

Studienarbeit

VFD-Studio

Betrieb und Ansteuerung eines *Noritake GU-256x64-372* Vakuum Fluoreszenz Display (VFD) am LPT-Port zur Anzeige von PC-Systeminformationen unter MS Windows®

von Philipp Bank
FH-Offenburg
2004, 2005

```
--- SYSTEMINFORMATIONEN ---  
Computer: KABAL / User: Administrator  
RAM: 1023 MB  
CPUName: Intel(R) Pentium(R) 4 CPU 2.80GHz  
CPUID: x86 Family 15 Model 2 Stepping 9  
CPUVen: GenuineIntel  
CPUSpeed: 2798 MHz  
Zeitzone: Westeuropaeische Normalzeit
```



Laufwerke: _____

SYSTEM (C:)
Festplatte
28,94 GB von 37,26 GB frei = 77,67%

Winamp: Version: 5.03
Titel:
Frank Sinatra - My Way
0:57 / 4:37

Ein paar Screenshots

Inhalt:

- 1. Die Studienarbeit*
- 2. VFD-Technologie*
- 3. Vor- und Nachteile der VFDs gegenüber LC-Displays*

Entwicklung des VFD-Studios :

- 4. Lastenheft*
- 5. Softwareentwurf*
- 6. Ein erstes Use-Case*

1. Die Studienarbeit

Thema dieser Studienarbeit ist die Ansteuerung eines VF-Displays am Druckeranschluss (LPT-Port) eines PCs und die Darstellung verschiedener Systeminformationen auf dem Display.

Es gibt eine Vielzahl kommerzieller Produkte und Freeware-Tools zur Ansteuerung von LC-Displays am Druckerport oder über die serielle Schnittstelle. Grund für diese Vielzahl ist der Modding-Trend.

Modding ist als das Umgestalten oder Tunen von Computern beschreibbar und bereits seit längerem verbreitet.

Seit etwa zwei Jahren bieten auch kommerzielle Anbieter von PC-Zubehör Modding-Artikel an, so dass ein regelrechter Markt entstanden ist.

Modding reicht vom Lackieren des Gehäuses über Innenraumbeleuchtung bis zu wassergekühlten Prozessoren. Mögen einige der „Modifications“ eher der optischen Aufwertung dienen, so gibt es auch manche mit sinnvollem Verwendungszweck.

Eines davon ist z.B. der Anschluss eines Displays am LPT-Port zur Anzeige verschiedenster Informationen wie Videomodus, Speichernutzung, Systemressourcen, etc.

EasyLCD, *LCD-Studio*, *Liquid-Mp3*, *LCD-Hype*, *jaLCD* und *LCDInfo* sind einige populäre Programme zur Ansteuerung von Displays.

Diese sind jedoch alle ausschließlich zur Verwendung mit LC-Displays gedacht – Unterstützung für VFDs findet sich selten⁽¹⁾.

Vor einiger Zeit erwarb ich nun ein GU256x64-372 VFD von Noritake und begann mangels passender Software mir selbst ein Programm zur Ansteuerung zu programmieren.

Die ersten Kommunikationsversuche mit dem Display fanden noch unter DOS mit TurboPascal mit direkten Adresszugriffen statt, später entstand ein Windows-Programm mit grafischer Oberfläche und Kommunikation über Treiber.

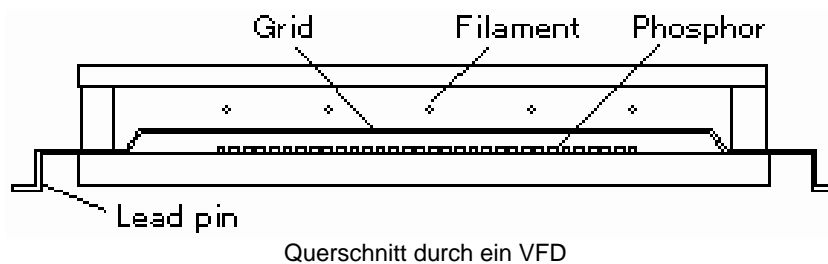
Im Laufe meiner Studienarbeit entstand daraus das *VFD-Studio*, welches in dieser Dokumentation beschrieben wird.

⁽¹⁾ Inzwischen (seit 5-04) kann auch LCDInfo Displays vom Typ GU256x64-372 ansteuern.

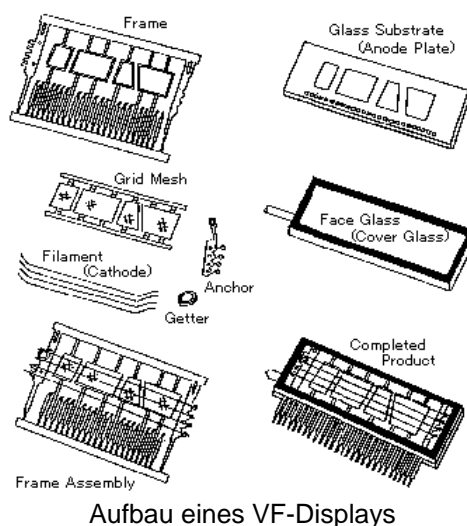
2. VFD-Technologie

Ein VF-Display besteht im wesentlichen aus einem evakuierten Glaskörper. Hinter der Displayscheibe befinden sich die Heizdrähte (Filament), darunter das Anodengitter (Grid) und ganz unten die mit Phosphor beschichteten Anoden.

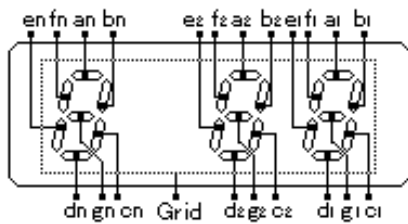
Die vom Heizdraht emittierten Elektronen können durch das Grid beeinflusst werden. Eine negative Gridspannung verhindert, dass die Elektronen zur Anode gelangen können, eine positive Spannung beschleunigt die Elektronen in Richtung Anoden. Einige der so angezogenen Elektronen treffen das Grid und bewirken damit den Gridstrom, viele jedoch gelangen durch das Gitter und treffen auf die Anoden. Die einzelnen Anodenpins können ebenfalls positiv oder negativ geladen sein. Negative Ladung verhindert, dass Elektronen vom Heizdraht auf die Phosphorschicht gelangen können – das Segment/Pixel bleibt dunkel. Ist das Anodenpin jedoch positiv geladen regen die dort einschlagenden Elektronen die Phosphorschicht zum leuchten an. Das Vakuum verhindert, dass das Phosphor zu brennen anfängt.



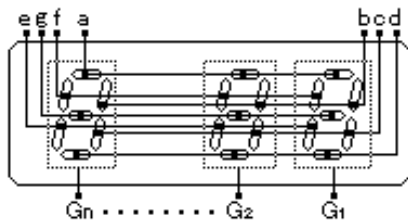
Folgendes Bild veranschaulicht die Schichten eines VFD:



VF-Displays können auf zwei Arten betrieben werden: statisch und gemultiplext.
 Beim statischen Verfahren gibt es nur ein Grid.
 Jedes Anodensegment hat einen eigenen Anschluss.



Dieses Verfahren hat den Vorteil einer sehr einfachen Handhabung, eignet sich jedoch nicht für Displays mit vielen Segmenten.
 (Bei dem verwendeten GU255x64 wären mit dem statischen Verfahren 16384 Anschluss-drähtchen nötig um alle Segmente/Pixel unabhängig voneinander ansteuern zu können!)
 Bei vielen Segmenten wird daher die Ansteuerung gemultiplext betrieben.
 Die Anodensegmente sind untereinander verbunden, jedoch unter separaten Grids.
 Es ist immer nur ein Grid positiv geladen, während die anderen Grids negativ bleiben.



Abhängig davon, welches Grid gerade positiv geladen ist und welche Segmente an dieser Stelle aufleuchten sollen, steuert der Anodentreiber die Anodensegmente (a..g) an.
 Durch die schnelle Folge der positiv geladenen Grids entsteht somit für das Auge so ein stehendes Bild.
 Dieses Verfahren hat allerdings den Nachteil eines hohen Schaltungsaufwandes für das Multiplexverfahren. Die meisten VF-Displays heute nutzen jedoch diese Technik, wenn viele Segmente anzusteuern sind.

3. Vor- und Nachteile der VFDs gegenüber LC-Displays:

VFDs vs. LCDs:

	VFD	LCD
Kontrast	sehr hoch	hoch
Ablesewinkel horizontal	sehr groß	gering - mittel
Ablesewinkel vertikal	sehr groß	gering
Helligkeit	hoch (selbstleuchtend)	muss extern beleuchtet werden
Stromaufnahme	hoch	gering (weunige μA - mA)
Spannungen	hohe Spannungen nötig	nur wenig Spannung (μV – mV)

Einsatzbereiche:

- Technische Geräte: Medizinische Geräte, Messinstrumente
- HiFi: Autoradio, Stereoanlage
- Überwachungsanzeigen
- Automaten

Die möglichen Einsatzgebiete sind wesentlichen identisch mit denen der LCDs.

Im HiFi-Bereich sind VFDs jedoch wegen ihres elegantes Aussehens weit häufiger anzutreffen – ebenso überall wo gute Ablesbarkeit von Bedeutung ist.

Wegen der hohen Leistungsaufnahme sind sie allerdings für mobile Geräte weniger geeignet.

Entwicklung des VFD-Studios

4. Lastenheft

Am Anfang jeder Entwicklung steht naturgemäß die Frage, welche Funktionen realisiert werden sollen, welche Features gewünscht sind.

Es ist sinnvoll sich hierfür ein Lastenheft, und sei es auf einem Notizzettel, anzufertigen um einerseits die Vorstellungen und Anforderungen etwas zu konkretisieren und andererseits sich bei der späteren Entwicklung auf das Wesentliche zu konzentrieren.

Das Lastenheft umfasste folgende Requirements:

Allgemein:

1. Das Programm soll im Hintergrund laufen und in einstellbaren Intervallen verschiedene Informationen, Bilder und Animationen auf dem Display ausgeben.
2. Der Benutzer soll möglichst viel Gestaltungsmöglichkeiten haben, woraus folgt, dass die Informationen nicht in einer vorgegebenen Darstellungsart angezeigt werden, sondern sich der Benutzer seine „Screens“ selbst zusammenstellen kann.
3. Es soll eine (oder mehrere) vom Benutzer erstellte Liste geben, in welcher steht, was auf dem Display angezeigt werden soll.
4. Das Programm soll diese Liste interpretieren und daraus einen (oder mehrere) „Screens“ erstellen, welcher auf dem Display für eine einstellbare Zeit angezeigt wird. Wenn das Ende der Liste erreicht ist, fängt die Interpretation wieder am Anfang der Liste an.
5. Der Benutzer soll die Möglichkeit haben aktiv auf das Programm, bzw. die Listeninterpretation, einzuwirken um z.B. den nächsten Screen vor Ablauf des Intervalls anzuzeigen oder eine andere Liste zu laden.
6. Die Entwicklung soll modular aufgebaut sein. Für jeden größeren Aufgabenbereich, bzw. jedes Teilproblem soll eine wiederverwertbare Komponente existieren.
7. Damit das Programm unter NT-basierten Betriebssystemen (Win2000) lauffähig ist, soll ein LPT-Treiber verwendet werden.
8. Das Projekt soll bis Ende 2004 abgeschlossen sein (reine Programmierung ohne Dokumentation).

Ausgabe auf dem Display:

1. Darstellen von Grafiken aus Bitmap-Dateien auf dem Display
2. Abspielen von Animationen
3. Ausgabe von Texten und Informationen
4. Grafische Darstellung der CPU- und Speichernutzung (ähnlich wie im Windows-Task-Manager)
5. Grafische Darstellung der Uhrzeit
6. Einzelne Pixel setzen
7. Helligkeit einstellen

Informationen über:

- Betriebssystem
- Computername, Username
- Netzwerkinformationen (IP-Nummer)
- CPU-Daten
 - Prozessorname, Hersteller, Technologien, ...
- Speichernutzung
- Angaben über Laufwerke
- Datum und Uhrzeit
- Winamp-Informationen
 - Titel und Interpret, Position und Dauer
- Videomodus
- Installierte Drucker
- ...

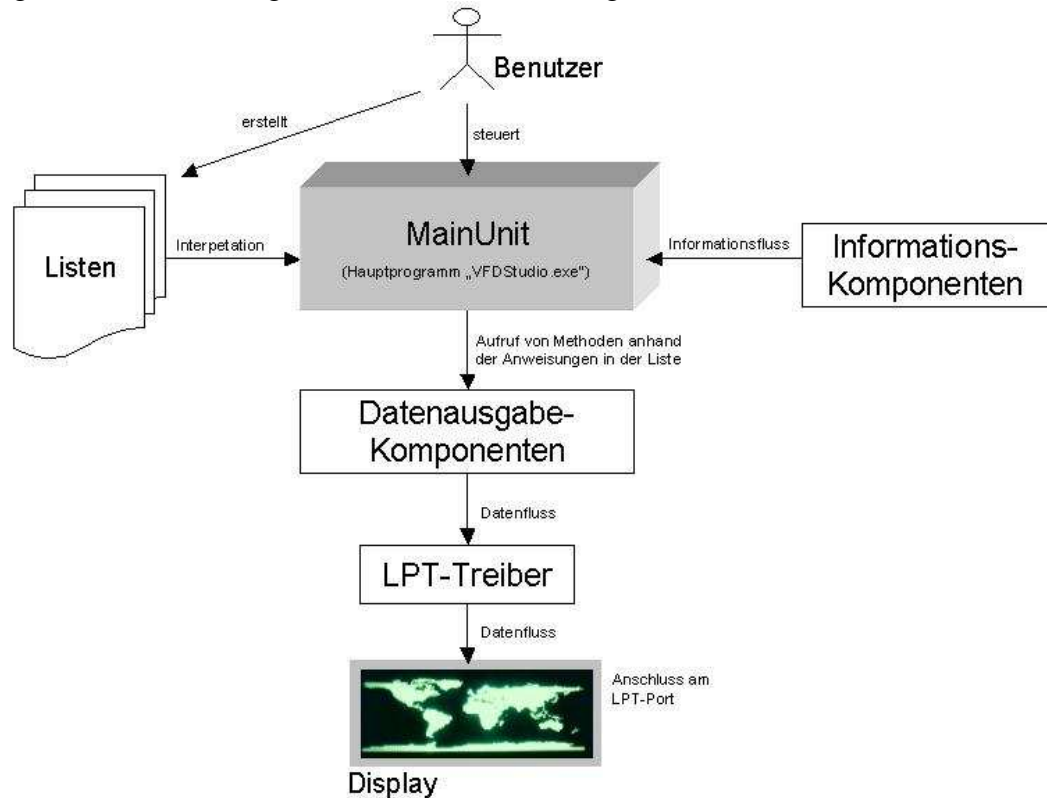
5. Softwareentwurf

Anhand des Lastenheftes und einiger weiterer Überlegungen kann das Software-Projekt nun in groben Zügen umrissen werden.

Die Kernpunkte lassen sich aus dem Lastenheft folgendermaßen zusammenfassen:

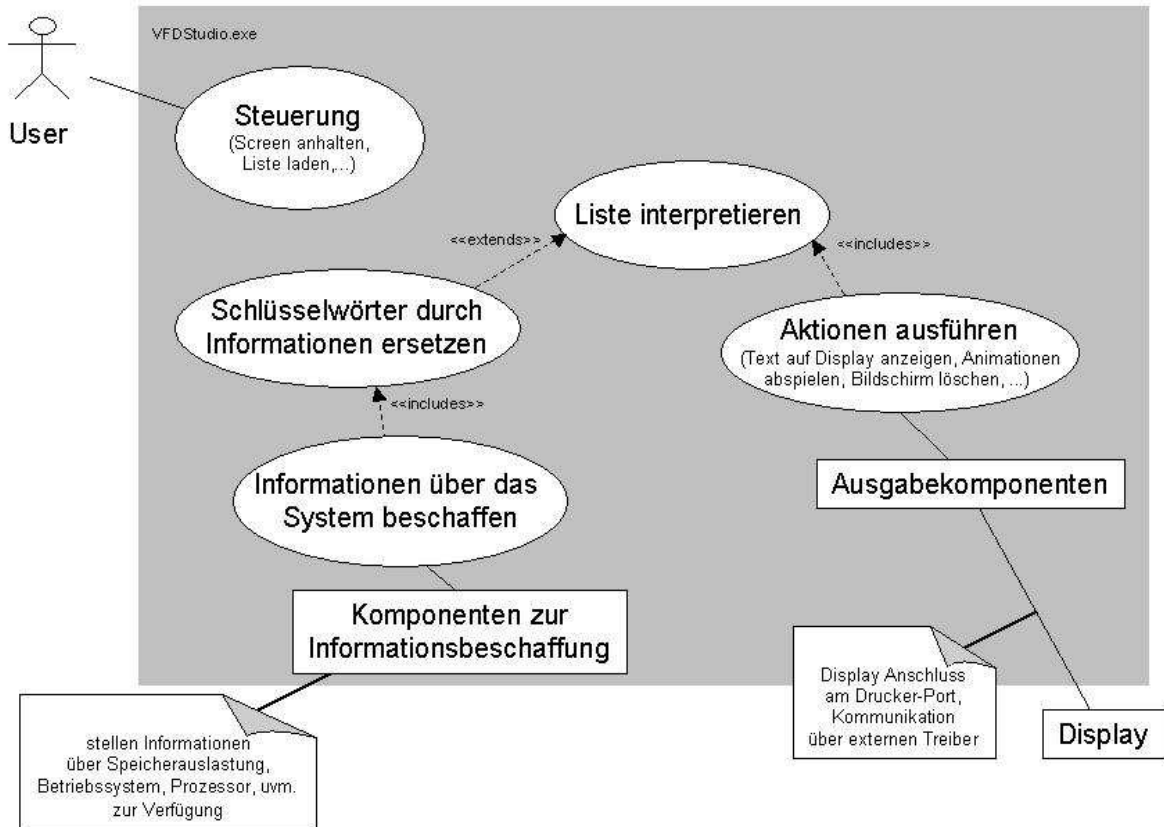
- Benötigt werden Komponenten zur Informationsbeschaffung und zur Datenausgabe auf dem Display.
- Die Interpretation der Listen kann vom Hauptprogramm übernommen werden, welches zugleich als Bindeglied zwischen Listen, Informationen und Datenausgabe zu sehen ist.
- Für die Listen wird zwecks einfacher Interpretation eine simple Scriptsprache benötigt (mit Befehlen wie z.B. „CLEARSCREEN“ um den Bildschirm zu löschen oder ähnlichem).
- Die Erstellung der Listen kann zwar mit einem gewöhnlichen Texteditor erfolgen, benutzerfreundlicher ist jedoch ein separates Programm zu diesem Zweck.
- Die Software muss sich Einstellungen merken, z.B. welche Liste sie zuletzt geladen hat, welche Port-Adresse sie verwenden soll, usw. und daher eine Ini-Datei besitzen.

Daraus ergibt sich in etwa folgende bildliche Darstellung:



6. Ein erstes Use-Case

Hieraus lässt sich nun das (stark vereinfachte) Use-Case Diagramm für das Hauptprogramm erstellen:



Use-Case-Diagramm des Projektes

Obiges Diagramm umreißt in groben Zügen die Funktionalität des Hauptprogrammes.

Der User kann das Programm dahingehend steuern, als dass er den Screen anhalten kann (d.h. es wird nach Ablauf der Screendauer nicht der nächste Screen angezeigt), den nächsten Screen anzeigen lassen kann, eine andere Liste laden, ...

Zentraler Punkt des Programms ist jedoch die Listeninterpretation, bei der (falls Informationen angezeigt werden sollen) Informationen von den entsprechenden Komponenten zur Verfügung gestellt werden und anhand der Anweisungen in der Liste verschiedene Aktionen ausgeführt werden – oder konkreter: Prozeduren und Funktionen der Ausgabekomponenten aufgerufen, die über einen externen LPT-Treiber mit dem Display kommunizieren.