



Kleiner Universal-Monitor

LCD-Colour-Panel LCP 100

Teil 2

Nicht immer muss es der große PC-Monitor sein – in vielen Fällen genügt auch ein kleines (Zusatz-) Display, um bestimmte Informationen anzuzeigen. Das kompakte LCD-Colour-Panel im Handy-Displayformat ermöglicht genau dies – E-Mail-Eingang, ID3-Tags, andere gewünschte Informationen, ja sogar komplette Farbbilder werden in guter Qualität dargestellt. Nach der Hardwarebeschreibung folgt nun die der Software und der Schnittstelle für dieses interessante Projekt.

Software ganz offen

Die Beschreibung der Hardware und des Aufbaus des kleinen Zusatz-Displays im vorangegangenen „ELVjournal“ hat ganz sicher Appetit auf die Software-Anbindung an den PC gemacht.

Dabei ist die „Hauptarbeit“ schon erledigt, denn das Windows-Programm steht inklusive Treiber als C#-Projekt zum Download bereit. Daneben gehen wir hier natürlich noch auf die serielle Verbindung via USB ein, auch deren Protokoll und der verwendete Datenrahmen werden für eigene Applikationen offengelegt.

Die serielle Schnittstelle

Die Datenübertragung per USB arbeitet im Prinzip wie die inzwischen vielfach genutzte serielle Übertragung mit einem virtuellen COM-Port. Dabei werden für diesen folgende Einstellungen zugrunde gelegt:

- 19.200 Baud
- gerade Parität
- 1 Stoppbit

- 8 Datenbits
- Sendeauszeit: 10 Sekunden

Diese Einstellungen befinden sich für die PC-Software im C#-Quelltext. Auf dem LCP 100 sind diese in der Firmware implementiert.

Der Protokollrahmen

Der verwendete Protokollrahmen dient der Synchronisation und Absicherung der Datenübertragung zwischen dem LCP 100 und der PC-Software. Er ist für eine Übertragung von Nutzdaten variabler Länge bis zu einer maximale Länge von 65.535 Byte ausgelegt. Nach der Bildung der CRC16-Prüfsumme werden die Daten codiert und anschließend gesendet.

Inhalt und Aufbau

Die Länge der Nutzdaten wird in zwei Bytes abgelegt. Danach folgen die Nutzdaten und die 2 Byte der CRC16-Prüfsumme. Die Speicherung erfolgt nach dem Byte-Reihenfolge-Verfahren „Big-Endian“. Der CRC16-Algorithmus arbei-

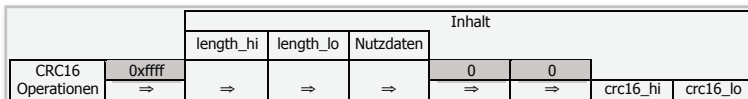


Tabelle 1: Die Bildung der CRC16-Prüfsumme

tet, wie in Tabelle 1 zu sehen, nach folgendem Prinzip. Erst wird die CRC16-Prüfsummenbildung mit 0xffff initialisiert und dann mit Hilfe des Polynoms 0x8005 über die Nutzdatenlänge und die Nutzdaten die Prüfsumme gebildet. Danach werden zwei Nullen in die CRC16-Prüfsumme „hineingeschiftet“. Die 2 Byte der Prüfsumme werden anschließend den Daten angehängt.

Codierung und Synchronisation

Natürlich wird neben der Synchronisation auch die Codierung der Steuerzeichen realisiert. Taucht eines der zwei Zeichen „STX“ oder „DLE“ in den Daten auf, sind diese zu codieren. Empfängerseitig wird diese Umsetzung jedoch automatisch wieder rückgängig gemacht, um die eigentlichen Nutzdaten wieder zu erhalten. Dabei gestaltet sich die eigentliche Codierung der Daten nach der Bildung der CRC16-Prüfsumme wie folgt:

In den Daten enthaltene Zeichen im Bereich von 0x00 bis 0x7f werden durch Steuerzeichen <DLE> (0x10) ersetzt, gefolgt von dem ursprünglichen Zeichen „verodert“ mit dem Wert 0x80. Danach werden im Inhalt enthaltene Steuerzeichen <STX> (0x02) und <DLE> (0x10) entsprechend der DLE-Codierung folgendermaßen ersetzt:

Inhalt	Codierung
<STX>	<DLE> + (<STX> 0x80)
<DLE>	<DLE> + (<DLE> 0x80)

Im letzten Schritt wird den so entstandenen DLE-codierten Daten vor der Übertragung das Steuerzeichen <STX> als Synchronisationsbyte vorangestellt:

<STX>	DLE-codierter Inhalt (kein <STX> mehr enthalten)
-------	--

Befehlssatz

Der LCP 100 verfügt für die Kommunikation mit der PC-Software über einen Befehlssatz mit einer Vielzahl von Instruktionen. Die Tabelle 2 gibt eine Übersicht über jeden Befehl und dessen zugehörige Parameter.

Die Installation der Hardware

Um den LCP 100 über den PC nutzen zu können, ist als Erstes der Virtual-Com-Port-Treiber von der Herstellerfirma Silabs (mit dem C#-Projekt als Download bereitstehend) zu installieren.

Danach wird das Gerät mittels USB-Kabel mit dem PC verbunden. Das Betriebssystem erkennt es als neues Gerät. Nach der durch den Assistenten erfolgreich ausgeführten Installation lässt sich die zugewiesene Nummer des COM-Ports im Gerätemanager von Windows nachschauen. Dies geschieht unter Windows XP über die Menüpunkte „Systemsteuerung“ -> „System“ -> Karteikarte „Hardware“ -> Button „Geräte-Manager“ -> Anschlüsse (COM und LPT).

Beschreibung	Befehl	Parameter (Anzahl an Bytes)	Parameterinformationen		Antwort
Allgemein					
PC-Verbindung	x	Status (1 Byte)	0x00 0x01	Verbindung deaktivieren Verbindung aktivieren	ACK NAK (Parameter falsch)
Geräteversion auslesen	v	kein Parameter notwendig			Version (2 Byte) Vorkomma-/Nachkommastelle NAK (Parameter falsch)
Display					
Display löschen (Display weiß)	c	kein Parameter notwendig			ACK NAK (Parameter falsch)
Testbild ausgeben	t	kein Parameter notwendig			ACK NAK (Parameter falsch)
Kommando senden (s. Datenblatt Sitronix)	o	Befehl (1 Byte)			ACK NAK (Parameter falsch)
Daten senden (s. Datenblatt Sitronix)	i	Daten (1 Byte)			ACK NAK (Parameter falsch)
Displaybeleuchtung ein/aus	z	Befehl (1 Byte)	0x00 0x01	Aus Ein	ACK NAK (Parameter falsch)
Daten					
Page speichern	p	Page nr (1 Byte) Daten (512 Byte)	0x0–0x1000 (0–4098)		ACK NAK (Parameter falsch)
Page anzeigen	s	Page nr (1 Byte) Anzahl der Datenbytes (2 Byte)	0x0–0x1000 (0–4098) 0x1–0x200 (1–512)		ACK NAK (Parameter falsch)
Page laden	l	Page nr (1 Byte) Anzahl der Datenbytes (2 Byte)	0x0–0x1000 (0–4096) 0x1–0x200 (1–512)		Datenrahmen mit den Nutzdaten NAK (Parameter falsch)

Tabelle 2: Der Befehlssatz für die Kommunikation mit dem PC

Das C#-Projekt

Im auf der ELV-Internetseite zum Download bereitgestellten C#-Projekt „LCP100Demo“ werden die Befehle durch einzelne Methoden realisiert. Durch die umfangreiche Kommentierung des Quelltextes kann diese Applikation schnell und einfach den Weg in die eigene Anwendung finden bzw. ist den eigenen Bedürfnissen anpassbar. Auch bietet diese Software für Schüler, Studenten, und nicht zu vergessen für Hobbyprogrammierer, die Chance, in die Programmerstellung mit C# einzusteigen. Zum Editieren und Kompilieren eignet sich u. a. die kostenlose Software „Microsoft Visual C# 2008 Express Edition“.

Die Testsoftware

Für den ersten Schritt, dem Laden von Bildern auf das Display, haben wir eine einfache Windows-Testsoftware erstellt, die ebenfalls zum Download bereitsteht.

Die Oberfläche der Testsoftware gestaltet sich sehr übersichtlich und zeichnet sich durch eine intuitive Bedienung aus. In Abbildung 6 ist die Oberfläche in Aktion zu sehen. Im Bereich „Status“ wird die Softwareversion der angeschlossenen Schaltung ausgegeben.

Im Bereich „Verbindung“ befindet sich die Auswahl des verwendeten virtuellen COM-Ports. Darunter erscheint der Verbindungsstatus zum LCP 100.

Der Bereich „Bild“ gliedert sich wie folgt:

Buttonbeschriftung	Erläuterung
Bild laden	das auf dem LCP 100 auf dem gewählten Speicherplatz gespeicherte Bild wird auf dem Display ausgegeben
Bild speichern	das in das Programm geladene Bild wird auf den LCP 100 auf den gewählten Speicherplatz gespeichert *
Testbild anzeigen	Ein Testbild wird auf dem LCP100 ausgegeben

*Nach dem Laden eines Bildes wird das Bild auf das Format 128 x 128 Pixel angepasst und in den 16-Bit-Farbraum konvertiert. Das Ergebnis dessen erscheint dann im Vorschaufenster.

Im Bereich „Displaybeleuchtung“ lässt sich die Displaybeleuchtung des LCP 100 wahlweise ein- bzw. ausschalten.

Um die Arbeitsweise der Bildspeicherung und -darstellung zu verdeutlichen, wollen wir zum Abschluss die Konvertierung, Übertragung und Darstellung einer Bilddatei einmal beispielhaft „durchspielen“:

Über den Ladedialog des Programms erfolgt das Laden des gewünschten Bildes, das im Format BMP/JPG/GIF/PNG vorliegen muss. Anschließend verkleinert die Software das Bild auf die Größe des Displays und konvertiert die Farbwerte entsprechend dem Display-Farbwertumfang (RGB 5:6:5).

Nun wird das Bild in Pages zu je 512 Byte aufgeteilt. Diese werden dann mit der Page-Nummer (0–4096) an das LCP 100 gesendet. Dabei entsprechen 64 Pages einem Vollbild

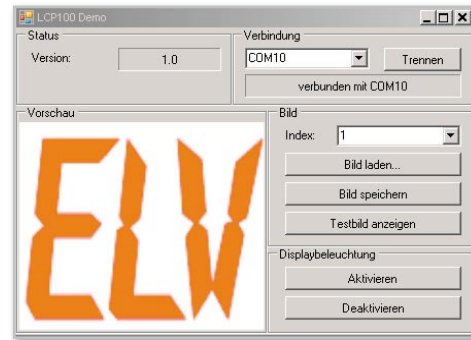


Bild 6: Das LCP-100-Demo-Programm für das Konvertieren, Übertragen und Anzeigen von Bildern

in einer Farbtiefe von 65.536 Farben. Nach der Übertragung wird nun das Bild Page für Page angezeigt. Allerdings kann die Darstellung jeder Page wie im Demoprogramm (Abbildung 7) auch direkt nach der Speicherung erfolgen.

Bleibt uns abschließend, viel Spaß beim Einsatz und der Programmierung des vielseitig einsetzbaren Mini-Displays zu wünschen!

ELV

Internet:

Microsoft Visual C# 2008 Express Edition:
www.microsoft.com/germany/express/product/visualcsharpexpress.aspx

Herstellerseite des Schnittstellenwandlers CP2102:
www.silabs.com
 (VCP-Treiber ebenfalls im Downloadarchiv enthalten)

Datenblatt des Sitronix ST7637:
[www.sitronix.com.tw/sitronix/SASpecDoc.nsf/File-Download/ST76372773498/\\$FILE/ST7637_v1.5.pdf](http://www.sitronix.com.tw/sitronix/SASpecDoc.nsf/File-Download/ST76372773498/$FILE/ST7637_v1.5.pdf)
 (Ebenfalls im Downloadarchiv enthalten)

Datenblatt des Displayherstellers:
www.wandisplay.com/UploadFiles/2007122716362267.pdf
 (Ebenfalls im Downloadarchiv enthalten)

Produktseite des LCP100:
www.elv.de

```

/// <summary>
/// Speichert ein Bild
/// </summary>
/// <param name="index">index des Bildes</param>
/// <param name="image">Bildpuffer (32 KByte)</param>
private void SavePicture(byte index, byte[] image)
{
    try
    {
        int offset = index * PAGES_PER_IMAGE;
        for (int i = 0; i < PAGES_PER_IMAGE; i++)
        {
            byte[] page = new byte[PAGE_SIZE];
            for (int j = 0; j < PAGE_SIZE; j++) { page[j] = image[j] + (i * PAGE_SIZE); }
            if (!lcp100.SavePage(offset + i, page)) { throw new Exception(); }
            lcp100.ShowPage(offset + i, PAGE_SIZE);
        }
    }
    catch
    {
        MessageBox.Show("Das Bild konnte nicht gespeichert werden.", "Bild speichern");
    }
}

```

Bild 7: Die Darstellung jeder Page kann direkt nach dem Speichern erfolgen.