

A dark blue vertical bar runs down the left side of the page. A horizontal arrow, also in dark blue, points to the right from the bar at the level of the 'Våren 2024' text.

Våren 2024

# Teknisk rapport

IN1060 – Bruksorientert design

Several thin, dark blue wavy lines originate from the bottom left corner and curve upwards and to the right, overlapping the vertical bar.

Mariell Ramberg, Erik Øverby, Maren Tveitereid,  
Jakob Solli og Ingvild Nissen-Lie

# Innhold

1. Introduksjon	2
1.1 Prosjektgruppen – Store variabler	2
2. Mål for vårt prosjekt	2
2.1 Prototypens hensikt	3
3. Videopresentasjon	4
4. Teknisk spesifikasjon	4
4.1 Komponenter	4
4.2 Gjennomgang av koden	5
4.2.1 Kodelikheter i Node A- oversiktsnode og Node B- pultnoden	5
4.2.2 Node A – oversiktsnode	6
4.2.3 Node B – pultnode	7
5. Avslutning	8
Referanser	9

# 1. Introduksjon

I denne rapporten har vi som formål å dokumentere og demonstrere den tekniske løsningen som er utviklet i dette prosjektet. Målet med rapporten er å gi en grundig beskrivelse av hvordan artefakten fungerer og vise hvordan den kan inngå i brukernes brukskontekst. Rapporten inneholder en presentasjon av prosjektgruppen og målet med prosjektet. Videre vil den forklare hva videoen viser, og til slutt gi en grundig gjennomgang av de tekniske spesifikasjonene, inkludert komponenter og koden som er brukt i dette prosjektet.

## 1.1 Prosjektgruppen – Store variabler

Gruppen vår består av fem medlemmer: Ingvild, Mariell, Maren, Erik og Jakob. Vi hadde lite kjennskap til hverandre fra før og vi har med oss en kombinasjon av ulike arbeidserfaring, livserfaring og alder, som har vært berikende i vår tilnærming til prosjektet. På bakgrunn av vårt mangfold i gruppa valgte vi å kalle oss for «Store variabler».

## 2. Mål for vårt prosjekt

Vårt prosjekt har fokusert på å utforske problemområdet forstyrrelser i et åpent kontorlandskap. I forkant av kontakten med brukere satte vi oss inn i temaet og fant ut at det er mange som opplever å bli forstyrret i kontorlandskapet de arbeider i og at disse forstyrrelsene ofte skyldes andre kollegaer (Lilleeng, 2023). Ved prosjektets start gjorde vi oss noen tanker om at brukerne muligens ville ha lignende opplevelser der de fort kunne bli forstyrret inne på arbeidsrommet ved at personer i landskapet oppsøker og tar kontakt med andre.

I løpet av denne våren har vi hatt kontakt med fire brukere. Disse brukerne arbeider ved en videregående skole i Oslo og sitter i et åpent kontorlandskap. Gjennom semistrukturerte intervjuer, flere observasjoner og analyser av data, fant vi ut at utfordringen med dette åpne kontorlandskapet var mer kompleks og at utfordringene i større grad handlet om inn- og utstrømninger av personer på arbeidsrommet. Brukerne satte ord på at de opplevde det som forstyrrende at andre ansatte og elever oppsøkte enkeltpersoner i arbeidsrommet ved deres arbeidsplass, og at de ofte ble stående å ta en lengre prat med den ansatte eller eleven. Ofte var det slik at andre ansatte gikk inn til den ansatte uten å nødvendigvis vurdere om den ansatte var tilgjengelig eller utilgjengelig, når personen satt ved plassen sin. Til tross for at arbeidsrommet også hadde glassdør, var det høye skillevegger som gjorde at man ikke hadde sikt til alle de ansatte. Dette gjorde at det var vanskelig å se om den ansatte virket tilgjengelig eller utilgjengelig, eller om den ansatte i det hele tatt var på plassen sin.

Brukerne våre var også opptatt av mulighetene til å redusere den generelle støyen og forstyrrelsene som brukerne innad i arbeidsrommet selv stod for. De fortalte oss at det fort kunne oppstå en del høylytt prating og sosialisering med de andre på rommet der man fort ble involvert uten at man nødvendigvis ønsket det. Til tross for at brukerne satte ord på at dette kunne være forstyrrende for

dem, ønsket de likevel ikke å oppleve at de i for stor grad ble regulert på dette området. De ønsket derimot fortsatt å ha muligheten til å kunne ha disse sosiale interaksjonene siden arbeidsrommet hadde et godt sosialt fellesskap.

På bakgrunn av våre analyser og datainnsamlinger var målet vårt å utvikle en artefakt med brukerne som kunne signaliserer brukernes tilgjengelighet eller utilgjengelighet, som både eksterne personer utenfor arbeidsrommet kunne oppfatte og som personer innad på arbeidsrommet kunne oppfatte. I og med at brukernes arbeidshverdag også innebar å kommunisere med andre ansatte som ledere og rådgivere, i tillegg til elever, var det naturlig å lage en artefakt der eksterne personer utenfor arbeidsrommet enkelt kunne signalisere direkte til brukeren at de ønsket å komme i kontakt med brukeren. Videre var målet at dette systemet også kunne ha en signaleffekt på brukerne innad på arbeidsrommet, om mange ansatte var satt som utilgjengelige, ville man kanskje forsøke å overholde arbeidsroen og minimere forstyrrelsene.

Gjennom workshops og evaluering av den første prototypen fant vi ut at brukerne hadde enkelte funksjonelle- og ikke-funksjonelle krav til løsningen. De ønsket at systemet skulle gi en oversikt over deres egen status og de ønsket å kunne endre statusen sin fra sin egen arbeidspult. Videre var de også spesielt opptatt av at interaksjonen med systemet skulle være enkelt og effektivt.

## 2.1 Prototypens hensikt

Vår prototype er utformet som et signalsystem der en artefakt er plassert utenfor arbeidsrommet og den andre artefakten er plassert på arbeidspulten til den ansatte. Artefakten utenfor har et brukergrensesnitt som viser en oversikt over ansatte som har arbeidsplass i det åpne kontorlandskapet. Denne oversikten er på artefaktens fremside og bakside. Til hvert navn er det et rødt og et grønt lys, i tillegg er det en knapp til hvert navn på artefaktens ene side. Denne knappen er tiltenkt at eksterne personer kan interagere med. Utenfor arbeidsrommet får du da en oversikt om personen du skulle ha tak i er tilgjengelig eller utilgjengelig. Om personen du ønsker å få tak i er satt som tilgjengelig kan interaksjonsmekanismen gjennom knappen tillate den eksterne personen å tilkalle ønsket person.

Artefakten på arbeidspulten har et brukergrensesnitt med to knapper og et RGB LED-lys. Brukeren kan trykke på den ene eller den andre knappen for å utløse interaksjonsmekanismen som sender et signal til noden utenfor arbeidsrommet om at man er tilgjengelig eller utilgjengelig. LED-lyset på noden på arbeidspulten vil også angi grønt eller rødt lys ut ifra oppgitt status. Om brukeren er satt som tilgjengelig og personen utenfor utløser interaksjonsmekanismen gjennom knappen på artefakten utenfor rommet, altså en ekstern person utenfor rommet tilkaller deg, vil RGB LED-lyset på artefakten på arbeidspulten blinke blått i 5 sekunder før det bytter tilbake til den opprinnelige oppgitte statusen.

I vårt prosjekt benyttet vi oss av arbeidsrommets glassdør. Vi valgte å henge artefakten med oversikt over de ansatte på glassdøren slik at artefaktens ene side, ville bli synlig for ansatte på arbeidsrommet.

På denne måten vil oversikten også kunne signalisere til de inne på arbeidsrommet hvem som er tilgjengelig og hvem som er utilgjengelige, og forhåpentligvis ha en signaleffekt på forstyrrelser og støy inne på arbeidsrommet.

Ideelt sett skulle vi laget to artefakter som viser oversikt over de ansattes status, en utenfor og en inne på arbeidsrommet, i tillegg skulle vi gjerne ha laget flere arbeidspult-artefakter: én til hver ansatt, men det ville blitt for omfattende for dette prosjektets rammer. Vil valgte derfor å lage én funksjonell prototype til arbeidspulten og én artefakt som gir oversikt over de ansatte på begge sider, der ett av lysene er koplet med noden på arbeidspulten, mens de andre lysene simulerer den tiltenkte funksjonen uten å være koplet opp mot den funksjonell pultnoden.

### 3. Videopresentasjon

a)

Videoen viser hvordan prototypen vår fungerer. I et tenkt scenario sitter man på arbeidspulten og stiller inn sin egen status med rød eller grønn knapp. LED- lyset viser oppgitt status på både pultnoden og oversiktsnoden. Videre viser videoen at det ikke blir gitt et ringeklokke-signal hvis den du ringer på er satt som utilgjengelig. Om den du derimot ønsker å få tak i er satt som tilgjengelig, vil ringeklokke-knappen signalisere til den ansatte gjennom et blått blinkende lys at noen ønsker å få tak i deg.

I videoen viser vi hvordan vi ville plassert oversiktstavlen på glassdøren inn til arbeidsrommet, slik at den er synlig både innenfra og utenfra arbeidsrommet. Vi viser også at noden til arbeidsplassen kan festes på både bord eller vegg ved hjelp av borrelås.

I videoen viser vi statusvinduet til navnet Ingvild, og dens tilhørende pultnoden som kommuniserer med hverandre. De andre lysene på tavlen er satt opp med vanlig Arduino Uno.

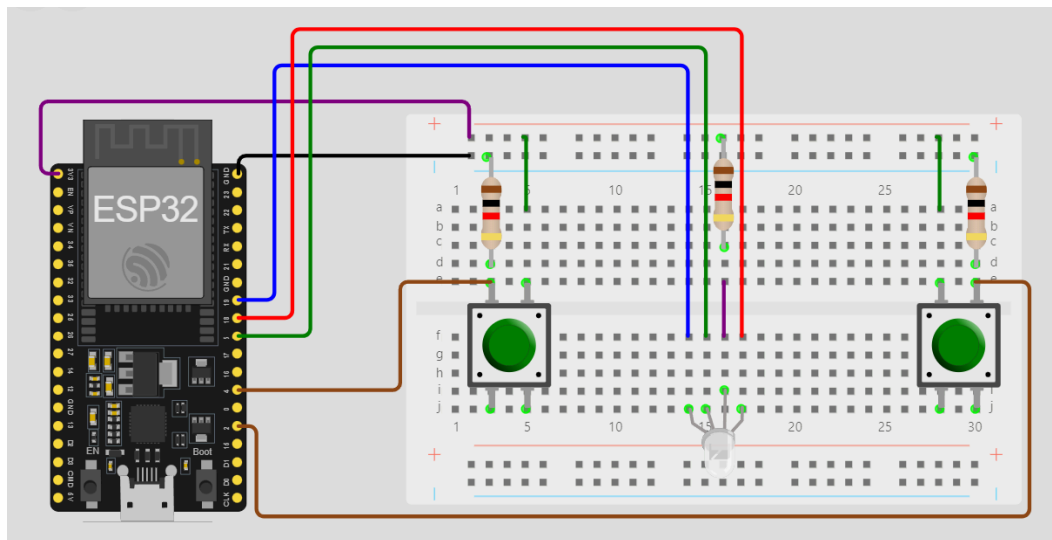
b)

<https://youtu.be/oP3tDwd05XI>

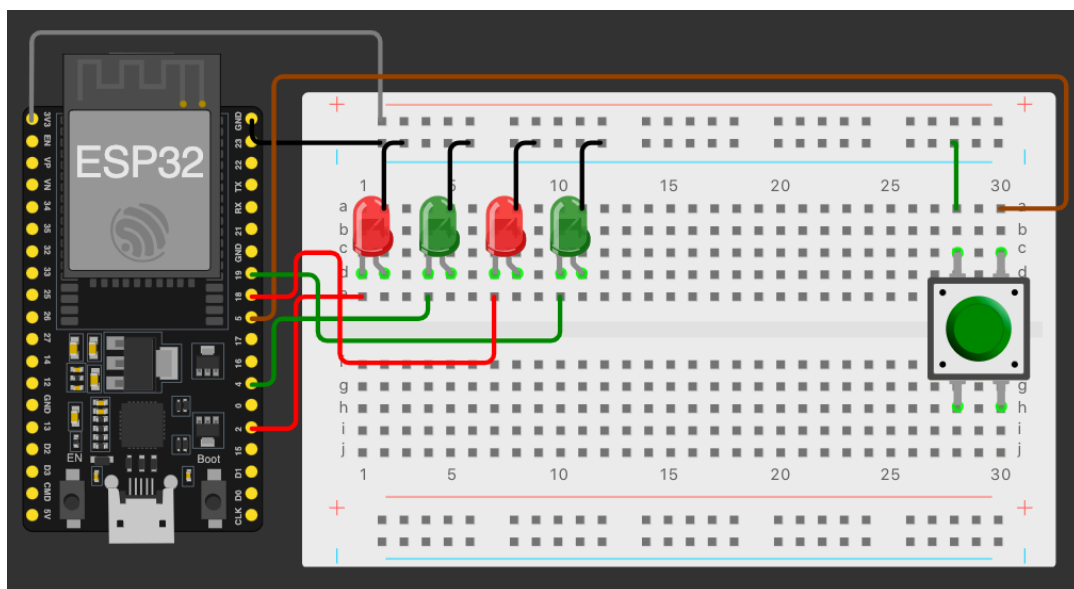
### 4. Teknisk spesifisering

#### 4.1 Komponenter

Oversikt over kretsløp for pultnoden:



Oversikt over kretsløp for oversiktsnoden:



Komponenter i pultnoden	Antall
ESP32 mikrokontroller	1
LED RGB	1
Resistorer	3
Breadboards	2
Pushbuttons	2

RGB dioden vi brukte hadde en felles anode som vil si at den må kobles til positiv spenning for å fungere. De andre pinnene for fargene måtte dermed stilles negativt for at den respektive fargen skulle vises.

Komponenter i oversiktsnoden	Antall
ESP32 mikrokontroller	1
LED rød	2
LED grønn	2
Breadboards	2
Pushbuttons	1

ESP32 mikrokontrolleren gir kun ut 3.3 volt i motsetning til Arduino UNO sine 5 volt. Vi valgte derfor å ikke benytte resistorer på oversiktsnoden slik at LED-lysene skulle bli litt sterkere.

Tabellen nedenfor viser komponentene vi brukte til å simulere de andre nodene med LED-lys på tavlen.

Komponenter for de simulerte lysene	Antall
Arduino UNO	2
LED rød	8
LED grønn	2
Breadboards	2

## 4.2 Gjennomgang av koden

I vår tekniske løsning har vi brukt to ESP 32 mikrokontroller og to Arduino Uno- brett. De to ESP 32 mikrokontrollere utgjør hovedartefaktene: én node utenfor arbeidsrommet og én node på arbeidspulten. De to Arduino uno-brettene vi har brukt, er for å sette opp de andre LED-lysene slik at prototypen i større grad iscenesetter og simulerer den tiltenkte funksjonen ved en videreutvikling av prototypen. Disse brettene er satt opp med en enkel kode der LED-lysene er output og koplet til en digital pin på brettet. Disse arduinobrettene er derfor ikke koplet opp på det trådløse nettverket og har derfor ikke de samme funksjonene som er implementert i de to andre ESP 32 brettene. På bakgrunn av dette vil gjennomgangen av koden belyse ESP 32 -brettene oppsett av kode.

Koden vi har laget til artefaktene er en videreutvikling av koden vi fant i Arduino IDE sin eksempelfil «ESPNOW\_MultiSlave\_Master» og «ESPNOW\_MultiSlave\_Slave». I videreutviklingen brukte vi flere Youtube-videoer som forklarte hvordan oppsettet av kommunikasjonen mellom to enheter kan skrives. Vi brukte spesielt videoen «Arduino IDE + ESP32 | Communication between ESP32 (Two or more ESP32) with ESP-NOW Protocol» fra Uteh Str som en inspirasjon til hvordan vi kunne utforme koden (Str, 2021). I tillegg brukte vi UiO ChatGPT når vi stod fast i feilsøkingen og trengte ytterligere forklaring på hva enkelte metoder fra de importerte bibliotekene innebar. Videre brukte vi ESP-NOW dokumentasjonen fra Espressif for å forstå syntaksen til bibliotekene og innebygde funksjoner vi måtte benytte oss av (Espressif).

#### 4.2.1 Kodelikheter i Node A- oversiktsnode og Node B- pultnoden

I begge kodefilene starter vi med å implementere vi bibliotek `<esp_now.h>` og `<WiFi.h>`. Begge bibliotekene spiller en viktig rolle i å muliggjøre trådløs kommunikasjon mellom enhetene ved å bruke ESP-NOW- protokollen og koble enhetene til et WiFi-nettverk. Med `<esp_now.h>`- biblioteket sørger koden for at enhetene kan kommunisere over WiFi uten en ruter og muliggjør en toveis kommunikasjon mellom noder (Santos, u.d.). Med `<WiFi.h>`-biblioteket kan man implementere funksjoner for å koble opp en enhet som en klient til et nettverk (Santos, u.d.).

Koden til node a og node b starter med å opprette en datatype `«esp_now_peer_info_t»` som oppbevarer informasjonen om `«nodePult»` og `«nodeOversikt»`. Videre valgte vi å sette kanalen nodene opererer på til å være konstant og på kanal 6 i og med at det var denne kanal som var tilgjengelig på vårt test-nettverk. Her kunne man valgt å gjøre koden mer dynamisk, men i og med at vi ønsket å oppnå en mer stabil og pålitelig kommunikasjon mellom nodene beholdt vi denne konstante variabelen.

I tillegg til å sette opp variabler til LED-lysene og knappene, satte vi opp noen boolske variabler som vi kunne bruke for å kontrollere om koplingen mellom nodene var vellykket og at de hadde kontakt: `«bool erParet = false;»` og `«bool nodeFunnet = false;»`. Disse opprettet vi også på bakgrunn av noen erfaringer vi gjorde tidligere i prosessen når vi ikke klarte å opprette kontakt mellom nodene. På denne måten kunne vi jevnlig skrive ut verdien i Serial Monitor verdien slik at vi lettere kunne lokalisere feil i koden.

I koden vår har vi to metoder for å kunne finne noden man ønsker å koble seg sammen med, `«void setupNode()»` og `«void sjekkNode()»`. Metoden `setupNode()` kan vi bruke når vi viste MAC-adressen til noden vi ønsket å kople noden vår med, mens `sjekkNode()` ville lete etter en node ved å skanne det tilgjengelige nettverket og identifisere nodens SSID for deretter å bruke denne nodens MAC-adresse for å opprette en kontakt. Vi valgte å holde på begge metodene i og med at vi ville gjøre koden vår fleksibel til å kunne finne noden uavhengig oppgitt MAC-adresse. I vårt tilfelle brukte vi som oftest noder der vi hadde nodens MAC-adresse.

I metoden `«configDeviceSTA()»` sørger man for at nodene kopler seg opp på et oppgitt eksisterende trådløst nettverk i «Station Mode», og dermed kan kommunisere med hverandre over dette nettverket. I vårt tilfelle ønsket vi at begge nodene skulle kunne sende og motta data til og fra hverandre over nettverket.

Begge nodene våre skulle kunne sende signaler til hverandre som vi ønsket at de skulle handle etter. Funksjonen `«void sendData(int verdi)»` er implementert i begge ESP 32-kontrollene våre. I Node A sender den ringeklokkens verdi, og i Node B sender den verdiene 1 eller 2, 1 om grønn knapp er trykket på og 2 om rød knapp er trykket på.



Funksjonen «**void OnDataSent(const uint8\_t \*mac\_addr, esp\_now\_send\_status\_t status)**» er en callback funksjon som gir oss en oversikt over hvilke data det er som har blitt sendt til en den andre noden etter at dataoverføringen er fullført. Funksjonen printer i tillegg ut sendestatusen til pakken, om sendingen var vellykket eller feilet. Denne funksjonen har vi brukt aktivt for å kunne se at riktig data ble sendt og for å få informasjon om eventuelle feil eller problemer ved sendingen.

Begge nodene våre har også en «**setup()**»-metode som konfigurerer ESP-kontrollene. I denne metoden starter man en seriell kommunikasjon med en baud rate på 115200, setter WiFi til station, skriver ut navnet på prosjektet, kaller på en funksjon som konfigurerer Wi Fi, initialiserer ESP-NOW protokollen, setter opp noden gjennom **setupNode()**-metoden, registrere en callback-funksjon som kalles når data er sendt og en callback- funksjon som kalles på når data er mottatt. I tillegg settes de ulike variablene til LED og knapper til å være output og input\_pullup.

#### 4.2.2 Node A – oversiktsnode

I node a er det enkelte kodelinjer som naturlig nok skiller seg fra node b. I node a har vi blant annet satt opp to grønne og to røde LED-lys, slik at vi har et lys til hver side av artefakten. I callback funksjonen «**void OnDataRecv(const uint8\_t \*mac\_addr, const uint8\_t \*data, int data\_len)**», får vi oversikt over hvilke data som er mottatt hos noden og hvor dataen kommer fra. Videre implementerer funksjonen en if-sjekk som behandler dataen som er sendt inn. Om dataen som er sendt inn tilsvarer verdien int 2 vil den gjøre at de to røde lysene vil skrus på, og de grønne lysene blir slått av. Om dataen derimot er 1 vil de røde lysene bli slått av og de grønne lysene bli slått på. På denne måten sørger noden for å oppdatere sine lys som både kan leses fra innsiden og utsiden av arbeidsrommet.

Hovedprogrammet i «**void loop()**» gjør først en sjekk om noden er koplet opp på nettverket til riktig kanal og at den er parett med peer-noden. Videre leser den av tilstanden til knappen og setter dette til ringeklokkeNyttTrykk, og deretter sjekker om knappen er trykket eller ikke før den oppdaterer tilstanden til knappen. Om knappen er trykket på, sender noden dataen til node b.

#### 4.2.3 Node B – pultnode

I callback funksjonen «**void OnDataRecv(const uint8\_t \*mac\_addr, const uint8\_t \*data, int len)**» til node b gjøres det også en if-sjekk for å kunne behandle dataene som sendes inn på riktig måte. If-sjekken her sjekker om dataen som er sendt inn er 1 og om string-objektet «status» er satt til grønn. Om status er satt til «grønn» betyr det at brukeren på arbeidsrommet er satt som tilgjengelig, noe som også betyr at det er tillatt for personen utenfor arbeidsrommet å ringe på til personen. Om if-sjekken resulterer i true, blir metoden «**settBlaa()**» kalt på. Om brukeren på arbeidsrommet derimot er satt til å være utilgjengelig, vil det ikke være mulig for eksterne personer å ringe på brukeren.

Node b, har flere metoder som vil endre nodens eget RGB-lys. I «**void settBlaa()**» setter vi rødt og grønt lys til high for å slå av disse anodespenningsnivåene, og vi setter blått lys til low for å skru på

den tilknyttede lysdioden. I denne metoden setter vi også opp en variabel som lagrer nåværende tid før programmet deretter kjører en while-løkke som varer i fem sekunder. I løkken vil det blå lyset bli slått av og på slik at det blinker i fem sekunder. Når de fem sekundene har gått kaller metoden enten på metoden «**settGronn()**» eller «**settRod()**» gjennom en if-sjekk. Dette gjør at lyset som tidligere blinket blått vil bli satt tilbake til sin opprinnelige status.

Hovedprogrammet i «**void loop()**» gjør, på lik linje med node a, ulike sjekker som sørger for at noden er koplet til et trådløst nettverk og koplet til den andre noden. Om digitalRead får signaler om at den grønne knappen er trykket på, vil programmet først kalle på «**sendData()**»-funksjonen med parameterverdien 1, deretter vil programmet kalle på «**settGronn()**»-metoden. I «**settGronn()**» metoden vil RGB-lyset settes til å lyse grønt, altså brukeren er satt som tilgjengelig. Om digitalRead derimot får signaler om at den røde knappen er trykket på, vil programmet kalle på «**sendData()**»-funksjonen med int 2 som parameter, deretter kaller programmet på «**settRod()**»-metoden. I denne metoden blir RGB-lyset satt til å lyse rødt, altså brukeren er satt til å være utilgjengelig. Om det derimot viser seg at noden ikke er koplet opp mot oppnevnt kanal vil det printes ut en melding med «venter på at noden skal bli funnet». I slutten av loopen har vi satt inn en delay på 100, slik at vi har kontroll over signalene og gir enhetene noe tid til å håndtere dataene.

## 5. Avslutning

I denne rapporten har vi presentert vår tekniske løsning på utfordringene med å redusere forstyrrelser i et åpent kontorlandskap, med utgangspunkt i det overordnede temaet: interaksjon uten skjerm og av/på-funksjonalitet. Ved hjelp av ESP 32-mikrokontroller har vi utviklet et brukervennlig system som signaliserer tilgjengelighet og utilgjengelig for både de ansatte og eksterne personer som ønsker arbeidsrommet.

Gjennom prosjektet har vi aktivt involvert brukerne gjennom intervjuer, workshops og evalueringer. Dette har sikret at vår løsning er godt tilpasset deres brukssituasjon og arbeidsmiljø. Den tette kontakten vi har hatt med brukerne har også sørget for både gjensidig læring og samskaping i flere ledd av prosjektet. På denne måten har vi fått et innblikk i brukernes arbeidshverdag og behov, samtidig som brukerne har lært om de tekniske mulighetene Arduino kan tilby. Brukernes verdifulle innsikter som har formet designet og funksjonaliteten til dette systemet.

Prosjektet har gitt oss verdifull erfaring med prosessarbeid der hovedfokus er brukerinvolvering og tangible interaction. Ved dette prosjektets ende har vi oppdaget flere muligheter for videreutvikling, som å inkludere flere funksjonaliteter og mer utforskning av «look and feel». I et større kontorlandskap ville det vært hensiktsmessig å inkludere flere noder og integrere ytterligere sensorer for bedre å kunne redusere forstyrrelser og utvide interaksjonsmulighetene.

## Referanser

Espressif. (u.d.). *ESP-NOW*. Hentet fra ESP-IDF Programming Guide:

[https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp\\_now.html](https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html)

Lilleeng, S. (2023, juli 4). *Åpne kontorlandskap øker risikoen for sjukemeldinger*. Hentet fra NRK:

[https://www.nrk.no/trondelag/arbeidsplassen\\_-flere-sjukemeldinger-seg-i-apne-kontorlandskap-e-nn-pa-eget-kontor-1.16455510](https://www.nrk.no/trondelag/arbeidsplassen_-flere-sjukemeldinger-seg-i-apne-kontorlandskap-e-nn-pa-eget-kontor-1.16455510)

Santos, R. (u.d.). *Getting Started with ESP-NOW (ESP32 with Arduino IDE)*. Hentet fra Random nerds tutorials: <https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/>

Str, U. (2021, Juli 14). *Arduino IDE + ESP32|Communication between ESP32 (Two or more ESP32) with ESP-NOW Protocol*. Hentet fra Youtube:  
<https://www.youtube.com/watch?v=xOLG-88Ld3A&t=541s>