

Lego Mindstorms RCX mit Java - Ein erstes Programm -

Durch das folgende Programm wird an Anschluss S1 ein Lichtsensor angemeldet. Der Motor an Ausgang A wird auf „vorwärts“ eingestellt.

In einer ständigen Wiederholung wird in der Anzeige der Wert des Lichtsensors angezeigt. Der Motor wird gestoppt, sobald der Lichtsensor einen bestimmten Wert (hier 60) überschreitet. (Erläuterungen zu den roboterspezifischen Befehlen auf der nächsten Seite.)

```
import josx.platform.rcx.*;

public class StopLinie
{
    public static void main(String[] aArg) throws Exception
    {
        Sensor.S1.setTypeAndMode(3,0x80);
        Sensor.S1.activate();
        Motor.A.setPower(3);
        Motor.A.forward();

        while(true)
        {
            LCD.showNumber(Sensor.S1.readValue());
            if(Sensor.S1.readValue()>60)
            {
                Motor.A.stop;
            }
        } // Ende von while(true)

    } //Ende von main
}
```

Durch `import josx.platform.rcx.*` werden die Klassen geladen, in denen die roboterspezifischen Befehle (Klassen) abgelegt sind.

Das eigentliche Programm ist durch eine Java-Klasse mit dem Namen „StopLinie“ definiert, daher ist auch der Dateiname als „StopLinie.java“ zu wählen!

Die Hauptmethode lautet `public static void main(String[] aArg) throws Exception`. Diese wird beim Start des Programmes ausgeführt und ist vergleichbar mit dem `task main()` in Not Quite C (NQC).

Ein minimaler Programmrumpf ist folgender:

```
import josx.platform.rcx.*;

public class NixZuTun
{
    public static void main(String[] aArg) throws Exception
    {
        // Hier kämen die Anweisungen hin.
    }
}
```

Lego Mindstorms RCX mit Java - kleine Befehlsübersicht -

Motoren

allgemeine Syntax:

`Motor.A.XXXX` (steuert den Motor an Ausgang A an)

wichtige Befehle:

<code>forward()</code>	vorwärts
<code>backward()</code>	rückwärts
<code>setPower(int aPower)</code>	Kraft einstellen (Werte von 0 bis 7)
<code>stop()</code>	anhalten
<code>isForward()</code>	liefert „true“ bei vorwärts
<code>flt()</code>	Leerlauf

Sensoren

allgemeine Syntax:

`Sensor.S1.XXXX` (steuert den Sensor an Einfang 1 an)

wichtige Befehle:

<code>activate()</code>	aktivieren
<code>readValue()</code>	liefert den aktuellen Wert
<code>setTypeAndMode(int aType, int aMode)</code>	Typ und Modus einstellen *)
<code>readRawValue()</code>	liefert den aktuellen raw-Wert

*)

aType: 0=RAW, 1=TOUCH, 2=TEMP, 3=LIGHT, 4=ROTATION

aMode: 0x00=RAW, 0x20=BOOL, 0x40=EDGE, 0x60=PULSE, 0x80=Percent,
0xA0=DEGC, 0xC0=DEGF, 0xE0=ANGLE

Details zu allen Befehlen kann man im Lejos-API nachlesen. Dieses kann man im Java-Editor im Menü durch Hilfe/Mindstorms aufrufen.