Erweiterte Display-Funktionen

Erweiterte Display-Funktionen

Durch Benutzung weiterer Standardfunktionen und unter Anwendung einiger grundlegender Programmierkonstruktionen lassen sich die Anzeigemöglichkeiten in unserem Programm modifizieren.

Zeichenweise Ausgabe

Der Text könnte zum Beispiel zeichenweise, mit kurzen Verzögerungen erfolgen:

```
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("G");
  delay(50);
  lcd.print("u");
  delay(50);
  lcd.print("t");
  delay(50);
  lcd.print("e");
  delay(50);
  lcd.print("n");
  delay(50);
  lcd.print(" ");
  delay(50);
  lcd.print("T");
  delay(50);
  lcd.print("a");
  delay(50);
  lcd.print("g");
  delay(50);
```

Nur – auf diese Weise programmiert man das natürlich nicht !

Zeichenweise Ausgabe mit Hilfe einer Schleife

Variablendeklaration
In einem Programm können Speicherplätze (Variablen) reserviert werden, um bestimmte Arten von Informationen aufzunehmen, z.B. "ganze Zahlen" (Datentyp:Integer), Zahlen mit Nachkommastellen (Datentyp:double) oder auch Texte (Datentyp:char).

Diese Variablen werden am Anfang des Programmes vereinbart (deklariert).

So könnten wir mit dieser Programmzeile

```
char ausgabe1[] = "Guten Tag";
```

eine Textvariable mit dem Namen "ausgabe1" deklarieren und diese mit dem Text "Guten Tag" vorbelegen

Wir benutzen nun eine "Schleife", in der derselbe Programmteil mehrfach durchlaufen wird, um jedes einzelne Zeichen des in der Variablen gespeicherten Textes nacheinander ausgeben zu lassen.
Das ist kürzer, übersichtlicher, wesentlich einfacher und nicht so fehleranfällig:

```
lcd.clear();
lcd.setCursor(3, 0);
for (int i=0; i<9; i++) {
  lcd.print(ausgabe1[i]);
  delay(100);
}
```

Erklärung
Ein Schleifenzähler (hier: "i") startet mit dem Wert "0" und wird - solange er kleiner ist als 9 (i<9) - bei jedem Schleifendurchlauf um 1 erhöht ("i++").
Der Programmcode innerhalb der geschwungenen Klammern "{" und "}" wird bei jedem Schleifendurchgang ausgeführt (hier: das "i"-te Zeichen unserer Textvariablen "ausgabe1" wird am Display ausgegeben).
Durch den Befehl "delay(100)" wird eine Verzögerung von einer Zehntelsekunde bei der Ausgabe der einzelnen Zeichen eingebaut.

Weitere Funktionen und Möglichkeiten, die Ausgabe zu beeinflussen

rightToLeft
Durch diesen Befehl (er ist Bestandteil der LCD-Bibliothek) werden die Buchstaben eines Textes nicht von links nach rechts auf das Display geschrieben, sondern umgekehrt von rechts nach links.

Wir können also eine (Text-)Variable "ausgabe2" deklarieren und vorbesetzen…

```
char ausgabe2[] = "! nemmoklliW";
```

…und diese zeichenweise von rechts nach links auf das Display schreiben lassen:

```
lcd.setCursor(13, 1);
lcd.rightToLeft();
for (int i = 0; i < 12; i++) {
  lcd.print(ausgabe2[i]);
  delay(100);
}
lcd.leftToRight();
```

Der letzte Befehl "lcd.leftToRight()" stellt die ursprüngliche Schreibrichtung wieder ein.

Display ein- und ausschalten

Mit den Befehlen "lcd.display()" und "lcd.noDisplay()" kann das Display bei Bedarf ein- und ausgeschaltet werden. Um also die Anzeige am Display fünfmal "blinken" zu lassen, könnten diese Programmzeilen verwendet werden:

```
for (int i = 0; i < 5; i++) {
  lcd.noDisplay();
  delay(500);
  lcd.display();
  delay(500);
}
```

Displaytext spaltenweise "scrollen"

Mit den Befehlen "lcd.scrollDisplayLeft()" und "lcd.ScrollDisplayRight()" kann man einen Text, der bereits am Display angezeigt wird, um eine Spalte nach links oder rechts "verschieben".

Mit folgendem Programmcode "scrollt" der Displaytext zuerst vollständig (16 Zeichen) nach links aus dem Display heraus, erscheint dann wieder und "läuft" 30 Zeichen nach rechts über das gesamte Display und aus diesem heraus, um dann – von rechts kommend – wieder (mit 14 Zeichen) auf dem Display zu erscheinen:

```
for (int i = 0; i < 16; i++) {
  lcd.scrollDisplayLeft();
  delay(50);
}
for (int i = 0; i < 30; i++) {
  lcd.scrollDisplayRight();
  delay(50);
}
for (int i = 0; i < 14; i++) {
  lcd.scrollDisplayLeft();
  delay(50);
}
```

Mehrzeiligen Text am Display ausgeben

Um einen Text, der aus mehr als zwei Zeilen besteht, nacheinander am Display anzeigen zu lassen, ist wieder eine Schleife hilfreich.

Zuerst wäre allerdings eine Textvariable (hier: "MultiRow") zu deklarieren, die aus mehreren (hier: 4) Textzeilen besteht:

```
char* MultiRow[4] = {"Dies ist ein" , "Beispiel fuer" , "einen Text mit" , "mehreren Zeilen "}
```

In einer Schleife können diese Zeilen dann nacheinander am Display angezeigt werden, wobei nach jeweils drei Sekunden die oberste Zeile verschwindet und unten die nächste Zeile angezeigt wird.

```
lcd.clear();
lcd.setCursor(0, 0);
for (int i = 0; i < 3; i++) {
  lcd.setCursor(0, 0);
  lcd.print(MultiRow[i]);
  lcd.setCursor(0, 1);
  lcd.print(MultiRow[i + 1]);
  delay(3000);
}
```

Wenn Sie die Funktionen ausprobieren wollen, finden Sie auf der nächsten Seite den Code für alle gezeigten Funktionen, den Sie per Copy&Paste in Ihren eigenen Sketch einfügen können.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

char ausgabe1[] = "Guten Tag";
char ausgabe2[] = "! nemmoklliW";
char* MultiRow[4] = {"  Dies ist ein  ", "  Beispiel fuer ", " einen Text mit ", "mehreren
Zeilen "};


void setup() {
  lcd.init();
  lcd.backlight();
}


void loop() {
  lcd.clear();

  lcd.setCursor(3, 0);
  for (int i = 0; i < 9; i++) {
    lcd.print(ausgabe1[i]);
    delay(100);
  }

  lcd.setCursor(13, 1);
  lcd.rightToLeft();
  for (int i = 0; i < 12; i++) {
    lcd.print(ausgabe2[i]);
    delay(100);
  }
  lcd.leftToRight();

  delay(3000);

  for (int i = 0; i < 5; i++) {
    lcd.noDisplay();
    delay(500);
    lcd.display();
    delay(500);
  }

  delay(3000);


  for (int i = 0; i < 16; i++) {
    lcd.scrollDisplayLeft();
    delay(50);
  }
  for (int i = 0; i < 30; i++) {
    lcd.scrollDisplayRight();
    delay(50);
  }
  for (int i = 0; i < 14; i++) {
    lcd.scrollDisplayLeft();
    delay(50);
  }

  delay(3000);

  lcd.clear();
  lcd.setCursor(0, 0);
  for (int i = 0; i < 3; i++) {
    lcd.setCursor(0, 0);
    lcd.print(MultiRow[i]);
    lcd.setCursor(0, 1);
    lcd.print(MultiRow[i + 1]);
    delay(3000);
  }

  delay(1000);

}
```