

## Übungsblatt 2    Formatierte Ausgabe


### Lösung

#### Lernziele:

- Ausgabe von Text auf dem EV3-Display erlernen
- das Formatieren von Text erlernen
- den Umgang mit `TermPrintf()` üben
- den Umgang mit `Ev3Clear()` üben

#### Legende:

 Lektüre, die vor der Übung gelesen werden muss

 Fragen/Aufgaben, die vor der Übung zu bearbeiten sind

 Aufgaben, die in der Übung zu bearbeiten sind

In der vergangenen Übung wurde bereits die Funktion `printf` vorgestellt, welche die formatierte Ausgabe von Text ermöglicht.

Für die Ausgabe von Text auf dem LEGO MINDSTORM EV3 gibt es eine sehr ähnliche Funktion.

```
TermPrintf(const char * format,...);
```

Die auszugebende Zeichenkette wird mit `format` übergeben. Der Text muss dabei mit `"` beginnen und mit `"` enden.

So könnte eine Anweisung zur Ausgabe auf dem Display des EV3 aussehen:

```
#include <ev3.h>
int main(void)
{
    TermPrintf("Ich studiere in Aschaffenburg.");
    Wait(SEC_5);
    Ev3Clear();
    return 0;
}
```

Durch das Einbinden der Datei `ev3.h` werden dem Compiler die EV3-spezifischen Funktionen bekannt gemacht. Die `TermPrintf`-Funktion sorgt dafür, dass der Satz

Ich studiere in Aschaffenburg.

ausgegeben wird. Die `wait`-Funktion lässt den EV3 fünf Sekunden warten, bevor die Ausgabe gelöscht wird.

**Tabelle 1** im Anhang zeigt eine Liste der Formatbuchstaben, mit der in der `TermPrintf`-Funktion verschiedene Zahlenformate ausgegeben werden können. In **Tabelle 2** sind die Escape-Sequenzen dargestellt. Mit Hilfe dieser Steuerzeichen lassen sich Texte schnell und einfach formatieren.

### Hinweis

Es sollte beachtet werden, dass das Display des Roboters nur eine bestimmte Größe hat. Am rechten Rand kommt es zu einem automatischen Zeilenumbruch und der Text wird in der nächsten Zeile fortgeführt. Ist das untere Ende erreicht, wird der Text um eine Zeile nach oben geschoben und ist somit nicht mehr sichtbar.

Zum Löschen des Inhaltes wird

```
Ev3Clear();
```

verwendet. Der Befehl setzt auch gleichzeitig den Cursor zurück.

## 2.1 Vorbereitung zur Übung

Die Vorbereitungen und Fragen sind vor dem Übungstermin zu bearbeiten. Sie sind als Hilfen gedacht, um Ihnen die Aufgaben, die an den Übungsterminen selbst zu bearbeiten sind, zu erleichtern.

📖 Lesen Sie zur Vorbereitung der Übung die Dokumente „Erstellung eines EV3-Projektes“, „Einführung Display“ und „Befehlsübersicht EV3“! Sehen Sie sich zusätzlich das Video „Erstellen und Übertragen eines EV3 Projektes“ an!

 *Einleitende Fragen:*

Welchen Project type und welche Toolchain benötigen Sie, um ein neues Projekt zu erstellen?

<i>ProjectType : Executable ⇒ HelloWorldEV3Project</i> <i>Toolchain : CrossARM/EV3GCC</i>
--

Vervollständigen Sie folgende Tabelle:

<b>Cross GNU EV3 Toolchain</b>	
Toolchain name	<i>arm – none – linux – gnueabi – gcc</i>
Toolchain path	<i>C : \csite\bin</i>
EV3 Uploader	<i>C : \ev3\uploader</i>

 **Eine kleine Übung zum Aufwärmen**


Schreiben Sie ein kleines Programm, welches Ihnen, wie in Übung 1 schon behandelt, Ihre Daten (Adresse, Telefonnummer, Matrikelnummer, Studiengang) auf dem EV3 ausgibt. Ergänzen Sie (analog zum Beispiel auf Seite 1) dazu folgenden Funktionsrumpf:

```
#include <ev3.h>
int main(void)
{
    TermPrintf( „Eclipsestraße 1, \n“);
    TermPrintf( „12345 Marsstadt, \n“);
    TermPrintf( „Tel.: 0123456789, \n“);
    TermPrintf( „Matrikelnr.: 99887766, \n“);
    TermPrintf( „Studiengang: Herr der Ringe“);
    Wait(SEC_ 5);
    return 0;
}
```

## 2.2 Ausgabe der Zahlen 8 bis 16 in verschiedenen Formaten

*Aufgabe:*

Lassen Sie die Zahlen 8 bis 16 auf dem Display des EV3 ausgeben. Die Zahlen sollen dabei als Dezimal-, Oktal- und Hexadezimalzahlen ausgegeben werden. Für die einzelnen Zahlenformate soll dabei eine neue Zeile begonnen werden.

 Wählen Sie aus der Tabelle 1 (siehe Anhang) die für die Aufgabenstellung passenden Formatbuchstaben aus!

Dezimal: %d ; Dezimalzahl mit Vorzeichen  
Oktal: %o ; Oktalzahl ohne Vorzeichen  
Hexadezimal: %x ; Hexadezimalzahl ohne Vorzeichen, klein geschrieben

❖ Schreiben Sie nun das Programm, in dem Sie die gewonnenen Erkenntnisse umsetzen!

Lösung:

```
#include <ev3.h>

int main(void)
{

    TermPrintf( "Dezimal:\n");
    TermPrintf( "%d, ", 8);
    TermPrintf( "%d, ", 9);
    TermPrintf( "%d, ", 10);
    TermPrintf( "%d\n", 11);
    TermPrintf( "%d, ", 12);
    TermPrintf( "%d, ", 13);
    TermPrintf( "%d, ", 14);
    TermPrintf( "%d, ", 15);
    TermPrintf( "%d\n", 16);

    TermPrintf( "Oktal:\n");
    TermPrintf( "%o, ", 8);
    TermPrintf( "%o, ", 9);
    TermPrintf( "%o, ", 10);
    TermPrintf( "%o\n", 11);
    TermPrintf( "%o, ", 12);
```

```
TermPrintf( "%o, ", 13);  
TermPrintf( "%o, ", 14);  
TermPrintf( "%o, ", 15);  
TermPrintf( "%o\n", 16);  
  
TermPrintf( "Hexadezimal:\n");  
TermPrintf( "%x, ", 8);  
TermPrintf( "%x, ", 9);  
TermPrintf( "%x, ", 10);  
TermPrintf( "%x\n", 11);  
TermPrintf( "%x, ", 12);  
TermPrintf( "%x, ", 13);  
TermPrintf( "%x, ", 14);  
TermPrintf( "%x, ", 15);  
TermPrintf( "%x\n", 16);  
Wait(SEC_10);  
  
return 0;  
}
```

HIER BENÖTIGTE FUNKTIONEN:

- TermPrintf("Textbeispiel");
- Wait(SEC\_30);

## 2.3 Formatierte Textausgabe

*Aufgabe:*

Geben Sie die beiden nachfolgenden Texte auf dem Display aus.

**Text1:**


```
Hochschule  
Aschaffenburg  
Wuerzburger  
Strasse 45  
63743 Aschaffenburg
```

```
Gebaeude: 40  
Raum: 130
```

**Text2:**

Seite 2 der formatierten Textausgabe:

Am Ende des Displays gibt es einen automatischen Zeilenumbruch.

 Wählen Sie aus der Tabelle 2 (siehe Anhang) die für die Aufgabenstellung passenden Steuerzeichen aus! Begründen Sie Ihre Auswahl!

<pre>\n: Neue Zeile \f: Seitenvorschub, neue Seite benötigt</pre>
---

❖ Schreiben Sie nun das Programm, in dem Sie die gewonnenen Erkenntnisse umsetzen!

Lösung:

```
#include <ev3.h>
```

```
int main(void)
{
```

```
TermPrintf("Adresse:\n");
TermPrintf(" Hochschule\n Aschaffenburg\n");
TermPrintf(" Wuerzburger\n Strasse 45\n");
TermPrintf(" 63743 Aschaffenburg\n\n");
TermPrintf(" Gebaeude: 40\n");
TermPrintf(" Raum: 130\n");
```

```
Wait(SEC_5);
```

```
Ev3Clear();
```

```
TermPrintf("Seite 2 der formatierten Textausgabe:\n\n");
TermPrintf("Am Ende des Displays gibt es einen automatischen
Zeilenumbruch.\n");
```

```
Wait(SEC_5);
```


```
return 0;
}
```

HIER BENÖTIGTE FUNKTION:

- `TermPrintf("Textbeispiel");`
- `Ev3Clear();`

## 2.4 Formatierung der Genauigkeit und Stellenangaben

Für die Ausgabe von Zahlen existieren eine Reihe von Zusatzoptionen, die zwischen % und Formatierungsbuchstaben geschrieben werden. Diese Zusatzoptionen wirken sich u.a. auf rechts-, linksbündige Ausgabe mit/ohne führenden Nullen, die Anzahl der Nachkommastellen (bei Fließkommazahlen) aus.

 Recherchieren Sie zunächst unter <http://www.tutorials.at/c/03-dateneingabe-ausgabe.html>, wie welche Zusatzoption in C realisiert wird.

 *Aufgabe:*

Überlegen Sie, welche Formatbuchstaben Sie für die Zahlen 4711 und 0.24124 benötigen.

Zahl	Formatbuchstabe
4711	%d
0.24124	%f

 Vervollständigen Sie nun folgende Tabelle:

Zusatzoption	4711	0.24124
rechtsbündig	%d	%f
linksbündig	%-d	%-f
mit Vorzeichen	%+d	%+f
mit Vorzeichen linksbündig	%+-d	%+-f
mit führender Null	%0d	%0f
mehrere Stellen einer Zahl (zB. 8) angeben	%8d	%8f
mit Vorzeichen, mehreren Stellen und führender Null	%+08d	%08f
mehrere Nachkommastellen angeben (zB. 8)	-	%.8f

❖ Lassen Sie sich nun die beiden Zahlen nacheinander mit den verschiedenen Zusatzoptionen auf dem Display des EV3 ausgeben. Beachten Sie dabei, dass die Ausgabe der ersten Zahl gelöscht werden muss, bevor die Ausgabe der zweiten Zahl erfolgen kann!

Lösung:

```
#include <ev3.h>

int main(void)
{

/* rechtsbündig Dezimal */
TermPrintf("%d\n",4711);

/* linksbündig Dezimal */
TermPrintf("%-d\n",4711);
```



Prof. Dr.-Ing. J. Abke  
Simón Rodríguez Pérez  
Patricia Stegmann

TH Aschaffenburg  
Informatik I

```
/* mit Vorzeichen Dezimal */
TermPrintf("%+d\n",4711);

/* mit Vorzeichen linksbündig Dezimal */
TermPrintf("%+-d\n",-4711);

/* mit führender Null Dezimal */
TermPrintf("%+d\n",4711);

/* mehrerer Stellen angeben Dezimal */
TermPrintf("%8d\n",4711);

/* Mit Vorzeichen, mehrere Stellen und führender Null Dezimal */
TermPrintf("%+08d\n",4711);

Wait(SEC_10);

/*Sauebern des Displays*/
Ev3Clear();

/* rechtsbündig Fließkomma */
TermPrintf("%f\n",0.24124);

/* linksbündig Fließkomma */
TermPrintf("%-f\n",0.24124);

/* Mit Vorzeichen Fließkomma */
TermPrintf("%+f\n",0.24124);

/* Mit Vorzeichen und linksbündig Fließkomma */
TermPrintf("%+-f\n",0.24124);

/* Mit führender Null Fließkomma */
TermPrintf("%0f\n",0.24124);

/* Mehrere Stellen angeben Fließkomma */
TermPrintf("%8f\n",0.24124);

/* Mit Vorzeichen, mehrere Stellen und führender Null Fließkomma */
TermPrintf("%+08f\n",0.24124);
```

Prof. Dr.-Ing. J. Abke  
Simón Rodríguez Pérez  
Patricia Stegmann

TH Aschaffenburg  
Informatik I

```
/* Mehrere Nachkommastellen Fließkomma */  
TermPrintf("%.8f\n",0.24124);  
Wait(SEC_10);  
  
return 0;  
}
```

## Anhang

**Tabelle 1: Liste der Formatbuchstaben**

Format	Argumenttyp	Erläuterung	Beispiel
c	char oder int	einzelnes Zeichen	<code>printf("%c", 'A');</code> druckt A
d	int	dezimal	<code>printf("%d", -45);</code> druckt -45
i	int	dezimal	<code>printf("%i", -45);</code> druckt -45
u	unsigned int	dezimal ohne Vorzeichen	<code>printf("%u", 3);</code> druckt 3
o	unsigned int	oktal ohne Vorzeichen	<code>printf("%o", 8);</code> druckt 10
x	unsigned int	hexadezimal 0-9abcdef	<code>printf("%x", 11);</code> druckt b
X	unsigned int	hexadezimal 0-9ABCDEF	<code>printf("%X", 11);</code> druckt B
f	float	Gleitpunktzahl	<code>printf("%f", 5.234);</code> druckt 5.234000
e,E	float	Exponentenschreibweise	<code>printf("%e", 0.000005);</code> druckt 5.000000e-6
g,G	float	das kürzere von f und e	
s	string	Ausgabe bis '\0'	<code>printf("%s", "string");</code> druckt string
ld	long int	dezimal	<code>printf("%ld", -45456256432);</code> druckt -45456256432
lf	double	Gleitpunktzahl doppelt genau	
le, lE	double	Exponentenschreibweise doppelt genau	
hd	short int	dezimal, Short-Int	
hx	short int	hexadezimal, Short-Int	

## Tabelle 2: Liste der Steuerzeichen

<code>\b</code>	Backspace (ein Zeichen zurück)
<code>\f</code>	Seitenvorschub
<code>\n</code>	Neue Zeile
<code>\r</code>	Wagenrücklauf (an den Anfang d. akt. Zeile)
<code>\v</code>	Vertikales Tabulatorzeichen
<code>\'</code>	Hochkomma
<code>\"</code>	Anführungsstriche
<code>\\</code>	Backslash