



Verfasser:
D. Gachet / HTA-FR - Telekommunikation

HTA-FR – Kurs Telekommunikation

Embedded systems 1 und 2 Programmiersprache C – Funktionszeiger

Klasse T-2 // 2018-2019



- ▶ **Deklaration**
- ▶ **Zuweisung und Aufruf**
- ▶ **Tabellen**



Funktionszeiger sind sehr mächtige Elemente der Programmiersprache C. Der Funktionszeiger ist eine Variable, die die Adresse einer Funktion (Eintrittspunkt der Funktion) enthält.

Funktionszeiger werden in C sehr oft verwendet, weil sie Folgendes ermöglichen:

- ▶ Schnittstellen mit Rückruffunktionen (callback) zu konzipieren.
- ▶ Generische Algorithmen für die Verarbeitung von Daten (OO) zu konzipieren.

Die Deklaration eines Funktionszeigers hat die folgende Form:

```
int(*function) (int, int);
```

Es ist oft praktisch, für die Deklaration eines Funktionszeigers eine `typedef` zu verwenden, z. B.:

```
typedef int(*function_t)(int, int);  
function_t function;
```



Die Zuweisung eines Funktionszeigers kann die folgenden Formen annehmen:

```
function = my_function
```

oder

```
function = &my_function
```

Für den Aufruf eines Funktionszeigers genügt es, einen normalen Funktionsaufruf zu verwenden, z. B.:

```
int value = function (10, 20);
```

es ist auch möglich, die für die Zeiger verwendete Syntax zu benutzen:

```
int value = (*function) (10, 20);
```



Natürlich ist es möglich, dass der Funktionsaufruf einen Funktionszeiger zurück gibt.

Es existieren zwei Arten für die Deklaration solcher Funktionen:

```
int (*get_function(int which_one)) (int, int);
```

oder

```
typedef int (*function_t) (int, int);  
function_t get_function (int which_one);
```



Die Deklaration einer Funktionszeiger-Tabelle kann die folgenden Formen annehmen:

```
int(*functions[20]) (int, int);
```

oder

```
typedef int(*function_t) (int, int);  
function_t functions[20];
```

den Funktionsaufruf:

```
int value = functions[1] (10, 20);
```