

Segment-Zauber - LCD-Simulator für das eigene LCD-Design

Sobald ein Entwickler ein eigenes LCD-Design für ein neues Projekt benötigt, wird es kompliziert - kundenspezifische LCDs stehen am Beginn einer Entwicklung kaum zur Verfügung. Der LCD-Simulator ermöglicht die komplette Simulation eines beliebigen LC-Displays mit bis zu 80 Segmenten und 16 Backplanes am PC-Bildschirm und stellt damit ein äußerst effizientes Entwicklungswerkzeug dar.

Teurer Weg zum neuen Display

Displays aller Art bilden das Frontend und die Benutzerschnittstelle eines Gerätes, ob dies eine einfache Digital-Uhrenanzeige ist oder ein komplex aufgebautes, mit eigener Intelligenz arbeitendes Punktmatrixdisplay, wie wir es als Bildschirm tragbarer PCs kennen.

Der Entwickler eines solchen Gerätes steht stets vor der Aufgabe, die Datenausgabe so effizient, sprich kostengünstig, wie möglich zu gestalten. Solange er dabei auf in Massenproduktion hergestellte Standard-Displays, etwa für einen Wecker, zurückgreifen kann, wirft die Aufgabe kein Pro-

blem auf. Sobald aber die Anzeige komplexer bzw. spezifischer werden soll, ist eine andere Vorgehensweise nötig. Für einfache Text- und Grafikanzeigen greift man dann zum gängigen Punktmatrix-Display, das durch den integrierten Controller relativ einfach programmierbar ist, jedoch nur über eine endliche Anzahl von programmierbaren Pixeln verfügt.

Will man beispielsweise auf relativ kleinem Raum angenehm anzusehende Grafiken darstellen, stoßen zumindest kleinere Punktmatrix-Displays schnell an ihre Grenzen.

So ist, wie anhand von Abbildung 1 leicht nachzuvollziehen ist, eine 45-Grad-Linie im Punktmatrix-Display eben eine

„Treppe“, im spezifischen LC-Display hingegen eine glatte Linie.

Außerdem steigt mit der Größe auch der Aufwand der Ansteuerung eines Punktmatrix-Displays. Spezielle Ansteuerprozessoren und hochwertige, weil schnelle Bild-

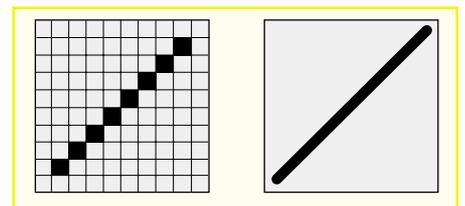


Bild 1: Der optische Unterschied zwischen Punktmatrix- und „normalem“ LC-Display.

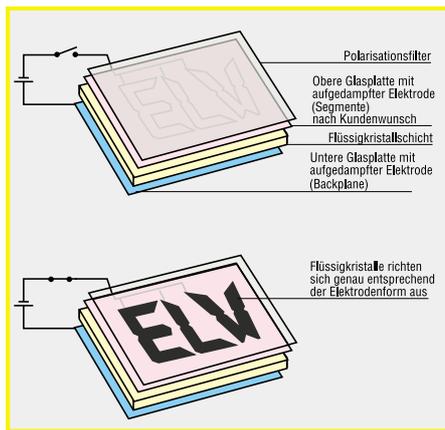


Bild 2: Der schematische Aufbau eines LC-Displays zeigt die Lage von Backplane- und Segment-Elektrode. Durch die Form der beliebig gestaltbaren Segmentelektrode wird die Abbildung bestimmt.

speicher treiben die Kosten in die Höhe. Diese hohen Kosten ziehen sich dann auch durch die gesamte Serienproduktion, da hier auch bei Massenproduktion nur geringe Rabatte zu erzielen sind.

All dies ist für den Entwickler ein Grund, ein kundenspezifisches LC-Display zu entwerfen, das dann, hat man erst einmal den Entwicklungsaufwand und die Testphase hinter sich, äußerst kostengünstig in großen Mengen produzierbar ist. Die Produktionskosten betragen bei großen Serien dann nur noch Pfennige bis maximal wenige Mark je Display.

Erinnert man sich an den Aufbau und die Funktion von Flüssigkristall-Displays, so kann man jeden beliebigen Punkt auf dem Display direkt durch Aufdampfen einer Elektrode in Form der später gewünschten, sichtbaren Kontur auf der Glasscheibe des Displays erzeugen (Abbildung 2). So sind geometrische Formen beliebiger Art erzeugbar, die völlig homogen darstellbar sind. Ein Kreis ist hier also wirklich ein Kreis!

Dazu kommt die relativ einfache Ansteuerung durch kostengünstig zu produzierende Standard-Prozessoren, es sind keine teuren Bildspeicher, Matrix-Ansteuerungen etc. nötig. Man muß ja nur die relativ wenigen Segmente ansteuern können, die man tatsächlich benötigt.

Das Problem ist die Prototypen-Entwicklung. Diese ist logischerweise sehr teuer und bewegt sich auch beim heute erreichten, hohen Grad der Erstellungstechnologie von LCDs im Bereich von mehreren tausend Mark. Erfahrungsgemäß kommt

es in der Entwicklungsphase zu zahlreichen Änderungen des Layouts, sei es durch Kundenwünsche, die Wünsche des Designers oder funktionale Änderungen. Dann kommen schnell Display-Entwicklungskosten von mehreren zehntausend Mark zusammen, die das endgültige Produkt unnötig verteuern.

Kosten sparen durch Simulation

Deshalb liegt es nahe, die Display-Musterherstellung durch eine geeignete Simulationsmöglichkeit so lange hinauszuzögern, bis man die Entwicklung des ansteuernden Gerätes vollständig abgeschlossen hat und das endgültige Design des Displays vorliegt.

Genau für diese Aufgabe wurde der ELV LCD-Simulator entwickelt. Er vermag Displays zu simulieren, die über bis zu maximal 80 Segmente und 16 Backplanes verfügen können, eine auch für große LC-Displays ausreichende Anzahl. Die Simulation erfolgt auf einem normalen PC-Bildschirm anhand der ohnehin erforderlichen Entwurfszeichnung für das Display.

Der Clou am Windows-Simulationsprogramm des Simulators ist der, daß man jede beliebige (einfarbige) Zeichnung heranziehen kann, die im Bitmap-Format vorliegt. Dies kann ein Scan ebenso sein wie eine in das Bitmap-Format umgewandelte Vektorzeichnung, z. B. aus Corel Draw, Micrografx Designer, Macromedia Freehand oder Adobe Illustrator. Natürlich kommen hier auch in reinen Bitmap-Zeichenprogrammen erstellte Entwürfe, z. B. aus MS-Paint, zum Einsatz. Die einzigen Grundvoraussetzungen für diese Dateien sind das Dateiformat „Bitmap“ und die Speicherung als einfarbiges Bild.

Simulator-Hardware mit Intelligenz

Der an eine beliebige serielle Schnittstelle des PCs anschließbare Simulator (Abbildung 3 zeigt die Frontansicht) bildet das Bindeglied zwischen dem Simulatorprogramm und der entwickelten Schaltung. Er verarbeitet die vom ansteuernden Prozessor bzw. den Anzeigentreibern kommenden Signale und speichert diese zwischen, bis sie vom PC über die serielle Schnittstelle abgefragt werden. Ein interner Prozessor steuert den Ablauf.

Die Verbindung zur ansteuernden Schaltung erfolgt über zwei 50polige Stiftleisten

Tabelle 1: Belegung der Stiftleisten

Linke Stiftleiste

Pin 1 bis Pin 40: SEG-Leitung 0 bis 39
Pin 41 bis Pin 48: COM-Leitung 0 bis 7
Pin 49: BIAS High
Pin 50: BIAS Low

Rechte Stiftleiste

Pin 1 bis Pin 40: SEG-Leitung 40 bis 79
Pin 41 bis Pin 48: COM-Leitung 8 bis 15
Pin 49: BIAS High
Pin 50: GND

(Belegung siehe Tabelle 1), die sowohl bis zu 80 Segment-, als auch bis zu 16 Backplane-Signale (folgend als COM bezeichnet) aufnehmen. Zusätzlich sind hier die sogenannten BIAS-Anschlüsse verfügbar, die den Spannungsbereich vorgeben, in dem sich die Signale der Segment- und COM-Leitungen bewegen (0 bis max. 9 V). Mit dem DIP-Schalter 4 auf dem rechts oben auf der Frontplatte liegenden 4fach-DIP-Schalterblock ist der BIAS-Low-Wert einheitlich auf GND festlegbar.

In der Mitte oberhalb der Stiftleisten befindet sich ein Einstell-Poti für die BIAS- bzw. Referenzspannung. Hier kann die Komparator-Schaltswelle eingestellt werden. Um also ankommende Signale detektieren zu können, müssen die Komparatoren der Eingangsbeschaltung des Simulators Spannungen innerhalb des BIAS-Bereiches als Vorgabe erhalten. Mit dem BIAS-Poti ist die Komparator-Referenzspannung für die Backplane-Eingänge (COM 0 - 15) im Bereich BIAS-High bis (BIAS-High + BIAS-Low)/2 und für die Segment-Eingänge im Bereich BIAS-Low bis (BIAS-High + BIAS-Low)/2 einstellbar.

Mittels der 3 weiteren DIP-Schalter des 4er-DIP-Schalter-Blocks kann man die Übertragungsrate der seriellen Datenkommunikation zum Rechner auswählen (115.200, 57.600, 38.400 Baud). Dabei darf natürlich stets nur ein DIP-Schalter aktiv, sprich auf „ON“ geschaltet sein.

Schließlich befindet sich oberhalb der beiden Stiftleisten noch je ein 8fach-DIP-Schalter-Block. Hier ist der Zustand aller 16 COM-Leitungen einstellbar. Das heißt, wird die jeweilige COM-Leitung durch die ansteuernde Schaltung als Backplane-Leitung genutzt, so ist der zugehörige DIP-Schalter auf „Active“ zu stellen, wenn nicht, dann zwingend auf „Inactive“.

Der 9polige serielle Anschluß auf der Geräterückseite ist über ein 9poliges Kabel mit einer freien seriellen Schnittstelle des PCs zu verbinden.

Schaltung

Anhand der vorangegangenen Funktionsbeschreibung ist die Schaltung des Simulators schnell erkennbar.

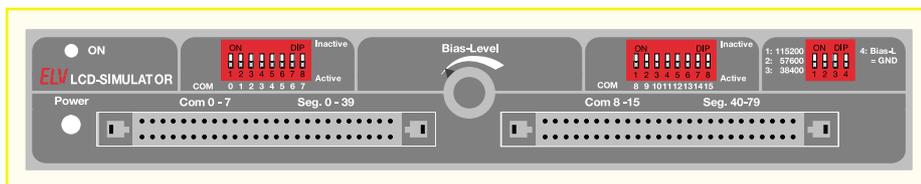


Bild 3: Die Frontansicht des LCD-Simulators

991744A04

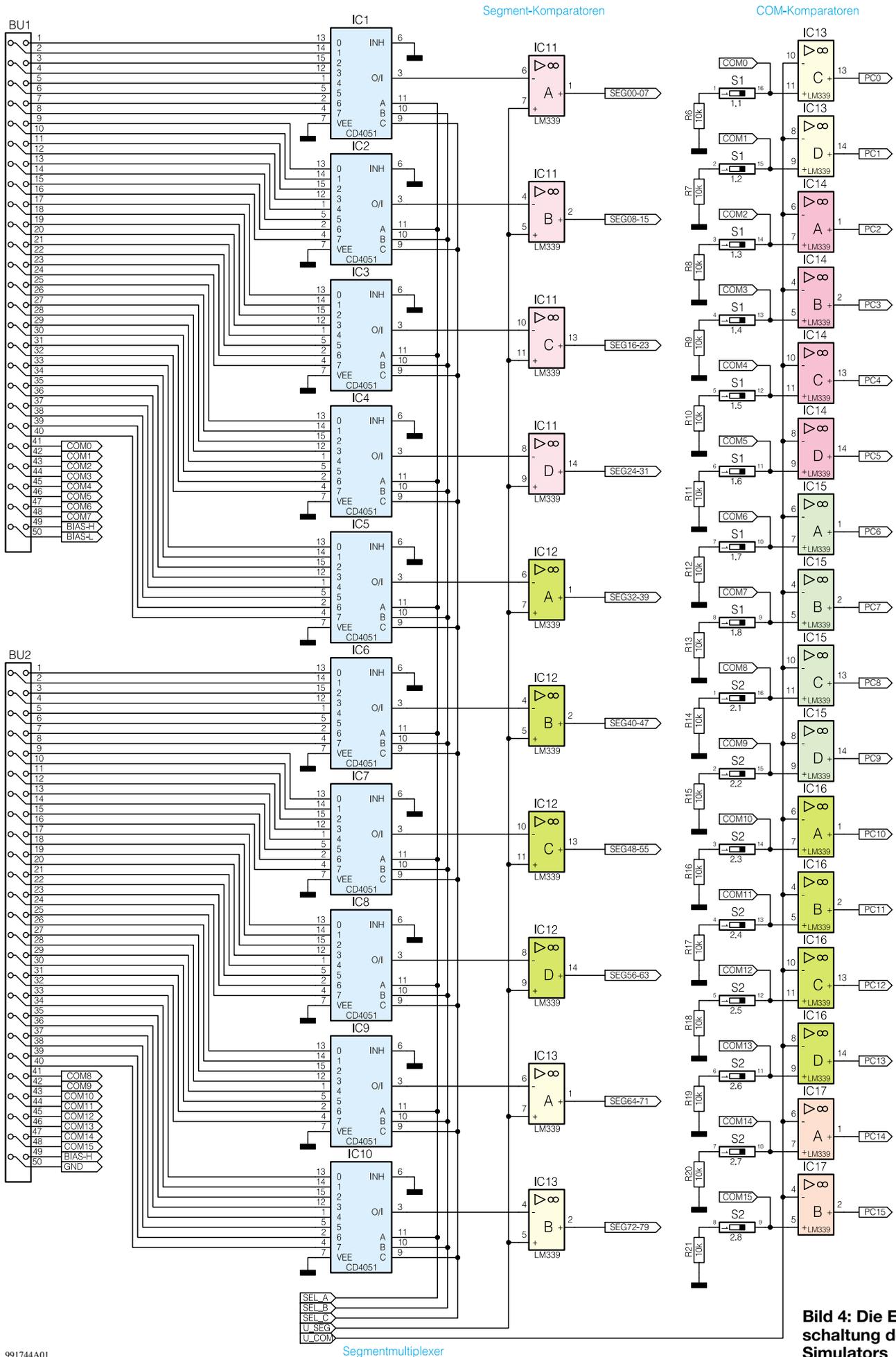


Bild 4: Die Eingangsschaltung des LCD-Simulators

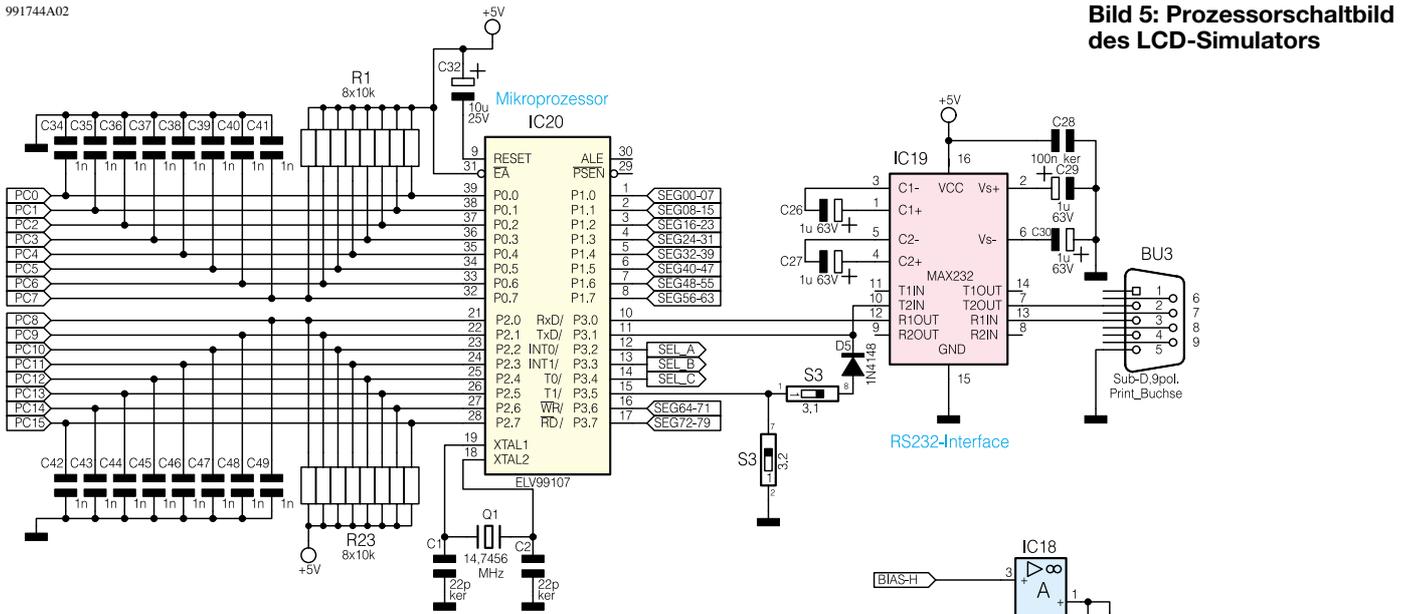


Bild 5: Prozessorschaltbild des LCD-Simulators

Kern des Simulators ist der Prozessor IC 20 (Abbildung 5), der sowohl die Multiplexerabläufe der Eingangsmultiplexer IC 1 bis IC 10 steuert (Abbildung 4) als auch die Abfrage der Komparatoren für die Backplane-Eingänge und die Umwandlung aller Daten in ein serielles Ausgangssignal vornimmt. Seine Peripherie besteht lediglich aus der Takterzeugung mit Q 1, C 1 und C 2, der Einschalt-Reset-Beschaltung C 32 und der Beschaltung für die Backplane-Signaleingänge.

Der Prozessor gibt das serielle Signal auf das RS232-Interface IC 19, der das Signal auf RS232-Pegel umsetzt.

Die Eingangsschaltung des Simulators besteht aus 10 Multiplexern (IC 1 bis IC 10), die, gesteuert durch den Prozessor, die hohe Anzahl der 80 Segmenteingänge in 10 Multiplex-Segment-Signale umsetzen. Diese gelangen über die Segment-Komparatoren IC 11 A bis IC 13 B an die zugehörigen Porteingänge P 1.0 bis P 1.7 sowie P 3.6 und P 3.7.

Die Backplane-Multiplexer IC 13 C bis IC 17 B sind, wie erwähnt, einzeln über die DIP-Schalter S1, S2 sperrbar.

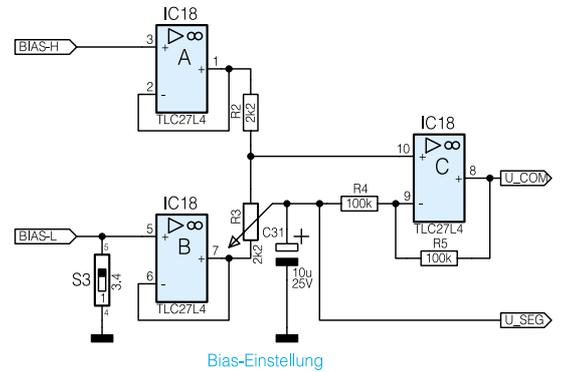
Beide Komparatorgruppen erhalten die Referenzspannung gegenphasig aus der

OP-Schaltung mit IC 18. An den nicht invertierenden Eingängen von IC 18 A und B liegen die beschriebenen BIAS-High- und BIAS-Low-Pegel. S 3.4 kann den BIAS-Low-Pegel auf GND festlegen.

IC 18 C sorgt für die invertierte Ausgabe des BIAS-Wertes an die Backplane-Komparatoren und die Erzeugung eines Offsets, um durch die Invertierung keine negative Referenzspannung zu erhalten. R 3 ermöglicht die Einstellung der Referenzspannung, wie zuvor beschrieben (Abbildung 6).

Die Stromversorgung wird durch das integrierte Netzteil sichergestellt. IC 21 erzeugt die stabilisierte 5V-Betriebsspannung für den Prozessor und den RS232-Interfacebaustein, während IC 22 eine ebenfalls stabilisierte Spannung von 10 V für alle restlichen Schaltungsteile bereitstellt.

Bild 6: Schaltbild der BIAS-Einstellung



Der Funktionsablauf im Simulator

Der Prozessor überwacht ständig alle 16 COM-Leitungen direkt. Sobald der Pegel auf einer dieser Leitungen die eingestellte Referenzspannung überschreitet (dies wird dem Prozessor über die Komparatoren IC 13 C bis IC 17 B mitgeteilt), tastet der Prozessor über den Eingangsmultiplexer IC 1 bis IC 10 alle 80 Segmentleitungen ab und ermittelt über die Komparatoren IC 11 A bis IC 13 B, welche davon unter der eingestellten Referenzspannung liegen. So wird zu jeder aktiven COM-Leitung die entsprechende, gerade aktive Segment-Leitung erkannt und im Prozessor gespeichert. Die Software auf dem angeschlossenen PC fragt diesen Datensatz zyklisch ab (Zyklus in der Windows-Software einstellbar, z. B. 20 ms) und aktiviert bzw. deaktiviert die entsprechenden Segmente in der simulierten Displayzeichnung.

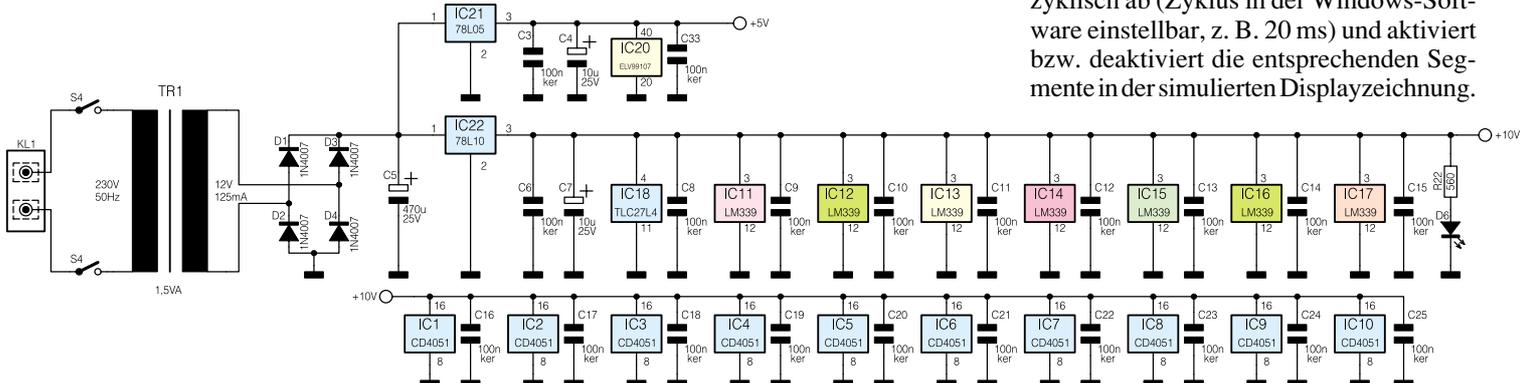


Bild 7: Schaltung des Netzteils

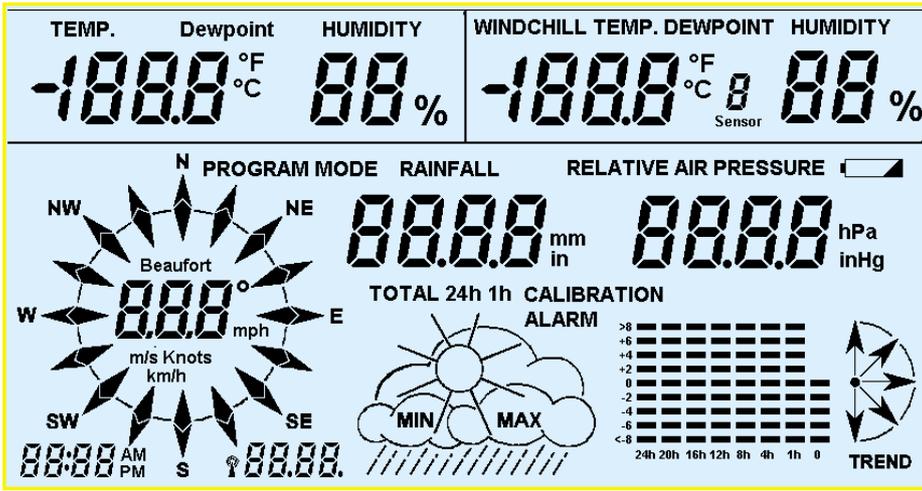


Bild 8: Ausgangspunkt für die LCD-Simulation ist eine Zeichnung, die im Bitmap-Format abzuspeichern ist. Dabei kann es durchaus filigran zugehen, lediglich ein Berühren benachbarter Elemente ist zu vermeiden.

Zusätzlich überwacht der Prozessor auch, welche COM-Signale in einem bestimmten Zeitraum aktiv waren und teilt dies der Windows-Software mit. Diese kann die Aktivitäten anzeigen. Diese Anzeige ist zur korrekten Einstellung des BIAS-Wertes notwendig, um eine sichere Erkennung der Backplane-Signale auf den COM-Leitungen zu gewährleisten.

Bedienung/Software

Die Installation der auf einer Diskette gelieferten Software erfolgt über den Aufruf des Setup-Programms.

Die Software legt standardmäßig eine eigene Programmgruppe im Windows-Startmenü an. Ein Rechnerneustart ist nach der Installation nicht erforderlich.

Displaybild erstellen

Bevor das Programm eingesetzt wird, ist das Displaylayout, wie bereits beschrie-

ben, als monochrome Bitmap, z. B. mittels „MS Paint“, zu erstellen und in einem anzulegenden Projektverzeichnis abzuspeichern.

Beim Entwurf des Displaybildes ist zu beachten, daß sich keine Segmente untereinander berühren dürfen, da zur Simulation eine Füllroutine eingesetzt wird (Abbildung 7). Die Folge wäre das gleichzeitige Füllen auch der benachbarten Segmente.

Nun ist das Bedienprogramm „LCD Simulator“ zu starten.

Projekt erstellen und konfigurieren

Über das Dateimenü ist ein neues Projekt zu erstellen über den „Speichern unter“-Dialog im Dateimenü kann der Name des Projekts festgelegt werden.

Per Menü „Bearbeiten - Konfiguration“ erfolgt nun das Festlegen der Vorgaben für das zu simulierende Display (Abbildung 8). Hier sind die Anzahl der Segmente, der Backplanes (COM) und die Abfragerate

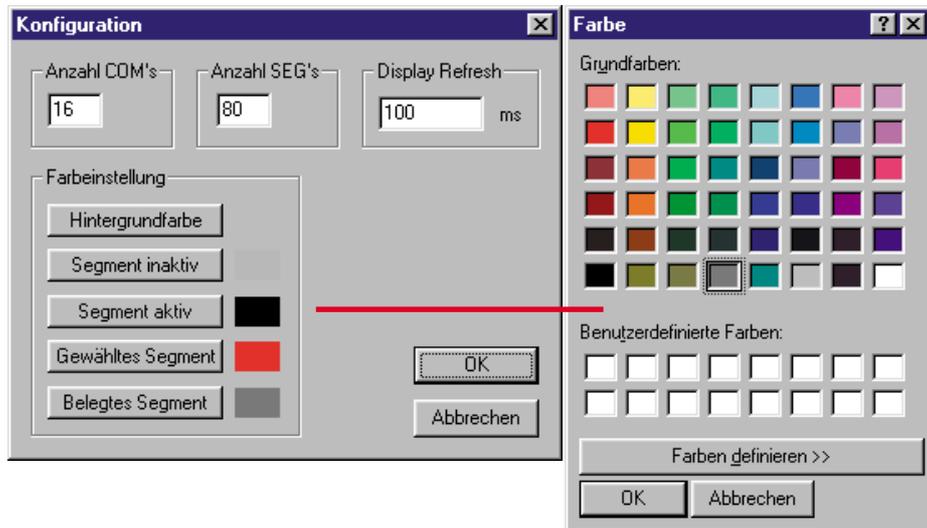


Bild 9: Konfigurationsmenü des Simulatorprogramms. Hier sind die Anzahl der Backplanes und der Segmente, die Refresh-Rate der Anzeige und die Segmentfarben einstellbar.

des Simulators ebenso festlegbar wie die Darstellungsfarben für die einzelnen Displayzustände.

Bei der Auswahl der Farben ist unbedingt zu beachten, daß man für alle Segmentzustände unbedingt zum gewählten Hintergrund unterschiedliche Farben wählen muß, da man sonst das betreffende Segment nicht mehr auf dem Hintergrund wiederfindet. Es ist dann auch nicht aktivierbar, da die Simulation, wie gesagt, mit Füllroutinen für Flächen arbeitet. Im Extremfall wird die gesamte Bildfläche (Hintergrund) gefüllt.

Als nächstes ist das Bitmap-Bild des Display-Layouts über „Bearbeiten - Bitmap festlegen“ zu laden (Abbildung 9).

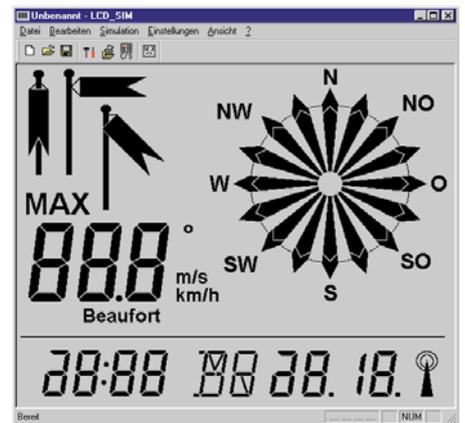


Bild 10: Die Menüansicht mit geladenem Bitmap-Bild

Segmente zuordnen

Ist dies erfolgt, kann man nun die Segmente des Displays zu den zugehörigen COM-Segment-Kombinationen zuordnen.

Nach Anwahl des Menüs „Bearbeiten“ wählt man dazu dort den Menüpunkt „Segmente zuordnen“ an. Daraufhin erscheinen in der Kopfleiste Auswahlfenster für die COM- und Datenleitungen. Betätigt man per Maus die Plus-Zeichen hinter den Fenstern, erfolgt ein automatisches Weitschalten der Leitungsbezeichnungen bei jeder neuen Segmentanwahl in der Zeichnung. Deaktiviert man diese Option bzw. benutzt man für die Zuordnung die rechte Maustaste, so kann man z. B. einer Segmentleitung mehrere Segmente in der Zeichnung zuordnen, z. B. für das Aufrufen einer bestimmten Segmentkombination, eines Wortes etwa. Bis zu 32 Segmente sind auf diese Weise für eine COM-/Segmentkombination definierbar.

Die Zuordnung erfolgt zunächst ganz einfach durch Anwahl eines Segments per Maus. Dieses färbt sich, falls die o. g. Plus-Option nicht gewählt ist, in der voreingestellten Standard-Konfiguration rot. Will man, wie beschrieben, einer COM-/Segment-Kombination mehrere Segmente zu-

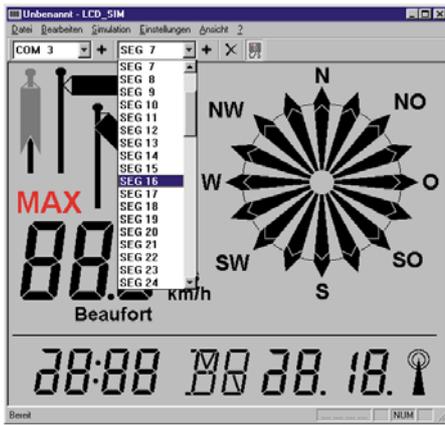


Bild 11: So erfolgt die Zuordnung der Segmente zu den Steuerleitungen.

Simulator auch über einen ELV-RS232-Multiplexer angeschlossen werden, die Software berücksichtigt diese Anschlußmöglichkeit bereits.

In der Windows-Software wählt man nun über das Menü „Einstellungen“ die vom Simulator bzw. RS232-Multiplexer belegte serielle Schnittstelle ebenso an wie die gewünschte Übertragungsrate (Abbildung 11). Letztere sollte zunächst niedrig gewählt werden, also 38.400 Baud. Erst, wenn die gesamte Simulatoranordnung komplett und störungsfrei funktioniert, sollte man eine höhere Baudrate anwählen. Um die Baudrate am Simulator zu ändern, ist dieser auszuschalten, die DIP-Schalter S3.1 - S3.3 entsprechend zu setzen und anschließend der Simulator wieder einzuschalten.

Ist ein RS232-Multiplexer angeschlossen, so kann man über „RS232-Multiplexer“ die Adressierung des Multiplexer-Ports vornehmen, an den der Simulator angeschlossen ist.

Simulatortest/Simulatorlauf

Nun ist zunächst nach Einschalten des Simulators und der zu testenden Schaltung die benötigte Referenzspannung einzustellen. Dazu dreht man den BIAS-Regler am Simulator auf Linksanschlag und wählt im Menü „Simulation“ die Option „Aktive COM testen“ an.

Es erscheint ein Menü, in dem alle als aktiv erkannten COM-Leitungen mit

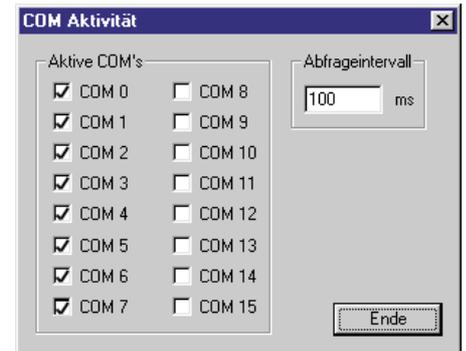


Bild 13: Als aktiv erkannte COM-Leitungen werden hier markiert.

einem Häkchen versehen sind (Abbildung 12).

Durch langsames Rechtsdrehen des BIAS-Reglers wird die Referenzspannung nun so eingestellt, daß das Programm alle erwarteten COM-Leitungen als aktiv erkennt. Während eines Simulatorlaufs kann es u. U. notwendig werden, die Referenzspannung etwas nachzustellen.

Nun kann nach Anwahl der Option „Simulation ein“ aus dem Menü „Simulation“ ein Simulatorlauf gestartet werden (Abbildung 13). Auf der Arbeitsfläche erscheinen alle gerade aktivierten Segmente in der Grundkonfiguration schwarz, alle inaktiven sind nicht bzw. in der gewählten Farbe für inaktive Segmente zu sehen.

Um bei schnellen Änderungen des Displayinhalts diese auch stets verfolgen zu können, erfolgt nicht ständig ein neuer Gesamtaufbau, sondern lediglich eine Veränderung der betroffenen Segmente.

Im zweiten Teil dieses Artikels beschreiben wir den Nachbau des LCD-Simulators. **ELV**

ordnen, so kann man dies durch weitere Segmentanwahl jetzt tun (Abbildung 10).

Nach der Segmentanwahl ist die zugehörige COM-Segment-Kombination zu bestimmen, das Segment färbt sich nun in der Standardeinstellung grau und kann nicht wiederholt zugeordnet werden. Auch eine bereits vergebene COM-/Segment-Kombination ist nicht versehentlich erneut vergebbar - bei einem Versuch bleibt das Segment rot gefärbt.

Durch den Lösch-Button (X) in der Statusleiste erreicht man ein Lösch-Menü, das die nachträgliche Löschung einzelner oder aller Zuordnungen des Projekts ermöglicht. So kann man die jeweils komplette Zuordnung einer COM-/Segmentleitung, einer einzelnen COM- bzw. Segment-Kombination oder aber, wie gesagt, alle Zuordnungen des Projekts löschen.

Nach der Zuordnung aller Displaysegmente und dem Abspeichern des fertigen Projekts erfolgt jetzt die Verbindungsaufnahme mit dem Simulator.

Anschluß des Simulators

Über die mitgelieferten Anschlußadapter ist die zu testende Schaltung an die Stiftleiste(n) des Simulators anzuschließen. Nichtbelegte COM-Leitungen sind mit den zugehörigen DIP-Schaltern zu deaktivieren und die gewünschte Übertragungsrate einzustellen.

Beträgt der BIAS-L-Pegel 0 V, ist der zugehörige DIP-Schalter (4fach-Block, Schalter 4) auf „ON“ zu schalten.

Nun erfolgt die Verbindung der seriellen Schnittstelle mit einer freien seriellen Schnittstelle des PCs. Alternativ kann der

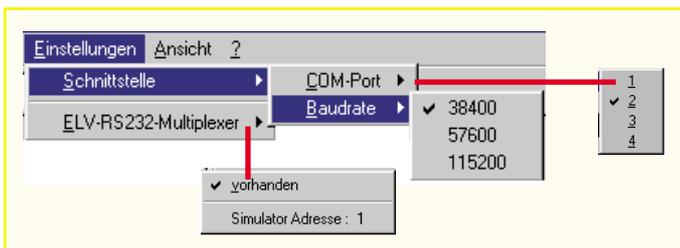


Bild 12: Einfache Einstellung der PC-Schnittstelle.

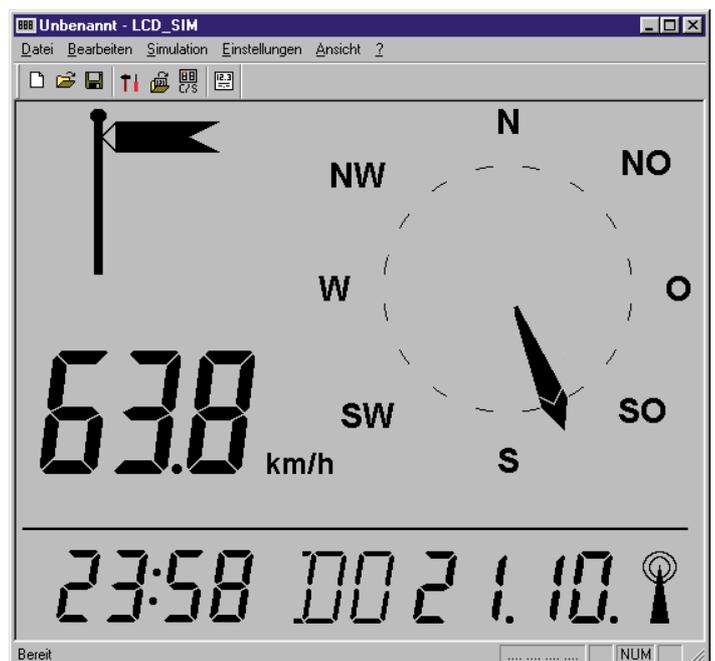


Bild 14: Im Simulatorlauf verhält sich das simulierte Display genau wie das spätere reale Display.