

A close-up, blue-tinted photograph of a microcontroller circuit board. The image shows various electronic components, including a large integrated circuit (microcontroller) in the center, surrounded by smaller components like capacitors and resistors. The background is blurred, creating a bokeh effect with light spots.

Mikrocontroller

PRÄSENTATION VON YANNIS BECKER UND LUIS
SCHÄFER

Gliederung

- 1) Motivation
- 2) Grundlagen
- 3) Aufbau/Funktionsweise
- 4) Programmierung
- 5) Praktisches Beispiel
- 6) Fazit

Motivation

Grundlagen - Definition

Halbleiterchip mit Prozessor,
Peripherie und Speicher

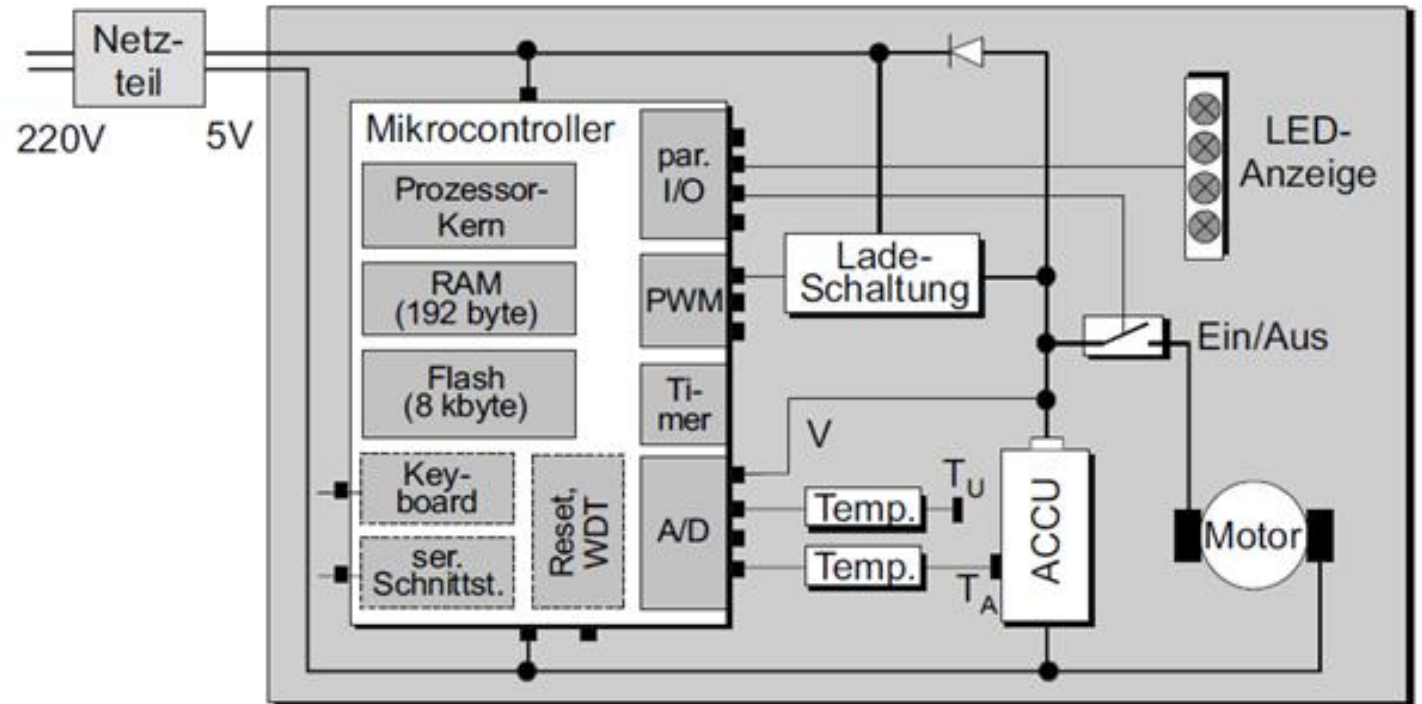


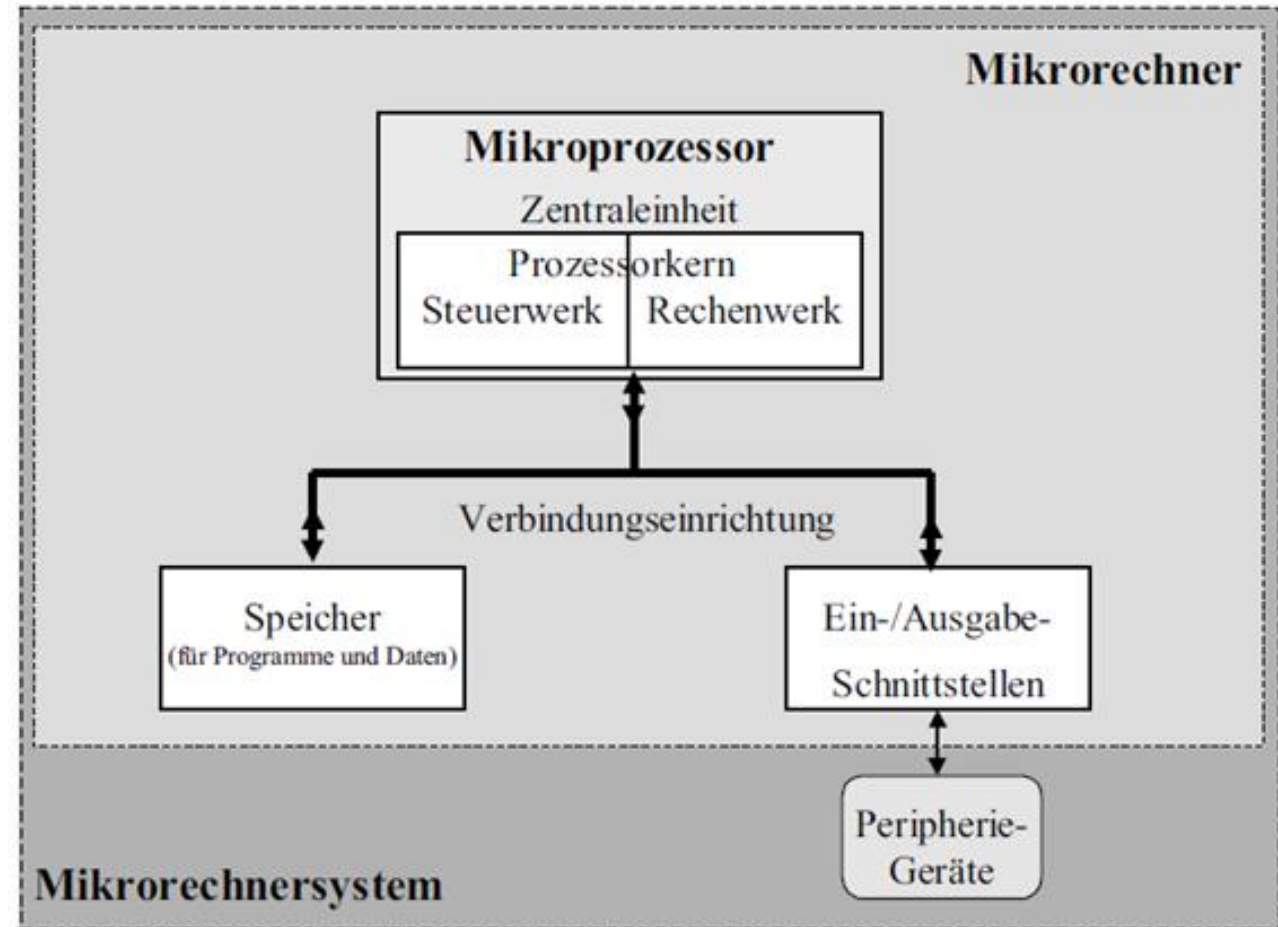
Abb. 2.2: Ein einfaches Mikrocontroller-System

/1/

Grundlagen – Unterscheidung Mikroprozessor & Mikrocontroller

Ein Mikroprozessor kann
ein Teil eines
Mikrocontrollers sein.

-> Ein Mikrocontroller
besitzt zusätzlich Speicher
und Ein- und
Ausgabeschnittstellen.



/2/

Abb. 1.1. Abgrenzung der Begriffe Mikroprozessor, Mikrorechner und Mikrorechnersystem

Grundlagen - Anwendungsfelder

Welche Anwendungsgebiete von Mikrocontrollern sind euch bekannt?

Grundlagen - Anwendungsfelder

Medizintechnik, Automatik und Robotik uvm.

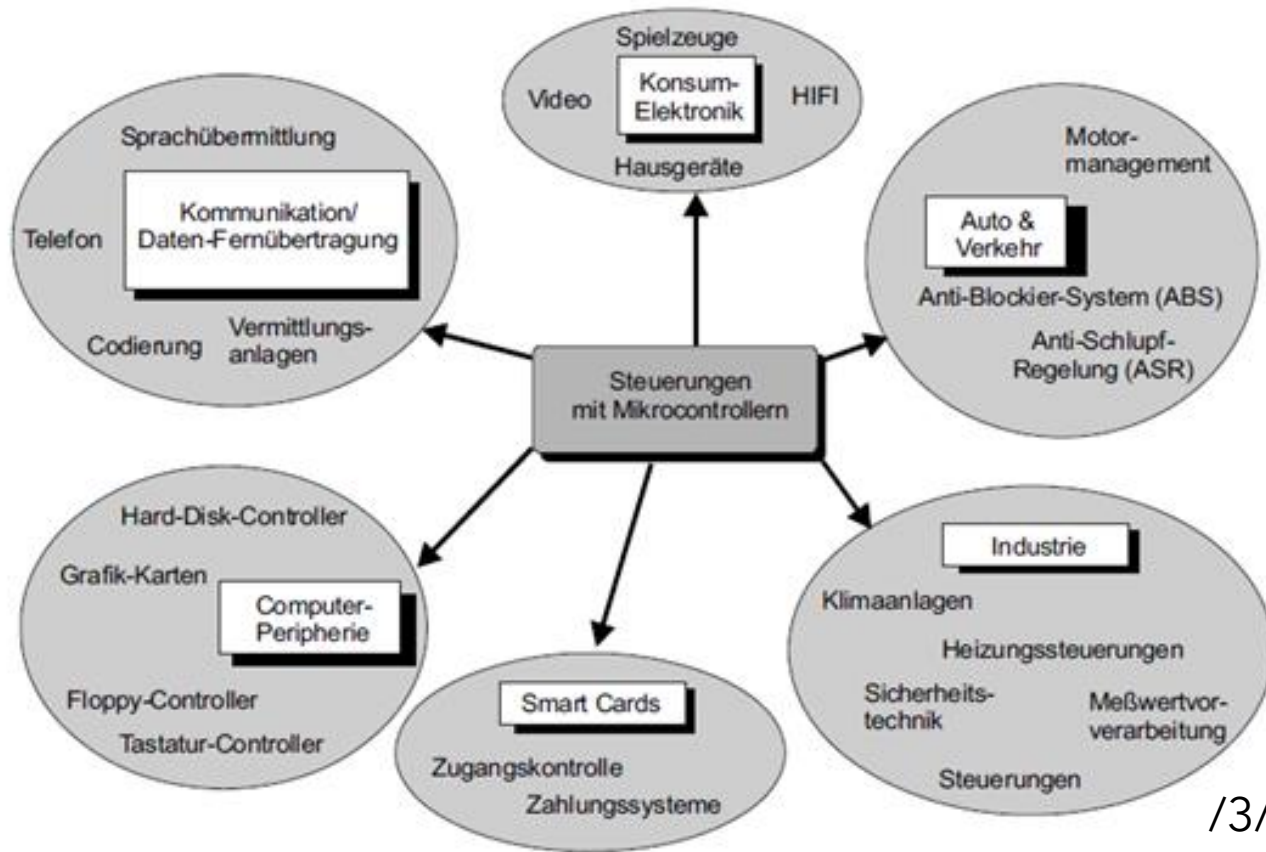


Abb. 2.5: Haupteinsatzgebiete für Mikrocontroller

Grundlagen - Anwendungsfelder

Automobil zur
detailreicheren
Betrachtung.

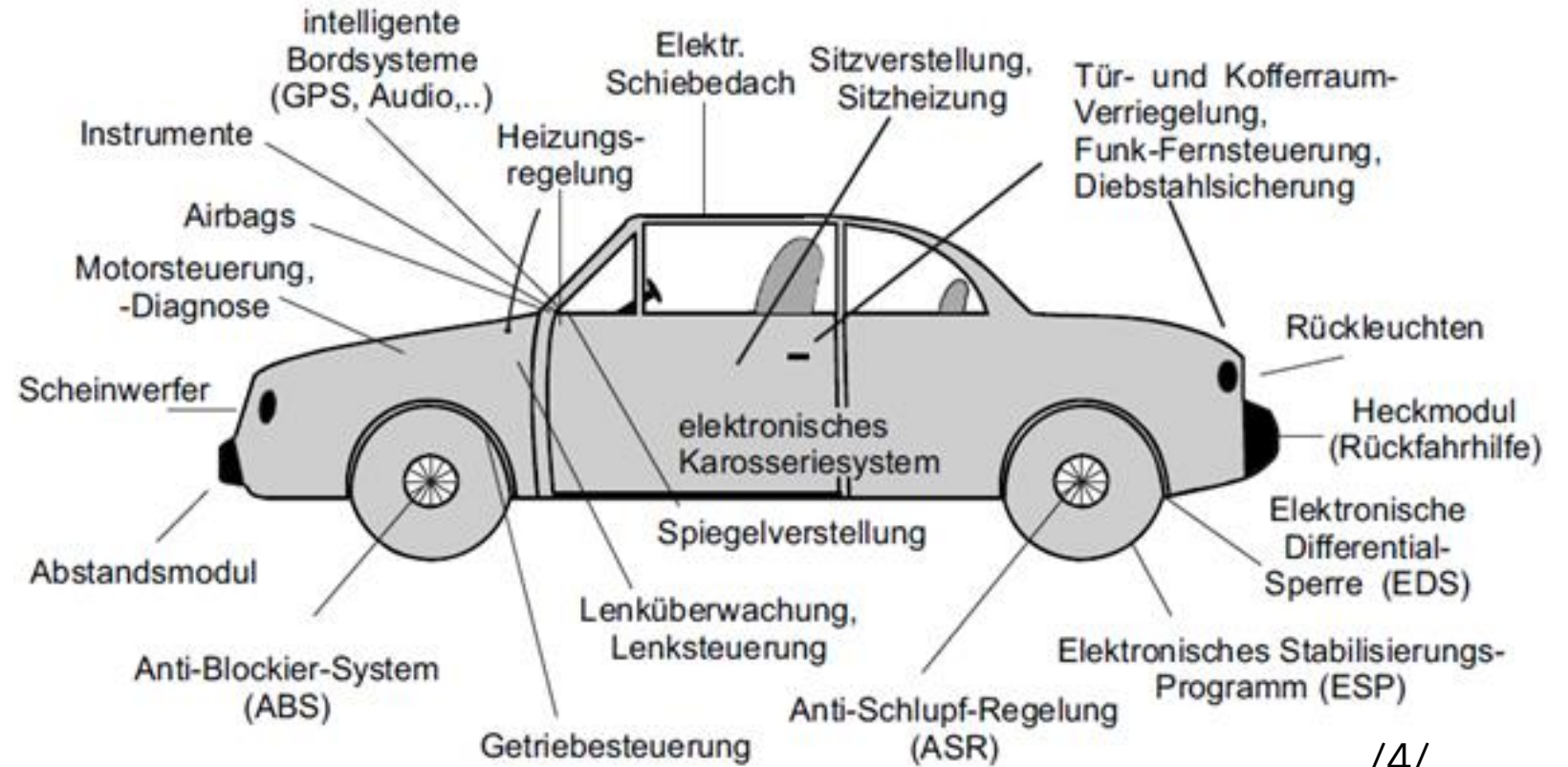
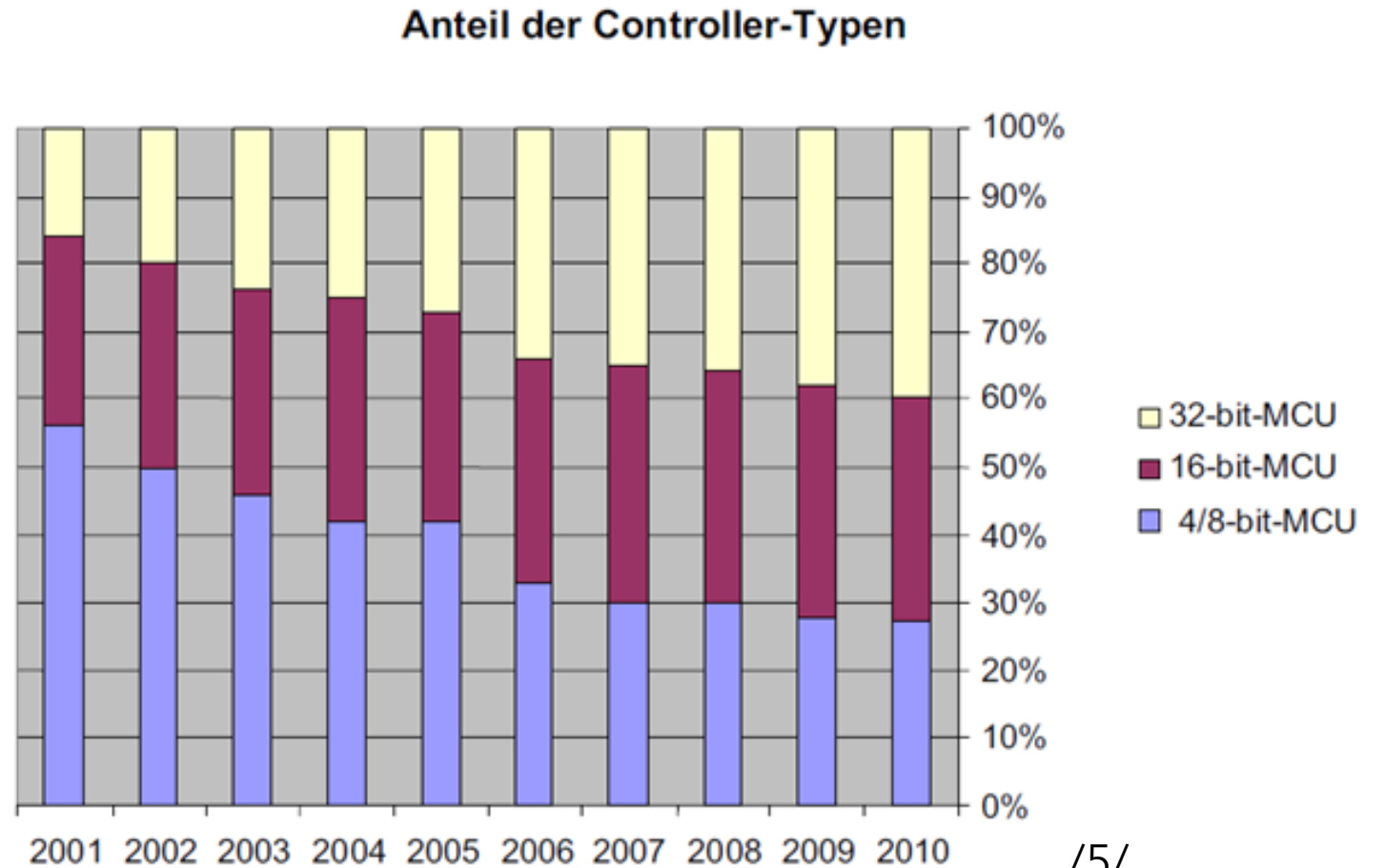


Abb. 2.6: Einsatz von Mikrocontrollern in einem Automobil

Grundlagen - Leistungsklassen

Leistungsklassen werden nach Datenbreiten der Prozessoren eingeteilt (4, 8, 16 & 32 Bit).



Grundlagen - Mikrocontrollerfamilien

Tabelle 3.3. Atmel Mikrocontrollerfamilien

Familie	Festwert- speicher (KBytes)	Schreiblese- -speicher (Bytes)	Zeit- geber (Anzahl)	Serielle E/A (Anzahl)	AD-Wandler (Anzahl x Bit)	Takt (MHz)
<i>Atmel</i>						
<i>8 Bit</i>						
AVR 8	1 - 256	128 - 8k	2	1 - 2	(1 - 16) x 10	16 - 20
MSC51 komp.	2 - 128	128 - 8k	2 - 4	0 - 3	(0-8) x (8-0)	16 - 60
<i>Atmel</i>						
<i>8/16 Bit</i>						
AVR 8/16	16 - 384	2k - 32k	4-8	4 - 12	(12 - 16) x 12	16 - 20
<i>Atmel</i>						
<i>32 Bit</i>						
AVR 32	16 - 512	128k	6	10	1 x 16	66
ARM	0 - 2048	8K - 256K	3 - 9	4 - 6	(8-16) x (10-12)	33-400

/6/

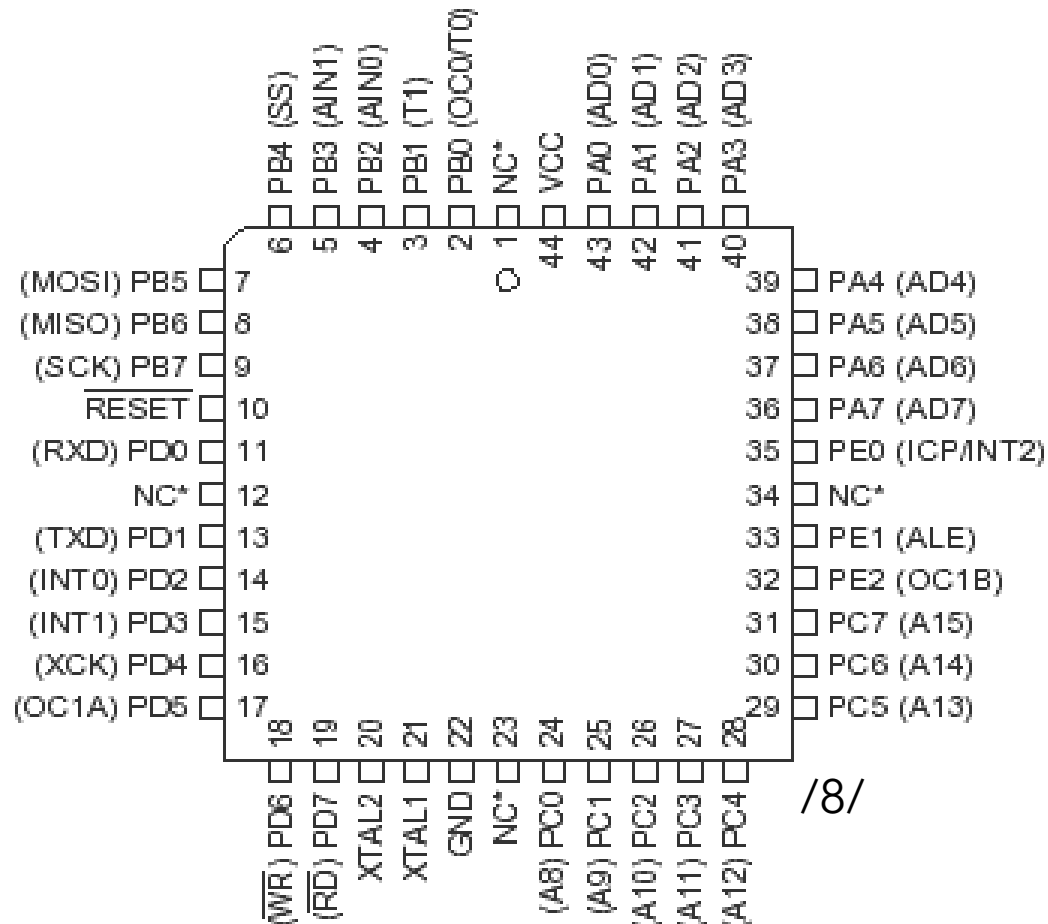
Grundlagen - Mikrocontrollerfamilien

Tabelle 3.4. Die Mitglieder der AVR8 Familie

ATtiny1X 2X 4X 8X	Grundmodelle mit 1, 2, 4 oder 8 kBytes ROM. Je nach X unterschiedliche Versionen, z.B. ATtiny11 mit 6 MHz, ATtiny12 mit 8 MHz und ATtiny13 mit 20 MHz Taktfrequenz.
ATmega8X 16X 32X 64X 128X 256X	Erweiterte Modelle mit 8 bis 256 kBytes ROM. Je nach X unterschiedliche Varianten, z.B. ATmega128A mit 4 kBytes RAM und ATmega1280 mit 8 kBytes RAM.
AT90X	Spezialversionen, z.B. AT90CAN128 mit CAN-Bus-Einheit oder AT90PWM1 mit Pulsweitenmodulator.
AT94X	Enthält programmierbare Logik (FPGA).

/7/

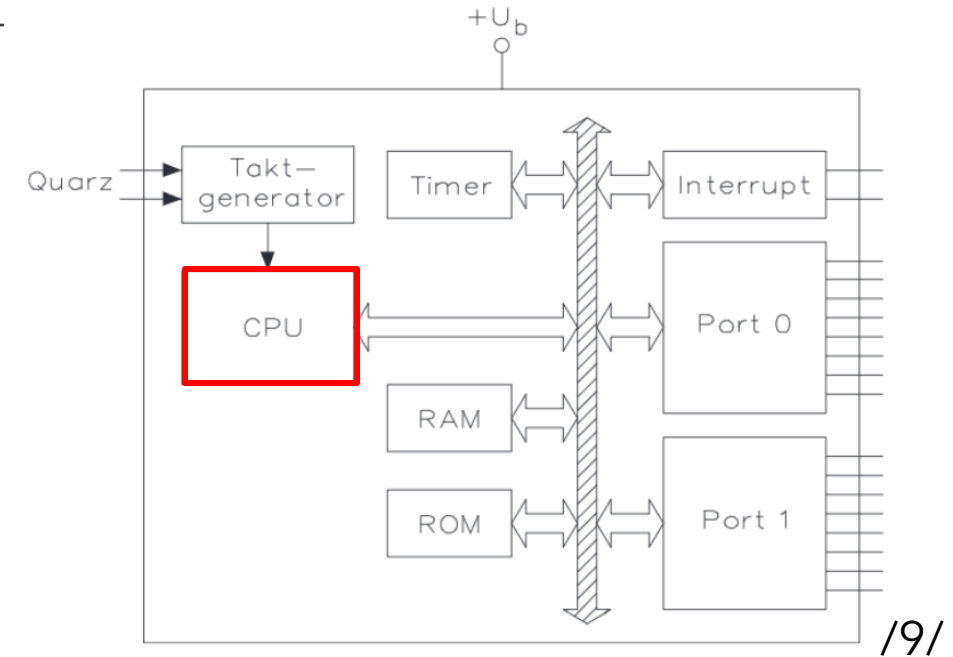
Grundlagen - Datenblätter



- Pinbelegung
- Kennzahlen
- Zeiteigenschaften
- Elektrische Eigenschaften
- ...

Aufbau - Prozessorkern

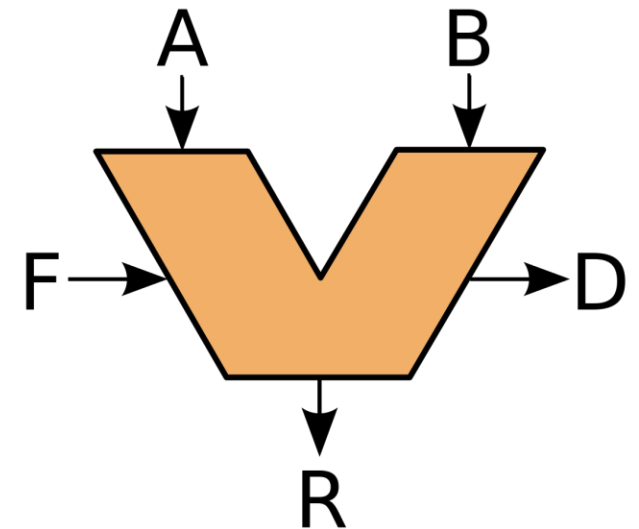
- bestimmt Befehlssatz, Daten- und Befehlsformate
- beinhaltet Rechen- und Steuerwerk, Register
- Verbindung mit allen anderen Komponenten des Mikrocontrollers über verschiedene Busse
- Ausführung von Befehlen → koordinierte Ansteuerung unterschiedlicher Komponenten durch Steuerwerk



Blockschaltbild eines vereinfachten Mikrocontrollers

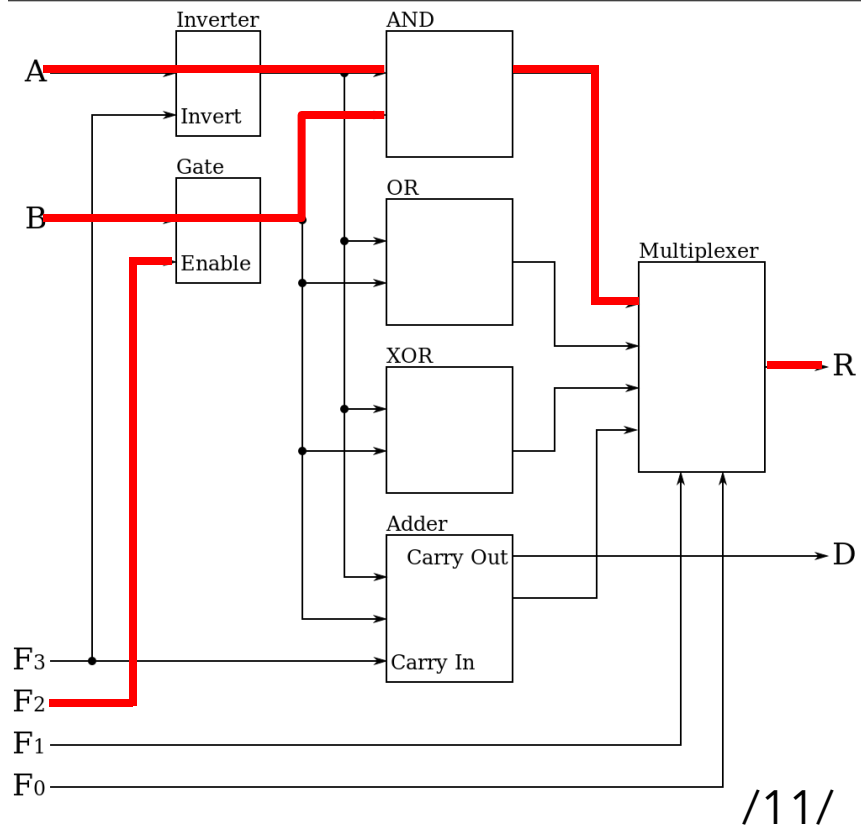
Aufbau - ALU

- **A**lgorithmic **L**ogic **U**nit
- Durchführung von mathematischen und logischen Operationen
- Register für Operatoren(F), Operanden(A,B), Ergebnis(R), Status(D)



Schema einer ALU /10/

Aufbau – einfache ALU



Einfache ALU

F_3	F_2	F_1	F_0	R
0	0	0	0	0
0	0	0	1	A
1	0	0	1	NOT A
0	1	0	0	A AND B
0	1	0	1	A OR B
1	1	1	0	A XOR B
0	1	1	1	A + B
1	1	1	1	B - A

Aufbau - Speicher

Programmspeicher (Flash)

- nicht flüchtig, beinhaltet Maschinencode des Programms
- unterteilt in Boot- und Anwendersektor
- Program Counter (=Zähler) zeigt auf aktuell auszuführende Speicherzelle
- ATmega32 z.B. 32 Kbyte

Datenspeicher (SRAM)

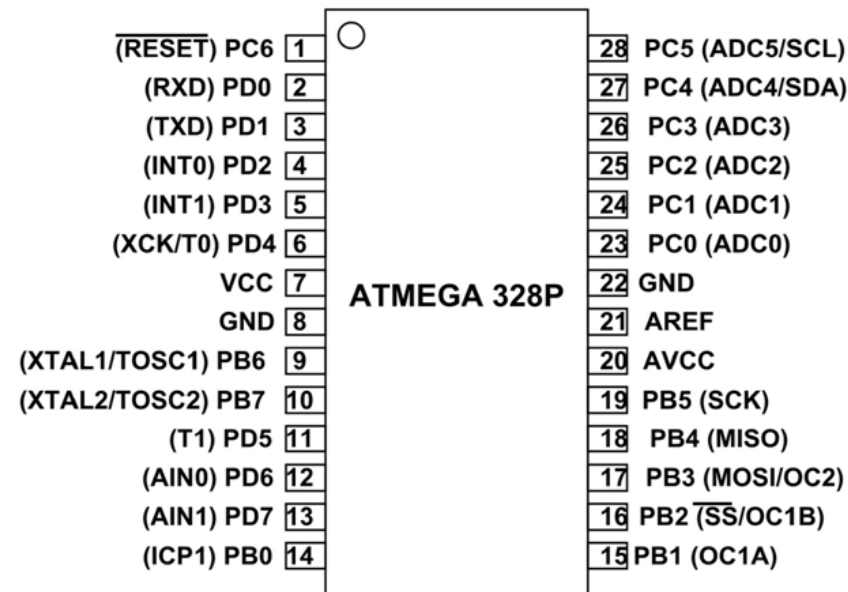
- flüchtig, Speicher für Werte der Programm-Variablen
- nutzt i.d.R. bistabile Kippstufen (geringe Stromaufnahme beim Halten)
- ATmega32 z.B. 2 Kbyte

EEPROM

- nicht flüchtig, vom Programm les- und schreibbar
- ATmega32 z.B. 2 Kbyte

Aufbau - Ein-/Ausgabeeinheiten

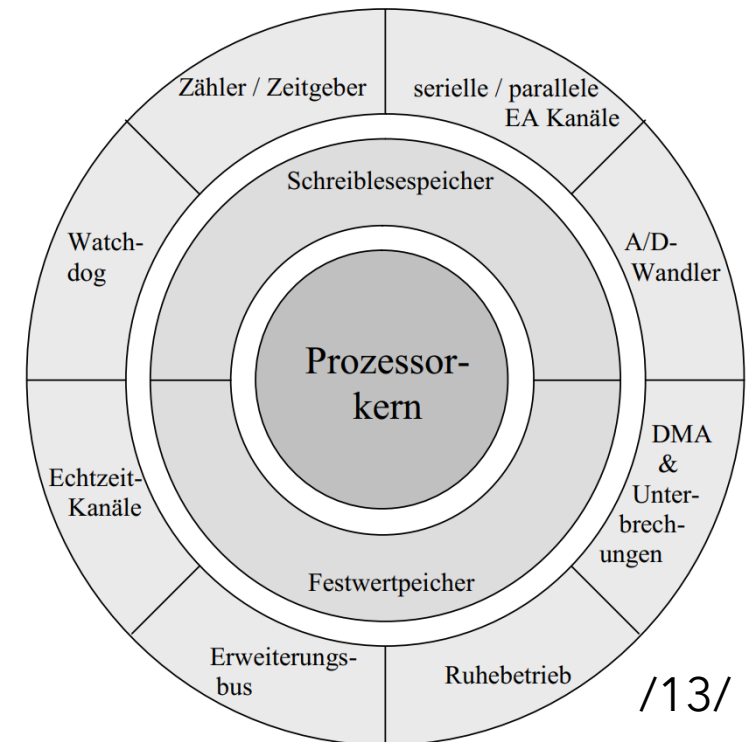
- Schnittstelle des Mikrocontrollers zu seiner Umwelt
- Übertragung digitaler (0/1) oder analoger Signale (V)
- Anbindung an den Prozessor über eigenen Adressraum (Konfiguration, Ein-/Ausgabe)



/12/

Funktionsweise

- Prozessorkern = primäre Komponente
- enge Anbindung an Speicher
- Prozessorkern koordiniert Ausführung von Befehlen durch Ansteuerung unterschiedlicher Komponenten
- E/A-Einheiten = Schnittstelle zur Umwelt
- diverse weitere Komponenten für spezifische Funktionen



/13/

Programmierung - Checkliste

- ✓ Computer
- ✓ Entwicklungsumgebung
- ✓ Programmierer (bzw. Entwicklungsboard)
- ✓ USB-Kabel sowie Kabel um Ein- und Ausgabegeräte zu verbinden
- ✓ Programmiersprache

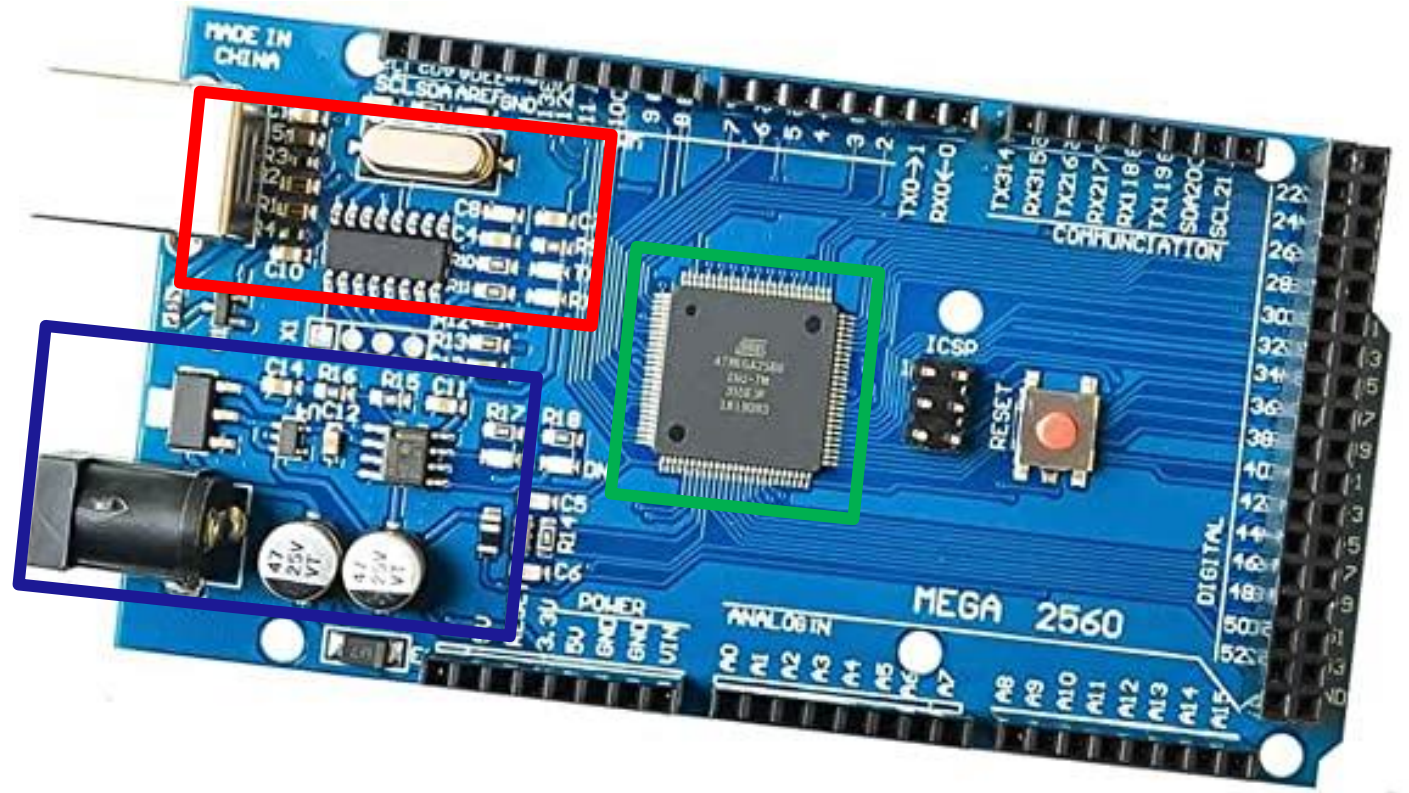
Programmierung - Entwicklerboard & Programmer

Rot: Programmer

Grün: Mikrocontroller

Blau: Stromversorgung

USB-Port, PINs



/14/

Programmierung - Programmiersprache

- Assembler
- C und auch C++
- weitere, nicht so gängige Optionen, wie MicroPython, Lua, ...
- Wahl der Sprache (insofern diese existiert) ist eine Wahl zwischen Programmierkomfort und technischer Effizienz

Programmierung – Besonderheiten

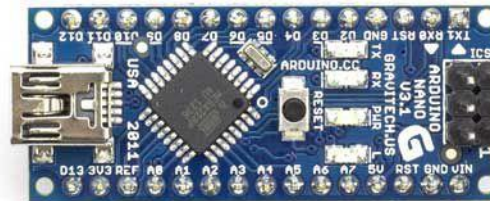
- verfügbarer Speicher ist eine knappe Ressource
- Adressfestlegung beim Binden
- Testen mittels Simulatoren
- Laden des ausführbaren Programms auf den Mikrocontroller
 - Mithilfe eines Monitorprogrammes
 - In den Festwertspeicher programmieren

Praktisches Beispiel



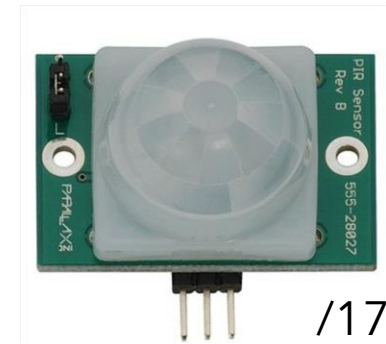
/15/

LED-Streifen

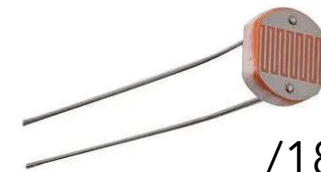


/16/

Mikrocontroller
(Arduino Nano)



/17/



/18/

Bewegungs- &
Helligkeitssensor

Praktisches Beispiel - Arduino

- quelloffene Soft- und Hardware-Plattform
- einfache Entwicklerboards in verschiedenen Bauformen
- Entwicklungswerkzeuge für Programmierung
- Einsteigerfreundlich, gute Dokumentation



Praktisches Beispiel - Voraussetzungen

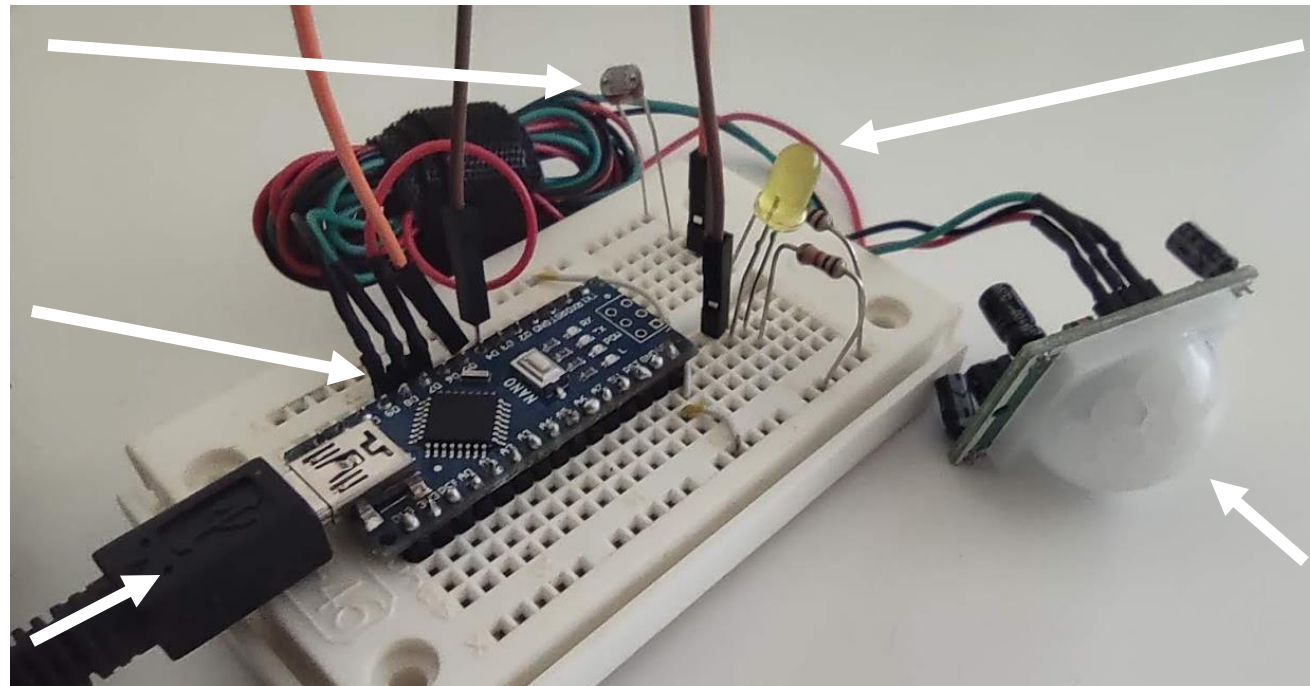
- ✓ Mikrocontroller (Arduino Nano Entwicklerboard)
- ✓ PC
- ✓ Arduino IDE (<https://www.arduino.cc/en/software>)
- ✓ MiniUSB- Kabel
- ✓ Hardware (LED-Streifen, Sensoren, einige passive Komponenten, Drähte)

Praktisches Beispiel

Helligkeitssensor

Arduino Nano

PC



LED (-Streifen)

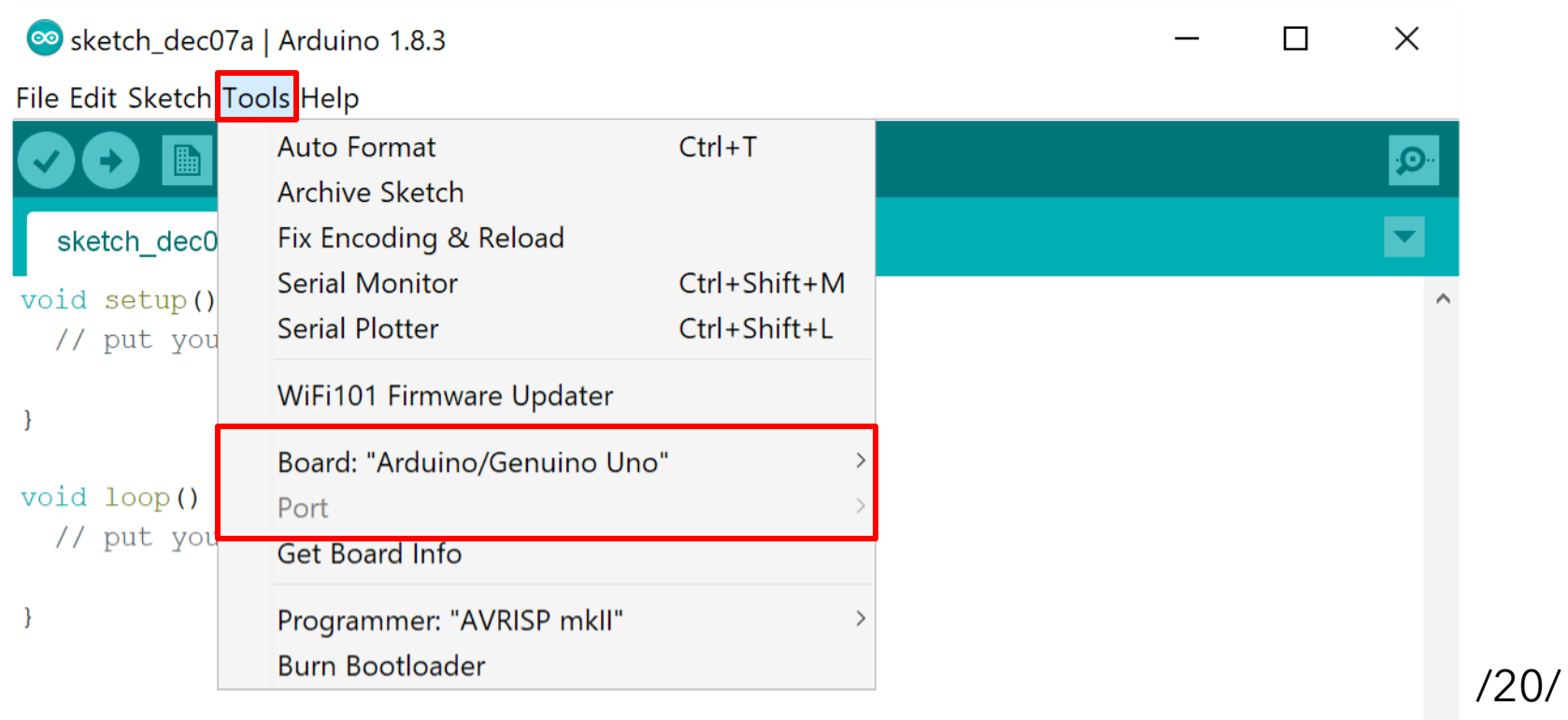
Bewegungssensor

Praktisches Beispiel - IDE



/20/

Praktisches Beispiel - IDE



Praktisches Beispiel - IDE



/20/

Praktisches Beispiel - Programmcode

```
long int lastMotion = 0; // Zeitpunkt letzter Bewegung
int timeout = 60 * 1000; // Einschaltdauer
bool enabled = false; // LEDs aktuell eingeschaltet?
int maxBrightness = 255; // Helligkeit der LEDs

void setup() {
    pinMode(2, INPUT); // Bewegungssensor
    pinMode(3, INPUT); // Helligkeitssensor
    pinMode(4, OUTPUT); // LED-Streifen
}
```

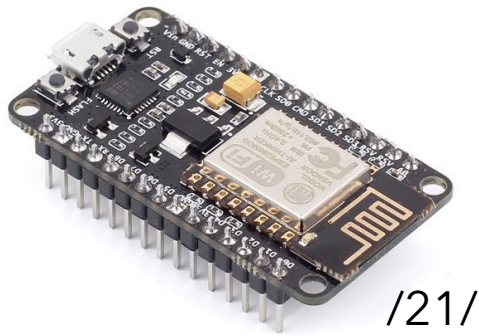
Praktisches Beispiel - Programmcode

```
void loop() {  
    int motion = digitalRead(2); // Bewegungssensor (0/1)  
    int light = analogRead(3); // Helligkeitssensor (0...255)  
    if(motion){ // Bewegung wurde erkannt  
        lastMotion = millis();  
        if(!enabled){  
            maxBrightness = 255 - (255 - light);  
            for(int brightness = 0; brightness <= maxBrightness; brightness++){  
                analogWrite(4, brightness); delay(5);  
            }  
            enabled = true;  
        }  
        // Licht nach einer bestimmten Zeit ausschalten  
        if(enabled && millis()-lastMotion > timeout){  
            for(int brightness = maxBrightness; brightness >= 0; brightness--){  
                analogWrite(4, brightness); delay(5);  
            }  
            enabled = false;  
        }  
    }  
}
```

Praktisches Beispiel - Ergebnis



Praktisches Beispiel - Ausblick



ESP8266

/21/



Arduino Nano

/16/



ATtiny45

/22/

Fazit

Bildquellen

1. Helmut Bähring: „Anwendungsorientierte Mikroprozessoren Mikrocontroller und Digitale Signalprozessoren“ , S. 12
2. Uwe Brinkschulte, Theo Ungerer: „Mikrocontroller und Mikroprozessoren“, S.2
3. Helmut Bähring: „Anwendungsorientierte Mikroprozessoren Mikrocontroller und Digitale Signalprozessoren“ , S. 14
4. Ebenda, S. 15
5. Ebenda, S. 16
6. Uwe Brinkschulte, Theo Ungerer: „Mikrocontroller und Mikroprozessoren“, S. 95
7. Ebenda, S. 96
8. <https://www.futurlec.com/Pictures/ATMega8515.gif>
9. Herbert Bernstein: „Mikrocontroller Grundlagen der Hard- und Software der Mikrocontroller ATtiny2313, ATtiny26 und ATmega32“, S. 3
10. https://upload.wikimedia.org/wikipedia/commons/thumb/8/82/ALU_symbol.svg/1195px-ALU_symbol.svg.png

Bildquellen

11. <https://upload.wikimedia.org/wikipedia/commons/thumb/1/17/Simplified-ALU.svg/1075px-Simplified-ALU.svg.png>
12. <https://www.elecparts101.com/atmega328p-datasheet-and-pinout-low-power-8-bit-atmel-microcontroller/>
13. Uwe Brinkschulte, Theo Ungerer: „Mikrocontroller und Mikroprozessoren“, S. 67
14. https://images-na.ssl-images-amazon.com/images/I/618IHEtRKEL._AC_SY355_.jpg
15. <https://ae01.alicdn.com/kf/H187878cb192a4d3e9de17298850c5b47j.jpg>
16. https://www.elektor.de/media/catalog/product/cache/23126780d17198beb71f0dbe2a3ae59a/a/r/arduinonanofront_3_sm_1.jpg
17. https://www.robotshop.com/media/catalog/product/cache/image/1350x/9df78eab33525d08d6e5fb8d27136e95/p/a/parallax-pir-motion-sensor_1.jpg

Bildquellen

18. <https://opencircuit.shop/resources/content/870f2b37c98bb/crop/400-300/GL12528-LDR-Lichtgevoelige-weerstand.webp>
19. https://upload.wikimedia.org/wikipedia/commons/thumb/8/87/Arduino_Logo.svg/1024px-Arduino_Logo.svg.png
20. <https://support.microsoft.com/de-de/topic/hochladen-von-brett-code-und-arduino-ide-a9723765-1314-49e0-a69b-bb5c3e1f628d>
21. <https://cdn.antratek.nl/media/product/5df/nodemcu-v2-lua-based-esp8266-development-kit-16608-42d.jpg>
22. https://cdn-reichelt.de/bilder/web/xxl_ws/A200/ATTINY45-PDIP-8.png

Quellen

Internet

- [https://de.wikipedia.org/wiki/Arduino_\(Plattform\)](https://de.wikipedia.org/wiki/Arduino_(Plattform))
- https://de.wikipedia.org/wiki/Arithmetisch-logische_Einheit
- <http://einsteiger.myavr.de/index.php?id=3>
- <https://www.exp-tech.de/mikrocontroller-eine-einfuehrung>
- <https://de.rs-online.com/web/generalDisplay.html?id=ideen-und-tipps/mikrocontroller-leitfaden>
- <https://www.mikrocontroller.net/>

Buch

- „Anwendungsorientierte Mikroprozessoren Mikrocontroller und Digitale Signalprozessoren“ von Helmut Bähning ISBN: 978-3-642-12291-0
- „Mikrocontroller Praxis Ein praxisorientierter Leitfaden für Hard- und Software-Entwicklung auf der Basis der 80(C)51x-Familie“ von Norbert Heesel, Werner Reichstein ISBN: 978-3-528-05366-6
- „Mikrocontroller und Mikroprozessoren“ von Uwe Brinkschulte, Theo Ungerer ISBN: 978-3-642-05397-9
- „Kompaktkurs Mikrocontroller Grundlagen Schaltungstechnik, Aufbau und Programmierung eines 8051-Systems, Kommunikation mit Windows-Rechnern, Debugging“ von Silvia Limbach ISBN: 978-3-528-05788-6
- „Mikrocontroller Grundlagen der Hard- und Software der Mikrocontroller ATtiny2313, ATtiny26 und ATmega32“ von Herbert Bernstein ISBN: 978-3-658-30066-1