

# Pseudonym presenterer



## TEKNISK RAPPORT

### **Medlemmer:**

*Camilla Christensen, Emma Tvinnereim,*

*Hedda Dyngeland og Maren Berge*

# Innholdsfortegnelse

<b>1. Introduksjon</b>	<b>2</b>
1.1 Hvem er vi?	2
<b>2. Mål med dette prosjektet</b>	<b>3</b>
2.1 Krav til prototypen	3
2.2 Teoretisk konsept bak lyslenken	3
<b>3. Presentasjon av video</b>	<b>4</b>
3.1 Målet med videoen	4
3.2 Link til video	4
<b>4. Teknikken bak løsningen</b>	<b>4</b>
4.1 Komponenter vi har brukt	4
4.2 Illustrasjon av kretsen	7
4.3 Valg som har ble tatt	7
4.3 Feiltesting og løsninger	9
<b>5. Presentasjon av koden</b>	<b>9</b>
5.1 Initiering av variabler	10
5.2 Metoder	10

# 1. Introduksjon

I denne rapporten blir den tekniske løsningen til den høyopløselige prototypen presentert. Det blir lagt fokus på bruken av de forskjellige komponentene inkludert i løsningen, og hvordan de fungerer sammen. Her blir det diskutert om målene satt før implementasjonen faktisk blir oppfylt. Fokuset i denne rapporten er hva som er blitt gjort og hvordan. Det legges også mye fokus på hvorfor.

Vi vil også inkludere valg og alternative retninger vi måtte ta som viker fra den originale planen. Aller først presenteres forfatterne av dette dokumentet:

## 1.1 Hvem er vi?



Gruppen vår består av Camilla Christensen, Emma Tvinnereim, Hedda Kjetun Dyngeland og Maren Ingunnsdatter Berge, og sammen er vi Pseudonym. Vi valgte gruppenavn ut ifra noe som reflekterte av målgruppen var, forfattere.

Prosjektet vårt har fått navnet Girograf. Dette ordet er bygget opp av to ord: giro og graf. Her har vi tatt betydningen til giro for repetitive handlinger, og graf for visualisering av utvikling.

For oss heter prosjektet dette, men vi har latt selve prototypen vårt få et navn etter brukerens smak/ønske, som da ble Rolf.



## 2. Mål med dette prosjektet

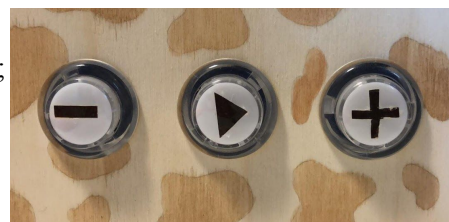
Målet vårt for dette prosjektet går ut på å finne, og lage en løsning som svarer på problemstillingen vår: *”Kan vi gi én barnebokforfatter motivasjon, via visualisering av tid til å gjennomføre dagsplaner?”*. Dette tilsvarer at vi finner en løsning som oppfyller behovene som dukket opp under datainnsamlingen, og etter analysen. For oss er det å skape en uforstyrrende samarbeidspartner for forfatteren, fordi vi har vært fokusert på kontekst hele veien. Dette blir det ikke fokus på i denne rapporten, men blir videre forklart i prosjektrapporten.

### 2.1 Krav til prototypen

- Den skal ha mulighet til å legge og trekke fra timer.
- Bruker skal kunne starte timeren etter valgte timer.
- Visuell visning av hvor mye tid som har gått, vise helheten i en arbeidsøkt.
- Tydelig varsling etter hver time med arbeid.
- Mulighet til å sette timeren i pause og deretter begynne igjen ved endt tidspunkt.

Etter utformingen og implementering av prototypen har vi oppfylt kravene over. Dette inkluderer at det er tre knapper som har funksjonene; *legge til timer, trekke fra timer og starte tidtakeren*.

Den midterste knappen, illustrert i bilde til høyre starter tidtakeren, og bruker antall registrerte timer til å lyse opp lyslenken basert på dette.



Det blir gitt en tydelig feedback til bruker for hver time som går mens Girografan er aktiv, i form av lyd. Lysene i nakken er ikke synkronisert med lyden, men de brukes for å vise den helhetlige oversikten over arbeidsøkten.

Det ble også implementert en funksjon som oppfylte kravet om at bruker skal kunne legge prototypen i pause, at dette påvirket hovedtidtakingen.

### 2.2 Teoretisk konsept bak lyslenken

Ut ifra funnene i dagboken, har SuperForfatter stor variasjon i hvor mye hun kan jobbe totalt på en dag. Fra loggen hun førte var det ikke vanlig for henne å jobbe mer enn 10 timer, og Girografan fikk derfor begrenset til 10 timer.

Videre ble det implementert at alle lysene skal lyses opp gjennom økten, selv om eksempelvis bare 4 timer blir valgt å jobbe for dagen, skal dette ikke resultere i opplysning av 4 lys men hele nakken over 4 timer. Dette ble gjort etter valg gjort av bruker.

Dette ble løst med et regnestykke med 6-gangen, fordi vi hadde 10 lys som skulle tilsvare 10 timer og det er 60 min i en time. Eksempelvis SuperForfatter velger én time for dagen, vil være 6 min mellom hvert lys. Altså ett lys per 6. min. Videre med to timer blir ett lys hvert 12 min, og tre timer blir ett lys hvert 18 min, osv.

## 3. Presentasjon av video

### 3.1 Målet med videoen

Målet med videoen vi har laget er å vise Girograf sine funksjoner, hvilket problem den er ment til å løse og hvordan den gjør dette i praksis. Dette ønsker vi å oppnå ved å vise et scenario av brukerens arbeidsdag først uten prototypen og deretter med bruk i konteksten den er ment. Ved å få med dette fremstilles problemet - en forfatter som ikke har motivasjon og struktur nok til å gjennomføre dagsplanene hun setter seg, og løsningen - en artefakt som visualiserer tid og gir feedback i form av lys og lyd.

*NB! Vær oppmerksom på at stemmen er ikke en del av Girografens originale funksjoner og egenskaper. Den ble lagt til for å redegjøre for sine funksjoner på egen hånd, i tillegg til å spille på humoren som går igjen i videoen.*

### 3.2 Link til video

[Her er linken til videoen!](#)

## 4. Teknikken bak løsningen

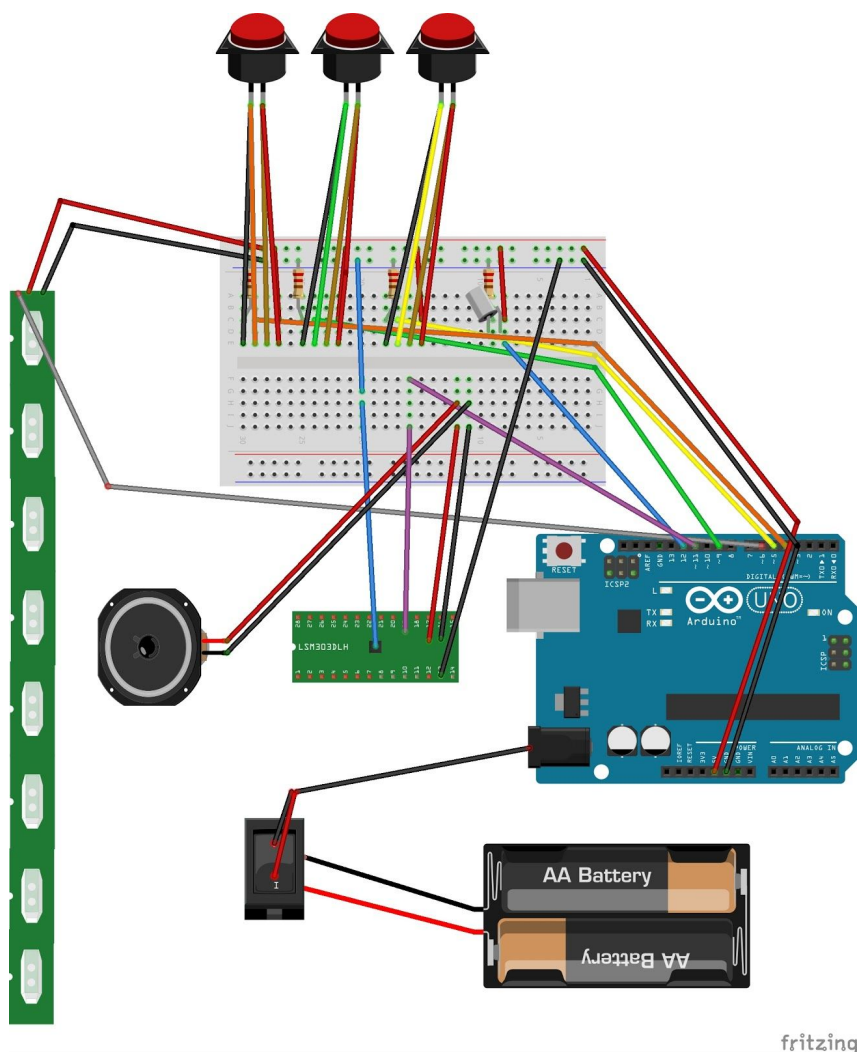
### 4.1 Komponenter vi har brukt

Komponenter	Antall	Bilde
Fra Arduino Starterpakke		<i>Bildene tatt fra skisseprogrammet Fritzing</i>

Arduino uno	1	
Breadboard	1	
Tiltsensor	1	 Bilde: <a href="http://www.varesano.net/files/imagecache/500width/tilt_sensor.jpg">http://www.varesano.net/files/imagecache/500width/tilt_sensor.jpg</a>
Resistorer 220 $\Omega$	4	
Fra Kjell og Company		<i>Bildene tatt fra netthandelen til Kjell og Company</i>
Strømbryter, 1-polet Av/på: 250 V / 6 A	1	
Luxorparts Koblingskabel som kan deles, 40-polet Hann-hann - 40 stk i en pakke	2	
Fra Teknikkmagasinet		<i>Bilde tatt fra netthandelen til Teknikkmagasinet</i>

Lydeffektboks - inkluderer høyttaler	1	  <p>(Bilde til høyre er det vi brukte - høyttaleren er kuttet ut i en sirkel)</p>
Fra Digital Impuls		<i>Bildene tatt fra netthandelen til Digital Impuls</i>
Arcade Button med LED	3	
Strømadapter 220V -> 9V	1	
Fra Verksted i Masterbygg		
Neopixel lyslenke, krever RGB-fargekoder	10 lys	

## 4.2 Illustrasjon av kretsen



## 4.3 Valg som har ble tatt

### Strømforsyning

Vi vurderte flere løsninger for å sende strøm til Arduino, men endte opp med å hente strøm fra stikkontakt i veggen. På illustrasjonen er det bilder av batterier som illustrerer denne strømkilden.

Strømtilførselen kontrolleres vha. en strømbryter som er loddet til strømforsyningen, og åpner og lukker kretsen.

### Kretskort med lydeffekter

I utgangspunktet falt valget på å bruke mp3-shield, men etter tilbakemelding fra gruppelærer kunne vi bruke en lyd-effektboks vi hadde kjøpt.

Den grønne figuren (sentrert i illustrasjonen over) representerer kretskortet. I lydboxen ble lyden spilt av når kretsen ble lukket ved hjelp





av strømførende materiale i knappene. Vi prøvde å løse dette ved uten bruk av de medfølgende knappene, ved å først lodde en ledning til hver side av kretsen til den ønskede lyden, slik at kretsen kunne bli lukket på vårt signal.

Etter litt testing viste seg at det kun var nødvendig å føre strøm på den ene siden, og dette ble løsningen vår. Høytalerne får strøm fra via et signal når kretsene lukkes og dette spiller av lyden.

En konsekvens ved at vi benyttet oss av dette kretskortet var at vi kun fikk implementert en lyd, og ikke fire som var brukerens ønske. Dette skyldes at kretskortet var av dårlig kvalitet og klarte ikke å stå imot varmen fra loddingen. Dette resulterte i at kortet begynte å smelte, så da det var en stor risiko for total ødeleggelse, prioriterte vi den ene lyden vi hadde fått til å fungere.

### Neopixel

Vi ble fortalt at man burde ha en ekstra strømkilde for å drive hele lenken med Neopixelsene, men fordi vi kun skulle bruke 10 lys kunne vi drive den med strømmen fra Arduino.

For å dekke kravet om å visualisere hvor mye tid som har passert ble det tatt et valg at lys var det mest visuelle virkemiddelet. Det endelige valget falt på Neopixel da det er mulig å kontrollere med Arduino. Ved bruk av indekser til å tenne lys, ville vår kunnskap om programmering være tilstrekkelig.

### Knapper

For å ha mulighet til å inkludere behovet for å velge antall timer, valgte vi å ta utgangspunkt i knapper. Her var det viktig at det var mulighet for å legge til timer, fjerne timer og starte tidtakingen, og ut i fra dette falt valget på en pluss- minus- og en startknapp.

Ved at bruker interagerer med knappene skulle dette bestemme hvilken intervall lysene skulle lyse, slik at det fungerte som en progressbar i sin helhet. Knappene fungerte derfor som en overgang for brukerens ønsker i form av knappetrykk.

### Tilt

Bruken av tilt ble tatt etter veiledning fra gruppelærere, da dette var en god komponent for å registrere om prototypen var lagt til siden eller ikke. Vi undersøkte andre muligheter, men det var tilt som var det beste alternativet.

### Ferdig()

Et valgt som ble gjort uten innspill fra SuperForfatteren, er slutteffekten. Når hun er ferdig med alle de satte timene, vil halsen lyse opp i form av alle regnbuens farger. Dette ble gjort for å gi bruker tilbakemelding om at prototypen har strøm ved begynnelsen, og som et tillegg når bruker har blitt ferdig med arbeidsøkt.

## settGrønt()

Denne metoden setter alle lysene i lenken til grønt i en kort periode, og dette ble gjort etter brukers ønske om å motta feedback på at start-knappen ble trykket.

## **4.3 Feiltesting og løsninger**

### Lyset

Etter gjennomført evaluering med bruker, ble det oppdaget noen problemer med tidtakingen for lys da det skulle telles med timer. Dette skyldes at vi kun hadde testet med sekunder og minutter under tidligere tester. Løsningen ble at variablene for tid (se kode) gradvis ble økt fra sekunder til minutter til timer under kontrollert testing av tidtaking, slik at det kunne kartlegges hvor problemet lå. Det viste seg at en tidligere brukt variabel var begrenset til en int verdi, og måtte byttes til unsigned long for å takle store tall.

### Lyden

Fram til dette punktet var ikke implementasjonen av lyden styrt av hovedprogrammet, kun av menneskelig manipulering eller et selvstendig kodeprogram. Dette var et av programmets største problemer, men ved hjelp av gruppelærer fikk vi testet alle funksjoner som hadde innvirkning på lyden, og problemet fant vi med tilstanden til port 11 som styrer lyden. Ved å endre tilstanden fra daværende HIGH til nåværende LOW, ville lyden fungere på signal fra Arduino.

### Pause

Det siste kravet for prototypen ble satt til å være implementasjon av pause-funksjon, som også skulle sørge for at brukers selvvalgte pause skulle holdes utenfor. Etter første implementering av denne funksjonen fortsatte tidtakingen under pausen, og lysintervallene ble dermed ikke korrekt. Løsningen ble at vi lagret tiden som var da pausen ble initiert, og deretter satte hovedtidsvariablene til den tiden, og da fortsatte tidtakingen fra der vi slapp.

## 5. Presentasjon av koden

Det anbefales å ha koden tilgjengelig under lesing av disse avsnittene ([koden finner du her](#)).

### 5.1 Initiering av variabler

Først og fremst bruker vi Neopixel sitt biblioteket, som gjør at vi kan styre lysene med koden i Arduino. Porten som skal sende data til Neopixels defineres, og denne er satt til å være “pin 6”. Videre blir antall led-lamper definert og satt til 10 stk. Neopixel lenken defineres til “strip”, da det var en gjenganger i flere andre prosjekter. De globale variablene blir også satt i begynnelsen av koden, og alle brukes videre i de ulike metodene. Her settes tiden som bestemmer utgangspunktet for tidsintervallet som lyser opp Neopixelene.

### 5.2 Metoder

#### setup()

Det første vi kommer til i koden er setup(). Her defineres alle tilstandene til portene, det vil si om de enten skal lytte eller sende ut signal. Vi satte alle knappene til å lytte etter input, fordi vi vil registrere når en knapp blir trykket. Neopixelene mottar data fra pin 6, og den defineres da til output. Lydporten settes også til output, fordi vi får lyd når den mottar strøm fra Arduino. Tilt defineres til input, fordi vi vil registrere om det føres strøm eller ikke. Da gjenstår det tre steg igjen; *sette lysene i gang med strip.begin(), kjøre ferdig()-metoden som bekreftelse på start og sette lysene til sort slik at de er klare for start.*

#### loop()

Det neste steget i koden er loopen. Denne kjøres kontinuerlig, så lenge det ikke er en annen metode som kjører. Her har det blitt lagt inn en ny metode som sjekker de tre knappene, lyttTilKnapp().

#### lyttTilKnapp()

I denne metoden vil programmet lese av portene som er koblet til knappene. Dersom en knapp registreres som trykket, vil programmet kjøre den tilhørende metoden. Dersom pluss- eller minusknappene blir trykket, vil antall lys og timer øke eller minke før den går videre tilbake til lyttTilKnapp().

Det vil si at knappene alltid sjekkes, med mindre start blir trykket. Hvis start-knappen blir trykket vil den gå inn i tilhørende metode, og fortsette helt til alle lys er tent og ferdig-metoden er kjørt. Etter dette vil programmet gå tilbake igjen, og knappene vil bli sjekket på nytt.

## start()

Grunnen til at vi ikke har delt denne metoden inn i flere metoder, er fordi vi ville vise flyten i koden i ett og et samlet bilde over hva som skjer etter hverandre og hva som er avhengighetene i de forskjellige metodene.

Det som skjer i start()-metoden er først og fremst at det blir lagret en tid, som blir tiden akkurat nå ved hjelp av millis(). Denne er satt til startTid. Variabelen tidForNyttLys setter til å være det samme som den globale variabelen tid, ganger antall timer.

Programmet vil alltid gå inn i den første while-løkken så lenge alle lysene ikke er tent. Deretter sjekker en if-test om det er tid for å tenne et nytt lys. Hvis dette blir sant vil et nytt lys tennes, og lysIndeks økes med én. Dersom det ikke er det, vil den gå videre til else if som sjekker om tilten har endret seg fra HIGH til LOW. Hvis dette er tilfelle vil alle lys settes til blått inntil videre. Tiden frem til dette punktet blir lagret. Hvis tilt er tilbake til HIGH så hopper den ut av den kjørende while-løkken og lysene settes tilbake til slik de var før pausen ble initiert. Variablene som regner ut tid for lyd og tidForNyttLys settes lik de lagrede tidene, slik at tiden som gikk under pausen blir ekskludert fra tidtakingen.

Når den ønskede tiden for lyd er det samme som en time, vil lyden spilles av. Variabelen lydTid settes på nytt, slik at det igjen vil gå en time til neste lyd spilles. Etter at alle lysene er tent, vil programmet gå ut av den første while-løkken fordi lysIndeksen da vil være 10. Det vil da spilles av lyd og lys som gir feedback til bruker om at arbeidsøkten er ferdig.

## ferdig()

Ferdig()-metoden setter lysene til forskjellige farger ved å bruke random(tall) tre ganger som gir oss en unik rgb-fargekode for hver runde. Dette vil skje fem ganger, fordi vi har definert variabelen som må oppfylles i while-løkken til fem. Etter dette vil programmet hoppe tilbake til start()-metoden, sjekker om alt er ferdig, og så gå videre til loop(), der alt sammen skjer om igjen.

## Metoder for å bestemme lys

Lysene i Neopixel bestemmes av rgb-koder, slik som (198,43,183) som gir rosa farge. Det er laget metoder for å gi sort lys, som på en måte "nullstiller" lysene, og for å gi grønt lys som brukes etter start-knappen er trykket.