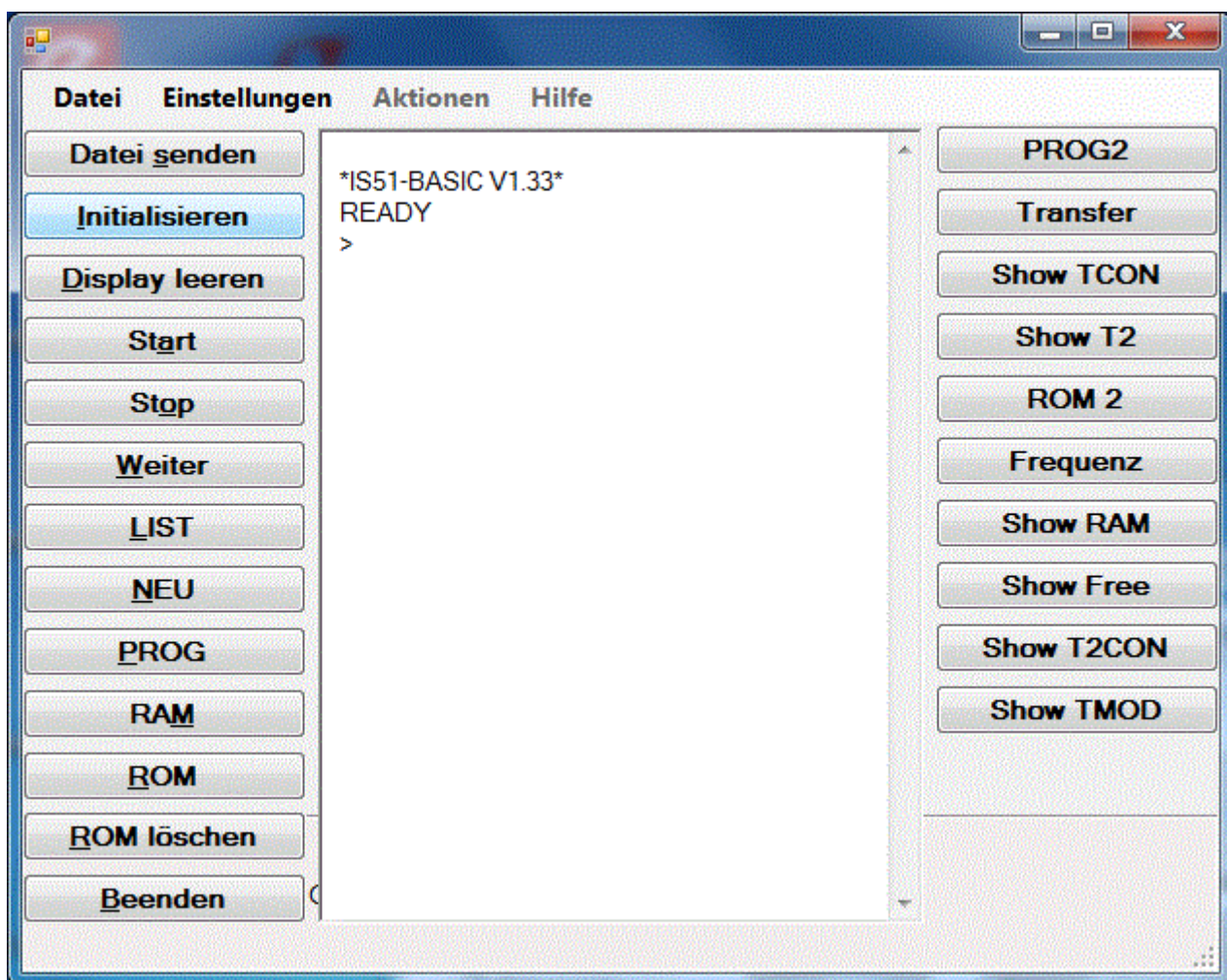


# Handbuch

## zum

# INTEL-MCS51

## Basic-Terminal





## Inhaltsverzeichnis

1 Was ist das Basic-Terminal.....	4
2 Terminal starten.....	5
3 Basic initialisieren.....	7
4 Basic-Datei laden.....	8
5 Datei zum Basic senden.....	9
6 Programm starten.....	11
7 Programm anhalten.....	12
8 Programm fortsetzen.....	13
9 Programm löschen.....	14
10 Programm dauerhaft speichern.....	15
11 Programmspeicher löschen.....	16
12 Eingaben durch Tastatur.....	17
13 UserButton definieren.....	18
14 UserButton benutzen.....	19
15 Anzeige löschen.....	21
15 Terminal beenden.....	21
16 Beispiel.....	22
16.1 Hardware Funktionstest.....	23
16.2 Testprogramm Tastatureingabe.....	23
16.3 Testprogramm laden.....	24
16.4 Programm dauerhaft speichern.....	24
16.5 Autostart einrichten.....	25

## 1 Was ist das Basic-Terminal

Das Basic-Terminal ist ein Programm, mit dem die Funktionen des MCS51-Basic von INTEL, von einem PC unter Windows gesteuert werden können. Mit dem Terminal lassen sich Befehle und Kommandos, sowie komplette Basic-Dateien übertragen.

Es ist an keine bestimmte Zielhardware gebunden, setzt aber voraus, dass der PC über eine serielle RS232 Schnittstelle verfügt und mit einer Basic-Hardware verbunden ist. Dazu genügt auch ein USB-RS232 Adapter. Das Terminal bietet im Auswahldialog nur im PC verbaute Schnittstellen an. Die Baudrate ist im Bereich 600 bis 115200Baud wählbar.

Im Anzeigefenster sind jeweils die, über RS232 empfangenen, Meldungen des Basic, oder eigene Eingaben der Tastatur zu sehen.

Eingaben erfolgen über die Tastatur oder per Maus und Funktionsbuttons. Zur Eingabe von Befehlen wird mit der Maus dem Anzeigefenster der Programmfokus übergeben. Ist dies geschehen, blinkt der Eingabecursor im Fenster. Nun können über die Tastatur Befehle eingegeben werden. Durch betätigen der ENTER-Taste wird ein Befehl abgeschlossen und der eingegebene Text, über die serielle Schnittstelle zur Basic-Hardware übertragen. Durch den Zeileneditor ist es möglich, Programme für das Basic zu schreiben. Für größere Programme ist es aber zweckmäßiger einen komfortablen Editor zu verwenden.

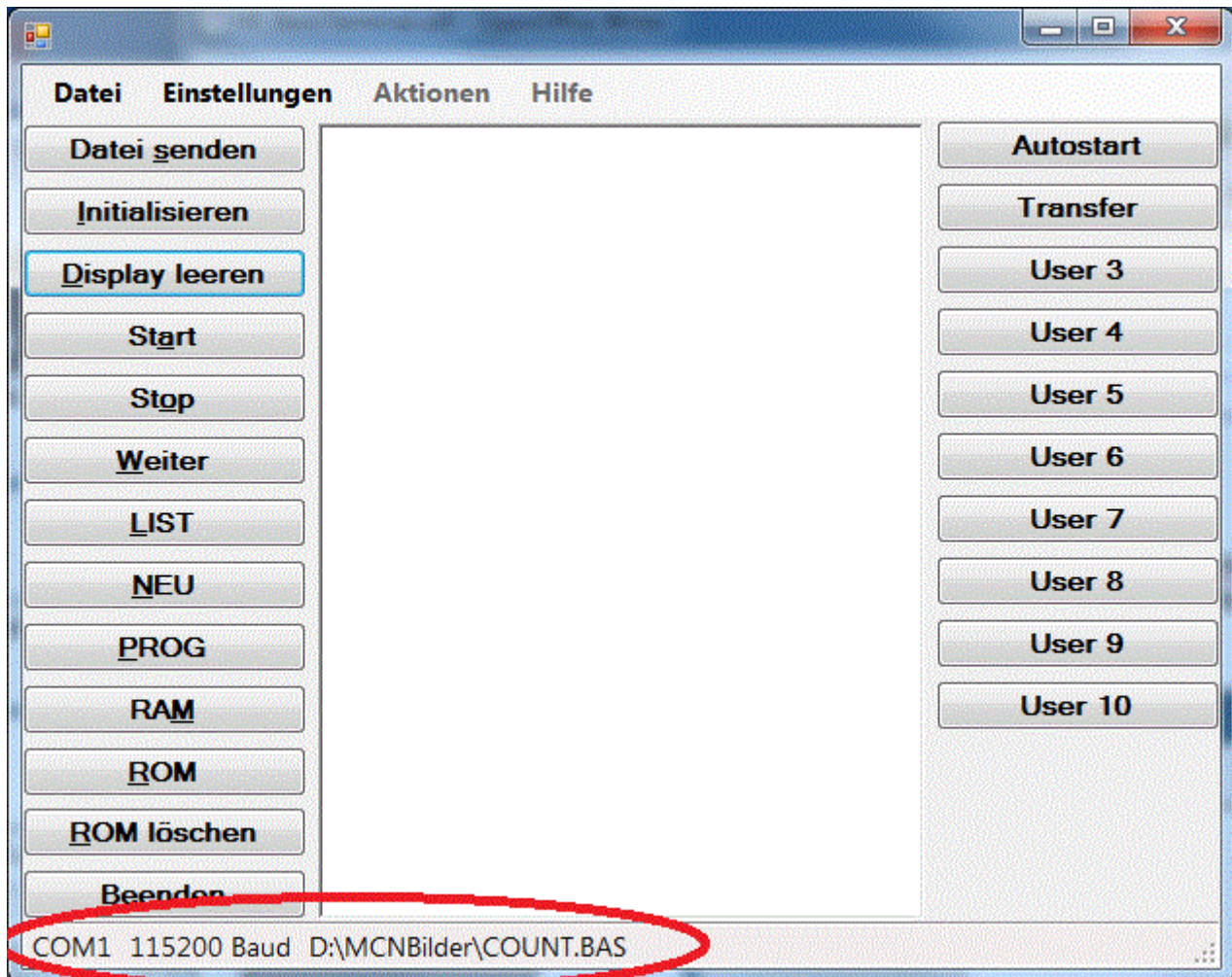
Eine andere Art der Eingabe läuft über die Buttons des Programms, die den schnellen Zugriff auf die Befehle ermöglichen. Es sind bereits vordefinierte Buttons vorhanden, dazu kommen noch frei Definierbare. Auf diese freien Buttons kann der Anwender selbst Funktionen legen.

Beim beenden des Programms werden die Einstellungen gespeichert und beim erneuten starten wird mit den gespeicherten Einstellungen gearbeitet.

Dieses Handbuch dient der Beschreibung der Bedienung und Funktionen des Basic-Terminalprogramms. Eine detaillierte Beschreibung des MCS51-BASIC ist in dessen Handbuch zu finden.

## 2 Terminal starten

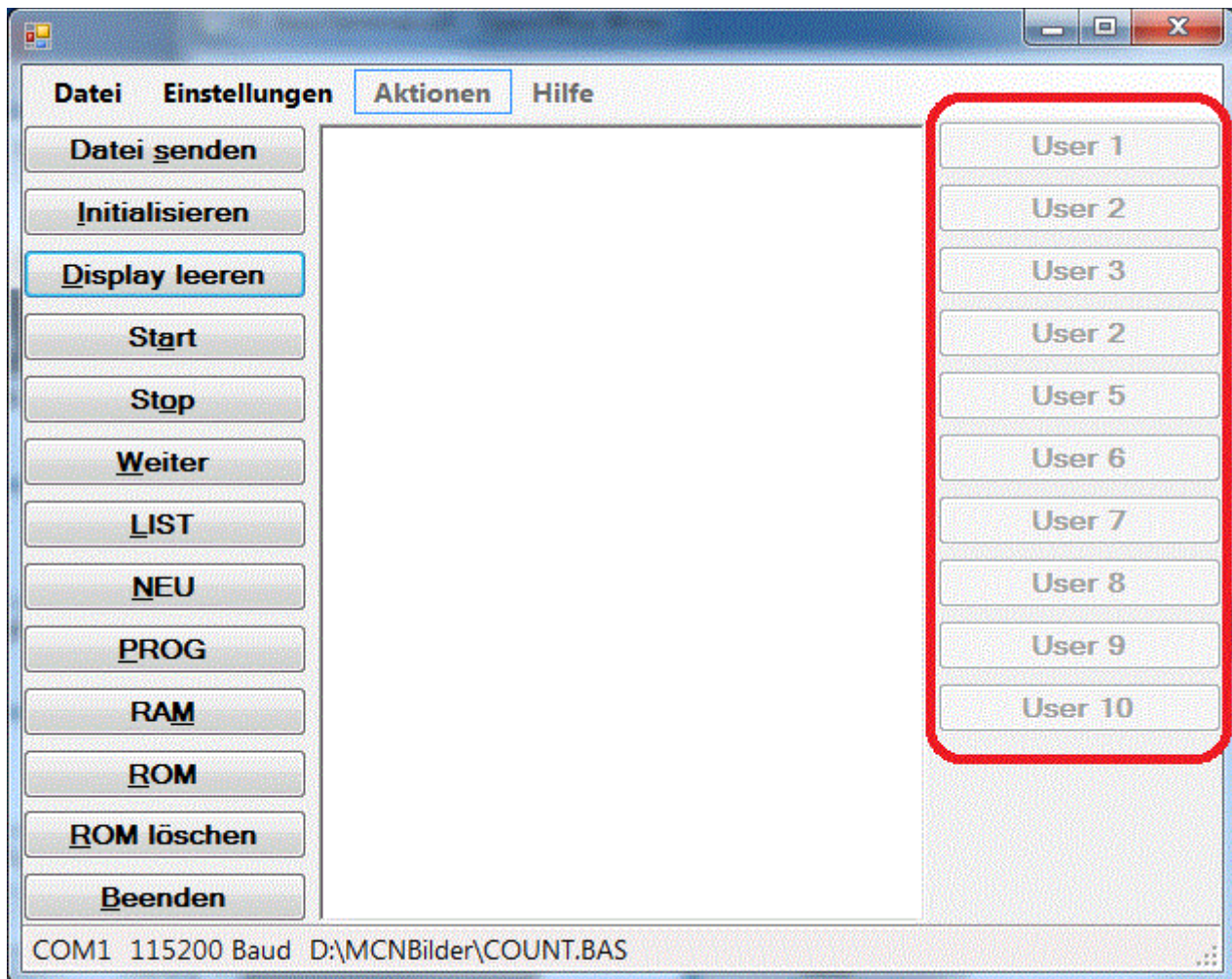
Beim ersten Start des Terminalprogrammes hat dieses noch keine Config-Datei und wird deshalb mit Grundeinstellungen gestartet.



Als Schnittstelle ist COM1 mit einer Übertragungsrate von 115200Baud eingestellt.



Die benutzerdefinierbaren Buttons sind noch deaktiviert und ihnen ist keine Funktion zugeordnet.



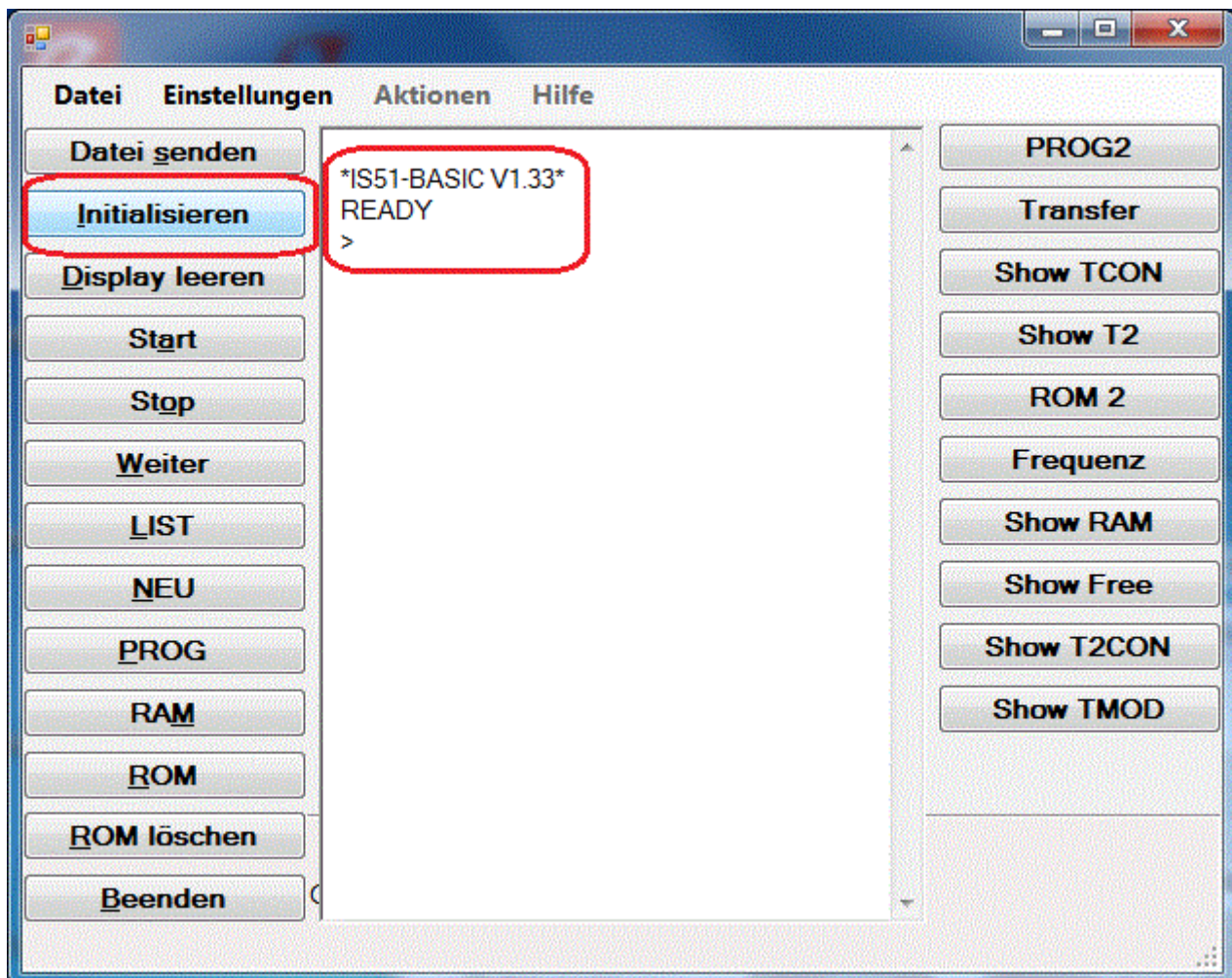
Nach Änderung der Einstellungen und beim beenden wird eine Config - Datei angelegt und das Terminal startet danach stets mit den darin enthaltenen Einstellungen. Auch die zuletzt verwendete Datei wird gespeichert.

### 3 Basic initialisieren

Zu Beginn sind im Menü die gewünschte Schnittstelle (Einstellungen – Schnittstelle) und Baudrate (Einstellungen - Baudrate) zu wählen. Welche Baudrate dies genau ist, spielt keine Rolle, da das Basic sich beim initialisieren auf die Baudrate des PC einstellt. Es sind Übertragungsgeschwindigkeiten von 600 Baud bis 115200 Baud möglich. Zunächst ist eine Baudrate von 19200 angemessen, gibt es damit keine Probleme, darf auch eine höhere Rate gewählt werden.

Um mit dem Basic in Kontakt zu treten ist der PC, über die serielle Schnittstelle mit der Basic-Hardware zu verbinden. Das Basic ist durch einschalten oder Reset in Eingabebereitschaft zu versetzen.

Betätigen des Button „Initialisieren“ sendet den ASCII-Code 20H (Space) zum Basic. Dadurch kann der Interpreter die richtige Baudrate ermitteln und sich darauf einstellen.

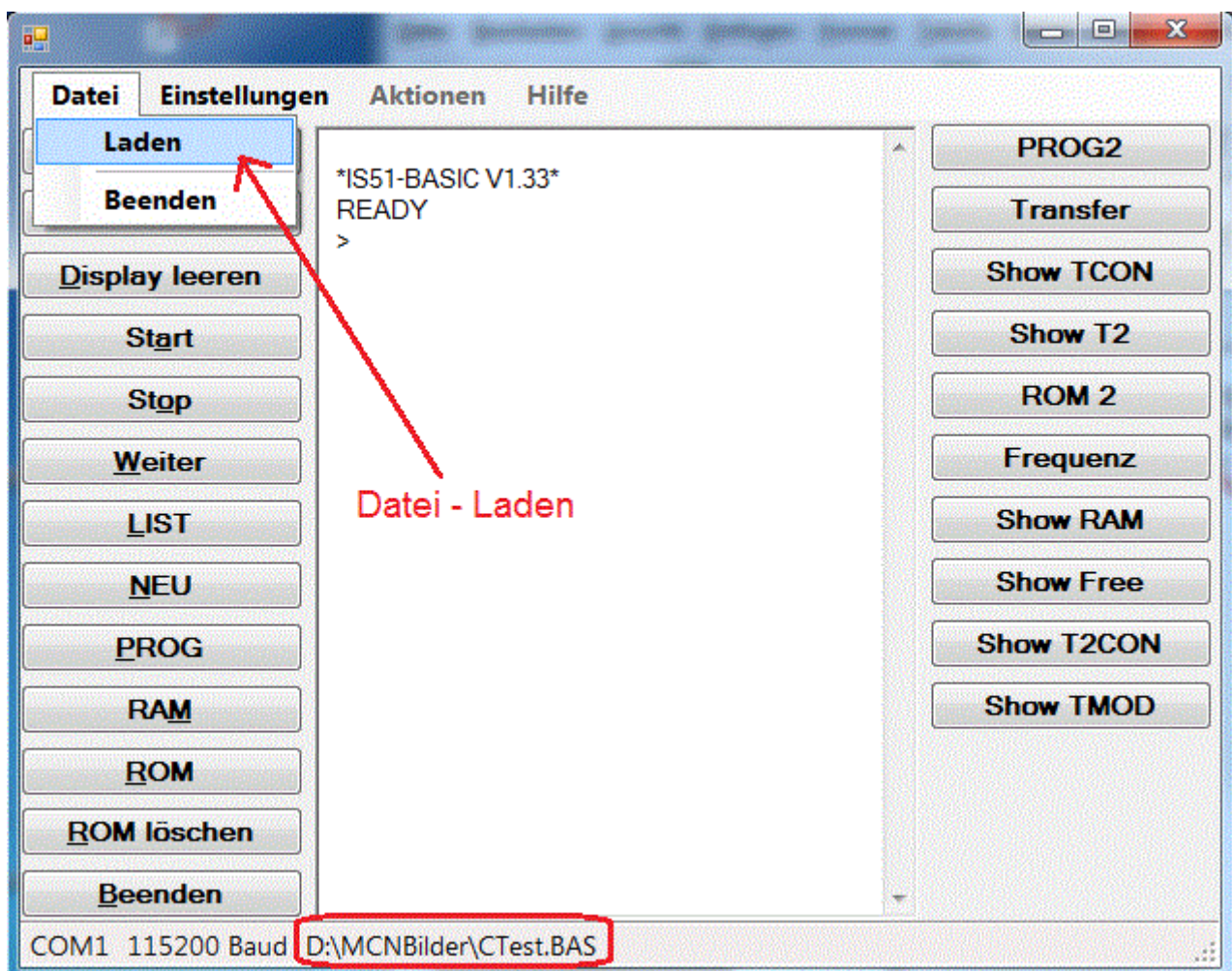


Hat das initialisieren funktioniert, gibt das Basic seine Bereitschaftsmeldung und sein Eingabeprompt „>“ zurück. Jetzt ist das Basic bereit für Kommandos und Befehle.



## 4 Basic-Datei laden

Im Menü kann unter „Datei – Laden“ die Datei mit dem Basic-Programmcode zur Übertragung zum Basic gewählt werden. Diese Datei beinhaltet den Quelltext eines Programms. Die Datei wird nicht in das Terminal geladen, sondern es wird nur der Dateiname mit Pfad gespeichert.



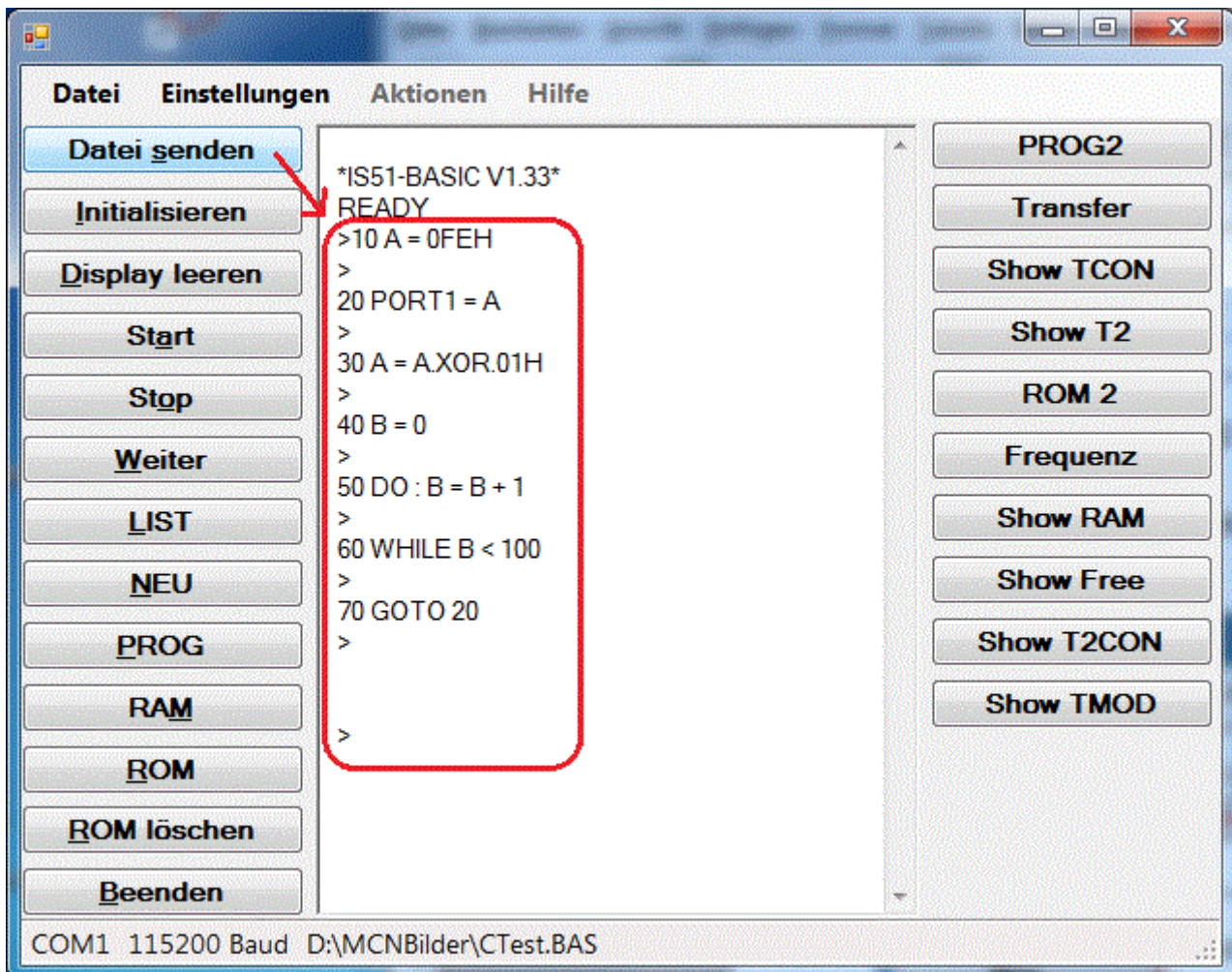
Der Dateiname mit Pfad wird in der Config - Datei gespeichert und bleibt solange erhalten, bis eine neue Datei gewählt ist.

Nachdem eine Datei gewählt ist, ist in der Statuszeile der komplette Pfad zu finden. Jedes mal wenn nun der Button „Datei senden“ betätigt wird, gelangt der Inhalt der gewählten Datei über die serielle Schnittstelle zum Basic. Der Dateiname bleibt solange erhalten, bis eine neue Datei gewählt wird.



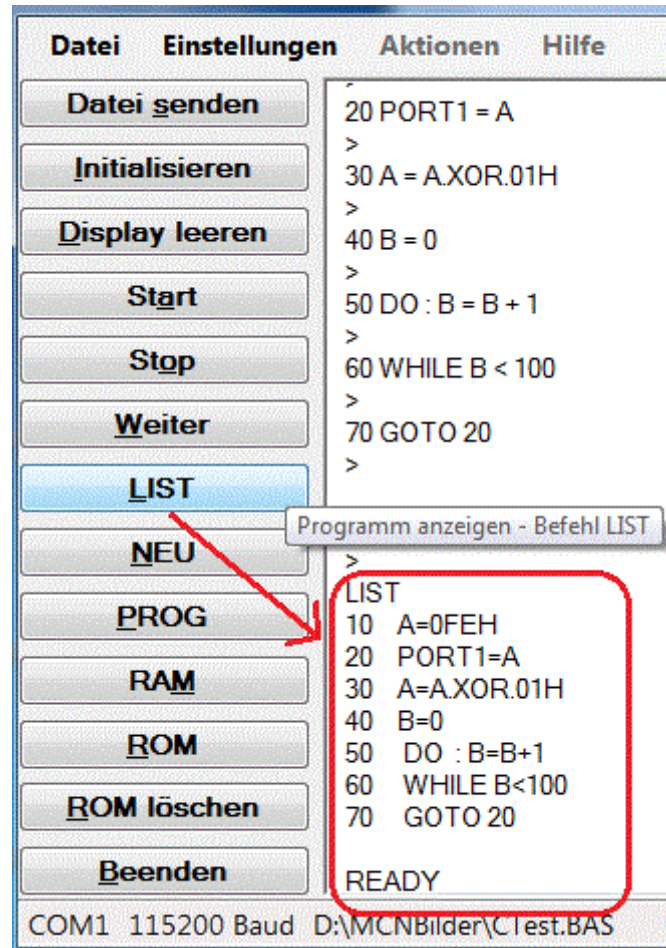
## 5 Datei zum Basic senden

Betätigen des Button „Datei senden“ überträgt die gewählte Datei zum Basic.



Das Basic bestätigt jede Zeile durch ein Echo und das Eingabeprompt „>“. Das Programm wird im Datenspeicher (RAM) der Basic-Hardware abgespeichert. Von dort kann es mit einem der PROG-Befehle in den Programmspeicher kopiert werden.

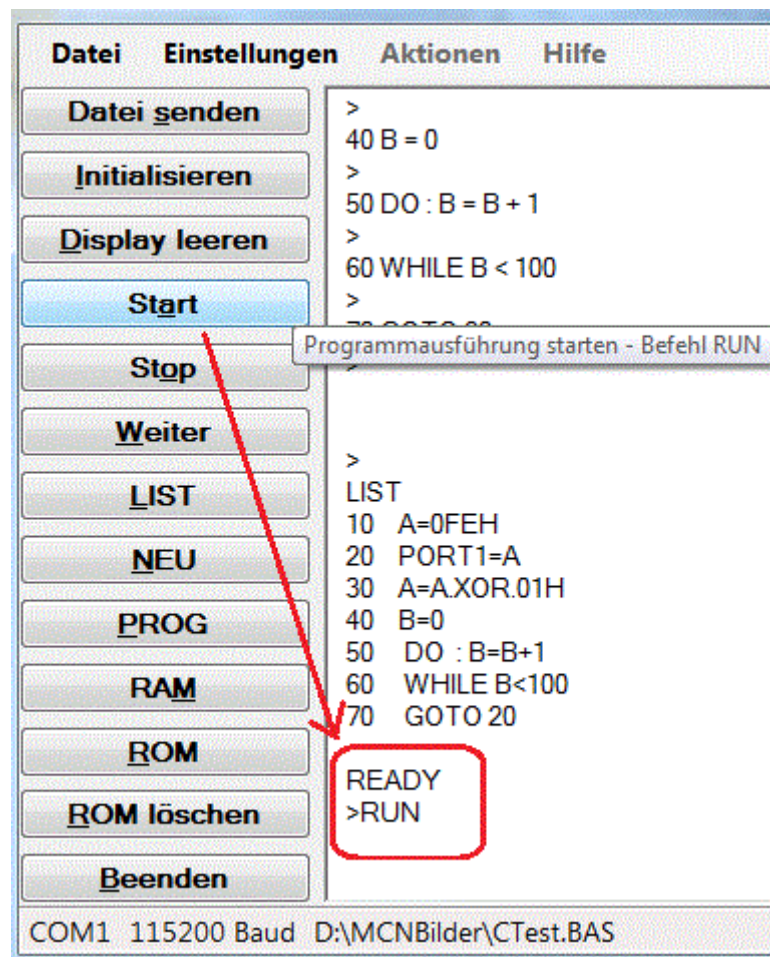
Durch betätigen des „LIST“-Button wird der Befehl „LIST“ zum Basic gesendet und das Basic zeigt sein gespeichertes Programm im Anzeigefenster.



Im Anzeigefenster wird alles vom Basic gesendete und Eingaben der Tastatur angezeigt. Nun kann das Programm überprüft und ggf. geändert werden. Um etwas einzugeben wird mit der Maus auf die letzte Zeile im Anzeigefenster geklickt, wodurch der Cursor dort blinkt. Jetzt können über die Tastatur Eingaben erfolgen, welche durch betätigen der Enter-Taste zum Basic gesendet werden.

## 6 Programm starten

Durch betätigen des „START“-Button wird der Befehl „RUN“ zum Basic gesendet und damit das im Speicher befindliche Programm gestartet.

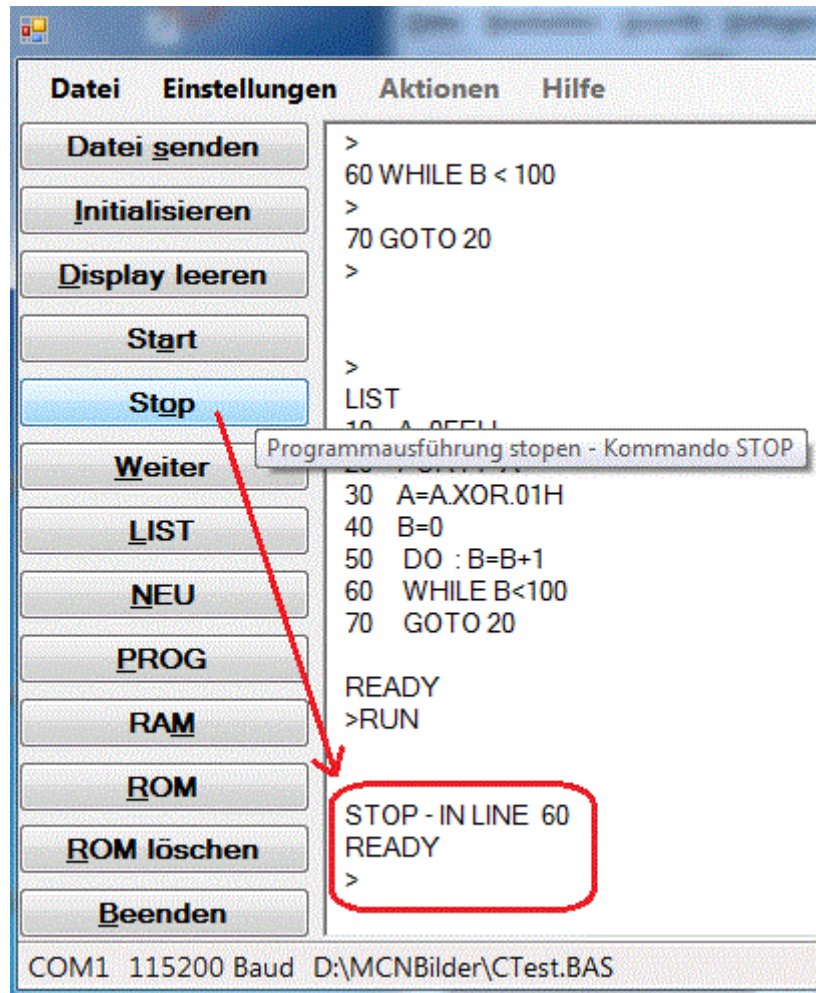


Nun wird das Programm durch den Mikrocontroller ausgeführt, weshalb das Basic nicht mehr mit seinem Prompt antworten kann.



## 7 Programm anhalten

Durch betätigen des „STOP“-Button wird das STOP - Kommando zum Basic gesendet.

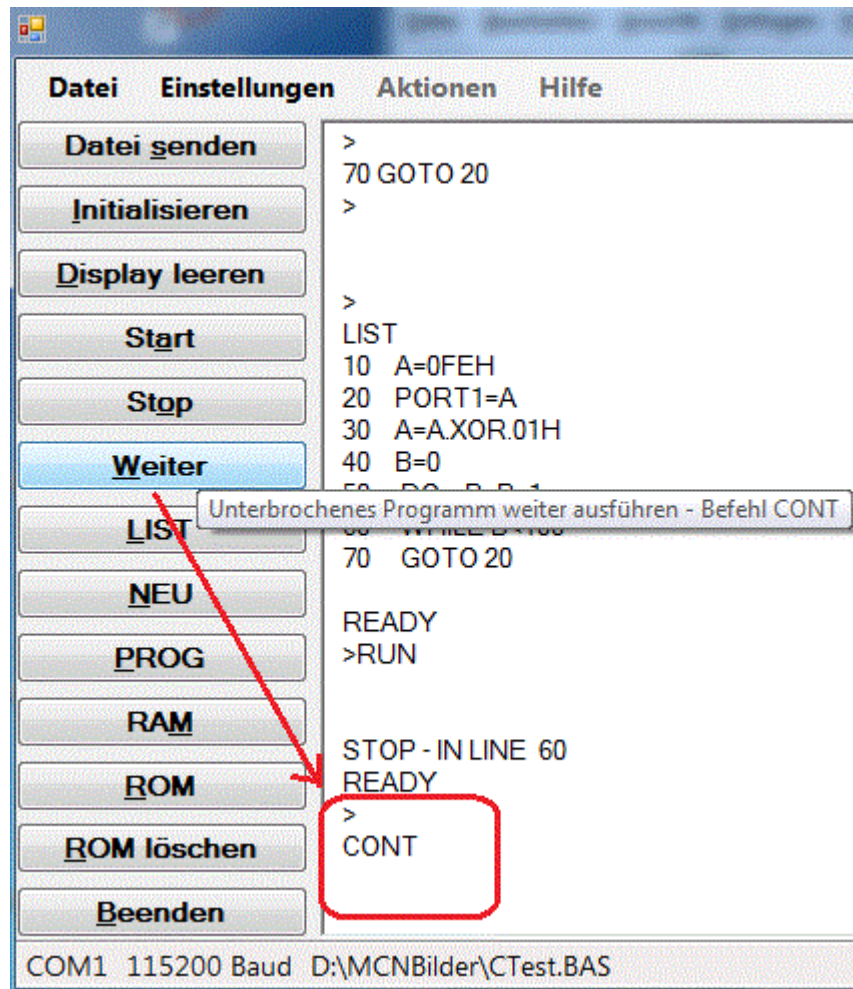


Das Basic quittiert dies mit der Stop-Meldung „STOP – IN LINE xx“. Jetzt ist es wieder möglich, mit dem Basic zu kommunizieren.



## 8 Programm fortsetzen

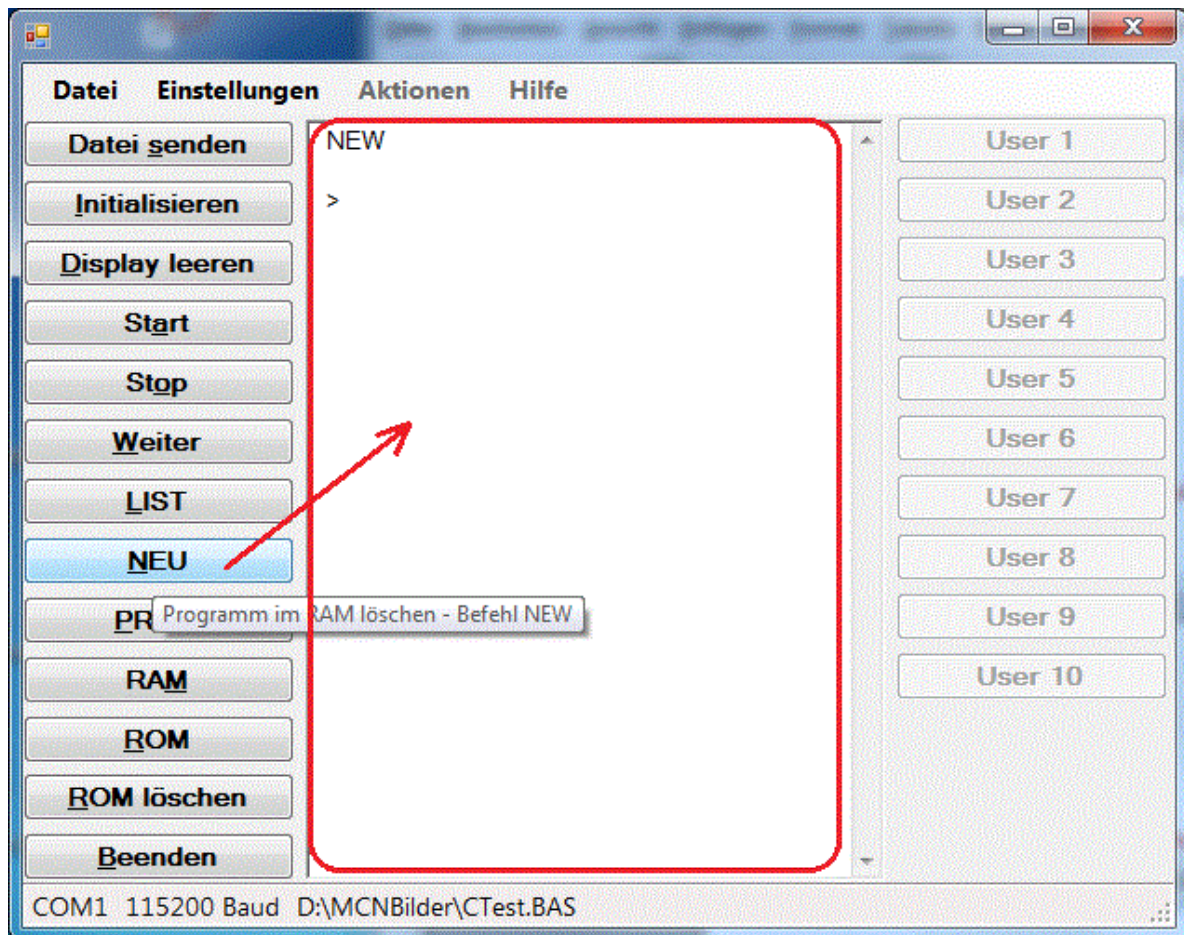
Durch betätigen des „Weiter“-Button wird der CONT-Befehl gesendet und das Programm, an der unterbrochenen Stelle, fortgesetzt.



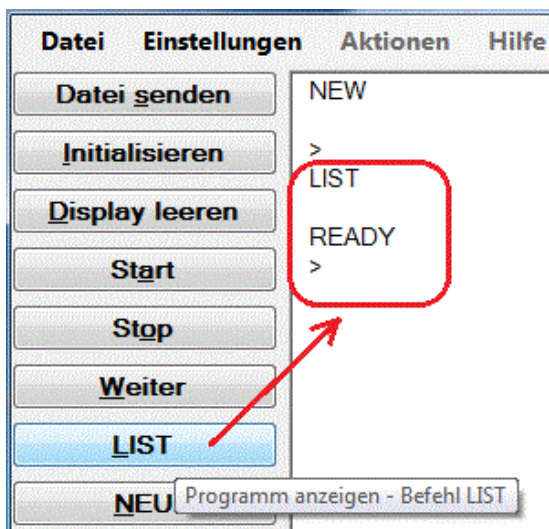
Eine Bestätigung des Basic fehlt, da nun wieder das Anwenderprogramm die Kontrolle über den Mikrocontroller hat.

## 9 Programm löschen

Durch betätigen des „Neu“-Button wird der NEW-Befehl zum Basic gesendet. Dadurch wird das Programm aus dem Datenspeicher gelöscht.



Es wird auch das Anzeigefenster gelöscht, da es jetzt ungültige Daten enthält.



Eine Kontrolle durch „LIST“ zeigt den leeren Programmspeicher.

## 10 Programm dauerhaft speichern

Befindet sich ein Programm im Datenspeicher, so kann es durch betätigen des „PROG“-Button in den Programmspeicher (EEPROM oder RAM mit Batteriepufferung) kopiert werden.

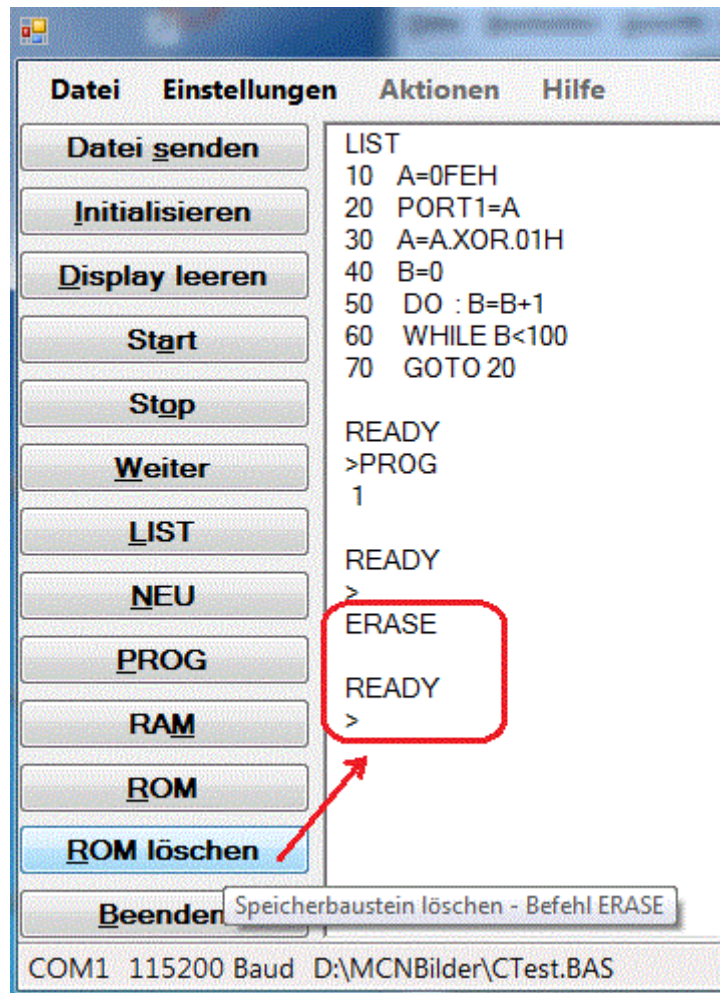


Es können mehrere Programme im Programmspeicher liegen. Beim abspeichern weist PROG jedem Programm eine fortlaufende Nummer zu. Unter dieser Nummer ist das Programm dann erreichbar. Im Beispiel wird das Programm unter der Nummer 1 gespeichert, das nächste Programm dann unter Nummer 2 usw.



## 11 Programmspeicher löschen

Durch betätigen des „ROM löschen“-Button wird der „ERASE“-Befehl gesendet und dadurch der Programmspeicher gelöscht.

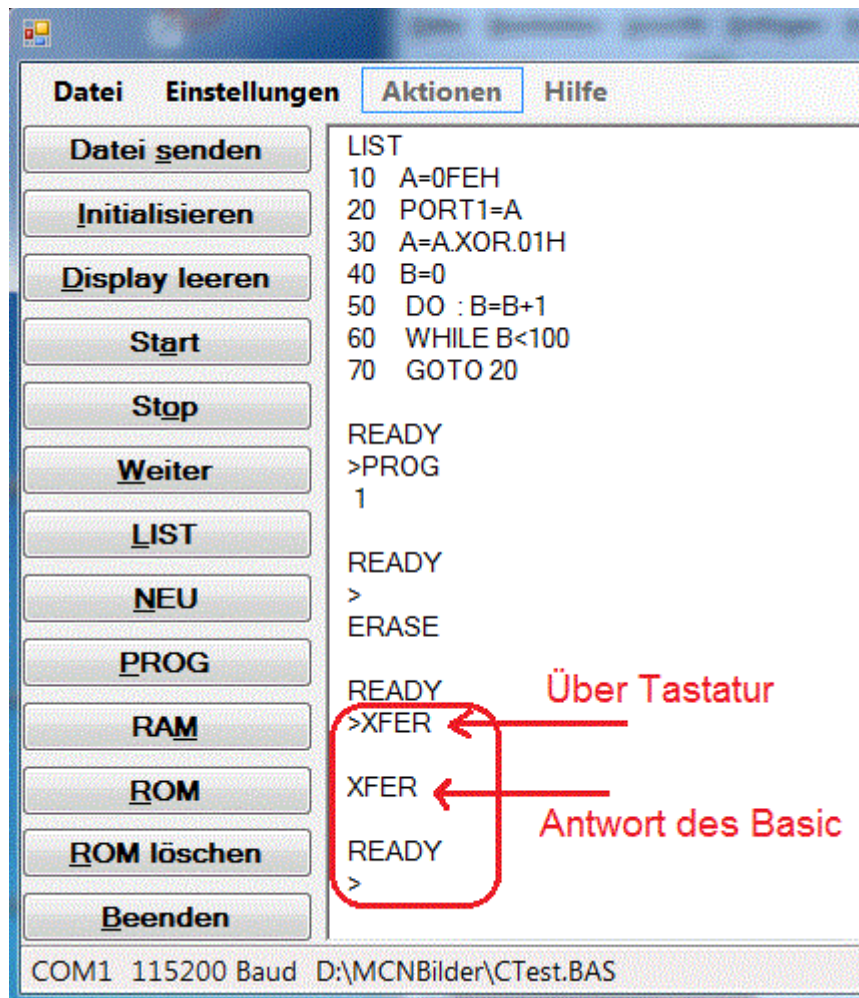


Nach einer Sicherheitsabfrage, ob wirklich gelöscht werden soll, wird der gesamte Programmspeicher und alle darin befindlichen Programme gelöscht. Dabei spielt es keine Rolle, ob EEPROM oder RAM als Programmspeicher verbaut ist.



## 12 Eingaben durch Tastatur

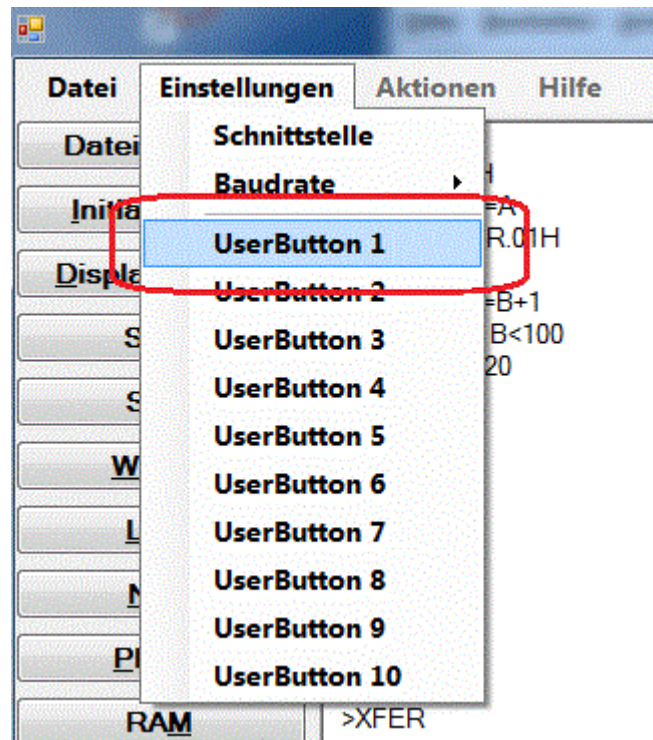
Für das kopieren eines Programms vom Programmspeicher in den Datenspeicher, gibt es keinen Button. Solche Befehle sind über die Tastatur einzugeben.



Dazu wird mit der Maus auf die letzte Zeile im Anzeigefenster geklickt. Am blinkenden Cursor können nun Tastatureingaben erfolgen. Durch betätigen der Eingabetaste (Enter) wird die Eingabe zum Basic gesendet und der Befehl dann ausgeführt.

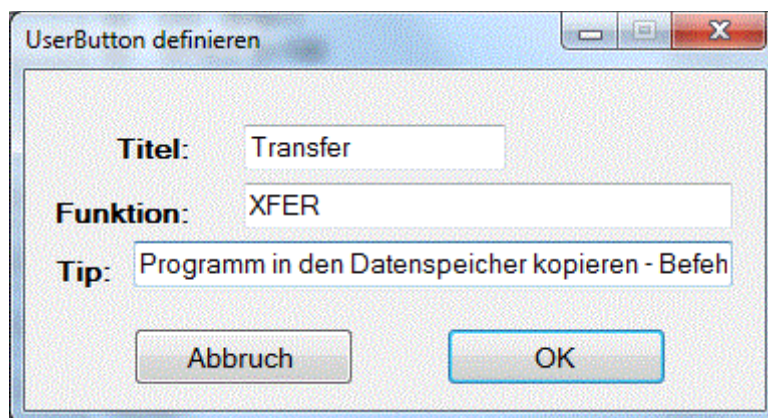
## 13 UserButton definieren

Eine andere Art Befehle einzugeben ist das benutzen der User-Button. Häufig benötigte Befehle sind auf diese Art schnell erreichbar.



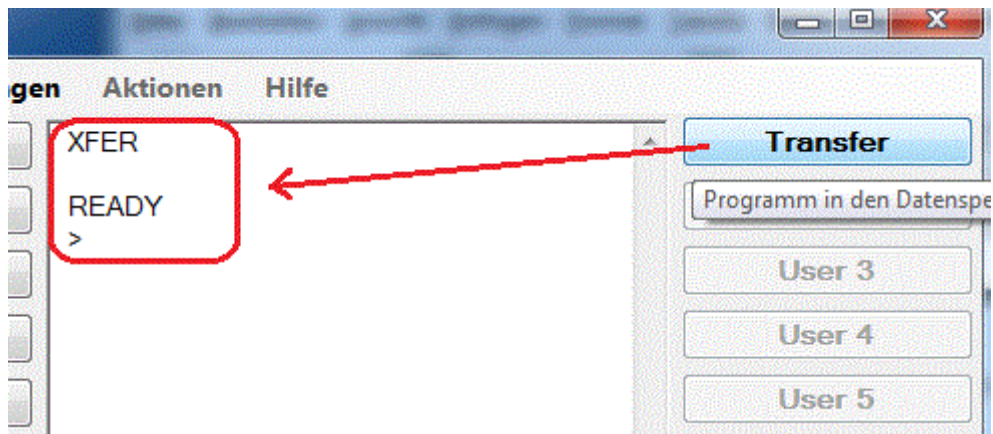
Im Menü unter Einstellungen sind die UserButtons zu definieren. Jeder Button kann, unabhängig von den Anderen festgelegt werden.

Titel ist der Beschriftungstext des Button im Terminal. Funktion ist der Befehl, welcher bei betätigen zum Basic gesendet wird. Tip ist der Hilfstext, welcher bei berühren des Button mit der Maus erscheint.

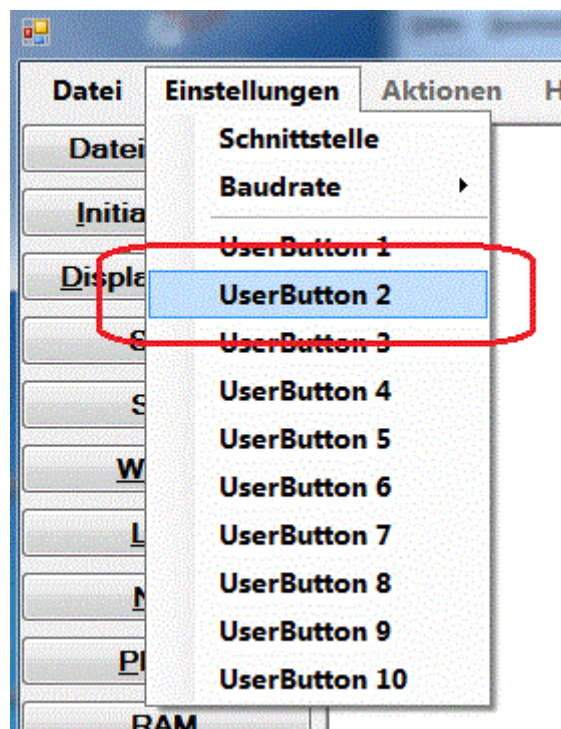


## 14 UserButton benutzen

Der neu definierte Button ist jetzt mit „Transfer“ beschriftet und sendet bei Betätigung den XFER-Befehl zum Basic. Berührt die Maus den Button, erscheint der Hilfstext „Programm in den Datenspeicher kopieren – Befehl XFER“.

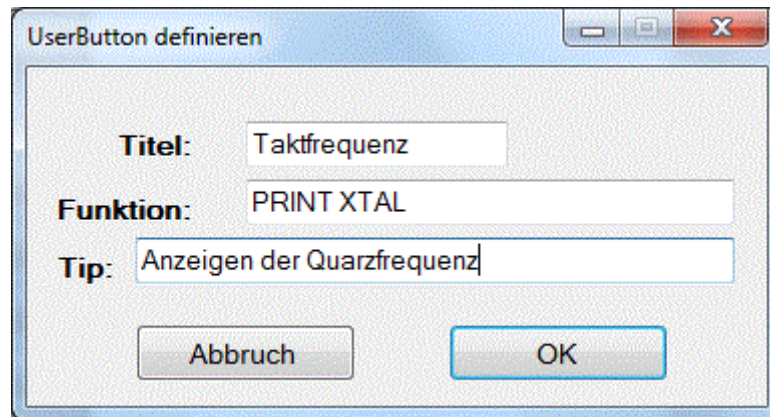


Auf den nächsten Button kann ein anderer Befehl gelegt werden.

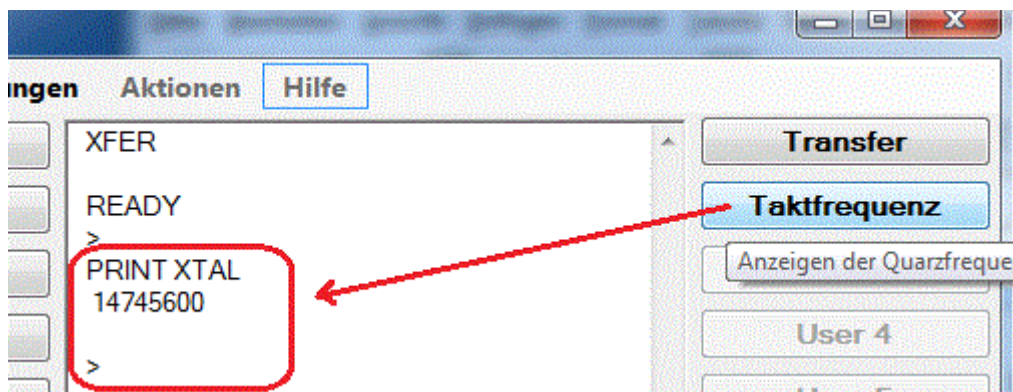




Als Beispiel wird hier das Kommando „PRINT XTAL“, zum anzeigen der Quarzfrequenz, definiert.



Bei Betätigung wird das eingetragene Kommando „PRINT XTAL“ zum Basic übertragen und das Basic antwortet mit seiner Taktfrequenz.

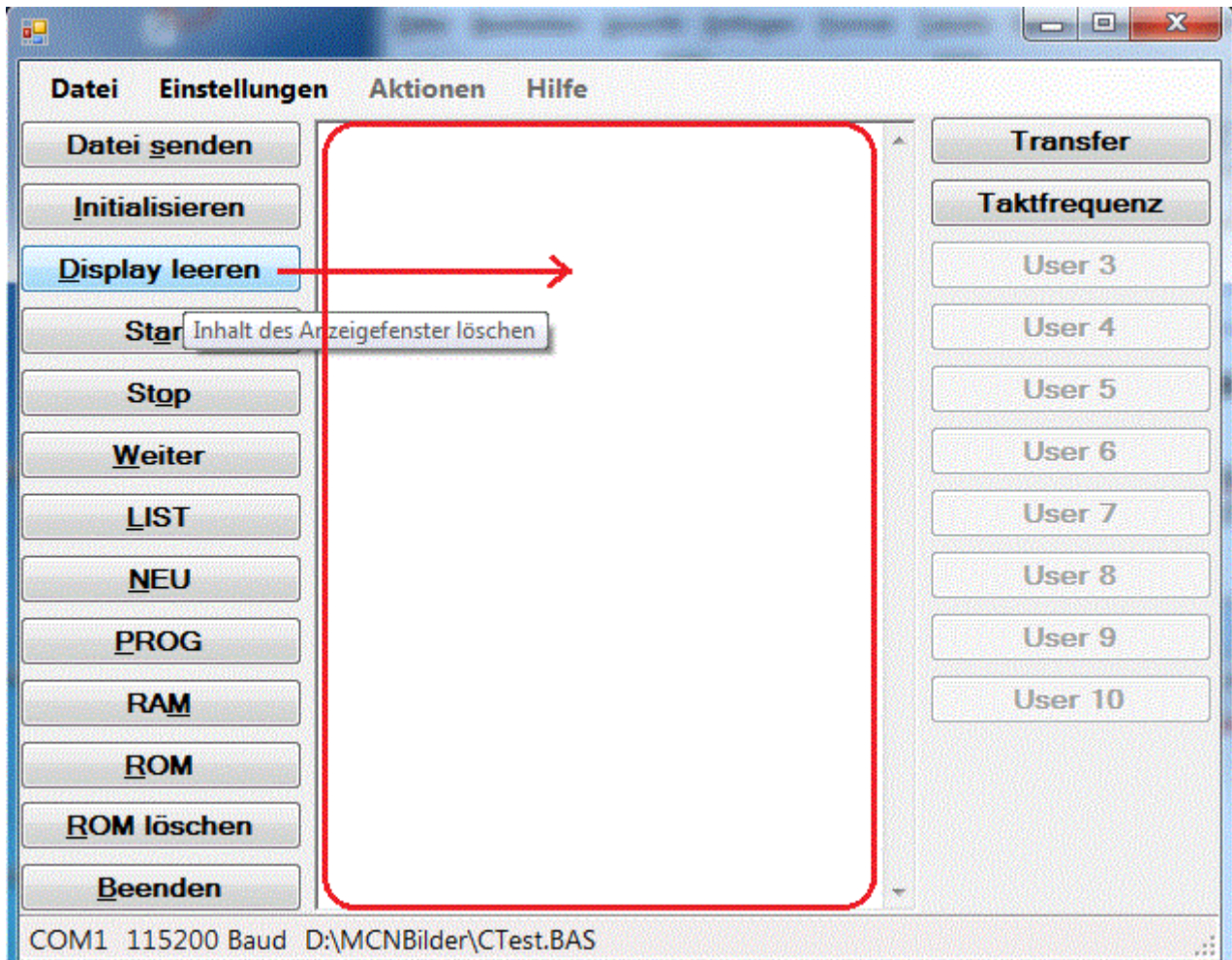


Auf die Weise können alle UserButtons mit beliebigen Befehlen und Kommandos belegt sein, um diese schnell per Knopfdruck zu erreichen. Ein belegter UserButton kann jederzeit mit anderen Werten definiert werden. Die Definition der UserButton ist in der Config-Datei gespeichert und wird so bei jedem Start des Terminals wieder hergestellt.



## 15 Anzeige löschen

Durch betätigen des „Display leeren“-Button wird der Inhalt des Anzeigefensters gelöscht. Dies dient nur der übersichtlichen Darstellung und hat keinerlei Einfluss auf das Basic.

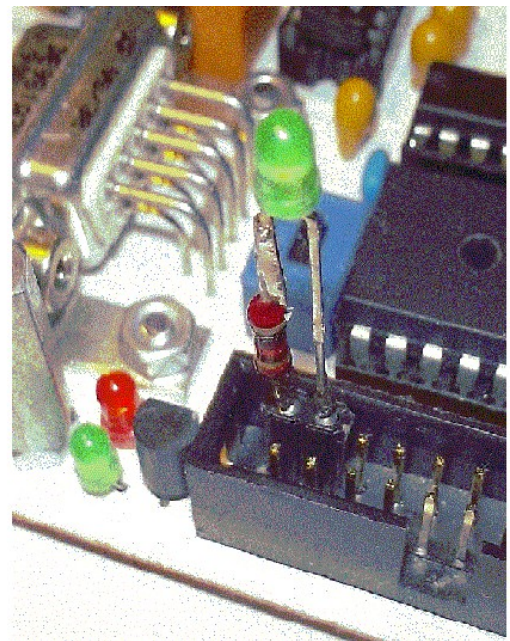
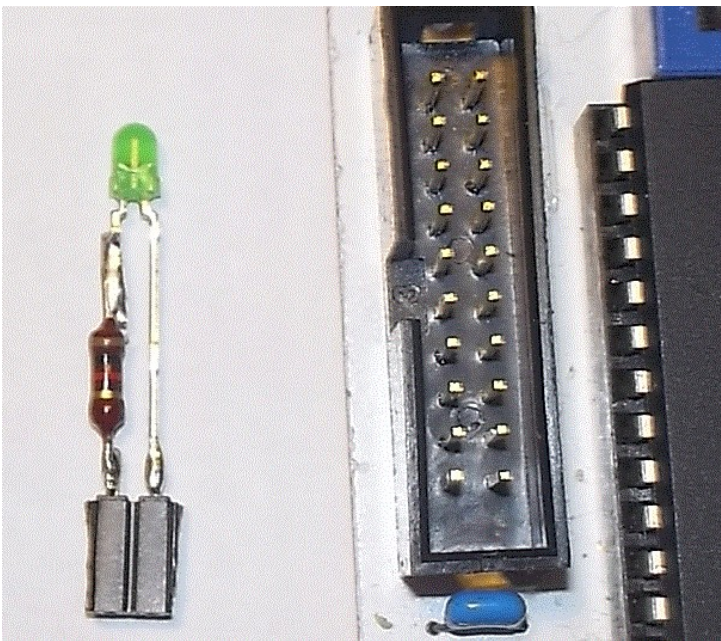


## 15 Terminal beenden

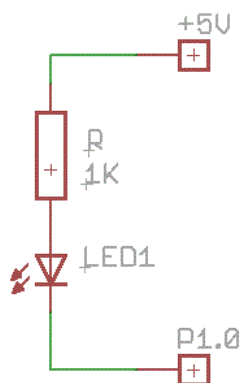
Betätigen des „Beenden“-Button oder im Menü „Datei – beenden“, beendet das Terminalprogramm.

## 16 Beispiel

Als Beispiel wird ein kleines Basic-Programm erstellt, das eine LED am Portbit P1.0 blinken läßt. Als Hardware wird nur ein Basic-Board und eine LED mit Vorwiderstand benötigt. Hier wird die IS51Basic-Platine verwendet. Die LED ist mit ihrem Vorwiderstand auf einen zweipoligen Rest Pfostenbuchse aufgebaut und kann so auf den kleinen Erweiterungsstecker gesteckt werden. Dort liegt der VCC-Pin direkt neben dem P1.0-Pin, wodurch diese Anzeige wenig Aufwand bereitet. Der Widerstand hat einen Wert von 1k-Ohm.



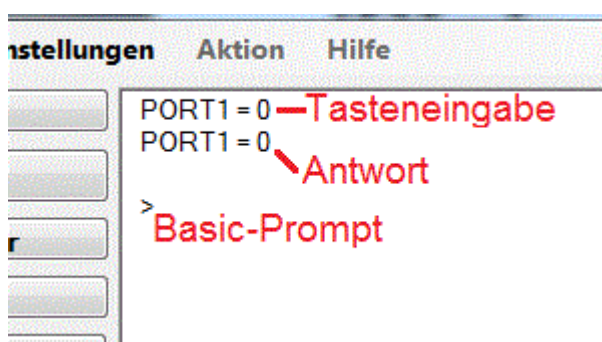
Das Schema zeigt die Verschaltung der Anzeige für das Portbit P1.0.



Ohne LED kann die Ausgabe auch mit einem Voltmeter oder Oszilloskop, direkt am Mikrocontroller Pin-1 nachgemessen werden. Anschaulicher ist es allerdings mit der LED.

## 16.1 Hardware Funktionstest

Für einen ersten Test der LED benötigen wir noch kein Programm, der Zustand des Port 1 wird direkt durch Tastatureingabe verändert. Durch das Kommando „PORT1 = 0“ werden alle 8 Bit des Port 1 zu 0, dadurch muss die LED leuchten. Im Bild ist die Anzeige nach



Ausführung des Befehls zu sehen. In der ersten Zeile der eingegebene Befehl, darunter die Antwort des Basic und zuletzt das Bereitschaftsprompt als Kennung, dass das Basic wieder bereit für neue Befehle ist.

Durch die Abarbeitung des Befehls wurden alle Bit des Port 1 zu 0, darunter auch P1.0, und die LED leuchtet. Damit ist die Funktionalität der Hardware bestätigt und wir können ein Programm schreiben.

## 16.2 Testprogramm Tastatureingabe

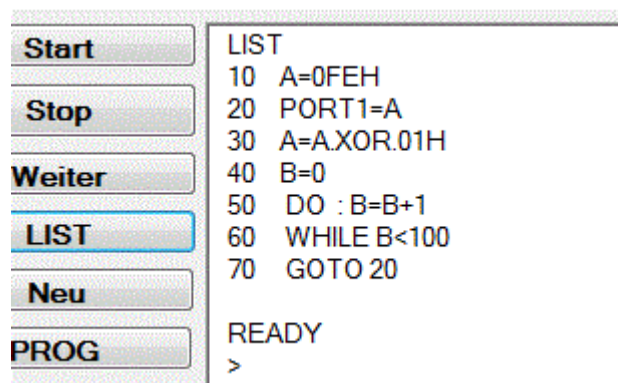
Dazu folgendes Programm (ohne Kommentare) mit der Tastatur in das Anzeigefenster eintippen:

```

10 A = 0FEH          ; Erster Ausgabewert
20 PORT1 = A         ; Zum Port ausgeben
30 A = A.XOR.01H     ; Bit 0 invertieren (umschalten)
40 B = 0             ; Zähler für Zeitverzögerung
50 DO : B = B + 1    ; Verzögerung zählen
60 WHILE B < 100     ; bis B = 100
70 GOTO 20           ; Endlos wiederholen

```

Die Eingabe mit „LIST“ überprüfen und ggf. korrigieren. Zur Korrektur muss nur die fehlerhafte Zeile neu geschrieben werden. Anschließend durch „START“ (RUN) starten.

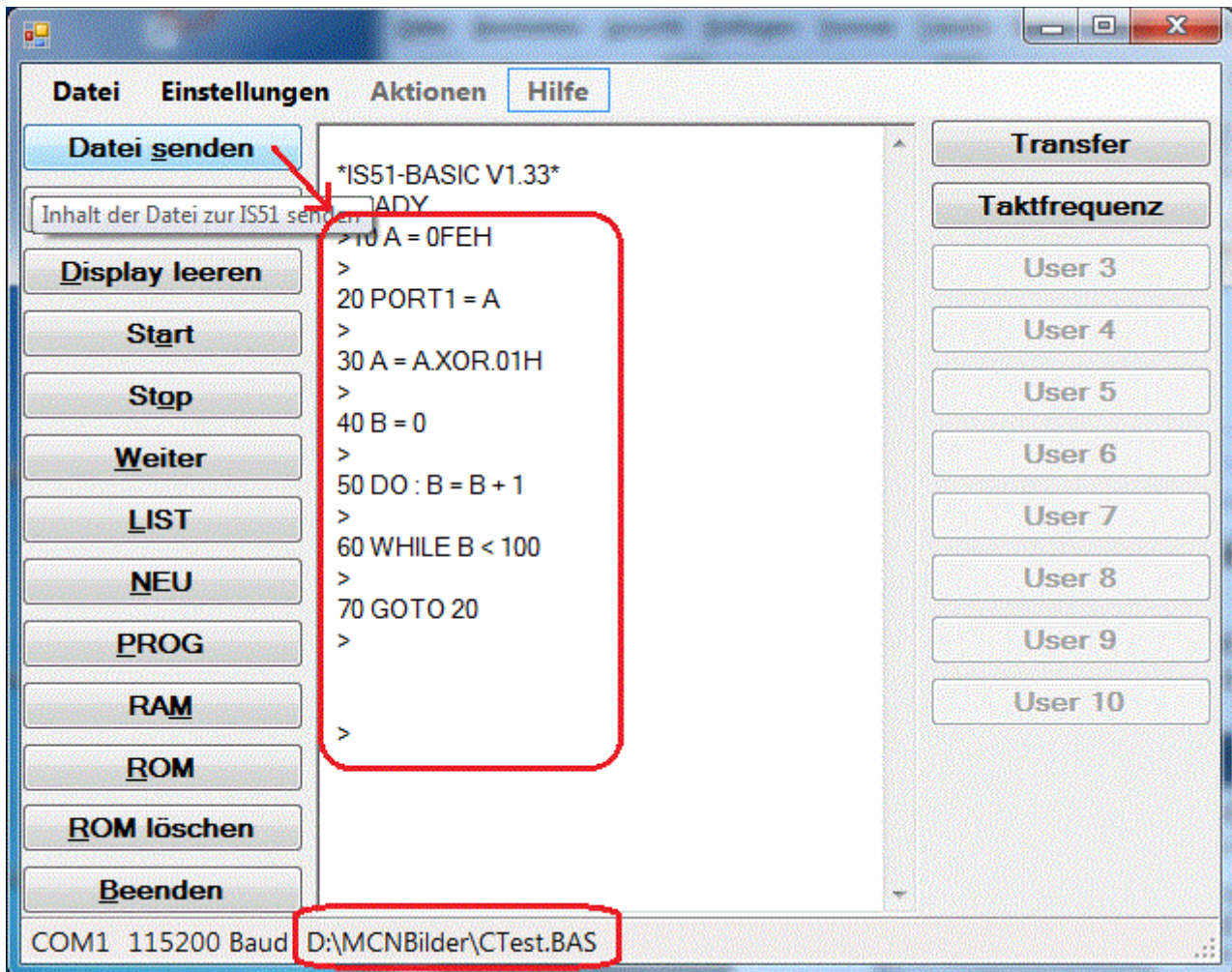


Jetzt sollte die LED an P1.0 blinken! Falls nicht, den Fehler suchen.



### 16.3 Testprogramm laden

Das Programm wird mit einem Editor erstellt und als Datei abgespeichert. Diese Datei wird im Terminal geöffnet und mit „Datei senden“ zum Basic übertragen.



### 16.4 Programm dauerhaft speichern

Wenn das Programm fehlerfrei läuft, kann es durch den PROG-Befehl in den Programmspeicher kopiert werden. Dort bleibt es auch beim abschalten der Versorgungsspannung erhalten.

Zum erneuten ausführen des Programms wechselt man mit ROM in den Programmspeicher und kann es dort mit LIST anzeigen oder mit RUN starten. Befindet man sich im Programmspeicher, kann das Programm von dort mit dem Befehl XFER in den Datenspeicher zurückkopiert werden, gleichzeitig wechselt XFER den Fokus in den Datenspeicher.

## **16.5 Autostart einrichten**

Befindet sich ein Programm im Programmspeicher, kann es durch den Befehl PROG2 (und Ähnliche – näheres im Handbuch zum Basic) zum Autostartprogramm werden. Von jetzt an wird es beim anlegen der Versorgungsspannung automatisch gestartet und ausgeführt. Mit einem STOP - Kommando (in der richtigen Baudrate) kann es wieder angehalten werden.

Vorgehensweise:

- Programm eintippen/übertragen
- Mit LIST überprüfen
- Funktionierendes Programm mit PROG abspeichern
- Programm mit PROG2 zum Autostartprogramm machen

Viel Spaß und Erfolg. Tom