Arduino verstehen und anwenden

Teil 18: Grafikdisplays

Im letzten Artikel zu dieser Serie wurde die Ansteuerung von alphanumerischen Displays ausführlich behandelt. Diese Displays sind für die Ausgabe von Texten, einfachen Daten und Messwerten vollkommen ausreichend. Über Sonderzeichen ermöglichen sie sogar in sehr beschränktem Umfang grafische Darstellungen.

Möchte man allerdings echte hochauflösende Grafiken erzeugen, so müssen entsprechend aufwendigere Displaytypen eingesetzt werden. Ähnlich wie bei den zeichenorientierten Anzeigen der TYP HD44780 hat bei den grafischen Displays die Controllerversion KS0108 eine Quasi-Standardfunktion übernommen.

Dieser Displaytyp bietet eine Auflösung von 128 x 64 Pixeln. Dies mag im Zeitalter der Megapixel-Kameras und HD-Monitore wenig erscheinen, für viele Mikrocontrolleranwendungen ist diese Auflösung aber vollkommen ausreichend. Displays vom Typ KS0108 oder einer kompatible Version sind bereits für deutlich unter 20 Euro erhältlich und stellen damit eine kostengünstige Möglichkeit dar, einen Arduino zu einem grafikfähigen System aufzurüsten.

Anschluss des Grafikdisplays an den Arduino

Im Vergleich zum zeichenbasierten Display sind beim Grafikdisplay deutlich mehr Verbindungsleitungen erforderlich. Zudem gibt es beim KS0108 mehrere Varianten, die in der Tabelle mit Typ 1 bis Typ 3 gekennzeichnet sind.

Die Tabelle 1 zeigt, wie die Anzeigeeinheit mit dem Arduino Uno zu verbinden ist. Für ein Display vom Typ 1 gilt dementsprechend z. B., dass der Digital-Pin Nr. 8 des Arduinos mit dem Pin Nr. 7 des Displays verbunden werden muss usw.

Wenn man methodisch vorgeht, ist der Anschluss des Grafikdisplays trotz der 16 erforderlichen Verbindungsleitungen rasch erledigt.

Die GLCD-Library

Die Ansteuerung eines Grafikdisplays ist natürlich mit erheblichem Aufwand verbunden. Wie im Arduino-Umfeld üblich, ist aber auch für

Bild 1: Arduino-Testausgabe auf dem Grafikdisplay

KS0108-kompatible Anzeigen eine Library verfügbar. Die Lib kann unter [1] kostenlos aus dem Internet geladen werden.

Die Bibliothek enthält mehrere Beispielprogramme. Um die korrekte Funktion des Displays zu testen kann der Beispielsketch

ks0108example.ino

geladen werden. Er zeigt nach einem Begrüßungsbildschirm verschiedene Fonts und schließlich eine sich bewegende Grafik an. Wenn das Display diese Ausgaben korrekt anzeigt, kann man davon ausgehen, dass die Verbindung zwischen Displayeinheit und Arduino einwandfrei ist (siehe Bild 1).

Die meisten KS0108-Varianten verfügen über ein eingebautes Backlight. Dieses kann über die Pins LED_A und LED_K versorgt werden.

Sie müssen über einen geeigneten Vorwiderstand (z. B. 220 Ω) mit 5 V bzw. Ground des Arduinos verbunden werden. Die mit „Pot" bezeichneten Display-Pins sind über ein Potentiometer (z. B. 10 KΩ)

Tabelle 1

				Digital-Pin							
Arduino-Pin	**5 V**	**Gnd**	**Pot**	**8**	**9**	**10**	**11**	**4**	**5**	**6**	**7**
Display-Typ 1	2	1	3	7	8	9	10	11	12	13	14
Display-Typ 2	1	2	3	4	5	6	7	8	9	10	11
Display-Typ 3	13	14	12	1	2	3	4	5	6	7	8

	Analog-Pin					Backlight-Versorgung		
Arduino-Pin	**0**	**1**	**2**	**3**	**4**	**LED_A**	**LED_K**	**Rst**
Display-Typ 1	15	16	5	4	6	19	20	17
Display-Typ 2	12	13	15	16	17	19	20	14
Display-Typ 3	15	16	10	11	9	19	20	18

mit dem Arduino zu verbinden. Der Schleifer des Potentiometers ist dabei an den jeweiligen Display-Pin zu legen. Die anderen beiden Anschlüsse kommen an die Versorgungsspannung bzw. den Ground des Arduinos. Damit kann dann der Kontrast des Displays manuell eingestellt werden. Im Zweifelsfall sollte man hierzu auch das Datenblatt der Anzeigeeinheit zu Rate ziehen. Weitere Informationen zum Anschluss von Displays mit Kontrastreglern finden sich auch im letzten Arduino-Beitrag „Alphanumerische LC-Displays" (ELVjournal 4/2016).

Nachdem das Grafikdisplay erfolgreich getestet wurde, können weitere interessante und nützliche Anwendungen auf den Arduino geladen werden.

Bild 2: Analog-Uhr

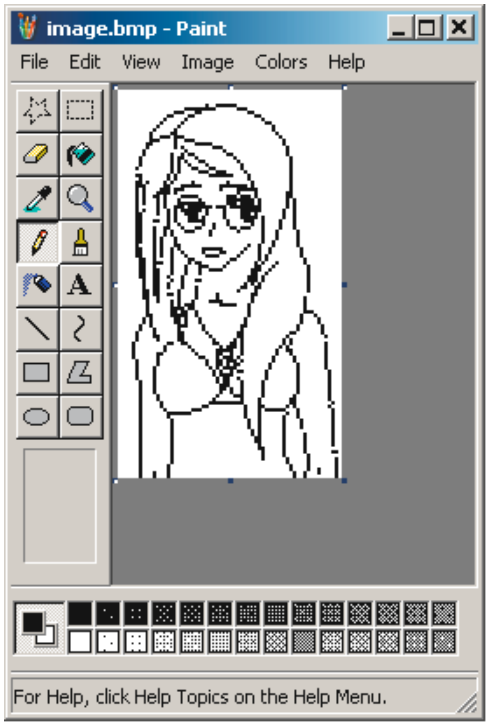

Bild 3: Erzeugung einer Bitmap

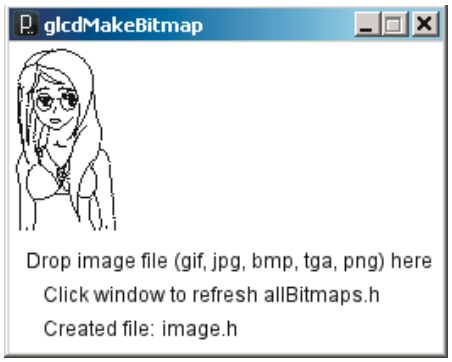

Bild 4: Das Konvertierungsfenster des Processing-Programms

Die Analog-Uhr

Obwohl die Digitaltechnik in alle Lebensbereiche eingezogen ist, erfreuen sich quasi-analoge Technologien immer noch großer Beliebtheit. Ein bekanntes Beispiel hierfür sind Analog-Uhren. Obgleich die eigentliche Zeittaktung vollkommen digital erfolgt, ist die Anzeige der aktuellen Uhrzeit mit Zeigern auf einem Ziffernblatt immer noch aktuell. Auch der Arduino kann auf dem Grafikdisplay eine Analog-Uhr anzeigen.

Der Sketch dazu findet sich ebenfalls unter den Beispielen zur GLCD-Library. Zusätzlich ist hier noch die Time-Bibliothek erforderlich. Diese kann unter [2] als time.zip geladen werden.

Nachdem die Bibliothek wie üblich installiert wurde, kann der Analoguhr-Sketch *clockFace.ino* geladen werden. Bild 2 zeigt die Darstellung der Uhr auf dem Display.

Ausgabe von Bitmap-Grafiken

Auf dem Display können nicht nur Messwerte und Daten in grafischer Form ausgegeben werden, sondern auch beliebige Abbildungen in Form von Bitmaps.

Um die Grafik anzeigen zu können, muss zunächst eine Bitmap (z. B. image.bmp) erzeugt werden. Hierfür können die bekannten Programme wie etwa Paint verwendet werden. Dabei ist darauf zu achten, dass die Datei bereits ein Format von 128 x 64 Pixel enthält (siehe Bild 3).

Dann muss diese Datei in ein Include-File (*.h-Datei) umgewandelt werden. Hierfür steht ein Programm namens glcdMakeBitmap zur Verfügung. Das in der GLCD-Bibliothek unter

/bitmaps/utils/glcdMakeBitmap

enthaltene Programm

glcdMakeBitmap.pde

ist nicht für die Arduino-IDE gedacht, sondern muss in „Processing" gestartet werden. Dabei handelt es sich um ein leistungsfähiges Grafikprogramm unter Windows. Es kann unter [3] kostenlos aus dem Internet geladen werden.

Processing wurde bereits in Teil 6 zu dieser Artikelserie (Sensortechnik und Messwerterfassung) kurz vorgestellt. Es weist starke Ähnlichkeiten mit der Arduino-IDE auf. Das ist kein Zufall, da die Arduino-Programmieroberfläche sehr eng an das ursprünglich am Massachusetts Institute for Technology (MIT) entwickelte Processing angelehnt ist.

Nachdem das Programm glcdMakeBitmap in Processing geladen und gestartet wurde, erscheint ein Fenster, in welches die zu konvertierende Grafik per Drag & Drop geladen werden kann (siehe Bild 4).

Auf diese Weise wird eine Datei „image.h" erzeugt, welche die Daten der Bitmap in einer für den Arduino verständlichen Form enthält.

Die Anweisungen

#include "image.h" und *GLCD.DrawBitmap(image, x, y);*

sorgen dann dafür, dass die Bitmap an der Position x, y auf dem Display erscheint. Die Datei image.h muss dazu natürlich mit in das Verzeichnis des aktuellen Sketches kopiert werden.

Das vollständige Programm dazu sieht so aus:

```
// Arduino graphics display

#include <glcd.h>
#include "image.h"

void setup()
{ GLCD.Init(NON_INVERTED);
  GLCD.ClearScreen();
  GLCD.DrawBitmap(image, 0,0, BLACK);
}

void  loop()
{}
```

Bild 5 zeigt das Ergebnis auf dem Display.

Der Thermograf

Moderne Heizungsanlagen verfügen über die vielfältigsten Steuer- und Regeleinrichtungen. Neben Zeitsteuerung und Vorlauftemperatur werden auch Parameter wie Innen- und Außentemperatur berücksichtigt.

Hier kann es schnell einmal vorkommen, dass ein kleiner Fehler in den Einstellungsparametern der Heizanlage zu unerwünschten Raumtemperaturprofilen führt. Dies ist natürlich neben dem mangelnden Komfort auch mit erhöhten Heizkosten verbunden.

Hier ist es deshalb sinnvoll, die Raumtemperatur unabhängig von der Heizanlage selbst zu überwachen. Dabei ist nicht nur die aktuelle Temperatur wichtig, sondern insbesondere auch der Temperaturverlauf über einen längeren Zeitraum wie etwa 24 oder 48 Stunden. Hier kommt das Grafikdisplay ins Spiel. Es erlaubt, eine Temperaturkurve (Bild 6) automatisch über einen längeren Zeitraum hinweg darzustellen.

Unerwünschte Temperaturverläufe, wie etwa eine überflüssige nächtliche Heizperiode, können so schnell erkannt werden.

Die Programmierung einer solchen Anwendung kann mit den Kenntnissen aus diesem Beitrag in kurzer Zeit durchgeführt werden.

Komplettgerät mit Grafikdisplay

Abschließend zeigt Bild 7 den kompletten Aufbau eines Geräts mit Grafikdisplay. Links unten ist das Potentiometer für die Einstellung des Kontrastes zu sehen. Ein solches Gerät macht durchaus einen bereits recht professionellen Eindruck und kann etwa als Thermograph auch in einem Wohnzimmer eingesetzt werden.

Ausblick

Mit diesem Artikel wird das umfassende Thema Displaytechnik abgeschlossen. Beginnend mit Sieben-Segmentanzeigen über LED-Matrizen und Alphanumerische LCD-Einheiten bis hin zu voll grafikfähigen Mini-Monitoren wurden die vielfältigen Möglichkeiten dieses wichtigen und interessanten Gebiets vorgestellt.

Im nächsten Beitrag wird es um das Thema Messtechnik gehen. Neben einfachen Messaufgaben wie die Erfassung von Strömen, Spannungen und Widerständen werden auch komplexere Geräte wie ein einfaches Oszilloskop und sogar ein Logic-Analyzer auf Arduino-Basis vorgestellt. ELV

Weitere Infos:

[1] http://code.google.com/p/glcd-arduino/downloads/list
[2] http://playground.arduino.cc/Code/Time
[3] http://processing.org/

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

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

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

G. Spanner: Elektor Praxiskurs AVR-XMEGA-Mikrocontroller, 2015, Best.-Nr. CJ-12 07 62, € 39,80

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. CJ-10 68 46, € 129,–

Preisstellung August 2016 – aktuelle Preise im Web-Shop

Empfohlene Produkte	Best.-Nr.	Preis
Arduino Uno	CJ-10 29 70	€ 27,95
Mikrocontroller-Onlinekurs	CJ-10 20 44	€ 99,–

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

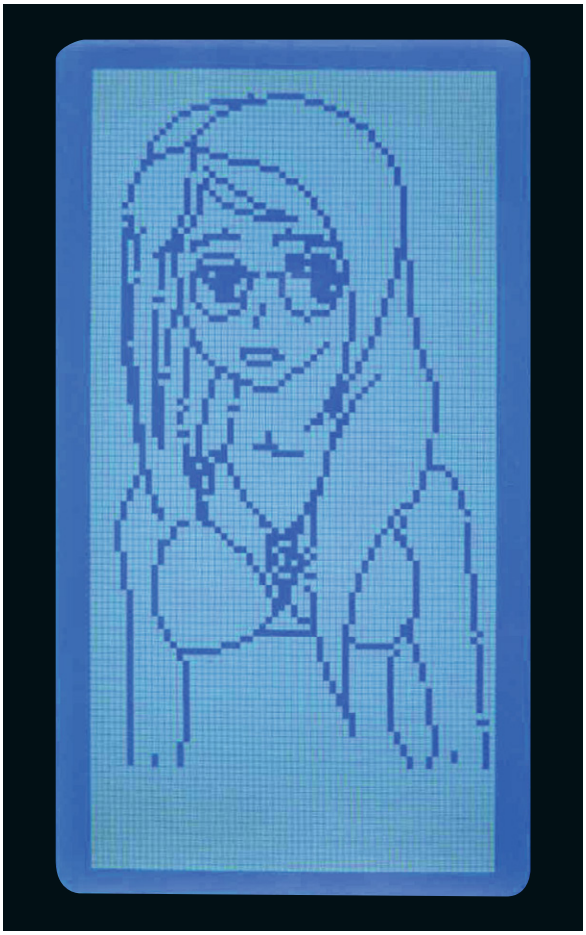

Bild 5: Ausgabe einer Bitmap auf dem Grafikdisplay

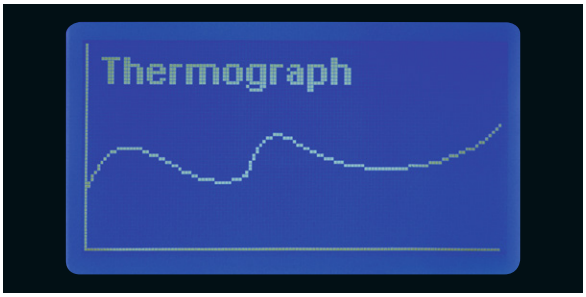

Bild 6: Der Thermograf in Aktion

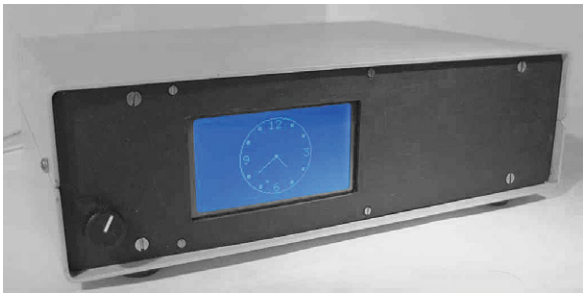

Bild 7: Komplettgerät mit grafischem Display