

Inhaltsverzeichnis:

Allgemein.....	2
ISP-Programmierung	2
Blockdiagramm	3
Bootstrap-Monitor.....	4
Hauptmenü	4
Hauptfunktionen.....	4
Autostart vom EEPROM	5
Program downloaden	5
Debugger.....	6
Programm im EEPROM ablegen.....	7
Download aus dem EEPROM.....	7
Interrupttabelle des Monitors	8
Schaltplan.....	9
Bestückung.....	10

Allgemein

Das uc-Board2 ist eine Weiterentwicklung des uC-Boards. Dieses BasisBoard wurde unter anderem für eine Anwendung entwickelt, welches eine Inhausvernetzung von Komponenten mit einer preiswerten Microcontrollervariante realisieren soll. Das Modul ist klein und flexibel genug, um in Unterputzdosen untergebracht zu werden.

Das uC-Board2 ist mit einem ISP-fähigen Microcontroller (AT89S52 PLCC44) ausgestattet. Da die CPU gesockelt ist, kann ein entsprechender pinkompatibler Microcontroller verwendet werden. Die zwei Anschlußleisten im 2,54mm Raster erlauben eine einfache Anbindung der Zusatz-Hardware. Die geringe Größe des Boards (50x80mm) erlaubt zudem eine kompakte Bauweise, um die Gesamtschaltung (BasisBoard + Zusatz-Hardware) in kleinen Gehäusen unter zu bringen.

Das uc-Board2 besitzt onboard bereits mehrere Basiskomponenten:

- PowerSupply
- ISP-Connector (CPU auf PLCC44-Fassung)
- Adress/Datenbus sowie Adressdekoder
- 32KB-ROM oder RAM
- serieller EEPROM (auf Faassung)
- COM Schnittstelle
- eine onboard-LED

Die 8051 CPU enthält bereits eine Firmware mit folgenden Features:

- Upload/Download eines Programms vom Eeprom (gesichert mit Checksumme)
- Download eines Programms per COM Schnittstelle
- Single-Step-Debugger
- Programmierung im 32KB RAM möglich (kein ständiges Flashen nötig)

ISP-Programmierung

Die Programmierung des internen Flashspeichers des Mikrocontrollers erfolgt über die ISP-Schnittstelle. Unter Windowsbetriebssystemen (bevorzugt Win9x) kann die

Programmersoftware „Aec_isp.exe“ der Firma AEC Electronics genutzt werden.

Diese Software ist unter <http://www.aec-electronics.co.nz> verfügbar. Die rechte

Abbildung zeigt das Programmierkabel.

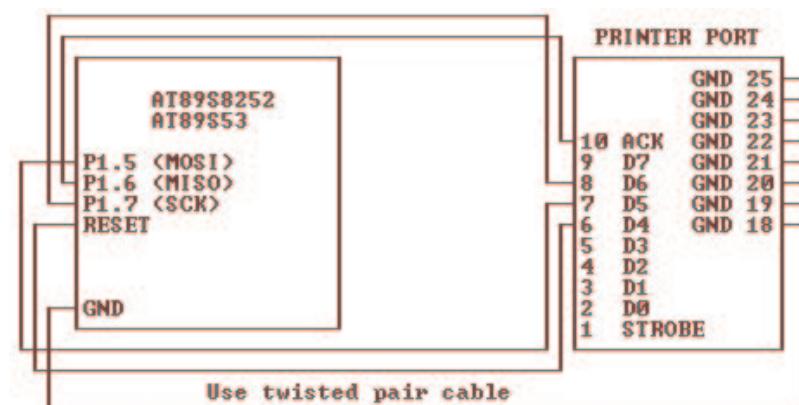
In der Version 3.0 werden folgende Mikrocontroller unterstützt:

- AT89S8252
- AT89S51 / 52 /53

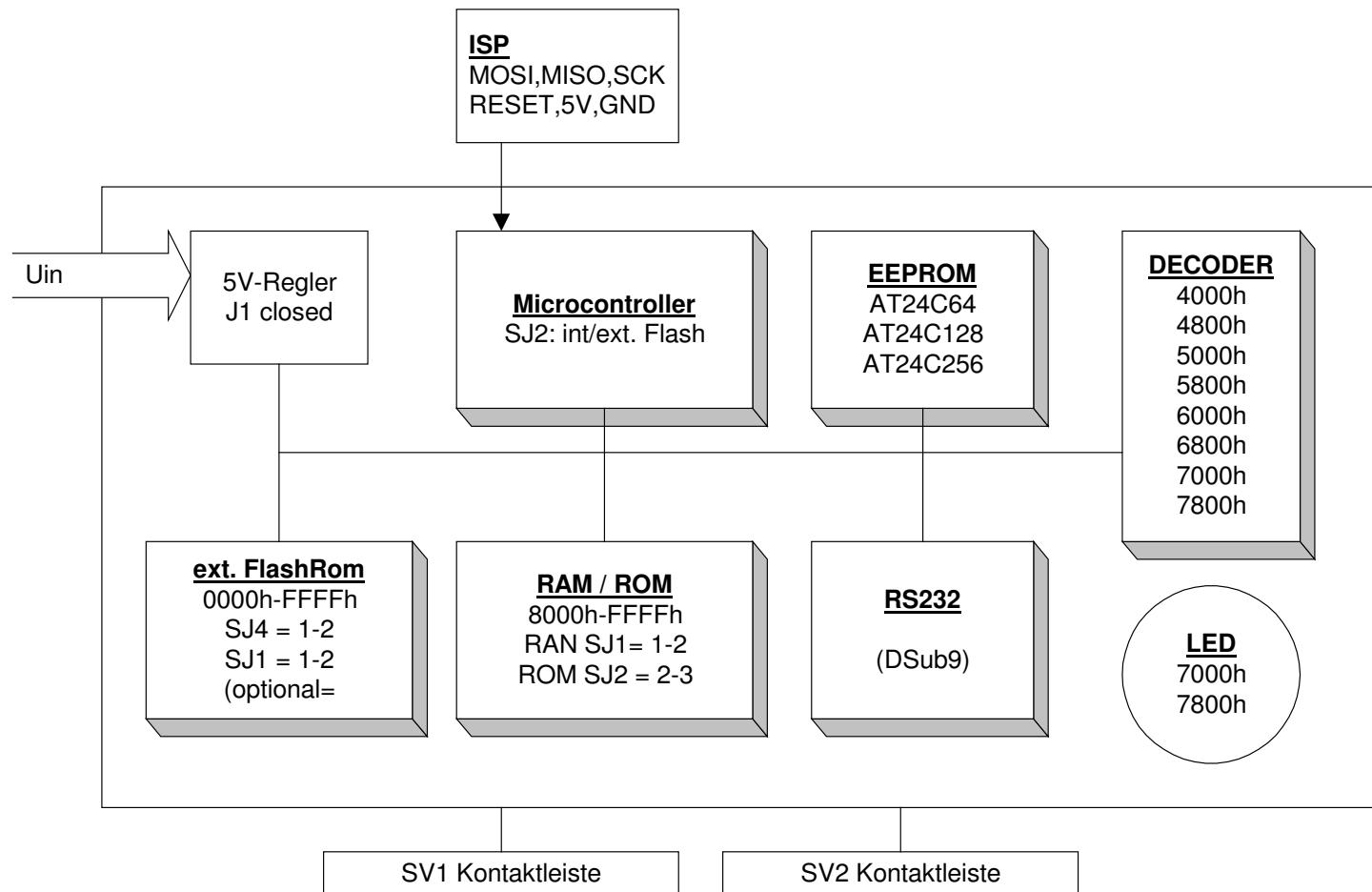
Zum Programmieren wird ein kompliertes Programm im HEX-Format benötigt.

Die Steuerung des Programmiertools ist menügeführt.

Hinweis: Unter Win2000/XP wird für den Zugriff auf den parallelen Port noch ein zusätzlicher Treiber (z.B. AllowIo) benötigt.



Blockdiagramm



Bootstrap-Monitor

Hauptmenü

So meldet sich der Monitor, wenn kein Programm automatisch vom EEPROM geladen wird.

```
Welcome to DEB51MON (www.Frydaysoft.de)
 [based on code from Paul Stoffregen - PAULMON]
```

some features:

- Download program (hex file)
- Run program from RAM (8000h)
- Normal, or single-stepping
- <ESC> key is supported
- List assemble code from memory
- Hex Dump and Edit external RAM
- On-line help (type ?)

```
>Loc=8000: (Version 1.50) Command>
```

Hauptfunktionen

Die nachfolgenden Kommandos stehen im Hauptmenu zur Verfügung.

These commands are currently supported:

```
?      Help ?
R      Run program
D      Download program (hex file)
N      New memory location
H      HEX Dump Memory to the Screen
L      List assembly code
E      Edit external ram
W      Write bytes from 8000h to eeprom
U      Upload bytes from eeprom to 8000h
```

All numerical values are shown in hex.

Pressing <ESC> key will exit the current command.

Most prompts require only single character.

For information regarding the single-step run feature,
type `?` when asked `Single-step/Normal` before running the program.

Autostart vom EEPROM

Beim Start des Boards wird geprüft, ob am I²C (Port 1.6 & 1.7 des Prozessors) ein EEPROM angeschlossen ist und dieser eine gültige Datensequenz (Header) liefert. Ist der Header im Eeprom gültig, wird das Programm vom EEPROM in den RAM kopiert und bei korrekter Checksum wird es gestartet. Ausgaben erfolgen nur noch im Fehlerfall!

DB0	DB1	DB2	DB3	DB4-N
Header Size	ProgLen HI	ProgLen LO	ChkSum	Program

Hinweis: Um das automatische Booten vom EEPROM zu unterlassen, muß während der Bootphase JP5 offen sein. Hierdurch wird der Eeprom nicht mehr erkannt.

Program downloaden

Im Hauptmenü Taste „D“ drücken und anschließend das Program im Hexformat an den Monitor senden. Das Programm muß auf Adresse 8000h gelinkt sein, da der Monitor nur Programme im RAM (0x8000h-0xFFFFh) ausführen kann.

Download program (hex file)

Begin Intel HEX-transfer or <ESC>

```
108000007581307801907800E012804C12804C121B
10801000804C907000E012804C12804C12804C809A
10802000E43092FDE804440FF812804C8890908070
10803000213090EDA312804C12804C12804C309174
10804000E012804C12804C12804C80D575F0007488
0880500000D5E0FDD5F0F82297
00000001FF
Download completed
```

>Loc=8000: (Version 1.50) Command>

Debugger

Um ein Programm zu debuggen, muß sich dieses im RAM befinden. Weiterhin muß JP4 gesteckt sein, damit das Programm im Schrittmodus per IRQ gestepppt werden kann. Im Hauptmenü ist nun Taste „R“ zu drücken. Die anschließenden Abfragen können im Regelfall mit „Enter“ bestätigt werden, da die vorgegebenen Defaultwerte bereits die gewünschten Einstellungen haben.

Run program

```
S=Single Step, N=Normal (default), ?=Help >
Run in single step mode
```

Run from memory loaction (8000=Default):

```
Interrupt priority> L=Low, H=High (default):
Now running in single step mode: <RET>=default, ?=Help
```

```
SP=3A PSW=20 C=0 A=00 B=12 DPTR=8000 P1=FF P3=F7
R0=00 R1=01 R2=00 R3=00 R4=00 R5=00 R6=00 R7=00
--> 8000: 75 81 30      MOV      SP, #30
```

```
SP=30 PSW=20 C=0 A=00 B=12 DPTR=8000 P1=FF P3=F7
R0=00 R1=01 R2=00 R3=00 R4=00 R5=00 R6=00 R7=00
--> 8003: 78 01      MOV      R0, #01
```

In diesem Mode sind die nachstehenden Kommandos verfügbar:

Single Step Commands:

<RET>	Print Status and execute the next instruction
<SPACE>	Execute next instruction (only one status lines)
`?`	on-line help
`R`	Print SFR
`H`	Hex dump of int. ram
`S`	Skip this instr
`A`	Change the Acc
`M`	back to Main-menu
`Q`	Quit to executing normally

Programm im EEPROM ablegen

Das Programm wird vom RAM ab Adresse 8000h in den EEPROM kopiert. Die Länge der zu schreibenden Bytes wird vor dem Programmieren des EEPROMs per Terminal abgefragt. Der Header muß vom Eeprom manuell abgezogen werden. Somit ergeben sich maximal folgende Programmgrößen:

- AT24C64 (8kB) $0x2000-3 = 0x1FFD$
- AT24C128 (16kB) $0x4000-3 = 0x3FFD$
- AT24CC256 (32kB) $0x8000-3 = 0x7FFD$

Die Ausgabe des Monitors ist nachstehend abgebildet.

```
>Loc=8000: (Version 1.50) Command>
  Write bytes from 8000h to eeprom
```

```
  EEPROM detected
  size: 1F00
```

```
  header: 03 1F 00 97
  .....
  ready...
```

Download aus dem EEPROM

Das Programm kann auch zum Test vom EEPROM in den RAM gelesen werden. In Anlehnung an das obige Beispiel wird nun das Programm vom Eeprom in den RAM zurückgelesen per Terminal-Befehl, da nach PowerOn ja keine Ausgaben mehr erfolgen. Während des Auslesen wird auf dem Terminal ein Fortschrittsbalken angezeigt.

```
>Loc=8000: (Version 1.50) Command>
  Upload bytes from eeprom to 8000h
```

```
  EEPROM detected
  header: 03 1F 00 97
  =====
```

Interrupttabelle des Monitors

```

start      equ 0000h
program   equ 8000h

ORG      start
rst:    lJMP   poweron      ;Monitor

ORG      start+3       ;ext int0
LJMP    program+3

ORG      start+0Bh     ;timer0
LJMP    program+0Bh

ORG      start+13h     ;ext int1
jnb     tcon.2,intr0 ;normal or debug ?
ljmp    program+13h  ;normal processing

intr0:   ajmp   step      ;step-debuging

ORG      start+1bh     ;timer1
ljmp    program+1bh

ORG      start+23h     ;serial
ljmp    program+23h

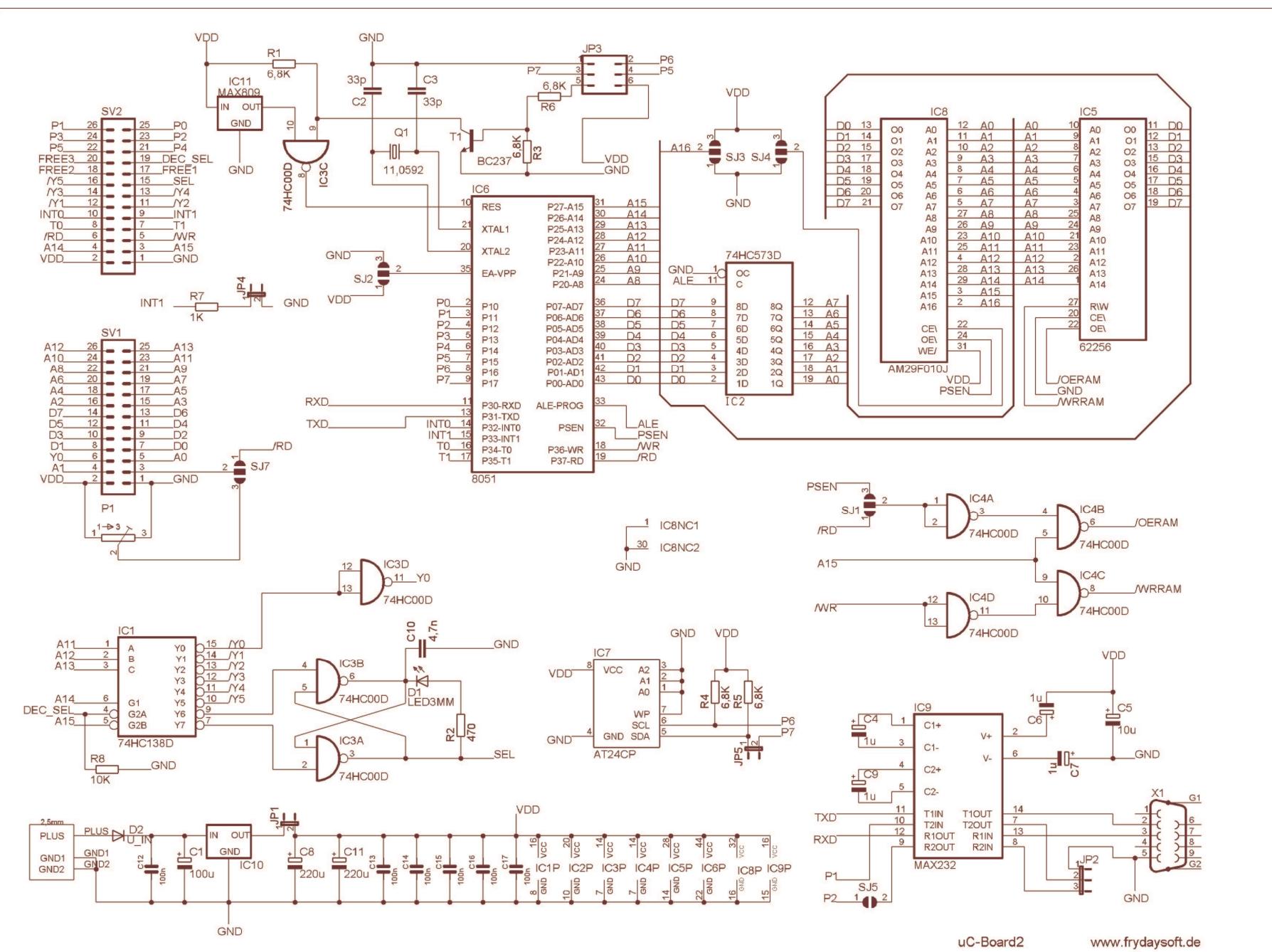
ORG      start+2bh     ;timer2 (8052 only)
ljmp    program+2bh

ORG      start+30h     ;the jump table
ajmp   cout           ;print accu
ajmp   cin            ;read keyboard, returned in accu
ajmp   phex           ;print accu as hex value
ajmp   phex16          ;print dptr as hex value
ajmp   pstr            ;print string pointed by dptr
ajmp   ghex             ;gets an 8-bit hex value from keyboard, returned in accu
ajmp   ghex16            ;gets an 16-bit hex value from keyboard, returned in dptr (C=1 --> escape pressed)
ajmp   esc              ;checks to see if <ESC> is waiting on serial port (C=0 --> escape not pressed)
ajmp   upper            ;convert accu to uppercase

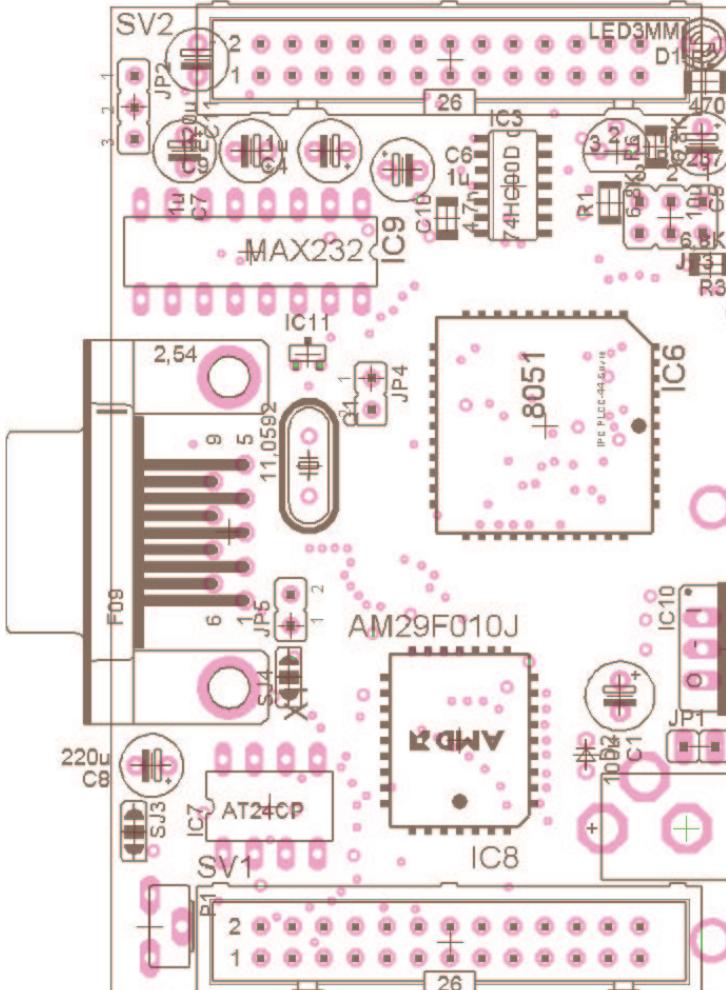
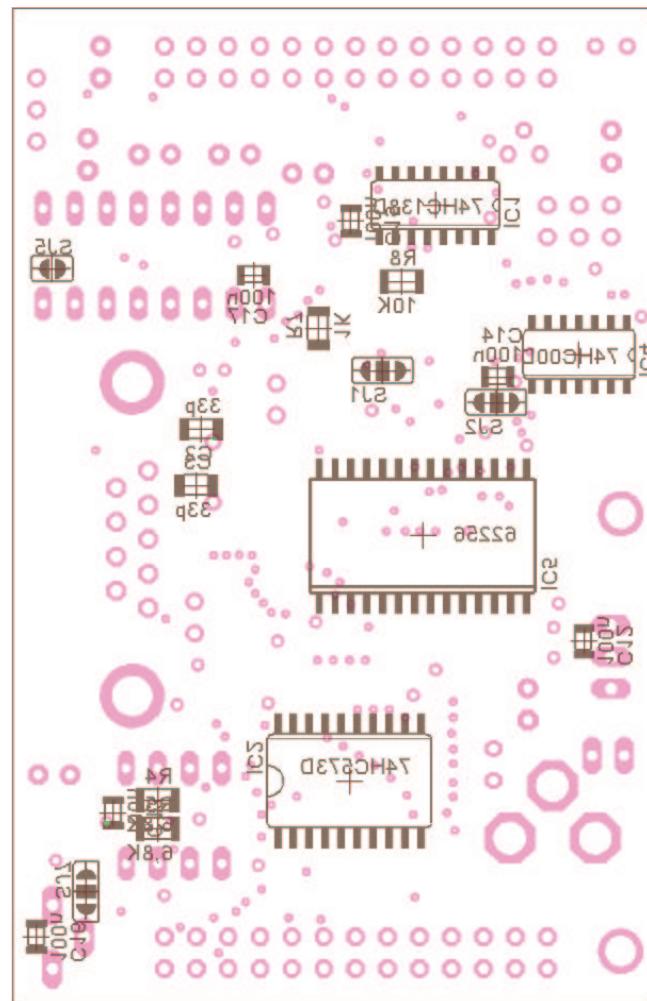
```

Hinweis: Die Routinen, die in der Jumptabelle gelistet sind, können von Userprogrammen genutzt werden und sind reentrant.

Schaltplan



Bestückung



C1	100u
C2	33p
C3	33p
C4	1u / 100n
C5	10u
C6	1u / 100n
C7	1u / 100n
C8	220u
C9	1u / 100n
C10	future use (4,7n)
C11	220u
C12	100n
C13	100n
C14	100n
C15	100n
C16	100n
C17	100n
D1	LED 3mm
D2	1N4148
IC1	74HC138
IC2	74HC573
IC3	74HC00
IC4	74HC00
IC5	SRAM 62256
IC6	CPU 89S52
IC7	Eeprom AT24CXX
IC8	future use (FLASH)
IC9	MAX232
IC10	VReg 78X05
IC11	Reset-IC Max809

JP1	JP1E	P1	poti 10K	R4	6,8K	SJ1	SJ2W	SV1	ML26
JP2	JP2E	Q1	11,0592 MHz bzw. 22,118MHz	R5	6,8K	SJ2	SJ2W	SV2	ML26
JP3	JP3Q			R6	6,8K	SJ3	SJ2W	T1	BC237
JP4	JP1E	R1	6,8K	R7	1K	SJ4	SJ2W	U\$1	Uin (7-12V)
JP5	JP1E	R2	470	R8	10K	SJ5	SJ	X1	SUB09
		R3	6,8K			SJ7	SJ2W		