bluetooth-teknologi. EKG-prosessdata er signaler generert ut i fra elektrisk aktivitet på hjertet. Bluetooth skal i denne sammenhengen fungere som et overføringsmedium, en bærer, for slik EKG-prosessdata mellom en pasient og en monitor. Videre tar oppgaven for seg interoperabiliteten med WAP, der WAP skal fungere som en alarm. Systemintegrasjonen mellom Bluetooth og WAP gjøres med bruk av en pasient-database.

Et EKG signal har en båndbredde fra 0-150 Hz. Det er mulig å kode dette signalet i forskjellige formater. Disse formatene har varierende kvalitet og i tillegg er båndbreddene forskjellige. Formatene som har blitt evaluert i denne oppgaven er WAV, MP3 og sampler.

Bluetooth er en teknologi, utviklet i fellesskap av en gruppe ledende IT- og telecomselskaper, for å tilkoble mobile enheter raskt og enkelt uten behov for kabler. Teknologien går ut på å gi støtte for trådløs kommunikasjon for tale og data ved bruk av kortbølge-radio.

Opgaven har vektlagt en lyd-overføring av slik EKG. Her finnes både overføring av filformater og sampler, som begge gir karakteristikk av EKG gjennom lyd. Overføringsmetodikken av filformater og sampler er ikke lik. Filformater tar bruk av en protokoll kalt OBEX. Denne protokollen gir også støtte for push-teknologi. Sampler derimot, kan overføres som tale via talekanal. Alternativt er å sekvensiere sampler, for så å produsere filer. Et slikt valg bringer oss tilbake til overføring av filformater.

Begrensningene i Bluetooth i dag er mobilitetsproblematikken, da teknologien isolert sett ikke gir noen støtte for mobilitet. Derfor egner ikke Bluetooth seg alene i et miljø basert på mobilitet, og er derfor ikke aktuelt i en slik sammenheng. Benytter man derimot Bluetooth i sammenheng med LAN-aksesspunkter vil mobilitetsspørsmålet løses.

Per dags dato finnes ingen interoptabilitet med WAP, der WAP er bærer for Bluetooth. Derfor er en systemintegrasjon av disse preget av moderate tilpasninger, for å få frem prosessdata. Den neste spesifikasjons-utgivelse av Bluetooth vil garantert ha bedre interoperabilitet med andre systemer, men en slik utgivelse er ikke ventet før 2001, kanskje så sent som i 2002. Da vil også GPRS være med på å gi ytterligere ”forbedringer” for WAP.

Forord

Denne rapporten er et resultat av en 10 vekttalls hovedoppgave ved sivilingeniørstudiet i informasjons og kommunikasjonsteknologi (IKT) ved Høgskolen i Agder, institutt for informasjonsteknologi i Grimstad.

Jeg vil takke Tien-Toan Do ved Universitetet i Hannover, institutt for kommunikasjonsteori og signalprosessering, som har kommet med utfyllende kommentarer og tips underveis. En stor takk sendes også veileder ved HiA, Rune Fensli, som har bidratt med råd og innspill forkant og underveis i arbeidet med rapporten.

Grimstad, Juni 2000

Per Rune Grønhovd

Innholdsfortegnelse

| | |
|---|-----------|
| <i>Sammendrag</i> | <i>1</i> |
| <i>Forord</i> | <i>2</i> |
| 1 Innledning | 7 |
| 1.1 Bakgrunn | 8 |
| 1.1.1 Elektrokardiogram | 8 |
| 1.2 Opprinnelig prosjektbeskrivelse | 9 |
| 1.3 Avgrensninger og målformulering | 9 |
| 1.3.1 Problemstillinger | 10 |
| 1.4 Motiv | 10 |
| 1.5 Rapportens struktur | 11 |
| 1.6 Litteratur | 11 |
| 1.6.1 Internet – den reddende engel | 11 |
| 1.6.2 Beslutninger | 11 |
| 1.6.3 Videre prosess | 11 |
| 1.7 IKT-senter | 11 |
| 2 Introduksjon – Bluetooth | 13 |
| 2.1 Visjonen | 13 |
| 2.2 The Bluetooth Special Interest Group (SIG) | 13 |
| 2.2.1 Spesifikasjonens oppbygning | 13 |
| 2.3 Teknologiens virkemåte | 13 |
| 2.3.1 Bluetooth radio | 13 |
| 2.3.2 Baseband | 14 |
| 2.3.3 Linker og Link Controller | 15 |
| 2.3.4 Link Manager og applikasjons software | 15 |
| 2.4 Nettverk | 15 |
| 2.5 Pakker | 16 |
| 2.5.1 Aksesskoder | 16 |
| 2.5.2 Pakkeheader | 16 |
| 2.5.3 Pakketyper | 16 |
| 2.6 Protokoller | 16 |
| 2.6.1 Protokoll-stakk | 16 |
| 2.6.2 Logical link control and adaption protocol (L2CAP) | 17 |
| 2.6.3 Link Manager Protocol | 18 |
| 2.6.4 RFCOMM | 18 |
| 2.6.5 Service Discovery Protocol | 18 |
| 3 Teoretisk løsningsforslag (Metode) | 19 |
| 3.1 Innledning | 19 |
| 3.2 Generelt om Ericsson Bluetooth Development Kit | 19 |

| | |
|---|-----------|
| 3.2.1 Tilkobling og overføringshastighet | 20 |
| 3.2.2 Overføringshastighet i RS232 | 20 |
| 3.2.3 Kontakt med Bluetooth-hardware – HCI kommandoer | 20 |
| 3.3 Planoversikt | 21 |
| 3.3.1 Innledning | 21 |
| 3.3.2 Bruk av bærbar PC | 22 |
| 3.3.3 Monitor | 22 |
| 3.3.4 Database | 23 |
| 3.4 Datarepresentasjon | 23 |
| 3.4.1 Båndbredde | 23 |
| 3.4.2 Sampling | 23 |
| 3.4.3 Digital representasjon av EKG | 24 |
| 3.4.4 Sanntidsløsning | 24 |
| 3.4.5 Audio-overføring | 24 |
| 3.5 Overføring av EKG i Bluetooth | 24 |
| 3.5.1 Generelt | 24 |
| 3.5.1 Protokollstakk for overføring av filer | 24 |
| OBEX | 25 |
| 3.5.2 Push-teknologi i Bluetooth | 25 |
| 3.6 Beskrivelse av filformater for overføring av data | 26 |
| 3.6.1 Wav | 26 |
| 3.6.2 Mp3 | 27 |
| 3.6.3 Sampler | 28 |
| 3.6.4 Komprimering av lydfiler | 28 |
| 3.7 Dynamisk båndbredde | 29 |
| 3.8 Generelt ved sending og mottak av data i Bluetooth | 29 |
| 3.8.1 Tidsdelt overføring | 29 |
| 3.8.2 Synkronisering | 29 |
| 3.8.3 Buffre ved sending | 29 |
| 3.8.4 Buffre i mottak | 30 |
| 3.8.5 Flytkontroll | 30 |
| 3.9 Til videre bruk | 30 |
| 3.9.1 Interoperabilitet med WAP | 30 |
| 3.9.2 Bruk av database og SQL | 31 |
| 3.9.3 Hjemme-database | 31 |
| 3.9.4 Sampler eller filer? | 31 |
| 3.10 Alternativ overføring | 32 |
| 3.10.1 TCP/IP og streaming audio | 32 |
| 4 Mobilitetsproblematikk i Bluetooth | 33 |
| 4.1 Innledning | 33 |
| 4.2 Definisjoner | 33 |
| 4.2.1 Piconett | 33 |
| 4.2.2 Master-enhet | 33 |
| 4.2.3 Slave-enhet | 33 |
| 4.2.4 Scatternet | 33 |

| | |
|--|-----------|
| 4.3 Nettverkstopologi | 33 |
| 4.4 Master-slave definisjon | 34 |
| 4.5 Hvordan tiden beregnes | 34 |
| 4.6 Datatransmisjon | 34 |
| 4.7 Etablering av forbindelse | 34 |
| 4.7.1 Standby | 35 |
| 4.7.2 Page- og forespørselprosedyre | 35 |
| 4.7.3 De andre tilstandene | 36 |
| 4.8 Etablert forbindelse | 36 |
| 4.8.1 Connection | 36 |
| 4.8.2 Modus | 36 |
| 4.9 Skjema | 37 |
| 4.9.1 Polleskjema | 37 |
| 4.10 Scattnet | 37 |
| 4.10.1 Kommunikasjonen mellom flere piconett | 37 |
| 4.11 Et sykehusmiljø | 38 |
| 4.12 Plassering av enhetene | 38 |
| 5 Sikkerhet | 39 |
| 5.1 Innledning | 39 |
| 5.2 Sikkerhetsregler ved bestemt bruk – i dette fall EKG | 39 |
| 5.3 Oversikt over sikkerhetsarkitektur i Bluetooth | 39 |
| 5.3.1 Autetisering og autorisasjon | 40 |
| 5.3.2 Kryptering | 40 |
| 5.3.3 Sikkerhetsnivå på enheter og tjenester | 40 |
| 5.3.4 FEC og ARQ - sikkerhet i pakkeoverføring | 41 |
| 5.4 Sikkerhet på linklag | 41 |
| 5.5 Spread-spectrum teknologi og frekvens-hopping | 42 |
| 5.6 Generelle krav til kompetanse, service og kontroll | 42 |
| 6 Bluetooth i et fremtidig perspektiv – medisinsk-teknisk hjelpemiddel? | 43 |
| 6.1 Generelt - Bluetooth vs. annet trådløs overføring | 43 |
| 6.2 Bluetooth og EKG | 43 |
| 7 Resultater | 44 |
| 7.1 Overføringskjema | 44 |
| 7.2 Programvare | 44 |
| 7.3 Oppkobling mot EBDK | 44 |
| 7.4 File Transfer applikasjon | 45 |
| 7.5 Empirisk kunnskap | 47 |
| 8 Drøfting og diskusjon | 49 |

| | |
|---|-----------|
| 8.1 EKG-prosessdata via Bluetooth | 49 |
| 8.2 Teoretisk løsningsforslag | 49 |
| 8.3 Mobilitet | 50 |
| 8.3.1 Generelt | 50 |
| 8.3.2 Et sykehusmiljø | 50 |
| 8.3.3 Et hjemmemiljø | 51 |
| 8.4 Sikkerhet | 51 |
| 8.4.1 Tiltak i teknologien | 51 |
| 8.4.2 Overføringer | 52 |
| 8.4.3 Dimesjonering av nettene i et større miljø | 52 |
| 8.4.4 I en hjemmesituasjon | 53 |
| 8.4.5 Behandling av utstyr | 53 |
| 8.4.6 Forskjellige fabrikanter | 53 |
| 9 Konklusjon | 54 |
| 10 Referanser | 55 |
| <i>Fra Bluetooth Specification Vol 1.0b – Core</i> | 55 |
| <i>Fra Bluetooth Specification Vol. 1.0b – Profiles</i> | 55 |
| <i>Fra Internett</i> | 55 |
| <i>Fra bøker</i> | 57 |
| <i>Annet</i> | 57 |
| Fra Internett – ekstra | 58 |
| 11 Vedlegg | 59 |
| 11.1 Vedleggsliste | 59 |
| 1 - Forkortelser benyttet i rapport | 60 |
| 2 - Etablering av slave | 62 |
| 3 - Etablering av master | 67 |
| 4 - HCI-kommandoer | 72 |

Figurliste

| | |
|---|----|
| <i>Figur 1 : Oversiktskart over radiodel og baseband i Bluetooth</i> | 14 |
| <i>Figur 2 : To piconett kan kobles sammen til et scatternett</i> | 15 |
| <i>Figur 3 : Pakkeformatet i Bluetooth</i> | 16 |
| <i>Figur 4 : Oversiktsbilde av protokollstakken</i> | 17 |
| <i>Figur 5 : L2CAP i protokoll-arkitekturen</i> | 18 |
| <i>Figur 6 : Ericsson Bluetooth Development Kit</i> | 20 |
| <i>Figur 7 : Bluetooth Host Controller</i> | 21 |
| <i>Figur 8 : Den ideelle planløsningen</i> | 21 |
| <i>Figur 9 : Den reelle planløsningen som muliggjorde EKG-prosessoverføring</i> | 22 |
| <i>Figur 10 : Et "normalt" EKG-signal</i> | 23 |
| <i>Figur 11 : Bruksmodell ved overføring av filer/sampler</i> | 25 |
| <i>Figur 12 : Protokoll-modell for push</i> | 25 |
| <i>Figur 13 : Push-teknologi danner rollemønster; Client og Server</i> | 26 |
| <i>Figur 14 : Formatdelen i RIFF</i> | 26 |
| <i>Figur 15 : Innledningen på datadelen i RIFF</i> | 27 |
| <i>Figur 16 : Perceptual encoding</i> | 27 |
| <i>Figur 17 : Buffer i sender</i> | 30 |
| <i>Figur 18 : Tilstandsdiagram for enheter</i> | 35 |
| <i>Figur 19 : Oversikt over sikkerhetsarkitekturen i Bluetooth</i> | 40 |
| <i>Figur 20 : Fire forskjellige benevnelser på linklag</i> | 41 |
| <i>Figur 21 : Overføring fra maskin til maskin vha EBDK</i> | 44 |
| <i>Figur 22 : Primærvinduet i programvaren til EBDK</i> | 44 |
| <i>Figur 23 : Overføring mellom PC og EBDK bestemmes</i> | 45 |
| <i>Figur 24 : Slave er blitt aktivisert</i> | 45 |
| <i>Figur 25 : Et utdrag av oppkoblingsprosedyren til slaven</i> | 46 |
| <i>Figur 26 : Oppgi enhetsadresse til den aktiverte slaven</i> | 46 |
| <i>Figur 27 : Master er blitt aktivisert</i> | 47 |

1 Innledning

1.1 Bakgrunn

I moderne tid har helsesektoren vært avhengig av avansert teknisk utstyr. Slikt medisinsk teknologi er med på å gi større informasjon og skape rasjonaliserte behandlingsmetoder. Moderne utstyr og ny teknologi er årsak til mesteparten av den velstanden vi har oppnådd [13]. Teknologi har også gjort pasienter mer selvhjulpne, samt frigjort arbeid for personell slik at de kan bruke mer tid på oppgaver som bare mennesker kan gjøre.

Ett av de mer kjente tekniske utstyrene en kjenner fra helsesektoren er Electrocardiogram (EKG), et apparat som måler forandringer i elektrisk aktivitet på hjertet. Ved forståelig visuell fremstilling gir slik teknologi verdifull informasjon. Slike apparater har tradisjonelt vært brukt ved fast fysisk tilkobling mellom pasient og overvåkningsutstyr/monitor.

Ved å innføre radiolinje mellom pasienten og monitoren vil man få en løsning som gir friheter for pasienten, da man får muligheter til å bevege seg samtidig med å være pålogget monitoren. Et slikt radiolinje-samband kan eksempelvis være Bluetooth.

Bluetooth er en trådløs teknologi som tillater brukere trådløs og umiddelbare tilkoblinger mellom diverse forskjellige enheter uten anstrengelse. Navnet har teknologien fått fra den danske kongen Harald Blåtann, Fordelen med Bluetooth-teknologien er at den gir mulighet for trådløs overføring, og dette til en relativt rimelig penge [12]. Nå finnes det ingen dokumentasjon på om Bluetooth kan bli noe besparende i helsesektoren, dette blir en faktor ut i fra behovet. Det som er helt sikkert, er at denne teknologien representerer en nyvinning, som i all hovedsak kan revolusjonere dagens drift. Uansett vil satsing på denne form for teknologi være en investering.

Før regjeringsskiftet ble det fra sentrale hold i Norge dokumentert at teknologiske investeringer i helsesektoren prioriteres. Den sittende regjeringen gikk inn for økte bevilgninger i utstyrs-investeringer i statsbudsjettet [14]. I Sosial- og helsedepartementets budsjettforslag for 2000 [14] dokumenteres det også for at Stortinget har vedtatt en femårsplan for å bedre utstyrssituasjonen på norske sykehus. I det reviderte nasjonalbudsjettet, fremlagt av den nye regjeringen, ble utstyrsplanen utvidet med ytterligere to år [15]. Dette kan blant annet bidra til en senere innføring av Bluetooth i helsesektoren.

Oppgaven er gitt i oppdrag fra høgskolelektor Rune Fensli ved Høgskolen i Agder, avdeling Grimstad. Oppgaven skal danne et grunnlag for videre klinisk undersøkelse i forbindelse med doktoravhandling.

1.1.1 Elektrokardiogram

Et elektrokardiogram (EKG) er et medisinsk apparat som måler forandringer i elektrisk aktivitet på hjertet. Når en kombinerer dette med en stress-test, er den nyttig i å gi diagnostikk på hjerteproblemer. Ved bruk av EKG benyttes elektroder, festet til armer, ben og/eller bryst, som formidler/fremskaffer elektriske pulser. Hjertets elektriske aktivitet blir vist på en skjerm/monitor og noen ganger også risset ut på papir.

Fra et normalt sunt hjerte vises en karakteristisk kurve. Hvis det forekommer noe uregelmessigheter i hjerterytmene, eller at det er en skade på hjertemuskelen, kan dette

endre den elektriske aktiviteten og vil dermed utmerke seg ved av den karakteristiske kurven blir endret.

EKG blir også benyttet i øyeblikkelig hjelp, større operasjoner og noen ganger under barnefødsler for å overvåke pasientens hjerte.

[16]

1.2 Opprinnelig prosjektbeskrivelse

Oppdragsgiver gav følgende prosjektbeskrivelse i forbindelse med utlysningen av oppgaven :

Det skal undersøkes hvilke muligheter som finnes for trådløs overføring av analoge og digitale prosessdata ved å benytte Bluetooth-teknologi.

Metoder for digitalisering og pakkesvitsjing skal evalueres med hensyn på mulig dynamisk båndbredde, og aktuelle komprimeringsteknikker skal beskrives og vurdres.

Det skal utvikles en testapplikasjon for å realisere de løsninger som en finner er mest egnet for slik dataoverføring, og systemet skal testes i forhold til pålitelighet og sikkerhet i overføringene.

1.3 Avgrensninger og målformulering

Bluetooth's spennvidde anses som stort, og en ville aldri ha kommet i mål med noen form for vitenskapelig avhandling hvis en ikke begrenset oppgaven. I og med at oppdragsgiver Rune Fensli har personlig interesse innen klinisk teknologi, og spesielt på EKG, ble det mer eller mindre et naturlig valg å rette seg inn på dette.

Det er lagt vekt på å få frem hvilken teknologi som ligger til rette i Bluetooth, som en løsning for bærer av EKG-prosessdata fra pasient til overvåkningsutstyr, eventuelt til en felles lagringsbase for flere pasienter. I denne løsningen er det ikke tatt hensyn til generell databehandling, ei heller sanntidsløsninger, men heller belyst mulighetene med Bluetooth som den integrert delen i denne løsningen.

I den opprinnelige prosjektbeskrivelsen er det lagt vekt på termer som digitalisering, pakkesvitsjing, båndbredde og komprimering. Pakkesvitsjing gir kun fornuft i en løsning med flere pasienter hvor vi "organiserer" disse. I et slikt miljø med flere pasienter er det naturlig å se på mobiliteten i forbindelse med pakkesvitsjingen. Båndbredden er også nært tilknyttet til antall pasienter. Derfor har det blitt lagt vekt på å også gi utredninger om bruk av teknologien i et sykehusmiljø, og derfor også hensynet til mobilitet.

Et annet aspekt i teknologien som bør belyses i en slik oppgave er sikkerheten. Det være seg sikkerhet som ligger i teknologi og sikkerhet i bruk av teknologi. Det er ikke gjort dype utgreiinger i dette emnet, men siden det så viktig har det blitt belyst, og gjort rede for relatert informasjon om sikkerhet i forhold til kjernen av oppgaven.

Ut i fra opprinnelig prosjektbeskrivelse gir den også i oppgave å utrede muligheter med Bluetooth. Denne delen inngikk ikke dype tekniske termer, men heller på mulighetene for Bluetooth som medisinsk hjelpemiddel.

Utformingen av konkrete mål for hovedoppgaven er en dynamisk prosess. Mål og målformuleringer forandret seg underveis, etterhvert som man arbeider med oppgaven.

Mål for oppgaven er :

- å gi en introduksjon om Bluetooth
- å belyse mulighetene som ligger til rette for å benytte Bluetooth som bærer for EKG-prosessedata, samtidig å gjøre tilpasninger av valgte verktøy
- å få frem aspekter rundt sikkerheten i Bluetooth, og belyse om en slik teknologi egner seg i "sykehusmiljø"
- å se på mobilitetsproblematikken i Bluetooth
- å belyse fremtidspektivene for bruk av Bluetooth i medisinsk sektor.
- å overføre EKG-prosessedata ved hjelp av EBDK

1.3.1 Problemstillinger

Utover målene som ble satt, dannes også flere problemstillinger. Disse problemstillingene har utviklet seg i takt med oppgaveutførelsen, og har vært en såkalt dynamisk prosess.

Punktvis er disse problemstillingene:

- få frem representative overføringsmål
- se på filformater og båndbredde
- legge frem forslag på digitalisering og samplingfrekvens
- se på protokoller ved overføring
- se på push-teknologi
- videre prosess, spesielt interoperabilitet med WAP

1.4 Motiv

Motivet for denne hovedoppgaven er at Bluetooth anses som et høyst aktuell teknologi i medisinsk sektor. I mest mulig grad vil en implementasjon få betydning for arbeids- og pasientsituasjonen. Å gjøre hverdagen kabelfri er en selvsagt ønsket situasjon.

Satt i perspektiv kan teknologen komme til å gi uante muligheter. Syke mennesker kan få mulighet til å være hjemme, logget til en terminal via Bluetooth, for så videre å sende datainformasjon videre, til en felles database, via andre alternative teknologer. Et slikt alternativ er WAP.

Det mest spennende er hvordan overføringen kan foregå i et sykehusmiljø, der det er flere brukere. Her ligger det nok et naturlig ønske om mobile løsninger. Dette igjen skaper et nytt fokuspunkt; hvordan data blir behandlet mellom pasienter. Med andre ord, hvilke løsninger, om noen, finnes for "pakkesvitsjing" mellom pasienter?

I og med at EKG gir livsviktig informasjon er sikkerheten et naturlig aspekt. Flere medier belyser sikkerhet i medisinsk teknologi, deriblant Teknisk Ukeblad [46]. Her ble det lagt spesielt vekt på de tekniske problemene, som i verste fall kan koste liv. Dette ga svar på at det er enormt viktig at ny teknologi tilfredsstiller sikkerhetsaspekter, slik at det kan skapes robuste systemer.

Motivet bak modellen, er å se hvordan en skal legge til rette for EKG-prosessedata. Hvilke hensyn skal man her ta og hvordan skaper en automatikk? Dette gir en fremstilt løsning, basert på en utredning av teknologien.

At det bygger seg opp mange spørsmål, må sees på som et godt motiv for en slik oppgave.

1.5 Rapportens struktur

De målformuleringene og avgrensningene som ble tatt, gjorde at oppgaven delte seg i tre hoveddeler; introduksjon av Bluetooth, et teoretisk løsningsforslag (metode) og emnet angående mobilitetsproblematikk. I tillegg har det blitt innført egne kapitler om generell sikkerhet i Bluetooth, og Bluetooth som et fremtidig medisinsk-teknologisk hjelpemiddel.

Videre er det resultater av overføring av lydfiler ved bruk av Ericsson Bluetooth Development Kit.

Den siste delen av rapporten inneholder resultater og drøfting. Her drøftes det teoretiske løsningsforslaget, samt mobilitetsproblematikken.

1.6 Litteratur

Å få gleden av å sette seg inn i en ny teknologi har vært stor. Men det har også hatt sin pris – ingen faglitterære bøker har til nå blitt utgitt. Følgene har blitt et totalt avhengighetforhold til Internet som fagformidler.

1.6.1 Internet – den reddende engel

På ”nettet” finnes det utallige sider som tar for seg teknologien Bluetooth, der noen er mer troverdige enn andre. For nettopp troverdighet er viktig i et arbeid som krever nøyaktige opplysninger. På grunnlag av min knappe kompetanse på dette feltet før prosjektetstart, trengtes mye informasjon. Det mest naturlige å starte med var spesifikasjonen. Utgiverne av spesifikasjonene i Bluetooth, SIG, ga ut i desember 1999 Spesifikasjon 1.0b, en fornyelse av den første spesifikasjonen 1.0a fra juli 1999 [12,17] Mine beslutninger i forhold til teknologien har til dels basis fra denne spesifikasjonen.

1.6.2 Beslutninger

Spesifikasjonen inneholder tørre fakta, og er ikke et godt beslutningsmatriell. For å lage løsninger trengs flere bein å stå på. I et vanlig arbeidsmiljø ville bøker være det primære valget i en slik sammenheng. I mitt tilfelle har jeg vært nødt til å stole på mine egne slutninger av spesifikasjonen, informasjon fra nyhetsgrupper samt utsagn fra personer i utviklingsmiljøer.

1.6.3 Videre prosess

Metodeløsningen er en skisse basert på teori og tilgjengelig utstyr. Den videre prosess må være å skape en automatikk i en slik løsning.

1.7 IKT-senter

Selskapene Telenor og Ericsson danner, sammen med en rekke andre bedrifter, Grimstad teknologipark. Høgskolen i Agder er en del av denne parken. På høgskolens område i Grooseveien er Agder IKT-senter etablert. Dette er en samlokalisering av forskere fra selskapene fra teknologiparken. Senteret skal skape en møteplass mellom

industri og høgskole. Det skal utvikle et sterkt miljø av forskere og ha rollen som kraftsenter og inspirasjonskilde for IKT-næringen på Agder.

IKT-senteret fungerer stort sett som et laboratorium. På denne "labben" ble to maskiner koplet opp med Ericsson Bluetooth Development Kit (EBDK). På denne måten kunne studenter med Bluetooth oppgave utføre eksperimenter med denne teknologien. [18]

2 Introduksjon – Bluetooth

2.1 Visjonen

Tidlig i 1998 begynte en gruppe av ledende IT- og telecomselskaper i fellesskap å utvikle en standard – en teknologi. Denne standarden gikk ut på å få brukere tilkoppet til mobile enheter raskt og enkelt, uten noen behov for kabling. Denne teknologien fikk navnet *Bluetooth*.

Teknologien består av en spesifisering for trådløs kommunikasjon av tale og data ved bruk av kortbølge radio. Det at teknologien er åpen betyr [19], i motsetning til kabler, at det kun vil være en type av Bluetooth. Hvilken som helst Bluetooth-sertifisert enhet vil kunne kommunisere med en annen Bluetooth-sertifisert enhet, uansett hvor en befinner seg hen i verden, noe som igjen var ideen bak denne standarden. [20]. Dette vil si at systemer fra de ulike leverandørene kan operere problemfritt sammen. [21]

2.2 The Bluetooth Special Interest Group (SIG)

Noen måtte stå ansvarlig for å videreutvikle og for å få markedsført teknologien. Dette ga grunnlaget for The Bluetooth Special Interest Group (SIG), som er en interessegruppe for teknologien. Denne gruppen var i starten dannet av de bedriftene som stod for utviklingen. Disse bedriftene var Ericsson, IBM, Intel, Nokia og Toshiba. I dag har det kommet ytterligere fire medlemmer i det såkalte ”promoter”-gruppa - 3Com, Lucent, Motorola og Microsoft. I tillegg finnes det ca. 1575 såkalte ”adopter”-selskaper [12].

For å bli *adopter* trengs kun å fylle ut en lisensavtale som gir rett til å benytte spesifiseringen for utvikling og produsere produkter og software. Denne lisensen er gratis, og gir også tilgang på dokumenter tidlig og oppdaterte versjoner av spesifiseringen. Videre kan en delta på ”online-forums”, og være med på konferanser.

2.2.1 Spesifikasjonens oppbygning

SIG er ansvarlig for utvikling av spesifiseringen. En slik notat inneholder all informasjon som trengs for Bluetooth-kommunikasjon mellom enheter. Etter at den første utgaven kom sommeren 1999 (se kapittel 1.6.1), ble en oppdatering utgitt i desember 1999 – Spesifisering 1.0b. Denne spesifiseringen er todelt. En del tar for seg kjernen av teknologiens oppbygning (Core), mens en annen del tar for seg de nødvendige prosedyrene for de forskjellige Bluetooth-applikasjonene (Profiles).

2.3 Teknologiens virkemåte

Grovt sett kan en dele opp Bluetooth-systemet følgende [22]:

- en radio som sender og mottar data og tale
- en baseband eller en link-kontroll enhet som prosesserer de sendte/mottatte data
- linkadministrasjons SW (Link Manager) som administrerer overføringene og gir støtte til applikasjons SW

2.3.1 Bluetooth radio

Radioen i Bluetooth er en kortbølge, laveffekts radio som opererer i den globale tilgjengelige 2,4 GHz-båndet. Rekkevidden er 10 meter ved nominell antennegain 0dBm (1mW) [23]. Når rekkevidden er 10 meter vil det bety at kommuniserende enheter må være mindre enn 10 meter fra hverandre. Alternativt vil en ha rekkevidde opp til 100 meter ved bruk av antennegain (effekt) på 20 dBm (100mW).

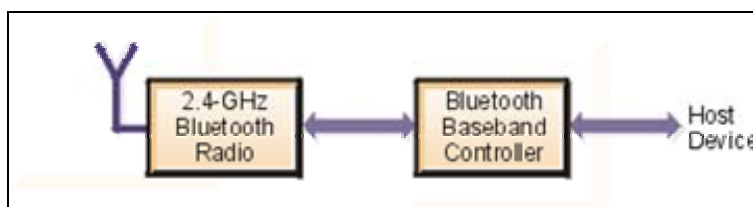
Overføringshastigheten er maksimalt 1Mbps, men på grunn av informasjon som ligger i protokoll-headere blir den nyttige informasjonsraten drøyt 721 Kbps.

Radiokommunikasjon er utsatt for både støy og interferens. Når 2,4 GHz-båndet er delt mellom flere typer utstyr som mikrobølgeovner, LAN's og andre applikasjoner, vil en tro at intereferens vil være et stort problem i Bluetooth. Dette er løst ved å anvende såkalt *spectrum-spreading* (se kapittel 5.5) [12], som gjør at Bluetooth-radioen hopper mellom flere frekvenser i et høyt tempo. Det finnes 79 forskjellige slike hopp, som starter på 2,402 GHz og stopper på 2,480 GHz. Interferens blir derfor unngått ved hoppingen, en kanalhoppesekvens som er unik for hvert piconett (se kapittel 4.2.1), som foregår 1600 ganger i sekundet. Dette gjør linken robust [22]. Hvis overføringen får et intereferens-sammenstøt, holdes tiden 1/1600-del, det vil si 625 μ s, til det neste frekvenshoppet og overfører igjen på den nye frekvensen. Frekvenshopping skaffer også datasikkerhet da to forskjellige datapakker aldri blir sendt over samme frekvens etter hverandre, samt datasikkerhet ved at de vekslende frekvensene er uforutsigbare.

Bluetooth hopper med et raskere tempo, og med mindre pakker (se kapittel 2.5) sammenliknet med andre systemer på samme frekvensbåndet. Dette gjør Bluetooth mer robust enn andre systemer [22]

2.3.2 Baseband

Når det gjelder trådløs kommunikasjon er baseband et hardware som former radiosignalene inn til et digitalt format som kan prosesseres av applikasjonen. Med andre ord så konverterer den digital- eller taledata inn på en slik form som kan bli overført ved bruk av radiosignal, ifølge en protokoll som tillater dekoding med ett den er mottatt.



Figur 1 : Oversiktskart over radiodel og baseband i Bluetooth [22]

Da Bluetooth enheter kan sende og motta signaler på samme tid må det være en form for å differensiere overføringer fra hverandre. Enhver enhet trenger altså å vite hvor informasjon kommer ifra. I trådløs kommunikasjon er det slik at data blir overført i pakker, der hver pakke inneholder en forhåndsbestemt mengde av data. Pakkene inneholder også informasjon om hvor den er sendt fra og hvor den skal sendes til. Derfor har pakker som kommer fra en bestemt enhet en bestemt ID, mens pakker som kommer fra en annen enhet har en annen ID. I pakkene finnes det også informasjon om hvordan dataene er blitt komprimert, rekkefølge slik de vare sendt og informasjon som blir brukt for å korreksjon. Ved mottak blir dataene kontrollert.

Alle oppgavene som er beskrevet ovenfor blir utført av baseband-prosessoren. Den er ansvarlig for å konvertere data fra en form til en annen, gjøre om til pakker samt tilegne både ID og feildeteksjonsinformasjon. Ved mottak reverseres prosessen for

dataene som er mottatt. I Bluetooth kaller man denne baseband-funksjonen for *Link Controller* [1].

2.3.3 Linker og Link Controller

Overføringer skjer ved hjelp av linker. Bluetooth støtter to forskjellige typer linker, Synchronous Connection-Oriented (SCO) og Asynchronous Connectionless (ACL) [1]. SCO benyttes i hovedsak for tale, mens ACL er for data.

Hvilken som helst enhet i et Bluetooth-system kan benytte begge linktypene, en av gangen, og kan endre sin linktype under overføring.

Link Controller er en oppsynsfunksjon som håndterer alle funksjonene i baseband og som støtter Link Manager (se neste kapittel). Denne funksjonen sender og mottar data, ber om ID fra enheter som har sendt, legaliserer (autentiserer) linker, setter opp linktype, determinerer hvilken rammetype i et pakke-til-pakke basis, og dirigerer hvordan enheter vil lytte til overføringer fra andre enheter eller eventuelt legge dem på vent [22].

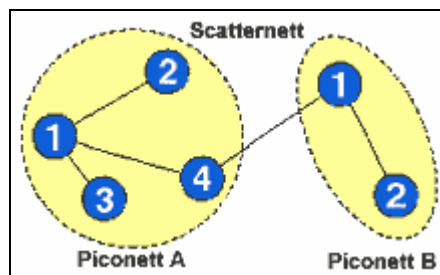
Hver pakke tar bruk av en tidsluke på $625\mu\text{s}$, det vil si varigheten mellom to frekvenshopp. Denne kan utvides til å dekke fem tidsluker. Bluetooth gir støtte til en asynkron datakanal, tre synkrone talekanaler, eller som en kombinasjon av disse. Den asynkrone datakanalen kan yte 721 kbps i en retningene i en asymmetrisk link. Returkanalen kan være opp mot 57,6kbps. I en symmetrisk link er hastigheten 432,6kbps [25].

2.3.4 Link Manager og applikasjons software

Link Manager er software som blir benyttet av en mikroprosessor [22]. Dette gjør det mulig å administrere kommunikasjonen mellom Bluetooth-enheter. Hver Bluetooth-enhet har ens egen Link Manager som oppdager andre fjerntliggende link-managere, og kommuniserer med disse for å håndtere ting som setup, autentisering og konfigurasjon og protokoller. Protokollen som benyttes er Link Manager Protocol (LMP), som blir beskrevet nærmere senere i rapporten.

2.4 Nettverk

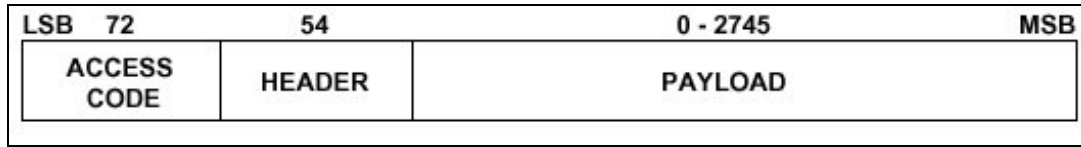
Alle Bluetooth-enheter innen rekkevidde vil automatisk kunne koble seg opp i et såkalt "piconett" (se kapittel 4.2.1). Dette nettverket støtter opp til åtte enheter (en master og syv slaver) og en av disse vil automatisk fungere som master [1]. Flere av disse piconettene kan kobles sammen for enda flere enheter i et "scatternett" (se kapittel 4.2.4). To piconett kan også fungere uavhengig av hverandre. Overføringskapasiteten på 1Mbps tildeles mellom alle enheter innenfor det samme piconettet.



Figur 2 : To piconett kan kobles sammen til et scatternett [22]

2.5 Pakker

Bitrekkefølgen følger Little Endian-formatet [1]. Det betyr i dette tilfellet at LSB er det første bitet som sendes. Dataene på en piconett-kanal blir overbrakt i form av pakker. Det generelle pakkeformatet inneholder 3 deler; aksesskode, header, og payload.



Figur 3 : Pakkeformatet i Bluetooth. Den består av 3 deler; aksesskode, header og payload [1].

Aksesskoden og headeren er av fast størrelse, mens payload kan variere fra null til 2745 bits.

2.5.1 Aksesskoder

Hver pakke starter med en aksesskode. Denne aksesskoden er benyttet for synkronisasjon og identifikasjon [1]; aksesskoden identifiserer alle pakkene som blir utvekslet på en piconett-kanal. Alle pakkene som sendes på den samme piconettet har den samme aksesskoden. I mottakeren til en Bluetooth-enhet finnes det en korrelator som korrelerer mot aksesskoden og bestemmer timingen til mottakeren

Aksesskoden blir også benyttet ved paging- og forespørselprosedyrer. I et slikt tilfelle finnes det ikke header og payload.

Det er tre typer aksesskoder :

- Channel Access Code (CAC) – identifiserer et piconett og benyttes i alle pakker sendt på en piconett-kanal.
- Device Access Code (DAC)– blir benyttet for signaleringsprosedyrer, herav paging og respons til paging.
- Inquiry Access Code (IAC) – blir benyttet for å oppdage andre enheter innen rekkevidden, og deres karakteristikk.

2.5.2 Pakkeheader

Headeren inneholder informasjon om Link Controlleren (LC). Her gis informasjon om blant annet om medlemsadresse i et piconett (AM_ADDR) og hvilken pakketype som sendes (TYPE) [1].

2.5.3 Pakketyper

Pakkene som benyttes i et piconett kobles til den fysiske linken de blir benyttet i (ACL eller SCO). For hver av disse linkene er det definert 12 forskjellige pakketyper. En av pakketypene er FHS (Frequency Hopping Synchronisation) som blant annet gir informasjon om den ”innfødte klokken” (se kapittel 4.5) [1].

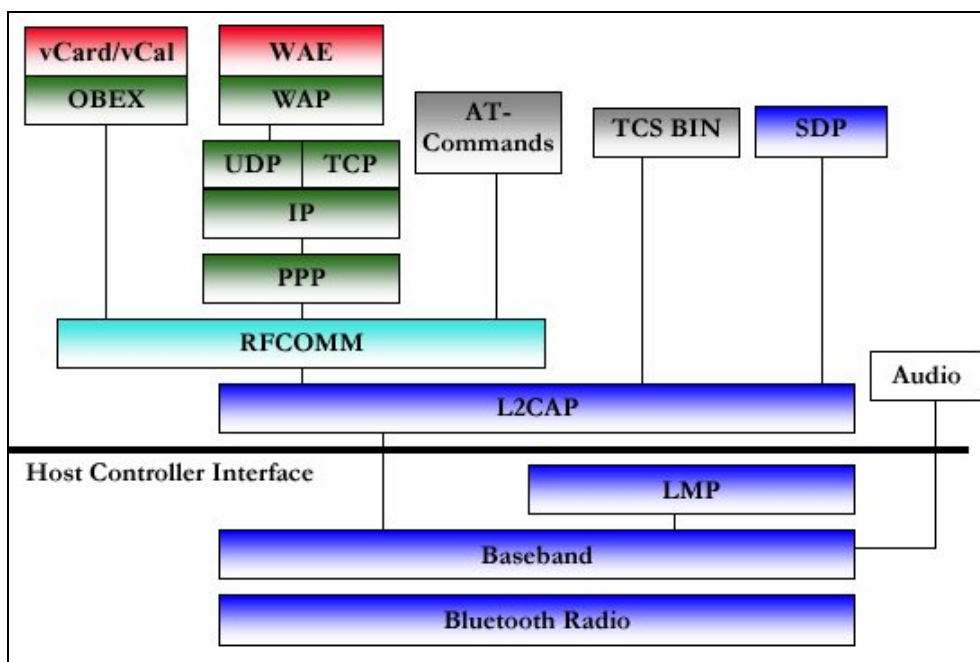
2.6 Protokoller

2.6.1 Protokoll-stakk

Målet med spesifikasjonen er å tillate applikasjoner, skrevet på en slik måte som er i samsvar med spesifikasjonen, til å operere med hverandre. For å få dette til å fungere

må korresponderende applikasjoner i kommuniserende enheter benytte samme protokoll-stakker.

Forskjellige applikasjoner kan føres over forskjellig protokoll-stakker. Hver eneste av disse forskjellige protokoll-stakkene bruker en felles Bluetooth datalink og fysisk lag. Som vi ser av protokollsammensetningen i Bluetooth, gir denne ingen fast stakk, men gir muligheter til å benytte forskjellige protokoller i de forskjellige lagen bortsett fra lagene baseband (fysisk lag) og radio (datalink) [47].



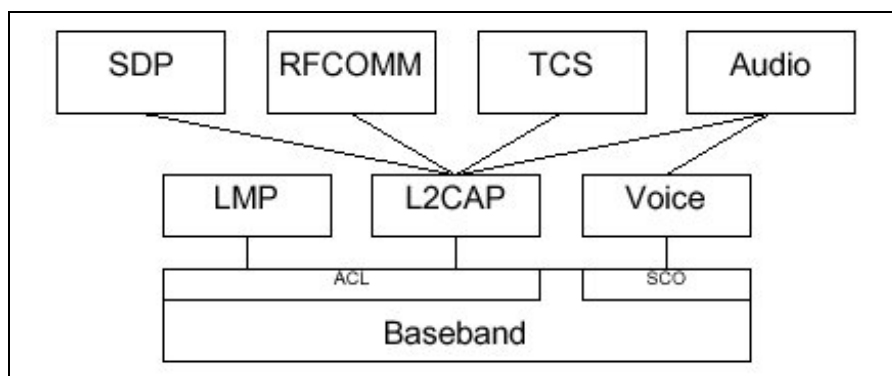
Figur 4 : Oversiktsbilde av protokollstakken [47]

2.6.2 Logical link control and adaption protocol (L2CAP)

Logical link control and adaption protocol, forkortet L2CAP, tilbyr forbindelsesorienterte og forbindelsesløse datatjenester til øvre lags protokoller. Her gis det muligheter for protokoll-multipleksing, segmentering og reassemblering. L2CAP ligger plasseringsmessig over Baseband Protocol og finnes i datalink-laget.

Som nevnt tidligere gir Bluetooth støtte til to linktyper, SCO og ACL. L2CAP derimot er definert kun for ACL og det finnes ingen påtenkte løsninger til SCO [2].

L2CAP skaper kontakt med andre kommunikasjonsprotokoller som f.eks Bluetooth Service Discovery Protocol (SDP) og RFCOMM.



Figur 5 : L2CAP i protokoll-arkitekturen [2]

2.6.3 Link Manager Protocol

Link manager protocol (LMP) er benyttet for link-setup, sikkerhet og kontroll [3]. Dette skjer ved hjelp av såkalte LMP-meldinger. Disse meldingene har en høyere prioritet enn brukerdata, som betyr at hvis Link Manager trenger å sende en melding så skal ikke dette blir forsinket av L2CAP-trafikk. Meldingene blir overført i payload isteden for L2CAP og kan skilles fra en bestemt verdi i payload-headeren.

2.6.4 RFCOMM

RFCOMM-protokollen gir seriell-port egenskaper over L2CAP, og er basert på ETSI¹ standard TS 07.10 [4]. RFCOMM er en enkel transport protokoll med tilleggsdata for å gi tilpasning til DB9-RS232 seriellporter.

RFCOMM støtter opptil 60 koblinger samtidig mellom to enheter. Antallet koblinger som kan bli benyttet samtidig i en enhet er implementasjons-spesifikk.

2.6.5 Service Discovery Protocol

Service Discovery Protocol (SDP) gjør det mulig for applikasjoner å oppdage hvilke tjenester som er tilgjengelige og bestemme egenskapene til disse tjenestene [5]. I et Bluetooth miljø endrer samlingen av tjenester som er tilgjengelige dynamisk, noe som gjør dette ulik andre tradisjonelle nettverksbaserte miljøer. SDP er påtenkt å adressere de unike egenskapene i et slikt Bluetooth-miljø.

¹ ETSI (the European Telecommunications Standards Institute) er en ikke-profitt organisasjon der ders mål er å lage telekommunikasjon-standarder.

3 Teoretisk løsningsforslag (Metode)

3.1 Innledning

Innledningsvis ble det fortalt at kjernedelen av oppgaven var å gi løsninger til en overføringsprosess. Denne prosessen er bygget opp med Bluetooth som ryggrad, og belyser denne teknologien i tråd med EKG-bruk. Løsningen på denne overføringsprosess kan kalles metode. Denne metoden skulle gjenspeile en situasjon av EKG-data fra en pasient til visuell representasjon. Utgangspunktet for modellen er en hjemmesituasjon der overføring skal skje fra pasient til hjemme-PC, hvor disse lagres i database. Mer fortalt vil dette omhandle alt fra digitalisering av det analoge EKG-signalet, til et mottak der lagring og monitorering skal foregå. I tillegg skal det være tilrettelagt for videre bruk av disse oversendte dataene, da spesielt videresending over WAP.

Metoden er kun en modell, det vil si at det er en systemoversikt. Løsningsproblematikken besto mer i av å fremstille en teknologisk plattform, som gir utrykk for muligheter som ligger til rette, rent teknisk, med Bluetooth som en aktivt medium i et EKG-miljø.

Metoden er i stor grad basert på teori, og består av en del forutbestemte elementer, som blant annet elektroder og EKG-forsterker. Dette var verktøy som måtte benyttes for å få frem signaler som var verdt å jobbe med. Med andre ord var ikke denne delen en del av selve løsningsrettede problematikken.

Utførelsen av overføringenprosessen ble foretatt ved hjelp av EBDK (se neste kapittel).

3.2 Generelt om Ericsson Bluetooth Development Kit

Ericsson har utviklet, i samarbeid med Symbiotics [26], et arbeidsverktøy - Ericsson Bluetooth Development Kit (EBDK). Dette verktøyet gjør det mulig for interesserte å jobbe med teknologien. EBDK består av flere deler som i sin helhet gir støtte for utvikling av flere applikasjoner.

Verktøyet gjør det lettere for ingeniører å integrere den nye åpne standarden med en rekke av digitale enheter for stor produksjon. Dets arkitektur demonstrerer kjernen av Bluetooth-teknologien, derfor tillater den utviklere til å lage integrerte applikasjoner til prototyper, altså tidsbesparende og man får tid til å konsentrere om andre aspekter av utviklingen. En mengde grensesnitt gjør at man raskt kan utvikle ferdige applikasjoner.

Dette utviklingsverktøyet ble installert på IKT-labben, og besto hovedsakelig av to kommuniserende enheter, hver del identisk som vist på figur 3.1. Disse delene ble koblet til hver sin PC, gjennom RS232-tilkobling. Ved hjelp av tilhørende software lot det seg gjennomføre å manøvrere disse enhetene. [27, 49, 50]



Figur 6 : Ericsson Bluetooth Development Kit. Figuren viser en av totalt to like deler, som igjen kunne kommunisere med hverandre etter installasjon [27]

Innebygget i verktøyet er Bluetooth-modulen ROK 101007/1.

3.2.1 Tilkobling og overføringshastighet

De to aktuelle metodene for å koble seg til dette verktøyet, USB² eller RS232³.

Den planlagte tilkoplingen til verktøyet skulle i utgangspunktet bli fortatt ved hjelp av USB, da særlig på grunn av den bærbare PCen (se kapittel 3.3.2). Denne løsningen ville avlastet den ene PCen på IKT-senteret. Å utføre den relle planløsningen (se kapittel 3.1) viste seg tidlig å bli ekstra mye merarbeid, og derfor ble den praktiske utførelsen foretatt ev gjeldende utstyr på IKT-senteret. PCene på IKT-senteret hadde en RS232-tilkobling til EBDK, og derfor også den valgte tilkoblingen [49, 50].

3.2.2 Overføringshastighet i RS232

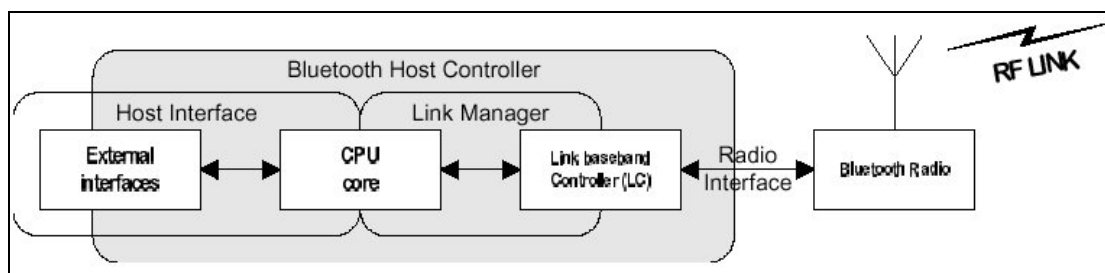
Båndbredden i overføringen til sende-modulen i EBDK bestemmes av tilkoblingen, og er bestemt av produsent.[50]. Denne overføringshastigheten (baud rate) var satt til å være 57600kbps.

3.2.3 Kontakt med Bluetooth-hardware – HCI kommandoer

For å få kontroll med Bluetooth-hardware i en enhet må man få kontakt med Host Controller i denne enheten. En Host Controller er fellesbetegnelse på hardware i teknologien. Et grensesnitt til en Host Controller kalles HCI (Host Controller Interface), og er en uniform metode for å aksessere Bluetooth's hardware.

² USB (Universal Serial Bus) er en standard for tilkobling av periferiutstyr. Med USB kan en enhet bli lagt til PC'en "plug & play", det vil si svært enkelt. De fleste nye PC'er og utstyr kommer i dag med USB. En er avhengig å ha et OS som støtter USB for å få til en slik overføring [28]. Maksimum overføringshastighet er 12 Mbps.

³ RS232 er en asynkron seriell overføringsmetode, og er en av de eldste kommunikasjonsstandardene i computer-teknologien. Denne måten å overføre på er robust, og ble i sin tid utviklet for tilkobling av periferiutstyr. Originalt var overføringshastigheten på maksimum 20 kbps, men det har vist seg i praksis at større båndbredde er mulig [29].



Figur 7 : Bluetooth Host Controller er selve fellesbetegnelsen for hardware [6]

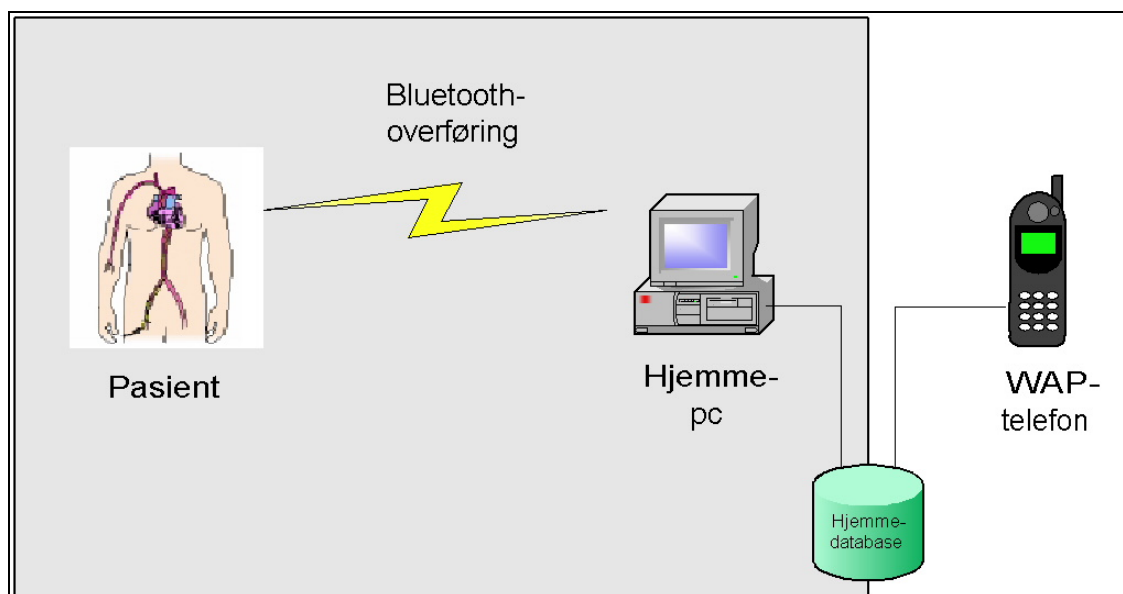
For å få akesseringen benyttes spesielle kommandoer, kalt HCI-kommandoer. Disse kommandoene gir muligheten for å kontrollere linklag-forbindelser [6].

I vedlegg 5 finnes en total oversikt over slike HCI-kommandoer.

3.3 Planoversikt

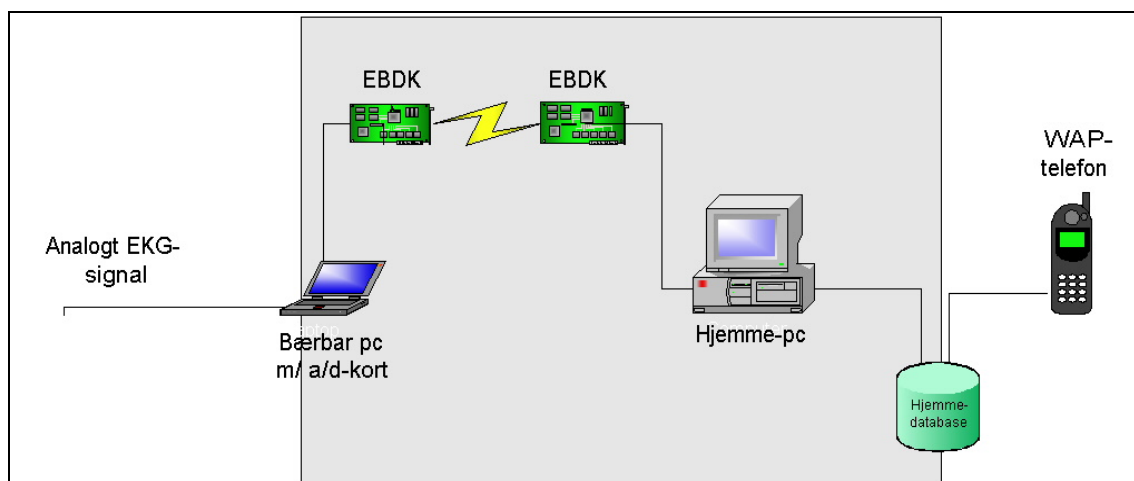
3.3.1 Innledning

En ideell planløsning vil være å ha en liten Bluetooth-sender ”plantet” på en pasient. Herfra vil da EKG-dataene gå direkte over til en hjemme-PC. Denne PCen vil ha en database for lagring av innkomne EKG-informasjon. Fra denne databsen kan informasjonen sendes videre med annen teknologi, til en sentral database, f. eks ved et sykehus. Denne sentrale databasen vil senere operere som referanse til pasienten.



Figur 8 : Den ideelle planløsningen

På grunn av tilgjengelig utstyr på HiA, må en slik tenkt planløsningen endres i praksis. Utgangspunktet er fortsatt den samme, men for å gjøre dette praktisk mulig, men det tilgjengelige utstyret, må tilpasninger til for å få ting til å fungere i praksis. En reell planløsning måtte skapes. Figuren nedenunder beskriver denne reelle planløsningen.



Figur 9 : Den reelle planløsningen som muliggjorde EKG-prosessoverføring

Oppgaven omhandlet ikke databehandling, noe som ga utslag på den praktiske gjennomføring av overføringen. Denne overføringen ble i praksis overført mellom to maskiner på IKT-sentert, der hver sin maskin hadde montert hvert sitt Bluetooth-overføringsmedium (les: EBDK-enhet). En systemoversikt over denne overføringen finnes i kapittel 7.1. Derfor er det i rapporten fokusert mest på hva som finnes i Bluetooth, som er EKG-bærer. Videre bære rapporten preg av fokusering på selve overføringen og ikke periferi-behandlingen av data. I tillegg er det blitt fokusert på interoperabiliteten mot WAP.

3.3.2 Bruk av bærbar PC

Som nevnt i kapittel 3.3.1 skulle den opprinnelige metodeløsningen inkludere en bærbar PC. Derfor ble det installert et a/d-kort⁴ på en bærbar PC, til det formål å gi en digital representasjon av EKG-dataene. Hensikten var å koble denne bærbare PCen til en av de to EBDK-enhetene ved hjelp av USB. På grunn av utviklingen av prosjektet, ble altså bruken av et slik bærbar PC utelatt.

Å behandle datasignaler fra hardware krever software-applikasjoner. Universal Library, fra ComputerBoards, er et slikt software som gir komplett tilgang til kortets funksjoner. Ved hjelp av dette programmet var det mulig å programmere egne programmer, som f.eks. et samplingsprogram. (se kapittel 3.5.2)

3.3.3 Monitor

Sentralt i kontrolleringen av databehandlingen var å gi en løsning på et eget EKG-program som skulle gjengi EKG-prosessdataene. Dette programmet skulle ikke bare være et bruksmiddel i databehandlingen, men også som en del i modellen. En kan kalle programmet for modellens vindu utad. Dette programmet skulle benyttes både

⁴ Det har i oppgaven tatt utgangspunkt i et PCMCIA-kort fra ComputerBoards (PCM-DAS08). Dette kortet er et 25kHz 12 bits- kort (angir 2^{12} mulige nivåer), og med få innganger og utganger. EKG-overføring stiller ikke store krav til båndbredde, ei heller til overføringsrate, derfor et relativt enkelt kort. Ved overføring av kun EKG, trengs en kanal; en inngang og en digital utgang. Med tanke på utvidelser av tjenestene, det vil si overføringer av ytterligere tjenester enn bare EKG-dataene, gir kortet også støtte til dette

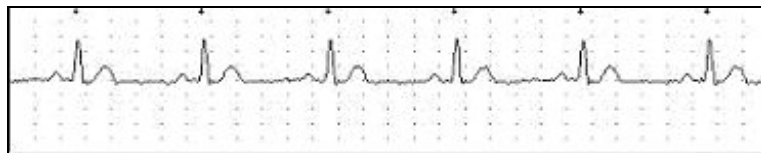
av den bærbare PCen og av PCen i mottaket. Dette for å gi et sammenligningsgrunnlag.

Programmets funksjonalitet

Programmets funksjonalitet skulle ikke være mer særegent enn vanlig EKG-monitorering. Faktisk skulle dette være et helt enkelt program som skulle formidle de digitale representasjonene på en synlig måte.

Bruksmateriale til dette programmet blir filer eller sampler. Før bruk må alle ha et punkt-representasjon, det vil si at filsekvens må reassembleres før bruk. Punktene, som refereres til, er den verdien av den digitale veriden av et sampel. Disse verdiene må representeres som et punkt i et diagram. Dette diagrammet ville lage "rammen" for den synlige delen av programmet. I denne rammen skal punktene gradvis forskyves til venstre når nye punkter av etterfølgende sampler ble lest inn, kalt panning, som er den vanlige måten å monitorere EKG.

Den totale levetiden på et sampel i rammen kan være varierende, men ca. 10 sekunder vil være tilstrekkelig for å få en oversikt over den siste tilstanden til en pasient. På figur 9 vises hvordan slike EKG vises. (NB! Dette er bare et tilfeldig bilde.)



Figur 10 : Et "normalt" EKG-signal [32]

For å klare å lese verdier fra en fil, kan dette gjøres ved å lese fra filen til en array. Ved 12 bit sampelstørrelse, må hvert array være av typen integer. Hvert array måtte være så stort som man har punkter langs x-aksen i diagrammet.

Til konstruksjonen av diagrammet kan putpixel (x, y, color)-funksjonen benyttes. For å få til en panning av diagrammet mot venstre trenger man bare å kopiere av et array-element til det foregående [51].

Alternativt til å konstruere eget program er å benytte allerede ferdige fremstilte verktøy, som MediTronGUI og CoolEdit.

MediTronGUI er Meditrons analytiske software, som tar bruk av lyd. Dette verktøyet har software for opptak og registrering av lyd og visning av fonocardiogram [30].

Cool Edit er en Wav-editor for Windows [31].

3.3.4 Database

Informasjon om denne finnes i kapittel 3.9.2.

3.4 Datarepresentasjon

3.4.1 Båndbredde

Den høyeste frekvenskomponenten i et analogt EKG-signal er på rundt 150Hz [32].

3.4.2 Sampling

Et analogt EKG-signal må digitaliseres for bruk i et Bluetooth-miljø. Den første fasen i den retning er bruk av a/d-kort. Å digitalisere data ved hjelp av a/d-kort medfører

ofte for høy signalrate, og krever unødvendig stor prosesseringshastighet for systemer. Ved bruk av sampling får man foretatt representative ”stikkprøver” av signalet. Et EKG-signal, som bare har en båndbredde på 150Hz, vil få en god representasjon med en samplingfrekvens på 300Hz, i følge Nyquist-teoremet [45]. Hvis man sampler med 300Hz eller mer vil en få en god digital representasjon av et EKG-signal.

3.4.3 Digital representasjon av EKG

Det allerede ferdig-installerte EBDK-utstyret, med tilhørende stasjonære PCer, ble benyttet i den praktiske gjennomføringen av prosessoverføringen av EKG-dataene. Siden det digitale representasjonen som skulle foreligge ved bruk av bærbar PC, nå ikke lot seg gjennomføre, ble det benyttet tidligere lagrede filer som utgangspunkt. Disse filene var av typen .wav og mp3 (3.6.1 og 3.6.2), og ga en god digital representasjon av EKG-data. Filene inneholdt et 5-sekunders opptak av normale hjerteslag, og ga en god representasjon av EKG-prosessdata.

3.4.4 Sanntidsløsning

Det er ikke tatt hensyn til å se på problematikken rundt sanntidsoverføring. I en EKG-overføring er alt annet enn en sannhetsoverføring i praksis ikke godt nok.

3.4.5 Audio-overføring

Det er lagt vekt på en overføring med audio-filer, samt overføring av sampler. Dette har tilknytning til oppdragsgivers ønske. Se kapittel 3.6 for ytterligere informasjon.

3.5 Overføring av EKG i Bluetooth

3.5.1 Generelt

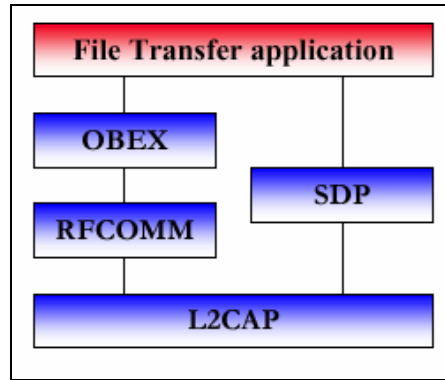
En ideell EKG-overføring krever vil være sømløs og foregå i sanntid. Derfor krever visse betingelser som bærer av slik tjeneste.

3.5.1 Protokollstakk for overføring av filer

Da protokoll-stakken kan bygges opp på forskjellige måter, alt avhengig av hvilken bruk, er det dannet forskjellige bruksmodeller [47]. Hver og en av disse modellene er forbundet med en profil, som nevnt tidligere definerer protokoller og deres egenskaper, nettopp for å støtte opp om den aktuelle profilen.

Ved overføringen av EKG-signalet, uansett vil det aktuelle overførings-objektet være en fil eller et samplel. Slik filoverføring har sin egen bruksmodell [47]. En slik modell gir støtte for overføringer til et ubegrenset antall filtyper.

Figur 10 nedenfor viser den nødvendige protokoll-stakken som presenterer den aktuelle bruksmodellen. Figuren viser ingen LMP, Baseband eller Radiolag, men dette er obligatorisk og blir benyttet som underliggende lag.



Figur 11 : Bruksmodell ved overføring av filer/sampler [47]

OBEX

IrOBEX [IrDA Interoperability], eller forkortet OBEX, er et sesjonslags-protokoll. Den er definert av IrDA (Infrared Data Association) i den hensikt å utveksle objekter på en enkel måte.

Denne protokollen har blitt adoptert til bruk i Bluetooth-teknologien, noe som gjør det mulig for enkelte applikasjoner å benytte enten radio eller IR.

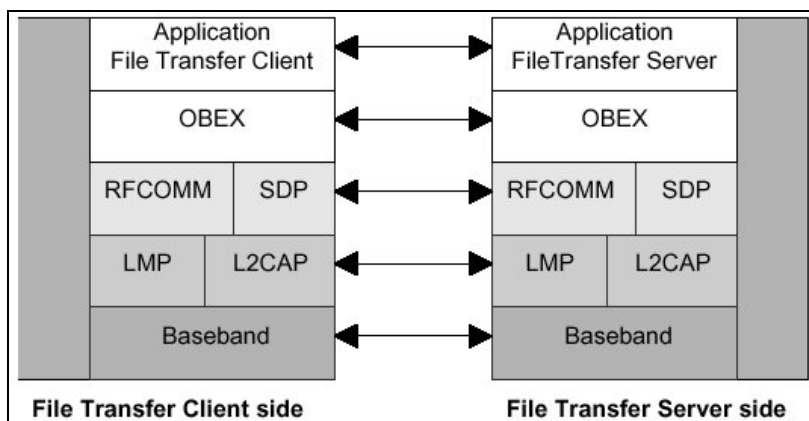
Selv om både IrDA og Bluetooth er laget for trådløs kommunikasjon ved korte distanser, har de noen fundamentale forskjeller når det gjelder lavere lags protokoller. Opprinnelig ble OBEX utviklet for å utveksle dataobjekter over en infrarød link, men kan opptre ovenpå andre transportlag, som Bluetooth's RFCOMM. OBEX er derfor blitt tilrettelagt plasseringsmessig over de lavere lagene som Bluetooth benytter.

En ser for seg bruken av OBEX over TCP/IP som en fremtidig trekk for Bluetooth-enheter [47].

3.5.2 Push-teknologi i Bluetooth

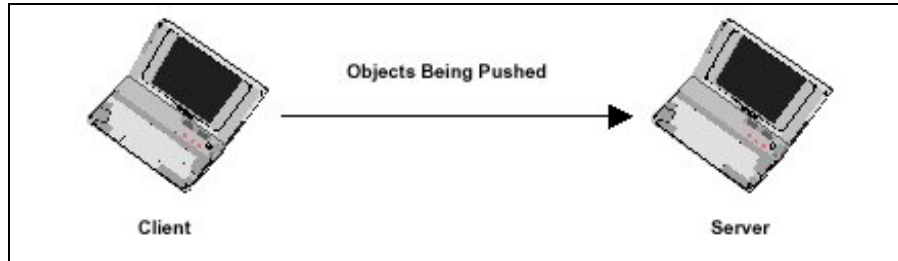
Som teknologi for overføring av EKG-data er det nødvendig å ha en sømløs overføring. Denne overføringen skal skje med automatikk og ikke ved forespørsler. Bluetooth har spesifisert push-teknologi for filer [8], noe som gir automatikk i overføringsprosessen.

I forrige kapittel ble protokoll-stakken for filoverføring beskrevet. I en filoverførings-sekvens er også noe som heter Client og Server. Interaksjonen mellom disse, samt deres protokoller, gir noe som kalles en protokoll-modell. Denne modellen vises i figur 11.



Figur 12 : Protokoll-modell for push [8]

Slik som denne overføringen skal foregå, vil det dannes et rollemønster mellom Client og Server. Client er den enheten som initierer operasjonen, altså sender-delen. Denne pusher objekter til Server.



Figur 13 : Push-teknologi danner rollemønster; Client og Server [9]

Det er OBEX som gir støtte for slik pushing [7]. En push-overføring foregår ved operasjoner i protokollen. For push benyttes operasjonen *Put*. En slik operasjon kan bare ta maksimum en fil (noen ganger trengs flere put-operasjoner for en fil).

Etablering av en OBEX-session finnes spesifikasjonen [7, 9, 10].

3.6 Beskrivelse av filformater for overføring av data

3.6.1 Wav

Filformatet wav (Windows WAVE) ble i sin tid utviklet av Microsoft og IBM [33], og er i virkeligheten en standard i Windows og DOS [34]. Wav er av ekstrem høy kvalitet [34] og som konsekvens av dette skapes forholdsvis store filer.

Wav er en del av filformatet RIFF, som håndterer mange typer data, men som i utgangspunktet er beregnet for multimediafiler. En RIFF-fil preger seg ved at den inneholder to deler; den ene forklarer selve filen (formatdel), mens den andre inneholder sampler fra den digitaliserte lyden (datadel). Formatdel beskriver filen, og her ligger blant annet informasjon om hvilken komprimeringsteknikk som er benyttet. Da wav er en typisk pulskode-modulert ⁵(PCM) signal, som ofte er blitt generert fra et lydkort, finnes denne informasjonen i formatdelen.

```
12      4 bytes  'fmt '
16      4 bytes  0x00000010      // Length of the fmt data (16 bytes)
20      2 bytes  0x0001          // Format tag: 1 = PCM
22      2 bytes  <channels>      // Channels: 1 = mono, 2 = stereo
24      4 bytes  <sample rate>   // Samples per second: e.g., 44100
28      4 bytes  <bytes/second>  // sample rate * block align
32      2 bytes  <block align>   // channels * bits/sample / 8
34      2 bytes  <bits/sample>  // 8 or 16
```

Figur 14 : Formatdelen i RIFF. I linje 3 angis den aktuelle komprimeringen (PCM), linje 4 angir om filen er i mono eller stereo, og linje 5 angir samplingfrekvensen [57]

⁵ Pulse Code Modulation. En vanlig metode som brukes til å konvertere analoge signaler til digitale.

| | | |
|----|---------|----------------------------|
| 36 | 4 bytes | 'data' |
| 40 | 4 bytes | <length of the data block> |
| 44 | bytes | <sample data> |

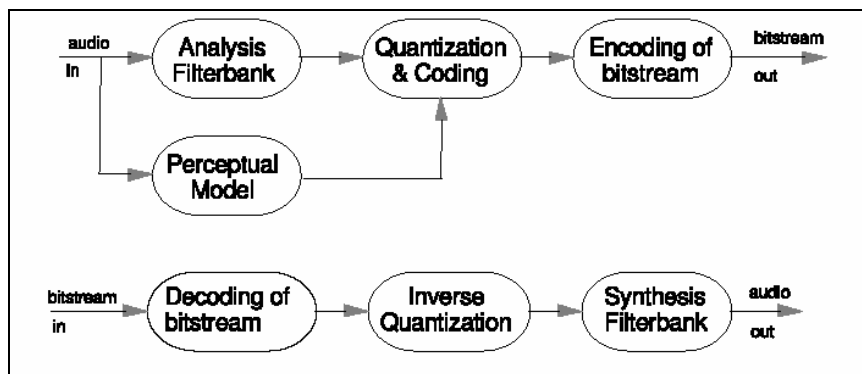
Figur 15 : Innledningen på datadelen i RIFF. Samplene vil bli listet opp etter denne innledningen [57]

Som det ble nevnt innledningsvis i dette kapittelet, vil wav-filene bli relativt store. Hvor store filene blir har sterk sammenheng med samplingfrekvensen. En hurtigere frekvens gir flere samplere per tidsenhet, noe som gir bedre representasjon av signalet. Figuren viser en frekvens på 44.1kHz, en frekvens som ifølge Nyquist er tilstrekkelig kvalitet for menneskets hørbare område (20- 20000Hz). En frekvens på 44.1kHz gir det vi ofte kaller ”CD-kvalitet”.

3.6.2 Mp3

Mp3-standarden ble i sin tid utviklet av MPEG (Moving Picture Expert Group), en arbeidsgruppe nedsatt av ISO. Denne arbeidsgruppen har hatt, og har, ansvaret for utviklingen av internasjonale standarder for blant annet komprimering av lyd. Det finnes foreløpig tre standarder, MPEG-1, MPEG-2 og MPEG-3. Mp3 er en kortform for det såkalte lag 3 av MPEG-1-standarden. Tre slike lag er definert, og alle med sine spesielle fordeler. Høyere lag, jo større kompleksitet [58].

Teknikken som brukes som komprimering i MP3 kalles ”perceptual encoding” [35]. Kort fortalt er dette komprimering på frekvenser som den menneskelige hjerne ikke registrerer. Med andre ord vil dette si at ikke-hørbare frekvenser fjernes, noe som utgjør størsteparten av komprimeringen. Når man da står igjen med det lyttbare frekvensområdet, blir det kun tatt hensyn til påfølgende ulike digitale verdier, med andre ord så kodes kun forholdet mellom to ulike verdier. Komprimeringsfaktoren er avhengig av hvilket frekvensområde lyden ”befinner” seg på, derfor er det ingen fast komprimeringsforhold [36].



Figur 16 : Perceptual encoding [36]

Ved bruk av komprimeringsalgoritmen for MP3, er det mulig å komprimere en wav-fil til nettopp en MP3-fil. Tilgjengelig utstyr for en slik prosess finnes enkelt ved søk i Internett, som ferdige software-program. Flere av disse programmen har også muligheter til å reversere prosessen, fra en MP3 til wav.

MPEG-standarden gir ingen garantier for kvalitet av komprimert lyd [36]. MP3 kan lage feil og mangler i gjengivelsen dersom det blir brukt for liten bitrate i kodingen og/eller når det er feil på parameterne. Feilene kan være:

- Forvrengning, men ikke harmoniske forvrengninger
- Støy, men bare ved bestemte frekvensområder
- ”Hard lyd”, men feilen endrer karakteristikk hvert 20ms

I kapittel 3.6.4 er det utregnet et forhold på komprimeringen fra wav til MP3 på to EKG-filer. Disse filene er lagrede 5-sekunders sekvenser av hjerterytme.

3.6.3 Sampler

Alternativt til å sende sekvensielle filer er overføring av hvert sampel. Disse samplene er representanter for hvert sampel (se kapittel 3.4.2) fra det analoge signalet. Det mest aktuelle overføringsmetodikken i så måte, kan være å benytte seg direkte av Bluetooth Audio-overføring. Dette er tale-overføring, som gir enten en 64kbps PCM overføring eller en 64 kbps CVSD (Continous Variable Slope Delta Modulation) overføring. Denne overføringshastigheten vil dekke behovet for Nyquist samplingsteorem på et EKG-signal. En slik tale-overføring vil ikke gå på ACL-ling, men SCO-linker [1, 47]. Ved bruk av talekanalen, vil en også få dekket sanntidskrav.

Sampler er råde bit å regne, det vil si at det finnes minimal eller ingen ekstrainformasjon. Ligger det mulighet for bruk av linjekoding, finnes mulighet for å spare båndbredde også her. Ved hjelp av linjekoding, som tar utgangspunkt i endringene i bitfølgetongen. I slikt bruk må det finnes en linje-dekoder ved mottak. Eksempler på slik linjekoding er Manchester og NRZI (No-Return-to-Zero_Inverted) [59]

Baseband i Bluetooth benytter Little-endian-prinsippet (se kapittel 2.5), det vil si at den først ankomne bitet fra høyere lag vil være den som blir overført først. På en slik måte vil man ikke rokkere rekkefølgen på ankomne bit, og derfor være mindre utsatt for hvor en linjekoder skal plasseres.

Alternativt til å sende over tale-kanalen, er å sekvensiere samplene i filer. Dette bringer oss tilbake til overføring av filer, og filformater.

3.6.4 Komprimering av lydfiler

Tidligere eksperiment hadde gitt to wav-filer. Disse inneholdt EKG-simulering, og besto av ca. 5 sekunders opptak, og ga et godt representativt materiale for et EKG-signal. Disse filene var henholdsvis 436 og 445 Kbyte i størrelse.

På en Bluetooth-link er båndbredden. eksklusiv header, 721 kbps. Disse filene krevde følgende båndbredde :

$$436 \times 1024 \times 8 = 3571712 \text{ bits}$$

$$445 \times 1024 \times 8 = 3645440 \text{ bits}$$

Nå er dette størrelsen på en 5-sekunders sekvens. Ved å dividere på 5 vil vi få hvor mye bps som kreves, og dermed også båndbredde :

$$3571712 \text{ bits} / 5 \text{ sekunder} = 714\text{kbps}$$

$$3645440 \text{ bits} / 5 \text{ sekunder} = 729\text{kbps}$$

Vi ser at dette går ikke. Selv om verdien ligger like under maksimale båndbredden på linken, så må vi huske på at dette er den teoretisk maksimale og ikke den reelle. Men med enkle komprimeringsteknikker er det mulig å få frem de samme verdiene uten at essensen i signalet går tapt. Med komprimeringsteknikken mpr [se kap x] kan Wav-filer enkelt gjengis i en langt mindre format. Det finnes utallige gratis-versjoner av komprimeringsprogrammer på Internett. Her ble MP3-Compressor benyttet. Resultatet ble følgende :

436KByte → 78.5KByte (et forhold på 5.5:1)

445KByte → 81.4KByte (et forhold på 5.5:1)

Forholdet på 5.5:1 er langt dårligere enn for vanlige musikkfiler, som normalt ligger i størrelsesorden 12:1 [34]. Dette har sin årsak i at et EKG-signal ikke har frekvenskomponenter utover det hørbare området som nettopp blir fjernet med MP3. (se kapittel 3.6.2)

3.7 Dynamisk båndbredde

Båndbredden i en link i Bluetooth er 1Mbps, eller 721Kbps eksklusiv headere, synkronisering etc (se kapittel 2.3.1). Denne båndbredden er en maksimumrate, det vil si at det er den høyest teknisk oppnåelig. En slik rate er i teorien holdt ved like i et enkelt piconett, og blir først lavere ved scatternett-dannelser.(se kapittel 4.10). EBDK består av to moduler, noe som danner et piconett. Overføringen har da en teoretisk maksimum på 721Kbps i nytte-data, en verdi som blir teoretisk konstant da det ikke finnes andre Bluetooth-enheter i den nærmeste omkrets.

Generelt sett vil ikke båndbredden i et piconett reduseres dersom det ikke inkluderes i et scatternett. Ved dannelse av et scatternett har de enhetene som opererer på to eller flere piconett en tidsdelt aksess (se kapittel 3.8.1). Dette gir reduksjoner på overføring per piconett (se kapittel 4.3). Dette er verdt å merke seg i dimensjonering og konstruksjon av tilkoblinger.

3.8 Generelt ved sending og mottak av data i Bluetooth

3.8.1 Tidsdelt overføring

Overføringen i Bluetooth er tidsdelt-duplex (TDD) [1]. Dette betyr at det vekselvis overfører og mottar, der overføringen skjer i intervaller. Hvilken modus en enhet befinner seg i bestemmer overføringen.

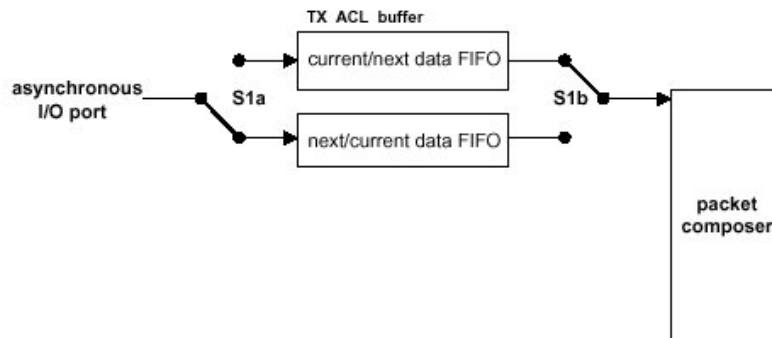
3.8.2 Synkronisering

Det er klokke i master som synkroniserer et piconett. Hvert intervall er alltid på 625µs (tiden mellom hvert frekvenshopp), og dette endrer seg aldri mens et piconett eksisterer (se kapittel) [1].

3.8.3 Buffre ved sending

Siden master styrer hver slave, finnes det her buffer. Det vil si i en ACL-forbindelse finnes det ett buffer på hver tilknyttede slave. Hvert buffer inneholder to FIFO (First-In-First-Out) registre. Det ene er et "current"-register, som kan aksesseres og leses av Link Controller for å lage pakker før overføring kan skje. Det andre registret er et "next"-register, som kan aksesseres av Link Manager for å fylle opp ny informasjon.

For å fastsette hvilke registre som er hva, finnes det svitsjer som styres av Link Controller. Det finnes slike svitsjer på både input og putput, og de kan aldri være koblet til den samme registret samtidig [1].



Figur 17 : Buffer i sender. Vi ser at den inneholder to registre som aksesseres ved svitsjer [1]

3.8.4 Buffre i mottak

I mottakk finnes også buffer [1]. I motsetning til sending må bufferet deles av hver slave. Dette bufferet inneholder også to FIFO registre. Det ene registret kan aksesseres og fylles av Link Controller med nytte-data fra den siste mottatte pakken, mens det andre registret kan aksesseres av Link Manager, som leser forrige nytte-data.

3.8.5 Flytkontroll

Siden ACL-bufferet kan være fullt når nye nytte-data fra neste pakke ankommer, er det nødvendig med flytkontroll [1]. Dette løses ved å sende en returmelding, der "flow"-biten i header er merket med "stop" eller "go". En slik melding blir alltid sendt tilbake når det mottas nye pakker.

3.9 Til videre bruk

Nå vil man gjerne se løsninger i større perspektiv. Mye tyder på at Bluetooth ble ikke utviklet for å være et trådløst medium bare for å tilby en plattform for høyere lags protokoller og applikasjoner. Dette bevises gjennom flere spesifikasjoner på interoperabilitet med annen teknologi i spesifikasjonen. Målet har alltid vært å forsyne mer, der umiddelbar interoptabilitet skulle være mulig når de første produktene kommer. De virkelige interessante applikasjonene vil være de som går hinsides dette.

3.9.1 Interoptibilitet med WAP

I løsningen ble det lagt vekt på interoptabiliteten med WAP, og WAP som videreformidler. Hvorfor skal vi gjøre dette? Et motiv for dette er være å løsrive seg fra fastnett, noe som er svært aktuelt i de tilfellene der pasienten befinner seg hjemme. I et slikt scenario er det aktuelt å sende over data, med personalopplysninger, til en felles-database, evt et sykehus, som en alarm. Denne basen vil da ha registrert hendelsene til en pasient, som gjør at forespørsler skjer opp mot denne databasen. Dermed vil behovet for toveis kommunikasjon til hjemme-databasen være uaktuelt.

En får ikke mulighet til sømløs og problemfri overføring til WAP sånn uten videre. På grunn av begrensninger i overføringshastighet, dvs prosessering og båndbredde, og begrenset display, er det nødvendig med tilpasninger. Vi valgte å legge en løsning med et klart separat skille – en database.

3.9.2 Bruk av database og SQL

Det er nytte­data som er EKG-infor­masjonen. Her finnes også infor­masjonen om eventuelle filtyper. Etter at Link Manager har lest siste nytte­data, er neste korstog å lagre disse filene. Slik lagring foregår meget strukturert i en database. I en tenkt hjem­mesituasjon (neste kapittel) vil dette være den databasen der alle filer/sampler legges inn.

Å legge inn data i en database kan SQL benyttes. Structured Query Language (SQL) er en standardisert interaktiv- og programmerings-språk for å hente informasjon fra, og for oppdatering av, en database. Selv om SQL er både en ANSI (American National Standards Institute) og en ISO (International Standards Organization), støtter mange database-produkter SQL, men har ofte egne utvidelser til standarden.

Forespørsler foregår ved forskjellige kommandoer, som blant annet select, insert, update, find etc. [37].

3.9.3 Hjemme-database

Bruk av en database i hjemmet er for å kunne koordinere filer/sampler videre i sammenheng med personalia. Derfor er en slik løsning kun egnet for en person. Ved bruk av database, er det aktuelt å benytte to tabeller, en for sampler/filer og en for personalia. I sample/fil-tabellen legges hver sample/fil inn med SQL-kommandoer. (se kapittel 3.9.2). Denne tabellen skal være koblet opp mot personalia-tabellen. Forbindelsen mellom disse tabellene må være one-to-many, det vil si at det må gis mulighet for en pasient å ha flere filer/sampler registrert i sitt navn. Den direkte koblingen mellom tabellene får en til ved å ha en felles nøkkel. I dette tilfellet er Pasient_ID det mest naturlige å ha som kobling, som da vil befinne seg i begge tabellene, og som også vil finnes som primær­nøkkel i personalia-tabellen. I sample/fil-tabellen må det finnes en egen primær­nøkkel, Fil_ID, og denne må settes til autonummer. Den tredje attributten i denne tabellen er Filnavn, som gjør identifisering enklere. Navnene skal gis i forhold til når andre filer/sampler, som tilhører samme pasient, ble lagret og hvilken filtype som er lagret. Et eksempel på et slikt filnavn kan være MP005, noe som angir at det er mp3 og at det er fil nummer 5 i overførings­sesjonen.

For å kunne aksessere databasene via WAP brukes ASP. ASP bygges inn i WML. ASP er et program beregnet for dynamiske websider. Inne i ASP-delen av programmet brukes SQL. SQL brukes blant annet for å hente ut og legge inn data i en database. Det er viktig at data i en database blir lagt og hentet fra riktig sted og dette er en av hovedfunksjonene til SQL som er et standard språk som fungerer for de fleste databaser [56].

3.9.4 Sampler eller filer?

I en sanntids­løsning, som baserer seg på en annen overføring enn den foretatte reserve-løsningen med sekvensiell audio-fil, er løsningen å sende hvert sampel. Et mulig hinder i å gjennomføre dette er den lave overføringsraten som i dag kan sendes over med WAP (9600 kbps). Hvor stor overføringsrate bestemmes i stor grad av hvor god oppløsning EKG-signalet har. Har man et 12-bit a/d-kort vil en få en overførings­hastighet på 4800kbps ved sampling på 400Hz. Husk at dette er kun nytte­data. I tillegg kommer bl.a. signalering, headere osv. En ser at en dårligere oppløsning enn 12 bit, ikke er særlig smart. Har man mulighet til å benytte linjekoding, som f.eks. Manchester eller NRZI, vil man ha mulighet til å spare noe (se kapittel 3.6.3)

Alternativet til å overføre samplene, er å gjøre benytte lagrede sekvenser. Dette blir på den samme måte som benyttede filene i realiseringen av prosessedata-overføringen. I følge tidligere beregninger i kapittel 3.6.4 vil ikke slike filer la seg gjennomføres som filformat. Ytterligere komprimering må finne sted. To ting som står sentralt for filvalg, og det er hvilken type som gjengir signalet best, og hvor mye er det på å spare på komprimeringsforskjellene.

Et resultat av sekvensoverføring, er at overføringsmetodikken blir mindre dynamisk. I et slikt tilfelle vil ikke hvert sampel være en enhet. Hver overføringsenhet blir nå flere sampler. Dette vil gi en såkalt "stillbilde"-løsning, hvis det formodningen ikke skulle eksistere en løsning i mottaker som "reassemblerer" disse sekvensene.

3.10 Alternativ overføring

3.10.1 TCP/IP og streaming audio

Spesifikasjonen gir støtte for bruk av protokollen OBEX over TCP/IP [7]. Dette kan også sees i protokoll-stakken på figur 4. Dette muliggjør overføring på Internett. En metode benyttet mye på World Wide Web (WWW) som tilfredsstillt sanntid er streaming-teknologi. Ved streaming sendes det over kontinuerlig strøm av informasjon, i egne formater. Et format for slik overføring er M3U. Ved streaming foregår det ingen lagring i mottaker.

Streaming-teknologi foregår på den måten at data sendes i sanntid. Et buffer blir benyttet for å korrigere overføringshastigheten med den raten filen bruker på å kjøres. Dette bufferet har first in – first out (FIFO) karakteristikk. Slik teknologi blir ofte benyttet i Internett for overføring av både audio og video, og gir muligheten for å høre og se på filer uten å måtte laste ned filene på forhånd. Streaming tar ikke bruk av mellomlagring, ei heller lagring i mottak [34].

4 Mobilitetsproblematikk i Bluetooth

4.1 Innledning

I et sykehusmiljø er det særs viktig å ha sikker teknologi. Det kan få svært alvorlige følger hvis ikke teknologien klarer å bringe kontinuerlig informasjonsstrøm. Som vi allerede vet er Bluetooth en teknologi som opererer på korte avstander. I dette tilfellet kommunikasjonen mellom enheter på avstander mindre enn 10 meter. Som nevnt skal ikke Bluetooth kun være til nytte som en teknologi for reduksjon av ledninger og kontakter, men også som langt større applikasjonsområder. Derfor er det interessant hva som ligger i dagens teknologi for behandling av enheter, det være seg pasienter, som befinner seg i et mobilt miljø.

Før en går direkte inn i et sykehusstilfelle, er det flere teknologiske begreper og forklaringer som må legges til grunn. Disse forklaringene er ment for å gi helhet i teorien. Sentralt i etableringen av koblinger i teknologien er piconett og scatternett, som ble kort presentert i introduksjonen av teknologien tidligere i rapporten.

På de neste sidene blir forklaringer hvordan piconett etableres, og hvordan enheter kan legges til og hvordan de kobles vekk. Videre går vi nærmere inn på tilstander i driftingen av Bluetooth-enheter, som er definert for å støtte funksjonaliteten.

4.2 Definisjoner

4.2.1 Piconett

Dette er et nett bestående av enheter som er koblet sammen *ad hoc*, det vil si at det trengs ingen forhåndsdefinisjoner eller planlegging, slik som i et standard-nettverk. To til åtte enheter kan være koblet til et slikt nettverk. Det er et såkalt likemanns-nettverk, det vil si at enhver enhet har lik aksess til andre straks den er tilkoblet. Uansett, er en enhet definert som master, mens resterende er slaver (se neste kapittel). I et piconett er det viktig at enhetene benytter den samme kanals aksesskode, at de benytter den samme kanals hoppesekvens og at klokkene i hver enhet er synkronisert. Disse parametrene avledes fra master i piconettet [38].

4.2.2 Master-enhet

En master i et piconett synkroniserer andre enheter ved dens klokke og hoppesekvenser [38].

4.2.3 Slave-enhet

Enheter i et piconett som ikke er master [38].

4.2.4 Scatternett

Flere piconett kan skape et større nett, såkalt scatternett, der hver piconett fortsatt holder på sin selvstendighet [38].

4.3 Nettverkstopologi

Topologien i Bluetooth-nettverket kan enten være et punkt-til-punkt eller punkt-til-multipunkt. Enhver enhet i et piconett kan etablere en forbindelse til et annet piconett, som skaper et scatternett. En oversikt på dette ble gitt i kapittel 2.4 Det er verdt å merke seg at en master i ett piconett ikke er linket til et annet piconett.

For punkt-til-multipunkt konfigurasjon, har man 1 Mbps som maksimum overføringshastighet innad i ett piconett. For et scatternett, minsker den maksimale overføringsrate i takt med økningen av antall piconett i scatternettet. Endringene i overføringsraten er avhengig av hvor mange enheter som befinner seg i piconettene [38].

4.4 Master-slave definisjon

Kanalen i piconettet er preget utelukkende av master. Enhetsadressen (BD_ADDR) til master bestemmer kanalhoppesekvens og kanal-aksesskodeen, med andre ord systemklokka til master bestemmer fasen i hoppingssekvensen og setter tidsberegning. I tillegg kontrollerer master trafikken med et polleskjema. (se kapittel 4.9.1)

Ut i fra definisjonen er master den enheten som initierer en kobling, til en eller flere slaver-enheter. Navnene master og slave henviser bare til protokollen på kanalen, det vil si at hvilke som helst enheter kan bli master og slaver i et piconett [1].

4.5 Hvordan tiden beregnes

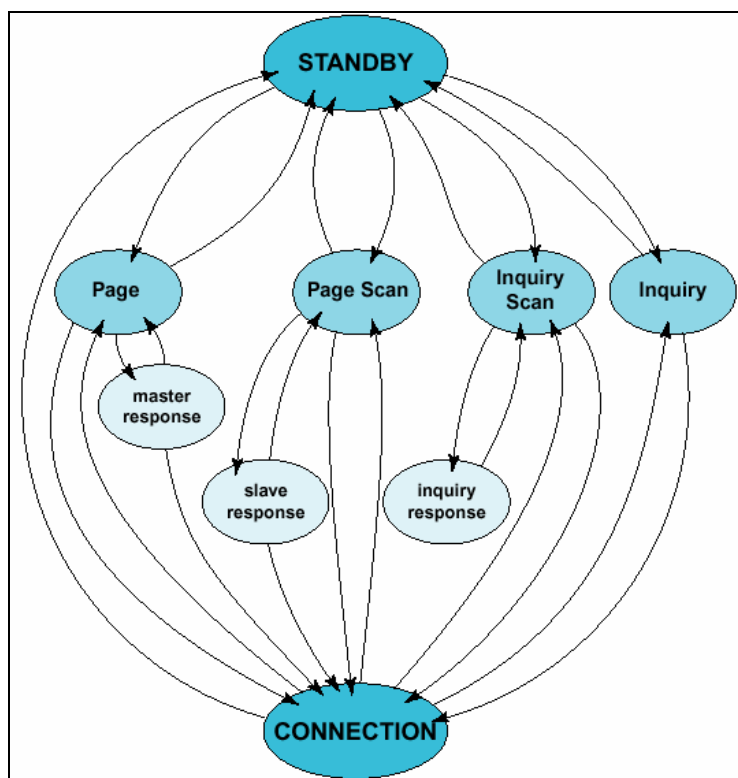
Enhver Bluetooth-enhet har en intern systemklokke som bestemmer styringen og hoppingen i transceiveren, og er svært viktig i synkroniseringen. Bluetooth-klokka er bestemt av en uavhengig ”innfødt” klokke som aldri er justert og som aldri er slått av. For synkronisering med andre enheter blir offset i tillegg til denne klokka benyttet, som gir midlertidige klokker, der disse er gjensidig synkronisert. Det er verdt å notere seg at Bluetooth-klokka har ingen forbindelse med klokketid i forhold til normal tid. Derfor kan den bli initialisert når som helst og med hvilken som helst verdi. Når piconettet er etablert, blir master-klokka overført til slavene. Hver slave legger til en offset til dens ”innfødte” klokke for å bli synkronisert til master-klokka. Siden klokkene er uavhengige, må offset oppdateres jevnlig [1].

4.6 Datatransmisjon

Som nevnt tidligere kan data bli overført både synkront og asynkront. Den synkrone metoden (SCO) er hovedsak øremerket tale, mens asynkrone (ACL) er primært for data, som for eksempel EKG-data. Innenfor et piconett kan hvert master-slave par benytte hver sin overføringsmetode, og disse metodene kan forandres til enhver tid. Tidsdelt duplex (TDD) er benyttet for både SCO og ACL. (SCO pakker blir vanligvis overlevert via reserverte intervaller, det betyr at pakkene er sendt i grupper uten å tillate forstyrrelser av andre overføringer. SCO pakker kan bli sendt uten polling.) ACL linker støtter både symmetriske og assymmetriske overføringer. Båndbredden kontrolleres av master, som bestemmer hvor mye hver slave kan beslaglegge. Slavene kan ikke overføre data før de har blitt pollet av master, og master kan kringkaste meldinger til slaver via ACL-linken [38].

4.7 Etablering av forbindelse

Figur 17 viser et tilstandsdiagram som illustrerer de forskjellige tilstandene som er benyttet i Link Controlleren. Som vi ser er det to hovedtilstander, STANDBY og CONNECTION. I tillegg er det syv undertilstander, som er midlertidige tilstander, benyttet for å legge til nye slaver i et piconett. Bevegelser mellom tilstandene bestemmes av kommandoer fra Link Controlleren.



Figur 18 : Tilstandsdiagram for enheter [1]

4.7.1 Standby

Denne tilstanden er standard-tilstanden til en Bluetooth-enhet. Her er enheten i en low-power modus. Kontrolleren kan gå fra denne tilstanden for å scanne for page- eller forespørselmeldinger, eller ved å utføre page eller forespørsel selv. En enhet vil ikke returnere til STANDBY etter å ha besvart på en page-melding, men vil heller gå over til CONNECTION som en slave. Ved å ha overført en vellykket page-forsøk, så vil enheten gå over i CONNECTION som en master [1].

4.7.2 Page- og forespørselprosedyre

For å få til nye tilkoblinger må prosedyrer som forespørsel og paging benyttes. *Forespørselprosedyren* gjør det mulig for en enhet å oppdage andre enheter innen rekkevidden, finne enhetsadressen og klokken deres. Med en *page-prosedyre* kan en reell kobling bli gjennomført. Det er bare enhetsadressen som trengs for å sette opp en oppkobling. Hvor fort dette lar seg gjøre bestemmes ut i fra kunnskapen om klokken. Det er viktig å legge merke til at en enhet som etablerer en forbindelse vil overføre en page-prosedyre og blir da automatisk master. I begge disse to prosedyrene benyttes henholdsvis forespørsels-aksesskoden (IAC – se kapittel 2.5.1) og enhets-aksesskoden (DAC). En enhet i tilstanden inquiry scan eller page scan samsvarer med disse respektive aksesskodene med en passende korrelator.

I page-prosessen kan flere paging-skjemaer benyttes. Det er derimot et obligatorisk skjema som må støttes av alle enhetene, og dette skjemaet benyttes når to enheter ”møtes” for første gang. I dette tilfellet følger en page-prosedyre direkte etter en forespørsel-prosedyre [1].

Ved hjelp av SDP (se kapittel 2.6.5) finner man enheters tjenester.

4.7.3 De andre tilstandene

Syv andre tilstander som også er beskrevet i figur 17. Disse tilstandene utgjør under-tilstander under etablering eller avkobling av forbindelser. Som figuren viser er det fire tilstander som etterfølger STANDBY. For å gjøre forklaringen kort, lytter enheten på "sitt eget område" ved scan, og blir slave ved oppkobling av en annens enhets interesse. Hvis en enhet tar initiativet selv, ved PAGE eller INQUIRY foresørsler, blir denne master [1].

(En mer utdypning her gir mye tallsustans som gir lite mening. Det viktigste er å forstå prosedyrene, og master-slave orienteringen.)

4.8 Etablert forbindelse

4.8.1 Connection

I denne tilstanden er forbindelse skapt og pakker kan bli sendt frem og tilbake. Den første informasjonspakkene i CONNECTION inneholder kontrollmeldinger som informasjon om linken og ytterligere detaljer om enhetene. Disse meldingene blir utvekslet mellom enhetenes Link Manager'e.

Å gå fra CONNECTION kan skje ved hjelp av kommandoene *detach* eller *reset*. *Detach* blir brukt hvis linken har blitt frakoblet på en normal måte. Med *reset* så resettes alle controller-prosessen, og controlleren må konfigureres på nytt [1].

4.8.2 Modus

En enhet kan være i forskjellige modus i en etablert forbindelse; *activ modus*, *sniff modus*, *hold modus* og *park modus*.

I *Activ modus* foregår det dataoverføring. Master planlegger overføringen ut i fra trafikkbehov til og fra forskjellige slaver.

Sniff modus så kan slavens lytteaktivitet bli redusert. En enhet i denne modusen har ikke en aktiv rolle i piconettet. Hvis en slave deltar på en ACL link er den nødt til å lytte til alle ACL-sporene til master-trafikken. Med *sniff modus* er tidslukene, der hvor master kan starte overføring til en slave, redusert. Dette gjør at master bare kan starte overføring i spesielle tidsluker.

Under CONNECTION kan linken til en slave bli satt i en *hold modus*, enten ved eget eller master ønske. Med denne modusen kan det frigjøres kapasitet for scanning, paging, inquiring eller betjene andre piconett.

Når en slave ikke lenger trenger å være delaktig på piconett-kanalen, men vil fremdeles fortsette å være synkronisert til kanalen, så kan den gå inn i *park mode*. Denne modusen er en *low-power modus* med veldig lite aktivitet i slaven.

Mange flere enn 7 enheter kan være tilkoblet en master, men bare syv vil være aktiv, og dermed ha en MAC adresse. De andre må da være i *parkmodus*, som er en modus der enhetene har "mistet" sin MAC-adresse (den fysiske adressen) [1].

4.9 Skjema

Prosedyrene på grensesnittnivå har visse ”regler” eller følgerutiner. Disse reglene kalles skjemaer, og er gir retningslinjer for hvordan kommunikasjon skal foregå.

4.9.1 Polleskjema

Master har alltid fullkontroll over piconettet, og slaver kan bare kommunisere med master og ikke til andre slaver [1] på grunn av bindende skjemaer. For å unngå kollisjoner på ACL-linkene, har en slave bare lov til å sende i bestemte luker. I SCO er reglene en smule annerledes. Her har slaven lov til å sende i luker som ikke er reserverte av andre.

4.10 Scatternet

Flere piconett kan dekke det samme arealet. Siden hvert piconett har hver sin master, hopper piconettene uavhengig av hverandre, og hvert nett har sin kanalhoppesekvens og fase bestemt av master. I tillegg er pakkene som blir overført på kanalene prioritert av forskjellige aksesskoder. Flere piconett som blir lagt til, øker sannsynlighetene for kollisjoner.

Når flere piconett dekker samme arealet, kan en enhet være delaktig i to eller flere overliggende piconett ved å anvende tidsmultipleksing. For å delta i kanalen som passer best, så skal enheten benytte den assosierte enhetsadressen til master. Den må også bruke den riktige klokkeforskyvningen (offset) for å få den korrekte fasen. En enhet kan fungere som en slave i flere piconett, men kan bare være master i ett. Hvis to piconett har samme master, er de synkronisert og bruker samme hoppesekvens, og derfor blir de til det samme piconettet. Eventuell ny master gir et nytt piconett, og nye parametre må settes. En gruppe av piconett der forbindelsene består av flere forskjellige piconett kalles *scatternett*.

En master eller slave kan bli en slave i et annet piconett ved at den blir kontaktet, ved paging, av en master i et annet piconett. På en annen side kan en enhet i et piconett kontakte master eller slave på et annet piconett. Som nevnt tidligere er den enheten som tar kontakt alltid den som ender opp som master. Hvis det er nødvendig for denne enheten å fungere som slave, er man nødt til foreta et master-slave bytte. Som en konsekvens av dette byttet, må de andre slavene flyttes til det nye piconettet, samt forandre tidsinnstilling og hoppeskjema [1].

4.10.1 Kommunikasjonen mellom flere piconett

Tidsmultipleksing må altså til for å veksle mellom flere piconett. For ACL-linker så kan en enhet spørre om å gå inn i hold- eller parkmodus i den nåværende piconettet i den tiden den kan slutte seg til et annet piconett. Dette gjøres ved å endre kanalparametrene. Enheter i sniffmodus kan ha tilstrekkelig tid til å besøke andre piconett mellom snifflukene. Hvis det er etablert SCO-linker, kan man bare besøke andre piconett i de ikke-reserverte lukene.

Siden klokkene til to master'e på to forskjellige piconett ikke er synkronisert, må en slave som deltar på to piconett passe på to forskjellige offset, som sammen med sin egen ”innfødte” klokke, skaper den ene eller den andre masterklokken. Siden disse to master-klokkene driver forskjellig, må det være jevnlig oppdateringer på disse offset'ene for at slaven kan bevare synkronisering til begge masterene [1].

4.11 Et sykehusmiljø

På steder der hensikten er mange stasjonære enheter, vil ikke mange nye sammensetninger av piconett finne sted, siden enhetene har hele tiden "kontroll" på sine nærmeste enheter. Nye sammensetninger skjer ved at nye enheter blir oppdaget og meldes inn i et piconett, og steder hvor det kan finnes mange bevegelige Bluetooth-enheter gir mange piconett. Et slikt sted kan være et sykehus.

På en sykehus-avdeling som skal benytte Bluetooth i forbindelse med EKG, er løsningen å ha en sende-enhet festet til pasienten. Denne enheten kommuniserer så til en mottaker-enhet, som i et slikt tilfelle kan være en monitor.

En innføring av en slik løsning, der Bluetooth vil være et suverent overføringsmedium fra pasienter til monitoreringsutstyr, gir stadige endringer i nettverkstopologien. Dette skjer fordi pasienter med sendere kan bevege seg rundt. Er situasjonen slik at et areal dekker mange pasienter som har behov for konstant tilkøpling, gir dette store trafikkmengder på kanalene mellom enhetene. Denne ekstra-trafikken er utslag av mye paging og forespørselsprosedyrer, samt overhead for kontrollering og synkronisering..

4.12 Plassering av enhetene

I utgangspunktet trengs ingen retningslinjer for hvor utstyr skal plasseres, da Bluetooth sendes uavhengig av bygningskonstruksjon (se kapittel 6.1). Men siden radiodelen har begrenset utgangseffekt, er det viktig å overholde rekkevidden. Dette er aktuelt for de som er i en hjemmesituasjon.

5 Sikkerhet

5.1 Innledning

EKG data inneholder personaldata, dette vil si data som er sensitive. Slike data er konfidensielle. Selv om det kanskje ikke er så stor interesse i å tyvlytte slike data, er all inntrenging en mulig påvirkning til dataene. Det er viktig å ha en teknologi som kan gi beskyttelse for unødige påvirkninger, derfor er sikkerhetsaspektet viktig å belyse. Sikkerheten i teknologien må kunne bære den sikkerheten som en slik applikasjon krever.

Investeringsplanen i det reviderte statsbudsjettet for 2000 går inn for investeringer i medisinsk teknisk utstyr på norske sykehus med nærmere 3 milliarder frem til 2004 [15].

Selv om den opprinnelige planperioden har blitt forlenget med to år, innenfor uendret bevilgningsramme, anses investeringen i sektoren som betydelig. Og en slik armada av teknisk utstyr krever ikke bare kompetanse, men også opplæring, service og kontroll, for å ivareta sikkerheten

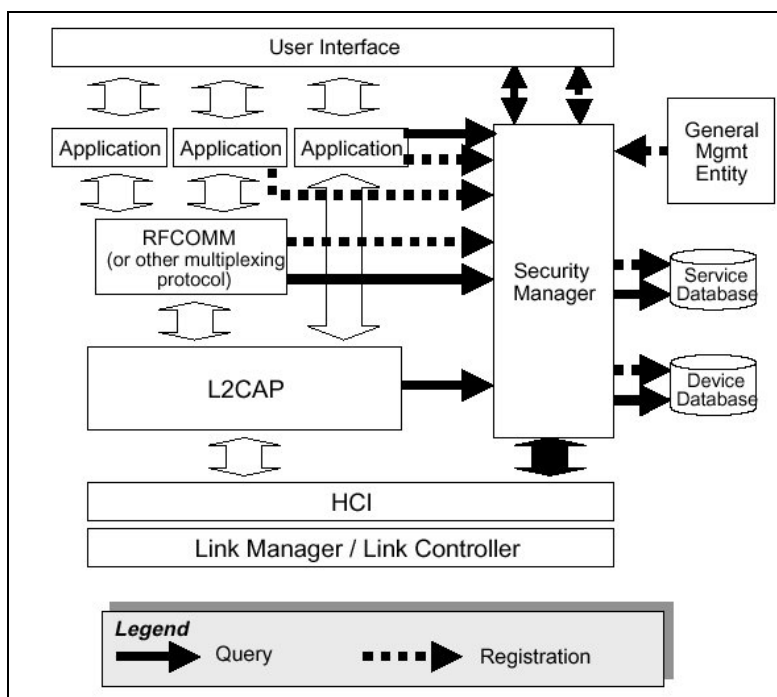
5.2 Sikkerhetsregler ved bestemt bruk – i dette fall EKG

For å få gi bruksbeskyttelse i EKG og informasjons-konfidensialitet i EKG-informasjonen er det nødt til å være sikkerhetsregler både på applikasjonslag og linklag. Disse reglene, eller lettere sagt forhåndsreglene, må være hensiktsmessig for miljøet som finnes. Med andre ord så betyr dette at autentiserings- og krypteringsrutiner må være implementert på samme måte i hver Bluetooth-enhet.

5.3 Oversikt over sikkerhetsarkitektur i Bluetooth

En generell arkitektur vises i figur 18. Hovedkomponenten i denne sammensetningen er en sikkerhets-manager. Denne "bestyreren" har følgende oppgaver [48] :

- lagre sikkerhetsrelatert informasjon på tjenester
- lagre sikkerhetsrelatert informasjon på enheter
- svare på aksess-forespørsler ved protokoll implementasjoner eller applikasjoner
- sikre autentisering og/eller kryptering før en forbindelse til applikasjonen finner sted



Figur 19 : Oversikt over sikkerhetsarkitekturen i Bluetooth. Legg merke til Security Manager, en sentral aktør i Bluetooth-sikkerhet [48]

Sikkerhets-manageren er delaktig i alt som omhandler sikkerhetkonseptet. Den er sentral i rutinene for sjekking om enheter har aksess. Videre blir aksess på tjenester styrt av autorisasjon ut i fra database. Dette gjør at Security Manager er hjernen bak prosessene om autentisering og autorisasjon. Det er denne enheten som skaper *trust*.

5.3.1 Autetisering og autorisasjon

Å utføre en prosess som verifiserer hvem som befinner seg på den andre siden av en link, kalles autentisering. Autentisering blir gjort for enheter. I et EKG-miljø vil dette tilsvare verifisering av pasienten. I Bluetooth blir dette gjort ved en autentiseringsprosedyre som bygger på den lagrede link-nøkkelen, eller ved pairing⁶.

Hvilken som helst enhet kan kommandere autentisering av en kommuniserende enhet og/eller at dataene på kanalen skal være kryptert. Slike kommandoer kan sendes via HCI, ved HCI-kommandoer (se kapittel 3.2.3) [48].

5.3.2 Kryptering

Overført informasjon kan bli beskyttet ved å kryptere payload til pakkene. Det foregår nemlig aldri kryptering på aksesskoden eller header [1].

5.3.3 Sikkerhetsnivå på enheter og tjenester

Bluetooth-teknologien definerer sikkerhetsnivåer på både enheter og tjenester. Dette bidrar til å øke sikkerhet både ved overføring, bruk og mot omverdenen. Hvilke sikkerhetsnivåer som trengs avhenger av applikasjonen, og kan konfigureres av bruker [48].

⁶ Pairing :Når to enheter ikke har en felles link-nøkkel, så lages det en initierings-nøkkel Deretter må gjennomføres autentisering, på grunnlag av denne nøkkelen, og ikke link-nøkkelen. Hvis dette ble gjort med hell, dannes link-nøkkelen.

5.3.4 FEC og ARQ - sikkerhet i pakkeoverføring

Bluetooth har feilkorreksjon (FEC) og automatisk "repeat-request" (ARQ) for å hindre feil i overførte pakker.

Hensikten med FEC er å hindre antall retransmisjoner. I et filfritt miljø, kan FEC gi unødvendig mye overhead, som reduserer den reelle overføringshastigheten.

Løsningen på dette er alternativ bruk av FEC.

ARQ's rolle er å sørge for at mottaker gir en bekreftelse på feilfri mottatt pakke [1].

5.4 Sikkerhet på linklag

Som det ble beskrevet i introduksjonen av teknologien, så foretar LMP sikkerhetsarbeidet, dette vil si på linklaget. Det blir benyttet fire forskjellige benevnelser for å holde sikkerhet ved like på linklag [1]; en allmenn kjent adresse som er unik for hver bruker, to hemmelige nøkler, og en random-number som er forskjellig for hver transaksjon.

| Entity | Size |
|---|------------|
| BD_ADDR | 48 bits |
| Private user key, authentication | 128 bits |
| Private user key, encryption configurable length (byte-wise) | 8-128 bits |
| RAND | 128 bits |

Figur 20 : Fire forskjellige benevnelser benyttes for å holde på sikkerhet ved like på linklag [1]

Den allmenn kjente adressen er enhetsadressen BD_ADDR. Dette er en 48-bit adresse som er unik for hver enhet. Man kan finne denne adressen ved å utøre en spørrerutine i en enhet, liknende forespørsel av IMEI-nummeret til en GSM-telefon.

De hemmelige nøklene er utledet under initialisering og er aldri senere synlig. Det normale er at krypteringsnøkkelen blir utledet av autentiseringsnøkkelen under autentiseringsprosessen. Størrelsen på nøkkelen som blir brukt i autentiseringsalgoritmen er alltid 128 bits. Nøkkelen for krypteringsalgoritmen kan variere mellom 1 og 16 oktetter (8 – 128 bits). Det er to grunner til at krypteringsnøkkelen skal være konfigurert. For det første så har det noe med de forskjellige lands pålagte krav på dette området. Den andre grunnen er å gjøre det lettere å gi retningslinjer for sikkerhetsoppgradering i fremtiden, uten å måtte bruke mye penger på omkonstruksjoner på hardware. Dette forklares med at ved å øke størrelsen på nøkkelen, vil dette være den enkleste måten å bekjempe økende maskinstyrke på den andre siden. I dag (1999) [1] synes en 64 bits krypteringsnøkkel å gi en tilfredsstillende sikkerhet for de fleste applikasjoner. Denne nøkkelen oppfyller kravet om at andre nærliggende enheter ikke skal benytte seg av andres tjeneste. Hvis denne nøkkelen ikke var til stede, ville det f.eks. vært mulig for en medpassasjer på bussen å ringe ut på din mobiltelefon med sin trådløse hodetelefon.

Random-nummeret RAND kan bli avledet fra en randomprosess i en Bluetooth-enhet. Dette er ingen statisk parameter, men vil endre seg ofte. Størrelsen på dette nummeret er såpass stort at dubletter anses som umulig.

5.5 Spread-spectrum teknologi og frekvens-hopping

Mange trådløse nett benytter spread-spectrum radiotransmisjonsteknikk. [39], heriblant de fleste LAN. En slik teknologi tar også Bluetooth bruk av [38]. Spread-spektrum teknologi ble først introdusert for ca. 50 år siden, i den hensikt å forbedre meldingsintegriteten og sikkerheten. Spectrum-spread systemer er laget for å være motstandsdyktige mot støy, interferens, jamming, og uautoriserte ”oppdagelser”. Senderne sender signalene ut over en rekke frekvenser på en meget lav effekt, noe som er til sterk kontrast til andre radioer som konsentrerer alt effekt inn på en frekvens. Det finnes flere måter å implementere denne teknologien, men en av de mest vanlige er frekvens-hopping. Med sitt frekvenshopp spread-spectrum (FHSS) danner Bluetooth et godt og sikkert overføringssystem.

For å tyvlytte til et system med FHSS må man først vite spredekoden. Deretter må man vite hoppemønsteret. Til slutt er man også nødt til å vite hvilket frekvensbånd som benyttes, noe som er velkjent i Bluetooth, og modulasjonsteknikken som er benyttet. Selv om man vet frekvensbåndet til Bluetooth, er ikke det til så mye hjelp da hoppene skjer i en hastighet på 1600 hopp i sekundet.

Ved bruk av FHSS vil en da unngå problemer med å forstyrre annet frekvent utstyr på sykehus.

5.6 Generelle krav til kompetanse, service og kontroll

Stadig mer avansert teknologi i helsesektoren krever flere sikkerhetstiltak. Dette er spesielt viktig i en innføringsfase av ny teknologi. En innføring av Bluetooth, vil gi forandring i vante rutiner, som kanskje er vonde å vende. Dette står i sterk grad i sammenheng til motstand til forandring. En nøye og planlagt innkjøringsfase er nødvendig, samtidig er det viktig å ha god opplæring av de som skal benytte nytt utstyr og enheter.

Siden bevilgningsrammen er øket med to år, vil dette føre til færre investeringer en planlagt. Et resultat av dette kan være at en innføring av eventuell Bluetooth-enheter vil gå over lengre tid. Dermed blir rutinene endret i et større tidsperspektiv. I en tenkt situasjon kan det være på ett tidspunkt være nødt til å ta hensyn til forskjellige utstyr som benytter vidt forskjellige teknologier, og som også krever forskjellige rutiner.

6 Bluetooth i et fremtidig perspektiv – medisinsk-teknisk hjelpemiddel?

Har Bluetooth noen fremtid? Blant analytikere finnes det sprikende meninger. Det er vel flere bruksområder som kanskje hadde vært bedre uten Bluetooth. Uansett, Bluetooth har potensial, og i mange sammenhenger kan teknologien gi de etterlengtede løsninger.

Siden initiativet så dagens lys i mai 1998, har nærmere 1600 selskaper meldt seg inn i SIG. Nye applikasjoner tilpasset Bluetooth dukker stadig opp, og den store utbredelsen av spesifikasjonen har fått SIG-partnerne til å utvide gruppen. De ni selskapene i promo-gruppa [jfr. SIG-kap] skal samarbeide for å løfte frem nye, kommersielle Bluetooth-produkter på markedet, deriblant medisinsk-teknisk utstyr.

6.1 Generelt - Bluetooth vs. annet trådløs overføring

I dag brukes infrarød stråling til å overføre data etter standarden IrDA (jfr. OBEX). Siden Bluetooth baserer seg på radiobølger, og ikke infrarøde stråler, trenger man ikke lenger å ha fritt rom mellom enhetene. Dette gir revolusjonerende endringer i et kontorlandskap der det ikke er fritt rom.

En av de store fordelene til Bluetooth er at den gir en kraftig trådløs forbindelse, og det til en relativt rimelig penge [40]. Dessuten er teknologien lett å implementere og systemer fra ulike leverandører kan operere problemfritt sammen. Dette er momenter analysefirmaet Frost & Sullivan har sett på som gode argumenter for at Bluetooth skal bli en suksess [41]. Et annet viktig moment er at teknologien er uavhengig av underliggende nettverk.

Standarden er åpen, noe som gjør at alle som vil anvende Bluetooth-teknologien i sine produkter kan gjøre det gratis. Ved å gjøre standarden åpen legges det til rette for produsenter å utvikle sine egne produkter basert på teknologien. Dette vil også være til nytte for videre eksperiment, da erfaringer med bruk av teknologien øker i et voldsomt tempo. I følge SIG ("promoter"-gruppen) er det primære målet å sørge for at teknologien blir så tilgjengelig og så rimelig som mulig [17]

At teknologien i tillegg har en bred industriell backing lover bra, med tanke på fremtidsutsiktene.

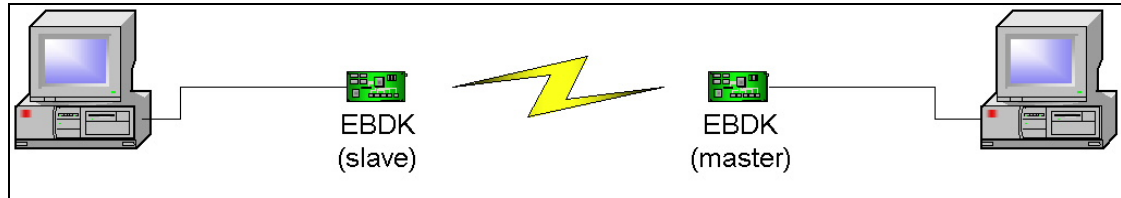
6.2 Bluetooth og EKG

Et australsk selskap som Micromedical Industries er allerede sterkt delaktig i SIG. Dette er et selskap som har som mål å bli en ledende leverandør av produkter innen hjertemedisin ved bruk av trådløst overvåkningssystemnettverk, såkalt e-health. Dette omfatter alt fra diagnostikk, behandling, rehabilitering og forebygging. De er allerede verdensledende innen portable hjertemonitoreringsutstyr og telemedisin [42]. Grunnen til deres delaktighet i SIG, er for å være med på å skape løsninger for sømløs, sikker og feilfri monitorering for hjertepasienter, uansett hvor disse pasientene måtte befinne seg. Flere av de verdensledende flyselskapene blir nå installert med hjertemonitor fra Micromedia. Videre hevdes det at teknologien kan passe godt i et kunstig hjerte som selskapet utarbeider i samarbeid med University of Technology i Sydney og University of New South Wales [43].

7 Resultater

7.1 Overføringsskjema

Overføringen ble foretatt mellom to maskiner på IKT-senteret. Begge maskinene hadde tilkoblet hver sitt EBDK.

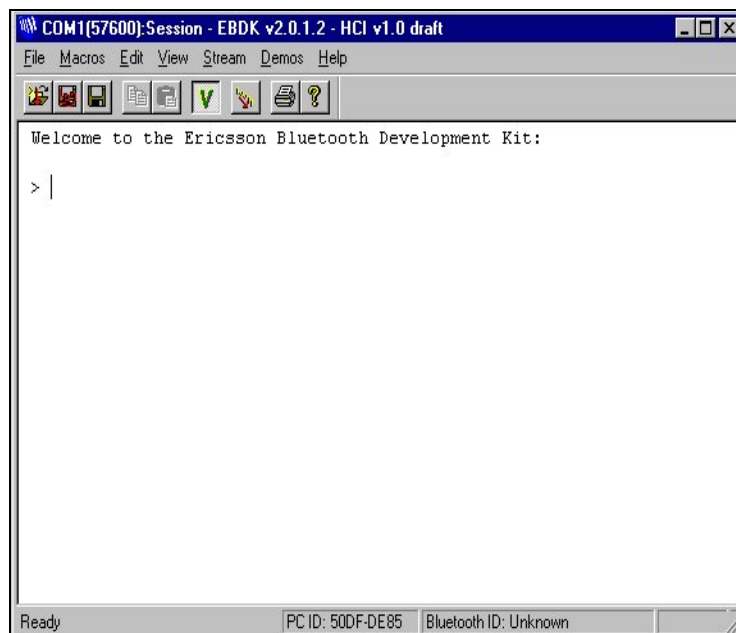


Figur 21 : Overføring fra maskin til maskin vha EBDK

7.2 Programvare

På begge maskinene på IKT-senteret var det i forbindelse med installeringen av EBDK, også blitt installert tilhørende programvare. Uten et slikt programvare, hadde det vært umulig å gjøre en brukersvennlig bruk av EBDK.

Programvaret hadde et primærvindu (se figur 21). Dette vinduet kunne brukes til å programmere HCI-kommandoer. Ved bruk av ferdige innebygde applikasjoner i programvaren ble dette vinduet mest brukt til å få frem oppkoblingsprosedyrene.

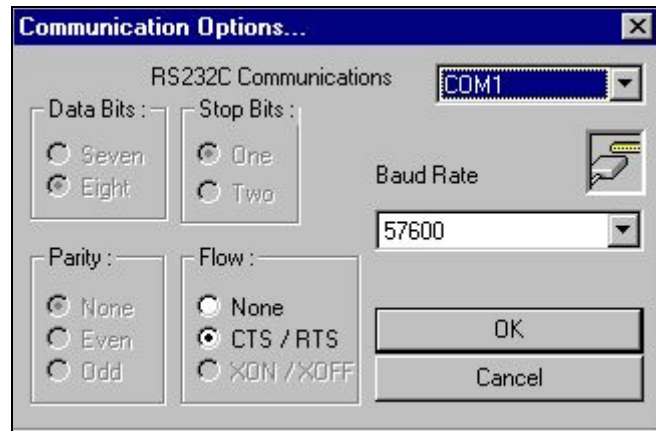


Figur 22 : Primærvinduet i programvaren til EBDK.

7.3 Oppkobling mot EBDK

En koble seg opp mot hardware i EBDK gjennom programvaren. I denne oppkoblingsfasen får man muligheter til å bestemme overføringen mellom PCen og EBDK. Her velges den serieporten EBDK er koblet til, samt hastighet denne overføringen skal foregå med.

Overføringer kunne bare foregå med en hastighet på 57600 baud. (se kapittel 7.5). Det ble ikke tatt hensyn til å endre "flow", som var satt til CTS/RTS.



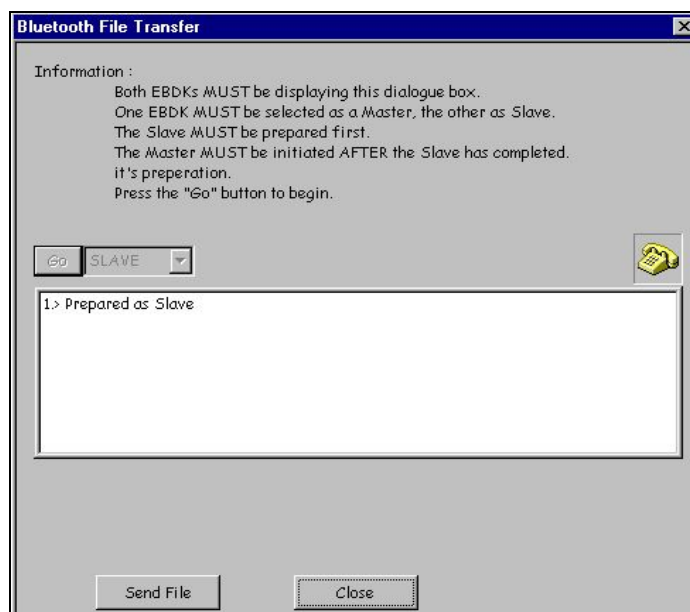
Figur 23 : Overføring mellom PC og EBDK bestemmes

7.4 File Transfer applikasjon

Overføringen av lagrede EKG-filene ble foretatt av en File Transfer applikasjon, som fantes i det tilhørende EBDK-software. Et slikt valg ville gjøre det mulig å overføre de lagrede filene mellom hver maskins harddisk.

Før overføringen kunne realiseres måtte maskinene være klargjort som master eller slave. Instruksjonene (se bilde) fortalte at slaven måtte være aktivisert først. Dette hadde sammenheng med at master ikke kunne foreta scanning av en ikke-aktivisert slave (se neste kapittel), da aktivisering av master spør etter enhetsadressen (BDDR) til en den aktiv slaven som den skal kommunisere med. Dessuten ble ble ikke denne enhetsadressen kjent før man hadde foretatt aktivisering av slave.

Ved aktivisering av master/slave dukket det opp en oppkoblingsprosedyre (se vedlegg 2 og 3) i primærvinduet.



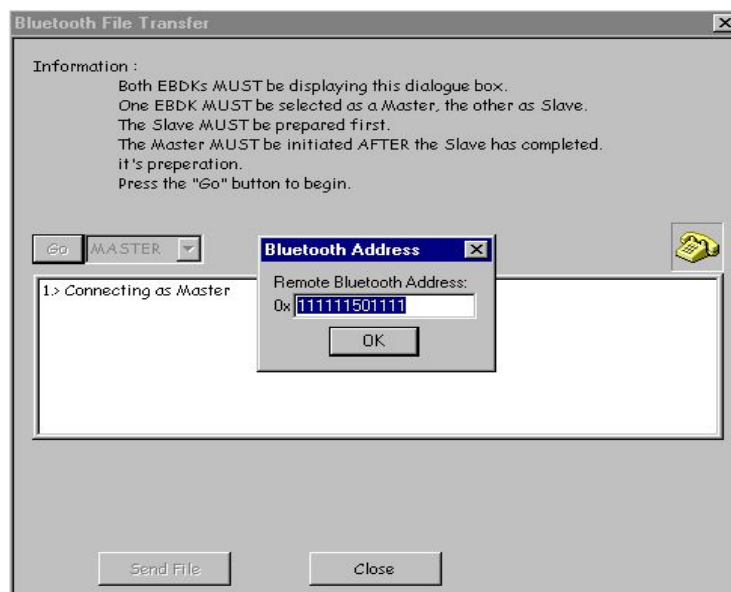
Figur 24 : Slave er blitt aktivisert

Nå var slave aktivisert. Dessuten var dens egen enhetsadresse blitt registrert. Denne enhetsadressen vises i figur 24. En slik aktiviseringsprosedyre gjør også scanning på andre enheter. Dette vises ved at registrerte enhetsadresser blir registrert i denne prosedyren.

```
COM1(57600):Session - EBDK v2.0.1.2 - HCI v1.0 draft
File  Macros  Edit  View  Stream  Demos  Help
#
# Length = 3 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 09 10 00
# HCI_PacketOpcode      [ 2 byte/s] = 0x1009 (4105)
# Parameter_Length     [ 1 byte/s] = 0x00 (0)
>
< HCIReadBDADDR - RBDADDR
# Mon Jun 05 16:36:09 2000
#
# Length = 12 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 0a 01 09 10 00 11 11 50 11 11 11
# HCI_PacketOpcode      [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x0A (10)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0x1009 (4105)
#                       (0x00001009)
# Status                [ 1 byte/s] = SUCCESS (0) (0x00000000)
# BD_ADDR               [ 6 byte/s] = 0x111115011111
>
Tx Packet Count 11, Rx Packet Count 11 PC ID: 50DF-DE85 Bluetooth ID: 0x111115011111
```

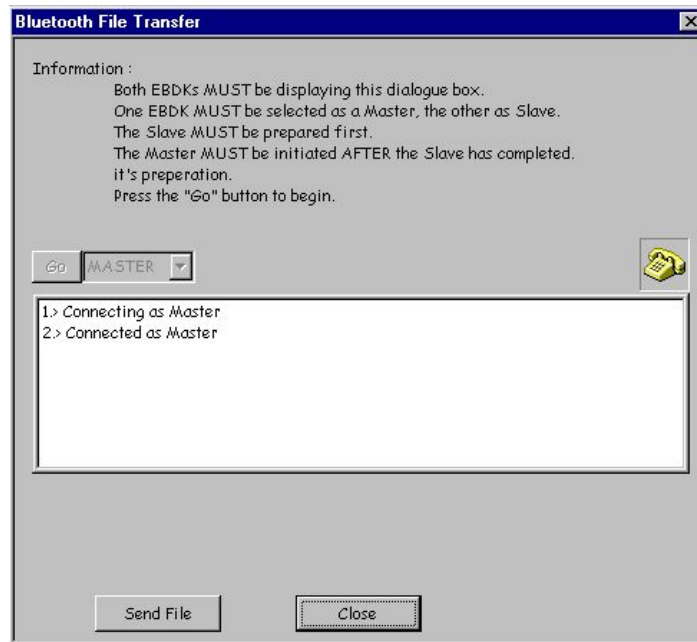
Figur 25 : Et utdrag av oppkoblingsprosedyren til slaven. Nederst til høyre er nå enhetsadressen til slaven blitt registrert. Legg også merke til at den har registrert enhetsadressen til en annen enhet (BD_ADDR)

Etter at slaven var aktivisert var det mulig å aktivisere master. Her fikk man spørsmål om enhetsadressen til slaven, som var mulig å lese i opp fra oppkoblingprosedyren til slaven (se vedlegg 2)



Figur 26 : Enhetsadressen til den aktiverte slaven må oppgis når man skal aktivisere master.

Ved å taste inn riktig enhetsadresse ble også master aktivisert.



Figur 27 : Master er blitt aktivisert

Nå fantes det en etablert forbindelse, og en filoverføring var mulig. Filoverføringen skulle i utgangspunktet kunne skje i begge retninger, men det ble valgt å sende fra slave til master. (se neste kapittel).

Ved mottak av en fil, kom et overføringsskjema opp i primærvinduet. Dette skjemaet besto av datainnholdet på den overførte filen. Størrelsen på dette skjemaet varierte med størrelsen på filene. Et slikt overføringsskjema inneholdt lange lister av overførte byte.

Det ble gjort forsøk på å sende over alle fire filene, det vil si begge wav-filene og deres tilknyttede MP3-utgaver. I mottaket ble overføringsskjemaet betydelig større ved overføring av wav-filene siden disse filene var langt større. I vedlegg x finnes skjemaet for overføringen av den ene MP3-filen. Denne filen hadde en størrelse på 80.5KByte, noe som også kan registreres i dette skjemaet.

7.5 Empirisk kunnskap

Ved testing av programvare og EBDK ble det oppdaget flere ting, som ga lærdom til videre testing:

- Filstørrelser større enn 2.5MB ble blokkert i overføringen
 - dette ble registrert ved testing av andre filer enn EKG-filene.
- Kun overføring i retningen fra slave til master lot seg gjennomføre.
 - overføring den andre retningen ga beskjeden "Transfer Aborted : Invalid Packet."
- Filene kunne bare sendes fra harddisk og ikke fra floppy.
 - lesing fra floppy tar lenger tid enn fra harddisk

- Resetting av EBDK var nødvendig etter flere forsøk
 - resettet man ikke etter flere forsøk ble som regel overføringen blokkert.
- Ved oppkobling kunne bare overføringstaten på 57600 baud gi muligheter for overføring
 - andre overføringshastigheter lot seg ikke gjennomføre, og ble blokkert allerede ved oppkobling av slave/master.

8 Drøfting og diskusjon

8.1 EKG-prosessedata via Bluetooth

Overføring av EKG-prosessedata krever sømløs overføring, og en automatikk i et slik system må sies å være obligatorisk. Hvilken grad av sanntid som skal kreves, er avhengig av hvilket bruk som skal føres. Skal det kun overføres data til en sentral database for at disse kan benyttes som referanse senere dette isolert sett ikke sanntid. I et slikt tilfelle kan heller ikke pasienten helsestatus anses som særlig kritisk. Nå vil vel kanskje heller ikke de mest "alvorlige" pasientene være hjemmeliggende. Sånn sett kan overføring av filer være aktuelt, men lite trolig. Overføring av sampler direkte vil gi en mer dynamisk overføring, og er gjerne det man kaller optimalt.

Overføringen av filer vil gå med protokollen OBEX. Denne protokollen støtter også push-teknologi, som er aktuell som bruk i selve Bluetooth-overføringen. For en sømløs og automatisk overføring er slik push det riktige valget. Det kan også velges å overføre filer på generelt grunnlag, noe som ble foretatt i min praktiske utførelse. Velges slik løsning, er man nødt til å lempe på sanntidskravet, og legge til rette løsninger som bærer minst preg av dette. Slike løsninger vil være uavhengige av Bluetooth.

Skal overføringen være direkte sampler, vil tale-overføring trolig være det beste alternativet. Her vil overføringen foregå med tilstrekkelig samplingsfrekvens, samtidig med at sanntid oppfylles hvis ønskelig.

Båndbredden i et EKG-signal er ikke stor, og Bluetooth vil være ypperlig i så måte. Problemet i overføringsfasen, hvis det kan kalles et problem i den relativt lave EKG-båndbredde, kommer først ved etablering av scatternett.

En automatisk overføring kan blant annet foregå med push-teknologien i Bluetooth, alternativt er en form for streaming-teknologi. Bakdelen med streamingteknologi er at den ikke tar i bruk mellomlagring, ei heller i mottak. Dette gjør teknologien mindre egnet til dette formålet.

Bruk av database blir et meget robust verktøy i databehandlingen, og er et nyttig verktøy hvis ikke sanntidskravet settes i fokus. Denne danner sikkerhet ved lagring før videresending med WAP. Da et grensesnitt mellom Bluetooth og WAP ikke foregår problemfritt og sømløst, er tilpasninger nødvendig. Bruk av database blir da et naturlig valg, som WAP-teknologien senere kan aksessere ved hjelp av ASP og SQL, for så å ta dataene videre.

Interoperabilitet med WAP finnes dokumentert i Bluetooth-spesifikasjonen, men da med Bluetooth som en WAP-bærer. I dette tilfellet er situasjonen omvendt.

Et viktig moment med en løsning som dette er at det vil kreve flere strømvhengigeheter. uten noen form for backup, blir forbindelsen til en sentral database brutt. Dette er ikke særlig helseforsvarlig.

8.2 Teoretisk løsningsforslag

Da løsningen ikke lot seg gjøre som opprinnelig planlagt, ble resultatet en praktisk gjennomføring av kun deler av denne reelle løsningen. Uansett komplett gjennomført løsning eller ikke; overføringsbiten hadde blitt den samme ved bruk av filer.

På grunn av at den reelle løsningen ikke ble gjennomført, fikk en heller ikke testet ut systemet direkte på en person. Hvis en i tillegg hadde vektlagt oppgaven mer i retning av databehandling, med en mindre fokus på Bluetooth, kunne kanskje en gjennomføring av EKG-overføring av sampler blitt gjennomført.

Denne såkalte reelle løsningsforslaget vil antakelig ikke være det mest praktiske ved reellt bruk. Målet er selvfølgelig å være mest mulig uavhengig av flere enheter enn mulig. En løsning, som tidligere i rapporten ble kalt den ideelle, vil i så måte være mest hensiktsmessig i bruk.

Det ble ikke lagt vekt på å få frem et selvprogrammert monitorerings-software. Begrunnelsen for dette er min begrensede programmeringskunnskap, som ville påføre stor arbeidstid. En voldsom innsats på dette feltet ville bare gått utover belysingen av Bluetooth som teknologi. Resultatet hadde lett blitt halvdårlig resultat i begge ender.

Resultatet av overføringene med EBDK ga store begrensninger, da en hadde liten fleksibilitet i valgmuligheter for overføringen. Da farten kun sendes i en hastighet av 57600 baud, vil overføringen stå i en sterk kontrast til det ideelle. Videre fikk man erfaringer som gikk på tvers av det Bluetooth-teknologien gir støtte for. Da tenker jeg mest på usikkerhet ved overføringer der EBDK-enhetene sto mer enn to meter fra hverandre. Dette må gå på kontoen til EBDK, som kun et utviklingsverktøy for Bluetooth.

Uansett er Bluetooth kun i startfasen, og min oppfatning er at enormt mye skjer på området. Siden blant andre Micromedical Industres i Australia er sterkt på banen med liknende løsninger vil det formodningen ikke bli alt for lenge til at en slik løsning blir opp mot min ideelle.

8.3 Mobilitet

8.3.1 Generelt

Bluetooth gir ingen støtte for mobilitet over større områder. Den gir kun støtte for bevegelse innenfor et begrenset område som dekkes av radiosenderen. Et større område kan dekkes opp av enten større effekt på sender-delen på radioen, eller ved å knytte seg til et LAN-punkt.

Det finnes ingen støtte i Bluetooth for å foreta handover mellom enheter, eksempelvis mellom monitor-enheter [52]. Det ideelle hadde jo vært å ha en handover på SDP-plan mellom monitorene.

8.3.2 Et sykehusmiljø

I et slikt miljø det pasienten må ha konstant tilkøpling, vil en løsning der pasientdel er master virke som den mest sannsynlige, med tanke på master-slave situasjonen. I disse tilfellene er det jo pasienten som "trenger" kontakt til nye monitoreringsenheter etterhvert som pasienten beveger seg utenfor rekkevidden til den sist benyttede monitoreringsenheten. Dette gir en løsning der alle pasientene er master og monitorene er slaver. Resultatet av dette er at piconettene blir minimale, bare to enheter. Videre gir dette mye hyppigere topologiforandringer i nettet. Siden monitoren blir slave, er den nødt til å ha forbindelser til mer enn en master, siden en-til-en løsning i et bevegelig miljø ikke anses som en ideell løsning, ei heller praktisk, da dette ikke ville tatt fordelene av scatternett.

Hvis vi skal snu på flisa, og si at monitoren blir master i en forbindelse, er man nødt til å ha en løsning som gjør at monitoren initierer forbindelsen. Dette virker litt uodynamisk og upraktisk, siden monitorene vanligvis kan sees på som stasjonære. En slik løsning hadde sannsynligvis gitt mye skanning. En kan vel si at mye av det som blir bespart av mindre topologiforandringer blir spist opp av mer skanning. Uansett, så gir denne løsningen mindre sannsynlighet for kollisjon – rent teoretisk. Årsaken til dette er at det hadde blitt færre piconett. En annen ting er at slavene opererer tidsmultiplekst med andre mastere de har forbindelse med, og i tillegg må operere med forskjellige offset, og for å bevare synkronisering med alle mastere må foregå jevnlig oppdateringer på hver offset. Delaktighet på flere piconett gir dermed utslag på dataratene på hver forbindelse.

Båndbredden til EKG-signal er relativt liten, bare 150 Hz. Men hvis en ha et miljø det det er mye mobilitet, vil hver pasient bli som nevnt master i en en-til-en løsning. Dette gir stadig nye etableringer av piconett. Resultatene av dette blir mye ekstra overhead til kontrollering og synkronisering, siden nye parametre må settes. Samtidig skaper det mye retransmisjon, og sannsynligheten for kollisjon øker også, som fører til at BER⁷ øker [53]. All slik ”ekstraoverføring” tar større kapasiteter på linjen, og da Bluetooth har en begrenset overføringskapasitet ved mange scatternett, vil kapasiteten på linjene være uheldig store.

Det er vanskelig å si om det gis støtte for mobilitet til neste generasjon Bluetooth. Den neste spesifikasjonen, Specification 2.0, er planlagt utgitt i 2001/2002 [54].

8.3.3 Et hjemmemiljø

I et hjemmemiljø, som løsningen er en del av, vil situasjonen være slik at mobilitets-spørsmålet faller i grus. Det mest interessant momentet her er hvordan en skal løse problematikken rundt begrenset rekkevidde. Hva skjer hvis hjemmepasienten går utenfor de 10 meterene som radioen gir støtte for? Derfor er plasseringene av utstyrsenhetene viktig.

Den mest naturlige løsningen på rekkeviddeproblemet er å øke effekten på radioen. Da vil også rekkevidden bli større. En annen løsning er å benytte andre teknologier som bærere, men da må disse bærerene støtte interoptibilitet med Bluetooth.

8.4 Sikkerhet

8.4.1 Tiltak i teknologien

Teknologisk sett har det i Bluetooth gjort mange sikkerhetstiltak. Det finnes sikkerhetstiltak på både tjenestene og på enhetene. Avhengig av applikasjon, trengs forskjellige krav til sikkerhet, og dette bli løst gjennom bruk av flere sikkerhetsnivåer. Sikkerhetsnivåene er konfigurerbare og kan settes av bruker. Foruten alle sikkerhetstiltakene teknologien er bygget opp rundt, er selve grunnpilaren i overføringen bruk av FHSS. Spread-spectrum, med frekvenshopp, hindrer inntrenging utenfra. Dessuten hindrer bruk av slik teknologi forstyrrelser på annet frekvent utstyr. Den teoretiske muligheten for å skape to identiske meldinger (dubletter), er mikroskopisk, da hvert randomnummer har begrenset levetid.

⁷ BER (Bit Error Rate) angir hyppigheten av feil i en telekommunikasjonsoverføring som en andel av det totale antall bit overført.

Et system er aldri mer sikker enn det svakeste leddet. Derfor er det viktig at teknologien ikke blir brukt feil eller blir misbrukt. En grunnleggende kompetanse er med på å forsvare en god bruk av teknologien.

8.4.2 Overføringer

Som formidler av signaloverføring for EKG-prosessdata, er det ingen direkte sikkerhetsproblemer. Tyvlytting er kanskje ikke noe stort problem på dette området, men uansett hindrer FHSS at slikt skal finne sted. Videre er kryptering og autentisering, i tillegg til strenge sikkerhetsnivåer, med på å sikre *trust*. Bluetooth vil, isolert sett som et ende-til-ende løsning til EKG-prosessdata, ikke vil skape problemer.

Bluetooth benytter seg også av feilkorleksjon og automatisk "repeat" for feil eller manglene data [44].

8.4.3 Dimesjonering av nettene i et større miljø

En dimensjonering av piconett til scatternet er med på å avgjøre sikkerheten. Flere antall scatternet [52] er med på å begrense overføringshastigheten per pasient. Samtidig trenger man like mange tilkoblingspunkter som 1/7 av maksimal pasientstørrelse, hvis alle pasienten skal være pålogget samtidig (alternativt er at enheter er i park-modus). Et tilkoblingspunkt er i dette tilfellet en EKG-monitor eller et annet Bluetooth-kompatibelt utstyr.

Siden Bluetooth ikke gir støtte for mobilitet per dags dato, har man mulighet til å benytte LAN-punkter [11] eller så må hver pasient være koblet opp mot en fast monitor. Slik jeg ser det er det, av hensyn til pasientsikkerheten, bare aktuelt med egen monitor for hver pasient. I et Bluetooth-miljø gir dette en-til-en forbindelse, noe som gir mange piconett. Mange piconett er ikke ugunstig i seg selv. Dette gir mye overhead, som igjen gir mye "ekstradata". Resultatene av dette blir økt sannsynlighet for kollisjon.

Hvis det er behov for å videresende disse ved kobling mellom flere enheter, vil dette gi scatternet. Hvis det er planer om å videresende EKG-dataene for felleslagring, anbefales ikke dette på grunn av dannelse av mange scatternet og derav gir reduksjoner på overføringene på EKG-prosessdataene. Dette kan i verste fall, med tanke på videre utvidelser av prosessdataene via Bluetooth, gi datatap.

En monitor kan behandle syv pasienter av gangen, uten å påføre noe tap for overføringene. Dette gir muligheten for å veksle mellom pasientene fra monitoren. En slik løsning påfører langt færre scatternet ved videresending, men gir et prioritetsproblem. For hvis det utløses en krisesituasjon, må selvfølgelig prosessdataene fra denne pasienten få "forkjørsrett". I praksis vil dette si at en må ha et varslingsutstyr. Da alle syv plassene er opptatt må enten redusere antall pasienter i et piconett, eller så må man ha et varslingsutstyr som går utenom Bluetooth. I tillegg er det ganske naturlig at i krisesituasjon får automatisk endret monitoreringsbilde til den aktuelle pasienten. Hvis en setter ting på spissen gir en slik løsning ytterligere problemer. Hva hvis to eller flere krisesituasjoner inntreffer i samme piconett?

8.4.4 I en hjemmesituasjon

En pasient som er hjemmeværende med sitt EKG-utstyr, vil ikke ha de samme nettproblemene som i et sykehus eller sykehjem. Her er antall Bluetooth-enheter redusert. Derfor vil ikke de ovennevnte problemene ha noe effekt.

Det som denne situasjonen krever, og som er dens svake punkt, er interoptibiliteten med andre teknologier som skal være bærere av prosessdataene videre. Per dags dato gir spesifikasjonen kun gode løsninger på interoptibiliteten mellom disse teknologiene der Bluetooth er bærer for WAP, og ikke omvendt.

I vår løsning er grensesnittet mellom Bluetooth og WAP en database. Dette er i seg selv er grei løsning, men for dynamikkens skyld skulle interoptibiliteten være uavhengig av flest mulig strømforsynende enheter. Pasient-sikkerhet setter et krav at Bluetooth-eneheten må ha backup-muligheter ved strømbrydd. Ved flere "innkoblede" enheter, krever dette større backup-kapasitet.

Som nevnt i kapittel 4.12, så er plasseringene av utstyrsenhetene viktig, med tanke på begrensninger i rekkevidden, som igjen kan skape brydd i overføringene. Dette løses ved å investere i den mer effektsterke radioen.

Videre blir det av naturlige grunner forsinkelser i systemet ved innføring av flere ledd i en slik overføringsløsning. Undertegnede setter også tvil i om WAP kan stille de sikkerhetskrav som EKG-diagnostikk krever, da denne teknologien er bygget opp om telefoni-nettet, som er kjent for å være en best-effort tjeneste. Dette emnet kan studeres i andre rapporter [55, 56].

8.4.5 Behandling av utstyr

Teknologien er aldri sterkere enn det svake ledd, derfor er det viktig at det finne god oppfølging av service og kontroll. En god opplæring av bruk i utstyrene er en selvfølge.

8.4.6 Forskjellige fabrikanter

Som jeg beskrev i kapittel 2.1 vil det kun være en type av Bluetooth, og alle sertifiserte Bluetooth-enheter kan kommunisere med hverandre. Dette gjør bruk av enheter fra forskjellig utstyrsleverandører problemfritt. En kan vel si at Bluetooth opererer uavhengig av fabrikat.

9 Konklusjon

Oppgaven har bestått av å se på mulighetene for Bluetooth som en EKG-bærer. I tillegg er flere formater for overføring blitt evaluert.

Bluetooth vil være et utmerket overføringsmedium for EKG ved bruk lydformater. Både EKG-karakteristikk og sikkerhet blir ivarettatt på en slik måte at Bluetooth kan sees på som et meget godt alternativ som en slik bærer. Hvilken overføringsmetode en skal benytte innad i Bluetooth, er avhengig av bruk, krav og format. En filformat støttes ved bruk av protokollen OBEX, som igjen gir støtte for push. Slikt push er nødvendig for å gi en mest mulig automatisk og sømløs overføring. Alternativt til filformatet er sampler, som fint kan overføres som lyd gjennom tale, og i tillegg oppfylle EKG-karakteristikka.

Begrensningene i Bluetooth i dag er mobilitetsproblematikken, da teknologien isolert sett ikke gir noen støtte for mobilitet. Derfor egner ikke Bluetooth seg alene i et miljø basert på mobilitet, og er derfor ikke aktuelt i en slik sammenheng. Benytter man derimot Bluetooth i sammenheng med LAN-aksesspunkter vil mobilitetsspørsmålet løses.

Per dags dato finnes ingen interoptabilitet med WAP, der WAP er bærer for Bluetooth. Derfor er en systemintegrasjon av disse preget av moderate tilpasninger, for å få frem prosessedata. Den neste spesifikasjons-utgivelse av Bluetooth vil garantert ha bedre interoperabilitet med andre systemer, men en slik utgivelse er ikke ventet før 2001, kanskje så sent som i 2002. Da vil også GPRS være med på å gi ytterligere "forbedringer" for WAP.

Konklusjonen blir at Bluetooth klarer å håndtere EKG-signaler, både båndbredde-messig og sikkerhetsmessig. Derfor egner den seg som et overføringsmedium for EKG, vel og merke i et "stasjonært" miljø. Er det behov for å integrere en slik teknologi i et mobilt miljø, må man gardere seg med annen teknologi, som for eksempel LAN-aksesspunkter.

10 Referanser

Fra Bluetooth Specification Vol 1.0b – Core

- [1] - Baseband Specification, s. 33 - 184
- [2] - Logical Link Control and Adaption Protocol Specification, s. 245 - 322
- [3] – Link Manager Protocol, s. 185 – 244
- [4] – RFCOMM with TS 07.10, s. 385 - 410
- [5] – Service Discovery Protocol, s. 323 - 384
- [6] – Host Controller Interface Functional Specification, s. 517 – 758
- [7] – IrDA Interoperability, s. 411 - 428

Fra Bluetooth Specification Vol. 1.0b – Profiles

- [8] – Object Push Profile, s. 329 - 354
- [9] – File Transfer Profile, s. 355 - 386
- [10] – Generic Object Exchange Profile, s. 301 - 328
- [11] – LAN Access Profile, s. 265 - 300

Fra Internett

- [12] - The Official Bluetooth Website – (www.bluetooth.com)
- [13] - Helse- og sosialdepartementet : handlingsplan for helse- og sosialpersonell 1998-2001 – (<http://odin.dep.no/shd/publ/haplan/haplan1998.html>)
- [14] - Sosial- og helsedepartementets budsjettforslag for 2000 – (<http://odin.dep.no/bud2000/sb/shd/kap-02.html>)
- [15] - Det reviderte nasjonalbudsjettet – (<http://odin.dep.no/shd/prm/2000/k2/000512.html>)
- [16] – UCSD Healthcare – (<http://health.ucsd.edu/guide/T0254.htm>)
- [17] – Digi.no, Bluetooth 1.0 er klar – gir trådløse nett for tale og data (Eirik Rossen, 29/7-1999) – (<http://www.digi.no/digi98.nsf/pub/dd81179458ero5301594568>)
- [18] – Agder IKT-senter – (<http://www.agder-ikt.hia.no>)

- [19] – Digi.no, Åpen løsning forenkler trådløs kommunikasjon (Frode Eriksen, 20/5-1998) – (<http://www.digi.no/digi98.nsf/pub/te43597023fe1743648432>)
- [20] – Intel Corporation – (<http://www.intel.com/mobile/bluetooth/index.htm>)
- [21] – Digi.no, Frost & Sullivan : Bluetooth går til himmels (Øystein Kvistad, 14/1-2000) – (<http://www.digi.no/digi98.nsf/pub/te20000114150700KVI2316472114>)
- [22] – Option International – (<http://www.option.com/techno/bluetooth02.htm>)
- [23] – Atmel Corporation – (<http://www.atmel.com/atmel/bluetooth/index.html>)
- [24] – Option International – (<http://www.option.com/techno/bluetooth05.htm>)
- [25] – Bluetooth – Tik-111.550 Seminar on multimedia – (<http://www.tcm.hut.fi/Opinnot/Tik-111.550/1999/Esitelmat/Bluetooth/bluetooth.html>)
- [26] – Symbionics Ltd. – EBDK – (<http://www.symbionics.co.uk/solutions/bluetooth/Bluetoothkit.shtml>)
- [27] – Ericsson – EBDK – (http://bluetooth.ericsson.se/ebc/implementation_kit.asp)
- [28] – USB Implementers Forum, Inc. – (www.usb.org)
- [29] – Lammert Bies (personal homepage) – (http://www.lammertbies.nl/comm/info/RS-232_specs.html)
- [30] – Meditron AS – (www.meditron.no)
- [31] – Syntrillium Software Corporation – (<http://www.syntrillium.com/cooledit>)
- [32] - Dean Jenkins and Stephen Gerred (ECG library homepage) – (<http://homepages.enterprise.net/djenkins/ecghome.html>)
- [33] – Marc Boots Ebenfiled (personal homepage) – (<http://lang.swarthmore.edu/mellon/soundnvideo/soundformats.htm>)
- [34] – MP3.com, Inc – (www.mp3.com)
- [35] – J. Matthew Miller (personal homepage) – (<http://devilnet.duke.edu/cyberia/cyberculture/mp3>)
- [36] – Centro Studi e Laboratori Telecomunicazioni S.p.A – (<http://drogo.cselt.stet.it/mpeg/tutorials.htm>)
- [37] – TechTarget.com, Inc – (<http://www.whatis.com/sql.htm>)
- [38] – EarthWeb, Inc – (<http://sysopt.earthweb.com/articles/bluetooth/index2.html>)

- [39] – The Wireless LAN Association – (www.wlana.com)
- [40] – Digi.no, Bluetooth-alliansen kan sprekke (Eirik Rossen 27/1-2000) – (<http://www.digi.no/digi98.nsf/pub/dd20000127010000ero5735713741>)
- [41] – Frost & Sullivan – (www.frost.com)
- [42] – Micromedical Industries Inc. – (www.micromed.com.au)
- [43] – Wapland, Bluetooth hjelper hjertepasienter (Marius Røstad 8/3-2000) – (<http://wapland.no/art/294.html>)
- [44] – Dell Computer Corporation – (http://www.euro.dell.com/countries/no/nor/dhs/topics/vectors_1999-blue.htm)

Fra bøker

- [45] - Fred Halsall : **Data Communications, Computer Networks and Open Systems**, 4th edition (1996), s. 69

Annet

- [46] – Teknisk Ukeblad nr. 8 (2000), s. **x**
- [47] – Riku Mettala : **Bluetooth Protocol Architecture**, white paper (1999)
- [48] – Thomas Mueller : **Bluetooth Security Architecture**, white paper (1999)
- [49] – **The Ericsson Bluetooth Development Kit**, pdf-fil (mottatt av Jonny Ervik hos Ericsson, Grimstad)
- [50] – **ROK 101 007/1 Bluetooth module**, pdf-fil (mottatt av Jonny Ervik hos Ericsson, Grimstad)
- [51] Johansen, Vegar, 1. amanuensis Høgskolen i Gjøvik. (Personlig meddelelse, 2. mai 2000). Email-adresse: vegar.johansen@hig.no
- [52] Do, Tien-Toan, Universitetet i Hannover. (Personlig meddelelse, 10. mai 2000). Email-adresse: toan@mst.uni-hannover.de
- [53] Do, Tien-Toan, Universitetet i Hannover. (Personlig meddelelse, 11. mai 2000). Email-adresse: toan@mst.uni-hannover.de
- [54] Do, Tien-Toan, Universitetet i Hannover. (newsgroup, 5. – 10 mai 2000)
Web-adresse: www.bluetooth.com
- [55] Storfjord, Monica. **Security in WAP**, [publiseres juni 2000]. Diplomoppgave ved Høgskolen i Agder, avdeling for informasjons- og kommunikasjonsteknologi.
Web-adresse: <http://siving.hia.no/ikt00/ikt6400/mstorfjo/>

[56] Bræck, Christer. **Overføring av prosessdata via WAP**. Diplomoppgave ved Høgskolen i Agder, avdeling for informasjons- og kommunikasjonsteknologi. Web-adresse: <http://siving.hia.no/ikt00/ikt6400/cbrack/>

Fra Internett – ekstra

[57] - Timothy John Weber (personal homepage) - <http://www.lightlink.com/tjweber/StripWav/WAVE.html>

[58] – PCInfobase - <http://www.pcinfobase.com/demo/indxmpeg.htm>

[59] – Knut Harald Nygaard (personal homepage) - <http://www.iu.hioslo.no/~knutn/artikler/linjekoding/linjetel.htm>

11 Vedlegg

11.1 Vedleggsliste

- 1 - Forkortelser benyttet i rapport
- 2 - Etablering av slave
- 3 - Etablering av master
- 4 - HCI-kommandoer

1 - Forkortelser benyttet i rapport

EKG – Electrocardiogram (eng : ECG)
WAP – Wireless Application Protocol
SIG – Special Interest Group
IKT – Informasjons- og kommunikasjonsteknologi
EBDK – Ericsson Bluetooth Development Kit
SW – Software
LAN – Local Area Network
ACL - Asynchronous Connectionless
SCO - Synchronous Connection-Oriented
LMP – Link Manager Protocol
LSB – Least Significant Bit
LC – Link Controller
L2CAP - Logical link control and adaption protocol
SDP - Service Discovery Protocol
HCI - Host Controller Interface
IrDA - Infrared Data Association
WAV - Windows WAVE
MP3 - MPEG-1 layer 3
MPEG - Moving Picture Expert Group
TDD – Time Division Duplex
SQL - Structured Query Language
FEC – Forward Error Correction code
ARQ - Automatic Repeat reQuest
FHSS – Frequency-Hopping Spread Spectrum

2 - Etablering av slave

Overføring av EKG-prosessdata via Bluetooth-teknologi.

Welcome to the Ericsson Bluetooth Development Kit:

```
>
! - Spooling Command "cmd r"
> cmd r
# Mon Jun 05 16:29:19 2000
#
# HCIReset - R
# Mon Jun 05 16:29:19 2000
#
# Length = 3 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 03 0c 00
# HCI_PacketOpcode          [ 2 byte/s] = 0x0C03 (3075)
# Parameter_Length         [ 1 byte/s] = 0x00 (0)
>
< HCIReset - R
# Mon Jun 05 16:29:19 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 03 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC03 (3075) (0x00000C03)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd rbs"
> cmd rbs
# Mon Jun 05 16:29:19 2000
#
# HCIReadBufferSize - RBS
# Mon Jun 05 16:29:19 2000
#
# Length = 3 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 05 10 00
# HCI_PacketOpcode          [ 2 byte/s] = 0x1005 (4101)
# Parameter_Length         [ 1 byte/s] = 0x00 (0)
>
< HCIReadBufferSize - RBS
# Mon Jun 05 16:29:19 2000
#
# Length = 13 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 0b 01 05 10 00 20 03 00 0a 00 00 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x0B (11)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0x1005 (4101) (0x00001005)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
# HC_ACL_Data_Packet_Length [ 2 byte/s] = 0x320 (800) (0x00000320)
# HC_SCO_Data_Packet_Length [ 1 byte/s] = 0x0 (0) (0x00000000)
# HC_Total_Num_ACL_Data_Packets [ 2 byte/s] = 0xA (10) (0x0000000A)
# HC_Total_Num_SCO_Data_Packets [ 2 byte/s] = 0x0 (0) (0x00000000)
>
! - Spooling Command "raw 01 05 0c 01 00"
> raw 01 05 0c 01 00
< Raw Data Transmitted
>
< HCISetEventFilter - SEF
# Mon Jun 05 16:29:19 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 05 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC05 (3077) (0x00000C05)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd wse INQUIRY_PAGE"
> cmd wse INQUIRY_PAGE
# Mon Jun 05 16:29:19 2000
#
# HCIWriteScanEnable - WSE
# Mon Jun 05 16:29:19 2000
#
# Length = 4 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 1a 0c 01 03
```

Overføring av EKG-prosessdata via Bluetooth-teknologi.

```
# HCI_PacketOpcode      [ 2 byte/s] = 0x0C1A (3098)
# Parameter_Length     [ 1 byte/s] = 0x01 (1)
# Scan_Enable          [ 1 byte/s] = INQUIRY_PAGE (3) (0x00000003)
>
< HCIWriteScanEnable - WSE
# Mon Jun 05 16:29:19 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 1a 0c 00
# HCI_PacketOpcode     [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0xC1A (3098) (0x00000C1A)
# Status                [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd wae DISABLED"
> cmd wae DISABLED
# Mon Jun 05 16:29:19 2000
#
# HCIWriteAuthenticationEnable - WAE
# Mon Jun 05 16:29:19 2000
#
# Length = 4 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 20 0c 01 00
# HCI_PacketOpcode     [ 2 byte/s] = 0x0C20 (3104)
# Parameter_Length     [ 1 byte/s] = 0x01 (1)
# Authentication_Enable [ 1 byte/s] = DISABLED (0) (0x00000000)
>
< HCIWriteAuthenticationEnable - WAE
# Mon Jun 05 16:29:19 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 20 0c 00
# HCI_PacketOpcode     [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0xC20 (3104) (0x00000C20)
# Status                [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd wvs 0x0061"
> cmd wvs 0x0061
# Mon Jun 05 16:29:20 2000
#
# HCIWriteVoiceSetting - WVS
# Mon Jun 05 16:29:20 2000
#
# Length = 5 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 26 0c 02 61 00
# HCI_PacketOpcode     [ 2 byte/s] = 0x0C26 (3110)
# Parameter_Length     [ 1 byte/s] = 0x02 (2)
# Voice_Channel_Setting [ 2 byte/s] = c 02 61 00 (97) (0x00000061)
>
< HCIWriteVoiceSetting - WVS
# Mon Jun 05 16:29:20 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 26 0c 00
# HCI_PacketOpcode     [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0xC26 (3110) (0x00000C26)
# Status                [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd sef 0x02 0x00 0x02"
> cmd sef 0x02 0x00 0x02
# Mon Jun 05 16:29:20 2000
#
# HCISetEventFilter - SEF
# Mon Jun 05 16:29:20 2000
#
# Length = 6 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 05 0c 03 02 00 02
# HCI_PacketOpcode     [ 2 byte/s] = 0x0C05 (3077)
# Parameter_Length     [ 1 byte/s] = 0x03 (3)
# Filter_Type          [ 1 byte/s] = CONNECTION_SETUP (2) (0x00000002)
# Filter_Condition_Type [ 1 byte/s] = ALL_DEVICES (0) (0x00000000)
# Condition             [ 1 byte/s] = DO_AUTO_ACCEPT (2) (0x00000002)
```


Overføring av EKG-prosessdata via Bluetooth-teknologi.

```
>
< HCISetEventFilter - SEF
# Mon Jun 05 16:29:20 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 05 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC05 (3077) (0x00000C05)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd wcat 0x2000"
> cmd wcat 0x2000
# Mon Jun 05 16:29:20 2000
#
# HCIWriteConnectionAcceptTimeout - WCAT
# Mon Jun 05 16:29:20 2000
#
# Length = 5 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 16 0c 02 00 20
# HCI_PacketOpcode          [ 2 byte/s] = 0x0C16 (3094)
# Parameter_Length         [ 1 byte/s] = 0x02 (2)
# Conn_Accept_Timeout      [ 2 byte/s] = 5120.00MS (8192) (0x00002000)
>
< HCIWriteConnectionAcceptTimeout - WCAT
# Mon Jun 05 16:29:20 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 16 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC16 (3094) (0x00000C16)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd WPT 0x3000"
> cmd WPT 0x3000
# Mon Jun 05 16:29:20 2000
#
# HCIWritePageTimeout - WPT
# Mon Jun 05 16:29:20 2000
#
# Length = 5 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 18 0c 02 00 30
# HCI_PacketOpcode          [ 2 byte/s] = 0x0C18 (3096)
# Parameter_Length         [ 1 byte/s] = 0x02 (2)
# Page_Timeout             [ 2 byte/s] = 7680.00MS (12288) (0x00003000)
>
< HCIWritePageTimeout - WPT
# Mon Jun 05 16:29:20 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 18 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC18 (3096) (0x00000C18)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd ewpcms 0x03"
> cmd ewpcms 0x03
# Mon Jun 05 16:29:20 2000
#
# HCIEricssonWritePCMSSettings - EWPCMS
# Mon Jun 05 16:29:20 2000
#
# Length = 4 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 07 fc 01 03
# HCI_PacketOpcode          [ 2 byte/s] = 0xFC07 (64519)
# Parameter_Length         [ 1 byte/s] = 0x01 (1)
# PCM_Settings             [ 1 byte/s] = PCM_MASTER_A_TRANSMIT (3) (0x00000003)
>
< HCIEricssonWritePCMSSettings - EWPCMS
# Mon Jun 05 16:29:20 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
```

Overføring av EKG-prosessdata via Bluetooth-teknologi.

```
# 0e 04 01 07 fc 00
# HCI_PacketOpcode      [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0xFC07 (64519) (0x0000FC07)
# Status                [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd rbdaddr"
> cmd rbdaddr
# Mon Jun 05 16:29:20 2000
#
# HCIReadBDADDR - RBDADDR
# Mon Jun 05 16:29:20 2000
#
# Length = 3 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 09 10 00
# HCI_PacketOpcode      [ 2 byte/s] = 0x1009 (4105)
# Parameter_Length     [ 1 byte/s] = 0x00 (0)
>
< HCIReadBDADDR - RBDADDR
# Mon Jun 05 16:29:20 2000
#
# Length = 12 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 0a 01 09 10 00 11 11 50 11 11 11
# HCI_PacketOpcode      [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x0A (10)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0x1009 (4105) (0x00001009)
# Status                [ 1 byte/s] = SUCCESS (0) (0x00000000)
# BD_ADDR               [ 6 byte/s] = 0x111150111111
>
```

3 - Etablering av master

Overføring av EKG-prosessdata via Bluetooth-teknologi.

Welcome to the Ericsson Bluetooth Development Kit:

```
>
! - Spooling Command "cmd r"
> cmd r
# Mon Jun 05 16:14:08 2000
#
# HCIReset - R
# Mon Jun 05 16:14:08 2000
#
# Length = 3 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 03 0c 00
# HCI_PacketOpcode          [ 2 byte/s] = 0x0C03 (3075)
# Parameter_Length         [ 1 byte/s] = 0x00 (0)
>
< HCIReset - R
# Mon Jun 05 16:14:08 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 03 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC03 (3075) (0x00000C03)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd rbs"
> cmd rbs
# Mon Jun 05 16:14:08 2000
#
# HCIReadBufferSize - RBS
# Mon Jun 05 16:14:08 2000
#
# Length = 3 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 05 10 00
# HCI_PacketOpcode          [ 2 byte/s] = 0x1005 (4101)
# Parameter_Length         [ 1 byte/s] = 0x00 (0)
>
< HCIReadBufferSize - RBS
# Mon Jun 05 16:14:08 2000
#
# Length = 13 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 0b 01 05 10 00 20 03 00 0a 00 00 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x0B (11)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0x1005 (4101) (0x00001005)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
# HC_ACL_Data_Packet_Length [ 2 byte/s] = 0x320 (800) (0x00000320)
# HC_SCO_Data_Packet_Length [ 1 byte/s] = 0x0 (0) (0x00000000)
# HC_Total_Num_ACL_Data_Packets [ 2 byte/s] = 0xA (10) (0x0000000A)
# HC_Total_Num_SCO_Data_Packets [ 2 byte/s] = 0x0 (0) (0x00000000)
>
! - Spooling Command "raw 01 05 0c 01 00"
> raw 01 05 0c 01 00
< Raw Data Transmitted
>
< HCISetEventFilter - SEF
# Mon Jun 05 16:14:08 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 05 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC05 (3077) (0x00000C05)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd wse INQUIRY_PAGE"
> cmd wse INQUIRY_PAGE
# Mon Jun 05 16:14:09 2000
#
# HCIWriteScanEnable - WSE
# Mon Jun 05 16:14:09 2000
#
# Length = 4 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 1a 0c 01 03
```

Overføring av EKG-prosessedata via Bluetooth-teknologi.

```
# HCI_PacketOpcode      [ 2 byte/s] = 0x0C1A (3098)
# Parameter_Length     [ 1 byte/s] = 0x01 (1)
# Scan_Enable          [ 1 byte/s] = INQUIRY_PAGE (3) (0x00000003)
>
< HCIWriteScanEnable - WSE
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 1a 0c 00
# HCI_PacketOpcode     [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode      [ 2 byte/s] = 0xC1A (3098) (0x00000C1A)
# Status              [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd wae DISABLED"
> cmd wae DISABLED
# Mon Jun 05 16:14:09 2000
#
# HCIWriteAuthenticationEnable - WAE
# Mon Jun 05 16:14:09 2000
#
# Length = 4 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 20 0c 01 00
# HCI_PacketOpcode     [ 2 byte/s] = 0x0C20 (3104)
# Parameter_Length     [ 1 byte/s] = 0x01 (1)
# Authentication_Enable [ 1 byte/s] = DISABLED (0) (0x00000000)
>
< HCIWriteAuthenticationEnable - WAE
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 20 0c 00
# HCI_PacketOpcode     [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode      [ 2 byte/s] = 0xC20 (3104) (0x00000C20)
# Status              [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd wvs 0x0061"
> cmd wvs 0x0061
# Mon Jun 05 16:14:09 2000
#
# HCIWriteVoiceSetting - WVS
# Mon Jun 05 16:14:09 2000
#
# Length = 5 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 26 0c 02 61 00
# HCI_PacketOpcode     [ 2 byte/s] = 0x0C26 (3110)
# Parameter_Length     [ 1 byte/s] = 0x02 (2)
# Voice_Channel_Setting [ 2 byte/s] = c 02 61 00 (97) (0x00000061)
>
< HCIWriteVoiceSetting - WVS
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 26 0c 00
# HCI_PacketOpcode     [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode      [ 2 byte/s] = 0xC26 (3110) (0x00000C26)
# Status              [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd sef 0x02 0x00 0x02"
> cmd sef 0x02 0x00 0x02
# Mon Jun 05 16:14:09 2000
#
# HCISetEventFilter - SEF
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 05 0c 03 02 00 02
# HCI_PacketOpcode     [ 2 byte/s] = 0x0C05 (3077)
# Parameter_Length     [ 1 byte/s] = 0x03 (3)
# Filter_Type          [ 1 byte/s] = CONNECTION_SETUP (2) (0x00000002)
# Filter_Condition_Type [ 1 byte/s] = ALL_DEVICES (0) (0x00000000)
# Condition            [ 1 byte/s] = DO_AUTO_ACCEPT (2) (0x00000002)
```

Overføring av EKG-prosessedata via Bluetooth-teknologi.

```
>
< HCISetEventFilter - SEF
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 05 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC05 (3077) (0x00000C05)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd wcat 0x2000"
> cmd wcat 0x2000
# Mon Jun 05 16:14:09 2000
#
# HCIWriteConnectionAcceptTimeout - WCAT
# Mon Jun 05 16:14:09 2000
#
# Length = 5 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 16 0c 02 00 20
# HCI_PacketOpcode          [ 2 byte/s] = 0x0C16 (3094)
# Parameter_Length         [ 1 byte/s] = 0x02 (2)
# Conn_Accept_Timeout      [ 2 byte/s] = 5120.00MS (8192) (0x00002000)
>
< HCIWriteConnectionAcceptTimeout - WCAT
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 16 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC16 (3094) (0x00000C16)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd WPT 0x3000"
> cmd WPT 0x3000
# Mon Jun 05 16:14:09 2000
#
# HCIWritePageTimeout - WPT
# Mon Jun 05 16:14:09 2000
#
# Length = 5 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 18 0c 02 00 30
# HCI_PacketOpcode          [ 2 byte/s] = 0x0C18 (3096)
# Parameter_Length         [ 1 byte/s] = 0x02 (2)
# Page_Timeout             [ 2 byte/s] = 7680.00MS (12288) (0x00003000)
>
< HCIWritePageTimeout - WPT
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 04 01 18 0c 00
# HCI_PacketOpcode          [ 1 byte/s] = 0x000E (14)
# Parameter_Length         [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode           [ 2 byte/s] = 0xC18 (3096) (0x00000C18)
# Status                    [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd ewpcms 0x03"
> cmd ewpcms 0x03
# Mon Jun 05 16:14:09 2000
#
# HCIEricssonWritePCMSSettings - EWPCMS
# Mon Jun 05 16:14:09 2000
#
# Length = 4 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 07 fc 01 03
# HCI_PacketOpcode          [ 2 byte/s] = 0xFC07 (64519)
# Parameter_Length         [ 1 byte/s] = 0x01 (1)
# PCM_Settings             [ 1 byte/s] = PCM_MASTER_A_TRANSMIT (3) (0x00000003)
>
< HCIEricssonWritePCMSSettings - EWPCMS
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
```

Overføring av EKG-prosessedata via Bluetooth-teknologi.

```
# 0e 04 01 07 fc 00
# HCI_PacketOpcode      [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0xFC07 (64519) (0x0000FC07)
# Status               [ 1 byte/s] = SUCCESS (0) (0x00000000)
>
! - Spooling Command "cmd rbdaddr"
> cmd rbdaddr
# Mon Jun 05 16:14:09 2000
#
# HCIReadBDADDR - RBDADDR
# Mon Jun 05 16:14:09 2000
#
# Length = 3 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 09 10 00
# HCI_PacketOpcode      [ 2 byte/s] = 0x1009 (4105)
# Parameter_Length     [ 1 byte/s] = 0x00 (0)
>
< HCIReadBDADDR - RBDADDR
# Mon Jun 05 16:14:09 2000
#
# Length = 12 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0e 0a 01 09 10 00 11 11 50 11 11 11
# HCI_PacketOpcode      [ 1 byte/s] = 0x000E (14)
# Parameter_Length     [ 1 byte/s] = 0x0A (10)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0x1009 (4105) (0x00001009)
# Status               [ 1 byte/s] = SUCCESS (0) (0x00000000)
# BD_ADDR              [ 6 byte/s] = 0x111150111111
>
! - Spooling Command "cmd cc 0x111111501111 DM1 R0 MANDATORY 0x0000"
> cmd cc 0x111111501111 DM1 R0 MANDATORY 0x0000
# Mon Jun 05 16:14:09 2000
#
# HCICreateConnection - CC
# Mon Jun 05 16:14:09 2000
#
# Length = 15 bytes (Dir = 1 ,where Rx=0 and Tx=1)
# 05 04 0c 11 11 50 11 11 11 08 00 00 00 00 00
# HCI_PacketOpcode      [ 2 byte/s] = 0x0405 (1029)
# Parameter_Length     [ 1 byte/s] = 0x0C (12)
# BD_ADDR              [ 6 byte/s] = 0x111150111111
# Packet_Type          [ 2 byte/s] = DM1 (8) (0x00000008)
# Page_Scan_Repetition_Mode [ 1 byte/s] = R0 (0) (0x00000000)
# Page_Scan_Mode       [ 1 byte/s] = MANDATORY (0) (0x00000000)
# Clock_Offset         [ 2 byte/s] = 0x0 (0) (0x00000000)
>
< CommandStatus - CS
# Mon Jun 05 16:14:09 2000
#
# Length = 6 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 0f 04 00 01 05 04
# HCI_PacketOpcode      [ 1 byte/s] = 0x000F (15)
# Parameter_Length     [ 1 byte/s] = 0x04 (4)
# Status               [ 1 byte/s] = SUCCESS (0) (0x00000000)
# Num_HCI_Command_Packets [ 1 byte/s] = 0x1 (1) (0x00000001)
# Command_Opcode       [ 2 byte/s] = 0x405 (1029) (0x00000405)
>
< ConnectionComplete - CC
# Mon Jun 05 16:14:11 2000
#
# Length = 13 bytes (Dir = 0 ,where Rx=0 and Tx=1)
# 03 0b 00 01 00 11 11 50 11 11 11 01 00
# HCI_PacketOpcode      [ 1 byte/s] = 0x0003 (3)
# Parameter_Length     [ 1 byte/s] = 0x0B (11)
# Status               [ 1 byte/s] = SUCCESS (0) (0x00000000)
# Connection_Handle    [ 2 byte/s] = 0x1 (1) (0x00000001)
# BD_ADDR              [ 6 byte/s] = 0x111150111111
# Link_Type            [ 1 byte/s] = ACL (1) (0x00000001)
# Encryption_Mode      [ 1 byte/s] = DISABLED (0) (0x00000000)
>
```

4 - HCI-kommandoer

Overføring av EKG-prosessdata via Bluetooth-teknologi.

```
# raw <n bytes>
# Første byte må være en transport byte, skriver du dette feil blir ikke pakken transportert

# 01 - Command
# 02 - ACL Data
# 03 - SCO Data
# 04 - Event

# I      - HCI_Inquiry                [args = 3]
# IC     - HCI_Inquiry_Cancel        [args = 0]
# PIM    - HCI_Periodic_Inquiry_Mode [args = 5]
# EPIM   - HCI_Exit_Periodic_Inquiry_Mode [args = 0]
# CC     - HCI_Create_Connection     [args = 5]
# D      - HCI_Disconnect            [args = 2]
# ASCOC  - HCI_Add_SCO_Connection    [args = 2]
# ACR    - HCI_Accept_Connction_Request [args = 2]
# RCR    - HCI_Reject_Connection_Request [args = 2]
# LKRR   - HCI_Link_Key_Request_Reply [args = 2]
# LKNR   - HCI_Link_Key_Negative_Request_reply [args = 1]
# PINCRR - HCI_PIN_Code_Request_Reply [args = 3]
# PINCNR - HCI_PIN_Code_Negative_Request_Reply [args = 1]
# CCPT   - HCI_Change_Connection_Packet_Type [args = 2]
# AR     - HCI_Authentication_Requested [args = 1]
# SCE    - HCI_Set_Connection_Encryption [args = 2]
# CCLK   - HCI_Change_Connection_Link_Key [args = 1]
# MLK    - HCI_Master_Link_Key       [args = 1]
# RNR    - HCI_Remote_Name_Request    [args = 4]
# RRSF   - HCI_Read_Remote_Supported_Features [args = 1]
# RRVI   - HCI_Read_Remote_Version_Information [args = 1]
# RCO    - HCI_Read_Clock_Offset      [args = 1]
# HM     - HCI_Hold_Mode               [args = 3]
# SM     - HCI_Sniff_Mode              [args = 5]
# ESM    - HCI_Exit_Sniff_Mode         [args = 1]
# PM     - HCI_Park_Mode               [args = 3]
# EPM    - HCI_Exit_Park_Mode          [args = 1]
# QOSS   - HCI_QOS_Setup               [args = 7]
# RD     - HCI_Role_Discovery          [args = 1]
# SR     - HCI_Switch_Role             [args = 2]
# RLPS   - Read_Link_Policy_Settings   [args = 1]
# WLPS   - Write_Link_Policy_Settings  [args = 2]
# RLVI   - HCI_Read_Local_Version_Information [args = 0]
# RLSF   - HCI_Read_Local_Supported_Features [args = 0]
# RBS    - HCI_Read_Buffer_Size        [args = 0]
# RCC    - HCI_Read_Country_Code       [args = 0]
# RBDADDR - HCI_Read_BD_ADDR           [args = 0]
# RFCC   - HCI_Read_Failed_Contact_Counter [args = 1]
# RFCC   - HCI_Reset_Failed_Contact_Counter [args = 1]
# GLQ    - HCI_Get_Link_Quality        [args = 1]
# RRSSI  - HCI_Read_RSSI               [args = 1]
# RLM    - HCI_Read_Loopback_Mode      [args = 0]
# WLM    - HCI_Write_Loopback_Mode     [args = 1]
# EDUTM  - HCI_Enable_Device_Under_Test_Mode [args = 0]
# SEM    - HCI_Set_Event_Mask          [args = 1]
# R      - HCI_Reset                   [args = 0]
# SEF    - HCI_Set_Event_Filter         [args = 3]
# F      - HCI_Flush                   [args = 1]
# RPINT  - HCI_Read_PIN_Type           [args = 0]
# WPINT  - HCI_Write_PIN_Type          [args = 1]
# CNUK   - HCI_Create_New_Unit_Key     [args = 0]
# RSLK   - HCI_Read_Stored_Link_Key    [args = 2]
# WSLK   - HCI_Write_Stored_Link_Key   [args = 3]
# DSLK   - HCI_Delete_Stored_Link_Key  [args = 2]
# CLN    - HCI_Change_Local_Name       [args = 1]
# RCAT   - HCI_Read_Connection_Accept_Timeout [args = 0]
# WCAT   - HCI_Write_Connection_Accept_Timeout [args = 1]
# RPT    - HCI_Read_Page_Timeout       [args = 0]
# WPT    - HCI_Write_Page_Timeout      [args = 1]
```

Overføring av EKG-prosessdata via Bluetooth-teknologi.

```
# RSE - HCI_Read_Scan_Enable [args = 0]
# WSE - HCI_Write_Scan_Enable [args = 1]
# RPSA - HCI_Read_PageScan_Activity [args = 0]
# WPSA - HCI_Write_PageScan_Activity [args = 2]
# RISA - HCI_Read_InquiryScan_Activity [args = 0]
# WISA - HCI_Write_InquiryScan_Activity [args = 2]
# RAE - HCI_Read_Authentication_Enable [args = 0]
# WAE - HCI_Write_Authentication_Enable [args = 1]
# REM - HCI_Read_Encryption_Mode [args = 0]
# WEM - HCI_Write_Encryption_Mode [args = 1]
# RCOD - HCI_Read_Class_Of_Device [args = 0]
# WCOD - HCI_Write_Class_Of_Device [args = 1]
# RVS - HCI_Read_Voice_Setting [args = 0]
# WVS - HCI_Write_Voice_Setting [args = 1]
# RAFT - HCI_Read_Automatic_Flush_Timeout [args = 1]
# WAF - HCI_Write_Automatic_Flush_Timeout [args = 2]
# RNBR - HCI_Read_Num_Broadcast_Retransmissions [args = 0]
# WNBR - HCI_Write_Num_Broadcast_Retransmissions [args = 1]
# RHMA - HCI_Read_Hold_Mode_Activity [args = 0]
# WHMA - HCI_Write_Hold_Mode_Activity [args = 1]
# RTPL - HCI_Read_Transmit_Power_Level [args = 2]
# SHCTHFC - HCI_Set_Host_Controller_To_Host_Flow_Control [args = 1]
# HBS - HCI_Host_Buffer_Size [args = 4]
# HNOCP - HCI_Host_Number_Of_Completed_Packets [args = 3]
# RLST - HCI_Read_Link_Supervision_Timeout [args = 1]
# WLST - HCI_Write_Link_Supervision_Timeout [args = 2]
# RNOSIAC - HCI_Read_Number_Of_Supported_IAC [args = 0]
# RCIACLAP - HCI_Read_Current_IAC_LAP [args = 0]
# WCIACLAP - HCI_Write_Current_IAC_LAP [args = 2]
# RPSM - HCI_Read_Page_Scan_Period_Mode [args = 0]
# WPSM - HCI_Write_Page_Scan_Period_Mode [args = 1]
# RPSM - HCI_Read_Page_Scan_Mode [args = 0]
# WPSM - HCI_Write_Page_Scan_Mode [args = 1]
# EWHWR - HCI_Ericsson_Write_HW_Register [args = 3]
# ERHWR - HCI_Ericsson_Read_HW_Register [args = 1]
# ERRI - HCI_Ericsson_Read_Revision_Information [args = 0]
# EWPCMS - HCI_Ericsson_Write_PCM_Settings [args = 1]
# EWST - HCI_Ericsson_Write_Scan_Triggering [args = 1]
# ESUARTBR - HCI_Ericsson_Set_UART_Baud_Rate [args = 1]
# EWDPIN - HCI_Ericsson_Write_Default_PIN [args = 2]
# EWIC - HCI_Ericsson_Write_I2C [args = 3]
# EETM - HCI_Ericsson_Enter_Test_Mode [args = 1]
# ETC - HCI_Ericsson_Test_Control [args = 9]
# EAUX - HCI_Ericsson_AUX1 [args = 2]
# EST - HCI_Ericsson_Self_Test [args = 1]
# EWXOT - HCI_Ericsson_Write_XO_Trim [args = 1]
# ERM - HCI_Ericsson_Read_Memory [args = 1]
# EWM - HCI_Ericsson_Write_Memory [args = 2]
# EWRSSIC - HCI_Ericsson_Write_RSSI_Calibration [args = 1]
# EWCC - HCI_Ericsson_Write_Country_Code [args = 1]
# EDC - HCI_Ericsson_Disable_Commands [args = 0]
# CCNO - CommandCompleteNoOperation [args = 1]
>
```