

## Lab Session 5

### Black White Detection

Vi startede ud med at bygge robotten fra første lab session og montere lyssensoren som anvist i instruktionerne. Herefter satte vi os ind i funktionalitet af BlackWhiteSensor klassen og gav os derefter til at skrive en ny klasse som anvendte denne.

Vores klasse er relativt simpel og indeholder et loop, som anvender busy-wait til konstant at checke hvorvidt venstre eller højre knap på NXT enheden bliver trykket på. Hvis dette er tilfældet, så foretager vi en måling, viser værdien og hvorvidt det er en sort/mørk farve eller en hvid/lys farve.

Indmaden i control-loopet ser således ud:

```
boolean left = waitForLeftRight();

if (left)
{
LCD.clear();
LCD.drawString("The color is", 0, 0);
LCD.drawString("black: " + blackWhiteSensor.black(), 0, 1);

LCD.drawString("Threshold="+blackWhiteSensor.getThreshold(), 0, 3);
LCD.drawString("Light value="+blackWhiteSensor.light(), 0, 4);
Thread.sleep(5000);
}
else
{
LCD.clear();
LCD.drawString("The color is", 0, 0);
LCD.drawString("white: " + blackWhiteSensor.white(), 0, 1);

LCD.drawString("Threshold=" + blackWhiteSensor.getThreshold(), 0, 3);
LCD.drawString("Light value="+blackWhiteSensor.light(), 0, 4);
Thread.sleep(5000);
}
```

### Line Follower with Calibration

I denne opgave anvendte vi de to klasser LineFollowerCal og Car sammen med BlackWhiteSensor klassen fra første opgave. Vores afprøvning af disse fungerede helt som forventet, og vores robot fik tilført power til enten venstre eller højre motor alt afhængigt af, om lyssensoren detekterede en mørk eller lys farve.

### ColorSensor with Calibration

Her lod vi os inspirere af BlackWhiteSensor klassen fra de forrige to opgaver og udvidede blot denne for også at kunne detektere en blå farve, da en sådan farve var tilstede på testbanen.

Vi tilføjede følgende metode blue til vores nye ThreeColorSensor klasse og tilføjede også et par nye fields, som skulle bruges i forbindelse med black/blue og blue/white threshold. Vores test klasse implementerede vi i meget samme stil som i forrige opgave, hvor vi testede BlackWhiteSensor klassen. På denne måde var vi i stand til at detektere både hvid, sort og blå farve på NXT enheden.

```
public boolean blue()
{
int value = ls.readValue();
return (value > blackBlueThreshold && value <>
}
```

Vi valgte ikke at detektere grøn farve, som der bliver lagt op til i opgaven, da "goal" zonen på testbanen er blå :)

## Line Follower that stops in a Goal Zone

I denne sidste opgave, hvor vi skulle anvende vores ThreeColorSensor, forsøgte vi først naivt blot at tilføje en case til LineFollowerCal klassen, sådan at vores robot ville stoppe, hvis lyssensoren målte en værdi, som repræsenterede en blå farve, jvf. nedenstående kode:

```
if (sensor.black())
  Car.forward(power, o);
else if (sensor.white())
  Car.forward(o, power);
else
{
  Car.stop();
  Sound.beepSequenceUp();
}
```

Dette virkede dog ikke efter hensigten, da det gik op for os, at lyssensoren kommer ind i "else" casen, når farven skifter mellem sort og hvid, så her stoppede robotten næsten med det samme under vores test.

Vi huskede Ole's vise ord og tilføjede en ekstra lyssensor til vores robot, som kun skulle bruges til at detektere blå farve. For at dette skulle virke, måtte vi være sikre på, at den ekstra lyssensor under kørslen kun detekterer den samme farve (vi har monteret sensoren, så den kun ser hvid), for ellers ville vi havne i samme situation som før. Vi ændrede vores ThreeColorSensor klasse til at tage to sensorer i konstruktøren og ændre control-loopet til følgende:

```
public ThreeColorSensor(SensorPort blackWhite, SensorPort blue)
{
  ls = new LightSensor(blackWhite);
  blueSensor = new RCXLightSensor(blue);
  // Use the light sensor as a reflection sensor
  ls.setFloodlight(true);
  blueSensor.setFloodlight(true);
}

while (! Button.ESCAPE.isPressed())
{
  LCD.drawInt(sensor.light(),4,10,2);
  LCD.refresh();
  if (sensor.blue() //Uses sensorport S1
  {
    Car.stop();
    Sound.beepSequenceUp(); //WEEE!
    Thread.sleep(1000);
    break;
  }
  else if (sensor.black() //Uses sensorport S4
  Car.forward(power, o);
  else if (sensor.white() //Uses sensorport S4
  Car.forward(o, power);
  else
  {
```

```
//We should not end up here, but just to be safe!  
Car.stop();  
break;  
}  
Thread.sleep(10);  
}
```

Desuden tilføjede vi et par threshold metoder, som vi brugte til atdetektere, hvorvidt robotens lyssensor målte en sort, hvid eller blå farve:

```
public boolean black()  
{  
return (ls.readValue() < blackWhiteThreshold);  
}  
public boolean white()  
{  
return (ls.readValue() > blackWhiteThreshold);  
}  
public boolean blue()  
{  
return (blueSensor.readValue() < blueWhiteThreshold);  
}
```

Vi havde dog problemer med at få roboten til at opføre sig som forventet, da RCX sensoren tilsyneladende har en meget lav granularitet. Den læste 22 ved detektering af en blå farve og 28 ved hvid - ikke den store forskel! Dette havde den uheldige effekt, at de beskidte dele af banen blev fortolket som blå med RCX sensoren, og derfor stoppede roboten hurtigt. Vi er dog rimelig sikre på, at roboten ville opføre sig korrekt med to NXT sensorer, da disse havde større granularitet i målingerne. F.eks. målte denne lyssensor følgende værdier:

```
32 = black  
40 = blue  
54 = white
```