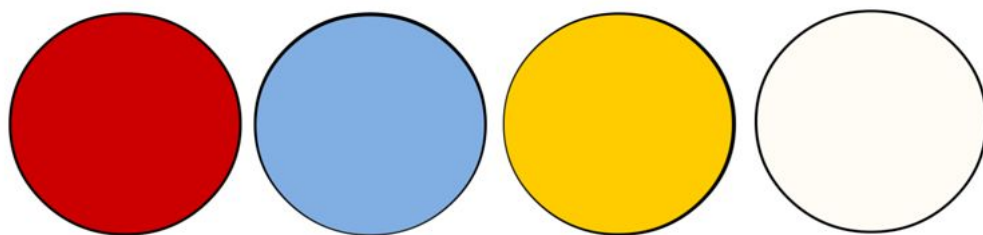


V E L F E R D S F O R E N I N G E N S

PODIO



TEKNISK RAPPORT

INF1510
Ellen Katrine Sveen
Anna Sofie Schei
Jostein Hellerud
Espen Wøien Olsen



Institutt for Informatikk
Universitetet i Oslo
VÅR 2016

Innholdsregister:

1 Introduksjon.....	3
1.1 Presentasjon av gruppen.....	3
1.2 Presentasjon av mål for prosjektet.....	3
2 Hva videoen viser.....	3
3 Lenke til video.....	4
4 Dokumentasjon av den tekniske løsningen.....	4
4.1 Helhetlig.....	4
4.2 Software.....	5
4.2.1 Kildekode.....	5
4.2.2 Liste over komponenter.....	6
4.2.3 Argumentasjon for bruk av software.....	7
4.2.4 Konklusjon og eventuelle løsninger knyttet til software.....	8
4.3 Hardware.....	8
4.3.1 Liste over komponenter.....	8
4.3.2 Argumentasjon for bruk av hardware.....	8
4.3.3 Konklusjon og eventuelle løsninger knyttet til hardware.....	10
5 Kilder og litteraturliste.....	11

1 Introduksjon

I denne rapporten vil vi presentere de tekniske detaljene i prosjektet samt videoen som viser frem problemområdet og vår løsning. I tillegg viser vi til kildekode som er brukt i selve prototypen. Vi vil argumentere for valg og bruk av både software og hardware.

1.1 Presentasjon av gruppen

Gruppen går under navnet Velferdsforeningen og består av Ellen Katrine Sveen, Anna Sofie Schei, Jostein Hellerud og Espen Wøien Olsen som alle studerer informatikk ved Universitetet i Oslo. Anna og Ellen studerer design, bruk og interaksjon på første året. Jostein og Espen studerer nanoelektronikk og robotikk på tredje året.

1.2 Presentasjon av mål for prosjektet

“Den eldre befolkningen har størst andel ikke-brukere av internett og størst andel med svak digital kompetanse. Sannsynligheten for nedsatt funksjonsevne øker også med alder. “ (Bufdir, 2014 - kilde finnes i hovedrapporten). Dette er analysen vi har jobbet etter gjennom hele semesteret. Målet vårt for prosjektet har dreid seg om å tilrettelegge for at eldre enklere kan ta i bruk radio/podkast ved hjelp av interaksjon uten skjerm.

2 Hva videoen viser

Filmen innledes med problemområdet vi har avdekket. Vi ser flere eldre som sliter med bruk av teknologier som radio og smarttelefon. Videre får vi en demonstrasjon av en pensjonists brukskontekst der både feedback på skjerm og via lyd er forvirrende, samt at knappene for små og uhåndterbare. Her ser vi i tillegg at brukeren er på farten, og ender opp med å måtte droppe å høre på nyhetene grunnet hastverk for å skynde seg på golfspilling. Videoen går i svart, og det sies: “Tenk om det fantes en mulighet for å få med seg det du ønsker å høre på på radio, selv når det kræsjer med golfspillingen?”

Løsningen Podio presenteres. En radio med podkastfunksjon, som lar deg høre på de radioprogrammene du selv ønsker, når du selv ønsker det. Videre vises en teknisk beskrivelse med oversikt over komponenter, samt en illustrasjon over hvordan disse arbeider sammen. Til sist er samme bruker som tidligere i filmen på plass i godstolen sin igjen - denne gangen har han Podio! Han øver litt spansk gjennom spanskurs-podden før han setter på siste nyhetssending etter ønske fra kona - akkurat når det passer dem. Velferdsforeningens logo runder av filmen.

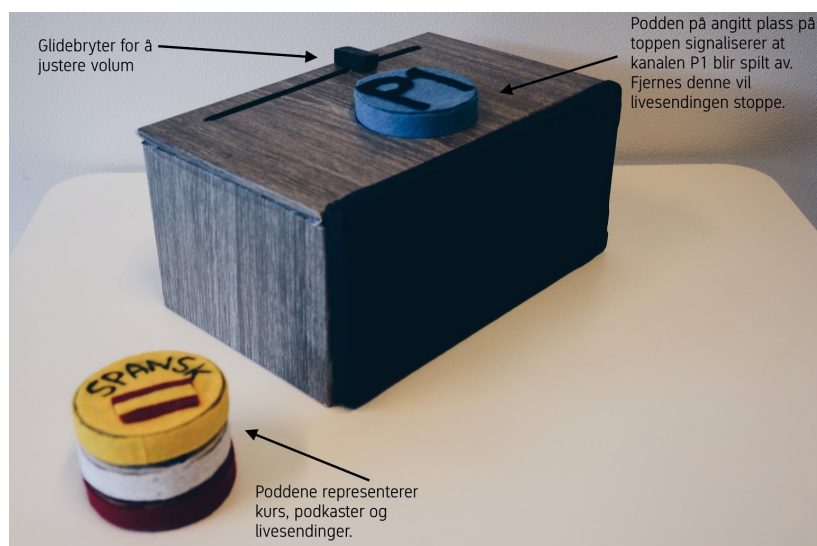
3 Lenke til video

<https://youtu.be/FtWz2O9eb-Q>

4 Dokumentasjon av den tekniske løsningen

4.1 Helhetlig

Podio består av en Arduino Uno, en Raspberry Pi model B, et USB-modem, en forsterker, et potensiometer, en RFID-leser, to transformatorer og to høyttalere. Arduino styrer systemet og leser av RFID-tagger ved hjelp av RFID-leseren. Taggen blir lest, prosessert og sendt til Raspberry pi'en om hvilken lydfile som skal spilles av. Nyeste episode av podkastene lastes ned automatisk og spilles av når brukerne er ferdig med forrige episode. Det spilles av ønsket radiokanal eller podkast, lyden sendes ut, forsterkes opp og nytes gjennom høyttalerne. Volumet kontrolleres gjennom et glider-potentiometer som sitter på toppen av podio.



4.2 Software

4.2.1 Kildekode

Vi viser her til kildekode. Etter instruks fra gruppelærer viser vi kun til github.

<https://github.com/skilbert/podio>

Forklaring til github-prosjektmappen:

Prosjektet inneholder flere mapper. Under “config” ligger filer som brukes når programmet kjører og bestemmer hvilke podkast og radiostasjoner vi ønsker.

I mappen “executables” ligger versjoner av programvaren som er eksportert som en jar-fil og klare til eksekvering. Her ligger også to shell-script som begge brukes for å få Raspberry Pi til å kjøre programmet når den starter opp. Mappen inneholder også en versjon av bibliotekene som er i bruk.

Kildekoden ligger under mappen “src” og er kommentert for letter å kunne sette seg inn i hva som er gjort. Her ligger alle java-filene og kildekode til Arduino i mappen ved navn “arduino”. I mappen “arduino” er det koden “rfidv2” som er i bruk og kjører på Arduinoen som er i bruk i prototypen. “Libs” er bibliotekene i bruk og “file” er tom på github, men inneholder vanligvis nedlastede podkaster.

I tillegg inneholder prosjektet en ekstra versjon av bibliotekene i bruk og en kjørbare jar versjon av prosjektet.

Forklaring til kildekode:

Javakodens main-metode ligger i klassen “Podio” og koden kjøres her i fra. Main-metoden oppretter et objekt av klassen Handler og overlater kjøringen av programmet til denne. Handler starter så kjøringen av programmet, oppretter kontakt med Arduino, starter og stopper tråder for nedlasting og avspilling av lyd, samt organiserer trådene under kjøring. For ytterligere detaljer til kildekode referer vi til github med tilhørende kommentarer under src/

Koden som kjører på Arduino starter med å sette opp serial kommunikasjon og kommunikasjon med RFID-shield. Den går så videre til loop hvor den lytter konstant etter forandringer på RFID-shield og kommanderer Raspberry Pi til handling ut i fra det den registrerer. For flere detaljer om koden henviser vi til den kommenterte koden på githuben under src/arduino/rfidv2.

4.2.2 Liste over komponenter

Software brukt i prosjektet:

Software	Beskrivelse
<i>Arduino</i>	<i>Arduino Software (IDE)</i>
<i>Raspberry Pi</i>	<i>Java-språk, shell-script + innebygde bibliotek</i>
<i>SketchUp</i>	<i>3D-modellering og design av prototype(r)</i>
<i>GoogleDocs</i>	<i>For utkast av rapport og generell fildeling gjennom hele prosjektet</i>
<i>GitHub</i>	<i>Fildeling og vertøy for samarbeid</i>
<i>Filmredigering</i>	<i>Kdenlive</i>

4.2.3 Argumentasjon for bruk av software

Arduino ble som forventet og ønsket en veldig sentral enhet for hele systemet. Den fungerer som en overordnet kontrollenhet for Podio. Siden vi tar i bruk RFID-teknologi, er dette mye data som må prosesseres. Hovedoppgaven er først og fremst å lese RFID-koden til radiobrikkene, prosessere disse til riktig form. Normalverdien til kodene er i det hexadesimale systemet, så de prosesseres til en håndgripelig kode. Dermed har vi satt opp en kommunikasjonsplattform med Arduino

som gjør at den enkelt kan lese av RFID-koden og kommunisere til en Raspberry Pi hvilken handling som ønskes. Arduino fungerer rett og slett som en master og sender kommandoer til Raspberry Pi via USB serial print.

Raspberry Pi fikk et innpass i prosjektet etter å ha forhørt oss om bruk og fått godkjent av gruppelærere. Denne fungerer som en slave og laster ned, samt spiller av podkaster og eller ønsket radiostasjon. Grunnen til at vi tok i bruk dette er rett og slett pga. muligheten til å bruke tråd-funksjoner i Java og lettere bruk av WiFi og lydutgang, samt hardware løsninger vi diskuterer under. Dette er meget nyttig ift. å kunne laste ned nye programmer samtidig som et annet blir spilt. Ved bruk av bare Arduino hadde vi fått problemer ved å spille av lyd, laste ned lydfiler og lytte etter forandringer på RFID-leseren på en gang. Dette løses altså ved at Arduino lytter på RFID-leser og styrer systemet, mens Raspberry Pi spiller av lyd og laster ned nye filer. Den tar imot de prosesserte kodene og kommando fra Arduino, som da peker på den aktuelle podkasten. Raspberry Pi blir dermed slaven i systemet.

4.2.4 Konklusjon og eventuelle løsninger knyttet til software

Løsningen som er valgt er hensiktsmessig i forhold til prosjektet. Vi føler at Arduino er et kraftig verktøy til enkel problemløsning og prototyping. Ved hjelp av programvaren til Arduino kan man enkelt laste opp og hurtig prøve ny programvare. Dette ble ofte benyttet i prosjektet. Videre finner vi at ved mer komplekse oppgaver oppleves funksjoner, støttet gjennom software, som utfordrende. For eksempel møter vi utfordringer der det er ønskelig med større operasjoner samtidig når flere, eller alle, prosesser krever totalen av prosessorens ressurser. Dette løser vi ved bruk av Raspberry Pi i prosjektet. Softwareløsningen fungerer godt og distribuerer arbeidsoppgavene i systemet til enheten som løser oppgaven best.

4.3 Hardware

4.3.1 Liste over komponenter

Hardware brukt i prosjektet:

Hardware	Beskrivelse
<i>Høytalerelementer</i>	<i>Montert ut av et høytalieranlegg</i>
<i>Potentiometer</i>	<i>Kontroll av lydstyrke</i>
<i>Sparkfun Evaluation Shield</i>	<i>Leser RFID-tags</i>
<i>RFID-tagger</i>	<i>Enheter som leses av RFID-shield</i>
<i>Arduino Uno</i>	<i>Mikrokontroller</i>
<i>USB modem</i>	<i>WiFi-funksjonalitet for Raspberry Pi</i>
<i>Raspberry Pi Model B</i>	<i>Mikrokontroller</i>
<i>Forsterkerkort</i>	<i>Driver høytalerelementene i stereo</i>
<i>Transformatorer x 2</i>	<i>Nødvendig for strømforsyning til komponentene</i>
<i>Treverk og skruer</i>	<i>Materiale for konstruksjon av enhet</i>

4.3.2 Argumentasjon for bruk av hardware

Mikrokontrollere er de absolutt mest sentrale enhetene i dette prosjektet. Med tanke på kobling av ledninger, er Arduino Uno en veldig god enhet. Den er godt støttet med ADC (Analog to Digital converter) og DAC (Digital to analog converter), og vi blir fritatt for mange potensielle problemer ved bruk av RFID-leser.

Raspberry Pi får en annen funksjon enn Arduino i vårt prosjekt og brukes på grunn av dens prosessor, støtte for mer avanserte funksjoner og innebygde hardware

komponenter. Pien har mange hardware komponenter innebygd som ville blitt et problem å få plass til på Arduino. Ved bruk av et USB-modem og den innebygde 3.5 mm minijack lydutgangen får vi mye funksjonalitet i en liten enhet. Hadde vi ønsket å løse alle disse med Arduino hadde vi måttet bruke WiFi-shield og Mp3-shield i tillegg til RFID-shield som vi allerede har i bruk. I og med at shield som regel bruker mange av de samme portene kunne dette ført til bruk av opp til tre Arduino enheter, hvilket er noe vi ikke hadde hatt plass til i enheten utformet.

Raspberry Pi og Arduino er koblet sammen med USB-kabel for å minimere potensielle feil ved sammenkobling og lodding av signalkabler. I tillegg er høyttalersystemet koblet direkte inn i Raspberry Pi med 3.5 mm mini-jack for å unngå unødvendig kompleksitet.

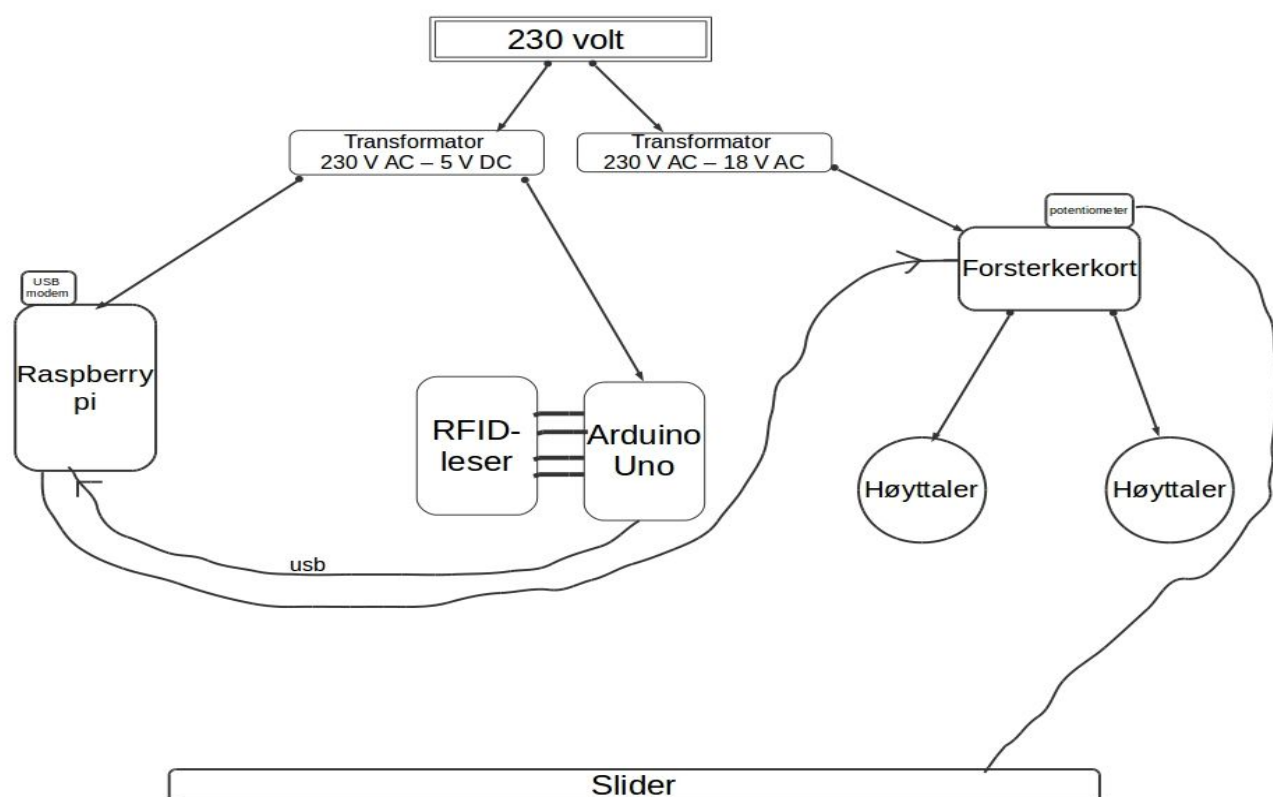
Sparkfun Evaluation Shield var en meget enkel komponent å ta i bruk med tanke på hardwareløsninger. Her var ressurser og koblingsmetoder godt støttet på Sparkfun sine nettsider¹. For å kunne plassere leseren der vi ønsker, uten å måtte være avhengige av at Arduinoen fikk plass på samme sted loddet vi kabler til leseren og dro disse til Arduino. I stedet for at de er plagget rett i hverandre.

Høyttalersystemet baserer seg på 18 V AC (vekselstrøm) til forsterkerkortet og får dette levert av en transformator. Målgruppen presenterte et tydelig ønske om god og høy lyd. Høyttalerelementene ble derfor valgt fordi de kunne levere dette og ble valgt til fordel for elementer på mindre størrelse for å best mulig møte behovene til brukergruppen. For best mulig lyd kunne det vært ønskelig med større høyttalerelementer, men dette ble et trade-off i forhold til størrelsen vi ønsket at Podio skulle være. Dette er også tilfellet for forsterkerkortet som både måtte levere ønskede kvaliteter ved lyden, samt være på en størrelse vi fikk plass til.

Mikrokontrollerne baserer seg på 5 V DC (likestrøm), så disse er koblet til en hjemmelaget transformator som omformer 230 V AC fra stikkontakt til den nødvendige likestrømmen.

¹ <https://www.sparkfun.com/products/10406>

Potentiometer viste seg å gi flere problemer. Ettersom det var ønskelig fra brukergruppen å ha en glidebryter som volumkontroll, ble vi nødt til å lage en egen komponent, da vi ikke fant glidebrytere av ønsket størrelse. En snusboks, brukt som et tannhjul, er koblet til potentiometer. Til denne har vi koblet på en tråd som gjør at tannhjulet kan snurre i to retninger. Denne tråden er så koblet til selve glidebryteren, som gir funksjonen som var ønskelig.



Illustrasjon 1: Oversikt over hardware komponenter og hvordan disse er koblet sammen

4.3.3 Konklusjon og eventuelle løsninger knyttet til hardware

Hardwareløsningene i prosjektet er preget av et fokus på funksjon. Først og fremst kom det veldig klart frem at burde blitt bestilt komponenter tidligere i prosjektløpet. Det er selvfølgelig vanskelig å planlegge dette siden vi har basert prototypen på intervjuer med brukergruppene, og vi ikke hadde et klart bilde av hvordan denne

skulle se ut. Det burde blitt bestilt med en gang vi hadde en idé om hva vi skulle gjøre. I tillegg er det enkelte problemer vi kunne forutsett. Volumkontrollen er et godt eksempel på dette, og vi kunne spesialbestilt et glider-potensiometer som er av den størrelsen det var behov for i stedet for løsningen vi har nå som er litt MacGyver-aktig - men det fungerer!

I tillegg kunne vi definitivt brukt 3D-printer for modellering og konstruksjon av selve prototypen. Men da vi skjønnte at dette ville krevd flere runder med tilpassing og mål av de forskjellige komponentene sparte vi mye tid og ressurser ved å ta i bruk treverk som vi konstruerte selv. Samtidig var dette materialet foretrukket av brukergruppen.

Videre gjør både Arduino og Raspberry Pi jobben som er ønskelig. Vi bruker fordelene til de to enhetene og prototypen fungerer stabilt, spiller høy og klar lyd.

5 Kilder og litteraturliste

API hentet fra NRK

<https://www.nrk.no/podkast/>

<https://www.nrk.no/mp3aac/>

Diverse bibliotek og ressurser

- Java Media Framework

<http://www.oracle.com/technetwork/java/javase/download-142937.html>

- Java Media Framework MP3

<http://www.oracle.com/technetwork/java/javase/download-137625.html>

- USB Communication RXTX

<http://rxtx.qbang.org/wiki/index.php/Download>

- Sparkfun ressurser til RFID-leser(fotnote s.9)

<https://www.sparkfun.com/products/10406><https://www.sparkfun.com/products/10406>