

Echtzeit-Roboter-Steuerung aus Hollabrunn

Ein Bericht über eine Diplomarbeit der Abteilung Steuerungs- und Regelungstechnik in Zusammenarbeit mit Opentech.at

Florian Bruckner, Georg Schiesser (betreut von Richard Smetana)

Projektziele

Ziel unseres Projektes war es, die Steuerung eines Industrie-Roboters über weite Entfernungen und ohne direkten Sichtkontakt zu ermöglichen. Da die Steuerung des Roboters in Echtzeit geschehen muss, verwendeten wir für unser Projekt das von FSMLabs entwickelte Echtzeit-Betriebssystem RTLinux. Das gesamte Projekt sowie alle Source-Codes und Dokumentationen sind frei im Internet erhältlich und wurden unter GPL veröffentlicht.

Zur Steuerung des Roboterarms sollte ein Standard-PC-Joystick verwendet werden, der über den Gameport mit dem Computer verbunden wird. Die vom Joystick eingelesenen Daten werden dann in Geschwindigkeiten der einzelnen Roboterachsen umgerechnet und über eine Netzwerkverbindung (Ethernet- bzw. COM-Schnittstelle) an den zweiten Computer geschickt, der dann den Roboter ansteuert.

Um die Steuerung des Roboters auch ohne direkten Sichtkontakt zu ermöglichen, wurden 2 USB-Webcams in das System integriert, die ein Bild des Roboters am Bildschirm anzeigen. Zusätzlich wird die aktuelle Position des Roboters durch ein OpenGL-Modell simuliert und dreidimensional dargestellt.

Projektübersicht

Unser gesamtes Projekt lässt sich in vier Teilbereiche einteilen. Zuerst das **Joystick-Modul**, das die Position des Joysticks über den Gameport einliest und diese in Geschwindigkeiten der einzelnen Roboter-Achsen umrechnet. Anschließend werden diese Daten über eine Netzwerkverbindung (Ethernet, COM) an den zweiten Teil, das **Roboter-Modul**, weitergereicht. Dieses Modul muss dann nur mehr die geeigneten Befehle an den Roboter schicken, um ihn an die berechnete Position zu bewegen. Die ersten beiden Module sind für die Steuerung des Roboters zuständig und müssen deshalb echtzeitfähig sein. Die nächsten beiden Module sind für die Visualisierung der Roboter-Position verantwortlich. Sie sind nicht echtzeitfähig, da sonst die Realisierung sehr aufwendig gewesen wäre und es eigentlich nicht notwendig ist, dass das Bild des Roboters in regelmäßigen Abständen angezeigt wird. Das dritte Modul übernimmt die **Visualisierung** des Roboters mittels eines OpenGL Modells. Dafür wird ihm die aktuelle Position des Roboters vom Roboter-Modul übergeben. Zusätzlich wird ein Bild des Roboters vom vierten Modul angezeigt. Es wird von den Webcams über USB eingelesen und über eine Netzwerkverbindung an die Visualisierung geschickt.

Das Joystick-Modul

Das Joystick-Modul basiert auf dem Linux-Joystick-Treiber für Logitech-Joysticks. Sobald das Modul eingefügt wird, wird zuerst die Soundkarte initialisiert und der Gameport aktiviert. Anschließend wird der Joystick initialisiert. Der verwendete Logitech-Joystick unterstützt 2 verschiedene Betriebsarten:

- **Analogmodus:** Es wird ein normaler Analog-Joystick emuliert. Es werden 2 Achsen und 2 Buttons unterstützt, die jeweils über eine eigene Leitung mit dem Gameport verbunden sind.
- **Digitalmodus:** Die Position des Joysticks wird digital über die zwei Button-Leitungen übertragen. Vorteile dieses Modus sind die höhere Genauigkeit und beliebige Anzahl von Achsen bzw. Buttons.

Der Joystick wird durch das Senden einer Trigger-Sequenz im digital Modus initialisiert. Danach wird ein ID-Paket übertragen, das bestimmte Informationen über den Joystick enthält (Achsen, Buttons, ...). Anschließend wird noch ein Device-File angelegt,



Projektteam

über welches die Joystick-Position eingelesen werden kann. Nach Aufruf der `read`-Funktion dieses Device-Files wird ein Triggersignal an den Joystick gesendet, welcher daraufhin ein Datenpaket sendet. Dieses Datenpaket beinhaltet die Position der einzelnen Achsen und die Zustände der Buttons.

Der zweite Teil des Joystick-Moduls braucht nur mehr regelmäßig die Daten vom Joystick-Device-File einlesen und über die serielle Schnittstelle an das Roboter-Modul senden. Ein RT-Treiber für die serielle Schnittstelle ist bereits vorhanden und heißt `rt_com`.

Das Robotermodul

Das Roboter-Modul liest regelmäßig Datenpakete von der seriellen Schnittstelle und berechnet anhand der übergebenen Geschwindigkeiten die Positionen der einzelnen Roboterachsen.

Diese werden dann in einem speziellen Format über eine zweite serielle Schnittstelle an den Roboter übergeben. Zusätzlich wird vom Roboter-Modul verhindert, dass sich die berechnete Position innerhalb von bestimmten, nicht erlaubten Bereichen befindet, in denen sich der Roboter selbst beschädigen würde. Ein weiterer Sicherheitsmechanismus kontrolliert, ob die Datenpakete vom Joystick-Modul in regelmäßigen Abständen gesendet werden. Falls ein Datenpaket ausfällt, wird das Roboter-Modul gestoppt, um zu verhindern, dass im Falle einer unterbrochenen Verbindung der Roboter nicht mehr gestoppt werden kann und sich somit mit der alten Geschwindigkeit weiterbewegt.

Die Visualisierung

Die Visualisierung liest regelmäßig die aktuelle Roboter-Position vom Roboter-Modul aus, und zeichnet anhand dieser Daten ein 3D-Modell des Roboters. Das Modell wurde auf die grundlegenden Merkmale des Roboters reduziert, um nicht zu viel Rechenzeit für die Visualisierung zu verbrauchen.

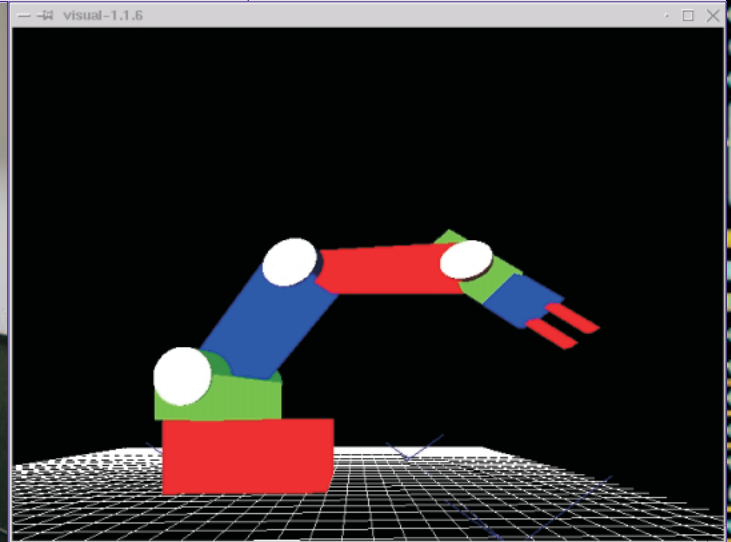
Zusätzlich hat der Benutzer die Möglichkeit, die gewünschte Ansicht einzustellen. Mit Hilfe der Maus ist es möglich an den Roboter heranzuzoomen oder vom Roboter wegzuzoomen. Außerdem kann man die Ansicht rotieren und verschieben.

Für die Realisierung des 3D-Modells wurde OpenGL verwendet, da es bereits in den meisten Linux-Distributionen standardmäßig enthalten ist. Außerdem bietet es die Möglichkeit, relativ einfach



Aufbau

Lichteffekte hinzuzufügen, wodurch das Modell viel realistischer wird.



Visualisierung

den Force-Feedback-Joystick zurückgeführt werden. Dies ermöglicht es dem Anwender, die vom Roboter aufgewendete Kraft zu spüren und dadurch steuern zu können.

Das Webcammodul

Dieses Modul hat die Aufgabe, ein Bild des Roboters von einer Webcam einzulesen, über eine Netzwerkverbindung an den lokalen Rechner zu senden und am Bildschirm darzustellen. Als Kamera wurde eine USB-Webcam ausgewählt, die Bilder in einer Auflösung von 640x480 ermöglicht. Da für die Webcam bereits Linux-Treiber zur Verfügung standen, beschränkte sich unsere Aufgabe darauf, die Bilder vom entfernten auf den lokalen Rechner zu übertragen.

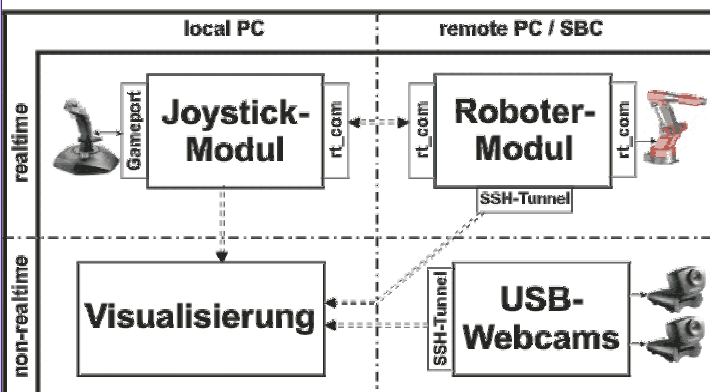
Aus Zeitgründen konnte dieses Problem jedoch noch nicht vollständig gelöst werden. Es wurde versucht, mit Hilfe eines SSH-Tunnels die Daten der Webcam von einem Rechner zum anderen zu senden. Der SSH-Tunnel hat den Vorteil, dass die Daten verschlüsselt übertragen werden, und dass keine eigene Anwendung für die Datenübertragung erstellt werden muss. Da der Webcam-Treiber nicht nur Daten von der Webcam ausliest sondern auch bestimmte Steuerbefehle sendet, reicht das Umleiten der Ausgabe nicht aus. Es muss eine Anwendung erstellt werden, die alle benötigten Datei-Operationen umleitet.

Zukunftsaussichten

Als zukünftige Erweiterung des Projektes ist die Datenübertragung über Ethernet geplant. Die dafür benötigten Netzwerkkarten-Treiber sowie das echtzeit-fähige Socket-Interface ist bereits im Versuchsstadium. Dadurch könnte in Zukunft sogar die Steuerung des Roboters über Internet möglich werden.

Eine weitere Erweiterung stellt die Verwendung eines Force-Feedback-Joysticks dar. Um ein Rückführung der aufgewendeten Kraft zu ermöglichen, muss diese zuerst gemessen werden. Dies könnte im einfachsten Fall durch die Messung des Motorstromes realisiert werden. Dieser Strom ist proportional der aufgewendeten Kraft. Nachdem die gemessene Kraft vom Roboter-Modul an das Joystick-Modul gesendet wurde, kann diese über

Blockdiagramm



RTLinux

RTLinux ist ein Echtzeit-Betriebssystem das von FSMLabs entwickelt wurde. Es bietet die Möglichkeit in einem „normalen“ Linux System auch Echtzeit-Tasks auszuführen. Dafür muss zuerst ein Standard Linux-Kernel heruntergeladen und eine geeigneter Patch angewendet werden. Nachdem mit dem neuen Kernel gebootet wurde, kann dann RTLinux gestartet werden. Daraufhin übernimmt RTLinux die vollständige Kontrolle über den Computer (z.B.: Interrupts,...). Das „normale“ Linux läuft dann als RTLinux Task weiter. Zusätzlich können aber auch Realtime-Tasks erzeugt werden, die dann in Echtzeit ausgeführt werden. Echtzeit bedeutet nicht, dass ein Task schneller ausgeführt wird, sondern nur, dass man mit Sicherheit sagen kann, dass er innerhalb einer bestimmten Zeitspanne ausgeführt wird. Dadurch kann gewährleistet werden, dass der RT-Task innerhalb einer gewissen, genau definierten Zeitspanne auf ein Ereignis reagieren kann.

Vorteile von RTLinux gegenüber anderen RT-Betriebssystemen

- Für die Initialisierung von Geräten können „normale“ non-RT-Tasks verwendet werden
- „normale“ Linux Anwendungen können parallel zu RT-Tasks ausgeführt werden (Netscape, Star Office,...)
- RTLinux ist frei erhältlich und der Source Code ist unter GPL veröffentlicht

Links

- RTLinux Homepage <http://www.opentech.at/>
- Projekt Homepage <http://www.opentech.at/~florianb/>
- Projekt Homepage (Mirror) http://www3.htl-h1.ac.at/homepages/IP_Joystick/
- IP Support für RTLinux <http://canals.disca.upv.es/~serpeal/RTL-1wIP/htmlFiles/>

http://www.opentech.at/~florianb/