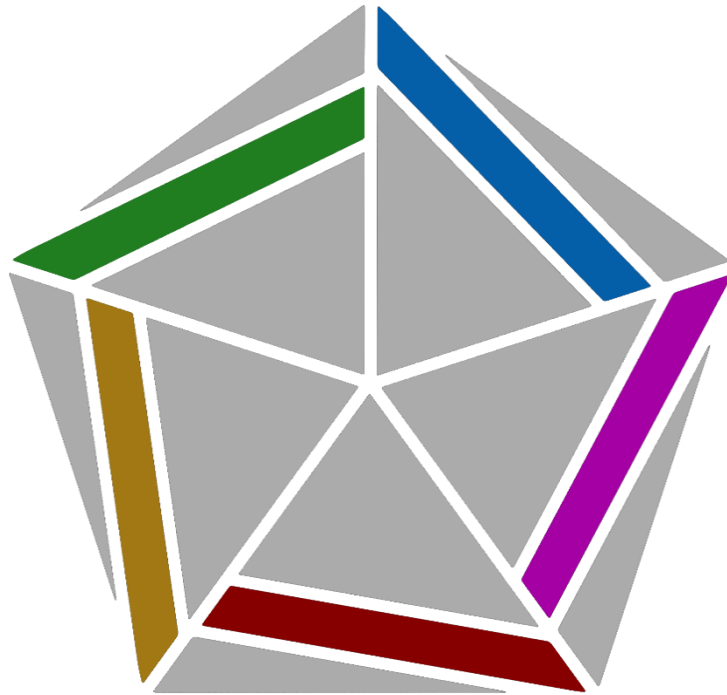


IN1060 Bruksorientert design

**TEKNISK RAPPORT**



Pentagon

Chris Kløv Andersen | Steffen Ekeberg Bråten | Steven Hoang Giang Nguyen

Thu Thuan Thi Vo | Trần Văn Ngọc Tân

Universitetet i Oslo

Institutt for Informatikk

Vår 2018

## Innholdsfortegnelse

<b>1. INTRODUKSJON .....</b>	<b>3</b>
<b>2. MÅL .....</b>	<b>3</b>
<b>3. VIDEO .....</b>	<b>3</b>
<b>4. TEKNISK SPESIFIKASJON .....</b>	<b>4</b>
4.1.    TEKNISK GRUNNLAG .....	4
4.2.    PROTOTYPE .....	4
4.3.    BYGGING.....	5
4.4.    UTFORDRINGER.....	6
4.5.    KOMPONENTER .....	6
4.6.    STRØMKILDE .....	6
4.7.    KRETSE OG SKJEMATISKE DIAGRAMMER.....	7
<b>5. KODER.....</b>	<b>9</b>
5.1.    ARDUINO 1 .....	9
5.2.    BEHANDLETE SIGNALER.....	9
5.3.    ARDUINO 2.....	10
<b>6. BIBLIOTEK.....</b>	<b>14</b>

## 1. INTRODUKSJON

I denne rapporten vil vi presentere de tekniske detaljene ved prototypen vår. Vi vil gå inn på de ulike komponentene som blir brukt og koden prototypen består av. Vi skal vise til prosjektvideoen vi har laget og en beskrivelse av videoen. Videre vil vi redegjøre valgene vi har tatt gjennom prosjektet med tanke på tekniske løsninger, og hva slags utfordringer vi har støttet på og hva vi har gjort eller ikke gjort med disse utfordringene.

## 2. MÅL

I dette prosjektet har vi jobbet med å forenkle og forbedre interaksjonen mellom eldre og hverdagslig teknologi. Dette er et problemområde vi ønsker å løse. I løpet av prosessen har vi hatt en hovedbruker og tolv andre innenfor målgruppen som vi har samarbeidet med. Hovedbrukeren elsker å høre på musikk, men har en CD-spiller som hun ikke greier å kommunisere med. Målet for prosjektet vårt er da å løse dette problemet.

Gjennom prosjektet har vi kommet på mange forskjellige ideer og varianter av flere løsninger, men gjennom undersøkelser og analyser har vi endt opp med CD COVER CONTROL (CCC), som er en type fjernkontroll. Den er formet som en CD-cover for å passe inn i brukskonteksten. Den har større og færre knapper (kun de nødvendige) for å tydeliggjøre knappenes funksjon og forminske forvirring. Håndgripen interaksjon er også implementert inn i løsningen, for å beholde og øve opp motorikken til eldre samtidig som det er intuitivt å gjøre disse bevegelsene. Den har selvforklarende ord istedenfor symboler på knappene. På denne måten kan CCC bidra med å forenkle og forbedre interaksjonen mellom eldre og hverdagslig teknologi som en CD-spiller.

## 3. VIDEO

Vi vil gjøre oppmerksom på at i videoen vår er det blitt brukt Wizard of Oz for å kunne vise interaksjonen mellom CCC og en potensell bruker. Dette er gjort fordi CCC kun fungerer på hovedbrukeren vår sin Onkyo CS-265 (CD-spilleren). Grunnen til dette er for at vi har måtte dekodet signalene som kommer fra brukerens originale fjernkontroll for å få signalene som sendes ut av fjernkontrollen. Vi kan, om spurt, dekode signalene fra en annen fjernkontroll slik at den hadde fungert på den fjernkontrollens CD-spiller. En funksjon som ikke ble vist på videoen, på grunn av tidsbegrensningen, som vi har implementert inn i CCC er hvordan CCC sender ut signaler når lydknappene eller sangbytteknappene holdes nede. Dette er beskrevet nøye i som kommentarer i koden.

Videoen på YouTube: [https://www.youtube.com/watch?v=gHnRS\\_58xFE](https://www.youtube.com/watch?v=gHnRS_58xFE)

Vi har video-dokumentasjon på at systemet fungerer i bruk, men siden vi ikke får linket til denne videoen får vi dessverre ikke inkludert den. Dersom dette er ønsket kan vi få sendt dette.

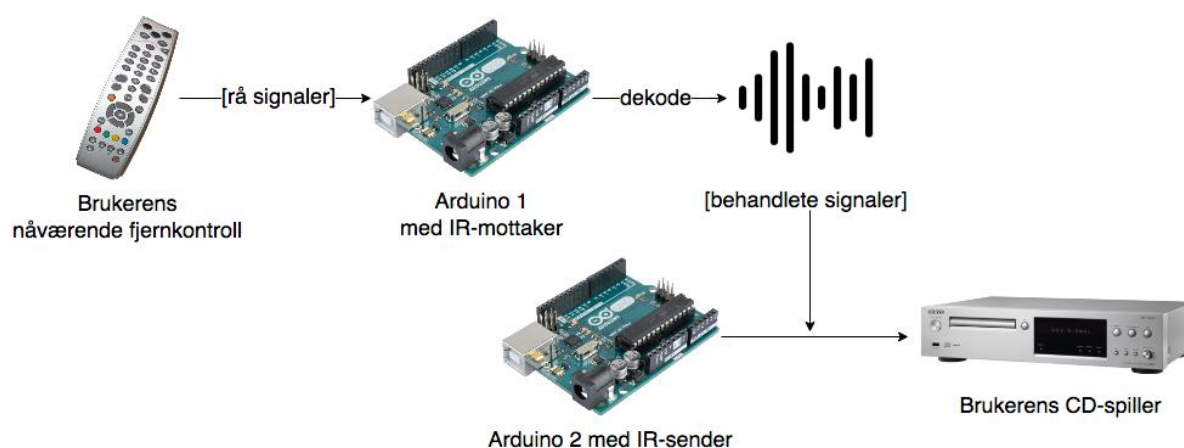
## 4. TEKNISK SPESIFIKASJON

### 4.1. Teknisk grunnlag

IR-fjernkontroller, som navnet tilsier, benytter pulser av infrarødt lys for å sende signaler til en mottaksenhet, for eksempel en TV eller en CD-spiller. Hver knapp på fjernkontrollen sender ut et unikt mønster av pulser som dekoderes av mottakeren, slik at riktig funksjon (dvs. øke volumet, skru av enheten) kan utføres.

### 4.2. Prototype

Den endelige prototypen består av én Arduino, men gjennom prosessen har vi brukt to Arduino. Grunnen til dette er fordi vi brukte en Arduino (Arduino 1) med en IR-mottaker som tar imot og dekode signalene fra brukerens nåværende fjernkontroll slik at vi kan ha data over signalene. Den andre Arduino (Arduino 2) med en IR-sender vil dermed sende videre de behandlede signalene til brukerens CD-spiller.



Når coveret løftes opp blir magneten som er limt på kanten av "coveret" dratt vekk fra Reed-switch som sitter under panelet. Dette indikerer at Arduino sender signalen til at CD-spilleren kan slå på og starte musikken. Samme måte kan man sette musikken på pause, alt brukeren

trenger å gjøre er å lukke coveret slik at Reed-switchen registrerer at magneten kommer nærmere og dermed sende pause signal til CD-spilleren. For å unngå at systemet er på hele tiden uten å være i bruk har vi gjort det slik at en såkalt nedtellingsvariabel starter når reedswitch registrerer at magneten kommer nærmere igjen. Denne nedtellingen starter på 20 minutter, og når disse 20 minuttene er over sender Arduionen slå av signal til CD-spilleren.

Det er fire forskjellige knapper på panelet til kontrolleren: høyere volum, lavere volum, neste sang og forrige sang. Ut fra observasjoner har vi sett at eldre har en tendens til å holde knapper nede lengre enn nødvendig og dermed sende flere enn et signal. Når det gjelder lydstyrke knappene er dette kjekt, for da slipper brukeren å trykke flere ganger for å øke/senke lyden mye. Når det gjelder sangbytte-knappen derimot kan dette være et problem når cd-spilleren hopper over flere sanger når brukeren egentlig bare vil bytte til neste sang. Derfor har vi implementert sangbytte-knappene slik at det kun sendes et signal uansett hvor lenge du holder den nede. For å hoppe over flere sanger må da brukeren trykke ned knappen flere ganger.

### 4.3. Bygging og utforming

Til bygging av prototypene og spesielt den endelige prototypen hadde vi flere muligheter og flere måter vi kunne gjennomføre det på. Vi kunne blant annet bruke laserkutter, det fant også blant annet nettsider på nettet som kan kutte treet på forhåndsbasert målinger. Det ble ingen av de som nevnt over, vi valgte heller å lage alt for hånd, dette er fordi håndkuttete plater gir prototypen et mer tradisjonell utseende og følelse som vil stå i stil med brukskontekst og brukernes personlige preferanser for stil og utforming.

Vi har valgt tre som materiale fordi hovedbrukeren vår var svært samfunnsengasjert og ikke ønsket å bidra til mer forurensing av hav ved bruk av plast, som motiv til coveret ønsket hun et bildet av hennes favoritoperasanger, Maria Callas. Prototypen er også hvit fordi det passer mer inn i kontekst – hun ønsket hvit for at den skulle stå i stil med bordet og musikkanlegget hennes.

For å gjøre prototypen brannsikker har vi brukt elektrisk teip rundt der vi har loddet sammen. Vi har også lagt over flere lag med elektrisk teipen på bunnen av prototypen slik at arudionen ikke får direkte kontakt med treet.

#### 4.4. utfordringer

I utviklingsprosessen møttet vi noen vanskeligheter blant annet hvordan oppdage coverets bevegelse. Den opprinnelige løsningen med lyssensor ble erstattet av Reed-switch. Funksjonaliteten om å automatisk skru av CD-spilleren etter en viss tid er ikke lett å implementere heller. Disse er beskrevet mer detaljert i del 5 KODER.

#### 4.5. Komponenter

Navn	Antall	Beskrivelse
Arduino UNO	1	
USB-A til USB-B kabel	1	3 m
AC-DC adapter	1	Input: 100 ÷ 240 V, 50/60 Hz. Output: 5 V, 1 A
Piezo	1	Piezo-høytaler
IR-mottaker	1	Universell IR-mottaker for vanlige fjernkontroller. Type: 1838. 2,7-5,5 V. 10-pk.
IR-dioder (sender)	1	IR-diode med klar linse: 1,5-1,6 V, 60 mA (120 mA maks.), 15-30° åpningsvinkel.
Reed-switch	1	14 mm. Magnetisk tungelementkontakt (Reed-switch). Sluttes ved magnetisk påvirkning (NO).
Sylindrisk magnet	1	Sylindrisk magnet for magnetkontakter og andre tungelementkontakter (f.eks reed-switch). Mål: Ø 5 x 16 mm.
Arkadeknapp 33 mm	4	Fjærende knapp (NO) i gjennomsiktig plast, til f.eks. Arkadespill. Gullbelagte kontakter. Stikker opp 6,5 mm fra monteringshullet (Ø 30 mm). Mål: Ø 33 x 31, 5 mm.
Motstander, 100 Ω	1	Koblet til piezo-høytaleren
Motstander, 22 Ω	1	Koblet til IR-dioderen
Motstandere, 10 kΩ	4	Koblet til arkadeknappene

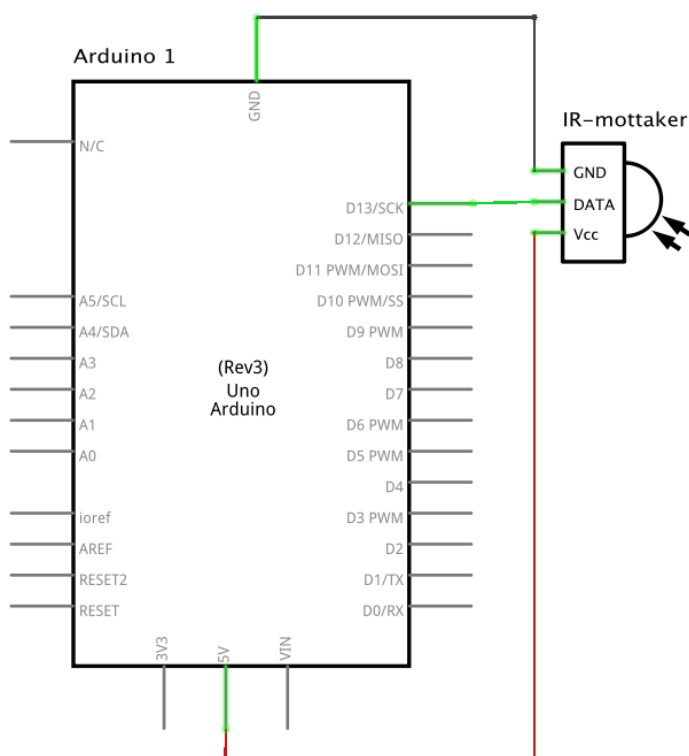
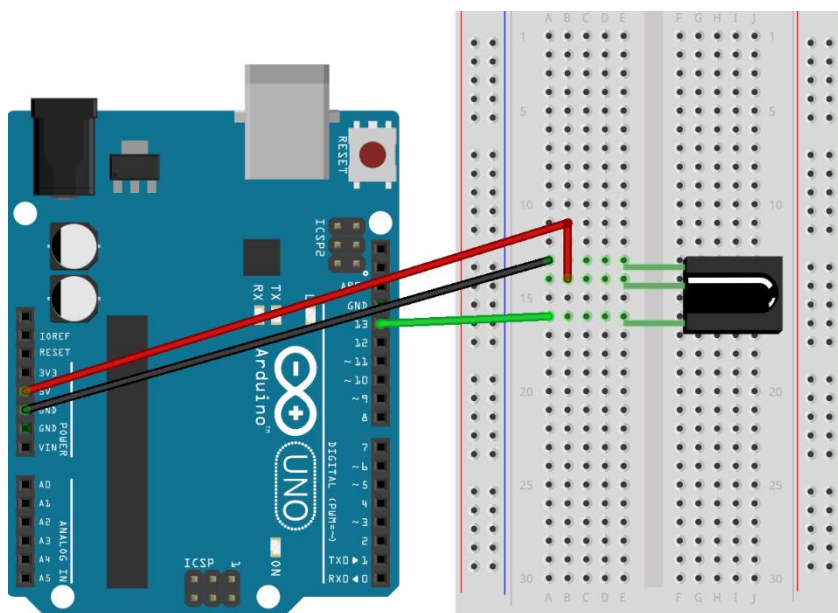
#### 4.6. Strømkilde

Det er mange måter å forsyne strøm til en Arduino blant annet med et batteri som er den enkleste måten. Nedsiden ved batteri er at det kan bli vanskelig for brukeren å skifte batteriet

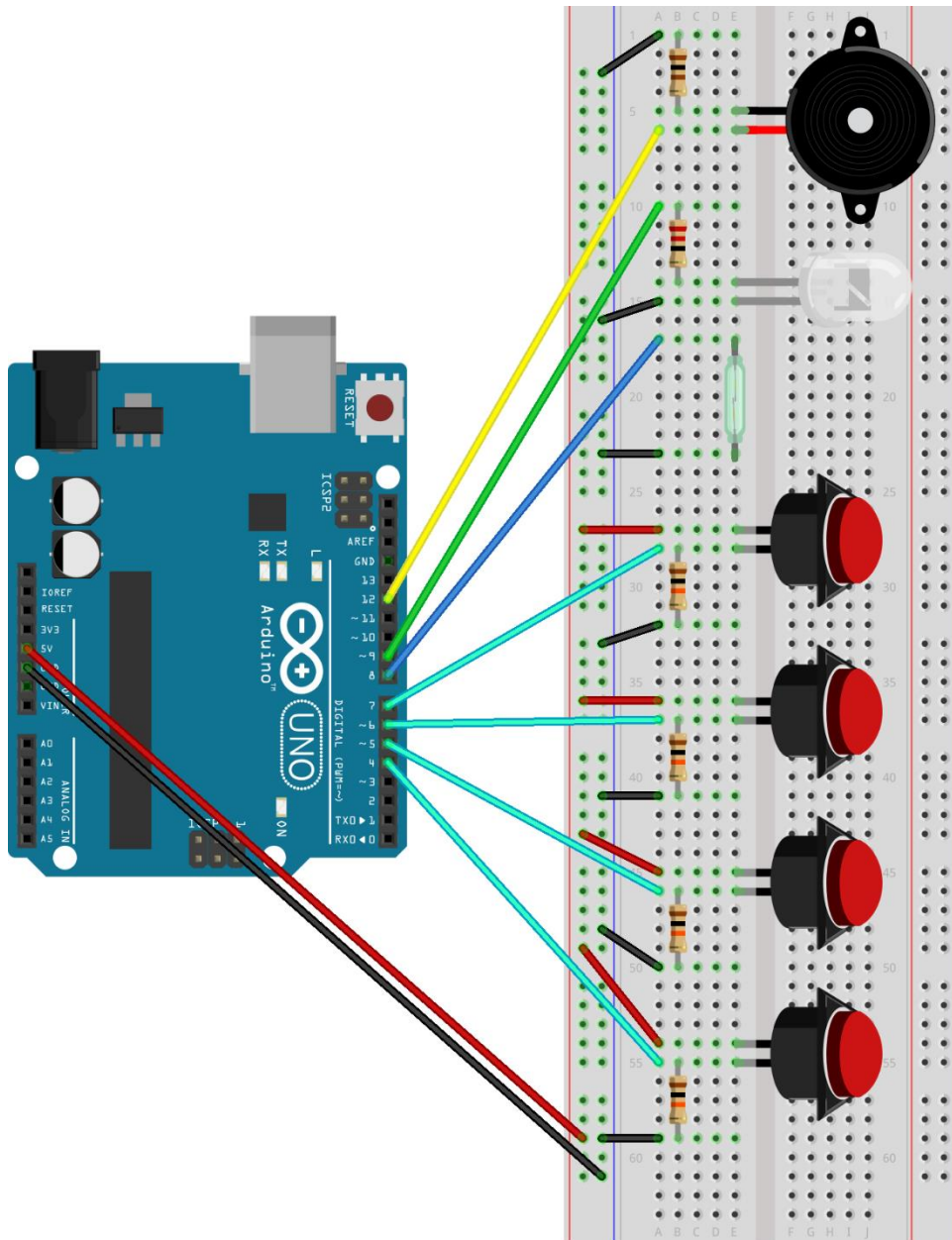
når det har gått tomt. I tillegg vil det være risikabelt at man må åpne esken for å bytte batteri, med tanke på at sensitive komponenter kan bli påvirket. Derfor bestemte vi oss for å bruke en kabel som kobler Arduinoens USB-B-port til en AC-DC-adapters USB-A-port. Kabelen har lengden på 3 meter slik at CCC kan stå fritt ganske langt unna strømkilden. Siden CCC uansett skal være stasjonær har det ikke så mye å si om det brukes kabel eller ikke.

#### 4.7. Kretser og skjematiske diagrammer

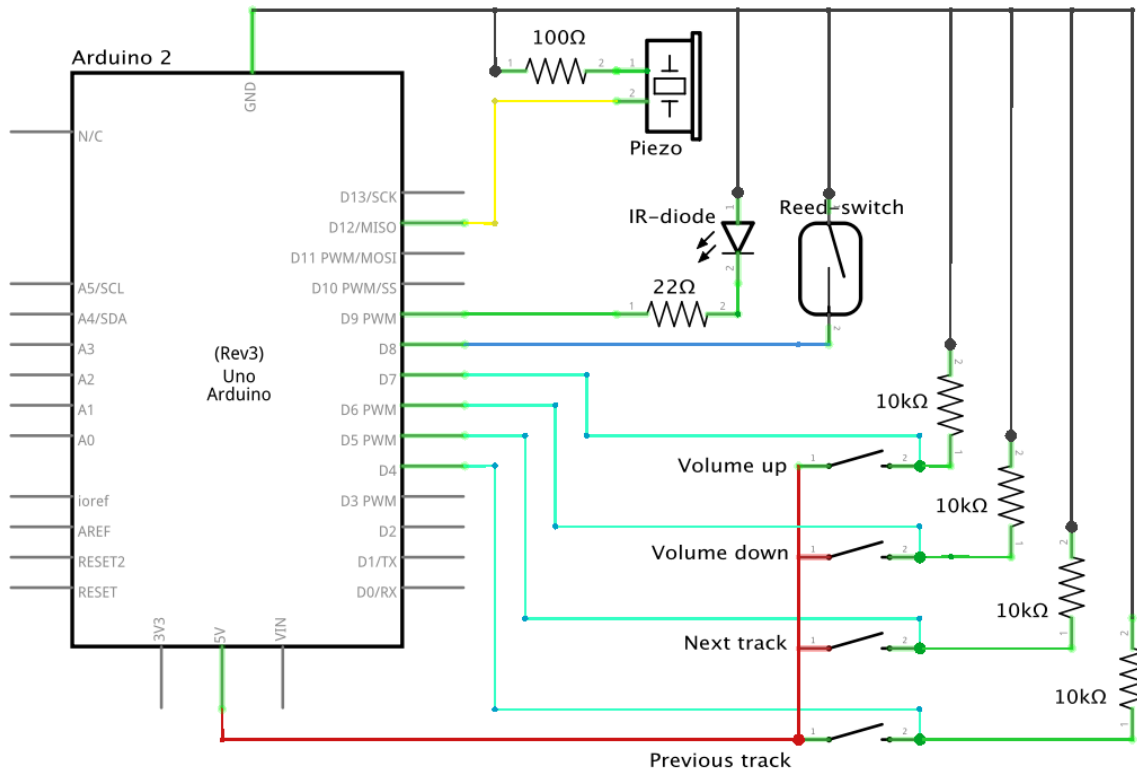
Arduino 1:



## Arduino 2:







## 5. KODER

### 5.1. Arduino 1

Koden som ble lastet ned på Arduino 1 er den opprinnelige koden av eksemplet `IRrecvDumpV2` inkludert i biblioteket `IRremote` (jf. del 6. BIBLIOTEK).

Koden på Github: [\url{https://goo.gl/pC3NsF}](https://goo.gl/pC3NsF)

### 5.2. Behandlete signaler

Ved å bruke Arduino 1 til å dekode rå signaler fra brukerens nåværende fjernkontroll, fikk vi en tabell med data tilsvarende seks funksjoner vi ønsket å gjenspeile: `høyere volum`, `lavere volum`,  `neste sang`, `forrige sang`, `spill av/stoppe musikk` og `skru av/på`. Følgende er det for eksempel dataen til funksjonen `neste sang`:

```
\begin{lstlisting}
```

```
Encoding : NEC
```

```
Code : 4BC0B847 (32 bits)
```

```
Timing[67]:
```

```
+8850, -4500 + 550, - 600 + 500, -1700 + 550, - 600
+ 500, - 600 + 550, -1700 + 500, - 600 + 550, -1700
+ 500, -1750 + 500, -1750 + 500, -1700 + 550, - 600
+ 550, - 550 + 500, - 600 + 550, - 600 + 500, - 600
+ 550, - 600 + 500, -1700 + 550, - 600 + 550, -1650
+ 550, -1700 + 550, -1700 + 500, - 600 + 550, - 600
+ 500, - 600 + 550, - 550 + 550, -1700 + 550, - 600
+ 500, - 600 + 500, - 600 + 550, -1700 + 550, -1700
+ 550, -1700 + 500
```

```
unsigned int rawData[67] = {8850,4500, 550,600, 500,1700, 550,600,
500,600, 550,1700, 500,600, 550,1700, 500,1750, 500,1750, 500,1700,
550,600, 550,550, 500,600, 550,600, 500,600, 550,600, 500,1700,
550,600, 550,1650, 550,1700, 550,1700, 500,600, 550,600, 500,600,
550,550, 550,1700, 550,600, 500,600, 500,600, 550,1700, 550,1700,
550,1700, 500}; // NEC 4BC0B847
```

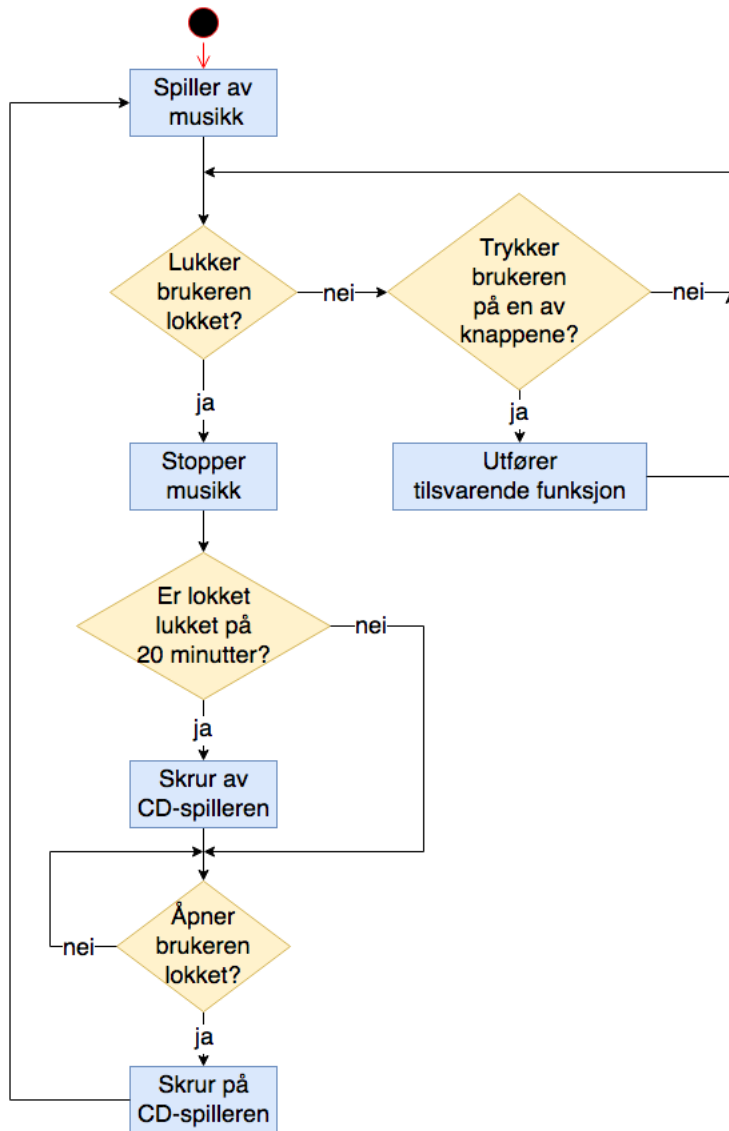
```
unsigned int data = 0x4BC0B847;
```

```
\end{lstlisting}
```

Hele tabellen: [\url{https://goo.gl/5V3pJB}](https://goo.gl/5V3pJB)

### 5.3. Arduino 2

Funksjonaliteten til CCC kan beskrives av følgende kodeflyten:



Fire funksjoner `\textit{høyere volum}`, `\textit{lavere volum}`, `\textit{neste sang}` og `\textit{forrige sang}`, blir styret av arkadeknapper. Hver gang en knapp trykkes, lager CCC en kort lyd, samt skriver en linje til serial-overvåker. \par

\begin{lstlisting}

```

if (digitalRead(nextTrack)) {
  irsend.sendNEC(0x4BC0B847, 32);
  Serial.println("Next track");
  simpleSound();
  delay(debounceTime);
}

```

```
\end{lstlisting}
```

Her er `{irsend}` et objekt av klassen `{IRsend}` som blir brukt av biblioteket til å sende verdier ved hjelp av metoden `{sendNEC()}`. Verdien `{0x4BC0B847}` er nettopp den vi fikk av Arduino 1:\par

```
\begin{lstlisting}
```

```
Encoding : NEC
unsigned int data = 0x4BC0B847;
```

```
\end{lstlisting}
```

Det å spille av/stoppe musikk når brukeren åpner/lukker lokket er litt annerledes. Før hadde vi tenkt å bruke en lyssensor (photocell) til å oppdage coverets bevegelse. Men da traff vi noe problem: hvis rommet er mørkt vil ikke bruker kunne få spilt av musikk. Gjennom mange forsøk innså vi også at analoge signaler som blir sendt av lyssensor ikke er veldig stabile. CCC risikerer derfor å miste funksjonaliteten. Harald Maartmann-Moe forslo å benytte i stedet en Reed-switch med magnet. Komponentens operasjonsprinsipp er veldig enkel. Når brukeren åpner coveret til CCC vil magneten bli adskilt fra Reed-switchen som skal returnere verdien `{false}` og vice versa. Signalene som blir sendt er dermed digitale, helt eksplisitte. Nå trenger vi bare å skrive ned verdier i starten og slutten av hver `{loop()}-løkke:\par`

```
\begin{lstlisting}
```

```
void loop() {
    proximity = digitalRead(reed);
    //...
    previousProximity = digitalRead(reed);
    delay(200);
}
```

```
\end{lstlisting}
```

og sammenligne dem:\par

```
\begin{lstlisting}
```

```
if (proximity != previousProximity) {  
    //...  
    irsend.sendNEC(0x4B5822DD, 32);  
    simpleSound();  
    //...  
    delay(debounceTime);  
}
```

```
\end{lstlisting}
```

Brukeren har en tendens til å trykke ned og holde lenge på knappene for å sikre at systemet registrerer hennes kommandoer. Dette er fint når det gjelder justering av lydstyrke. Det sendes altså kontinuerlig med signaler så lenge volumknappen er holdt nede.

```
if (digitalRead(volumeUp)) {  
    irsend.sendNEC(0x4BC040BF, 32);  
    Serial.println("Volume up");  
    simpleSound();  
    delay(debounceTime);  
}
```

Når det gjelder sangbytte-knappene har vi brukt debounce-handling for å unngå at CCC hopper over flere sanger når brukeren trykker en gang. Med andre ord vil kommandoene bli registrert kun én gang uansett hvor lenge hun holder knappen nede.

```
if (digitalRead(nextTrack) && digitalRead(nextTrack) !=  
previousNextTrack) {  
    irsend.sendNEC(0x4BC0B847, 32);  
    Serial.println("Next track");  
    simpleSound();  
    delay(debounceTime);  
}
```

```
}
```

Når det gjelder funksjonaliteten om å automatisk skru av CD-spilleren etter en viss tid, blir det enda mer komplisert. Da bruker vi `millis()` til å registrere tidspunkt når brukeren lukker coveret, sammen med en rekke logiske beregninger for å implementere funksjonen.

```
\begin{lstlisting}
```

```
if ((millis()-pauseMillis) > (1000 * turnOffTime) && (millis()-
pauseMillis) < (1000 * turnOffTime + 250) && pauseMillis != 0 &&
!proximity && !previousProximity) {

    irsend.sendNEC(0x4B20D32C, 32);

    Serial.println("Turn off");

    powerSound();

    isOff = true;

    delay(debounceTime);
```

```
\end{lstlisting}
```

Koden på Github: [\url{https://goo.gl/gLMNDk}](https://goo.gl/gLMNDk)

## 6. BIBLIOTEK

Prototypen anvender biblioteket **IRremote** av Ken Shirriff.

I utgangspunktet var det umulig å få piezo fungere med biblioteket. Dette skyldes konflikt mellom biblioteket og metoden `tone()` (som brukes for å lage lyd med piezo) fordi de bruker samme *timer*. Vi tar i bruk en tidligere versjon (1.0.0) istedenfor den siste (2.2.3) og endrer filen *IRremoteInt.h* sine kodelinjer angående *timer* slik:

```
\begin{lstlisting}
```

```
#define IR_USE_TIMER1 // tx = pin 9
//#define IR_USE_TIMER2 // tx = pin 3
```

```
\end{lstlisting}
```

Biblioteket, versjon 2.2.3: [\url{https://github.com/z3t0/Arduino-IRremote}](https://github.com/z3t0/Arduino-IRremote). Versjon 1.0.0 får vi ved å installere i programvaren.

Den endrete \textit{IRremoteInt.h}: \url{https://goo.gl/tUpxNL}\par