

9.7 GTK/Eingabefelder

9.7.1 Problem

An einem digitalen Speicheroszilloskop wird eine sinusförmige Spannung mit einem bestimmten Spitze-Spitze-Wert U_{p-p} gemessen. Daraus soll durch ein GUI-Programm der Effektivwert U_{eff} der Spannung berechnet werden.

Dazu muss zuerst der Spitze-Spitze-Wert (in V) eingegeben werden. Nach dem Klick auf die Schaltfläche mit dem Titel *Berechnen* soll in einem Textfeld der Effektivwert der Spannung ausgegeben werden.

9.7.2 Eingabefeld

Hier fehlt bisher eine Funktion wie `scanf()`, die die Eingabe eines Wertes ermöglicht. Statt einer Funktion findet man wiederum ein Widget für ein Eingabefeld (*entry*). Zuerst muss es erzeugt werden:

```

1   GtkWidget *eingabe;    // wie immer
2   eingabe=gtk_entry_new();           // entweder so
3   eingabe=gtk_entry_new_with_max_length(40); // oder so

```

Dann kann man, wenn man will, einen Vorgabetext festlegen.

```

1   gtk_entry_set_text(GTK_ENTRY(eingabe), "Hallo");

```

Man kann durch Gtk jederzeit die Eingabe überwachen lassen:

```

1   g_signal_connect(eingabe, "changed", funktion, NULL);

```

An den Inhalt der Eingabe kommt man mit einer weiteren Funktion:

```

1   const char *p=gtk_entry_get_text(GTK_ENTRY(eingabe));

```

In `p` steht dann die Startadresse einer Zeichenkette zur Verfügung. Für die Umwandlung in eine Zahl braucht man danach weitere Funktionen wie `atof()`, `atol()` oder `sscanf()`.

Alles zusammen sieht man im Programm `eingabel.c`:

```

1 #include <gtk/gtk.h>
2 #include <math.h>
3 #include <stdlib.h>
4 void ueffberechnen(GtkWidget *eingabe, gpointer muell)
5 {
6     double upp, ueff;
7     const char *p;
8     p=gtk_entry_get_text(GTK_ENTRY(eingabe));
9     upp=atof(p);
10    ueff=upp/(2.0*M_SQRT2);
11    g_print("Effektivwert: %f_V\n", ueff);
12 }
13 int main(int argc, char *argv[])
14 {
15     GtkWidget *fenster;
16     GtkWidget *eingabe;
17     gtk_init(&argc, &argv);
18     fenster=gtk_window_new(GTK_WINDOW_TOPLEVEL);
19     eingabe=gtk_entry_new();
20     gtk_container_add(GTK_CONTAINER(fenster), eingabe);
21     g_signal_connect(G_OBJECT(fenster), "delete_event",

```

```

22         gtk_main_quit, NULL);
23     g_signal_connect(G_OBJECT(eingabe), "changed",
24                     G_CALLBACK(ueffberechnen), NULL);
25     gtk_widget_show_all(fenster);
26     gtk_main();
27     return 0;
28 }

```

Zeile 16 Hier wird der Merker für das Eingabefeld vereinbart.

Zeile 19 Das Eingabefeld wird angelegt.

Zeile 20 Das Eingabefeld wird in das Fenster gelegt (hier ohne Schachtel, weil es kein weiteres Widget im Fenster geben soll).

Zeile 23 Bei jeder Eingabe in das Eingabefeld wird das Signal "changed" aktiviert. Damit wird die entsprechende Funktion aufgerufen.

Zeile 4 Der Aufruf der Callback-Funktion wurde vom Widget `eingabe` veranlasst. Damit erhält die Callback-Funktion den Merker für dieses Widget als ersten Parameter. Der zweite Parameter wird diesmal also gar nicht gebraucht.

Zeile 8 Mit dieser Funktion holt man den aktuellen Wert aus dem Eingabefeld. Das Makro dient wieder nur zur Überprüfung, ob es sich bei `eingabe` wirklich um ein Eingabefeld handelt.

Zeile 9 Die Zeichenkette wird in eine Gleitkommazahl umgewandelt.

Zeile 11 Die Ausgabe geht auf das Konsolenfenster.



Abbildung 1: `eingabel.c`

Das Programm führt in offenbar in die Sackgasse. Startet man `eingabel` und gibt 2,825 ein, erhält man die Ausgabe:

```

Terminal
schueler@debian964:~$ eingabel
Effektivwert: 0,707107 V
Effektivwert: 0,707107 V
Effektivwert: 0,989949 V
Effektivwert: 0,997021 V
Effektivwert: 0,998788 V

```

Bei jeder Änderung irgendeines Wertes wird sofort etwas ausgegeben. Das ist für den Benutzer verwirrend. Besser ist es, erst nach dem Klick auf eine Schaltfläche die Ausgabe zu starten.

9.7.3 Eingabefeld, Schaltfläche und Ausgabefeld

Wie das geht, sieht man im Programm `eingabe2.c`. Hier wird erst nach Druck auf die Schaltfläche `knopf` die Funktion aufgerufen.

```

1 #include <gtk/gtk.h>
2 #include <math.h>
3 #include <stdlib.h>
4 #include <stdio.h>
5 struct gui_t
6 {
7     GtkWidget *fenster;
8     GtkWidget *schachtel;
9     GtkWidget *eingabe;
10    GtkWidget *knopf;
11    GtkWidget *ausgabe;
12 };
13 void ueffberechnen(GtkWidget *knopf, gpointer daten)
14 {
15     char ergebnis_str[80];
16     double upp, ueff;
17     const char *p;
18     struct gui_t *pgui=daten;
19     p=gtk_entry_get_text(GTK_ENTRY(pgui->eingabe));
20     upp=atof(p);
21     ueff=upp/(2.0*M_SQRT2);
22     sprintf(ergebnis_str, "Effektivwert: %f_V", ueff);
23     gtk_label_set_text(GTK_LABEL(pgui->ausgabe), ergebnis_str);
24 }
25 int main(int argc, char *argv[])
26 {
27     struct gui_t gui;
28     gtk_init(&argc, &argv);
29     gui.fenster=gtk_window_new(GTK_WINDOW_TOPLEVEL);
30     gui.schachtel=gtk_vbox_new(TRUE, 0);
31     gui.eingabe=gtk_entry_new();
32     gtk_entry_set_text(GTK_ENTRY(gui.eingabe), "Hier_Upp_eingeben...");
33     gui.knopf=gtk_button_new_with_mnemonic("_Effektivwert_berechnen");
34     gui.ausgabe=gtk_label_new("Effektivwert: _____");
35     gtk_container_add(GTK_CONTAINER(gui.fenster), gui.schachtel);
36     gtk_box_pack_start(GTK_BOX(gui.schachtel), gui.eingabe, TRUE, FALSE, 0);
37     gtk_box_pack_start(GTK_BOX(gui.schachtel), gui.knopf, TRUE, FALSE, 0);
38     gtk_box_pack_start(GTK_BOX(gui.schachtel), gui.ausgabe, TRUE, FALSE, 0);
39     g_signal_connect(G_OBJECT(gui.fenster), "delete_event",
40                     gtk_main_quit, NULL);
41     g_signal_connect(G_OBJECT(gui.knopf), "clicked",
42                     G_CALLBACK(ueffberechnen), &gui);
43     gtk_widget_show_all(gui.fenster);
44     gtk_main();
45     return 0;
46 }

```

Hier nur die Erweiterungen, die mit dem Eingabefeld zu tun haben:

Zeile 32 Der Startwert wird festgelegt. Man kann ihn auch als Eingabeaufforderung benutzen.

Zeile 41 Der Aufruf der Callback-Funktion wird durch die Schaltfläche veranlasst, nicht durch das Eingabefeld. Damit die Callback-Funktion den Inhalt der Schaltfläche bekommt, wird als zweiter Parameter ein Zeiger auf das ganze GUI des Programms übergeben.

Zeile 19 Es wird wieder die Funktion `get_entry_get_text()` benutzt. In `pgui>eingabe` steht der Merker für das Eingabefeld.

9.7.4 Auswertung

Mit Etikett, Schaltfläche und Eingabefeld ist die Funktionalität von `printf()` und `scanf()` abgedeckt. Damit kann man jetzt jedes konsolenbasierte Programm auf die graphische Benutzeroberfläche (GUI) holen. Für den Benutzer ist es etwas komfortabler, weil er sein Programm jederzeit beenden kann. Andererseits ist für den Programmierer der Aufwand stark gewachsen.