CODEN:LUTEDX/(TEIE-3063)/1-70/(2016)

Industrial Electrical Engineering and Automation

Utvärdering och design av trådlös kommunikation mellan smart dosett och smartphone

- En undersökning och implementation av trådlös kommunikation åt företaget Finn Medicinen AB

Jonas Voigt
Kristoffer Voigt

Division of Industrial Electrical Engineering and Automation
Faculty of Engineering, Lund University

Utvärdering och design av trådlös kommunikation mellan smart dosett och smartphone

- En undersökning och implementation av trådlös kommunikation åt företaget Finn Medicinen AB

LUNDS UNIVERSITET
Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg
Industriell elektroteknik och automation

Examensarbete:
Jonas Voigt
Kristoffer Voigt

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige

Lunds universitet
Lund 2016

Sammanfattning

En produkt som har ett behov av en trådlös kommunikation mellan sig själv och Smartphone utvecklas och ett lämpligt protokoll med en godtycklig modul ska väljas och utvärderas.

Hur bör mjukvaran se ut? Vilka funktioner bör finnas tillgängliga?

Modulen och dess protokoll ska också testas i så kallade fälttester.

För protokoll diskuterades ZigBee, Z-Wave, Bluetooth Low Energy, WIFI och vanligt 3G/4G med avseende på punkterna energiförbrukning, räckvidd, överföringshastighet, behovet av en mellanliggande basstation och hur vanlig tekniken är. Ur produktens synpunkt ses dessa i ordning: behov av basstation -> energiförbrukning -> räckvidd -> teknikens utbredning -> överföringshastighet.

Bluetooth Low Energy valdes framför allt på grund av etableringen inom Smartphone industrin och att en basstation inte behövs. Energiförbrukning talar väl för ZigBee och Z-Wave medan WIFI och 3G/4G har väldigt bra överföringshastighet och täckning.

I brist på tid valdes endast protokollet Bluetooth Low Energy (BLE) att implementeras i ett chip, PSoC BLE, från Cypress Semiconductors.

Under projektets gång fördes en diskussion med Finn Medicinen över produktens funktion och behov vilket omsattes till ett flödesschema. Utan färdig produkt kunde inte alla funktioner implementeras.

Modulen PSoC BLE från Cypress Semiconductors ansågs ha goda möjliga att uppfylla Finn Medicinens behov. Fälttesterna visade att denna enhet, med denna implementation, klarade flera rimliga fall väl.

Nyckelord

Bluetooth Low Energy, PSoC Creator, Cypress Semiconductors, Finn Medicinen, Smartphone

Abstract

A product with the need for a wireless communication between itself and a smartphone will be developed. A suitable wireless protocol and module will be chosen and evaluated.

How should the software be designed? What functions should be available?

In the end the module and software will be tested in the field to see how it works in reality.

For the choosing of protocol, the following will be discussed; ZigBee, Z-wave, Bluetooth Low Energy, WIFI and the common mobile network. The main point of the discussion will be energy consumption, range, transfer rate and the need for additional hardware in order to work with a smartphone.

Bluetooth Low Energy was chosen mainly because the wide use of the product with in the smartphone market. Due to the fact that Bluetooth Low Energy already is integrated in most of the new smartphones no additional hardware was needed in order to make the communication work. Both ZigBee and Z-wave also had a very low power consumption and both WIFI and the mobile network had a great data transfer rate.

A flowchart was designed to show the need of functions within the software and the task of the communication between the product and the smartphone. From the flowchart all the functions needed was defined. Unfortunately, the company's own product was at this time not completed. Without it, not all functions were possible to implement.

The module PSoC BLE from Cypress Semiconductors was believed to be able to perform as Finn Medicinen required. The field tests proved that this module, with this implementation, handled several reasonable situations well.

Keywords

Bluetooth Low Energy, PSoC Creator, Cypress Semiconductors, Finn Medicinen, Smartphone

Förord

Som två bröder inom samma utbildning var ett delat examensarbete av stort intresse. Det har varit väldigt intressant att skriva ett examensjobb för ett företag i sin uppstartningsfas och få delta i, och se hur denna process går till. Examensjobbet självt har varit högst relevant för vår utbildning inom automation där informationsflödet ofta är av högsta vikt.

Vi vill tacka Finn Medicinen för denna möjlighet; deras önskan att se sin produkt men även deras hjälp i att vårat examensarbete skall gå bra för oss. Personligt tack till VD Ulrika Landin och delägare Niklas Landin som även varit handledare åt oss.

All hjälp vi fått ifrån LTH och ingenjörsinstitutionerna har varit till stor hjälp och vi vill utrycka särskilda tack till vår handledare Christian Nyberg och vår examinator Mats Lilja.

Innehållsförteckning

Innehåll

Sammanfattning iii
Nyckelord iii
Abstract iv
Keywords iv
Förord v
Innehållsförteckning vi
1 Inledning 1
1.1 Bakgrund 1
1.2 Syfte och mål 1
1.3 Problemformulering 2
1.3.1 Vad Finn Medicinen vill ha 2
1.3.2 Krav 3
1.4 Avgränsningar 3
2 Teknisk bakgrund 4
2.1 Trådlösa protokoll 4
2.1.1 Bluetooth Low Energy 4
2.1.2 Z-Wave 4
2.1.3 WIFI 4
2.1.4 ZigBee 5
2.1.5 Mobilnät 5
2.2 Utvidgad Bluetooth bakgrund 6
2.2.1 Bluetooth 6
2.2.2 Bluetooth Low Energy 6
2.2.2.1 Inledning 6
2.2.2.2 Övergripande struktur 7
2.2.2.3 GAP och GATT 7
2.2.2.4 Bluetooth SIG 9
2.3 Utvecklingskit från Cypress Semiconducters 10
2.3.1 PSoC BLE 4200 11
2.3.2 PRoC BLE 4200 12
2.3.3 Sleep mode 13
2.4 PSoC Creator 15
2.5 CySmart 15

3 Metod och analys 16
3.1 Inlärningsprocess 16
3.1.1 Implementation av BLE med färdigt kit 16
3.2 Lösningen 17
3.3 Mjukvaran 19
3.3.1 Topdesignen 19
3.3.2 Main.c 20
3.3.3 UpdateGateDatabase 22
4 Analys 25
4.1 Utvärdering av trådlös kommunikation 25
4.2 Vilken kommunikation 25
4.2.1 Hastighet 25
4.2.2 Räckvidd 25
4.2.3 Bandbredd 25
4.2.4 Batteriförbrukning 25
4.2.5 Säkerhet 25
4.2.6 Färdiga moduler 26
4.2.7 Etablerat märke 26
4.2.8 Basstation 26
4.3 Kommunikationsprotokoll 26
4.4 Tankar och funderingar på att välja modul och processor själv 26
4.5 Val av färdigt kit 27
4.6 Tester 27
4.6.1 Varför testa? 27
4.6.1.1 Dosboken 27
4.6.1.2 Programvara 27
4.6.1.3 Förväntad funktionalitet 28
4.6.1.4 Genomförande 28
4.7 Källkritik 29
5 Resultat 30
5.1 Modern villa (markplan) – Bra placering av sändare 30
5.2 Modern villa (ovanvåning) – Bra placering av sändare 31
5.3 Modern villa (markplan) – Dålig placering av sändare 32
5.4 Modern villa (ovanvåning) – Dålig placering av sändare 33
5.5 Lägenhet (Våning 7) - Bra placering av sändare 34
5.6 Lägenhet (Våning 7) – Dålig placering av sändare 35

5.7 Gammal villa (markplan) – Bra placering av sändare 36
5.8 Gammal villa (Källarvåning) – Bra placering av sändare 37
5.9 Gammal villa (Markplan) – Dålig placering av sändare 38
5.10 Gammal villa (Källarvåning) – Dålig placering av sändare 39
5.11 Utvärdering av testresultat 40
5.11.1 Mjukvaran 40
5.11.1.1 Kölistan 40
5.11.1.2 Anslutning 40
5.11.1.3 Realtidsklockan 40
5.11.2 Fältstudien 41
5.12 Gammal villa 41
5.12.1 Bra placering 41
5.12.1.1 Ovanvåning 41
5.12.1.2 Källarvåning 41
5.12.2 Dålig placering 41
5.12.2.1 Ovanvåning 42
5.12.2.2 Källarvåning 42
5.13 Modern villa 42
5.13.1 Bra placering 42
5.13.1.1 Markplan 42
5.13.1.2 Ovanvåning 42
5.13.2 Dålig placering 42
5.13.2.1 Markplan 42
5.13.2.2 Ovanvåning 43
5.14 Lägenhet 43
5.14.1 Bra placering 43
5.14.1.1 Våning 7 43
5.14.2 Dålig placering 43
5.14.2.1 Våning 7 43
5.14.3 Felkällor 43
6 Slutsats 44
6.1 Vilka aspekter bedöms vara viktiga för produktens trådlösa kommunikation? 44
6.2 Vilket protokoll vore lämpligast för Finn Medicinens produkt? 44
6.3 Bör ett färdigt utvecklingskit eller eget val av mikroprocessor och modul väljas? 44
6.4 Hur skulle en implementation av mjukvaran se ut? 45
6.5 Vilka problem har uppstått på grund av avsaknad av företagets färdiga produkt? 45

6.6 Hur går man tillväga för att ta fram en sådan implementation när arbetsmiljön är okänd? 45
6.7 Hur väl kommer en implementation med det valda protokollet hantera normala situationer? 45
6.8 Vilka avstånd kan företaget vänta sig från sändaren? 46
6.9 Har uppgiften klarats? 46
6.10 Framtida utvecklingsmöjligheter och optimeringar 47
6.10.1 Egengjord applikation 47
6.10.2 Flera tester 47
6.10.3 Bättre signalstyrka 48
6.10.4 Bättre kölista 48
6.10.5 Batterioptimering 48
6.10.6 Utökad felhantering 48
6.10.7 Analog till digital 48
7 Terminologi 49
8 Referenser 50
9 Bilagor 52
9.1 Källkod 52
9.1.1 Main.c 52
9.1.2 Main.h 56
9.1.3 updateGattDatabase.c 57
9.1.4 updateGattDatabase.h 61

1 Inledning

1.1 Bakgrund

Finn Medicinen Scandinavia AB (556879-35) är ett företag som startades 2012. På den tiden utvecklade företaget en internetbaserad tjänst som lät privatkunder söka efter aktuellt lagersaldo av ett visst läkemedel hos ett specifikt apotek. År 2014 började företaget att titta på andra idéer inom e-hälsa eller ”connected health”. En idé som kläcktes var en smart uppkopplad version av den klassiska dosetten. Finn Medicinen utvecklar idag en avläsare som elektroniskt kan läsa av och registrera innehåll och förändringar i en läkemedelsbehållare, en så kallad Dosett. Denna information skall kommuniceras till smartphone där en applikation använder informationen för att stötta läkemedelsanvändningen med påminnelser, uppmuntran och dokumentation. Företaget har idag inga andra produkter på marknaden och det fokuseras just nu enbart på att komma igång med den smarta dosetten, som fått namnet Dosboken. Produkten är i prototypfas och uppdraget i examensarbetet är att ta fram förslag på en trådlös kommunikationslösning mellan hårdvaran och smartphone. I uppdraget ingår också att ta fram en fungerande mjukvara till lösningen.
Finn Medicinen har för tillfället sitt kontor i Malmös hamn nära Malmö högskola vid en så kallad företagsinkubator vid namn MINC. Genom MINC finns tillgång till nätverk med kompetens kring produktutveckling i tidig fas. Nätverket skall också utnyttjas vid examensarbetet för att få en inblick i möjligheter och utmaningar i startup-företags förutsättningar.

1.2 Syfte och mål

Syftet med produkten som ska utvecklas är att underlätta för de personer som är inblandade i privatpersoners läkemedelsanvändning. Produkten är tänkt att fungera som stöd för både patienten själv men även för nära anhöriga och för vårdpersonal. Patienten ska med hjälp av produkten kunna känna sig säkrare på att inte glömma sitt läkemedelsintag. Anhöriga ska också kunna känna sig trygga med att läkemedlen tas i tid. För vårdens skull ska det kunna innebära en bekräftelse på att läkemedel tagits eller inte tagits. Vården kan då i bättre uträckning förstå varför en patient mår bättre eller sämre.

För att hjälpa Finn Medicinen att hitta en bra lösning på detta problem kommer det i detta arbete att tittas på lämpliga trådlösa överföringsprotokoll, lämpliga moduler och mikroprocessorer och i slutändan göra riktiga fälttester för att visa på hur bra kommunikationen kommer att fungera.

Vi kommer med detta examensarbete utföra följande moment:

1. Titta på olika lämpliga kommunikationsmetoder för att sända data trådlöst
2. Argumentera för den metod som verkar lämpligast
3. Utveckla en mjukvarulösning för vald metod
4. Redogöra för hur inlärningsprocessen gått till
5. Dokumentera mjukvaran och se till att den med största möjliga enkelhet kan överlämnas till tredje part för vidare utveckling av total prototyp

6. Ta fram ett testunderlag för att kunna visa på att mjukvaran fungerar och hur bra den fungerar i olika miljöer
7. Diskutera kring vilka metoder som använts och vilka resultat som erhållits

1.3 Problemformulering

1.3.1 Vad Finn Medicinen vill ha

Finn Medicinen vill ha ett sätt att förmedla information från företagets produkt Dosboken. Dosboken är en digitaliserad dosett designat för att hjälpa människor att komma ihåg att ta sina läkemedel. Med hjälp av ett omslag för att hålla en dosett på plats och med placerade sensorer under dosetten kan man läsa av dosettens innehåll och på så sätt se om piller blivit tagna vid aktuella tidpunkter. För att behandla den data som kommer från dosetten vill man förmedla denna till en smartphone som stödjer Bluetooth 4.0 eller senare. Konceptet illustreras i figur 1.

Figur 1 - Produktillustration

För att beskriva behovet och definiera produkten har företaget skapat ett kravdokument. I detta kravdokument finns punkter som beskriver den trådlösa kommunikationens behov. Hela lösningen går som sagt ut på att ta sensorvärden från dosetten och förmedla dem till en applikation på en smartphone.

För att enklare i slutändan kunna svara på om lösningen som tagits fram och uppfyller målen för detta examensarbete bör problem brytas ner i frågeställningar. Dessa kan enkelt besvaras i slutsatsen.

1. Vilket protokoll vore lämpligast för Finn Medicinens produkt?
2. Vilka aspekter bedöms vara viktiga för produktens trådlösa kommunikation?
3. Bör ett färdigt utvecklingskit eller eget val av mikroprocessor och modul väljas?
4. Hur skulle en implementation av mjukvaran se ut?
5. Vilka problem kan uppstå på grund av att varken applikation eller hårdvara finns att tillgå?
6. Hur går man tillväga för att ta fram en sådan implementation när tekniken, programspråket och utvecklingsverktygen är okända?
7. Hur väl kommer en implementation med det valda protokollet hantera normala situationer?
8. Vilket är det största möjliga avstånd mellan dosett och smartphone vid vilket kommunikation är möjligt?

1.3.2 Krav

Kraven från företaget som berörde kommunikationen såg ut som följande:

1. Sändningen ska ske via Bluetooth Low Energy
2. Sändningen ska vara på formen [Sensorvärden][Tidsstämpel]
   Till exempel [01010010010101001000100101010][160424135223]
   Dvs [28 sensorer][ååååmmddhhmmss]
3. Batteristatus ska också kunna begäras

Dessa krav samt den problemformulering som gjorts resulterade i en konkret lista med problem som behövde hanteras;

- Inlärning av nytt system (PSoC Creator)
- Definition av flödesschema över lösningen
- Definition och insamling av information över viktiga aspekter för ett trådlöst protokoll
- Definition av fälttester och utförande av dessa

1.4 Avgränsningar

Finn Medicinens idé är att läsa av facken i en dosett för att sedan med hjälp av Bluetooth sända över data för vidare behandling i iOS applikation.
Dessa tre steg är väldigt närliggande och för att visa på vad vår uppgift är så behövs vissa avgränsningar:

1. Sensorvärdena från Dosboken sköter den/de som utvecklar sensorriggen. Endast en funktion som bearbetar data och förbereder för överföring med BLE behövs
2. Företaget utvecklar själva en smartphone applikation
   Om deras applikation inte blir klar ska endast tester mot en egen utvecklad Android applikation alternativt CySmart PC applikation med tillhörande Dongel utföras

2 Teknisk bakgrund

I detta kapitel presenteras teknisk bakgrund över de olika trådlösa protokollen som undersökts samt detaljerad information om BLE och utvecklingskitet som valt att användas.

2.1 Trådlösa protokoll

I slutet på kapitel 8.1 sammanställs data över protokollen i tabell 1.

2.1.1 Bluetooth Low Energy

Bluetooth Low Energy, Bluetooth Smart eller bara BLE är ett relativt nytt protokoll för trådlös kommunikation. Protokollet sänder på det olicensierade nätet 2,4-2,5 Ghz. I protokollet finns inbyggda funktioner för att på ett säkert sätt kunna skicka information. BLE erbjuder en sändningshastighet på 270kb/s. Viktiga fördelar är protokollets förmåga att kunna kommunicera direkt med dagens smartphones. BLE är utvecklat för att minska energiförbrukningen, framförallt gentemot sin föregångare Bluetooth och för att kunna konkurrera med andra energisnåla protokoll.

2.1.2 Z-Wave

Z-Wave är ett MESH-nätverk där noder kan vidarebefordra information om en signal inte är avsedd för just denna apparat. Detta möjliggör enkel utvidgning till större ytor om sådant skulle behövas och löses nästan automatiskt när ett hem utnyttjar flera apparater och en enskild nod har upp till 100m räckvidd (vid optimala förhållanden). Z-Wave kan skicka data på upp mot 100Kb/s, arbetar på under 1Ghz (868.42 MHz för Europa) och använder sig av liknande kryptering som banköverföring över internet. [1]

2.1.3 WIFI

WIFI behöver en nod, eller router, som olika klienter kan ansluta mot. Denna nod tillåter sedan kommunikation med andra klienter anslutna till routern, framförallt internet i sin helhet. WIFI jobbar på bandbredden 2,4GHz och på senare tid även 5GHz och WIFI är väl utvecklat för höga hastigheter vilket leder till hög energiförbrukning.
I många avseende är WIFI svårdefinierat på grund av dess stora (och spridda) utveckling, speciellt med exakta siffror på exempelvis hastighet, räckvidd och säkerhet.
Säkerhet finns tillgängligt genom utnyttjande av WPA2.[2][3]

2.1.4 ZigBee

ZigBee är en konkurrent till Z-Wave och är en standard för trådlös kommunikation. Plattformarna är helt trådlösa och energisnåla. De har stöd för så kallad MESH-nätverk och hela tekniken är gjord för att vara mycket lätt att installera.

I Europa sänder ZigBee på 868MHz och har en överföringshastighet på 20kbit/s. Den maximala räckvidden ligger på upp till 100 meter. ZigBee erbjuder verktyg för att säkerställa en säker kommunikation. Deras säkerhet bygger på industristandardeden AES-128 [4]

2.1.5 Mobilnät

Generellt sett skulle något av det vanliga mobilnätet GSM, 3G eller 4G kunna utföra uppgiften.

Samtliga svenska operatörer använder sig av 2600Mhz bandet och har max överföringshastighet på 300Mb/s för 4G[5]. Räckvidden för dessa nät kan anses som internationella även om täckningen kan vara dålig långt ifrån bebyggda områden*.

	BLE	**Z-Wave**	**ZigBee**	**Wifi**	**4G**
Hastighet	270kbit/s	<100Kbit/s	<20kbit/s	Varierar	300Mb/s
Räckvidd	100m	100m	100m	Varierar	Världen
Bandbredd	2400Mhz	868.42 MHz(Sve)	868MHz(EU)	2400MHz (Huvudsakligen)	2600MHz (Sve)
Batteriförbrukning	Låg	Låg	Låg	Hög	Hög
Etablering	Hög	Medel	Medel	Hög	Hög
Säkerhet	Tillgänglig	Tillgänglig	Tillgänglig	Tillgänglig	Tillgänglig
Anslutning direkt	JA	NEJ	NEJ	JA	JA

Tabell 1 – Översikt av markörer som är viktiga

För mer information se [6]

2.2 Utvidgad Bluetooth bakgrund

2.2.1 Bluetooth

Bluetooth skapades 1994 som ett alternativ för trådlös radiokommunikation. Protokollet utnyttjar mindre tillfälliga nätverk som kallas ad-hocs, eller mer specifikt piconets. Dessa nät kräver ingen infrastruktur likt en router eller annan sändare. Noderna etablerar sig som kontrollnoder (master node) eller mottagarnoder (slave node). Dessa noder kan sedan utbyta information mellan varandra. Ett pico-nät kan ha mellan 2 och 8 noder varav en måste ha rollen som kontrollnod. Flera noder kan anslutas och befinna sig i ett passivt tillstånd där deras anslutning snabbt kan upprättas genom att hoppa över anslutning- och synkroniseringstiden. För den klassiska Bluetooth finns två lägen för dataöverföring, *Basic Rate* och *Enhanced Data Rate*.[7]

2.2.2 Bluetooth Low Energy

2.2.2.1 Inledning

Bluetooth Low Energy, som härefter kallas BLE, är en light-version av det klassiska Bluetooth. BLE utvecklades från början av Nokia under namnet Wibree. Det var till en början bara ett in-houseprojekt hos Nokia men blev så småningom adopterat av Bluetooth Special Interestgroup.

Det finns många varianter av trådlösa protokoll som kan användas för att skapa trådlös kommunikation. Men vad som gör BLE så speciellt är att det är enkelt att skapa produkter som kan kommunicera med alla moderna mobila plattformar. Både iOS, Android och Windows Phones.

Kompatibla system innefattar:

- iOS5+ (helst iOS7+)
- Android 4.3+
- Apple OS X 10.6+
- Windows 8+ (XP, Vista and 7 stöder bara Bluetooth 2.1)
- GNU/Linux Vanilla BlueZ 4.93+

För mer information se [8]

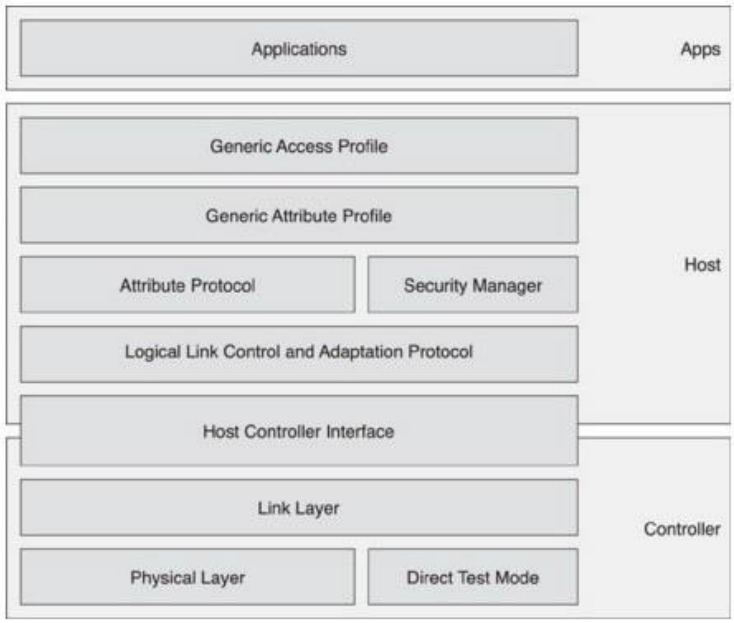
Om man ser till klassisk Bluetooth så är BLE ämnat att ge en markant minskning i strömförbrukning, se tabell 2 i kapitel 8.3.3. BLE riktar sig till produkter vars funktion behöver vara strömsnåla. Bluetooth SIG, se kapitel 7.2.2.6, har nu i sin specifikation av BLE tagit fram standardprofiler för ett antal väl valda områden som är tänkta att använda BLE. Det finns färdiga standardprofiler inom

- Health Care
- Sports and fitness
- Internet Connectivity
- HID (Human Interface device)
- Proximity
- Alerts and Time
- Battery

BLE är inte bakåtkompatibelt.
Idag är det mest vanligt att till exempel smarta mobiler kommer utrustade med hårdvara och mjukvara som stödjer BLE. År 2018 räknar SIG med att 90% av alla smarta telefoner kommer ha BLE-möjlighet.

2.2.2.2 Övergripande struktur

Tittar man på Figur 2 ser man att den innehåller tre kategorier; Controller, Host och Apps. I Host kategorin ligger de speciellt viktiga underkategorierna Generic Access Profile (GAP) och Generic Attribute Profile (GATT).



Figur 2 – BLE Arkitekturen - Från boken Bluetooth Low Energy The Developer's Handbook

Controller motsvarar de analoga och digitala komponenterna samt hårdvara som sköter sändningen. Controller-gränssnittet är den del som sköter det faktiska gränssnittet mot omvärlden. Under Host ligger de två viktiga underkategorierna GAP och GATT.

För mer information se [9].

2.2.2.3 GAP och GATT

För att förstå och konfigurera anslutningar och sändningar med BLE behöver man förstå de två grundläggande begreppen; GAP och GATT

GAP står för Generic Access Profile och är den del som hanterar anslutningar och annonsering. Det är alltså GAP som sköter synligheten mot omvärlden för produkten. Den bestämmer även hur två enheter får eller inte får kommunicera med varandra.
GAP kan anta två olika roller hos enheten beroende på vad den ska användas till, perifer eller central.
Perifera enheter är små lågenergikonsumerande enheter är ansluta till en större och mer kraftfull enhet. Det kan till exempel vara en hjärtövervakare (EKG) eller en närhetssensor.

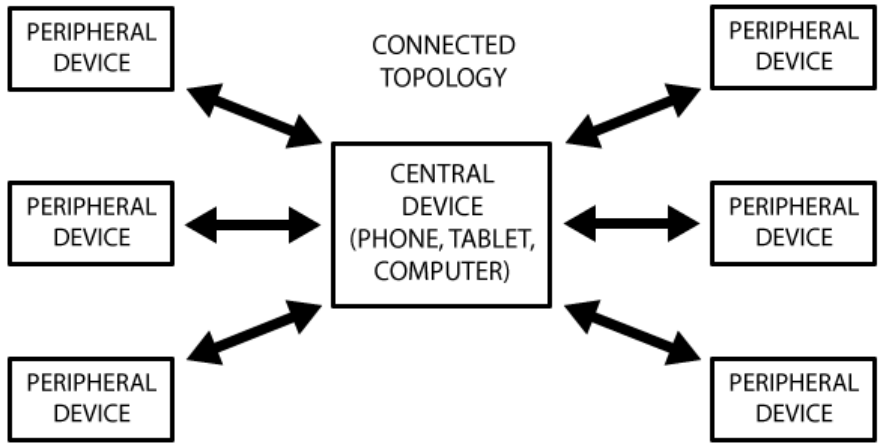
En central enhet är ofta en mobiltelefon eller surfplatta och har då oftast mycket större kapacitet när det gäller både batteri och minne.

För mer information se [10]

GATT står för Generic Attribute Profile och denna profil definierar hur två enheter med BLE skickar data fram och tillbaka mellan varandra. Detta görs genom ett koncept som kallas ”Services” och ”Characteristics”.

GATT börjar inte jobba förrän anslutningsprocessen hos GAP är avklarad. Det är också viktigt att veta att när man använder GATT så är anslutningen exklusiv. Det betyder att en perifer enhet bara kan vara ansluten till en central enhet åt gången.

Om man tittar på Figur 3 ser man hur anslutningar fungerar i GATT.

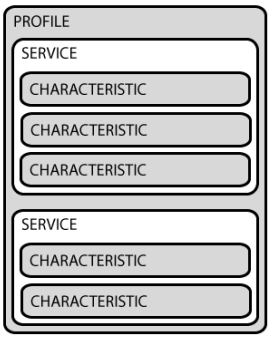


*Figur 3 - Connected Topology – **[11]***

Skulle en perifer enhet vilja prata med en annan perifer enhet måste detta gå en igenom en central enhet.

I GATT används server/klient modellen. Den perifera enheten kallas GATT Server och den centrala enheten kallas GATT klient. Vid en anslutning föreslår GATT server ett intervall där kontroll av anslutningen ska göras. Vid detta intervall kommer det förfrågas om en ny anslutning för att se om ny data finns tillgänglig.

GATT transaktionerna är baserade på objekten ”Profiles”, ”Services” och ”Characteristics”.



*Figur 4 - GATT uppbyggnad – **[12]***

Profilerna är fördefinierade av Bluetooth SIG och kan exempelvis vara:

- Battery Service
- Blood Pressure Profile
- CurrentTime Service

Full lista finns här:
https://developer.bluetooth.org/TechnologyOverview/Pages/Profiles.aspx

”Service” används för att bryta upp datamängder i logiska enheter men data lagras i olika ”Characteristics”.

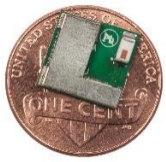
För mer information se [13]

2.2.2.4 Bluetooth SIG

Bluetooth Bluetooth Special Interest Group eller SIG är ett fristående och icke vinstdrivande företag som sköter och underhåller utveckling av Bluetoothteknologin. De hanterar även de licenser man behöver ha för att utveckla och marknadsföra sina Bluetooth produkter. [14]

2.3 Utvecklingskit från Cypress Semiconducters

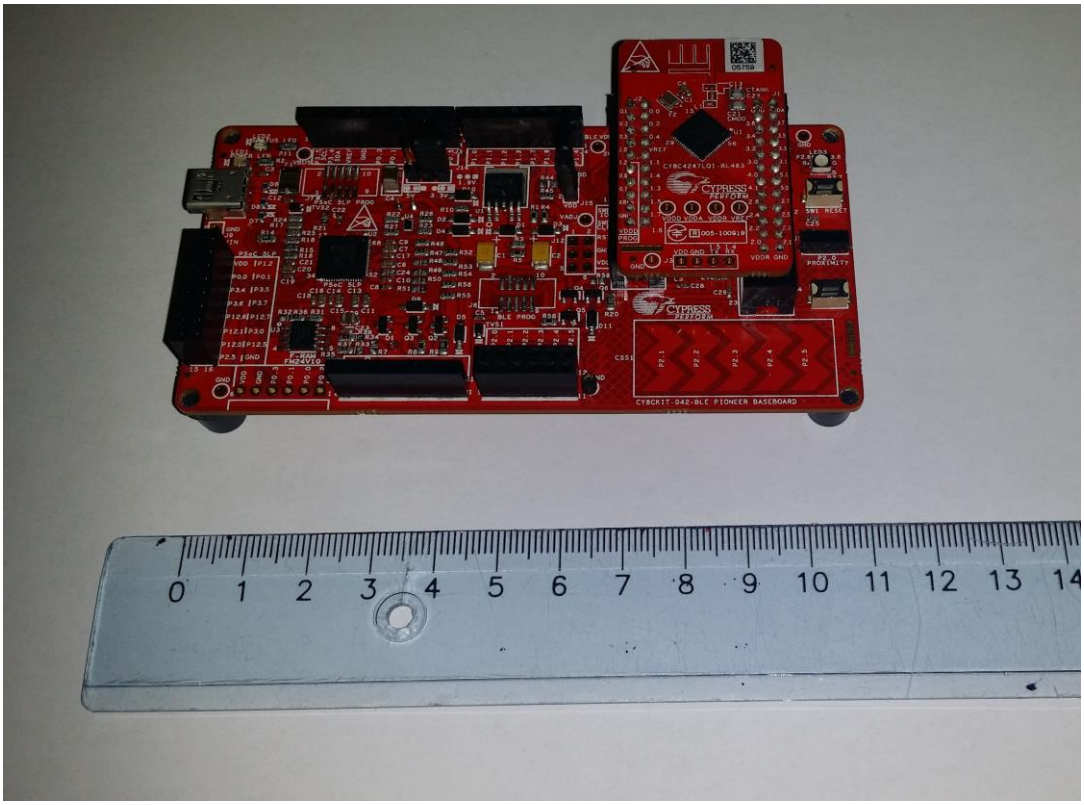
Cypress tillhandahåller flera programmerbara enheter bland annat PSoC, Programmable System on Chip och PRoC, Programmable Radio on Chip. Båda dessa finns som applicerbara storlekar under PSoC/PRoC BLE EZ. Till dessa kommer även PSoC Creator vilket är en kostnadsfri IDE som diskuteras mer under 8.4. I figur 5 visas en PSoC/PRoC EZ modul.



Figur 5 PSoC/PRoC BLE EZ modul ***[15]***

Specifikt valt kit är "CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit" som stöds av utvecklingsmiljön PSoC Creator. Kittet tillåter enkel användning av både PSoC 4 BLE och PRoC BLE enheter. Kittet inkluderar även CY5670 - CySmart USB Dongle (BLE Dongle) som tillsammans med Cypress applikation CySmart ger möjlighet att felsöka perifera enheter. Se figur 6 för bild på utvecklingskitet.

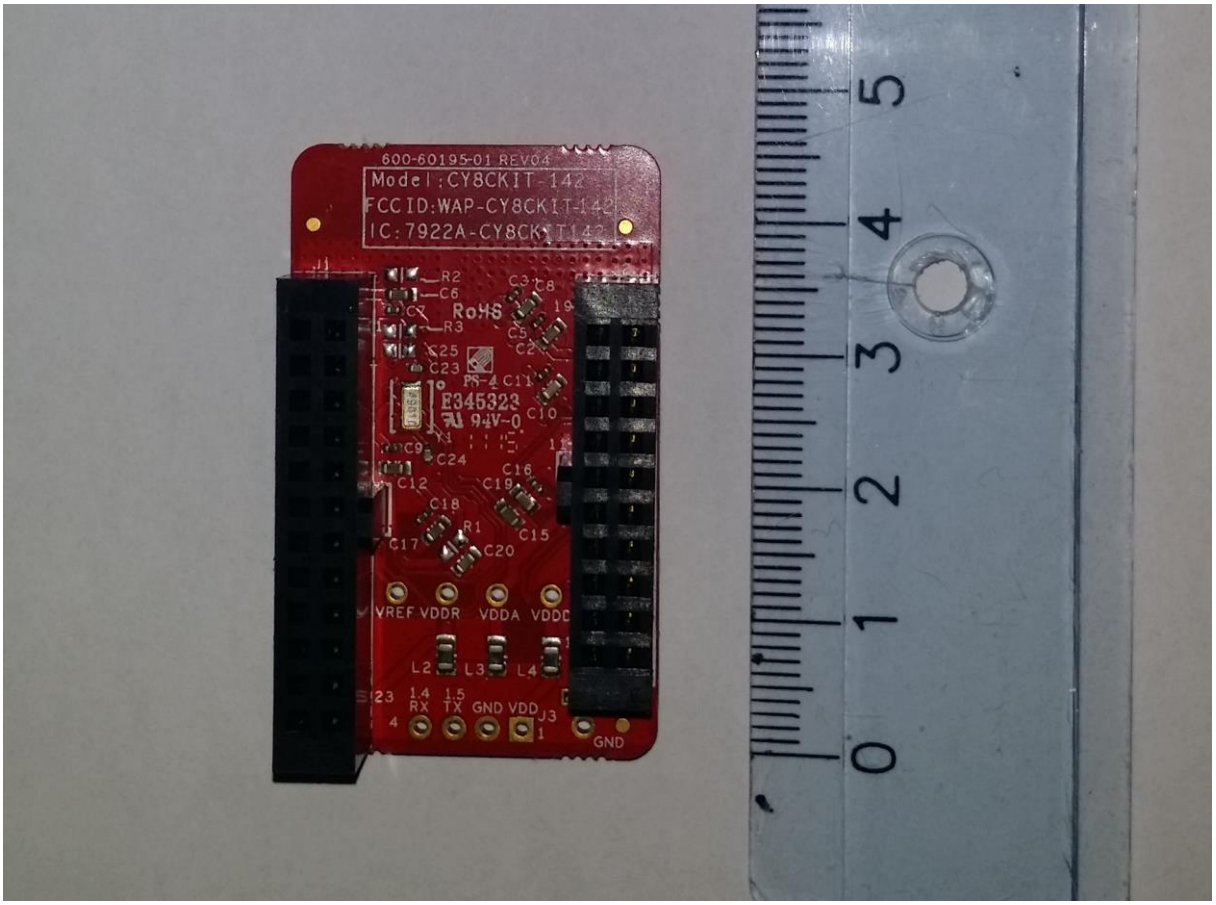
Kittets komponenter:
• BLE Pioneer Baseboard
• CY8CKIT-142 PSoC 4 BLE modul
• CY5671 PRoC BLE Module
• CY5670 - CySmart USB Dongle (BLE Dongle)
• Snabbstarts guide
• USB Standard A to Mini-B cable
• Fyra jumper wires (4 inch) och två proximity sensor wires (5 inch)
Coin cell (3V CR2032)



Figur 6 - CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit

2.3.1 PSoC BLE 4200

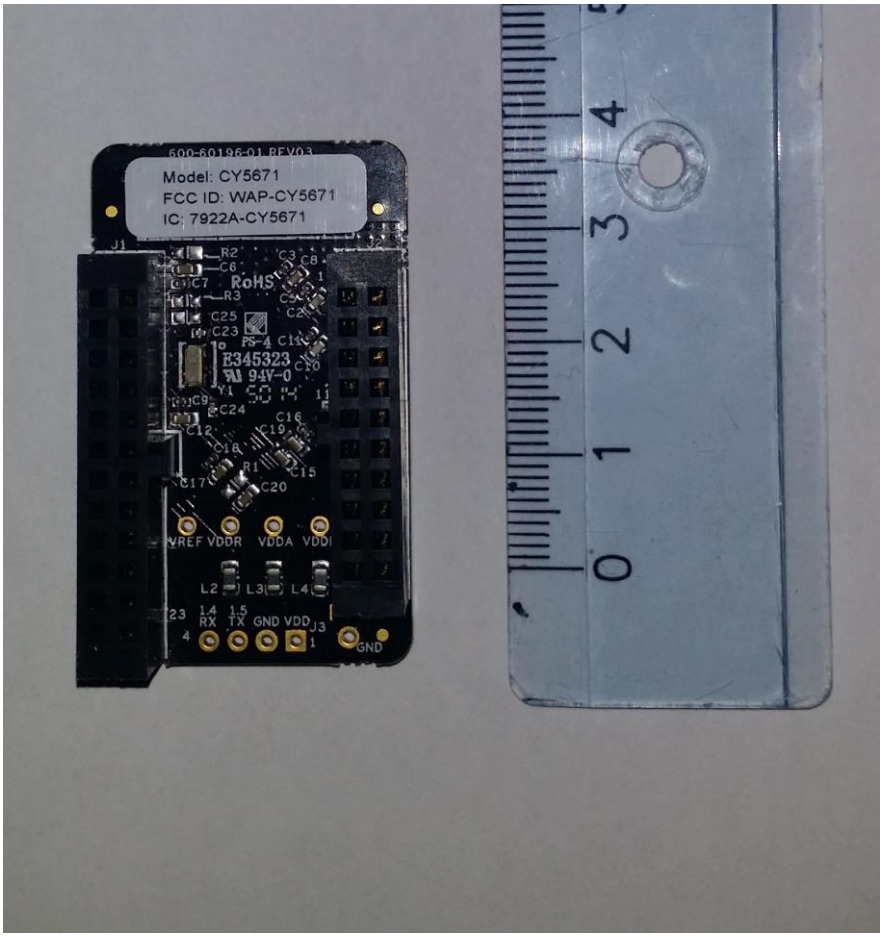
PSoC BLE är en PSoC enhet som utnyttjar en 48-MHz ARM® Cortex™-M0 CPU och en Bluetooth Low Energy modul. PSoC BLE finns med 256/128KB Flash och 32/16KB SRAM och har programmerbara analoga signal kretsar och programmerbar digital logik och flera andra egenskaper.[16]
Se figur 7 för bild på PSoC-modulen.



Figur 7 - PSoC BLE 4200

2.3.2 PRoC BLE 4200

PRoC BLE är en PRoC enhet som utnyttjar en 48-MHz ARM® Cortex™-M0 CPU och en Bluetooth Low Energy modul. PRoC BLE finns med 256/128KB Flash och 32/16KB SRAM programmerbar digital logik.[17]
Se figur 8 för bild på PRoC-modulen.



Figur 8 - PRoC BLE 4200

Även om PSoC och PRoC är väldigt lika varandra är de viktigaste skillnaderna enligt Cypress ” PRoC BLE provides the pre-configured schematic view of PSoC Creator with preconfigured peripherals for key applications, such as HID, remote controls, toys and BLE connectivity. PSoC 4 BLE uses the schematics view of PSoC Creator and offers programmable digital blocks and programmable analog blocks to create custom sensor-based BLE products.” [18]

Kortfattat är PRoC en enklare version med det mesta förinställt, avsett att använda för vanliga applikationer. Denna modul saknar då framför allt fler analoga signal kretsar.
För en mer detaljerad beskrivning se länkar för respektive modul eller besök Cypress Semiconducters hemsida.[16][17][18]
Cypress BLE utveckling har godkänts av BlueTooth SIG’s tester och är certifierat för BLE lösningar.[19]

2.3.3 Sleep mode

PSoC 4200 erbjuder en rad olika ”Sleep Modes” som kan användas för energireducering. Det finns fyra olika varianter; Sleep Mode, Deep Sleep Mode, Hibernate Mode, Stop Mode. Dessa olika ”Sleep Modes” brukar olika mängder energi genom att stänga ner olika mycket hårdvara. Ju djupare ner i ”Sleep Modes” desto lägre energiförbrukning får enheten. Ju längre ner desto längre tid tar det också för enheten att vakna upp. Energiförbrukning och uppvakningstider visas i tabell 2.

Power Mode	Current Range (typical) (Vdd = 3.3 V to 5.0 V)	WakeupTime PSoC 4200	WakeupTime PSoC 4200M	WakeupTime PSoC 4200L
Active	1.3 mA to 14 mA	-	-	-
Sleep	1.0 mA to 3 mA	0	0	0
Deep Sleep	1.3 uA to 15 uA	25 uS	25 uS	25 uS
Hibernate	150 nA to 1 uA	2 ms	0.7 ms	0.7 ms
Stop	20 nA to 80 nA	2 ms	2 ms	1.9 ms

Tabell 2 - Lista över "sleep modes"

För mer information se [20]

2.4 PSoC Creator

PSoC Creator är en Integrerad Design Miljö (IDE) som tillåter editering, kompilering och debugging av hårdvara och firmware. PSoC Creator är kompatibelt med PSoC 3, PSoC 4, PSoC 4 BLE, PRoC BLE och PSoC 5LP.
PSoC Creator använder sig av virtuella chips som enkelt definierar och kontrollerar valda komponenter.
Till PSoC Creator finns en väldokumenterad API för BLE-modulen och datablad för alla komponenter.

PSoC Creator inkluderar:

- Hårdvarudesign med komplett schematisk bild
- Över 120 förtestade, produktionsredo, PSoC-komponenter
    - Komplett kommunikationsbibliotek som I2C, USB, UART, SPI och Bluetooth Low Energy.
    - Verktyg för att utveckla specialgjorda komponenter I Verilog eller tillståndsdiagram.
    - Dynamiskt skapade API-bibliotek.
- Integrerad C-kompilator och editor.
- Inbyggd debugger.

För mer information se [21]

2.5 CySmart

CySmart är ett BLE emulatorverktyg för PC med Windows. CySmart tillhandahåller en Graphical User Interface (GUI) för debugging och tester av perifera BLE enheter. För att emulera rollen som en klient kräver CySmart Cypress BLE Dongle.

CySmart stöder bland annat anslutning, ”pairing”, ”bonding” och sökning efter enheter samt ger tillgång till GATT databaser för PRoC BLE, PSoC BLE enheter och familjen EZ-BLE moduler.

För mer information se [22]

3 Metod och analys

3.1 Inlärningsprocess

3.1.1 Implementation av BLE med färdigt kit

För att jobba med en ny produkt och ett nytt programspråk bör en tidsperiod för inlärning avsättas. Detta för att gör det möjligt att förstå konceptet och kunna formulera sin egen lösning hellre än att endast kunna följa liknande lösningar.

Cypress erbjuder sitt egenskrivna utvecklingsverktyg, PSoC Creator, vilket är en EDI som kombinerar blockprogrammering och C programmering. Cypress utvecklingskit med PSoC Creator är designat för en enkel inlärning och snabb produktutveckling.

På Cypress finns det utbildningsvideos i hur man jobbar med utvecklingskitet i BLE syfte tillsammans med PSoC Creator.

En uppskattad planering gjordes och ett arbetsschema med tid avsattes för att lära sig PSoC Creator.
Ambitionen var att till viss del lära sig PSoC Creator innan den riktiga uppgiften började utföras för att bäst kunna utnyttja PSoC Creators potential.

Beskrivning av inlärningsmetodiken:

1. **Börja titta på kodexempel**
   I IDE'n finns ett flertal färdigskrivna program som kunde användas. Det fungerade bra att ladda över till kittet för att sedan användas tillsammans med deras egna Android applikation CySmart.
   Genom att börja titta på kodexempel ville vi få en första inblick hur IDE'n var uppbyggd, hur man gjorde när man kompilerade och hur olika delar av IDE'n användes.

2. **Manipulera kodexempel**
   Att ta ett kodexempel och manipulera ett perfekt sätt att börja sätta sig in vad olika delar av koden gör och hur funktioner fungerar.

3. **Titta på utbildningsvideos.**
   När man efter ett tag inte får ut mer av att studera kodexempel är det en bra idé att gå över till de utbildningsvideos som finns. Tyvärr fanns det bara fem av dessa och de var rätt korta. Fördelen med videos är att det blir mycket och koncentrerad information på en kort tid. Dessutom vet man att det korrekt information.

4. **Hitta exempel och guider på nätet**

5. **Börja skriva eget och försöka bygga på mer**

3.2 Lösningen

Utöver att från grunden lära sig ett nytt utvecklingsverktyg behövde vi en strategi för hur vi skulle lägga upp arbetet. Genom att analysera kraven och tänka igenom produkten uppställdes olika delmål som projektet skulle bestå av.

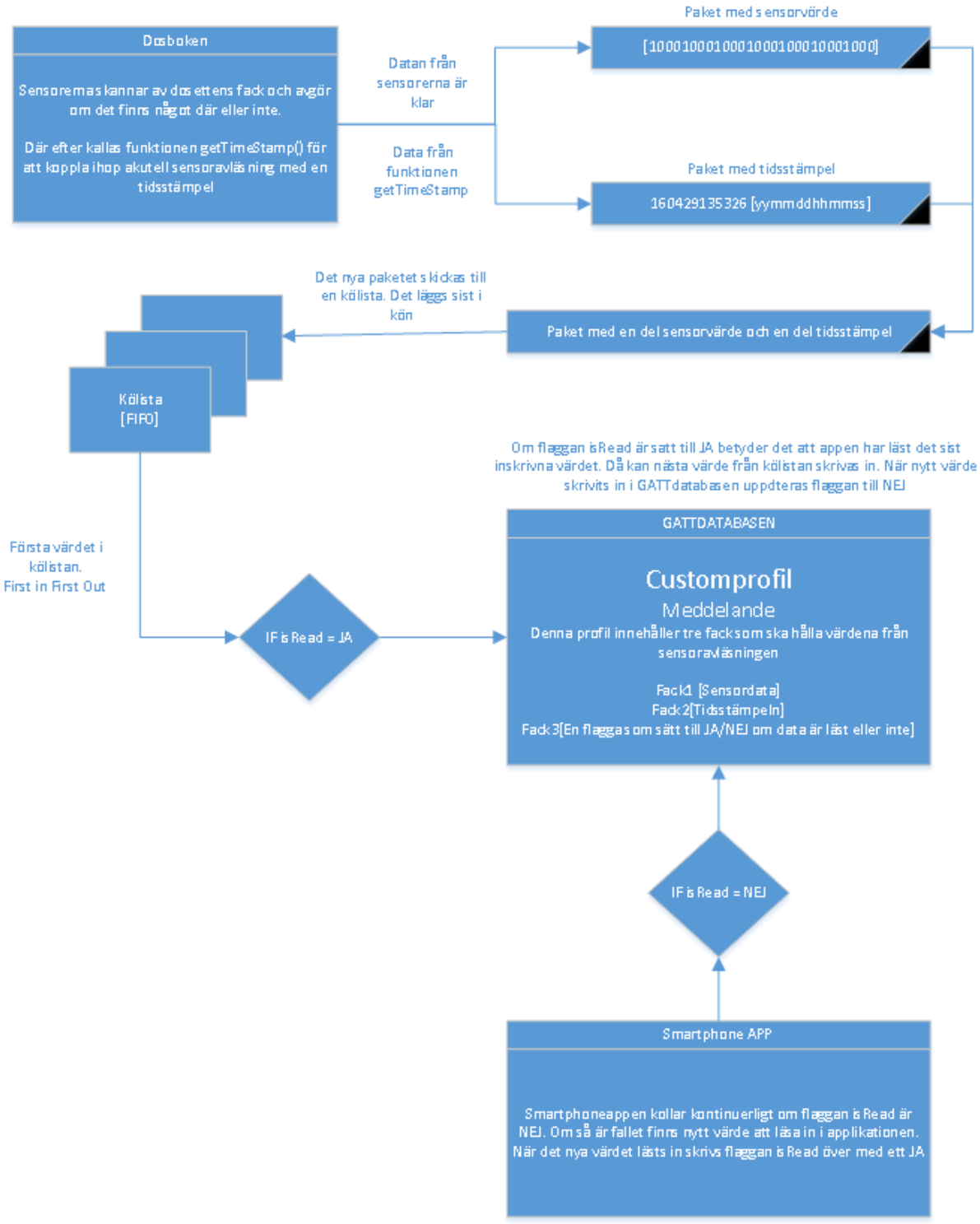
Hela uppgiften består egentligen av tre separata delar

1. Kunna skicka över sensordata från Dosboken
2. Kunna skapa/skicka med en tidsstämpel till sensordatan
3. Kunna skicka över batteristatus för Dosboken

Dessa delmål har sedan ett antal olika underkrav

1. Data
    1. Funktion som tar in sensorvärden
    2. Funktion som gör om sensorvärden till lämpligt paket att skicka via BLE. Omvandling från binär sekvens till decimalt bedömdes lämpligt. 28 bitars sträng av sensorvärden omvandlas till fyra stycken 8 bitars decimala tal.
    3. Funktion som skriver in sensorvärde + tidsstämpel i GATT-databasen. Flaggan isReadyFlag avgör om nytt värde kan skriva in eller skall sparas tills möjlighet uppkommer. Datan lagras i kronologisk ordning tills dess att de kan skrivas in i databasen.
2. Tidsstämpel
    1. Funktion(klocka) som kan skapa en tidsstämpel
3. Kunna skicka över batteristatus för Dosboken
    1. Funktion som uppdaterar batteristatus vid efterfrågan

För att enklare kunna få en överblick skapade vi ett flödesschema som visas i figur 9



Figur 9 - Flödesschema över programmets funktion

3.3 Mjukvaran

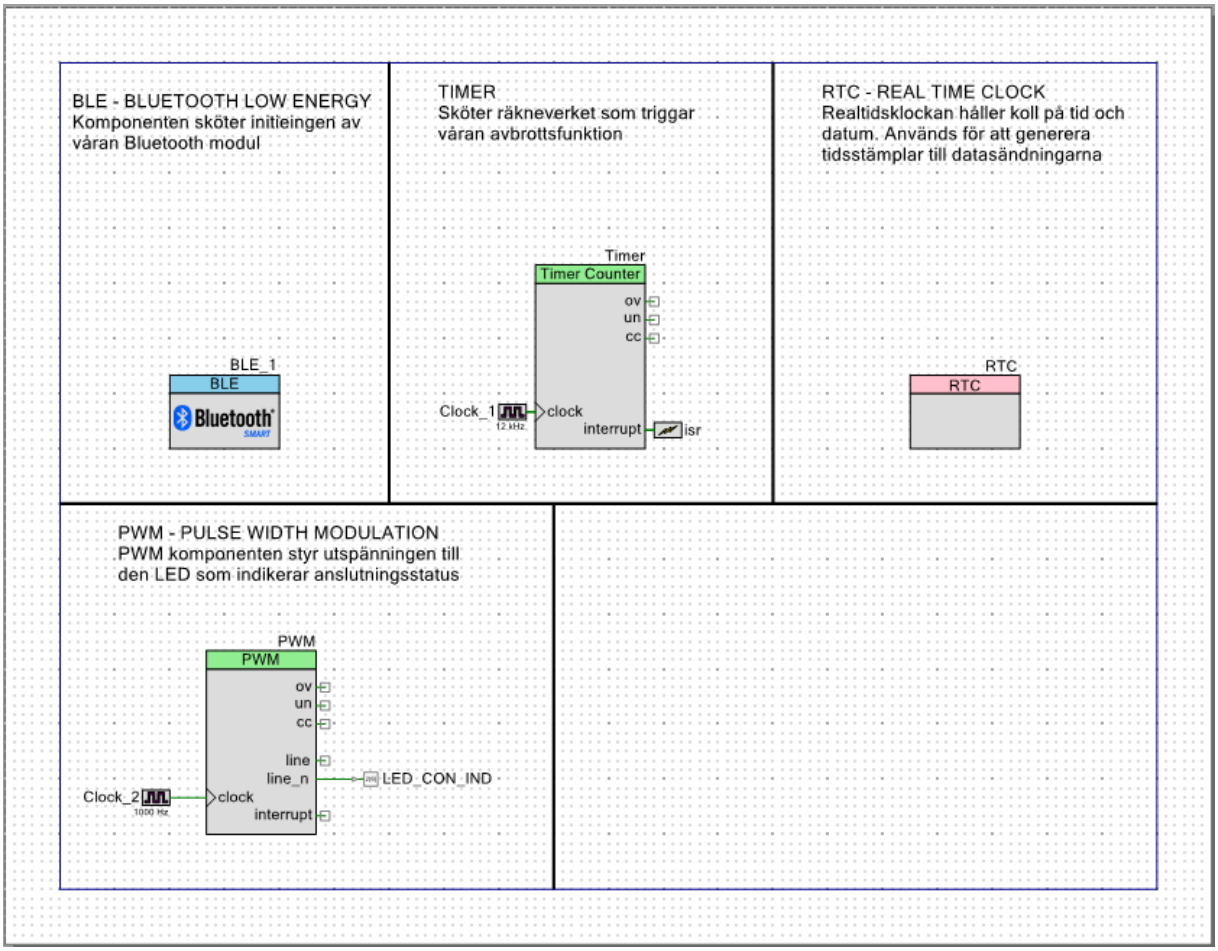
Lösningen vi har gjort består till huvuddel av tre komponenter.

1. Top Design
2. Main
    a. Main.c
    b. Main.h
3. updateGattDatabase
    a. updateGattDatabase.c
    b. updateGattDatabase.h

Hela lösningens kod finns bifogad i Appendix A.

3.3.1 Topdesignen

Till en början skapades en topdesign för projektet. Här sätts de komponenter ut som kommer att krävas för produkten. Får att utveckla kommunikationen krävdes ett antal komponenter. Se figur 10



Figur 10 - Top Design i PSoC creator

BLE_1 – Detta är själva komponenten för BLE modulen i kommunikationen.

Timer – Denna komponent är en klocka som kan räkna. Den används för att få ett tidsintervall för när vår avbrottsfunktion skall triggas.

RTC – Detta är en så kallad ”Real Time Clock”. Den räknar mer exakt än en vanlig räknare och har inbyggda funktionen för att kunna hålla reda på tid och datum. Den används för att vi ska kunna skapa en tidsstämpel

PWM – ”Pulse Width Modulation”. Denna komponent används för att skapa en PWM signal till en LED indikator på kortet. LED lampan används i syfte att visa när anslutningen är CONNECTED eller DISSCONNECTED.

3.3.2 Main.c

Det första som händer är att alla variabler deklareras. Mest speciellt här är deklareringen av ett par specifika variabler som krävs för skrivandet och läsandet av data från GATT-databasen.

```
CYBLE_GATT_HANDLE_VALUE_PAIR_T handleValue;
CYBLE_GATT_HANDLE_VALUE_PAIR_T handleValueFlag;
CYBLE_CONN_HANDLE_T connHandle;
CYBLE_CONN_HANDLE_T connHandleFlag;
CYBLE_GATT_ERR_CODE_T apiResult;
CYBLE_GATT_HANDLE_VALUE_PAIR_T handleValueFlagInsert;
CYBLE_GATTS_WRITE_REQ_PARAM_T *wrReqParam;
```

CYBLE_GATT_HANDLE_VALUE_PAIR_T är en variabel som krävs för att lagra värden och parametrar som krävs för inskrivning till en vald ”characteristic” i GATT-databasen.

```
handleValueFlag.attrHandle = wrReqParam->handleValPair.attrHandle;
        handleValueFlag.value.val = wrReqParam->handleValPair.value.val;
        handleValueFlag.value.len = 1;
        CyBle_GattsWriteAttributeValue(&handleValueFlag, 0, &cyBle_connHandle,
CYBLE_GATT_DB_LOCALLY_INITIATED);
```

attrHandle – Information om de fält i GATT-databasen som data ska skrivas till
value.val – Information om vilket värde som ska skrivas in i GATT-databasen
value.len – Information om hur långt fältet som data ska skrivas till ska bli

CYBLE_CONN_HANDLE_T är en variabel som hanterar information för själva anslutningen till GATT-databasen.

CYBLE_GATTS_WRITE_REQ_PARAM_T används för att hantera värden som läses från GATT-databasen

CYBLE_GATT_ERR_CODE_T används för att spara returnerade felmeddelande

Vidare används en avbrottsfunktion vars uppgift är att skapa ett avbrott varje sekund för att läsa av flaggan isReadyFlag och se om ny data kan skrivas in GATT-databasen.

```
CY_ISR(isr_Handler){}
```

Här skapas bland annat ett par tidsstämplar för våra tester.

```
createTimeStamp();
```

```
apiResult = CyBle_GattsReadAttributeValue(&handleValueFlag, &connHandleFlag,
CYBLE_GATT_DB_LOCALLY_INITIATED);
    flag = *handleValueFlag.value.val;


    if (apiResult == CYBLE_GATT_ERR_NONE){
        if (flag == 0){
            updateGattDataBase();
        }
    }
```

Här läses flaggan isReadyFlag av och sedan avgörs det om det är dags att skriva in ett nytt värde.

Stackhandler som kan ta hand om alla events som kan ske.

```
void stackHandler(uint32 event, void *eventParam){
```

void stackHandler(uint32 event, void *eventParam) bearbetar händelser genererade av BLE modulen. Dessa är grunden för BLE kommunikationen och essentiellt för övergången mellan hårdvara och Smartphone.

Hanterar de händelser vi bedömt mest nödvändiga CYBLE_EVT_STACK_ON, CYBLE_EVT_GAP_DEVICE_DISCONNECTED,

CYBLE_EVT_GAP_DEVICE_CONNECTED, CYBLE_EVT_GATTS_WRITE_REQ. Alla händelser kommer med en pekare till eventuella parametrar.

Följande viktiga händelser kan inträffa men är inte begränsade till:

- CYBLE_EVT_STACK_ON
  Stacken har initierats, generas när CyBle_Start() kallats.
- CYBLE_EVT_GAP_DEVICE_DISCONNECTED
  Genereras när den Centrala enheten (Smartphone) avslutat anslutningen eller försök till anslutning misslyckats. Ny annonsering startas här.
- CYBLE_EVT_GAP_DEVICE_CONNECTED
  Genereras när ny anslutning gjorts. Tänder diod på anslutningskittet för indikering på att anslutning lyckats.
- CYBLE_EVT_GATTS_WRITE_REQ
  Genereras vid begäran från klient att skriva till GATT databasen.
  Innehåller data och adress över vad som begärts ska skrivas.
  Här sätts flaggan som visar att datan lästs av.

```
CySysTickStart();


/* Configure SysTick timer to generate interrupt every 100 ms */
CySysTickSetReload(SYSTICK_RELOAD);


for (i = 0u; i < CY_SYS_SYST_NUM_OF_CALLBACKS; ++i)
{
  if (CySysTickGetCallback(i) == NULL)
  {
    /* Set callback */
    CySysTickSetCallback(i, SysTickIsrHandler);
    break;
  }
}
```

CySysTickStart() är metoden för att initiera avbrottsfunktionen som ska styra våran realtidsklocka. Denna kod fanns fördefinierad i ett kodexempel given av PSoC creator. Kodexemplet heter RTC_P4_SysTick_Example.

3.3.3 UpdateGateDatabase

De hjälpvariabler som krävs för att hålla den data som ska skrivas till GATT-databasen skapas här samt de variabler som krävs för att hålla tid och kölistorna på data och tidsstämplar.

```
void intialiseGattDataBase()
```

Denna funktion åkallas för att skapa den kölista som ska hålla den data som kommer från Dosbokens sensorer. Den skapar en kö som anses vara lagom stor och fyller dessutom den med tomma värden.

Vidare skriver den för våra tester skull in en rad värden i kölistan via funktionen insertStorage()

Den sätter även ett startvärde på flaggan isReadyFlag. Det bidrar till att programmet inte skriver något till GATT-databasen förrän telefonen ändrar flaggan och därmed säger att den är redo att ta emot data.

```
void createTimeStamp(){
```

Denna funktion åkallas då en tidsstämpel ska skapas.

Funktionen hämtar värden om tid och datum från komponenten RTC (real time clock och sparar dessa i variablerna. Detta görs genom den inbyggda funktionen RTC_GetDateAndTime();

Vidare placeras de på den tidigaste tomma platsen i kölistan vilket korresponderar mot samma position i datalistan.

```
void updateGattDataBase(){
```

Denna funktion åkallas då flaggan isReadyFlag blivit satt till noll. Det betyder att ny data är redo att skrivas in i GATT-databasen.

Här skapas en temporär vektor som innehåller både data från sensorerna och tidstämpeln. Den ser ut på följande sätt:

[data][data][data][data][ss][mm][hh][dd][mm][yy][yy]

Vill man förstå exakt hur de olika fälten i vektorn sätts kan man läsa koden i Appendix 1.

I slutet körs kommandona:

- CyBle_GattsWriteAttributeValue(&handleTimeHour, 0,&cyBle_connHandle, CYBLE_GATT_DB_LOCALLY_INITIATED);
- CyBle_GattsWriteAttributeValue(&handleValueFlagg, 0, &cyBle_connHandle, CYBLE_GATT_DB_LOCALLY_INITIATED);

Dessa skriver in datamängden och ändrar flaggan isReadyFlag till 1 igen.

Slutligen åkallas funktionen updateStorage()

```
void updateStorage(){
```

Denna funktion ser till att det första värdet i kölistan som precis blivit sänt till GATT-databasen tas bort. Efter det flyttas alla värdena fram ett steg. På så vis fås funktionen FIFO – ”first in first out” i kölistan.

```
void insertStorage(){
```

Denna funktion åkallas när man vill skriva in ett värde i kölistan. Funktionen ser till att hitta den första tomma platsen i kölistan och därefter skrivs värdet in.

4 Analys

I detta kapitel analyseras och utvärderas de frågeställningar som tagits upp.

4.1 Utvärdering av trådlös kommunikation

För att kunna visa vilken metod av trådlös kommunikation som lämpar sig bäst vill vi identifiera ett par punkter som är viktiga för den typ av dataöverföring som kommer att krävas av oss. Tanken är att dessa punkter ska kunna presenteras i en enkel tabell för att lättöverskådligt förstå vårt val. Denna tabell kan även användas i framtiden vid utveckling av nya prototyper med andra kommunikationslösningar, då kraven för produkten kan ha ändrats.

4.2 Vilken kommunikation

Följande punkter har vi kommit fram till är värda att ta med i beräkningen för valet av den trådlösa kommunikationen.

4.2.1 Hastighet

Hur snabbt kan data överföras? Vid en tillfällig anslutning bör all data förväntas hinna sändas.

4.2.2 Räckvidd

Den slutgiltiga produkten ska kommunicera med en smartphone och det typiska användningsområdet är hemmet. Det betyder att produkt och smartphone oftast kommer att vara inom samma hus men kanske inte ligga i närheten. Man vill så klart helst ha det till att det i största möjliga mån alltid finns en anslutningsmöjlighet mellan produkt och smartphone. Ju längre räckvidd det finns ju större är sannolikheten att data inte kan skickas över direkt.

4.2.3 Bandbredd

Det finns två huvudsakliga saker att ta hänsyn till när det gäller bandbredd:

1. Störningar
   En bandbredd som inte är mycket brukad minskar de störningar som sändningar utsätts för.
2. Licenser
   Att sända data trådlöst är ofta betingat med licenser och avgifter. Om möjligt bör ett bandbreddsspann som inte innefattar licenskrav och avgifter väljas.

4.2.4 Batteriförbrukning

Energiförbrukning är alltid en aktuell fråga. En låg energiförbrukning alternativt med möjlighet till att minska energiförbrukningen är önskvärt.

4.2.5 Säkerhet

Kontroll av anslutningar bör finnas tillgängligt för att kontrollera att rätt användare får tillgång till enheten, inte bara av sekretesskäl utan även för funktionaliteten.

Om enheten förväntas hantera känslig data måste säkerhet kunna implementeras.

4.2.6 Färdiga moduler

Att själv konfigurera en Bluetooth modul med mikrokontroller ger hög kontroll. Men att sätta ihop egen lösning av nödvändig hårdvara är ett projekt som kräver hög teknisk kunskap, större omkostnader och mycket mer tid. Ett företag som Finn Medicinen tjänar mycket på att utnyttja de färdiga lösningar som andra företag redan tagit fram. Det finns ingen anledning för ett litet entreprenörsföretag att ta fram teknik som redan finns.

Färdiga utvecklingskit för dessa moduler ger mindre kontroll men en snabbare produktutveckling. Ett färdigt utvecklingskit med integrerade kretsar, väl valda komponenter samt en utvecklingsmiljö avsedd att fokusera på produktens behov. Dessa ger väldigt snabb produktutveckling.

4.2.7 Etablerat märke

Ju större ett märke är och ju mer etablerat det är desto mer finns dokumenterat. Det finns sannolikt mer inlärningsmaterial och det finns flera ställen att vända sig till om för hjälp eller stöd. Det är också mer sannolikt att en tredje part har lättare att ta vid där man slutade om det är ett etablerat märke.

4.2.8 Basstation

En viktig detalj för företaget var att protokollet redan kan kommunicera med en Smartphone och inte behöver ha någon form av basstation mellan produkten och telefonen.

4.3 Kommunikationsprotokoll

Efter att ha undersökt dessa fem trådlösa protokollen närmare kom vi fram till at BLE skulle bli det bästa för produkten. Företaget hade indirekt redan bestämt detta men ville för framtida bruk ändå låta oss utvärdera och jämföra fler trådlösa kommunikationer.

Anledningen till att vi valde BLE var att det var det enda protokollet som uppfyllde alla tre viktiga kraven:

a. Låg batteriförbrukning
b. Räckvidd
c. Har möjlighet att kommunicera direkt till en Smartphone

Punkt c ansågs tillsammans med batteriförbrukning vara den viktigaste punkten att uppfylla. Hastighet ansågs mer än tillräcklig för produktens ändamål.

4.4 Tankar och funderingar på att välja modul och processor själv

Att välja modul och processor själv erbjuder en möjlighet att på ett mer exakt sätt skräddarsy en uppsättning som precis passar kundens behov. Det medför dessvärre också ett större krav på teknisk kompetens då man måste ha avancerad kunskap om flera tekniker.

Då detta examensarbete hoppas kunna nyttjas till en färdig produkt bedömdes det att en integrerad modul med mikrokontroller vara den bästa vägen.

På marknaden finns det utvecklingskit avsedda för snabb, enkel och prisvärd utveckling av BLE kommunikation vilket skulle passa företagets behov.
Ett av dessa bestämdes vara det effektivaste sättet att utveckla en lämplig prototyp för att matcha kundens behov.

4.5 Val av färdigt kit

Eftersom en stor del av detta examensarbete handlar om att leverera en produkt till företaget i fråga kände vi snabbt att tiden skulle rinna iväg om vi först skulle sitta ner och välja mellan tusentals av modeller och fabrikat av BLE moduler och mikroprocessorer. Jobbet med att sedan sätta ihop dessa för att kunna börja programmera bedömde vi efter tag blev för stort. För att hinna leverera det företaget ville ha och inte få fokus flyttat från att utveckla och utvärdera kommunikationen till att välja rätt mikroprocessor. Under en mässa vid namn ”Hardware Connect” Malmö erhölls kontakt med ett företag som sålde BLE utvecklingskit. Det var ett företag som var återförsäljare åt ett mycket större Amerikanskt företag vid namn Cypress Semikonduktors. Detta företag erbjöd ett BLE utvecklingskit med deras egna tillhörande utvecklingsverktyg PSoC Creator. Det var ett utvecklingskort gjort för att kunden lätt skulle kunna komma igång och börja testa att utveckla sin prototyp. Det passade oss helt perfekt. Denna direktkontakt med en återförsäljare gjorde det möjligt att komma igång tidigt.

4.6 Tester

4.6.1 Varför testa?

Vid testning och utvärdering av funktionaliteten och lämpligheten hos den utvecklade programvaran och lämpligheten i att använda BLE protokollet behövs en beskrivning av hur produkten (i sin helhet) skall fungera följt av en definition av vad produkten bör klara samt vilka funktioner produkten måste ha. Från dessa funktioner kan rimliga fälttester definieras för utvärdering.

4.6.1.1 Dosboken

För långtidssjuka människor som behöver ta medicin över längre perioder vid specifika intervaller har en elektronisk påminnelsemetod utvecklats. Dosboken, med tillhörande hårdvara och Smartphone applikationen, ska med hjälp av ett veckolångt schema kunna övervaka när medicinen tagits ut. Varje gång en position förändrats ska en uppdatering av nuvarande tillstånd skickas till applikationen som jämför med det veckolånga schemat och avgör om det efterföljs. Dessa uppdateringar ska kunna lagras om anslutning mellan applikationen och Dosboken förlorats och sedan överföras när möjlighet ges.

Applikationen ska kunna vidareförmedla denna information till användaren.

4.6.1.2 Programvara

- PSoC BLE ska kunna skriva och läsa från GATT databasen
- PSoC BLE ska kunna annonsera sin närvaro
- PSoC BLE ska kunna läsa av hårdvaran
- PSoC BLE ska kunna skapa tidsstämplar för varje tillståndsavläsning
- PSoC BLE ska kunna lagra och sortera en rimlig mängd data

4.6.1.3 Förväntad funktionalitet

- PSoC BLE bör kunna ansluta över rimliga avstånd
- PSoC BLE bör kunna ansluta trots rimliga hinder
- PSoC BLE bör kunna hålla anslutning över en rimlig tid
- PSoC BLE bör ha implementerat energi reduktion

4.6.1.4 Genomförande

Programvaran testas utan att hänsyn till den förväntade funktionaliteten. Dessa funktioner finns specificerade i kapitel 9.3.

Funktionerna, updateGattDataBase(),createTimeStamp(),updateStorage(), insertStorage samt eventHandlers CY_ISR(isr_Handler), void stackHandler(uint32 event, void *eventParam)utvärderades med CySmart.

Enskilda tester med CySmart visar att dessa events på ett lämpligt sätt hanterar och skapar en funktionell miljö.

Övriga händelser bör omhändertas på liknande sätt där utskrifter till en ”debugger” lämpligen görs. Eventuella specifika händelser för nödvändiga situationer kan uppkomma, se API dokumentation över ”*BLE COMMON EVENTS”*. [23]

De funktioner som handhar lagring av data och uppdatering av GATT databasen testades enligt den ordning de var tänkta att användas. När updateGatt DataBase testats följde övriga funktioner efter i ordning; updateGattDataBase -> insertStorage , createTimestamp->updateStorage där till sist alla funktioner testades tillsammans med flera värden för att verifiera att flödesdiagramet stämmer, se figur 9.

Avbrottsfunktionerna CY_ISR(isr _Handler) och void SysTickIsrHandler(void)har testats på liknande sätt genom att låta data skrivas in till GATT databasen och avläsas med CySmart.

Varje funktion testas till önskat resultat uppnåtts med hjälp av CySmart. Då hårdvaran ännu inte var färdig skapades ingen metodik för avläsning av aktuella tillstånd.

Funktionaliteten testas med hjälp av två fälttester där ett scenario får goda förutsättningar och ett scenario får dåliga förutsättningar görs. Båda anses ändå vara rimliga i avseendet att sändaren alltid placeras på markplan och aldrig i skåp eller dylikt. Med dessa scenarion kontrolleras anslutning, skrivning och läsning av/till GATT databasen från minst en position i varje rum och graderas med ett medelvärde från fem försök där alla värden avrundas uppåt.

- Goda förutsättningar antas vara i lokalens centrum där medelavståndet, som är en nyckelfaktor, är lägst
- Dåliga förutsättningar antas vara en rimlig del av lokalen (kök, sovrum eller toalett) i utkanten av huset där längst avstånd kan uppnås
- Varje försök betygsätts med *n* där:
  n = 1: Anslutning möjlig, (minst) två läs/skrivningar innan avbrott
  n = 2: Anslutning möjlig, (högst) en läs/skrivning innan avbrott
  n = 3: Anslutning ej möjlig

Varje medelvärde *m*, *m*= (n1 + n2 + n3 + n4 + n5)/ 5 avrundas uppåt för att undvika missvisande positiva resultat och tilldelas en färg.
m = 1: Grön
m = 2: Gul
m = 3: Röd

4.7 Källkritik

Z-Wave.com tillhör Sigma Designs, ett styrande medlemsföretag i Z-Wave Alliance. Z-Wave Alliance är en grupp företag som jobbar för utveckling av Z-Wave. De framhåller antagligen Z-Wave bra förhållande gentemot övrig information de visar kring andra protokoll.

Wi-fi.org underhålls av WIFI Alliance och antas vara en korrekt källa.

Telegesis är ett företag som utvecklar ZigBee moduler. De kan antas vara partiska när det informerar om andra protokoll.

Kjell&Company är ett svenskt detaljhandelsföretag som marknadsför sig som ett teknikföretag. De är oberoende och ger en opartisk alla sina produkter.

ZigBee.org eller ZigBee Alliance handhar utveckling och standardisering av ZigBee protokollet.

Bluetooth.com underhålls av Bluetooth SIG, som utvecklar och standardiserar Bluetooth protokollet. Alla företag som vill utveckla Bluetooth produkter måste certifieras av Bluetooth SIG.

Adafruit är ett företag i New York med avsikten att skapa en hemsida för utbildning inom elektronik och att skapa egna elektroniska produkter.

Cypress Semiconductor är ett företag som tillhandahåller tekniska lösningar till integrerade kretsar inom en rad inriktningar. I det här fallet är det tillverkaren av det valda utvecklingskitet och den valda modulen.

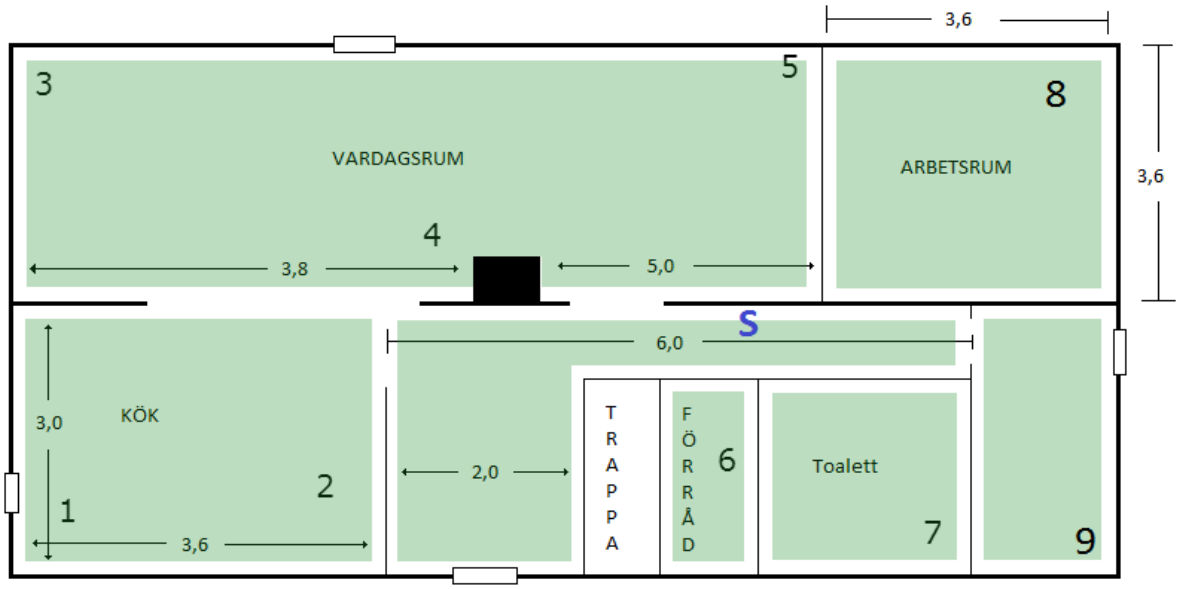
5 Resultat

Dessa testen syftar till att visa anslutnings- och sändningsräckvidd och hur väl den trådlösa kommunikationen fungerar.

Testfallen här under visar de tre olika boendeformerna; modern villa, lägenhet och gammal villa. För varje boendeform finns det en bra placering och en dålig placering. Exakta definitioner av hur testen ska gå till samt hur resultaten skall tolkas finns beskrivet under kapitel 9.6.1.

5.1 Modern villa (markplan) – Bra placering av sändare

Sändaren är placerad centralt i huset för att få kortast genomsnittliga avstånd till mätpunkterna. Huset har genomgående gips och träväggar och WIFI signal finns. Mätpunkt 4 lades till för att ge en klarare bild av hela rummet. Inga problematiska mätpunkter hittades.



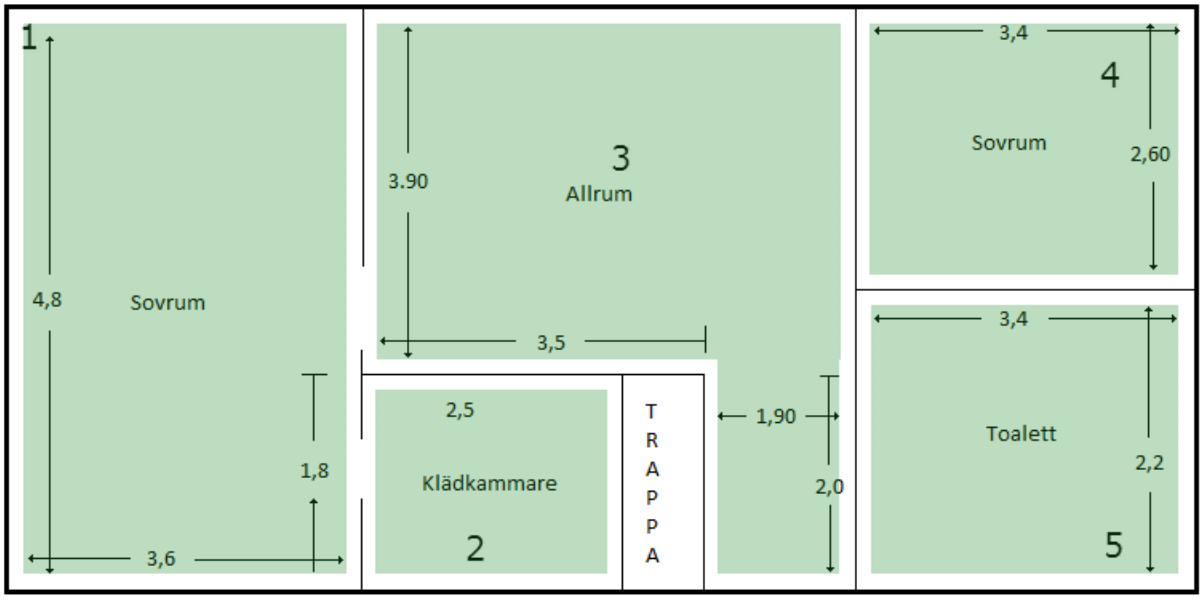
Figur 11 – Modern villa, Trä/Gips, Markplan, S = Sändare, Bra placering av sändare

Mät punkt	Mätvärden	Avrundat medel värde	Medelvärde	Avstånd till sändare (approx)
1	[1,1,1,1,1]	1	1,0	7,0m
2	[1,1,1,1,1]	1	1,0	4,0m
3	[1,1,1,1,1]	1	1,0	7,0m
4	[1,1,1,1,1]	1	1,0	3,0m
5	[1,1,1,1,1]	1	1,0	3,0m
6	[1,1,1,1,1]	1	1,0	1,5m
7	[1,1,1,1,1]	1	1,0	3,0m
8	[1,1,1,1,1]	1	1,0	5,0m
9	[1,1,1,1,1]	1	1,0	5,0m

Tabell 3 – Mätvärden Modern villa, Markplan, Bra placering av sändare

5.2 Modern villa (ovanvåning) – Bra placering av sändare

Ovanvåning av modern villa. Sändaren är fortfarande placerad centralt markplan. Även här är väggarna gjorda av gips och trä. WIFI signal finns. Inga problematiska mätpunkter hittades.



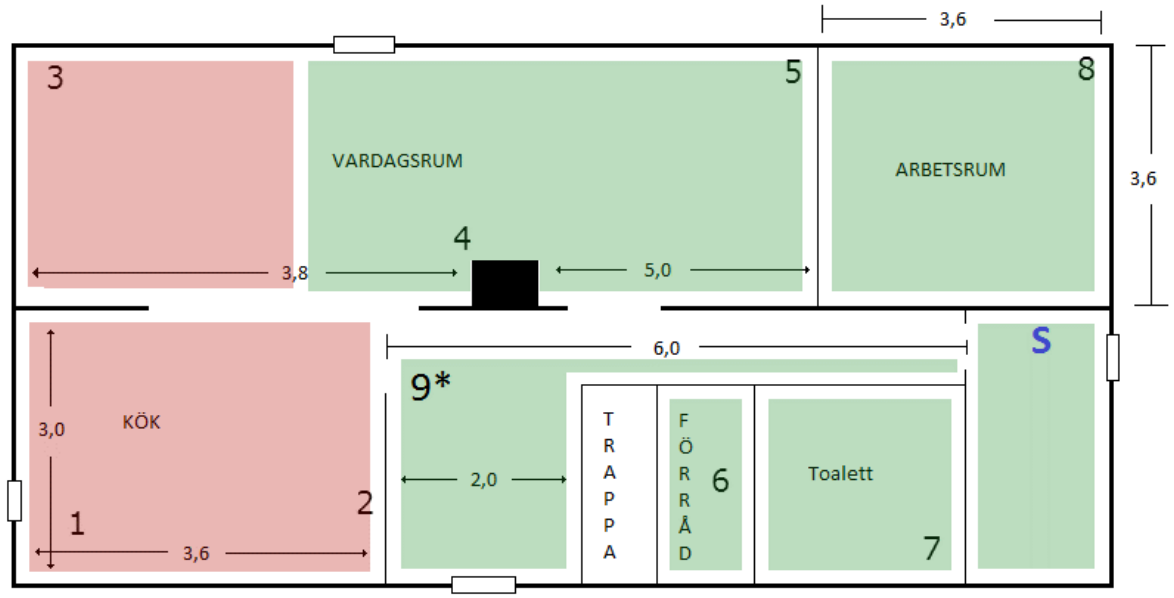
Figur 12 – Modern villa, Trä/Gips, Ovanvåning, S fortfarande placerad centralt på markplan, Bra placering av sändare

Mät punkt	Mät värden	Avrundat medelvärde	Medelvärde	Avstånd till sändare (approx)
1	[1,1,1,1,1]	1	1,0	8.0m
2	[1,1,1,1,1]	1	1,0	5.0m
3	[1,1,1,1,1]	1	1,0	4.0m
4	[1,1,1,1,1]	1	1,0	6.0m

Tabell 4 - Mätvärden modern villa, Ovanvåning, Bra placering av sändare

5.3 Modern villa (markplan) – Dålig placering av sändare

Sändaren är placerad i utkanten i huset. Detta ger potentiellt stora genomsnittliga avstånd till mätpunkter. Huset har genomgående gips och trä. WIFI signal finns. I detta fall är sändaren placerad väldigt nära WIFI-enheten.



Figur 13 - Modern villa, Trä/Gips, Markplan, S = Sändare, Dålig placering av sändare

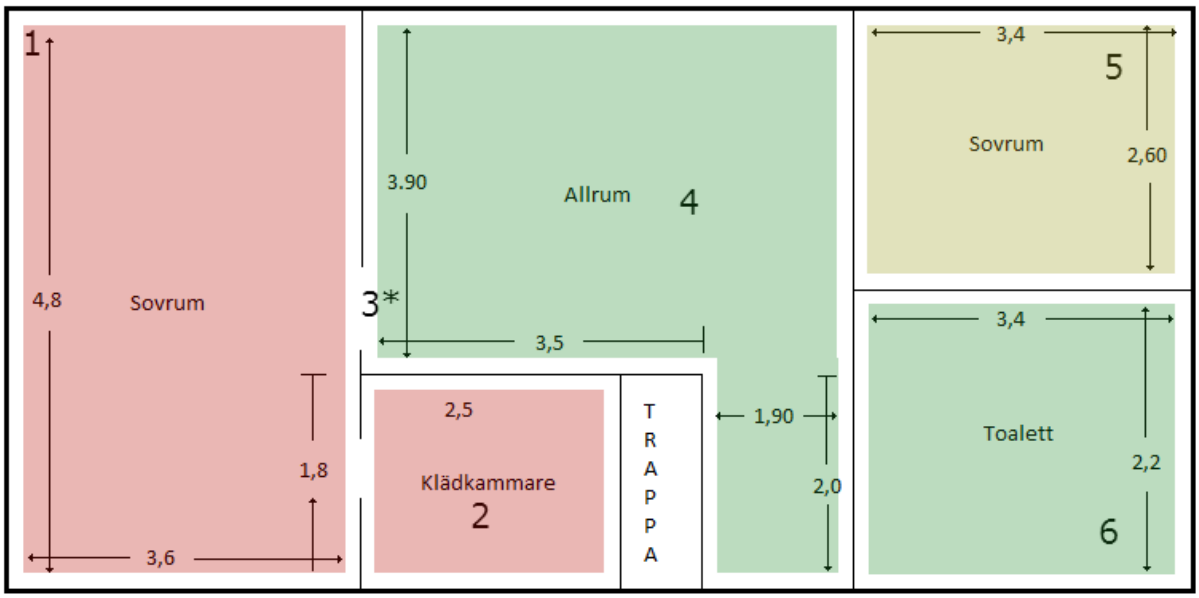
Mät punkt	Mät värden	Avrundat medelvärde	Medelvärde	Avstånd till sändare (approx)
1	[2,2,3,2,3]	3	2,4	12,0m
2	[3,2,3,2,1]	3	2,2	7,0m
3	[2,2,3,3,2]	3	2,4	12,0m
4	[1,1,1,1,1]	1	1,0	7,0m
5	[1,1,1,1,1]	1	1,0	5,0m
6	[1,1,1,1,1]	1	1,0	5,0m
7	[1,1,1,1,1]	1	1,0	3,5m
8	[1,1,1,1,1]	1	1,0	3,5m
9*	[1,1,1]	n/a	n/a	6,5m

Tabell 5 - Mätvärden modern villa, Markplan, Dålig Placering av sändare

* Mätpunkt 9 visar var övergången sker

5.4 Modern villa (ovanvåning) – Dålig placering av sändare

Sändaren finns fortfarande placerad i utkanten på markplan. Väggar och tak består av gips och trä. WIFI signal finns.



Figur 14 - Modern villa, Trä/Gips, ovanvåning, Sändare finns fortfarandeplacerad på markplan, Dålig placering av sändare

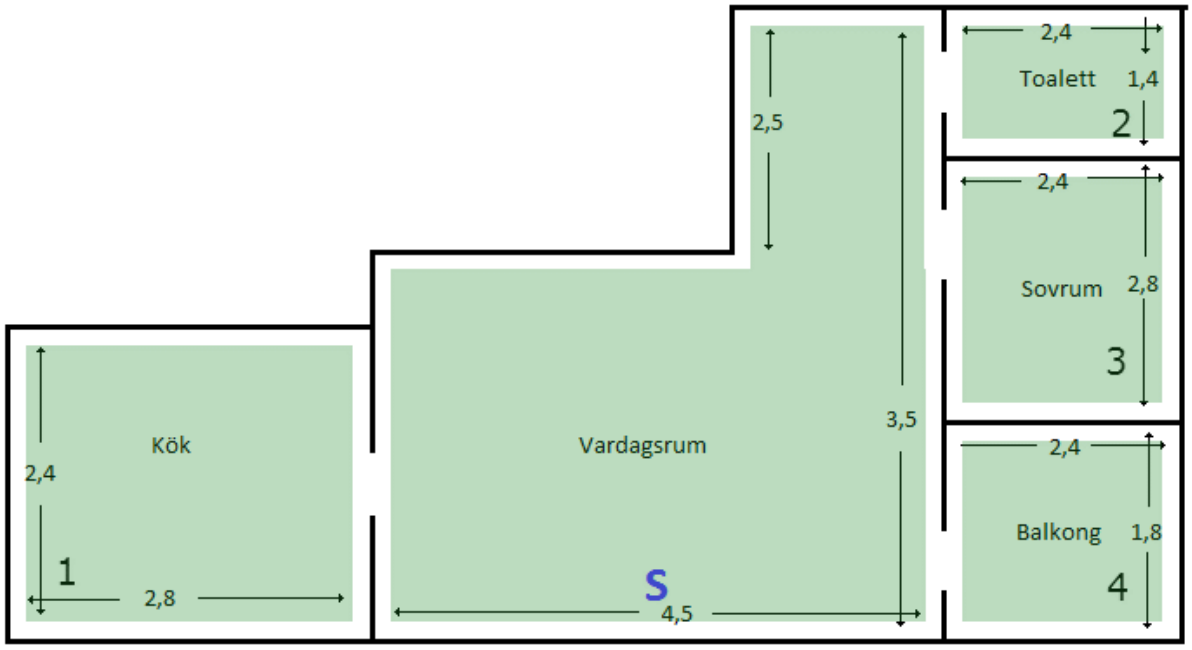
Mät punkt	Mät värden	Avrundat medelvärde	Medelvärde	Avstånd till sändare (approx)
1	[2,3,3,3,3]	3	2,8	13,0m
2	[2,3,3,3,3]	3	2,8	8,0m
3*	[3,1,1,]	n/a	n/a	8,0m
4	[1,1,1,1,1]	1	1,0	7,0m
5	[1,1,1,1,2]	2	1,2	4,0m
6	[1,1,1,1,1]	1	1,0	4,0m

Tabell 6 - Mätvärden modern villa, Ovanvåning, Dålig placering av sändare

* Extra mätpunkt för att få en tydligare bild

5.5 Lägenhet (Våning 7) - Bra placering av sändare

Lägenheten har relativt korta avstånd och med en bra placering av sändaren blev avståndet till den mätpunkt längst bort 7,0m. Väggarna består av betong och det finns flera stycken olika WIFI signaler.



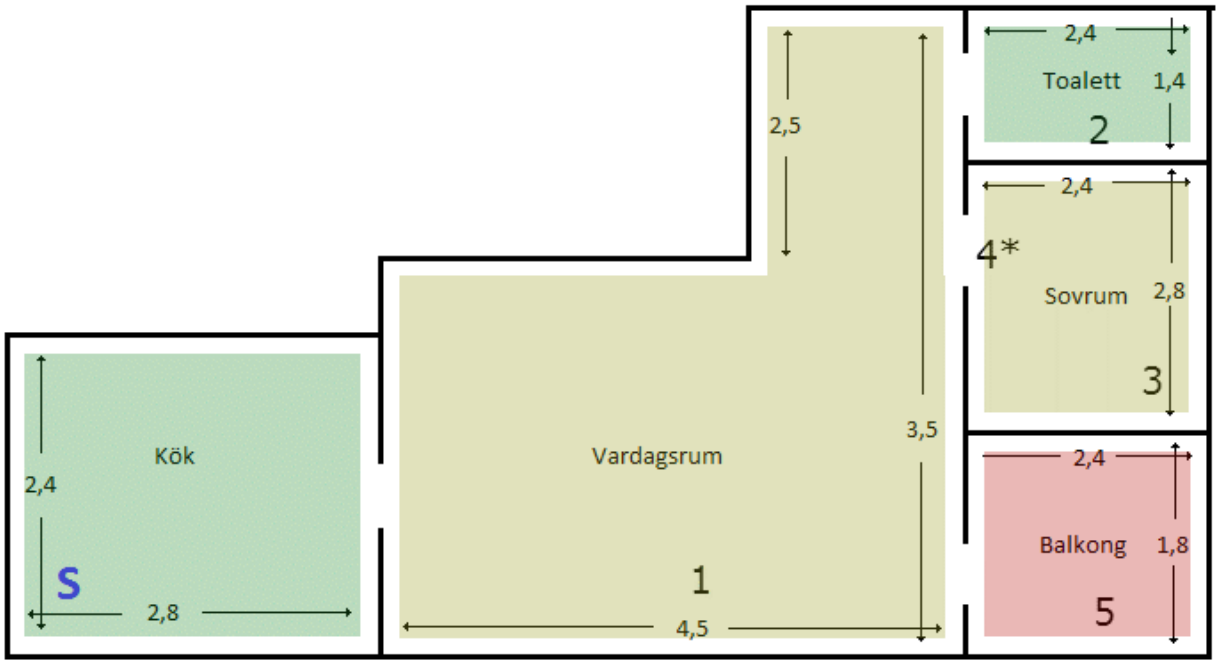
Figur 15 - Lägenhet, Betong, Våning 7, S = Sändare, Bra placering av sändare

Mät punkt	Mät värden	Avrundat medelvärde	Medelvärde	Avstånd till sändare (approx)
1	[1,1,1,1,1]	1	1,0	5,5m
2	[1,1,1,1,1]	1	1,0	7,0m
3	[1,1,1,1,1]	1	1,0	5,0m
4	[1,1,1,1,1]	1	1,0	3,5m

Tabell 7 - Mätvärden Lägenhet, Våning 7, Bra placering av sändare

5.6 Lägenhet (Våning 7) – Dålig placering av sändare

Lägenheten har relativt korta avstånd och med en dålig placering av sändaren blir avståndet till den mätpunkt längst bort 13,0m. Väggarna består av betong och det finns flera stycken olika WIFI signaler.



Figur 16 - Lägenhet, Betong, Våning 7, S = Sändare, Dålig placering av sändare

Mät punkt	Mät värden	Avrundat medelvärde	Medelvärde	Avstånd till sändare (approx)
1	[1,2,1,1,1]	2	1,2	5,5m
2	[1,1,1,1,1]	1	1,0	13,0m
3	[1,2,2,2,1]	2	1,6	10,0m
4*	[1,1,1]	n/a	n/a	9,5m
5	[3,3,3,3,3]	3	3,0	9,0m

Tabell 8 - Mätvärden Lägenhet, Våning 7, Dålig placering av sändare

* Extra mätpunkt för att få en tydligare bild

5.7 Gammal villa (markplan) – Bra placering av sändare

Gammal villa med sändaren placerad centralt på markplan. Längsta avstånd till mätpunkt uppgår till 10,0m. Väggar består av trä och gips. WIFI signal finns.

Figur 17 – Gammal villa, Trä/Gips, Markplan, S = Sändare, Bra placering av sändare

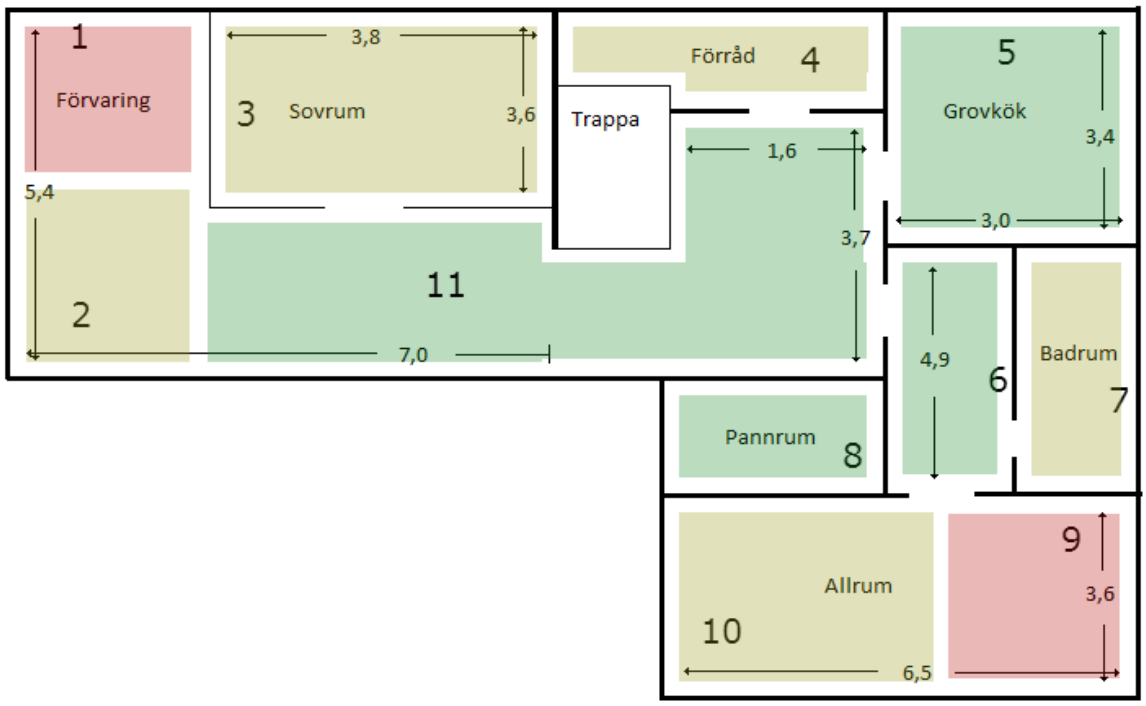
Mät punkt	Mät värden	Avrundat Medelvärde	Medelvärde	Avstånd till sändare(approx)
1	[1,1,1,1,1]	1	1.0	9.0m
2	[2,1,1,2,2]	2	1.6	10.0m
3*	[1,1,1]	n/a	n/a	8.0m
4	[1,1,1,1,1]	1	1.0	6.0m
5	[1,1,1,1,1]	1	1.0	4.0m
6	[1,1,1,1,1]	1	1.0	3.0m
7	[1,1,1,2,1]	2	1.2	10.0m
8	[1,1,1,1,1]	1	1.0	7.0m

Tabell 9 - Mätvärden gammal villa, Markplan, Bra placering av sändare

*Extra mätpunkt för att få en tydligare bild

5.8 Gammal villa (Källarvåning) – Bra placering av sändare

Källarens väggar består av sten. WIFI sändaren är placerad i källaren. Längsta avstånd till mätpunkt uppgår till 9,5m. Sändaren finns fortfarande placerad på markplan.



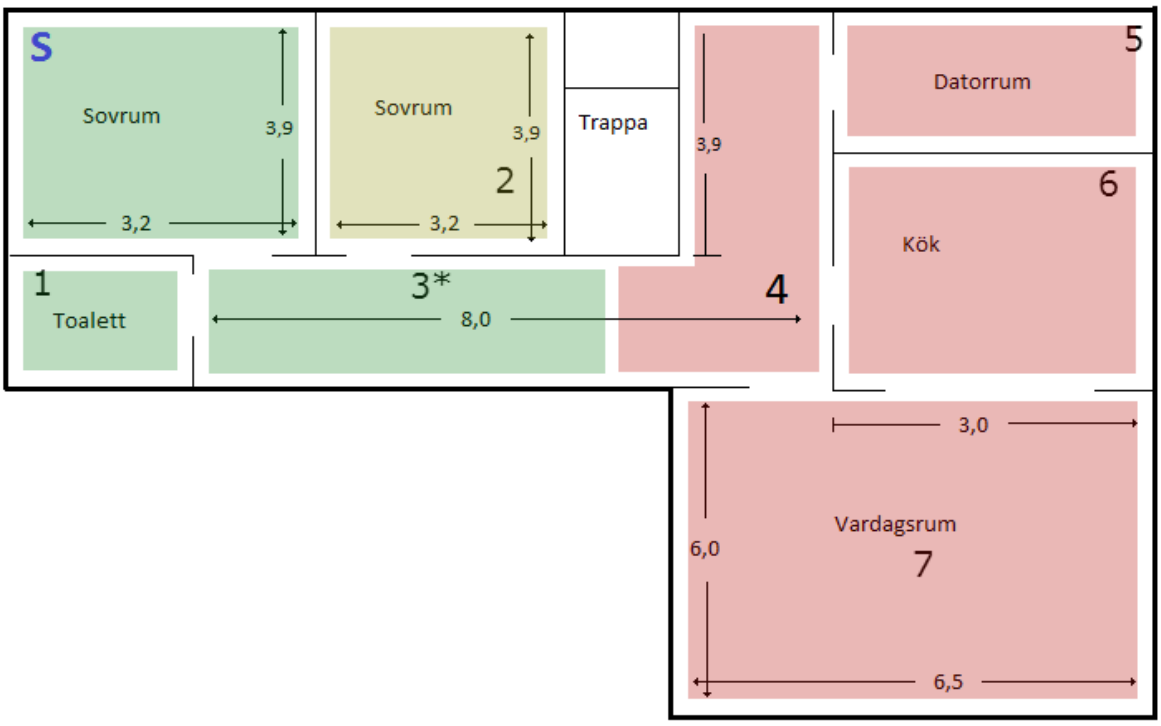
Figur 18 – Gammal Villa, Sten, Källarvåning, Sändare finns fortfaranade placerad på markplan, Bra placering av sändare

Mät punkt	Mät värden	Avrundat medelvärde	Medelvärde	Avstånd till sändare (approx)
1	[3,3,3,3,3]	3	3.0	9.5m
2	[3,1,1,1,1]	2	1.4	9.5m
3	[3,2,3,1,1]	2	2.0	7.5m
4	[3,3,1,1,1]	2	1.8	4.0m
5	[1,1,1,1,1]	1	1.0	5.0m
6	[1,1,1,1,1]	1	1.0	4.0m
7	[2,2,2,2,1]	2	1.8	5.0m
8	[1,1,1,1,1]	1	1.0	4.0m
9	[3,2,2,3,2]	3	2.4	6.0m
10	[2,1,1,2,1]	2	1.4	6.0m
11	[1,1,1,1,1]	1	1.0	5.0m

Tabell 10 - Mätvärden gammal villa, Källarvåning, Bra placering av sändare

5.9 Gammal villa (Markplan) – Dålig placering av sändare

Sändaren placerades i utkanten av bostaden för att få så långt genomsnittligt mätvärde till mätpunkt. Med denna placering blir längsta avståndet till mätpunkten 15,0m. Väggarna består av trä och gips. WIFI signal finns.



Figur 19 – Gammal villa, Trä/Gips, Markplan S = Sändare, Dålig placering av sändare

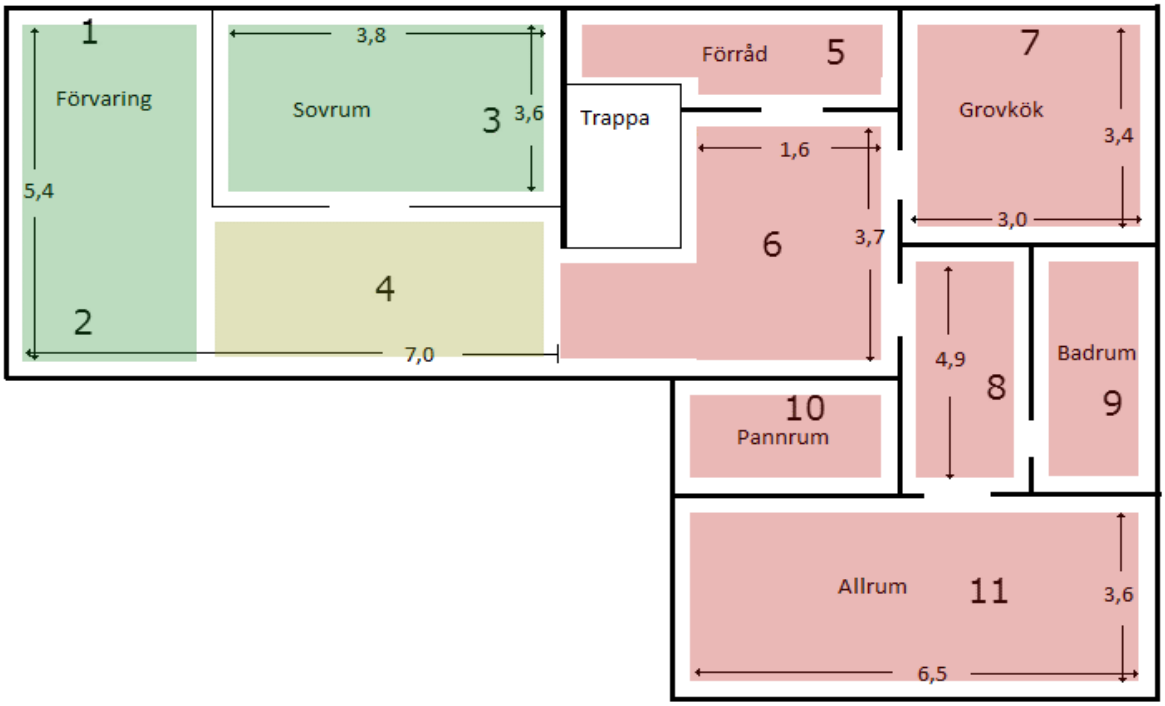
Mät punkt	Mät värden	Avrundat Medelvärde	Medelvärde	Avstånd till Sändare (approx)
1	[1,1,1,1,1]	1	1.0	4,0m
2	[1,2,1,1,1]	2	1.2	6,5m
3*	[1,1,1]	n/a	n/a	5,0m
4	[2,2,3,3,2]	3	2.4	11,0m
5	[3,3,3,3,3]	3	3.0	13,0m
6	[3,3,3,3,3]	3	3.0	13,0m
7	[3,3,3,3,3]	3	3.0	15,0m

Tabell 11 - Mätvärden gammal villa, Markplan, Dålig placering av sändare

*Extra mätpunkt för att få en tydligare bild

5.10 Gammal villa (Källarvåning) – Dålig placering av sändare

Längsta avstånd till mätpunkt uppgår till 17,0m. Källarens väggar består av sten. WIFI-enheten finns placerad i källaren.



Figur 20 – Gammal villa, Sten, Källarvåning, Sändare finns fortfarande på markplan, Dålig placering av sändare

Mät punkt	Mät värden	Avrundat Medelvärde	Medelvärde	Avstånd till Sändare (approx)
1	[1,1,1,1,1]	1	1.0	2.0m
2	[1,1,1,1,1]	1	1.0	6.0m
3	[1,1,1,1,1]	1	1.0	7.5m
4	[2,1,1,1,1]	2	1.2	5.5m
5	[1,2,2,3,3]	3	2.2	9.0m
6	[3,3,3,3,3]	3	3.0	9.0m
7	[3,3,3,3,3]	3	3.0	11.0m
8	[3,3,3,3,3]	3	3.0	12.0m
9	[3,3,3,3,3]	3	3.0	13.0m
10	[3,3,3,3,3]	3	3.0	11.0m
11	[3,3,3,3,3]	3	3.0	17.0m

Tabell 12 - Mätvärden gammal villa, Källarvåning, Dålig placering av sändare

5.11 Utvärdering av testresultat

I kapitel 11.11 utvärderas mjukvaran och fälttesternas resultat.

5.11.1 Mjukvaran

Mjukvaran fungerar som önskat enligt flödesschema i figur 9 men vid verkliga tillämpningar kan extrema fall uppnås. Exempel på detta skulle kunna vara att kölistan blir full. Det har för testernas skull antagits räcka med en plats på 20 för att tillgodose funktionaliteten. I den verkliga produkten kan denna kölista behövas ändras.

5.11.1.1 Kölistan

Följande frågor måste behandlas då det potentiellt utsätter systemet för problem:

- Hur stor ska den rimligt vara för att undvika problem?
- Vad händer om kölistan överskrids?

I och med att kölistan är fördefinierad till ett visst antal platser finns alltid risken att den blir full och värden går förlorade. Företag bör göra undersökningar på hur många värden dem tror kommer att behöva sparas som mest innan de kan föras över till telefonen.

5.11.1.2 Anslutning

Följande fråga måste behandlas då det potentiellt utsätter system för problem:

- Vad händer om anslutning bryts mitt i en sändning?

Protokollet hanterar felsändningar eller avbrutna sändningar men mjukvarulogiken i våran egen lösning måste undvika att data skrivs över. Det är mycket viktigt att se till att data aldrig förloras. En funktion som håller reda på om sändningen lyckades eller inte bör implementeras för att förhindra att problem uppstår i sådana lägen.

5.11.1.3 Realtidsklockan

Följande fråga måste behandlas då det potentiellt utsätter system för problem:

- Vad händer om realtidsklockan börjar gå fel?

Om realtidsklockan börjar gå fel kan tidsstämpeln bli felaktig. Detta problem skulle kunna lösas genom att implementera en noggrannare realtidsklocka eller låta den befintliga realtidsklockan synka sig efter mobiltelefonens klocka.

5.11.2 Fältstudien
Produkten förväntas inte behöva en kontinuerlig anslutning. Det räcker med att anslutning kan upprättas kortare tidsperioder för att man ska kunna föra över data. Därför bedömer vi enligt våran egen definierade betygsskala:

- 1,0 = m – Väldigt bra
- 1,0 ≤ m < 1,5 – Bra
- 1,5 ≤ m < 2,5 – Fungerar
- 2,5 ≤ m < 3,0 – Dåligt
- 3,0 = m – Ej funktionellt

5.12 Gammal villa
Störningskällor och byggnadsinformation:

- Byggdes på 40 talet. Tegelfasad, med trä och gipsväggar. Källare med stenväggar.
- Huset har WIFI.

5.12.1 Bra placering
Bra placering av sändare bedömdes vara i mitten av husets ovanvåning.

5.12.1.1 Ovanvåning
Vid avstånd mindre än 10,0 meter blev resultatet väldigt bra, eller bra. De rum som låg på längsta avstånd fick något sämre resultat men fortfarande inom godkända ramar för produktens behov.

Antalet väggar bedöms även påverka resultatet.

5.12.1.2 Källarvåning
Vid avstånd mindre än 5,0 meter blev resultatet Väldigt bra vilket bedöms vara rimligt då endast korta avstånd kan gå genom taket. Avstånd större än 5,0 meter blir tvingar signalen att gå igenom källarväggar där de tappar mycket styrka.

Kringliggande rum, avstånd mindre än 10,0 meter avstånd och större än 5,0 meter får ändå resultat mellan Bra och Fungerar.

Mätpunkt 1 fick betyget 3.0 vilket innebär att anslutning troligtvis inte kan göras.

5.12.2 Dålig placering
Dålig placering av sändare bedömdes vara i husets yttre sovrum.

5.12.2.1 Ovanvåning
Vid avstånd mindre än 6,5 meter blev anslutningen Väldigt bra eller Bra. Vid mätpunkt 4 blev anslutningen Fungerande även om den tappat i styrka.

Avstånd mindre än 11,0 meter bedömdes Ej funktionella. Detta på grund dels sträckan men även de många väggarna.

5.12.2.2 Källarvåning
Endast avstånd mindre än 7,5 meter får en Väldigt Bra eller Bra anslutning därefter faller anslutningsmöjligheterna snabbt. Detta antas vara dels det långa avståndet men även det höga antal väggar långa sträckor måste passera.

Källarväggar är också tjockare och av annorlunda material.

5.13 Modern villa
Störningskällor och byggnadsinformation:

- Byggdes 2010. Träfasad, med trä och gipsväggar. Ovanvåning med trä och gipsväggar.
- Huset har WIFI.

5.13.1 Bra placering
God placering av sändaren bedömdes vara hallen i mitten av markplan.

5.13.1.1 Markplan
Avstånd mindre än 7,0 meter gav Mycket bra anslutningsmöjligheter. Inga avstånd större än 7.0 meter fanns att mäta.

5.13.1.2 Ovanvåning
Avstånd mindre än 8,0 meter gav Mycket bra anslutningsmöjligheter. Inga avstånd större än 8.0 meter fanns att mäta.

5.13.2 Dålig placering
Dålig placering av sändaren bedömdes vara i utkanten av huset på markplan nära WIFI sändaren.

5.13.2.1 Markplan
Avstånd mindre än 7,0 meter gav Mycket bra anslutningsmöjligheter. Avstånd större än 7.0 meter och mindre än 8,0 meter ger Fungerande anslutning.

5.13.2.2 Ovanvåning
Avstånd mindre än 7,0 meter gav Mycket bra eller Bra anslutningsmöjligheter. Avstånd större än 7.0 meter gav Dåliga anslutningsmöjligheter.

Avstånd mindre än 7,0 meter på ovanvåning gav fler väggar att passera.

5.14 Lägenhet
- Byggd på -60talet. Betong Väggar. Endast en våning.
- Huset har flera WIFI sändare.

5.14.1 Bra placering
God placering av sändaren bedömdes vara i mitten av lägenheten, vardagsrum.

5.14.1.1 Våning 7
Avstånd mindre än 7,0 meter gav Mycket bra anslutningsmöjligheter. Inga avstånd större än 7.0 meter fanns att mäta.

5.14.2 Dålig placering
Dålig placering av sändaren bedömdes vara i utkanten av lägenheten, köket.

5.14.2.1 Våning 7
Avstånd mindre än 5,0 meter gav Mycket bra anslutningsmöjligheter. I övrigt gav mät punkterna skilda resultat. Väggar och andra möjliga störningskällor verkar vara en större faktor här. Notera att de flesta rum fortfarande har Bra eller Fungerande anslutningsmöjligheter.

5.14.3 Felkällor
Mät metoden bör göras med känsligare instrument och under mer kontrollerade former där exakta avstånd och förhållanden kan bestämmas.

Olika tillfälliga störningar och speciella förhållanden kan inte undvikas.

Höga antal mätningar ger alltid en bättre bild av situationen och skulle vara rekommenderat. Fler testfall bidrar också till en mer tillförlitlig bild.

6 Slutsats

I kapitel 7.3.1 identifierades de problem som skulle lösas för Finn Medicinen. Genom detta arbete har dessa frågeställningar besvarats och kommer att redogöras i detta kapitel.

6.1 Vilka aspekter bedöms vara viktiga för produktens trådlösa kommunikation?

Först definierades de aspekter som är viktiga i allmänhet för trådlös dataöverföring. Hastighet, räckvidd, energiförbrukning, bandbredd, etablering och behov av basstation. I samråd med Finn Medicinen bestämdes de viktigaste punkterna för deras produkt nämligen:

- Behovet av att klara sig utan basstation
- Energiförbrukning
- Räckvidd

Avsaknaden av basstation är förmågan att kommunicera direkt med smartphone. Detta bestämdes vara den viktigaste punkten. Dels för den eventuella kostnaden för produkten men även för förmågan att röra sig utanför hemmet med produkten. Denna aspekt är den viktigaste fördelen Bluetooth hade över Zigbee och Z-Wave.

Utöver detta kommer energiförbrukning som för alla trådlösa enheter är väsentlig. BLE har låg energiförbrukning och stöder olika alternativ av ”Sleep modes” för reducering av energiförbrukning. Jämförelser med ZigBee och Z-Wave energiförbrukning har inte gjorts. Från källorna i teknisk bakgrund kan referensförbrukningar för olika ”Sleep modes” utläsas. Bedömning att möjligheten till optimering för låg energiförbrukning för BLE är tillräckligt. Om lägre energiförbrukning önskas senare är ZigBee och Z-Wave intressanta alternativ.

Hög räckvidd ökar tillgängligheten för enheten. BLE, Zigbee och Z-Wave har alla samma optimala räckvidd. Zigbee och Z-Wave som båda stöder Mesh-nätverk kan enklare öka sina respektive räckvidder i hemmet om flera noder kan göras aktuella.

6.2 Vilket protokoll vore lämpligast för Finn Medicinens produkt?

BLE är ett lämpligt protokoll för Finn Medicinens produkt framför allt med avseende på avsaknaden av behovet av basstation. Om räckvidd i hemmet och/eller energiförbrukning visar sig vara otillräckliga är ZigBee och Z-Wave intressanta alternativ.

6.3 Bör ett färdigt utvecklingskit eller eget val av mikroprocessor och modul väljas?

Efter en kort undersökning antas att för ett företag som Finn Medicinen är ett färdigt kit bäst. För egen implementering av själva modulen krävs hög teknisk kompetens såväl tid, pengar och möjlighet att konstruera/beställa egna integrerade chip efter egen design. Denna uppgift hade varit svår och tidskrävande att utföra tillräckligt väl för ett examensarbete.

6.4 Hur skulle en implementation av mjukvaran se ut?

Kod finns bifogat i bilagorna. Under metodik finns beskrivning av implementationen och hur för produkten den bedömdes behöva se ut.

6.5 Vilka problem har uppstått på grund av avsaknad av företagets färdiga produkt?

Att ha en färdig produkt och en färdig applikation skulle gjort tolkningen och implementationen av den trådlösa kommunikationen enklare då exaktare behov kunnat definieras.

Noggrannare tester skulle också kunna utföras vilket skulle ha gett en bättre bild av produktens beteende i verkliga situationer. Dessa testresultat hade då bättre kunnat motivera förbättringar och problem som måste hanteras. Alla funktioner hade då eventuellt kunnas implementerats.

6.6 Hur går man tillväga för att ta fram en sådan implementation när arbetsmiljön är okänd?

Att först studera arbetsmiljön, lägga tid på att tolka och definiera uppgiften är väl värt sin möda. Vid förstudier bör källor till information sammanställas; utbildningsvideos, utbildningsdokument, forum med mera. Denna period bör däremot planeras noga med ett fixerat tidsschema för att undvika att projektets moment stannar upp.

6.7 Hur väl kommer en implementation med det valda protokollet hantera normala situationer?

Denna implementation fungerar väl med avseende på de fälttester vi utfört. Produktens behov och önskade funktioner visar att endast små mängder data kommer behöva föras över och kontinuerliga anslutningar kommer inte heller behövas. Fälttesterna visar att anslutningar kommer kunna göras i större delarna av bostäderna med tid för minst två manuella överföringar. En applikation förväntas kunna jobba betydligt snabbare än våra manuella tester vilket bör resultera i att anslutning och överföring bör hinnas med. Detta ger slutsatsen att data med sannolikhet kommer kunna föras över förutsatt att smartphone rör sig i huset även om den vid en tidpunkt inte är kapabel att upprätta anslutning och göra överföring.

Även om implementationen kan hålla data tills överföring är möjlig måste företaget rationalisera hur många avläsningar som rimligt skall kunna lagras. I nuläget kan 20 stycken lagras och utrymme för fler bedöms inte vara ett problem. Man kan enkelt utöka kölistan.

Räckvidd kan undersökas om den kan ökas via signalstyrkan och känsligheter på antennen.

Inga mätningar över energi förbrukningen har gjorts och inga implementationer försöker begränsa den. Olika ”Sleep Modes” kan implementeras för både PSoC BLE och PRoC BLE men för det behövs hela produkten och applikationen för att mer exakt kunna avgöra vilken ”Sleep Mode” som är lämplig. Se teknisk bakgrund, 8.3.3 ”Sleep Modes”.

6.8 Vilka avstånd kan företaget vänta sig från sändaren?

BLE modulen klarar med hög sannolikhet upprätta en anslutning om avståndet är mindre än 10,0 m. och rimliga väggar innefattas.

Över 10,0 m blir anslutningen ostabil och kan inte hållas länge. Däremot så kan den hållas tillräckligt länge för att hinna göra två manuella överföringar. Med detta motiveras att en applikation på en Smartphone bör hinna göra en delvis överföring av all sparad data på en anslutning.

Näst efter avstånd är väggar och hinder det största bekymret. Dessa är i sin tur beroende av tjocklek, antal och material. För specifika krav på väggar måste en mer noggrann studie, ej fältstudie, genomföras.

Våningsskillnader är bara ett problem om en av våningar är en källarvåning eller om avståndet ökar.

Inget av de tre exempel vi utfört våra fälttester för har haft sträckor >20.0m och det vanliga avståndet ligger på ~10m med 1-3 väggar.

Med avseende på vår tolkning av den slutgiltiga produkten bedöms BLE med denna modul som en lämplig metod för Finn Medicinens produkt.

6.9 Har uppgiften klarats?

Uppgiften av Finn Medicinen gick ut på tre olika saker. Dels skulle det levereras en kommunikationslösning som uppfyllde de krav som fastställts. Kommunikationen skulle kunna leverera information från sensorerna i Dosboken till en smartphone via Bluetooth. Företaget ville också veta hur bra denna kommunikation skulle fungera i en verklig situation.

Eftersom Finn Medicinens egenutvecklade applikation inte blivit klar blev den slutgiltiga testningen något modifierad. Istället för att testa mot deras applikation fick vi använda oss av Cypress egna CySmart applikation till PC samt deras BLE Dongel. Själva produkten Dosboken fanns inte heller tillgänglig så riktiga sensorvärden fanns inte att tillgå. Istället valdes att testa med fejkade värden vilket bedömdes, för testning av kommunikationen, fungerade likvärdigt. Eftersom att arbetet inte handlar om att behandla sensorvärden snarare än att se att värden kommer över som dem ska.

Under alla tester som körts under fälttestningen har samtliga funktioner i lösningen testats varje gång.
Under testerna uppkom aldrig ett avvikande värde i sändning av data och tidsstämpel samt att flaggans funktion alltid fungerade. Slutsatsen blir därför att med hög sannolikhet fungerar funktionerna.

Den andra delen av arbetet handlade om att visa hur pass bra lösningen blev. Tanken fälttesterna var att för olika boendeformer visa på hur bra kommunikationen skulle fungera. Hur nära måste produkt och telefon hålla sig förhållande till varandra för att man ska kunna säkerställa att de kan synkronisera sin information med varandra. Genom att utföra fälttester med bra respektive dålig positionering av sändaren hoppades man kunna få en rimlig uppskattning av BLE sändarens prestanda. Genom att göra grova skisser på planritningar och

sedan rita in färgkodningar för sändningskvalitén fås ett mycket enkelt och lättöverskådligt sätt att visa hur resultatet blev. För detta arbetes omfattning fick det räcka med tre olika typer av hus och två olika placeringar av sändare och med mätningar från alla rum med fem test för varje mätpunkt. Om man hade velat ha en ännu noggrannare bild över kvalitén på sändningen och mer detaljerade övergångar skulle man kunna göra om testerna med ännu fler mätpunkter i såväl liknande förutsättningar men även andra typer av boende.

Man måste förstå att dessa mätresultat visar endast på sändnings kvalité för just dessa hus. Men eftersom att vi valt hus av olika karaktär, gammaldags hus av tegel/sten/trä, modernt hus av gips/trä samt en höghuslägenhet av betong, representerar dessa val väl de vanliga boendeformer som finns. Även om lokala skillnader kan förekomma.

En tredje sak Finn Medicinen gärna ville veta var hur bra denna lösning står sig mot liknande produkter. Om man läser på Bluetooth specifikationen så ska BLE ha en optimal sändningsradie på 100 meter. Dessa tester visar att i hus och lägenheter försämras sändningsradien avsevärt av alla hinder och störningar. Det blir således svårt att jämföra olika fabrikat med varandra då de optimala värdena som de skriver ut inte stämmer med den verkligheten man faktiskt tillämpar produkterna på. Det man skulle kunna jämföra är sändningsräckvidder för produkterna under deras optimala förutsättningar. Men den undersökningen kommer inte betyda så mycket för det finns inget som säger att den jämförelsen kommer att stämma i den verkliga tillämpningen.
För att göra en riktig jämförelse av andra produkter så skulle man behöva göra implementeringar av alla de produkter man vill jämföra med och sedan utföra samma tester för alla lösningarna. Inte förrän det är gjort kan man med säkerhet visa på vilka eller vilken som fungerar bäst.

6.10 Framtida utvecklingsmöjligheter och optimeringar

6.10.1 Egengjord applikation

Den initiala tanken var att utveckla en enkel applikation med verktyget MIT App Inventor 2.0. Men det visade sig att det verktyget inte är uppdaterat för Bluetooth Low Energy. Avsaknad av kunskap i vanlig applikationsutveckling för Android eller iOS system uteslöt möjligheten att göra en egen. Företagets egen applikation blev dessutom försenad och det gjorde att ingen riktig applikation fanns att testa mot. För testernas skull hade det kanske varit bättre om tester kunde testats mot den riktiga applikationen.

6.10.2 Flera tester

För att öka noggrannheten och mer exakt kunna se var gränserna för sändnings kvalitéer gick kunde man lagt fler mätpunkter. Det bedömdes dock att den mängden som valts räckte för detta arbete.
Men vill man kan man enkelt ta denna produkt och återskapa dessa tester i samma eller andra hus för att få en ännu mer detaljerad bild.

6.10.3 Bättre signalstyrka
Om räckvidden på signalen inte anses som tillräcklig bör man utforska möjligheten att antingen öka signalstyrkan eller känsligheten på antennen.

6.10.4 Bättre kölista
I denna lösning har vi en kölista för att hantera alla sensoravläsningar. Eftersom att enheten inte alltid är inom räckhåll för telefonen är det viktigt att ha en kölista som kan hantera alla värden som kommer. Man skulle behöva se över denna kölista och försäkra sig om att den är tillräckligt lång. Ett annat alternativ skulle kunna vara att utveckla en självväxande kölista som själv anpassar sig efter mängden data den ska innehålla.

6.10.5 Batterioptimering
En väldigt väsentlig sak för BLE är energi. En sak som inte tagits upp i detta arbete men som skulle kunna erbjuda längre batteritider för den färdiga produkten är en sådan funktion. Som en vidareutveckling skulle man kunna ta våran lösning och optimera den med kod för att lägga produkten i ”Sleep Modes” mellan uppgifter. Detta sätt skulle spara energi.

6.10.6 Utökad felhantering
För att erbjuda en bättre felhanteringsmöjlighet är det rätt vanligt att man lägger till utskrifter av felmeddelande i realtid. Detta skulle vara något som hade varit bra om man planerar att fortsätta vidareutveckla detta program.

6.10.7 Analog till digital
Istället för att låta sensortillverkaren göra en funktion som läser av sensorerna kunde man utveckla denna själv. Denna funktion kan ges till sensortillverkarna så att dem inte behöver göra sin egen funktion.

7 Terminologi

API	Application Programming Interface
BLE	Bluetooth Low Energy
Dosboken	Finn Medicinens elektroniska dosett
Dosett	Läkemedelslåda med indexering för dag och tid
Finn Medicinen	Företaget var exjobbet gjordes
GAP	Generic Access Profile
GATT	Generic Attribute Profile
IDE	Integrated Development Environment
I2C	Inter-Integrated Circuit
MINC	Malmö baserad incubator
PWM	Pulse With Modulation
PRoC	Programmable Radio on Chip
PSoC	Programmable System on Chip
RTC	Real Time Clock
SPI	Serial Peripheral Interface
Verilog	Hårdvarubeskrivande språk
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

8 Referenser

[1] http://www.z-wave.com/about [2016-04-15]

[2] Computer Networking – A Top Down Approach sixth edition, ISBN13;978-0-273-76896-8, James F.Kurose, Keith W.Ross, s552 [2016-04-15]

[3] http://www.wi-fi.org/discover-wi-fi/security[2016-04-15]

[4] http://www.telegesis.com/about-us/zigbee-overview/ [2016-04-15]

[5] http://www.kjell.com/se/fraga-kjell/hur-funkar-det/mobilt/mobil-kommunikation/mobilnaten [2016-04-15]

[6] http://www.zigbee.org/zigbee-for-developers/network-specifications/ [2016-04-15]
http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeepro/ [2016-04-15]
http://www.z-wave.com/about [2016-04-15]
https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy [2016-04-15]

[7] https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics [2016-04-15]

[8] https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction [2016-04-15]

[9] Bluetooth Low Energy The Developer's Handbook [ISBN-13: 978-0-13-288836-3] October 2012 [2016-04-15]

[10] https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap [2016-04-15]

[11] https://cdn-learn.adafruit.com/assets/assets/000/013/826/medium800/microcontrollers_ConnectedTopology.png?1390836031 [2016-04-15]

[12] https://cdn-learn.adafruit.com/assets/assets/000/013/828/medium800/microcontrollers_GattStructure.png?1390836057 [2016-04-15]

[13] https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt [2016-04-15]

[14] https://www.bluetooth.com/membership-working-groups/why-how-to-become-a-member [2016-04-15]

[15] http://www.cypress.com/sites/default/files/inline/fckImages/PRoC_EZ-BLE_Module_popup.jpg [2016-04-15]

[16] http://www.cypress.com/products/psoc-4-ble-bluetooth-smart [2016-04-15]

[17] http://www.cypress.com/products/proc-ble-bluetooth-smart [2016-04-15]

[18] http://www.cypress.com/documentation/frequently-asked-questions/ble-frequently-asked-questions "PSoC and PRoC BLE" "What is the difference between PRoC BLE and PSoC 4 BLE" [2016-04-15]

[19] http://www.cypress.com/products/psoc-4-ble-bluetooth-smart "Certifications" [2016-04-15]

[20] http://www.cypress.com/file/121271/download, s 2-5 [2016-04-15]

[21] http://www.cypress.com/products/psoc-creator-integrated-design-environment-ide [2016-04-15]

[22] http://www.cypress.com/documentation/software-and-drivers/cysmart-bluetooth-le-test-and-debug-tool [2016-04-15]

[23] http://www.cypress.com/file/139326/download , s 44 [2016-04-15]

9 Bilagor

9.1 Källkod

9.1.1 Main.c

```
/* ========================================
 *
 * Copyright YOUR COMPANY, THE YEAR
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF your company.
 *
 * ========================================
*/

/*
* ================================== *
* |        Inlcudes           | *
* ================================== *
*/

#include <project.h>
#include <updateGattDAtaBase.h>
#include <main.h>


/*
* ================================== *
* |     Hjälpvariabler för      | *
* |     Skrivning till GATT     | *
* ================================== *
*/

CYBLE_GATT_HANDLE_VALUE_PAIR_T handleValue;
CYBLE_GATT_HANDLE_VALUE_PAIR_T handleValueFlag;
CYBLE_CONN_HANDLE_T connHandle;
CYBLE_CONN_HANDLE_T connHandleFlag;
CYBLE_GATT_ERR_CODE_T apiResult;
CYBLE_GATT_HANDLE_VALUE_PAIR_T handleValueFlagInsert;


/*
* ================================== *
* |       Hjälpvariabler        | *
* ================================== *
*/

uint8 kalle[2];
uint8 flag;
CYBLE_GATTS_WRITE_REQ_PARAM_T *wrReqParam;
int cnt = 1; // För räknare att lägga in timestamp i avbrottsfunktionen
```

```
/*
* ================================= *
* |      RTC-initeringar        | *
* ================================= *
*/

uint8 i; // Hanterar en for-sats i sysTick

CY_ISR_PROTO(SysTickIsrHandler);

#define TIME_HOUR          (0x00u)
#define TIME_MIN           (0x00u)
#define TIME_SEC           (0x00u)
#define TIME_HR_MIN_SEC     ((uint32)(TIME_HOUR << RTC_HOURS_OFFSET) | \
                        (uint32)(TIME_MIN << RTC_MINUTES_OFFSET)    | \
                         TIME_SEC)

#define DATE_MONTH         (RTC_MARCH)
#define DATE_DAY           (0x22u)
#define DATE_YEAR          (0x1845u)
#define DATE_MONTH_DAY_YEAR ((uint32)(DATE_MONTH << RTC_MONTH_OFFSET)   | \
                        (uint32)(DATE_DAY << RTC_DAY_OFFSET)       | \
                         DATE_YEAR)

#define SYSTICK_EACH_10_HZ  (10u)
#define SYSTICK_RELOAD      (CYDEV_BCLK__SYSCLK__HZ / SYSTICK_EACH_10_HZ)


/*
* ================================= *
* |      Avbrottsfunktion       | *
* |       Clock: 12000 HZ       | *
* |      Counter: 12000 HZ      | *
* ================================= *
*/

CY_ISR(isr_Handler){

   /* Enkel metod för att skriva in tio tidsstämplar*/
   if (cnt <= 10){
      createTimeStamp();
      cnt++;
   }

 /*
* ================================= *
* |      FLAGGAVLÄSNING         | *
* |      FLAGGSKRIVNING         | *
* ================================= *
*/
   apiResult = CyBle_GattsReadAttributeValue(&handleValueFlag, &connHandleFlag,
CYBLE_GATT_DB_LOCALLY_INITIATED);
   flag = *handleValueFlag.value.val;

   if (apiResult == CYBLE_GATT_ERR_NONE){
      if (flag == 0){
         updateGattDataBase();
      }
   }
   /* Rensar interupt så att vi inte fastnar här */
```

```
    Timer_ClearInterrupt(Timer_INTR_MASK_TC);
}


 /*
* ================================== *
* |      StackHandler           | *
* |    Här hanteras alla        | *
* |   saker som kan inträffa    | *
* |   under kommunikationen     | *
* ================================== *
*/

void stackHandler(uint32 event, void *eventParam){

    switch(event){
        /* Om stacken är aktiverad */
        case CYBLE_EVT_STACK_ON:




        /* Om enheten inte har anslutning */
        case CYBLE_EVT_GAP_DEVICE_DISCONNECTED:
            PWM_WriteCompare(500);

            /* Startar en advertising så att enheten kan bli upptäckt */
            CyBle_GappStartAdvertisement(CYBLE_ADVERTISING_FAST);
            break;

        /* Om enheten har anslutning */
        case CYBLE_EVT_GAP_DEVICE_CONNECTED:
            PWM_WriteCompare(1000);

        /* Om enheten har fått en begäran om att något skall skrivas in i GATT-Databasen */
        case CYBLE_EVT_GATTS_WRITE_REQ:

            wrReqParam = (CYBLE_GATTS_WRITE_REQ_PARAM_T *) eventParam;
            if(CYBLE_FLAG_ISREADY_CHAR_HANDLE == wrReqParam->handleValPair.attrHandle){

                handleValueFlag.attrHandle = wrReqParam->handleValPair.attrHandle;
                handleValueFlag.value.val = wrReqParam->handleValPair.value.val;
                handleValueFlag.value.len = 1;
                CyBle_GattsWriteAttributeValue(&handleValueFlag, 0, &cyBle_connHandle,
CYBLE_GATT_DB_LOCALLY_INITIATED);
                }

            CyBle_GattsWriteRsp(cyBle_connHandle);
            break;
    }
}

/*
* ================================== *
* |   SysTick Avbrottsfunktion    | *
* ================================== *
*/

void SysTickIsrHandler(void)
```

```
{
    RTC_Update();
}


/*
* ================================ *
* |     Mainfunktionen          | *
* ================================ *
*/

int main()
{
    /* Initiering av vektorerna i kölistan */
    /*Inskrivning av data till kölistan */

    /* Startar komponent Timer */
    Timer_Start();
    PWM_Start();

    /* Startar avbrottsfunktionen */
    isr_StartEx(isr_Handler);

    /* Enable global interrupts. */
    CyGlobalIntEnable;
    CyBle_Start(stackHandler);

    /* RTC + SyStick */


    CySysTickStart();

    /* Configure SysTick timer to generate interrupt every 100 ms */
    CySysTickSetReload(SYSTICK_RELOAD);

    for (i = 0u; i < CY_SYS_SYST_NUM_OF_CALLBACKS; ++i)
    {
        if (CySysTickGetCallback(i) == NULL)
        {
            /* Set callback */
            CySysTickSetCallback(i, SysTickIsrHandler);
            break;
        }
    }


    RTC_Start();
    RTC_SetDateAndTime(0x00000000, 0x20160514);
    RTC_SetPeriod(1u, SYSTICK_EACH_10_HZ);
    intialiseGattDataBase();

    for(;;)
    {
        CyBle_ProcessEvents();
    }
}

/* [] END OF FILE */
```

9.1.2 Main.h

```
/* ========================================
 *
 * Copyright YOUR COMPANY, THE YEAR
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF your company.
 *
 * ========================================
*/

/* [] END OF FILE */
```

9.1.3 updateGattDatabase.c

```
/* ========================================
 *
 * Copyright YOUR COMPANY, THE YEAR
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF your company.
 *
 * ========================================
*/

/*
* ================================== *
* |        Includes            | *
* ================================== *
*/
#include <project.h>
#include <updateGattDataBase.h>
#include <main.h>

void createTimeStamp(); //tasbort?


/*
* ================================== *
* |     Hjälpvariabler för      | *
* |     Skrivning till GATT     | *
* ================================== *
*/

CYBLE_GATT_HANDLE_VALUE_PAIR_T handleValue;
CYBLE_CONN_HANDLE_T connHandle;
CYBLE_GATT_HANDLE_VALUE_PAIR_T handleValueFlagg;


/*
* ================================== *
* |       Hjälpvariabler        | *
* ================================== *
*/

uint8 storage[20][4];
uint8 isReadyFlag;
uint8 tempArray[4]; //Tas bort?

/*
* ================================== *
* |       RTC Variabler         | *
* ================================== *
*/

uint32 time;
uint32 date;
RTC_DATE_TIME dateTime;
```

```
uint8 timeHourMinSec[3];
CYBLE_GATT_HANDLE_VALUE_PAIR_T handleTimeHour;
CYBLE_CONN_HANDLE_T connHandle;
uint32 timeDate[20][2];
uint8 dataTimeDate[11];
uint16 year;

/*
* ================================== *
* |     Initierar vektorn som    | *
* |     ska fungera kölista      | *
* ================================== *
*/

void intialiseGattDataBase(){
   for (int i = 1; i<21; i++){
      for(int j = 0; j<4; j++){
         storage[i][j] = 0;
      }
   }

   insertStorage(1,1,1,1);
   insertStorage(2,2,2,2);
   insertStorage(3,3,3,3);
   insertStorage(4,4,4,4);
   insertStorage(5,5,5,5);
   insertStorage(6,6,6,6);
   insertStorage(7,7,7,7);
   insertStorage(8,8,8,8);
   insertStorage(9,9,9,9);


   isReadyFlag = 1;
   handleValueFlagg.attrHandle = CYBLE_FLAG_ISREADY_CHAR_HANDLE;
   handleValueFlagg.value.len = 1;
   handleValueFlagg.value.val = &isReadyFlag;
   CyBle_GattsWriteAttributeValue(&handleValueFlagg, 0, &cyBle_connHandle,
CYBLE_GATT_DB_LOCALLY_INITIATED);
}


/*
* ================================== *
* |       Funktion för att       | *
* |       skapa tidsstämpel      | *
* ================================== *
*/

void createTimeStamp(){
   /*Kallas var gång ett värde förs in i storage för att ge en tidstämpel till var mätning*/

   RTC_GetDateAndTime(&dateTime);
   time = dateTime.time;
   date = dateTime.date;
   for(int i = 0; i<19; i++){
      if(timeDate[i][0] == 0){
         timeDate[i][0] = time;
         timeDate[i][1] = date;
         i = 20;
      }
```

```
    }

}

/*
* ================================== *
* |       Funktion för att       | *
* |     uppdatera databasenl     | *
* ================================== *
*/

void updateGattDataBase(){
    /*Kallas om flaggan isReady = 0*/
    /*Hjälpvektorn dateTimeDate [data][data][data][data][ss][mm][hh][dd][mm][yy][yy] används för inskrivning
till gattDatabasen */

    /* Data */
    dataTimeDate[0] = storage[0][0];
    dataTimeDate[1] = storage[0][1];
    dataTimeDate[2] = storage[0][2];
    dataTimeDate[3] = storage[0][3];

    /* tid */
    dataTimeDate[4] = RTC_GetSecond(timeDate[0][0]);
    dataTimeDate[5] = RTC_GetMinutes(timeDate[0][0]);
    dataTimeDate[6] = RTC_GetHours(timeDate[0][0]);

    /* datum */
    dataTimeDate[7] = RTC_GetDay(timeDate[0][1]);
    dataTimeDate[8] = RTC_GetMonth(timeDate[0][1]);
    dataTimeDate[9] = (((RTC_GetYear(timeDate[0][1])>>8)&0xff));
    dataTimeDate[10] = (((RTC_GetYear(timeDate[0][1])>>0)&0xff));

    /* handleTimeHour tilldelas önskade värden; destination, aktuell storlek, aktuelt värde*/
    handleTimeHour.attrHandle = CYBLE_MEDDELANDE_MED_CHAR_HANDLE;
    handleTimeHour.value.len = 11;
    handleTimeHour.value.val = dataTimeDate;

    CyBle_GattsWriteAttributeValue(&handleTimeHour, 0,&cyBle_connHandle,
CYBLE_GATT_DB_LOCALLY_INITIATED);

    /* isReadyFlag sätts för att markera att gattdatabasen är uppdaterad*/
    isReadyFlag = 1;
    handleValueFlagg.attrHandle = CYBLE_FLAG_ISREADY_CHAR_HANDLE;
    handleValueFlagg.value.len = 1;
    handleValueFlagg.value.val = &isReadyFlag;
    CyBle_GattsWriteAttributeValue(&handleValueFlagg, 0, &cyBle_connHandle,
CYBLE_GATT_DB_LOCALLY_INITIATED);

    /*Matriserna Storage och timeDate updateras */
    updateStorage();

}

/*
* ================================== *
* |       Funktion för att       | *
* |       att sätta in värde     | *
* |         i kölistan           | *
```

```
* ================================== *
*/

void insertStorage(uint8 rowOne, uint8 rowTwo, uint8 rowThree, uint8 rowFour){
/*Vid sensor avläsning placeras datan i Storage*/
/*Talen skall uttryckas som decimalatal*/

for (int i = 0; i <20; i++){
   if (storage[i][0] == 0){
      storage[i][0] = rowOne;
      storage[i][1] = rowTwo;
      storage[i][2] = rowThree;
      storage[i][3] = rowFour;
      i = 20; // avslutar for satsen
   }
}

}

/*
* ================================== *
* |       Funktion för att       | *
* |      uppdatera kölistan      | *
* ================================== *
*/

void updateStorage(){
   /* j = rader , i = kolumner*/
   /*Update Storage skiktar fram alla värden ett steg för både storage och timeDate*/

   for (int i = 1; i<20; i++){
      for(int j = 0; j<4; j++){
         storage[i-1][j] = storage[i][j];
      }
   }
   for(int i= 0; i<20; i++){
      for (int j = 0; j<2; j++){
         timeDate[i-1][j] = timeDate[i][j];
      }
   }
}

/* [] END OF FILE */
```

9.1.4 updateGattDatabase.h

```
/* ========================================
 *
 * Copyright YOUR COMPANY, THE YEAR
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF your company.
 *
 * ========================================
*/

/* [] END OF FILE */
#include <project.h>

/*
* ================================================ *
* |     Funktioner för updateGattDataBase       | *
* ================================================ *
*/

/*
* ============================================== *
* |    Hämtar värden från matriserna          |*
* |     storage och timeDate                  |*
* ============================================== *
*/
void updateGattDataBase();

void insertStorage(uint8 rowOne, uint8 rowTwo, uint8 rowThree, uint8 rowFour);

void updateStorage();

void intialiseGattDataBase();

void createTimeStamp();
```