CHALMERS

Träningsgadget

Taktstyrd mediaspelare

JONAS EKSTRÖM

MIKAEL ANDERSSON YNGHAMMAR

Examensarbete

Högskoleingenjörsprogrammet för datateknik

CHALMERS TEKNISKA HÖGSKOLA
Institutionen för data- och informationsteknik
Göteborg 2012

Sammanfattning

Att hålla den takt man vill när man springer till musik kan vara svårt, därför eftersträvar vi att skapa ett hjälpmedel som kan hjälpa motionärer att hålla den takt de vill. Vi eftersträvar även att detta hjälpmedel skall varna användaren om något närmar sig bakom dess rygg, som till exempel en bil eller cykel. Vi vill med hjälp av sensorer och en mikrokontroller bestämma takten vilken användaren rör sig med, med utgång från den aktuella takten jämfört med en vald referenstakt, för att sedan alternera musikens uppspelning beroende på resultatet av jämförelsen. En annan typ av sensor skulle sedan behövas för att upptäcka om ett objekt dyker upp bakom användaren för att därefter genom ljud varna användaren om faran. Den här rapporten beskriver en lösning som har tagits fram i två delar, ett tillbehör till befintliga portabla mediaspelare och en Android-applikation. Den beskriver även de problem som vi stött på under arbetets gång och hur vi gjorde för att lösa dessa. Lösningen har tagits fram genom grundliga tester av sensorer och ytterligare nödvändig hårdvara tillsammans med mjukvara. Någon lösning för en sensor som känner av om något kommer bakom användaren har inte tagits fram, detta på grund av att vi fick prioritera om vissa delar då det hände saker som gjorde att vi ej kunde följa vår tilltänkta tidsplan.

Nyckelord: accelerometer, takt, Arduino, Android, träningshjälpmedel

Abstract

Keeping the pace you want when you run to the music can be quite difficult, therefore, we strive to create a tool that can help athletes to keep the pace they want. We also seek for this tool to warn the user of something approaching behind their back, like a car or bicycle. We would like, by using sensors and a microcontroller, to determine the rate which the user is moving at. Starting with the current pace compared to a chosen reference rate, and then alternates the music playback depending on the results of the comparison. Another type of sensor would be needed to detect if an object appearing behind the user and then alert the user through audio to warn about the potential danger. This report describes a solution that has been developed in two parts, an accessory to existing portable media players and an Android application. It also describes the problems we encountered during the work on this project and what we did to solve them. The solution has been developed through thorough testing of sensors and additional required hardware together with software. A solution for a sensor to alert if something is approaching behind the user has not been developed, due to obstacles that we encountered and had to solve first. Because we had to give priority to other things that was not planned for in our time schedule we had to choose to skip something and the choice fell on the back sensor.

Keywords: accelerometer, pace, Arduino, Android, exercise aid

Innehållsförteckning

1. INLEDNING

1.1 Bakgrund

Vi har viljan att utveckla ett eget träningshjälpmedel som skall kunna användas när man till exempel springer eller cyklar. Idén kommer ifrån att vi själva tränar frekvent, Mikael är långdistanslöpare och Jonas cyklar längs med landsvägar, och när vi gör detta så brukar vi lyssna på musik i en mp3-spelare. Utöver detta har vi noterat att en stor del av de vi möter ute och tränar även de lyssnar på musik. Vid undersökningar bland våra bekanta stärks dessa observationer då merparten säger sig föredra att träna till musik.
Vi började se möjligheter för att kunna utföra denna idé när vi under den tidigare projektkursen DAT065 skapade en handske som styrde ett datorspel genom att man rör på handen.

1.2 Problemställning

Vi har upptäckt att det kan vara svårt att hålla en jämn takt vid träning när man lyssnar på musik med varierande tempo, skillnaden i takt medför ofta att ens egen takt på till exempel löpstegen blir varierande. Vårt mål är att utveckla ett hjälpmedel som på något sätt gör användaren uppmärksam på att den avviker från vald takt. Vi har som mål att ta fram dels en lösning som kopplas till en befintlig portabel mediaspelare, dels en applikation till Androidtelefoner och möjligtvis en komplett hårdvarulösning med inbyggd mediaspelare.
När väl olika lösningar tagits fram skall jämförelser utföras där fördelar respektive nackdelar ställs emot varandra. Jämförelser skall innefatta dels prestanda och funktionalitet och dels användarupplevelse och möjlighet till försäljning som en eventuell kommersiell produkt.

Ett annat problem som vi söker lösning på är vid löpning eller cykling på landsvägar så uppkommer svårigheter att uppfatta när det närmar sig en bil bakifrån, speciellt om man samtidigt lyssnar på musik, därför strävar vi efter att finna en lösning att uppmärksamma användaren att det kommer något bakifrån så att denne får en chans att reagera.

1.3 Syfte

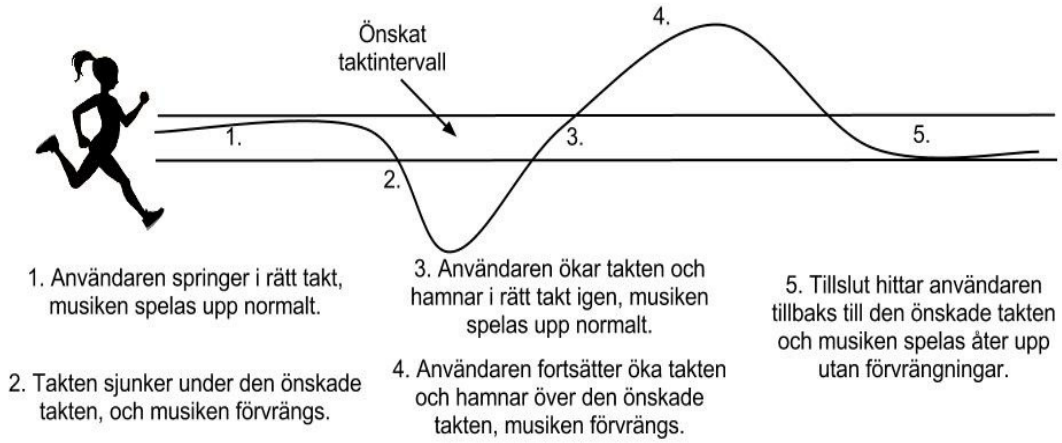
Vi strävar efter att undersöka möjligheten att ta fram ett helt nytt träningshjälpmedel som ännu inte finns på den kommersiella marknaden, vilket kan ge möjligheter till ekonomiska vinster. Idén är att med hjälp av musik hjälpa en person hålla vald takt vid till exempel löpning. Vi vill även undersöka om det går att kombinera detta hjälpmedel med en sensor för att varna för bilar som kommer bakifrån, för att minska risken för att bli påkörd för man inte varit uppmärksam på trafiken.

1.4 Avgränsningar

Vi kommer inte ta hänsyn till vilken takt musiken har, utan kommer enbart gå efter på den fysiska takten vid rörelse.
Vi kommer inte använda oss av andra mikrokontrollers än ATMEGA328 med Arduino bootloader(se teknisk bakgrund), anledningen är att projektet skulle bli för stort för att kunna genomföras inom rimlig tid.
Vi strävar inte efter att göra fullständigt kompletta produkter för varje plattform, utan målet ligger i att testa ett koncept på flera möjliga plattformar och sedan jämföra möjligheter och eventuella hinder hos dessa.

2. METOD

När vi arbetade med vårt projekt i föregående läsperiod så använde vi bland annat en Arduino LilyPad, en mikrokontroller som är utvecklad för att sys fast i tyg. Detta fick oss under projektets gång att fundera; "Vad mer kan vi göra med den här tekniken?". Vi kom då fram till en gemensam idé att vi skulle kunna utveckla ett träningshjälpmedel som skulle kunna känna av vilken takt som man springer/cyklar i och som sedan skickar signaler till mp3-spelaren så att musiken till exempel saktas ned om man ligger under den önskade takten, eller spelas för fort om man ligger över. (Se figur 2.1)

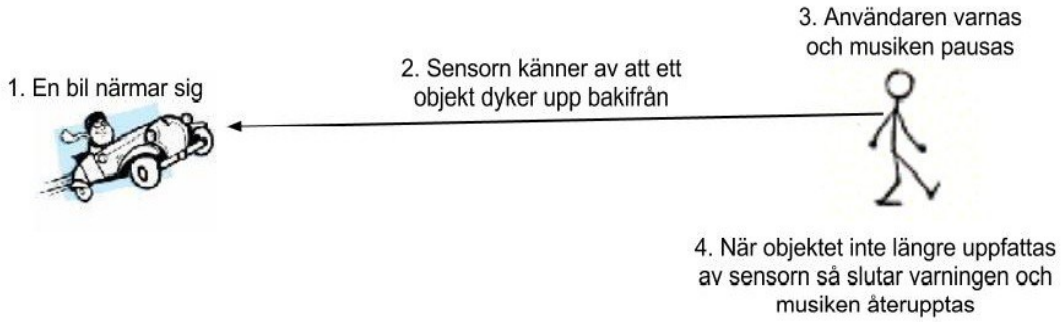


Figur 2.1 Principskiss för taktfunktion

Detta problem har vi tänkt lösa med hjälp av en multiplexer (se teknisk bakgrund) för att byta mellan mp3-spelare och egna ljud från LilyPaden, eller bygga en equalizer/ljudprocessor för att förvränga musiken från själva mediaspelaren.
Eftersom de flesta nya telefonerna idag har en inbyggd accelerometer och musikspelare så har vi tänkt utveckla en applikation som skulle kunna göra samma sak som vår hårdvara, det vill säga skicka signaler till musikspelaren om man ligger över eller under takten. På detta sätt så skulle personer som föredrar att ha med sig sin telefon när de tränar inte behöva ha på sig något extra. Kanske skulle även man kunna spara GPS positioner så att man skulle kunna se vilken väg man sprungit eller cyklat och kanske även spara någon annan sorts statistik, till exempel hur många gånger man var i otakt och liknande.
Programmet för hårdvaran kommer att kodas i Arduino Programming Language (se teknisk bakgrund) och appen kommer att programeras i Java (se teknisk bakgrund).

För att lösa problemet med att man inte märker om det kommer en bil bakifrån så tänkte vi att man skulle kunna ha en sensor på till exempel hjälmen om man cyklar eller på en mössa, pannband eller kanske på en tröja om man springer. Vi ska sedan försöka utveckla ett sätt att skärma av sensorn så att den inte varnar för att det till exempel är träd bakom

längs med vägen. När väl något objekt kommer upp bakom sensorn så ska användaren varnas, antingen via en vibrator eller att någon slags varningssignal skickas till musikspelaren och pausar musiken. (Se figur 2.2) I det sistnämnda fallet så kommer Bluetooth kommunikation att behövas.



Figur 2.2 Principskiss för ryggsensor.

Vi tror att en sådan sensor skulle vara något som inte bara skulle vara användbart för personer som tränar utan även för människor i allmänhet som befinner sig på vägar eller ute i trafiken, vi kan tänka oss att det skulle hjälpa att förhindra flera olyckor där cyklister är inblandade.

Det tidiga målet att utveckla en komplett kommersiell produkt har ändrats till att utveckla och testa konceptet på flera plattformar, med avsikt att göra en komparativ studie där plattformarnas för- och nackdelar ställs mot varandra. De principiella funktionernas värde och dess skillnader väger tyngre än att göra en perfekt mediaspelare med många funktioner.

3. TEKNISK BAKGRUND

3.1 Begreppsförklaringar

Arduino är en plattform med verktyg för att få datorer att kommunicera med omvärlden via en rad olika sensorer. Projektet bygger på open-source för både hård och mjukvara, vilket ger möjligheter för att bygga egna versioner och även omarbeta befintliga produkter för att passa användarens specifika behov. Tillverkarens beskrivning av sitt koncept lyder

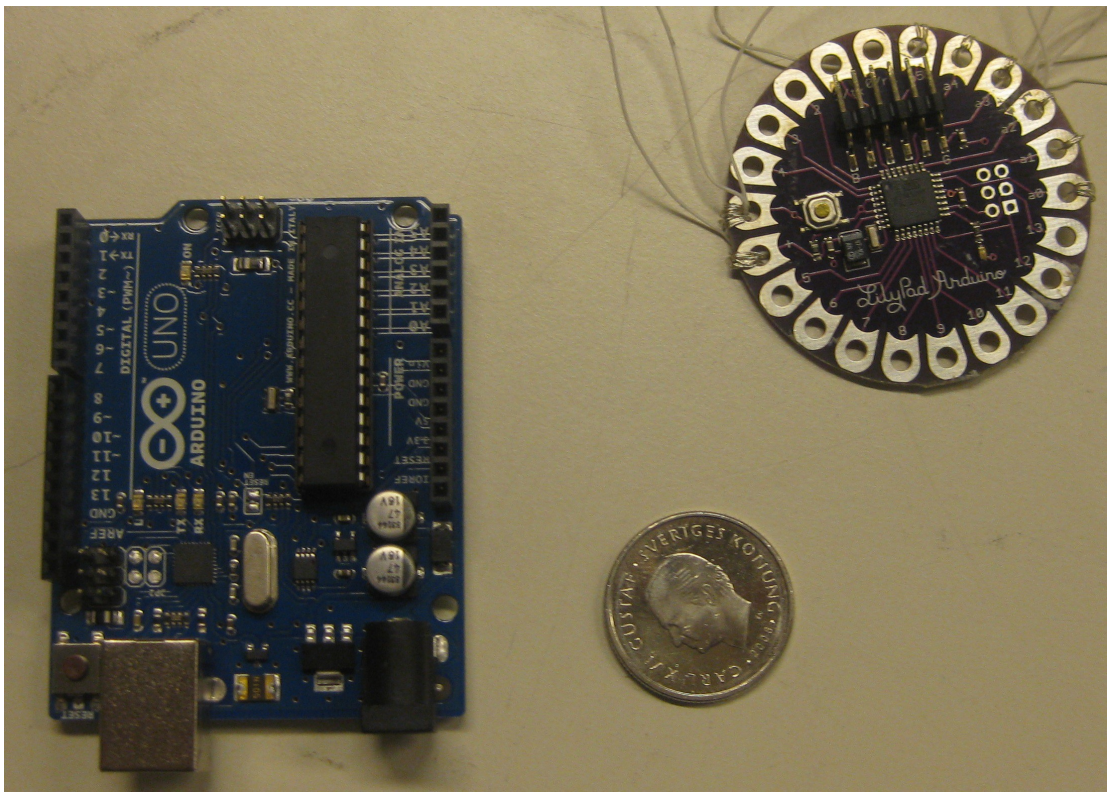
"Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board." [1]

Varför vi valde att arbeta med just Arduino var att vi hade mycket positiva erfarenheter av plattformen sedan ett tidigare projekt där vi utnyttjade handens rörelser för att styra ett datorspel. I de projektet användes LilyPad huvudenhet, tillsammans med LilyPad accelerometer och en Bluetoothmodul för att sköta kommunikationen.

Till alla olika versioner av Arduinos finns en uppsjö av tillbehör för olika ändamål, de handlar dels om sensorer för att känna av omvärlden och dels möjligheten att styra lampor, motorer, spela upp ljud eller för att kommunicera med andra enheter vilka kan vara såväl andra Arduinos som datorer eller telefoner.

I detta projekt används främst två olika Arduinomodeller, främst minstingen i familjen kallad LilyPad[2]. LilyPaden skiljer sig från övriga genom sin lilla och platta form och möjligheten att sy fast på tyg med hjälp av konduktiv tråd (en speciell typ av tråd som leder ström). Dess karaktär ger utmärkta möjligheter för att skapa klädesplagg med roliga funktioner såsom lampor som blinkar beroende hur personen som bär tröjan rör sig, exempelvis en tröja med blinkers som aktiveras när bäraren lyfter upp ena armen.

Den andra modellen vi använt oss av är Arduino Uno, vilken är Arduinos referensmodell[3]. Dess storlek (se figur 3.1) gör den lite svår att bära med sig, i detta projekt är de trots allt inget hinder då enbart funktionaliteten undersöks, i en komplett produkt skulle storleken kunna minskas ner genom att ta bort överflödiga komponenter.



Figur 3.1 Storleksjämförelse mellan Uno och LilyPad

Huvuddelen i bägge modellerna är mikrokontrollern ATMEGA328 från Atmel[4]. En 8-bitars mikroprocessor utrustad med 16KB minne i LilyPad och 32KB i Uno. Mikrokontrollern på de bägge modellerna har låg strömförbrukning, den kräver endast 0.2mA vid drift. Spänningsmatning till de bägge enheterna möjliggörs enklast genom USB kopplat direkt från en dator, om så är möjligt. För extern strömförsörjning skiljer sig modellerna drastiskt, Uno kan drivas av spänning mellan 6 och 20Volt, dock rekommenderas mellan 7 och 12Volt för att fungera medans LilyPad kan drivas av en spänning från 1.8Volt upp till 5.5Volt, vilket är samma gränser som mikrokontrollern klarar. Den mer påkostade Unon är utrustad spänningsregulator för att kunna leverera stabila utspänningar på 3.3Volt respektive 5Volt för drivning av yttre komponenter, LilyPaden saknar alla former av skydd eller regulatorer för felaktiga spänningar och behöver därför hanteras med varsamhet.

Arduinos produkter kommer laddade med företagets egen bootloader(se nedan) och programmeras enkelt genom Arduinos egenutvecklade utvecklingsmiljö. Programmeringsspråket som används kallas för Arduino Programming Language är en implementation av programmeringsramverket Wiring[5] vilket är baserat på projektet Processing[6], som syntaxmässigt påminner väldigt mycket om vanlig C/C++ programmering. Till skillnad från vanlig mikroprocessorprogrammering krävs väldigt lite kunskaper om initiering av hårdvaran, de som behöver göras är att i programmet specificera en port på kortet för att vara in eller utgång, sedan går de väldigt enkelt att bara läsa eller skriva från porten.

Bootloader är namnet på den första programvaran som laddas in när en dator startas. Dess uppgift är att möjliggöra för inläsning av den programvara användaren vill köra, exempelvis ladda in operativsystem vid uppstart av en dator. I inbyggda system används bootloader på samma sätt, när mikroprocessorn startas upp läser den först i sin bootloader för att antingen via en datalänk ladda in ny programvara eller för att köra ett program som redan ligger i dess flashminne. Bootloader är en programvara som ofta ligger inbränd i ett ROM minne, alltså ett minne som bara går att läsa ifrån, då den väldigt sällan (i de flesta fallen aldrig) behöver bytas ut.[7]

Accelerometer[8] är en komponent som möjliggör mätning av rörelse med hjälp av att mäta g-krafter. Användningsområdena för accelerometrar är många, enkelt sätt kan de delas in i två huvudområden när de gäller användning. Det första huvudområdet för användning är att mäta olika typer av rörelse, nedan följer en lista på exempelapplikationer:

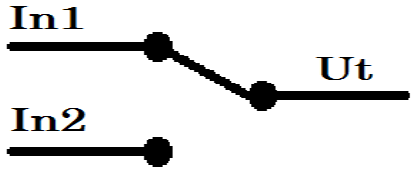
- Mäta vibrationer i byggnadskonstruktioner.
- Mäta seismisk aktivitet för att övervaka vulkaner.
- De kan fungera som krocksensorer i bilar genom att mäta kraftig negativ acceleration för att bestämma krockvåldet för att ge signaler för att lösa ut till exempelvis air-bags.
- Användas i digitalkameror som skaksensor för att hjälpa bildstabilisatorer, för att ta bild då kameran är helt stilla.
- Användas i bärbara datorer som sensor ifall datorn tappas för att ge bland annat låsa fast hårddiskars läsarm för att minska risken för att data går förlorad.

Det andra huvudområdet är att mäta lutning, exempel på detta är:

- Känna av orienteringen på smarta telefoner för att kunna rotera innehållet på skärmen så att användaren får texten åt rätt håll.
- Känna av lutningen på objekt i rörelse, exempelvis raketer.

Är i vårt fall den centrala delen för att känna av rörelse, för att ge oss möjligheten att representera en översättning till takt.
Det finns många olika typer av accelerometrar, den som testats i hårdvarudelen av denna rapport är en treaxlig accelerometer vid namn ADXL335 från Analog Devices[9], vilken ger utslag i X,Y och Z led likt ett ortogonalt koordinatsystem.

Multiplexer[10] elektronisk komponent med flera ingångar vilka exklusivt kan väljas separat att gå genom kretsen till en utgång. Kan ses som en elektroniskt styrd växel(se figur 3.2).



Figur 3.2 Principskiss multiplexer.

Kondensator[11] elektrisk komponent används typiskt för att blockera likspänning eller för att jämna ut växelspänningars styrka.

Resistor[12] elektriskt motstånd för att begränsa ström och spänning i en elektrisk krets.

Förstärkare[13] komponent som används för att öka styrkan på en elektrisk signal, utan att ändra signalens innehåll.

Distorsion[14] förvrängning av originalsignalen, i ljudsammanhang uppfattas detta som störningar, missljud eller bara ljud som blir en aning färgat.

Bluetooth[15] teknik för trådlös överföring mellan två eller flera komponenter. Används exempelvis för trådlösa handsfree-lösningar till mobiltelefoner.

Java[16] programmeringsspråk ursprungligen utvecklat av amerikanska Sun Microsystems, stor utbredning för mobila enheter. Togs fram för att vara portabelt, med andra ord kunna användas på flera plattformar(exempelvis fungera under Linux/Windows/Mac) utan att modifiera källkoden.

Android [17][18] open source plattform för mobila enheter. Stora möjligheter att skapa egna applikationer, så kallade "appar", via Java. Utvecklades av Android inc. och ägs numera av Google.

GUI [19](Graphical User Interface) kallas de grafiska gränssnitt som har som uppgift att interagera mellan dator och användare. Bilder och text används för att visa användaren vad som sker i systemet. Kan även användas av användaren för att styra systemet via inmatningsfunktioner, såsom pekskärm eller mus/tangentbord.

GPS [20](Global Positioning System) system för positionering med hjälp av satelliter utvecklat av amerikanska militären. Relevant användning av GPS systemet i detta projekt är enbart för mätning av hastighet.

Mp3 [21] (MPEG Audio Layer III) teknik för komprimering av ljud. Den vanligaste komprimeringsformen för portabla musikspelare (förutom Apples produkter som använder sig av AAC). Musik som lagras på CD-skivor kräver mycket lagringsutrymme, ungefär 10MB per minut, musik i Mp3-format kräver endast en tiondel (beroende på kvalitetsinställningar) av detta utrymme på grund av att den komprimerats.

Open Source[22][23] är programvara som fritt kan laddas ner och användas som den är eller modifieras för att passa andra projekt. Namnet som översatt till svenska är ”öppen källkod” syftar på att källkoden till programmet lämnas ut öppet så vem som helst kan ta del av den, till skillnad från till exempel operativsystemet Windows där källkoden är

hemlig. Kan även syfta på öppen hårdvara där de är fritt att kopiera och använda för egna projekt.

En *mjukvarulicens*[24] är ett juridiskt redskap för att styra användandet eller återutgivandet av mjukvara. All mjukvara är i regel copyright skyddade. En typiskt mjukvarulicens ger slutanvändaren rättighet att använda en eller flera kopior av mjukvaran på sådant sätt som annars skulle kunna klassas intrång på mjukvarans ägares rättigheter enligt uppbehovslagen.

Apache License[25] är en gratis mjukvarulicens från Apache Software Foundation. Denna licens ger mottagaren omfattande rättigheter att använda och återutge material som annars vore förbjudet enligt lagarna om uppbehovsrättigheter. Apache Licensen är så kallad permissive, på svenska "tolerant", detta betyder att modifierade versioner av mjukvaran inte behöver återutges med samma licens.
I varje licensierad fil så måste eventuell copyright, patent samt varumärke behållas.

3.2 Mjukvara

Eclipse IDE [26] är världens nästa största utvecklingsverktyg för utveckling i programmeringsspråket Java. Detta är mycket tack vare att det har sina egna moduler för olika användningsområden. Eclipse är kompatibelt med flera olika operativsystem och finns på många olika språk som till exempel engelska och svenska. Eclipse distruberas som open source.

Android SDK [27][28] innehåller mjukvara för debugging och även en emulator för att testköra program direkt på dator utan att ladda upp till en fysisk androidenhet.

Arduino[29] utvecklingsmiljö för programmering av Arduinos mikrokontrollers. Innehåller verktyg för att ladda in program från dator till mikrokontrollern.

Tera Term[30] är en terminalemulator vilken kan kommunicera med bland andra seriella portar på en dator. Används här vid tester av mätvärden, möjliggör för skapande av loggfiler med data från tester för senare analys.

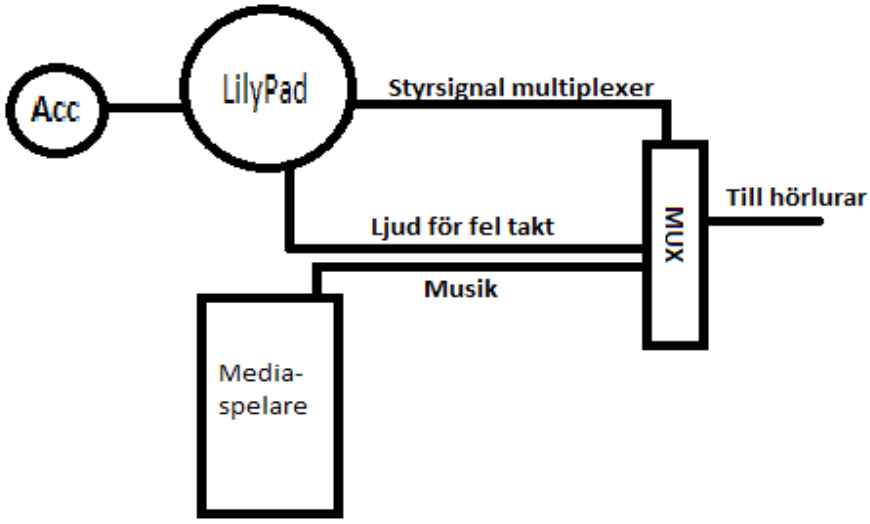
4. GENOMFÖRANDE

4.1 Hårdvara

4.1.1 Ursprunglig lösningsidé

Den ursprungliga idén att alternera musikens uppspelningshastighet konstaterades var omöjlig med hjälp av Arduino. Två anledningar till att detta inte var möjligt var först att de finns ingen generell metod att externt styra uppspelningshastigheten på en mediaspelare, den andra anledningen är att Arduino på grund av dess klena processor inte kan spela upp musik i annat än att skicka ut 8-bits ljud[1].

För att lösa problemet med hjälp av en Arduino bestämdes att istället skifta mellan två olika musikkällor. För att åstadkomma skiftningen används LilyPad och Accelerometer för att känna av takten i löpstegen, kopplat till en multiplexer för att välja antingen en separat ljudkälla för musiken när användaren är i rätt takt eller för att skicka ut ljud från LilyPaden vid fel takt.



Figur 4.1 Principskiss för lösningen

Principen fungerar rakt av, vissa komplikationer stöttes dock på under arbetets gång vilka krävde ytterligare planering och komponenter. För att kunna arbeta med en del i taget delades undersökningarna upp i två separata delar, en för accelerometerhanting och en för ljuduppspelning och multiplexer, för att sedan sammanfoga dessa till en enhet.

4.1.2 Accelerometerpositionering

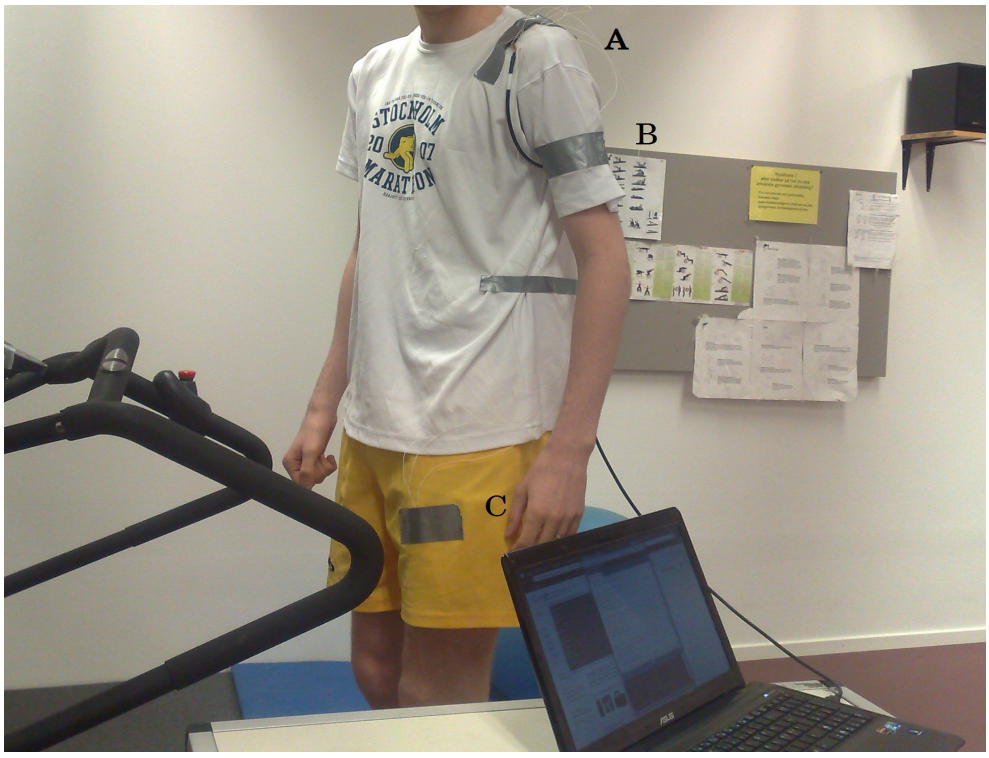
Då avläsningarna för takt bygger på avläsningar från en accelerometer vilken mäter krafter vid förändringar i hastighet krävs det att accelerometern placeras på en del av kroppen som rör sig mycket vid

1 Likt ljudet på Nintendos TV-spel ifrån mitten av 80-talet.

löpning. Ytterligare krav för att hitta en bra position är att minimera antalet sladdar för att koppla ihop produkten med mediaspelare och hörlurar, produkten måste även vara komfortabel för användaren att både ta på sig och inte vara i vägen vid träning.

Efter en del funderingar fann vi tre möjliga placeringar lämpliga, och lämpliga placeringar vars positioner utom inbördes ordning skulle vara skuldra, överarm och lår. Ytterligare placeringsmöjligheter diskuterades men uteslöts främst p.g.a. bekvämlighet och kabellängd. En placering som förmodades besitta bra egenskaper var foten, med andra ord monterad i alternativt på en löparsko. Det stora hindret för fotplacering ligger i avståndet till huvudet, dit musiken i slutändan måste komma. Ett alternativ vilket diskuterades för att lösa den problematiken var någon form av trådlös överföring, exempelvis Bluetooth. Om en trådlös lösning väljs uppkommer ett behov av två strömkällor, en för accelerometern vid foten och en för huvudenheten någonstans på överkroppen.

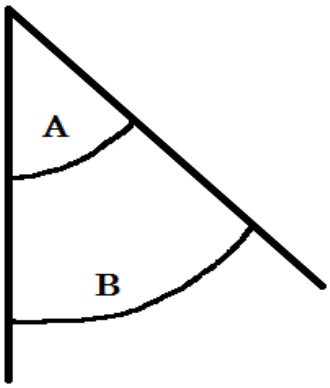
För att undersöka de tre valda positionerna och dess funktionella karakteristik utfördes undersökningar i form av löptester på löpband, se figur 4.2. Testernas huvudmål var att urskilja var en färdig produkt kan placeras, med hänsyn till rent funktionella aspekter. De tre undersökta positionerna gav upphov till liknande resultat, alla med god marginal för att urskilja skillnader i takt vid varierande hastighet. Av rent designmässiga skäl övergavs skuldran som alternativ redan vid tidigt stadium. Att lyckas ta fram en produkt för att bära på skulderbladet uppfattades som ett för stort problem att lösa, redan när en väldigt liten accelerometer tejpades fast i en t-shirt uppfattades den som störande för användaren.



Figur 4.2 Bild på accelerometerplacering vid löptest

De återstående valen mellan armen och benet har både för och nackdelar vad de gäller utslag på takten och även vad gäller bekvämlighet för användaren. Ytterligare en aspekt där armen har fördel över benet är det kortare avståndet mellan enheten och huvudet för inkoppling av hörlurar.

En färdig produkt kommer ha möjlighet att fästas både på arm och ben, avläsningarna sker med samma metod. Skillnaden är att utslaget på takt ger olika intervall beroende på val av placering för accelerometern. Då exakt placering på till exempel armen även de spelar roll saknas garantier att två separata träningstillfällen ska ge exakt samma resultat, och därav bör referenstakt väljas för varje enskilt träningstillfälle. Nedan följer ytterligare förklaring av hur hävarmen påverkas beroende på accelerometerns placering, figur 4.2.



Figur 4.2 Illustration av hävarm.

Beroende på hur lång väg accelerometern förflyttas påverkas den av olika stora krafter. Ur ovanstående principskiss kan enkelt konstateras att om de raka strecken är en persons arm, och jämförelser görs med accelerometern på två olika punkter. Båge A illustrerar placering nära axeln och båge B är en placering närmre armbågen, då längden på de bägge bågarna A och B är olika långa och tiden mellan ändlägena är exakt samma är de naturligt att B utstått en högre kraft och därav ger större utslag på accelerometern.

4.1.3 En till tre axlar

Vid genomförandet av löptester på löpband där enbart z-axeln på accelerometern används uppfattades ett något inkonsekvent beteende, till en viss del kunde lite av beteendet förklaras med glappkontakt från kablar. I de första löptesterna uppstod nämligen lite problem med att få allting att sitta fast som det var tänkt, men med en mer noggrann justering av kablarna samt med hjälp av silvertejp lyckades kablarna fästas på ett sådant sätt att det ej längre uppstod något glapp. Men även utan glappkontakt så uppstod ett inkonsekvent beteende. De tidiga testerna hade som huvuduppgift att lokalisera en position där slutprodukten med fördel kunde placeras, målet att få en liten produkt med så få sladdar som möjligt förhindrar att fler än en accelerometer används. Då tidigare tester med enbart en axel uppvisade något inkonsekvent beteende utvecklades en plan att använda en version som

slog samman resultatet av accelerometerns tre axlar och sedan delade det på tre, detta för att fånga rörelser i flera riktningar och på så vis få ett mer balanserat resultat. Förändringen gav mycket lyckat resultat där tydligt linjärt samband med bra amplitud kunde avläsas från analyseringen av värdena.

Tester som gjordes med tre axlar uppvisade mycket konsekventa mätvärden oavsett vinkeln vilken accelerometern fästes på armen/benet, dessvärre saknas loggar på detta då testerna utfördes i realtid. Jämfört med tidigare tester där enbart en axel registrerades och resultatet varierade när accelerometern fästes vid arm och ben med olika lutningar.

4.1.4 Formatering av loggvärden

Vid löptester för analys av accelerometeravläsningar och uträkningar för takt användes först följande formatering (se figur 4.3) för enkel avläsning direkt i terminalfönster.

acc one: leg
18 488
acc two: shoulder
11 516
acc three: arm
485 274

Figur 4.3 Formatering 1

Vid närmare studie av dessa värden uppmärksammades behovet av att studera flertalet avläsningar i följd för att kunna räkna ut medelvärden. Dessa avläsningar blev svåra att göra på grund av dåligt genomtänkt formatering. Att göra dessa analyser upptog för mycket tid och därav ändrades programmet för att skicka ut nedanstående formatering (figur 4.4) innan nästa löptest genomfördes.

8,518; 8,344; 0,475; 6,398; 8,619; 1,526;

Figur 4.4 Formatering 2

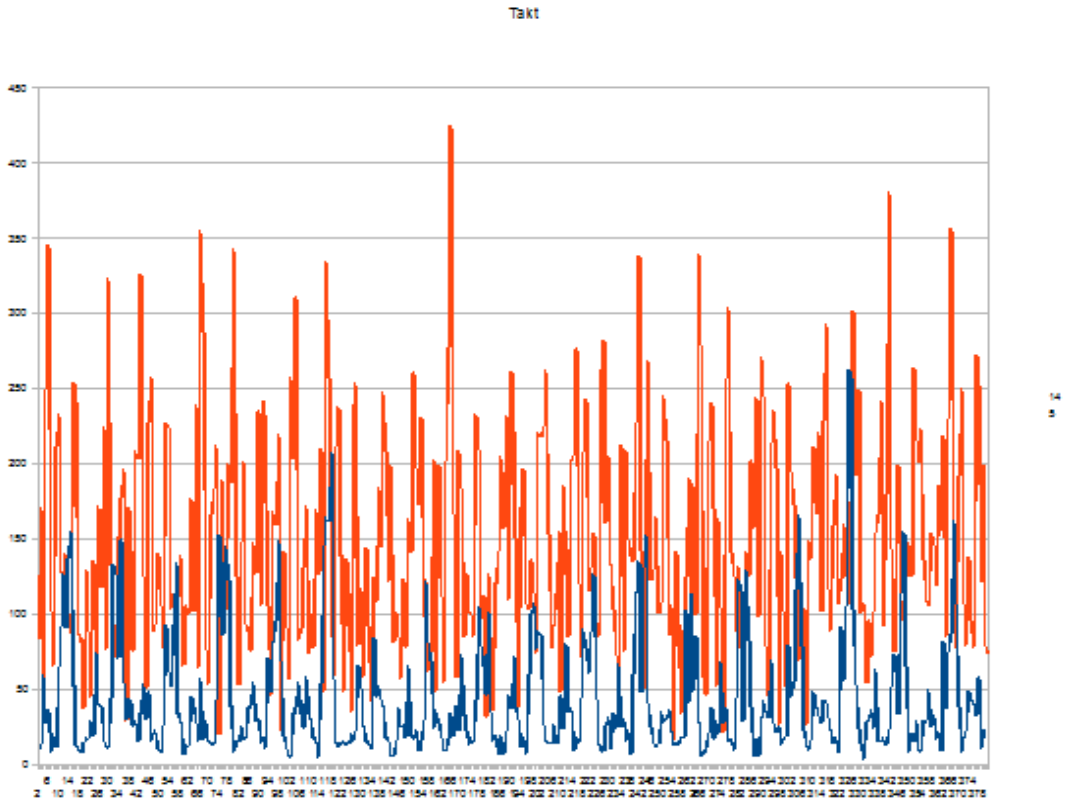
Formateringen ovan enligt följande: "Ben X-axel takt, accelerometervärde; Ben Y-axel takt, accelerometervärde; Ben Z-axel takt, accelerometer värde; Arm X-axel takt, accelerometervärde; Arm Y-axel takt, accelerometer värde; Arm Z-axel takt, accelerometervärde;"
Denna formatering medförde drastisk förenkling av analyseringsprocessen. Vid import i OpenOffice Calc separerades avläsningarna till dess kolumner redan vid inklistring av data från logfilen. Tack vare denna enkla justering minskades tiden för analys av värden från flertalet timmar till ett par minuter. Ytterligare en positiv bi-effekt var möjligheten att analysera större mängder data än tidigare. För att sätta skillnaden i siffror analyserades cirka 300 värden på den första formateringen, för att sedan använda över 3000 värden på den andra formateringen, tack vare detta gav en tiofaldig ökning av antalet värden en bättre möjlighet att få mycket mer precisa genomsnittsvärden.

4.1.5 Takt vs. Hastighet

Nuvarande avläsningsmetod att jämföra storleksskillnaden mellan två på varandra följande accelerometervärden medför god förmåga att jämföra samma person med sina egna tidigare resultat. Att jämföra två helt olika personer med avseende på takt speglar inte nödvändigtvis personernas inbördes hastigheter. Problemet ligger i att generellt översätta en takt till en verklig hastighet som inte är möjligt utan avancerade beräkningar, då hastigheten hos en person i rörelse dels beror på stegfrekvens och dels längden på varje individuellt steg. Steglängd i sin tur beror inte bara på personens längd utan skiljer även med avseende på löpteknik och detta återspeglas alltså tillbaks till hastigheten. Steglängd kan även ändras med trötthet, då en trött person tenderar att ta kortare steg än en mindre trött motsvarighet.
Summan av ovanstående är att översätta takten till reell hastighet ligger utanför ramarna för detta projekt. Anledningen till detta är att uträkningarna får för många variabler att ta hänsyn till, variabler som på något sätt måste matas in till hårdvaran under drift. Att mäta varje steg för sig och sedan med realtidsklocka ta fram hastighet är förutom GPS de möjligheter som finns för att mäta hastighet, och de bägge alternativen är uteslutna. Att mäta steglängden med enbart en accelerometer är inte möjligt och GPS är för batterikrävande.

4.1.6 Medelvärde

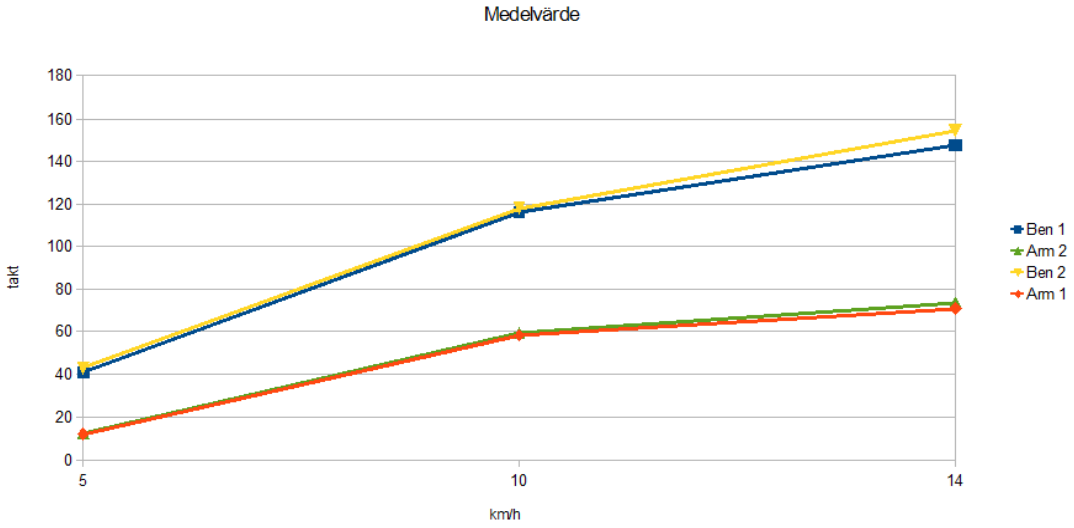
Vid tester upptäcktes programmets representation av takt uppvisade stora variationer över tiden(se figur 4.5).



Figur 4.5 Variationer i taktavläsningar

Enskilda avläsningar från de bägge mätningar överlappar med varandra, detta faktum kan tyckas att nuvarandeavläsningsmetod ej är tillräcklig för att ge något som helst resultat att använda. Fenomenet uppfattades redan under det första löptestet och var ett av de största frågetecknen och potentiella hinder för hela hårdvaruprojektet.

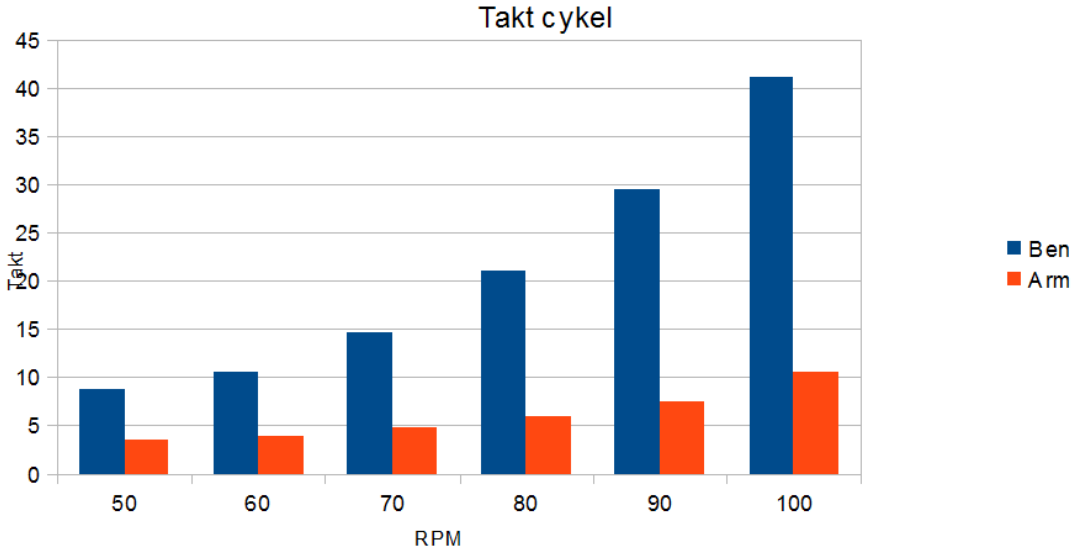
Vid närmare studier av mätningarna beräknades istället medelvärden på flera efter varandra följande mätningar. Dessa medelvärden bidrar till möjligheten att urskilja de tydliga skillnader vilka uppstår vid olika löphastigheter (se figur 4.6).



Figur 4.6 Medelvärde på takt vid olika hastigheter

4.1.7 Tester på träningscykel

Vid testtillfälle två utfördes även tester på motionscykel, då testpersonen var uppriggad med accelerometer på både arm och ben registrerades bägge värdena. Benets utslag var mycket konsekvent och tydlig ökning av takt i samband med ökning av kadensen vilken pedalerna rördes. Vad som skapade viss förvåning efter testerna var resultatet från accelerometern på armen, det visade sig att det gick att se en linjär ökning på takten även där, om än relativt små ökningar, se figur 4.7. För att få ett konsekvent beteende och kunna ha marginaler i avläsningarna rekommenderas icke att använda armen för placering av accelerometer. Då tester enbart gjorts i fast miljö på motionscykel har inte tillräckliga studier gjorts på hur dåligt väglag påverkar accelerometern, det finns möjlighet att vibrationerna från vägen skulle överskugga takten hos personen.



Figur 4.7 Takt vid motionscykel.

RPM-värden i figuren är tagna från motionscykelns informationsdisplay och när konstant RPM uppnåtts startades mätningar av takt.

4.1.8 Multiplexer

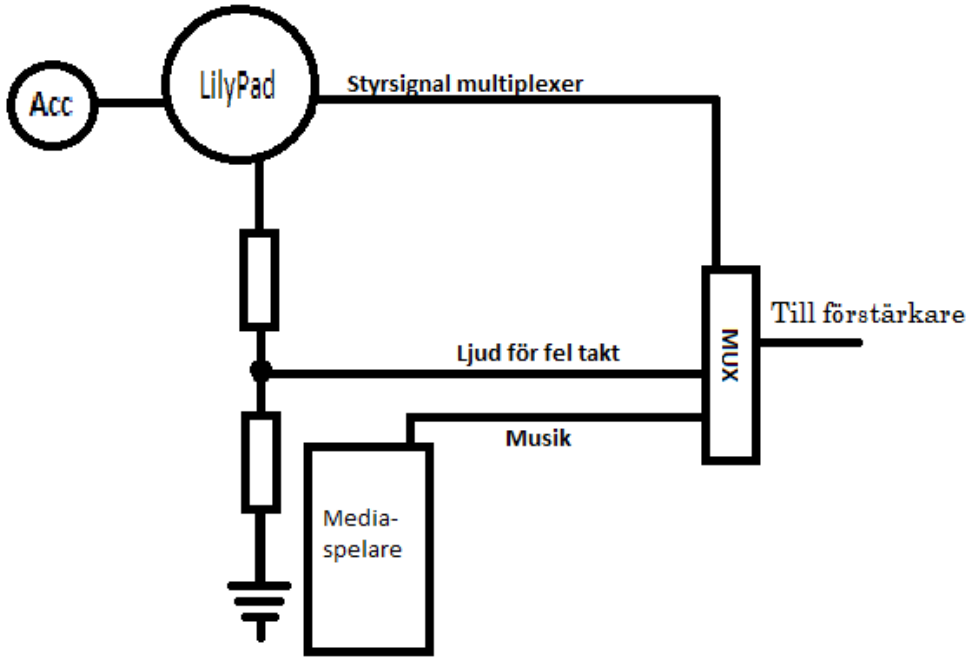
För att kunna alternera uppspelnings möjligheterna mellan Mp3 spelare och LilyPad som ljudkällor uppfattades den bästa möjligheten som att använda en analog multiplexer. Att enkelt och snabbt kunna byta ljudkälla med en enkel styrsignal från LilyPaden medför snabb omkoppling. Idén kommer ifrån det analoga telenätets circuit switching, principen att koppla mellan olika telefoner för att få en temporär koppling. Förhållandet mellan multiplexerns relativt höga inre resistans jämfört med högtalare eller hörlurars impedans ger en stor effektförlust med resultatet att musiken blir knappt hörbar. Sökningar efter bättre lämpad multiplexer misslyckades, resultatet är visserligen inte uppseendeväckande i sig då en multiplexer har som uppgift att alternera mellan ingångar för att skicka exempelvis mätvärden. Att driva en högtalare genom en krets som inte är anpassad för ändamålet var en misslyckad idé från början. Mellan multiplexerns utgång och högtalarnas ingång kommer det krävas en förstärkare för att kompensera för effektförlusten i multiplexern.

4.1.9 Förstärkare

Den förstärkare som testats i prototypen är av modellen TPA152 75mW stereoförstärkare från Texas Instruments, vilken enligt tillverkaren är konstruerad för just drift av hörlurar från portabla mediaspelare. För att få förstärkaren att fungera används värden på kondensatorer och resistorer från tillverkarens exempelschema från kretsens datablad.

Vid testkörning av komplett krets uppskattades ljudnivån att vara likvärdig genom koppling via multiplexer och förstärkare som en direktkoppling från Mp3-spelaren till högtalare. Ytterligare mätningar för att säkerställa samma signalnivå ansågs överflödiga då upplevd ljudnivå är vad en användare ställer krav på.

Nivån på de ljud LilyPaden sänder till multiplexern är mycket högre än de Mp3-spelaren sänder. Eftersom LilyPaden saknar någon volymkontroll, krävs extern sänkning där behov fanns att minska signalstyrkan mellan LilyPadens utgång och multiplexerns ingång. Nuvarande lösning är en enkel spänningsdelare med två resistorer, möjligen bör ena resistorn ersättas med potentiometer för justering beroende på olika hörlurar. De nuvarande resistorerna är ett 100k ohm och ett 1k ohm, dessa valdes genom att helt enkelt byta till en resistor med högre resistans tills det att en godtycklig ljudnivå uppnåddes, se figur 4.8 för uppdaterad principskiss med spänningsdelare på rätt plats.



Figur 4.8 Principskiss med spänningsdelare.

4.1.10 Distorsion i förstärkarkrets

Någonstans på väg från mediaspelare till hörlurar uppstår det viss distorsion, dock är dess uppkomstkälla ej specifikt bekräftad. Uppkomsten är svår att helt isolera då de ingående komponenterna motverkar varandras effekt, multiplexern stjäl effekt och förstärkarens uppgift är förstärka signalen för att kompensera för förlusten. De resistorer och kondensatorer som används kan även de bidraga till brus i signalerna. Viss störning kan även härstamma ifrån ojämn strömförsörjning till förstärkare eller multiplexer, då dessa kopplingar ej innefattar stabila spänningsregulatorer.

För att teoretiskt undersöka störningskällan konsulterades de ingående kretsarnas datablad med hänsyn till dess distorsionsegenskaper. Den siffra som är intressant är Total Harmonisk Distorsion[31] vilket kan sammanfattas som summan av alla störningar på en signal.

Multiplexern CD4053BE som används har specificerad total harmonisk distorsion på 0.3% vid 2V sinusformad insignal med frekvensen 1kHz. Vad som sker vid lägre inspänning och högre frekvens finns ej specificerat i databladet. Förstärkarens motsvarande THD vid samma frekvens är 0.01%. Motsvarande siffror för THD hos resistorer och kondensatorer finns inte att tillgå i datablad och dessa komponenters inverkan är således oklar. Totala siffran för de förstärkare och multiplexer bör alltså hamna på under 0.5% vilket verkar vara en liten siffra. Enligt hemelektroniktillverkaren Yamaha[32] påstås störningar under 1% inte vara märkbara för lyssnare. Med enbart dessa siffror i åtanke borde inte störningarna uppfattas, trots detta upplevs störningarna vara klart märkbara vid lyssningstester där samma musikstycke spelas alternerande mellan förstärkare och direkt koppling från mediaspelare. Möjligtvis är multiplexerns distorsion högre vid de frekvenser som musik innehåller, och därav ger upphov till de störningar som hörs. Ett annat möjligt problem med just multiplexern är att signalerna från den bärbara mediaspelaren är så svaga så ytterst små störningar ger upphov till stora variationer i signalerna, då inspänningen till multiplexern ökar, sänks THD värdet något. Enligt detta förmodas inspänningar på under 2V ge upphov till högre distorsion än de specificerade 0.3% vid ovan nämnda 2V.

Då förstärkarkretsen är designad för att klara av att spela upp ljud utan större förvrängningar förmodas störningarna härstamma ifrån multiplexern. Ett av alternativen till multiplexern skulle vara att de keramiska kondensatorer som används inte klarar av att hantera de frekvenser som bearbetas. Ytterligare undersökningar utelämnas då projektmålet inte är att tillverka en perfekt produkt utan endast att få fram fungerande principer.

4.1.11 Batteri

Målet att tillverka en liten produkt ställer stora krav på en stabil strömförsörjning, som fortfarande är kompakt och väger lätt. Då relativt många komponenter används krävs hög kapacitet för att få acceptabel driftstid. Ett antal tänkbara alternativ finns att tillgå, såsom inbyggt laddningsbart eller utbytbara AA/AAA batterier.

Bägge alternativen har sina klara fördelar och nackdelar. Laddningsbart har låg vikt i förhållande till den kapacitet de innehar, dess största nackdel är att tar batteriet slut under en löprunda går de inte att byta mot nytt batteri likt att bara byta ut ett AA batteri. Detta kanske skulle kunna lösas med att det laddningsbara batteriet sitter på ett sådant sätt att även det kan bytas ut, liknande som på en mobiltelefon, så att det skulle finnas möjlighet att byta ut det med ett fullt laddat batteri. Dock så kanske det inte är så stor sannolikhet att en användare tar med sig extra batterier ut när de springer.

Det mest lämpliga är ett uppladdningsbart batteri av typen Lithium-Ion eftersom de räcker mycket längre än motsvarande Nickel-Metallhydrid, dessutom lider de inte av samma grad av självurladdning som de

sistnämnda tenderar att göra. Ytterligare en anledning till att använda denna lösning är att andra bärbara produkter på marknaden idag kommer med inbyggda laddningsbara batterier, och på så sätt få potentiella kunder att känna igen sig.

4.1.12 Kretskort

Då samtliga tester och utveckling skett med färdiga prototypkort för LilyPad och accelerometer krävs en omarbetning och planering för att ha möjlighet till en egen säljbar produkt. Lösningen ligger i att tillverka ett eget kretskort med nödvändiga komponenter monterade. Tiden för att designa ett kretskort från grunden avsattes aldrig vid planering av projektet. Försök påbörjades att designa en komplett produkt från grunden, dessa försök avbröts dock främst på grund av tidsbrist.

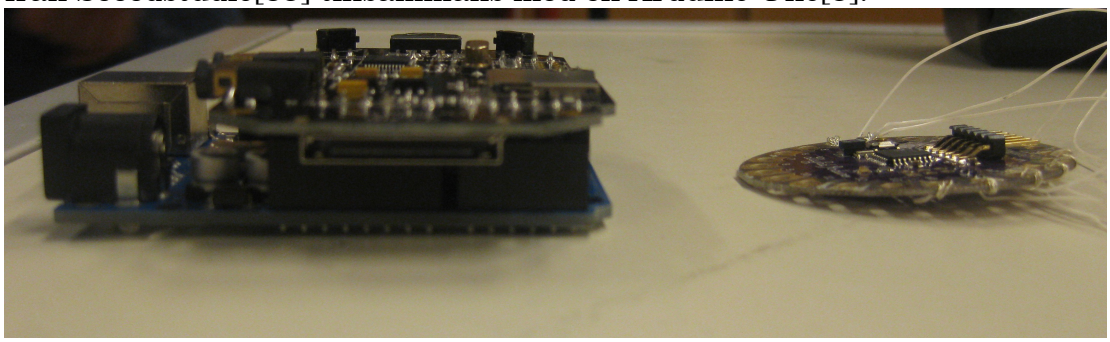
4.2 Undersökta alternativ till nuvarande hårdvarulösning.

4.2.1 Inledande tankar

Under projektets genomförande diskuterades och utvärderades ytterligare möjligheter till att lösa problemet hårdvarumässigt på andra sätt än den multiplexerlösning som tidigare presenterats. Två av de utvärderade idéerna beskrivs i följande del.

4.2.2 Music Shield

Ett alternativ till att tillverka ett tillägg för befintliga mediaspelare är att tillverka en komplett mediaspelare med inbyggd accelerometer för att känna av takten. Detta alternativ undersöktes med hjälp av Music Shield från Seeedstudio[33] tillsammans med en Arduino Uno[3].



Figur 4.9 Storleksjämförelse Uno med Music Shield och LilyPad

Ur evalueringssynpunkt är inte storleken ett problem(se figur 4.9), men att bära runt på något i denna storlek känns otänkbart ur användarsynpunkt. Den skrymmande storleken går givetvis precis som multiplexerlösningen att minska avsevärd genom att tillverka allting på samma kretskort.

Själva uppspelningsfunktionerna i Music Shielden fungerade utan några som helst problem. Det stora funktionella problemet som fick oss att överge detta lösningsalternativ ligger i avsaknaden av möjlighet att ändra uppspelningshastighet under drift, med detta i åtanke försvinner den

största fördelen med hela idén om en integrerad produkt. Trots förhoppningar att slippa använda multiplexer och förstärkare hade dessa fortfarande krävas för att bryta ljudet och spela upp ljud från en annan källa. Ytterligare en anledning till att vi övergav just denna lösning var att produkten hade blivit dyrare än den variant som kopplas samman med befintlig Mp3-spelare, och därav troligtvis mer svårsåld som färdig produkt.

Givetvis finns andra möjligheter till att konstruera kompletta mediaspelare på hårdvarunivå. Dessa innefattar mer avancerade mikroprocessorer eller andra chip än VS1053b[34], de chip Music Shielden baseras på, som möjligtvis stödjer alternering av uppspelningshastighet under drift.

Att gå ifrån Arduino till fördel en annan plattform hade inneburit stora ändringar i projektet. Ny bootloader[7] hade behövts och likaså hade de krävts byte till ny utvecklingsmiljö. En så drastisk ändring skulle kräva väldigt mycket extra arbete och framförallt tid, den tiden vilken det krävts för att undersöka möjligheter på andra hårdvaruplattformar än Arduino avsattes inte i detta projekts planering. Ifall vidareutveckling skulle göras vore de lämpligt att undersöka kraftfullare plattformar än Arduino för tillverka en komplett produkt.

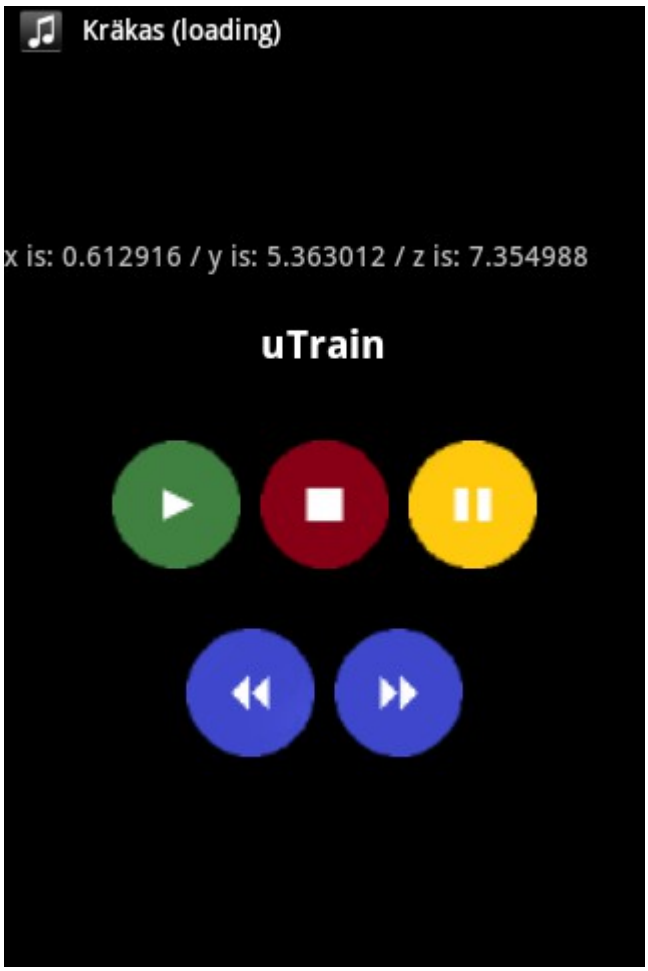
4.2.3 Equalizer

En alternativ lösning för att åstadkomma den egentliga idén med att förvränga originalsignalen hade varit att konstruera en analog ljudprocessor att koppla mellan mediaspelare och hörlurar. Detta konstaterades på tidigt stadium vara för avancerat med tanke på våra väldigt begränsade kunskaper inom analog signalhantering.

4.3 Applikationsutveckling för Android

4.3.1 Uppstart av applikationsarbete

Vid uppstart av applikationsutvecklingen beslutades det att utgå ifrån befintlig open source kod för en mediaspelare eftersom det skulle ge en bra grund att stå på. Ytterligare en anledning till att utgå från open source var att det skulle spara tid för utvecklingen i ett redan pressat tidsschema, detta är inte optimalt då det kan ta tid att sätta sig in i och förstå en annan programmerares kod och programmet blir mer optimalt om man själv kodar det som man vill ha det. Open source koden är använd enligt Apache License, version 2.0[25]. Koden har sedan modifierats för att den ska passa in mer med koden för träningsappen. Den kod som är specifik för träningsappen har stegvis utvecklats mer och mer. Till en början programmerades den grafiska layouten (se figur 4.10), det så kallade GUI:t, med knappar, text och liknande. Bilderna för knapparna och loggan för applikationen skapades med programmet Adobe Photoshop.



Figur 4.10 Bild på GUI, aktuell låttitel visas längst upp.

När GUI:t var skapat flyttades fokus till det som hela programmet bygger på, nämligen accelerometern som finns inbyggd i nyare telefoner. Först skapades ett enklare program separat för att undersöka accelerometerns funktion och för att se vilka värden som accelerometern skickar ut vid olika positioner av telefonen. Mer specifikt om testningen av accelerometern går att läsa i ett avsnitt längre ner. När programmet fungerade och värdena från accelerometern analyserats så började arbetet med att sammanfoga de två programmen till ett enda.

4.3.2 Kodexempel accelerometeravläsning

Då programmets funktionalitet till stor del bygger på avläsningar av accelerometervärden följer ett par stycken av kod för att visa hur avläsningen går till:

För att utelämna en hel del kod såsom variabler och liknande så visar ”…..” att det finns mer kod innan eller efter den visade koden.

Funktionen onCreate (kodavsnitt 1) körs när programmet startas upp, den skapar en sensorManager som har hand om sensorn samt sensorn som läser av accelerometern. Funktionen skriver även ut resultaten från accelerometern till en så kallad TextView, ett textfönster, som visas på

skärmen. Om sensorn inte fått något resultat från accelerometern så skrivs ”No result yet” ut på skärmen istället. Dessa utskrifter användes främst under utvecklingsfasen av applikationen för att kontrollera accelerometerns funktionalitet. Denna kod finns dock fortfarande kvar i programmet för att det är tänkt att takten som användaren håller skall skrivas ut i det textfönstret istället.

```
…..
" public void onCreate(Bundle savedInstanceState) {
        …..
        sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        sensor =
sensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER).get(0);

        result = (TextView) findViewById(R.id.result);    //Prints
accelerometer readings to screen
        result.setText("No result yet");    //If no results from accelerometer,
print this.
….. "
```

Kodavsnitt 1 onCreate

Funktionen onSensorChanged (kodavsnitt 2) är, precis som namnet indikerar, det som händer när sensorn som läser av accelerometern läser av ett nytt värde som inte var det samma som den läst av tidigare. Det som funktionen då gör är helt enkelt att den sparar X, Y och Z värdena från accelerometern till varsin variabel och anropar sedan på funktionen refreshDisplay.

```
 " public void onSensorChanged(SensorEvent event) {
        x = event.values[0];    //Load X-value from accelerometer
        y = event.values[1];    //Load Y-value from accelerometer
        z = event.values[2];    //Load Z-value from accelerometer
        refreshDisplay();
….. "
```

Kodavsnitt 2 onSensorChanged

Funktionen refreshDisplay (kodavsnitt 3) anropas alltså av onSensorChanged och dess uppgift är helt enkelt att uppdatera de värden som skrivs ut på skärmen. I den kompletta versionen är inte utskriften av accelerometervärden relevant, utan detta var nödvändigt under utvecklingsfasen för att få koll på accelerometerns beteende. De viktiga som anropas under onSensorChanged är då taktuträkningen.

```
 " private void refreshDisplay() {
        String output = String
        .format("x axis: %f ; y axis: %f ; z axis: %f", x, y, z);
        result.setText(output);
   } "
```

Kodavsnitt 3 refreshDisplay

4.3.3 Tester av accelerometervärden

Då telefonens accelerometer skiljer sig ifrån ADXL335 vad det gäller signaler och därför krävdes liknande tester som utfördes med hårdvaran för att bestämma dess beteende vid skiftande hastigheter.
Den största anledningen till att koden från hårdvaran ej gick att överföra rakt av till applikationen var att ADXL335 skickar ut värden i ett intervall mellan 0 till 1052, medans det största intervallet vi lyckades få ut från två stycken telefoner från olika tillverkare var -10 till 10.
Överraskande nog så erhölls samma resultat från accelerometern på de båda telefonerna fast den ena var från HTC och den andra från Samsung. Detta kanske beror på att båda mobiltelefontillverkarna köper in samma sorts accelerometer till sina telefoner. Detta är till fördel för applikationen eftersom att det annars skulle behövas någon sorts logik för att bestämma vilken tillverkare telefonen kom från och vilka värden som då skickas från accelerometern.
Även ett mindre löptest genomfördes där de vanligaste resultaten från accelerometern granskades och bestämdes, detta test var ganska likt de löptester som genomfördes med hårdvaran.

4.3.4 Activity vs Service

Under arbetets gång så framgick det att om man använder sig av så kallade activitys, aktiviteter, så tappar applikationen fokus när telefonen går in i sömn/stand-by läge. En typisk applikation i Android består oftast av flera aktiviteter som jobbar med varandra, där en aktivitet tjänar som en huvudaktiviteten och det är den som visas när applikationen startas upp. En activity beskrivs enligt följande på Androids hemsida:

"An Activity is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map. Each activity is given a window in which to draw its user interface. The window typically fills the screen, but may be smaller than the screen and float on top of other windows." [35]

Därför påbörjades ett intensivt arbete med hur detta problem skulle kunna lösas. Det visade sig att om applikationen är konstruerad med service, tjänster, istället för aktiviteter så kommer det att fortsätta köras även om telefonen går in i stand-by och de kan även hantera att flera olika tjänster som kör i bakgrunden på samma gång. För att beskriva tjänster lite mer ingående citeras än en gång androids hemsida:

"A Service is an application component that can perform long-running operations in the background and does not provide a user interface. Another application component can start a service and it will continue to run in the background even if the user switches to another application. Additionally, a component can bind to a service to interact with it and even perform interprocess communication (IPC). For example, a service might handle network transactions, play music, perform file I/O, or interact with a content provider, all from the background."[36]

Omkonstruktionen av applikationen skedde i flera steg för att säkerställa att dess funktionalitet inte förstördes. Först så testades det att förflytta mediaspelarens funktionalitet till en tjänst och detta gick tämligen enkelt att utföra. När sedan accelerometerns funktionaliteter skulle flyttas till en tjänst så blev det mer problematiskt eftersom bland annat taktuträkningen och if-satser som bestämmer vad applikationen skall göra vid olika takter använder sig utav accelerometerns uträkningar. När applikationens huvudaktivitet försökte kalla på funktioner som numera låg i en annan klass så uppstod ett fel med static references. Därför påbörjades förflyttningar av så mycket logik och funktioner som möjligt till accelerometertjänsten. Resultatet blev tillslut att mediaspelaren och accelerometern körs parallellt i två separata tjänster, medans logiken som påverkar mediaspelartjänsten ligger i accelerometertjänsten. Applikationens funktionalitet fastslogs i ett kortare löptest där det provades att springa under, i och över den satta takten. Applikationen fungerade på önskvärt sätt i alla av dessa tre testfall.

4.4 Problem under applikationsutvecklingen

4.4.1 Problem med Eclipse

För det första så har programmeringsverktyget Eclipse ställt till en hel del problem, programmet har krånglat många gånger på flera olika sätt. Eclipse har behövt startas om med jämna mellanrum på grund av att vissa fel uppstått vid kompilering och de gick sedan ej att få bort om programmet inte stängdes ner och startades upp igen på nytt. Ett annat problem var att när GUI:t utvecklades så vägrade koden för knapparnas layout gå igenom kompilering, detta på grund av ett felmeddelande som sa att knapparna ej kunde ritas ut. Vid felsökning så framkom det att lösningen på detta problem skulle vara att lägga de önskade bilderna i rätt mapp i projektet, det enda problemet var att bilderna redan låg i exakt den mappen. Problemet kvarstod fast bilderna togs bort och lades in på nytt och även när de flyttades till en annan undermapp i projektets resurskatalog. I ren desperation kopierades bilderna över till varje enskild undermapp i projektet och förvånande nog så resulterade detta i att programmet kunde kompileras. Detta ledde dock till att programmet ökade drastiskt i storlek och detta är inte särskilt önskvärt, därför togs bilderna bort från de olika undermapparna en efter en och programmet kompilerades efteråt. Tillslut hade alla bilder som låg i fel undermappar tagits bort och bilderna låg alltså enbart i den mapp som de låg i från början, men programmet kunde ändå kompileras utan problem.

4.4.2 Operativssystemsberoende problem

Ett annat problem som stöttes på vid användningen av Eclipse under Windows var att en viss kod kunde ge uppstått till ett fel "x", men när exakt samma kod kördes i Eclipse under Linux så uppstod felet "y" istället. Detta ledde till problem när koden skulle felsökas eftersom programmeringen oftast skedde i olika operativsystem.

Ytterligare ett problem upptäcktes vid användning av @override för att överskugga metoder från basklasser, där en av Eclipseinstallationerna som kördes under Windows påpekade att de var felaktig användning av ovan nämnda annotation, medans den andra installationen som även den kördes under Windows inte uppvisade samma fel, likaså fungerade annotationen i den version av Eclipse som kördes under Ubuntu i en virtuell dator genom VMWare Workstation.

4.4.3 Problem med Android SDK

Även vid användning av Android SDK upptäcktes fel med Eclipse eller möjligen fel med Android SDK kombinerat med Eclipse. Under projektets gång så har SDK:n ominstallerats totalt 5 gånger för att Eclipse tappat kontakten med SDK installationen, Android kod gav felmeddelanden och det gick inte att skapa nya Android-projekt.
Ett annat problem som varit riktigt irriterande under arbetet med applikationen är att den Androidemulator som följer med Android SDK inte fungerar något vidare. Det kan vara allt ifrån att den fastnar i laddningen av emulatorn, att applikationen inte startas upp, att applikationen eller hela emulatorn har väldigt hög fördröjning innan saker sker och dessutom så fungerar vissa funktioner inte i emulatorn. Exempel på de funktioner som inte fungerar är bland annat accelerometeravläsningar och eftersom vår applikation bygger på användning av accelerometern så har detta orsakat problem för testning.

4.4.4 Kompatabilitetsproblem äldre telefoner

För att försöka lösa problemet med den felande Androidemulatorn så användes istället fysiska Android telefoner för testningen av applikationen. Men även detta visade sig vara problematiskt. På en av de testade telefonerna, en HTC Hero, så startar applikationen upp men när man trycker på knappen för att starta musikuppspelningen så slutar programmet att svara. Först misstänktes att detta var på grund av något fel i applikationen, men efter felsökning så hittades inget som antydde på detta. När applikationen senare testades på en kraftfullare Samsung Galaxy Ace så fungerade den utan någon antydan på problem. Detta kan antyda på att vissa äldre Android telefoner skulle kunna få problem att köra applikationen på grund av telefonens hårdvara, dock så visar våra tester att programmet i sig inte är särskilt krävande så det skulle kunna bero på just denna telefon.

4.5 Ryggsensor

På grund av tidsbrist uteblev undersökningar av ryggsensor. De alternativ som diskuterades innehöll bland annat ultraljud och infraröda avståndskännare. Planer fanns på att undersöka rörelsedetektorer likt utelampor som tänds när någon går förbi inom dess detektionsområde.

5. RESULTAT

5.1 Sammanfattning

Den produkt som tagits fram räknar ut takten som användaren håller vid till exempel löpning och jämför sedan den aktuella takten med den takt som användaren valt. Användaren kan antingen använda en fördefinierad takt eller ställa in sin egen takt genom att trycka på en knapp och sedan springa en kort stund i den takten som önskas. Programmet räknar då ut den genomsnittliga takten som hölls och sätter denna takt till den valda takten. Takten räknas ut med hjälp av en accelerometer som använder sig utav tre axlar för att få en så exakt beräkning som möjligt. För att programmet inte ska bli alltför känsligt för eventuella fel i avläsningen från accelerometern så filtreras den aktuella takten genom att räkna ut medelvärdet av ett flertal avläsningar. Eftersom produkten dessutom läser av accelerometern ett flertal gånger per sekund så hjälper denna filtrering på det sättet att användaren inte varnas om det skulle vara så att takten ligger fel i en halvsekund eller något liknande.

Produkten kan placeras på armen eller benet, de positionerna som testats är överarmen respektive framsidan av låret. Det skulle säkert fungera även på fler ställen då det noterats att värden från skulderbladet fungerade förvånansvärt bra, men det är inget som rekommenderas och funktionaliteten kan inte garanteras. Enligt de tester som utförts så fungerar produkten bäst på benet vid cykling medans det inte verkar spela någon större roll om den sitter på benet eller armen vid löpning.

En multiplexer används för att byta mellan olika ljudkällor, när användaren håller rätt takt så skickas ljudet från mediaspelaren ut till hörlurarna och när användaren håller fel takt så skickas ett ljud ut från mikrokontrollern. Vid för hög takt skickas högfrekventa toner med i snabbt tempo och när för låg takt hålls så skickas lågfrekventa toner i långsamt tempo till hörlurarna.
På grund av att multiplexrar inte är designade för att behandla ljud så erhölls en effektförlust när ljudet kopplades igenom denna. För att kompensera dessa förluster användes en förstärkare som är konstruerad för att driva hörlurar, dock så bidrog förstärkaren till att ljudet från mikrokontrollern blev extremt högt och detta kunde ej regleras från på grund av att det ej fanns någon inbyggd volymkontroll. Därför så skapades en spänningsdelare mellan mikrokontrollern och den ena ingången till multiplexern för att få ner styrkan på ljudet innan det nådde förstärkaren.
I detta läge sker strömförsörjningen med matning från USB, detta sker på grund av att produkten hittills bara testats i labbmiljö.

5.2 Jämförelse mellan de olika lösningsmetoderna.

Vid jämförelse mellan hårdvarulösningen och en komplett Androidapplikation finns flera avvägningar att göra. Kostnad, användarvänlighet, funktionalitet, extra funktioner som kan byggas in, attraktionskraft med möjlighet till kommersiell framgång och givetvis ekonomiska förtjänstmöjligheter.

Icke tekniska aspekter såsom attraktionskraft är författarnas högst personliga åsikter och funderingar. Att springa runt med en stor telefon fastsatt på armen känns, enligt oss, klumpigt. Även tanken att springa runt med en telefon som kostar ett par tusen kronor i inköp såpass oskyddat har en risk i sig, inte enbart för telefonen utan det skulle även kunna innebära en risk för användaren för att till exempel locka till sig rånare. Dock så vet vi att även fast vi har såpass negativa tankar om detta så finns det personer som har telefonen med sig när de tränar, ofta genom att köpa ett armband så att telefonen går att fästa på armen.

Att bära en liten mediaspelare kopplad till en liten kompakt produkt, såsom den multiplexerlösning som utvecklats här är, kan kännas lättare och även mindre avskräckande med sitt lägre pris och mindre ömtåliga fysiska karaktär. En stor nackdel med en liten enhet att koppla till en befintlig mediaspelare är att de ska drivas av separata strömkällor,

Funktionsmässigt fungerar både hård- och mjukvarulösningarna. Möjligheterna att utöka med extra funktionalitet utan extra tillbehör eller kostnad är större som Androidapplikation, möjligheten att utnyttja GPS bland andra.

6. SLUTSATS

6.1 Resumé

Den produkt som tagits fram räknar ut takten som användaren håller vid till exempel löpning och jämför sedan den aktuella takten med den takt som användaren valt. Användaren kan antigen använda en fördefinierad takt eller ställa in sin egen takt genom att trycka på en knapp och sedan springa en kort stund i den takten som önskas. Programmet räknar då ut den genomsnittliga takten som hölls och sätter denna takt till den valda takten. Takten räknas ut med hjälp av en accelerometer som använder sig utav tre axlar för att få en så exakt beräkning som möjligt. För att programmet inte ska bli alltför känsligt för eventuella fel i avläsningen från accelerometern så filtreras den aktuella takten genom att räkna ut medelvärdet av ett flertal avläsningar. Eftersom produkten dessutom läser av accelerometern ett flertal gånger per sekund så hjälper denna filtrering på det sättet att användaren inte varnas om det skulle vara så att takten ligger fel i en halvsekund eller något liknande.

Produkten kan placeras på armen eller benet, de positionerna som testats är överarmen respektive framsidan av låret. Det skulle säkert fungera även på fler ställen då det noterats att värden från skulderbladet fungerade förvånansvärt bra, men det är inget som rekommenderas och funktionaliteten kan inte garanteras. Enligt de tester som utförts så fungerar produkten bäst på benet vid cykling medans det inte verkar spela någon större roll om den sitter på benet eller armen vid löpning.

En multiplexer används för att byta mellan olika ljudkällor, när användaren håller rätt takt så skickas ljudet från mediaspelaren ut till hörlurarna och när användaren håller fel takt så skickas ett ljud ut från mikrokontrollern. Vid för hög takt skickas högfrekventa toner med i snabbt tempo och när för låg takt hålls så skickas lågfrekventa toner i långsamt tempo till hörlurarna.
På grund av att multiplexrar inte är designade för att behandla ljud så erhölls en effektförlust när ljudet kopplades igenom denna. För att kompensera dessa förluster användes en förstärkare som är konstruerad för att driva hörlurar, dock så bidrog förstärkaren till att ljudet från mikrokontrollern blev extremt högt och detta kunde ej regleras från på grund av att det ej fanns någon inbyggd volymkontroll. Därför så skapades en spänningsdelare mellan mikrokontrollern och den ena ingången till multiplexern för att få ner styrkan på ljudet innan det nådde förstärkaren.

I detta läge sker strömförsörjningen med matning från USB, detta sker på grund av att produkten hittills bara testats i labbmiljö.

På Androidapplikationen har ett GUI skapats för att vara enkelt och lättförståeligt.

Applikationens logik är uppbyggt på liknande sätt som för hårdvaran, hela programmet bygger alltså på användning av telefonens inbyggda accelerometer som konstant läses av. Den största skillnaden mellan applikationen och hårdvaran är att accelerometrarna är av olika fabrikat och skickar därför ut olika värden, detta fick tas hänsyn till och alla beräkningar fick göras om på nytt. Applikationen använder sig utav tjänster istället för aktiviteter vid användning mediaspelaren och accelerometern eftersom programmet måste fortsätta att köra även om telefonen går in i viloläge, och detta klarar enbart en tjänst av. Själva fönstret som visas för användaren i programmet är dock en aktivitet, detta eftersom knapparna enbart behöver fungera när telefonen är aktiv och inte när den är i viloläge.

6.2 Kritisk diskussion

Distorsionen i förstärkaren är ju självklart helt oönskad, dock nöjer vi oss med att bevisat att vår idé fungerar. Att göra en perfekt förstärkare ligger utanför projektets ramar, detta och ytterligare detaljer behöver fintrimmas innan en produkt kan göras kommersiell. Givetvis skulle vi kunnat undersöka flera olika multiplexrar för att bestämma vilken som ger minst störningar, liknande undersökningar skulle även göras med övriga komponenter.

En lösning som hade vart tänkbar är en mer avancerad mikroprocessor för att ta in ljud från mediaspelare och sedan förvränga originalljudet, detta för att slippa byta ljudkälla och de problem de innebär. Funderingar på detta fanns redan relativt tidigt i projektets skede, men övergavs på grund av tidsaspekter.

Alternativet att göra en komplett mediaspelare hade givetvis fungerat, dock är författarnas uppfattning att det hade blivit svårare att sälja in en sådan produkt än ett tillägg till befintlig mp3-spelare. Detta anser vi på grund av att om kunden redan köpt en dyr mp3-spelare med stort lagringsutrymme, touch display och en massa andra finesser så vill nog hellre en kund använda sig av den än att köpa en hel produkt som troligtvis innehåller en sämre mp3-spelare än vad kunden hade från början. Den kompletta mediaspelaren skulle dessutom bli dyrare att tillverka än att tillverka en sådan produkt som vi har utformat som använder sig av en redan befintlig mp3-spelare. På samma sätt resonerar vi om telefoner och vi tror därför att i det här läget har en app till en befintlig androidtelefon har större chans att sälja än den kompletta mediaspelaren.

Anledningen till att vi ej har utvecklat någon ryggsensor som varnar för objekt som kommer bakifrån användaren är helt enkelt att det uppstod diverse problem under projektets gång som gjorde att vi tappade tidsplanen, och då bestämde vi oss för att det var det som var enklast att välja bort ifrån projektet. Problemen var främst med väntan på beställda komponenter samt effekt förlusten på grund av multiplexern som gjorde

att vi fick lov att spendera mer tid på detta än vad vi planerat. Kanske var vi överambitiösa när vi planerade hur lång tid arbetet skulle ta, men bortsett från detta så har vi hållit väldigt bra takt under hela projektets gång.
Dock av det vi hann planera och fundera över så känns ryggsensorn som något helt genomförbart och ingen omöjlig uppgift, det största problemet hade troligtvis blivit arbetet med att försöka skärma av signalen så att sensorn inte skulle varna för till exempel träd på sidan av vägen om man springer i skogen.

Om vi skulle göra om hela projekt med den kunskapen som vi fått under arbetets gång så skulle vi ändrat mer eller mindre hela produktens uppbyggnad. Vi skulle använt oss utav en annan mikrokontroller istället för ATMEGA328 som kunde gett oss möjligheten att sköta musikuppspelningen direkt från den. På så sätt så hade vi sluppit många onödiga komponenter som dessutom krävt väldigt mycket av vår tid. Vi skulle inte behövt en multiplexer för att välja vilken ljudkälla som ska spelas upp, och då hade vi inte heller behövt en förstärkarkrets och inte heller någon spänningsdelare eftersom det inte hade uppstått skillnader i volym mellan ljudkällor.
Vi hade även använt oss utav det formatet som vi skickar värden från LilyPaden med på en gång, istället för att börja med det första formatet som vi förlorade många timmars arbete på att försöka analysera.

Vid utvecklingen av Androidapplikationen så hade vi, om det fanns möjlighet för det tidsmässigt, utvecklat vår egen applikation från grunden. Vi hade även börjat arbeta med tjänster direkt istället för aktiviteter, då vi fick lov att omstrukturera i koden när vi bytte mellan dessa upplägg.

Vi hade troligen inte heller arbetat med Eclipse i Windowsmiljö eftersom vi stött på alla möjliga besvär när vi programmerat, och detta händer tydligen mycket mer frekvent i Windows än i Linux när man utvecklar mjukvara i just Java/Android.

6.3 Möjligheter till fortsättning

6.3.1 Hårdvara

Det finns stora möjligheter till utveckling av hårdvaran. Det första naturliga steget är att konstruera ett komplett kretskort med alla komponenter istället för nuvarande lösa moduler. Anledningar till detta är främst pris och användarvänlighet, att köpa in lösa komponenter istället för kompletta LilyPad-moduler skulle bli mycket billigare. Den andra stora anledningen är storlek på produkten, ett kretskort ger möjlighet att krympa storleken och på så sätt bidra till smidigare placering och mindre störande för användaren att bära på.

En självklar fortsättning är även ryggsensorn som var planerad att ingå i projektet från början. Detta var bland det mest spännande enligt för oss

och vi tror att det absolut skulle vara genomförbart, det skulle närmast kunna jämföras med en sådan sensor som varnar bilförare när de backar nära något objekt som kan finnas i nyare bilar. Den största skillnaden skulle troligen vara att vår sensor skulle behöva ha något längre räckvidd.

6.3.2 Androidapplikation

Även här finns möjligheter till fortsättning på flera plan. Bland annat att lägga till stöd för GPS positionering för hastighetsuträkning som komplettering för enbart taktfunktionen. En annan möjlighet vore att lägga till en funktion för styrd intervallträning där måltakten alterneras mellan en högre takt och en lägre med förbestämda intervall.
Applikationen skulle egentligen kunna förbättras och utökas nästan hur mycket som helst, direkt en ny idé dyker upp så skulle det bara vara att uppdatera koden och lägga till den nya funktionen.
En sådan idé skulle till exempel kunna vara den idén som vi haft om hur man skulle kunna integrera en ryggsensor. Om ryggsensorn skulle utvecklas som ett tillbehör som kan kommunicera trådlöst via Bluetooth så skulle man helt enkelt kunna utveckla en ny version av applikationen som stödjer användning av denna sensor.

Referenslista

1. Arduino – Introduction http://www.arduino.cc/en/Guide/Introduction
2. Arduino - ArduinoBoardLilyPadhttp://arduino.cc/en/Main/ArduinoBoardLilyPad
3. Arduino – ArduinoBoardUno http://arduino.cc/en/Main/ArduinoBoardUno
4. doc8161 http://www.atmel.com/Images/doc8161.pdf
5. Wiring http://wiring.org.co/
6. Processing_org http://www.processing.org/
7. Bootloader – Wikipedia http://sv.wikipedia.org/wiki/Bootloader
8. Accelerometer – Wikipedia http://en.wikipedia.org/wiki/Accelerometer
9. ADXL335 http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf
10. Multiplexer – Wikipedia http://en.wikipedia.org/wiki/Multiplexer
11. Capacitor – Wikipedia http://en.wikipedia.org/wiki/Capacitor
12. Resistor – Wikipedia http://en.wikipedia.org/wiki/Resistor
13. Förstärkare – Wikipedia http://sv.wikipedia.org/wiki/F%C3%B6rst%C3%A4rkare
14. Distorsion (teleteknik) – Wikipedia http://sv.wikipedia.org/wiki/Distorsion_(teleteknik)
15. Fast Facts Bluetooth Technology Website http://www.bluetooth.com/Pages/Fast-Facts.aspx
16. Java (programming language) – Wikipedia http://en.wikipedia.org/wiki/Java_(programming_language)
17. What is Android - Android Developers http://developer.android.com/guide/basics/what-is-android.html
18. Android (operating system) – Wikipedia http://en.wikipedia.org/wiki/Android_(operating_system)
19. Graphical user interface – Wikipedia http://en.wikipedia.org/wiki/Graphical_user_interface
20. Global Positioning System – Wikipedia http://sv.wikipedia.org/wiki/Global_Positioning_System
21. MP3 – Wikipedia http://en.wikipedia.org/wiki/Mp3
22. The Open Source Definition - Open Source Initiative http://www.opensource.org/docs/osd
23. Öppen källkod – Wikipedia http://sv.wikipedia.org/wiki/%C3%96ppen_k%C3%A4llkod
24. Software license – Wikipedia http://en.wikipedia.org/wiki/Software_license
25. Apache License, Version 2_0 http://www.apache.org/licenses/LICENSE-2.0
26. Eclipse - The Eclipse Foundation http://www.eclipse.org/
27. Android SDK - Android Developers http://developer.android.com/sdk/index.html
28. Android SDK - Swedroid Wiki http://www.swedroid.se/wiki/index.php/Android_SDK
29. Arduino – Software http://arduino.cc/en/Main/Software
30. Tera Term http://ttssh2.sourceforge.jp/index.html.en
31. Total harmonic distortion – Wikipedia http://en.wikipedia.org/wiki/Total_harmonic_distortion
32. Total Harmonisk Distortion (THD) http://www.yamaha-hifi.com/characteristics.php?idlex=79&lang=w
33. Music Shield – Wiki http://seeedstudio.com/wiki/Music_Shield
34. vs1053 http://www.vlsi.fi/fileadmin/datasheets/vlsi/vs1053.pdf
35. Activities - Android Developers http://developer.android.com/guide/topics/fundamentals/activities.html
    Services - Android Developers http://developer.android.com/guide/topics/fundamentals/services.html