



Høgskulen  
på Vestlandet

**BACHELOROPPGAVE:**  
**BO21E-26 - ABB: QR-kodebasert  
navigasjonssystem for mobile enheter**

**Monica Lunde**

**Ingrid Finnøy**

**Lise Pettersen**

Kommunikasjonssystemer

Institutt for datateknologi, elektroteknologi og realfag

Veileder Adis Hodzic

1.juni 2021

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 12-1.

## Dokumentkontroll

<i>Rapportens tittel:</i> QR-kodebasert navigasjonssystem for mobile enheter	<i>Dato/Versjon</i> 01.06.21/1.0
	<i>Rapportnummer:</i> B021E-26
<i>Forfatter(e):</i> Monica Lunde Ingrid Finnøy Lise Pettersen	<i>Studieretning:</i> HKOM18
	<i>Antall sider m/vedlegg</i> 40
<i>Høgskulens veileder:</i> Adis Hodzic	<i>Gradering:</i> Åpen
<i>Eventuelle Merknader:</i> Vi tillater at oppgaven kan publiseres.	

<i>Oppdragsgiver:</i> ABB	<i>Oppdragsgivers referanse:</i> Simon Hamilton
<i>Oppdragsgivers kontaktperson(er) (inkludert kontaktinformasjon):</i> Simon-John Hamilton: simon-john.hamilton@no.abb.com Ørjan Tveit: orjan.tveit@no.abb.com Kjell-Arne Gjerde: kjell-arne.gjerde@no.abb.com	

Revisjon	Dato	Status	Utført av
v.0.1	29/04/2021	Førsteutkast	Ingrid Finnøy Lise Pettersen Monica Lunde
v.0.2	20/05/2021	Andreutkast	Ingrid Finnøy Lise Pettersen Monica Lunde
v.1.0	01/06/2021	Ferdig rapport	Ingrid Finnøy Lise Pettersen Monica Lunde

## Forord

Denne rapporten dokumenterer arbeidet på prosjektet “QR-kodebasert navigasjonssystem for mobile enheter”. Prosjektet ble fullført våren 2021, og er den avsluttende oppgaven på bachelorstudiet *Kommunikasjonsingeniør ved Høgskulen på Vestlandet*. Prosjektet er gjennomført av Ingrid Finnøy, Lise Pettersen og Monica Lunde.

Prosjektet innebærer å utvikle et program som skal ta i mot en videostrøm fra et kamera, som går langs en skinne. Kameraet skal fange opp QR-koder med unike tag for forskjellige enheter som er registrert i klasser, og deretter gi en lokasjon på hvor enhetene befinner seg i rommet.

Prosjektgruppen vil gjerne takke ABB for denne unike muligheten, og for god oppfølging underveis i prosjektet. Dette prosjektet har gitt oss en verdifull kompetanse innenfor prosjekt, API og andre teknologier som vi kan ta med oss videre i arbeidslivet.

Prosjektgruppen vil også gi en stor takk til den engasjerte, interne veilederen Adis Hodzic ved Høgskulen på Vestlandet som har gitt gode råd og veiledning gjennom hele prosjektperioden.

-----

Ingrid Finnøy

-----

Monica Lunde

-----

Lise Pettersen

## Sammendrag

Bacheloroppgaven har som formål å lage et QR-kodebasert navigasjonssystem for mobile enheter som skal kunne brukes på blant annet ubemannede plattformer.

Ønsket til ABB gikk ut på å lage et program som kommuniserer med et kamera festet til en pod på skinne. Skinnen skal være festet i taket på et tenkt teknisk rom, der det står kabinetter som inneholder forskjellige enheter. Når en alarm går på en av enhetene, skal kameraet kunne forflytte seg automatisk til der alarmen har gått. Hver enhet vil ha en unik QR-kode som inneholder en tag, som skal brukes til identifikasjon.

Programmet kjører autonomt, og skal kunne observeres via et grafisk brukergrensesnitt. Programmet vil få inn en videostrøm, som skal kunne leses av og identifisere QR-koder. Ved hjelp av QR-kodene skal programmet kunne avgjøre hvilken enhet kameraet befinner seg ved. Når det befinner seg ved riktig QR-kode skal det sendes bekreftelse på posisjon. Dette er løst ved å lage programmet i programmeringsspråket C#.

Det som har vært viktigst å få plass i prosjektet er at programmet selv navigerer seg ut i fra QR-kodene, og henter ut nødvendig informasjon. Programmet oppfylder de ønskede kravene, samtidig som det er noe forbedringspotensiale for noen av funksjonene.

Prosjektet har totalt sett gått bra. Underveis har det dukket opp utfordringer som er blitt løst i løpet av prosjektperioden. Sluttproduktet tilfredsstillter prosjektets mål.

# Innhold

## Dokumentkontroll

## Forord

## Sammendrag

## Innhold

## Figurliste

<b>1. Innledning</b>	<b>1</b>
1.1 Oppdragsgiver	1
1.2 ABBs ønske	1
1.3 Problemstilling	2
<b>2. Kravspesifikasjon</b>	<b>2</b>
2.1 Tre nivåer av funksjonalitet	2
2.2 Autonomi	3
2.3 Posisjon av QR-kode	3
2.4 Vurdering av implementasjon mot Milestone VMS	3
2.5 API gitt av ABB	4
2.6 Lav forsinkelse	4
2.7 Alarmsimulator	5
<b>3. Analyse av problemet</b>	<b>5</b>
3.1 Funksjonalitet	5
3.2 Arkitektur	6
3.3 Teori om, og valg av løsninger	8
3.3.1 QR-kode	9
3.3.2 NSSM og API	10
3.3.3 C#	11
3.3.3.1 Bibliotek i C# Visual Studio	11
3.3.3.2 ZXing	11
3.3.3.3 AForge.NET	11
3.3.4 Dele opp oppgaven	12
3.5 Fysisk utstyr	12
3.5.1 Poden	12
3.5.2 Skinne	13
3.5.3 Mikrokontrolleren	14
3.5.4 DMX-boks	15
3.5.5 Webkamera	16

<b>4. Realisering av valgt løsning</b>	<b>17</b>
4.1 QR-kode klasser	18
4.2 Timer	18
4.3 Setup	19
4.4 Simulering av alarm	19
4.5 Skrive til, og lese fra tekstfil	20
<b>5. Testing av program og utstyr</b>	<b>20</b>
5.1 Refleksjoner fra lys	20
5.2 Tilkobling av mikrokontroller	21
5.3 Lagring av posisjon	21
5.4 Funksjonaliteten til kameraet	22
5.5 Krav til QR-kodene	22
5.6 Testing av alarm	23
5.7 Brukervennlighet	24
<b>6. Drøfting</b>	<b>24</b>
6.1 Styrker og begrensninger i prosjektet	24
6.2 Videre utvikling av programmet	25
<b>7. Konklusjon</b>	<b>26</b>
<b>Litteraturliste</b>	<b>27</b>
<b>Appendix A - Forkortelser og ordforklaringer</b>	<b>30</b>
<b>Appendix B - Prosjektforslag fra ABB</b>	<b>31</b>
<b>Appendix C - Fremdriftsplan</b>	<b>32</b>
<b>Appendix D - Forstudie</b>	<b>33</b>

## Figurliste

Figur 1a:	Logo til ABB	1
Figur 2a:	Kommandoer som API-et håndterer	4
Figur 3a:	Skisse av produktet utformet av ABB. Markerer oppgavens grensesnitt	6
Figur 3b:	Skisse av tiltenkt program (GUI)	7
Figur 3c:	Skjerm bilde av sluttproduktet (GUI)	7
Figur 3d:	Løsningsskisse	8
Figur 3e:	Skisse av komponentene brukt i prosjektet	9
Figur 3f:	QR-kode som viser bokstaven "A"	10
Figur 3g:	Poden på skinnen	13
Figur 3h:	Skinne hengt opp i kabelskinne	14
Figur 3i:	Mikrokontroller og skjerm	15
Figur 3j:	DMX-boks	16
Figur 3k:	Webkamera festet på poden	17
Figur 4a:	Oppsett av QR-koder	18
Figur 5a:	QR-kode som holdes opp ned	23

## 1. Innledning

I denne delen av rapporten vil prosjektgruppen presentere bakgrunnen for prosjektet. Det vil gi et innblikk i oppdragsgiver, bedriftens ønsker og problemstilling. Oppdragsgiver er bedriften ABB, heretter kun omtalt som ABB.

### 1.1 Oppdragsgiver

ABB er et stort internasjonalt teknologifirma med 110 000 ansatte i over 100 land (ABB, 2021a). De har virksomhet i hele Norge med 1900 ansatte, og har eksistert i Norge siden 1880-årene. Omsetningen i Norge i 2019 var på 7 milliarder (ABB, 2021b). ABB sier selv: «For ABB handler bærekraftig utvikling om å balansere økonomisk utvikling, miljøforbedringer og samfunnsansvar for å skape verdier for våre eiere, partnere og andre interessenter.» (ABB, 2021c). Oppdragsgiver har flere virksomhetsområder innenfor robotisering, elektrifisering, automatisering, motorer og omformere (ABB, 2021d). Gjennom årene har ABB bygget en spesialisert telekommunikasjonsavdeling. Denne konsentrerer seg om behov knyttet til segmenter innenfor off- og onshore, flytende naturgass (LNG) og rørledninger (ABB, 2021e).



*Figur 1a: Logo til ABB (ABB, 2021a).*

### 1.2 ABBs ønske

ABB ønsker at det skal lages et navigasjonssystem for mobile enheter. Navigasjonssystemet fungerer ved at et kamera skal kunne styres til en forhåndsdefinert posisjon. Dette ved hjelp av for eksempel rovere, droner eller kamera på skinneløsninger. Den mobile enheten, altså kameraet festet til en av de tre nevnte løsningene, skal kunne bekrefte posisjon ved hjelp av enten RFID, QR-kode eller radio. De vil at prosjektet skal fungere autonomt, uten behov for manuell styring eller fysisk tilstedeværelse. ABB ønsker at prosjektgruppen skal integrere videostrøm fra en mobil enhet mot for eksempel Milestone VMS, som



er en tjeneste for videoovervåkning. ABB ønsker hovedsakelig en skinneløsning og avlesning av QR-kode med så lite forsinkelser som mulig.

### 1.3 Problemstilling

Problemstillingen har som målsetting å lage et program som skal kommunisere med et kamera festet til en pod på skinne. Kommunikasjon mellom programmet og kameraet skal gjøres via et API. Når en alarm går av skal kameraet kunne forflytte seg automatisk til der alarmen har gått. Programmet skal få inn en videostrøm, som skal kunne leses av og identifisere QR-koder. For at programmet skal kunne bekrefte at kameraet befinner seg på riktig plass, skal det brukes QR-kode som inneholder en tag som identifikator. Det skal plasseres en unik QR-kode over hvert kabinett, samt nedover i kabinettet for hver enhet som kan utløse en alarm. Kameraet som er festet til poden vil kjøre bortover en skinne, samtidig skal programmet skanne hvert kabinett for QR-koder. Dermed skal programmet kunne sende bekreftelse på at kameraet befinner seg ved riktig enhet ved hjelp av QR-koden.

## 2. Kravspesifikasjon

Kravspesifikasjon gitt fra ABB omhandler løsning, utviklingsmetoder og verktøy. Det anses som viktig at disse kravspesifikasjonene blir fulgt for å oppnå et resultat med ønsket funksjonalitet og som lett kan videreutvikles. I punktene under er disse kravene beskrevet.

### 2.1 Tre nivåer av funksjonalitet

I prosjektbeskrivelsen er det gitt tre nivåer av funksjonalitet fra ABB: basis, utvidet og avansert. Punktene under hvert nivå beskriver hvilken funksjonalitet ABB forventer av prosjektgruppen. Meningen med dette er at basisfunksjonene er viktigst å få på plass. Dersom prosjektgruppen fikk til dette skulle det arbeides videre med utvidet funksjonalitet, og til slutt avansert funksjonalitet. Punktene under avansert funksjonalitet var tenkt som teknisk utfordrende for prosjektgruppen. Disse funksjonalitetene skulle implementeres dersom det ble tid, og de andre funksjonene var utført. I punktene under er disse tre nivåene beskrevet av ABB.

**Basis funksjoner:**

- Dekode QR-kode med å lese data fra webkameras videostrøm.
- Vise posisjon av QR-kode i bilde.
- Lage Windows applikasjon (GUI).

**Utvidet funksjonalitet:**

- Motta alarm-objekt fra tredjepartssystem, for å kunne styre kamera til riktig posisjon.
- Sende bekreftelse på at riktig tag er funnet.

**Avansert funksjonalitet:**

- Implementere letefunksjon for å finne ønsket QR-kode og posisjonere kamera slik at QR-kode kommer midt i bildet.
- Lagre kjente posisjoner for tag-nummer, for å raskere posisjonere kamera.
- Manuell kontroll av posisjon, for å kunne lagre kjente QR-kode posisjoner.

## 2.2 Autonomi

ABB har et krav om at programmet skal arbeide autonomt, i den forstand at det skal kunne kjøres på ubemannede plattformer eller lignende. Dette vil si at det skal kunne fungere og utføre handlinger uten menneskelig interaksjon.

## 2.3 Posisjon av QR-kode

Et viktig krav fra ABB er at posisjonen til de ulike QR-kodene, som blir avlest av kameraet festet til poden, skal kunne lagres. Slik at når en alarm går skal programmet raskt kunne finne tilbake til riktig enhet, ved hjelp av lagrede posisjoner. I sammenheng med dette er det også viktig at kameraet er i stand til å posisjonere seg slik at QR-koden sentreres midt i bildet.

## 2.4 Vurdering av implementasjon mot Milestone VMS

Kravspesifikasjon for kommunikasjon med overliggende systemer innebærer at programmet skal kunne utveksle informasjon med andre programmer. I dette tilfellet var det snakk om Milestone VMS. Siden prosjektgruppen ikke har tilgang til dette programmets kode, ble det bestemt at prosjektet skulle utvikles uten å ta videre hensyn til Milestone VMS.

## 2.5 API gitt av ABB

API står for “application programming interface” (Rossen, 2020a). Et API er en form for et mellomledd eller en kontaktflate mellom ulike programmer eller tjenester (Rossen, 2020a). Tanken med bruk av API i dette prosjektet er at API-et skal håndtere kommunikasjonen mellom programmet og motoren som kjører poden på skinnen. På denne måten slipper prosjektgruppen å ha kjennskap til programkoden som styrer motoren. Det er tilstrekkelig å vite hvilke kommandoer som kan håndteres av API-et gitt av ABB, heretter omtalt som API-et. I figur 2a vises kommandoene som kan brukes. Hensikten med tallene bak kommandoene er å justere intensiteten til signalet (Philips, 2008, s. 6). Intensiteten styrer blant annet hastigheten til motoren.

```
//MOTORPWR      0-1      0: Trackrunner not allowed to move
//              1: Trackrunner allowed to move
//RESETBCK      0-255    Reset backwards. 255 resets backwards at maximum speed
//RESETFWD      0-255    Reset forwards. 255 resets forwards at maximum speed
//MOTORBCK      0-255    Trackrunner runs manually backwards. Limits are not overwritten.
//MOTORFWD      0-255    Trackrunner runs manually forwards. Limits are not overwritten.
//MOTORPOS      0-9999   Trackrunner position. 9999 is all the way to the forwards limit
//MOTORSPD      0-255    Maximum speed of trackrunner. 255 is max.
```

*Figur 2a: Kommandoer som API-et håndterer.*

API-et gitt av ABB kjøres ved hjelp av windows service. Dette vil si at det kontinuerlig kjører i bakgrunnen og utfører programkode. ABB har et krav om at API-et skal brukes, fordi videre utvikling og bruk av programmet avhenger av dette.

## 2.6 Lav forsinkelse

ABB har satt krav om at programmet skal kjøre med minst mulig forsinkelse. Prosjektgruppens mulighet til å påvirke dette har vært gjennom å sørge for lagring av posisjonene til QR-kodene. Slik vil programmet slippe å bruke tid på å søke seg frem til en QR-kode når alarmer går, men kan bevege seg ut i fra lagrede posisjoner.

## 2.7 Alarmsimulator

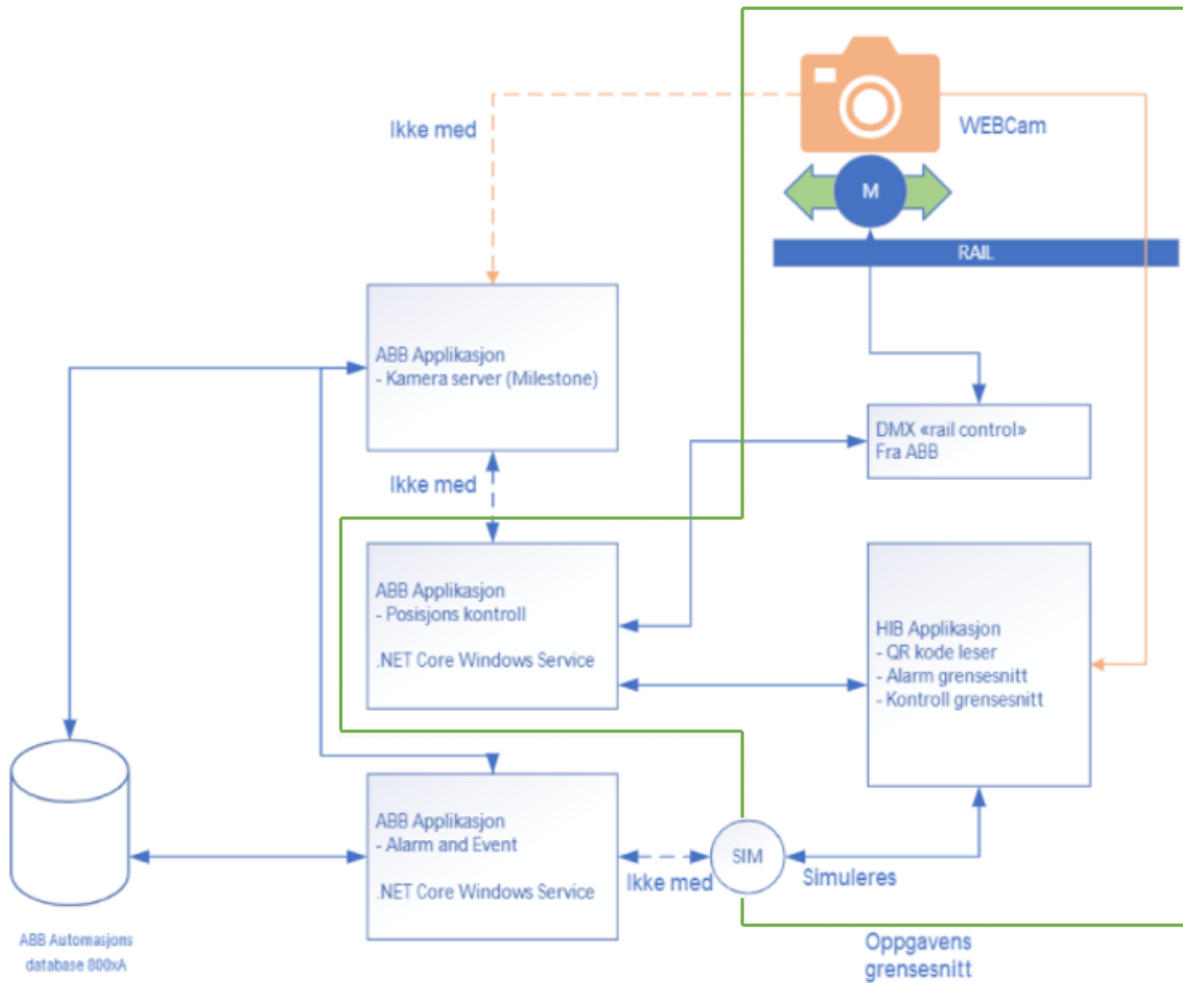
Prosjektgruppen har ikke tilgang til Milestone VMS. Optimalt ville en alarm kommet fra dette programmet. For å omgå dette problemet har ABB satt krav om at alarm skal simuleres, på valgfri måte.

## 3. Analyse av problemet

Prosjektgruppen, i samråd med ABB, har diskutert kravspesifikasjonene, brukergrensesnitt, utvikling og arkitektur av oppgaven. Dette var viktig for å få et overblikk over hva som måtte kartlegges i forbindelse med forstudiet og fremdriftsplanen. På denne måten ble det tydelig hvilke utfordringer prosjektgruppen måtte ta høyde for og hvilke deler av oppgaven som ville kreve ekstra ressursbruk.

### 3.1 Funksjonalitet

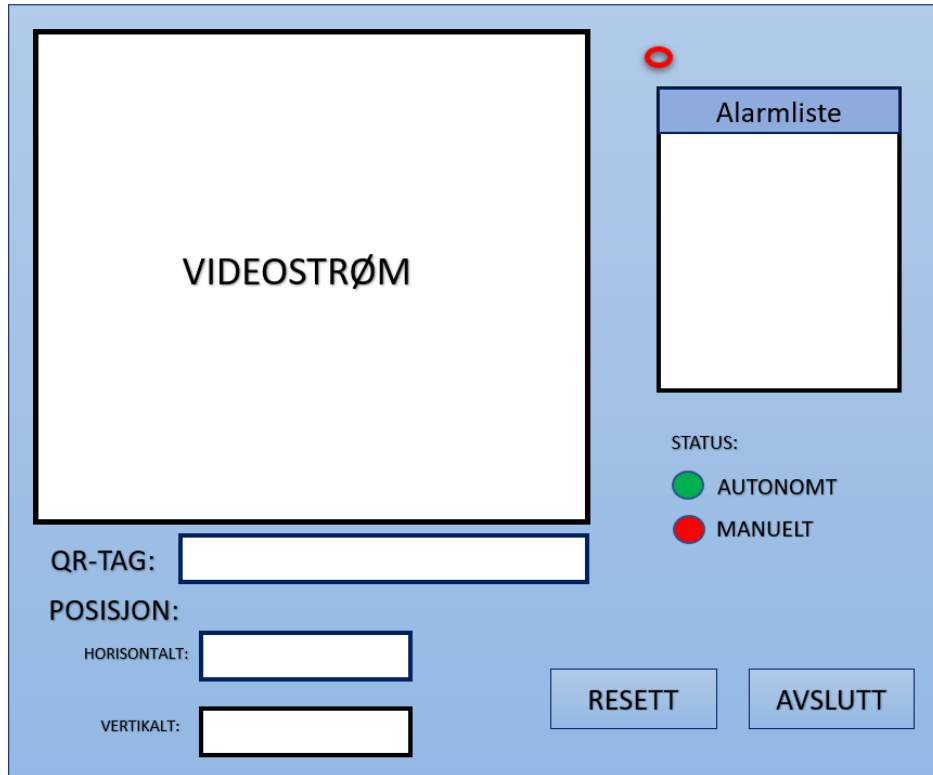
Figur 3a viser en skisse utformet av ABB. Denne viser hvordan sluttproduktet er tenkt og hva oppgavens grensesnitt er. Skissen har gitt oss en indikasjon på hva som må prioriteres, og brukes som en del av fremdriftsplanen. Figuren under har rammet inn hvilke funksjonaliteter som inngår i prosjektets grensesnitt. Prosjektgruppen skal konsentrere seg om applikasjonen, simulering av alarm og kommunikasjon mellom enhetene. ABB bruker ordet applikasjon, heretter bruker prosjektgruppen ordet program. Når funksjonalitetene er på plass vil det være et autonomt program som kan selges til ABBs kunder innenfor olje og gass. Derfor er det viktig at produktet har god funksjonalitet når det kommer til autonomi, og at programmet kan navigere seg ut i fra QR-kodene uten menneskelig interaksjon.



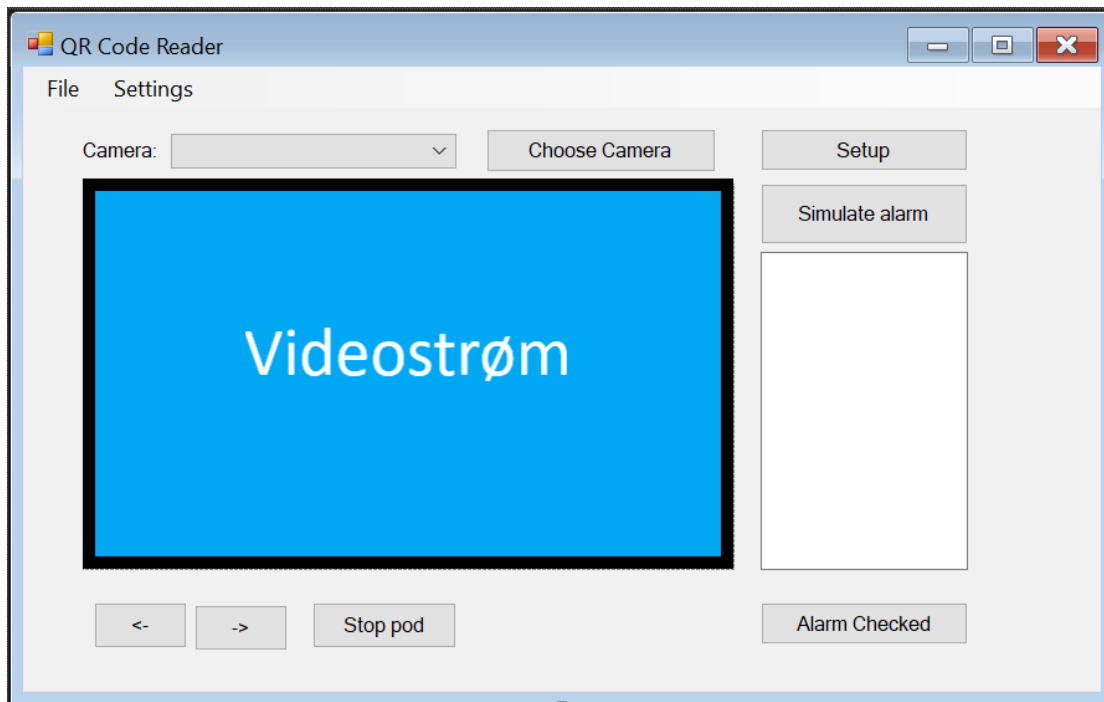
Figur 3a: Skisse av produktet utformet av ABB. Markerer oppgavens grensesnitt.

### 3.2 Arkitektur

Figur 3b er utformet av prosjektgruppen, i samråd med ABB, og viser hvordan programmets grafiske brukergrensesnitt (GUI) var planlagt å se ut. Figur 3c viser hvordan GUI til sluttproduktet er utformet. Prosjektgruppen har gjort egne valg og tilpasninger ut fra hva kravspesifikasjonene tilsier. Dette blir løst ved å lage et program i programmeringsspråket C# ved hjelp av Visual Studio. I kapittel 4, Realisering av valgt løsning, blir det lagt frem en mer detaljert beskrivelse av design rundt programmet og resterende valgte løsninger rundt arkitektur av prosjektet.

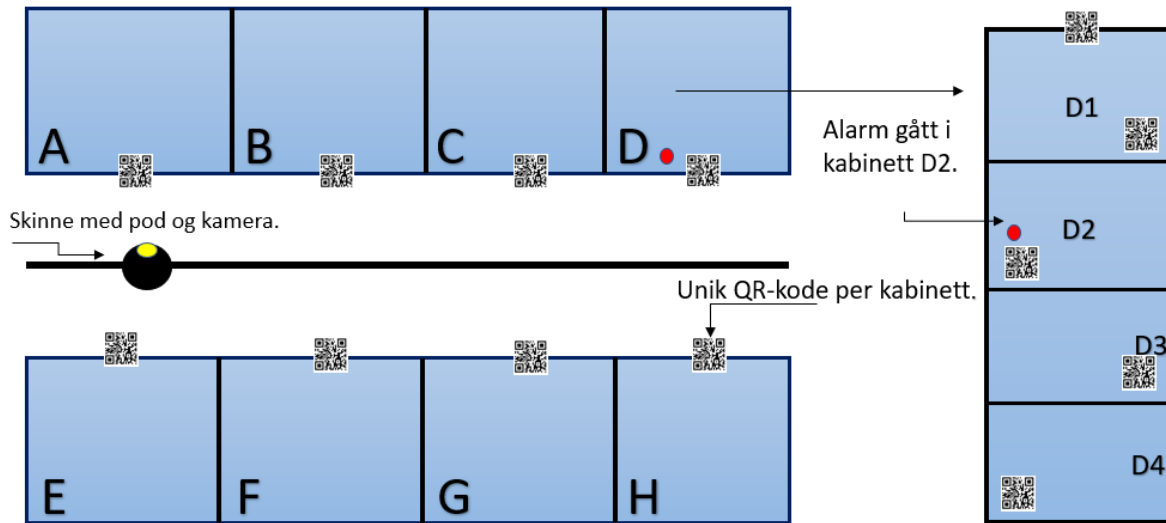


Figur 3b: Skisse av tiltenkt program (GUI).



Figur 3c: Skjerm bilde av sluttproduktet (GUI).

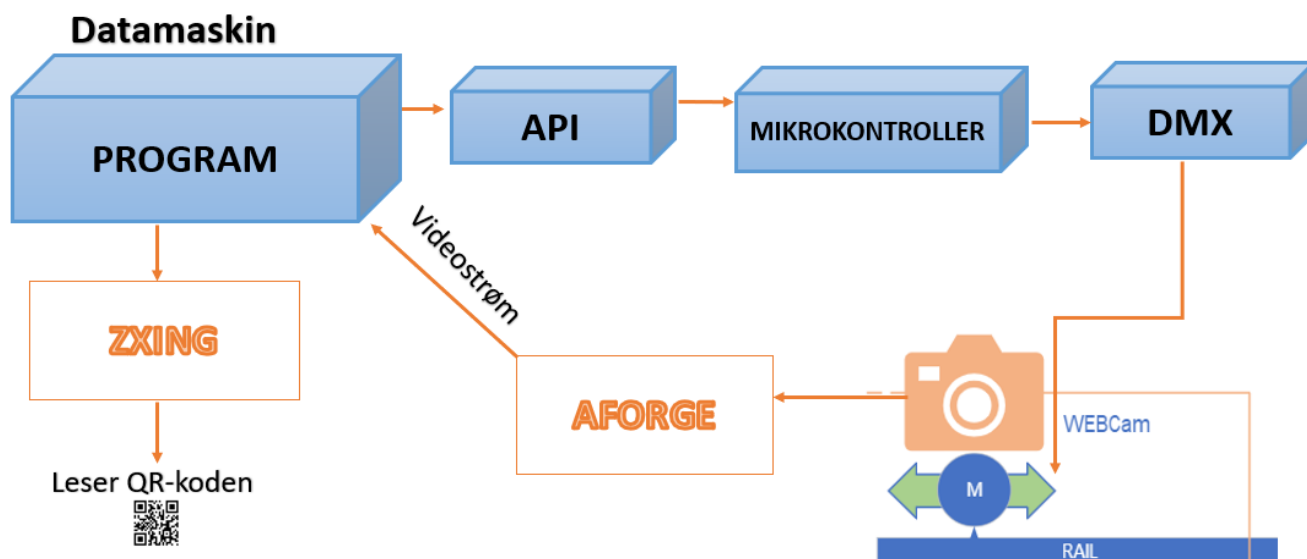
For å få en tydelig felles forståelse av hva som skulle være programmets funksjonalitet ble en løsningsskisse utformet (figur 3d). Denne viser tanken bak oppsettet med at ulike enheter er plassert i ulike kabinett, og at hvert kabinett er sortert med en egen overordnet QR-kode. Den viser også at alle enhetene i et kabinett har en unik QR-kode, og hvordan skinne med pod og kamera plasseres i forhold til kabinettene.



Figur 3d: Løsningsskisse.

### 3.3 Teori om, og valg av løsninger

Det er tatt utgangspunkt i skissen av oppgavens grensesnitt (figur 3a). Ut i fra denne har prosjektgruppen valgt å utvikle produktet med komponentene vist i figur 3e. Prosjektgruppen har undersøkt funksjonaliteten til de utvalgte komponentene og hvordan de skal brukes videre i prosjektet.



Figur 3e: Skisse av komponentene brukt i prosjektet.

### 3.3.1 QR-kode

Prosjektet hadde flere muligheter for løsning. Det ble utlevert tre kommunikasjonsmuligheter: RFID, QR-kode eller radio. Valget falt på QR-kode, siden ABB hadde størst ønske om dette.

QR står for “quick response” og er en type kode som fremstilles ved hjelp av en matrise av lyse og mørke piksler (Hayes, 2020). QR-kode er en forbedring av tradisjonelle strekkoder. En av fordelene med QR-koder er at de kan lagre mye mer informasjon enn strekkoder (Beaconstac, 2021). Pikslene i en QR-kode representerer siffer og bokstaver (TechTerms, 2015). Eksempel på hvordan en QR-kode ser ut, vises i figur 3f. Ved hjelp av et kamera, som har tilhørende program, kan pikslene i koden konverteres. QR-koder finnes i flere versjoner og varianter som er tilpasset forskjellige formål. De kan for eksempel brukes til å åpne en nettside eller inneholde bildegallerier, karthenvisninger og tekst.





*Figur 3f: QR-kode som viser bokstaven "A".*

I dette prosjektet skal QR-kode brukes til å bekrefte fysisk posisjon til et kamera på skinner. Flere koder vil bli brukt til å lage et navigasjonssystem slik at kameraet finner frem til ønsket posisjon, og kan bekrefte denne.

### 3.3.2 NSSM og API

NSSM står for "Non-Sucking Service Manager" og behandler windows servicer som kjøres i bakgrunnen av et program (Patterson, 2017). Ved bruk av NSSM blir servicer mer pålitelig, ved at hvis et program er dødt eller har feil vil servicen starte programmet på nytt (Patterson, 2017). Dette gjør det svært fordelaktig å benytte denne service-manageren. Prosjektgruppen ble anbefalt å bruke NSSM til å kjøre API-et av ABB.

Siden prosjektgruppen ikke har erfaring med bruk av API og NSSM fra før, var dette en potensiell utfordring. ABB hadde som krav å bruke API i oppgaven, derfor har de hjulpet prosjektgruppen med hvordan dette kan implementeres i programmet.

### 3.3.3 C#

Prosjektgruppen har valgt å bruke programmeringsspråket C# i Visual Studio, siden prosjektgruppen har mest erfaring med dette. ABB hadde ingen preferanse på hvilket programmeringsspråk som skulle benyttes i prosjektet. "C Sharp" er utviklet av Microsoft og er et objektorientert programmeringsspråk som benytter klasser og interface som objekter i programmeringen (Rossen, 2020b).

#### 3.3.3.1 Bibliotek i C# Visual Studio

Bibliotek er forhåndsskrevet programmeringskode av andre utviklere (Redhat Inc, 2021). Dette kan implementeres og brukes i egen kode for å slippe å lage all funksjonalitet selv. I dette prosjektet implementeres biblioteker til bruk av videostrømbehandling og avlesning av QR-koder.

#### 3.3.3.2 ZXing

Et av bibliotekene prosjektgruppen har benyttet er ZXing som behandler bilder som strekkoder, i dette prosjektet brukes det for å håndtere QR-kode i programmet (GitHub Inc, 2021). ZXing uttales "zebra crossing" og er en åpen kildekode (GitHub Inc, 2021). Dette er et bildebehandlingsbibliotek for strekkoder implementert i Java, og kan også brukes i andre programmeringsspråk (GitHub Inc, 2021).

Programmet håndterer en innkommende videostrøm. ZXing sørger for å ta bilder av videostrømmen og skanne bildene for QR-koder. Dersom en QR-kode blir funnet, dekker ZXing informasjonen som finnes i denne.

#### 3.3.3.3 AForge.NET

AForge.NET består av et rammeverk av flere biblioteker, i C#, for digital lesing av bilder og kunstig intelligens (Kirillov, 2016). Biblioteket som er relevant i denne oppgaven omfatter bildebehandling.

Prosjektgruppen benytter AForge.Video. I programmet skal AForge.Video brukes til å behandle videodata, for å gi tilgang til direkte videostrøm fra webkamera.

### 3.3.4 Dele opp oppgaven

For å håndtere kompleksiteten av prosjektet har prosjektgruppen valgt å dele oppgaven opp i mindre moduler. Moduleringen vil være i prioritert rekkefølge, basert på ABB sin inndeling av nivåer.

1. Motta videostrøm.
2. Håndtere videostrøm, samt vise denne i GUI.
3. Lese QR-kode.
4. Hente ut informasjon fra QR-kode.
5. Beregne posisjon til kamera.
6. Lagre posisjon.

### 3.5 Fysisk utstyr

Utstyret som blir brukt i prosjektet har ABB valgt å bestille fra sin leverandør Wahlberg i Danmark. Prosjektgruppen fikk tilsendt en skinne, pod, DMX og mikrokontroller. ABB var også behjelpelige med å bestille et godt webkamera.

#### 3.5.1 Poden

Poden består av en motor med hjul som driver den fremover og bakover på en skinne (figur 3g). Gjennom kommandoer bestemmes hastigheten og retningen til poden, og motoren kan skrus av og på.



*Figur 3g: Poden på skinnen.*

### 3.5.2 Skinne

Skinnen består av to delskinner som er skrudd sammen, slik at den blir ca. seks meter lang.

Prosjektgruppen vurderte å henge skinnen opp med strips. Etter diskusjon om tyngde på skinnen og hvordan ustabilitet kunne påvirke avlesing av QR-koder, ble det valgt å gå bort fra dette. Derfor besluttet prosjektgruppen å montere skinnen med bolter, skive og låsemutter for å få mest mulig stabilitet.

Skinnen er hengt opp i en kabelskinne i taket, der poden skal kjøre frem og tilbake. Det har ikke vært noe problem for poden å kjøre over skjøten mellom de to delskinnene.

I hver ende av skinnen er det montert fast bremseklosser. Når poden kommer til en bremsekloss vil det trykkes inn en knapp på poden, slik at den stopper. Bremseklossene skal sikre at poden ikke kan kjøre ut av skinnen.



*Figur 3h: Skinne hengt opp i kabelskinne.*

### 3.5.3 Mikrokontrolleren

I prosjektet brukes en mikrokontroller som er koblet til en liten skjerm (figur 3i). Skjermen benyttes til å vise hvilken IP-adresse og hvilket portnummer som kan benyttes for å koble til en datamaskin som kjører programmet. Programmet sender kommandoer til mikrokontrolleren ved hjelp av API-et.

Mikrokontrolleren sørger for videre kommunikasjon med DMX-boks.



Figur 3i: Mikrokontroller og skjerm.

### 3.5.4 DMX-boks

I dette prosjektet er det brukt en DMX-boks. DMX står for “digital multiplex” (Aleksandersen, 2018). DMX-boksen er en digital nettverkskontroller som styrer motoren på poden. Den brukes for å utføre kommandoene som blir sendt fra programmet (Aleksandersen, 2018). DMX-boksen styrer intensiteten på signalene (Philips, 2008, s. 6). Et eksempel er når poden får kommando om å kjøre fremover, forhåndsinnstilles hastigheten ved å sende et tall mellom 0 og 255.



Figur 3j: DMX-boks.

### 3.5.5 Webkamera

Prosjektgruppen, i samråd med ABB, kom frem til å bruke Logitech BRIO 4K STREAM Edition Webkamera. Kameraet har 4k (3840 x 2160 piksler) oppløsning, og gir derfor en jevn, høy kvalitet på videostrømmen (Logitech, 2021). Ved å bruke et kamera med så god oppløsning tok prosjektgruppen høyde for at man kunne lese QR-kode på lengre avstander.

I dette prosjektet brukes det et webkamera. Sluttproduktet er tiltenkt å benytte seg av et overvåkingskamera med PTZ(Pan-tilt-zoom) (Canon, 2021). PTZ har en annen funksjonalitet ved at kameraet beveger seg og panorerer horisontalt i bilde, tilter vertikalt i bildet og zoomer inn med høy oppløsning (Canon, 2021). Med en maskinvarekontroller som kan styre kameraet vil man kunne kontrollere et område uten fysisk tilstedeværelse. Prosjektgruppen kom frem til, i samråd med ABB, at Logitech webkamera var et godt utgangspunkt for dette prosjektet.

Prosjektgruppen har festet kameraet med borrelåstape til poden slik at det lett kan tas av når det ikke er i bruk.



*Figur 3k: Webkamera festet på poden.*

#### 4. Realisering av valgt løsning

ABB har en lab i Oslo som prosjektgruppen kunne ha benyttet seg av, men på grunn av koronasituasjonen lot det seg ikke gjøre. Derfor fikk prosjektgruppen tilsendt utstyr til Bergen. Utstyret besto av en skinne, mikrokontroller og pod fra Wahlberg i Danmark.



## 4.1 QR-kode klasser

Prosjektgruppen har valgt å lage klasser for QR-koder. Der man har en klasse for QR-kodene som tilhører hvert kabinett, og en klasse for QR-kodene til hver enhet inne i kabinettene. Dette illustreres i figur 4a. Her kan man se at hvert skap har en overordnet QR-kode, markert med rødt. Enhetene i kabinettene har hver sin unike QR-kode, markert med blått.



Figur 4a: Oppsett av QR-koder

Klassen til QR-kodene inne i kabinettene arver innhold fra klassen til kabinettene. Klassene brukes for å kunne opprette et objekt som refererer til en enkelt QR-kode, og dermed lagre posisjon og tag. Dermed kan objektene lett hentes opp igjen etter de er lagret.

## 4.2 Timer

Prosjektgruppen valgte å bruke en timer som kjører kontinuerlig så lenge programmet er i gang. Timeren fungerer som en klokke som tikker. Hyppigheten på tikkingen kan endres. I dette programmet er timeren innstilt til å tikke en gang hvert sekund. For hver gang timeren tikker, vil det tas et bilde av videostrømmen og bildet skannes for QR-koder. Bibliotekene ZXing og Aforge benyttes i denne prosessen.

### 4.3 Setup

Oppstartsprosessen setup kan kjøres første gang programmet starter, hver gang det skjer endringer i hvilke QR-koder som er i bruk, eller en endring i deres plassering. Setup kan startes manuelt ved behov. Denne prosessen vil sette motoren til poden i gang med lav fart og programmet vil lete etter QR-koder. Om programmet leser et kabinett, for eksempel "A", skal denne QR-koden lagres som et objekt. Motoren skal fortsette å kjøre til QR-koden er sentrert i bildet horisontalt. Dette sikrer riktig lagring av posisjon.

Når dette er utført skal bildet tiltes nedover for å lete etter QR-koder inne i kabinettet. Hvis det oppdages en QR-kode som er ukjent, skal denne lagres. Om QR-koden er kjent fra før skal denne ignoreres. Bildet fortsetter å tilte ned, til det når bunnen av kabinettet. Prosessen videre er at bildet tiltes opp igjen og motoren settes i gang for å lete etter neste kabinett. Dette pågår til programmet ikke har funnet en ny QR-kode innen fem minutter. Dermed anses setup som ferdig og poden kjører tilbake til startposisjon, i enden av skinnen.

### 4.4 Simulering av alarm

Simulering av alarm skjer i egen tråd. Forskjellige tråder i programmering vil gi muligheten til å kjøre forskjellige programkoder samtidig. I dette tilfellet for at GUI-programmet skal fortsette å være responsivt og ikke fryse når programkoden kjører. Simulering av alarm skjer ved at det velges et tilfeldig objekt i listen av QR-koder som er lagret. Dermed hentes det ut informasjon om tagen til det tilfeldige objektet. Denne tagen representerer en enhet, og blir brukt for å simulere at en alarm har gått.

Når en alarm går henter programmet ut informasjon om den horisontale posisjonen til enheten alarmen går ved. Denne informasjonen hentes fra det objektet som er lagret med samme tag som alarmens tag. Det blir gitt kommando til motoren at den skal sette i gang og gå til den når posisjonen som er lagret for enheten. Når poden har funnet frem til riktig horisontal posisjon, begynner webkamera å tilte nedover til den får treff på riktig QR-kode. Dette vil bekreftes ved at programmet viser en melding om at alarm er funnet. Hvis den ikke finner enheten der alarmen går, vil brukeren få melding om dette.

## 4.5 Skrive til, og lese fra tekstfil

Optimalt sett burde prosjektgruppen brukt databaser for å lagre QR-kodene med tag og tilhørende posisjon. Grunnet for lite kunnskap om databaser, og begrenset tid til å sette seg inn i emnet, ble det valgt å skrive til tekstfil i dette prosjektet. For at det ikke skal være behov for å kjøre setup hver gang programmet startes, blir det benyttet lagring til tekstfil. Hvis det er lagret en tekstfil som inneholder QR-kodenes tag og posisjon fra før, kan dette hentes ved hjelp av nedtrekksmeny i GUI.

Hvert objekt blir lagret som vanlig tekst i tekstfilen, med informasjon om tag og posisjon. Hvis bruker velger å hente objekter fra tekstfil, slettes først referanser til tidligere objekter. Deretter lagres det nye objekter med informasjonen som hentes i tekstfilen. Dette gjøres ved at linje for linje leses, og hver linje deles opp i mindre biter for å kunne skille mellom QR-kodenes tag og posisjon.

Et av problemene med å lagre informasjonen i tekstfil handler om sikkerheten til innholdet. Det er veldig lett å endre informasjonen i en slik fil. Dette kan føre til at programmet får problem med å lese tekstfilen eller at det blir feil informasjon knyttet til hver QR-kode.

## 5. Testing av program og utstyr

Testing av utstyret og koden til programmet har blitt gjort kontinuerlig, underveis i utviklingen. Under testing har det blitt oppdaget feil og svakheter som skyldes enkle programmeringsfeil. Disse har prosjektgruppen i stor grad vært i stand til å rette opp i. I tillegg har det vært oppdaget hindringer som skyldes at utstyret ikke er optimalt, eller som krever såpass omfattende koding at dette prosjektet ikke har hatt tilstrekkelig med tid.

### 5.1 Refleksjoner fra lys

Prosjektgruppen har testet hvordan refleksjon i glass påvirker avlesningen av QR-kodene. Der utstyret er plassert er det store vinduer med mye lysinnslipp. Lyset har skapt refleksjoner i glasset der QR-kodene er festet, og gjort at QR-kodene ikke alltid er synlig i videostrømmen. Dermed får ikke programmet lest av disse.

Etter diskusjon med ABB er prosjektgruppen kommet frem til at dette ikke vil påvirke sluttproduktet i stor grad, fordi produktet mest trolig skal plasseres i et rom uten mye lys. Derfor ble videre testing utført med persiennene nede, og normal belysning. Dette har fungert godt. Det er likevel viktig å være oppmerksom på at sterkt lys kan påvirke autofokus-funksjon og justering av lysbalansen til kameraet. Dette kan føre til at det går med ekstra tid for at kameraet skal justere seg.

## 5.2 Tilkobling av mikrokontroller

Under testing av prosjektet har det til tider vært vanskelig å opprette nettverkskontakt mellom datamaskin som kjører programmet og mikrokontrolleren. Prosjektgruppen fikk informasjon fra produsenten av utstyret om hvilken rekkefølge man skulle koble til kabler og strømledninger. Det ble sagt at ethernetkabelen mellom datamaskin og mikrokontroller måtte være tilkoblet før man koblet i mikrokontrollerens strømledning.

Gitt fremgangsmåte viste seg å ikke nødvendigvis fungere. Prosjektgruppen prøvde forskjellige løsninger. Noen av løsningene var å skifte ethernetkabel og fysisk tilkoblingsport i datamaskinen og koble til ledningene og kablene i forskjellig rekkefølge. Prosjektgruppen noterte når det fungerte og ikke. Resultatet tyder på at det er helt tilfeldig hva som skal til for å opprette nettverkskontakt, og derfor er eneste løsning å forsøke gjentatte ganger til det fungerer. Prosjektgruppen antar at problemet skyldes en svakhet i utstyret.

## 5.3 Lagring av posisjon

Prosjektgruppen hadde planlagt å bruke en kommando knyttet til API-et som kalles MOTORPOS. Denne kommandoen gir mulighet til at poden kan kjøre en gang frem og tilbake på skinnen, slik at lengden av denne blir lagret. Med utgangspunkt i lengden til skinnen ville programmet kunne lagre posisjoner som en avstand i forhold til skinnens totale lengde. Denne kommandoen ville gitt muligheten til å lagre og kjøre til posisjoner langs skinnen ved hjelp av avstand som enhet.

Etter flere mislykkede forsøk med å bruke kommandoen MOTORPOS, tok prosjektgruppen kontakt med produsenten av utstyret. De kunne informere om at det manglet en kabel til poden for at dette skulle fungere. Derfor så prosjektgruppen seg nødt til å finne en annen løsning, og valgte dermed å bruke tid

som en enhet for å lagre posisjon. Denne prosessen fungerer ved at en stoppeklokke starter når poden starter å kjøre, og at den stopper når en QR-kode er sentrert i bildet. På denne måten vet man hvor lang tid poden trenger, i en gitt hastighet, for å komme fra startposisjon til gjeldende QR-kode. Tid viste seg å være en upresis enhet å bruke. Tiden som lagres under setup samsvarer ikke med tiden poden bruker for å kjøre til samme posisjon når alarmen går. Prosjektgruppen tror at små variasjoner i motorens hastighet når den starter og stopper påvirker den totale tiden. I dette prosjektet fungerer tid som enhet godt nok, men ideelt sett ville man ha benyttet seg av kommandoen MOTORPOS.

### 5.4 Funksjonaliteten til kameraet

Webkameraet som er i bruk har mulighet for zoom. Dette har gjort det mulig å håndtere et snevrere bildeutsnitt, enn hvis zoom ikke var en mulighet. På denne måten kan QR-kodene være mindre, og/eller plasseres lengre unna kameraet før de ikke lenger kan leses. Når det zoomes inn er det også mulig å tilte bildet i vertikal retning. Dette er fordi utsnittet som vises og håndteres kun er en mindre del av det opprinnelige utsnittet. Ved at bildeutsnittet kan forflyttes opp og ned, gir det mulighet til å lese QR-koder som er plassert i ulike høyder. Prosjektgruppens analyse er at et kamera med høyere oppløsning, og mulighet for mer zoom, vil gjøre det mulig å håndtere QR-koder som er mindre og står tettere. I dette prosjektet er det brukt QR-koder som er 11x11cm store, og er plassert 3-3,5 meter unna kameraet.

### 5.5 Krav til QR-kodene

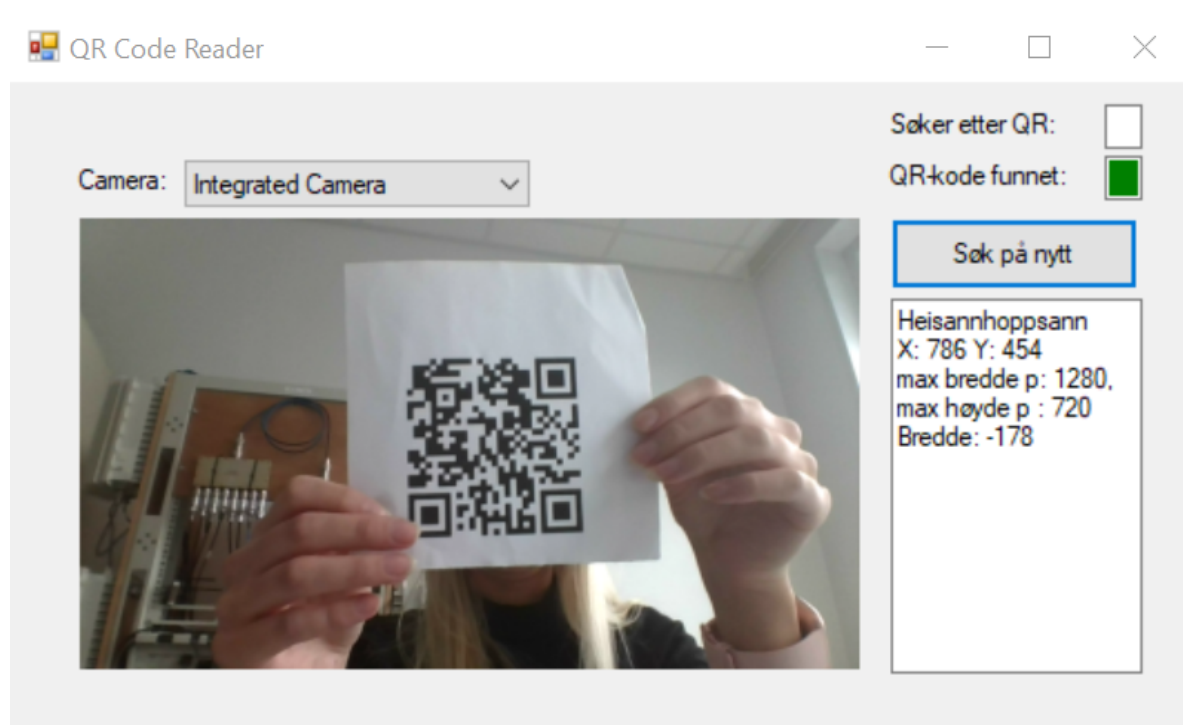
Programmet er laget slik at QR-kodene som representerer hvert kabinett kun består av en bokstav. Prosjektgruppen har brukt A, B, C, osv. QR-kodene til hver enhet består av en bokstav som representerer hvilket kabinett det tilhører, og et tall. Her er det brukt A1, A2, B1, osv. Programmet vil overse QR-koder som er i feil kabinett, for eksempel enhet A3 i kabinett C.

Et tilbakevendende spørsmål har vært håndtering av flere QR-koder i samme bildeutsnitt.

Prosjektgruppen mener at dette skal være mulig å få til, men at det krever langt mer omfattende programmering enn det er mulighet til i dette prosjektet. Slik programmet er nå er det nødvendig at QR-kodene ikke står for tett. Det sørger for at en og en QR-kode kan avleses om gangen.

Det er nødvendig at siste QR-kode ikke står for nære enden av skinnen. Gjør den dette, vil bremsene i enden av skinnen stoppe poden og avslutte setup før alle QR-kodene er skannet og lagret. Siste QR-kode må plasseres slik at den er sentrert i bildet når poden er minimum 30 cm fra enden av skinnen.

Dersom en QR-kode plasseres opp ned gir den negativ verdi når størrelsen avleses, dette illustreres i figur 5a, hvor bredden vises som -178. Dette påvirker posisjoneringen som gjøres for å få QR-koden sentrert i bildet. Programmet har per tid ikke en funksjon for å håndtere dette, og det er derfor viktig at QR-kodene plasseres riktig vei.



Figur 5a: QR-kode som holdes opp ned.

## 5.6 Testing av alarm

Sluttproduktet av programmet skal videreutvikles ved benytte seg av Milestone VMS for å motta alarmer. Det er nevnt tidligere at det derfor brukes simulering av alarm i dette prosjektet.

I noen tilfeller oppdages det en feil i programmet når en alarm simuleres. Denne feilen har prosjektgruppen ikke tilstrekkelig med tid til å rette opp. Feilen går ut på at programmet ikke begynner å

lete etter riktig QR-kode når en alarm simuleres. Siden dette kun oppstår de gangene setup er hentet fra lagret fil, og ikke når setup er kjørt fra start, mistenker prosjektgruppen at feilen ligger ved filavlesningen og ikke i simuleringen av alarm.

## 5.7 Brukervennlighet

GUI-programmet er laget for å være mest mulig brukervennlig. Se figur 3c. Prosjektgruppen har etterstrebet at det skal kunne brukes uten å ha kjennskap til programmet fra før. Dette er gjort ved at det er færrest mulig knapper. Alle knapper og informasjonstekster er enkle og tydelige. I tillegg deaktiveres knapper når de ikke skal eller kan trykkes på, og aktiveres når de skal kunne brukes. Prosjektgruppen har valgt å bruke engelsk for at språk ikke skal være en hindring for bruk av programmet.

## 6. Drøfting

I løpet av prosjekttiden er det blitt gjort en del tanker og refleksjoner rundt hvordan prosessen har foregått, måten oppgaven har blitt løst på og sluttproduktet som prosjektgruppen har kommet frem til. I de neste avsnittene skal vi gå nærmere inn på disse refleksjonene.

### 6.1 Styrker og begrensninger i prosjektet

Slik programmet er per dags dato, fungerer det godt som en demonstrasjon for hvilke bruksområder det er tiltenkt. Det fungerer også som et førsteutkast som kan videreutvikles, eller til inspirasjon for hva som er mulig å få til. Prosessen med utvikling av programmet er gjennomført i henhold til de gitte kravene og rammene i prosjektet.

Mange arbeidsoppgaver i starten av prosjektet ble gjort felles, dette viste seg etterhvert å ta lang tid. Prosjektgruppen kom frem til at det ville være bedre å dele arbeidsoppgaver mellom seg. På denne måten har oppgaven blitt løst på en god måte innen tidsfristen. Prosjektgruppen har vært i stand til å implementere de funksjonene som var ønsket fra starten, både de som ABB hadde sortert under basis, utvidet og avansert funksjonalitet.

Prosjektgruppen tror det kunne vært en fordel å ha mer struktur og et tydeligere bilde av hvordan sluttproduktet skulle være, fra starten av prosjektet. Dette bedret seg betraktelig utover i prosjektperioden og rammen rundt oppgaven ble utarbeidet kontinuerlig de første ukene med arbeid. Da kravene, ønskene og mulighetene for løsning av oppgaven ble tydeliggjort, gikk arbeidet lettere fremover. En god dialog med oppdragsgiver har vært essensielt for at prosjektgruppen oppnådde et godt sluttresultat. Måten kravene til oppgaven ble utarbeidet parallelt med at arbeidet pågikk har vært en viktig erfaring, som prosjektgruppen mener er relevant å ta med seg til senere prosjektarbeid.

Det er flere av teknologiene som er brukt i prosjektet som prosjektgruppen ikke hadde erfaring med fra tidligere. Ved å være nøye på å lese seg opp og innhente kunnskap om emnene, har det vært mulig å ta disse teknologiene i bruk på gode måter. Prosjektgruppen har brukt mye ressurser på å sørge for at alt av teknologier og metoder som har blitt implementert i prosjektet, har blitt brukt på en god og fornuftig måte. Dette er noe prosjektgruppen mener har blitt gjort så godt det lar seg gjøre, innenfor gitte tidsrammer.

Oppsummert har det blitt en god løsning på oppgaven, med et program som utfører alle de ønskede funksjonene. Programmet har blitt testet gjentatte ganger, og med få unntak, oppfører det seg som tiltenkt og er i stand til å arbeide autonomt slik kravet var.

## 6.2 Videre utvikling av programmet

Programmet er tiltenkt og brukes videre i reelle situasjoner. Dette krever optimalisering av programkoden og endringer i noen av funksjonene som tas i bruk. Det er enkelte deler av programmet prosjektgruppen mener er spesielt viktig å finne bedre løsninger for å håndtere. De er nevnt tidligere i rapporten, men oppsummeres her.

Det er nødvendig å ta stilling til håndtering av posisjoner. Nå brukes det tid som enhet for å lagre og hente ut podens posisjon når QR-koden til en bestemt enhet avleses. Dette er lite nøyaktig, og prosjektgruppen mener at bruk av kommandoen MOTORPOS ville egnet seg bedre. Da er det nødvendig med utstyr som støtter denne funksjonen.



Slik programmet er nå, lagres QR-kodene med tilhørende posisjon til en tekstfil. Dette er en lite sikker måte å lagre data på. Det bør tas stilling til om dette er tilstrekkelig, eller om det er mer hensiktsmessig å lagre denne informasjonen i en database. Dette vil kreve mye endring i koden, siden det håndteres på ulike måter ved lagring, bruk og henting av posisjonene.

Avlesning av QR-koder i vertikal retning er nå tilpasset bruk av webkamera. Skal det brukes et PTZ-kamera har dette andre funksjoner, og koden må tilpasses dette. Det vil være nødvendig å endre på zoom- og tiltfunksjonen som brukes i sluttproduktet.

## 7. Konklusjon

Prosjektgruppen opplever at oppgaven oppfyller kravene i problemstillingen og tilfredsstillende forventningene til ABB. Problemstillingen har fokus på at programmet skal arbeide autonomt. Programmet finner QR-koder, lagrer posisjoner og kan finne tilbake til disse når alarmer går, uten menneskelig interaksjon. Programmet utfører alle disse funksjonene slik det var ønsket.

Underveis i prosessen har alle frister blitt overholdt. Dette som følge av jevnt arbeid og at fremdriftsplanen har blitt fulgt. Resultatet er at prosjektet har blitt ferdigstilt innen tidsfristen. Godt samarbeid innad i prosjektgruppen, med ABB og med veileder Adis Hodzic har bidratt til at prosjektet i stor grad har gått som ønsket. Faste møtetidspunkter har sørget for lav terskel når prosjektgruppen har stoppet opp og hatt behov for hjelp.

Prosjektet har ført til en bratt læringskurve hos hver enkelt i prosjektgruppen, og har gitt en forsmak på arbeidslivet. Prosjektgruppen har opparbeidet seg erfaring med å arbeide mot tidsfrister, samarbeide med andre, samt å jobbe selvstendig for å møte krav satt av ABB. Prosjektgruppen har spesielt fått erfare viktigheten av å utnytte tiden og å ta aktive valg av løsninger, basert på opparbeidet kunnskap.

Oppsummert er prosjektgruppen fornøyd med det som har blitt gjort, og synes det ferdige produktet tilfredsstillende prosjektets mål. Prosjektperioden har stått til forventningene, det har vært en lærerik og morsom erfaring.

## Litteraturliste

ABB (2021a) *Om oss*. Tilgjengelig fra: <https://new.abb.com/no/om-oss> (Hentet: 25. januar 2021)

ABB (2021b) *Kort om oss*. Tilgjengelig fra: <https://new.abb.com/no/om-oss> (Hentet: 25. januar 2021)

ABB (2021c) *Bærekraftig utvikling*. Tilgjengelig fra:

<https://new.abb.com/no/om-oss/baerekraftig-utvikling> (Hentet: 25. januar 2021)

ABB (2021d) *Våre fire virksomhetsområder*. Tilgjengelig fra:

<https://new.abb.com/no/om-oss/var-virksomhet> (Hentet: 25. januar 2021)

ABB (2021e) *Telecoms and security systems*. Tilgjengelig fra:

<https://new.abb.com/oil-and-gas/systems-and-solutions/telecommunications> (Hentet: 25. januar 2021)

Aleksandersen, D. (2018) *How To Understand DMX And Avoid Potential Pitfalls*. Tilgjengelig fra:

<https://newsandviews.dataton.com/how-to-understand-dmx-and-avoid-potential-pitfalls> (Hentet: 24. mai 2021).

Beaconstac (2021) *Types of QR Codes: what are the different types?* Tilgjengelig fra:

<https://www.beaconstac.com/types-of-qr-codes-examples> (Hentet: 02. februar 2021)

Canon (2021) *Canon PTZ-kameraer*. Tilgjengelig fra: [https://www.canon.no/ptz-cameras/#id\\_2063777](https://www.canon.no/ptz-cameras/#id_2063777)

(Hentet: 21. april 2021).

GitHub Inc (2021) *Zxing*. Tilgjengelig fra: <https://github.com/zxing/zxing> (Hentet: 20. april 2021)

Hayes, A. (2020) *Quick Response (QR) Code*. Tilgjengelig fra:

<https://www.investopedia.com/terms/q/quick-response-qr-code.asp> (Hentet: 02. februar 2021)

Kirillov, A. (2016) *AForge.NET*. Tilgjengelig fra: <http://www.aforgenet.com/> (Hentet: 20. april 2021)

Logitech (2021) *Brio Ultra HD Pro Bedriftswebkamera*. Tilgjengelig fra:

<https://www.logitech.com/no-no/products/webcams/brio-4k-hdr-webcam.960-001106.html> (Hentet: 20. mai 2021)

Milestone VMS (2021) *Videobehandlingsprogramvare*. Tilgjengelig fra:

<https://www.milestonesys.com/no/solutions/platform/video-management-software/> (Hentet: 24. mai 2021)

Patterson, I. (2017) *NSSM*. Tilgjengelig fra: <https://nssm.cc/description> (Hentet: 19. april 2021)

Philips (2008) *Introduction to DMX*. Tilgjengelig fra:

<https://erg.abdn.ac.uk/users/gorry/eg3576/resources/Introduction%20to%20DMX%20philips.pdf>  
(Hentet: 31. mai 2021)

Redhat Inc (2021) *What is open source software?*. Tilgjengelig fra:

<https://opensource.com/resources/what-open-source> (Hentet: 24. mai 2021)

Rossen, E. (2020a) *API*. Tilgjengelig fra: <https://snl.no/API> (Hentet: 19. april 2021)

Rossen, E. (2020b) *C#*. Tilgjengelig fra: <https://snl.no/C%23> (Hentet: 19. april 2021)

Sweetwater (2021) *Understanding DMX*. Tilgjengelig fra:

<https://www.sweetwater.com/sweetcare/articles/understanding-dmx/#DMX-controllers> (Hentet: 24. mai 2021)

TechTerms (2021) *GUI*. Tilgjengelig fra: <https://techterms.com/definition/gui> (Hentet: 24. mai 2021)

TechTerms (2009) *RFID*. Tilgjengelig fra: <https://techterms.com/definition/rfid> (Hentet: 24. mai 2021)

TechTerms (2015) *QR code*. Tilgjengelig fra: [https://techterms.com/definition/qr\\_code](https://techterms.com/definition/qr_code) (Hentet: 2. februar 2021)

UiB (2021) *Tagg*. Tilgjengelig fra:

[https://ordbok.uib.no/perl/ordbok.cgi?OPP=tag&ant\\_bokmaal=5&ant\\_nynorsk=5&begge=+&ordbok=be](https://ordbok.uib.no/perl/ordbok.cgi?OPP=tag&ant_bokmaal=5&ant_nynorsk=5&begge=+&ordbok=be)  
gge (Hentet: 24. mai 2021)

## Appendix A - Forkortelser og ordforklaringer

**DMX** – Står for «Digital Multiplex» og er en protokoll brukt til å kontrollere forskjellige enheter som for eksempel lys, røykmaskiner eller i dette tilfellet pod (Sweetwater, 2021).

**GUI** – Står for «Graphical User Interface». Brukergrensesnitt som inneholder grafiske elementer som for eksempel vinduer, ikoner og knapper (TechTerms, 2021).

**Milestone VMS** – «Milestone videobehandling-programvare (VHP) bringer alle puslespillbitene rundt en videoovervåkningsinstallasjon sammen i en perfekt kombinasjon for å skape en løsning som holder folk og eiendommer trygge, i dag og i morgen» (Milestone VMS, 2021).

**QR-kode** – QR står for «Quick Response» og er en type strekkode som inneholder en matrise med prikker. Ved scanning konverteres prikkene i koden til tall eller tekst (TechTerms, 2015).

**RFID** – Står for «Radio-Frequency Identification». RFID er et system brukt for å spore objekter, personer eller dyr ved hjelp av tags som responderer til radiobølger (TechTerms, 2009).

**TAG** – En merkelapp festet på noe eller noen med hensikten om identifikasjon eller å gi annen informasjon (UiB, 2021).

## Appendix B - Prosjektforslag fra ABB

29.09.2020

### Mal for prosjektforslag til Bachelorprosjekt ved Data- og Informasjonsteknologi-studiene for våren 2021.

---

## Telecomoppgave Oil and Gas (ABB AS)

Kokstadveien 23B, 5257 Kokstad

[www.abb.no](http://www.abb.no)

Telecom avdeling

### Oppgave Oil and Gas

Navigasjonssystem for mobile enheter. Autonom operasjon. Dette gjelder for både kamera på skinneløsninger, mobile kameraløsninger (Rowere) og droner. Disse enhetene må kunne operere autonomt på anlegget. Mulige kommunikasjonsmuligheter:

- I. RFID
- II. QR
- III. Radio

#### Integrasjon av video fra Mobile enheter mot Milestone VMS

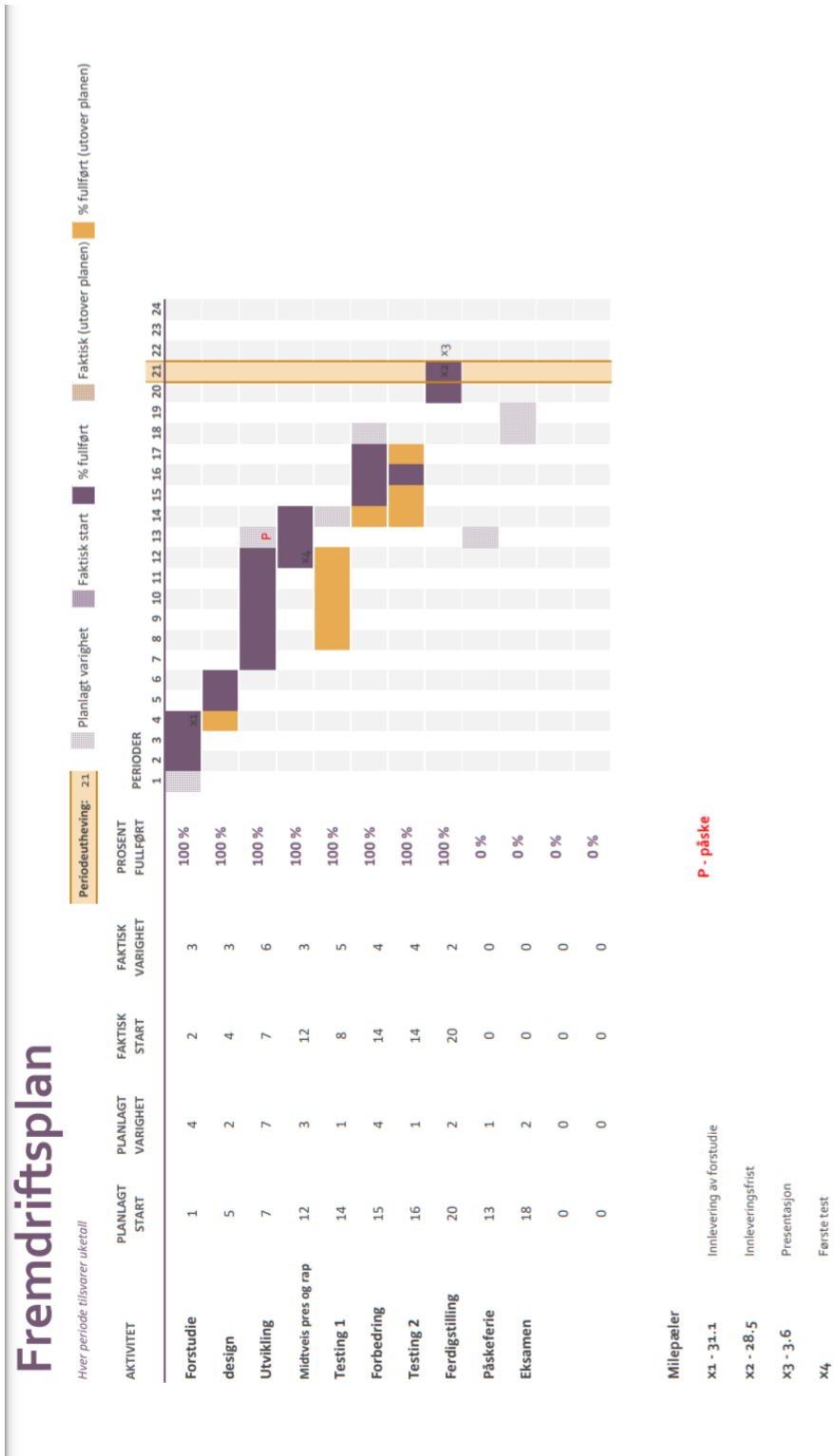
- IV. Det skal være mulig å overføre video fra et kamera montert på et mobilt enhet. Man skal her kunne se i hvilke retning kamera på det mobile enheter ser i forhold til enhetens posisjon og retning på selve kamera.
- V. Kan man redusere latency fra operatør til kamera og fra kamera til operatør for bedre opplevelse av manuell kontroll i forbindelse med manuell operasjon av enhet. (Ref. Colloboration teknologier)

### Kontaktpersoner

Aktuelle kontaktpersoner / veiledere:

- Erik Serck-Hanssen
  - Seksjonsleder
  - erik.serck-hanssen@no.abb.com
  - 922 22 141
-

## Appendix C - Fremdriftsplan



## Appendix D - Forstudie

Se vedlagt pdf-fil: "Appendix D - Forstudie.pdf".