

c't 2/2006

Spielgefährten

Questions / Read:

- Wie ist der Bot aufgebaut?
Runde Grundplatte, damit sich der Bot auf der Stelle drehen kann.
- Was für Sensoren hat der Bot?
Zwei Reflexlichtschranken messen die Drehgeschwindigkeit der Räder.
Zwei Infrarotsensoren messen den Abstand nach vorne.
Zwei Lichtschranken die den Boden beobachten um einer schwarzen Linie zu folgen.
Zwei Lichtschranken die im vorderen Teil des Bots sind um einen Abgrund zu erkennen.
Zwei Photosensoren, damit der Bot Lichtquellen finden kann.
- Wie wird der Bot angetrieben?
Durch 2 Gleichstrommotoren (6 Volt) mit einem Untersetzungsgetriebe (33:1) erreicht der Bot 0,03 Nm. Bei maximal 151 U/min und einem Raddurchmesser von 57 mm erreicht der Bot 0,45 m/s (1,6 km/h).
- Wie wird der Bot mit Energie versorgt?
Fünf NiMh-Akkus (Mignon-Zellen) liefern 6 Volt Gleichspannung.
Stabile 5 Volt für die Elektronik erzeugt ein Low-Drop-Regler.
- Wie wird der Bot gesteuert?
Mikrocontroller (Atmel Atmega32)
- Kann der Bot kommunizieren? Wenn ja, wie?
Soll später per Funk mit einem PC oder mit anderen Bots kommunizieren können.
Bekommt Befehle per IR-Fernbedienung.
- Was kostet der Bot?
200 Euro

Reflect:

- In groben Zügen wird der Aufbau des Bots beschreiben.
- Programmierung ist kein Thema.
- Bauteile werden angerissen.
- Motorsteuerung und Energieversorgung auf elektrotechnischer Ebene dargestellt.
- Blick in die Zukunft was noch kommen soll.

Recite:

Es wird geschildert für wen der c't-Bot ist und das kein große Löterfahrung vorausgesetzt wird. Bot kann erstmal auch nur simuliert werden, dazu steht der c't-Sim zu Verfügung.

Aufbau des Bots wird beschrieben, warum die Grundplatte rund ist und warum zwei Räder verwendet werden.

Es werden die vorhandenen Sensoren beschrieben. Welcher Art sie sind und was ihre Aufgabe ist. Die Motoren werden beschrieben. Das sie genügend Kraft haben, leicht zu steuern sind und auch noch bei fast leeren Akkus funktionieren.

Energieversorgung wird beschrieben

Der Typ des Controllers wird erwähnt.

Review:

- Allgemeine Informationen über den Bot.
- Fast nichts zur Programmierung.
- Hardware steht im Vordergrund.
- Nichts zum Sim

Question / Read:

- Was ist der c't-Sim?
Soll eine möglichst genaue Nachbildung des c't-Bots und seiner Umwelt sein.
Besteht aus der Simulator-Umgebung und mindestens einem simulierten Bot. Java-Projekt c't-Sim steckt den Rahmen der Welt ab. C-Code des Bots ist gekapselt, so wie er auch auf dem Bot selbst ablaufen könnte. Die Kommunikation findet über TCP/IP statt.
- Wie funktioniert der Sim?
Sim wird gestartet wie jedes Java-Programm. Dann startet man den Bot, der sich per TCP/IP am Simulator anmeldet.
Zusätzlich zur simulierten Welt sieht man eine Kontrolltafel, die den Zustand des Bots mit den Daten die seine Sensoren empfangen anzeigen.
Mit Java3D lassen sich Welten erstellen und diffuses Licht simulieren.
Für jeden Bot wird ein eigener Thread angelegt.
Es lassen sich auch Bots direkt für den Simulator programmieren. Dabei kann man auf den C-Code verzichten.
- Wie funktioniert Zugriff auf die internen Zustände des Bots?
- Wie sieht das Datenmodell des Sim aus?
Sim ist dreigeteilt: Darstellung, Modellierung und Kontrollelemente.
Durch die konzeptionelle Trennung können weitere Views erzeugt werden, die z.B. die Sicht des Roboters darstellen.
- Dokumentation des Sim?
- Was für Werkzeuge kommen zum Einsatz?
Eclipse als Entwicklungswerkzeug.
- Synchronisationsprobleme?
Es gibt eine Zeit für die Welt und jeder Bot hat seine eigene Zeit. Nach jedem Durchlauf werden die Zeiten synchronisiert.

Reflect:

- Einstieg in die Programmierung des Simulators.
- Einblick in Funktionsweise des Sim

Recite:

Im Sim lässt sich der original Steuercode des Bot testen. Mit Java3D lassen sich leicht Welten modellieren. Der Sim teilt sich auf in die Modellierung der Sicht (Fenster), der Welt und der Bots. Dabei werden zwei Bot-Typen unterschieden: einmal der mit dem original C-Code und der in Java programmierte Bot. Es lassen sich leicht weitere Robotermodelle simulieren, dank der Vererbungshierarchie. Die Welt und jeder Bot laufen in einem eigenen Thread, nach 10ms werden die Zeiten der einzelnen Threads synchronisiert.

Des weiteren wird beschrieben wie die neue Position berechnet wird. Hierfür stellt Java Vektor-Klassen bereit. Es kommt hierbei zu einem Fehler aufgrund der Threads und aufgrund dessen, dass die Motoren linear simuliert werden.

Der Schnelleinstieg wird im Kasten „Werkzeuge“ erklärt.

Review:

- Einstieg in Programmierung des Sim

c't 4/2006
Hallo Welt!

Question / Read:

- Wie wird der Flash-Speicher befüllt?
Die Firmware des Controllers spielt man z.B. mit dem Programm PonyProg2000 auf. Ohne Programmierung dreht sich der Bot nur im Kreis.
- Erste Programmierung zum Funktionstest, Fehlersuche?
Es gibt fertige Testprogramme die Testen ob die Hardware korrekt funktioniert. Der Status wird über 8 LEDs angezeigt. Mit einem Display kann man alle Funktionen auf einmal testen, da die Programme immer gleich testen und auf dem Display alle Informationen anzeigen können.
- Programmierung des MC?
Beim kompilieren des C-Codes gibt man an, ob das Programm auf Sensoren zugreifen soll, oder auf den Sim

Recite:

Das Zusammenbauen des Bots wird beschrieben. Wie die Firmware auf den Controller überspielt wird und wie ein erster Test gemacht wird, ob die Hardware funktioniert. Ein kleines Beispiel zeigt wie dem Bot ein Verhalten einprogrammiert wird.