

Funktionen („Methoden“) mit/ohne Übergabe- und Rückgabewert

Mit dem Begriff Funktion ist ein Programmteil gemeint, der zum Zwecke der besseren Übersichtlichkeit (besseren Programm-Struktur) ausgelagert wurde. Das Schreiben von Funktionen hat darüber hinaus den großen Vorteil, dass eine solche Funktion von verschiedenen Programmstellen aus aufgerufen werden kann. Sich wiederholende Programmanteile werden folglich in eine Funktion ausgelagert.

Es sind grundsätzlich vier verschiedene Funktionstypen zu unterscheiden:

- Funktionen ohne Übergabe- und Rückgabewert
- Funktionen mit Übergabewert
- Funktionen mit Rückgabewert
- Funktionen mit Übergabe- und Rückgabewert

Beispiel: Funktion ohne Übergabe- und Rückgabewert

```
void blinken(void)    // void bedeutet „nichtig, ungültig, leer“
{
    // Funktion zum „Blinken“ lassen von zwei LEDs an Port D
    PORTD = 16; delay(500); PORTD = 32; delay(500);
}

void loop (void)
{
    ...
    blinken(); // Funktionsaufruf der Funktion „blinken“
    ...
}
```

Der Befehl „blinken()“ ruft die Funktion auf. Nach Abarbeitung der Funktion kehrt der Programmablauf an die Stelle des Funktionsaufrufs zurück und setzt das Programm hinter dem Funktionsaufruf fort.

Beispiel: Funktion mit Übergabewert

```
void leuchten(int zahl)    // `15` tritt an die Stelle der
{                          // Variable „zahl“
    for (int t = zahl; t > 0; t--) {
        PORTD = 64; delay(100); PORTD = 128; delay(100);
    }
}

void loop (void)
{
    ...                      // Übergabe des Werts `15`
    leuchten(15);           // an die Funktion „leuchten“
    ...
}
```

Der Befehl „leuchten(15)“ ruft die Funktion auf und übergibt in den Klammern die ganze Zahl 15. Die aufgerufene Funktion übernimmt den übergebenen Wert in der „Schublade“ mit dem angegebenen Namen „zahl“.

Beispiel: Funktion mit Rückgabewert

```
// int bedeutet: „diese Funktion ist vom Typ
// int und gibt eine ganze Zahl zurück !!!
int tasterabfrage(void) {
    int taste = PINC & 0b00001111;
    return taste;
}
```

Der Funktionsaufruf in der loop() kann dann so aussehen:

```
taster = tasterabfrage();
if (taster == 1) {
    PORTD = 1; delay(2000); PORTD = 2; delay(2000);
}
```

Der Befehl „tasterabfrage()“ ruft die Funktion auf. Da die Funktion einen Wert vom Typ integer zurückgibt, muss vor dem Funktionsaufruf eine Variable (hier: taster) stehen, die den Rückgabewert aufnehmen kann.

Beispiel: Funktion mit Übergabe- und Rückgabewert

```
int allesdrin(int wert)
{
    int rueck = wert * (PINC & 0b00001111);
    PORTD = 255; delay(250); return rueck;
}

void loop (void)
{
    ... // Übergabe des Werts '1000' an allesdrin
    int echtjetzt = allesdrin(1000);
    delay(echtjetzt);... // Rückgabewert in echtjetzt speichern
}
```

Der Befehl „allesdrin(1000)“ ruft die Funktion auf und übergibt in den Klammern die ganze Zahl 1000. Die aufgerufene Funktion übernimmt den übergebenen Wert in der „Schublade“ mit dem Namen wert. Diese Variable ist somit innerhalb der Funktion verfügbar. Durch „return rueck“ wird der Wert des Integers rueck an die Aufrufstelle zurückgegeben und in der Variablen echtjetzt gespeichert. In Abhängigkeit von der gedrückten Taste wird anschließend eine entsprechend lange bzw. kurze Pause gemacht.