



Denis Hatebur, Friedrich Hülscher, Christoph Hüntemann, Thomas Rottke

Die Bluetooth-Portierung

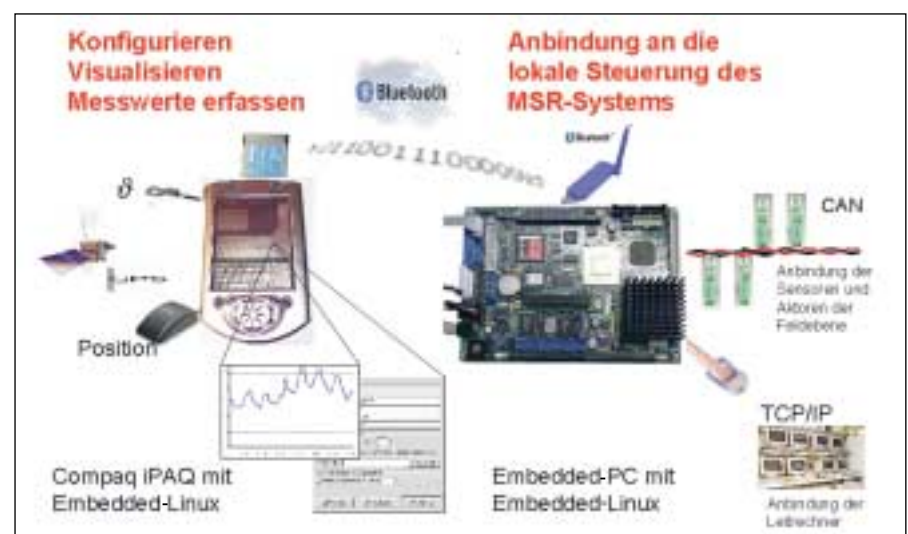
Drahtgebundene Anschlüsse an lokalen Komponenten eines verteilten MSR-Systems sind nicht unproblematisch – oft sind diese Schnittstellen nur schwer zugänglich. Mobile Wartungseinheiten mit „Air-Interfaces“ können hier Abhilfe schaffen. Welcher Aufwand hinter einer solchen Lösung steckt, zeigt folgende Fallstudie einer Bluetooth-Anbindung.

Für Wartungs- und Konfigurationsaufgaben in verteilten MSR-Systemen (Messen, Steuern, Regeln) sind mobile Wartungsgeräte zum Beispiel in Form eines Handheld-Computers ideal geeignet. Sie lassen sich gut transportieren und sind in der Lage, Informationen der Leitebene über den Zustand des Systems und der Systemkomponenten darzustellen. Ein weiterer Reiz dieser Geräte liegt darin, mit ihnen zusätzlich Messdaten zu erfassen und diese online an das MSR-System zu übermitteln.

Trotz der offensichtlichen Vorteile einer solchen Lösung gegenüber einer drahtgebundenen Schnittstelle sind „Air-Interfaces“ bislang nur unzureichend in Embedded-Systeme für MSR-Anwendungen integriert. Eigentlich unverständlich, da sich die Integration mittlerweile mittels kostengünstiger Standardtechnologien und ohne Programmieraufwand bewerk-

stelligen lässt. Dies zeigen die Erfahrungen im Zusammenhang mit einer Fallstudie bei Itesys.

Konkret ging es dabei darum, einen Wartungs-PDA (Personal Digital Assistant) mit Embedded-Linux zur Visuali-



Die Struktur des realisierten Bluetooth-Systems.

sierung von Messdaten, zur Konfiguration des MSR-Systems und zur Messwert-erfassung einzusetzen. Der PDA ist über Bluetooth mit einer Unterstation verbunden. Diese besteht aus einem Embedded-PC, auf dem ein speziell angepasstes Linux und der Bluetooth-Stack „BlueZ“ laufen. Über Ethernet und TCP/IP ist die Unterstation mit der Leitstation verbunden und über CAN mit der Feldebene. Alternativ ließe sich auch ein PDA mit Windows-CE einsetzen, sofern die entsprechende Windows-Variante Bluetooth-Unterstützung bietet. Da der Unterstationsrechner aber mit Linux betrieben wird, bot es sich aus Gründen der Homogenität, Quelloffenheit und Kompatibilität an, Linux auch auf dem PDA einzusetzen. Aus dieser Architektur ergeben sich verschiedene Vorteile:

- ▷ Bluetooth hat sich als Standard für drahtlose Übertragung bei Distanzen bis zu 100 m bewährt und ist für mobile Wartungsaufgaben besonders geeignet.
- ▷ Linux erfreut sich auch im Embedded-Bereich in der Variante „Embedded Linux“ zunehmender Beliebtheit. Gerade die Offenheit des Systems macht Linux besonders interessant, wenn in einem Entwicklungsprojekt bei der Integration mit Anpassungsarbeiten zu rechnen ist.
- ▷ Das Open-Source-Konzept garantiert, dass für Embedded-Linux und den Bluetooth-Stack keine Lizenzkosten anfallen. Die Entwicklungsumgebungen für Linux – bestehend aus GCC mit Compiler und Debugger – sind gut dokumentiert und ebenfalls kostenfrei.
- ▷ Für Embedded-Linux ist kostengünstige Standardhardware verfügbar; ein Bluetooth-Dongle kostet weniger als 50 Euro.
- ▷ Für Standardkonfigurationen lassen sich fertige Embedded-Linux- und Bluetooth-Installationspakete direkt und kostenlos aus dem Internet herunterladen. Für speziellere Konfigurationen ist die Installationsarbeit natürlich aufwendiger.

Die Realisierung im Detail

Für den verwendeten PDA, einen Compaq iPAQ 3660, gibt es als Linux-Distribution das „Familiar Linux“ mit den Oberflächen Opie (Qt-basiert) oder GPE (GTK-basiert). Beide Versionen werden auf der Entwickler-Homepage www.handhelds.org als fertige Images angeboten, die

direkt in den Flash-Speicher des Handhelds geladen werden können. Familiar Linux enthält bereits eine Bluetooth-Unterstützung in Form von BlueZ, dem Standard-Protokollstack von Linux. Beide Varianten von Familiar bieten die gewohnten Programme wie bash, ssh-Server und ssh-Client, sftp-Server und sftp-Client. Der Umgang mit diesem System erfordert also wenig Einarbeitung, sofern man bereits Desktop-Erfahrung mit Linux

hat. Noch fehlende Komponenten wie zum Beispiel ein Web-Browser, um mit dem Unterstationsrechner kommunizieren zu können, lassen sich ebenso unter der genannten Web-Adresse herunterladen und installieren.

Die Installation der Familiar-Distribution läuft wie folgt ab: Da das bestehende System eventuell später wieder hergestellt werden soll, empfiehlt es sich, als ersten Schritt ein Backup davon zu machen.

Anschließend ist der Bootloader zu installieren, mit dem das Flash-Image zum Beispiel mit Hilfe eines Terminal-Programms über die serielle Schnittstelle geladen werden kann. Der gesamte Zeitaufwand hierfür beträgt etwa 20 Minuten zuzüglich der Ladezeit für das Backup.

Das speziell angepasste Linux, das auf der Unterstation läuft, wurde auf der Basis von LEAF realisiert. Der Begriff LEAF steht für Linux Embedded Appliance Firewall. Das Ziel dieses Projektes ist es, einen Linux-Router mit Firewall-Funktion zu erstellen, der möglichst wenige Ressourcen benötigt. Die Grundlage hierfür ist die Distribution des Linux Router Project (LRP). Dieses „minimalistische“ Linux benötigt nur wenig Festpeicher (kleiner 2 MByte) und relativ wenig Hauptspeicher (kleiner 12 MByte). Die Hauptherausforderung hinsichtlich der Anpassung bestand darin, Bluetooth in das Embedded-System zu integrieren und so zu compilieren, dass der Speicherbedarf des Bluetooth-Stack und der erforderlichen Tools den üblichen Rahmen für Embedded-Systeme (kleiner 2 MByte) nicht überschreitet. Die LRP-Distribution benötigt nur eine Diskette und läuft nach dem Start des Systems komplett im Hauptspeicher.

Mit der Distribution „Bering“ des LEAF-Projektes, deren Code-Basis aus dem Linux Router Project (LRP) stammt und die im Gegensatz zu anderen Ableitungen aus dieser Code-Basis den Kernel 2.4 verwendet, steht von Seiten des Kernels her erstmals eine Bluetooth-Unterstützung zur Verfügung. Allerdings ist der Kernel 2.4 um einiges größer als Linux 2.2 oder gar Linux 2.0, die beide

auf Grund ihrer Größe und Stabilität in sogenannten Mini- oder Eindisketten-Linux-Versionen eingesetzt werden. Allein der Kernel benötigt etwa 520 kByte, und ein lauffähiges Bering – ohne wirkliche Anwendungen und nur mit Systemtools – belegt rund 1,3 MByte. Da jedoch – was eine Bluetooth-Unterstützung angeht – derzeit keine Alternative zur Bering-Variante besteht, sind diese Nachteile in Kauf zu nehmen.

Für den vorliegenden Anwendungsfall mussten am Bering-System einige Anpassungen vorgenommen werden. Als Ausgangspunkt diente ein LEAF-Bering-System in der Version 1.2, das jedoch noch keine aktuelle, vollständige und funktionierende Bluetooth-Unterstützung enthält. Auf Grund der schnellen Entwicklung und sicherheitskritischer Fehler in den letzten Kernel-Versionen war es ratsam, einen aktuellen Kernel mit aktivierten Bluetooth-Optionen zu installieren. Da der Speicherplatz bei einem Embedded-System begrenzt ist, musste der Kernel vor der Installation noch mit einem Executable-Packer komprimiert werden. Man erreicht so eine Verkleinerung auf etwa 85 % der Originalgröße. Die Erstellung eines eigenen Kernels bietet den Vorteil, alle benötigten Module schon statisch in den Kernel compilieren zu können. So entfällt die Notwendigkeit, die Kernelmodule in das System zu integrieren.

BlueZ-Erweiterung für LEAF

Die größte Schwierigkeit beim Aufbau des Systems bestand in der Integration des Bluetooth-Subsystems in LEAF-Bering. Um dies zu erreichen wurde der offizielle

Linux-Bluetooth-Protokollstack BlueZ portiert. Das BlueZ-System setzt sich aus drei Komponenten zusammen: Module, Applikationen und Libraries.

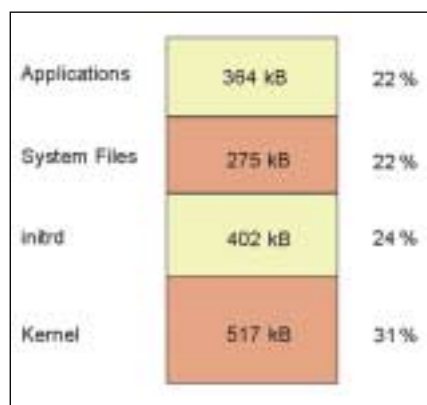
Module

Bei BlueZ gibt es drei Arten von Kernelmodulen. Die oberen Schichten sind Bluetooth-Protokollmodule. Diese sind Implementierungen der gleichnamigen Protokolle der Bluetooth-Spezifikation beziehungsweise der Bluetooth-Profil-spezifikation. Darunter befindet sich „bluez.o“, das zentrale Modul des BlueZ-Stacks. Dieses Modul fungiert als HCI-Treibermodul (Host Controller Interface) und reicht die HCI-Pakete an die unteren Hardware-Treiber weiter. Die HW-Treibermodule dienen als Schnittstelle zwischen dem HCI-Treiber bluez.o und anderen Hardware-Treibern, welche dann direkt auf die Geräte zugreifen.

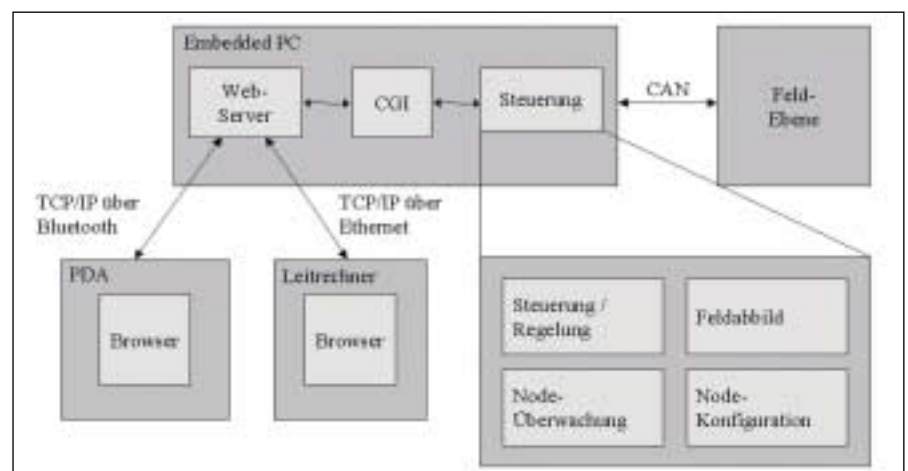
Applikationen

Um den BlueZ-Stack nutzen zu können, ist es nicht nur wichtig, die Kernelmodule des Stacks zu kennen. Auch die zugehörigen Anwendungen, Test- und Konfigurationstools sind von Bedeutung. Wichtig sind in diesem Zusammenhang der HCI-Daemon „hcid“, der SDP-Daemon „sdpd“ (Service Discovery Protocol), das Konfigurationstool „hciconfig“ und der PAN-Daemon „pand“ (Personal Area Network).

Der Daemon hcid, der im Hintergrund läuft, erfüllt im Wesentlichen drei Aufgaben. Bei seinem Start werden alle bis dahin vorhandenen HCI-Geräte initialisiert und mit Standardwerten konfiguriert. Die Konfiguration erfolgt dabei auf der Grundlage einer Konfigurations-



Aufteilung des Festpeicherbedarfs von LEAF-Bering: Über 75 % des Platzes auf der Diskette (1,68-MByte-Image) benötigt allein das System.



Das Komponentendiagramm: Ein Wartungs-PDA und die Steuerungsrechner der Leitebene werden über ein Web-Interface mit dem MSR-System verbunden.



datei, in der alle grundsätzlichen Einstellungen für die jeweiligen Geräte getroffen werden können. Die zweite Aufgabe des Programms ist es, im Betrieb neu hinzugekommene Geräte entsprechend zu konfigurieren. Darüber hinaus erfüllt das Programm noch eine Protokollierungsfunktion.

Das Service Discovery Protocol (sdpd) dient dazu, anderen Geräten die Dienste, die das eigene Gerät anbietet, zu nennen. Der SDP-Daemon ist das dafür zuständige Programm.

Mit dem Tool „hciconfig“ kann ein HCI-Gerät, also ein Bluetooth-Gerät, manuell konfiguriert werden. Außerdem lassen sich damit der Name und die MAC-Adresse des Gerätes abfragen. Das Tool ist vergleichbar mit dem Tool „ifconfig“ für Netzwerkkarten.

Das Personal Area Network ermöglicht es, ad hoc ein Netzwerk aufzubauen. Das dabei eingesetzte Protokoll heißt „BNEP“ und steht für „Bluetooth Network Encapsulation Protocol“. Dieses Protokoll verpackt Ethernet-Frames in Bluetooth-Pakete. Die beteiligten Geräte erhalten bei Verwendung des „pand“ ein virtuelles Netzwerk-Interface mit Namen „bnep0“, dem eine IP-Adresse zugewiesen werden kann. Über diese IP-Adresse ist das Gerät dann ansprechbar.

Libraries

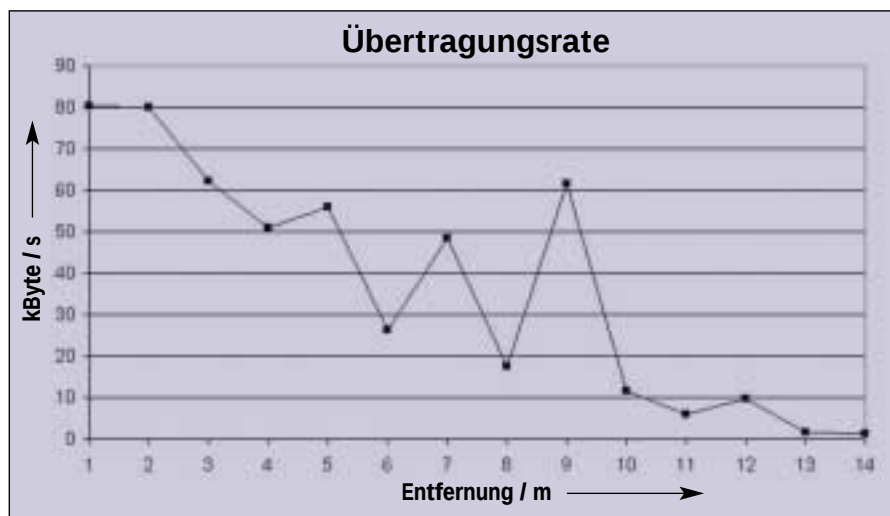
Zusätzlich zu Modulen und Applikationen gehören auch noch Libraries zu BlueZ. Darin sind die wichtigsten und immer wieder benötigten Funktionen enthalten.

Die Anbindung an das Feldbus-System

Die Steuerung der Feldebene erfolgt bei der beschriebenen Applikation über CAN. Die Steuerungssoftware läuft auf dem Embedded-PC und setzt sich aus den Komponenten Steuerung/Regelung, Feldabbild, Node-Überwachung und Node-Konfiguration zusammen. Die Konfiguration und Überwachung des Systems erfolgt über einen Web-Browser auf dem PDA oder dem Leitrechner. Dazu ist es notwendig, dass auf dem Unterstationsrechner ein Web-Server läuft. Dieser wird von LEAF-Bering in Form des Paketes „weblet.lrp“ bereitgestellt. Das Paket enthält einen Shellskript-Web-Server, der normalerweise dazu dient, Statusinformationen über das Bering-System abzufragen. Die Kommunikation zwischen Web-Browser und Web-Server erfolgt über TCP/IP, wobei der Leitrechner Ethernet und der PDA Bluetooth als Übertragungsweg nutzt. Die Anfragen des Browsers werden auf Seiten des Embedded-Systems von CGI-Skripten beantwortet, welche über Unix-Sockets mit der Steuerungskomponente Daten austauschen.

Portierung auf das Zielsystem

Programme, die für ein „normales“ Linux-System kompiliert wurden, können nicht ohne weiteres auf einer Mini-Distribution wie LEAF-Bering gestartet werden – der Grund dafür sind unterschiedliche glibc-Versionen. Die glibc ist die GNU-Variante der libc. Alle in C geschriebenen Programme werden dagegen gelinkt. LEAF-Bering verwendet aus Platzgründen eine ältere glibc-Version, da diese im Vergleich zu neueren Versionen deutlich kleiner ist. Programme, die gegen neuere glibc-Versionen



Das Ergebnis der empirischen Messung: Bis zu einem Abstand von 5 Metern kann mit einer relativ gleichmäßigen, hohen Datenrate gerechnet werden. Bei Abständen von 6 bis 10 Metern ist die Transferrate stärker vom Umfeld abhängig, und es kommt je nach äußeren Bedingungen zu stärkeren Schwankungen. Abstände von über 10 Metern führen zu einer extremen Verschlechterung der Übertragungsleistung; bei diesen Entfernungen ist praktisch keine sinnvolle Anwendung mehr möglich.

gelinkt worden sind, müssen daher portiert werden.

Zur Portierung ist es erforderlich, auf dem Entwicklungsrechner eine Entwicklungsumgebung für das entsprechende Zielsystem zu installieren. Mit diesem „Cross-Compiler“ ist es möglich, auf dem Entwicklungssystem Programme für das Zielsystem zu übersetzen. Nach der Portierung sind die Applikationen und Libraries in das Standard-Archivformat der LEAF-Bering-Distribution zu bringen. Diese Archive haben die Endung „.lrp“ und sind im Wesentlichen „tar.gz“-Archive mit spezifischen Datei-Erweiterungen, die von Bering zur Konfiguration des Systems benutzt werden.

Damit die Applikationen automatisch starten, empfiehlt es sich, ein Startup-Skript in „/etc/init.d/“ zu erstellen und entsprechende Links anzulegen. Nutzt man LEAF-Bering in der Standardversion

(Bilder: Itesys)

mit einer Diskette, ist hierfür erst noch Platz auf der Diskette zu schaffen und nicht benötigte Pakete müssen gelöscht werden. Für das Paket mit den Anwendungen und das um die Bluetooth-Treiber erweiterte Modul-Paket werden rund 220 kByte benötigt. Diese Archive sind auf die Diskette zu kopieren, anschließend ist das (Applikations-)Paket in die Konfigurationsdatei „syslinux.cfg“ des Bootloaders „syslinux“ einzutragen. Die für eine solche Bluetooth-Applikation unbedingt notwendigen Hardware-Treiber und Protokollmodule bluez.o, l2cap.o, bnep.o, hci_usb.o, usbcore.o, und usb_ohci.o beziehungsweise usb_uhci.o sind mit dem entsprechenden Kernel zu compilieren und in das System zu integrieren.

Nach dem Neustart des LEAF-Systems ist mittels „hciconfig“ überprüfbar, ob der eingesteckte Bluetooth-Dongle erkannt wird. Wenn nicht, sind wahrscheinlich nicht alle benötigten Module geladen oder die notwendigen Applikationen laufen nicht.

Der Verbindungs-Aufbau

Nach einer vollständigen Konfiguration der Unterstation kann der PAN-Daemon „pand“ auf dem Embedded-PC mit dem Parameter „--listen“ gestartet werden. Der Daemon wartet dann im Hintergrund auf eine eingehende Verbindung. Vom PDA kann sich der Bediener anschließend mittels „pand --connect BLUETOOTH_ADRESSE“ mit der Unterstation ver-

binden. Kommt eine Verbindung zustande, wird ein Netzwerk-Interface erzeugt, welches nur für die Dauer der Verbindung existiert. Mit „ifconfig“ lässt sich eine IP-Adresse für dieses Interface einstellen, zum Beispiel mit

```
:root@linux#> ifconfig bnep0 10.0.0.1
```

Der Datenaustausch zwischen PDA und Unterstation kann nun über übliche Internet-Techniken erfolgen.

Sinnvoll bis 10 Meter Entfernung

Um Daten über die Leistungsfähigkeit und Zuverlässigkeit der Bluetooth-Übertragungsstrecke zu ermitteln, wurden empirische Messungen durchgeführt. Bei der Aufzeichnung der Datenübertragungsraten in Abhängigkeit von der Entfernung stellte sich heraus, dass in einem Abstand von knapp 10 Metern die Datenrate zur Messwert-Erfassung noch ausreicht. Der Bereich jenseits der 10 Meter wurde durch eine Wand gemessen, was zwar durchaus Praxisbedingungen entspricht, aber auch das Ergebnis entsprechend beeinflusst. Die Messungen erfolgten unter Büro-Bedingungen mit Class-3-Bluetooth-Adaptoren. Da Bluetooth extrem störungsfähig ist, dürfte es auch im Industrie-Umfeld zu keinen größeren Verschlechterungen kommen. gh

Nähere Informationen:

Bluetooth-Pakete für LEAF-Projekt: www.itesys.de
Embedded Linux LEAF Project: <http://leaf.sf.net>
Embedded Linux für PDAs: <http://www.handhelds.org>
Installationsanleitung von Familiar Linux: <http://familiar.handhelds.org/familiar/releases/v0.7.1/install/>

Denis Hatebur

ist Geschäftsführer
der Firma Itesys Institut für Technische Systeme.

Friedrich Hülischer

ist Student der technischen Informatik
an der FH-Dortmund.

Christoph Hüntemann

ist Student der technischen Informatik
an der FH-Dortmund.

Thomas Rottke

ist IT-Manager bei DNV Det Norske Veritas und
Dozent an der FH-Dortmund.