

# Mikrocontroller



von Markus Koch

April 2011

# Inhaltsverzeichnis

1 Was ist ein Mikrocontroller?.....	3
1.1 Einleitung.....	3
1.2 Aufbau.....	3
1.2.1 CPU.....	3
1.2.2 Speicher.....	3
RAM.....	3
ROM.....	4
1.2.3 Takt.....	4
1.2.4 Peripherie.....	4
1.2.5 Unterschied zwischen Mikroprozessor und Mikrocontroller.....	5
1.3 Unterschiede verschiedener Controllerarten.....	5
1.3.1 Allgemein.....	5
1.3.2 Beispiele für bekannte Mikrocontroller (8-Bit).....	6
1.3.3 Bauformen.....	6
1.4 Unterschiede zum PC.....	6
2. Wie fange ich mit Mikrocontrollern an?.....	7
2.1 Einleitung.....	7
2.2 Wo kann ich Mikrocontroller und die benötigte Hardware kaufen?.....	8
2.3 Der Anfang.....	8
2.4 Wie programmiere ich einen Mikrocontroller?.....	8
3. Verwendung von Mikrocontrollern.....	9
3.1 Verwendung.....	9
3.2 Warum gibt es nicht nur Mikrocontroller?.....	9
4. Quellen.....	10
4.1 Informationsquellen.....	10
4.2 Bilder.....	10
5. Selbstständigkeitserklärung.....	11

# 1 Was ist ein Mikrocontroller?

## 1.1 Einleitung

Ein Mikrocontroller (Abk.  $\mu$ C oder MCU) ist ein IC, das von Anwender frei programmiert werden kann. IC kommt aus dem englischen „Integrated Circuit“ und heißt auf deutsch „Integrierte Schaltung“. Ein solches Bauteil enthält viele andere elektronische Bauelemente wie zum Beispiel Transistoren oder Widerstände. Ein Mikrocontroller besteht aus Millionen solcher Elemente.

## 1.2 Aufbau

Ein Mikrocontroller lässt sich in vier Bereiche aufteilen: Die CPU, den RAM, den Takt und die Peripherie.

### 1.2.1 CPU

Die Central Processing Unit, besser bekannt als Prozessor führt die Berechnungen und logische Operationen durch und ist somit das Herzstück des Mikrocontrollers. Diese CPU kann man grundsätzlich mit dem Prozessor eines PCs vergleichen, nur ist sie bei einem Mikrocontroller nicht wechselbar.

### 1.2.2 Speicher

#### **RAM**

Im Arbeitsspeicher werden während des Betriebs temporär Daten wie zum Beispiel Variablen abgelegt. Nach dem Ausschalten, also dem Entfernen der Betriebsspannung, gehen diese Daten verloren und können nicht wieder ausgelesen werden.

Der Arbeitsspeicher zeichnet sich durch eine lange Lebensdauer und schnelle Lese- sowie Schreibgeschwindigkeit aus.

Beispiel: Ein Gerät soll die Temperatur im Raum messen und anzeigen. Hierbei wird der Temperaturwert erst im RAM zwischengespeichert, bevor er auf dem Display angezeigt wird. Der Wert ist nach dem Ausschalten nicht mehr vorhanden. Dies ist jedoch nicht mehr schlimm da später sowieso der Wert neu eingelesen und angezeigt wird.

Der RAM im Mikrocontroller entspricht somit dem RAM im PC, jedoch ist dieser meist auch schon im Mikrocontroller eingebaut und kann nicht gewechselt oder erweitert werden.

## **ROM**

Das ROM (engl. **Read-Only-Memory**) ist der Festspeicher des Mikrocontrollers. In diesem werden das Programm und dessen Einstellungen gespeichert. Da diese Speicher nur für eine bestimmte Anzahl an Schreibvorgängen ausgelegt sind, werden oft Programmspeicher und Einstellungsspeicher getrennt.

So verwendet man für den Programmspeicher, welcher normalerweise nur ein Mal bei der Herstellung (des Produkts) programmiert wird, ein Flash-ROM. Dieses lässt sich etwa 10'000 Mal neu beschreiben. Nicht selten werden auch so genannte Mask Programmable ROM-Chips verwendet, welche sich nur einmal bei der Herstellung programmieren lassen.

Beim Einstellungsspeicher wird oft ein so genanntes EEPROM (Electrically Erasable Programmable ROM – dt. Elektronisch löschbarer programmierbarer Festwertspeicher) verwendet, welches 100'000 Schreibzyklen hält.

Ein ROM hat eine schnelle Lesegeschwindigkeit aber langsame Schreibgeschwindigkeit. Jedoch bleiben beim ROM die Daten nach entfernen der Betriebsspannung erhalten.

Das ROM würde im PC der Festplatte entsprechen. Beim PC wird jedoch nicht zwischen Anwendungs- und Einstellungsspeicher getrennt.

### **1.2.3 Takt**

Der Prozessortakt gibt vor, mit welcher Geschwindigkeit die Daten vom Prozessor verarbeitet werden. So bedeutet ein Takt von 1 MHz, dass eine Millionen Befehle pro Sekunde verarbeitet werden. Ohne einen Takt kann ein Mikrocontroller bzw. Mikroprozessor nicht arbeiten.

Nach oben hin ist die Taktrate durch die verwendeten Transistoren limitiert.

Typische 8-Bit Mikrocontroller (Dazu später mehr) gehen von 1MHz bis 30MHz. Solche Taktraten werden mit Hilfe von Quarzen bzw. Quarzoszillatoren erreicht.

### **1.2.4 Peripherie**

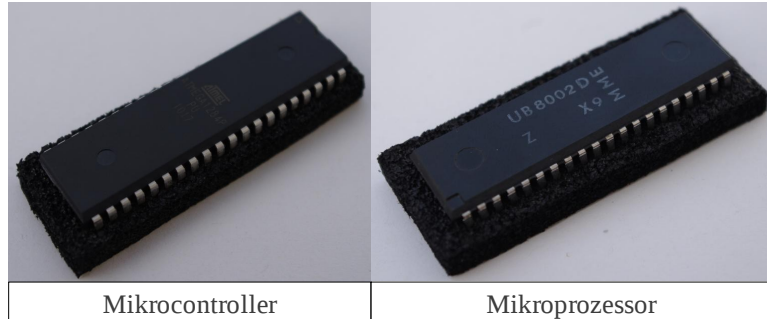
Die Peripherie ist mit der wichtigste Teil eines Mikrocontrollers. So ist sie dafür verantwortlich, dass man überhaupt sieht, was der Prozessor berechnet. Ohne Peripherie fehlt also die Verbindung nach außen.

Typische Peripherie eines Mikrocontrollers sind einerseits so genannte I/O-Pins. I/O-Pins sind nichts anderes als Ein- und Ausgänge. Damit kann man zum Beispiel Schalter abfragen oder LEDs ein- und ausschalten. Ebenso lassen sich über diese I/O-Pins Daten schicken oder auch Motoren schalten. Die Möglichkeiten sind nahezu unbegrenzt. Des Weiteren sind Schnittstellen zur di-

rekten Kommunikation mit einem PC oder anderer Hardware vorhanden. Beispiele dafür sind eine serielle Schnittstelle oder I<sup>2</sup>C.

## 1.2.5 Unterschied zwischen Mikroprozessor und Mikrocontroller

Sehr oft wird ein Mikroprozessor mit dem Mikrocontroller verwechselt. Der Unterschied liegt darin, dass der Mikroprozessor wirklich nur die Recheneinheit ist. Ein Mikroprozessor benötigt noch RAM,



ROM und die anderen oben genannten Komponenten um arbeiten zu können.

Ein Mikrocontroller hingegen ist eine Kombination aus dem Prozessor und den anderen Teilen. Ein Mikrocontroller benötigt eigentlich nur die Betriebsspannung und Peripherie um zu arbeiten.

Rein optisch gibt es keinen Unterschied zwischen den beiden Bauteilen.

## 1.3 Unterschiede verschiedener Controllerarten

### 1.3.1 Allgemein

Es gibt sehr viele verschiedene Mikrocontroller. Diese unterscheiden sich in den eben oben genannten Punkten. Es gibt Mikrocontroller mit viel RAM, wenn zum Beispiel große Datenmengen verarbeitet werden müssen. Andere hingegen haben wenig RAM dafür aber spezielle Schnittstellen, also andere Peripherie. Ebenso gibt es Chips, die mehr auf hohe Taktraten ausgelegt sind oder einen besonders großen Programmspeicher haben.

Der wichtigste Unterschied ist jedoch die Bit-zahl, also wie viele Bit die Register<sup>1</sup> der CPU und des RAMs breit sind. Ein kurzer Beispiel zur Veranschaulichung:



Ein 8-Bit-Mikrocontroller hat wie der Name bereits vermuten lässt, eine 8-Bit breite Speicherarchitektur. Das bedeutet der Prozessor kann theoretisch nur Zahlen von 0-255 (-128 – 127) verarbeiten. Durch Software lassen sich dann zwar höhere Zahlen berechnen, jedoch dauern diese Vorgänge wesentlich länger.

Bei einem 32-Bit Prozessor sind die Speicher nun 32-Bit breit. Man kann nun also Zahlen von über vier Milliarden direkt in der Hardware berechnen.

---

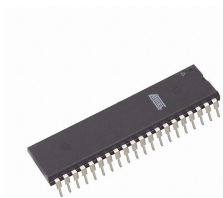
<sup>1</sup> Register: Speicherbereiche innerhalb der CPU

### 1.3.2 Beispiele für bekannte Mikrocontroller (8-Bit)

Hersteller	Bezeichnung	Speicher; RAM	Takt	Preis
	AtMega	4kB – 256kB; 1kB - 16kB	32kHz - 20MHz	1€ - 10€
	AtTiny	512B – 16kB; 32B - 512B	32kHz - 20MHz	1€ - 10€
	PIC	512B – 256kB; 16B - 12kB	32kHz - 64MHz	50ct - 15€

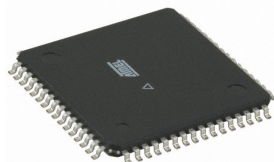
### 1.3.3 Bauformen

Mikrocontroller gibt es wie die meisten ICs, in vielen Bauformen. Die sogenannte DIP-Bauform ist die alte Bauform, bei der man die Beinchen noch durch die Platine steckt. Diese wird in der Industrie immer seltener



DIP-Bauform

verwendet, bei Hobbyelektronikern ist diese Form jedoch noch sehr beliebt, da sie vergleichsweise einfach zu löten ist.



SMD-Bauform

Der aktuelle Standard nennt sich SMD (Surface Mount Device). Bei diesem werden die Beinchen nicht mehr durch gesteckt, sondern direkt auf der Ober-

seite verlötet.

### 1.4 Unterschiede zum PC

Man kann einen Mikrocontroller grundsätzlich mit einem PC vergleichen und sie haben einige Gemeinsamkeiten. Jedoch gibt es auch einige Unterschiede. Hier mal die Wichtigsten:

	Mikrocontroller	Desktop-PC
<b>Prozessortakt (Bereich)</b>	MHz	GHz
<b>Speicherbreite</b>	Meist 8-Bit; aber auch bis 32-Bit	32-Bit
<b>Speicher (Bereich)</b>		
<b>Festspeicher</b>	Kilobyte	Gigabyte – Petabyte
<b>RAM</b>	Kilobyte	1 – 4 Gigabyte (bei 32-Bit)
<b>Peripherie</b>	I/O-Ports; sonstige Schnittstellen	Bildschirm, Tastatur, ...
<b>Aufgabe</b>	Meist eine bestimmte Aufgabe	Viele Aufgaben (parallel)

Nach dieser Tabelle sieht es so aus, als hätte der Mikrocontroller vor Allem Nachteile, jedoch gibt es auch einige Vorteile der Mikrocontroller. So sind diese Beispielsweise sehr klein und verbrauchen vergleichsweise sehr wenig Strom. Ein Mikrocontroller benötigt weniger als 0,05 Watt – ein PC hingegen 400 Watt. Es ist also kein Problem, ihn mit Batterien zu versorgen. Die Größe von oft weniger als 1 auf 1cm ermöglicht es, die Controller in kleinen Geräten, wie Computermäusen oder Handys zu verbauen. Ein Mikrocontroller kostet auch nur 1-10€.

Ein wichtiger Unterschied ist, dass Mikrocontroller meist nur eine bestimmte Aufgabe ausführen, während PCs viele Programme gleichzeitig ausführen. So ist ein Mikrocontroller zum Beispiel dafür da, die Tastatureingaben an den PC weiterzuleiten. Dies ist seine einzige Aufgabe in dieser Schaltung. Am PC hingegen kann man Musik hören, ein Dokument schreiben oder im Internet surfen – Alles parallel.

Ebenso gibt es auf einem Mikrocontroller nur sehr selten ein Betriebssystem. Die Programme werden direkt in eine Sprache umgewandelt, die der Prozessor ohne weitere Ergänzungen versteht.

## 2. Wie fange ich mit Mikrocontrollern an?

### 2.1 Einleitung

Wenn man anfangen möchte, sich mit Mikrocontrollern zu beschäftigen, gibt es 2 Wege. Entweder man fängt gleich richtig an und kauft sich alles einzeln oder man geht den Weg über einen Experimentierkasten.

Der Einstieg mit Hilfe eines Experimentierkastens ist meist wesentlich einfacher, da man fertige Experimente hat und diese ausführlich erklärt sind. Der Nachteil dieser Lösung ist, dass diese Kästen meist Hard- und Softwaremäßig eingeschränkt sind. So sind nur wenige Pins herausgeführt und es wird eine vereinfachte Programmiersprache verwendet.

Der andere Weg ist für Einsteiger wesentlich schwerer, da man den Mikrocontroller nicht ohne Weiteres an den PC anschließen kann. Man benötigt ein Programmiergerät. Ebenso muss man selbst etwas aufbauen, um überhaupt etwas damit machen zu können. (Z.B. LEDs und Taster.) Dann fehlen einem hier die Beispiele und Erklärungen, wobei es auch gute Bücher gibt, die auf so etwas aufbauen.

Eine Lösung, die beides vereint sind Experimentierplatten oder Evaluationsboards. Diese bestehen nur aus einer Platine mit Mikrocontroller und diverser Peripherie, die aber in einer richtigen Programmiersprache programmierbar ist. Häufig werden diese vom Hersteller selbst angeboten.



C-Control Pro Evaluationsboard

## **2.2 Wo kann ich Mikrocontroller und die benötigte Hardware kaufen?**

Mikrocontroller gibt es in jedem Elektronikgeschäft wie Conrad oder Reichelt, um die vermutlich Bekanntesten zu nennen. Diese haben meist auch die benötigten Programmiergeräte da, welche leider oft sehr teuer (über 100€) sind. Jedoch gibt es die Alternative, diese Geräte selbst zu bauen. Dann kann man sie bereits für unter 10€ bekommen.

## **2.3 Der Anfang**

Sollte man sich für einen Experimentierkasten entschieden haben, sollten an dieser Stelle keine Fragen mehr auftreten.

Bei der zweiten Lösung ist es jedoch wichtig, sich entweder ein gutes Buch oder im Internet eine gute Anleitung zu suchen. Eine empfehlenswerte Internetseite ist beispielsweise <http://mikrocontroller.net/>. Mit einer guten Anleitung sollte es Möglich sein bereits innerhalb weniger Tage oder sogar Stunden die ersten Erfolge zu erleben.

## **2.4 Wie programmiere ich einen Mikrocontroller?**

Zur Programmierung eines Mikrocontrollers wird ein Programmiergerät benötigt. Als Software auf dem PC benötigt man einen Compiler und eine Software, um das Programm auf den Mikrocontroller zu übertragen. Der Compiler erstellt aus dem geschriebenen Quelltext eine Binärdatei, die der Controller versteht. Ein Beispiel für die Sprache C bzw. C++ wäre der avr-gcc. Möchte man in Basic für den Controller programmieren, so wäre Bascom die richtige Wahl.

Um nun die Software auf den Mikrocontroller zu übertragen, benötigt man ein Programm („Programmer“), welches den verwendeten Programmieradapter unterstützt. Ein Beispiel für ein Programm, dass alle gängigen Adapter unterstützt ist avrdude.

Die ersten Programme werden zweifellos eine blinkende LED oder ein Lauflicht sein. Weiter geht es dann mit Displays, Motoren und vielem mehr.



## **3. Verwendung von Mikrocontrollern**

### ***3.1 Verwendung***

Mikrocontroller sind in so gut wie jedem Gerät enthalten. Von Waschmaschinen bis zu Fernsehgeräten ist Alles dabei. Oft stecken auch mehrere Controller in einem Gerät.

Ein Mittelklasseauto hat beispielsweise bereits 30 oder mehr Mikrocontroller verbaut. So gut wie jeder Computerzubehör enthält mindestens einen Controller.

Sie führen somit vorwiegend Steuerungsaufgaben durch, jedoch gibt es sie auch in Spielkonsolen und anderer Unterhaltungselektronik.

### ***3.2 Warum gibt es nicht nur Mikrocontroller?***

Die Mikrocontroller sind grundsätzlich ein ganzer Computer auf einem Chip. Theoretisch könnte man sie auf die gleichen Geschwindigkeiten wie die aktuellen Prozessoren takten. Warum trennt man also bei einem PC noch CPU, RAM und so weiter und benutzt keinen Mikrocontroller.

Ein Grund ist auf jeden Fall, dass die CPU dann um einiges größer werden würde und die Produktionskosten drastisch ansteigen würden. Des Weiteren wäre es nicht möglich, Reparaturen vor zu nehmen, da die einzelnen Komponenten nicht austauschbar sind. Man müsste den PC also wegschmeißen, wenn beispielsweise nur ein Teil des RAM kaputt geht. Eine Erweiterung beziehungsweise ein Update der Hardware wäre dadurch auch nicht mehr möglich.

## 4. Quellen

### 4.1 Informationsquellen

- Wikipedia (Englisch): <http://en.wikipedia.org/wiki/Microcontroller> [11.04.2011]
- YouTube (Englisch): <http://www.youtube.com/watch?v=jKT4H0bstH8> [11.04.2011]
- Homepage Atmel (Englisch): <http://atmel.com/> [13.04.2011]
- Homepage Microchip (Englisch): <http://microchip.com/> [13.04.2011]

### 4.2 Bilder

- Mikrocontroller: Markus Koch
- Firmenlogos: Atmel/Microchip
- C-Control Evaluationsboard: Conrad
- Bauformen: Digikey

## **5. Selbstständigkeitserklärung**

Ich habe diese GFS ohne fremde Hilfe und nur mit den im Verzeichnis angegebenen Hilfsmitteln angefertigt. Ich habe alle Stellen, die im Wortlaut oder dem Sinn nach anderen Veröffentlichungen entnommen sind, durch Angabe der Quellen als Entlehnungen kenntlich gemacht.

Markus Koch