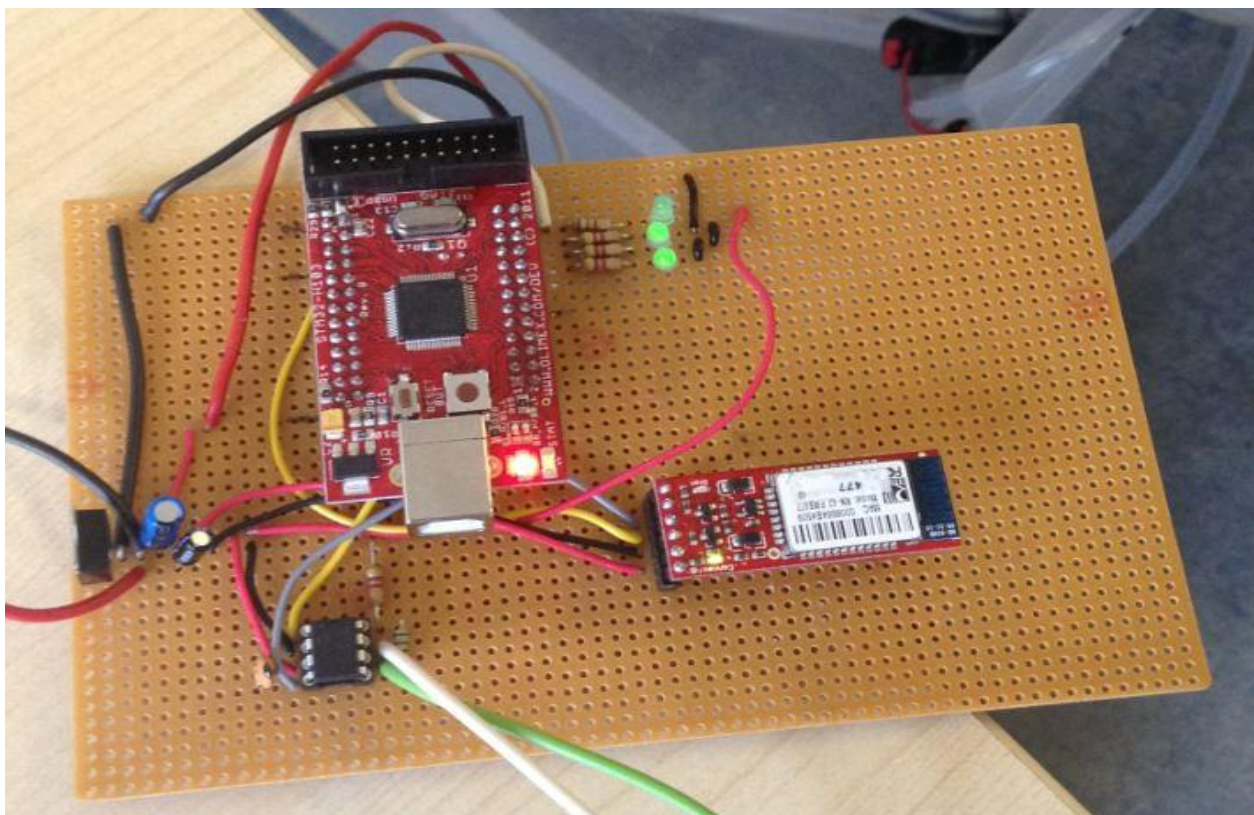




# CHALMERS

---



## Bränsleövervakningssystem för marina fartyg

## Fuel monitoring system for marine vessels

Examensarbete inom Högskoleingenjörsprogrammet Mekatronik

Daniel Johansson  
Tobias Olsson

Fuel monitoringsystem for marine vessels  
DANIEL JOHANSSON  
TOBIAS OLSSON

©DANIEL JOHANSSON, 2014.

©TOBIAS OLSSON, 2014.

Department of Computer Science and Engineering  
Chalmers University of Technology  
SE-412 96 Göteborg  
Sweden  
+46 31 772 10 00

Göteborg, Sweden 2014

## FÖRORD

Detta projekt har genomförts av två studenter från Mekatronikprogrammet på Chalmers. Denna rapport är en del i examensarbetet som utförts under läsperiod 2 på företaget Broccoli Engineering AB. Arbetet motsvarar 30 högskolepoäng, 15 per vardera student. Examensarbetet skrevs för Institutionen Data- och Informationsteknik med Lars Svensson som examinator. Ett flertal personer har varit bidragande till examensarbetets slutresultatet och framförallt skulle vi vilja tacka Björn Bergholm VD för Broccoli Engineering AB och uppdragsgivare samt vår handledare på Broccoli Peter Jansson. Likaså har vår handledare på Chalmers, Sakib Sisteck, varit bidragande med många nya infallsvinklar och idéer, som vi är mycket tacksamma för.

Daniel Johansson & Tobias Olsson

## **Sammanfattning**

Rapporten beskriver utvecklingen av ett tekniskt system för att mäta och presentera bränsleförbrukning hos marina fartyg. Det slutliga målet är en demo som är användarvänlig och kan ge korrekta mätvärden av ett fartygs bränsleförbrukning. Projektet innefattar inte installation och implementation av systemet i ett fartyg men utvecklas med detta i åtanke. I denna rapport beskrivs projektets tillvägagångssätt och tankegången kring utvecklingen. Resultatet mynnade ut i en demo där kraven uppfylldes. För vidareutveckling anses utökad interaktion med NMEA 2000, ytterligare konfigurerbara parametrar och utveckling av användargränssnittet som de största punkterna.

## **Abstract**

This report focus on the development of a system that allows the user to monitor the fuel consumption of a marine vessel. The aim with this project is a functional demo that is user friendly and reliable. The showing results should be accurate to the fuel consumption of that vessel. To install this demo in a marine vessel is not a part of the project but the development will take this in consideration. Through this report the approach and background of the project is described from beginning to the final result. The result was a demo that meets all requirements for this project but leaves some opinions on improvements that should be considered if further development would take place. The main areas are judged to be more cooperation between NMEA 2000 and the system, additional parameters with configurable and further development of the Android application.

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
1.1	Bakgrund . . . . .	1
1.2	Syfte . . . . .	1
1.3	Mål . . . . .	1
<b>2</b>	<b>Teknisk bakgrund</b>	<b>2</b>
2.1	Microcontroller . . . . .	2
2.2	UART . . . . .	2
2.3	Bluetooth . . . . .	3
2.4	Flödesmätare . . . . .	3
2.5	CAN/NMEA 2000 . . . . .	4
2.5.1	CAN . . . . .	4
2.5.2	NMEA 2000 . . . . .	5
2.6	CANalyzer . . . . .	6
2.7	Android . . . . .	7
<b>3</b>	<b>Metod</b>	<b>8</b>
<b>4</b>	<b>Demoutveckling</b>	<b>9</b>
4.1	Val av komponenter . . . . .	9
4.1.1	Processor/Utvecklingskort . . . . .	9
4.1.2	Bluetoothmodul . . . . .	10
4.2	Val av utvecklingsmiljö . . . . .	12
4.2.1	Utvecklingsmiljö för beräkningsenhet . . . . .	12
4.2.2	Utvecklingsmiljö för Androidapplikation . . . . .	12
4.3	Hårdvara . . . . .	13
4.3.1	Utvecklingskort . . . . .	13
4.3.2	Flödesmätare . . . . .	13
4.3.3	Kommunikationer . . . . .	14
4.3.4	Bluetoothmodul . . . . .	14
4.4	Mjukvara . . . . .	15
4.4.1	Flödesmätning . . . . .	15
4.4.2	Ta emot CAN/NMEA 2000 . . . . .	15
4.4.3	Haversine beräkning . . . . .	16
4.4.4	Skicka CAN/NMEA 2000 . . . . .	17
4.4.5	Androidapplication . . . . .	17

4.4.6	Bluetooth för Android . . . . .	18
4.5	Utveckling av Bluetoothprotokoll . . . . .	18
4.6	Tester . . . . .	19
4.6.1	Flödestestning . . . . .	19
4.6.2	Positionstestning . . . . .	20
4.6.3	Kommunikationstestning . . . . .	20
<b>5</b>	<b>Resultat</b>	<b>21</b>
5.1	Konfigurerbarhet . . . . .	21
5.2	Demo . . . . .	22
<b>6</b>	<b>Diskussion</b>	<b>23</b>
6.1	Utökat utvecklingsarbete . . . . .	23
6.1.1	Konfiguration . . . . .	23
6.1.2	Lagring av data . . . . .	23
6.2	Miljöaspekter . . . . .	23
6.3	Viskositet . . . . .	24
6.4	Utvärdering av examensarbete . . . . .	24
6.4.1	Den tänkta arbetsmetodiken . . . . .	24
6.4.2	Vad kunde ha gjorts annorlunda . . . . .	25
<b>A</b>	<b>Bluetooth Protokoll</b>	<b>I</b>
<b>B</b>	<b>Flödesschema</b>	<b>IV</b>
<b>C</b>	<b>Elschema</b>	<b>V</b>
<b>D</b>	<b>Planeringsrapport</b>	<b>VI</b>

# Förkortningar

ACK - Acknowledge

API - Application Programming Interface

ARM - Branschledande inom utveckling utav processorarkitekturer för inbyggda system och bärbara enheter.

CAN - Controller Area Network

CAPL - Communication Access Programming Language, Programspråk som används i Vector produkter t.ex. CANalyzer. [11]

CRC - Cyclic Redundancy Check

EWARM - Embedded Workbench for ARM, en produkt från IAR Systems

GPIO - General Purpose Input Output

GPS - Global Position System

J1939 - Populärt CAN protokoll som används i bland annat lastbilar och marina fartyg [2]

NMEA - National Marine Electronics Association

PCB - Printed Circuit Board

PGN - Parameter Group Number

UART - Universal Asynchronous Receiver Transceiver

WIFI - En typ av WLAN "Wireless Local Area Network"

XML - Extensible Markup Language



# 1 Inledning

## 1.1 Bakgrund

Företaget är ett konsultföretag som i dagsläget har stora kunder inom fordonindustrin men vill bredda sin kunskap och få en teknisk lösning som kan intressera kunder inom marinteknik. Uppdraget syftar därför till att kunna läsa och eventuellt skicka data via NMEA 2000, en databuss som är vanlig i dagens båtar. Idén till projektet kom från företagets VD som kände behovet av att övervaka bränsleförbrukning i fritidsbåtar på ett enkelt sätt. För att åstadkomma ett resultat utan några stora ingrepp i båten så ville man kunna övervaka det hela trådlöst. Resultat ska helst vara mycket skalbart och konfigurerbart efter användarens önskemål och förutsättningar.

## 1.2 Syfte

Uppdraget består i att utveckla ett system där data från flödesgivare och GPS används för beräkning av bränsleförbrukning i marina fartyg. De analyserade dataflödet ska sedan till viss del bevaras i systemet samt relevant information ska redovisas med hjälp av ett Android-gränssnitt som ansluts trådlöst. Modulen för beräkningar skall vara konfigurerbar och kunna fungera utifrån olika förutsättningar när de gäller givare och datatillgångar.

## 1.3 Mål

Målet med projektet är att kunna uppvisa en demo, bestående av en mikroprocessor som mäter ett bränsleflöde och tar in GPS-signaler via en Androidenhet alternativt NMEA 2000. Dessa signaler ska processas för att sedan sända resultatet trådlöst till en Androidenhet som presenterar statistiken.

## 2 Teknisk bakgrund

Detta kapitel innehåller en kort teoretisk sammanfattning av tekniska delar som förekommer i rapporten och under projektet. Grundläggande förståelse för dessa delar kan vara nödvändigt för att få en förståelse för projektet och dess omfattning.

### 2.1 Microcontroller

Microcontrollers är vanligt förekommande i små inbyggda system för att övervaka eller styra processer. Microcontrollers kan vara lite olika uppbyggda och ha lite olika struktur för att möta kunders olika behov och krav. Men gemensamt för microcontrollers är att de måste programmeras på ett eller annat sätt för att åstadkomma någon uppgift. Vanligtvis används programmeringsspråk så som C eller assembler för att skriva program på en PC som sedan kan skrivas över och köras på en microcontroller. Under projektet har microcontrollern STM32F103 använts, en controller som bygger på ARM Holdings processorstruktur, Cortex M3. En 32-bitars struktur som tillåter klockfrekvenser på upp till 72 Mhz vilket ger snabba beräkningar och kort operationshantering. STM32F103 har ett stort antal inbyggda periferienheter för kommunikation och tidshantering, exempel på detta är CAN, USART och timers [10].

### 2.2 UART

UART står för Universal Asynchronous Receive Transmit och är en seriell kommunikationsstandard för digitala enheter. Eftersom UART sänder data asynkront krävs det endast två förbindelser mellan de kommunicerande enheterna, en för att sända data och en för att ta emot data. För synkron kommunikation krävs ytterligare förbindelser för att enheterna skall kunna synkronisera sina klockor, då kallas standarden USART. Det möjliggör för sändning av längre meddelanden än de oftast 8 bitar men ibland 9 som går att skicka med asynkron kommunikation. Hastigheten för att överföra data är standardiserad, där det finns nivåer för att sända hundratals bitar per sekund till miljontals bitar per sekund[7].

## 2.3 Bluetooth

Bluetooth använder sig av radiovågor för att kommunicera mellan olika enheter och kräver att det finns minst två enheter för att upprätta kommunikation, båda enheterna behöver någon form av antenn för att kunna sända och ta emot data. Radiofrekvensen som Bluetooth använder sig av är inom ett intervall på 2,4 till 2,485 GHz, detta är frekvenser som i de flesta länder är licensfritt att använda. Att Bluetooth använder sig av ett intervall med frekvenser istället för låsa sig till en fast frekvens är för att undvika störningar när flera enheter försöker sända samtidigt. Enheterna känner av vilka frekvenser andra enheter i närheten använder sig av och anpassar sig ständigt till lediga frekvenser. Detta ger en stabilare och effektivare kommunikation även om flera enheter i närheten nätverkar samtidigt. Bluetooth är en standard för att kommunicera korta sträckor, från 1 till 100 meter, och är indelat i tre olika klasser. Klass 3 har ett kommunikationsavstånd på ca en meter medan den mer populära klass 2 enheten klarar av upp till ca 10 meter och en klass 1 enhet klarar av kommunikation upp till 100 meter. Bluetooth begränsas inte rent fysiskt till dessa korta avstånd utan har med avseende på energiförbrukning valts att begränsas för att bättre lämpas åt mobila och trådlösa enheter. Tack vare att Bluetooth-protokollet är anpassat för att hålla en låg energiförbrukning och därmed begränsats till korta kommunikationsavstånd har det blivit ett populärt protokoll för mobila/trådlösa enheter. Den låga kostnaden för Bluetoothtekniken är också en bidragande orsak till det växande utbudet av enheter som utnyttjar tekniken [5].

## 2.4 Flödesmätare

Den flödesmätare som användes i projektet var utvald av uppdragsgivaren och var av typen som ger ett bestämt antal pulser per liter som passerar igenom rotorhuset. Vid standardutförande så genereras 2500 pulser/liter och med en strypning som medföljde höjdes noggrannheten till per 8500 pulser/liter [14].

## 2.5 CAN/NMEA 2000

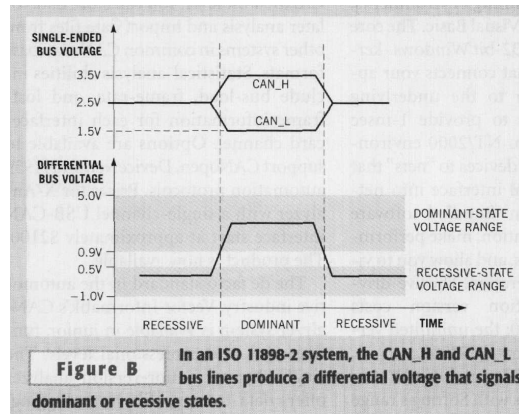
### 2.5.1 CAN

CAN, Controller Area Network, är en standard för hur processorer kan skicka information mellan varandra. Tekniken utvecklades av företaget Bosch och standarden presenterades 1986 [4]. Den lägger stor tyngdpunkt på kommunikation där felsända och icke fullständiga meddelande ska minimeras. CAN specificerar det fysiska och datalänk lagret, dessa lager sätter standarden för hur data skickas rent fysiskt med etta och nolla samt paketering av data i frames.

För att kunna koppla upp en enhet på en fältbuss av CAN-typ så krävs det två hårdvarukomponenter. Den första är en CAN-controller som ofta är implementerad i enheter som är avsedda för kommunikation via CAN-bussar. Controllern ser till att meddelanden paketeras enligt CAN protokollets specifikationer. Meddelandena kallas oftast för frame och det finns fyra typer av frames, Data frames, Remote frames, Error frames och Overload frames. Dessa frames har lite olika uppbyggnad och syfte men under detta projekt har fokus endast varit på Data frames. En Data frame innehåller information från sändaren som kan vara av intresse för andra enheter, en frame adresseras alltså inte till en mottagare utan skickas ut på bussen så att samtliga noder kan avläsa meddelandet. Det som beskriver vad meddelandet innehåller är Identifier, en del av ramen som avgör vilken enhet som har skickat och vad meddelandet innehåller. Identifier kan vara antingen 11 eller 29 bitar lång beroende på vilken mjukvarustandard som använts, t.ex. NMEA 2000, J1939 eller CANopen. Utöver Identifier innehåller ett meddelande även delar som Control, Data, CRC och ACK. Control består av 6 bitar varav 2 bitar är reserverade och 4 bitar talar om hur mycket data meddelandet innehåller. Meddelandets datainnehåll kan variera i storlek mellan en till åtta bytes, därav definieras detta i Control. CRC och ACK används av CAN för att kontrollera huruvida störningar och fel har påverkat meddelandet, se tabell 2.1 [2].

Data frame				
11(standard) / 29(extended)	6	0-64	16	2
Identifier	Control	Data	CRC	ACK

Tabell 2.1: Utformning av CAN-protokollets Data frame



Figur 2.1: Elektrisk signal enligt signalstandarden ISO 11898-2

Den andra hårdvarukomponenten som krävs är en CAN-transceiver som anpassar signalerna ut från enheten till någon av CANs signalstandarder. En vanligt förekommande signalstandard för CAN är ISO11898, denna standard tillåter en överföringshastighet upp till 1Mbit/s och använder sig av två ledningar, CANH och CANL. Det är spänningsskillnaden mellan CANH och CANL som avgör hurvida det är en etta eller nolla som skickas på bussen, då spänningsskillnaden är ca 2 volt skickas en nolla och då ingen spänningsskillnad är på bussen skickas en etta, se figure 2.1. Denna standard är det som tillämpats enligt NMEA 2000 och därmed även under detta projekt [6].

### 2.5.2 NMEA 2000

NMEA 2000 är ett högre protokoll för CAN som beskriver hur den frame som ska skickas får se ut. Den har stora likheter med protokollet J1939 som används för tyngre fordon men NMEA 2000 är anpassad för marin miljö. Likt J1939 använder sig NMEA 2000 av 29-bitars Identifier och delar upp den i tre delar, prioritet, PGN och källa. Identifierns första tre bitar anger meddelandets prioritet, följande 18 bitar anger meddelandets PGN som uppger vad meddelandet innehåller. Identifierns sista åtta bitar anger vilken enhet som skickar meddelandet, källan bakom meddelandet [6].

## 2.6 CANalyzer

CANalyzer är ett analyseringsverktyg som företaget använder på sina datorer för att övervaka och skicka kommunikation på CAN-buss. För att kunna koppla upp en dator mot CAN krävs en hårdvara till exempel Kvaser USBcan Professional, se figur 2.2 alternativt referens [13], som CANalyzer jobbar mot. Programmet har många konfigureringsmöjligheter för att hantera data och meddelande, t.ex. kan meddelande filtreras för att bara visa de som användaren är intresserad av. Det finns även möjlighet att skriva script som utför olika operationer som är händelsestyrda, dvs. att ett visst meddelande mottagits eller användaren har genom en knapptryckning aktiverat en händelse. Möjligheten att skriva denna typ av script ger användaren möjligheter att kunna simulera olika händelser och scenarion. Men för den som inte vill skriva script är det enkelt att analysera och logga de meddelanden som går på bussen utan programmeringskunskaper [1].



Figur 2.2: Kvaser USBcan Professional för övervakning av CAN-buss

## 2.7 Android

Android är ett operativsystem som är utvecklats i första hand för mobila enheter så som smartphones och tablets. Android Inc. köptes upp utav Google 2005 som sedan dess lett utvecklingen av plattformen och i dagsläget innehar Android omkring 80% av marknaden för smartphones [3]. Operativsystemet är baserat på Linux och är open source, vilket betyder att plattformen får modifieras och sedan distribueras i enlighet med GNU-manifestet [15]. För utveckling av applikationer har Android en egen framtagen utvecklingsmiljö, Android SDK som står för: Software Development Kit, där programutvecklingen sker till största del i JAVA. För utveckling av grafiska gränssnitt i Android används XML mestadels. Det mesta av plattformen bygger på händelsestyrd programmering där systemet reagerar på yttre källor så som gester på enhetens skärm [10].

### 3 Metod

Vid projektets uppstart togs en projektplan fram för att åskådliggöra tidsrymden för projektet och hur denna tid skulle fördelas på projektets olika faser. Planeringsrapporten satte upp mål för varje vecka vilket gav en bra uppfattning om hur projektet låg till tidsmässigt. Projektets genomförande skedde i olika faser, den första fasen var att identifiera lämplig hårdvara för projektet. Hårdvaran skall uppfylla de krav som ställs på projektet och under rimlig tid gå att införskaffa samt implementera. Efter att lämplig hårdvara har kommit tillhanda tar nästa fas vid, lämplig mjukvara för att kunna programmera och testa projektet skall identifieras utifrån ställda krav. När detta är klart kan utvecklingen påbörjas och de första enheterna kopplas samman. Genom att dela upp projektet i tre delsystem, CAN, flödesmätning och Bluetooth, kunde varje del testas var för sig vilket underlättade felsökning. När varje del testats och fungerade kunde delarna knytas samman i mjukvaran för att så småningom bli en fungerande demo.

Under hela utvecklingen har olika tester körts kontinuerligt för att minska antalet fel och kontrollera att mjukvaran genomför rätt operationer. En dialog har kontinuerligt förts med handledaren på Broccoli, Peter Jansson, för att stämma av att resultaten stämmer överens med företagets förväntningar. Utifrån företagets önskemål har även en blogg förts på företagets hemsida under projektets gång, en form av dokumentation som varit tillgänglig för företags anställda samt allmänheten.

Projektet har utförts i Broccolis lokaler i Göteborg och företaget har tillhandahållit alla de verktyg och komponenter som varit nödvändiga för projektet.



## 4 Demoutveckling

Detta kapitel beskriver projektets tillvägagångssätt när det gäller att ta fram både hård- och mjukvara för den demo projektet kräver. Från val av komponenter till de slutliga testerna av systemet.

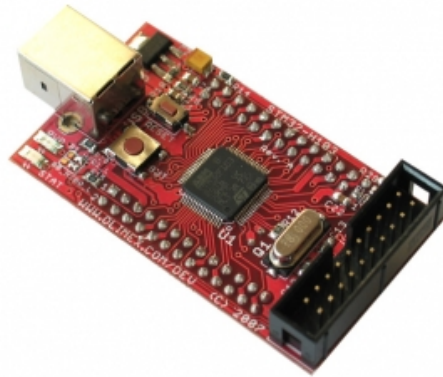
### 4.1 Val av komponenter

#### 4.1.1 Processor/Utvecklingskort

Den inledande delen av projektet bestod av undersökning kring projektets omfattning och identifiera de krav som processorn måste uppfylla. Processorn skall kunna mäta flödet från en flödesgivare, den skall kunna läsa och skriva på CAN-bussen och den skall kunna kommunicera trådlöst med en surfplatta eller annan bärbar enhet. Detta innebär att den processorn för projektet behöver ha minst en ingång som kan konfigureras som externt avbrott för att underlätta och garantera att alla pulser från flödesgivaren registreras i realtid. För att hantera CAN bör processorn ha ett inbyggt protokoll för att underlätta utvecklingen av mjukvaran. Den bör ha stöd för UART(serie)-kommunikation för kunna använda en extern modul för trådlös kommunikation.

Efter identifierade krav så undersöktes företagets erfarenheter kring olika processorkretsar, den krets som företaget främst använt sig av under tidigare projekt var Cortex M3 därav togs beslut att använda denna processorkrets även under detta projekt. För att kunna hålla projektets tidsram så togs beslut om att använda ett utvecklingskort med denna processorkrets. Utvecklingskorten har fördel av att de har de mest grundläggande komponenterna som krävs för att köra processorn, så som spänningsmatning och kristall, samt underlättande anslutningar för in- och utportar.

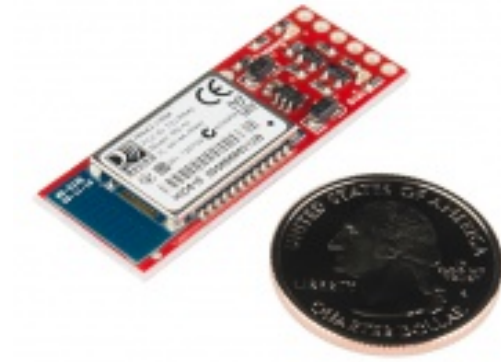
Det slutliga beslutet blev att införskaffa ett utvecklingskort från Sparkfun som byggde på en Cortex M3 krets, figur 4.1, som hade gott om anslutningsmöjligheter och en hög klockfrekvens. Kortet klarar av att köra processorn på 72MHz vilket innebär att beräkningar för distans och flöde kan göras relativt avancerade och geometriska utan att de tar märkbart lång tid, se kapitel 4.4.3.



Figur 4.1: Utvecklingskort STM32H103 från Sparkfun

#### 4.1.2 Bluetoothmodul

Efter att ha valt ett utvecklingskort för projektet som saknade trådlös kommunikation så behövs en extern modul för att lösa den uppgiften. Efter att ha granskat surfplattor och de vanligaste trådlösa enheterna så tycks Wifi och Bluetooth vara de två vanligaste standarderna för trådlös kommunikation. Då projektet inte kräver den högre dataöverföringshastighet som Wifi tillåter och inte heller dess längre räckvidd utan istället prioriterades energiförbrukning där Bluetooth är det energisnålare alternativet. Den Bluetooth som de flesta trådlösa enheter använder sig av i dagsläget är klass 2, vilket innebär en räckvidd på ca 10 meter. För detta projekt anses Bluetooth vara den kommunikation som lämpar sig bäst då 10 meters räckvidd räcker för de allra flesta båtar för privat bruk. Den Bluetoothmodul som valdes att användas under detta projekt var BlueSMiRF Silver, figur 4.2 en modul som enkelt kunde kopplas mot utvecklingskortet via seriekommunikation, UART. Detta gör det hela enkelt att skicka data trådlöst över Bluetooth.



Figur 4.2: Bluetoothmodulen BlueSMiRF

## **4.2 Val av utvecklingsmiljö**

### **4.2.1 Utvecklingsmiljö för beräkningsenhet**

Utvecklingsmiljön för projektets beräkningsenhet var tvungen att uppfylla vissa önskemål för att underlätta arbetet och därmed se till att så mycket som möjligt hinns med inom projektets tidsram. Ett önskemål var att utvecklingsmiljön ska vara kompatibel med operativsystemet företaget använder sig av, i detta fall Windows XP. Då projektet har som mål att hålla en stram budget sattes målet att hitta en utvecklingsmiljö som var så billig som möjligt men ändå uppfyllde projektets behov. Den utvecklingsmiljö som ansågs bäst lämpad för detta projekts beräkningsenhet, utifrån tidigare nämnda önskemål var IAR EWARM, ett verktyg som finns i ett flertal olika versioner varav en är gratis. Denna utvecklingsmiljö har stöd för bland annat C, C++ och assemblerspråk vilket underlättar utvecklingen då erfarenheten av dessa språk är god inom både företaget och projektgruppen.

### **4.2.2 Utvecklingsmiljö för Androidapplikation**

Även en utvecklingsmiljö för programmeringen av en Androidapplikation krävdes i projektet. Utbudet av utvecklingsmiljöer för Androidapplikationer domineras av Eclipses tillägg för Android som stöds av Androids egna hemsida där information finns om Androidspecifik mjukvara. Utöver det finns möjligheter för virtuella test för de flesta Android versionerna som släppts. Även för testning på en fysisk enhet finns det stöd för, där enda kravet är att koden är skriven för den aktuella programvaruversionen som denna använder. En annan utvecklingsmiljö som också stödjer applikationsutveckling för Android är Netbeans, men eftersom den Eclipsebaserade miljön har ett mycket stort utbud av information och material valdes den som utvecklingsmiljö för projektet.

## 4.3 Hårdvara

### 4.3.1 Utvecklingskort

Utvecklingskortet för detta projekt har många bra funktioner, några som är klart nödvändiga för projektet medan andra inte kommer fylla någon funktion men trots de så måste vissa yttre komponenter och signaler anslutas. För att enkelt göra dessa anslutningar används ett labbkort med kopparledning där komponenterna enkelt kan lödas på plats, kopparledningarna fungerar i sin tur som ledare precis som en kabel. Kopparledningarna kan enkelt kapas för att inte skapa kortslutningar eller felaktiga signaler. Utvecklingskortet som är försett med två kontaktlistor på totalt 52 anslutningar kopplas till labbkortet via motsatta kontaktlistor som löds fast på labbkortet. Detta gör alla anslutningar tillgängliga på labbkortet, dvs vi kan ansluta en ödesgivare eller CAN-signaler utan att modifiera själva utvecklingskortet. Efter att utvecklingskortet har anslutits till labbkortet kan övriga komponenter börja lödas på plats, först behövs en spänningsmatning som klarar av att driva utvecklingskortet och övriga komponenter som kan behövas. Utvecklingskortet har en spänningsregulator för att förse processorn med rätt spänningar men det behövs en stabil inspanning till kortet på 5 volt. Dessa 5 volt åstadkoms genom en spänningsregulator på labbkortet som i sin tur matas med 12 volt från spänningsaggregat eller ett batteri, spänningsregulatorn klarar stora variationer på spänningen in men ger konstant 5 volt ut.

### 4.3.2 Flödesmätare

För att kunna mäta det bränsle som fartyget förbrukar behöver bränsleslangen förses med en flödesmätare. Denna flödesmätare ska sedan anslutas till labbkortet för att kunna förse processorn med data på hur stort bränsleflödet till motorn är. Detta är en signal som ger pulser och ju oftare dessa pulser skickas från givaren ju större är flödet. Dessa pulser är perfekta att använda som avbrott på processorn då varje puls garanterat kommer detekteras, därför ansluts signalen till en inport som kan konfigureras för att hantera externa avbrott. Flödesgivaren behöver spänningsmatas och beroende på vilken spänning givaren matas med så kommer signalen ha samma spänning vid hög puls. Då utvecklingskortet klarar av signaler mellan 3-5 volt så matas därför flödesgivaren med 5 volt.

### **4.3.3 Kommunikationer**

Utvecklingskortet har en inbyggd CAN-controller men vad den gör är att den har vissa funktioner för att hantera kommunikation enligt CAN-protokollet men för att fysiskt skicka och ta emot meddelanden behövs en CAN-transceiver. Denna enhet kan översätta signaler från CAN-bussens två linor och skicka denna signal vidare som en seriekommunikation till processorn och tvärtom. Transceiver monteras till labbkortet med hjälp av en lämplig kontakt för att enkelt kunna byta enhet om den går sönder. Enheten spänningsmatas med 12 volt men ger endast 5 volts signaler så den kan direkt anslutas till tänkta in-/utportar för CAN på utvecklingskortet.

### **4.3.4 Bluetoothmodul**

För att kunna skicka data trådlöst mellan beräkningsenheten och surfplattan så behöver Bluetoothenheten anslutas till labbkortet. Detta görs genom att förse modulen med en stiftlist bestående av 6 stift och en motsatt list på labbkortet sedan kopplas modulens signaler till någon utav utvecklingskortets seriella portar, UART eller SPI.

## 4.4 Mjukvara

För att komma igång med programmering av beräkningsenheten så hämtades ett standardbibliotek med de vanligaste periferienheterna, anpassat för projektets processor och utvecklingsmiljö. Ibland standardbiblioteken fanns bland annat seriekommunikation via UART, CAN, timers och externa avbrott. Biblioteken underlättade utvecklandet avsevärt då många standardfunktioner så som skicka meddelande över CAN fanns färdiga att tillämpa. Men många av funktionerna som ingår i biblioteken kräver att man sätter sig in i och förstår hur de fungerar då många använder sig av structer och pekare.

### 4.4.1 Flödesmätning

För att detta projekt ska kunna presentera relevant data över bränsleförbrukningen behöver detta göras i realtid. För att åstadkomma detta har flertalet avbrottsrutiner använts för kommunikation och mätning av bränsleflöde. Detta kräver att avbrottsvektorer och avbrottsrutiner implementeras i mjukvaran, något standardbiblioteken hjälpte till med. Avbrott implementeras för tex CAN, UART/Bluetooth, flödesgivare osv.

### 4.4.2 Ta emot CAN/NMEA 2000

Den hårdvara som används är avsedd för CAN men beräkningsenheten måste klara av att hantera NMEA, ett protokoll som använder sig av CAN som grund. Att åstadkomma en enhet som följer NMEA 2000 standard till 100 procent ryms inte i projektets tidsram utan måste begränsas till endast nödvändiga funktioner för projektets syfte. Därför anpassas projektet till att endast lyssna på ett meddelande och till att kunna skicka ett meddelande. För att få detta att fungera måste NMEA 2000 studeras, företaget har inte tillgång till någon enhet som kommunicerar med detta protokoll därför görs endast en teoretisk studie utifrån den information som finns att få tag på internet och via databaser. Därefter skall beräkningsenheten konfigureras för att efterlikna NMEA 2000 så mycket som möjligt. Till att börja med används standardbiblioteket för CAN för att sätta upp kommunikation med rätt hastighet vilket enligt NMEA 2000 standard är 250 kbit/s. När detta är konfigurerat och fungerar så granskas vilka signaler som kan vara av intresse för att uppnå målet. För att beräkna förflyttning behövs det ett meddelande med GPS-positioner, enligt standarden skall meddelanden med den data

skickas med ett visst PGN. Genom att låta processorns CAN-controller sortera ut de meddelanden som innehåller den önskade data och därefter ger ett avbrott. När detta har åstadkommits och kontrollerats att endast relevanta meddelanden ger ett avbrott så ska informationen i meddelandet behandlas. Detta görs genom att läsa ut meddelandets Longitud och Latitude information och utifrån detta beräkna förflyttning från tidigare position.

#### 4.4.3 Haversine beräkning

Att kunna beräkna avstånd mellan GPS-positioner är nödvändigt för att kunna beräkna bränsleförbrukning med avseende på avstånd, något som är avgörande för projektet. Det finns era sätt att beräkna förflyttning utifrån GPS-positioner enligt longitude och latitude. Efter att ha undersökt de vanligast förekommande formlerna fattades beslut om att använda haversine. En formel som inte tar hänsyn till förflyttning i höjd, något som inte anses nödvändigt då tillämpningen är tänkt att användas i marina fartyg där inga större höjdskillnader förekommer. Haversine räknar däremot med jordens krökningsradie vilket gör den väldigt exakt på både långa och korta avstånd. För att genomföra dessa beräkningar i C användes standardbiblioteket math.c, detta innehåller funktioner så som sin, cos och tan [9].

$$haversine(\frac{d}{r}) = haversine(\phi_2 - \phi_1) + \cos(\phi_1) \cos(\phi_2) haversine(\lambda_2 - \lambda_1) \quad (1)$$

$$haversine(\theta) = \sin^2(\frac{\theta}{2}) \quad (2)$$

$$d = r haversine^{-1}(\frac{d}{r}) = 2r \arcsin \sqrt{\frac{d}{r}} \quad (3)$$

$$d = 2r \arcsin(\sqrt{\sin^2(\frac{\phi_2 - \phi_1}{2}) + \cos(\phi_1) \cos(\phi_2) \sin^2(\frac{\lambda_2 - \lambda_1}{2})}) \quad (4)$$

1

---

<sup>1</sup> d = distans  
r = jordradie  
 $\phi_1, \phi_2$  = latitude för position 1 & 2  
 $\lambda_1, \lambda_2$  = longitude för position 1 & 2



#### 4.4.4 Skicka CAN/NMEA 2000

När beräkningar för förflyttning och förbrukat bränsle har gjorts över samlingsintervallet skickas denna data med 1 sekunds intervall över NMEA 2000. Detta meddelande skall uppfylla standarden för att kunna läsas ut av andra enheter på bussen och för att kunna göra det måste meddelandesuppbbyggnaden granskas. För att skicka data om bränsleförbrukning enligt NMEA 2000 så skall detta paketeras i ett meddelande kallat, Engine Parameters - Dynamic, detta meddelande innehåller mängder av olika data. För att en mottagare skall kunna urskilja vad som är bränsleförbrukningen måste denna data placeras korrekt i meddelandet. Då Engine Parameters, Dynamic innehåller mer data än vad som stöds i en CAN-frame löser NMEA 2000 detta genom att segmentera upp meddelandet i era frames. Dessa frames skickas i följd med ett fast tidsintervall mellan varje, tidsintervallet är enligt NMEA 2000 protokollet 50 ms. Detta tidsintervall åstadkoms under projektet genom att skapa en delay som fördröjer varje meddelande så att de skickas med rätt tidsmellanrum.

#### 4.4.5 Androidapplication

Vid utveckling av Androidapplikationer är det två delar som ska integreras, ena delen är de XML filer för den grafiska layout som sedan kommer att visas i enheten. Den andra den javakod som binder samman XML filerna med funktioner på enheten, så som GPS, Bluetooth och datalagring. För att kunna hantera de signaler som enheten tar emot användes en avbrottshantering "Handlers" som kan hantera den kontinuerliga strömmen av mottagen data. Den Bluetoothkommunikation som enheten använde konfigurerades i enlighet med Androids Bluetooth klass. De moment som användes i detta projekt möjliggjorde för att aktivera, söka efter närliggande enheter och koppla upp sig mot den aktuella processor samt att överföra och ta emot data. I projektets mål var skalbarhet en viktig faktor och därför implementerades Androidenhetens GPS för att skapa valmöjlighet för användaren, där även tidsintervallet för GPS uppdateringar infördes. Den API version som valdes var API 8, som gör programmet kompatibelt med de flesta Androidenheter.

#### 4.4.6 Bluetooth för Android

För att använda Bluetooth i Androidapplikationer så finns det bibliotek tillgängliga där informationsutbyte mellan enheter möjliggörs och även funktioner för mer specifika ändamål så som uppkoppling mot headsets eller streama ljud. Den data som skickas mellan enheterna är paketerad enligt ett protokoll speciellt utvecklat för denna kommunikation och sänds med hjälp av standardfunktioner.

### 4.5 Utveckling av Bluetoothprotokoll

För att skicka data mellan beräkningsenhet och Androidenhet så används Bluetoothmodul. För att mottagaren skall kunna veta vad för data den tar emot och hur den skall bearbetas togs ett eget Bluetoothprotokoll för projektet fram. Detta protokoll reglerar inte hur kommunikationen sker rent fysiskt utan endast hur data skall skickas och hur den skall utläsas. Protokollet skall vara lätt att förstå och lätt implementerat i båda enheterna. Efter att har studerat CAN meddelanden och hur GPS signalen skickas där så togs beslut om att meddelande som skickas över Bluetooth bör efterlikna CAN. CAN-protokollet bygger på att varje meddelande har ett ID och max åtta bytes. Det egenutvecklade protokollet för överföring via Bluetooth ges istället en fast data mängd som är åtta bytes, dessa åtta bytes måste skickas i varje meddelande för att inte skapa problem hos mottagaren. Hur data disponerar dessa åtta bytes finns dokumenterat, se bilaga A. Då Bluetooth inte har stöd för att skicka meddelanden med ID så nyttjar protokollet för detta projektet istället en byte som motsvarar ID. Denna byte kallas i protokollet för kontrollbyte och anger vad meddelandet innehåller för data eller hur meddelandet skall tolkas. Kontrollbyten skickas som första byte för att mottagaren skall kunna identifiera meddelandet och göra rätt återgärder. Efter att protokollet testats upptäcktes att kontrollbyten i vissa fall inte utlästes korrekt då detta var första byten som skickade/togs emot, detta problem löstes genom att införa en startbyte som inte fyller annan funktion än att väcka den mottagande enheten.

## 4.6 Tester

Den första delen i processen för testning var att fastställa hur systemets funktioner skall kunna verifieras. För detta konstruerades en enklare testtrigg. Till testtriggen krävdes att data som inhämtas av systemet simuleras, där identifierades två huvudparametrar som krävdes. Dessa var flödet som flödesgivaren mäter och GPS-signalen som tillhandahölls från NMEA 2000 anslutning alternativt Androidenhetens GPS.

### 4.6.1 Flödestestning

För flödesmätning användes ett slutet system bestående av en pump som upprätthöll ett vattenflöde genom givaren där vattnet därefter returnerades i en uppsamlingsbehållare där pumpen åter cirkulerade vätskan. Flödesgivarens korrekthet verifierades genom att den kopplades samman med mikroprocessorn, som genom en avbrottsrutin summerade de pulser givaren angivit som indikation på mängden vätska som passerat. Mikroprocessorn programmerades sedan för att indikera med en diod när jämna liter passerat igenom givaren. Det testades med att en uppmätt mängd vatten tilläts rinna genom givaren.



Figur 4.3: Testtrigg för verifiering av flöde och distans

#### **4.6.2 Positionstestning**

Simuleringar av GPS-signaler via NMEA 2000 görs via ett CAPL-script som sänder ut fasta värden på CAN-bussen med ett bestämt mellanrum. Det gjordes i analysprogrammet CANalyzer. Även tester där uppmätta avstånd jämfördes med systemets mätdata gjordes för att säkerställa tillförlitligheten i Androidenhetens GPS.

#### **4.6.3 Kommunikationstestning**

Kommunikation mellan CAN-buss med NMEA 2000 protokoll och mikroprocessor samt mellan mikroprocessor till Androidenhet är de datakommunikationer som sker. Testningen för dessa signalvägar utfördes i tre steg. I det första steget testades den monterade hårdvaran, till det användes oscilloskop samt multimeter för att fastställa om signalernas korrekthet. Steg två i processen, när biblioteken för kommunikationerna implementerats, skickades förutbestämda bytes mellan enheterna det som verifiering av att ihopkopplade signalvägar. Det tredje och sista steget var att säkerställa värden mellan enheterna konverterades och mottogs på korrekt sätt. Det övervakades genom ett fönster i CANalyzers som visar alla de meddelande som skickas på CAN-bussen samt via Androidenheten där mottagna och skickade värden även visades i en textview.

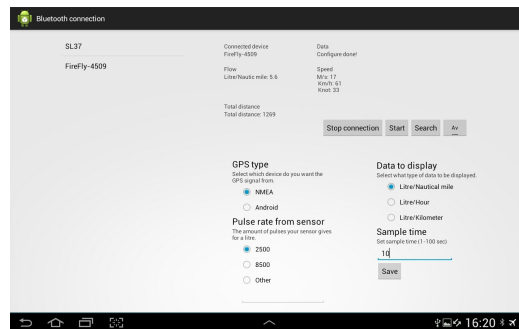
## 5 Resultat

Slutresultatet av projektet är en demo som uppfyller de krav som ställts vid uppstarten av projektet. Där ett system skulle kunna inhämta flödesdata från en flödesgivare ämnad för att mäta bränsleåtgång på en båtmotor. Datan kan tillsammans med lägesdata från en GPS via antingen NMEA 2000 eller en Androidenhet användas för att beräkna era typer av relevant statistik som användaren kan välja att visa på Androidenheten. Kommunikationen mellan Androidenheten och mikroprocessorn i systemet sker trådlöst via Bluetooth som specificerat. Vidare har målet med flexibilitet uppfyllts då ett flertal olika relevanta data kan presenteras så som bränsleförbrukning, per distans och tidsenhet.

### 5.1 Konfigurerbarhet

Ett av målen med projektet var att åstadkomma en demo som är användarvänlig och konfigurerbar. Den ska gå att anpassa utifrån användarens förutsättningar och önskemål. Variabler som är intressanta för projektets flexibilitet är inställningar för givare, demon skall enkelt kunna anpassas för att fungera med större eller mindre givare. Givaren måste fortfarande vara av samma typ, dvs. pulser per liter, men mjukvaran skall vara konfigurerbar efter antalet pulser givaren genererar per liter.

Något som användaren kan vara intresserad av att ställa in är samplingsintervallet, detta avgör hur ofta data på surfplattan uppdateras, vid korta samplingsintervall ökar risken för fel men ger ett mer direkt svar på nuvarande förbrukning. Andra parametrar som kan vara av intresse att ställa in är hur flödet skall presenteras, antingen i liter/nautisk mil, liter/timme eller liter/km. Detta ger användaren möjlighet att avgöra vilken enhet som lämpar sig bäst utifrån användningsområde och personlig önskemål. Då samtliga båtar inte är utrustade med NMEA 2000 och GPS så anpassas projektet genom att kunna hantera GPS-signal över Bluetooth. En signal som de flesta surfplattor och Androidenheter har att tillgå.



Figur 5.1: Skärmdump på Android application

## 5.2 Demo

Den demo som tagits fram under projektet har de funktioner som projektet krävde men för att implementeras i en båt krävs ytterligare tester och kunskap. Dessutom skulle labbkortet behöva skyddas genom att installeras i fukttät låda för att inte få elektriska problem. Labbkortet skulle även behöva minimeras genom en PCB design för att kallas prototyp och vara redo för produktion.

Denna demo har inte heller rätt anslutning för att kopplas upp mot NMEA 2000 utan använder sig av en anslutning för att passa testverktyget som använts för CAN.

## 6 Diskussion

### 6.1 Utökat utvecklingsarbete

#### 6.1.1 Konfiguration

Den demo som tagits fram under detta projekt tar endast hänsyn till marina fartyg med en bensinmotor, vid vidareutveckling skulle även fartyg med två motorer eller dieselmotorer kunna tas i åtanke.

För Androidapplikationen finns det utvecklingsmöjligheter både utöka valmöjligheten för inställningsbara parametrar och att integrera applikationen mer med bland annat Google olika tjänster, så som kart- och söktjänster.

#### 6.1.2 Lagring av data

Vidareutveckling av detta projekt skulle kunna innefatta lagring av data på ett SD-minne genom antingen android eller direkt från processorenheten. Detta skulle ge en samlad statistik över körningen och kunna analyseras i efterhand. Utifrån statistiken skulle distansen till tom tank kunna räknas ut om NMEA 2000 signalen från tankmätaren implementerades in i mjukvaran. Resultatet skulle ge användaren bättre överblick över sin körning och utifrån de kunna anpassa sin körning för att spara mer bränsle.

### 6.2 Miljöaspekter

Ur miljösynpunkt ger detta bränsleövervakningssystem möjlighet för användaren att med den data som presenteras optimera sin bränsleförbrukning. Utöver denna aspekt så finns flera möjligheter för att optimeringar och vidareutveckla systemet så att användaren kan ytterligare minska sin bränsleförbrukning. Mjukvaran skulle kunna kompletteras med programvara för visualisering av den rutt som är planerad och alternativa rutter som skulle ge mindre förbrukning. Även den möjligheten att utöka systemet med minne för tidigare statistik och historik skulle kunna få användaren att göra miljömedvetna val. För att bland annat minska energiåtgången vid kommunikation mellan enheterna valdes Bluetooth istället för WIFI som standard. Det tillsammans med att GPS koordinaterna kan avläsas från NMEA 2000 så undviks de mest energidränerande komponenterna i Androidenheten.

## 6.3 Viskositet

I alla försök som utfördes i testtriggen användes vatten som medium, dess viskositet ligger relativt nära bensenens värde. Men då det finns variationer i hur blandningen av bensen är mellan kontinenter och länder så kan det förekomma skillnader. Då detta sidospår ligger långt ifrån själva uppgiften har endast en enklare efterforskning genomförts för att bekräfta vattnets och bensenens snarlika viskositet. Då det finns en stor mängd båtmotorer som är av typen 2-takt där extra olja blandas i bensenen för att smörja förbränningsmotorn så förändrar även den viskositeten i mediumet. Ytterligare en faktor som kan påverka mätsystemet om det skulle anpassats för dieseldrift är att med den nya miljödieseln bildas alger som kan sätta igen bränslefilter och även flödesmätaren för projektets system.

## 6.4 Utvärdering av examensarbete

### 6.4.1 Den tänkta arbetsmetodiken

Innan projektet påbörjades gjordes en planeringsrapport som delade upp det tänkta projektet i olika delmål och moment för att enklare kunna överblicka projektets omfattning, se appendix D. Denna planeringsrapport har visat sig stämma relativt väl överens med projektets genomförandetakt och ordning med endast ett fåtal undantag. Undantagen berodde framförallt på omvärderingar och prioriteringar, t.ex. behövdes väldigt lite data sparas därför utgick momentet lagring av statistik och mer tid lades på Androidapplikationen som påbörjades tidigare än planerat. En del som krävde mer resurser än vad som var planerat, både när det gäller kunskapsinhämtning och implementation, var kommunikation GPS/NMEA.



#### 6.4.2 Vad kunde ha gjorts annorlunda

Det egenutvecklade Bluetooth protokollet hade i efterhand kunnat modifieras för att minska och effektivisera datatrafiken mellan enheterna. Istället för att skicka de olika flödena i olika frames kunde de komprimerats till frame med de tre olika ödena, liknande den speedframe som används.

Ytterligare en begränsning i protokollet är antalet bytes som skickas i speedramen, att skicka en byte med m/s är inga problem då max blir 255m/s något en båt inte kommer uppnå. Men då en decimal används innebär de att m/s multipliceras med 10 och då blir begränsningen 25,5 m/s, dvs. 49 knop, något få båtar kommer att uppnå och därför har inte hänsyn till detta tagits.

## Figurer

2.1	Elektrisk signal enligt signalstandarden ISO 11898-2 . . . . .	5
2.2	Kvaser USBcan Professional för övervakning av CAN-buss . . .	6
4.1	Utvecklingskort STM32H103 från Sparkfun . . . . .	10
4.2	Bluetoothmodulen BlueSMiRF . . . . .	11
4.3	Testtrigg för verifiering av flöde och distans . . . . .	19
5.1	Skärmdump på Android application . . . . .	22
B.1	Flödesschema för avbrottsrutiner . . . . .	IV
B.2	Blockschema för systemet . . . . .	IV
C.1	Elschema för labbkort . . . . .	V

## Referenser

- [1] Vector CANtech. (2004). Programming with CAPL.  
[http://vector.com/portal/medien/vector\\_cantech/faq/ProgrammingWithCAPL.pdf](http://vector.com/portal/medien/vector_cantech/faq/ProgrammingWithCAPL.pdf)  
(2014-01-18).
- [2] Di Natale, M. Zeng, H. Giusto, P. Ghosal, A. (2012). Understanding and Using the Controller Area Network Communication Protocol: Theory and Practice.
- [3] Fingas(2013) Android tops 81 percent of smartphone market share in Q3  
<http://www.engadget.com/2013/10/31/strategy-analytics-q3-2013-phone-share/>  
(2014-01-19).
- [4] Marsh, David. (Aug 22 2002). CANbus networks break into mainstream use.  
[www.EDN.com](http://www.EDN.com).
- [5] Bluetooth SIG, Inc (2013) A Look at the Basics of Bluetooth Technology.  
<http://www.bluetooth.com/Pages/Basics.aspx>  
(2014-01-19).
- [6] Cassidy, Frank. (1999), NMEA 2000 Explained - The Latest Word.  
<http://www.nmea.org/Assets/2000-explained-white-paper.pdf>  
(2014-01-19).
- [7] Balch, Mark. (2003) Complete Digital Design. The McGraw-Hill Companies Inc.
- [8] OLIMEX Ltd. (February 2008) Rev.A. STM32H103 development board - User Manual  
<https://www.olimex.com/Products/ARM/ST/STM32-H103/resources/STM32-H103.pdf> (2014-02-11)
- [9] Haversine  
<http://www.longitudestore.com/haversine-formula.html> (2014-02-11)

- [10] Google Inc & Open Handset Alliance.(2013) Android, The World's Most Popular Mobile Platform.  
<http://developer.android.com/index.html>(2014-01-31)
- [11] Schwager, Mark. (2008-04-28). Using MATLAB with CANoe - Vector.  
[http://vector.com/portal/medien/cmc/application\\_notes/AN-IND-1-007\\_Using\\_MATLAB\\_with\\_CANoe.pdf](http://vector.com/portal/medien/cmc/application_notes/AN-IND-1-007_Using_MATLAB_with_CANoe.pdf)(2014-03-08)
- [12] Diffen.com. Bluettoth Vs Wifi.  
[http://www.diffen.com/difference/Bluetooth\\_vs\\_Wifi](http://www.diffen.com/difference/Bluetooth_vs_Wifi)(2014-03-09)
- [13] Kvaser. (2014). Kvaser USBcan Professional.  
[http://kvaser.hosterspace.com/datasheets/kvaser\\_datasheet.php?ean=73-30130-00357-6&page\\_format=USLETTER](http://kvaser.hosterspace.com/datasheets/kvaser_datasheet.php?ean=73-30130-00357-6&page_format=USLETTER)(2014-03-16)
- [14] B.I.O-TECH e.K. (2011). Low Flow Flowmeter.  
[http://www.produktinfo.conrad.com/datenblaetter/150000-174999/150392-da-01-m1-DURCHFLUSSM\\_FCH\\_M\\_POM\\_LC\\_ANSCHL\\_6MM\\_de\\_en.pdf](http://www.produktinfo.conrad.com/datenblaetter/150000-174999/150392-da-01-m1-DURCHFLUSSM_FCH_M_POM_LC_ANSCHL_6MM_de_en.pdf)(2014-03-16)
- [15] Free Software Foundation. (2014). The GNU Manifesto.  
<http://www.gnu.org/gnu/manifesto.en.html#content>(2014-05-11)

## A Bluetooth Protokoll

Frame Uppbyggnad

Start	Ctrl	Data
1 byte	1byte	8 bytes

Tabell A.1: Frame uppbyggnad

Start frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	AA	–	–	–	–	–	–	–	–

Tabell A.2: Start Frame

Config frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	CC	xx	yy	zz	zz	qq	–	–	–

Tabell A.4: Config frame, xx=Flöde, yy=GPS, zz=Givare(pulser/Liter), qq=Samplingsintervall.

Config klar frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	CD	–	–	–	–	–	–	–	–

Tabell A.5: Config klar frame

Flöde(Liter/Nautisk mil) frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	20	–	–	–	–	–	–	xx	xx

Tabell A.6: Flöde(Liter/Nautisk mil)

Flöde(Liter/Timme) frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	21	–	–	–	–	–	–	xx	xx

Tabell A.7: Flöde(Liter/Timme)

Flöde(Liter/km) frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	22	–	–	–	–	–	–	xx	xx

Tabell A.8: Flöde(Liter/km)

Hastghet frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	30	–	–	–	–	–	xx	yy	zz

Tabell A.9: Hastighet, xx=Meter/sekund, yy=Knop, zz=Kilometer/timme.

GPS frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	99	Lat1	Lat2	Lat3	Lat4	Long1	Long2	Long3	Long4

Tabell A.10: GPS

Stop frame

Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	AF	–	–	–	–	–	–	–	–

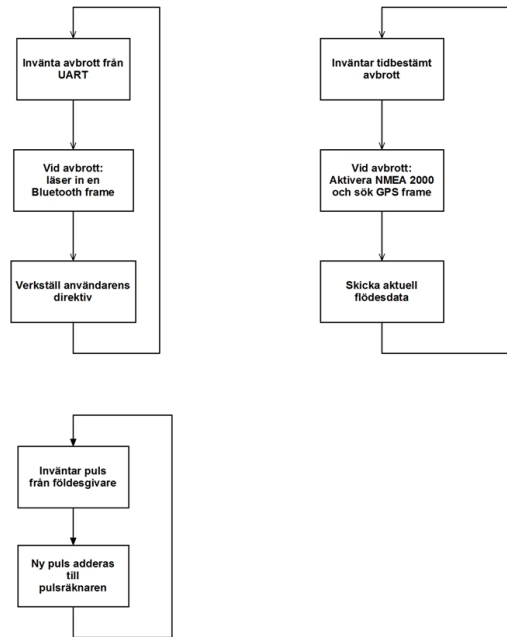
Tabell A.11: Stop

Distans frame

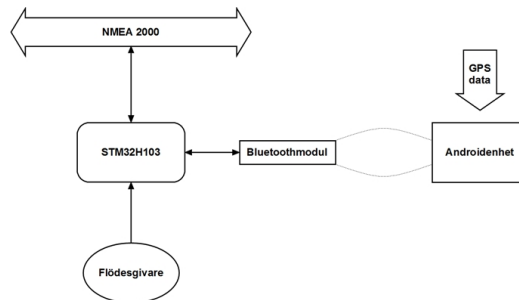
Start	Ctrl	D0	D1	D2	D3	D4	D5	D6	D7
00	BB	–	–	–	–	–	–	xx	xx

Tabell A.12: Distans, xx=meter

## B Flödesschema



Figur B.1: Flödesschema för avbrottsrutiner

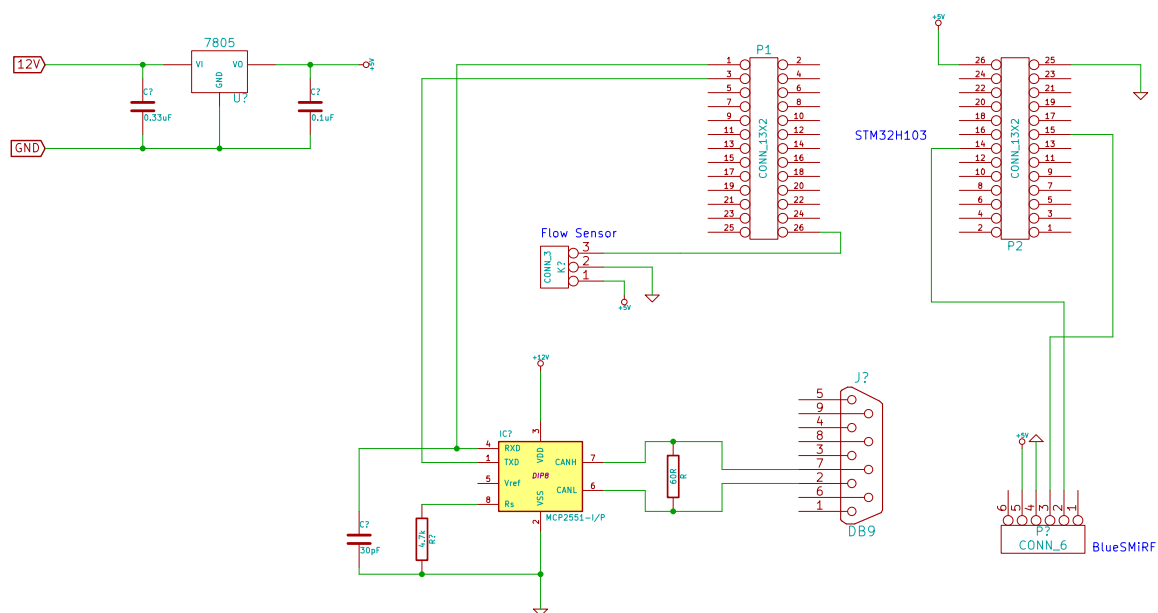


Figur B.2: Blockschema för systemet



## C Elschema

Figur C.1: Elschema för labbkort



File: Elschema-demokort.sch		
Sheet: /		
Title:		
Size: A4	Date: 11 feb 2014	Rev:
KiCad E.D.A.		Id: 1/1

## D Planeringsrapport

Tobias Olsson      MEI  
Daniel Johansson   MEI

# Planeringsrapport

## 1. Inledning

### 1.1 Bakgrund

Företaget är ett konsultföretag som i dagsläget har stora kunder inom fordonsindustrin men vill bredda sin kunskap och få en teknisklösning som kan intressera kunder inom marinteknik. Uppdraget syftar därför till att kunna läsa och ev. skicka data via NMEA 2000, en databuss som är vanlig i dagens båtar.

### 1.2 Syfte

Uppdraget består i att utveckla ett system där data från flödesgivare och GPS analyseras för beräkning av bränsleförbrukning i marina farkoster. De analyserade dataflödet ska sedan till viss del bevaras i systemet samt relevant information ska redovisas med hjälp av ett Androidgränssnitt som ansluts trådlöst. Modulen för beräkningar skall vara konfigurerbar och kunna fungera utifrån olika förutsättningar när de gäller givare och datatillgångar.

### 1.3 Avgränsningar

Den microcontroller som kommer att användas ska vara utav liknande modell som företaget redan använt i tidigare projekt. Gränssnitt för konfigurering och presentation av data begränsas till Androids operativsystem. Beräkning av bränsleförbrukning avser endast en flödesgivare och tar inte hänsyn till antal motorer. Kommunikationen och signaler begränsas till NMEA 2000, ev. seriekommunikation och pulståg.

## 2. Metod

Detta projektet avses delas upp i två delar, en del innefattar en microcontroller som ska behandla data från en flödesgivare och GPS för att utifrån detta beräkna en bränsleförbrukning. Den andra delen avser en Andoridenhet som ska kommunicera med microcontrollern för att enkelt kunna presentera beräknad data och även kunna konfigurera microcontrollern efter stående förutsättningar så som ansluten flödesgivare och GPS möjligheter. Modulen för beräkningar ska bygga på en ARM-processor ur Cortex-M3 serien, exakt vilken processor som antas till projektet ska utredas men den ska ha stöd för CAN-/NMEA 2000 kommunikation och möjliggöra trådlös kommunikation. Processorn ska ha tillgång till flashminne i lagom storlek för lagring av data.

Daniel Johansson MEI

Detta system avses användas i marina farkoster där farkostens GPS användas för att avgöra farkostens hastighet och förflyttad sträcka. Data från GPS ska utläsas via NMEA 2000 som är den databuss som i dagsläget är standard för kommunikation mellan enheter i båtar. Om ytterligare arbete görs kan modulen med microcontrollern läsa av ytterligare information från NMEA 2000 för att t.ex. beräkna mest ekonomiska varvtalet och ev. presentera nuvarande tanknivå om sådan utrustning finns installerad.

### 3. Tidplan

[illegible]