






# Kompetenzraster „Roboter und Software“



Total: \_\_\_\_\_ / 50 P. (LZ erreicht mit 30 P.) Note: \_\_\_\_\_ Visum Eltern: \_\_\_\_\_

Kompetenzbereiche	 0 P.	 5 P.	 8 P.	 10 P.
<b>Robustheit und Stabilität</b> <i>Wie robust und stabil ist der Roboter gebaut? Ist die Hauptstruktur des Roboters zweckmässig und technisch elegant konstruiert?</i>	Der Roboter ist fragil oder zerfällt sogar während der Fahrt und bleibt deswegen stecken. Die Konstruktion ist instabil gebaut und verunmöglicht genaue, wiederholbare Manöver.	Wenige unwichtige Teile sind beweglich oder lose, aber der Roboter bleibt noch funktionsfähig. Die Konstruktion ist zu aufwändig mit zu vielen unnötigen Teilen gebaut. Manöver lassen sich meistens gut kontrollieren.	Der Roboter ist in wesentlichen Teilen stabil gebaut und kann seine Aufgabe zuverlässig erfüllen. Die Konstruktion ist zweckmässig und ermöglicht genaue Manöver..	Der Roboter ist mit einem Minimum an Teilen sehr solide und stabil gebaut. Die Konstruktion ist technisch elegant und ermöglicht sehr präzise Manöver.
<b>Mechanik und Design</b> <i>Wie gut funktionieren die beweglichen Teile des Roboters?</i>	Einfache Mechanik wird nur ansatzweise eingesetzt. Der Roboter kann sich schlecht oder gar nicht bewegen. Das Design ist klobig.	Zahnradgetriebe und andere Mechanismen funktionieren meist gut, unterstützen die Beweglichkeit des Roboters aber nur teilweise. Das Design ist etwas klobig, erfüllt aber seinen Zweck.	Zahnradgetriebe und andere Mechanismen unterstützen die Beweglichkeit des Roboters und laufen reibungslos. Das Design ist durchaus zweckmässig.	Neben Zahnradgetrieben und anderen Mechanismen werden weitere Hilfsstrukturen (Arme, usw.), eingesetzt, um die Effizienz des Roboters zu maximieren. Alle Getriebe und Mechanismen bewegen sich einwandfrei.
<b>Softwaredesign</b> <i>Wird der Funktionsumfang der Programmiersprache NEPO optimal ausgeschöpft?</i>	Im Programm werden keine Schleifen oder Verzweigungen eingesetzt.	Im Programm werden einfache Schleifen und Verzweigungen meist sinnvoll eingesetzt. Variablen oder Mathematik-Blöcke sind nicht sinnvoll verwendet worden oder fehlen ganz. Konstruierter Ro-Bot mit nur einfachen Teilen.	Im Programm werden Schleifen und Verzweigungen sinnvoll kombiniert eingesetzt. Über Variablen und Mathematik-Blöcke werden vereinzelt Sensorwerte weiterverarbeitet.	Im Programm werden komplexe und elegante Kombinationen aus Schleifen und Verzweigungen eingesetzt. Mit Variablen, Mathematik- und Logik-Blöcken werden Sensorwerte weiterverarbeitet. Funktions-Blöcke werden als Unterfunktionen an geeigneter Stelle verwendet.
<b>Funktionalität</b> <i>Sind Software und Hardware gut auf einander abgestimmt?</i>	Hardware und Software arbeiten nicht gut zusammen. Das Programm steuert die meisten wesentlichen Komponenten nicht an.	Das Programm steuert die meiste Hardware an, aber einige Komponenten funktionieren nicht zuverlässig oder werden vom Programm gar nicht verwendet.	Das Programm verwendet die Komponenten des Roboters wie vorgesehen. Alle Komponenten funktionieren zuverlässig und erfüllen ihre Aufgaben.	Das Programm steuert alle Komponenten präzise an. Die Software ist flexibel genug gestaltet, um leicht weitere Teile hinzuzufügen und bei Bedarf verwenden zu können.
<b>Effizienz und Innovation</b> <i>Kann der Roboter durch innovatives Zusammenspiel von mechanischem Design und Softwaredesign die Aufgabe gut lösen? Wird der Roboter dadurch vielseitig einsetzbar?</i>	Die Software kann Schlüsselemente der gestellten Aufgabe nur teilweise oder gar nicht ausführen. Das Roboterdesign ist für die gestellte Aufgabe nicht geeignet.	Die Software erfüllt die grundlegenden Anforderungen und reagiert auf einzelne, einfache Änderungen. Das Roboterdesign ist wenig innovativ, eignet sich aber, genau die gestellte Aufgabe im Wesentlichen lösen.	Die Software erfüllt alle wesentlichen Anforderungen und reagiert auf einfache Änderungen. Das Roboterdesign zeigt interessante Ansätze und eignet sich gut, die gestellte Aufgabe zu lösen.	Der Roboter reagiert auf unvorhersehbare Änderungen und die Software funktioniert fehlerlos. Das Roboterdesign überrascht mit innovativen Ansätzen und kann die gestellte Aufgabe hervorragend lösen. Der Roboter kann auch für neue Aufgaben verwendet werden.



# Kompetenzraster „Roboter und Software“

Total: \_\_\_\_\_ / 50 P. (LZ erreicht mit 30 P.) Note: \_\_\_\_\_ Visum Eltern: \_\_\_\_\_

Kompetenzbereiche	 0 P.	 5 P.	 8 P.	 10 P.
<b>Robustheit und Stabilität</b> <i>Wie robust und stabil ist der Roboter gebaut? Ist die Hauptstruktur des Roboters zweckmässig und technisch elegant konstruiert?</i>	Der Roboter ist fragil oder zerfällt sogar während der Fahrt und bleibt deswegen stecken. Die Konstruktion ist instabil gebaut und verunmöglicht genaue, wiederholbare Manöver.	Wenige unwichtige Teile sind beweglich oder lose, aber der Roboter bleibt noch funktionsfähig. Die Konstruktion ist zu aufwändig mit zu vielen unnötigen Teilen gebaut. Manöver lassen sich meistens gut kontrollieren.	Der Roboter ist in wesentlichen Teilen stabil gebaut und kann seine Aufgabe zuverlässig erfüllen. Die Konstruktion ist zweckmässig und ermöglicht genaue Manöver..	Der Roboter ist mit einem Minimum an Teilen sehr solide und stabil gebaut. Die Konstruktion ist technisch elegant und ermöglicht sehr präzise Manöver.
<b>Mechanik und Design</b> <i>Wie gut funktionieren die beweglichen Teile des Roboters?</i>	Einfache Mechanik wird nur ansatzweise eingesetzt. Der Roboter kann sich schlecht oder gar nicht bewegen. Das Design ist klobig.	Zahnradgetriebe und andere Mechanismen funktionieren meist gut, unterstützen die Beweglichkeit des Roboters aber nur teilweise. Das Design ist etwas klobig, erfüllt aber seinen Zweck.	Zahnradgetriebe und andere Mechanismen unterstützen die Beweglichkeit des Roboters und laufen reibungslos. Das Design ist durchaus zweckmässig.	Neben Zahnradgetrieben und anderen Mechanismen werden weitere Hilfsstrukturen (Arme, usw.), eingesetzt, um die Effizienz des Roboters zu maximieren. Alle Getriebe und Mechanismen bewegen sich einwandfrei.
<b>Softwaredesign</b> <i>Wird der Funktionsumfang der NXT-Software / EV3-Software optimal ausgeschöpft?</i>	Im Programm werden keine Schleifen oder Schalter eingesetzt.	Im Programm werden einfache Schleifen und Schalter meist sinnvoll eingesetzt. Variablen oder Datenleitungen sind nicht sinnvoll verwendet worden oder fehlen ganz. Konstruierter Ro-Bot mit nur einfachen Teilen.	Im Programm werden Schleifen und Schalter sinnvoll kombiniert eingesetzt. Über Datenleitungen werden vereinzelt Sensorwerte weiterverarbeitet.	Im Programm werden komplexe und elegante Kombinationen aus Schleifen und Schaltern eingesetzt. Mit Datenleitungen und Variablen werden Sensorwerte weiterverarbeitet. Eigene Blöcke werden als Unterfunktionen an geeigneter Stelle verwendet.
<b>Funktionalität</b> <i>Sind Software und Hardware gut auf einander abgestimmt?</i>	Hardware und Software arbeiten nicht gut zusammen. Das Programm steuert die meisten wesentlichen Komponenten nicht an.	Das Programm steuert die meiste Hardware an, aber einige Komponenten funktionieren nicht zuverlässig oder werden vom Programm gar nicht verwendet.	Das Programm verwendet die Komponenten des Roboters wie vorgesehen. Alle Komponenten funktionieren zuverlässig und erfüllen ihre Aufgaben.	Das Programm steuert alle Komponenten präzise an. Die Software ist flexibel genug gestaltet, um leicht weitere Teile hinzuzufügen und bei Bedarf verwenden zu können.
<b>Effizienz und Innovation</b> <i>Kann der Roboter durch innovatives Zusammenspiel von mechanischem Design und Softwaredesign die Aufgabe gut lösen? Wird der Roboter dadurch vielseitig einsetzbar?</i>	Die Software kann Schlüsselemente der gestellten Aufgabe nur teilweise oder gar nicht ausführen. Das Roboterdesign ist für die gestellte Aufgabe nicht geeignet.	Die Software erfüllt die grundlegenden Anforderungen und reagiert auf einzelne, einfache Änderungen. Das Roboterdesign ist wenig innovativ, eignet sich aber, genau die gestellte Aufgabe im Wesentlichen lösen.	Die Software erfüllt alle wesentlichen Anforderungen und reagiert auf einfache Änderungen. Das Roboterdesign zeigt interessante Ansätze und eignet sich gut, die gestellte Aufgabe zu lösen.	Der Roboter reagiert auf unvorhersehbare Änderungen und die Software funktioniert fehlerlos. Das Roboterdesign überrascht mit innovativen Ansätzen und kann die gestellte Aufgabe hervorragend lösen. Der Roboter kann auch für neue Aufgaben verwendet werden.

```

+ Programmstart  zeige Sensordaten
Wiederhole bis Taste Mitte gedrückt?
mache
  Zeige Text in Spalte 1
  in Zeile 1
  Warte 100 ms

```

```

+ Programmstart  zeige Sensordaten
Wiederhole bis Taste Mitte gedrückt?
mache
  wenn < 15
    gib Licht Farbsensor Port 3
  mache
  Motor Port B an Tempo 10
  Stoppe Motor Port C bremsen
  sonst
  Motor Port C an Tempo 10
  Stoppe Motor Port B bremsen

```

```

+ Programmstart  zeige Sensordaten
QuadratFahren

```

```

+ QuadratFahren
wiederhole 4 mal
mache
  Fahre vorwärts Tempo 30
  Strecke cm 20
  Drehe rechts Tempo 10
  Grad 90

```

```

+ Programmstart  zeige Sensordaten
Fahre vorwärts Tempo 30
Strecke cm 20
Drehe rechts Tempo 10
Grad 90
Fahre vorwärts Tempo 30
Strecke cm 20
Drehe rechts Tempo 10
Grad 90
Fahre vorwärts Tempo 30
Strecke cm 20
Drehe rechts Tempo 10
Grad 90
Fahre vorwärts Tempo 30
Strecke cm 20
Drehe rechts Tempo 10
Grad 90

```

```

+ Programmstart  zeige Sensordaten
Motor Port B an Tempo 60
für Umdrehungen 1
Motor Port C an Tempo 30
für Umdrehungen 1

```

```

+ Programmstart  zeige Sensordaten
Wiederhole bis Taste Mitte gedrückt?
mache
  Zeige Bild Augen auf
  x 0
  y 0

```