

# Hardwarenahe Programmierung

Prof. Dr. rer. nat. Peter Gerwinski

19. November 2018

# Hardwarenahe Programmierung

<https://gitlab.cvh-server.de/pgerwinski/hp.git>

## 1 Einführung

## 2 Einführung in C

...

### 2.15 String-Operationen

## 3 Bibliotheken

### 3.1 Der Präprozessor

### 3.2 Bibliotheken einbinden

### 3.3 Bibliotheken verwenden

### 3.4 Projekt organisieren: make

## 4 Hardwarenahe Programmierung

## 5 Algorithmen

### 5.1 Differentialgleichungen

...

...

## 3 Bibliotheken

### 3.1 Der Präprozessor

**#include**: Text einbinden

- **#include <stdio.h>**: Standard-Verzeichnisse – Standard-Header
- **#include "answer.h"**: auch aktuelles Verzeichnis – eigene Header

## 3 Bibliotheken

### 3.1 Der Präprozessor

**#include**: Text einbinden

- **#include <stdio.h>**: Standard-Verzeichnisse – Standard-Header
- **#include "answer.h"**: auch aktuelles Verzeichnis – eigene Header

**#define VIER 4**: Text ersetzen lassen – Konstante definieren

## 3 Bibliotheken

### 3.1 Der Präprozessor

**#include**: Text einbinden

- **#include <stdio.h>**: Standard-Verzeichnisse – Standard-Header
- **#include "answer.h"**: auch aktuelles Verzeichnis – eigene Header

**#define VIER 4**: Text ersetzen lassen – Konstante definieren

- Kein Semikolon!

## 3 Bibliotheken

### 3.1 Der Präprozessor

**#include**: Text einbinden

- **#include <stdio.h>**: Standard-Verzeichnisse – Standard-Header
- **#include "answer.h"**: auch aktuelles Verzeichnis – eigene Header

**#define VIER 4**: Text ersetzen lassen – Konstante definieren

- Kein Semikolon!
- Berechnungen in Klammern setzen:  
**#define VIER (2 + 2)**

## 3 Bibliotheken

### 3.1 Der Präprozessor

**#include**: Text einbinden

- **#include** <stdio.h>: Standard-Verzeichnisse – Standard-Header
- **#include** "answer.h": auch aktuelles Verzeichnis – eigene Header

**#define** VIER 4: Text ersetzen lassen – Konstante definieren

- Kein Semikolon!
- Berechnungen in Klammern setzen:  
**#define** VIER (2 + 2)
- Konvention: Großbuchstaben

## 3 Bibliotheken

### 3.2 Bibliotheken einbinden

Inhalt der Header-Datei: externe Deklarationen

```
extern int answer (void);
```

```
extern int printf (__const char *__restrict __format, ...);
```

Funktion wird „anderswo“ definiert

- separater C-Quelltext: mit an `gcc` übergeben
- Zusammenfügen zu ausführbarem Programm durch den *Linker*
- vorcompilierte Bibliothek: `-lfoo`  
= Datei `libfoo.a` in Standard-Verzeichnis



## 3.3 Bibliothek verwenden (Beispiel: GTK+)

- `#include <gtk/gtk.h>`

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

- `#include <gtk/gtk.h>`
- Mit `pkg-config --cflags --libs` erfährt man, welche Optionen und Bibliotheken man an `gcc` übergeben muß:

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

- `#include <gtk/gtk.h>`
- Mit `pkg-config --cflags --libs` erfährt man, welche Optionen und Bibliotheken man an `gcc` übergeben muß:

```
$ pkg-config --cflags --libs gtk+-3.0
-pthread -I/usr/include/gtk-3.0 -I/usr/include/at-spi2-atk/2.0 -I/usr/include/at-spi-2.0 -I/usr/include/dbus-1.0 -I/usr/lib/x86_64-linux-gnu/dbus-1.0/include -I/usr/include/gtk-3.0 -I/usr/include/gio-unix-2.0/ -I/usr/include/cairo -I/usr/include/pango-1.0 -I/usr/include/harfbuzz -I/usr/include/pango-1.0 -I/usr/include/atk-1.0 -I/usr/include/cairo -I/usr/include/pixman-1 -I/usr/include/freetype2 -I/usr/include/libpng16 -I/usr/include/gdk-pixbuf-2.0 -I/usr/include/libpng16 -I/usr/include/glib-2.0 -I/usr/lib/x86_64-linux-gnu/glib-2.0/include -lgtk-3 -lgdk-3 -lpangocairo-1.0 -lpango-1.0 -latk-1.0 -lcairo-gobject -lcairo -lgdk_pixbuf-2.0 -lgio-2.0 -lgobject-2.0 -lglib-2.0
```

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

- `#include <gtk/gtk.h>`
- Mit `pkg-config --cflags --libs` erfährt man, welche Optionen und Bibliotheken man an `gcc` übergeben muß.

→ Compiler-Aufruf:

```
$ gcc -Wall -O hello-gtk.c -pthread -I/usr/include/gtk-3.0 -I/usr/include/at-spi2-atk/2.0 -I/usr/include/at-spi-2.0 -I/usr/include/dbus-1.0 -I/usr/lib/x86_64-linux-gnu/dbus-1.0/include -I/usr/include/gtk-3.0 -I/usr/include/gio-unix-2.0/ -I/usr/include/cairo -I/usr/include/pango-1.0 -I/usr/include/harfbuzz -I/usr/include/pango-1.0 -I/usr/include/atk-1.0 -I/usr/include/cairo -I/usr/include/pixman-1 -I/usr/include/freetype2 -I/usr/include/libpng16 -I/usr/include/gdk-pixbuf-2.0 -I/usr/include/libpng16 -I/usr/include/glib-2.0 -I/usr/lib/x86_64-linux-gnu/glib-2.0/include -lgtk-3 -lgdk-3 -lpangocairo-1.0 -lpango-1.0 -latk-1.0 -lcairo-gobject -lcairo -lgdk_pixbuf-2.0 -lgio-2.0 -lgobject-2.0 -lglib-2.0 -o hello-gtk
```

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

- `#include <gtk/gtk.h>`
- Mit `pkg-config --cflags --libs` erfährt man, welche Optionen und Bibliotheken man an `gcc` übergeben muß.

→ Compiler-Aufruf:

```
$ gcc -Wall -O hello-gtk.c $(pkg-config --cflags --libs)
    -o hello-gtk
```

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

- `#include <gtk/gtk.h>`
- Mit `pkg-config --cflags --libs` erfährt man, welche Optionen und Bibliotheken man an `gcc` übergeben muß.

→ Compiler-Aufruf:

```
$ gcc -Wall -O hello-gtk.c $(pkg-config --cflags --libs)  
-o hello-gtk
```

Optionen:

u. a. viele Include-Verzeichnisse:

`-I/usr/include/gtk-3.0`

Bibliotheken:

u. a. `-lgtk-3 -lcairo`

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

Selbst geschriebene Funktion übergeben: *Callback*

```
gboolean draw (GtkWidget *widget, cairo_t *c, gpointer data)
{
    /* Zeichenbefehle */
    ...

    return FALSE;
}

...
```

```
g_signal_connect (drawing_area, "draw", G_CALLBACK (draw), NULL);
```


→ GTK+ ruft immer dann, wenn es etwas zu zeichnen gibt,  
die Funktion `draw` auf.

### 3.3 Bibliothek verwenden (Beispiel: GTK+)

Selbst geschriebene Funktion übergeben: *Callback*

```
gboolean draw (GtkWidget *widget, cairo_t *c, gpointer data)
{
    /* Zeichenbefehle */
    ...
    return FALSE;
}
...
```

repräsentiert den  
Bildschirm, auf den  
gezeichnet werden soll



```
g_signal_connect (drawing_area, "draw", G_CALLBACK (draw), NULL);
```

→ GTK+ ruft immer dann, wenn es etwas zu zeichnen gibt,  
die Funktion `draw` auf.



### 3.3 Bibliothek verwenden (Beispiel: GTK+)

Selbst geschriebene Funktion übergeben: *Callback*

```
gboolean draw (GtkWidget *widget, cairo_t *c, gpointer data)
{
    /* Zeichenbefehle */
    ...
    return FALSE;
}
```

repräsentiert den  
Bildschirm, auf den  
gezeichnet werden soll

optionale Zusatzinformationen  
für draw(), typischerweise  
ein Zeiger auf ein struct

```
g_signal_connect (drawing_area, "draw", G_CALLBACK (draw), NULL);
```

→ GTK+ ruft immer dann, wenn es etwas zu zeichnen gibt,  
die Funktion `draw` auf.

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

Selbst geschriebene Funktion übergeben: *Callback*

```
gboolean timer (GtkWidget *widget)
{
    /* Rechenbefehle */
    ...

    gtk_widget_queue_draw_area (widget, 0, 0, WIDTH, HEIGHT);
    g_timeout_add (50, (GSourceFunc) timer, widget);
    return FALSE;
}

...

g_timeout_add (50, (GSourceFunc) timer, drawing_area);
```

→ GTK+ ruft nach 50 Millisekunden die Funktion `timer` auf.

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

Selbst geschriebene Funktion übergeben: *Callback*

```
gboolean timer (GtkWidget *widget)
{
    /* Rechenbefehle */
    ...
```

Dieser Bereich soll  
neu gezeichnet werden.



```
    gtk_widget_queue_draw_area (widget, 0, 0, WIDTH, HEIGHT);
    g_timeout_add (50, (GSourceFunc) timer, widget);
    return FALSE;
}
```

```
...

g_timeout_add (50, (GSourceFunc) timer, drawing_area);
```

→ GTK+ ruft nach 50 Millisekunden die Funktion `timer` auf.

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

Selbst geschriebene Funktion übergeben: *Callback*

```
gboolean timer (GtkWidget *widget)
{
    /* Rechenbefehle */
    ...
```

Dieser Bereich soll  
neu gezeichnet werden.



```
    gtk_widget_queue_draw_area (widget, 0, 0, WIDTH, HEIGHT);
    g_timeout_add (50, (GSourceFunc) timer, widget);
    return FALSE;
}
```

In weiteren 50 Millisekunden soll

... die Funktion erneut aufgerufen werden.

```
g_timeout_add (50, (GSourceFunc) timer, drawing_area);
```


→ GTK+ ruft nach 50 Millisekunden die Funktion *timer* auf.

## 3.3 Bibliothek verwenden (Beispiel: GTK+)

Selbst geschriebene Funktion übergeben: *Callback*

```
gboolean timer (GtkWidget *widget)
{
    /* Rechenbefehle */
    ...
```

Dieser Bereich soll  
neu gezeichnet werden.



```
    gtk_widget_queue_draw_area (widget, 0, 0, WIDTH, HEIGHT);
    g_timeout_add (50, (GSourceFunc) timer, widget);
    return FALSE;
}
```

In weiteren 50 Millisekunden soll  
die Funktion erneut aufgerufen werden.

Explizite Typumwandlung  
eines Zeigers (später)



```
g_timeout_add (50, (GSourceFunc) timer, drawing_area);
```

→ GTK+ ruft nach 50 Millisekunden die Funktion `timer` auf.

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$\frac{dx}{dt} = v_x \quad \Rightarrow \quad x(t) = x_0 + v_x t$$

$$\frac{dx}{dt} = v_x \quad \Rightarrow \quad dx = v_x \cdot dt \\ x += dx$$

$$\frac{dy}{dt} = v_y \quad \Rightarrow \quad y(t) = y_0 + v_y t$$

$$\frac{dy}{dt} = v_y(t) \quad \Rightarrow \quad y(t) = y_0 + \int v_y(t) dt = y_0 + v_{y0} t - \frac{1}{2} g t^2$$

$$\frac{dv_y}{dt} = -g \quad \Rightarrow \quad v_y(t) = v_{y0} - g t$$

$$\frac{dv_y}{dt} = -g \quad \Rightarrow \quad dv_y = -g \cdot dt \\ v_y += -g \cdot dt$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t)$$

$$y'(t) = v_y(t)$$

$$v'_x(t) = 0$$

$$v'_y(t) = -g$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t) \qquad x(t) = \int v_x(t) dt$$

$$y'(t) = v_y(t) \qquad \Rightarrow \qquad y(t) = \int v_y(t) dt$$

$$v'_x(t) = 0 \qquad v_x(t) = \int 0 dt$$

$$v'_y(t) = -g \qquad v_y(t) = \int -g dt$$



# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t) \qquad x(t) = \int v_x(t) dt$$

$$y'(t) = v_y(t) \qquad y(t) = \int v_y(t) dt$$

$\Rightarrow$

$$v'_x(t) = 0 \qquad v_x(t) = \int 0 dt = v_{0x}$$

$$v'_y(t) = -g \qquad v_y(t) = \int -g dt$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t) \qquad x(t) = \int v_x(t) dt = \int v_{0x} dt$$

$$y'(t) = v_y(t) \qquad y(t) = \int v_y(t) dt$$

$\Rightarrow$

$$v'_x(t) = 0 \qquad v_x(t) = \int 0 dt = v_{0x}$$

$$v'_y(t) = -g \qquad v_y(t) = \int -g dt$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t) \qquad x(t) = \int v_x(t) dt = \int v_{0x} dt = x_0 + v_{0x} \cdot t$$

$$y'(t) = v_y(t) \qquad y(t) = \int v_y(t) dt$$

$\Rightarrow$

$$v'_x(t) = 0 \qquad v_x(t) = \int 0 dt = v_{0x}$$

$$v'_y(t) = -g \qquad v_y(t) = \int -g dt$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t) \qquad x(t) = \int v_x(t) dt = \int v_{0x} dt = x_0 + v_{0x} \cdot t$$

$$y'(t) = v_y(t) \qquad y(t) = \int v_y(t) dt$$

$\Rightarrow$

$$v'_x(t) = 0 \qquad v_x(t) = \int 0 dt = v_{0x}$$

$$v'_y(t) = -g \qquad v_y(t) = \int -g dt = v_{0y} - g \cdot t$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t) \qquad x(t) = \int v_x(t) dt = \int v_{0x} dt = x_0 + v_{0x} \cdot t$$

$$y'(t) = v_y(t) \qquad \Rightarrow \qquad y(t) = \int v_y(t) dt = \int v_{0y} - g \cdot t dt$$

$$v'_x(t) = 0 \qquad v_x(t) = \int 0 dt = v_{0x}$$

$$v'_y(t) = -g \qquad v_y(t) = \int -g dt = v_{0y} - g \cdot t$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t) \quad x(t) = \int v_x(t) dt = \int v_{0x} dt = x_0 + v_{0x} \cdot t$$

$$y'(t) = v_y(t) \quad \Rightarrow \quad y(t) = \int v_y(t) dt = \int v_{0y} - g \cdot t dt = y_0 + v_{0y} \cdot t - \frac{1}{2}gt^2$$

$$v'_x(t) = 0 \quad v_x(t) = \int 0 dt = v_{0x}$$

$$v'_y(t) = -g \quad v_y(t) = \int -g dt = v_{0y} - g \cdot t$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

### Beispiel 1: Gleichmäßig beschleunigte Bewegung

$$x'(t) = v_x(t) \quad x += v_x * dt;$$

$$y'(t) = v_y(t) \quad y += v_y * dt;$$

$\Rightarrow$

Siehe: [gtk-13.c](#)

$$v'_x(t) = 0 \quad v_x += 0 * dt;$$

$$v'_y(t) = -g \quad v_y += -g * dt;$$

# 5 Algorithmen

## 5.1 Differentialgleichungen

**Beispiel 1: Gleichmäßig beschleunigte Bewegung**

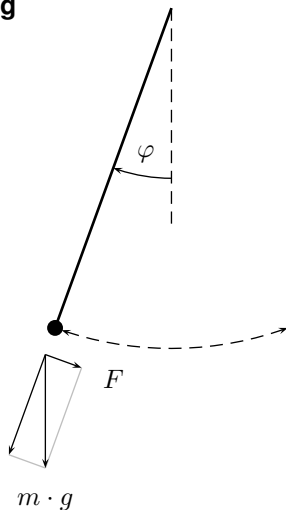
**Beispiel 2: Mathematisches Pendel**

$$\varphi'(t) = \omega(t)$$

$$\omega'(t) = -\frac{g}{l} \cdot \sin \varphi(t)$$

- Von Hand (analytisch):  
Lösung raten (Ansatz), Parameter berechnen
- Mit Computer (numerisch):  
Eulersches Polygonzugverfahren

```
phi += dt * omega;  
omega += - dt * g / l * sin (phi);
```





# 5 Algorithmen

## 5.1 Differentialgleichungen

**Beispiel 1: Gleichmäßig beschleunigte Bewegung**

**Beispiel 2: Mathematisches Pendel**

$$\varphi'(t) = \omega(t)$$

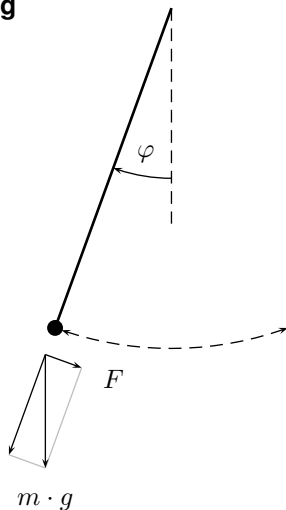
$$\omega'(t) = -\frac{g}{l} \cdot \sin \varphi(t)$$

- Von Hand (analytisch):  
Lösung raten (Ansatz), Parameter berechnen
- Mit Computer (numerisch):  
Eulersches Polygonzugverfahren

```
phi += dt * omega;  
omega += - dt * g / l * sin (phi);
```

**Beispiel 3: Weltraum-Simulation**

Praktikumsaufgabe



# Hardwarenahe Programmierung

<https://gitlab.cvh-server.de/pgerwinski/hp.git>

## 1 Einführung

## 2 Einführung in C

## 3 Bibliotheken

3.1 Der Präprozessor

3.2 Bibliotheken einbinden

3.3 Bibliotheken verwenden

3.4 Projekt organisieren: make

## 4 Hardwarenahe Programmierung

4.1 Bit-Operationen

4.2 I/O-Ports

4.3 Interrupts

...

## 5 Algorithmen

5.1 Differentialgleichungen

...

...