# Program

Programmet består af to tasks hvor den ene er main tasken og den anden er musik tasken. Derudover er det lavet adskillige hjælpefunktioner der primært kaldes fra main task.

## Main task

I main tasken ligger den basale navigations kode, der gør robotten i stand til at følge en linje og detektere sorte missionsmarkører. Hele navigations koden ligger i et uendeligt loop, der kaldes efter musik tasken er startet og farvesensoren er blevet kalibreret. Når navigationskoden begynder forventer programmet at robotten er placeret max 7,5 cm til højre for linjen.

Selve navigationsloopet er opdelt i 5 faser:

1. Initialisering af refleksionsmåling og nulstilling af timer1
2. Gå ind i loop der kører så længe refleksionsværdien er højere end skilleværdien. I loopet sættes de to hovedmotorer til at køre fremad med asynkron hastighed, hvor den højre motor kører hurtigst. Dette får robotten til at køre fremad samtidigt med at den drejer til venstre. Den egentlige effekt af loopet er at robotten vil køre fremad imens den langsomt søger ind mod stregen. Hver gang loopet kører læses der også op farvesensoren om refleksionen ændrer sig. Det sidste der bliver gjort inden loopet gentages er at kalde checkIfLost() funktionen. Når loopet afsluttes på grund af for lav refleksionsværdi nulstilles timer1 før næste loop påbegynddes.

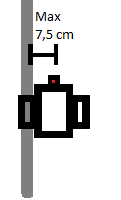


Figure 1 - Illustration af anbefalede robotplacering.

1. Dette loop køres så længe robotten er cirka midt på linjen. Det vil sige at så længe reflektionsværdien er ± 10 % af skilleværdien kører robotten lige fremad. Her bruges checkIfLost funktionen ikke. Når dette loop brydes bliver timer1 igen nulstillet før det næste loop begynder.
2. Når reflektionen er mindre end skilleværdien, men højere end stopLine værdien vil robotten antage at sensoren er over den grå streg. I det tilfælde vil den køre fremad mens den drejer langsomt til højre. Ellers fungere denne fase ligesom fase 2.
3. Den sidste fase er at tjekke efter om robottens farvesensor er over en sort missionsmarkering. Hvis dette er tilfældet vil robotten lægge en til stopLineCounts og derefter bruge denne globale variabel til at vurdere hvilken mission robotten er nået til.

### Musik task

Musik tasken består af et uendeligt loop der står for at der spilles musik så snart musik tasken er startet. Musik tasken vil dynamisk ændre musik nummer alt efter robottens status. Hvis robotten er i en fejltilstand, spilles der en mere dyster lyd, hvorimod der spilles en glad melodi mens robotten kører som den skal.

### Farvekalibrering – manualCallibColor()

For at sikre at robotten er I stand til at navigere I forskellige lysforhold skal farvesensoren kalibreres ved hver opstart. Når kalibreringsfunktionen kaldes, bliver der skrevet simple hjælpe instruktioner EV3 skærmen. Først skal robotten placeres så farvesensoren kan se den hvide baggrunds overflade, den målte refleksionsværdi vises på skærmen. Når værdien er stabil, trykkes der på enter knappen og programmet vil måle gennemsnitsværdien af overfladen over 10 samples, ved at kalde avgReflectedLight() funktionen. Når værdien er gemt, gentages det samme for den grå streg og de sorte missionsmarkører.

Når alle tre nuancer er blevet opmålt udregner programmet gennemsnitsværdien af den lyse værdi og den grå værdi. Dette gemmes som en skilleværdi der bestemmer om farvesensoren er over stregen eller ej.

Værdien for den sorte missionsmarkør gemmes i en global variabel. Denne vil senere blive brugt til at bedømme hvornår sensoren er over en missionsmarkør.

### CheckIfLost(float lostTimer, bool direction)

Denne funktion kontrollerer om robotten har set enten stregen eller baggrunden inden for de seneste 1500 ms. Hvis det ikke er tilfældet vil robotten stoppe med at føre fremad og begynde en stationær søgning efter stregen eller baggrunden. I søgningstilstand vil robotten dreje rundt om sig selv på stedet og fortsætte i den omdrejningsretning den kørte i før de 1500 ms er udløbet.

### calcDistMoved()

I en eller flere af missionerne, er der brug for at vide hvor langt robotten, har bevæget sig på et givet stykke tid. Derfor er der lavet en funktion, der kan udregne denne distance ved følgende formel: