

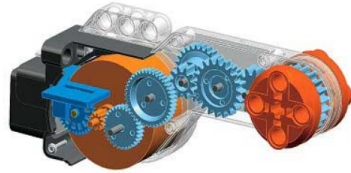
NNXT - Tutorials

[Startseite](#)
[Tutorial #01](#)
[Tutorial #02](#)
[Tutorial #03](#)
[Tutorial #04](#)
[Tutorial #05](#)
[Tutorial #06](#)

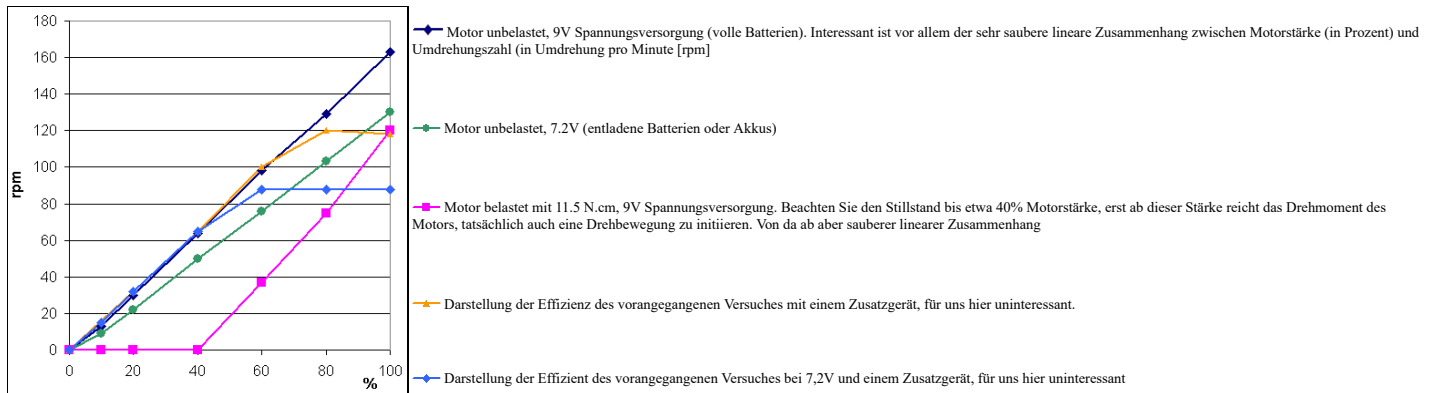
Tutorial #05 - Drehzahlsensoren

Grundsätzliches

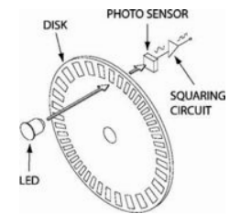
Die Motoren des NXT-Legosystems enthalten neben dem reinen Elektromotor auch noch ein Getriebe, dass eine entsprechende Über/Untersetzung auf die Drehachse gibt. Eine Vorstellung über den Aufbau der Motoren kann dem folgenden Bild entnommen werden:



Die Motoren der NXT-Baureihe verfügen insgesamt über eine Untersetzung, d.h. die Umdrehungsgeschwindigkeiten auf der Drehachse, an der die Räder aufmontiert sind, ist relativ gering verglichen mit anderen Elektromotoren. Der Vorteil dieser Bauweise liegt in dem größeren Drehmoment, was damit auf der Drehachse erzeugt werden kann und somit können auch schwere Aufbauten relativ zuverlässig bewegt werden. Die maximal zu erreichende Drehzahl hängt zum einen natürlich an der Motorstärke, die über die Motorfunktionen **Motor_Drive()** übergeben werden, zum anderen an der Güte der Spannungsversorgung (die Batterien werden nach einigem Gebrauch entladen und damit weniger Spannung liefern) und an der Last, gegen die der Motor arbeiten muss. Typische Kennlinien der Motoren ergeben das folgende Bild (angegeben ist hier die Umdrehungszahl der Drehachse in Umdrehungen pro Minute [rpm] in Abhängigkeit von der eingestellten Motor-Stärke in Prozent):



Was lernen wir aus diesen Kurven? Wenn Sie die Motorstärke über **Motor_Drive()** einstellen, wissen Sie noch lange nicht, wie schnell sich die Achse, an der Sie die beweglichen Teile angeschlossen haben, in Wirklichkeit dreht; dies hängt von sehr vielen Faktoren ab, die mitunter gar nicht vorherzusagen sind (z.B. Batteriespannung). Da man aber häufig eine genaue Drehzahl einstellen muss (z.B. beim Einhalten einer Geschwindigkeit des als Fahrzeug aufgebauten Roboters), müssen wir die Drehzahl des Motors an der Drehachse messen können. Hierzu ist ein Drehsensor in den Motor eingebaut, mit dem die Umdrehungszahl indirekt gemessen werden kann. Das Grundprinzip dieses Drehsensors gleicht dem einer Lichtschranke: In einem Zahnrad an dem Motor sind im Wechsel lichtdurchlässige und lichtundurchlässige Schichten eingebaut, die mit einem Lichtstrahl bestrahlt wird, der auf der anderen Seite des Zahnrads auf einen lichtempfindlichen Sensor fällt. Bei Drehung des Zahnrads ergibt sich damit an dem Lichtsensor ein stetiger Signalwechsel von Licht an/Licht aus, der gemessen werden kann. Die folgende Abbildung macht dieses Verfahren deutlich:



Bei der Messung im Motor des Lego-Roboters werden diese Wechsel von Licht an/Licht aus gezählt und dieser Zählstand kann nun ausgelesen werden. Aus diesem Wert kann geschlussfolgert werden, um wieviel Grad sich das Zahnrad mit den wechselnden Lichtdurchlässigen Schichten gedreht hat und daraus kann wiederum die Umdrehungsgeschwindigkeit in Grad pro Zeiteinheit errechnet werden. Der Aufbau der Lego-Motoren ist so gestaltet, das ein um 1 höherer Wert des Zählers bedeutet, dass sich die Drehachse (also dort, wo die Räder angebracht sind) um N° gedreht hat. Unglücklicherweise ist diese Zahl N nicht bei jedem Motor die gleiche. Sie müssen also diese Zahl für jeden Motor bestimmen. Der Beispielwert in unten stehender Beispielapplikation hat sich an dem betrachteten Roboter an beiden Motoren mit $N=1,33^\circ$ ergeben, an Ihrem Roboter wird dieser Wert aber ein anderer sein und muss gemessen werden (am einfachsten geschieht dies durch einen an das Rad angebracht langen Legostein, den Sie eine Weile drehen lassen, den dabei zurückgelegten Wert in Grad messen (z.B. 2 1/4 Drehungen entsprechen dann 710°) und dabei die Differenz der Werte des Drehsensors vom Start der Bewegung bis zum Ende der Bewegung ermitteln; stoppen Sie die Motoren dann mit brake und nicht mit float, damit sie durch ihren Schwung nicht nachlaufen).

Die Drehzahlsensor-API

Der Zähler des Drehzahlsensors kann pro Motor (an Port_A, Port_B oder Port_C) mit der folgenden Funktion ausgelesen werden:

```
sensor_error_t Motor_Tacho_GetCounter(const motorport_t port, uint32_t *counter);
```

Wie bei Sensorfunktionen üblich erfolgt die Rückgabe über einen Call-by-reference Parameter (hier **counter**). Der Datentyp ist dabei ein 32-Bit unsigned integer. Als Fehlerwert liefert die Funktion grundsätzlich **sensor_error_NoError**, so dass Sie den Rückgabewert ignorieren dürfen.

Mittels dieser Funktion kann man nun also die Umdrehungsgeschwindigkeit eines Motors messen, wie im unten angegebenen Beispiel zu sehen ist (hier wird alle 500ms gemessen und die Umdrehungsgeschwindigkeit des Motors wird in Umdrehungen pro Minute in der Variablen **rpm** errechnet und auf dem Display ausgegeben):

```

#include "nnxt.h"
#include <stdio.h>

#define DEG_TO_RPM 166.6667          // Umrechnungsfaktor [°/ms] in [U/min]

int main() {
    char rpmMsg[10];                // enthält Umdrehung/min als String für Ausgabe auf Display
    uint32_t prev_deg, deg;         // deg enthält Differenz vom Zähler jetzt und bei der letzten Messung vor 500ms (prev_deg)
    uint32_t dt, prev_tick = GetSysTime(); // dt enthält Differenz der Zeiten jetzt und der Zeit der letzten Messung (prev_tick)
    double rpm;                     // enthält Wert der Umdrehungen pro Minute

    MotorPortInit(Port_A);
    Motor_Tacho_GetCounter(Port_A, &prev_deg);
    Motor_Drive(Port_A, Motor_dir_forward, 20); // hier gerne mal anpassen und probieren
    while(1) {
        Delay(500);                // Abtastintervall 500ms
        dt = GetSysTime() - prev_tick; // Zeitdifferenz seit letzter Messung ermitteln
        Motor_Tacho_GetCounter(Port_A, &deg);
        deg = deg - prev_deg;       // um wieviel ist der Umdrehungszähler seit letzter Messung gewachsen
        rpm = (deg*1.33) * DEG_TO_RPM / (double)dt; // deg*1.33 weil Auflösung Drehzahlsensor 1 Digit entspr. 1,33 Grad
        Motor_Tacho_GetCounter(Port_A, &prev_deg); // Zählerstand für nächste Messung vorbereiten
        prev_tick = GetSysTime(); // Zeitmessung für nächste Messung vorbereiten
        sprintf(rpmMsg, "%d ", (int)rpm);
        NNXT_LCD_DisplayStringAtLine(0, rpmMsg);
    }
    return 0;
}

```

Der Umrechnungsfaktor **DEG_TO_RPM** kommt folgendermaßen zustande: **deg*1.33** ergibt die Anzahl an °, die das Rad sich seit der letzten Messung gedreht hat. Dies geteilt durch **dt** (was Näherungsweise 500 sein wird wg. **Delay (500)**) ergibt die Umdrehungsgeschwindigkeit in °/ms. Um dies in °/min um zu rechnen, muss der Wert mit 1000 (auf Sekunden) und dann nochmals mit 60 (auf Minuten) multipliziert werden. Danach hat man die Umdrehungsgeschwindigkeit in °/min. Da wir aber Umdrehungen pro Minute haben möchten und eine Umdrehung des Rades 360° umfasst, muss der Wert durch 360 geteilt werden, so dass sich ergibt: $1000*60/360 = 166.6667$.

Die Zeiterfassung wurde bei der Berechnung der Umdrehungsgeschwindigkeit hier allgemein über die Systemzeit und **GetSysTime()** erledigt, damit das Programm flexibler ist; es reicht jetzt die Wartezeit in der **Delay**-Funktion zu verändern und die Berechnung muss dann nicht angepasst werden, da die Zeitdifferenz seit der letzten Messung aus der Differenz der Systemzeiten ermittelt wird.

Variieren Sie bei gerne mal die Geschwindigkeit und schauen Sie bei langsamen Geschwindigkeiten auf das Rad und versuchen abzuschätzen, ob der angegebene Wert auf dem Display ungefähr stimmt.

WICHTIG: Der Drehzahlsensoren liefert bis zu einem Geschwindigkeitswert von 80% verlässliche Drehzahlwerte! Daher sollten Sie, bei Verwendung des Drehzahlsensors, den Motor nur zwischen 0% und 80% betreiben.