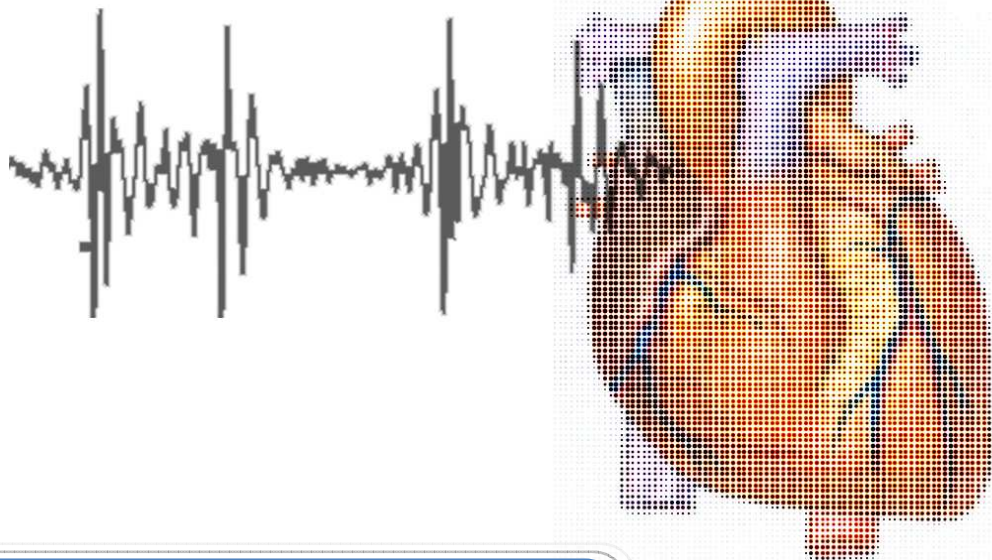


Projektrapport

Grafisk visning af hjertelyde

ITWEM1 – Ingeniørhøjskolen i Aarhus



Bartosz Trzebuchowski 04552

Dennis Francek 05731

04-06-2009

Indholdsfortegnelse

Introduktion.....	3
Formål med dette dokument	3
Baggrund for projektet	3
Krav til projektet	3
Arkitekturdesign	5
Hardwaredesign	8
Softwaredesign.....	10
SPI	10
Grafik & programfunktionalitet.....	11
Navigering mellem vinduer	12
Resultater	13
Konklusion	13
Referencer	14

Introduktion

Formål med dette dokument

Dette dokument er en rapport for et eksamensprojekt gennemført på Ingeniørhøjskolen i Århus i faget ITWEM1. Formålet med dokumentet er at beskrive arbejds- og udviklingsforløbet i projektforløbet. Det omfatter alle aspekter af arbejdet herunder: introduktion til emnet, kravspecifikation, arkitekturdesign, hardwaredesign, softwaredesign, resultater samt konklusion.

Dokumentet er struktureret i afsnit, som følger, og er navngivet efter de forskellige aspekter af arbejdet. Ord, udtryk og koncepter som kræver en uddybelse, er markeret med et notetal og forklaringen vil kunne findes i bunden af samme side som notetallet optræder på. Referencer til uddybende information markeres som en notebogstav i [] parenteser og referencer til disse findes sidst i dokumentet under Referencer.

Baggrund for projektet

Idéen til dette eksamensprojekt udspringer af et afgangprojekt gennemført på IHA. Projektet handlede om bl.a. at udvikle et digitalt stetoskop, iform af hardware og et program til en PDA og på den måde få udvidet funktionalitet, som ellers ikke er muligt med de gammeldags stetoskoper. Man skulle f.eks. kunne se, optage og afspille hjertelyde i form af .wav filer, samt detektere evt. hjertefejl. Dette er udført i programmet LabVIEW. I dette projekt gøres idéen med at plotte data på en graf efter, ved at anvende et Tahoe kort i stedet for en PDA og på den måde udvikle et program i .NET Micro Frameworket.

Krav til projektet

Der skal udvikles et system bestående af både software og hardware til aflytning af hjertelyde. Systemet skal kunne opfange lyde fra hjertet ved hjælp af et accelerometer (figur 1) og fremvise dem grafisk på Tahoe II kortets LCD display.

For at softwaren kan udvikles på Tahoe kortet, skal det udvikles i .NET Micro Frameworket i programmeringssproget C#.

Før hjertelydene kan vises på Tahoe kortets display, skal de konverteres fra analoge til digitale signaler. Der skal derfor udvikles et hardwarekredsløb med en A/D konverter, som vil kunne foretage denne konvertering.

Projektet er et undersøgelsesprojekt med det formål at vurdere teknologiens fremtidige udsigter med henblik på måling af lyde og mislyde i hjertet. Systemet skal ikke betragtes som en prototype på et færdigt produkt men snarere et springbræt til videreudvikling, hvis resultaterne ellers viser grundlag for det. Der bliver i projektet ikke fokuseret på systemets fysiske udformning, brugervenlighed, samt drifts- og brugssikkerhed. Formålet med udviklingen er alene et Proof of concept¹.

¹ En hurtig eller ukomplet realisering af en metode eller et koncept, udført for at bevise teorien eller princippet bag.

Omfanget af udviklingen

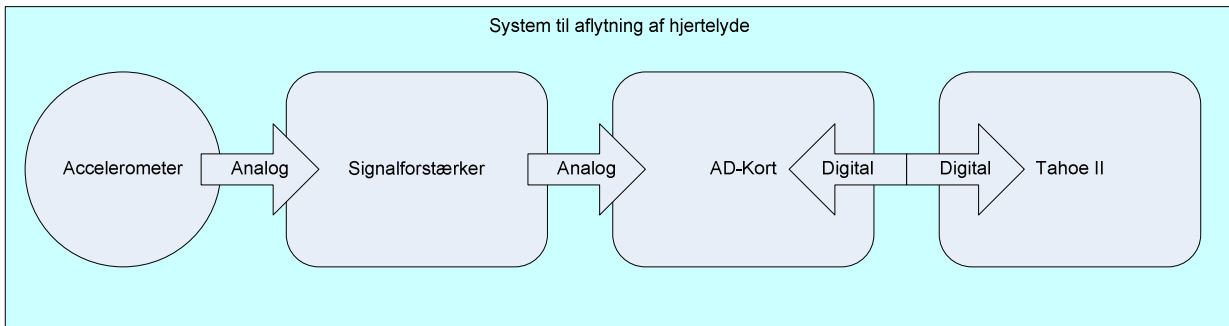
Udviklingsarbejdet kan betragtes som bestående af tre dele:

- Hardware - udvikling af et kredsløb til konvertering af analoge signaler til digitale. Printet er af eget design- og produktion og udføres inhouse på skolens laboratorier.
- Software - udvikling af et program til modtagelse af digitale signaler over SPI² og visning af disse på et farvedisplay.
- Systemsammensætning – da systemet består af flere selvstændige enheder, skal der bruges lidt tid til opsætning af enhederne.

² SPI – Serial Peripheral Interface Bus

Arkitekturdesign

Systemet består af fire komponenter der tilsammen danner det man under en betegnelse kan kalde: apparat til måling og visning af hjertelyde. Systemets opbygning kan ses på figur 1.



Figur 1 - Oversigt over systemets dele

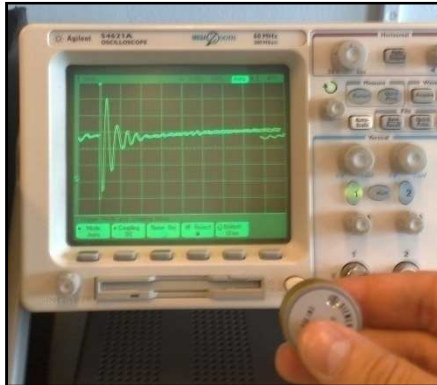
Fra venstre et accelerometer som opfanger vibrationer fra hjertet i form af små spændinger. Disse bliver dernæst forstærket. Derfra bliver det forstærkede analoge signal konverteret til et digitalt signal af en A/D konverter og sendt via SPI til Tahoe kortet. Der bemærkes at kommunikationen mellem A/D konverteren og Tahoe kortet er tovejs. Det er Tahoe kortet som forsyner A/D konverteren med clockimpulser som styrer hastigheden for hvor hurtigt dataene modtages.

Accelerometret kommer fra svenske Siemens-Elema AB. Det opfanger hjertets lyde og via en ledning sender dem som analoge signaler videre til en signalforstærker. Accelerometret påmonteres en persons brystkasse umiddelbart under de store brystmuskler på hjertets side af kroppen. Det er vigtigt, at man ved brug, holder hånden helt stille da accelerometret er meget følsomt overfor støj.



Figur 2 - Det anvendte accelerometer

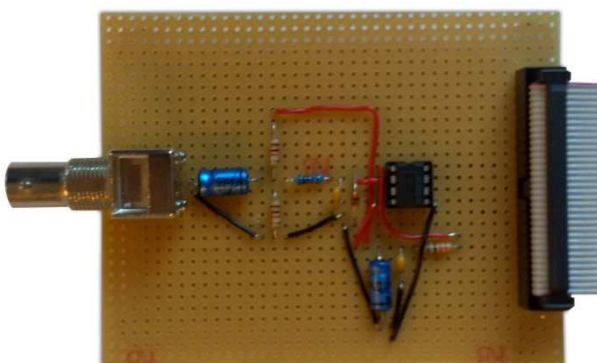
Signalforstærkeren fra danske Bruël & Kjør model 2635^[1] sørger for at forstærke signalet fra accelerometret tilstrækkeligt meget til at A/D konverteren kan sample det i så høj kvalitet som muligt. Man har fundet det optimale udsvingsområde ved at forsøge sig frem. Ved at påvirke accelerometret og måle dets udsving ved hjælp af et oscilloskop, kunne man fastslå den optimale indstilling for forstærkeren (figur 3).



Figur 3 - Udsving måles med et oscilloskop mens man påvirker accelerometeret med en finger.

Med det optimale svingningsområde menes der, at ved kraftig påvirkning, svinger signalet omkring 0 og peak³ op til $\pm 10V$ afhængigt af forstærkerens indstilling.

A/D konverterkredsløbet modtager signalet fra signalforstærkeren gennem dets BNC stik. Dets centrale komponent er en 12bit Maxim Max1241 A/D konverter^[1]. Desuden er kredsløbet forsynet med en overføringskondensator og spændingsdeler. Det signal som kommer fra forstærkeren svinger symmetrisk omkring 0, typisk med $\pm 10V$. A/D konverterens virkeområde ligger mellem 0 og 3V. Det betyder, at signalet skal shiftes, dvs. flyttes op, hvis A/D konverteren skal kunne arbejde med det. Hvordan det løses er beskrevet Hardwaredesign afsnittet. Man har valgt Max1241 chippen fordi den er simpel i sin udformning og leverer 12bit sampling, hvilket er nok til at dække behovet i dette projekt. Det samlede signal bliver sendt til Tahoe II kortet.

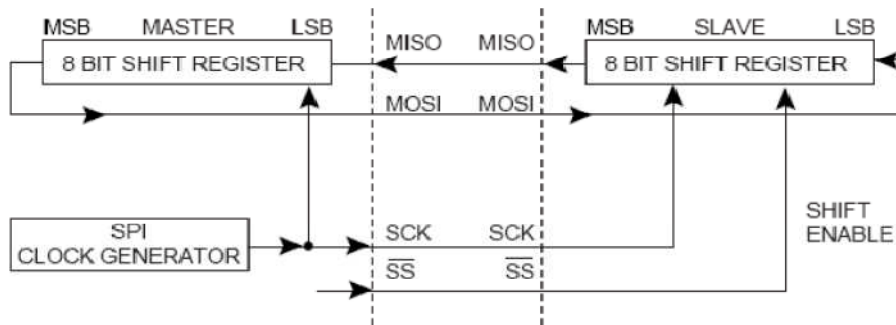


Figur 4 - A/D konverter kredsløbets udformning.

³ Peak – bjergtop som i denne sammenhæng skal forstås som det sted hvor signalet topper.



Tahoe II kortet modtager de digitale samples via dets SPI interface, som er en synkron seriel bus til datakommunikation over korte afstande. Princippet ved SPI bussen ses på figur 5.

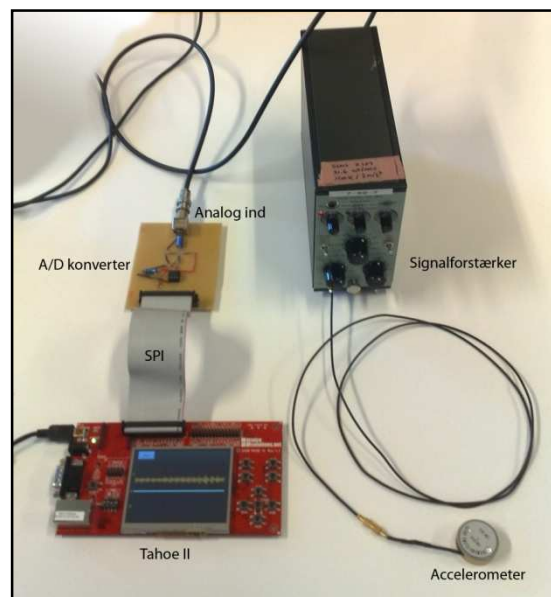


Figur 5 Principper i SPI interfacet.

SPI bussen er bygget omkring en master, (en konfiguration med flere mastere er muligt), og en til flere slaver. Hver master og slave har et 8 bit skifteregister som kan ses på figur 5. For hver clock cyklus flyttes der en bit fra masteren til slaven og den anden vej rundt fra slaven til masteren, efter princippet om first in first out⁴. Bittene udveksles serielt mellem MOSI⁵ og MISO⁶ ben på hhv. master- og slavehardwaren. Når en byte er overført, stopper SPI clock generatoren og der signaleres om at overførslen er slut. I vores tilfælde har A/D converteren kun en MISO og der sendes derfor ikke noget fra master til slave.

Modtaget data behandles og vises grafisk på displayet. Hvordan behandlingen foregår, er beskrevet i afsnittet Softwaredesign.

Tahoe kortet afslutter rækken af de enheder, som tilsammen udgør systemet til aflytning af hjertelyde (figur 6). For mere information om Tahoe kortets funktionalitet henvises til dets datablad.^[iii]



Figur 6 - Systemopstilling

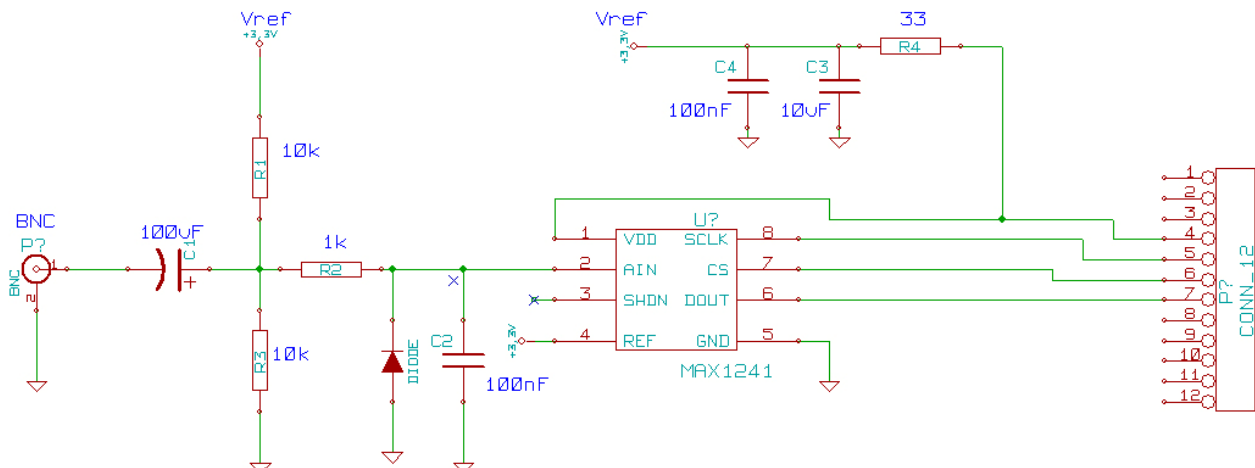
⁴ First in first out princippet – den første bit der kommer ind i køen, er også den der får lov at forlade den først.

⁵ MOSI – Master Out Slave In

⁶ MISO – Master In Slave Out

Hardware design

Til konvertering af analoge signaler fra signalforstærkeren til digitale, som Tahoe kortet kan forstå, blev der designet og produceret et kredsløb med en 12bit Maxim Max1241 A/D konverter. Figur 7 viser hvordan kredsløbet er designet.



Figur 7 Schematics for A/D konverterkredsløbet.

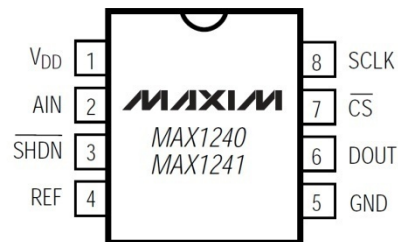
Som det ses på figur 7, kommer signalet fra signalforstærkeren ikke direkte ind i A/D konverteren. Det kommer først igennem en overføringskondensator og to modstande på 10kOhm som fungerer som en spændingsdeler. Deres opgave er at flytte signalet fra forstærkeren, som svinger omkring 0 med $\pm 10V$, op over 0V og til 3.3V. A/D konverterens arbejdsområde ligger i det interval og spændinger under 0V og over 3,6V kan brænde konverteren af. Som yderligere sikring, i tilfælde af at accelerometeret bliver påvirket meget kraftigt, er der monteret en diode, som dæmper alle signaler over 3.3V. En let rystelse ved en stor forstærkning kan ellers nemt brænde A/D konverteren af.

Man ønsker den analoge og digitale strømforsyning adskilt for at undgå forstyrrelser. Desuden er den digitale Vref spænding på 3,3V yderligere sikret stabilitet med to kondensatorer og én modstand som fungerer som et filter. Det som ses i toppen af diagrammet.

Max1241 chippen er udstyret med 6 ben som vist på figur 7:

- **VDD** - driftsspænding på 3,3 V. Denne forsyning kommer fra Tahoe kortet.
- **AIN** - analog input på mellem 0V og Vref, i vores tilfælde 3,3V.
- **SHDN** - når lavt, vil chippen gå i dvæle. Denne funktionalitet er designet med henblik på strømbesparelse, men gøres ikke brug af i denne opgave.
- **REF** - referencespænding Vref på 3,3V som bruges til analog til digital konvertering.
- **GND** - analog og digital ground.

- **DOUT** – Det digitale signal kommer ud i form af 13 bits, 12 bits der beskriver den samlede analoge værdi og 1 start bit. Når DOUT går fra lav til høj, er konverteringen slut og der står 12 bits klar. De flyttes ud serielt i takt med clockslag som kommer ind på SCLK benet. Det tager 13 clockslag for at tømme hele registret for alle 13 bits. Når disse er flyttet ud går DOUT lavt igen indtil de næste 12 bits står klar.
- **CS** (aktiv lav) - iværksætter samplingsprocessen. CS sættes lav for at starte en konvertering.
- **SCLK** - ekstern clock, må ikke være over 2,1MHz. Clocken må ikke være aktiv mens CS er lav og dermed når samplingsprocessen er i gang.



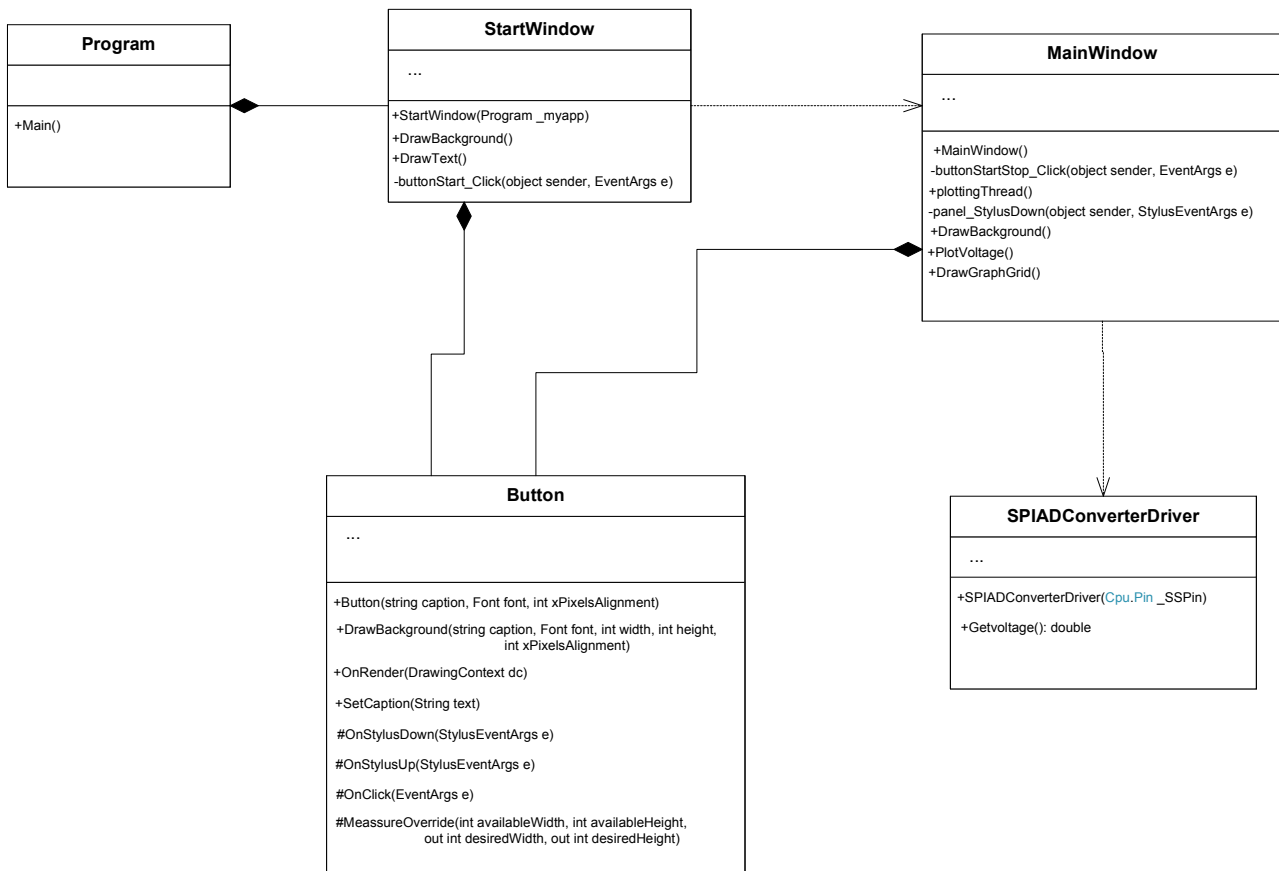
Figur 8 Maxim Max 1240/1241 chip. I dette projekt bruges Max1241 fordi den giver mulighed for egen referencespænding. Max1240 har en fikset REF på 2,5V.

Konverteren virker ved, at der kommer et analog signal ind på chippens AIN indgang. Når CS sættes lavt, vil chippen stoppe for ny input fra AIN, og holde det analoge spændingsniveau konstant, som der var på det tidspunkt hvor CS gik lavt. Derefter vil signalet blive samlet med en opløsning på 12 bits. Det betyder, at den samlede værdi beskrives med et tal på mellem 0 og 4095. Det er her vigtigt at bemærke, hvorfor vi var interesserede i at forstærke signalet sådan, at det peaker lige ved konverterens grænser for måleområdet. Jo mere man kan strække signalet ud over de tilgængelige 4095 målepunkter, jo mere nøjagtigt bliver det digital signal.

Når samplingsprocessen er slut, vil DOUT gå højt som tegn på, at der er bits klar til at blive hentet ud. CS vil sættes højt og den eksterne clock begynder at sende impulser ud til SCLK. For hvert clockslag vil der komme en bit ud af DOUT, MSB først. Efter 13 clockslag vil alle 1 startbit + 12 samplebits være ude. Derefter vil clock stoppe, CS gå høj og hele processen kan gentages.

Softwaredesign

I dette afsnit beskrives det overordnede softwaredesign samt de enkelte dele af softwaren. På nedenstående figur ses et klassediagram for software designet som det endeligt blev realiseret.



Figur 9 - Endeligt klassediagram

Her består softwaren af 5 klasser med hvert deres formål. Program klassen indeholder Main funktionen hvorfra programmet startes op. Fra program klassen oprettes StartWindow, som er det vindue man præsenteres for når programmet afvikles. Fra StartWindow oprettes vores hovedvindue, som er klassen MainWindow. Det er her programmets hovedfunktionalitet finder sted. Begge vinduer anvender Button klassen som implementerer en klasse til at oprette knapper som reagerer på tryk på den trykfølsomme skærm. Fra MainWindow klassen anvendes SPIADConverter klassen som er vores SPI driver.

SPI

Driveren til at kommunikere med vores hardware over SPI bussen, var den første del af softwaren der blev lavet. Dette blev lavet som det første, da det var det mest kritiske at få til at virke korrekt.

Vi havde tidligere anvendt SPI til datakommunikation, men kun i forbindelse med et STK 500 kit i C, hvor der skal skrives til registre og overholdes strenge tidskrav. Så databladet for vores A/D converter blev nærstuderet for at finde ud af hvilke tidskrav der skulle overholdes. Det viste sig dog at være langt nemmere og unødvendigt at overholde tidskravet for konverteringstiden.

For at anvende SPI i .NET MF(Micro Frameworket), opretter vi en SPI configuration, hvor SPI interfacet konfigureres.

```
new SPI.Configuration(SSPIn,
                    false,
                    0, //Not important, pin is floating
                    0, //Not important, pin is floating
                    false, //(Clock Polarity)Idlestate of clock = false (0)
                    false, //(Clock Phase)Data is sampled at the falling edge of the clk
                    2, //2.0 KHz
                    Microsoft.SPOT.Hardware.SPI.SPI_module.SPI1);
```

Ovenstående kodeblok viser opsætningen af SPI interfacet i forhold til databladet for A/D converteren. Her sættes clock hastigheden til 2 KHz, clock polariteten til lav og clock fasen sættes til false, som betyder at data samples på den faldende flanke af clocken. Med denne opsætning er man klar til at kommunikere via SPI interfacet. Ifølge databladet skal Chip Select på A/D converteren sættes lav for at starte en konvertering, hvorefter der maximum går 7.5us til konverteringen er slut. Vi hiver derfor pin GPIO4 lav i softwaren, da det er denne der er forbundet til CS på A/D converteren. Når dette er gjort er vi klar til at læse dataene modtaget over SPI. Til dette bruges en funktion til både at skrive til og læse fra SPI bussen. Skrivningen bruges, da der ofte ved SPI kommunikation skal sendes data til slaven før man modtager noget tilbage. Dette skal der ikke i vores tilfælde, da der ikke er noget input på A/D converteren. Der sendes derfor en tom buffer, da funktionen skal have noget med som parameter. I den anden parameter modtager vi i vores buffer på 16bit, 2 efterfølgende bytes med vores data i. Dataene i vores buffer skal så konverteres før vi har den reelle værdi af den målte spænding. Den første bit(msb) i bufferen er altid 1, så denne *shifter* vi væk. Derefter *shifter* vi de 3 sidste bits væk, da de kun indeholder nuller.

Dette skyldes converterens opløsning på 12 bit.

For at kunne konvertere vores data i form af bits om til en spænding, ganges hver sample med koefficienten – $(3.3 / 4095)$, hvor 3.3 V er vores reference spænding og 4095 er decimalværdien for 12 bit. Når det er gjort har vi vores reelle spænding fra A/D converteren til videre behandling i softwaren.

Grafik & programfunktionalitet

Målet med programmet var at plote dataene fra SPI driveren i en graf. Vi valgte derfor som en start at lave programmet som en console applikation, hvor vi fra programmets opstart står i en uendelig while løkke og løbende tegner dataene oven på et graf gitter ud fra hver sample. Når grafen var fyldt ud skulle grafen cleares og dataene skulle plottes fra starten igen. Da vi havde fået dette til at virke valgte vi at lave en Window applikation i stedet, da det gav bedre udvidelses muligheder.

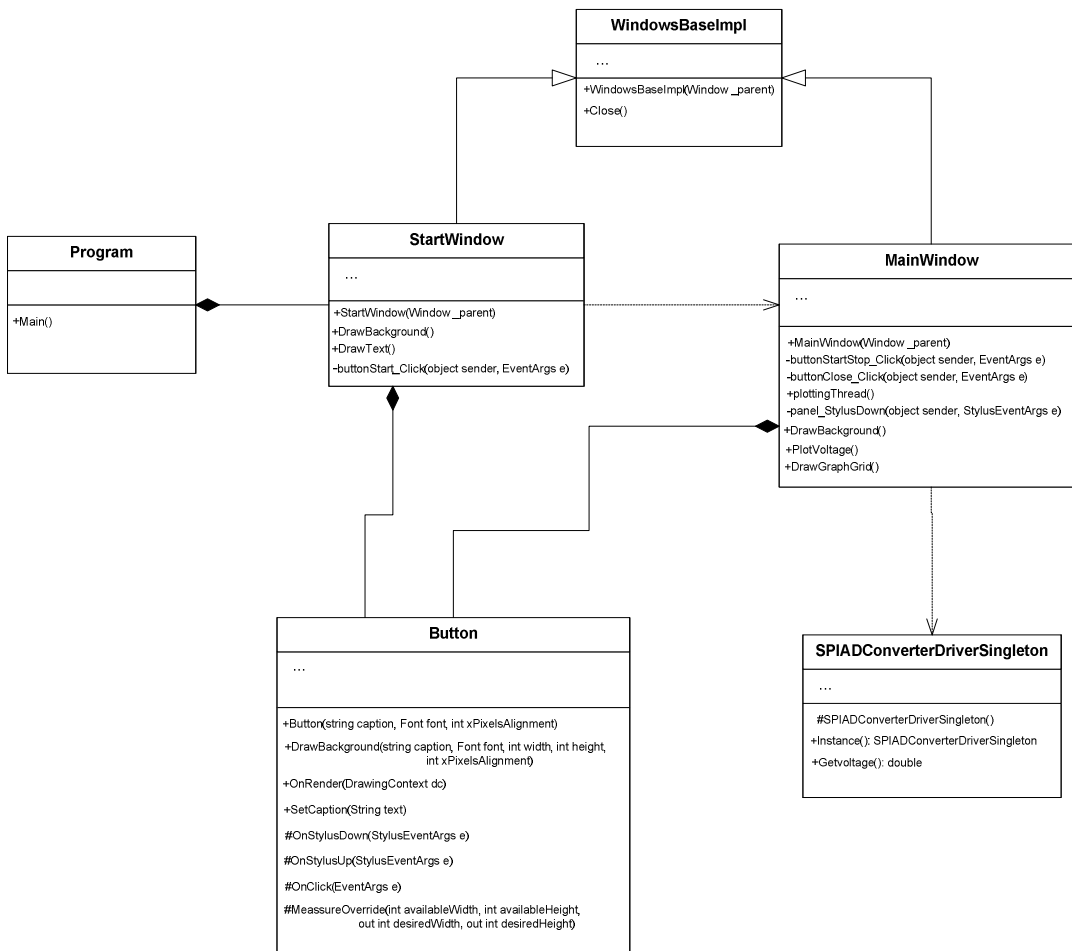
Største delen af applikationen består af det grafiske programmering til LCD skærmen. Vi har delt det grafiske op i 3 klasser. 2 klasser til de to vinduer vi har. Et start vindue, hvor programmet præsenteres og vores hovedvindue, hvor programmets hovedfunktionalitet finder sted. Derudover er der lavet en Button klasse til oprettelse af knapper som reagerer på interaktionen med den Touchscreen der sidder på Tahoe-II boardet.

Start vinduet består af lidt grafik og en knap til at starte hovedvinduet med. Hovedvinduet er bygget op af et panel til at strukturere de grafiske elementer på skærmen. Her oprettes der en knap til at starte og stoppe dataplotningen på grafen. Der udover oprettes der et Image med en bitmap, hvorpå alt graf relateret tegnes. Grafen består af et gitter og en X-akse som indikerer nul værdien for indgangssignalet. Ovenpå grafen, tegnes en linje i mellem det tidligere tegnede punkt og den nye punkt som udregnes ud fra grafens placering og hver sample vi modtager fra SPI interfacet.

Til løbende at kunne plote hver sample på skærmen var vi nød til at oprette en ny tråd sideløbende med hovedtråden. Tråden oprettes og lukkes når der trykkes på start/stop knappen. I tråden ventes der i en uendelig while løkke, indtil en boolean sættes fra hovedtråden. For hvert loop tegnes den aktuelle sample på grafen.

Navigering mellem vinduer

Der blev brugt en del tid på at finde ud af, hvordan der kunne navigeres imellem flere vinduer. Løsningen blev en ekstra klasse som extender den indbyggede Window klasse. På denne måde kunne et parent vindue gemmes og sættes til programmets aktuelle MainWindow, når et vindue lukkes med den nye implementerede .Close() metode.

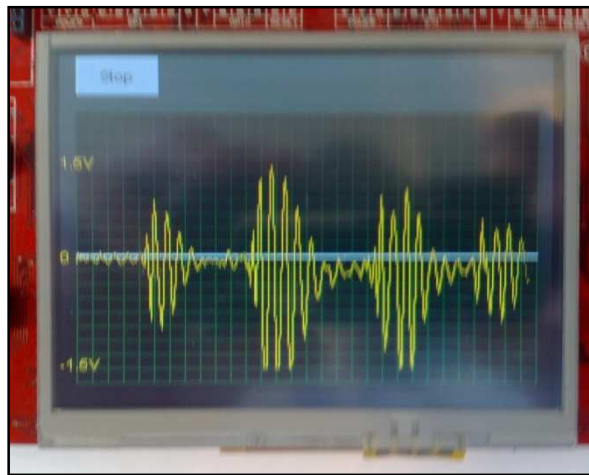


Figur 10 – Klassediagram for det design til navigering mellem vinduer der kun virker i emulatoren

Ovenstående figur viste sig at virke fint i emulatoren til navigering frem og tilbage, men ikke da det endeligt blev testet på Tahoe-II. Her blev der smidt en exception ved oprettelsen af SPI interfacet selvom dette var forsøgt implementeret som singleton designmønstret. Fejlen i dette program er højst sandsynligt en lille ting vi har overset men da det ikke virker på Tahoe-II boardet er løsningen som på figur 9 blevet valgt som den endelige.

Resultater

Der blev i dette projekt udviklet både et fungerende A/D konverterkredsløb samt software til Tahoe kortet. De enkelte delelementer blev forbundet med hinanden og tilsammen dannede de et apparat som er i stand til at registrere lyde og vise dem grafisk på et display (figur 10) på Tahoe kortet. Resultaterne indfrier altså de krav som blev stillet i kravene til projektet. Set ud fra den betragtning må man konkludere, at resultatet af projektarbejdet er tilfredsstillende.



Figur 11 Screenshot af målte lyde vist på Tahoe kortets display.

Projektgruppen opnåede nyt viden omkring udvikling af software i .NET Micro Frameworket til indlejrede systemer og man har fået en bedre forståelse for virkemåden af elektroniske komponenter gennem udviklingen af A/D konverteren.

Man har stiftet bekendtskab med Tahoe II kortet, som er et eksempel på et nyt og stærkt voksende marked for små indlejrede enheder med egne trykfølsomme skærme, navigationsknapper, accelerometre og et hav af udvidelses- og anvendelsesmuligheder. Af samme grund vil deres udbredelse vokse og sammen med den, behovet for udvikling af software til dem.

Konklusion

Ser man mere generelt på brugbarheden af løsningen i forhold faktiske målinger af hjertelyde, er resultaterne lidt mere tvivlsomme. Selvom lydene bliver registreret og vist på displayet, og dét med hurtig respons og god dynamik, så er systemet behæftet med flere svagheder. Den største af dem er at lydene bliver opfanget lige kraftigt. Det betyder, at systemet er meget følsomt overfor støj, som meget nemt opstår blot ved fysisk kontakt med accelerometeret eller støj fra omgivelserne. Løsningen ville nok være dels at bruge et accelerometer indbygget i et lydafsærmet hus, og dels udvikle intelligente filtre som vil kunne filtrere uønskede frekvenser bort og identificere hjertelyde fra andre støjkilder. For at kunne bruge systemet til noget fornuftigt, er der selvfølgelig behov for meget mere udvikling af avanceret software som skal udvikles i samarbejde med eksperter fra medico faggrupper.

Gennem arbejdet i dette projekt har man taget det første skridt til en prototype på et digitalt stetoskop. En stadig udvikling af bedre og mindre indlejrede computere, samt projekter som OpenCare^[iv], gør det ikke usandsynligt, at idéen en dag vil finde vej til kommerciel anvendelse.

Generelt betragter gruppen udbyttet af både WEM1 kurset så vel som det efterfølgende projektarbejde, som meget inspirerende og fagligt berigende.

Referencer

- ⁱ CD rom > datablade > BRUEL&KJAER_2635.pdf
- ⁱⁱ CD rom > datablade > MAX1240-MAX1241.pdf
- ⁱⁱⁱ CD rom > datablade > Tahoe-IITechnicaReferenceManual.pdf
- ^{iv} <http://opencareproject.wikispaces.com/>