

# Teknisk rapport

IN1060 - Bruksorientert design

Vår 2021



Universitetet i Oslo - Institutt for informatikk

Edvard Strand Edding  
Lone Øye Johnsen  
Nilofar Bateni  
Sandra Klonteig  
Sara Sobut Talib



## Innholdsfortegnelse

<b>1.0 Introduksjon</b>	3
<b>2.0 Presentasjon av gruppen</b>	3
2.1 Mål for prosjektet	3
<b>3.0 Presentasjon av prosjektvideo</b>	3
<b>4.0 Prototypen</b>	4
4.1 Komponenter	5
4.2 Kretstegning fra Tinkercad	6
<b>5.0 Kode</b>	6
5.1 Forklaring av kode: Arduino	6
5.2 Forklaring av kode: Processing	7
<b>6.0 utfordringer og veien videre</b>	8
<b>7.0 Referanseliste</b>	9

## 1.0 Introduksjon

I denne rapporten presenterer vi den tekniske løsningen i prosjektarbeidet til Den digitale meteren. I rapporten inngår en kort forklaring av målet for prosjektet og prosjektgruppen, etterfulgt av en forklaring av prosjektvideoen, komponentene vi har brukt, kretstegningen og forklaring av kodesnuttene. Avslutningsvis vil vi diskutere utfordringer vi har møtt på og veien videre hvis vi hadde hatt mer tid.

## 2.0 Presentasjon av gruppen

«Den digitale meteren» består av fem informatikk-studenter; Edvard Strand Edding, Lone Øye Johnsen, Nilofar Bateni, Sandra Klonteig og Sara Sobut Talib.

### 2.1 Mål for prosjektet

Med temaet “ensomhet og isolasjon” for årets prosjekt, diskuterte vi i hvor stor grad vi kan bidra med å redusere denne følelsen hos målgruppen vår. Etter innledende datainnsamling og workshops fikk vi avklart hvilke behov og ønsker brukeren vår har. Deretter fant vi ut av at han ønsket seg en løsning som skulle gi ham en oversikt over aktivitetene som foregår i nærområdet i løpet av en uke, og muligheten for å få en påminnelse av de interessante aktivitetene på et senere tidspunkt.

## 3.0 Presentasjon av prosjektvideo

Prosjektvideoen tar for seg hvordan prototypen vår “Jungeltelegrafan” er en løsning til problemet målgruppen opplever med å ha for dårlig oversikt over aktiviteter i nærområdet, og skal bidra til motivasjon for å komme seg ut og sosialisere med andre. Først viser den hvordan brukeren enkelt kan flytte på et lokk for å sette i gang prototypen. Deretter kan brukeren trykke seg gjennom de daglige aktivitetene i en uke, som både leses opp høyt som en radio og vises tekstlig på skjermen. Dersom man er interessert i aktiviteten kan man velge å lagre den for å få en påminnelse slik at brukeren kan huske på og forberede seg på planen man har den aktuelle dagen.

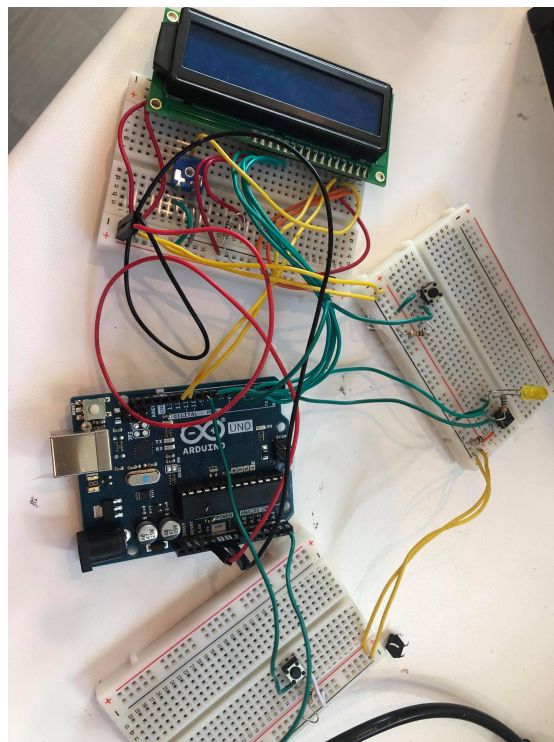
Link til video:

- 1) [YouTube](#)

## 4.0 Prototypen



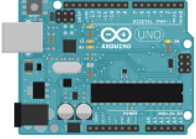

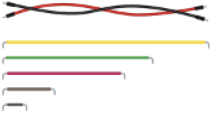





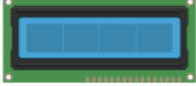
*Figur 1: Ferdigstilt prototype*



*Figur 2: Prototypens innmat*

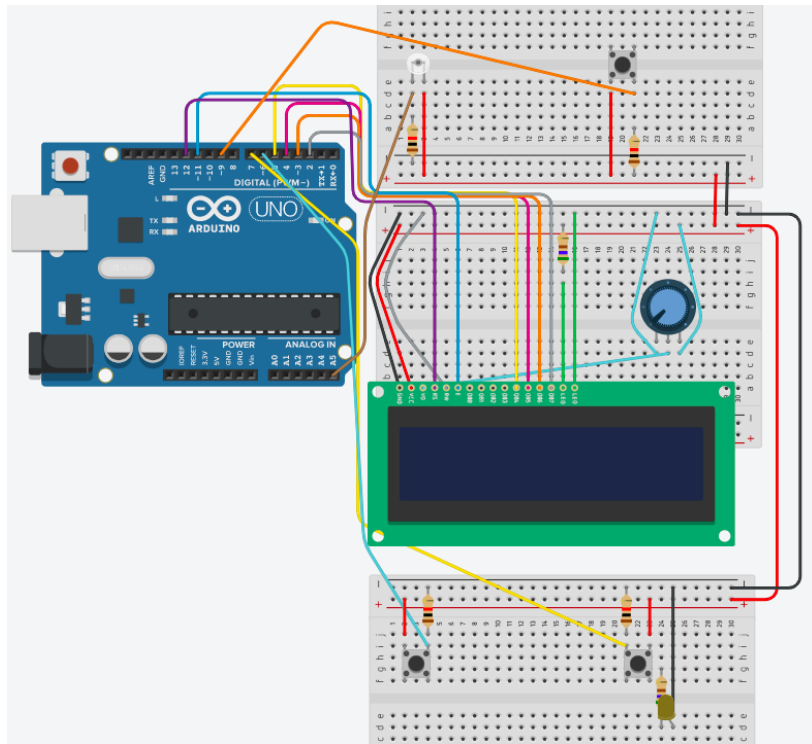


## 4.1 Komponenter

Komponent	Antall	Illustrasjon
Arduino Uno	1	
Breadboard	3	
Elektriske kabler	Ca. 30	
Photoresistor	1	
Push button	3	
Resistor	5	
Lysdiode (LED)	1	
Potentiometer	1	
LCD display	1	

(Fitzgerald & Shiloh, 2012, s. 6-8)

## 4.2 Kretstegning fra Tinkercad



Figur 3: Kretstegning lagd i Tinkercad

## 5.0 Kode

Koden er skrevet i både Arduino 1.8.13 og Processing 3.4.5. Processing har vi brukt til å få lyd fra prototypen, og er ment for å lese opp aktivitetene for de ulike dagene i en uke. Dette gjøres gjennom Serial, som kommuniserer med Arduino.

Link til kodene våre:

- 1) [Arduino-kode](#)
- 2) [Processing-kode](#)

### 5.1 Forklaring av kode: Arduino

I starten av koden settes alle de relevante variablene som benyttes under setup() og loop(). I tillegg til dette opprettes det tre array-lister “tidspunkter”, “aktiviteter” og “interesser”. De førstnevnte brukes for å vise de aktuelle aktivitetene og deres tidspunkter på LCD-displayet, og den sistnevnte brukes for å holde oversikt over om brukeren er interessert (true) eller ikke interessert (false) i aktiviteten. Delen av koden som skal gi brukeren påminnelse om aktiviteten mangler, og grunnlaget for hvorfor beskrives under utfordringer senere i rapporten.

I `setup()` blir modusen for “neste”, “lagre”, “LEDlys” og “nullstill” satt til input og output. `Serialbegin(9600)` åpner serial-porten og setter datahastigheten til 9600 bps slik at LCD-skjermen leser dataen raskt nok. Variabelen `value` leser av den analoge inputen i form av lysstyrke fra photoresistoren. While-loopen fortsetter så lenge verdien på lysstyrken er under 200, og når den overgår denne verdien vil brukeren se “velkommen” og “trykk på nesteknappen” på LCD-skjermen.

I `loop()` vil knappene for “neste”, “lagre” og “nullstill” kontinuerlig bli sjekket om de trykkes på av brukeren. Dersom man trykker på neste-knappen vil variabelen “hits” bli oppdatert med +1 slik at den neste aktiviteten, tidspunktet og interessen i array-listen blir vist og slik at riktig aktivitet blir lagret på riktig nummer i arrayet av interesser. I tillegg til dette vil `Serial.write(hits-1)` kommunisere med processing-appen for å få denne til å starte avspillingen av lyd gjennom processing. Når brukeren når slutten av array-listene, vil LCD-skjermen gi beskjed om at man kan starte på nytt og hits settes tilbake til 0.

Når man trykker på lagre-knappen vil lysdioden lyse opp to ganger etter ønske fra brukeren. Samtidig vil den aktuelle aktiviteten og interessen settes til true i arrayet interesser. Deretter viser koden at nullstill-knappen setter alle verdiene i interesser tilbake til false, og brukeren blir sendt tilbake til start. Dette kan skje når som helst under interaksjonen, og skal bidra til at brukeren kan rette opp i feil ved bruk.

## 5.2 Forklaring av kode: Processing

I starten av koden settes alle de relevante variablene som benyttes under `setup()` og `draw()`. Under `setup()` benyttes “`minim`” for å laste opp lydfile og “`port`” for å lese av porten som brukes under kompileringen av koden i Arduino. Deretter tilordnes `Audioplayer`-variablene de rette mp3-filene for hver aktivitet. Under `draw()` vil metoden `serialEvent()` først lese av verdien som kommer fra `Serial.write()` i Arduino-koden. Første gangen man trykker på neste-knappen i Arduino-koden vil lydfile for den første aktiviteten leses av, så de neste helt til den når siste aktiviteten. `Rewind()` brukes for å spole lydfile tilbake, slik at de begynner fra start dersom brukeren velger å gå gjennom aktivitetene flere ganger i Arduino-koden. Denne koden er inspirert av og revidert av tilsvarende kode (BGSU, 2017; Park, 2016).

## 6.0 utfordringer og veien videre

Arbeidet med Arduino og Processing har vært en lærerik prosess, og underveis i prosjektet møtte vi på noen utfordringer. En av utfordringene vi støtte på var under byggingen av prototypen, hvorav nullstill-knappen og photoresistoren ikke ville fungere som de skulle. Da vi opplevde at det kan være lett å bli “blind” på egen kobling og kode, tok det tid før vi innså at pluss- og minuskilden var koblet på feil side av breadboardet og at vi hadde glemt å tillegge nullstill-knappen INPUT under setup(), som førte til at photoresistoren fungerte slik nullstill-knappen skulle. Vi opplevde også at knappen ble noe sensitiv i form av at vi ikke trengte å trykke ned knappen, men at det holdt å være borti den for at den skulle virke. En annen utfordring var å finne en løsning for å lagre de aktivitetene brukeren var interessert i, men ved research og hjelp fra gruppelærer fant vi ut at dette kunne løses ved arraylister.

Den største utfordringen vår var imidlertid å få Processing og lydavspilling til å fungere optimalt. Feilmeldingen vi slet mest med var at Processing-applikasjonen ikke fant serial-porten som ble benyttet i Arduino. Dette gjorde oss usikre på om koden i Processing i det hele tatt var riktig skrevet, og det førte til mye prøving og feiling. Her valgte vi å distribuere Arduinoen og kodene videre til et annet gruppelem, som fikk skrevet inn riktig serial-port, og til vår begeistring så viste det seg at det kun var serial-porten som hadde blitt skrevet feil og koden var korrekt. Dette ble en tidkrevende prosess, og førte til at vi ikke fikk gjennomført koden for påminnelsene brukeren ønsket å få gjennom sine lagrede aktiviteter i interesser-arrayet. Hvis vi hadde hatt mer tid, ville vi valgt å løse denne problemstillingen med millis() som kunne ventet en tidsperiode før den leste opp aktivitetene som var lagret av brukeren. Denne tidsperioden ville for øvrig blitt kortere enn den ville vært i realiteten, da brukeren setter interessene uken før og ønsker påminnelsen uken etter.

En annen problemstilling vi diskuterte var at det burde vært mulig å slette en spesifikk aktivitet, fremfor å nullstille alle aktivitetene for en uke. Derimot anså vi det som enklere for vår bruker å nullstille alle aktivitetene og ha muligheten til å starte på nytt, ettersom nullstilling av én og én aktivitet kunne blitt mer utfordrende for brukeren i vår målgruppe med mindre teknologisk kompetanse. Videre ville vi i en ideell situasjon ønsket å ha muligheten for å kommunisere med nettsider som kunne hentet inn og oppdatert aktivitetene for hver uke i koden. Vi valgte imidlertid å løse det med forhåndsbestemte aktiviteter for å visualisere hvordan den tiltenkte ideelle løsningen ville fungert i realiteten.

## 7.0 Referanseliste

BGSU (2017) Week 14 Day 1 - Processing Sound [webside]. Tilgjengelig fra:  
<<https://bgsu.instructure.com/courses/1157282/pages/week-14-day-1-processing-sound>> [10.05.2021]

Fitzgerald, S. & Shiloh, M. (2012) *Arduino Project Books*. Torino, Italy. Arduino LLC. pp 6-8

Park, J. (2016) Using Sound in Processing with Minim. Tilgjengelig fra:  
<[https://www.youtube.com/watch?v=LcX36OxgZgg&ab\\_channel=JohnPark](https://www.youtube.com/watch?v=LcX36OxgZgg&ab_channel=JohnPark)> [10.05.2021]