8. Beginn Handbuch - Verwendungsmöglichkeiten des Simulators

Hier wird nun noch einmal ein Verbindungsaufbau mit Telnet und dem Programm PacketSender für Skripte ohne C-Programm gezeigt. Im Anschluss wird die Methode vorgestellt, mit der das Skript zuerst in der Kommandozeile als C-Programm geschrieben, kompiliert und dann an den Simulator gesendet wird. Für Skripte ist generell eine Kenntnis der möglichen Befehle nötig. Daher finden Sie hier eine Übersicht zu den Kommando-Befehlen[8] und ihren Bedeutungen:

Hex-Code	Display Reaktionen
00:XX	Lässt den Cursor an eine beliebige gültige Adresse springen. „XX“ ist die Hex-kodierte Dezimalzahl von 0 und 79.
01	Gesamte Anzeige Löschen und Datenspeicheradresse (Cursor) auf null setzen
02	Datenspeicheradresse wird auf null gesetzt. Anzeige an Originalposition (keine Verschiebung). Der Dateninhalt bleibt erhalten.
10	Display aus, Cursor aus, Blinken aus
11	Display ein, Cursor aus, Blinken aus
12	Display ein, Cursor ein, Blinken aus
13	Display ein, Cursor ein, Blinken ein
20	Verschiebt den Cursor um eine Stelle nach links
21	Verschiebt den Cursor um eine Stelle nach rechts
22	Verschiebt das Display um eine Stelle nach links
23	Verschiebt das Display um eine Stelle nach rechts
30	Übertragungsmodus 4-Bit, 2-Zeilen, 5x8 Pixel
31	Übertragungsmodus 8-Bit, 2-Zeilen, 5x8 Pixel
RS=0	Der Simulator befindet sich im Kommando-Modus
RS=1	Der Simulator befindet sich im Daten-Modus
RW=0	Der Simulator befindet sich im Schreib-Modus
RW=1	Der Simulator befindet sich im Lese-Modus
Delay=X	Übergabe einer neuen Zeitverzögerung; X in Millisekunden

8.1 Telnet

Starten Sie den Telnet-Client, indem Sie in die Kommandozeile den Befehl **telnet** eintippen. Geben Sie anschließend **open localhost 2000** ein, um über den Port 2000 eine Telnet-Verbindung zwischen dem (virtuellen) Server des Display-Simulators und Ihrem lokalen System aufzubauen. Sobald die Verbindung besteht, sehen die Kontrollsignale wie folgt aus:

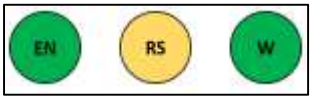

Abbildung 18 - Kontrollsignale im Kommando-Modus

Das **EN**able-Kontrollsignal sollte jetzt durch seine grüne Farbe die aufgebaute Verbindung signalisieren. Das **R**egister-**S**elect-Signal dagegen sollte gelb sein (dies signalisiert, dass Sie das Befehlsregister nutzen) und das **R**ead-**W**rite-Signal sollte ebenfalls grün sein. Grün steht bei dieser

[8] Die Datenmodus-Befehle befinden sich unter Kapitel 8.7. Dort können Sie einsehen, wie Sie Displaybuchstaben kodieren.

Kontrollleuchte für den Schreibzugriff. Sie können jetzt die Befehle einzeln, das heißt jeden Befehl durch die Enter-Taste bestätigen oder mehrere Befehle durch ein Leerzeichen trennen und abschließend durch die Enter-Taster an den Simulator senden. Mit Telnet ist das Versenden von Skripten unter Windows nicht möglich.

8.2 PacketSender

Um sich über PacketSender mit dem LCD-Simulator zu verbinden, muss dieser zuerst gestartet werden. Das Programm befindet sich auf der Daten-CD im Anhang. Im folgenden Fenster geben Sie bitte **127.0.0.1** als „Address“ und **2000** als „Port“ ein. Klicken Sie anschließend auf den Button „**Send**“ wie folgende Abbildung illustriert:

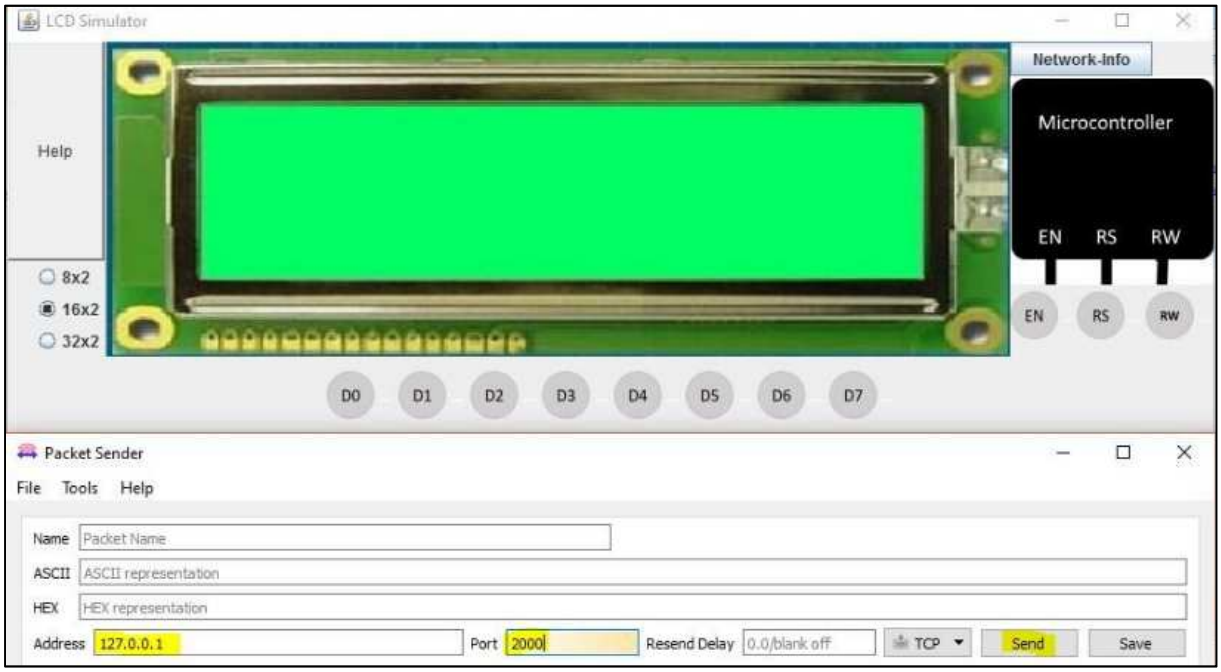

Abbildung 19 - Öffnen der TCP/IP-Verbindung über PacketSender

Die nächsten Schritte nach dem Verbindungsaufbau werden jetzt beschrieben.

Nur im Datenmodus können Sie Buchstaben auf das Display schreiben. Er wird durch den Befehl „RS = 1“ aktiviert. Sobald dieser Modus aktiviert ist, sehen die Steuersignalleuchten auf der rechten Seite des Simulators folgendermaßen aus (beachten Sie, dass RS jetzt grün ist).

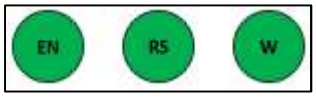

Abbildung 20 - Steuersignalleuchten im Daten-Modus

Um ein Gefühl dafür zu bekommen, verbinden Sie sich mit dem Simulator, wenn Sie dies noch nicht getan haben. Danach starten Sie ein Beispielprogramm mit Telnet oder führen Sie ein Beispiel-Skript mit PacketSender aus. Auch die Beispiel-Skripte finden Sie auf der Daten-CD im Anhang.

8.3 Beispielprogramm mit Telnet

Senden Sie das Kommando „**13**“, um die Anzeige des LC-Displays einzuschalten. Drücken Sie anschließend die Enter-Taste oder geben Sie ein Leerzeichen ein. Geben Sie nun „**RS=1**“ ein, um den Datenmodus zu aktivieren. Drücken Sie Enter oder geben Sie ein Leerzeichen ein. Geben Sie nun folgende[9] Hex-Codes jeweils getrennt durch ein Leerzeichen ein: „**48 61 6C 6c 6F 20 57 65 6c 74**“. Drücken Sie die Enter-Taste. Sie sollten die Worte "Hallo Welt" auf dem Display sehen. Diese Hex-Codes stammen aus der Zeichensatztabelle. Dies beschreibt das Kapitel 8.6.

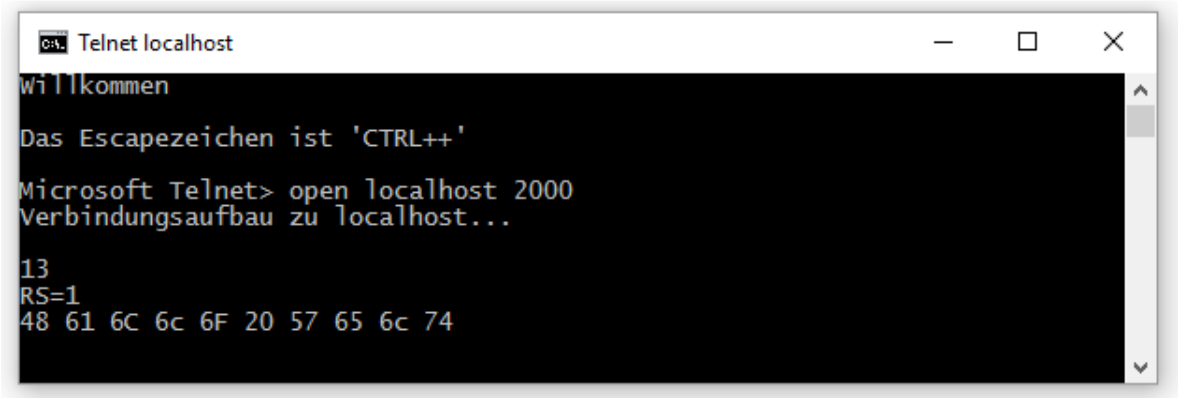

Abbildung 21 - "Hallo Welt" mit Telnet

8.4 Beispielprogramm mit PacketSender

Nach dem Sie die den Abschnitt 8.2 durchgeführt haben, sollen Sie jetzt mit PacketSender arbeiten. Durch die Eingaben der IP-Adresse und des Ports, haben Sie durch den Klick auf „Send“ ein weiteres Programmfenster geöffnet, mit dem nun die Kommunikation mit dem Simulator möglich ist.

Geben Sie wie in Abbildung auf der nächsten Seite zu sehen ist zunächst „**13**“ ein, um das Display einzuschalten. Danach „**RS = 1**“ für den Datenmodus und nun die Anzeigedaten: „**48 65 6c 6c 6f 20 73 75 64 65 6e 74**“ und klicken Sie mit der linken Maustaste auf die Schaltfläche „**Send“**.

Der Text "Hello student" erscheint. Auch hier spielt es keine Rolle, ob die Eingaben immer direkt oder alle zusammen an den Simulator gesendet werden.

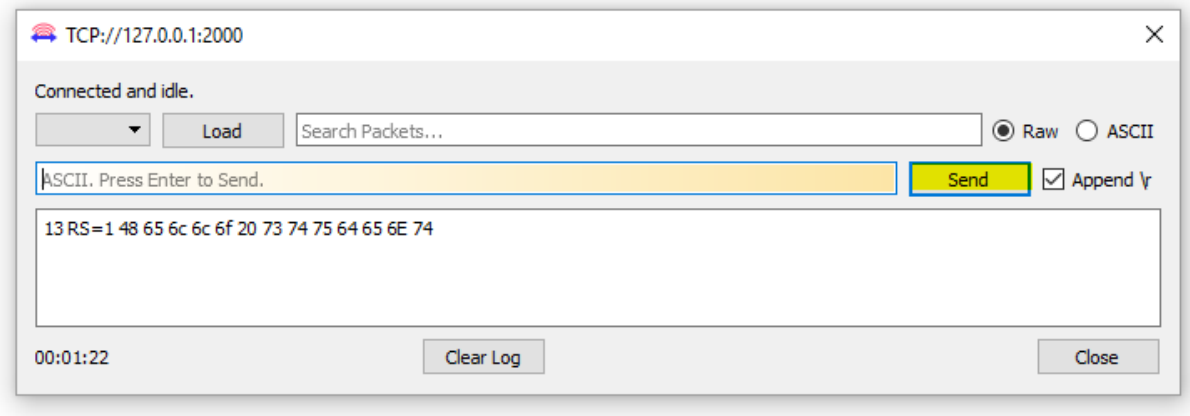

Abbildung 22 - "Hello student" mit PacketSender

[9] Die Buchstaben des Displays sind hexadezimal kodiert. Beachten Sie hierzu die Zeichensatztabelle in Kapitel 8.6, wenn Sie einen eigenen Text schreiben möchten.

Natürlich können auch Inhalte aus der Zwischenablage über den PacketSender an den Simulator gesendet werden. Mit dem Wissen aus dem Kapitel 5, ist es nun möglich einfach Skripte zu verfassen. Diese können in einer .txt-Datei geschrieben werden.

Zu Erinnerung: Kommentare können Sie mit dem /*…*/ anlegen.

Hier nun ein Beispiel-Skript:

```
/*Willkommen zur Display-Simulation*/
Delay=800                                          /*Zeitverzoegerung in Millisekunden*/
RS=0                                               /*Kommando Modus*/
01                                                 /*Loeschen des DRAMs*/
10                                                 /*LCD ausschalten*/
13                                                 /*LCD einschalten mit blinkendem Cursor*/
01                                                 /* Clear LCD*/
/*21*/                                             /* Cursor Shift right - auskommentiert */
RS=1                                               /*Daten Modus*/
57 69 6c 6c 6b 6f 6d 6d 65 6e  20  7a 75 72        /* Willkommen zur */
RS=0 00:28                                         /*Cursor springt 2. Zeile, 1. Stelle*/
21                                                 /* Cursor Shift right */
RS=1
4c 43 44  20  53 69 6d 75 6c 61 74 69 6F 6E 21     /* LCD Simulation! */
RS=0                                               /*Kommando Mode*/
```

Der fettgedruckte Befehl erzeugt ein Hinweisfenster, weil ein Löschen des Display-Inhaltes nur durchgeführt werden kann, wenn das Display auch eingeschaltet ist.

Abbildung 23 - InfoBox - You can't del...

Danach wird das Skript aber ordnungsgemäß ausgeführt und schreibt folgendes auf das Display.

Abbildung 24 - LCD-Anzeige nach Ausführung des Skriptes

8.5 Bash-Kommandozeile zur Befehlsübertragung verwenden

Sie können dasselbe Skript auch mittels der Bash-Kommandozeile[10] an den Display-Simulator verschicken. Legen Sie dazu folgende .c-Datei mit z.B. nano an:

```
#include <stdio.h>
int main(){
  printf("/*Herzlich-Willkommen*/                                  \r\n");
  printf     ("Delay=0      /*     Zeitverzoegerung*/             \r\n");
  printf     ("RS=0 /*Kommando Modus*/                             \r\n");
  printf     ("01   /*Loeschen des DRAMs*/                         \r\n");
  printf     ("10          /*LCD ausschalten*/                     \r\n");
  printf     ("13   /*LCD einschalten mit blinkendem Cursor*/     \r\n");
  printf     ("01                /*LCD clear*/                                  \r\n");
  printf     ("/*21*/            /* Cursor Shift right */                       \r\n");
  printf     ("RS=1 /*Daten Modus*/                                             \r\n");
  printf     ("57 69 6c 6c 6b 6f 6d 6d 65 6e 20  7a 75 72 /*Willkommen zur*/  \r\n");
  printf     ("RS=0 00:28   /*Cursor springt in die zweite Zeile*/             \r\n");
  printf     ("21           /* Cursor Shift right */                          \r\n");
  printf     ("RS=1                                                           \r\n");
  printf  ("4c 43 44 20 53 69 6d 75 6c 61 74 69 6F 6E 21 /*LCD Simulation!*/  \r\n");
  printf     ("RS=0              /*Kommando Mode*/                  \r\n");
  printf     ("/*02*/      /*Return Home*/                          \r\n");
getchar();
}
```

Es wurde also ein Header:

```
#include <stdio.h>
int main(){
```

und ein Footer hinzugefügt:

```
getchar();
}
```

Daten die an den Simulator gesendet werden, müssen nun in das Konstrukt: `printf("…\r\n");` eingeschlossen werden.

Dazu wurde die vorherige Datei „HerzlichWillkommen.txt“ einfach in „HerzlichWillkommen.c“ umbenannt und die oben beschriebenen Änderungen vorgenommen. Über die Kommandozeile kann jetzt in den Ordner navigiert werden, in dem sich die C-Datei befindet. Mit dem Befehl „**ls -l**“ können Verzeichnisse und Inhalte von Verzeichnissen angezeigt werden. Dies zeigt die nächste Abbildung.

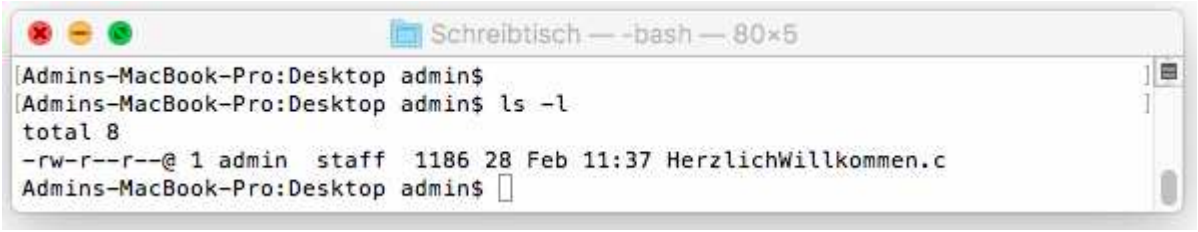

Abbildung 25 - Der Befehl "ls -l"

[10] Die Bash-Kommandozeile können Sie bei Windows 10 unter folgender Anleitung als Betaversion installieren: https://www.heise.de/newsticker/meldung/Hands-on-Das-neue-Linux-Subsystem-in-Windows-10-3163994.html. Bei Mac OS und Linux ist die „**B**ourne **a**gain **sh**ell“ standardmäßig installiert.

Mit dem Befehl „**nano HerzlichWillkommen.c**“ kann die Datei betrachtet und weiter bearbeitet werden. Das zeigt die folgende Abbildung.

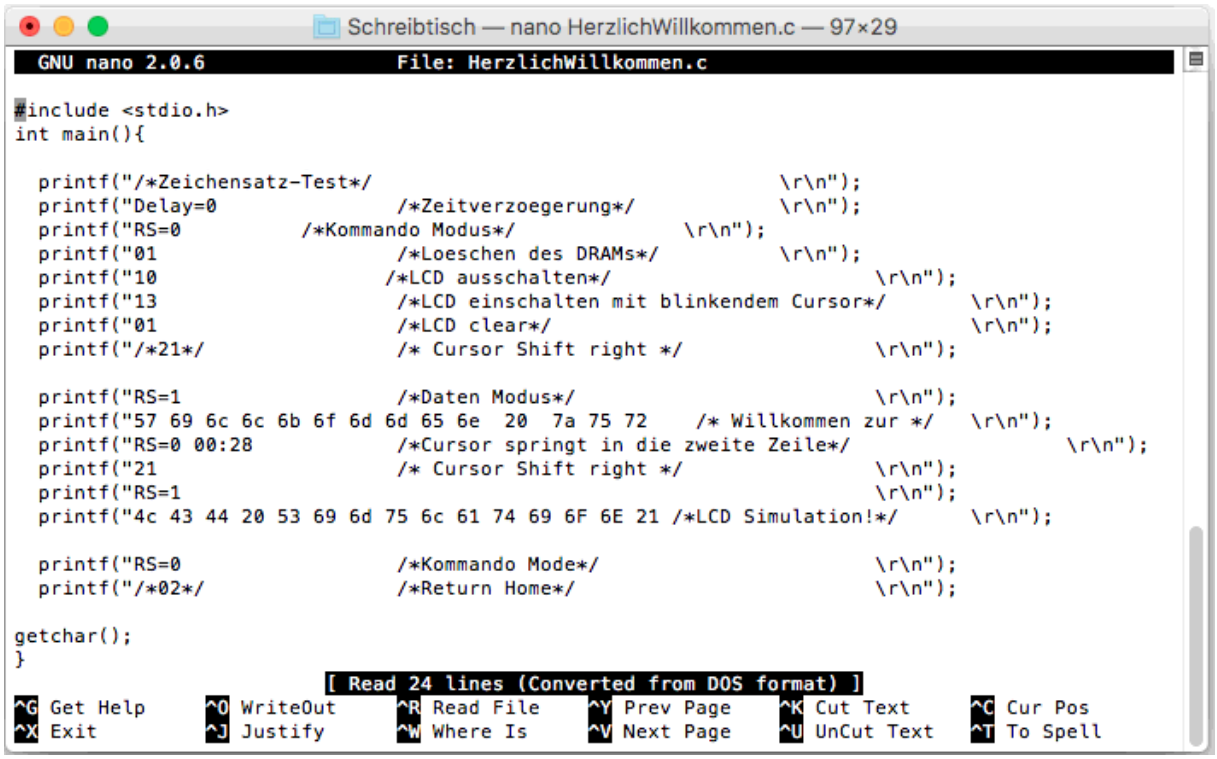

Abbildung 26 - Skriptbearbeitung mit nano

Mit „**Steuerung + o“** kann die Datei gespeichert werden. Mit „**Steuerung + x“** wird Nano wieder verlassen.

Nun muss das C-Programm noch kompiliert werden, damit es danach ausgeführt werden kann. Dazu wird der Befehl „**gcc -o hw HerzlichWillkommen.c**“ in die Konsole eingegeben. Der „**gcc**“ ist ein Standard-Kompilierer, der Parameter „**-o**“ bedeutet es soll eine Ausgabedatei mit dem dahinter stehenden Namen (hier „**hw**“) erzeugt werden. Die Eingabe C-Datei ist das „**HerzlichWillkommen.c**“. Die soeben erzeugte Datei wird im aktuellen Verzeichnis abgelegt. Nun muss das C-Programm noch ausgeführt werden und der Ausgabestrom muss zum Simulator gelangen.

Mit diesem Befehl "**./hw | nc localhost 2000**“ wird das C-Programm ausgeführt und der Ausgabestrom in Form der einzelnen Display Befehle "**./hw**“ wird in ein anderes Programm geleitet „**|**“. Das hier genutzte Programm heißt Netcat „**nc**“, welches noch die Parameter für den Verbindungsaufbau zum Simulator benötigt „**localhost 2000**“. Dieser Befehl muss **zwei** Mal durch die Enter-Taste bestätigt werden. Mit dem ersten Enter wird die Verbindung zum Simulator aufgebaut. Mit dem zweiten Enter wird der Ausgabestrom auf die gerade erzeugte Verbindung umgeleitet. Dies zeigt auch die nächste Abbildung, wo die Leerzeile deutlich zu sehen ist.

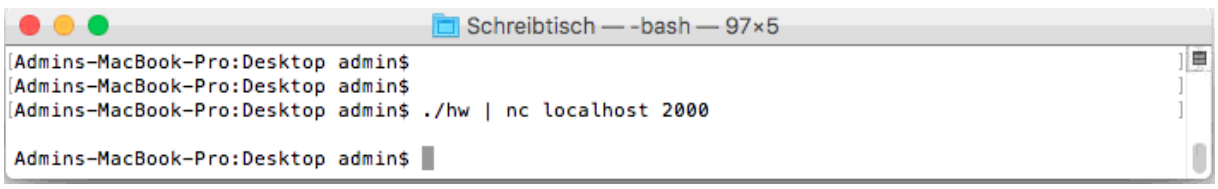

Abbildung 27 - Befehl zum TCP/IP-Socket öffnen und Senden des Streams

8.6 Auslesen von Zeichen

Das Auslesen von Zeichen ist möglich, wenn die beiden Steuersignal RS und RW logisch Eins sind. Dann müssen nur die Positionen der Display-Stellen übergeben werden, die ausgelesen werden sollen. Dies zeigt der folgende Code-Abschnitt. Ein vollständiges Beispiel-Skript ist auf der Daten-CD vorhanden.

```
...
printf("RW=1  /* Lese-Modus */                                                \r\n");
printf("RS=1  /* Daten-Modus */                                               \r\n");
printf("00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f /* Willkommen zur */  \r\n");
printf("28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 /* LCD Simulation! */ \r\n");
printf("RW=0  /* Schreib-Modus */                                              \r\n");
printf("RS=0  /* Kommando-Modus */                                            \r\n");
...
```

Der Hex-Code und das entsprechende Zeichen werden in einer Zeile neben einander dargestellt.

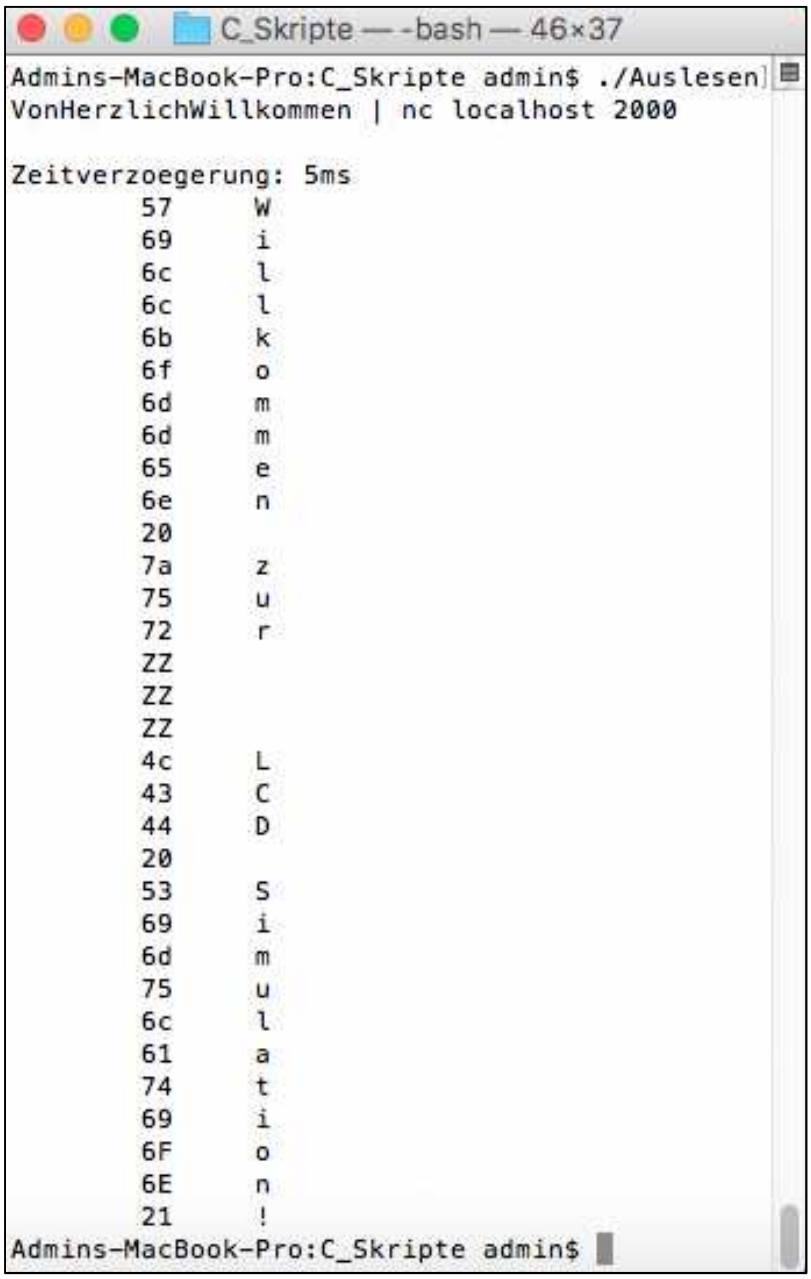

Abbildung 28 - Auslesen von Zeichen

8.7 Zeichensatz des Simulators

Die hier abgebildete Tabelle erlaubt es alle hier dargestellten Zeichen auf dem Simulator anzeigen zu lassen. Diese Zeichen sind in hexadezimaler Schreibweise kodiert. Eine dezimale „4“ hat den Hex-Code: 34. Das bedeutet, man legt die 3x und die x4 übereinander und erhalt somit 34.

Abbildung 29 - Zeichensatz des LCD-Simulators

Nicht alle Zeichen des Rasters 5x10 können auf dem Simulator auch richtig angezeigt werden.

Die 32 Zeichen der Adressen E0-EF und F0-FF sind in der folgenden Abbildung dargestellt. Ein direkter Vergleich mit der letzten Abbildung zeigt, dass die entsprechenden Zeichen am unteren Rand abgeschnitten sind.

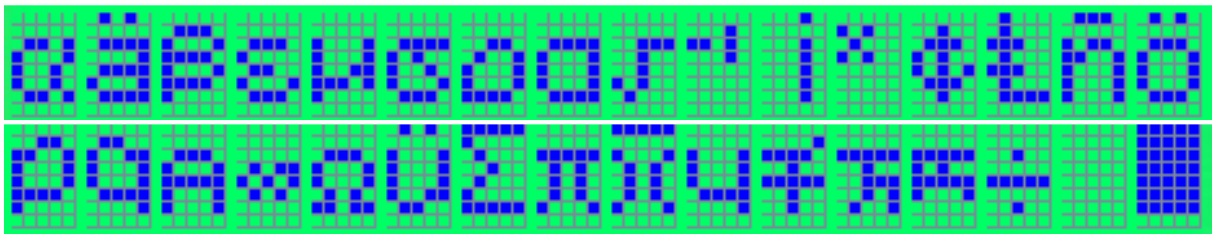

Abbildung 30 - Zeichen der Adressen E0-EF, F0-FF

Die Adressen 00-0F und 10-1F wurden nicht verwendet, da keine selbst erstellten Zeichen in den Simulator eingeladen werden können. Wird im Datenmodus der Adressraum trotzdem benutzt, erscheint eine entsprechende Fehlermeldung.

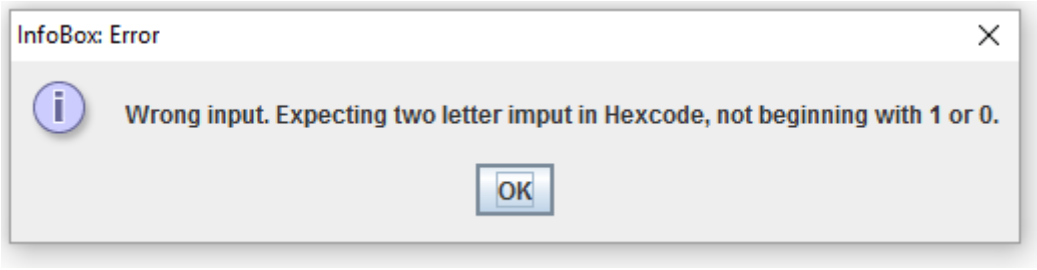

Abbildung 31 - Falscher Adressraum im Datenmodus