

LEGO Mindstorms EV3

Java en LEGO hand in hand?

Sinds september ligt de vierde generatie LEGO-robotkits in de winkel: de Mindstorms EV3. De kern van de kit bestaat uit een programmeerbare besturingsmodule, waarop je een heel scala aan motoren en sensoren kunt aansluiten. De redactie van Java magazine bemachtigde drie van deze LEGO-sets en ging ermee aan de slag. Belangrijkste vraag die we onszelf stelden: draait er ook Java op?

LEGO Mindstorms bestaat sinds 1998. In 2006 kwam Mindstorms NXT uit en in 2009 de NXT 2.0. De EV3 is de vierde en meest geavanceerde variant. Het hart van de set is de Mindstorms 'Brick': een blok met een ARM9-processor en een Linux OS. De Brick heeft een USB-aansluiting, een microSD-slot en een klein display. Het heeft nog het meeste weg van een Gameboy (de meeste lezers zullen die nog wel kennen).

De standaard set bevat onder andere: een aantal servo-motoren, een kleursensor, een druksensor en een infraroodsensor. Via de Nederlandse distributeur kregen wij de Education-kit toegestuurd. Deze kit bevat in plaats van de infraroodsensor een afstands-sensor die werkt op ultrasoon geluid en een gyro-sensor. Bij de set is een uitgebreid softwarepakket meegeleverd. Hiermee kun je de Brick op



Inhoud van de doos

een visuele, drag-and-drop manier programmeren. Het duurt even voordat je daaraan gewend bent, maar dat zou ook te maken kunnen hebben met bepaalde verwachtingen over hoe een IDE (want dat is het eigenlijk) in elkaar zou moeten zitten...

De Education-kit bevat daarnaast uitgebreidere software dan de standaard-set. Docenten kunnen hiermee zelf content voor lesprogramma's samenstellen. Met deze ingrediënten kun je direct aan de slag en het in elkaar zetten van je eerste robot (al dan niet gebruik makend van de meegeleverde instructies) is dan ook zo gebeurd. De bijgeleverde software is bruikbaar, maar als Java-ontwikkelaar blijft er toch een vraag in je achterhoofd hangen: zou Java ook werken?

Mindstorms en Java

Het antwoord hierop is kort maar krachtig: jazeker! Er is zelfs een uitgebreide community rondom het draaien van Java op de Mindstorms-set in de vorm van leJOS: het LEGO Java Operating System.

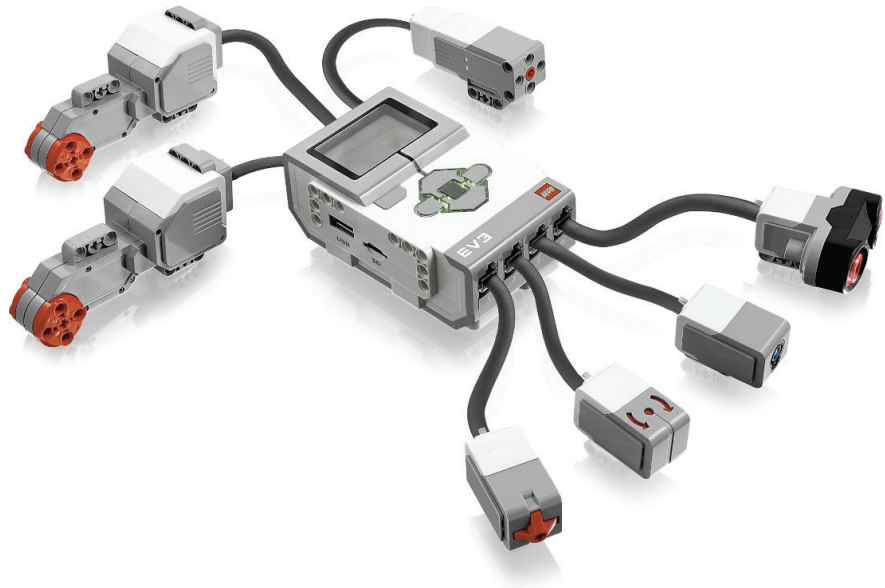
leJOS wordt ontwikkeld sinds 1999 en is een vervanging voor de standaard firmware van de Mindstorms Bricks. leJOS biedt een Java API voor het aanroepen en aansturen van de Mindstorms Brick. Sinds augustus vorig jaar is er support voor de EV3. Deze support is nog experimenteel, maar wel bruikbaar.

Getting started: Java op de EV3

De eerste stap om Java op de Mindstorms EV3 aan de praat te krijgen is het prepareren van een microSD-kaart met een aangepaste Linux-versie inclusief Java 7. Ons boodschappenlijstje daarvoor zag er als volgt uit:

- MicroSD-kaart (1 GB is voldoende)
- Goedkope WiFi dongle (wij hebben een TP-LINK TL-WN723N gebruikt)
- De leJOS Linux-distributie (wij hebben v0.4.0a gebruikt, zie de referenties onderaan dit artikel voor een link)
- Speciale embedded Java-versie voor de EV3 (Oracle speelt blijkbaar ook graag met LEGO): Oracle Java SE Embedded version 7 Update 40, ARMv5 Linux
- Linux-omgeving om de SD-kaart te installeren (bijvoorbeeld Ubuntu)

De eerste stap is het prepareren van de SD-kaart volgens de stappen op de leJOS-wiki. Daarna kun je de SD-kaart in je EV3 stoppen, waardoor hij nu opstart met de speciale Linuxdistributie en Java. Je hebt twee mogelijkheden om verbinding te maken met de Brick: via een USB WiFi-dongle of via een



De EV3 Brick met 3 servomotoren en sensoren: druk, gyro, kleur en infrarood

USB-kabel. In het tweede geval herkent een Linux PC de Brick als tweede netwerkinterface. Als je via WiFi werkt, probeert de EV3 verbinding te maken met het netwerk dat je in wpa_supplicant.conf op het Linux-filesystem geconfigureerd hebt. Als het lukt om te verbinden, start het apparaat op en zie je het verkregen IP-adres op het display. Je kunt vervolgens verbinding maken met de EV3 via telnet of SSH. Lukt verbinden niet, dan blijven de rode leds op de Brick branden.

De leJOS API

Zoals het elke rechtgeaarde programmeur betaamt, begin je natuurlijk met het schrijven van een 'Hello World' applicatie. Dat blijkt relatief eenvoudig:

```
import lejos.nxt.Button;
import lejos.nxt.LCD;

public class HelloIvoNet {

    public static void main(String[] args) {
        LCD.clear();
        LCD.drawString("Hello IvoNet", 0,
            5);
        Button.waitForAnyPress();
        LCD.clear();
        LCD.refresh();
    }
}
```



Bert Jan Schrijver is
Redacteur voor Java
magazine



Jeroen Resoort is
Application architect
bij TNT



Ivo Woltring is IT
architect bij Ordina

Naast de standaard functionaliteit om motoren, sensoren, display, knoppen en geluid te gebruiken, biedt leJOS nog meer. De leJOS libraries zijn er namelijk al veel langer. Ze bestaan ook voor de RCX- en NXT Mindstorms-sets. Er zijn libraries om bijvoorbeeld kaarten te tekenen en libraries voor kortste pad algoritmen. Wanneer je met wat geavanceerdere robots aan de slag gaat, komt dat goed van pas.

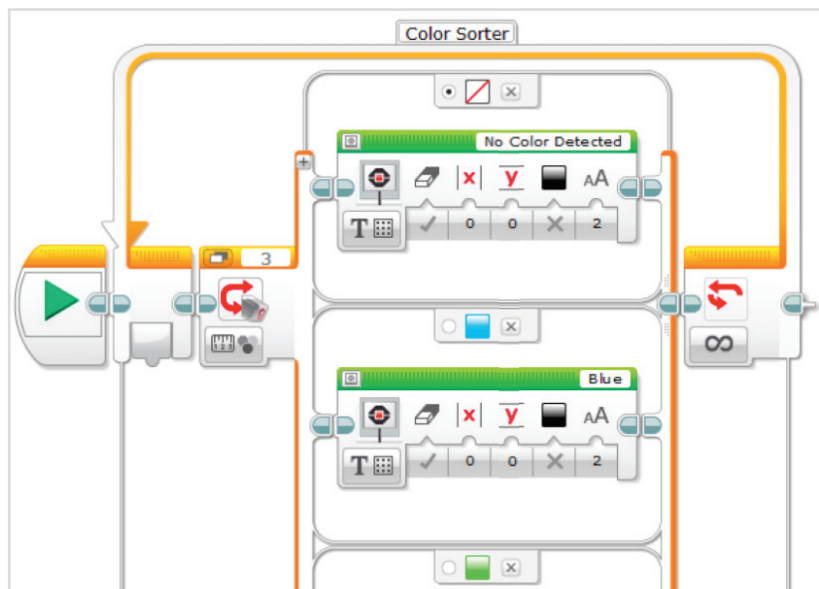
EV3's aan een gedistribueerde Event Bus

'Hello World' is natuurlijk leuk, maar we hebben drie EV3's tot onze beschikking. Het is dus tijd om een stapje verder te gaan. We willen ze alle drie tegelijk op afstand aansturen. Dit doen we met behulp van Vert.x; een lichtgewicht, polyglot applicatie-platform. Vert.x biedt standaard een paar hele handige features, waarmee we onze EV3's op afstand kunnen aansturen. Vert.x heeft een HTTP-server (op basis van Netty) en een gedistribueerde event-bus (gebaseerd op Hazelcast). Hazelcast bouwt een cluster op waarbinnen we events kunnen delen. Elke node in onze cluster is een EV3 of een andere deelnemende applicatie. We hebben het cluster met multicast geconfigureerd, zodat de verschillende nodes elkaar zelf kunnen vinden. We hoeven dan geen administratie van IP-adressen bij te houden, want de nodes registreren zichzelf. Het opstarten van het cluster verloopt op deze manier lekker vlot. Omdat we IP-adressen uitdelen via WiFi, moeten we de IP-range van het WiFi-netwerk expliciet vermelden. Dit resulteert in de volgende Hazelcast cluster.xml configuratie (zie listing 2).

Tenslotte moeten we Vert.x laten weten, dat

```
<network>
...
<port auto-increment=>true>5701</port>
<join>
  <multicast enabled="true">
    <multicast-group>224.2.2.3</multicast-group>
    <multicast-port>54327</multicast-port>
  </multicast>
  <tcp-ip enabled="false">
    ...
  </tcp-ip>
  <aws enabled="false">
    ...
  </aws>
</join>
<interfaces enabled="true">
  <interface>192.168.1.*</interface>
</interfaces>
...
</network>
```

Listing 2



Een greep uit de bijgeleverde 'IDE'

we op een specifiek IP-adres luisteren. Dat is nodig, omdat Vert.x anders alleen lokaal

```
Vertx vertx = VertxFactory.newVertx("192.168.1.100");
```

werkt. Er wordt dan geen cluster gevormd. Om zeker te weten dat onze configuratie werkt, bouwen we een simpele chat-applicatie. Deze applicatie stuurt een bericht via Vert.x, zodra er een knop is ingedrukt op één van de EV3-controllers. Zo weten alle nodes welke knop op een specifieke EV3 is ingedrukt.

Swarming EV3's

Nu we weten dat de clusterconfiguratie werkt, schrijven we een nieuwe applicatie die

**HET IS NIET
LASTIG OM JAVA
APPLICATIES TE
DRAAIEN MET
BEHULP VAN
JAVA EN DE
LEJOS IMPLI-
MENTATIE,
MAAR HET
VERGT WEL WAT
CONFIGURATIE.**

een motor aanstuurt, zodra er een event via Vert.x binnenkomt. De events zijn simpele Strings zoals: vooruit, links, rechts en achteruit. De software vertaalt de events naar commando's, die de motoren aansturen. De leJOS API werkt erg intuïtief en heeft geen verdere toelichting.

Een client op een PC stuurt de commando's naar de verschillende EV3-controllers. Nu we de software draaiend hebben, rest er niets anders dan onder het genot van een potje bier drie rijdende robots te bouwen. We kunnen ze nu alle drie synchroon aansturen. Met dit voorbeeld hebben we het principe van communicatie en de aansturing van de motoren aangetoond. De sensoren werken op een zelfde manier. Er mist nu nog de vereiste logica om de verschillende robots autonoom te laten rijden of ze op elkaars sensor input te laten reageren. Dat laten we als oefening aan de lezer over ;-)

Tenslotte

Een LEGO Mindstorms EV3 is heel leuk, maar wel prijzig speelgoed. De kit is erg robuust. Het geeft een leuke en zeer leerzame introductie in procestechologie. Met LEGO (en eventueel met je kinderen) kun je eindeloos veel creaties bouwen. De meegeleverde software is wel wat minder geschikt om complexere processen uit te programmeren. Het is niet lastig om Java applicaties te draaien met behulp van Java en de leJOS-implementatie, maar het vergt wel wat configuratie. Het is erg mooi dat de embedded versie van Java (nagenoeg) alle features heeft van een gewone Java SE runtime en dat de controller krachtig genoeg is om Java-frameworks zoals Vert.x te draaien. Dit vergroot de (communicatie)mogelijkheden enorm. Voor het uploaden en draaien van de software moet je wel wat geduld hebben, want dit werkt vrij traag. Niet alle USB WiFi-dongles worden ondersteund door leJOS. Dit is afhankelijk van de chipset. Helaas staat de chipset lang niet altijd vermeld als je een dongle koopt. Je hebt de meeste kans met een goedkope dongle, die ook door een Raspberry Pi wordt ondersteund. De prijs van de kit is een behoorlijke drempel, want die ligt namelijk tussen de €300 en €400. Dit maakt de afweging tussen het aanschaffen van een Mindstorms-set of een Raspberry Pi met wat motoren en sensoren lastig. Bij een Raspberry Pi met doe-het-zelf-materiaal is de leercurve wat hoger als je geen achtergrond in elektronica hebt,

```
192.168.1.100 - PuTTY
INFO: [192.168.1.100]:5701 [dev] Connecting to master node: Address[192.168.1.101]:5701
Jan 01, 2000 12:14:25 AM com.hazelcast.cluster.ClusterManager
INFO: [192.168.1.100]:5701 [dev]
Members [2] {
  Member [192.168.1.101]:5701
  Member [192.168.1.100]:5701 this
}
Jan 01, 2000 12:14:46 AM com.hazelcast.impl.LifecycleServiceImpl
INFO: [192.168.1.100]:5701 [dev] Address[192.168.1.100]:5701 is STARTED
Vert.X launched
I received a message Hello world from EV3
I received a message EV3 buttons pressed: Right
I received a message EV3 buttons pressed: Right
I received a message EV3 buttons pressed: Right
I received a message EV3 buttons pressed: Left
I received a message EV3 buttons pressed: Left
I received a message EV3 buttons pressed: Left
I received a message EV3 buttons pressed: Left
I received a message EV3 buttons pressed: Up
I received a message EV3 buttons pressed: Up
I received a message EV3 buttons pressed: Enter
I received a message EV3 buttons pressed: Enter
I received a message EV3 buttons pressed: Enter
I received a message EV3 buttons pressed: Down
I received a message EV3 buttons pressed: Enter
I received a message EV3 buttons pressed: Up
I received a message EV3 buttons pressed: Enter
```

Screenshot van de applicatie

maar op de lange termijn heb je er misschien meer plezier van. De investering is dan niet zozeer in geld, maar eerder in tijd. Wij hebben ons in ieder geval erg goed met de LEGO-kits vermaakt. ■



REFERENTIES

leJOS: <http://www.lejos.org>

Prepareren van een SD-kaart: <http://bit.ly/LAwzBY>

Java SE Embedded voor de EV3: <http://bit.ly/KtoLhg>

leJOS voor de EV3: <http://bit.ly/1eK7KeS>

Vert.x: <http://vertx.io>

Speciale dank aan Eelco Meuter voor zijn hulp bij het project.