

# Teknisk rapport for prototype

## «BOWMASTER»

designet av Team Trøkk



## **Innholdsfortegnelse**

|                                 |          |
|---------------------------------|----------|
| <b>1. Prosjektets mål</b>       | side 2.  |
| <b>2. Videoens innhold</b>      | side 2.  |
| <b>2.1 Videoens innhold</b>     | side 2.  |
| <b>2.2 Link til videoen</b>     | side 2.  |
| <b>3. Teknisk dokumentasjon</b> | side 3.  |
| <b>3.1 Hardware</b>             | side 4.  |
| <b>3.2 Software</b>             | side 7.  |
| <b>3.3 Veien videre</b>         | side 10. |
| <b>4 Referanser</b>             | side 11. |
| <b>4.1 Spesielle takk</b>       | side 11. |
| <b>4.2 Referanse</b>            | side 11. |

## **1. Prosjektets mål**

Målet for prosjektet er å lage et apparat som bidrar til å utfordre konsentrasjonen til bueskytteren. Grunnen til dette er for at bueskyttere skal kunne bidra til å forbedre konsentrasjonen under mesterskap der det er mange distraherende elementer.

For å oppnå dette har vi et vibrasjonsarmbånd for å simulere utmattelse i arm. Lysarmbånd for å simulere eventuelle lys som kan forstyrre, samt det kan hjelpe med å trene mot bevegelses distraksjoner.

I tillegg til dette har vi en lydenhet som skal spille av lyder som er distraherende for skytteren.

Satt sammen gir dette en rekke muligheter for treneren å distrahere skytteren med de problemområdene han/hun eventuelt har.

## **2. Videoen**

### **2.1 Videoens Innhold**

Videoen viser løsningen i en simulert bruksituasjon. Boye stilte opp som skytter og ble filmet flere ganger mens han skøt pil med bue. Denne videoen ble klippet og lagt til voice over.

Ettersom det ikke kom så godt frem funksjonaliteten under video med skyting så har vi lagt et innslag der det fremvises at den faktisk fungerer.

Den viser også bruksituasjon slik den er tenkt, men på grunn av apparatets natur viste det seg at en voiceover var mest gunstig.

Vi viste ikke frem det audioative aspektene med prototypen ettersom disse viste seg å ikke være tilstrekkelige. Ytterlige viste det seg at lysene på prototypen er alt for svake og ikke er tilstrekkelige til å distrahere skytteren.

### **2.2 Link til videoen**

<https://www.youtube.com/watch?v=G9OkpOAdc3Y>

### 3. Den tekniske løsningen.

#### 3.2 Hardware:

Vi begynte med et veldig utssende basert design, dette var avhengig av at trenerne kunne stille opp til et workshop for en deling av ideer. Dette skjedde imidlertid ikke, noe som medførte at vi fokuserte på et mer funksjonsdrevet design, heller enn at vi baserer funksjonen på utseende. Ettersom vi har en erfaren bueskytter i teamet (Boye Mathias Molteberg) innså vi at hele systemet kan til en del gjemmes bort i utstyret til bueskytteren. Derfor baserer det seg på to armbånd og en sentral kontroll boks som ligger i kogeret<sup>1</sup> til skytteren.

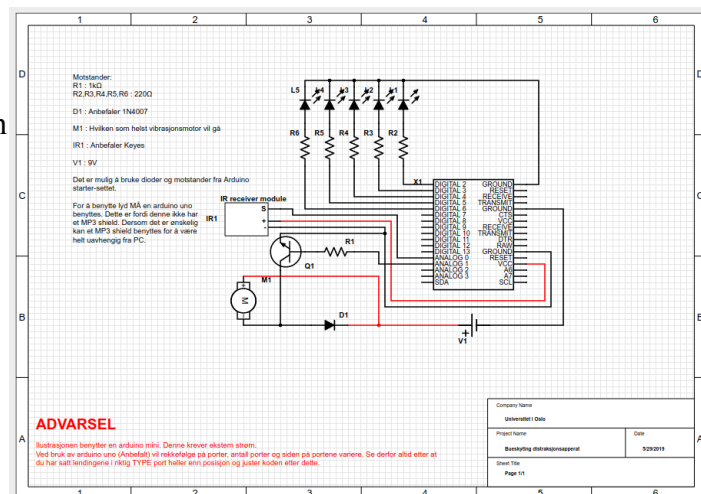
Ettersom vi av logistiske grunner ikke endte opp med å anskaffe et MP3 shield så endte vi opp med følgende oppsett (Figur 1). For vibrasjonsmotoren ble arduino brukt som av og på, dessverre gir det ikke nok kraft til å få en betydelig vibrasjon. Derfor koblet ble et 9V batteri koblet opp for å øke amperen til en tilstrekkelig mengde (1).

Ved å benytte et breadboard kan dette oppnås helt uten å måtte lodde. Dette var et viktig faktum for oss da vi ønsker å gi noe tilbake til bueskytterne som har hjulpet oss så veldig med dette prosjektet.

Under utviklingen var hvordan enheten skulle brukes et stort spørsmål. Vi ønsket gjerne en trådløs løsning noe som er mindre invasivt for skytteren. Dessverre på grunn av logistikk, økonomi og gjennomførbarhet endte vi opp med en enklere løsning.

Denne løsningen kan godt visualiseres som en trådløs løsning ettersom dette er det logiske steget videre.

Dersom vi skal fortsette med dette prosjektet så ville vi satt inn et MP3 shield for trådløs lydavspilling, mulighet for bluetooth ørepropper og trådløse armbånd. Dette er ting vi ikke fikk gjort på grunn av tidligere nevnte



Figur 1: Figuren viser bildet ment for at bueskyttere skal kunne sette opp sitt eget system. Her har vi også listet opp alle mulige verdier for motstander, dioder og strømkilde. Samt anbefalinger for IR mottaker og hva motoren er for.

<sup>1</sup> Et pilkoger er en en pilholder som er festet på et utstyrsbelte under skyting. Denne benyttes for å ha lett tilgang til nye piler og relevant utstyr imens man skyter.

årsaker, men som ville vært muliggjort med mer tid. Derfor ble det benyttet processing for å spille av lyd via datamaskin. Denne datamaskinen blir derfor også strømforsyning, selv om dette ikke er nødvendig.

Det skal nevnes at vibrasjonsbåndene nesten endte opp med å ikke bli realisert. Boye brukte nesten en måned på å få de, samt koden for disse, til å fungere. Dette til tross at en av de gikk i stykker i løpet av uken etter.

Vi ønsket at bueskytterne skal tilpasse oppsettet sitt. Dette er noe som mange bueskyttere tar veldig seriøst. Derfor er det viktig å ha mulighet for at skytterne kobler inn og ut armbåndene slik de selv ønsker.

For å oppnå dette brukte vi audio jack for å koble opp vibrasjonsmotor, og en 9-pins VGA kabel for å koble opp LED pærene. Denne hadde hver separate LED kontrollert av Arduino, med mulighet for totalt 8 LED pærer i koblingen. For vibrasjonsmotorene bruker vi en 3.5mm MONO jack, ettersom denne har kun + og GRND på ledningen. Derfor benytter vi en slags AND gate for å støtte opp mot et 9V batteri som aktiveres av en analog port på arduino.

LED pærene hadde problemer med å kobles opp mot arduino i serie. Derfor benyttet vi oss av en VGA som kan koble opp i parallell gjennom kabelen.

Når vi har muligheten til å koble ut armbåndene så gjør dette at skytteren har mulighet til å sette boksen på enten belte, koger eller et annet sted den ikke er i veien for han/hun. Dette er veldig viktig for en bueskytter og vi mener dette bidrar til støtte opp under funksjonen av systemet på en god måte.

Komponentene ble satt inn i en boks, rekonstruert av en gammel spikersorteringsboks. Denne har fordelen av å være gjennomsiktig noe som bidrar til at det er lett å få godt signal for IR avleseren til fjernkontrollen.

I selve boksen ble skilleveggene fjernet. Dette var for å gi rom for de individuelle komponentene. Deretter ble tre hull konstruert. En av disse var for arduinoledningen som fungerer som høyttalerkontakt, ettersom vi ikke fikk anskaffet MP3 shield. En annen var for en aux kabel og det siste for en VGA 9-pin åpning (Figur 2). For å gjøre dette ble det hovedsakelig brukt fil og en elektrisk graveringsmaskin for å lage åpningene for boksen. Disse ble slipt til nøyaktig mål, slik at de måtte presses inn med en del kraft.



*Figur 2: Boksen uten skillevegger, presentert på arbeidsbenken. Bildet er tatt etter hullene er laget. Disse måtte forstørres etter bildet ble tatt.*

Deretter ble bunnen og veggene dekket av hvit EL-teip av PVC. Denne tapen blir normalt brukt for isolering av ledninger, men i dette tilfellet ble den benyttet for å beskytte komponentene fra skarpe kanter etter fjerningen av skilleveggene(Figur 3).

Etter dette var gjort så ble utkanten dekket av et lag mørkt (sort) vevteip. Dette fjerner det transparente egenskapene til boksen samtidig som det gir to ulike fordeler.

Hovedsakelig er det en fordel at boksen ikke er like mottakelig for skraper og sprekker, da vevet og limet i tapen hjelper med å holde denne sammen. Dette er en fordel da det oppstod en håndfull sprekker under fjerningen av skilleveggene.

I tillegg til dette blir boksen som nevnt dekket av et opakt lag. Et produkt med synlig elektronikk kan i noen tilfeller oppfattes som mindre ferdig og mindre politlig enn et som elektronikken er gjemt. Derfor vil det for brukeren sannsynligvis være en fordel at den dekket (figur 4).

Farge og utseende på boksen er ikke av høy viktighet ettersom den skal ligge i en lomme på kogeret (pilholderen). Disse har den fordel at de ha store lommer til scorekort, men de er som oftest tomme ettersom store lommer er moten innenfor idretten heller enn basert i noe mer praktisk. Derfor er fargen endt på sort slik at den ikke fremstår like skitten som den kan hvis vi hadde brukt eksempelvis hvit.



*Figur 3: Her har boksen fått et beskyttende underlag av hvit EL-teip.*



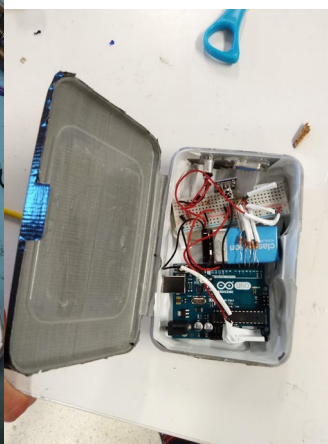
*Figur 4: Boksen med dekkete overflater*

Etter å ha dekket den i sort teip, ble Aux inngangen og VGA inngangen satt inn (figur 4 & figur 5). Disse er begge av typen hunn. For aux er det ingen forskjell dersom det er ønskelig å benytte han istedenfor en hunn. Derimot for VGA må den settes inn fra andre siden hvis det er en han heller enn en hunn. Disse ble satt på plass og deretter limt med varmt lim (figur 5, under).

Som tidligere nevnt satt disse godt uten lim, derimot ved å lime disse er vi sikre på at den ikke lar seg falle ut. Dette er veldig viktig siden vi skal presse inn kontakter, dette kan være nok trykk til å flytte på porten.



*Figur 5: Her har en 3.5 mm jack (AUX) hunn samt en VGA hunn presset inn i hullene på boksen.*



*Figur 6: Her sees innsiden av beholderen etter at alle delene har blitt koblet sammen og limt på plass.*



*Figur 7: Endelig utseende før brukertesting.*

Etter at portene hadde blitt satt på plass så ble breadboard og batteriet satt inn. Deretter ble alt sammen koblet sammen.

For å få plass til breadboard ble dette delt i to og limt på plass inne i arduinoen. Dette viste seg imidlertid å være en dårlig ide siden koblingen til 3.5mm jack kontakten var hakket for liten. Dette førte til at breadbordet måtte limes enda en gang og legges halvveis under VGA kontakten og står derfor skjevt i boksen. (figur 6, over)

Etter at alle delene hadde blitt festet sammen ble det også funnet et problem med å få avlesninger fra fjernkontrollen. Dette viste seg etter litt feilsøking å stamme fra tapen på utsiden. Tapen viste seg til å være opak nok til at den blokkerte IR sendingene fra fjernkontrollen.

Etter å ha kuttet ut en liten skjerm til IR signalet (figur 7, over) så kunne fjernkontrollen registreres fra en halvmeter unna med 4 lag med bomull mellom boksen og fjernkontrollen. På dette punktet er det all grunn til å tro at den vil motta signal gjennom lommen på et buekogger.

Dersom vi skal forbedre designet på apparatet så vil stegene fremover være å skaffe et MP3 shield samt en fjernkontroll basert på bluetooth heller enn IR. Dette gir mer brukbarhet og mer rekkevidde enn det nåværende utseende.

### 3.2 Software:

For softwaren begynte vi med to separate program. Et for arduino og et for processing. Tanken var at disse kunne i ettertid kombineres for at man kan oppnå lyd uten MP3 shield.

Først startet vi med Arduinokretsen. Denne ble modellert inn i flere deler. Hovedsakelig siden lys armbåndet har flere lys (fem) og disse er koblet direkte inn i arduinoen. Dette gir oss flere muligheter. To av disse lysene er hvite og brukes derfor til en blitsfunksjon. De resterende lysende (rød, gul, grønn) skal fungere som startlys. Dette kombineres med en timer som sjekkes hver gang opp mot en variabel. Måten funksjonen aktiveres på er ikke gjennom kall, men rettere endring av denne variabelene.

Derfor endte vi opp med følgende funksjon for lys.

```
void lys(int x){  
  digitalWrite(lys1,x);  
  digitalWrite(lys2,x);  
}
```

Denne lyser kun to hvite dioder. De siste tre blir brukt til å styre skytelys.

```
void skytelys(){  
  if (SKYTESTART + 200000 > millis()){  
    digitalWrite(grontLys, HIGH);  
    digitalWrite(gultLys, LOW);  
    digitalWrite(rodLys, LOW);  
  }  
  else if (SKYTESTART + 240000 > millis()){  
    digitalWrite(grontLys, LOW);  
    digitalWrite(gultLys, HIGH);  
    digitalWrite(rodLys, LOW);  
  }  
  else if (SKYTESTART + 300000 > millis()){  
    digitalWrite(grontLys, LOW);  
    digitalWrite(gultLys, LOW);  
    digitalWrite(rodLys, HIGH);  
  }  
  else{  
    digitalWrite(grontLys, LOW);  
    digitalWrite(gultLys, LOW);  
    digitalWrite(rodLys, LOW);  
  }  
}
```



```
} }
```

Dette ble satt inn i et modifisert eksempel fra IR biblioteket til kontrolleren(2).

Dette fører til en del endringer i utformingen. Vi fant også ut at motoren på dette punktet har kun av og på. Inntil dette punktet har det blitt brukt en funksjon for å skru den av og på.

Dette ble kobinert med en lydavspillingsmal(3) via Serial. Dette viste seg imidlertid å bli svært vanskelig å gjennomføre siden det påkrevde å sette seg inn i Processing.

Etter hjelp fra Jakob så muliggjorde det seg å benytte Prossesising med Minmin. For å spille av lyder så snappet opp de signaler fra Serial og koblet dette opp med ulike lydfiler. For å oppnå mulighet for å spille av flere lyder ble flere ulike AudioPlayer objekter benyttet.

Disse ble kontrollert med signalene av serial som styrte en .play() .pause(). .rewind() sekvens.

For å forhindre at disse signalene ble blandet med signaler fra andre prosesser så startet vi signalet med "PAS", kort for Play Audio Sound, etterfulgt av et tall.

```
if(myString.equals("PAS0")){                               if(myString.equals("PAS1")){
    if(beep.isPlaying() == false){                          if(klikk.isPlaying() == false){
        beep.play();                                        klikk.play();
        delay(1500);                                       delay(1500);
        beep.pause();                                       klikk.pause();
        beep.rewind();                                       klikk.rewind();
    }                                                         }
}                                                         }
```

Dette resulterte i følgende kode på arduino siden.

```
void lyd(int x){
    String stringAaSende = "PAS" + String(x);
    Serial.println(stringAaSende);
    delay(1000);
}
```

For at dette skulle funke med lyd på en god måte endte vi opp med skytesekvens endret til:

```

void skytelys(int piler){
  lyd(0);
  lyd(0);
  int long tid = millis() + 10000;
  while(tid > millis()){
    digitalWrite(rodLys, HIGH);
  }
  tid = millis() + piler * 30000 - piler * 5000;
  lyd(0);
  while(tid > millis()){
    digitalWrite(grontLys, HIGH);
    digitalWrite(gultLys, LOW);
    digitalWrite(rodLys, LOW);
  }
  tid = millis() + piler * 5000;
  int rist = millis ();

  while(tid > millis()){
    digitalWrite(grontLys, LOW);
    digitalWrite(gultLys, HIGH);
    digitalWrite(rodLys, LOW);
    if (millis() > rist){
      rist = millis() + 500;
      SHAKE(300, 255);
    }
  }
  lyd(0);
  lyd(0);
  lyd(0);
  digitalWrite(grontLys, LOW);
  digitalWrite(gultLys, LOW);
  digitalWrite(rodLys, HIGH);
  delay(1000);
  digitalWrite(grontLys, LOW);
  digitalWrite(gultLys, LOW);
  digitalWrite(rodLys, LOW);
}

```

Her er kan det sees at der den forrige enheten var fastsatt til kun 6 piler, så åpner denne for en muligheten for mer fleksibilitet.

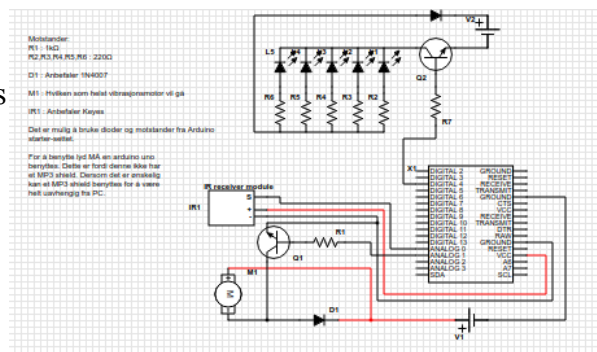
Alle tidligere kode deler ble satt sammen i Arduino kode og Processing kode. (4,5,6)

### 3.3 Veien videre

Etter brukertesting viser det seg at det må introduseres større strøm inn i systemet for å få tilstrekkelig effekt av lysdiodene. Ytterligere vil det lønne seg å kvitte seg med hele Processing ettersom lyd ga liten til ingen effekt.

Lys armbåndet er derfor ganske unødig som et hjelpemiddel. Ettersom den foreløpig har for lite strøm. Derfor er det bedre å fokusere på en vertikal prototype videre.

Denne vil da kobles noe lignende slik vist på figur 8.



Figur 8: Kobling av lysblits med Av/på funksjon med ca 10V effekt

## Spesiell Takk til:

Alaa Sami for å intro til Circuit.io som var ganske god hjelp for å finne ut koblingen av en 9V batteri på en krets.

Jakob Kongrud for hjelp med generelle spørsmål og Processing.

## Referanser

1. Circuto.io , <https://www.circuito.io/app?components=8449,10218,11021> , lest 04/6/2019.
2. IR kontroller, <https://www.kjell.com/no/produkter/elektro-og-verktoy/utviklerkit/arduino/moduler/luxorparts-ir-mottaker-og-fjernkontroll-p87895>, Bibliotek under mer informasjon , lest 20/5/2019.
3. Mal for lydavspilling med Processing,  
<https://gist.github.com/timpulver/5ba4a29cddd543b4a900>
4. Repository for all kode brenyttet i prosjektet, <https://github.com/boyemm/IN1060>, lest 11/6/2019.
5. Endelig kode for arduino,  
[https://github.com/boyemm/IN1060/blob/master/Satt\\_sammen\\_med\\_lyd.ino](https://github.com/boyemm/IN1060/blob/master/Satt_sammen_med_lyd.ino) , lest 11/6/2019
6. Endelig kode for processing,  
[https://github.com/boyemm/IN1060/blob/master/LydAvspilling\\_pde.pde](https://github.com/boyemm/IN1060/blob/master/LydAvspilling_pde.pde) , lest 11/6/2019.