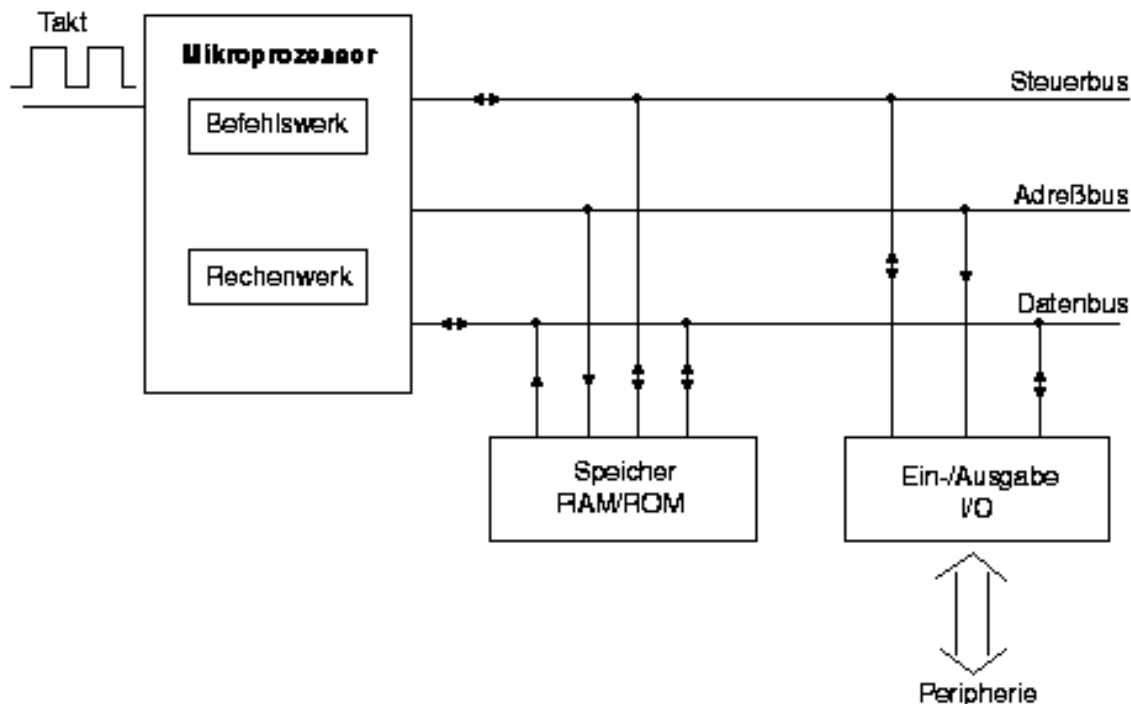


# Grundlagen Mikrocontroller ( $\mu\text{C}$ )

## 1. Prinzipieller Aufbau am Beispiel eines 8-Bit-Mikrocontrollers

Den Grundstein für einen Mikrocontroller bilden drei Baugruppen: die **Zentraleinheit**, die unter dem Namen Mikroprozessor oder **CPU** (Central Processing Unit) bekannt ist, der **Zentralspeicher**, der in ROM (Read Only Memory) und RAM (Random Access Memory) unterteilt ist, und die **Ein- /Ausgabe-Einheiten**, den sogenannten **Inputs und Outputs (I/O)**. Im Gegensatz zu einem Mikroprozessorsystem sind diese Baugruppen bei einem Mikrocontroller stets (manchmal aber nur teilweise) in einem einzigen Baustein integriert, weshalb man auch von „Single Chip Computer“ spricht.



Quelle: [www.fernando-heitor.de](http://www.fernando-heitor.de)

Die Zentraleinheit setzt sich zusammen aus dem **Befehlswerk** und dem **Rechenwerk**. Im **Befehlswerk**, der sogenannten **Control Unit (CU)**, befindet sich das Befehlsregister, das alle Befehle enthält, die die Zentraleinheit ausführen kann. Auch die Steuerung der Ein-/Ausgabe-Einheiten und des Steuerbusses gehört zu den Aufgaben des Befehlswerkes. Das **Rechenwerk**, die **Arithmetic Logic Unit (ALU)**, führt alle logischen und mathematischen Operationen (z.B. Addition, Subtraktion, Und-/Oder-Verknüpfungen) durch. Hinzu kommen noch die Operationen Multiplikation und Division, die zwar früher in Co-Prozessoren integriert waren, heutzutage aber in fast allen Zentraleinheiten zu finden sind. Das Rechenwerk bearbeitet zusätzlich die Flags (Ereignisspeicher) sowie unterschiedliche Registersätze, die nach einem bestimmten Zustand einer Funktion oder einer Berechnung gesetzt oder zurückgesetzt werden.

Der Zentralspeicher ist in einen „Nur-Lese-Bereich“ (ROM) und in einen Bereich mit „wahlfreiem“ Zugriff (RAM) eingeteilt. Der ROM-Speicher enthält das Programm, weshalb man ihn auch als Programmspeicher bezeichnet. Der RAM-Speicher dient zur Bearbeitung von Daten und heißt deshalb auch Datenspeicher.

Zur Kommunikation mit der „Außenwelt“ werden Ein-/Ausgabe-Einheiten (= Ports) benötigt. Zusammen mit weiteren Komponenten (z.B. Timer) bilden sie die sog. Peripherie eines Mikrocontrollersystems.

Verbunden sind diese drei Baugruppen durch die drei Bussysteme **Adressbus**, **Datenbus** und **Steuerbus**.

Der **Adressbus** ist bei den meisten 8-Bit-Mikrocontrollern 16-Bit breit, über ihn teilt die CPU den Speicher-Bausteinen mit, mit welchen Speicherzellen Daten ausgetauscht werden sollen.

Der **Datenbus** ist 8-Bit breit und transportiert reine Informationsdaten zwischen den Einheiten hin und her. Physikalisch ist er häufig ein Teil des Adressbusses, d.h. die unteren Leitungen (Bit0 – Bit7) werden auch als Datenleitungen benutzt.

Die Breite des **Steuerbusses** ist je nach CPU verschieden und liegt zwischen vier und zehn Leitungen. Er sorgt dafür, dass die einzelnen Bausteine getrennt voneinander gesteuert werden können.

Jeder Mikrocontroller hat einen Eingang für ein rechteckförmiges Taktsignal, das in der Regel von einem Quarz erzeugt wird. Mit diesem Taktsignal werden alle Abläufe der Zentraleinheit und des gesamten Mikrocontrollersystems gesteuert.

## 2. General Purpose Input Output (GPIO)

Die Anschlüsse eines Mikrocontrollers werden Pins genannt. Einige haben eine feste Funktion (z.B. Stromversorgung), die Anderen können je nach Bedarf genutzt werden. Die frei verwendbaren Pins werden als General Purpose Input Outputs, kurz GPIOs, bezeichnet. GPIOs werden allgemein als Ein- bzw. Ausgänge für digitale Signale eingesetzt. Je nach Mikrocontroller ist es aber auch möglich, damit analoge Werte zu messen oder mit einer externen Hardware zu kommunizieren, die ein bestimmtes Interface erfordert (z.B. OLED-Display mit I2C-Bus-Schnittstelle).

### **Wichtig:**

Die konkrete Funktion eines GPIOs muss bei den meisten Mikrocontrollern über die Programmierung festgelegt werden!

Gehäuseintern sind GPIOs in Ports zusammengefasst, wobei normalerweise 8 Pins einem Port zugeordnet sind, es können jedoch auch weniger Anschlüsse sein. Ob ein Pin direkt angesprochen werden kann oder über eine Portangabe, hängt vom verwendeten Mikrocontroller ab.

Jeder GPIO-Pin besitzt bestimmte elektrische Eigenschaften (z.B. wie viel Strom er als Ausgang liefern kann), die **unbedingt** beachtet werden müssen, da sonst der Mikrocontroller zerstört werden kann. Genaue Angaben hierzu stehen im zugehörigen Datenblatt (engl. „Datasheet“).