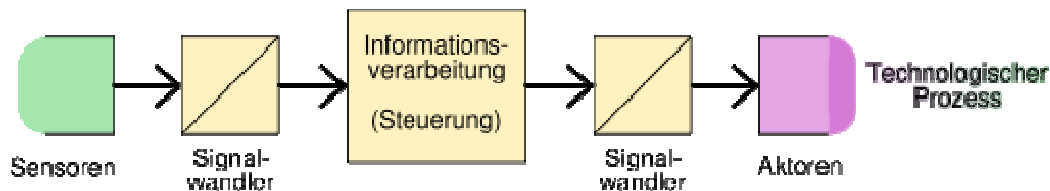
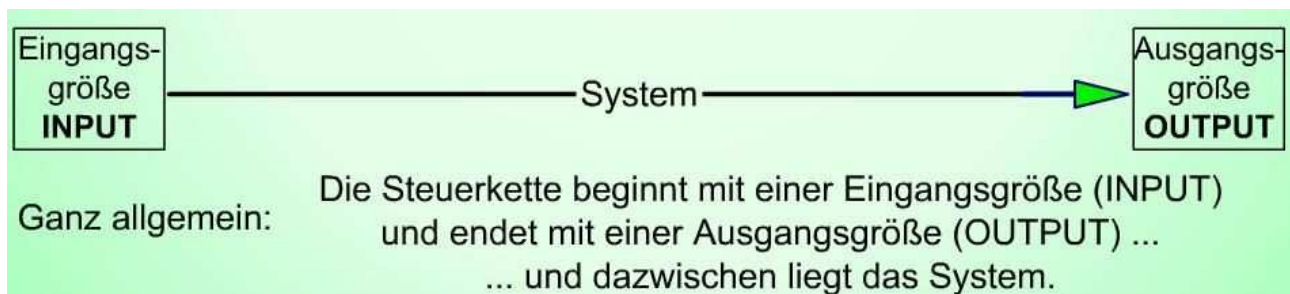


## Steuern von technischen Prozessen

### Grundlagen und Begriffe

**Steuerung:** Ziel einer Steuerung ist es, den Zustand eines Systems oder Objektes zu verändern. Dabei soll das System oder Objekt von einem Ist-Zustand in einen Soll-Zustand gebracht werden.

### Steuerkette:



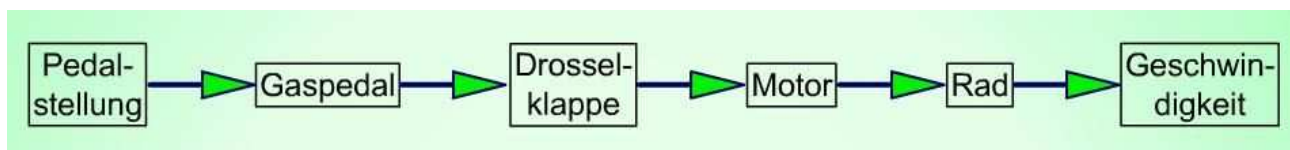
### Störgröße:

Ist eine Größe, die ungewollt von außen in Steuerabläufe eingreift. (Bsp: Rückenwind, Gegenwind; Neigungswinkel der Fahrbahn, ...)

**Aktor:** Gegenstück zu Sensor (Messfühler, Messaufnehmer); Ein Aktor setzt die Signale einer Steuerung in (meist) mechanische Arbeit um

### Blockschaltbild einer Steuerungskette:

Beispiel: Fortbewegung eines Autos mit unterschiedlichen Geschwindigkeiten



### Kernsätze:

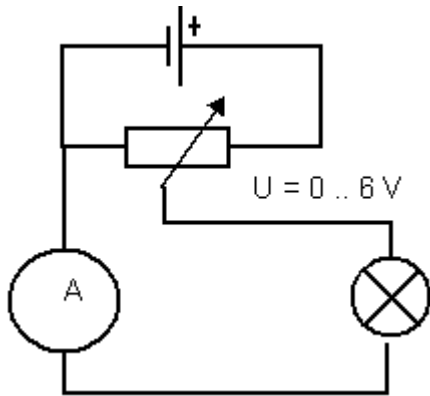
- Störgrößen auf das System verändern die Ausgabe
- Abweichungen in der Ausgabe bleiben bestehen. Das System korrigiert nichts. Der Mensch muss eingreifen.
- Abweichungen in der Ausgabe beeinflussen die Eingabe nicht.
- Von der Ausgabe erfolgt keine Rückmeldung.
- Die Steuerkette ist offen.

**Beispiele für Steuerketten**



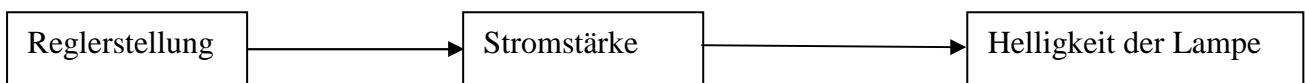
Beispiel	Steuerungsvorgang	auslösendes Signal	Sensor
Zeitschaltuhr	Strom ein-, ausschalten	Uhr abgelaufen	Schaltkontakt
Pflanzenwachstum	Wachstum steuern	Nährstoffmangel	Wurzel
Rauchmelder	Alarm auslösen	Temperatur steigt an Brandgas wird registriert keine klare Sicht	Temperatursensor Gassensor optischer Sensor
Patellarsehnenreflex	Kniegelenk strecken	Kurzer Schlag auf die Patellarsehne bei gebeugtem Kniegelenk	Nervenzelle (nervus femoralis)
Automatische Tür	Tür öffne / schließen	Lichtschranke unterbrechen	optischer Sensor

**SE: einfache Steuerung der Helligkeit einer Lampe**

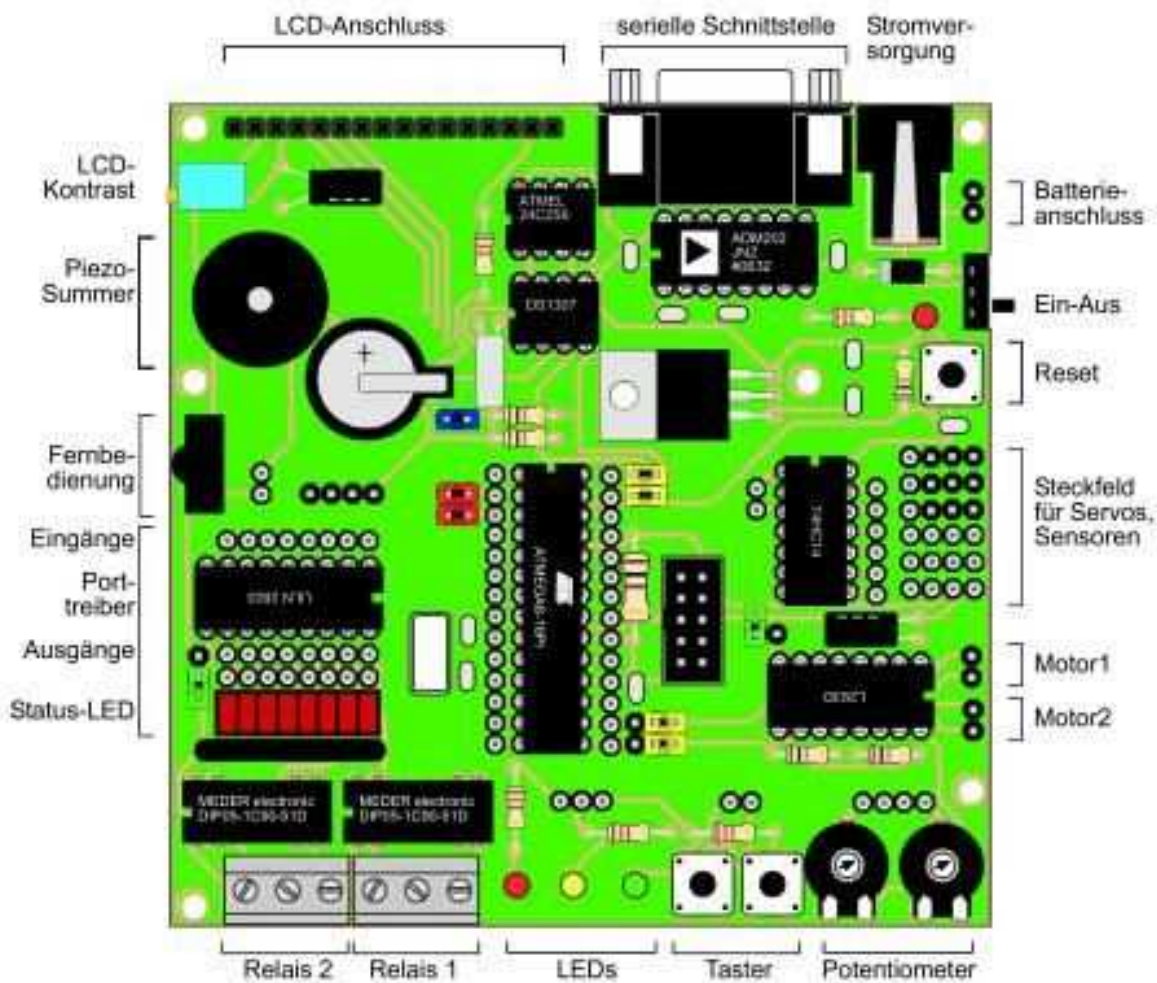


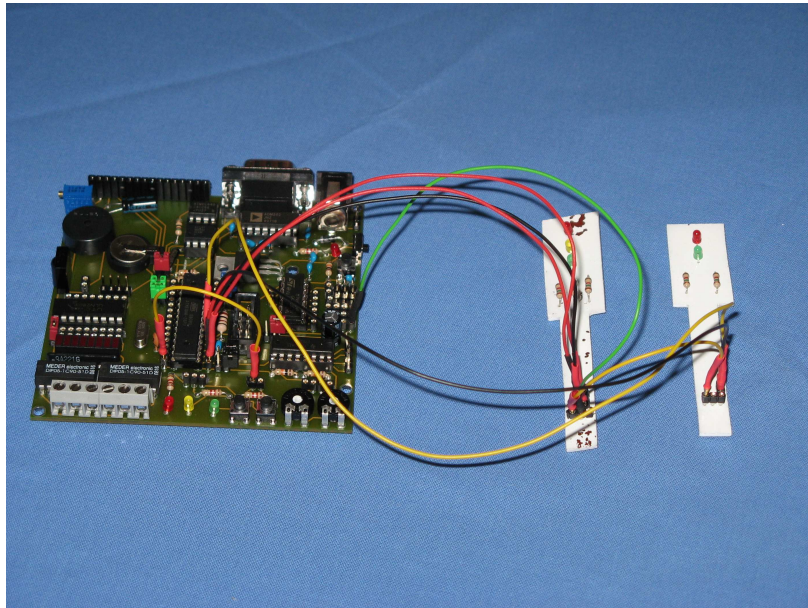
I in mA	Helligkeit (verbal)
0	dunkel
	glimmt
	hell
	sehr hell

Blockschaltbild der Steuerkette:



**Ampelsteuerung mit einem Mikrocontroller**





```

program Bedarfsampel;

{ $BOOTRST $00C00}          {Reset Jump to $00C00}
{$NOSHADOW}
{ $WG}                      {global Warnings off}

Device = mega8, VCC=5;

Import SysTick, SerPort, LCDmultiPort, TWImaster, BEEPport;

From System Import ;

Define
  ProcClock      = 16000000;      {Hertz}
  SysTick        = 10;           {msec}
  StackSize      = $0032, iData;
  FrameSize      = $0032, iData;
  SerPort        = 9600, Stop2;   {Baud, StopBits|Parity}
  RxBuffer       = 8, iData;
  TxBuffer       = 8, iData;
  TWIpresc       = TWI_BR400;
  LCDmultiPort   = I2C_TWI;
  LCDrows_M      = 2;            {rows}
  LCDcolumns_M   = 16;           {columns per line}
  LCDtype_M      = 44780;
  BeepPort       = PortD,7;

Implementation

{$IDATA}

{-----}
{ Type Declarations }

```

type

```
{-----}
{ Const Declarations }

{-----}
{ Var Declarations }
{$IDATA}
var
  taster[@PinB,0]:bit;
  led_rt_A[@PortB,1]:bit; // rote LED fuer Auto
  led_ge_A[@PortB,2]:bit;
  led_gr_A[@PortB,3]:bit;
  led_rt_F[@PortB,4]:bit; // rote LED fuer Fussgaenger
  led_gr_F[@PortB,5]:bit;

{-----}
{ functions }

procedure InitPorts;
begin
  PortB:= %00000001;
  DDRB:= %00111110;
  DDRD:= %10000000;
end InitPorts;

{-----}
{ Main Program }
{$IDATA}

begin
  InitPorts;
  EnableInts;
  loop
    // Anfangszustand setzen
    led_gr_A := 1;
    led_rt_A := 0;
    led_ge_A := 0;
    led_rt_F := 1;
    led_gr_F := 0;
    BeepClick;
    mdelay(500);
    if not taster then // Fussgaenger drueckt Taster
      led_gr_A := 1;
      led_rt_F := 1;
      mdelay(5000);
      led_gr_A := 0;
      led_ge_A := 1;
      mdelay(2000);
```

```
    led_ge_A := 0;
    led_rt_A := 1;
    mdelay(3000);
    led_rt_F := 0;
    led_gr_F := 1;
    BeepStepLH;
    mdelay(15000);
    led_gr_F := 0;
    led_rt_F := 1;
    mdelay(3000);
    led_ge_A := 1;
    mdelay(1000);
    led_rt_A := 0;
    led_ge_A := 0;
    led_gr_A := 1;
    mdelay(1000);
endif;
endloop;
end Bedarfssampel.
```