Technische Hochschule Rosenheim
Technical University of Applied Sciences

Embedded Systems
Kapitel 1: Grundlagen

Prof. Dr. Wolfgang Mühlbauer
Fakultät für Informatik
wolfgang.muehlbauer@th-rosenheim.de

Sommersemester 2020

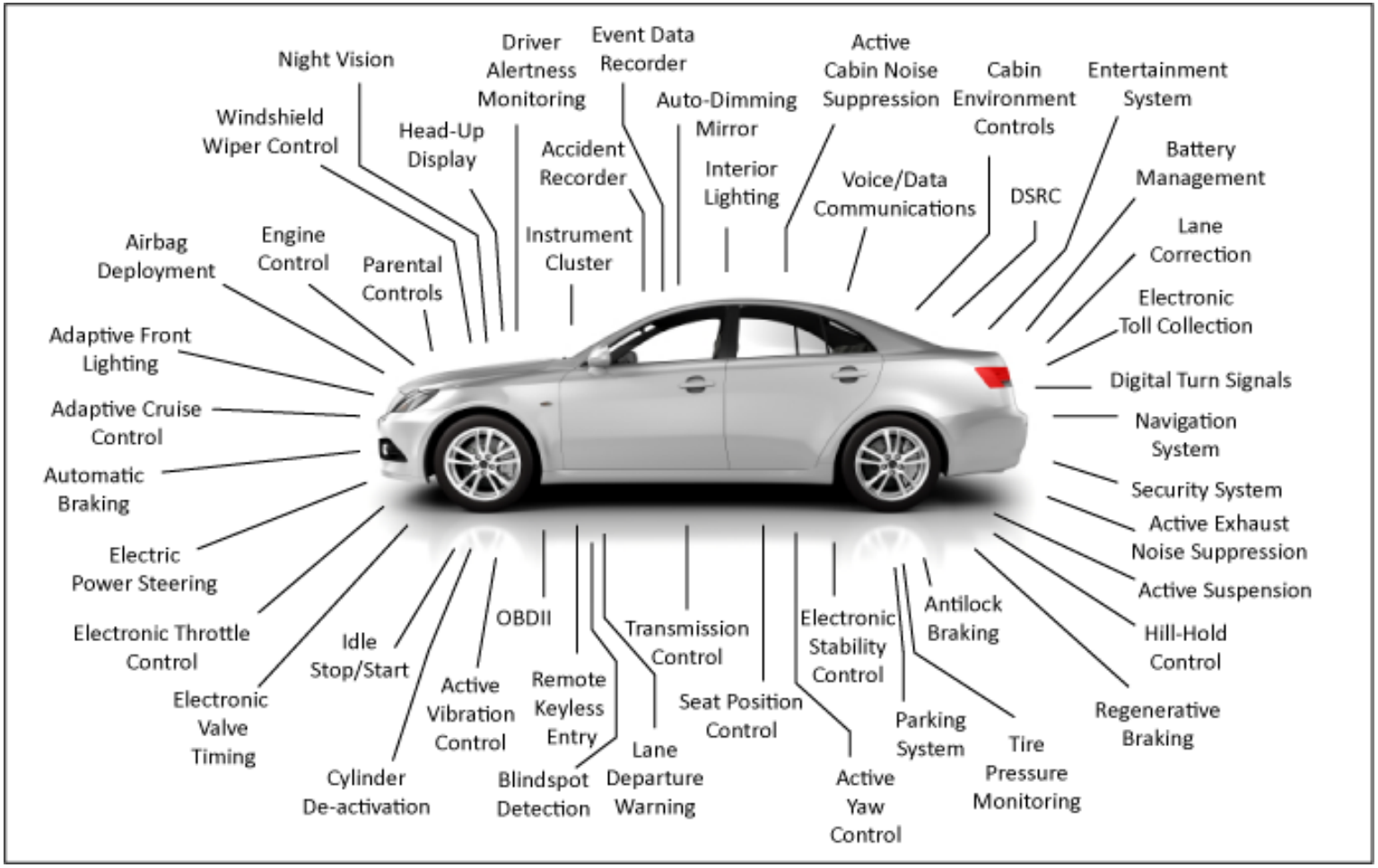
Eingebettetes System: Definition

Quelle: [4]

- *"An embedded system can be broadly defined as a device that contains **tightly coupled hardware and software** components to perform a **single function**, forms **part of a larger system**, is **not** intended to be independently **programmable by the user**, and is expected to work with **minimal or no human interaction**.* [1]

- *“An embedded system is a computer that does not look like a computer” (*D. Gajski, University of California)

Komplexe eingebettete Systeme



Quelle: [8]

In Autos arbeiten eine Vielzahl an Mikrocontroller zusammen!

Eingebettete Systeme: Anforderungen

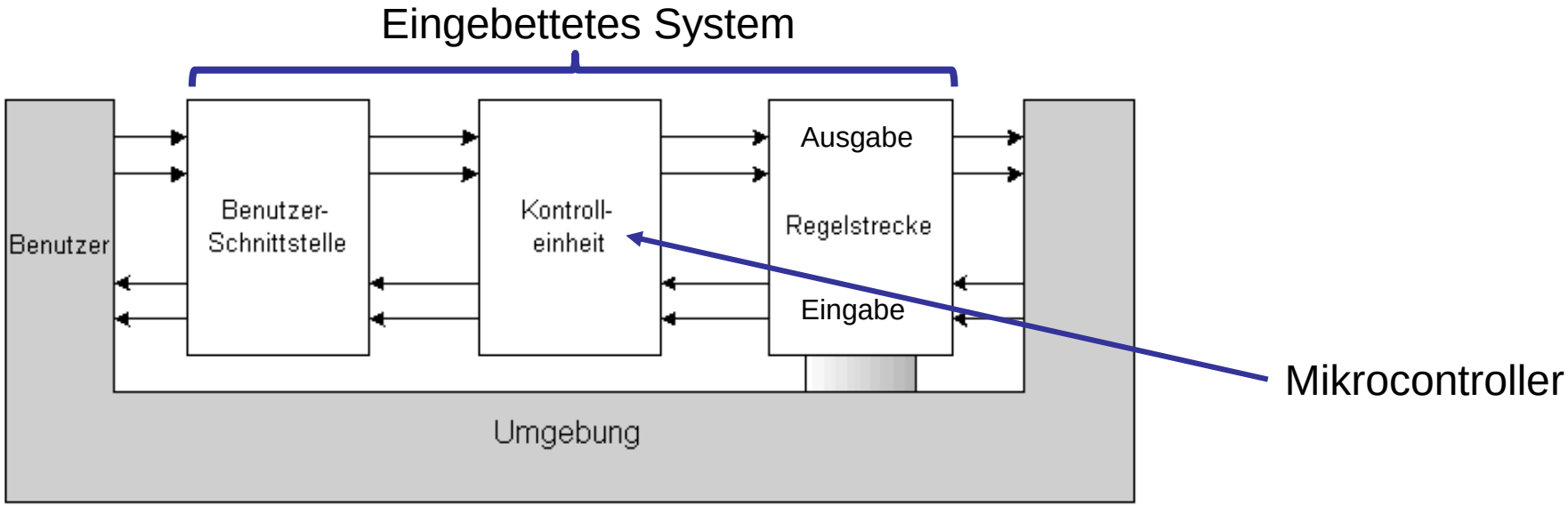
- ***Schnittstellen***
  - SPI, USB, Ethernet, I2C, …
- ***Mechanisch***
  - Größe
  - Robustheit gegen mechanische Belastungen,
  - Hitze, Kälte
- ***Elektrisch***
  - Geringer Energieverbrauch
- ***Zuverlässigkeit***
  - Geringe Ausfallwahrscheinlichkeit
  - Notbetrieb
- ***Echtzeit***
  - Ausführen von Aktionen innerhalb einer vorgegebenen Zeit

Der SW-Entwickler muss all diese Anforderungen beachten!

Aufbau eines eingebetteten Systems

- ***Hardware***
  - CPU, Speicher, Ein-/Ausgabe Ports, Timer, usw.
  - In der Praxis meist Mikrocontroller!

- ***Software***
  - Steuerungslogik bzw. Programm auf Hardware
  - Code in der Regel durch "Anwender" nicht veränderbar → **Firmware**!



[5]

(Programmierbare) Hardware

- ***Mikrocontroller***
  - Allgemeine, steuerungsorientierte Aufgaben
  - Integrierter Speicher, integrierte Schnittstellen zur Kommunikation mit Außenwelt
  - Geringe Performanz (ca. 1 µs/Instruktion)

- ***Digitale Signalprozessoren (DSP), anwendungsspezifische Prozessoren***
  - Spezialisierter Instruktionssatz, spezielle Funktionseinheiten.
  - Dominanz von Datenfluss (Sprachverarbeitung)
  - Komplexe, regelmäßige arithmetische Operationen (z.B. FFT)
  - Festkomma- und Gleitkommaarithmetik

- **Programmierbare Hardware (FPGA / CPLD)**
  - Field Programmable Gate Array
  - Programmierbare Schaltungen
  - Look-Up Tables (LUTS) realisieren beliebige n-stellige Binärfunktionen

- **Anwendungsspezifische integrierte Schaltung (ASIC)**
  - Kundenspezifischer Entwurf von integrierten Schaltungen
  - Interessant für hohe Stückzahlen

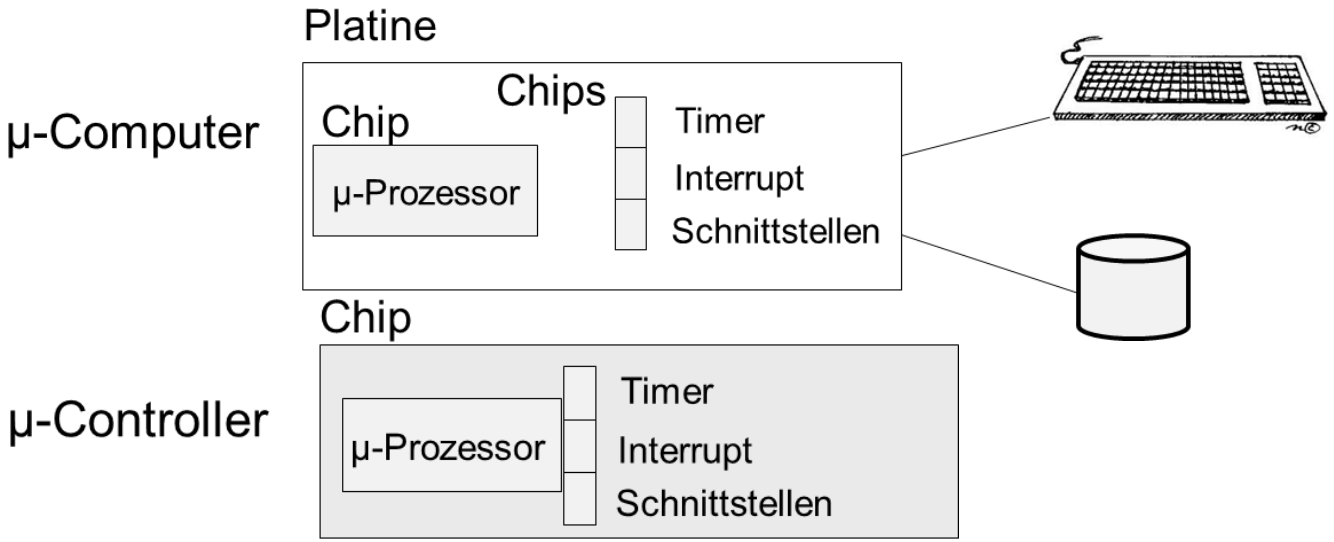
Übung

- Ordnen Sie die Komponenten bzgl. der Flexibilität bei der Realisierung eines eingebetteten Systems!
  - DSP
  - ASIC
  - Mikrocontroller
  - FPGA

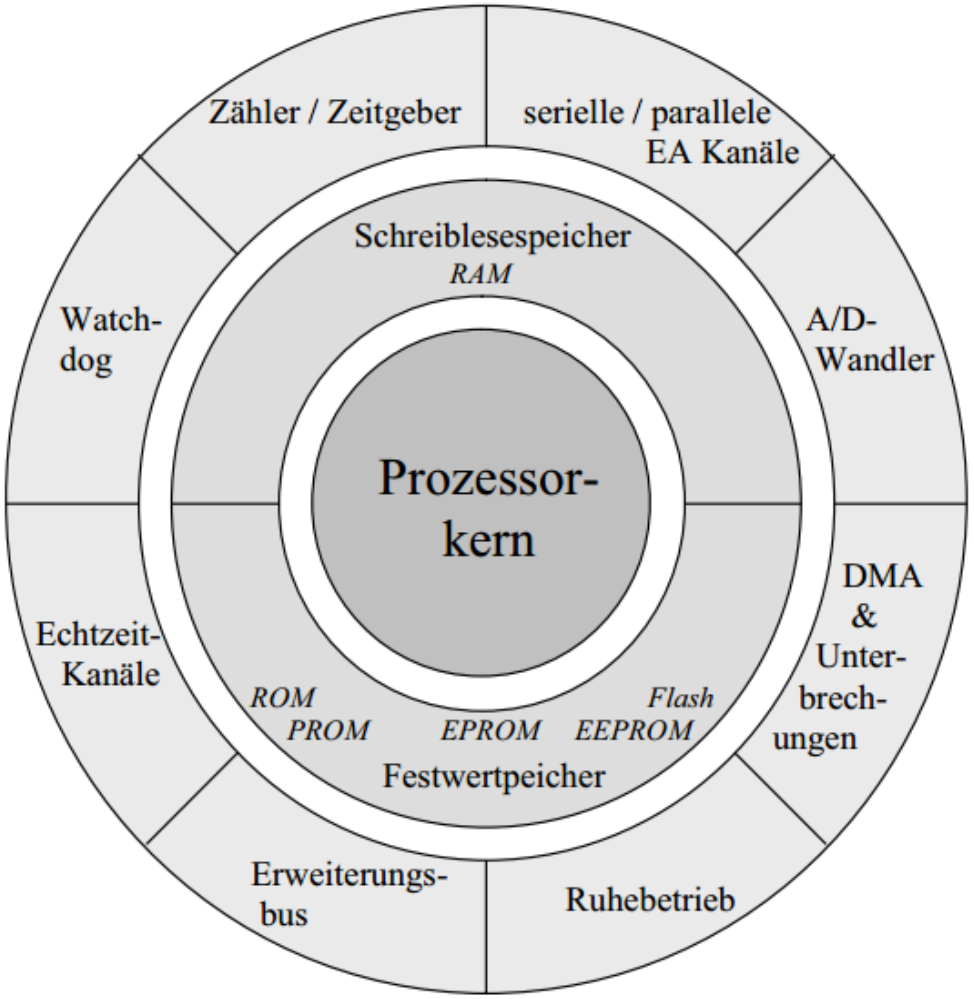
Mikrocontroller vs. Mikroprozessor

- ***Definition: Mikrocontroller***
  - *"Small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals." (Wikipedia)*

- ***Mikrocontroller = Mikroprozessor + Peripherie***
  - Peripherie (Speicher, Schnittstellen, Timer, usw.) auf gleichem Chip wie CPU.
  - µController benötigt weniger Signalleitungen nach außen als Mikroprozessor.



Aufbau eines typischen Mikrocontrollers

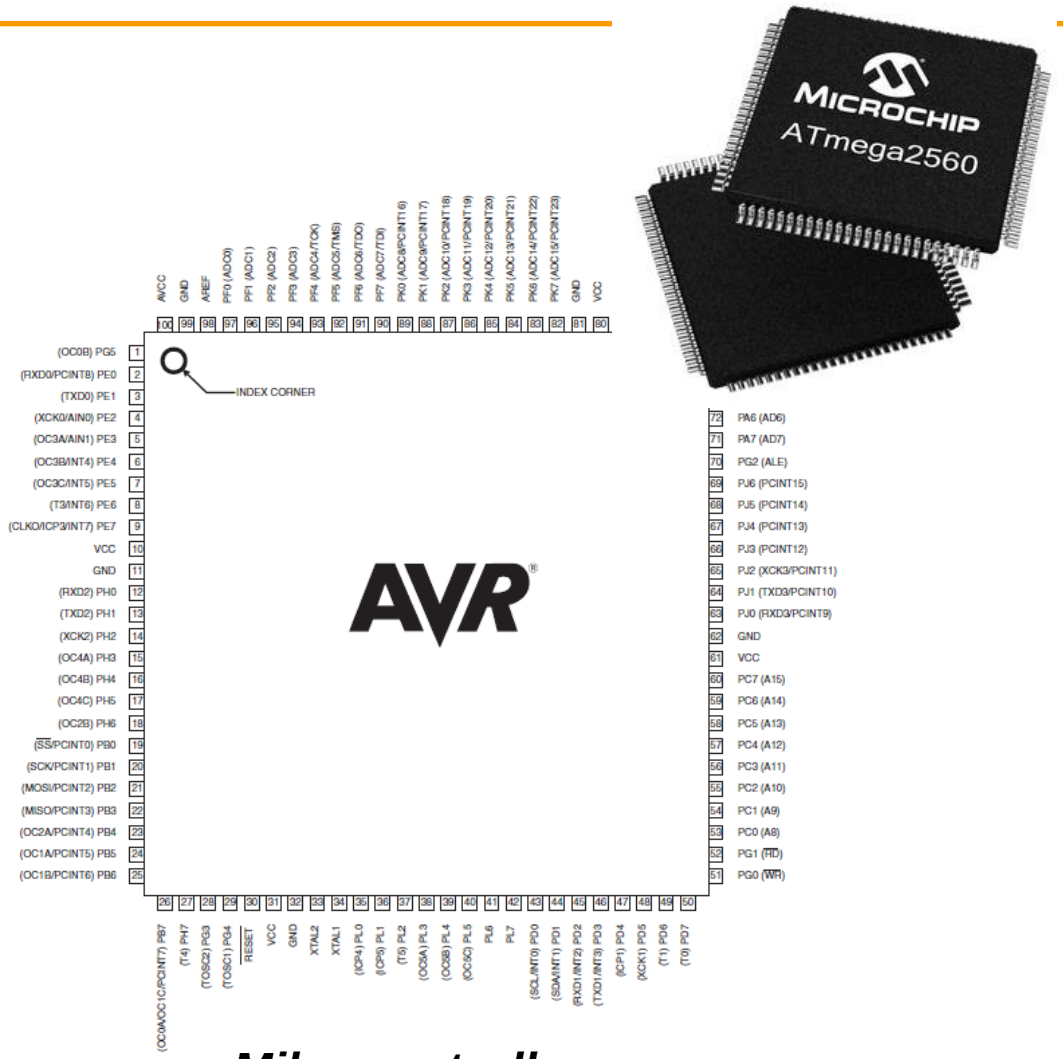


Mikrocontroller enthält CPU + zahlreiche Peripheriekomponenten (Quelle [2]).

- **Speicher**
  - Programme und Daten
  - ROM and RAM
  - Flüchtig vs. nicht-flüchtig
- **Zähler (Timer)**
  - Messen bzw. Abwarten von festen Zeiten
- **Unterbrechungen (Interrupts)**
  - Asynchrone Ereignisse
- **Schnittstellen**
  - Ein- und Ausgabe Pins
  - Serielle, parallele Kommunikation mit anderen Komponenten
- **Watchdog**
  - Überwachung der korrekten Funktionsweise andere Komponenten
- …

Mikrocontroller ATmega2560 von Microchip/Atmel

- 8-Bit Architektur
- Programmspeicher: 256K
- CPU Geschwindigkeit: 16 MIPS
- Flash, EEPROM, RAM
- JTAG
- Timer/Counter
- PWM
- UART, SPI, I2C
- Watchdog
- Interrupts
- Keine Memory Management Unit (MMU)
- Kein Direct Memory Access (DMA)

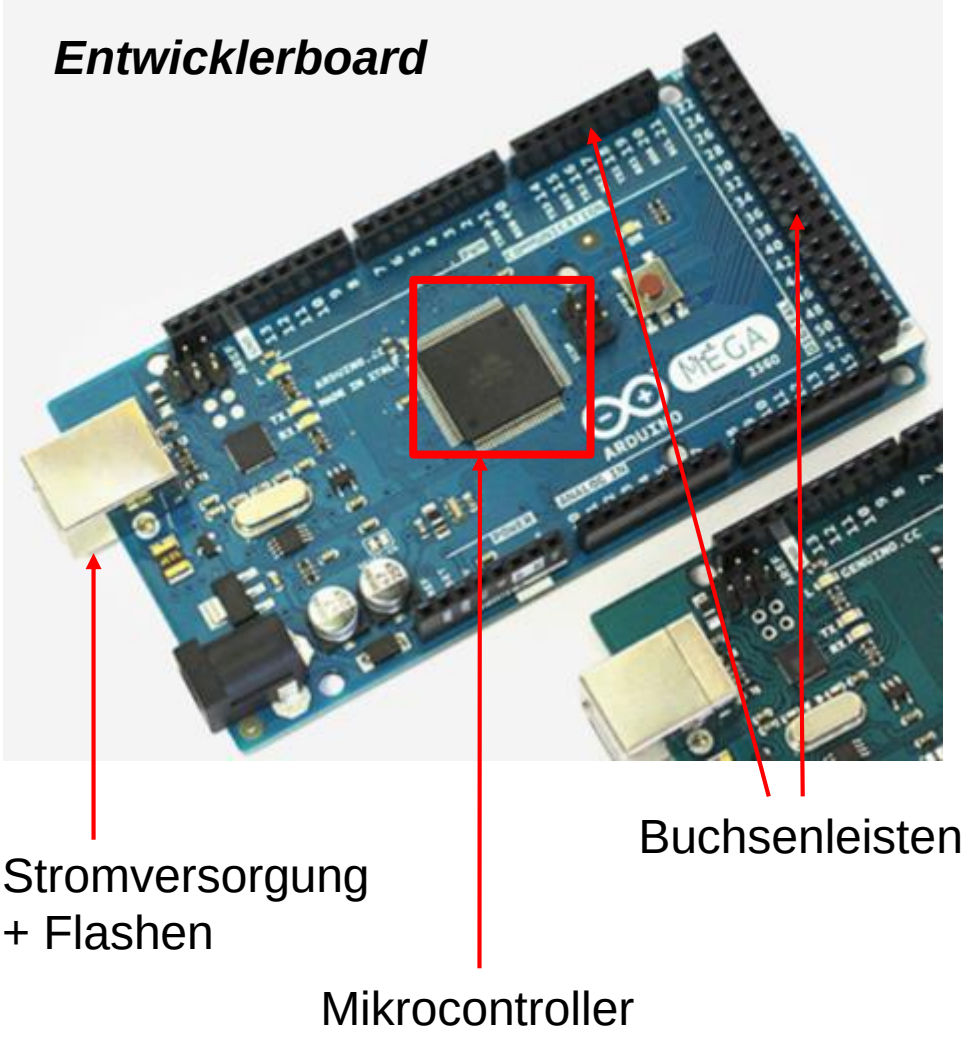


Mikrocontroller

Datenblatt: siehe Learning Campus oder [7]

Entwicklerboard: Arduino Mega

- Entwicklerboard Arduino Mega
  - https://www.arduino.cc/en/Main/ArduinoBoardMega2560
- Nicht verwechseln!
  - ***Entwicklerboard:*** Arduino Mega
  - ***Mikroprozessor***: ATmega2560
- Warum Entwicklerboard?
  - Einfaches Laden („Flashen“) von Programmcode
  - Stromversorgung durch USB des Computers
  - Einfacher Zugriff auf Pins des Mikrocontrollers über Buchsenleisten
  - Verfügbarkeit von Erweiterungsboards ("Shields")



Mikrocontrollerprogramm: Aufbau

- ***Sketch***
  - == Mikrocontrollerprogramm für Arduino
  - Code steht in einer *.ino Datei.
  - Automatische Einbettung in C++-Programm, ohne dass es Nutzer merkt.
  - *Vorteil:* Einfaches Programmieren und einfacher HW-Zugriff.

- **`setup(.)`**
  - Einmalige Ausführung nachdem Mikrocontroller durch Strom versorgt wird.
  - Roter Knopf startet Mikrocontroller ebenfalls neu.
- **`loop(.)`**
  - Wird danach immer wieder (=Endlosschleife) ausgeführt.

```
void setup() {
  // put your setup code here, to
run once:
}

void loop() {
  // put your main code here, to
run repeatedly:
}
```

Aufbau eines Sketch

Was ist ein Sketch?

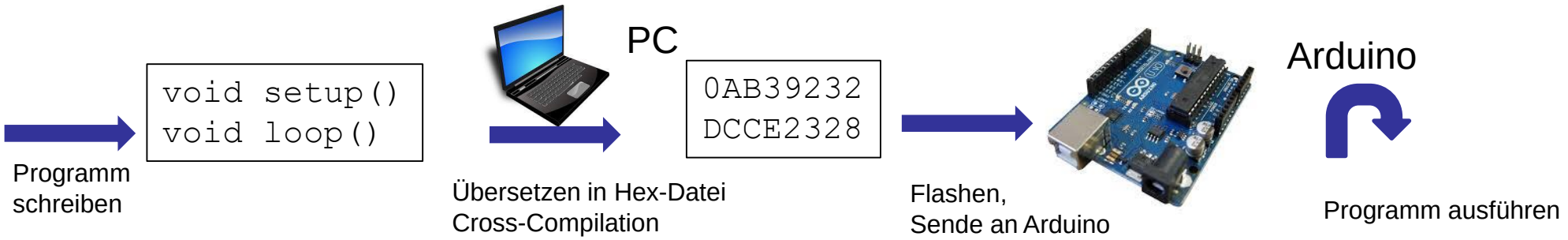
- Sketch == Arduino Programm
  - Vereinfacht Programmierung
  - C++-Bibliotheken, die HW-Programmierung leichter machen.
- Normalerweise sieht ein Programm wie rechts abgebildet aus:
  - Endlosschleife
  - Main-Methode.
  - *.cpp-Datei
- Die Arduino IDE verwendet unsichtbar eine main-Methode, in der `setup` und `loop` aufgerufen wird.
- Weiterführende Info:
  - https://github.com/arduino/Arduino/wiki/Build-Process

```
#include <avr/io.h>

int main(void)
{
    /* Replace with your application code */
    while (1)
    {
    }
}
```

So sieht ein Mikrocontrollerprogramm normalerweise aus (kein Sketch, *.cpp-Datei)!

Toolchain



- ***Programmieren***
  - Programmiersprache **C/C++,** selten direkt Assembler
  - Häufig in Entwicklungsumgebung, z.B. **Arduino IDE, Eclipse, CLion**

- ***Kompilieren, Cross-Compiling***
  - Programm wird auf anderer Plattform gebaut, nicht auf Zielplattform (= Mikrocontroller)
    Compiler: **avr-gcc,** häufig in Entwicklungsumgebung integriert.
  - Ergebnis: Bytecode, **hex**-Datei.

- ***SW Download / "Flashen"***
  - Sende hex-Datei von PC zu Mikrocontroller
  - Meist automatisch von Entwicklungsumgebung erledigt.
  - Kommandozeile: **avrdude**

Bibliotheken

- Bibliotheken erlauben den Zugriff auf Hardwarekomponenten des Mikrocontrollers.

- Er gibt beim Arduino Mega 2 Möglichkeiten.

- ***avr-libc:*** Standard C-Library des `avr-gcc`
  - Direkter Zugriff auf Hardware, Register, Timer etc. des Mikrocontrollers.
  - Beispiel: `_delay_ms` verwendet im Hintegrund einen Timer.
  - Komplexer und mächtiger als Arduino Library.
  - https://www.nongnu.org/avr-libc/

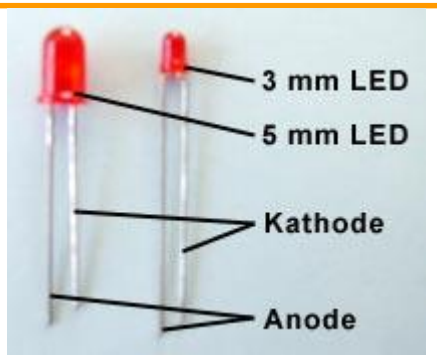
- ***Arduino Library / Sketch***
  - Vereinfacht Zugriff auf Hardware des Mikrocontrollers.
  - Kapselt `avr-libc`.
  - Beispiel: `digitalWrite(.)`
  - https://www.arduino.cc/reference/en/

Integrated Development Environments

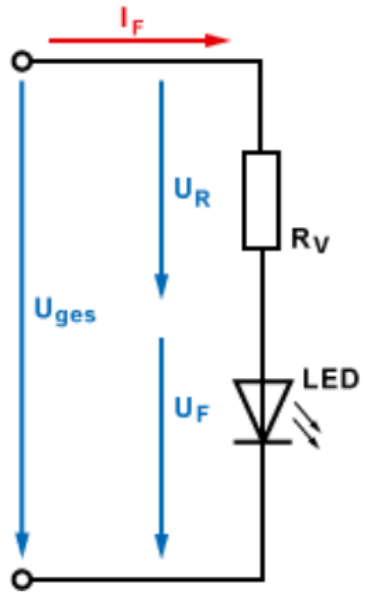
- ***Arduino IDE*** (im Labor vorhanden, Standardtool)
  - Genügt für Großteil der Übungen
  - Leider kein Syntax Highlighting
  - Enthält AVR-Libc
- ***Atmel Studio*** (im Labor vorhanden, Profitool)
  - Sketchprogrammierung nicht nativ unterstützt.
  - Erlaubt zusätzlich Simulation, JTAG Debugging, In-System Programming (siehe spätere Übung)
- ***Visual Studio***
  - Für Visual Studio Fans
- ***Eclipse mit AVR Eclipse Plugin***
  - "Schöner Editor"
  - https://www.eclipse.org/community/eclipse_newsletter/2017/april/article4.php
  - Empfehlenswert, siehe Aufgabe 1, Übung 4

Anschluss einer externen LED

- Light Emitting Diode
- Halbleiterdiode, die **nur(!)** in Durchlassrichtung Licht erzeugt
  - Langer Pin == Anode ("+")
- Je nach Material unterschiedliche Farben.
- LEDs vertragen nur bestimmten maximalen ***Durchflussstrom*** I_F bei einer ***Durchflussspannung*** U_F
  - Vorwiderstand R_F zur Begrenzung des Stroms notwendig.
  - $R_V = \frac{U_{ges} - U_F}{I_F}$



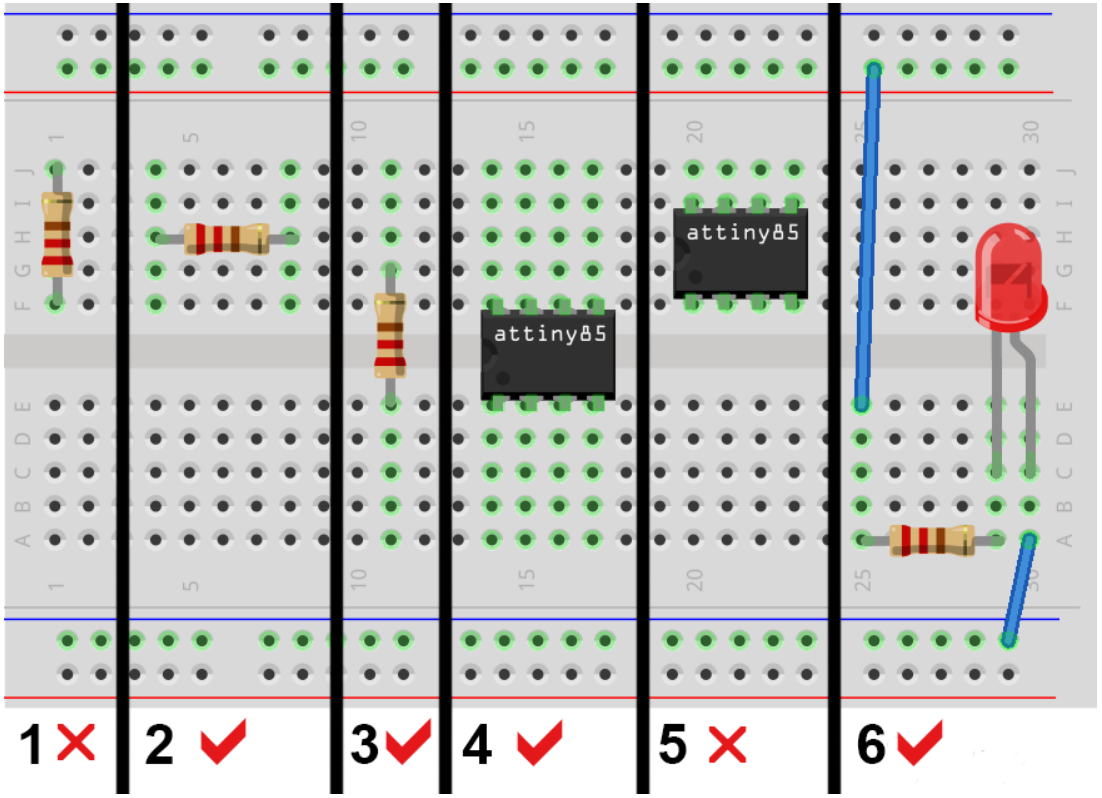
Quelle: [9]



Quelle: [9]

Steckbrett (engl. „Breadboard“)

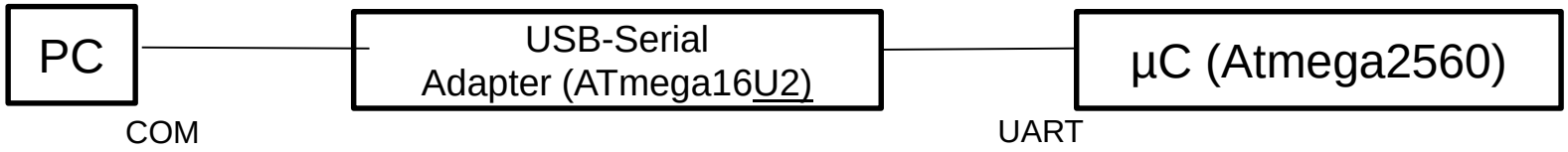
- Einfaches Testen von Schaltungen durch Einstecken von Bauteilen.
- Festes, genormtes Raster mit Lochabstand 2,45 mm.
- Manche Löcher sind leitend miteinander verbunden.
  - Fall 1 und 5 in rechter Grafiken zeigen, wie man Bauteile nicht einsteckt!



Quelle: [10]

Serieller Monitor

- Wie kann laufendes µC-Programm Textdaten an PC senden und empfangen?



- **Datenfluss**
  - µC sendet empfängt Daten über serielle Schnittstelle (UART)
  - Serial-to-USB Adapter (ebenfalls µC) auf dem Arduino Board
  - PC empfängt sendet Daten über USB Schnittstelle
  - PC emuliert über USB Schnittstelle einen COM Port

- **Stromlaufplan/Schaltplan** des Arduino Boards
  - https://www.arduino.cc/en/uploads/Main/arduino-mega2560_R3-schematic.pdf

- Arduino-Bibliothek **`Serial`** für Zugriff auf serielle Schnittstelle.
  - https://www.arduino.cc/en/Reference/Serial

Quellenverzeichnis

[1] G. Gridling und B. Weiss. *Introduction to Microcontrollers*, Version 1.4, 26. Februar 2007, verfügbar online: https://ti.tuwien.ac.at/ecs/teaching/courses/mclu/theory-material/Microcontroller.pdf (abgerufen am 08.03.2017)

[2] M. Jiménez, R. Palomero und I. Couvertier. *Introduction to Embedded Systems – Using Microcontrollers and the MSP430*, Springer, 2014
(eBook in Bibliothek)

[3] U. Brinkschulte und T. Ungerer. *Mikrocontroller und Mikroprozessoren*, 3. Auflage, Springer, 2010 (eBook in Bibliothek).

[4] http://webuser.hs-furtwangen.de/~spale/forall/PES/Vorlesung/ppt/embedded-einf.pdf
(abgerufen am 10.03.2020)

[5] https://de.wikipedia.org/wiki/Embedded_Software_Engineering#/media/File:ESE_Referenzarchitektur.png
(abgerufen am 08.03.2017)

[6] L. Thiele und J. Teich. Eingebettete Systeme – Materialien zum Kapitel "Architekturentwurf".

[7] Datenblatt ATmega2560, https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

[8] https://www.chipsetc.com/computer-chips-inside-the-car.html (abgerufen am 10.03.2020)

[9] https://www.elektronik-kompendium.de/sites/bau/0201111.htm (abgerufen am 10.03.2020)

[10] https://www.kollino.de/elektronik/breadboard-steckplatine/ (abgerufen am 10.03.2020)