

# Mikrocontroller: 8 bit oder 16 bit?

Hermann Kramer

Mikrocontroller werden heute in allen Ausführungsklassen von 4 bit bis 32 bit Datenbreite eingesetzt. Die 8-bit-Architekturen beherrschen noch immer den Markt, allerdings mit geringer Zuwachsrate. Die größte Zuwachsrate verzeichnen derzeit die 16-bit-Architekturen, deren Verkaufszahlen mit denen der 8-bit-Architekturen im Jahr 1999 gleichziehen werden.

## Welcher Datenbreite soll in der HTL-Elektronik-Ausbildung der Vorzug gegeben werden?

Vor dieser Frage stellt sich mir allerdings das grundsätzliche Problem: Wie weit ist es für im Rahmen einer praktischen Ausbildung sinnvoll, sich überhaupt mit Mikrocontrollern näher auseinanderzusetzen?

Gehen wir von zwei extremen Ansatzpunkten aus:

1. Es genügt eine grundsätzliche Vorstellung von der Architektur. Viel wichtiger ist die Arbeit mit fertigen industriellen Einheiten wie SPS oder ausgereiften PC-Karten und das Programmieren auf hoher Ebene unter Windows mit mächtigen Software-Werkzeugen.
2. Die Schüler werden mit einem speziellen Mikrocontroller sehr detailliert vertraut gemacht, beherrschen den gesamten Befehlssatz und plagen sich auf der untersten Ebene mit den einzelnen Bits herum.

So wichtig mir der erste Ansatz erscheint, so sehr ist es doch für einen Techniker sinnvoll, ein Bewußtsein und ein Gefühl für das zu entwickeln, was da hintersteht. Daher scheint mir ein näheres Kennenlernen eines Mikrocontrollers, mit "Augenmaß", nach wie vor wünschenswert.

Wenn 8-bit-Systeme für eine gediegene Grundausbildung völlig ausreichen und wenn man im Rahmen des Unterrichts kaum an die Grenze der Ressourcen eines 8-bit-Systems stößt, warum sollte man dann die Schüler mit einer 16-bit-Architektur überfordern?

## Programmieren in C

Zur Einführung und zum grundsätzlichen Kennenlernen des Programmablaufs ist zweifellos ein einfaches 8-bit-System, wie

die 8051-Familie, überschaubarer. Allerdings halte ich es für problematisch, größere Projekte in Assembler zu programmieren. Das ist ein strukturierter Denkweise sicher nicht förderlich. Hier sollte unbedingt eine höhere Programmiersprache ins Auge gefaßt werden. Es gibt natürlich C-Compiler für 8051, aber es gibt meines Erachtens keinen guten, effektiven C-Compiler, einfach deswegen, weil der Befehlssatz nicht für Hochsprachen entwickelt wurde. Die Geschwindigkeitsverluste gegenüber einem Assemblerprogramm ist deutlich. Das ist für mich das Hauptargument für ein 16-bit-System.

## Gibt es Argumente gegen 16 bit?

Von der Hardware-Seite kaum, denn es gibt Konfigurationen, die nach außen mit einem 8-bit-Bus arbeiten, wodurch sich relativ einfache Minimalsysteme bauen lassen. Vom Preis her kaum, die Starter-Kits kosten annähernd gleich viel. Eigentlich kann man nur anführen, daß für den Lehrer die Einarbeitung in die sehr mächtigen und umfangreichen Entwicklungstools und die schülergerechte Einrichtung eine große Zeit in Anspruch nimmt.

Dank der sehr guten Unterstützung von Siemens fiel unsere Wahl auf den Siemens-Mikrocontroller C167CR, den wir im Projektunterricht der 5. Klasse mit Starter-Kits von Phytex einsetzen. Einige attraktive Einrichtungen seien aufgezählt:

- Trotz hoher Komplexität übersichtliche Architektur
- Hohe Rechenleistung (die meisten Befehle dauern nur 1 Zykluszeit)
- Übersichtliches Interrupt-System
- 9 Timer, Reload-, Capture-, Compare-Möglichkeit
- PWM
- 10-bit-ADC mit 16 Eingängen
- Schnittstellen: RS232, schnelle, vielseitige synchrone Schnittstelle, CAN.

## CAN

Besonders attraktiv erscheint mir der integrierte CAN-Kontroller. Im Bereich der Sensoren und Aktoren besteht die Tendenz zur Vernetzung ähnlich wie bei den Computer-Netzen. Die dabei eingesetzten

Feldbusse haben im Vergleich zu Computer-Netzen folgende Eigenschaften:

- Die anfallenden Datenraten sind eher gering.
- Die Echtzeit-Anforderungen sind wesentlich größer.

Der CAN-Bus (Control Area Network) zeichnet sich durch hohe Störsicherheit, große Geschwindigkeit und gute Echtzeiteigenschaften aus. Die CAN-Kontroller sind aufgrund der hohen Produktionsstückzahlen zudem äußerst preiswert. CAN-Kontroller findet man in letzter Zeit häufig in Mikrocontrollern integriert. Der Siemens C167CR besitzt eine CAN-Einheit, die praktisch mit dem bekannten Intel-CAN-Kontroller 82527 identisch ist. Durch vorhandene CAN-Bibliotheken (eine solche ist zum Beispiel in der neuen Version des Tasking-C-Compilers integriert) ist die Bedienung des CAN-Busses nicht problematisch.

## Projekt mit C167CR und CAN

Zum Abschluß möchte ich ein im Schuljahr 1997/98 an der HTL Braunau durchgeführtes Ingenieur-Projekt vorstellen. Es geht dabei um die innovative Ansteuerung von Beleuchtungsfeldern (z. B. Werbeflächen, Weihnachtsbeleuchtung). Auf dem PC wird mit einem Zeichenprogramm unter Windows ein Bitmuster entworfen, wo bei je des Bit einer Lampe des Beleuchtungsfeldes entspricht. Außerdem wird die Abfolge festgelegt (Scroll-Möglichkeit vertikal und horizontal, Bildwechsel, ...). Diese Informationen werden in ein geeignetes Format gebracht und über die serielle Schnittstelle einem C167CR-Mikrocontroller-System übergeben. Dieses Master-System verteilt die Informationen an beliebig viele Slave-Systeme (einfache 8751-Systeme mit Intel CAN-Kontroller 82527), die ihrerseits für die Ansteuerung von Modulen mit je 5 x 5 Lampen verantwortlich sind. Die serielle Verbindung des Masters zum Computer ist nur bei einer Neuprogrammierung nötig.

Die Einarbeitung und der Umgang mit dem an der HTL Braunau neu eingeführten 16-Bit-Mikrocontroller-System gestalte sich auch bei einer weit entfernten Projektarbeit im Großen und Ganzen unproblematisch.