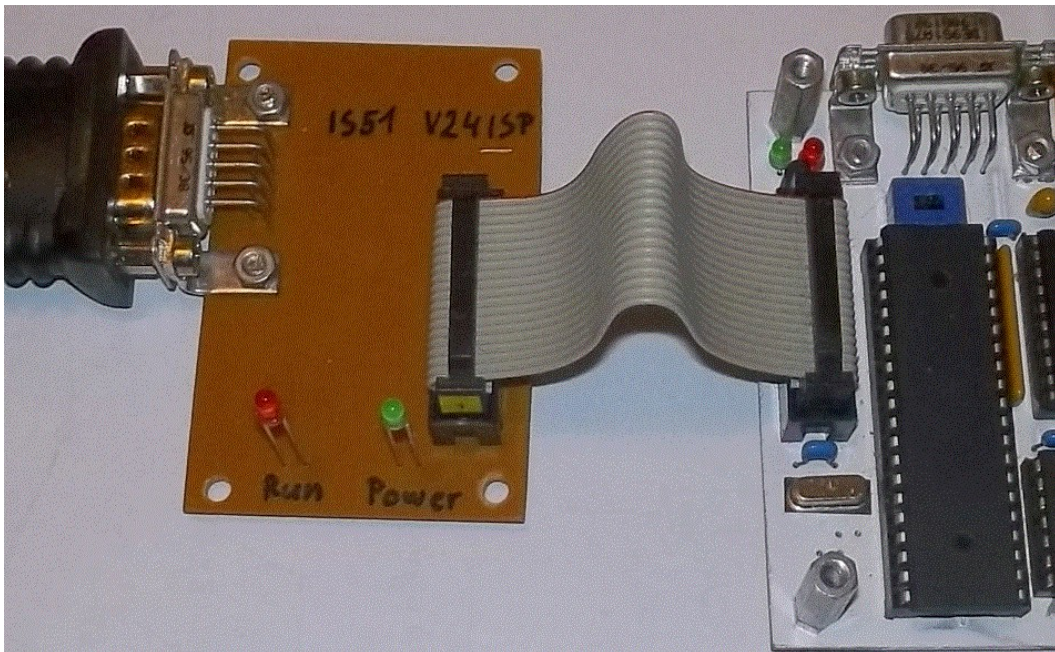


# Handbuch

## IS-Programmiergerät

### für RS232



# Inhaltsverzeichnis

1. Allgemeine Beschreibung.....	3
1.1 Wer braucht das Programmiergerät ?.....	3
2 Hardware.....	4
2.1 Schaltplan.....	4
2.2 Beschreibung der Hardware.....	4
2.2.1 Zuordnung der Signale.....	5
2.2.2 Spannungsversorgung.....	5
2.2.3 Verbindungskabel Zielsystem.....	5
2.2.4 Verbindungskabel PC.....	6
3 Layout und Bestückungsplan.....	7
4 Software.....	8
4.1 Aufrufparameter.....	8
4.2 Beispiele Konsole.....	9
4.2.1 Bildschirmbeispiele.....	10
4.3 Beispiel eingebettet in Keil $\mu$ Vision.....	11
5 Bemerkungen.....	13

# 1. Allgemeine Beschreibung

Bei den meisten Programmiergeräten erledigt ein programmierbarer Baustein die Aufgabe, den Zielchip zu programmieren. Um diesen Baustein programmieren zu können, braucht man aber ein Programmiergerät. Dieses Henne-Ei Problem zu umgehen, gibt es das V24ISP-Programmiergerät. Es kommt ohne spezielle Bauteile, vor Allem ohne programmierbaren Baustein aus. Zentraler Baustein ist ein TTL-RS232 Wandler zum anpassen der Spannungspegel zwischen dem RS232 Leitungspegel und dem TTL-Pegel des Zielsystems. Arbeitet das Zielsystem mit anderen Pegeln (z.B: 3,3V), muss nur ein geeigneter Pegelwandler-Baustein verwendet werden.

Die Programmierlogik befindet sich im PC-Programm und kann deshalb für viele verschiedene Zielsysteme geschrieben sein. Das programmieren des Zielbausteins ist nicht besonders schnell, da das Programm die Leitungen der RS232 einzeln bedient, um die nötigen Steuersignale zu erzeugen.

Dieses Handbuch beschreibt die Hardware des IS-Programmers und ein Programm für die MCS51-Mikrocontroller AT89S82, AT89S8252 und AT89S8253 für die IS51-Platine. Das vorgestellte Layout passt direkt zu dieser Platine. Das Programm ist eine Windows-Konsolenanwendung und kann von der Konsole oder aus einer Entwicklungsumgebung heraus, gestartet werden

Die Hardware ist so einfach, dass sie auch auf einer Lochrasterplatine und sogar auf einem Steckbrett funktioniert.

## 1.1 Wer braucht das Programmiergerät ?

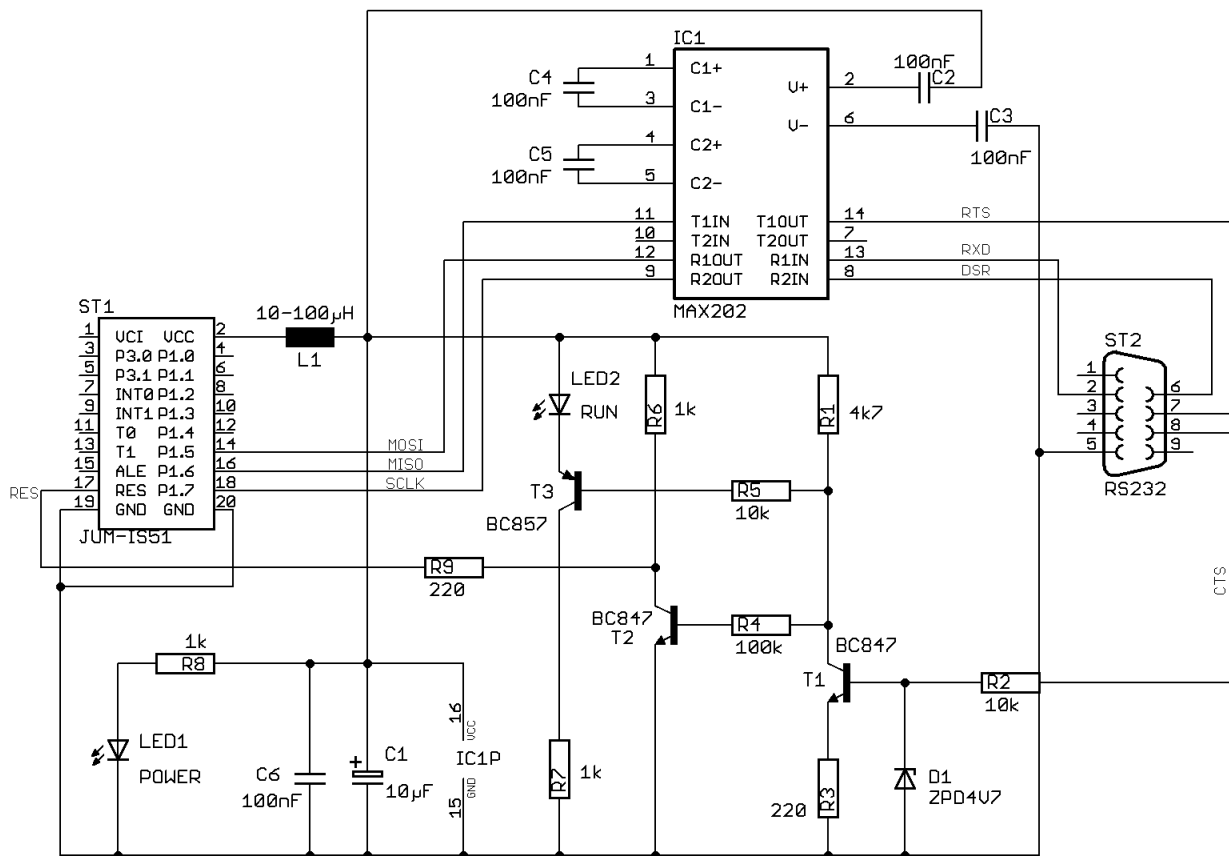
Wer in die Arbeit mit Mikrocontrollern einsteigt und noch nicht viel Geld für ein kommerzielles Gerät ausgeben möchte. Oder wer nicht viele Bausteine programmieren muss, sondern nur gelegentlich einen Baustein programmiert um etwas auszuprobieren.

Jemand der nur ein einziges mal einen Baustein programmieren möchte, kann die Schaltung auf einem Steckbrett auf- und nach Verwendung wieder abbauen.

Wer häufig ein Programmiergerät für verschiedene Zielsysteme braucht, kommt mit einem kommerziellen Gerät besser zurecht. Natürlich besteht die Möglichkeit, sich eine Software für ein Programmiergerät im Zielsystem zu schreiben und dieses mit dem V24ISP erstmalig in den Chip zu schreiben.

## 2 Hardware

### 2.1 Schaltplan



### 2.2 Beschreibung der Hardware

Zentraler Baustein ist hier ein MAX202. Er hat den Vorteil mit 0,1µF Kondensatoren zurecht zu kommen. Der MAX232 kann natürlich auch verwendet werden, braucht aber Kondensatoren mit größerer Kapazität (z.B: 10µF). Aufgabe dieses Bausteins ist die Anpassung der hohen Spannungspegel der RS232 Schnittstelle an die niedrigeren TTL-Pegel des Zielsystems. Da der Pegelwandler-Baustein nur zwei Treiber für jede Richtung besitzt, wird das dritte Signal (CTS) mit einem Transistor-Pegelwandler angepasst. Dieses bildet das Reset-Signal, dessen Geschwindigkeit nicht hoch zu sein braucht. Es wird zu Beginn des Programmierens aktiviert und am Ende wieder deaktiviert, während des Programmierens bleibt es aktiv. Die LED „RUN“ (D2) zeigt durch leuchten an, dass die Programmierung aktiv ist. So wie die LED jetzt beschaltet ist, leuchtet sie bei Highpegel am RESET des Zielsystems. Bei Lowaktiven RESET muss T3 ein NPN-Typ (z.B: BC847) sein und sein Emitter liegt an R7, der Kollektor an der LED. Das Reset-Signal für das Zielsystem wird aus dem CTS-Signal der RS232 gebildet. Die Software im PC-Programm hat dafür zu sorgen, dass nach aktivieren von Reset ausreichend lange (>100ms) gewartet wird, bis der Pegel sich stabilisiert hat.

Das Taktsignal SCLK des Zielsystems wird vom RS232-Signal DSR gebildet und sein Pegel vom MAX202 nach TTL gewandelt.

Das Datensignal MOSI des Zielsystems wird vom RS232-Signal RxD gebildet und sein Pegel vom MAX202 nach TTL gewandelt.

In der Richtung vom Zielsystem zum PC wird das Datensignal MISO im MAX202 auf RS232-Pegel gewandelt und als RTS zurückgegeben.

## 2.2.1 Zuordnung der Signale

Vom PC zum Zielsystem:	RESET	CTS
	SCLK	DSR
	MOSI	RXD

Vom Zielsystem zum PC	MISO	RTS
-----------------------	------	-----

## 2.2.2 Spannungsversorgung

In der gegebenen Schaltung wird das Programmiergerät aus dem Zielsystem (IS51) versorgt. Auf dem Stecker ST1 liegt auch die Spannung +5V und der Systemground. Für ein stabiles System muss die Spule L1 so nah als möglich an den Stecker ST1. Der Wert der Spule ist unkritisch und liegt zwischen 10 $\mu$ H bis 100 $\mu$ H.

Der Kondensator C6 ist so nahe wie möglich an den Versorgungspins des Pegelwandler-Bausteins MAX202 anzubringen.

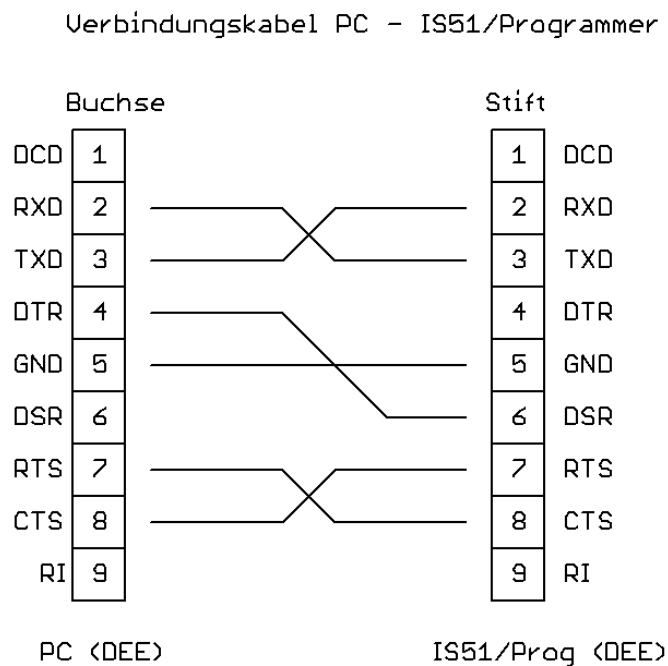
## 2.2.3 Verbindungskabel Zielsystem

In der gegebenen Schaltung passt der Verbindungsstecker ST1, über 20 polige Flachbandleitung, direkt zum Erweiterungsstecker der IS51. Für andere Zielsysteme sind die Signale des 20pol. Steckers ST1, über ein Adapterkabel, an den entsprechenden Stecker anzupassen.

## 2.2.4 Verbindungskabel PC

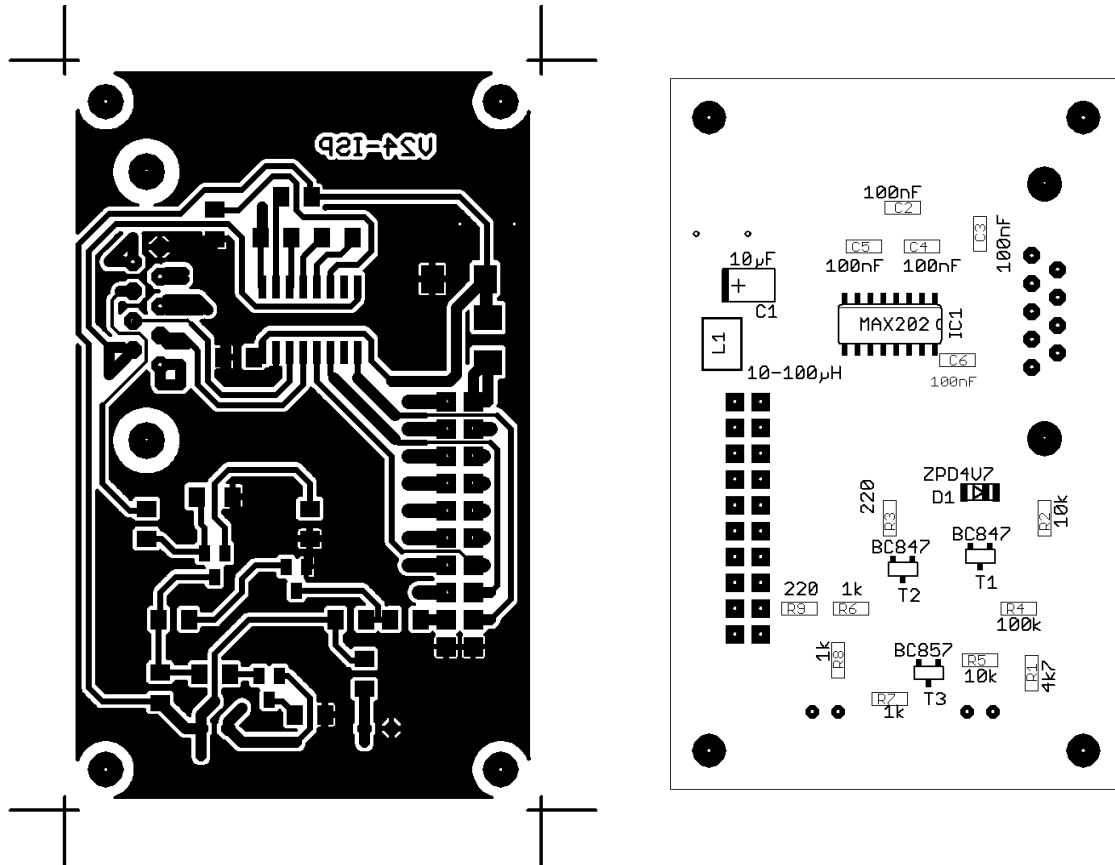
Damit nur ein Kabel für die Verbindung zwischen der IS51 zum PC oder dem Programmiergerät zum PC gebraucht wird, orientiert sich die Belegung an der IS51.

Die Geräte sind, wie der PC, als DEE definiert, weshalb ein gekreuztes Kabel benötigt wird. Man kann dafür ein handelsübliches NullModem-Kabel verwenden, oder selbst Eines fertigen. Es sind nicht alle Signale benutzt und müssen nicht vorhanden sein. Das Bild zeigt die minimalen Verbindungen für ein Kabel, dass sowohl für die IS51 (Download oder Basic), als auch für das Programmiergerät ausreicht:

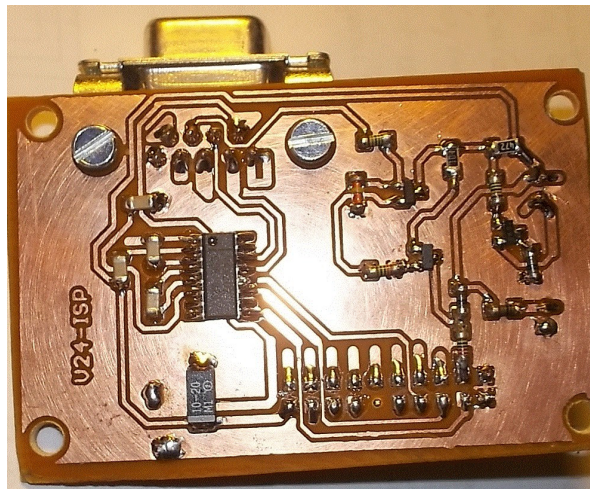
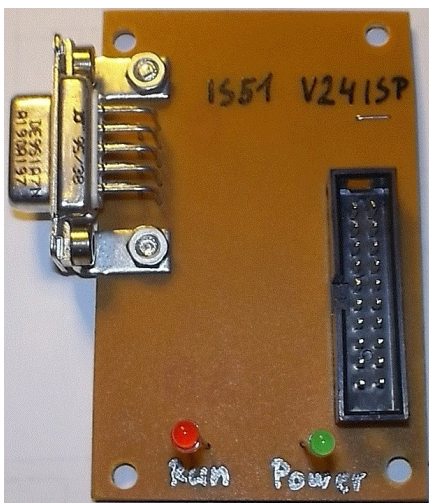


### 3 Layout und Bestückungsplan

Hier ein Layout für das Programmiergerät zur IS51, mit dem zugehörigen Bestückungsplan.



Die Größe des Originals zwischen den Kreuzungspunkten der Eckkreuze ist 44,5mm auf 70,0mm. Die beiden Punkte über Kondensator C1 gehören zu einer nicht eingezeichneten Brücke. Auf einer Platine muss die Brücke bestückt sein.



## 4 Software

Das Programm für die ATMEL Mikrocontroller der IS51 nennt sich „V24PG.EXE“. Es ist ein Windows Konsolenprogramm und kann von der Konsole, oder aus anderen Programmen heraus aufgerufen werden.

### 4.1 Aufrufparameter

Die Funktionen und Einstellungen werden über Aufrufparameter getroffen. Die allgemeine Form eines Programmaufrufes lautet:

#### ***V24PG Dateiname Funktion Schnittstelle Steuerbit Mikrocontroller***

**Dateiname:** ist der Name der IntelHex Datei die benutzt werden soll. Dabei wird der Dateierweiterung ".HEX" nicht mit angegeben, er wird vom Programm vorausgesetzt. Bei manchen Funktionen (z.B: löschen) wird kein Dateiname gebraucht, es muss trotzdem ein beliebiger Name angegeben sein.

**Funktion:** Dabei ist die Vorgabe, Großbuchstaben Kennzeichnen den Programmspeicher (Flash), Kleinbuchstaben Kennzeichnen den Datenspeicher (EEPROM)

- W**    Programmspeicher schreiben (Groß W)
- w**    Datenspeicher schreiben (Klein w)
- B**    Programmspeicher AT89S52 im Bytemode schreiben
- R**    Programmspeicher lesen
- r**    Datenspeicher lesen
- L**    Leertest Programmspeicher
- I**    Leertest Datenspeicher
- V**    Programmspeicher vergleichen (Groß V)
- v**    Datenspeicher vergleichen (Klein v)
- Kk**   Speicher löschen (Programm- & Datenspeicher gleichzeitig)
- Ss**   Baustein sichern

Bei K und S ist jeweils nur ein Buchstabe anzugeben, es ist aber egal ob Groß- oder Kleinbuchstabe.



**Schnittstelle:** ist die Nummer der seriellen RS232-Schnittstelle zur Übertragung. Es sind COM1 bis Com99 (1-99) erlaubt.

**Steuerbit:** dienen der Programmeinstellung

Bit 0 = Anzeige am Bildschirm: 0 = ohne, 1 = mit Anzeige

Ohne Anzeige kann das Programm im Hintergrund laufen.

Mit Anzeige wird am Ende auf Tastendruck gewartet.

**Mikrocontroller:** der gewünschte Mikrocontroller wird über eine zugeordnete Nummer gewählt.

1 = AT89S52

2 = AT89S8252

3 = AG89S8253

## 4.2 Beispiele Konsole

**"V24PG TEST W 1 1 2"** = Datei TEST.HEX über COM1 in den Programmspeicher eines AT89S8252 schreiben mit Anzeige.

**"V24PG TEST w 1 1 2"** = Datei TEST.HEX über COM1 in den Datenspeicher eines AT89S8252 schreiben mit Anzeige.

**"V24PG TEST k 2 1 3"** = Programm- & Datenspeicher eines AT89S8253 löschen. Programmiergerät an COM2, mit Anzeige. Dateiname wird nicht gebraucht, muss aber angegeben sein.

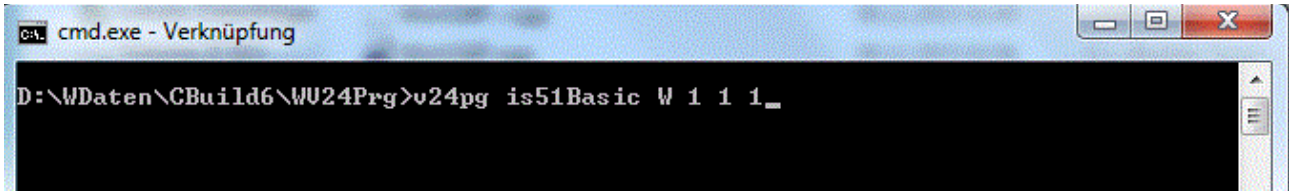
**"V24PG READ1 R 1 0 1"** = Programmspeicher eines AT89S52 in die Datei READ1.HEX lesen. Schnittstelle COM1, ohne Anzeige

**"V24PG ABCD W 9 0 1"** = Datei ABCD.HEX über COM9 in den Programmspeicher eines AT89S52 schreiben ohne Anzeige.

**"V24PG TEST S 3 1 2"** = AT89S8252 sichern mit Anzeige. Schnittstelle COM3.

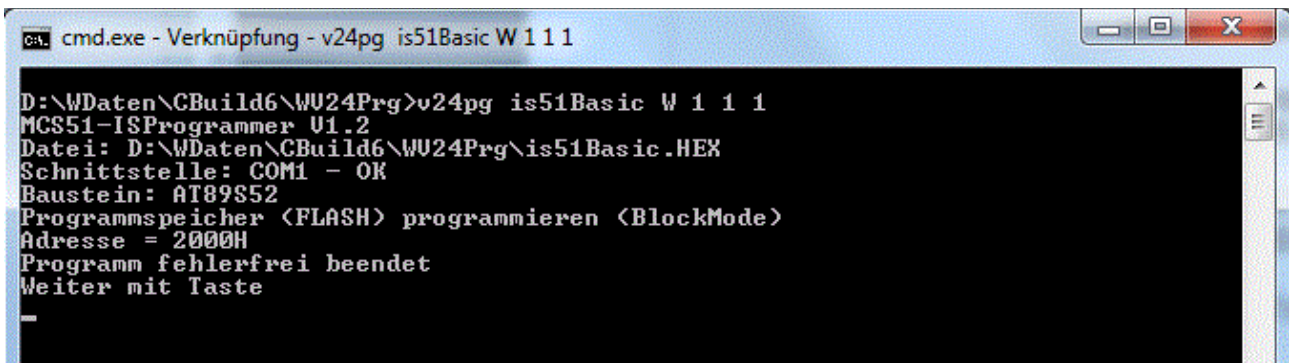
Der Speicher des Zielchip wird vor programmieren des Programmspeichers (Flash) automatisch gelöscht.

## 4.2.1 Bildschirmbeispiele



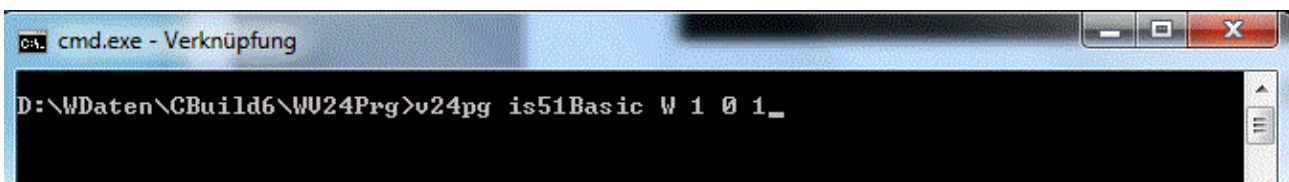
```
cmd.exe - Verknüpfung
D:\WDaten\CBuild6\WU24Prg>v24pg is51Basic W 1 1 1_
```

Beispiel eines Aufrufes von der Konsole. Die Datei IS51Basic.HEX wird mit Anzeige über COM1 in den Programmspeicher eines AT89S52 geschrieben.



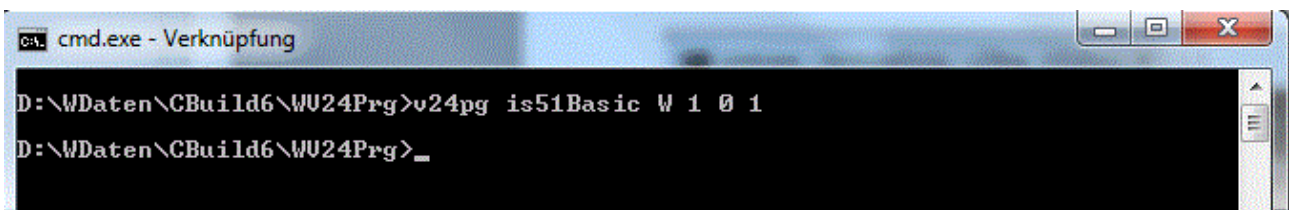
```
cmd.exe - Verknüpfung - v24pg is51Basic W 1 1 1
D:\WDaten\CBuild6\WU24Prg>v24pg is51Basic W 1 1 1
MCS51-ISProgrammer U1.2
Datei: D:\WDaten\CBuild6\WU24Prg\is51Basic.HEX
Schnittstelle: COM1 - OK
Baustein: AT89S52
Programmspeicher (FLASH) programmieren (BlockMode)
Adresse = 2000H
Programm fehlerfrei beendet
Weiter mit Taste
_
```

Hier die Reaktion des Programms. Da mit Anzeige gewählt war, erscheinen die Anzeigen des Programms auf der Konsole. Nach Beendigung wartet das Programm auf betätigen einer Taste.



```
cmd.exe - Verknüpfung
D:\WDaten\CBuild6\WU24Prg>v24pg is51Basic W 1 0 1_
```

Hier der Aufruf mit Parameter „ohne Anzeige“.

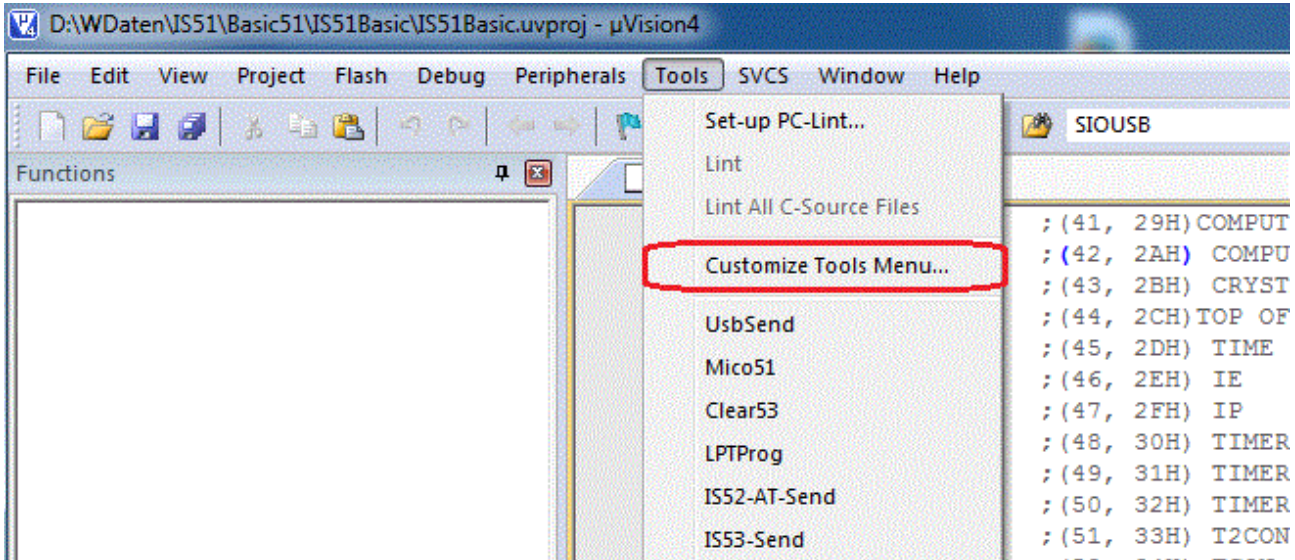


```
cmd.exe - Verknüpfung
D:\WDaten\CBuild6\WU24Prg>v24pg is51Basic W 1 0 1
D:\WDaten\CBuild6\WU24Prg>_
```

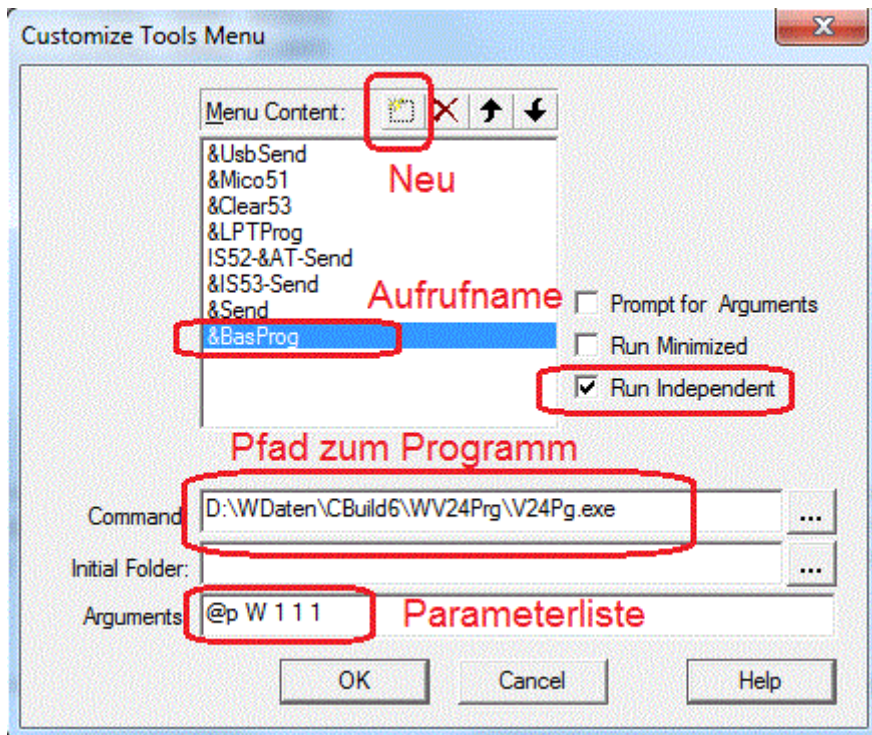
Jetzt wird nichts in die Konsole geschrieben. Nach Beendigung des Programms erscheint das Eingabeprompt der Konsole. Das Programm wartet nicht auf einen Tastendruck.

### 43 Beispiel eingebettet in Keil $\mu$ Vision

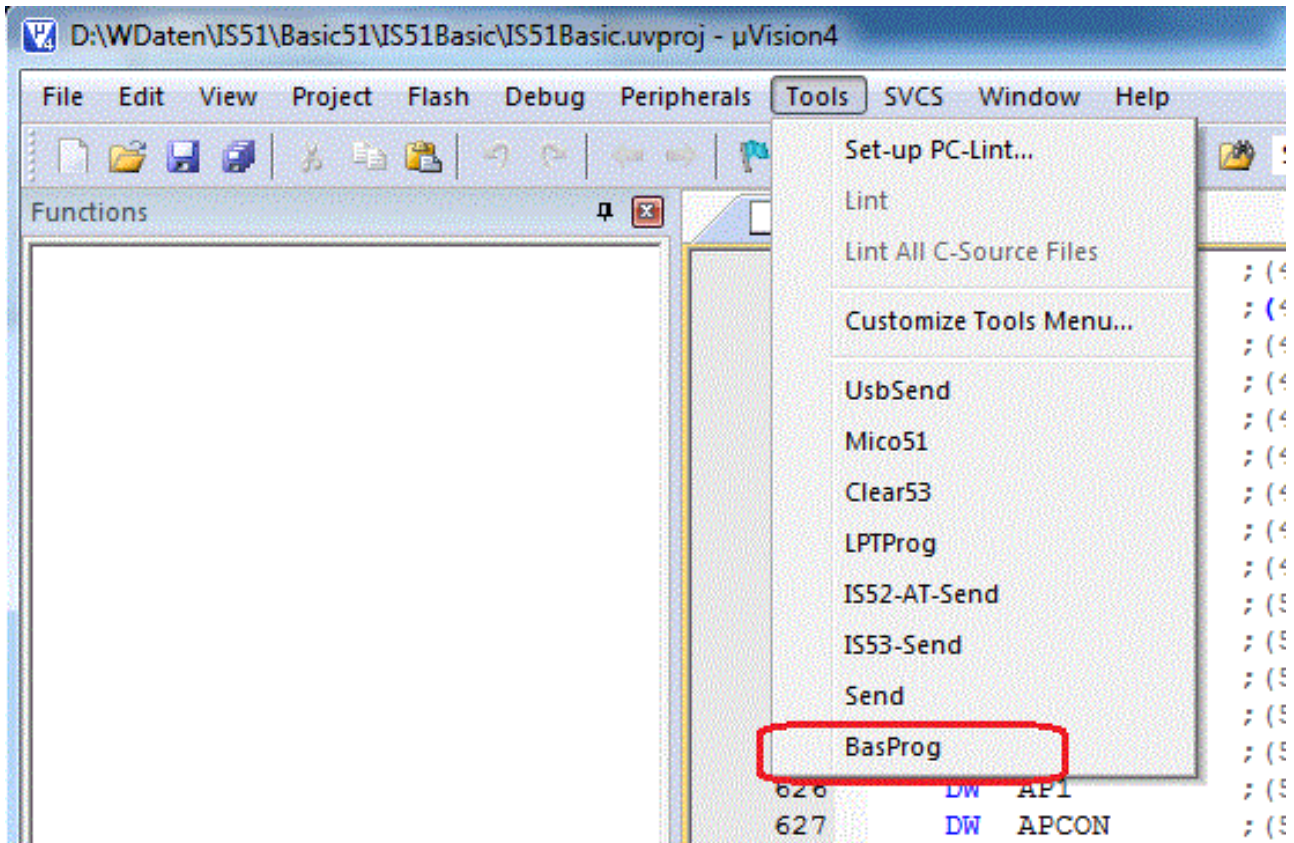
Um den IS-Programmer in Keil  $\mu$ Vision zu integrieren, ist im Menü unter „Tools“ der Item „Customize Tools Menu...“ zu öffnen.



Hier wird ein neuer Eintrag gewählt und sein Aufrufname eingetragen. „Run Independent“ erzeugt für die Ausgaben des Programms ein eigenes Fenster, sonst wird im Ausgabefenster von  $\mu$ Vision angezeigt.

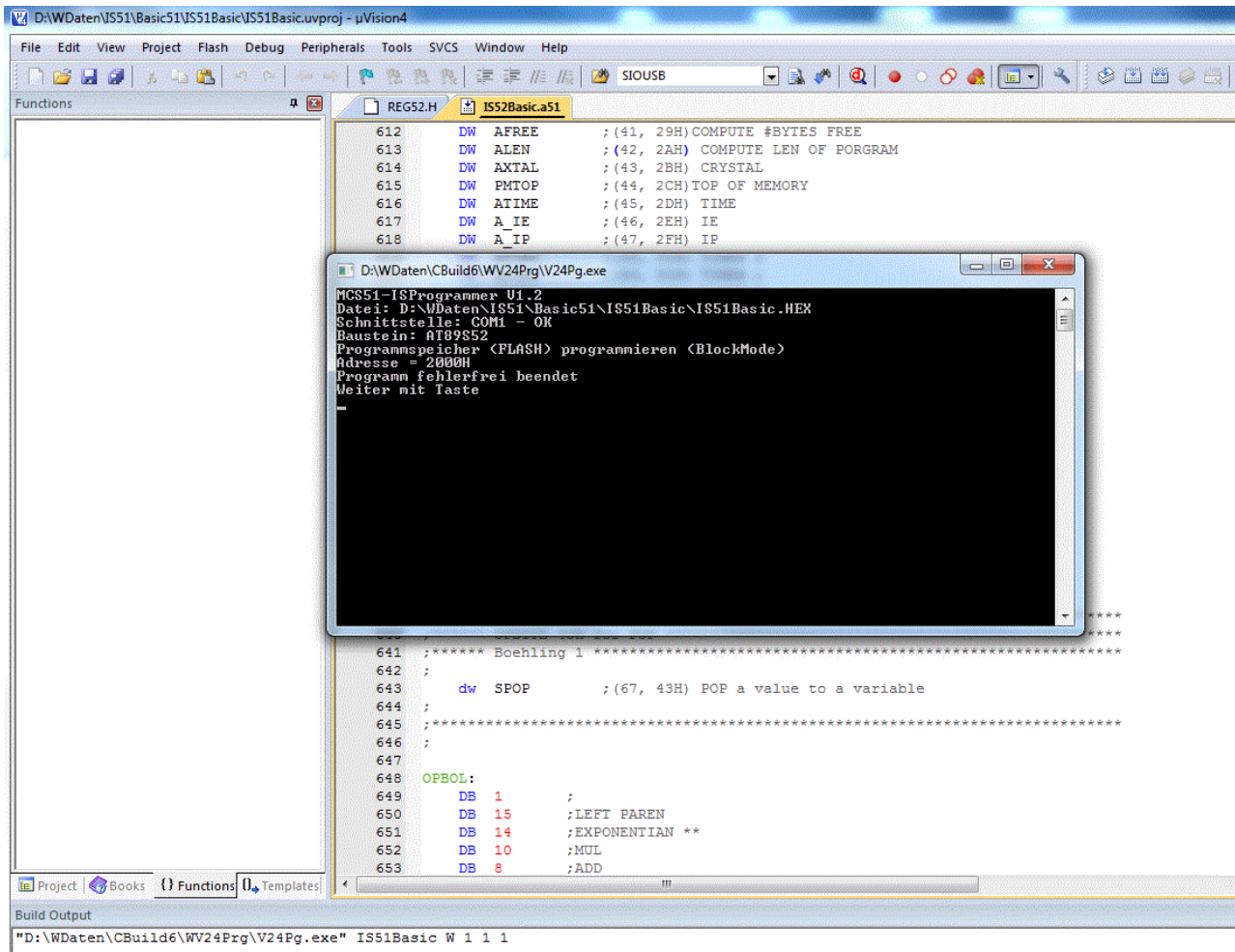


Bei „Command“ wird der Pfad zum Programm eingestellt und bei Arguments die Aufrufparameter für den Programmierer. Das „@p“ ist eine Variable von µVision, welche den Namen der aktuellen Datei beinhaltet. So wird der Aufruf stets mit dem Programm, an dem gerade gearbeitet wird, erfolgen.



Nun ist ein neuer Eintrag im Tools-Menü über den der Code des gerade bearbeiteten Programms (Assembler oder C) in das Zielsystem geflasht wird.

Beim anklicken des neuen Menü-Item wird das Programm in einem eigenen Fenster gestartet. Wurde „Run Independent“ nicht markiert, schreibt der Programmierer seine Ausgaben in das Meldfenster unten, bei „Build Output“.



## 5 Bemerkungen

V1.2 AT89S8252 alle Funktionen zugänglich.

AT89S8253 schreiben, lesen, vergleichen, Leertest und löschen

AT89S52 schreiben, lesen, vergleichen, Leertest und löschen

Sichern des AT89S8252 setzt die höchste Sicherheitsstufe.