

Das „Crystalfontz CFA-635“ unter Linux

Allgemeines zum Display

Das von der amerikanischen Firma „Crystalfontz“ hergestellte Displaymodul „CFA-635“ besteht aus einem LCD-Display, das je nach Ausführung weiße Schrift auf blauem Hintergrund oder schwarze Schrift auf grünem Hintergrund darstellt, vier Leuchtdioden, deren beiden Farben Grün und Rot jeweils unabhängig gedimmt werden können und einem 6-Tasten-Keypad, das angenehmerweise auch hinterleuchtet ist. Die Helligkeit des Backlights kann per Software eingestellt werden, ebenso wie der Kontrast des LCD-Displays. Leider kann man die Beleuchtung des Displays und des Keypads nicht getrennt regeln, was aber nicht weiter negativ auffällt, da das Verhältnis der beiden zumindest bei dem Modell mit der grünen Beleuchtung gut abgestimmt ist.

Der Anschluss an den PC erfolgt komfortabel über einen USB-Anschluss. Somit entfallen jegliche Lötarbeiten und es ist möglich, mehrere Module gleichzeitig an einem PC zu betreiben. Es gibt Kabel zum Anschluss des Displays mit einem gewöhnlichen USB-Stecker und es gibt Kabel, um das Modul direkt an einem internen USB-Port des Mainboards anzuschließen. Hierbei sind alle Pins einzeln am Mainboard anzuschließen, was den Vorteil bietet, dass das Kabel auf alle Mainboards passt, egal welche Belegung der Hersteller vorgesehen hat.

Zusätzlich zum Modul gibt es einen Einbaurahmen für einen 5¼“-Laufwerksschacht und diverse Frontblenden, etwa Grau, Schwarz oder sogar Schwarz mit Alulook.

Ich habe mich für das Modell mit der grünen Beleuchtung und schwarzer Frontblende entschieden, da es so am besten zu meinem PC passt.

Montage der Hardware

Da man bei den folgenden Schritten an der Hardware eines PCs Änderungen vornimmt, sollte dieser zuerst einmal heruntergefahren werden.

Es empfiehlt sich, die Schutzfolie des LCDs abzuziehen, bevor es in den Rahmen gebaut wird, da sie sonst eingeklemmt werden könnte und danach nicht mehr komplett zu entfernen ist. Zuerst steckt man die weißen Abstandshalter auf die Aufhahmegewinde des Rahmens, danach kommt das Modul drauf. Der Einbau des Moduls in den Rahmen sollte recht reibungslos verlaufen, eventuell müssen die Aufnahmeschrauben des Rahmens etwas zurechtgebogen werden um in die Löcher des Moduls zu passen. Anschliessend werden die beim Rahmen mitgelieferten Muttern auf die Gewinde geschraubt und festgezogen.

Nun wird die Frontblende von der hinteren Folie befreit und vorne auf den Rahmen geklebt. Man sollte sie an den vier LEDs auf der linken Seite ausrichten, weil man sonst den Rahmen an dieser

Stelle durchsieht und die Leuchtdioden zum Teil verdeckt werden.

Nun kann das komplette Modul in einem 5¼“-Schacht verbaut werden, allerdings sollte man das Verbindungskabel vom Modul zum USB-Anschluss schon jetzt anschliessen und von vorne in das Gehäuse fädeln. Schraubt man zuerst den Rahmen fest, wird man feststellen, dass man die Rückseite des Moduls von innen nicht erreicht und somit das Kabel im Nachhinein nicht mehr anschliessen kann. Der Einbaurahmen wird mit den vier mitgelieferten Schrauben im PC-Gehäuse befestigt, Standardschrauben für PC-Laufwerke passen im Normalfall nicht, da sie minimal dicker sind. Hat man auch die andere Seite des Kabels verbunden (entweder den Stecker in eine freie USB-Buchse gesteckt oder die vier Steckkontakte auf die entsprechenden Pins auf die Steckleiste des Mainboards gesteckt), kann man seinen Rechner wieder einschalten. Manch einen wird es wundern, dass das Display nicht sofort nach dem Einschalten beginnt zu leuchten, obwohl die Stromversorgung der USB-Anschlüsse schon vorhanden ist. Dies ist allerdings kein Fehler des Displays, es sollte sich allerdings irgendwann beim Booten (genauer gesagt, wenn der USB-Controller aktiviert wird) initialisieren.

Erster Kontakt zwischen Hard- und Software

Der Kernel sollte das neue USB-Gerät mit einer Zeile wie etwa

```
usb *; new full speed USD device using * and address *
```

melden. Bei meiner Version des CFA-635 gab der verwendete USB-Chip die Vendor-ID *0x0403* (= *Future Technology Devices International, Ltd*) und die Product-ID *0xFC0D* an. Allerdings enthält der Informationsfeld *iManufacturer* den Wert *Crystalfontz* und das Feld *iProduct* wird mit der Angabe *Crystalfontz CFA-635 USB LCD* noch deutlicher. Mit diesen Informationen, die sich alle per „*lsusb -v*“ abrufen lassen, sollte es jedem gelingen, das CFA-635 an seinem Rechner zu finden.

Wie sich an der VendorID schon erkennen lässt, befindet sich ein USB-to-Serial-Converter der Firma Future Technology auf dem Modul. Glücklicherweise befindet sich der Treiber hierfür bei den neueren Versionen direkt im Kernel, es muss also nichts extern installiert werden, um mit dem Display zu kommunizieren. Bei meinem Kernel (2.6.17) war der Treiber hier zu aktivieren:

```
„Device Drivers -> USB support ->USB Serial Converter support -> USB FTDI Single Port Serial Driver“
```

Es spricht nichts dagegen, den Treiber als Modul zu übersetzen, man kann das Display dann ohne Neustart nutzen, muss aber daran denken, das Modul bei jedem Neustart zu laden. Nachdem man den neuen Kernel gebootet hat oder das Modul in den laufenden Kernel inserted hat, sollte sich der Treiber, ein korrekt angeschlossenes CFA-635 vorrausgesetzt, mit diesen Zeilen in den Kernelnachrichten melden (wobei Versionsnummern, Chiptyp und andere Angaben abweichen

können):

```
ftdi_sio 1-3:1.0: FTDI USB Serial Device converter detected
```

```
drivers/usb/serial/ftdi_sio.c: Detected FT8U232AM
```

```
usb 1-3: FTDI USB Serial Device converter now attached to ttyUSB0
```

Besondere Beachtung verdient das letzte Argument der letzten Zeile, in diesem Fall „*ttyUSB0*“.

Dies ist der Device-Node, über den das Display anzusprechen ist, in diesem Fall „*/dev/ttyUSB0*“.

Dieses Argument ist später dem displaysteuernden Programm anzugeben.

Hat man die oben genannten Meldungen des Kernels erhalten, kann man direkt mit dem Display kommunizieren und es ist einsatzbereit.

Die Software

Das Programmpaket, das ich verwende um das LCD unter Linux anzusteuern ist *LCDproc*. Es basiert auf dem Client-Server-Prinzip, es gibt also einen Server, der Daten von anderen Programmen annimmt und an das Display weiterreicht und im Gegenzug Events des Displays (Tastendrucke am Keypad) an die Clients übergibt, wenn diese dem Server gemeldet haben, solche Daten empfangen zu wollen. Aufgrund dieses Designs können nicht nur mehrere Programme Informationen auf dem Display darstellen, es besteht auch die Möglichkeit, dass entfernte Maschinen sich per Netzwerk mit dem Server verbinden und man somit auf einem Display den Status mehrerer Rechner verfolgen kann.

Ein weiterer Vorteil von *LCDproc* ist, dass es sehr viele verschiedene Displays unterstützt, wie etwa alle HD44780-basierten Module, viele Displays von *Crystalfontz*, *Matrix Orbital*, *Bayrad*, *SED1330*- und *SED1520*-basierte Grafikmodule, *Cwlinux*- und „*pyramid computer*“-Displays, *EyeboxOne*-Module, das LCD auf *Logitech G15* Gamingtastaturen, das Anzeigemodul des *Medion 8800* PCs und die Displays der *Tyan GS10* und *Tyan GS12* Barebones. Hierbei ist es für die Clients fast irrelevant, welches Display die Informationen wiedergibt, da die Darstellung auf dem Display die Aufgabe des Servers ist. Dieser teilt einem Client beim Anmelden mit, wie viele Zeichen das Display in einer Zeile darstellen kann, wie viele Zeilen es gibt und aus wie vielen Pixeln sich ein Buchstabe zusammensetzt.

LCDProc und die meisten Clients sind unter der *GNU General Public License (GPL)* lizenziert, d.h. der Einsatz der Programme ist kostenfrei, der komplette Quellcode ist frei verfügbar und jeder darf ihn ändern, unter der Voraussetzung, dass diese Änderungen ebenfalls wieder frei verfügbar gemacht werden. Es kann also jegliches Verhalten von *LCDproc* geändert werden.

Um die Kommunikation zwischen Entwicklern und Nutzern aufrecht zu erhalten, gibt es eine sehr gute Mailingliste, auf der alle auftretenden Fragen schnell und kompetent beantwortet werden sowie alle Vorschläge lebhaft diskutiert werden.

Installation und Konfiguration von LCDproc

Für den Einsatz eines CFA-635-Moduls sollte man den Quellcode per CVS herunterladen, da nur so aktuelle Verbesserungen seit der letzten stabilen Version berücksichtigt werden. Um sich den aktuellen Quellcode per CVS auf den Rechner zu laden, werden folgende Kommandos benutzt:

```
cvs -d:pserver:anonymous@lcdproc.cvs.sourceforge.net:/cvsroot/lcdproc login
```

Auf die nun folgende Aufforderung, ein Passwort einzugeben, drückt man einfach die Return-Taste, da kein Passwort benötigt wird. Nun ist man bei dem CVS-Server angemeldet und kann sich den Quelltext herunterladen:

```
cvs -z3 -d:pserver:anonymous@lcdproc.cvs.sourceforge.net:/cvsroot/lcdproc co -P lcdproc
```

Anschliessend wechselt man mit „*cd lcdproc*“ in das Verzeichnis, wo der Quellcode abgelegt wurde. Das bekannte Script „*configure*“ muss noch mit Hilfe von „*sh autogen.sh*“ erstellt werden. Nun kann sich der interessierte Nutzer mit „*./configure --help*“ alle möglichen Optionen ansehen und entscheiden, welche für sein System am besten geeignet sind, für das CFA-635 sollte ein „*./configure --enable-drivers=CFontzPacket --enable-seamless-hbars*“ alles Nötige einstellen. Ist dieser Schritt beendet, wird das Programm mit „*make*“ übersetzt. Für die nachfolgende Installation per „*make install*“ sind Superuser-Rechte notwendig, die mit „*su*“ erlangt werden.

Hat keiner der oben genannten Schritte einen Fehler produziert, sollten der Server zur Ansteuerung des Displays (*/usr/sbin/LCDD*) und ein Client zur Anzeige einiger Systeminformationen (*/usr/bin/lcdproc*) vorhanden sein.

Nach der erfolgreichen Installation des Pakets muss es nun konfiguriert werden, damit etwa der Server weiß, welches Display er wie anzusteuern hat und der Client weiß, welche Informationen an welchen Server weitergeleitet werden.

Die Konfigurationsdatei befindet sich standardmäßig in */etc/LCDD.conf*, dieser Speicherort kann aber beliebig geändert werden, da dem Serverprogramm per Kommandozeile eine andere Konfigurationsdatei zugewiesen werden kann. Eine gut kommentierte Standardkonfigurationsdatei mit Beispielen für alle unterstützten Treiber wird mit dem Programmpaket mitinstalliert. Auf den ersten Blick erkennt man, dass sich die Konfiguration des Servers in 3 Abschnitte gliedert:

- die globale *[server]*-Sektion, die dem Server Basisinformationen wie etwa den Netzwerkport und die zu verwendenden Treiber mitteilt
- die *[menu]*-Sektion, die dem im Serverprogramm integrierte Menü Tasten zuweist
- die Treiber-Sektionen, die zu jedem Treiber seine spezifischen Parameter speichern

In der *[server]*-Sektion ist zuerst der verwendende Treiber zu bestimmen, in unserem Fall für das CFA-635: „*Driver=CFontzPacket*“. Weiterhin ist der Pfad anzugeben, in dem alle Treiber bei der Installation abgelegt wurden. Dies ist bei Verwendung der oben genannten Befehle „*DriverPath=/usr/share/lcdproc/drivers/*“. Alle weiteren Einstellungen können bei ihren

Standardwerten belassen werden, ich empfehle allerdings, sich die Kommentare durchzulesen um dann individuell entscheiden zu können, wie sich der Server verhalten soll. Gleiches gilt für die [menu]-Sektion.

Alle weiteren Sektionen bis auf die „[CfontzPacket]“-Sektion können aufgrund der Übersichtlichkeit gelöscht werden, da die dort beschriebenen Treiber weder geladen noch genutzt werden.

Nun folgt die Konfiguration des Treibers für das CFA-635 selbst, wozu in der „[CfontzPacket]“-Sektion zuerst die Option „Model“ auf „635“ gesetzt wird, da der Treiber die Modelle 631, 633 und 635 unterstützt. Die nächste Option „Device“ bekommt den Wert, den der Kerneltreiber des USB-Seriell-Wandlers in die Kernellog schreibt. In meinem Fall ist das, wie im Abschnitt „Erster Kontakt zwischen Hard- und Software“ dieses Dokuments erwähnt, „/dev/ttyUSB0“. Auf diese Option muss besonders geachtet werden, wenn mehrere USB-Seriell-Wandler (wie weitere Displays, GPS-Empfänger, ...) am selben PC verwendet werden. Die Option „Size“ erhält den Wert „20x4“, die Option „Speed“ wird auf „115200“ eingestellt und bei „Reboot“ sollte „yes“ angegeben werden, damit das Display bei jedem Start des Serverprogramms auf einen bekannten Status gesetzt wird. Die Option „Contrast“ musste ich leicht korrigieren, ich befand „475“ als guten Wert, dieser kann aber von Modul zu Modul schwanken. Die letzten beiden Optionen „Brightness“ und „OffBrightness“ sollten individuell den lokalen Gegebenheiten angepasst werden. Weiterhin soll erwähnt werden, dass sich die letzten drei erwähnten Einstellungen (*Contrast*, *Brightness* und *OffBrightness*) während des Betriebs mit dem im Server integrierten Menü komfortabel ändern lassen. Solche Änderungen gelten allerdings nur bis zum nächsten Neustart des Serverprogramms, das dann die Werte wieder aus der Konfigurationsdatei einliest.

Da nun die Konfiguration des LCD-Servers beendet ist, wird die Datei gespeichert und der Server mit „LCDd“ gestartet. Ich empfehle, dies als Superuser zu tun, in der Konfigurationsdatei lässt sich ein Benutzer einstellen, dessen Rechte LCDd nach der Initialisierung erhält.

Nach dem Start des Programms sollte nach kurzer Zeit auf dem Display zu lesen sein, dass noch keine Clients verbunden sind und somit auch noch kein Screen angelegt wurde. Mit einem Druck auf das rote X am Modul erscheint das Menü indem u.a. Kontrast und Helligkeit der Beleuchtung eingestellt werden kann. Um einige Systemdaten des Rechners anzeigen zu lassen, startet man „lcdproc L M T C S“, wobei jeder Buchstabe für eine bestimmte Informationsseite (*Screen*) steht, eine Übersicht der möglichen Screens liefert „lcdproc --help“. Weitere Clients können auf der Homepage von LCDproc heruntergeladen werden, es können aber mit geringen Programmierkenntnissen eigene Clients erstellt werden. Ich verweise hierfür auf die *Developer Documentation*, die bei der Installation mitkopiert wurde aber auch auf der Homepage zu finden ist.

Besonderheiten des CFA-635 aus Sicht der Clients

Wie bereits oben erwähnt, kann es den Clients des LCDd ziemlich egal sein, auf welchem Display ihre Daten dargestellt werden. Da sich die hardwarebedingten Darstellungsmöglichkeiten von Modell zu Modell unterscheiden gibt es einige kleine Dinge zu beachten. Als erstes ist hier der verwendbare Zeichensatz auf dem Display zu nennen. Jedes Charakter-Display hat einen fest im Controller eingebauten Zeichensatz, der festlegt, welches Zeichen bei welcher Steuernummer ausgegeben wird. Der Zeichensatz des CFA-635 unterscheidet sich zum Teil stark vom dem eines HD44780-basierten Moduls. Während sich alle alphanumerischen Zeichen (A-Z;a-z;0-9) ohne Probleme auf dem CFA-635 darstellen lassen, bereiten einige Sonderzeichen Probleme. So ist es zum Beispiel nicht möglich, dass Grad-Zeichen „°“ auf dem Display gezeigt zu bekommen, das es einfach nicht im Zeichensatz enthalten ist. Positiv fällt allerdings auf, dass alle deutschen Umlaute und die meisten latin-1-Zeichen definiert sind. Die nachfolgende Tabelle vergleicht den Zeichensatz des HD44780 und den des CFA-635:

Upper 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	(0)	!	1	A	Q	a	q									
xxxx0001	(2)	"	2	B	R	b	r									
xxxx0010	(3)	#	3	C	S	c	s									
xxxx0011	(4)	\$	4	D	T	d	t									
xxxx0100	(5)	%	5	E	U	e	u									
xxxx0101	(6)	&	6	F	V	f	v									
xxxx0110	(7)	'	7	G	W	g	w									
xxxx0111	(8)	<	8	H	X	h	x									
xxxx1000	(1)	>	9	I	Y	i	y									
xxxx1001	(2)	*	:	J	Z	j	z									
xxxx1010	(3)	+	;	K	L	k	l									
xxxx1011	(4)	,	<	L	l											
xxxx1100	(5)	-	=	M	J	m	j									
xxxx1101	(6)	.	>	N	^	n	^									
xxxx1110	(7)	/	?	O	_	o	_									
xxxx1111	(8)															

upper 4 bits	0s	16s	32s	48s	64s	80s	96s	112s	128s	144s	160s	176s	192s	208s	224s	240s
lower 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0s	CGRAM [0]	▶	◀	◀	◀	◀	◀	◀	◀	◀	◀	◀	◀	◀	◀	◀
1s	CGRAM [1]	◀	!	1	A	Q	a	q	!	!	!	!	!	!	!	!
2s	CGRAM [2]	◀	"	2	B	R	b	r	!	!	!	!	!	!	!	!
3s	CGRAM [3]	◀	#	3	C	S	c	s	!	!	!	!	!	!	!	!
4s	CGRAM [4]	◀	\$	4	D	T	d	t	!	!	!	!	!	!	!	!
5s	CGRAM [5]	◀	%	5	E	U	e	u	!	!	!	!	!	!	!	!
6s	CGRAM [6]	◀	&	6	F	V	f	v	!	!	!	!	!	!	!	!
7s	CGRAM [7]	◀	'	7	G	W	g	w	!	!	!	!	!	!	!	!
8s	CGRAM [0]	◀	<	8	H	X	h	x	!	!	!	!	!	!	!	!
9s	CGRAM [1]	◀	>	9	I	Y	i	y	!	!	!	!	!	!	!	!
10s	CGRAM [2]	◀	*	:	J	Z	j	z	!	!	!	!	!	!	!	!
11s	CGRAM [3]	◀	+	;	K	L	k	l	!	!	!	!	!	!	!	!
12s	CGRAM [4]	◀	,	<	L	l			!	!	!	!	!	!	!	!
13s	CGRAM [5]	◀	-	=	M	J	m	j	!	!	!	!	!	!	!	!
14s	CGRAM [6]	◀	.	>	N	^	n	^	!	!	!	!	!	!	!	!
15s	CGRAM [7]	◀	/	?	O	_	o	_	!	!	!	!	!	!	!	!

Figure 8. Character Generator ROM (CGROM)

Zeichensatz des HD44780

Zeichensatz des CFA-635

Eine weitere Besonderheit des CFA-635 betrifft die vier Leuchtdioden auf der linken Seite des Moduls. Diese können auch per LCDd angesteuert werden, allerdings lassen sich diese nur schalten, nicht stufenlos dimmen, obwohl es technisch möglich wäre. Die Ansteuerung geschieht

über das *output* Kommando an den Server, gefolgt von einer Zahl, die beschreibt, welchen Status die Leuchtdioden erhalten. Diese Zahl besteht aus acht Bit, also einem Byte und ist bitmaskiert. Das erste Bit bestimmt den Status des grünen Teils der ersten Leuchtdiode, das zweite den grünen Teil der zweiten Leuchtdiode, usw. Das fünfte Bit bestimmt den roten Teil der ersten LED, das sechste den roten Teil der zweiten LED, usw. Möchte man also, dass die erste Leuchtdiode orange leuchtet (grün und rot), die zweite gar nicht, die dritte rot und die vierte grün, so ist die Bitfolge „10011010“ die richtige Lösung. Das Byte entspricht Dezimal 154 (und Hex 0x9A), es ist also „*output 154*“ von einem Client an LCDd zu senden.

Fazit

Das CFA-635 macht einen durchweg guten Eindruck. Bis auf die im Vergleich zu einem HD44780-basierten Modul etwas langsame Aktualisierungsrate des Displays (wahrscheinlich bedingt durch eine vom LCDd eingefügte Pause, um Timingprobleme zu verhindern) und das Fehlen des Grad-Zeichens im Zeichensatz ein gelungenes Produkt. Das Display ist gut lesbar, das Backlight und der Kontrast ist per Software einstellbar, das Keypad macht einen soliden Eindruck und reagiert gut. Der Preis ist mit ca. 80\$ noch in Ordnung, wenn auch nicht ganz billig. Der Einsatz unter Linux funktioniert einwandfrei, bis auf die fehlende Möglichkeit, die LEDs zu dimmen, was aber nicht wirklich störend ist.