



sevenstax

Embedded Internet Technologie

sevenstax GmbH
info@sevenstax.de
www.sevenstax.de

Revision 4.3



Embedded Internet

Inhaltsverzeichnis

Internet-Kommunikation.....	3
Basis-Protokolle.....	3
TCP/IP.....	4
PPP / Ethernet.....	4
Applikationsprotokolle.....	5
HTTP.....	5
SMTP / POP3.....	6
XML / SOAP.....	6
FTP.....	6
Applikationen.....	7
E-Mail.....	7
Embedded Web-Server.....	8
Embedded Client für Web Services.....	8
Anwendungen.....	9
Konfiguration von einzelnen Geräten.....	9
Echtzeit Kommunikation mit Geräten	10
Anwendungsspezifischen Kommunikation über IP-Netze.....	11
Protokolldaten und Fehlermeldungen verteilter Geräte.....	12
Wartung und Diagnose von Geräte- und Maschinenpools.....	12
sevenstax Protokoll-Implementationen.....	15
sevenstaxTCP/IP.....	16
sevenstaxPPP.....	17
sevenstaxXML-Parser/Generator.....	18
sevenstaxSOAP-Client.....	19
sevenstaxWeb-Server.....	20
sevenstaxMail-Client.....	21
sevenstax – das Unternehmen.....	22
Kontakt.....	23
Glossar.....	23

Die Vernetzung der verschiedensten eingebetteten Systeme ist eine wichtige Anforderung an neue Geräte-Generationen, unabhängig davon, ob diese neu entwickelt oder im Zuge der Produktpflege erweitert werden. Während die Vernetzung von PCs und ähnlich leistungsfähigen Architekturen bereits Standard ist, trifft dies auf das breite Feld der Embedded Systeme noch nicht zu. Oftmals erscheint die Erweiterung dieser Geräte um die Netzwerk-Technologie als zu teuer, zu kompliziert oder der Bedarf an Ressourcen wie Speicher und CPU zu groß. Die nachfolgenden Seiten möchten Ihnen Lösungsansätze zeigen, wie ein Zugang zum Internet auch mit günstigen Mikrocontrollern und geringem Aufwand zuverlässig erreicht werden kann.

Internet-Kommunikation

Das Internet ist die weltweit verfügbare Infrastruktur zur Übertragung von Daten. Der entscheidende Vorteil liegt in der kostengünstigen Nutzung. Da die Übertragungsmedien bereits vorhanden sind, fallen für den Nutzer nur geringe, meist zeitabhängige, Gebühren an, wobei es keine Rolle spielt, wohin die Daten übertragen werden. Ist ein Gerät einmal mit dem Internet verbunden, so kann sich sein Kommunikationspartner irgendwo auf der Welt befinden. Der Zugang zum Netz kann auf verschiedene Weisen erfolgen, am gebräuchlichsten sind hierfür das Telefonnetz oder ein vorhandenes (Ethernet-)Netzwerk, z.B. ein Intranet. Aber auch drahtlose Anbindungen kommen vermehrt zum Einsatz, wie zum Beispiel GPRS (General Packet Radio Service) oder zukünftig UMTS (Universal Mobile Telecommunication System) im Bereich der Mobiltelefonie. All diese Verfahren haben als Ziel die Vernetzung der verschiedensten Geräte untereinander oder den Datenaustausch zwischen diesen Geräten und Servern.

Basis-Protokolle

Die wohl bekanntesten Protokolle des Internet sind TCP (Transmission Control Protocol) und IP (Internet Protocol). Im allgemeinen Sprachgebrauch wird allerdings oftmals der Begriff TCP/IP für die Gesamtheit aller für die Kommunikation verwendeten und in einem TCP/IP-Stack integrierten Protokolle benutzt. Grundsätzlich sind TCP/IP-Stacks in verschiedene Schichten unterteilt, die unterschiedliche Aufgaben erfüllen und aus deren Kombination sich die jeweils passende Lösung für eine Aufgabenstellung ergibt. Angelehnt an das OSI-Referenzmodell werden bei TCP/IP vier der möglichen sieben Schichten implementiert.

Schichten	Beispiele
Applikationsschicht	Telnet, FTP, SMTP, POP3, Web-Server
Transportschicht	TCP, UDP
Vermittlungsschicht	IP
Sicherungsschicht	PPP, ARP, RARP

Die oberste Schicht ist die Applikationsschicht. In dieser laufen alle Programme, die Daten über das Internet austauschen wollen, sowie die hierfür spezifischen Protokolle. Typische Standard-Anwendungen hierfür sind z.B. Telnet, die Übertragung von Dateien (FTP-Protokoll), E-Mail (SMTP- und POP3-Protokoll) oder Web-Server (HTTP-Protokoll). Neben den erwähnten Anwendungen enthält die Applikation-Schicht aber auch alle Benutzerprogramme, die beispielsweise Geräte konfigurieren oder Messdaten an einen Server übertragen können.

Die Transportschicht besteht aus zwei Protokollen, TCP und UDP, die der darüber liegenden

Applikationsschicht zwei verschiedene Methoden zur Kommunikation bieten.

TCP bietet eine sichere Übertragung von Daten an. Hierfür werden beispielsweise Checksummen verwendet und einzelne Pakete mit Sequenznummern versehen, um die richtige Reihenfolge beim Empfang zu garantieren. Da TCP selbst für die Integrität der Daten sorgt, ist es das am häufigsten verwendete Transportprotokoll.

UDP bietet diese Sicherheit nicht, dafür liegt die Geschwindigkeit bei der Verarbeitung von Daten aber höher. Auf Grund der ungesicherten Verbindung wird UDP hauptsächlich für Anfragen an Server verwendet, die nur geringe Datenmengen enthalten und unter Umständen verloren gehen dürfen, da sie jederzeit wiederholt werden können. Ein Beispiel hierfür ist die Abfrage von Domainnamen unter Verwendung des Domain Name Service (DNS). Falls UDP als Transportprotokoll eingesetzt wird, muss die Applikationsschicht die Übertragung sichern, andernfalls kann der Sender nie wissen, ob seine Daten tatsächlich empfangen wurden.

Die Vermittlungsschicht enthält das eigentliche Internet-Protokoll. Dieses ist für die Adressierung und die Zustellung der Daten verantwortlich. Zu diesem Zweck besitzt jedes Gerät, das sich im Internet befindet und Daten sendet oder empfängt, eine eigene IP-Adresse (in der Form 192.168.130.0), die es entweder bei der Einwahl von dem Internet Service Provider zugeteilt bekommt oder die dem Gerät fest zugewiesen ist.

Die Sicherungsschicht bildet den Anschluss an das Übertragungsmedium. Hier sind die Protokolle implementiert, die die TCP/IP Daten an das Medium anpassen, z.B. zum Transport über ein Modem oder über ein Ethernet-Netzwerk.

TCP/IP

Das Transmission Control Protocol und das Internet Protocol sind die am häufigsten verwendeten Protokolle im Internet. Sie sorgen für eine fehlerfreie Übertragung von Daten zwischen zwei Kommunikationspartnern. Dabei spielt es keine Rolle, welche Art von Daten transferiert wird, da deren Interpretation erst von der Applikationsschicht vorgenommen wird. Es gibt einige Standard-Applikationsprotokolle, wie etwa HTTP, SMTP oder FTP, die auf den folgenden Seiten kurz beschrieben werden. Es kann aber durchaus sinnvoll sein hier ein eigenes Übertragungsformat zu wählen, beispielsweise wenn es darum geht Geräte zu vernetzen, die schon über ein proprietäres Protokoll verfügen oder wenn zwischen zwei Geräten nur geringe Mengen an Informationen auszutauschen sind.

Der große Vorteil von TCP/IP ist seine Verbreitung. Seit Jahren ist es als Standard zum Datenaustausch zwischen verschiedensten Rechnern in Verwendung, bisher jedoch größtenteils bei leistungsfähigen Maschinen, wie etwa PCs. Für den Einsatz in Embedded Systemen ergibt sich aus den geringen verfügbaren Ressourcen jedoch der Bedarf nach speziellen, auf Mikrocontroller zugeschnittene TCP/IP-Stack-Implementierungen. Diese sollten sich leicht in bestehende Software integrieren lassen, kompatibel zu den Stacks anderer Betriebssysteme sein und mit wenig RAM, ROM und CPU-Belastung, auch auf kleinen Controllern, einen angemessenen Datendurchsatz erzielen.

PPP / Ethernet

Unterhalb von TCP/IP liegt die Sicherungsschicht, die die vorliegenden Daten an das verwendete Übertragungsmedium anpasst. Hauptsächlich werden in der Praxis zwei verschiedene Medien verwendet, entweder ein Netzwerk nach Ethernet-Standard oder eine serielle Anbindung, meistens ein Modem. Je nach Art des gewählten Übertragungsweges werden unterschiedliche Protokolle in der Sicherungsschicht eingesetzt.

Bei serieller Übertragung ist dies das Point to Point Protocol (PPP). Seine Aufgabe besteht in der Kapselung der Daten zur Übertragung, der Steuerung der Verbindung und ihrer Konfiguration. Außerdem bietet es mehrere Möglichkeiten der Authentifizierung gegenüber der Gegenstelle, von der nahezu alle Internet Service Provider Gebrauch machen (z.B. PAP, CHAP).

Bei Verwendung eines Ethernets wird die Datenübertragung im allgemeinen durch einen zusätzlichen Chip durchgeführt, der die TCP/IP-Daten in Ethernet-Pakete umsetzt, die entsprechende Adressierung vornimmt und Kollisionen von Paketen im Netzwerk erkennt. Da im Ethernet eine andere Art der Adressierung als bei TCP/IP verwendet wird (MAC-Adressen), ergibt sich das Problem, dass die Applikation, die den Transfer initiiert, zwar die IP-Adresse ihrer Gegenstelle kennt, nicht aber deren MAC-Adresse. Deshalb werden zwei weitere Protokolle für die Adressauflösung benötigt, eines zur Umsetzung der IP-Adressen in MAC-Adressen und das zweite für die umgekehrte Zuordnung. Diese Protokolle heißen Address Resolution Protocol (ARP) und das Reverse Address Resolution Protocol (RARP).

Applikationsprotokolle

Die Applikationsprotokolle ermöglichen die Nutzung der heutigen Standardanwendungen, wie sie im Internet täglich verwendet werden. So werden E-Mails mit Hilfe von SMTP und POP3 versendet und empfangen, Dateien unter Verwendung von FTP übertragen und Web-Seiten mit dem HTTP-Protokoll abgerufen und übermittelt. Die Applikationsprotokolle benutzen die darunter liegende Transportschicht zur Übertragung von Daten, bei allen nachfolgend aufgeführten Protokollen ist dies TCP.

HTTP

Das Hyper Text Transfer Protocol findet hauptsächlich Anwendung in der Kommunikation von Web-Servern und Internet-Browsern. Letztere können vom Server HTML-formatierte Seiten anfordern um sie dann darzustellen und dem Benutzer Informationen zu liefern oder Eingabemöglichkeiten anzubieten. Hierfür werden so genannte Formulare eingesetzt, die die Eingabe von Text oder die Auswahl aus vorgegebenen Werten ermöglichen. Vom Browser können dann die geänderten Werte zurück an den Server geschickt werden, der dann die Daten auswertet.

Kennzeichnend für den Einsatz von HTTP ist eine Client-Server Struktur, bei der ein Client Daten von einem Server anfordert und ggf. Antworten zurück sendet. Zu beachten ist beim Einsatz von HTTP, dass der Datenfluss vom Client zum Server auf kleine Datenmengen beschränkt bleibt, da diese Daten in einem Block zum Server gesendet werden und dort relativ aufwendig analysiert werden müssen. HTTP bietet sich auf Embedded Systemen, insbesondere in Kombination mit HTML, gut zur Konfiguration dieser Geräte über einen Browser an. Sollen Konfigurationen zwischen Maschinen ausgetauscht werden, bietet sich die Kombination von HTTP und XML (bzw. SOAP) an, die später eingehender erläutert wird.

SMTP / POP3

Die beiden Protokolle Simple Mail Transfer Protocol und Post Office Protocol dienen zur Übermittlung von E-Mails, wobei SMTP für den Versand und POP3 für den Empfang zuständig ist. Neben der üblichen Anwendung, der Mensch-zu-Mensch-Kommunikation, eignet sich E-Mail auch zur Statusmeldung von Geräten, die keine sofortige Reaktion erfordern, wie beispielsweise der Meldung von Beständen in Lagern oder Verkaufsautomaten.

XML / SOAP

Die Extensible Markup Language ist eine textbasierte Auszeichnungssprache und weist auf den ersten Blick viele Ähnlichkeiten mit HTML auf. Es gibt aber zwei wichtige Unterschiede, durch die XML sehr gut für die Maschinenkommunikation einsetzbar ist:

- Bei XML werden der Dateninhalt und seine Darstellung strikt getrennt. Das ermöglicht bei einem Mensch-Maschinen-Schnittstelle eine formatierte Ausgabe des Inhalts, während bei der Kommunikation zwischen zwei Maschinen die Darstellung ignoriert wird.
- XML ist erweiterbar, eigene Tag- und Attributnamen können frei gewählt werden. Hierdurch ist eine Anpassung an das jeweilige System und dessen Anforderungen einfach möglich.

Auf Grund der beiden Punkte haben sich einige XML-basierte Übertragungsstandards gebildet, zu denen auch das oben genannte Simple Object Access Protocol (SOAP) gehört. Bei diesen Standards wird ein Mechanismus definiert, der es allen Rechnern ermöglicht, selbstständig Daten auszutauschen, auch zwischen den unterschiedlichsten Rechnerarchitekturen. SOAP wird von allen großen Softwarefirmen unterstützt, wie etwa Sun, IBM oder Microsoft.

FTP

Das File Transfer Protocol dient zur Übertragung von Dateien über das Internet, das heißt dem Kopieren einer Datei auf ein anderes Gerät. Kennzeichnend für den Einsatz von FTP ist das Vorhandensein eines Dateisystems zur Verwaltung der Dateien sowie die gleichzeitige Verwendung von zwei TCP-Verbindungen (jeweils eine Verbindung zur Steuerung und eine zur Datenübertragung) zwischen den Kommunikationspartnern. Durch diese Charakteristika ist FTP für den Einsatz in kleinen Embedded Geräten nur bedingt zu empfehlen, da oftmals das Filesystem und der zusätzliche Speicherbedarf für die zweite TCP-Verbindung die vorhandenen Ressourcen stark belasten. Sollen dagegen größere Datenmengen übertragen werden, ist FTP die Standardlösung.

Applikationen

Neben den im vorherigen Kapitel erläuterten Applikationsprotokollen ist für viele Anwendungen eine weitere Logik notwendig. So ist es mit der Implementierung der E-Mail-Protokolle SMTP und POP3 zwar möglich, auf die entsprechenden Server zuzugreifen, die Darstellung und Verwaltung der Mails erfordert allerdings eine zusätzliche Applikation. Bei der Kombination aus HTTP-Protokoll und Web-Server gilt ähnliches. Soll mittels eines Browsers der Zugriff auf ein Embedded System erfolgen, so ist im allgemeinen die Verwaltung bzw. die dynamische Erstellung der HTML-Seiten notwendig. Diese Funktionalität ist nicht Teil des HTTP-Protokolls, sondern liegt in der Web-Server-Applikation.

E-Mail

Neben den für die Zugriffe auf SMTP- und POP3-Server notwendigen Protokollen gibt es für E-Mails eine Reihe weiterer internationaler Standards (RFCs), die den Dateninhalt von Mails betreffen. Diese beziehen sich beispielsweise auf das Format des Mail-Headers (der u.a. den Absender, die Sendezeit und die Betreff-Zeile der Mail enthält), auf die Kodierung von Sonderzeichen, wie etwa Umlauten, und die Verwendung verschiedener Sprachen. Die entsprechenden RFCs müssen beim Erstellen und beim Empfang von E-Mails eingehalten werden, damit diese von einem beliebigen Mail-Client fehlerfrei dargestellt werden können. Bei empfangenen Mails stellt sich im Zusammenhang mit der Applikation zusätzlich die Frage nach der Darstellung und in wie weit diese sich an bestehenden Mail-Clients orientieren soll. So ist z.B. zu überlegen, wie eine Liste der gesamten Mails angezeigt werden soll, wodurch der Benutzer gelesene und ungelesene Mails unterscheiden kann, ob die Mails als Kopie vom Mailserver geladen werden sollen und ob das Löschen von Mails, auch auf dem Server, unterstützt wird. Bei Überlegungen zum Einsatz von E-Mails ist natürlich auch nach den eigenen Zielen zu unterscheiden. So stellt ein Mail-Client zur Benachrichtigung über Betriebszustände eines Embedded Systems geringere Anforderungen an die Komplexität der Software als die Erstellung einer frei editierbaren Mail durch einen Benutzer. Zieht man zusätzlich den Bedarf an Ressourcen in Betracht, so stellt man fest, dass eine Universallösung für den Einsatz von E-Mails nicht existiert. Durch eine sinnvolle Auswahl der Anforderungen und der umgesetzten RFCs kann man allerdings eine Applikation entwickeln, die einen Großteil der Aufgaben ohne weitere Anpassungen erfüllt.

Bei der Entscheidungsfindung über den Umfang der benötigten Funktionalität können folgende Punkte helfen:

- Ist das Senden und/oder Empfangen von E-Mails erforderlich?
- Wofür soll die E-Mail eingesetzt werden (z.B. Benachrichtigung oder für durch den Benutzer einzugebenden freien Text)?
- Welche Sprachen müssen unterstützt werden?
- Sollen neben dem Mailtext auch binäre Daten (z.B. als Anhang) eingesetzt werden?
- Wie viel RAM und ROM stehen für die Funktionalität zur Verfügung?
- Welche Anforderungen stellt die restliche Software an das System?

Auf Basis der Antworten lässt sich ein Profil erstellen, anhand dessen man beurteilen kann, wie komplex die Lösung sein wird und ob es geeignete Software-Module bereits auf dem Markt gibt, bzw. mit welchem Aufwand man für Änderungen bzw. Erweiterungen zu rechnen hat.

Embedded Web-Server

Ein Web-Server ermöglicht es einem Client sich unter Verwendung des HTTP-Protokolls mit dem Embedded System zu verbinden. Welches Datenübertragungsformat dabei eingesetzt wird, hängt entscheidend von der zu erledigenden Aufgabe ab. Soll eine grafische Ausgabe auf einem Internet-Browser erfolgen, so ist dieses Format meistens HTML. Unter dessen Verwendung lassen sich Gerätezustände und Daten auf nahezu jedem Rechner darstellen, der mit dem Internet verbunden und auf dem ein Browser installiert ist. Zu beachten ist hierbei, dass der Zugriff auf das Gerät nur autorisierten Personen gestattet wird.

Als neues standardisiertes Datenübertragungsformat wird zunehmend XML eingesetzt. Die ersten Generationen der Browser, die XML neben HTML unterstützen, sind verfügbar. Es gibt allerdings noch Unterschiede in der Behandlung, so dass für grafische Ausgaben XML noch nicht so gut geeignet ist. XML bietet aber, wie oben bereits erwähnt, große Vorteile in der Kommunikation zwischen zwei Maschinen.

Embedded Client für Web Services

Webservices sind spezielle Dienstleistungen eines Servers im Internet: Diese sind nicht auf die Bereitstellung von Internetseiten (HTTP) oder Dateien (FTP) beschränkt, sondern können ganz spezielle an Kundenanforderungen angepasst sein. Beispiel: Ein Server bietet einen Wetter-Service an, der es Clients ermöglicht, sich mit ihm zu verbinden, zu authentifizieren und gezielte Wetterdaten abzufragen.

Der Embedded Client unterstützt die Funktionen des Web Services, indem er das standardisierte Austauschformat XML beherrscht, Anfragen an den Server artikuliert und Antworten interpretiert (sog. Parser). Die empfangenen – durch XML wohl strukturierte Daten – werden grafisch aufbereitet und der Anwendung bzw. dem Kunden zur Verfügung gestellt.

Anwendungen

Konfiguration von einzelnen Geräten

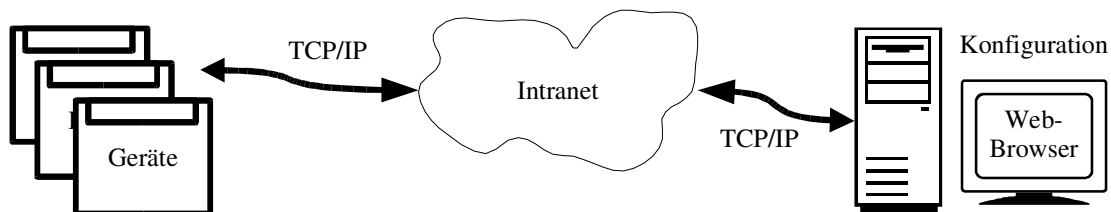
Maschinen und Geräte lassen sich einfach über ein Webinterface konfigurieren. Die Konfiguration über einen "Embedded Webserver" eignet sich besonders für einzelne Geräte. Im Browser stellt der Server ein Dialog (Formular) dar, in dem aktuelle Daten des Gerätes angezeigt werden und mit Hilfe von Edit-Feldern und Select-Boxen eingestellt werden können.

Vorteile

- Konfiguration kann mit einem Standard-Browser (Internet Explorer, Netscape, Opera etc.) von jedem PC vorgenommen werden.
- Es wird keine Installation von Software benötigt.
- Es bestehen keine Abhängigkeiten vom Betriebssystem
- Eine Konfigurationsanwendung mit HTML-Seiten kann mit geringen Aufwand erstellt werden.
- Die Hardwareanforderungen sind beim Einsatz der sevenstax-Module gering .

Nachteile

- Es wird eine Verbindung zum Gerät benötigt, das heißt:
 1. es existiert eine permanente Verbindung (z.B. Ethernet) oder
 2. der PC mit dem Browser baut die Verbindung auf oder
 3. es wird ein Mechanismus zur Einwahl auf Anforderung integriert
- Webseiten sind für eine Realtime-Kommunikation ungeeignet. Mit einer Erweiterung der HTML Seiten um Java-Applets lässt sich der Nachteil aber gut kompensieren.



Realisierung

Die Firmware der Geräte wird um die Basis-Internet-Protokolle und einen kleinen Webserver erweitert. Er bildet in Form von hierarchisch strukturierten HTML-Seiten die aktuellen Informationen (Status, Einstellungen, Fehler) ab. Diese Webseiten können vom Anwender oder Administrator (auch passwortgeschützt) abgerufen werden. Darüber hinaus beherrscht der Webserver sog. Formulare. Hierbei werden in den Webseiten Eingabemasken abgelegt über die der Benutzer die Geräteeinstellungen manipulieren kann.

Anwendungsbeispiele

Drucker, Kopierer, Heizungsanlagen, TK-Anlagen, Home Automation

Echtzeit Kommunikation mit Geräten

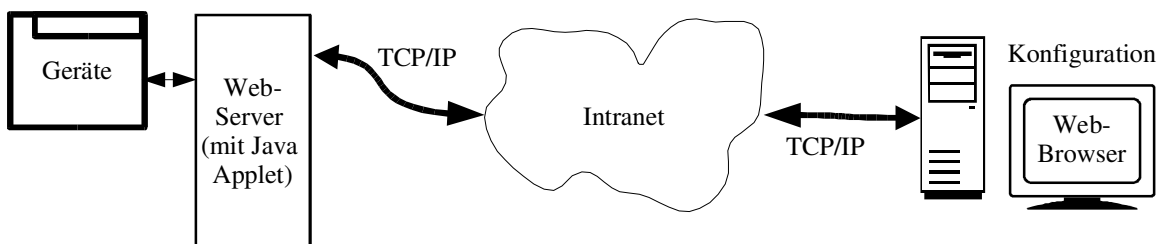
Für die Konfiguration von einzelnen Maschinen und Geräten eignet sich besonders ein Webinterface. Die Realtime-Anzeige von Daten oder die direkte Steuerung der Geräte sind mit HTML-Seiten nicht zu realisieren. Durch die Integration von Java Applets den Webseiten stehen diese Funktionen auch im Browser zur Verfügung. Die Applets können auch für die Auswertung der Daten genutzt werden. Da die Applets auf dem Computer und nicht auf dem Gerät gestartet werden, steht auch die Rechenleistung der Computers zur Verfügung, die i.a. deutlich höher ist als die des Gerätes.

Vorteile

- Echtzeit-Darstellung der Gerätedaten
- Auswertung der Daten kann am PC mit deren Rechenleistung erfolgen
- Größere Flexibilität und Interaktion mit dem Anwender möglich
- Lauffähig an verschiedenen Clients (z.B. PC, PDA, Mobiltelefon)
- Keine Installation am Client notwendig
- Hohe Sicherheit der Applets

Nachteile

- Die Java Virtual Machine muss beim Browser vorhanden sein
- Die Java-Applets werden beim Aufruf erst über das Netz geladen.
- Die Applet-Entwicklung ist aufwendiger als das erstellen von Webseiten



Realisierung

Die in Java implementierten Applets werden als Link in die Templates der Web-Seiten integriert. Die Applets selbst werden auf dem Embedded Gerät genauso behandelt wie Webseiten (siehe auch Webserver). Beim Aufruf der HTML-Seiten über einen Browser wird das Applet vom Browser erkannt und geladen. Das gestartete Applet stellt (rückwärts) eine Anfrage an die IP-Adresse des Webserver. Ein Applet-Server beantwortet die Anfrage des Applets und sendet die notwendigen Daten vom Gerät zum Applet. Für den Applet-Server und für die Applets bietet sevenstax vorgefertigte Module.

Anwendungsbeispiele

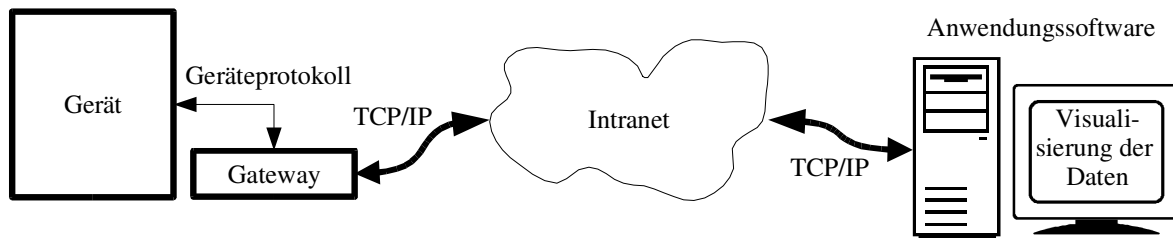
Windkraftanlagen, Überwachungsanlagen, Motorsteuerungen, medizinische Geräte

Anwendungsspezifischen Kommunikation über IP-Netze

In Geräte und Maschinen mit vorhandenen anwendungsspezifischen Protokollen zwischen PC-Applikation und Gerät kann ohne Wechsel der Protokolle die Internetfähigkeit integriert werden. Beim Internetprotokoll werden die Nutzdaten als einzelne Pakete übertragen. Die Daten selbst werden nicht ausgewertet, dies ist Aufgabe der Applikation. So können die Daten auch ein anwendungsspezifisches Protokoll enthalten. Auf diese Weise ist es möglich die vorhandene Kommunikation durch das Internet zu „tunneln“. Dabei sind einige Besonderheiten des Internets zu beachten:

- Die Laufzeit vom Sender zum Empfänger ist unbestimmt und von der Anzahl der beteiligten Netzknoten und von der Netzlast abhängig.
- Bei der Übertragung können theoretisch Datenpakete verloren gehen, die dann erneut gesendet werden müssen.
- Die Reihenfolge der empfangenen Pakete muss nicht mit der Sendereihenfolge übereinstimmen.

Die Geräte und Maschinen werden um die Funktion eines internen oder externen Internet-Gateways erweitert. Auf Seiten des PCs ist die Erweiterung der Auswertungssoftware notwendig.



Vorteile

- Die vorhandenen Software-Entwicklungen (PC-Applikation und Embedded Software) können weiter genutzt werden.

Nachteile

- Es wird immer eine anwendungsspezifische Applikation benötigt.
- Die Verfahren ist nicht für alle anwendungsspezifischen Protokolle geeignet.
- Die zu verwendende Internetinfrastruktur (z.B. Firewalls und Gateways) muss in die Planung mit einbezogen werden, da diese die Datenpakete durch- bzw. weiterleiten können müssen.

Protokolldaten und Fehlermeldungen verteilter Geräte

Für verteilte Geräte und Maschinen bietet sich als kostengünstige und zuverlässige Protokollieremethode der automatische und regelmäßige Versand von vorbereiteten E-Mails an. In Form eines Email-Moduls in das Gerät eingebracht, kann dieses über einen längeren Zeitraum die Daten sammeln, in eine E-Mail verpacken und diese Informationen regelmäßig versenden.

Der Empfänger kann sowohl ein Mensch als auch ein Server sein. Das regelmäßige Eintreffen der Mails erstellt im Eingangsfach ein sehr einfach zu verwaltendes Protokoll. Die Betreffzeile der Mail kann darüber hinaus Warnungen oder Zusatzinformationen bieten.

Vorteile

- geringer administrativer Aufwand, vorhandene Software
- E-Mail-Server und Gateways zur Sprachausgabe und SMS sind allgemein Verfügbar
- geringen Verbindungskosten

Nachteil

- keine gesicherte Übertragung, theoretisch können E-Mails verloren gehen
- es fehlen Verwaltungsstrukturen, für eine große Anzahl von Geräten ist die Methode zu unstrukturiert

Wartung und Diagnose von Geräte- und Maschinenpools

Verkaufsautomaten werden dort aufgestellt, wo viele Menschen sie schnell und einfach erreichen können.

Die regelmäßige Wartung von von Geräte- und Maschinen-Pools an verteilten Standorten bedeuten einen erheblichen personellen Aufwand. Treten Probleme auf, so entstehen weitere kostenintensive Aufwände. Zur Vermeidung bzw. zur Minimierung des Aufwands bietet sich die Fernwartung der Geräte über das Internet an.

Das Gerät nimmt über eine telefonische Einwahl in das Internet Verbindung zu einem Wartungsserver auf, um Fehler zu melden oder Wartungsdaten zusenden. Der Wartungsserver bekommt zusätzlich einen bedingten Zugriff auf das Gerät, um Statusdaten zurücksetzen zu können. Der Kunde kann sich über seinen eigenen Rechner mit dem Wartungsserver verbinden und sich die von dem Gerät empfangenen Daten auf einem Web-Browser anzeigen lassen.

Dieses Szenario hat zwei Ziele: Optimierung des Wartungsaufwands, da Servicepersonal nur im Bedarfsfall vor Ort sein muss und die Verringerung von Ausfallzeiten, da die Maschine eingetretene oder sich abzeichnende Fehler eigenständig melden und so für eine schnelle Beseitigung oder gar Vermeidung des Ausfalls sorgen kann.

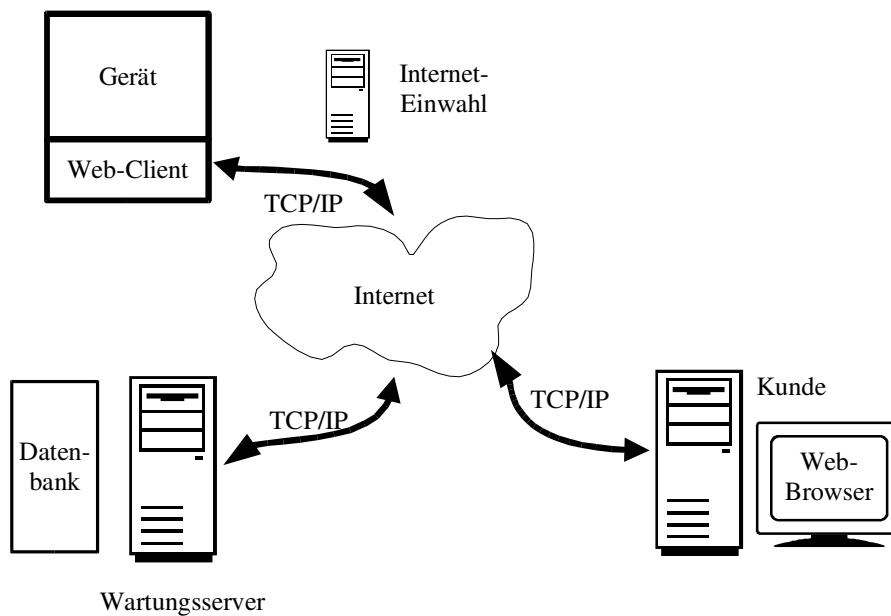
Vorteile

- Einfache Verwaltung viele Geräte
- Strukturierte Verwaltung nach Gerätegruppen und -untergruppen
- Rechtevergabe auf Nutzer und Nutzergruppen möglich
- Zusatzinformationen durch Statistiken und Datenanalyse

- Integration in die vorhandene IT-Infrastruktur
- Alle Zugriffe sind protokollierbar

Nachteile

- Es ist immer ein Server erforderlich
- Die Initialkosten zum Aufbau der Infrastruktur sind höher als bei den anderen Lösungen



Realisierung

In die Geräte und Maschinen wird ein XML/SOAP-Modul integriert. Die Integration kann in die vorhandene Hardware oder durch ein zusätzliches Modul erfolgen. Das Modul enthält die notwendige Hardware und Software um die Geräteinformationen abzurufen. Meist werden diese standardmäßig zu Wartungszwecken für die Techniker angeboten. Darüber hinaus ist das Modul in der Lage, sich nach Bedarf - gegebenenfalls sofort - mit dem Internet zu verbinden, und in einer kostengünstigen und kurzen Verbindung relevante Daten an den Wartungsserver zu übertragen.

Da in diesem Fall eine Kommunikation zwischen zwei Maschinen stattfindet, wird die Kommunikation über XML und SOAP durchgeführt. Die Vorteile der XML-Übertragung sind:

- Maschinenlesbarkeit der Daten in beiden Kommunikationsrichtungen
- Verwendung von Standard-Servern zur Vermeidung von Eigenentwicklungen
- XML-Parser für PCs und andere Rechnerarchitekturen sind verfügbar, woraus eine einfache Anbindung resultiert.
- Vermeidung proprietärer Lösungen des Datenaustauschformats durch die Verwendung von Standards



sevenstax Protokoll-Implementationen

Die **sevenstax**-Protokoll-Stacks sind für Embedded Systeme mit 8...32-Bit-Mikrocontrollern ausgelegt, wobei besonderer Wert auf die leichte Integrierbarkeit in schon existierende Software gelegt wurde. Zur Gewährleistung des störungsfreien Betriebs gibt es keine blockierenden Zustände in den Protokoll-Implementationen. Die Programmierer-Schnittstelle (API) ist sehr schmal gehalten, unterstützt aber alle zur Kommunikation notwendigen Aktionen.

Rückmeldungen der Protokolle erfolgen grundsätzlich über Callback-Routinen, an die aufgetretene Events von den Stack-Modulen gemeldet werden. Diese Routinen integrieren die Protokolle in die Software, ohne dass an dieser selbst größere Änderungen notwendig sind. Selbstverständlich bietet eine schmale API auch Vorteile bei der Erstellung neuer Software mit Internetanbindung, da Mehrdeutigkeiten in der Benutzung vermieden werden.

Der Ressourcenbedarf der **sevenstax**-Software ist gering, sowohl in Bezug auf den Bedarf an RAM und ROM, als auch im Hinblick auf die Belastung der CPU. Dadurch wird oftmals die Erweiterung bestehender Software möglich, ohne die Hardware-Plattform ändern zu müssen. Die Protokolle fügen sich sowohl in Systeme mit als auch ohne Betriebssystem nahtlos ein.

sevenstaxTCP/IP

Einsatzgebiet

Der *sevenstaxTCP-Stack* ist für Geräte mit Datenschnittstellen, die kleinere Datenmengen (in der Größenordnung bis einige hundert Kilobyte) aufweisen, konzipiert, beispielsweise zum Herunterladen von Parametern, Firmware, Web-Seiten, oder zur Konfiguration über das Internet, sowie für Fernwartung und -überwachung.

Vorteile

- Schnelle Integration und geringe Entwicklungszeiten. Es werden nur die für den Betrieb notwendigen Funktionen integriert und getestet.
- Hohe Betriebssicherheit durch definierte Systemzustände und kurze Tests
- Hohe Sicherheit durch klar definierte Funktionssets.
- Geeignet für den Einsatz in Mikrocontrollern von 8...32 Bit.
- Asynchrones Verhalten auch ohne multithreading-fähiges OS, d.h. blockiert keine andere Software.
- Getestet unter CMX RTOS von CMX Systems und embOS von SEGGER Microcontroller Systeme
- Benötigt keine eigenen Sende-/Empfangspuffer, da die Puffer der unteren Netzwerkprotokoll-Schicht verwendet werden können.
- Leichte Integration in vorhandene Software, da nur wenige Schnittstellen existieren.
- Portabilität durch Programmierung in ANSI C
- Optionale Funktionserweiterungen vorhanden (auf Kosten der Codegröße):
- Unterstützung von UDP und ICMP (Ping)
- TCP-Server-Funktionalität
- Konfigurierbare Anzahl paralleler Verbindungen
- In Deutschland, den USA, Asien und Südamerika erfolgreich eingesetzt

Speicherbedarf, Systemressourcen

- ROM: **2,6...8.2 kByte** Code, abhängig von den gewählten Optionen (Standard 16-Bit CPU)
- RAM: **95...250 Bytes** für Variablen.
- 32-Bit-Timer/Counter mit 1 ms Auflösung
- Typische CPU-Gesamtlast < 5% (16-Bit-Controller, 10 MHz, während einer Modem-Datenübertragung mit 14.400 Bits/s)

sevenstaxPPP

Einsatzgebiet

Das *sevenstaxPPP-Stack* ermöglicht die Übertragung von TCP/IP-Daten über ein serielles Kabel, Modem, DECT oder ISDN. Besonders häufig wird PPP bei Geräten eingesetzt, die als Client im Netzwerk auftreten, wie z.B. bei Web-Browsern oder Geräten, die selbstständig Daten anfordern oder senden.

Vorteile

- Geringe Gerätekosten, da wenig RAM und ROM benötigt wird und der Prozessor nur geringe Rechenleistung zur Verfügung stellen muss.
- Die Sende- und Empfangspuffer können den nächsten Protokollschichten zur Weiterverarbeitung der Daten zur Verfügung gestellt werden.
- Schnelle Integration, geringe Entwicklungszeiten und damit geringe Entwicklungskosten, da nur die für den Betrieb notwendigen Funktionen integriert und getestet werden müssen. Der Stack weist nur wenige Schnittstellen auf und lässt sich gut in vorhandene Software integrieren.
- Hohe Betriebssicherheit, da die Systemzustände sehr gut erfassbar und damit testbar sind.
- Optimiert für Verbindung zu Internet Service Providern (Einwahlverbindung zu ISPs).
- Leichte Integration zwischen Modem-Treiber und TCP/IP-Stack
- Läuft auf allen üblichen Mikrocontrollern (ab 8 Bit aufwärts)
- Software geschrieben in ANSI C
- Integrierte Sub-Protokolle LCP, PAP und IPCP.
- 'On the fly' HDLC encoding/decoding.
- Asynchrones Verhalten auch ohne multithreading-fähiges OS. Blockiert keine andere Software.
- Getestet unter CMX RTOS von CMX Systems und embOS von SEGGER Microcontroller Systeme
- Bereits in Deutschland, den USA, in Asien und Südamerika erfolgreich eingesetzt

Optionale Funktionen (auf Kosten der Codegröße)

- Authentifizierung mit CHAP
- Unterstützung von ACCM (Asynchronous Control Character Map)
- Header-Kompression
- Protokoll-Kompression

Speicherbedarf, Systemressourcen

- ROM: **3,5...6,0 kByte** Code, abhängig von gewählten Optionen (16-Bit CPU)
- RAM: **160...280 Bytes** für Variablen, Typ. 2x 600 Bytes für Empfangs- und Sendepuffer (einstellbar)
- 32-Bit-Timer/Counter mit 1 ms Auflösung
- Typische CPU-Gesamtlast < 5% (16-Bit-Controller, 10 MHz, während einer Modem-Datenübertragung mit 14.400 Bits/s)

sevenstaxXML-Parser/Generator

Einsatzgebiet

Das *sevenstaxXML-Modul* (Parser+Generator) ermöglicht den Einsatz von XML-basierten Sprachen (z.B. WML, SOAP) in Embedded Systemen. Es besteht aus einem Parser, der eintreffende XML-Daten auswertet, und einem Generator, der XML-Code zum Versenden zur Verfügung stellt. Bei der Implementierung wurde insbesondere auf die einfache Integrierbarkeit in bestehende Software und einen sparsamen Umgang mit dem RAM geachtet.

Vorteile

- Anpassung an beliebige XML-basierte Sprachen durch die freie Definition mehrerer Namensräume (Tag-Namen, Attribut-Namen etc.)
- Streaming Parser, d.h. Auswertung des XML-Codes schon während des Empfangs der Daten (geringer Bedarf an RAM, da nur ein sehr kleiner Puffer zur Analyse bereitgestellt werden muss)
- Unterstützung aller XML-Entitäten
- Syntaxcheck bis zu einer vordefinierten Verschachtelungstiefe
- Integration des Parsers in bestehende Software über Registrierung von Handler- (Callback-) Funktionen, die bei bestimmten Events (z.B. nach Erkennung eines XML-Tags oder nach Fehlererkennung) aufgerufen werden
- Vorgabe einer zu sendenden XML-Einheit an den XML-Generator in vordefinierter Struktur
- Zurückliefern des resultierenden XML-Codes in beliebig großen Paketen zum Versand

Zur Analyse benötigt der XML-Parser die unterstützten Tag- und Attributnamen. Da XML mit so genannten Namensräumen arbeitet, in denen u.a. die zugehörigen Tag- und Attribut-Namen definiert sind, können solche Namensräume definiert und dem XML-Modul bekannt gemacht werden. Jeder Tag-Name ist eindeutig bestimmt durch den Index des zugehörigen Namensraums und den Index der Tagliste dieses Namensraums. Bei Attribut-Namen verhält es sich analog.

Speicherbedarf, Systemressourcen

- ROM: **11,0 KByte** (Standard 16-Bit CPU)
- RAM: **350 Bytes**

sevenstaxSOAP-Client

Einsatzgebiet

Mit Hilfe des *sevenstaxSOAP-Clients* kann ein Datenaustausch mit einem existierenden SOAP-Server leicht auf einem Embedded System realisiert werden. Die Lösung basiert auf dem oben beschriebenen XML-Modul, dem die SOAP-spezifischen Namensräume (Tags, Attribute etc.) sowie die äußere Hülle der Anfrage-/Antwort-Formate (XML) bekannt gemacht werden. Darüber hinaus ist die SOAP-Fehlerbehandlung implementiert.

Neben dem XML-Modul benutzt der SOAP-Client den HTTP-Generator/-Parser, da die XML-Anfragen/-Antworten i.a. in einem HTTP-Header eingebettet sind. Für den Einsatz von SOAP via SMTP könnte die SOAP-Client-Software auf das SMTP-Modul zurückgreifen.

Eine Anwendung, die den *sevenstaxSOAP-Client* benutzt, muss nur noch die notwendigen SOAP-Parameter sowie den applikationsspezifischen Teil der XML-Anfragen und entsprechende Handlerrountinen für die XML-Antwort definieren. Die Anwendung muss keine Kenntnisse über die SOAP-Mechanismen besitzen.

Vorteile

- Verwendung für alle existierenden Web-Services, die mit SOAP-Technologie arbeiten
- Integration der Client-Software in bestehende Software über Registrierung der SOAP-Parameter und der applikationsspezifischen (nicht SOAP-spezifischen) XML-Eigenschaften (Namensräume, Struktur der Anfragen, Handle-Funktionen für Antworten etc.)
- Modularer Aufbau durch Benutzung separater XML-, HTTP- und Internet-Service-(TCP/IP, PPP) Modulen

Speicherbedarf, Systemressourcen

- ROM: **14 kByte Code** (Standard 16-Bit CPU)
- RAM: **700 Byte**

sevenstaxWeb-Server

Einsatzgebiet

Der *sevenstaxWeb-Server* ermöglicht den Zugriff auf ein Embedded System von jedem Gerät, das einen Web-Browser unterstützt.

Besonderer Wert wurde auf den sparsamen Umgang mit Mikrocontroller-Ressourcen (RAM, ROM) gelegt. Die dadurch entstehenden Funktionseinschränkungen gegenüber einem klassischen Web-Server für PCs oder UNIX-Server haben jedoch keinen Einfluss auf Webserver-Standardanwendungen, wie beispielsweise die Abfrage von HTML-Formst. Als Grundlagen der HTTP-Kommunikation unterstützt der Server die GET- und POST-Methoden. Der Server ist in der Lage, Web-Seiten unter Verwendung spezieller HTML-Tags dynamisch zu erzeugen, beispielsweise zur Generierung von HTML-Formst. Hierfür liest der Web-Server eine Vorlage (Template) aus dem ROM und gibt einer Applikation die Möglichkeit zum Einfügen von Parametern in die Web-Seite. Sollte dies nicht gewünscht sein, werden z.B. die Gefault-Werte aus dem Template übernommen. Typische Beispiele für den Einsatz sind die Fernkonfiguration von Geräten und die Präsentation von Messdaten.

Statt über HTML kann der Web-Server auch Daten austauschen, die in XML-Seiten enthalten sind, z.B. über das SOAP-Protokoll. Dies bietet sich für die Kommunikation mit Maschinen an, da diese Daten nicht dargestellt werden müssen.

Vorteile

- Läuft auf allen üblichen Mikrocontrollern (ab 8 Bit aufwärts)
- Unterstützung mehrerer paralleler Verbindungen
- Dynamische Seitenerzeugung (HTML/XML-Formst, HTML/XML-Seiten)
- Authentifizierung (Basic HTTP Authentication) verhindert ungewollten Zugriff auf das Gerät
- Eigener Sendepuffer (abschaltbar - falls der TCP/IP-Stack über einen eigenen Puffer verfügt, kann dieser mit verwendet werden)
- Alle Ressourcen (Seiten etc.) können im Speicher (ROM/Flash) liegen
- Unabhängig vom Einsatz eines Betriebssystems
- Benötigt kein Filesystem
- Zur Reduzierung des Speicherbedarfs nicht auf Socket-Interface angewiesen (Sockets für die Anbindung fremder Software optional einsetzbar)
- Getestet mit den gängigen Browsern (Mozilla 1.0, Netscape 4.7, IE 5.x, IE 6, StarOffice 5, Opera 5.x, Opera 6.x)
- Software geschrieben in ANSI C

Speicherbedarf, Systemressourcen

- ROM: **1.4 ... 4.5 kByte Code**, abhängig von den gewählten Optionen
465...690 Bytes für Konstanten
- RAM: **160...280 Bytes**
- TCP/IP-Stack zur Anbindung des Internets erforderlich

sevenstaxMail-Client

Einsatzgebiet

Der *sevenstaxMail-Client* stellt Funktionen zum Empfang und/oder zum Versenden von Mails auf Embedded Systemen zur Verfügung. Die Verwaltung der empfangenen bzw. versendeten Mails ist abhängig von der jeweiligen Applikation und somit nicht Bestandteil des Mail-Clients.

Die Client-Software verwendet mehrere andere Sub-Module:

- den POP3-Generator, der abhängig von der durchzuführenden Aktion, eine Sequenz von POP3-Befehlen für das Abrufen von Mails liefert
- den POP3-Parser, der die jeweiligen POP3-Antwort-Codes auswertet
- den SMTP-Generator, der eine Sequenz von SMTP-Befehlen zum Senden von Mails liefert
- den SMTP-Parser, der die jeweiligen SMTP-Antwort-Codes auswertet
- dem IMF/MIME-Generator, der eine Mail nach dem Internet Message Format und der MIME-Kodierung zusammensetzt
- dem IMF/MIME-Parser, der eine Mail nach dem Internet Message Format und der MIME-Kodierung auswertet und in ihre Bestandteile zerlegt

Vorteile

Leichte Erstellung von Embedded-Mail-Anwendungen ohne Kenntnisse der zugrunde liegenden Protokolle

- Versionen nur für den Empfang oder nur für den Versand oder für beides leicht konfigurierbar
- Streaming Parser, d.h. bei allen verwendeten Parsern Auswertung der Antworten schon während des Empfangs der Daten (geringer Bedarf an RAM, da nur ein sehr kleiner Puffer zur Analyse bereitgestellt werden muss)
- Integration des Clients in bestehende Software über Registrierung von Handler-Funktionen (für den Empfang), die bei bestimmten Events (z.B. nach Erkennung des Absenders einer Mail) aufgerufen werden bzw. durch Übergeben der Mail-Attribute (für den Versand)
- Modularer Aufbau durch Benutzung separater POP3-, SMTP-, IMF/MIME- und Internet-Service-(TCP/IP, PPP) Modulen

Speicherbedarf, Systemressourcen

- ROM: **23,0 KByte** (Standard 16-Bit CPU)
- RAM: **2,0 KByte**



sevenstax – das Unternehmen

Die Firma

Die Entwicklungen der **sevenstax GmbH** sind auf Systeme mit hohen Anforderungen an die Betriebssicherheit und geringen Hardwarekosten spezialisiert. **sevenstax** verbindet die Kommunikationstechnik, insbesondere das Internet, mit beliebigen Geräten und Maschinen. So schaffen wir mit unseren Kunden ganz neue Anwendungen - die Fernwartung von Geräten, Gerätediagnose oder Telemetrie-Anwendungen über das Internet.

Sie haben die Geräte, die mit Hilfe des Internets neue Funktionen bieten. **sevenstax** hat das Know-How für den Datentransfer. **sevenstax** hilft Ihnen, Ihre Projekte effizient und qualitätsgerecht durchzuführen. Unser Maßstab ist Ihr Erfolg. Das **sevenstax**-Team hat das Know-How, serienreife Produkte für den Markt zu entwickeln. Qualitätsmanagement in der Entwicklung ist ein Schwerpunkt von **sevenstax**.

Das Team

Erfahrung, Kreativität und Vielseitigkeit sind unsere Basis. Das Team von **sevenstax** arbeitet bereits seit mehr als 5 Jahren zusammen. Gemeinsam blicken die Entwickler auf mehr als 40 Jahre Berufserfahrung in Entwicklungsprojekten namhafter internationaler Unternehmen zurück.

Zeit genaue, den Anforderungen entsprechende, wirtschaftliche Projekte wurden von den **sevenstax**-Entwicklern in den letzten Jahren realisiert. Marktreife Produktentwicklungen entstanden unter der Leitung von **sevenstax**-Mitarbeitern ebenso wie Entwicklungen im Rahmen von Dienstleistungen.

Als Gruppenleiter, Projektmanager, Qualitätsmanager, Software- und Hardware-Designer, MMI-Designer und Tester sammelte die **sevenstax**-Crew ihre Erfahrung.

Das Qualitätsmanagement und die damit verbundene Kundenzufriedenheit haben in unserer Firma einen besonderen Stellenwert und sind unser oberstes Ziel. Unzufriedene Kunden durch fehlerhafte Produkte, überzogene Budgets und nicht eingehaltene Termine sind in dieser Beziehung genauso kontraproduktiv wie überlastete Entwickler nach 80 Stunden-Wochen und Wochenendeinsätzen. Suchen Sie einen Dienstleister mit Termintreue? Dann sind Sie bei **sevenstax** an der richtigen Adresse.

Unsere Leistungen

Angeboten werden alle Entwicklungsschritte von der Konzeptionierung bis zur Produktionsvorbereitung. Wenn Sie es möchten, übernehmen wir komplette Entwicklungsmodule oder arbeiten mit Ihnen im "Joint Team". Die Entwicklungsdienstleistungen rechnen wir sowohl nach Festpreis als auch nach Aufwand ab. Sie bekommen über jeden Entwicklungsschritt die volle Kontrolle.

Kontakt

sevenstax GmbH
Hamburger Allee 43
30161 Hannover
Germany

Tel. +49(0)511/ 47386 04
Fax +49(0)721/ 151 505 622
Internet: www.sevenstax.de
E-Mail: info@sevenstax.de

Glossar

Begriff	Bedeutung
ARP, RARP	Address Resolution Protocol und Reverse Address Resolution Protocol: Zuordnung der im Ethernet zur Adressierung verwendeten MAC-Adressen zu der korrespondierenden IP-Adressen und umgekehrt.
CHAP	Challenge Handshake Authentication Protocol: Protokoll zur verschlüsselten Authentifizierung eines Benutzers gegenüber dem angewählten ISP. CHAP ist ein Teil des PPP-Mechanismus.
DECT	Digital Enhanced Cordless Telecommunication: Drahtloser Kommunikationsstandard mit geringer Reichweite. Neben der Sprachübertragung (wie sie beispielsweise in vielen schnurlosen Telefonen verwendet wird) können mit DECT auch Daten übertragen werden.
DNS	Domain Name Service: DNS ordnet einen Domain-Namen (z.B. www.google.de) der zur Adressierung im Internet benötigten IP-Adresse (hier 216.239.33.100) zu.
FTP	File Transfer Protocol: Protokoll zur Übertragung von Dateien über das Internet.
GPRS	General Packet Radio Service: Weiterentwicklung der Mobiltelefonie-Standards zur packetorientierten Übertragung von Daten. Vorteil ist die bessere Nutzung der Bandbreite und eine maximale Übertragungsrate von 115KBit/s.
HDLC	High-Level Data Link Control: Bitorientiertes, synchrones Protokoll für den Einsatz in der Datenverbindungsschicht des OSI-Referenzmodells. HDLC ist von der ISO genormt.
HTTP	Hyper Text Transfer Protocol: Protokoll zur Übertragung von Web-Dokumenten
IPCP	IP Configuration Protocol: Sub-Protokoll von PPP zur Konfiguration des Internet Protokolls
LCP	Link Control Protocol: Sub-Protokoll von PPP zur Konfiguration der Verbindung
PAP	Password Authentication Protocol: Protokoll zur unverschlüsselten Authentifizierung eines Benutzers gegenüber dem angewählten ISP. PAP ist ein Teil des PPP-Mechanismus.
POP3	Post Office Protocol Version 3: Standard-Protokoll zum Empfangen von E-Mails.

Begriff	Bedeutung
PPP	Point To Point Protocol: Protokoll zur Übertragung von Daten über eine serielle Verbindung oder ein Modem. Bei der Verwendung von TCP/IP zur Datenübertragung werden die TCP/IP-Pakete in PPP-Frames verpackt, mit einer Checksumme versehen und dann über die Leitung verschickt. Auf der Empfangsseite wird das TCP/IP-Paket nach Überprüfung der Checksumme aus dem PPP-Frame extrahiert und an den TCP/IP-Stack weitergeleitet.
RFC	Request For Comment: Norm für einzelne Themen im Internet. Es gibt beispielsweise ein eigenes RFC für jedes verwendete Protokoll, wie SMTP oder HTTP oder den Aufbau und die Kodierung von E-Mails.
SMTP	Simple Mail Transfer Protocol: Standard-Protokoll zum Versenden von E-Mails.
SOAP	Simple Object Access Protocol: SOAP stellt einen einfachen Mechanismus zum Austausch von strukturierter und typisierter Information zwischen kommunizierenden Rechnern in einer verteilten Umgebung zur Verfügung. Grundlage von SOAP ist XML. Der Transport der Informationen erfolgt unter Verwendung von HTTP.
TCP/IP	Transmission Control Protocol / Internet Protocol. Diese beiden Protokolle stellen die Übertragung von Daten über das Internet sicher. Das TCP-Protokoll ist dabei zuständig für die Gewährleistung des eigentlichen Transports, d.h. für die Sicherstellung, dass alle Daten in der gleichen Reihenfolge und mit dem gleichem Inhalt beim Empfänger ankommen, mit dem sie auch verschickt wurden. Das Internet Protokoll sorgt in dieser Kombination für die Adressierung von Sender und Empfänger.
UDP	User Datagram Protocol: Protokoll zur ungesicherten Übertragung von Daten über das Internet. Die Applikation, die UDP verwendet, muss dafür Sorge tragen, dass alle gesendeten Daten unter Einhaltung der korrekten Reihenfolge empfangen wurden. Der Vorteil von UDP gegenüber TCP liegt in der höheren Geschwindigkeit. Müssen Daten aber zuverlässig beim Kommunikationspartner ankommen, ist TCP die bessere Wahl.
UMTS	Universal Mobile Telecommunication System: System zur Bereitstellung von Mobilfunk-Diensten der 3. Generation.
XML	Extensible Markup Language: Meta-Seitenbeschreibungssprache zur Beschreibung von strukturierten Daten. Der große Vorteil liegt in den präziseren Deklarationen des Inhalts und aussage kräftigeren Suchergebnissen über verschiedene Plattformen hinweg. Darüber hinaus ist XML die Grundlage für eine neue Generation von webbasierten Anwendungen zum Anzeigen und Bearbeiten von Daten.

