

Fordeler slides:

Sylwester farge:

- Oppdraget / Oppgaven
- Klienten / Webgrensesnittet
- Selvkjøring
- Demo

Lex farge:

- Roboten
- Bilde, Video og Lydopptak
- Tilbakemelding

Aram farge:

- Løsningen
- Sensorer
- Alternative bruk av løsningen

Slide 1 - Om Gruppen:

Hei sann! Mitt navn er Aram Ghazi. Jeg studerer dataingeniørfaget og har tidligere jobberfaring som dataelektroniker for selskapet Data Respons AS. Jeg har erfaring innen Windows-servere og interesser meg i fagfeltene databaser og robotteknikk.

Hei, jeg heter Sylwester. Allerede fra jeg var 16 år gammel ville jeg lære meg å programmere for å prøve meg på utvikling av software og på grunn av det lærte jeg meg selv å programmere i Java. Er interessert i Operativsystemer, mer spesifikt Linux, og er også interessert i elektronikk og selvfølgelig softwareutvikling, men er kanskje mer interesert i low-level programming nå for tiden.

Jeg heter Lexander. Jeg er en dataingeniørstudent som har stor interesse for all slags teknologi spesielt elektronikk og biler. Jeg har erfaring innen robotteknikk, elektronikk og softwareutvikling.

Slide 2 - Oppdraget / Oppgaven :

Det oppdraget handler om er å utvikle en spionrobot. En spionrobot vil si en robot som kan brukes til forskjellige spionoppdraget eller til spionasje generelt. Eksempler hvor roboten vårt kan brukes er for eksempel til spionase på et lager. Roboten har blitt utstyrt med komponenter og funksjonaliteter som gjør spionoppdrag mulig. Komponentene og funksjonalitetene som roboten vårt har blitt utstyrt med kommer vi til å snakke litt mere om senere i presentasjonen.

Hovedmålet for oppdraget er å få bedre kjennskap til roboter og deres egenskaper for spionasje av personer.

Kort oppsummert så har vi utviklet og demonstrert gjennom arbeidet på oppdraget en mulig løsning på en spionrobot ved å bruke GoPiGo 3 og Raspberry Pi 3 Model B, som er noen av komponentene vi brukte til å bygge roboten.

Delmålet for oppdraget er å utvikle en robot med nødvendige funksjonaliteter for spionasje. Oppdraget var ganske fritt som vil si at vi selv bestemte hva slags egenskaper ved roboten eller funksjoner som skal følge med. Gjennom møter i gruppen bestemte vi oss over hva slags funksjonaliteter som vi skal implementere, og dette har vi gjort etter at vi har studert andre eksisterende løsningen på en spionrobot men også av å gjøre research når det gjelder spionasje.

Det var et par antagelser vi har gjort i samarbeid med oppdragsgiver og veileder som har med roboten og funksjonalitetene til roboten å gjøre. Vi ble enige om å betrakte roboten mer som et leketøy enn ordentlig spionrobot. Dette gjorde vi etter oppdagelse av 'begrensninger' til platformen vi måtte benytte under oppdraget, nemlig gopigo3.

Og videre som konsekvens av dette har det blitt bestemt at deteksjon av roboten er ikke en kritisk problem eller oppgave som må løses av gruppen.

Slide 3 - Løsningen:

Før vi kommer innpå oversikten rundt løsningen, vil jeg kjapt ta dere gjennom hver del som gjorde helheten av spionroboten en realitet. Hva vi som en gruppe har oppnådd gjennom prosjektperioden tok for seg

demonstrering, dokumentering og testing av produktet. Her implementerte vi funksjonaliteter rettet mot spionasje ved hjelp av sensorer koblet til produktet. Sensorene hjelper oss med indikere robotens omgivelser. Med et webgrensesnitt kan en bruker styre roboten ved hjelp av en PC eller annet foretrukket enhet. Tilslutt blir data registrert og lagret ved å ta bilde, video eller lydopptak gjennom en overføringsenhet, vanligvis en USB-minnebrikke for videre bruk og analyse. For å vite om data er overført, vises dette på webgrensesnittet. For å få tilgang, må man være tilkoblet et Wi-Fi aksesspunkt som videre åpner opp din standard nettleser. Løsningen vi benyttet var å opprette en tjenerdel som benytter Flask.

Men hva er Flask? Jo Flask er et rammeverk som tar for seg kommunikasjonen mellom klientdel og webtjener. Som et eksempel gjør den mottakelse og besvarelse på forespørsler fra klienten enkel å implementere.

Med disse løsningene tatt i betraktning under prosjektperioden mener vi at vi har levert de forventede resultatene og utviklet en best mulig spionrobot.

Slide 4 - Roboten:

Oppdraget handlet om å bygge en spionrobot basert på GoPiGo3. Dette er et robotbyggesett fra Dexter Industries som gruppen fikk tildelt av oppdragsgiveren.

Vi merket med en gang at roboten egner seg ikke til bruk utendørs på grunn av dens fysiske egenskaper. Dette er på grunn av hjulene og egentlig det meste fysiske som har med roboten å gjøre. Det ble da bestemt at roboten skulle bare utføre spionoppdrag innendørs.

I kjernen av roboten er det et GoPiGo3 kretskort som er koblet på I/O pinnene av en Raspberry Pi. Kretskortet har en innebygd strømtilkobling slik at man kan koble til den batteripakke som er inkludert i byggesettet. Denne gir strøm til både kretskortet og Raspberry Pi som gjør det mulig

for roboten å kjøre uten en kablet strømkilde.

Roboten er utstyrt med et kamera, en mikrofon, en distansesensor og en lys- og fargesensor. Kamera kobles til Raspberry Pi via CSI-porten og mikrofon via USB, mens distanse- og lys- og farge-sensorene er koblet til GoPiGo3 kretskortet via GROVE porter.

Slide 5 - Klienten / Webgrensesnitt:

Klienten vårt er et webgrensesnitt skrevet i HTML, CSS og JavaScript. Den kan man se på figuren til høyre for nøkkelordene.

Av å gjøre research på GoPiGo3 for å få mer kjennskap til plattformen for å så kunne begynne på å bygge roboten og se mulighetene som følger med, så kom vi bort i et eksempel som passet perfekt som et startpunkt for oppdraget vårt.

Eksemplet er fra Dexter Industries, som er produsenten av GoPiGo3, og er et eksempel på et live-streaming robot. Vi har bestemt oss for å bygge opp på dette eksempelet istedenfor å begynne fra scratch for å få mer tid til implementasjon av videre funksjonalitet.

Ved en videreutvikling gjort av oss på eksemplet fra Dexter industries så har vi kommet til et resultat som er en webgrensesnitt eller klient opp mot roboten som kan brukes til å både styre selve roboten eller for eksempel til styring av de tidligere nevnte funksjonalitetene.

Det å ha klienten som et webgrensesnitt har med seg positive side-effekter. For det første så muliggjør det kontroll av roboten ved hjelp av hvilken som helst enhet som har mulighet for tilkobling til wifi. Dette vil si da i praksis at klienten vår er platform-independent. Den er ikke bundet fast til en spesifikk platform. Og dette gjør at man slipper å utvikle en klient for hvert individuell plattform som sparer ressurser.

Til å snakke med backend eller tjener-delen benytter vi et JavaScript library som heter JQuery. Denne gjør sendingen av forespørsler og analyse av returdata fra tjeneren lett og oversiktlig.

Noen av funksjonene vi har utviklet når de blir eksekvert på tjenerdelen kjører en bash script, dette scriptet kan for eksempel manipulere filer, få informasjon av disken og vite hvor mye ledig plass det er igjen.

Vi mener at webgrensesnittet vårt er oversiktlig på grunn av logisk oppdeling av funksjonene. Til høyre har vi knappene tydelig markert med navn på hva de gjør for styring av funksjonalitetene og til venstre har vi forskjellige informasjon som ledig plass på roboten eller lesing fra sensorer. Det å ha en omfattelig HMI-felt med informasjon om alt slags mulig og mulighet for for eksempel manipulering av filer i form av sletting gjør at roboten ikke trengs å kobles manuelt opp mot pcen for vedlikehold. Klienten utviklet av oss gir brukeren total kontroll over roboten og alt som følger med.

Slide 6 - Bilde, Video og Lydopptak:

Oppdragsgiveren ga gruppen friheten til å velge hvilke funksjonaliteter de synes som passer for å bygge spionroboten.

Vi diskuterte først om oppdraget, som var spionasje, så søkte vi opp hva definisjonen av spionasje var. Spionasje betyr å hente inn informasjon fra kilder, enten åpne eller lukkede, uten tillatelse. Vi kom på at interessante informasjon som kunne bli innhentet var da: om omgivelsene, altså hvordan det ser ut og samtalene mellom personene i nærheten. Til slutt ble de funksjonalitetene som vi valgte først å implementere var mulighet for roboten til å ta bilde, lyd og video.

Kameraet som fulgte med GoPiGo3 byggesettet hadde ikke mulighet for nattsyn, et nytt kamera ble kjøpt inn da vi mener det er viktig for en spionrobot å ha nattsynkamera. Dette er fordi, når roboten skal utføre spionoppdrag er det viktig å ikke bli oppdaget, da bør brukeren plassere roboten i dårlig belyste steder i området slik at den blir vanskelig å finne. I tillegg til det beholder roboten også funksjonen for bilde- og videoopptak i mørke områder. Dette gir også mulighet for videreutvikling ved at det kan implementeres en funksjon 'beside' selvkjøring hvor roboten gjemmer seg i mørke steder når den detekterer mennesker.

Etter å ha slått på roboten, koblet til dens Wi-Fi og åpne webgrensesnittet kan man se knapper for å ta bilde, video og lyd. Etter å ha trykket på knappen sendes det ett POST forespørsel med en del av URL strengen til funksjonen i Python som tilhører denne strengen. F.eks når 'Start audio'(lydopptak) knappen trykkes blir det sendt denne strengen som vises på slide. Denne strengen mottas av Python og den tilhørende funksjonen for strengen er da å kjøre lydopptak funksjonen. Denne prosessen blir det samme på bilde- og videoopptak knappene også.

Slide 7 - Sensorer:

Produktet vårt er utstyrt med to sensorer med hver sin hensikt som muliggjør behovet for ønskelige spionoppdrag. Første sensor det er snakk om er distansesensor, som oppdager avstander fra hindringer og gjenstander, og gir roboten muligheten til å navigere. Andre sensor er lys-og-fargesensor som bruker fargespekteret RGB, altså fargene rød, grønn og blå. Til å programmere sensorene har vi brukt Dexter Industries sin API kalt DI_Sensors. Denne gjør programmeringen av sensorer lett og oversiktlig. Via webgrensesnittet klarer vi å lese sensorverdiene over til HMI-feltet som vist på bilde her. Verdiene illustrert til høyre viser testen av roboten i et vanlig rom med lyset på. I et mørkt rom vil verdiene være vesentlig mindre. Begge sensorer er nødvendige med tanke på at navigasjon og ulike lysnivåer gjør produktet vårt essensielt for spionasje.

Slide 8 - Selvkjøring:

Selvkjøring er som alle funksjonene styrt fra webgrensesnittet. Mer presis er det ett knapp på webgrensesnittet som kontroller om roboten er i vanlig manuell modus eller om den er i selvkjøringmodus. I denne modusen vil roboten kjøre av seg selv og samtidig unngå kollidering med objekter i dens vei.

Deteksjon av gjenstander eller objekter i veien til roboten skjer ved å

lese verdien fra distanse sensoren koblet til roboten og som vi har snakket om allerede i denne presentasjonen. Dersom roboten kommer en viss lenge eller avstand til et objekt vil den prøve å kjøre en annen retning for å fortsette å patruljere et rom eller område.

Den opprinnelige planen for selvkjøring var å samspille sammen med et system som skulle detektere personer i et rom. Etter deteksjon skulle roboten gjemme seg for å unngå detektering. Dessverre på grunn av mangel på tid ble selvkjøring funksjonaliteten ikke så avansert som vi opprinnelig hadde planlagt og samtidig ble det verken ingen tid for deteksjon av mennesker.

Slide 9 - Demo:

Nå vil vi vise et kort video som demonstrerer roboten og en del av funksjoner som følger med, den fulle videon har blitt lagt ut på vår bloggsiden:

Slide 10 - Alternative bruk av løsningen:

Vår løsning byr på fleksibilitet som kan bli brukt til andre systemer. Med det så kan Raspberry Pi 3 Model B monteres ut fra resten av spionroboten og monteres inn på en annen type robot. Webgrensesnitt, bilde, lyd og videoopptak er noen funksjonaliteter som er avhengig av dette kretskortet. På denne måten kan man utvide en annen eksisterende løsning og montere inn det gruppen har utviklet for å legge de nevnte funksjonalitetene inn. Et godt eks kan være en robotstøvsuger. Da har man plutselig en robotstøvsuger som kan utføre spionoppdrag.

Slide 11 - Tilbakemelding:

Etter all funksjonalitet ble implementert på roboten og webgrensesnittet utførte gruppen deretter en brukertest. Dette var for å skaffe tilbakemeldinger om hva som er bra og/eller dårlig samt hva som kan forbedres og/eller legges til med tanke på funksjonaliteter. Før testen fikk også brukerne tildelt brukermanual slik at gruppen for også testet kvaliteten på den.

Brukertestene gikk problemfritt, brukerne mente det var lett å bruke roboten og kjøre funksjonene på webgrensesnittet. Brukerne hadde også tilbakemeldinger og forslag på hva som kan forbedres, som gruppen skrev i en liste og forbedret.

Etter brukertesten utførte gruppen akseptansetesten. Hensikten med denne var å få oppdragsgiverens inntrykk av vår løsning, altså roboten og webgrensesnittet.

På grunn av koronapandemien hadde ikke oppdragsgiveren muligheten til å teste roboten selv personlig. Allikevel mente vi det var viktig for oppdragsgiveren å se roboten i bruk og få tilbakemelding om løsningen gruppen har laget tilfredsstillende oppdragsgiverens ønsker. Som løsning lagde vi da en video som demonstrerer funksjonalitetene på både webgrensesnittet og roboten, og deretter sendte det til oppdragsgiveren.

Vi fikk svar fra oppdragsgiveren og fikk positiv tilbakemelding. Oppdragsgiveren var fornøyd med vår løsning samt at den tilfredsstilte oppdragsgiverens ønsker.