

Grundlagen Mikrocontroller

Was Mikrocontroller von Mikroprozessoren unterscheidet

Mikrocontroller sind allgegenwärtig, mehr als 20 Milliarden kommen jährlich auf den Markt – viele als unsichtbarer Bestandteil anderer Chips. Mikrocontroller dienen anderen Zwecken als Mikroprozessoren.

Von Christof Windeck

Ein Blick auf Extremfälle verdeutlicht Unterschiede: Der winzige Mikrocontroller NXP Kinetis KLO3 passt mit $1,6 \times 2$ Millimeter Kantenlänge in die Delle eines Golfballs, der mächtige Mikroprozessor AMD Epyc 7002 steckt in einem fast handtellergroßen Gehäuse. Der KLO3 kostet unter 1 Euro, hat einen einzigen 32-Bit-Rechenkern vom Typ ARM Cortex-M0+, taktet mit 48 MHz und kommt mit 8 Mikrowatt bis 400 Milliwatt aus. 32 KByte Flash-Speicher sowie 2 KByte Arbeitsspeicher sind eingebaut und mehr RAM kann er auch nicht ansteuern. Ganz anders der AMD Epyc mit 64 64-Bit-Kernen, der bis zu 4 TByte RAM und 128 PCI-Express-Lanes anbindet, aber auch 280 Watt verheizt.

Es gibt allerdings eine breite Grauzone, in der die Unterschiede zwischen Mikrocontrollern und Mikroprozessoren verschwimmen. Erstere gibt es nämlich auch mit mehreren Kernen, die über 1 GHz Taktfrequenz erreichen, mit Virtualisierungsfunktionen und Anschluss für externes DRAM. Umgekehrt begnügen sich manche Systems-on-Chip (SoC) mit x86-Kernen mit höchstens 6 Watt und besitzen typische Mikrocontroller-Schnittstellen. Einen Mikrocontroller nennt man auch Micro Controller Unit (MCU) oder kürzer μC , für einen Mikroprozessor gibt es die Abkürzungen Central Processing Unit (CPU) und Micro Processing Unit (MPU).

Prozessor oder Controller?

Die grundsätzlichen Unterschiede zwischen μC und CPU liegen auf anderen Ebe-

nen. Eine CPU führt typischerweise ein funktionsreiches Betriebssystem aus, das die Installation und Ausführung vieler verschiedener Programme (oder Apps) ermöglicht, auch gleichzeitig (Multitasking). Die CPU verarbeitet Tausende laufende Prozesse und Threads sehr schnell, aber es kommt selten darauf an, dass diese auf wenige Millisekunden genau vorhersagbar ablaufen.

Ein Mikrocontroller steuert hingegen meistens nur wenige immergleiche Funktionen, verarbeitet also eher eine Art Firmware. Als Herz einer Steuerung muss ein μC in vorhersagbarer Zeit – quasi in Echtzeit (Real Time) – Berechnungen erledigen. Das Steuergerät eines Verbrennungsmotors muss Messwerte etwa von Kurbelwellensensor, Luftmassenmesser und Gaspedal innerhalb von Millisekunden zu einem Signal für die Einspritzdüsen verarbeiten. In anderen Anwendungen geht es um „funktionale Sicherheit“: Im Regler eines Antiblockiersystems oder Herzschrittmachers dürfen keine Programmfehler auftreten. Dafür gibt es Prüfverfahren wie IEC 61508 oder für Autos auch ISO 26262 ASIL A bis D (Automotive Safety Integrity Level).

Der μC -Programmcode soll folglich möglichst schlank sein und komplett ins RAM passen; virtuelle Speicherverwaltung sowie die dabei hilfreiche Hardwareeinheit (Memory Management Unit, MMU) sind oft überflüssig oder stören sogar. Für einfache Aufgaben kann man Mikrocontroller ohne Betriebssystem programmieren. Gängig sind jedoch modulare und für bestimmte Anwendungsbereiche optimierte μC -Betriebssysteme, beispielsweise für Echtzeit-Steuerungsaufgaben (Real Time Operating System, RTOS). Wikipedia verzeichnet mehr als 140 RTOS-Varianten.

In vielen μC ist Flash-Speicher für Code und RTOS eingebaut, um Zusatzbausteine einzusparen. Mikrocontroller für Steuerungsaufgaben haben eine Fülle von Ein- und Ausgängen: Analog-Digital-(A/D-) und Digital-Analog-(D/A-)Wandler, digitale Ein- und Ausgänge (General Purpose I/O, GPIO), Funktionsblöcke für

Pulsweitenmodulation (PWM) sowie für standardisierte Schnittstellen wie I²C, SPI und UART, letztere oft als RS-232 oder RS-485 nutzbar. Bei „Automotive“- μC s für Fahrzeuge ist eine CAN-Bus-Schnittstelle üblich. Viele Hersteller kombinieren einen μC mit zusätzlichen Funktionsblöcken zu einem SoC, etwa Espressif den ESP32 mit WLAN-Einheit für Smart-Home- und IoT-Geräte.

Mikroprozessoren haben tendenziell wenige eingebaute I/O-Schnittstellen, dafür jedoch besonders leistungsstarke wie PCI Express, die Datentransferraten von mehreren Gigabyte pro Sekunde schaffen. Darüber wiederum ist dann oft ein Chipsatz angeflanscht, der etwa SATA, USB, SPI und I²C anbindet.

MCU-Vielfalt

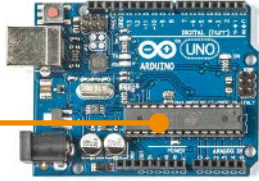
Bei CPUs dominieren wenige Befehlssatzarchitekturen (Instruction Set Architectures, ISA), weil es auf (Binär-)Kompatibilität zu Betriebssystemen und Software ankommt. Beispiele sind x86 (AMD, Intel, VIA), ARM Cortex-A (Apple, Qualcomm, Samsung, MediaTek), IBM Power, MIPS und RISC-V RV64GC. Bei einem μC sind dagegen die für bestimmte Einsatzbereiche optimierten Funktionsblöcke oft wichtiger als der Rechenkern, von dem es viele verschiedene gibt wie ARM Cortex-M, Tensilica Xtensa (ESP32), Microchip/Atmel AVR (Arduino), PIC und 8051. Weniger bekannt sind ARC, LatticeMico, NEC 78K, Renesas RL78, TI MSP430, RISC-V RV32I. Eine μC -Baureihe umfasst oft zahlreiche Varianten, etwa mit unterschiedlichen Taktfrequenzen, SRAM- und Flash-Kapazitäten, Schnittstellen und Zusatzfunktionen, aber auch zertifiziert für unterschiedliche Temperaturbereiche und Industriespezifikationen. Diese Spezialisierung macht das μC -Angebot sehr unübersichtlich. Zudem sind μC in vielen anderen Chips integriert: In WLAN-, Ethernet-, Bluetooth-, USB-Adaptern, in SSD- und Festplattencontrollern, sogar in x86-CPU von AMD und Intel, etwa für ein Trusted Platform Module (TPM 2.0).

(ciw@ct.de) 

Unterschiede zwischen Mikrocontrollern und Mikroprozessoren

Boards mit Mikrocontrollern

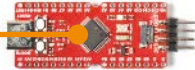
Arduino Uno
Microchip (Atmel) ATmega 328p
(1 × AVR, 8 Bit)



BBC Micro:Bit
Nordic Semi nRF51822
(1 × ARM Cortex-M0, 32 Bit)



Sipeed Longan Nano
GigaDevice GD32VF
(1 × RISC-V RV32I, 32 Bit)



Raspberry Pi Pico RP2040
RP2040
(2 × Cortex-M0+, 32 Bit)

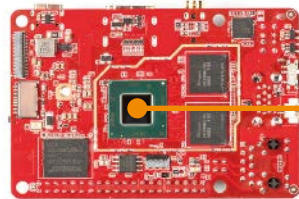


Boards mit Mikroprozessoren (und CPU)

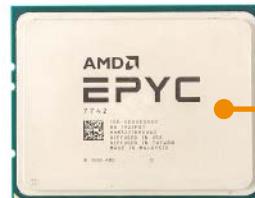
Raspberry Pi 4:
Broadcom BCM2837
(4 × ARM Cortex-A72)



Radxa Rock Pi X:
Intel Atom x5-Z8350
(4 × x86-64)



AMD Epyc 7742
(64 × x86-64)



Mikrocontroller und Mikroprozessoren: Typische Eigenschaften



Der NXP-Mikrocontroller Kinetis KL03 ist so winzig, dass er in die Delle eines Golfballs passt.

Eigenschaft	Mikrocontroller	Mikroprozessor
gängige Abkürzungen	µC, MCU (Micro Controller Unit)	CPU (Central Processing Unit), MPU (Micro Processing Unit)
Architektur Rechenkern	8, 16, 32 Bit	64 Bit (32-Bit-kompatibel)
Umfang Befehlssatz	eher klein	umfangreich
Anzahl Rechenkern	meist 1, mehrere möglich	1 bis über 100
Taktfrequenz Rechenkern	1 MHz bis über 1 GHz	500 MHz bis über 5 GHz
Leistungsaufnahme	wenige Mikrowatt bis circa 5 Watt	2 bis 500 Watt
adressierbares RAM	wenige Kilobyte bis einige Megabyte	mehrere Gigabyte bis Terabyte
Caches	selten	✓
eingebautes SRAM	einige Kilobyte bis Megabyte als Arbeitsspeicher	bis über 500 Megabyte als Cache
bindet externes DRAM an	manche	✓
Speichercontroller integriert	manche	✓ (für 1 bis 12 RAM-Kanäle)
Memory Management Unit (MMU) ¹	– (manchmal „MPU“ ²)	✓
typische Betriebssysteme	Echtzeit-OS oder gar keins	Windows, Linux, macOS, *BSD, z/OS, IBM i
optimiert für Echtzeit-Interrupts	✓	– (selten)
eingebauter Flash-Speicher für Firmware/Betriebssystem	oft	–
PCI Express Root Complex	–	✓ (für bis zu 128 PCIe-Lanes)
Analog-Digital-Wandler (A/D, D/A)	✓	– (bei Spezial-SoCs)
GPIO, UART, PWM integriert	✓	– (bei Spezial-SoCs)
SPI, I ² C integriert	✓	✓ (über Chipsatz)
Grafikkern integriert	– (bei Spezial-SoCs)	✓ bei PC-CPUs
Anzahl Transistoren	einige Tausend bis Millionen	oft über 1 Milliarde
Größe des Chips (Die)	wenige Quadratmillimeter	unter 100 bis über 800 Quadratmillimeter
als Funktionsblock in Systems-on-Chip integriert	häufig	selten
Stückzahl weltweit jährlich	über 20 Milliarden	2 – 2,5 Milliarden
Preisbereich	unter 10 Cent bis etwa 20 €	30 bis 30.000 €

¹ MMU verwaltet virtuellen Speicher, Caches, Zugriffsrechte, Ausführbarkeit (NX)

² Memory Protection Unit verwaltet Zugriffsrechte, Ausführbarkeit (NX)