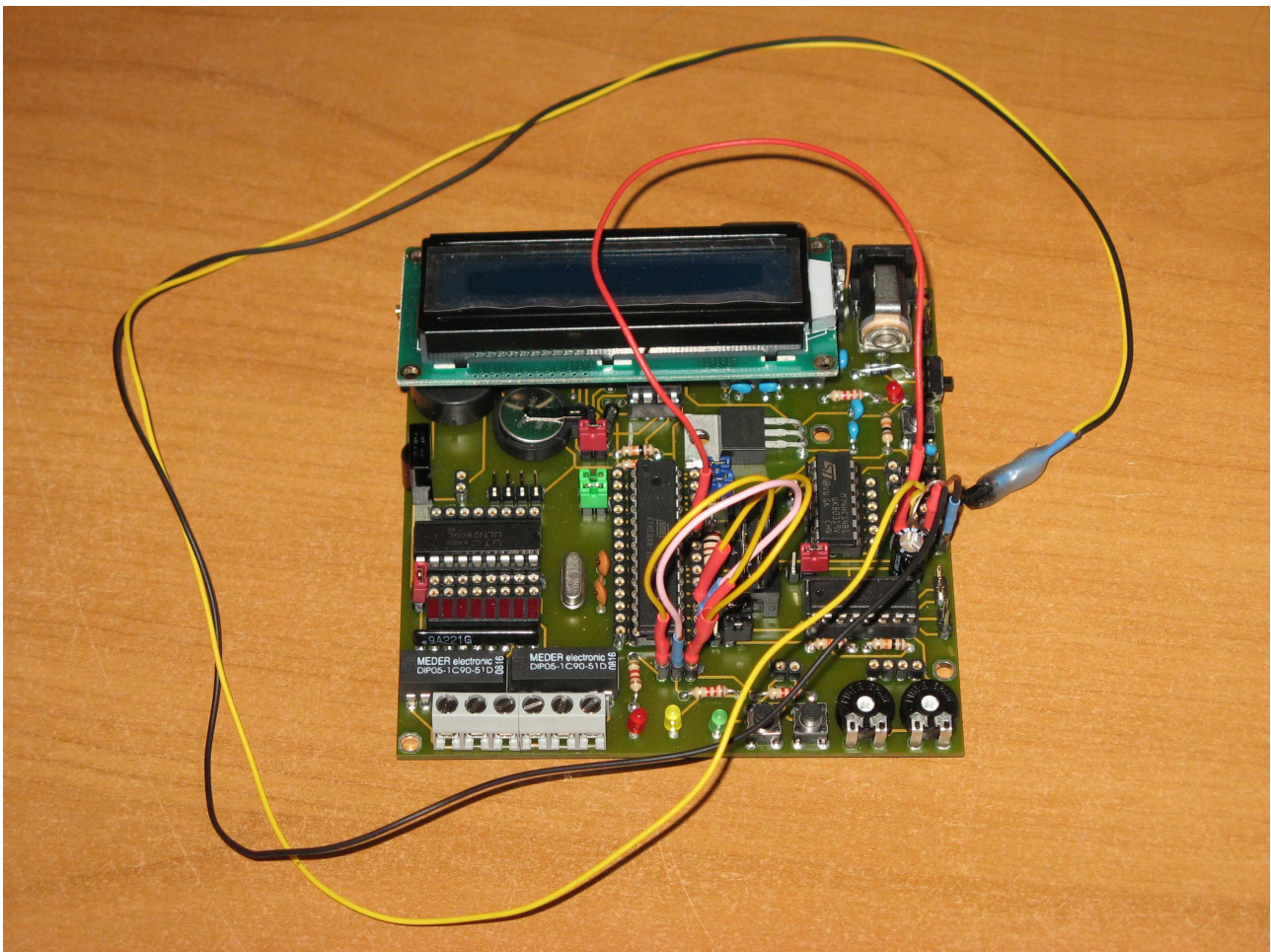


## Mikrocontroller

Arbeiten Sie die Folien 1 bis 13 des Vortrages „*Mikrocontroller -Einsatzmöglichkeiten und Entwicklungsrichtungen*“ von Prof. Dr. Ing. Peter Protzel durch und notieren Sie:

- Was ist ein Mikrocontroller?
- Vergleichen Sie einen herkömmlichen Mikroprozessor wie er in einem normalen PC-System zu finden ist mit einem Mikrocontroller!
- Notieren Sie Einsatzgebiete von Mikrocontrollern! Siehe hierzu auch [http://de.wikipedia.org/wiki/Reduced\\_Instruction\\_Set\\_Computing](http://de.wikipedia.org/wiki/Reduced_Instruction_Set_Computing) und [http://de.wikipedia.org/wiki/Eingebettete\\_Systeme](http://de.wikipedia.org/wiki/Eingebettete_Systeme).

### Simulation einer Heizungsanlage mit einem Mikrocontroller



```
program tempmessung;
```

```
Device = mega8, VCC=5;
```

```
Import SysTick, SerPort, ADCPort, LCDmultiPort, TWImaster;
```

```
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;
ADCchans       = 1, iData;
ADCpresc       = 128;
TWIpresc       = TWI_BR400;
LCDmultiPort   = I2C_TWI;
LCDrows_M      = 2;            {rows}
LCDcolumns_M   = 16;          {columns per line}
LCDtype_M      = 44780;

```

## Implementation

```

{$IDATA}

{-----}
{ Const Declarations }
const
  deg: char= #SDF; // Gradzeichen für Ausgabe
{-----}
{ Var Declarations }
{$IDATA}
var
  led_rt[@PortB,1] : bit;
  led_ge[@PortB,2] : bit;
  led_gr[@PortB,3] : bit;
  Temperatur       : integer;

{-----}
{ functions }
procedure LCDInit; // LCD wird initialisiert
begin
  LCDsetup_m(lcd_m1);
  LCDon_m(lcd_m1); // LCD anschalten
  LCDclr_m(lcd_m1); // LCD loeschen
  LCDcursor_m(lcd_m1,false,false); // Cursor ausschalten
end;

procedure LCDClear; // LCD loeschen
begin
  LCDclr_m(lcd_m1); // LCD loeschen
  LCDhome_m(lcd_m1); // Schreibmarke auf Anfang setzen
end;

procedure InitPorts; // wichtig! Konfiguration des ADC auf externe
Referenz 5V
begin
  ADMUX := ADMUX and %01111111; // Referenz auf AVCC setzen

```

```

ADMUX := ADMUX or %01000000; // Bit 7 loeschen, Bit 6 setzen
DDRB:=  %00001110; // Ausgaenge fuer LED

end;

function Messung: Integer; // Temperaturmessung in °C
var
  wert: Word;
  temp: Integer;
begin
  wert := GetADC; // Abruf des Wandlungsergebnisses als
  Spannungswert
  temp := ((Integer(wert)*63) div 125)-194;
  // sensorspezifische Gleichung
  Return(temp);
end;
{-----}
{ Main Program }
{$IDATA}

begin

  InitPorts;
  EnableInts;
  LCDInit; // Display initialisieren
  write(LCDOut_m, 'Temp: '); // Ausgabe schreiben
  loop
    LCDxy_m(lcd_m1,6,0); // Cursor positionieren Spalte 7,
  Zeile1
    Temperatur := Messung; // Sensor abfragen
    write(LCDOut_m, IntToStr(Temperatur)+deg+'C ');
  // Messung ausfuehren + Ausgabe
  case Temperatur of 0..28: // Heizen
    led_rt:=0;
    led_ge:=0;
    led_gr:=1;
    |
  28..30: // Heizen, aber Achtung!
    led_rt:=0;
    led_ge:=1;
    led_gr:=1;
    |
  31..40: // Heizung abschalten
    led_rt:=1;
    led_ge:=0;
    led_gr:=0;
    |
  endcase;
  mdelay(1000); // kurze Pause nach jeder Messung

  endloop;
end tempmessung.

```

Vergleich  
Soll- mit  
Ist-Wert

Regelalgorithmus  
Anweisungen an Stellglied

Sensor  
abfragen