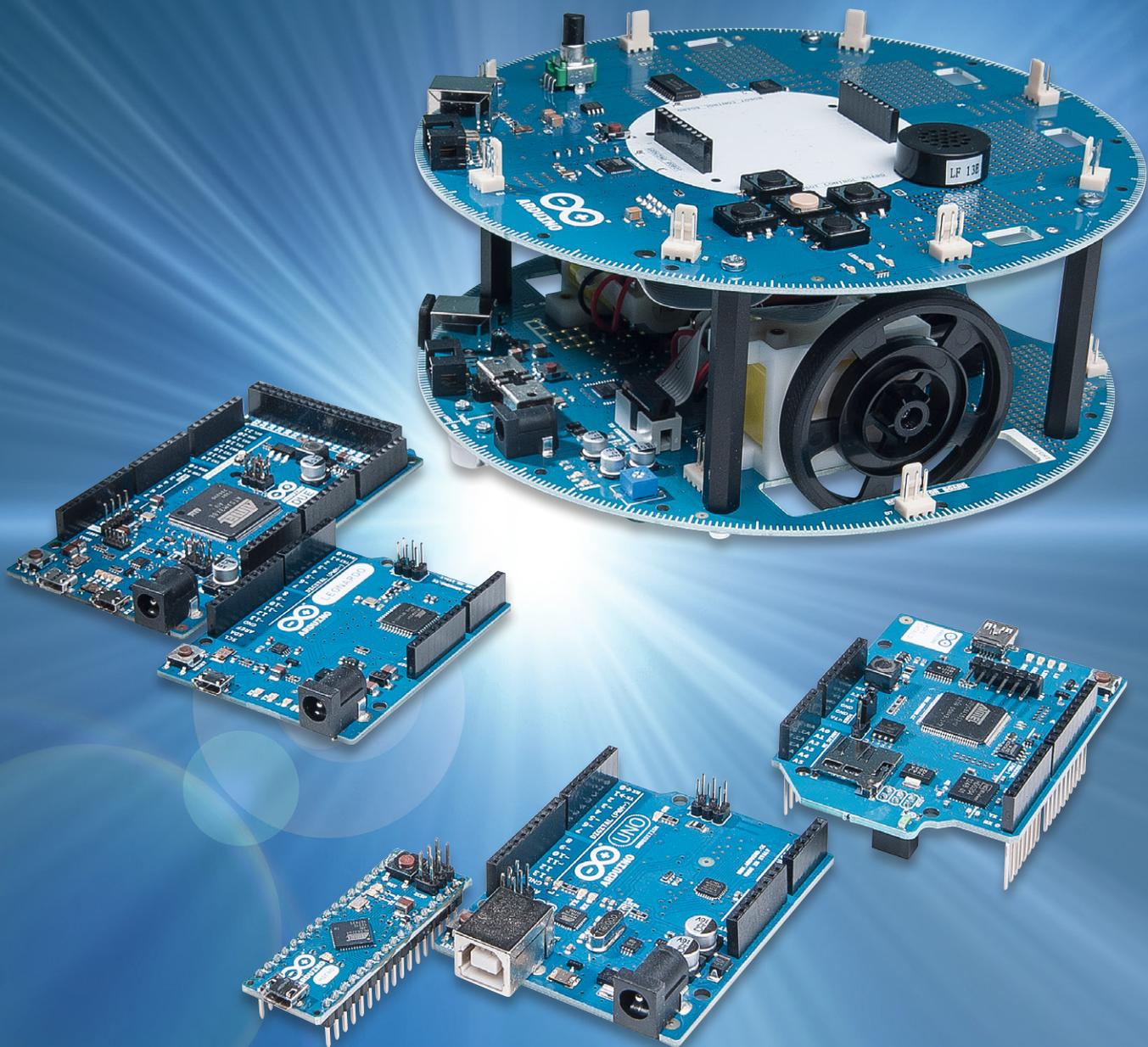




Arduino verstehen und anwenden

Teil 17: Alphanumerische LC-Displays





Will man mit einem Mikrocontroller umfangreichere Informationen darstellen, sind LC-Displays das Mittel der Wahl. Im Gegensatz zu 7-Segment-Displays können LCD-Punktmatrix-Anzeigen neben Ziffern auch Buchstaben und andere Zeichen gut lesbar anzeigen. Derartige Displays sind weit verbreitet und finden sich an Kaffee- oder Fahrkartenautomaten, Getränkemaschinen, Geschäftstelefonen usw. Entsprechend groß ist die Vielfalt der am Markt verfügbaren Typen.

Ein Display mit zwei Zeilen und 16 Zeichen pro Zeile kann bereits 32 Zeichen darstellen. Meist werden für die Anzeige eines einzelnen Zeichens $5 \times 8 = 40$ Pixel verwendet. Damit verfügt bereits ein kleines, zwei-zeiliges Display über $32 \times 40 = 1280$ Bildpunkte. Hier wird sofort klar, dass diese nicht mehr alle einzeln über die Arduino-Pins angesteuert werden können. Deshalb werden im Arduino-Umfeld auch praktisch ausnahmslos LC-Displays mit integrierten Hardware-Treibern eingesetzt.

Der Displaytyp HD44780 als Quasi-Standard

Für kleinere Displays mit ein bis vier Zeilen und 16 Zeichen pro Zeile hat sich hierfür im Lauf der Zeit ein gewisser Quasi-Standard etabliert. Die meisten LC-Displays dieser Größe verwenden den Controller-typ HD44780. Dieser Typ wurde ursprünglich von der Firma Hitachi entwickelt, ist aber inzwischen von vielen anderen Herstellern übernommen worden.

Dieser Displaytyp wird über 14 Pins angesteuert, bei vielen Modellen sind zwei weitere Pins für den Anschluss einer Hintergrundbeleuchtung vorgesehen. Die Standard-Pinbelegung zeigt [Tabelle 1](#).

LCD-Anzeigemodule mit Hitachi-HDD44780-kompatiblen Controllern existieren in zwei unterschiedlichen Varianten. Die eine verfügt über 16 Pins und über eine Hintergrundbeleuchtung. Die einfachere Version kommt ohne Beleuchtung aus und besitzt entsprechend nur 14 Pins.

Die 14 Pins können von einem Arduino wieder problemlos angesteuert werden. Die Spannung VEE sollte idealerweise über ein $10\text{-k}\Omega$ -Potentiometer zwischen GND und 5 V geschaltet werden, dann lässt sich der Displaykontrast gut einstellen. Bei vielen Displaytypen genügt aber auch bereits ein Widerstand von etwa $1\text{ k}\Omega$ nach Ground für einen guten Kontrast.

Das Display kann auf zwei verschiedene Arten angesteuert werden, im 8-Bit- oder im 4-Bit-Mode:

- Im 8-Bit-Mode werden alle acht Datenleitungen zur Ansteuerung verwendet, somit kann immer ein komplettes Byte übertragen werden.
- Im 4-Bit-Mode werden nur die oberen vier Datenleitungen (D4 bis D7) verwendet. Für die Übertragung eines Bytes sind somit zwei Zugriffe erforderlich. Dabei wird zunächst das höherwertige „Nibble“ (= 4 Bits), also Bit 4 bis Bit 7 übertragen, dann folgt das Niederwertige, also Bit 0 bis Bit 3. Die Datenleitungen D0 bis D3 des LCDs bleiben unbeschaltet.

Der 4-Bit-Mode hat den Vorteil, dass vier IO-Pins weniger benötigt werden. Der Nachteil der etwas geringeren Schreibgeschwindigkeit spielt meist keine Rolle, sodass die 4-Bit-Ansteuerung sehr häufig zum Einsatz kommt.

Für die Displaysteuerung sind neben den vier Datenleitungen (D4, D5, D6 und D7) noch die Anschlüsse RS, R/W und E erforderlich.

- Über RS (Register Select) wird ausgewählt, ob man einen Befehl oder ein Datenbyte an das LCD schicken möchte. Ist RS low, dann wird das ankommende Byte als Befehl interpretiert, ist RS high, dann wird das Byte direkt auf dem LCD als Zeichen angezeigt.
- R/W (Read/Write) legt fest, ob geschrieben oder gelesen werden soll. High Potential bedeutet lesen, bei Low können Daten auf die Displayeinheit geschrieben werden. Da man in den meisten Fällen nur Daten zum LCD sendet, kann man R/W fest auf GND legen und

Pin	Symbol	Level	Funktion
1	VSS	L	Stromversorgung 0V (GND)
2	VDD	H	Stromversorgung + 5 V
3	VEE	analog	Kontrastspannung
4	RS	H / L	Umschaltung Befehl / Daten
5	R/W	H / L	H = Read, L = Write
6	E	H	Enable (fallende Flanke)
7	D0	H / L	Display Data, LSB
8	D1	H / L	Display Data
9	D2	H / L	Display Data
10	D3	H / L	Display Data
11	D4 (D0)	H / L	Display Data
12	D5 (D1)	H / L	Display Data
13	D6 (D2)	H / L	Display Data
14	D7 (D3)	H / L	Display Data, MSB
15	-	frei	(evtl. LED-Hintergrundbeleuchtung)
16	-	frei	(evtl. LED-Hintergrundbeleuchtung)

Tabelle 1

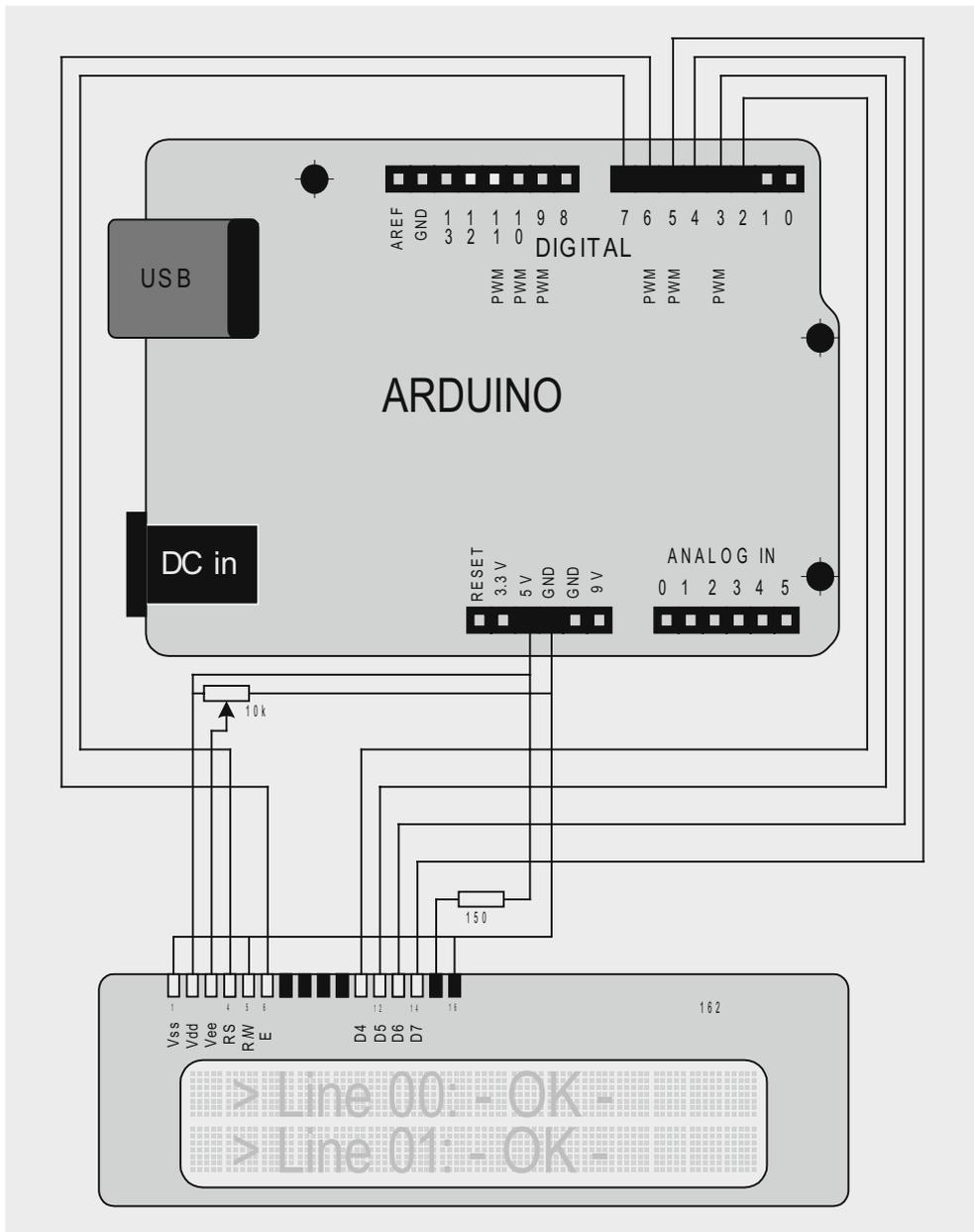


Bild 1: Anschluss eines zweizeiligen LCDs an den Arduino UNO

so einen weiteren IO-Pin am Controller einsparen. Man sollte dann jedoch nach jedem Befehl eine kurze Wartepause einlegen, um dem LCD-Controller Zeit zum Ausführen des Befehls zu geben.

- Der E-(Enable)-Anschluss signalisiert dem LCD, dass die Datenleitungen die korrekten Pegel angenommen haben und die Daten übernommen werden können.

Bereits in Teil 5 „Nutzung und Erstellung von Programmbibliotheken“ im ELVjournal 04/2014 wurde gezeigt, wie ein Arduino Micro mit einer LCD-Einheit verbunden werden kann. Bild 1 zeigt eine häufig verwendete Möglichkeit zur Verbindung eines LCDs mit dem Arduino UNO.

Wie arbeitet die LCD-Anzeige?

Im Rahmen des Beitrags zu den LED-7-Segment-Anzeigen und -Punktmatrixen wurde bereits diskutiert, wie einzelne Ziffern oder auch Buchstaben angezeigt werden können. Doch wenn es darum geht, das gesamte Alphabet oder auch Sonderzeichen wie z. B. *, #, % etc. auf diese Weise darzustellen, dann haben

diese Anzeigetypen die Grenzen des Möglichen erreicht. Immer wenn es erforderlich ist, kleine Datenmengen in Form von kurzen Textmitteilungen oder Zahlenreihen auszugeben, kommen deshalb häufig LCD-Anzeigen zum Einsatz (Bild 2).

LCDs weisen in ihrem Inneren Flüssigkristalle auf, die in Abhängigkeit von einer angelegten Spannung ihre Ausrichtung ändern. Dadurch wird auf das Display treffendes Licht polarisiert. Unter Verwendung geeigneter, fest eingebauter Polarisationsfilter kann so erreicht werden, dass ein bestimmter Displaybereich hell oder dunkel erscheint.

Auf diese Weise kann man ähnlich wie bei der LED Matrix-Displays nahezu beliebiger Größe aufbauen. Solche Anzeigeelemente nutzen meist aus einzelnen Punkten zusammengesetzte Matrizen. Damit lassen sich alle Arten von Zeichen (Ziffern, Buchstaben oder Sonderzeichen) darstellen. Darüber hinaus existieren auch Sonderformen, bei welchen die anzuzeigenden Muster bereits fest vorgegeben sind. Diese finden u. a. in Anzeigen für bestimmte Betriebszustände einer Maschine, in der Unterhaltungselektronik und in der Kfz-Technik Anwendung.

Bei Punktmatrix-Displays können durch geeignete Ansteuerung der einzelnen Punkte alle alphanumerischen Zeichen, d. h. alle Buchstaben und Zahlen, gut lesbar dargestellt werden. Darüber hinaus können auch einfache Grafiksymbole angezeigt werden (siehe Abschnitt „Erzeugung von Sonderzeichen“).



Echte großflächige Grafiken bleiben dagegen den sogenannten Grafikdisplays vorbehalten. Diese werden ausführlich im nächsten Beitrag zu dieser Artikelserie beschrieben.

Darstellung von Zeichen und Texten

Für die Ansteuerung des LC-Displays muss die Library LiquidCrystal eingebunden werden (siehe Teil 5 „Nutzung und Erstellung von Programm Bibliotheken“).

Für die Erzeugung eines LCD-Objekts müssen die folgenden Parameter übergeben werden:

- Pin Register Select (RS)
- Pin Enable (E)
- Pins der Datenleitungen D4 bis D7

Dies erfolgt über die Programmzeile

```
LiquidCrystal lcd(RS, E, D4, D5, D6, D7);
```

Die Klasse LiquidCrystal stellt dann eine Reihe von Methoden zur Verfügung, die die Ausgabe von Texten ermöglichen.

Die Methode „begin“ teilt dem LCD-Objekt die Anzahl der Spalten und der Zeilen des angeschlossenen Displays mit. Über

```
lcd.begin(COLS, ROWS);
```

wird die Anzahl der Zeichen pro Zeile (COLS) und die Zeilenzahl übergeben. Für ein 16-stelliges Display mit zwei Zeilen ist also

```
#define COLS 16
#define ROWS 2
```

zu setzen.

Die Methode „print“ teilt dem LCD-Objekt mit, welche Zeichen an das Display gesendet werden sollen. Sie ist vergleichbar mit der Printanweisung des Serial-Monitors.

Werden keine Angaben zur Position des anzuzeigenden Textes gemacht, wird er am Anfang der ersten Zeile gestartet.

Die Methode „setCursor“ positioniert den Cursor an die Stelle, an der die nachfolgende Textausgabe starten soll. Die Methode ist nullbasiert, d. h., die erste Zeile bzw. Spalte ist mit dem Index 0 versehen. Um in die zweite Zeile zu gelangen, ist daher der Wert 1 im Zeilenindex anzugeben:

```
lcd.setCursor(0, 1); // go to line 2
```

Die LCD-Methode „clear“ besitzt keine Parameter, löscht alle Zeichen aus der Anzeige und positioniert den Cursor in der linken oberen Ecke an der Koordinate 0, 0.

Es gibt HD44780-Varianten mit integriertem Vorwiderstand für die Hintergrundbeleuchtung. Hier können die Versorgungspins (Pins 15 und 16) direkt mit +5 V und GND verbunden werden. Im Allgemeinen ist allerdings ein Vorwiderstand erforderlich. Ein Blick in das entsprechende Datenblatt hilft in diesem Falle weiter. Liegt kein Datenblatt vor, so kann man testweise einen 1-k Ω -Vorwiderstand einsetzen.

Falls die Anzeige dann noch sehr dunkel erscheint, kann der Widerstandswert entsprechend angepasst werden. Natürlich muss man sich bei diesem Vorgehen bewusst sein, dass man das Display eventuell außerhalb seiner Spezifikationen betreibt und deshalb mit einer verkürzten Lebensdauer der Hintergrundbeleuchtung rechnen muss.

Will man die Helligkeit der Displaybeleuchtung softwareseitig steuern, so kann man die Versorgungsspannung der Hintergrund-LEDs mit einem PWM-Pin verbinden (z. B. Pin 9). Über

```
#define LCD_backlight 9
```

...

```
pinMode(LCD_backlight, OUTPUT);
analogWrite(LCD_backlight, 255);
```

kann man die Helligkeit der Hintergrundbeleuchtung softwaretechnisch steuern. Falls das verwendete Display eine sehr hohe Stromstärke benötigt (> 20 mA), ist ein entsprechender Treibertransistor vorzusehen.

Das vollständige Programm zur Ansteuerung eines zweizeiligen LCDs sieht damit so aus:

Bild 2: LCD-Anzeige in Aktion

```
// LCD control
#include <LiquidCrystal.h>

#define LCD_backlight 9
#define RS 7 // Register Select
#define E 6 // Enable
#define D4 5
#define D5 4
#define D6 3
#define D7 2
#define COLS 16
#define ROWS 2

LiquidCrystal lcd(RS, E, D4, D5, D6, D7);

void setup()
{ pinMode(LCD_backlight, OUTPUT);
  analogWrite(LCD_backlight, 255);
  lcd.begin(COLS, ROWS);
  lcd.print(" Line 01 -> OK");
  lcd.setCursor(0, 1); // go to line 2
  lcd.print(" Line 02 -> OK");
}

void loop(){}
```

Erzeugung von Sonderzeichen

Eine weitere interessante Methode ist

```
createChar()
```

Damit ist es möglich, eigene Zeichen zu erstellen und diese auf dem Display auszugeben.

Das Pixelmuster eines beliebigen Zeichens ist hier einfach als binäres Feld zu definieren, ähnlich wie es bereits bei den LED-Punktmatrizen (siehe vorhergehenden Artikel) zur Anwendung kam:

```
byte smiley[8] =
{ B00000,
  B00000,
  B01010,
  B00000,
  B00000,
  B10001,
  B01110,
  B00000
};
```

Über

```
lcd.createChar(CHAR, smiley);
```

wird dann der entsprechende Charakter erzeugt. Dieser kann dann wieder mit

```
lcd.write();
```

auf dem Display dargestellt werden.

Der folgende Sketch gibt so eine ganze Reihe von Sonderzeichen aus:



```
#define LCD_backlight 9
char love = 1, kid = 2, death = 3;

byte smiley[8] =
{ B00000,
  B00000,
  B01010,
  B00000,
  B00000,
  B10001,
  B01110,
  B00000
};

byte heart[8] =
{ B00000, B00000, B01010, B11111, B01110,
  B00100, B00000, B00000
};

byte cross[8] =
{ B00000, B00100, B00100, B11111, B00100,
  B00100, B00100, B00100
};

// include the library code:
#include <LiquidCrystal.h>

// initialize the library
// with interface pins
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup()
{ pinMode(LCD_backlight, OUTPUT);
  analogWrite(LCD_backlight, 255);
  lcd.begin(16, 2);
  lcd.createChar(kid, smiley);
  lcd.createChar(love, heart);
  lcd.createChar(death, cross);
}

void loop()
{ lcd.setCursor(0, 0);
  lcd.print("The brave man:");
  lcd.setCursor(0, 1);
  lcd.print("* ");
  lcd.write(love); lcd.print(" ");
  for (int i=1; i<=4; i++) lcd.write(kid); lcd.print(" ");
  lcd.write(death);
}
```

Aus Platzgründen wurden hier die Bitmaps für „Heart“ und „Cross“ in komprimierter Form dargestellt.

Das Ergebnis auf dem Display sieht so aus wie in Bild 3 dargestellt. Es stellt die Geschichte eines braven Iren in Kurzform dar [1].



Bild 3: LCD-Anzeige mit selbst definierten Sonderzeichen

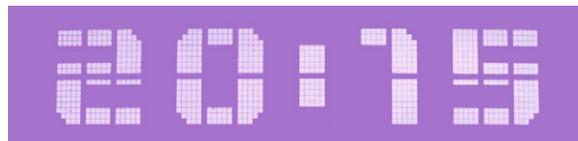


Bild 4: Großer Font

Mithilfe selbst definierter Fonts ist sogar die Darstellung großer Ziffern und Zeichen möglich (Bild 4). Hierzu müssen lediglich die erforderlichen Sonderzeichen festgelegt werden. Daraus können dann die großen Schriftzeichen zusammengesetzt werden. Eine Library hierzu findet sich unter [2].

Ausblick

Wie in diesem Artikel gezeigt wurde, eignen sich textbasierte LC-Displays nicht nur zur Anzeige von einfachen Ziffern und Zeichen, sondern sie erlauben darüber hinaus, wenn auch in beschränktem Umfang, die Darstellung einfacher Grafiken.

Will man dagegen umfangreichere grafische Darstellungen anzeigen, kann man ein Display des Typs KS108 einsetzen. Ähnlich wie das HD44780 werden KS108-Typen von verschiedenen Herstellern angeboten und es existiert eine Vielzahl von kompatiblen Modellen. Dieser Displaytyp bietet eine Grafikauflösung von immerhin 128 x 64 Pixeln.

Der nächste Artikel wird sich eingehend mit den Möglichkeiten der Grafikdarstellung auf diesem Displaytyp beschäftigen. Es wird dann u. a. eine Quasi-Analoguhr entstehen, und auf einem PC erzeugte Bitmaps werden vom Arduino auf das Display gezaubert werden. **ELV**



Weitere Infos:

[1] Die Geschichte des braven Iren:

[2] Library für große Fonts:

<http://code.google.com/p/phi-big-font/>

G. Spanner: Arduino – Schaltungsprojekte für Profis, Ektor-Verlag 2012, Best.-Nr. CI-10 94 45, € 39,80

Mikrocontroller-Onlinekurs, Franzis-Verlag, exklusiv für ELV, 2011, Best.-Nr. CI-10 20 44, € 99,-

Grundlagen zur elektronischen Schaltungstechnik finden sich in der E-Book-Reihe „Elektronik!“ (www.amazon.de/dp/B000XNCB02)

Lernpaket „AVR-Mikrocontroller in C programmieren“, Franzis-Verlag 2012, Best.-Nr. CI-10 68 46, € 129,-

G. Spanner: Lernpaket „Physical Computing“, Franzis-Verlag 2015, Best.-Nr. CI-12 21 81, € 99,-

Er wurde geboren
nahm ein Weib
zeugte 4 Kinder
und starb

Preisstellung Mai 2016 – aktuelle Preise im Web-Shop

Empfohlene Produkte	Best.-Nr.	Preis
Arduino UNO	CI-10 29 70	€ 27,95
Mikrocontroller-Onlinekurs	CI-10 20 44	€ 99,-
STN-LCD-Anzeigemodul, 2x 16 Zeichen	CI-05 41 84	€ 9,95

Alle Arduino-Produkte wie Mikrocontroller-Platinen, Shields, Fachbücher und Zubehör finden Sie unter: www.arduino.elv.de