

Rapport

IDI-Rally lego Team 3

Mikael Kalstad, Bård Hestmark, Henrik Tronstad, William Dalheim, Didrik Ring

01.10.2018

Innhold

[Del 1. Rolle og arbeidsfordeling 2](#_Toc525721073)

[Del 2. Problemstillinger og løsninger 3](#_Toc525721074)

[Del 3. Klassediagram for egenutviklet kode 4](#_Toc525721075)

[Del 4. Programkode i Java 5](#_Toc525721076)

[Del 5. Oppsummering 6](#_Toc525721077)

# Del 1. Rolle og arbeidsfordeling

Vi har brukt de generelle rollene som ble bestemt i arbeidskontrakten som teamet lagde i starten av prosjektarbeidet. I arbeidskontrakten står det teamleder skal fordele arbeidsoppgavene på de ulike team-medlemmene. Slik var vår fordeling under prosjektet:

*Navnene under på hver av de ulike arbeidsoppgavene hadde hovedansvaret for at denne oppgaven ble utført. Alle arbeidsoppgavene var gjennomført som et team-arbeid, derfor har alle bidratt til alle de ulike delene, til tross for at hver har kun navnet på noen.*

Programmerer: William Dalheim

William sto hovedsakelig for skrivingen av koden som ble brukt i prosjektet. Grunnen til hvorfor det ble skrevet på William sin datamaskin er beskrevet nærmere i del 2; *Problemstillinger og løsninger.*

Bygg/design: Bård Hestmark, William Dalheim, Didrik Ring

De som ikke programmerte eller hadde andre arbeidsoppgaver brukte tiden sin på å designe og bygge lego-modellen.

Rapportskriving: Mikael Kalstad

Team-lederen har god oversikt over alle arbeidsoppgavene som måtte gjøres og har derfor god oversikt over prosjektet i sin helhet. Derfor ble denne rapporten skrevet av teamlederen.

# Del 2. Problemstillinger og løsninger

*Underveis i prosjektet oppsto det flere problemer. Som et team måtte vi bruke tiden godt å finne gode løsninger på disse problemstillingene. Her er noen problemer som vi måtte:*

Kommunikasjons-problem

**Problem**: PC-ene til flere på teamet ville ikke kommunisere med EV3 brikken.

**Løsning**: En på teamet fikk til å kommunisere med EV3 brikken. PC-en til denne personen ble brukt til programmeringen og kommunikasjonen for å overføre filer.

**Beskrivelse**: Mikael, Bård og Henrik fikk problemer med å kommunisere med EV3 brikken. Vi prøvde å reinstallere drivere, kommunisere over Bluetooth og skifte IP-adresse. Lærerstudentene klarte heller ikke å finne en løsning på dette med det første. For å ikke kaste bort all tiden vår på dette, bestemte vi oss for å bruke William sin PC for kommunikasjon, siden han var den eneste som fikk kontakt med EV3 brikken.

Sensor justering

**Problem**: Få stilt inn sensorene slik at de reagerer på svart.

**Løsning**: Bruke den innebygde «tools» funksjonen på lego brikken for å sjekke verdi til sensorene.

**Beskrivelse**: Vi fikk problemer med at sensorene ikke reagerte på svar med våre verdier. Derfor måtte vi bruke sensortesting i «tools» for å få ut verdiene som tilsvarer svart på underlaget.

Holde seg på den svarte linjen

**Problem**: Få lego-roboten til å holde seg på den svarte linjen med svinger.

**Løsning**: Bruke tre sensorer og programmere en logisk løsning.

**Beskrivelse**: For å få lego-roboten til å holde seg på den svarte linjen brukte vi 3 forskjellige sensorer, en i midten med en på hver side. Med dette kan vi bruke de to sensorene på siden til å sjekke om roboten er ute av kurs, og justere til den i midten ser den svarte linjen.

# Del 3. Klassediagram for egenutviklet kode

Beskrivelse på virkemåte for kode:

For å gjøre logikken i koden enklere lagde vi en klasse med metoder som vi brukte i en annen klasse. Metodene i denne klassen inneholder metoder som vi fikk bruk for under prosjektet. Det var enklere for oss å lage en metode med en del kode, også bruke kun en enkel metode i «hovedfilen». Ved å gjøre dette gir koden mer mening og det er enklere for oss å implementere ønsket virkemåte.

KLASSEDIAGRAM

|  |
| --- |
| Legobil |
| - Brick brick  - Port s1  - Port s1  - Port s3  - NXTLightSensor lysSensorVenstre  - SampleProvider lysLeserVenstre  - float[] lysSampleVenstre  - EV3ColorSensor fargeSensorMidten  - SampleProvider fargeLeserMidten  - float[] fargeSampleMidten  - NXTLightSensor lysSensorHøyre  - SampleProvider lysLeserHøyre  - float[] lysSampleHøyre |
| + void framoverUtenStopp(int fart)  + void svingUtenStopp(String retning)  + void stoppMotorer()  + double[] lesFarge()  + void fartA(int fart)  + void fartB(int fart) |

# Del 4. Programkode i Java

# Del 5. Oppsummering

Prosjektet har vært lærerikt, men også utfordrende for teamet. Underveis dukket det opp flere problemer, flere av disse krevde at vi bygget om lego-roboten eller endret strukturen i koden. Vi løste disse problemene på en god måte ved å samarbeide og komme opp med smarte løsninger.

Vi mener at det var dårlig med ressurser til prosjektet. Alle teamene hadde ulik fordeling av delene, noe som var utfordrende siden vi ofte manglet deler til løsninger vi ønsket å lage. I tillegg hadde vi mindre kabler enn sensorer og motorer vi hadde på roboten, men vi fikk ordnet ekstra kabler. Derimot så fungerte lego-roboten på en tilfredsstillende måte til slutt.