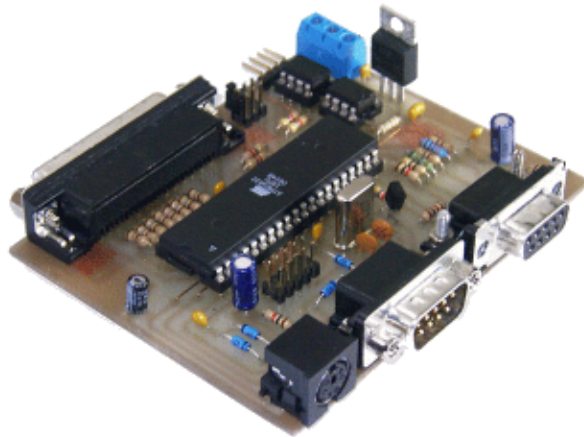


AVR-ChipBasic2: Ein BASIC-programmierbarer Einchip-Computer mit dem ATMega644

V0.69 (c) 2006-2008 Jörg Wolfram



1 Allgemeines

1.1 Rechtliches

Das Programm unterliegt der GPL (GNU General Public Licence) Version 3 oder höher, jede Nutzung der Software/Informationen nonkonform zur GPL oder ausserhalb des Geltungsbereiches der GPL ist untersagt!

Die Veröffentlichung dieses Projekts erfolgt in der Hoffnung, daß es Ihnen von Nutzen sein wird, aber OHNE IRGEND-EINE GARANTIE, auch ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK.

Alle im Text genannten Marken sind Eigentum des entsprechenden Inhabers.

1.2 Geschichte

Angefangen hat es mit der Idee, ein TV-Gerät zum Testen und Debuggen eines Formelparsers zu nutzen, da so mehrere Werte übersichtlich und gleichzeitig darstellbar waren.

Irgendwann im August 2006 kam dann die Idee, einen BASIC-Interpreter mit TV-Ausgabe in den Chip zu integrieren. Um das ganz einfach zu halten, habe ich mich auf einen TINY-BASIC ähnlichen Dialekt beschränkt. Und so kamen dann nacheinander Farbausgabe, ein Menü, ein integrierter Editor und weitere Funktionen dazu. Anfänglich konnte nur ein Programm im EEPROM-Speicher des ATMega16 Platz finden, beim Mega32 wurde dann der Flash benutzt. Nach den Versionen für die ATMega16 und ATMega32 Mikrocontroller stellt die Version für den ATMega644 die logische Weiterentwicklung bei gleicher Hardwarebasis dar.

1.3 Features

Als Eingabegerät dient eine normale PS2-Computertastatur, als Ausgabegerät ein Fernsehgerät mit Scart-Eingang (Farbe) oder BAS-Eingang (Graustufen) oder auch verschiedene PAL-/NTSC-taugliche TFT-Displays.

- 95 Programmzeilen mit maximal 32 nutzbaren Zeichen, Fullscreen-Editor
- 8 Programme im internen Flash
- Tiny-Basic-Programmiersprache mit Erweiterungen

- Eigene Fehlerbehandlung mit ONERR...
- 30x23 Zeichen mit maximal 8 Vorder- und Hintergrundfarben und Pseudografik
- 4 Videomodi, davon 3 mit Vollgrafik
- Sprites im Textmodus, Anzahl ist nur durch Array limitiert
- Schneller Bitblock-Transfer im Textmodus
- PAL/NTSC und Synchronsignale über Jumper einstellbar
- PS2-Tastatur zur Eingabe, Keyboard-Layout ist umschaltbar
- 1-Kanalige Audioausgabe (Noten in 2 Klangfarben, Rauschen) mit Hüllkurve
- Einfacher, im Hintergrund laufender Sequenzer
- serielle RS232-Schnittstelle mit 1200 Baud und Ladungspumpe
- parallele Druckerschnittstelle, auch als I/O und Analogeingänge nutzbar
- optionales Daten-EEPROM (24C64) für Datenlogger etc
- I2C-Anschluß für bis zu 8 LM75 Temperatursensoren oder andere I2C Devices
- Up/Download von Programmen über die serielle Schnittstelle
- Listingdruck über die Druckerschnittstelle
- automatischer Start des Programmes nach Einschalten über Jumper einstellbar
- Tastenkombinationen für Abbruch, Neustart, Screenshot
- Monitor mit Variablen- und Stackanzeige, Einzelschrittbetrieb
- Universelle I2C-Ansteuerung
- Einfaches Dateisystem auf ATMEL Dataflash
- Clone-Funktion zum Kopieren der Software auf weitere Controller

1.4 Hard- und Softwarekonzept

Kernstück des Ganzen ist ein ATmega644, die restliche Hardware besteht im Wesentlichen aus passiven Bauelementen und Steckverbindern. Timer 1 ist für das Videotiming zuständig. Kanal A arbeitet als PWM und erzeugt den Zeilen-Synchronimpuls. Im CSYNC-Mode wird während des vertikalen Synchronimpulses die PWM so umkonfiguriert, dass der Ausgang invertiert wird.

Kanal B von Timer 1 erzeugt den (Horizontal-) Videointerrupt. Dieser enthält eine Synchronisation auf den Timer, dadurch erzeugen auch EEPROM-Lesezugriffe mit 5 Takten CPU-Stop keine Bildstörungen. Timer 2 erzeugt PWM mit dem Audiosignal. Dieses wird entweder mit dem (Pseudo-)Zufallsgenerator oder per DDS mit Wavetable und Hüllkurve erzeugt.

Entgegen den meisten anderen Lösungen ist die PS2-Tastatur und den USART des ATmega angeschlossen. Eine Nutzung externer Interrupts verbietet sich schon wegen des Videotimings, man kann zwar das Clock-Signal während der Bilddarstellung auf LOW ziehen und damit die Tastatur zum Warten verdonnern, aber nicht jede Tastatur schafft es, ein Datenpaket während der Austastlücke zu senden und bei NTSC klappt das Ganze dann aber gar nicht mehr, weil dort die Austastlücke noch kürzer ist. Die zweite Möglichkeit wäre die SPI-Schnittstelle, aber die ist dann auch belegt. Und gerade während der Entwicklungsphase ist es lästig, wenn man ständig die Tastatur abziehen und anstecken muss. Die Nutzung des USART hat hingegen den grossen Vorteil, dass alles automatisch geht und man nur noch das fertige Zeichen abholen muss.

Wobei sich natürlich dann die Frage stellt „und was ist mit seriell?“ Mit reduzierter Bitrate (1200Baud) lässt sich die serielle Schnittstelle im Horizontal-Interrupt mit zwei Portpins realisieren. Timerausgang von Timer 0 treibt eine Ladungspumpe, die ca. -4,7V für die serielle Schnittstelle bereitstellt, dadurch sind keine Spezialbausteine (MAX232 etc.) notwendig. Um Strom zu sparen, lässt sich diese Funktion auch per Software abschalten.

Die parallele Schnittstelle wird über Port A realisiert (+2 Steuerleitungen aus Port B). Wenn keine Druckerschnittstelle benötigt wird, können die 8 Pins als Ein-/ oder Ausgang konfiguriert oder auch als Analogeingang genutzt werden.

1.4.1 Systemvoraussetzung Host für das Assemblieren

Da fast alle I/O-Funktionen von verschiedenen Bibliotheken bereitgestellt werden, sind zum Assemblieren `avr_libmake` (läuft nur unter Linux und organisiert die Makros für die Bibliotheksfunktionen, es geht aber auch ohne) und der AVRA-Assembler notwendig. Das Hex-File sollte sich auch unter anderen Betriebssystemen brennen lassen.

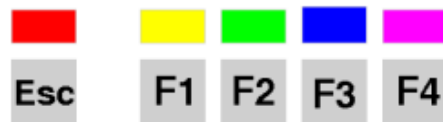
2 Bedienung

AVR-ChipBasic2 ist kurz nach dem Anlegen der Versorgungsspannung einsatzbereit. Wenn nicht der Run-Jumper gesetzt ist, wird das Menu dargestellt.

2.1 Die Tastatur

Die vorliegende Version benötigt eine deutsche oder US-Tastatur (umschaltbar). Bei Bedarf lässt sich das Programm auch an andere Tastenbelegungen anpassen, da noch eine dritte Tastenbelegung wählbar ist.

Zur einfacheren Bedienung gibt es am unteren Bildrand 5 farbige Felder mit Funktionen, die den Tasten **ESC** sowie **F1...F4** entsprechen. Mit einem Aufkleber etc. auf der Tastatur lassen sich so die wichtigsten Tasten schnell finden.

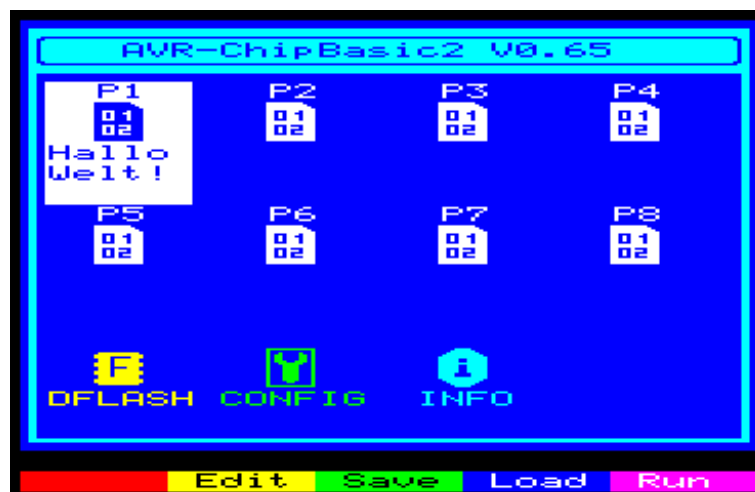


Daneben gibt es noch ein paar Tastenkombinationen, die zum Teil auch vom PC her bekannt sein dürften:

- CTRL+ALT+DEL startet komplett neu, dabei werden alle Variablen gelöscht
- CTRL+C unterbricht das Programm zum nächstmöglichen Zeitpunkt
- CTRL+P sendet einen Screenshot an die serielle Schnittstelle
- F12 unterbricht das Programm und ruft den Monitor auf, funktioniert NICHT in den Grafikmodi

2.2 Das Hauptmenü

Das Hauptmenü wird nach jedem Reset angezeigt, ausser der RUN-Jumper ist gesetzt und das Programm startet automatisch.



Gegenüber den bisherigen Versionen hat sich das Erscheinungsbild recht radikal geändert. Möglich wurde das durch die bessere Farbauflösung des Mega644. In einem „Fenster“ sind die 8 Programme sowie 3 zusätzliche Funktionen als Symbole angeordnet, darunter die Bezeichnung bzw. der Programmname. Mit den Cursortasten kann die Funktion des invertiert dargestellten Feldes aktiviert werden. Ist das Feld ein Programm, so wird dieses gestartet.

In der untersten Zeile stehen die möglichen Kommandos, die Farben entsprechen den Tasten **ESC** und **F1...F4** und haben folgende Funktionen:

Esc		Taste hat keine Funktion
F1	EDIT	Bei Programmen wird der Editor aufgerufen
F2	SAVE	Speichert das Programm auf einem Dataflash-Modul
F3	LOAD	Lädt das Programm von einem Dataflash-Modul
F4	RUN	Startet das Programm

2.2.1 Programm speichern (F2)

Diese Funktion bezieht sich auf das gerade ausgewählte Programm. Wird sie aufgerufen und ist kein formatierter Dataflash angeschlossen, erscheint folgende Fehlermeldung:



Wenn ein formatierter Dataflash angeschlossen ist, kann der Speicherplatz mit der Dateiauswahlbox bestimmt werden:



Dabei ist zu beachten, dass nur auf leere Plätze oder auf BAS Files geschrieben werden kann. Mit der Taste **ESC** kann die Auswahl auch abgebrochen werden. Bevor ein File überschrieben wird, muß dies nochmals mit „Y“ bestätigt werden.

2.2.2 Programm laden (F3)

Diese Funktion bezieht sich auf das gerade ausgewählte Programm. Wird sie aufgerufen und ist kein formatierter Dataflash angeschlossen, erscheint die oben gezeigte Fehlermeldung. Wenn ein formatierter Dataflash angeschlossen ist, kann das Programm mit der Dateiauswahlbox bestimmt werden:



Dabei ist zu beachten, dass nur BAS Files gelesen werden können. Das gewählte File wird ohne nochmalige Nachfrage in den Flash des Controllers kopiert. Mit der Taste **ESC** kann die Auswahl auch abgebrochen werden. Mit den 3 Symbolen in der dritten Zeile können der Flashmanager, die Config-Page und ein Programminfo angezeigt werden. Näheres dazu in den entsprechenden Abschnitten.

2.3 Der Flashmanager

Der Flash-Manager (Symbol **DFLASH** im Hauptmenü) funktioniert natürlich nur dann, wenn auch ein Dataflash-Modul angesteckt ist. Ansonsten gibt es nur die Möglichkeit zum Rescan und Abbruch.



Wurde ein Dataflash-Modul (es funktionieren nur 4 und 8 MBit Bausteine der serie AT45) erkannt, wird auf gültige Formatierung getestet. Formatieren?? Ja das gibt es hier auch, neben dem Blockstatus werden auch solche Dinge wie Schreibzähler angelegt. In der untersten Zeile stehen die möglichen Kommandos, die Farben entsprechen den Tasten **ESC** und **F1...F4** und haben folgende Funktionen:



Esc	EXIT	Rückkehr zum Hauptmenü
F1	INFO	Zeigt den Belegungszustand des Filesystems an
F2	BACKUP	Speichert alle Programme auf das Dataflash-Modul
F3	LOAD	Lädt alle Programm vom Dataflash-Modul
F4	RESCAN	Testet den Dataflash erneut (z.B. nach Wechsel)

2.3.1 Info (F1)

Die freien Dateien/Blöcke werden gezählt und in einer Infobox angezeigt.



2.3.2 Backup (F2)

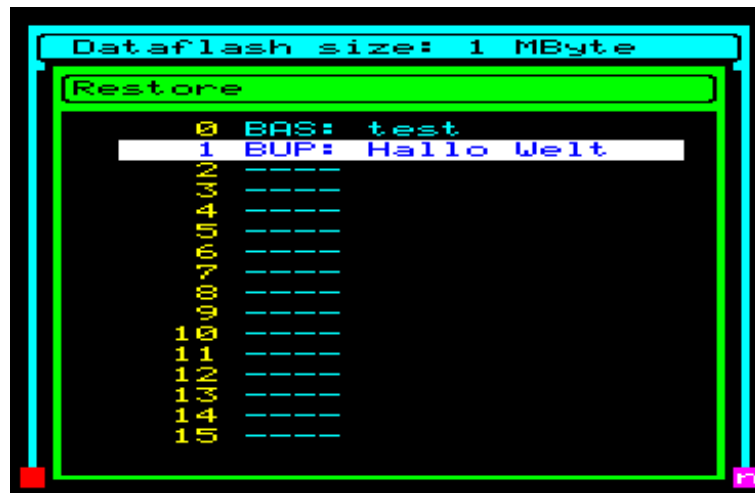
Wenn ein formatierter Dataflash angeschlossen ist, kann der Speicherplatz mit der Dateiauswahlbox bestimmt werden:



Dabei ist zu beachten, dass nur auf leere Plätze oder auf BUP Files geschrieben werden kann. Mit der Taste **ESC** kann die Auswahl auch abgebrochen werden. Bevor ein File überschrieben wird, muß dies nochmals mit „Y“ bestätigt werden. Beim Backup werden alle Programme in eine Datei geschrieben. In der Fileselect-Box wird bei BUP-dateien immer der Programmname des ersten Programms angezeigt.

2.3.3 Restore (F3)

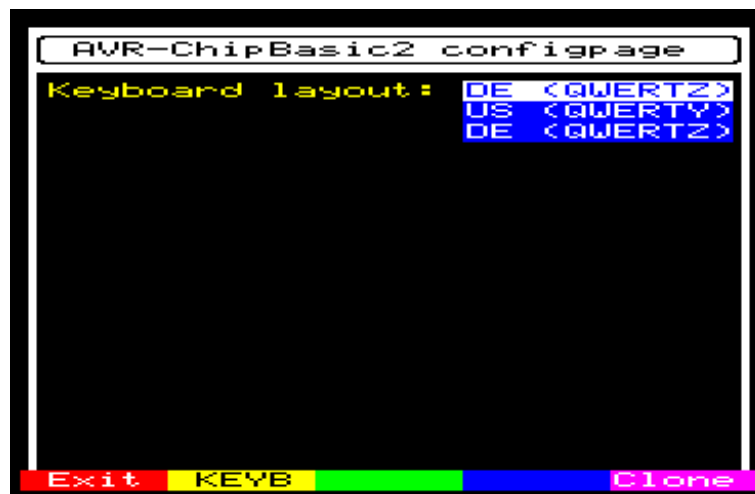
Wird sie aufgerufen und ist kein formatierter Dataflash angeschlossen, erscheint die oben gezeigte Fehlermeldung. Wenn ein formatierter Dataflash angeschlossen ist, kann das zu lesende Backup mit der Dateiauswahlbox bestimmt werden:



Dabei ist zu beachten, dass nur BUP Files gelesen werden können. Das gewählte File wird ohne nochmalige Nachfrage in den Flash des Controllers kopiert. Mit der Taste **ESC** kann die Auswahl auch abgebrochen werden.

2.4 Die Konfigurationspage

Auf der Config-Page können verschiedene Einstellungen vorgenommen werden. Das Hauptmenü wird nach jedem Reset angezeigt, ausser der RUN-Jumper ist gesetzt und das Programm startet automatisch.



In der untersten Zeile stehen die möglichen Kommandos, die Farben entsprechen wieder den Tasten **ESC** und **F1...F4** und haben folgende Funktionen:

Esc	EXIT	Config-Page wieder verlassen
F1	KEYB	Tastaturbelegung ändern
F2		Taste hat keine Funktion
F3		Taste hat keine Funktion
F4	CLONE	Erstellt eine Flash-Kopie des Controllers

Im unteren Bereich des Bildschirms ist eine 8-stellige Checksumme zu sehen. Diese steht auch in der LIESMICH Datei und sollte natürlich (insbesondere nach dem Clonen) übereinstimmen.

2.4.1 Tastaturbelegung ändern (F1)

AVR-ChipBasic2 unterstützt verschiedene Tastaturlayouts. Zur Zeit sind nur DE (QWERTZ) und US (QWERTY) realisiert, ein drittes Layout kann z.B. eine länderspezifische Tastatur sein. Bei jedem Tastendruck von F1 wird die Tastaturbelegung um weitergeschaltet. Die gewählte Einstellung wird im internen EEPROM gesichert und bleibt auch bei Abschalten der Spannungsversorgung erhalten.

2.4.2 System clonen (F4)

Mittels dieser Funktion lässt sich eine identische Kopie des Flash-Inhalts vom Mikrocontroller erstellen. Der Target-Mikrocontroller muss auch ein Mega644 sein und einen Quarz oder externen Oszillator angeschlossen haben. Denn neben dem Flash werden auch die Fusebits programmiert. Eventuell vorhandene BASIC-Programme werden natürlich auch mit übertragen. Damit ist es möglich, einen weiteren Computer aufzubauen ohne einen PC zur Verfügung zu haben / zu benutzen.

Nach Anwahl der Funktion folgt noch eine Sicherheitsabfrage, danach wird der Kopiervorgang gestartet. Um das System zu clonen ist ein Kabel mit einer speziellen Belegung notwendig, mit welchem dann die SPI/ISP-Schnittstellen beider Systeme verbunden werden.

Host Signal	Host PIN	Target Signal	Target PIN
GND	4,6,8,10	GND	4,6,8,10
VCC	2	VCC	2
MOSI	1	MOSI	1
MISO	9	MISO	9
SCK	7	SCK	7
SS	3	RESET	5

2.5 Der Editor

```

PROGRAM 1:Hallo Welt!
OK
01 ? "Hallo Welt!"
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
EXIT NAME SAVE RELOAD RUN

```

Die Programme werden mit einem einfachen Fullscreen-Editor geschrieben. Ganz oben stehen Programmnummer (1-8) und Programmname. Sobald der Text verändert wird, erscheint in der oberen linken Ecke ein Stern. Darunter befindet sich eine farbige Statuszeile, in der z.B. Fehler angezeigt werden. Die Position des Cursors wird durch ein invers blinkendes Zeichen dargestellt. Es wird immer eingefügt, Zeichen unter/rechts vom Cursor wandern nach rechts. Die Cursor-Tasten bewegen den Cursor im Textfeld, die Page-tasten bewegen den Cursor um 8 Zeilen nach oben/unten. Bei Bedarf wird der Text gescrollt. In der untersten Zeile sind 5 Farbfelder zu sehen. Diese entsprechen den Tasten **ESC** und **F1...F4** und haben folgende Funktionen:

Esc	EXIT	Abbruch, ohne zu speichern
F1	NAME	Änderung des Programmnamens
F2	SAVE	Speichert das Programm im Flash
F3	LOAD	Lädt das Programm aus dem Flash
F4	RUN	Startet das Programm

Die Tasten F1 bis F4 haben eine Zweitfunktion, wenn zusätzlich die **CTRL** oder **STRG** Taste gedrückt wird. Der Text in den 5 Farbfeldern ändert sich entsprechend:

Esc		Taste ist nicht belegt
F1	PRINT	Ausdruck des Listings über die Druckerschnittstelle
F2	SEND	Ausgabe des Programms über die serielle Schnittstelle
F3	RECV	Empfang eines Programms über die serielle Schnittstelle
F4	NEW	Löscht das Programm (alle Zeilen leer)

Bei den meisten Funktionen erscheint in der Statuszeile eine Sicherheitsabfrage, die entweder mit (Y)es oder (N)o beantwortet werden muss.

Ausserdem haben die folgenden Tasten / Tastenkombinationen spezielle Funktionen:

- Mit der Taste **POS1** wird der Cursor auf das erste Zeichen der aktuellen Zeile gesetzt.
- Mit der Taste **ENDE** wird der Cursor auf das letzte Zeichen der aktuellen Zeile gesetzt.
- Mit der Taste **DEL** wird das Zeichen unter dem Cursor gelöscht, von rechts rücken Zeichen nach.
- Mit der **Backspace**-Taste wird das Zeichen links neben dem Cursor gelöscht, von rechts rücken Zeichen nach.
- Mit der **ENTER** Taste wird an den Anfang der nächsten Zeile gesprungen
- Mit der Tastenkombination **ALT+INS** wird an der aktuellen Cursorposition eine Zeile eingefügt. Alle Zeilen ab der Cursorposition wandern nach unten, die bisherige letzte Zeile geht dabei verloren.
- Mit der Tastenkombination **ALT+DEL** wird die Zeile der aktuellen Cursorposition gelöscht. Alle Zeilen unterhalb der Cursorposition wandern nach oben, unten wird eine Leerzeile angefügt.
- Mit der Tastenkombination **CTRL+C** wird die aktuelle Zeile in den Puffer kopiert. Angezeigt wird das durch ein # und die Zeilennummer in der rechten oberen Bildschirmecke.
- Mit der Tastenkombination **CTRL+X** wird die aktuelle Zeile in den Puffer verschoben. Angezeigt wird das durch ein # und die Zeilennummer in der rechten oberen Bildschirmecke.
- Mit der Tastenkombination **CTRL+V** wird der Puffer in die aktuelle Zeile kopiert. Die Puffer-Anzeige rechts oben verschwindet wieder.

2.6 Serieller Programmtransfer

2.6.1 Kommunikationsparameter

Für die Funktionen der seriellen Schnittstelle sollte am PC folgendes eingestellt werden

8 Bit — 1200 Baud — no Parity — 1 Stopp-Bit

Für alle Funktionen wird normale Textübertragung verwendet. Unter Windows kann man z.B. das Hyperterminal verwenden (Text senden/aufzeichnen) verwenden, unter Linux minicom oder das Program **chiptrans.pl** im Examples-Ordner. Das Programmende wird mit einem „#“ als erstes Zeichen gekennzeichnet.

2.6.2 Senden zum AVR

1. **RECV** (CTRL+F3) im Editor auswählen und mit „Y“ bestätigen
2. Transfer am PC mit **./chiptrans.pl -w Dateiname** starten
3. am AVR wird das Programm, am PC der Status angezeigt

2.6.3 Empfangen vom AVR

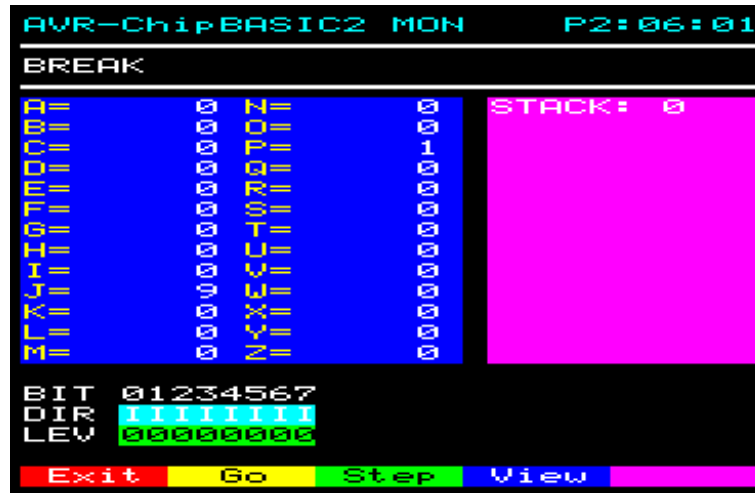
1. Transfer am PC mit **./chiptrans.pl -r Dateiname** starten
2. **SEND**(CTRL+F2) im EDITOR auswählen und mit „Y“ bestätigen
3. am PC wird der Status angezeigt

2.6.4 Nutzung der Screenshot-Funktion

Für den Screenshot kann das Programm **screenshot.pl** im libmio/tools-Verzeichnis benutzt werden. Es benötigt ImageMagick und wird mit „./screenshot.pl dateiname.endung“ aufgerufen. Danach kann dann der Screenshot vom Controllerboard mittels **CTRL+P** ausgelöst werden.

2.7 Der integrierte Monitor/Debugger

ChipBasic2 enthält auch einen kleinen Monitor zum debuggen. Angerufen wird er mit der Taste **F12** während des Programmlaufes oder mit einer **BREAK**-Anweisung im Programm.



Oben rechts steht durch Doppelpunkte getrennt das Programm, die Programmzeile und das gerade abgearbeitete Statement. Darunter die gerade aktive Programmzeile im Quelltext, da nur 30 Zeichen je Zeile dargestellt werden können eventuell die letzten 2 Zeichen der Quelltextzeile fehlen.

Danach folgt im linken Teil des Bildschirms die Anzeige aller Variablenwerte und rechts daneben der Stack. Jede FOR-Schleife und jedes CALL/XCALL erzeugt einen Eintrag auf dem Stack. Unter der Variablenliste wird der Status der parallelen Schnittstelle angezeigt. Zuerst die Pin-Nummern (1 entspricht D0), darunter die Datenrichtung (I oder O) und ganz unten der aktuelle Pegel.

In der untersten Zeile stehen die möglichen Kommandos, die Farben entsprechen wieder den Tasten **ESC** und **F1...F4** und haben folgende Funktionen:

Esc	EXIT	Beendet den Monitor und löst ein Break aus (beendet das Programm)
F1	GO	Beendet den Monitor und setzt die Programmausführung fort
F2	STEP	Führt das nächste Statement aus und kehrt danach zum Monitor zurück
F3	VIEW	Zeigt den aktuellen Ausgabebildschirm an, jeder weitere Tastendruck wechselt zur Monitoranzeige zurück
F4		Taste ist nicht belegt

3 Changelog

14.01.2008 Erste öffentliche Mega644-Version (0.65)

- ein paar Funktionen sind noch nicht vollständig getestet

- Beispielprogramme fehlen noch

31.01.2008 Zweite öffentliche Mega644-Version (0.69)

- ein paar Bugfixes
- Neue Sprite-Routinen
- im Hintergrund arbeitender Noten-Sequenzer