Another Small and Unique Robot from Oberpfaffenhofen

Einleitung

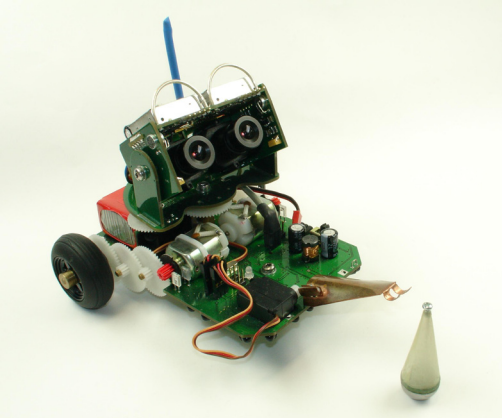


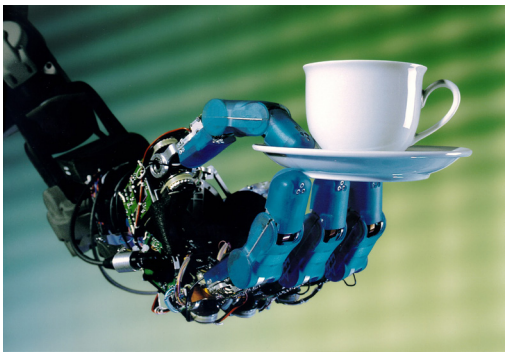
Abb. 1: Der Asuro – ein Roboter für Schüler! Dieses Exemplar haben wir zusätzlich mit „Augen“ ausgestattet … Bild: DLR

Schülerinnen und Schüler der Mittel- und Oberstufe hauchen begeistert dem kleinen Roboter ASURO Leben ein!

Am Anfang steht dabei ein Bausatz mit über 130 Teilen. Am Ende folgt der Roboter „ASURO“ – wenn alles richtig zusammengelötet wurde – selbstständig einer Linie und kann noch vieles mehr. Dazwischen befinden sich ein paar Stunden höchster Konzentration beim Löten, viel Kreativität beim Programmieren und einiges handwerkliches Geschick für die mechanischen Arbeiten.

ASURO wurde am DLR-Institut für Robotik und Mechatronik für das DLR_School_Lab Oberpfaffenhofen entwickelt. Er wird selbst zusammengebaut und ist frei in C programmierbar. Neben Leuchtdioden als Anzeigeinstrumente hat ASURO sechs Taster, um seine Umgebung zu erforschen. Die beiden Motoren lassen sich einzeln stufenlos ansteuern und über Reflexlichtschranken kann man ihre Drehzahl auswerten. Vervollständigt wird die Sensorik durch zwei Fotodioden auf der Unterseite, mit denen ASURO Helligkeitsunterschiede des Untergrundes erfassen kann. Darüber hinaus lässt sich ASURO mit verschiedenen Zusätzen erweitern. Neben Ultraschallortung (wie bei einer Fledermaus) und Wärmesensor gibt es ein LC-Display und einen Funkaufsatz. ASURO wird unter DLR-Lizenz bei verschiedenen Elektronik-anbietern vertrieben. Der Bausatz erfreut sich insbesondere bei Schulen großer Beliebtheit. Neben handwerklichen Fähigkeiten, elektrotechnischem Wissen und Programmiergrundkenntnissen vermittelt dieses Experiment nämlich vor allem eines: Spaß und Freude an moderner Technik und Wissenschaft.

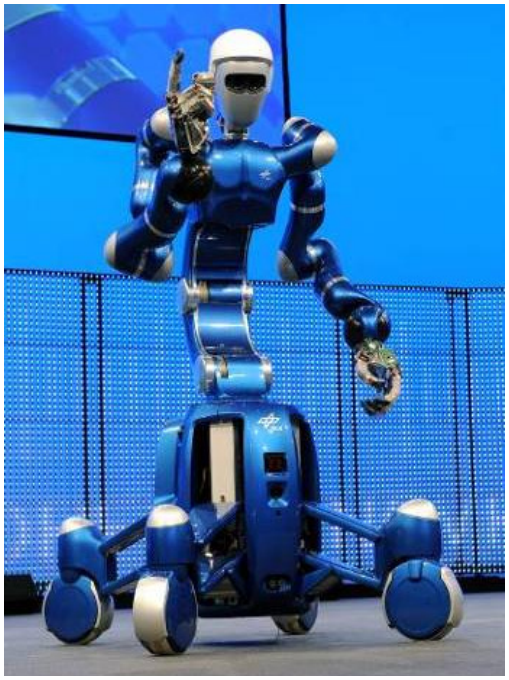
Bezug zur Forschung



Wird der Roboter bald den Menschen ablösen?

Roboter werden heute zu den unterschiedlichsten Zwecken eingesetzt. Die Bandbreite reicht vom Industrieroboter über kleine Rover-Fahrzeuge auf dem Mars bis hin zu medizinischen Instrumenten, mit denen man Menschenleben retten kann.

In der Medizin ist der Einsatz von Robotern zum unverzichtbaren Hilfsmittel geworden. Innovative Forschung hilft Chirurginnen und Chirurgen bei der Arbeit und verkürzt die Heilungszeiten – zum Beispiel mithilfe der minimal-invasiven Chirurgie, bei der robotische Systeme ebenfalls zum Einsatz kommen.



Das DLR beschäftigt sich mit Robotik in vielen Bereichen. Neben den in Wirtschaft und Industrie nutzbaren Anwendungen steht dabei natürlich die Weltraumforschung im Mittelpunkt. Dabei arbeiten Wissenschaftlerinnen und Wissenschaftler unterschiedlicher Disziplinen wie Elektrotechnik, Maschinenbau und Physik, aber auch der Medizin zusammen, um neue Maßstäbe in der modernen Robotik zu setzten.

Eine wegweisende DLR-Entwicklung ist JUSTIN: ein Roboter, dessen Oberkörper „humanoid“ – also dem Menschen nachgebildet – ist. Nur statt Beinen hat er ein Fahrwerk. Er kann seine Umgebung wahrnehmen, indem Kameras das Sehen übernehmen, Mikrofone für das Gehör sorgen und Kraft-Momenten-Sensoren das Fühlen und Tasten übernehmen. So kann JUSTIN selbstständig komplizierte Aufgaben ausführen – nicht nur um Astronautinnen und Astronauten später einmal zu unterstützen. Zu Demo-Zwecken bereitet er auch Eis-Tee zu oder fängt mehrere Bälle gleichzeitig.

Das Experiment

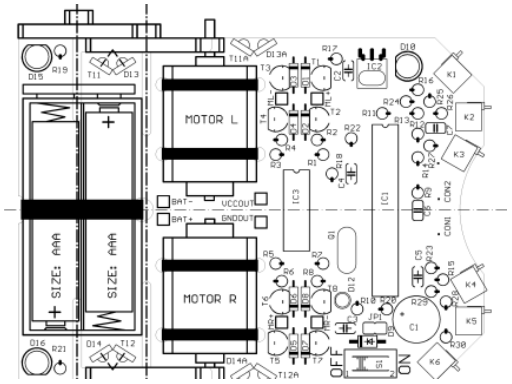
Euer Auftrag lautet, den Roboter ASURO zusammenzubauen und ihn dann zu programmieren und auf Testfahrten zu erproben – als ob ihr einen Rover auf eine Mission zum Mars vorbereitet.

Materialien und Hilfsmittel

- [ ] **1 Bausatz ASURO für zwei Schüler (ca. 40 €)**
- [ ] **Teppichmesser oder Säge**
- [ ] **feine Zange**
- [ ] **Abisolierzange**
- [ ] **1 Lötkolben pro ASURO**
- [ ] **Lötzinn: 1mm dickes Elektroniklot, bleifrei**
- [ ] **Seitenschneider**
- [ ] **Entlötlitze**
- [ ] **Schleifpapier mit feiner Körnung**
- [ ] **Kleber (Sekunden, Zweikomponenten oder Heißkleber)**
- [ ] **1 Computer pro ASURO**
- [ ] **Schwarzes Klebeband**

Vorbereitung, Aufbau und Durchführung

Step 1:
ASURO bauen



Bildet Zweiergruppen und bastelt nach der Bauanleitung je Gruppe den Roboter ASURO. Nachdem ihr die Lötfibel (siehe mitgeschickte Bauanleitung) gründlich studiert habt, startet mit der Roboterplatine!

Tipp: Es erspart viel Zeit, wenn je einer von euch die Bauteile vorbereitet und der andere sie einlötet!

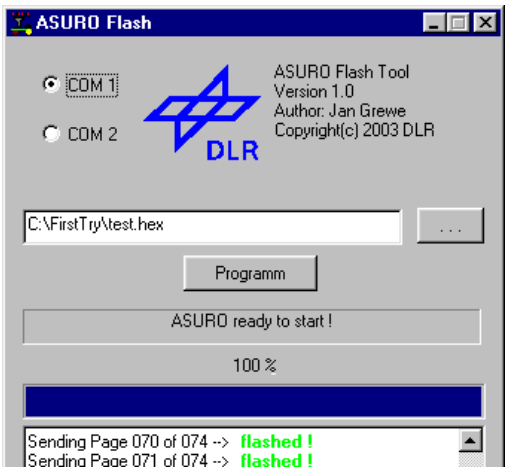
Achtung!

Folgt genau den Anweisungen der Bauanleitung! Dazu aber noch folgende Hinweise:

Achtet vor allem darauf, dass

- die Achsen zuerst und gerade eingelötet werden.
- das Getriebe getestet wird, indem ihr einmal die Räder ohne Motoren anbaut. Jetzt sollte alles ganz reibungslos laufen.
- nur die Sockel – ohne Prozessor – richtig eingelötet werden.
- Dioden und Transistoren nur in einer bestimmten Richtung funktionieren.
- die Widerstände im vorderen Teil korrekt eingelötet werden. Daher: Nehmt den Widerstand und biegt ihn an einer Seite ungefähr 3 mm hinter dem Bauteilkörper um 180°. Jetzt habt ihr automatisch ein langes und ein kurzes „Bein“. Das lange kommt in die Bohrung mit dem Kreis, das kurze in das Loch, zu dem der Punkt an dem Kreis zeigt.

Step 2:
Laufzeitmessung mit Stoppuhr



Installiert die Software von der ASURO-CD wie in der Anleitung beschrieben. Der ASURO wird in der Programmiersprache C programmiert und verfügt zusätzlich über einige vorbereitete ASURO-Funktionen. Nachdem ihr alles installiert habt und der Anleitung bis Seite 45 gefolgt seid, öffnet ihr die Datei; **test.c** im Programmers Notepad! Es erscheint eine Maske, die ASURO-Funktionen übersetzt.

Für die ersten Versuche reicht es aus, wenn ihr diese Datei etwas abändert!

Zum Beispiel wenn die StatusLED rot oder grün leuchten soll:

```
#include „asuro.h“
Int main(void)

{
        Init():
                StatusLED(RED);

        while(1);
return0;

}
```

Sobald ihr euer erstes Programm fertig geschrieben habt, klickt ihr auf Tools und dann auf make! Hier wird euer Programm für den ASURO übersetzt in eine .hex Datei. Und wenn das Programm keine Fehler enthält, schreibt es: **Errors none**.

Der erste Kontakt!

Öffnet das ASURO Flash Tool und ihr könnt das Programm per Infrarot-Transceiver downloaden. Testet als erstes euren ASURO! Der Ablauf des SelfTest ist im Handbuch beschrieben.

Step 3:
ASURO lernt blinken

Die Front-LED (D11) kann ein- bzw. ausgeschaltet werden.
Mögliche Übergabeparameter sind: ON, OFF

Beispiel:
Die Front-LED soll hell leuchten. Der Funktionsaufruf sieht dann folgendermaßen aus:

```
FrontLED(ON);
```

Versucht ebenso die Rück-LED mit folgender Funktion zum Leuchten zu bringen:

```
BackLED(unsigned char left, unsigned char right)
```

Step 4:
ASURO lernt fahren

Ihr könnt den beiden Motoren eine Drehrichtung und jeweils eine eigene Geschwindigkeit geben. Dabei ist es wichtig, die Drehrichtung der beiden Motoren vor der Geschwindigkeits-einstellung aufrufen.

Die Parameter für die Drehrichtung sind:
FWD (Vorwärtsrichtung)
RWD (Rückwärtsrichtung)
BREAK (Bremsen)

Beispiel:
Der linke Motor soll sich vorwärts drehen, während der rechte Motor stillsteht.

```
MotorDir (FWD,BREAK);
```

Die Geschwindigkeit kann in einem Bereich von 0 bis 255 angegeben werden.
Beispiel:
Der rechte Motor soll sich mit maximaler Geschwindigkeit drehen, der linke Motor gar nicht.

```
MotorSpeed(0,255);
```

Eure Aufgabe ist es nun, ASURO geradeaus fahren zu lassen!
Hört sich leicht an, ist es aber gar nicht!

Step 5:
ASURO kann auch schlafen

Überlegt euch wie ASURO mit seinen Lichtern blinken kann!
Es klappt nicht?

- **Hilfe 1:**
  Vielleicht denkt der Prozessor schneller als eure Augen es wahrnehmen können!
  Wie könnte man das Problem lösen?

- **Hilfe 2:**
  Ihr müsst eine Verzögerung programmieren mit der sleep()-Funktion.
  Diese Funktion legt den Prozessor für eine einstellbare Zeit schlafen. Sie besitzt einen 72kHz Timer und der Parameter kann Werte von 0 bis 255 (unsigned char) annehmen.

- **Hilfe 3:**
  Der Prozessor muss kurz schlafen, dabei entsprechen 3ms Schlaf 216 Arbeitsschritten!
  Beispiel: Der Prozessor soll für ca. 6ms warten, der Funktionsaufruf sieht
  dann folgendermaßen aus:

  Sleep(432);

Step 6:
ASURO folgt der Linie

Klebt eine Linie mit schwarzem Klebeband auf weißem Grund! Der ASURO hat 2 Fototransistoren auf der Unterseite. Mit diesen kann er Helligkeiten wahrnehmen und sie euch als eine Zahl zwischen 0 und 1.023 angeben, 0 heißt sehr dunkel, 1.023 sehr hell. Um diese aber bearbeiten zu können, braucht der ASURO eine spezielle Art von Speicher von euch, ein Array, das ihr euch als eine Art Tabelle vorstellen könnt. In jeder Zelle steht ein anderer Wert. Schreibt also in euer Programm noch vor dem Init()-Befehl:

```
unsigned int data[2];
```

Jetzt haben wir ein Array mit Namen data, welches 2 Felder besitzt. Verwendet jetzt in eurem Programm in die while-Schleife den Befehl:

```
LineData(data);
```

Jetzt hat der ASURO die Helligkeiten gemessen und in das Array geschrieben.

Der Wert vom linken Fototransistor steht in data[0], dem ersten Feld der Tabelle, der Wert vom rechten in data[1]. Diese Felder könnt ihr jetzt wie eine Variable verwenden, z.B. so:

```
if (data[0]>data[1])
        {MotorDir(BREAK,BREAK);}
```

Hier steht, dass der ASURO bremst, wenn er links einen helleren Wert gemessen hat als rechts.

Versucht nun mit eurem bisherigen Wissen, den ASURO auf der Linie fahren zu lassen!

- **Hilfe 1:**
  Überlegt euch, welche Werte gemessen werden, wenn die Linie genau zwischen den Fototransistoren durchgeht.

- **Hilfe 2:**
  Überlegt euch, was der ASURO sieht, wenn die Linie eine Rechtskurve macht?
  In welche Richtung sollte der Roboter dann fahren?

- **Hilfe 3:**
  Wenn die Linie nun aber eine Rechtskurve macht, dann rollt der ASURO mit dem rechten Fototransistor auf die Linie und sieht so dort etwas Dunkleres. Also muss, wenn es rechts dunkler ist als links, der ASURO nach rechts fahren. Das könnte dann so funktionieren:

  ```
  if (data[1]<data[0])
      {MotorSpeed(230,170);}
  ```

  Dasselbe müsst ihr nun noch für die linke Seite schreiben. Die Werte für die Geschwindigkeit sind nur grobe Vorgaben. Diese müsst ihr noch optimieren und beobachten, wie gut euer ASURO der Linie folgt.

Step 7:
ASURO lernt tanzen

Rechteck fahren

Kombiniert euer Wissen aus Aufgabe 4 und 5, um den ASURO Figuren fahren zu lassen, zum Beispiel ein Viereck oder eine Acht (lange Linkskurve, dann lange Rechtskurve).
Überlegt euch, was der ASURO sieht, wenn die Linie eine Rechtskurve macht?

Step 8:
Parcours

Der ASURO hat 6 Taster an seiner Vorderseite. Werden diese gedrückt, zum Beispiel weil er gegen eine Wand gefahren ist, dann könnt ihr euch mithilfe eines Codes genau sagen, welcher der Taster gedrückt wurde.

Mit dem Befehl **PollSwitch()** könnt ihr diesen Code abfragen.

Zum Beispiel so:

```
unsigned char tasterwert;

tasterwert = PollSwitch();
```

In der Variable Tasterwert steht nun der Code, mit dem ihr entschlüsseln könnt, welcher Taster gedrückt ist.

Aber woher kommt der Code?
Der ASURO vergibt jedem Taster eine Zahl. Von hinten betrachtet von rechts nach links haben die Taster die Zahlen 1,2,4,8,16 und 32. Wird einer der Taster gedrückt, so bekommt ihr diese Zahl. Werden mehrere gleichzeitig gedrückt, dann werden diese Zahlen addiert.
Zum Beispiel heißt ein Wert von 7, dass die 3 rechten Taster gedrückt wurden.

Versucht den ASURO bremsen zu lassen, wenn er gegen eine Wand gefahren ist!

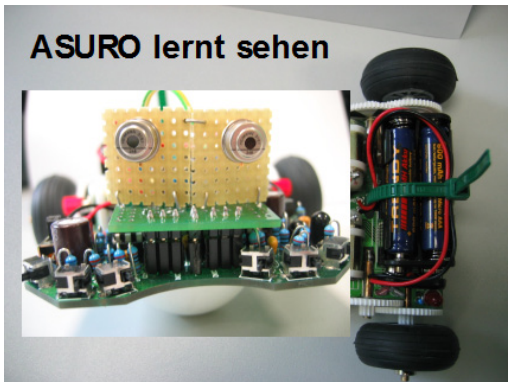
- **Hilfe 1:**
  Probiert doch einmal folgen Code in deinem Programm aus:

  ```
  if (PollSwitch()>0 && PollSwitch()>0 && PollSwitch()>0)
      {MotorDir(BREAK,BREAK);}
  ```

  Was bedeutet diese If-Abfrage?
  PollSwitch()>0 vergleicht direkt den Code von den Tastern, ob dieser größer 0 ist. Das ist immer dann der Fall, wenn irgendeiner der Taster gedrückt ist. Mit dem Befehl werden einzelne Bedingungen verbunden, so dass in diesem Fall alle Bedingungen stimmen müssen. Dies hilft, Messfehler zu umgehen, so dass der ASURO sicher sein kann, dass wirklich ein Taster gedrückt wurde.

Jetzt könnt ihr euch einen Parcours überlegen, den euer ASURO dann überwinden muss!

Aktuelles und Ausblick



ASURO befindet sich in ständiger Weiterentwicklung. In dem Buch „Mehr Spaß mit ASURO“ wird der Roboter mit seiner technischen Funktionsweise in verständlicher Form beschrieben. Zudem enthält das Buch viele Tipps und Informationen zum Erstellen von Zusatzplatinen. Im DLR_School_Lab Oberpfaffenhofen gibt es dazu übrigens ein spannendes Experiment:

Ähnlich wie der Mars-Rover Curiosity erkundet der ASUROnaut eine den Schülerinnen und Schülern unbekannte, nicht einsehbare Landschaft. Der ASUROnaut ist ein auf ASURO-Basis entwickelter Erkundungsroboter mit Stereokamera und erweiterter Sensorik.

Weiterführende Links

Roboternetz.de - Das große Portal für Roboter, Elektronik und Microcontroller Bastler http://www.roboternetz.de
Wikipedia Artikel: ASURO
http://de.wikipedia.org/wiki/ASURO
DLR-Programme für Schülerinnen und Schüler
http://www.dlr.de/desktopdefault.aspx/tabid-4741/7888_read-12417/
DLR_next: High-Tech für den Alltag: Roboter & Co.
http://www.dlr.de/next/desktopdefault.aspx/tabid-6306/11086_read-25290/

HINWEIS

Die hier beschriebenen Mitmach-Experimente wurden sorgfältig ausgearbeitet. Sie können jedoch auch bei ordnungsgemäßer Durchführung und Handhabung mit Gefahren verbunden sein. Die hier vorgeschlagenen Mitmach-Experimente sind ausschließlich für den Einsatz im Schulunterricht vorgesehen. Ihre Durchführung sollte in jedem Fall durch eine Lehrkraft betreut werden. Die Richtlinien zur Sicherheit im Schulunterricht sind dabei einzuhalten.

Das DLR kann keine Garantie für die Richtigkeit, Vollständigkeit und Durchführbarkeit der hier beschriebenen Experimente geben. Das DLR übernimmt keine Haftung für Schäden, die bei Durchführung der hier vorgeschlagenen Mitmach-Experimente entstehen.

Informationen für Lehrkräfte

Fächer

Informatik, Physik, Mathematik

Alter/Schwierigkeitsgrad

Ab ca. 9. Klasse

Dauer des Experiments

Das Löten des ASURO dauert in der Regel fünf Schulstunden. Sollen die Schülerinnen und Schüler nur eine einfache Linienfolge programmieren, benötigen sie ca. weitere vier Schulstunden. Bei der Programmierung sind jedoch keine Grenzen gesetzt: Ein Roboter, der einen Parcours bewältigt, kann als Projekt über ein ganzes Schuljahr hinweg entwickelt werden.

Lernziele

Löten
Kennenlernen elektronischer Bauteile
Programmierung in C

Kontakt

Britta Kästner, DLR_School_Lab Oberpfaffenhofen