

IN1060 - TEKNISK RAPPORT



Medlemmer:

Vilde Støen Vrenne, Agon Krasniqi,

Håkon Myhre og Christian Olsen

Innholdsfortegnelse

Introduksjon	3
Hvem er vi?	3
Kort om prototypen	3
Mål med prosjektet	3
Video	4
Hvordan løsningen fungerer	5
Komponenter vi har brukt:	5
Kretsdiagram	6
Forklaring av kode:	8
Utfordringer	11
Valg gjort:	11
Endring av prototype	11
Ramme	11
Knappene	12
Feiltesting	12

Introduksjon

I denne tekniske rapporten skal vi presentere mer i detalj den tekniske løsningen av vår endelige høyoppløselige prototype. Det blir lagt fokus på hvordan de forskjellige komponentene fra Arduino settet blir inkludert i løsningen, hvordan de fungerer og hvorfor vi bruker de komponentene vi gjør.

I rapporten vil vi også gå igjennom hva vårt mål med prosjektet er. Her vil vi reflektere om prototypen tilfredsstiller de kravene og behovene vi har satt eller ikke, og om målene har endret seg. Vi vil også gå inn på hvordan utformingen til slutt ble sånn den ble.

Hvem er vi?

Gruppen består av fire studenter på Universitet i Oslo, vi heter Christian Olsen, Håkon Myhre, Agon Krasniqi og Vilde S. Vrenne. Vi har gruppenavnet CHAV som er satt sammen av våre forbokstaver.



Bilde 1: Gruppen CHAV, fra venstre: Christian Olsen, Håkon Myhre, Agon Krasniqi og Vilde Vrenne

Kort om prototypen

Prosjektet har fått navnet “Arbeidsrammen”. Navnet er inspirert av en fotoramme for å gi det et litt mer diskret utseende i sammenheng med konteksten det skal brukes i, nemlig på kontor og i arbeidssituasjoner.

Prototypen er en litt tykk fotoramme. Den skal gi informasjon om aktive arbeidsoppgaver og skal kunne kommunisere med arbeidsgiver.

Mål med prosjektet

Vårt mål med prosjektet er å lage en løsning som svarer på problemstillingen vår: “*hvordan strukturere arbeidsdagen for hjemmekontorarbeidere.*” Med problemstillingen er det viktig å ta hensyn til behovene som oppstår og kravene som settes til hjemmekontor. Mer om mål for prosjektet og problemstillingen finnes i prosjektrapporten.

Med evaluering fant vi ut at for å strukturere bedre arbeidsdagen var det behov for at arbeidsgiver har kontroll på hvor mye hver enkelt arbeidstaker har, sånn at det er lettere å velge hvem som kan ta nye/flere arbeidsoppgaver. Dette er et behov vi i praksis ikke fikk prototypet reelt, men som vi i teorien sier den skal gjøre. Dette fordi vi hadde mangel på ressurser, og at det i realiteten ville kunne trenge tilgang til å hente informasjon fra database eller lignende.

Noen av kravene vi satt til prototypen var:

- *Prototypen skal kunne vise flere arbeidsoppgaver*
Dette er implementert ved å ha to LCD skjermer for å vise arbeidsoppgavene
- *Bruker må kunne velge mellom arbeidsoppgaver*
Dette er implementert med to knapper (en “opp”, en “ned”), hvor man kan bla igjennom en liste med arbeidsoppgaver.
- *Bruker må få feedback for å indikere arbeidsmengde*
Implementert med en fargekodet rgb LED pære man kan endre farge på
- *Systemet kommuniserer med arbeidsgiver (ikke implementert, forklart i avsnittet over)*
- *Bruker må kunne signalisere at en arbeidsoppgave er ferdig.*
Implementert på den måte at bruker trykker på en knapp for å fjerne “aktiv” arbeidsoppgave fra listen. I teorien ville vi ha lagt den fullførte arbeidsoppgaven i en egen liste, slik at arbeidsgiver og kollegaer kan vite hva som er gjort. (Er ikke gjort, da vi måtte ha hatt et database system)

Video

Videoen starter med å demonstrere noe av problemstillingen vi har prøvd å løse; en kontorarbeider som er ukonsentrert i hjemmekontoret.

Så er det en oversikt over hovedfunksjonaliteten til løsningen. Der demonstreres interaksjon med alle knappene med feedback på skjermene og led-pæren i mens rammen ligger horisontalt i kontrollerte omgivelser. Knappene er sårbare, så dette demonstreres bare mens rammen ligger. Dette blir etterfulgt av at personen bruker artefaktet i naturlige omgivelser på kontorpulten sin. Her demonstrerer vi brukskonteksten.

Til slutt har vi nærbilder av utformingen til artefaktet og et klipp der vi ser implementasjonen av arduinoen med funksjonalitet.

Link til videodemonstrasjon av “Arbeidsrammen”:

<https://youtu.be/8nDayPxqla0>

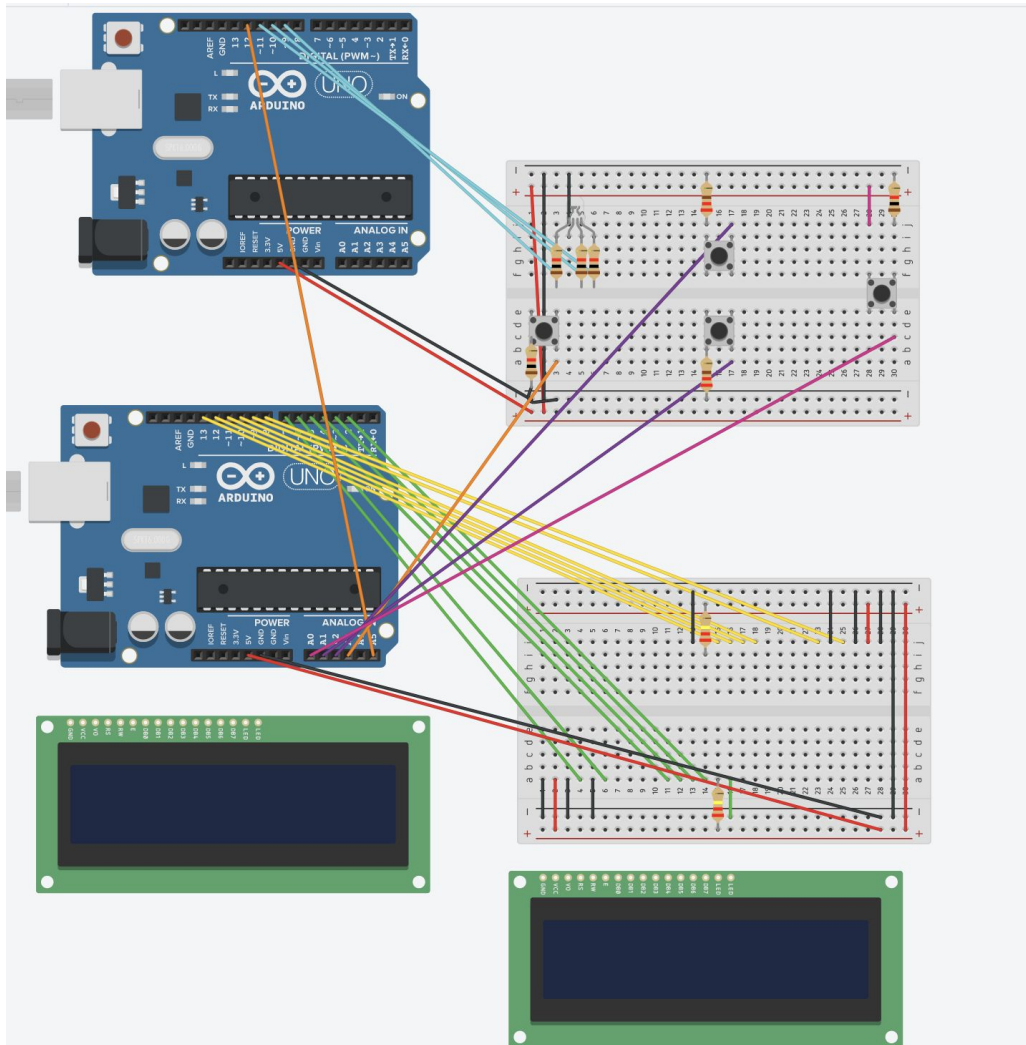
Hvordan løsningen fungerer

Prototypen har to led skjermer på høyre side som skal vise informasjon om arbeidsoppgaver eier av rammen har å gjøre. Den har også en rgb LED pære over skjermene, med en tilhørende knapp for å endre fargen. Pæren skal indikere til sjefen og arbeidstaker hvor mange arbeidsoppgaver han/hun har. Grønn farge betyr få/ingen arbeidsoppgaver, gult betyr passe mengde arbeidsoppgaver mens rødt betyr mange arbeidsoppgaver. Det er to knapper man kan trykke på for å bla gjennom aktive arbeidsoppgaver. Til slutt er den en knapp som skal fjerne arbeidsoppgaven på nederste skjerm for når man er ferdig med den. Til venstre for arduinoen er det et bilde som bruker kan bytte ut til det de føler passer (vi har i dette tilfelle valgt et vilkårlig fjell).

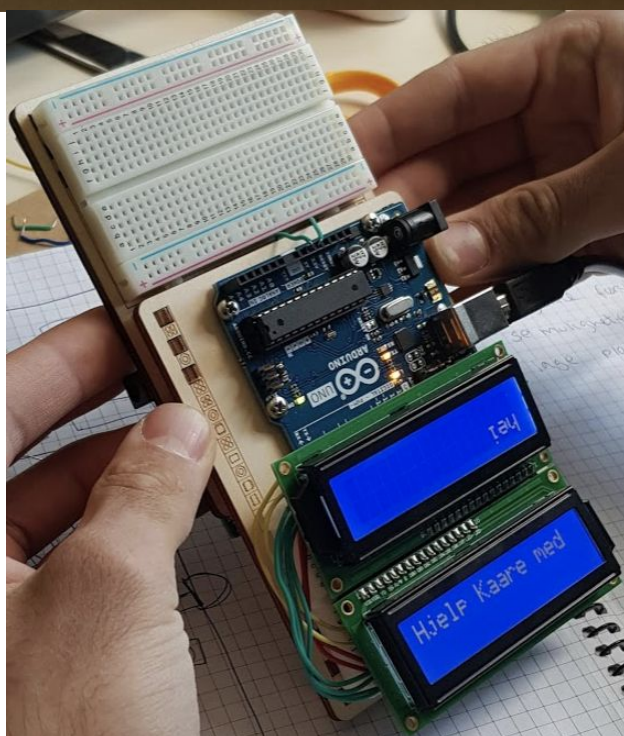
Komponenter vi har brukt:

- ledninger for å koble sammen alle komponentene med arduino og for å gi strøm
- 220 ohm resistorer
- 2 Liquid Crystal Display-er for å gi informasjon om aktive gjøremål for arbeidstaker
- 1 rgb LED pære
- Totalt 4 knapper: 1 til å endre lys på pære, 2 til å gå fram og tilbake for å vise gjøremål og 1 til å fjerne gjøremål som er ferdig
- 2 arduinoer
- 2 breadboards

Kretsdiagram

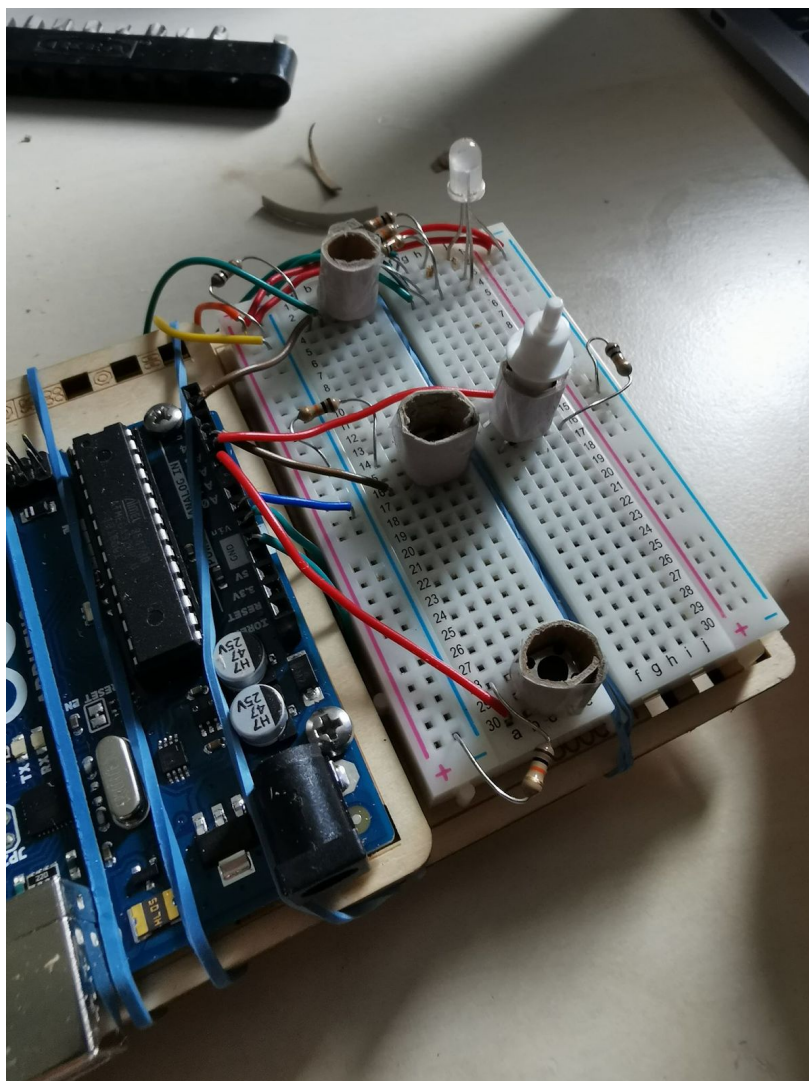


Bilde 2: Kretsdiagram (Lagd så likt som mulig med programmet TinkerCad)



Bilde 3: Ferdig prototype i hvordan den ser ut.

Bilde 4: Arduino-delen av prototypen uten implementeringen av knappene og rgb-LED pæren



Bilde 5: Viser breadboardet ferdig implementert uten rammen. "paprullene" er for å støtte opp knappene

Forklaring av kode:

Dummy data

Vi har ikke implementert en måte for brukeren eller sjefen til å legge til gjøremål i systemet, uten å måtte gå inn i koden. Det er halvveis hardkodet, der vi i setup kallet på metoden `leggTilGjøremaal(String tekst)` flere ganger for å legge til eksempler på gjøremål, altså dummydata.

Statusindikatoren for hvor opptatt arbeideren er med gjøremål er ikke koblet opp mot en mottaker. Det er kun en indeks 0-3 som styrer RGB-pæren her. Her har det altså for øyeblikket ingen virkning når brukeren trykker på statusknappen, annet enn kosmetisk med fargeskiftet.

Tilrettelegging for utvidelse

Koden er tilrettelagt slik at det skal enkelt være mulig å legge til flere skjermer i arrayet over skjermer. Gjøremålene som vises på hver enkelt skjerm er ikke “hardkodet”, men dynamisk i forhold til hvor mange skjermer som er lagt til.

Bruken av flere arduinoer

Vi var nødt til å bruke 2 arduinoer pga. mangel på pins. Mye av grunnen til dette var at vi ønsket å benytte to fungerende LCD-skjermer. “Hovedarduinoen” styrer to begge skjermene, og alle 4 knapper. Input fra knappen som angir lysstatusen går til begge arduinoene, slik skal begge arduinoene i teorien få lik statusindeks. Den andre arduinoen har kun ansvar for å få input fra lysknappen og outputter RGB-verdier til statuspæren.

- INO-koden ved navn “*Arbeidsrammen.ino*” er lastet opp til hovedarduinoen med skjermene og knappene.
- INO-koden ved navn “*Statussetter.ino*” er lastet opp til den arduinoen som kun har ansvar for å styre LED-lyset.

Biblioteker

#include <QList.h>

Kilde: Martin Dagarin (<https://github.com/SloCompTech/QList>)

Bibliotek for en dynamisk dobbel-lenket liste.

Brukt for å forenkle koden vår når vi skal legge til gjøremål til systemet. Har desverre ikke har tid til å implementere en dynamisk liste selv, men vi prioriterte ikke dette siden vi har gjort dette og er kjent med prinsippet fra java.

#include <LiquidCrystal.h>

Kilde: Adafruit, Arduino (<https://github.com/arduino-libraries/LiquidCrystal>)

Dette er et standardbibliotek som skaperne av arduino tilbyr. Biblioteket tilbyr metoder for å enkelt konvertere tekst i koden til standard LCD-komponentet som medfølger Arduino Uno pakken. Vi brukte denne for å vise gjøremålene til brukeren på LCD-skjermene.

#include <ezButton.h>

Kilde: ArduinoGetStarted.com (<https://github.com/ArduinoGetStarted/button>)

Bibliotek for enklere oppretting og lesing av knapp-komponenter. Dette forenkler slik at man ikke trenger å kode debounce manuelt, samt at biblioteket tilbyr en rekke nyttige metoder.

Implementasjonen av debounce er noe alle på gruppen er kjent med fra tidligere oppgaver, så dette valgte vi ikke å bruke tid på her. Samt. bruker vi såpass mange knapper at vi føler et slikt bibliotek ville vært naturlig å bruke uansett.

Arduino 1 kode:

setup()

Åpner en port for utskrivning til terminal, slik at det er enklere for utvikler å se hva som foregår
Legger til div gjøremål ved kall på metoden leggTilGM(String tekst).

loop()

Vår ambisjon var å ha så lite kode som mulig i loopen og kun kalle på andre metoder her. Det eneste som skjer i loop-metoden er at den sjekker om knappene bli trykket på.

leggTilGM(String tekst)

Tar imot et gjøremål i form at tekst og legger denne til i den dynamiske, globale listen som inneholder alle foreløpige gjøremål.

fjernGM()

Fjerner gjøremålet som vises på den nederste “hovedskjermen”. Den tar ingen parameter og vil kun fjerne gjøremålet med indeksen for gjøremålet som er valgt (altså det som vises på nederste skjerm).

navigerOpp()

Flytter alle gjøremål på skjermene opp et hakk. Endrer gjøremålet på hovedskjermen som brukeren kan interagere med. Øker gjøremålsvalg.

navigerNed()

Samme som navigerOpp(), bare navigerer gjøremål nedover.

visTekst()

Viser gjøremålene på alle skjermene. Itererer gjennom alle skjermene og viser ulike gjøremål på hver enkelt skjerm. Det er ikke hardkodet antall skjermer i metoden, så det globale arrayet over skjermer kan endres og denne metoden vil fortsatt fungere. Metoden kalles kun på når gjøremålene endres, altså brukeren navigerer opp/ned eller legger til/fjerner gjøremål. Den kalles ikke i loopen, slik unngår vi flickering i skjermen og unødvendig prosessering av data.

settFarge()

Redundant kode i denne arduinoen. Beholdt denne metoden, for vi ønsker at indeksen for statusen til brukeren skal kunne hentes fra “hovedarduinoen”. På denne måten har begge arduinoene en kopi av statusen til brukeren. Denne er senere planlagt til å kunne hentes ut slik at arbeidsgiver eller den som tildeler arbeidsoppgaver skal kunne se.

Arduino 2 kode:

setup()

Velger pins til å outputte til rgb LED-pæren.

loop()

Sjekker kontinuerlig om statusknappen blir trykket på.

void settFarge()

Når knappen for å endre status trykkes, blir denne metoden kalt og endrer farge på led pæren ved å kalle på metoden **visFarge** med fargeverdier som parameter. Fargene bestemmes i denne metoden ettersom hvilken indeks den globale variabelen `fargeStatus` får.

void visFarge(int rVerdi, int gVerdi, int bVerdi)

Tar inn verdier for rød, grønn og blå og skriver disse verdiene til rgb LED-pæren som skal indikere statusen til brukeren.

Utfordringer

Den ene skjermen i den endelige prototypen er opp-ned for å få plass på ett og samme breadboard. I teorien så burde ikke dette vært noe problem når man kan snu teksten ved bruk av software. Vi fant ingen enkel måte å gjøre dette på, uten å måtte lage custom tegn som som vises på skjermen som snur alle bokstavene vi ville vise. Dermed valgte vi ikke å bruke tid på dette.

Knappene var en stor utfordring. Mangel på en lab og loddeutstyr gjorde at vi var nødt til å bli kreative med utformingen av disse. Vi kunne ikke plassere knappene hvor vi ville, men var bundet opp til breadboardet. Vi løste dette med å (på godt norsk) "jallamekke" standoffs. Dette var pinner som man kunne trykke på, som vi leder inn mot knappene ved hjelp av hjemmelagde spor. Dette var en sårbar konstruksjon og fungerte så lenge vi holdt prototypen flatt og opp mot rundt 40 grader helling.

Valg gjort:

Endring av prototype

På grunn av evaluering, fant vi ut at prototypen vår måtte bli litt mer diskre, i den forstand at vår første høyoppløselige prototype var en "boks" klippet ut som et fjell. På denne prototypen skulle en figur styrt av en strikk på tre motorer "klatre" opp fjellet, til arbeidsdagen var over. Dette gikk vi helt bort ifra, da dette virket litt uprofesjonelt på et kontor og bruker syntes det hadde vært rart og få kommentarer som "Hvorfor har du et fjell på pulten?". Vi fikk også tilbakemelding om hvorfor figuren ikke passet så bra var at hun mente de viktigste funksjonene var uansett en oversikt over oppgaver og tilbakemelding til sjefen over hva man har gjort.

Ramme

Utfra evalueringen vi hadde av første høyoppløselige prototype fikk vi et par forslag om hvilken retning hun mente vi kunne ta prototypen. Hun foreslo først å gjøre det om til en heis, som kunne løftes opp og ned. Det andre forslaget hennes var å lage en fotoramme, som bygget mye på "diskre" delen hun hadde snakket om tidligere. Vi i gruppen følte rammen ville passet best til tilbakemeldingene hun allerede hadde gitt og valgte derfor dette som utgangspunkt for denne høyoppløselige prototypen.

Knappene

Vi valgte å bare bruke knapper for å interagere med systemet. Dette gjorde vi med tanke på designprinsippene affordance, consistency, feedback og constraints. Knapper er noe som lett gir feedback til bruker når den har blitt trykket på. Vi vurderte potensiometer på å f.eks. bla igjennom arbeidsoppgaver, men fant ut at den ble mer uforutsigbar og ikke ga god feedback får når den bladde igjennom. Knapper gir god affordance og consistency i prototypen fordi det er lettere å gi symbol til de, og alle vet hvordan knapper fungerer intuitivt. Dette gjør det også lett å lære seg å bruke systemet. Med vår minimalistiske stil på prototypen og de relativt få knappene, hindres bruker i å gjøre feil, som bidrar til designprinsippet constraints.

Feiltesting

Da vi satt sammen den høyoppløselige prototypen var det vanskelig å få knappene til å sitte godt, med de festet til breadboardet, når rammen ble festet. Etter et strev endte vi med at noen av knappene ikke kunne bli trykket på en god måte (isærs LED knappen). Vi valgte å bruke den slik vi hadde den, da vi fortsatt fikk til å bytte farge på lysene og de andre knappene fungerte. Grunnet mangel på ressurser fikk vi ikke til å lage en permanent løsning, med f.eks. bruk av lodding.