

# EINFÜHRUNG LEGO MINDSTORMS

## LabVIEW 2016

Prof. Dr.-Ing. Dahlkemper

### 1 NXT DIRECT COMMANDS

Es gibt zwei verschiedene Möglichkeiten, die Lego-Roboter zu programmieren: Man kann das Programm direkt auf den Microcontroller des Roboters laden oder man nutzt die sogenannten *Direct Commands*. Die *Direct Commands* bieten den Vorteil, dass das eigentliche Programm auf dem PC läuft und man die gesamte Funktionalität von LabVIEW nutzen kann.

Im Rahmen dieses Kurses steht die Vermittlung der graphischen Programmierung mit LabVIEW im Vordergrund. Daher beschränkt sich die Programmierung auf den Einsatz der *Direct Commands*.

#### 1.1 VERBINDUNG

Um mit den *Direct Commands* zu arbeiten, erstellt man ein herkömmliches VI über **New VI** und **nicht** über **New NXT/EV3** und muss in einem ersten Schritt den gewünschten NXT oder EV3 Roboter spezifizieren. Hierzu wählt man aus der Functions Palette das Menu **MINDSTORMS Robotics > I/O > DirectOnly > Specify NXT/EV3**.

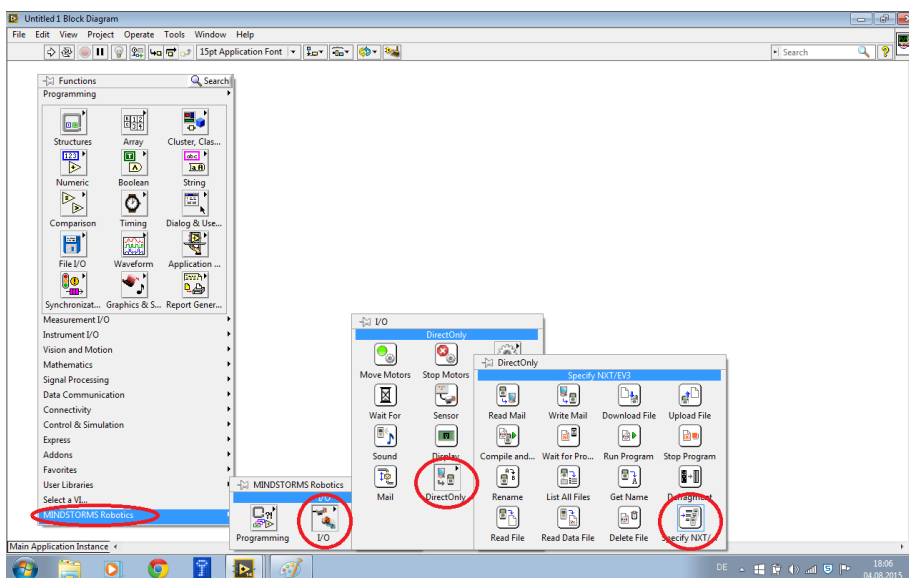


Abb. 1: Herstellen der Verbindung zu dem Roboter

Hierbei kann man auswählen, ob der Mindstorms Roboter über Bluetooth oder USB angeschlossen wird. Bei einer Bluetooth-Verbindung kann zwar drahtlos mit dem Roboter kommuniziert werden, was sich für mobile Roboter anbietet. Allerdings gibt es eine Latenzzeit von fast 60 Sekunden zwischen Compilieren und Ausführen des Programmes. Daher ist für die Entwicklung dringend von der Nutzung der Bluetooth-Verbindung abzuraten. Erst wenn das Programm vollständig getestet ist, kann man anschließend auf die Bluetooth-Verbindung umschalten.

Hierzu ist bei beiden Eingängen „Connection Type“ und „NXT Name“ mit einem Rechtsklick und *Create* → *Constant* eine Konstante zu erstellen. Der Verbindungstyp muss auf USB stehen, der Name ist auf den im Roboterdisplay angezeigten Roboternamen einzustellen.

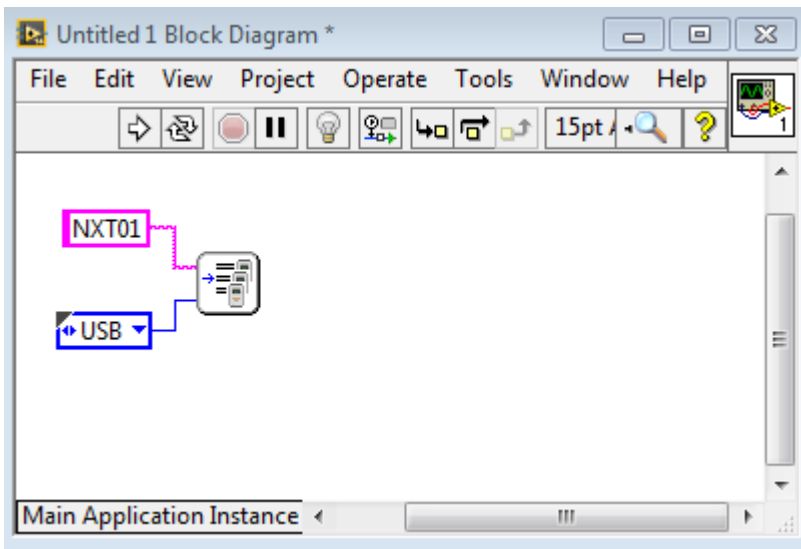


Abb. 2: Bezeichnung des Roboters muss mit dem Namen im Roboterdisplay übereinstimmen

## 1.2 MOTOR DREHEN UND ANHALTEN

Zum Drehen eines Motors benötigt man die Funktion *Move Motors*, die in der *Function Palette* unter *MINDSTORMS Robotics > I/O > Move Motors* zu finden ist.

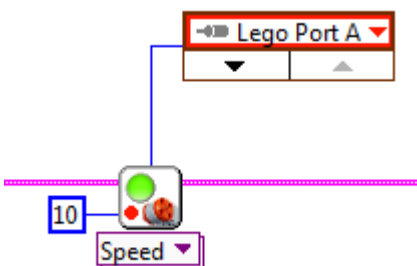


Abb. 3: Motor starten

Oberhalb des Funktionsblocks lässt sich mittels der rechten Maustaste eine Konstante oder ein *Control* (Steuerfeld im *Front Panel*) für den jeweiligen Port A, B, C, ... erstellen, an den der Motor angeschlossen ist. Dazu ist das Selektionsfeld „Unselected“ auf den gewünschten Motorport einstellen. Darüber hinaus kann die Motorauswahl auch über die Pfeiltaste auf mehrere Motoren ergänzt werden. Über den linken Eingang Power/Speed kann die Geschwindigkeit und

Drehrichtung im Bereich von -100 bis + 100 eingestellt werden. Es besteht über die untere Auswahlbox die Möglichkeit, eine Geschwindigkeitsregelung (*Speed*) oder eine Stromregelung (*Power*) zu aktivieren.

Erreicht der Programmfluss nun durch dieses Objekt, so beginnt sich der Motor zu drehen.

Um den Motor zu stoppen, erstellt man die Funktion *Stop Motors* aus der *Function Palette* unter *MINDSTORMS Robotics > I/O > Stop Motors*.

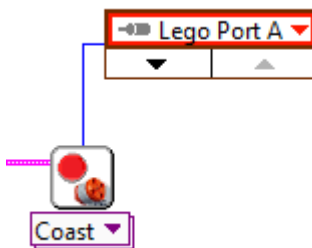


Abb. 4: Motor anhalten

Auch hier ist als Input anzugeben, welcher Motor gestoppt werden soll. Man kann den Motor über die Betriebsart *Coast* ausrollen lassen oder abrupt über *Brake* zum Stoppen bringen.

### 1.3 INKREMENTALGEBER DES MOTORS

Alle drei Motoren besitzen Inkrementalgeber. Sie speichern den zurückgelegten Winkel der Motoren in Grad. Bevor man den Wert das erste Mal auslesen kann, sollte man ihn zunächst auf 0 setzen. Dies geschieht durch das Einfügen des Objektes *Reset Encoders* aus der *Function Palette* unter *MINDSTORMS Robotics > I/O > Advanced Motors > Reset Encoders*

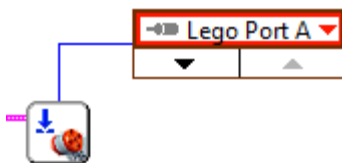


Abb. 5: Drehgeber zurücksetzen

Der Wert des Encoders kann mit Hilfe der Funktion *Motor Status* abgefragt werden:

*MINDSTORMS Robotics > I/O > Advanced Motors > Motor Status*

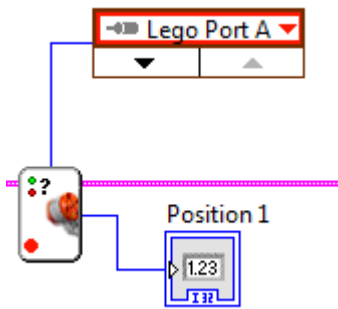


Abb. 6: Drehgeber auslesen über Motor Status

Alternativ kann die Position des Motors auch über den universellen Sensor-Block und Auswahl der Option „Rotation“ abgefragt und zurückgesetzt werden: MINDSTORMS ROBOTICS > I/O > Sensor mit der Auswahl „Read Rotation“ oder „Reset Rotation“.

## 1.4 BESCHLEUNIGUNGSSENSOR

Der Beschleunigungssensor wird von der Firma HiTechnic produziert. Daher muss dieser entsprechend eingebunden werden:

*MINDSTORMS Robotics > I/O > Additional Sensors > HiTechnic Sensors > HT Read*

Anschließend kann man die Funktion *HT Read* für die Messaufgabe konfigurieren. Dazu ist die Betriebsart *Read Accelerometer* im Dropdown-Menü der Funktion auszuwählen. Da der Beschleunigungssensor über drei Messachsen x, y und z verfügt, ist anschließend die gewünschte Messachse auszuwählen. Die Ausgabe erfolgt als Fließkommazahl in der Einheit  $m/s^2$ .

Wenn mehrere Achsen ausgelesen werden sollen, ist es am recheneffizientesten, alle drei Achsen gleichzeitig als Array auszugeben. Auf die einzelnen Werte kann man mittels der Array Funktion `Array > Index Array` zugreifen.

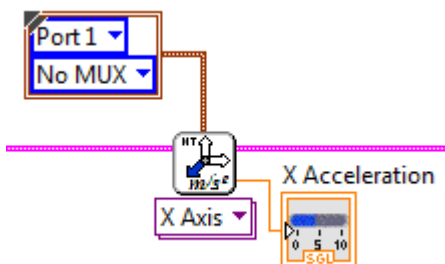


Abb. 7: HiTechnics Beschleunigungssensor über HT Read auslesen

## 1.5 DREHRATENSOR (GYRO)

Der Drehratensensor der Firma HiTechnic (Gyro) wird ebenso wie der Beschleunigungssensor über die folgende Funktion abgefragt:

*MINDSTORMS Robotics > I/O > Additional Sensors > HiTechnic Sensors > HT Read*

Bei der Auswahl der Betriebsart ist entsprechend *Read Gyro* zu wählen.

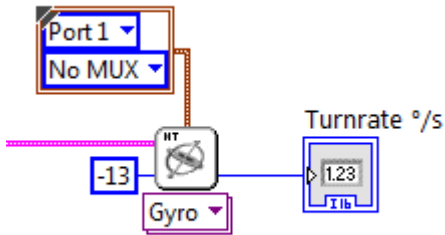


Abb. 8: Gyro-Sensor über HT Read auslesen

In Ruhelage liefert der Sensor einen leicht von Null abweichenden Wert. Dieser Wert unterscheidet sich von Sensor zu Sensor und verändert sich auch im Laufe der Betriebszeit u.a. aufgrund der Erwärmung des Sensors. Durch die Wahl eines geeigneten Wertes für den Eingang *Offset* ist sicherzustellen, dass der Sensor in Ruhelage auf einen Ausgabe-wert von Null kalibriert wird.

Der Sensor liefert die Drehrate in Grad pro Sekunde. Zuverlässige Werte lassen sich bei Drehgeschwindigkeiten unterhalb von  $\pm 360^\circ/\text{Sekunde}$  erzielen. Bei schnelleren Drehraten ergeben sich große Abweichungen, die insbesondere bei der Winkelberechnung über eine Integration auffallen.

Laut Datenblatt weist der Sensor eine Messrate von 3,3 ms auf. Es hat sich bewährt, die Werte mit 10 bis 50 Hz auszu-lesen.

## 1.6 ANDERE SENSOREN

Das Auslesen der Lego-Sensoren, z.B. Ultraschallsensor, Reflexlichttaster oder Taster erfolgt über die Funktion

*MINDSTORMS Robotics > I/O > Sensor*

Diese Funktion ist über das Dropdown-Menü auf den jeweiligen Sensor anzupassen. Das folgende Beispiel verwendet den Ultraschallsensor und muss entsprechend mit *Read EV3 Ultrasonic > CM* (Vorsicht: Nicht *Read NXT Ultrasonic*, dies ist für die Vorgängerversion der Sensoren) konfiguriert werden.

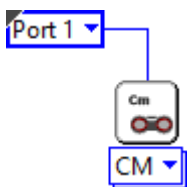


Abb. 9: Lego Sensor über die Funktion Sensor auslesen

Nach dem Erstellen des Sensor-Objektes muss lediglich der Port beispielsweise über eine Konstante eingestellt werden. Der Ausgang liefert die gemessene Entfernung in cm.

## 2 BEISPIELPROGRAMM

Das folgende Beispielprogramm bewirkt, dass sich der Motor dreht und der Drehgeber ausgelesen wird. Beim Drücken einer Stop-Taste läuft der Motor aus. Ein Drücken des Reset-Tasters bewirkt, dass der Drehgeber auf Null zurückgesetzt wird.

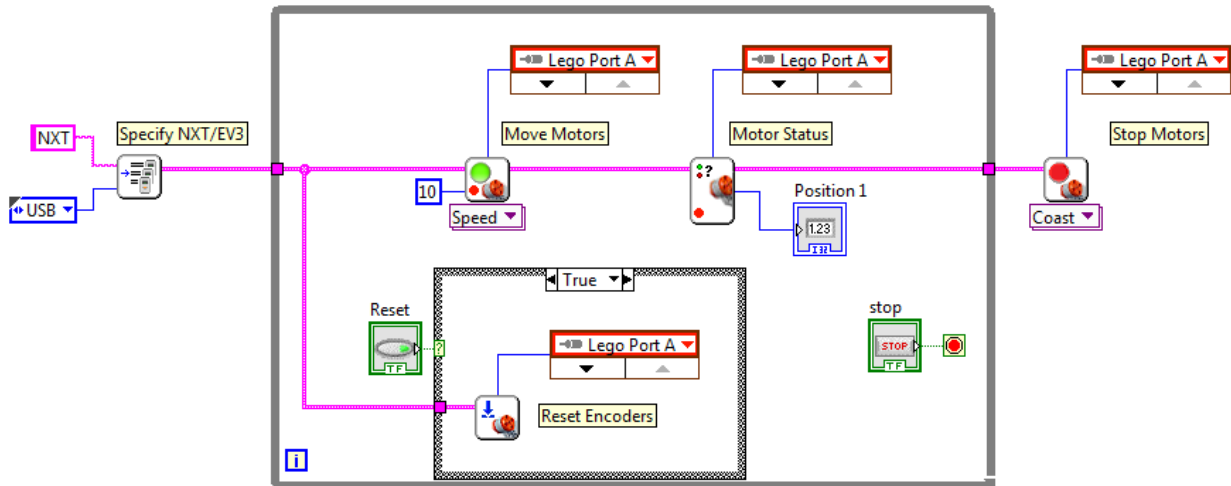


Abb. 10: Beispielprogramm zur Ansteuerung eines Motors mit Auslesen des Encoders