**Lichtsensor**

|  |  |  |  |
| --- | --- | --- | --- |
| Der EV3 ColorSensor kann Farben erkennen und kann darüber hinaus wie ein Lichtsensor für die Messung der Lichtintensität verwendet werden. Ein LightSensor-Objekt aus der Klassenbibliothek EV3JLibA unterstützt den EV3 ColorSensor.  Der Sensor ist mit Leuchtdioden (LEDs) zur Beleuchtung einer reflektierenden Fläche und einer Fotodiode ausgestattet, welche die Lichtintensität des einfallenden Lichtes messen kann. |  | http://www.java-online.ch/lego/EV3Autonom/bilder/colorsensor.png EV3 ColorSensor |  |

Die Methode getValue() gibt den Wert, welchen der Lichtsensor registriert, zurück. Die Werte liegen zwischen 0 und 1023 (je grösser der Wert, umso heller).

Es gibt grundsätzlich zwei Möglichkeiten, in einem Programm die Sensorwerte zu erfassen:

* Mit Pollen:   
  Die Sensorwerte werden in einer while-Schleife mit der Methode *getValue()* ständig abgefragt. Mit einer while- oder if-Struktur wird geregelt, was geschehen soll, wenn der registrierte Wert einen bestimmten Wert erreicht.
* Mit Java-Event-Modell:   
  Wenn die registrierten Sensorwerte einen bestimmten Wert (*Triggerlevel*) erreicht haben, wird ein Event ausgelöst. Dabei wird eine Callbackmethode aufgerufen, in welcher deklariert wird, wie der Roboter reagieren soll.

**Beispiel 1:** **Auf hell/dunkel reagieren (mit Pollen)**

Der Roboter bewegt sich auf einem Tisch. Wenn er über den Rand fährt und dunkel "sieht", fährt er einen kurzen Weg zurück, dreht um 90° nach links und fährt anschließend wieder vorwärts. Außerhalb des Tisches liegen die vom Lichtsensor erfassten Werte unter ca. 100. Auf dem Tisch gibt die Methode getValue() die Werte, die rund bei 200 liegen, zurück.

**Programmcode**

// LightsensorEx1.java  
  
import ch.aplu.ev3.\*;  
  
class LightSensorEx1  
{  
  LightSensorEx1()  
  {  
    LegoRobot robot = new LegoRobot();  
    LightSensor ls = new LightSensor(SensorPort.S1);  
    robot.addPart(ls);  
    ls.activate(true);  
    Gear gear = new Gear();  
    robot.addPart(gear);  
    gear.forward();  
    while (!robot.isEscapeHit())  
    {  
      int v = ls.getValue();  
      if (v < 100)  
      {  
        gear.backward(1000);  
        gear.left(600);  
        gear.forward();  
      }  
    }  
    robot.exit();  
  }  
  
  public static void main(String[] args)  
  {  
    new LightSensorEx1();  
  }

**Erklärungen zum Programmcode:**

|  |  |
| --- | --- |
| new LightSensor(SensorPort.S1) | Am Sensorport 1 ist ein Color-Sensor angeschlossen, der als Lichtsensor, d.h. zur Messung der Helligkeit, verwendet wird |
|  |  |
| getValue() | Gibt den Sensorwert zurück |
| while(!robot.isEscapeHit()) {} | Der Roboter bewegt sich so lange bis die ESC-Taste auf dem EV3 gedrückt wird. Diese Abbruchbedingung ist wichtig, sonst kann man den Roboter nur mit Herausnehmen der Batterie zum Stillstand bringen. Dabei darf man auch nicht die darauffolgende Zeile robot.exit() nicht vergessen. |
| if (v < 100) | Wenn der Sensor einen kleinen Lichtwert registriert, werden nachfolgende Programmschritte ausgeführt |

**Beispiel 2: Der Roboter wird mit Events gesteuert**

Der Lichtsensor wird durch einen *LightListener* mit einem wählbaren Schwellenwert (*Triggerlevel*) überwacht. Der *LightListener* hat zwei ***Callbackmethoden bright()*** und ***dark()***, die beide implementiert werden müssen. Wird der Schwellenwert unterschritten, so wird die Callbackmethode *dark()* aufgerufen, der Roboter fährt eine kurze Strecke zurück, dreht um 90° nach links und fährt anschließend wieder vorwärts. Die Roboter-Steuerung mit Lichtsensoren bietet eine anschauliche Möglichkeit, Erfahrungen mit dem Java-Event-Model zu sammeln.

**Programmcode**

// LightSensorEx2.java  
  
import ch.aplu.ev3.\*;  
  
class LightSensorEx2 implements LightListener  
{  
  int triggerLevel = 100;  
  Gear gear;  
  
  LightSensorEx2()  
  {  
    LegoRobot robot = new LegoRobot();  
    LightSensor ls = new LightSensor(SensorPort.S1);  
    robot.addPart(ls);  
    ls.addLightListener(this, triggerLevel);  
    gear = new Gear();      
    robot.addPart(gear);  
    gear.forward();  
    Tools.waitEscape();  
    robot.exit();  
  }  
  
  public void bright(SensorPort port, int level)  
  {  
  }  
  
  public void dark(SensorPort port, int level)  
  {

gear.backward(1000);  
    gear.left(600);  
    gear.forward();  
  
  }  
  
  public static void main(String[] args)  
  {  
    new LightSensorEx2();  
  }  
}

**Erklärungen zum Programmcode:**

|  |  |
| --- | --- |
| implements LightListener | LightListener ist ein Interface. Interfaces in Java enthalten Methodendeklarationen ohne Programmcode. Während in Java nur eine einfache Vererbung erlaubt ist, ist es möglich, dass eine Klasse mehrere Interfaces implementiert. Sie muss alle Methoden des Interfaces deklarieren. Da das Interface LightListener zwei Callbackmethoden (bright() und dark()) hat, muss auch eine leere Methode dark() deklariert werden |
| ls.addLightListener(this, triggerLevel) | LightListener wird mit der aktuellen Instanz und dem Triggerlevel registriert, damit der Lightsensor die Callbackmethoden aufrufen kann |
| while (!Tools.isEscapeHit()) {   Tools.delay(1); } | Der Roboter läuft so lange, bis der Escape-Taste auf dem EV3 gedrückt wird. |