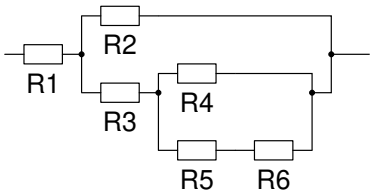
2.6 Funktionen/Rückgabewert

2.6.1 Situation

Es soll der Ersatzwiderstand einer komplizierten gemischten Schaltung berechnet werden: $R_{ers} = R_1 + (R_2||(R_3 + (R_4||(R_5 + R_6))))$ mit $R_1 = 1\,\Omega$, $R_2 = 2\,\Omega$, $R_3 = 3\,\Omega$, $R_4 = 4\,\Omega$, $R_5 = 5\,\Omega$ und $R_6 = 6\,\Omega$.



In der Formel wird zweimal die Berechnung einer Parallelschaltung gefordert. Da bietet es sich an, dies Berechung in eine Funktion zu verlagern. Schön wäre es, wenn eine Funktion — ähnlich wie die Sinus-Funktion aus `math.h` — die Eingabewerte annimmt, das Ergebnis berechnet und an das Hauptprogramm zurückgibt. Wie macht man das in C?

2.6.2 Beispiel

In einem Forum findet man folgendes Beispiel:

```
1  #include <stdio.h>
2  int xquadrat(int x)
3  {
4      int erg;
5      erg = x*x;
6      return erg;
7  }
8  int main(void)
9  {
10     int a, b;
11     printf("Eingabe einer Zahl: ");
12     scanf("%i", &a);
13     b = xquadrat(a);
14     printf("Ergebnis: %i\n", b);
```

Zunächst wird die Funktion `xquadrat()` betrachtet:

Zeile 2: Im Funktionskopf noch vor dem Funktionsnamen findet sich der so genannte Rückgabetyp der Funktion, nämlich `int`. Die Funktion gibt einen Wert mit dem Datentyp `int` zurück.

Zeile 4: Dieser Rückgabetyp hat aber – im Gegensatz zum Parameter `x` – in der Funktion noch keine zugehörige Variable. Man muss sich also eine Variable diesen Typs in der Funktion anlegen. Darum wird hier die `int`-Variable `erg` angelegt.

Zeile 5: Das Ergebnis der Funktion wird berechnet und in `erg` abgelegt.

Zeile 6: Die `return`-Anweisung hat hier zwei Auswirkungen:

a) Der Inhalt von `erg` wird an eine besondere Stelle kopiert, und zwar an die Stelle, die (im Hauptprogramm) für das Ergebnis der Funktion reserviert wurde.[1]

b) Rücksprung ins Hauptprogramm

Nun die Besonderheiten in `main()`:

[1] Man kann hier von einer namenlosen Variable sprechen, in die hineinkopiert wird.

Zeile 13: Hier steht eine Zuweisung (Gleichheitszeichen). Zuerst wird die rechte Seite berechnet (das Ergebnis dieser Berechnung wird danach an die linke Seite zugewiesen).

- Die Funktion `xquadrat()` wird aufgerufen, d.h. der aktuelle Parameter `a` wird in den formalen Parameter `x` kopiert und es wird an den Anfang der Funktion gesprungen.
- Die Funktion wird abgearbeitet (s.o.).
- Nach dem Rücksprung aus der Funktion wird das Ergebnis dieser Funktion (die Kopie von `erg` an der reservierten Stelle, s.o.) an die linke Seite des Zuweisungsausdrucks, also an `b`, zugewiesen.

2.6.3 Lösung

Jede Funktion kann also, wenn man will, nicht nur beliebig viele Parameter aufnehmen, sondern auch, wenn man will, *genau einen* Wert an die aufrufende Funktion zurückgeben.

In der Funktion braucht man dazu einen Datentyp für den Rückgabewert im Funktionskopf sowie mindestens eine `return`-Anweisung im Funktionsrumpf.

Wenn man die Funktion benutzt, kann man sie wie einen mathematischen Ausdruck (des richtigen Datentyps) verwenden, man kann also z.B. `xquadrat(a)` überall dort verwenden, wo auch `a*a` stehen könnte.

```
1  #include <stdio.h>
2  double rparallel(double ra, double rb)
3  {
4      double erg;
5      erg = (ra*rb)/(ra+rb);
6      return erg;
7  }
8  int main(void)
9  {
10     double r1=1.0, r2=2.0, r3=3.0, r4=4.0, r5=5.0, r6=6.0, rers;
11     rers=r1+rparallel(r2, r3+rparallel(r4, r5+r6));
12     printf("Rers = %lf Ohm\n", rers);
13     return 0;
14 }
```

2.6.4 Ergänzung

Auch die `main()`-Funktion gibt einen `int`-Wert an die aufrufende Umgebung zurück. Auf der Kommandozeile (Konsole) kann man diesen Wert abfragen.

- in Linux: `echo $?`
- in Wind.: `echo %errorlevel%`

Weiter kann man ihn auch in eigenen Shell-Skripten oder Batch-Dateien verwenden, um z.B. nur dann eine Aktion auszuführen, wenn eine andere zuvor geklappt hat.