

Aufgrund des noch kleinen Marktes für die mobile Robotik – im Vergleich zur industriellen Automatisierungstechnik – ist das Angebot an Hardware- und Softwareprodukten, die auf die Anforderungen der mobilen Robotik zugeschnitten sind, relativ klein. Gernot Gebhard und Matthias Bengel vom Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA geben einen Überblick, wie durch den Einsatz etablierter Standardkomponenten, wie PC-Control von Beckhoff und EtherCAT als Kommunikationssystem, hier Abhilfe geschaffen werden kann. Somit werden deren Potenziale in puncto Einsparungen, Performance, Robustheit usw. auch für die Servicerobotik verfügbar.

EtherCAT für mobile Roboter

Als Integrationsplattform wurde der vom Fraunhofer IPA entwickelte, mobile Sicherheitsroboter Secur-O-bot zur Überwachung von Gebäuden und Industrieanlagen, verwendet. Bei dem in Abb. 1 dargestellten Roboter handelt es sich um eine dreirädrige Plattform mit Differentialantrieb. Die Steuerungssoftware von Secur-O-bot basiert auf dem Navigationssystem des Fraunhofer IPA, das sich schon in den unterschiedlichsten Anwendungen und Umgebungen – teilweise im mehrjährigen Dauerbetrieb – bewährt hat. Damit sich der Roboter autonom fortbewegen kann, wird die Navigationssoftware mit hochpräzisen Daten des Laserscanners versorgt. Neben diesen Sensoren verfügt der Roboter über weitere Sensoren zur Überwachung.

Bisher erhältliche Überwachungsroboter überwachen jeweils nur den von der aktuellen Roboterposition aus „sichtbaren“ Bereich der Umgebung. Um jedoch auch ohne fest installierte Sensoren eine flächige Überwachung von Anlagen und Gebäuden zu erreichen, setzt der Roboter mit Hilfe eines Zusatzmoduls Sensoren ab. Durch das automatische Ausbringen und Wiederaufnehmen kleiner autarker Sensorsonden durch die mobile Robotereinheit kann die Überwachungsfähigkeit eines einzelnen Secur-O-bot durch jede zusätzlich ausgebrachte Sensorsonde um ein Vielfaches vergrößert werden, ohne die hohen Kosten einer flächendeckenden Festinstallation zu verursachen.

Schnittstellenvielfalt

Die bisherigen mobilen Roboter des Fraunhofer IPA wurden anwendungsgetrieben entwickelt, d.h. die anzuschließenden Geräte wurden rein nach Applikationsgesichtspunkten und nicht nach den bereits vorhandenen Schnittstellen ausgewählt. Da in den verschiedenen Robotern durchaus unterschiedliche Geräte, wie beispielsweise Antriebscontroller und Motoren verbaut sind, gibt es eine Hardwareabstraktionsschicht in der Robotersteuerung. Je nach verwendeter Schnittstelle und Komponente wird das entsprechende Modul selektiert und konfiguriert. Die Robotersteuerung enthält dazu verschiedene Treiberklassen, auf die mittels einer einheitlichen Schnittstelle zugegriffen wird.

Als Konsequenz kann die Schnittstellenvielfalt in Hardware große Ausmaße annehmen, was dazu führt, dass stets neue Treiberklassen implementiert werden



Abb. 1: Der Überwachungsroboter Secur-O-bot im Application-Park-Stand „PC-Control“ auf der Hannover Messe 2008.

müssen. Die Bibliothek an verfügbaren Treiberklassen wächst also stetig, was auch den Aufwand für die Wartung der Software nach oben schraubt.

Einheitliche Steuerungsarchitektur durch den Einsatz von Embedded-PC, TwinCAT und EtherCAT

Um die oben angeführten Defizite zu beseitigen, wurde prototypisch der eingangs beschriebene, mobile Roboter umgebaut, um für eine Vereinheitlichung der Schnittstellen in Software und Hardware zu sorgen.

Der Bordrechner des Roboters wurde durch einen Beckhoff Schaltschrank-PC C6920, auf Basis von Windows XP Embedded und TwinCAT I/O, ersetzt. Die Automatisierungssoftware TwinCAT ermöglicht den einheitlichen Zugriff auf die angeschlossene Hardware mit Hilfe einer Abstraktionsschicht. Die angeschlossenen I/O-Kanäle werden durch einen Echtzeittreiber miteinander in Beziehung gebracht. TwinCAT I/O ermöglicht den zyklischen Datenaustausch von Prozessvariablen auf Task-Ebene.

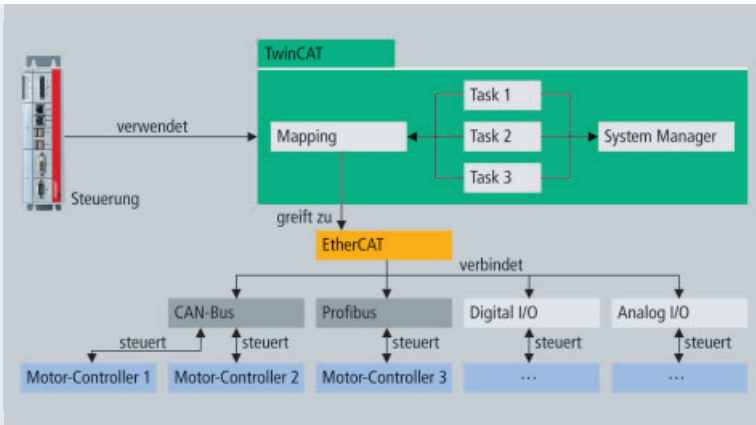


Abb. 2: Hardwareschicht mit der Verwendung von TwinCAT

Die Hardwareschicht der Rechnerarchitektur ist in Abb. 2 dargestellt. Von dieser Seite aus betrachtet, kommuniziert die Steuerung mit den entsprechenden EtherCAT-Klemmen über EtherCAT, d.h. es wird nur ein einziger Feldbus benötigt. Die I/O-Station besteht im Wesentlichen aus digitalen Ein-/Ausgabeports und aus Feldbusklemmen für andere Bussysteme, wie beispielsweise dem CAN-Bus. Auch andere Schnittstellen, wie PROFIBUS, serielle Schnittstellen oder analoge Ein-/Ausgänge sind möglich. Durch die Buskoppler werden die Daten von anderen Schnittstellenprotokollen in das EtherCAT-Protokoll gekapselt, so dass eine einheitliche Sicht auf die Daten entsteht. Die Konfiguration der Hardwareschnittstellen erfolgt mit dem TwinCAT System Manager. Die Robotersteuerung greift über das Mapping auf die konfigurierten Prozessdaten zu.

Dabei vereinfacht sich die Steuerungsanbindung auf die Verwendung eines automatisch generierten Headerfiles. Wird Hardware geändert, ohne dass sich die Typisierung der Prozessvariablen ändert, so geschieht dies für die Steuerung vollkommen transparent, da die Konfigurationsschritte innerhalb von TwinCAT ablaufen. Nur bei Änderungen des Headerfiles ist die Steuerung anzupassen. Dies geschieht im Allgemeinen jedoch nur, wenn zusätzliche Hardware verwendet werden soll, die zum Zeitpunkt der Steuerungsentwicklung noch nicht vorgesehen war. Somit ist eine weitere Implementierung von Hardwaretreibern in der Steuerung nicht mehr notwendig.

Das Erweiterungsmodul zum Absetzen von Sensorsonden wird „Huckepack“ von der mobilen Plattform mitgeführt. Vor dem Umbau der Hardware wurde das Erweiterungsmodul durch einen Mikrocontroller angesteuert. Nun werden die I/O-Klemmen, die über EtherCAT am IPC angeschlossen sind, direkt angesteuert, und die relative teure Anschaltung durch den Mikrocontroller entfällt. Damit ist EtherCAT bis hinunter in die Feldebene verfügbar. Durch das Auslagern der I/Os über den EtherCAT-Koppler kann der PCI-Steckplatz des IPCs anderweitig verwendet werden.

Konsequenzen für Programmierung und Konfiguration

Um auf die Variablen der Bussysteme aus der Robotersteuerung heraus zugreifen zu können, wird bei diesem Ansatz TwinCAT I/O verwendet, anstatt direkt einen Treiber für EtherCAT einzusetzen. Dazu wird ein weiterer Task in TwinCAT angelegt, der zyklisch Variablen liest und schreibt. Diese Variablen bilden die Kommunikation zur Robotersteuerung ab. Schreibt die Robotersteuerung in diese Variablen, so stehen sie im nächsten Zyklus in TwinCAT zur Verfügung und werden auf die entsprechend zugewiesene Variable am Bus geschrieben (vgl. Abb. 3). Der Lesezugriff erfolgt analog: Die Taskvariablen werden zyklisch von TwinCAT befüllt, so dass sie im nächsten Zyklus von der Robotersteuerung ausgelesen werden können.

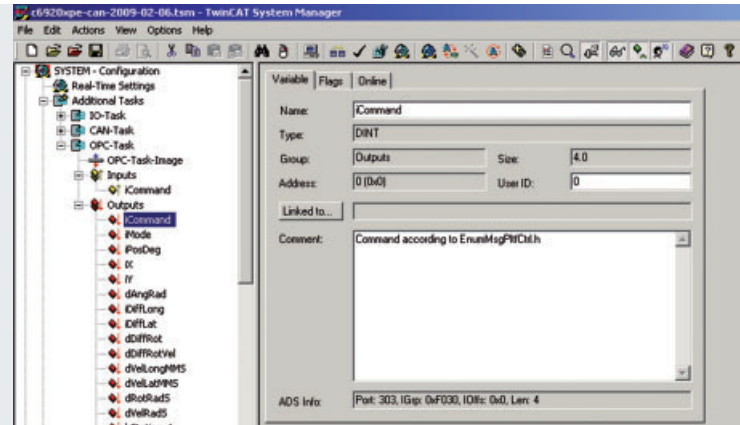


Abb. 3: Konfiguration der Prozessvariablen in TwinCAT

Der eigentliche Clou besteht aus Sicht der Robotersteuerung in der Verwendung dieser zusätzlichen Taskvariablen. Der einmalige Konfigurationsaufwand enthält zum einen das Anlegen der Task. Dabei werden im Wesentlichen Zykluszeit und Priorität festgelegt. Zum anderen werden innerhalb dieser Task Variablen definiert. Für jede Variable muss definiert werden, ob es sich um eine Ausgangs- oder Eingangsvariable handelt und welchem Datentyp sie entsprechen soll. Innerhalb des System Managers werden diese Variablen mit den Prozessvariablen der Geräte an den Feldbussen verknüpft. Um die Werte an die Robotersteuerung anzubinden, wird aus dieser Task ein Headerfile in der Programmiersprache C++ exportiert, das wiederum in die Robotersteuerung eingebunden wird. Mit Hilfe dieses Headerfiles und einer zusätzlich zu linkenden Bibliothek erfolgt der Zugriff auf die Prozessvariablen.

Der Vorteil besteht darin, dass die angeschlossene Hardware über den grafischen System Manager den Variablen des zusätzlichen Tasks zugeordnet werden kann. Bei veränderter Hardware muss also nur dieser eine Konfigurationsschritt erneut erfolgen, ohne dass die Robotersteuerung selbst davon etwas mitbekommt.

Werden Geräte wie Motorcontroller an einen Feldbus – wie hier den CAN-Bus – angeschlossen, so wird die Hardwareschicht CAN-Bus mittels EtherCAT gekapselt. Zur Robotersteuerung hin ist nur das EtherCAT-Gerät sichtbar. Mit Hilfe der Geräteprofile werden sowohl die Controller als auch das Bussystem hinter EtherCAT austauschbar, ohne dass tiefgreifende Softwareanpassungen notwendig sind.

Ausblick

Die Antriebscontroller für die Fahrmotoren des Roboters sind aktuell per CAN-Bus an die übergeordnete Software angebunden. Die Regelalgorithmen der Antriebe werden über den Bus im IPC geschlossen und haben dementsprechend hohe Anforderungen an den Bus. Dabei ist festzustellen, dass die Übertragung die Bandbreite des CAN-Busses bis an ihre Grenzen ausreizt. Vor allem bei komplizierteren Kinematiken – und damit entsprechend komplexeren Regelalgorithmen – wird klar, dass der CAN-Bus das limitierende Element bezüglich der Zykluszeit ist. Um hier Abhilfe zu schaffen, wird EtherCAT zukünftig als Antriebsbus eingesetzt.