

Pointer in C

Ein Pointer ist eine Variable. Der Inhalt eines Pointers ist eine Adresse.

man sagt: der Pointer "zeigt" auf einen Speicherplatz

```
/*AVR und Pointer(c) Kner 2007 */

char global1;          // Speichersegment .bss wird am Programmanfang
                       // gelöscht;
                       // wird verwendet für nicht initialisierte
globale und static Variable
char global2 = 3;     // .data wird am Programmanfang ins RAM kopiert

char err_str[] = "Your program has died a horrible death!"; //
im .data und im Flash

char uebergabe_Wert(char x); //Prototypen
void uebergabe_Adresse(char *x);

int main()
{
    char i;
    char local1 = 'A'; //am Stack
    char local2;      //am Stack

    char *pointer1 = &local1; //auf welche Adressen zeigen die
Pointer?
    char *pointer2 = &global1;
    char *pointer3 = &global2;

    pointer1 = &global2; // "Pointer umbiegen"

    *pointer1 = 0x11; // dem Pointer nachgehen

    *pointer2 = 0x22;
    *pointer3 = 0x33;
    pointer3++;
    *pointer3 = 0x34; // schwerer Programmierfehler:
                     // der Speicherbereich, auf den Pointer3
                     //jetzt zeigt gehört eigentlich err_str

    pointer3 = err_str; // err_str ist ein konstanter Pointer
                       // (kann man nicht verändern

    //err_str++;      // funktioniert nicht
    for (i = 0; i < 10; i++)
        *pointer3++ = 0;
        // besser lesbar: { *pointer3 = 0; pointer3++;}

    //ist das Gleiche wie
    for (i = 0; i < 10; i++)
        err_str[i] = 0; //besser lesbar

    err_str[3] = 'B';
    *(err_str + 3) = 'C'; //komplizierter
```

```

//praktisches Beispiel:
//in einem Integer-Array sollen alle MSB-Bytes gelöscht werden

int ary1[] = {0xaaaa, 0xbbbb, 0xcccc};
char* pointer4 = (char*)ary1; //Pointer auf den Anfang des
Arrays
for (i = 0; i < sizeof(ary1) / sizeof(int); i++)
{
    *pointer4 = 0;
    pointer4 += 2; //auf die nächste Zahl stellen
}

//eigentlicher Hauptzweck von Pointern: Austausch von Daten

global1 = 3;
pointer1 = &global1;
*pointer1 = 2; //verändert global1

// Austausch von Daten mit einem Unterprogramm
char local3 = 7;
local3 = uebergabe_Wert(local3);
uebergabe_Adresse(&local3); //jetzt weiß das Unterprg.,
//wo local3 "wohnt" und kann
// dort ein Paket ablegen

while (1)
;
}

char uebergabe_Wert(char x)
{
    return x * 2;
}
void uebergabe_Adresse(char *x)
{
    char h = *x;
    *x = h + 13;
}

```