

Bluetooth ausgereizt

Der Bluetooth-Standard eignet sich ideal für die kabellose Datenerfassung per Pocket-PC. Applikationsseitig sind bei diesen Handhelds jedoch kleinere Erweiterungen notwendig, um die Potenziale dieser Funktechnologie optimal ausnutzen zu können.

Wer Kabel durch Nahbereichs-Funktechnologie ersetzt, kann eine ganze Reihe verschiedener Vorteile für sich verbuchen. Die drahtlose Datenübertragung hilft nicht nur, unübersichtlichen Kabelsalat auf das absolut notwendige Maß zu reduzieren. Sie vermag auch viele Kontrollaufgaben substanzial zu erleichtern beziehungsweise ermöglicht diese überhaupt erst. Sensoren an rotierenden Teilen oder schwer zugänglichen Orten zum Beispiel erfordern bis dato häufig eine aufwendige

Verkabelung. Nicht so bei der Datenübertragung per Funk, die auch das leidige Steckerproblem löst -- ein enormes Vorteil in anspruchsvollen Applikationen wie etwa dem Ex-Bereich.

Nicht selten weisen solche Funklösungen jedoch einen hohen Komplexitätsgrad auf, der dem eigentlichen Ziel -- komplizierte Verkabelungen durch einfachere Entwicklungen zu ersetzen -- zuwiderläuft. Mit anderen Worten: Speziell entwickelte und hoch spezialisierte Funk-Lösungen schaffen zwar die störenden Leitungen ab; unter Umständen jedoch zum Preis hoher Entwicklungskosten, geringer Flexibilität in Punkto Erweiterbarkeit und erschwerter Anpassung an geänderte Aufgabenstellungen.

Bluetooth als Standard-Technologie aus dem Consumer-Umfeld bietet hier den Vorteil, dass für diese Technologie günstige Komponenten zur Verfügung

stehen, die sich einfach in bestehende Systeme integrieren und hervorragend für spontane Kommunikationsverbindungen einsetzen lassen. Und: Da Bluetooth heute standardmäßig auf vielen mobilen Pocket-PCs integriert ist, die bekanntlich auch in Ex-geschützten und äußerst robusten Ausführungen zu bekommen sind, muss diese Funk-Funktionalität nicht für jede Anwendung in eigenen Handgeräten neu implementiert werden. Außerdem lassen sich viele Diagnose- und Konfigurationstools, die auf Windows-basierten Desktops oder Notebooks zum Einsatz kommen, ohne weiteres auf Pocket-PCs portieren.

Noch etwas spricht für Bluetooth: Diese Technologie ist geradezu prädestiniert, Industriebusse abzubilden. Da sie ursprünglich für zeitkritische Anwendungen wie die drahtlose Sprachübertragung



bei Mobiltelefonen entwickelt wurde, verfügt sie über eine gewisse Echtzeitfähigkeit, die durch die Intelligenz des Bluetooth-Standards bereits auf unterer Protokollebene gewährleistet ist. Dies ist ein entscheidender Unterschied zu WLAN, dem im Bereich Nahbereichsfunk (im 2,4 Ghz-Band) mindestens ebenso weit verbreiteten Übertragungsprotokoll: Hier findet – ähnlich wie bei Ethernet -- ein Verdrängungswettbewerb der Telegramme statt, der die Echtzeitzuverlässigkeit erheblich einschränkt – auch wenn auf höheren Protokollschichten versucht wird, durch Zusatztools einen gewissen Determinismus zu bewirken.

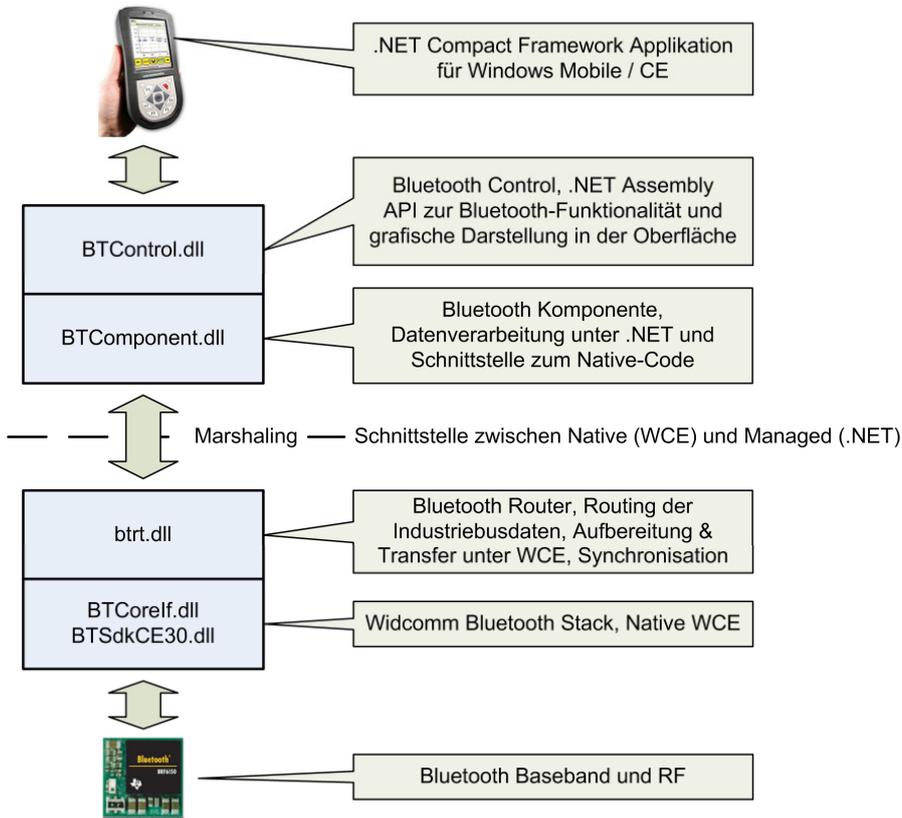
Der Aufwand, um die Bluetooth-Technologie für ein von Grund auf neues Projekt zu nutzen oder ein bestehendes seriell kommunizierendes System auf diesen Standard umzustellen, ist vergleichsweise gering. Der zunächst häufigste Fall ist sicherlich das „Upgrade“ – sprich das Bluetooth-„Enablen“ – eines bestehenden Systems, bei dem mehrere Sensoren oder Aktoren „klassisch“ über einen Industriebus an ein Bediengerät angeschlossen sind. Bei Pocket-PCs ist die Bluetooth-Hardware bereits integriert, bei PCs ist diese leicht nachrüstbar. Bei den Sensoren und Aktoren ist der Aufwand nur geringfügig höher: Auch hier hält der Markt eine Vielzahl von Komponenten bereit, so genannte „Bluetooth-Enabler“. Mittels solcher fertiger Module von verschiedenen Anbietern lassen sich bestehende Schnittstellen oder Microcontroller mit RS232- oder UART-Pegel nahtlos in ein Bluetooth-Funknetzwerk einbinden. Der Enabler muss hierzu im Idealfall lediglich mit den entsprechenden Pins des Microcontrollers verbunden werden. In vielen Fällen sind die Bluetooth-Enabler sogar vorkonfiguriert über UART-Ports in das System integrierbar. Für das Gros der Anwendungen ist damit hardwareseitig alles nötige getan.

Die Applikations-Problematik

Auf der Applikationsseite hingegen gestaltet sich der Bluetooth-Einsatz – will man die Potenziale dieses Systems voll ausnutzen – etwas umständlicher: Serielle Bluetooth-Verbindungen werden selbst in den aktuellsten Microsoft-Betriebssystemen Windows XP und Pocket-PC 2003 derzeit in Form eines virtuellen COM-Ports abgebildet. Das hat zwei Nachteile: Zum einen muss, um vom Pocket-PC aus mit einem Bluetooth-befähigten Sensor kommunizieren zu können, die Verbindung noch manuell aufgebaut werden; und zwar über eine entsprechende Verknüpfung auf dem PC-Desktop oder über die Windows-



Die kabellose Datenübertragung via Nahbereichsfunktechnologie hat nicht nur Vorteile in Ex-geschützten Bereichen. Sie kann auch aufwendige Wartungsvorgänge erheblich erleichtern.



Um eine hohe Betriebssicherheit zu gewährleisten, beschränkt das .NET-Framework den Zugriff auf sämtliche systemnahe Hardware des Pocket-PC. Dadurch wird die Nutzung des Bluetooth-Stacks von .Net-Compact-Framework-Anwendungen heraus unmöglich. BT-Control ergänzt diese Laufzeitumgebung um eine zusätzliche API, die Bluetooth optimal in das System einbindet.

Systemsteuerung. Zweitens lässt ein Pocket-PC-basiertes Gerät nur eine einzige virtuelle COM-Port-Verbindung zu. Für einfachere Anwendungen mag dies ausreichen; die wahren Potenziale der Bluetooth-Technik lassen sich damit allerdings nicht ausreizen. Ein großes Plus der Datenübertragung per Funktechnologie ist immerhin die Möglichkeit, mehrere parallele Verbindungen zu einer Vielzahl Sensoren gleichzeitig „im Blick“ zu haben. Außerdem wäre es sinnvoll, die Bluetooth-Funktionalität und ihre Konfi-

guration direkt in die Applikation zu integrieren. Das würde es zum Beispiel ermöglichen, verschiedene Sensoren gezielt über ihre eindeutige Bluetooth-ID anzu-steuern. Ein anwenderspezifisch maßge-schneidertes Bluetooth-Profil würde zu-dem helfen, Verwechslungen bei der Ver-bindungskonfiguration zu beseitigen -- und damit den Bediener erheblich entlas-ten.

.NET als Entwicklungsplattform

Grundsätzlich machbar ist dies alles mit Bluetooth -- dafür ist lediglich eine kleine Hürde zu nehmen: Wer für die Abbildung von Industriebussen heute Bluetooth-Funktionalitäten unter .NET erschließen will, muss den Umweg über spezielle .NET-Assemblies gehen, die den Zugriff auf die Bluetooth-Protokoll-Stacks (Bluetooth-Stacks) erlauben. Besonders interessant und verbreitet ist die Entwicklung im .NET-Compact-Framework von Microsoft, das mit dem Betriebssystem ab Pocket-PC 2003 ausgeliefert wird: Es bietet eine Fülle von gra-fischen Softwarekomponenten, die den Programmierer bei der Applikationsent-wicklung effektiv unterstützen. Auch gel-ten .NET-Applikationen als besonders stabil und eignen sich damit für den Ein-satz in wenig fehlertoleranten Einsatzbe-reichen.

Die „Schattenseite“ dieser hohen Einsatzsicherheit ist jedoch, dass dem Programmierer der Zugriff auf systemnahe Bereiche des PCs -- und damit auch auf die Bluetooth-Funktionalität -- erschwert

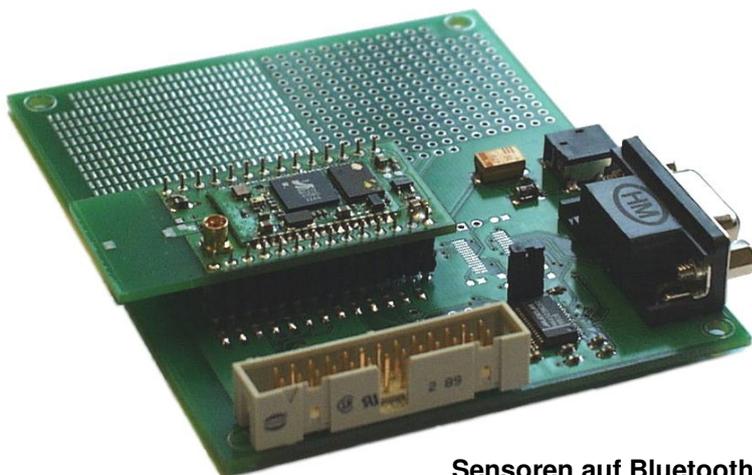
wird: Bluetooth ist ein Standard auf der Kommunikations-Ebene, nicht aber auf der Integrationsseite. Auf Pocket-PCs sucht man daher vergeblich nach einer Software-Schnittstelle, die aus dem .NET-Compact-Framework den Zugriff auf die Bluetooth-Funktionalität des Gerätes erlaubt. Microsoft hat in sein neues Betriebssystem CE.NET 5.0 zwar ein .NET-Assembly für den Zugriff auf Bluetooth integriert, allerdings wird dabei nur der „hauseigene“ Bluetooth-Stack unterstützt.

Spezielles API für den Bluetooth-Zugriff

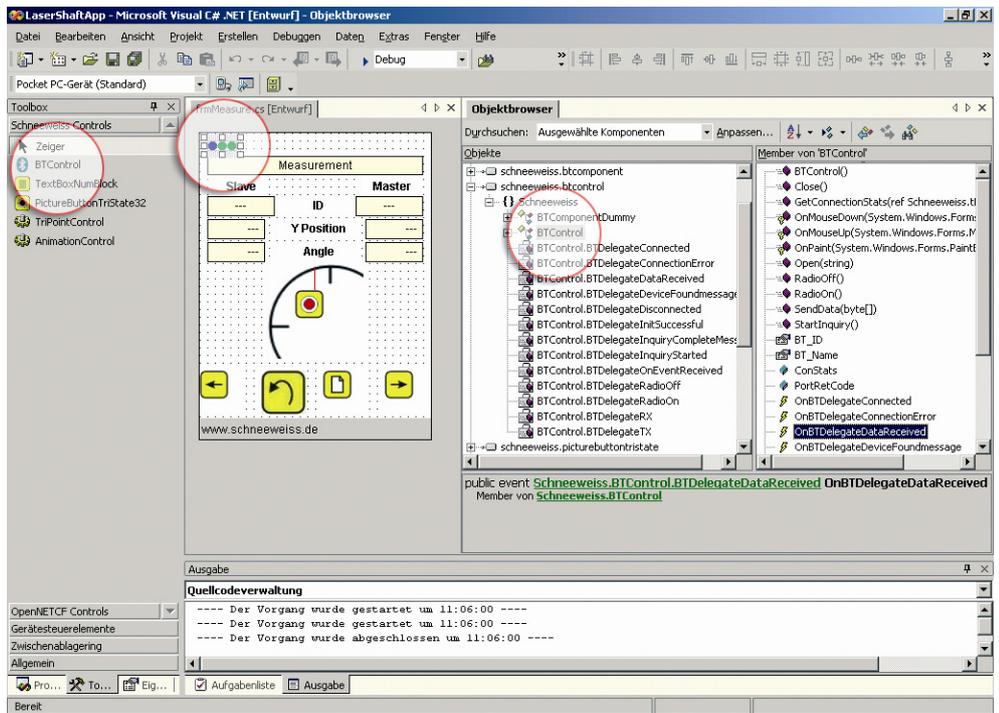
In der Praxis haben sich bei den Bluetooth-Stacks eine Vielzahl alternativer Anbieter etabliert, zum Beispiel Microsoft, Widcom (früher Broadcom), Nokia oder auch Toshiba. Besonders stark verbreitet ist der Widcom-Stack, der zum Beispiel in Windows-betriebenen Pocket-PCs von HP/Compaq arbeitet. Diesen Stack nutzt das von der Firma Schneeweiß Technische Systemprogrammierung entwickelte Tool „BT-Control“, um dem .NET-Framework eine API für den Zugriff auf Bluetooth zur Seite zu stellen. Diese API ermöglicht es, Industriebusprotokolle über das serielle Bluetooth-Profil abzubilden. Damit ist es machbar, mit Hilfe bewährter Standard-Entwicklungs-Werkzeuge wie Visual Studio und dem .NET-Compact-Framework in kürzester Zeit Messtechnik-Applikationen für Pocket-PCs zu programmieren und via Bluetooth mit entsprechend befähigten Sensoren und Aktoren zu verbinden.

Die Bluetooth-Control Software-Bibliothek besteht aus mehreren Schichten: Auf unterster Ebene wird in nativem Code eine Schnittstelle zum Bluetooth-Stack von der Firma Widcomm/Broadcom) geschaffen. Eine weitere native Schicht dient der Datenaufbereitung und dem Datentransfer zum .NET-Code sowie der Pufferung und Synchronisation der Zugriffe aus den gemanageten .NET-Schichten. Die erste CE.NET-Schicht dient der Datenaufbereitung und dem Verteilen der Telegramme auf .NET-Seite; sie wird von der eigentlichen Komponente -- dem Control, welches aus dem Anwendungscode referenziert wird -- genutzt. Das Anwender-Control bietet die Benutzer-API an und sorgt sowohl im Designmodus als auch zur Laufzeit der Applikationen für die Darstellung auf dem Mobilgerät.

Die Herausforderung bei der Implementierung dieser Schnittstelle bestand darin, die Daten auf geeignete Art und Weise zwischen der .NET-Umgebung und dem systemnahen Code zu transferieren. Hier wurde durch eine Kombination aus Windows-Benachrichtigung, Platform-Invoke und Marshaling ein optimales Ergebnis in Bezug auf Asynchronität und Funktionalität erreicht.



Sensoren auf Bluetooth umzustellen ist nicht schwer – hier bietet der Markt inzwischen zahlreiche Möglichkeiten. Manche Bluetooth-Enabler können sogar vorkonfiguriert an den UART-Ports in das System integriert werden.



BT-Control integriert sich vollkommen in die Visual-Studio-Umgebung von Microsoft. Dem Anwendungsentwickler steht die Komponente über die Toolbox per Mausklick zur Verfügung, und darüber hinaus auch die komplette API. So kann die Bluetooth-Funktionalität effizient während der „üblichen“ Entwicklung mit dem .NET Compact Framework genutzt werden.

Das Benutzer-Control steht nach der Installation in der Visual Studio Toolbox zur Verfügung. Bei der Verwendung dieses Controls wird auf die Bluetooth-Control-Assembly verwiesen. Hier stehen Methoden zur Verfügung, die die Nutzung von Kanälen zu den Bluetooth-Knoten ermöglichen. Weiterhin sind Informationen wie Feldstärke oder Informationen zu benachbarten Bluetooth-Knoten abrufbar oder werden durch Benachrichtigungen abgebildet. In der Designer-Oberfläche stellt sich die Komponente als Symbol mit drei Signalen dar: Status der Verbindung, gesendete Daten und empfangene Daten. Diese werden zur Laufzeit visualisiert.

Mittels BT-Control lassen sich somit Telegramme, die bisher in haus-eigenen Formaten über RS-Schnittstellen umständlich per Kabel eingesammelt werden mussten, über das serielle Bluetooth-Profil abbilden. Das Pocket-PC-basierte Gerät kommuniziert direkt über Bluetooth mit den Sensoren und übernimmt die Datenerfassung und -Verarbeitung. Mit anderen Worten: Auf Basis der geschilderten Tool-Unterstützung lassen sich die Potenziale der drahtlosen Nahbereichs-Funktechnologie im vollen Umfang mit Hilfe der ausgesprochen wirtschaftlichen Bluetooth-Hardware realisieren. Das bedeutet: Voller gleichzeitiger Zugriff

auf mehrere Sensoren, sichere Steuerung der Kommunikation durch die Applikationssoftware sowie mobile Auswertung der drahtlos erhaltenen Messwerte auf dem Pocket-PC.

Dieser integrative Ansatz bietet dem Anwender somit einerseits die Vorteile der kabellosen Datenerfassung, und eröffnet ihm zusätzliche Optionen in Punkto mobiler Datenverarbeitung. Denkbar ist zum Beispiel eine Pocket-PC-gestützte Software, die „im Vorübergehen“ mit den zu wartenden Einheiten korrespondiert, über die Wartungsnotwendigkeit entscheidet und zurück am zentralen Rechner für einen automatischen Datenabgleich mit dem Prozessleitsystem sorgt. Zugleich hält die Software in ihrer Datenbank die „Geschichte“ jedes einzelnen Geräts vor Ort parat.

Nähere Informationen:
www.schneeweiss.de



Jens Schneeweiß

Ist Inhaber eines Ingenieurbüros für Technische Systemprogrammierung in Herten