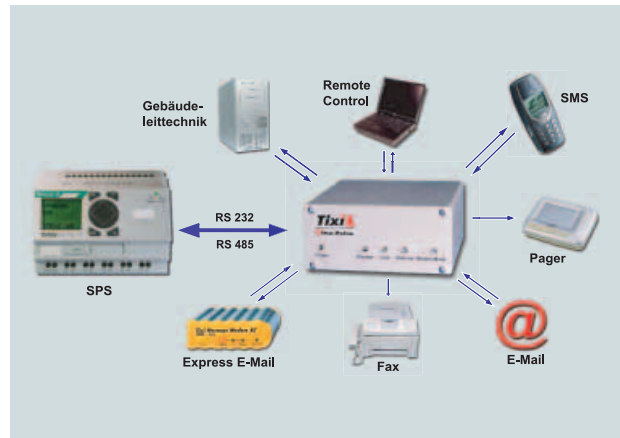


Intelligente Modems: Schluss mit AT-Befehlen?



Halle 9, Stand 259

Die Computer- und Elektronikwelt hat sich in den letzten 20 Jahren völlig verändert, doch ein wichtiges Gerät ist quasi auf dem Entwicklungsstand von 1981 stehen geblieben: das Modem. Man muss es heute mit exakt den gleichen AT-Befehlen steuern wie vor 20 Jahren. Welche Alternativen es gibt, zeigt der folgende Beitrag.

Der bekanntesten AT-Befehl kennen viele Techniker:

ATDT 03012345678 bedeutet, dass das Modem die angegebene Telefonnummer wählen soll (DT=Dial Tone = wählen mit "Tönen") und sich mit dem Modem auf der anderen Seite verbinden soll, damit Daten übertragen werden können. Doch was passiert dann? Wie werden Daten, Faxe, Emails usw. übertragen? Das kann ein Modem bis heute nicht, das muss immer ein Computer, meist ein PC, z.B. mit Windows oder Linux, machen.

Industrielösungen heute

Für die Industrie-Elektronik hat dieser "AT-Modem-Urgroßvater" jedoch fatale Folgen, denn diese Elektronik hat meist keine leistungsfähige Hardware und auch kein Betriebssystem wie Windows, das Kommunikationsfunktionen und -protokolle kostenlos zur Verfügung stellt. Industrie-Steuerungen haben keine schnellen CPUs, sehr wenig RAM, sehr wenig ROM und kein Standard-Betriebssystem mit eingebauten Kommunikationsfunktionen: Sie können ein AT-Modem nicht so wie

einen Windows-PC steuern und deshalb nicht mit der Außenwelt kommunizieren. Die Protokoll-Intelligenz fehlt in der CPU und im AT-Modem. Sie lässt sich auch nicht so leicht in SPS-Controller oder AT-Modems einbauen. Das ist der Grund, warum die Fern- und Internetkommunikation in der Industrie-Elektronik 8-10 Jahre hinter der PC- und Office-Welt hinterherhinkt. Heutige Kommunikationsprotokolle, wie z.B. das TCP/IP, POP3, SMTP, PPP (Internet, E-Mail), haben eine Vielzahl von Parametern. Außerdem erwartet man heute von Fernkommunikationsgeräten (z.B. Komfortfax, Komforttelefon, Handy) mehr als die bloße Sprach- oder Datenübertragung, z.B. Funktionstasten, Adressbücher für Telefonnummern u.v.a.. All diese vielen Parameter, die ständig mehr werden, sind mit den "einsilbigen" AT-Befehlen nicht mehr sinnvoll zu beherrschen.

XML als Alternative

In der Internetwelt ist HTML, die Hypertext Markup Language, die Grundlage des weltweiten Erfolgs der www-Web Server für den Austausch von formatierten Dokumenten mit

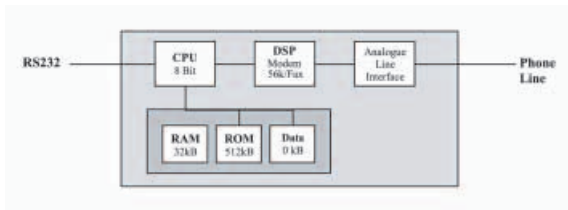
Graphiken, Java-Applets u.a. geworden. Eine ähnliche Revolution bahnt sich beim Austausch von fast beliebigen Daten zwischen beliebigen, plattformunabhängigen Programmsystemen an: XML, die Extensible Markup Language.

Was ist XML?

XML ist eine Metasprache zur Definition von Markup-Sprachen. Die bekannteste Markup-Sprache ist sicher HTML. An dieser Stelle sollen keine langen theoretischen Definitionen folgen. Ein kurzer Vergleich mit HTML mag genügen:

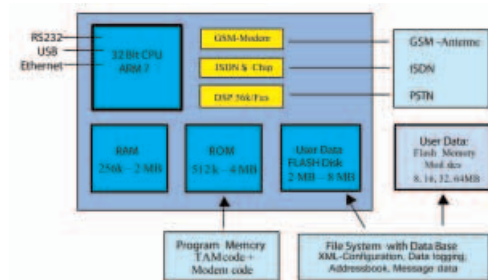
- Mit XML kann man die logische Bedeutung von Daten, Informationen und Texten definieren - ähnlich wie die Tabellen- und Spalten-Bezeichnungen in Datenbanken und Tabellenkalkulationen.
- XML ermöglicht im Gegensatz zu HTML die Definition eigener oder zusätzlicher "Befehle" (Tags) - ähnlich wie bei der Definition von Macros in der Textverarbeitung.
- XML-Applikationen eignen sich als Plattform- und Software-unabhängiges Austausch-Format für Daten zwischen verschiedenen Programmen und Rechnern - ähnlich wie RTF für Texte, CVS für Tabellen, EDI für kommerzielle Anwendungen - aber in einem einheitlichen, allgemein verwendbaren, herstellerunabhängigen Format.
- XML-Dokumente lassen sich in

Vergleich Standard Modem und Tixi Alarm Modem



Standard Analog 56k Modem: 8 Bit CPU, minimaler RAM, ROM und kein User-Datenspeicher. Jedes Analogmodem hat drei Hauptbestandteile (meist Einzelchips) sowie einen Line-Interface und einen DSP /AD/DA-Wandler). Zur Steuerung des

DSP, zum Verbindungsaufbau und -abbau und für die Konfigurierung und für die Kommunikation mit PC oder SPS wird meist ein kleiner 8-Bit-Controller (CPU) mit wenig RAM und ROM benutzt.



- *Kommunikationscomputer mit 32 Bit RISC CPU und 2MB-66MB User-Datenspeicher, 56k/V.90-Modem, ISDN, GSM/GPRS).*
- *Störungsmelder: sendet Nachrichten per SMS, Fax, E-Mail, Express E-Mail mit Quittungsempfang und Alarm-Kaskade.*
- *Fernwirkmodem: empfängt Befehle per SMS, E-Mail,*

Express E-Mail.

- *Dataloggermodem: zeichnet Daten auf und sendet diese als E-Mail-Attachement.*
- *SPS-Modem: unterstützt viele SPS-Typen (3S, ABB, Allen Bradley, Conrad, Mitsubishi, Moeller, Saia Burgess, Siemens, Theben, Vipa sowie M-Bus, ModBus, TixiBus, CAN, ..) und kann SPS-Variablen lesen und schreiben und Events auslösen (ohne dass in der SPS ein Treiber oder Baustein geladen werden muss).*
- *SPS-Funktionen: Input-Ports am Tixi-Modem können über Events gelesen und Output-ports gesetzt werden. Logische Operationen sind mit Ports, Variablen, Timer und Kalender möglich.*
- *Web Server Modem: SPS-Daten können über das Internet beobachtet und geändert werden.*
- *SPS-Fernwartung: RS232 Transparent-Mode zur SPS für die Fernwartung.*

Web-Browsern strukturiert anziehen.

- Die Syntax von XML ist streng festgelegt, so dass XML-Anwendungen wesentlich einfacher und effizienter von Programmen weiterverarbeitet werden können als HTML-Files.

XML-Syntax

Im Folgenden soll ein kurzer Einblick in XML-Strukturen zur Darstellung von Daten gegeben werden (weitergehende Informationen zu Stylesheets u.a. findet man bei www.XML.org oder als kurzen Überblick bei Hubert Partl 1.).

Start- und End-Tags

Die meisten Befehle (Tags) in XML-Anwendungen treten - wie auch in HTML - paarweise als Start- und End-Tags auf und geben an, welche Bedeutung der dazwischen liegende (eventuell durch weitere Tags unterteilte) Text hat. Bei den Tag-Namen ist die Groß- und Kleinschreibung zu beachten. Parameter zu den Tags werden als Attribute

bezeichnet. Die Werte der Attribute werden in Anführungszeichen oder Apostroph eingeschlossen

(z.B. Autor= "Meyer").

```
<Buch>...Text...</Buch>
```

oder

```
<Buch      Autor="Meyer"...>...
```

```
Text...</Buch>
```

Ein kleinen Adress-Datenbank-Beispiel soll das XML-Prinzip aufzeigen:

```
<person id="p4681">
  <vorname>Hubert</vorname>
  <zuname>Partl</zuname>
  <titel>Dr.</titel>
  <strasse>Mgasse118</strasse>
  <plz>A-1220</plz>
  <ort>Wien</ort>
  <geburtstag>
    <tag>28</tag>
    <monat>März</monat>
    <jahr>1949</jahr>
  </geburtstag>
</person>
```

Da Tags geschachtelt werden können, lassen sich auch komplizierte Datenstrukturen über-

Konfigurationsbeispiel SPS-Protokolle

Konfiguration von Variablen der SPS, die das Tixi-Modem zyklisch lesen oder setzen soll: (I=Input, acc=Zugriffsrechte, R=lesend, W=schreibend, def=Startwert vor erster Zuweisung, ind=index, format=Formatierung der Ausgabe in Textnachrichten.)

```
<External>
  <Bus _="C1" protocol="Moeller, Easy 800">
    <Device _="5" Pollrate="1s">
      <Eingang1 _="I" acc="R" def="0" ind="1"/>
      <Eingang2 _="I" acc="R" def="0" ind="2"/>
      <Eingang3 _="I" acc="R" def="0" ind="3"/>
      <Eingang4 _="I" acc="R" def="0" ind="4"/>
      <Ausgang1 _="Q" acc="RW" def="0" ind="1"/>
      <EasyUhr _="U" acc="R"/>
    </Device>
  </Bus>
</External>
```

sichtlich darstellen. Einen Spezialfall stellt ein Element ohne End-Tag dar:

Aus <Buch></Buch>

wird <Buch/>

oder

<Buch Autor="Meyer" ... />

Hier braucht der Tag-Name am Ende nicht wiederholt werden, was aus Platzgründen für kleine CPUs mit wenig Speicher von Vorteil ist.

Datenaustausch

Einige PC-Softwarehersteller bieten heute schon XML-Daten-Schnittstellen für ihre Programme an. Seit auch Microsoft angekündigt hat, XML als Datenformat für Word und andere Microsoft-Programme in Zukunft zu verwenden, dürfte der Siegeszug von XML vorprogrammiert sein.

Intelligente XML-Modems

Im Folgenden soll am Beispiel des Tixi Alarm Modems der Firma Tixi.Com die Relevanz von XML für die Automatisierungstechnik demonstriert werden. So werden die Konfigurationsmöglichkeiten mit XML für ein zeitgemäßes Modem gezeigt. Dieses Modem hat das Internet-Protokoll TCP/IP, sowie Fax- und SMS-Protokolle eingebaut, so dass der Anwender nur die notwendigen Parameter angeben muss, jedoch nichts programmieren muss, um Daten oder

Nachrichten mit diesen Protokollen übertragen zu können.

SPS-Unterstützung

Im Tixi Modem sind Protokolle von SPSen verschiedener Hersteller und allgemeine Feldbusprotokolle eingebaut (siehe Tabelle 1). Dadurch kann die Kommunikation von Tixi Modems mit einer SPS sehr einfach erfolgen, denn in der SPS brauchen keine Treiber oder Funktionsbausteine geladen werden, damit Tixi Variablen oder I/O-Ports lesen und setzen kann.

Beispiel Störmeldung

Wenn das Tixi Modem der Master der Kommunikation ist, fragt es die SPS-Variablen zyklisch ab (z.B jede Sekunde einmal) und speichert so eine Kopie der aktuellen, relevanten Werte der SPS. Zum Auslösen und zum Versenden von Störmeldungen aus dem Tixi Modem reicht es aus, wenn in der SPS ein Alarm-Flag von "0" auf "1" gesetzt wird. Beim nächsten Pollzyklus erkennt Tixi die Änderung des überwachten Flags und speichert diese in einer internen Prozessvariablen, mit der weitere Berechnungen (z.B Vergleiche mit anderen Ports oder einem Schichtplan) einen Status (Eventstate) ergeben. Der Eventhandler startet dann den Job-Generator, der über einen Textprozessor die Nachricht mit den aktuellen

Inhalten (z.B. Variablen-Werten) erzeugt. Zum Schluss sorgt das Job-System für den Versand der Nachricht über den jeweiligen Kommunikationsweg an den Empfänger.

Alarm-Kaskaden

Bei vielen Anwendungen ist es wichtig, dass Störungsmeldungen unbedingt in einer bestimmten Zeit zugestellt werden, damit die Störung schnell behoben werden kann. Wenn nun der Empfänger der Störungsmeldung nicht erreichbar ist, kann ein Tixi Modem automatisch weitere Störungsmeldungen an andere Ziele versenden. Erst wenn eine Empfangsbestätigung von einem Empfänger zum Tixi Modem zurück kommt, wird dieser Vorgang gestoppt.

XML fürs Modem

Aus der Sicht eines Modemnutzers und Entwicklers ergeben sich viele praktische Vorteile bei der Nutzung von XML:

- Erweiterungen des Funktionsumfangs sind leicht möglich (kein starrer Rahmen wie bei der Speicherung in festen Strukturen).
- XML passt sehr gut in das objektorientierte Design heutiger Software, da es selbst hoch strukturiert sein kann.
- Es ist kein Übersetzer in interne Strukturen nötig, denn die Abfrage von konfigurierten Werten erfolgt direkt auf XML.
- XML ermöglicht den Einbau von Scripten für Sonderprotokolle.
- XML kann sowohl für die Modem-Konfiguration als auch für Nutzerdaten verwendet werden.

Das wichtigste Argument für XML ist, dass man gute XML-Dokumente daran erkennt, dass man sie wie eine Beschreibung der Daten lesen und verstehen kann, ohne Handbuch.

XML-Nachteil: großes Dateivolumen

Neben den vielen Vorteilen hat XML auch einen nicht unbedeutenden Nachteil: die Daten werden stark aufgebläht. Das ist zwar für heutige PC-Systeme mit 2000MHz CPU, 500MB Hauptspeicher und Gigabyte-Festplatten kein Problem - doch

Dienste und Ereignisse

Das Tixi Modem System besteht aus verschiedenen Modulen, die mit einzelnen TiXML-Kommandos konfiguriert werden, z.B.

- Adressbuch
- Nachrichtebuch (Message Templates)
- Internet-Zugang
- Standortangaben
- Modemdaten
- Alarmkonfiguration
- Datenloggerkonfiguration
- Web-Serverkonfiguration
- Eventsteuerung mit logischen Operationen
- SPS-Variablen-Konfiguration

Alle Aktionen werden durch Events (Ereignisse) ausgelöst. Ein Ereignis kann sein:

- Änderung eines Inputports
- Änderung einer Variablen im Tixi-Modem oder in der SPS
- Tastendruck an der SPS oder am Tixi Modem
- Timer- bzw. Kalenderereignis (Schichtplanwechsel)
- Eingehende Nachricht (SMS, Email, Express E-Mail)
- Eingehender Ruf (ohne Abnehmen)
- Ein Fehler u.a.

für kleine CPUs mit 10-30MHz, wenig RAM, wenig ROM und nur wenigen Kilo- oder Megabytes an Userdatenspeicher ist das ein Problem. Die Designer von XML haben leider Klein-CPU-Anwendungen nicht bedacht und eine so simple Anweisung wie

```
<Vorname = "Meyer" />
```

vergessen. Das hätte die XML-Darstellung in vielen Fällen leichter überschaubar gemacht, weniger Platz verbraucht und Klein-CPU's weniger belastet.

TiXML - Datenreduzierung

Tixi.Com hat für die Konfiguration der Alarm-Modems einen XML-Trick eingeführt und damit zwei Fliegen mit einer Klappe geschlagen: Das Datenvolumen verringert sich um 30-40%, und die Lesbarkeit der XML-Datei verbessert sich, denn es stehen keine doppelten Informationen mehr in einer einzelnen Zeile.

Normale XML-Darstellung:

```
<vorname>Hubert</vorname>
```

TiXML-Darstellung:

```
<vorname _="Hubert" />
```

Das in XML zugelassene Zeichen "_" wird hier als Dummy-Attribut für den Tag benutzt, dem als Variable ein Wert zugewiesen werden soll.

Das Beispiel von oben sieht also in TiXML so aus:

```
<person id="p4681" >
  <vorname _="Hubert" />
  <zuname _="Partl" />
  <titel _="Dr." />
  <strasse _="Mgasse 118" />
  <plz _="A-1220" />
```

```
<ort _="Wien" />
<geburtstag>
  <tag _="28"/>
  <monat _="März"/>
  <jahr _="1949"/>
</geburtstag>
</person>
```

Vielleicht könnte diese Form in eine nächste XML-Version als normale Option aufgenommen werden (der Autor wird sich darum bemühen.)

Zusammenfassung

Man stelle sich vor, man müsste für all die hier gezeigten Funktionen 200 neue kryptische AT-Befehle lernen und anwenden... Modems mit AT-Befehlen sind offenkundig reif fürs Museum. Mit XML können die Anforderungen heutiger Automatisierungslösungen viel besser gelöst werden. Demzufolge entsteht eine neue Generation von intelligenten Modems, die Internet- und Fernkommunikation nun auch zu geringen Kosten und mit sehr wenig Aufwand in die Industrieelektronik bringen. Durch die Integration von SPS-Protokollen und Kommunikationsprotokollen in ein Modem, wird dieses zum zeitgemäßen Kommunikationscomputer. ■

18248

www.Tixi.com

Autor: Dipl.-Ing. Martin Brand ist Geschäftsführer der Tixi.Com GmbH, Berlin.