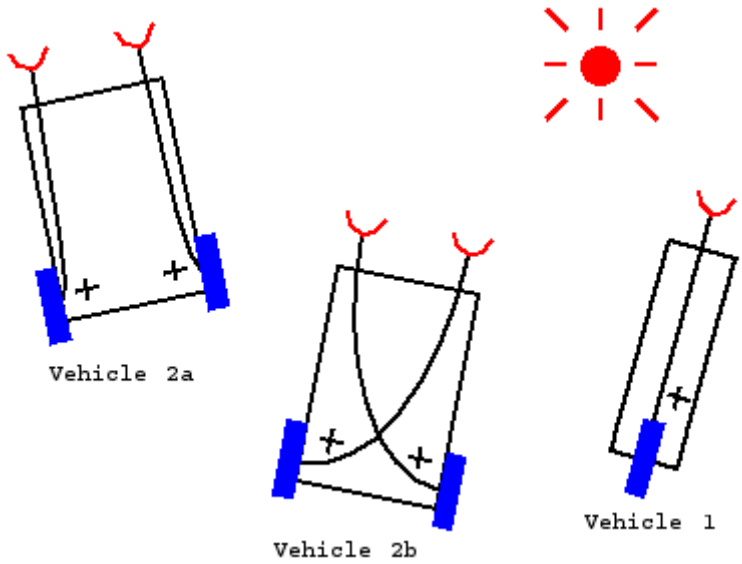
Marc Ilsø Poulsen, Peter Bartram Hansen og Bjarne Antonsen 4/11-2010
dLego: Lab Session 7 Rapport
Varighed: 3 timer

Formål:

Formålet med denne lab session er at bygge forskellige former for Braitenberg køretøjer, som inspireret af nedenstående figur:



Robotterne skal reagere på input i form af lyd eller lys og ændre deres hastighed proportionelt med niveauet for disse input.

Plan:

Vores plan var at anvende 9797 LEGO robotten til denne lab session. For at gøre robotten i stand til at reagere på input fra lyd- og lyskilder var vi nødt til først at anvende en mikrofon og senere et par lyssensorer, som blev monteret på robotten.

Resultater:

Vi forsøgte først at implementere Braitenberg køretøj 1, hvor vi implementerede den ønskede opførsel ved blot at give samme power til begge motorer. Her valgte vi først en direkte proportionel styring af robotten udfra lydniveauet, men dette virkede ikke helt efter hensigten. Robotten bevægede sig nemlig kun en ganske kort distance, når man eksempelvis klappede, grundet den meget korte varighed af lyden. Vores program udskrev det målte lydniveau på NXT enheden og ved hjælp af dette kunne vi se, at robotten tilsyneladende ikke reagerede på power under en værdi af ca. 60. Dette var dog grundet i, at der var for meget friktion til at robotten kunne bevæge sig, hvis den fik for lidt power (under 60).
Vores control loop til første Braitenberg køretøj endte med at se således ud og forklarer i meget høj grad sig selv:

```
while (!Button.ESCAPE.isPressed())
{
	int value = ss.readValue();

	if (value <= 70) {
```

```
                m1.controlMotor(70, forward);
                m2.controlMotor(70, forward);
        }
        else if (value > 100)
        {
                m1.controlMotor(100, forward);
                m2.controlMotor(100, forward);
        }
        else {
                m1.controlMotor(value, forward);
                m2.controlMotor(value, forward);

                Thread.sleep(900); //To better see the power applied.
        }
}
```

For at implementere Braitenberg køretøj 2a monterede vi to RCX lyssensorer på hver sin side af NXT enheden. Fra lab session 5 havde vi erfaret, at disse sensorer ikke målte så høje værdier som de nyere NXT lyssensorer og derfor kunne vi ikke bruge en direkte proportionel styring mellem lysniveauet og power til motorerne. Vi forsøgte at kalibrere ved hjælp af calibrateHigh() og calibrateLow() metoderne, men disse virkede ikke som forventet. Vi forventede, at kald af de to metoder ville sætte intervallet for værdier af lyssensorerne, men efter kalibrering målte sensorerne stadig de samme værdier som før. Et eksempel på vores brug af disse kan ses her:

```
LCD.clear();
LCD.drawString("Press LEFT to", 0, 0);
LCD.drawString("calibrate HIGH", 0, 1);
LCD.drawString("lightvalue for", 0, 2);
LCD.drawString("sensor 1", 0, 3);

while (!Button.LEFT.isPressed())
{
        LCD.drawString("Value: " + ls1.readValue(), 0, 5);
}
ls1.calibrateHigh();
```

Vi endte med at tage højde for de snævre interval for de målte værdier ved at gange en konstant med værdi 2 på det målte lysniveau.
Det indre control loop til denne opgave kom til at se således ud:

```
while (!Button.ESCAPE.isPressed())
{
        int value1 = ls1.readValue() * 2;
        int value2 = ls2.readValue() * 2;

        if (value1 > 100) m1.controlMotor(100, forward);
        else m1.controlMotor(value1, forward);
```

```
        if (value2 > 100) m2.controlMotor(100, forward);
        else m2.controlMotor(value2, forward);

        LCD.clear();
        LCD.drawString("Motor 1: "+ value1, 0, 0);
        LCD.drawString("Motor 2: "+ value2, 0, 1);

        Thread.sleep(100);
}
```

I koden tager vi også højde for, at power til motorerne ikke overstiger en værdi af 100, ellers bruges der en proportionel styring.

Vi løste opgave 2b ved at bytte om på tilslutningen af de to lyssensorer :) Dette resulterede i at robotten i stedet for at dreje imod lyskilden nu i stedet drejede væk fra den.

Hele programmet til implementering af opgave 2a var implementeret med brug af blot en enkelt tråd. I stedet anvendte vi en tråd til hvert lyssensor/motor par, hvori vi overfører den målte sensorværdi (gange 2) direkte til den ene motor. Implementering er lavet ved hjælp af to klasser, som kan ses nedenfor:

```
public class BB2Threads {

        public static void main(String[] args) {
                Thread a = new Thread(new LightThread(
                                new RCXLightSensor(SensorPort.S1), MotorPort.A));
                Thread b = new Thread(new LightThread(
                                new RCXLightSensor(SensorPort.S4), MotorPort.C));

                a.start();
                b.start();
        }
}

public class LightThread implements Runnable {

        private MotorPort motor;
        private RCXLightSensor ls;

        public LightThread(RCXLightSensor RCXls, MotorPort m) {
                motor = m;
                ls = RCXls;
        }

        @Override
        public void run() {
                while (!Button.ESCAPE.isPressed())
                {
```

```
                        int value = ls.readValue() * 2;

                        if (value > 100) value = 100;

                        motor.controlMotor(value, 1);
                }

                motor.controlMotor(0, 3);
                try {
                        Thread.sleep(1000);
                } catch (InterruptedException e) {
                }
        }

}
```

Vi kunne dog ikke se den store forskel på denne implementation med flere tråde kontra implementationen som kun anvendte en enkelt tråd. Vi havde dog kun en enkelt lyskilde til rådighed, så vi kunne forestille os, at der måske ville kunne ses forskel, hvis vi havde været i stand til at lyse på begge sensorer på en gang.

Konklusion:

Vi har opnået vores mål med denne øvelse. Gennem brug af lyd- og lyssensorer har vi konstrueret forskellige robotter, som vist i den indledende figur. Ved at anvende input fra disse sensorer til at styre motorerne med har vi kunnet observere forskellig opførsel.