Arm-Board (55800/63200)

Hardware

Spezifikation

Programmbeschreibung

Document Revision:	1.00
Document Status:	Released
Date last changed:	14.10.02
Written by:	Sven Freitag
Comments:	Software Rev. 1.00

Distribution:

Name	Internet	Tel.	Fax	E-Mail	O K
Sven Freitag	www.FrydaySoft.de			Svenfreitag@gmx.de Info@frydaysoft.de	

Document Revision History

Revision	Update	Author	Change / Chapter
1.00		Sven Freitag	Dokumentation zum Arm-Board

Related Documents

Document	Revision	Author/Company/Departm.
	1.00	S. Freitag

Inhaltsverzeichnis:

1. EINLEITUNG **4**

2. ALLGEMEINES ZUM ARM-BOARD **5**

3. BLOCK-SCHALTPLAN **6**

- PINBELEGUNG DER KONTAKTLEISTEN **7**

4. BOOT-LOADER **8**

5. ARM-BOARD (55800) **9**

6. ARM-BOARD (63200) **10**

7. BESTÜCKUNG **11**

8. JUMPEREINSTELLUNGEN **13**

9. ENTWICKLUNGSSOFTWARE **14**

ANHANG SCHALTPLAN 1/2 ..... **15**

ANHANG SCHALTPLAN 2/2 ..... **16**

1. Einleitung

Moderne Prozessoren im Embedded-Bereich werden immer komplexer und leistungsfähiger.
Es werden immer mehr Funktionseinheiten im Prozessor integriert, so daß auch die Programmierung dieser Prozessoren immer umfangreicher wird.
Hinzu kommt, das diese Prozessoren mit immer kleineren Spannungen arbeiten.
Dadurch ist die Anbindung an die „alte 5V-Technik“ nur noch mit Spannungswandlung möglich.

Um dennoch einen preiswerten Einstieg in diese Prozessorgeneration zu ermöglichen,
wurde dieses Arm-Board entwickelt.

Das Board unterstützt die High-performance 32-bit RISC-Prozessoren:

- AT91M63200
  - High-density 16-bit Instruction Set
  - 2K Bytes Internal RAM
  - Fully-programmable External Bus Interface (EBI)
  - Multi-processor Interface (MPI)
  - 8-channel Peripheral Data Controller
  - 8-level Priority, Individually Maskable, Vectored Interrupt Controller
  - 58 Programmable I/O Lines
  - 6-channel 16-bit Timer/Counter
  - 3 USARTs
  - Master/Slave SPI Interface
  - Programmable Watchdog Timer
  - Power Management Controller (PMC)
  - IEEE 1149.1 JTAG Boundary-scan on All Active Pins
  - Fully Static Operation: 0 Hz to 25 MHz (12 MHz at 1.8V)
  - 1.8V to 3.6V Core Operating Voltage Range
  - 2.7V to 5.5V I/O Operating Voltage Range

- AT91M55800
  - High-density 16-bit Instruction Set
  - 8K Bytes Internal SRAM
  - Fully-programmable External Bus Interface (EBI)
  - 8-level Priority, Individually Maskable, Vectored Interrupt Controller
  - 58 Programmable I/O Lines
  - 6-channel 16-bit Timer/Counter
  - 3 USARTs
  - Master/Slave SPI Interface
  - Programmable Watchdog Timer
  - 8-channel 10-bit ADC
  - 2-channel 10-bit DAC
  - Clock Generator with On-chip Main Oscillator and PLL for Multiplication
  - Real-time Clock with On-chip 32 kHz Oscillator
  - Battery Backup Operation and External Alarm
  - 8-channel Peripheral Data Controller for USARTs and SPIs
  - Advanced Power Management Controller (APMC)
  - IEEE 1149.1 JTAG Boundary-scan on all Digital Pins
  - Fully Static Operation: 0 Hz to 33 MHz
  - 2.7V to 3.6V Core Operating Range, 2.7V to 5.5V I/O Operating Range
  - 2.7V to 3.6V Analog Operating Range
  - 1.8V to 3.6V Backup Battery Operating Range
  - 2.7V to 3.6V Oscillator and PLL Operating Range

2. Allgemeines zum Arm-Board

Der große Vorteil dieses Boards ist die einfache Anbindung an die herkömmliche TTL-technik sowie das reichhaltige Angebot an freier Software. Dazu zählen z.B. die GNU-Compiler und das embedded Betriebssystem EcOS von Redhat.

Hier die wichtigsten Features des Arm-Boards:

- 5V I/O
- Two serial ports
- Reset push button
- 4 user-defined push buttons
- 12 LEDs
- 1M byte SRAM (upgradable)
- 2M bytes Flash (upgradable)
- 20-pin JTAG interface connector
- Connector for an alphanumeric LCD-display
- I/O expansion connector
- Memory expansion connector
- External Bus Interface (EBI) expansion connector

Bei der Softwareentwicklung für das Arm-Board kann das Armtool genutzt werden.
Das JTAG-Programm „Armtool“ ermöglicht das Laden eines Programmes in den SRAM des Boards.
Es können Programme auf dem Board gestartet werden. Das Tool kann auch Daten aus dem Board rücklesen. Das Armtool unterstützt somit die wichtigsten Kommandos zur Programmausführung.
Das Tool ist kommandoorientiert, somit kann es leicht in Makfiles eingebunden werden.
Der Datenaustausch vom PC aus erfolgt über die parallele Schnittstelle.
Eine einfache Treiberschaltung verbindet den LPT-Port mit dem JTAG-Port des Arm-Boards.
Da dieses Board kompatibel zu TTL ist, reicht auch eine direkte Verbindung aus.

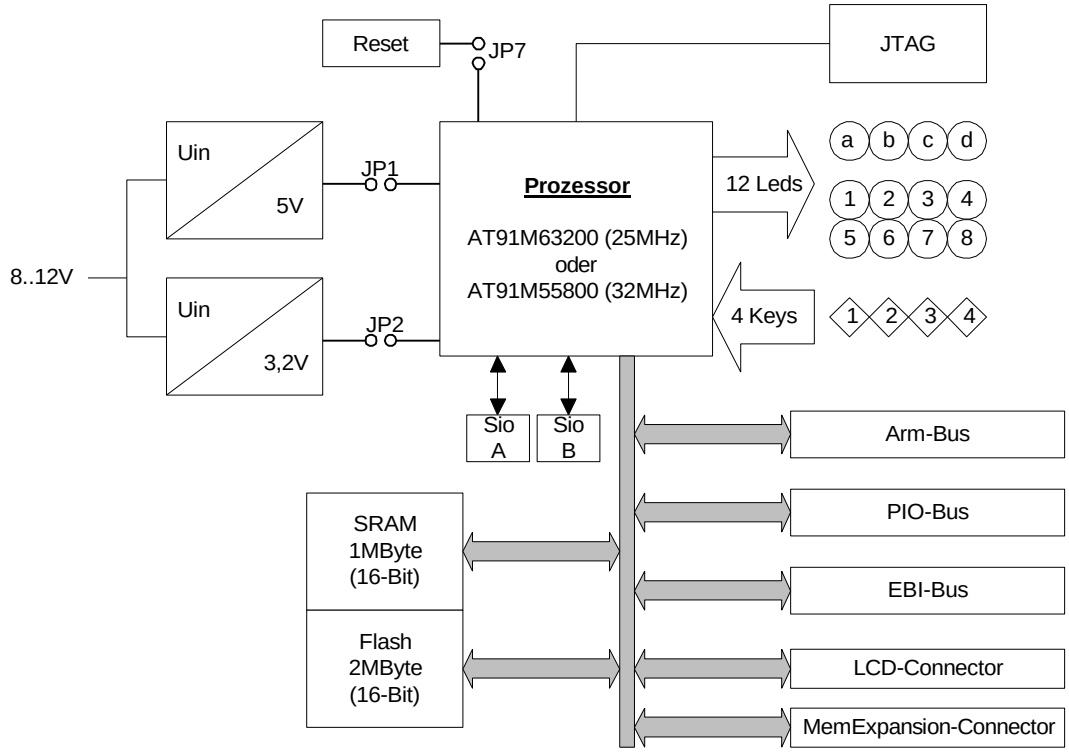
Die JTAG-Schnittstelle unterstützt aber auch Background Debugging (BDM).
Dazu sind jedoch professionelle und teuere Adapter nötig.
Ein gutes Preis/Leistungs-verhältis bietet z.B. der BDI2000 von Abatron.
Der BDI2000 kann mit dem freiem GNU-Debugger (z.B. GDB) genutzt werden.

Eine preiswertere Variante ist das Debuggen mit einem GDB-Stub. Hierfür kann ebenfalls das schon genannte EcOS genutzt werden.
Im EcOS (config-tool) ist das Anlegen der GDB-Stubs zu aktivieren. Die GDB-Stubs kommunizieren per serieller Schnittstelle mit dem GDB (Debugger).

Ist die Software mit den GDB-Stubs erstellt, sollte diese in den Flash des Boards abgelegt werden, so daß die weitere Softwareentwicklung über einen Debugger läuft. Dieser übernimmt dann auch den Download. Um die Software mit den Stubs aus dem Flash zu starten, ist noch ein Bootloader zu erstellen bzw. anzupassen. Der Bootloader mit der Debug-Software (GDB-Stubs) sollte als Flashimage erstellt werden.

Mit dem Armtool (+ kleinem Flashprogrammer im SRAM) kann dieses Flashimage in den Flash programmiert werden. Der Flashprogrammer selbst ist eine Softwareroutine, die das Image an einer definierten Adresse im SRAM erwartet. Der Flashprogrammer löscht das Flash und überträgt das Image vom SRAM in den Flash.
Nun ist die weitere Softwareentwicklung über den Debugger möglich.

3. Block-Schaltplan



8-er Block		Status-Block		Keys	
LED1	PB8	LEDa	PB26	S1	PA9
LED2	PB9	LEDb	PA27	S2	PB16
LED3	PB10	LEDc	PA28	S3	PB19
LED4	PB11	LEDd	PA29	S4	PB20
LED5	PB12				
LED6	PB13				
LED7	PB14				
LED8	PB15				

Memory-Expansion Connector (2x26 Pins):

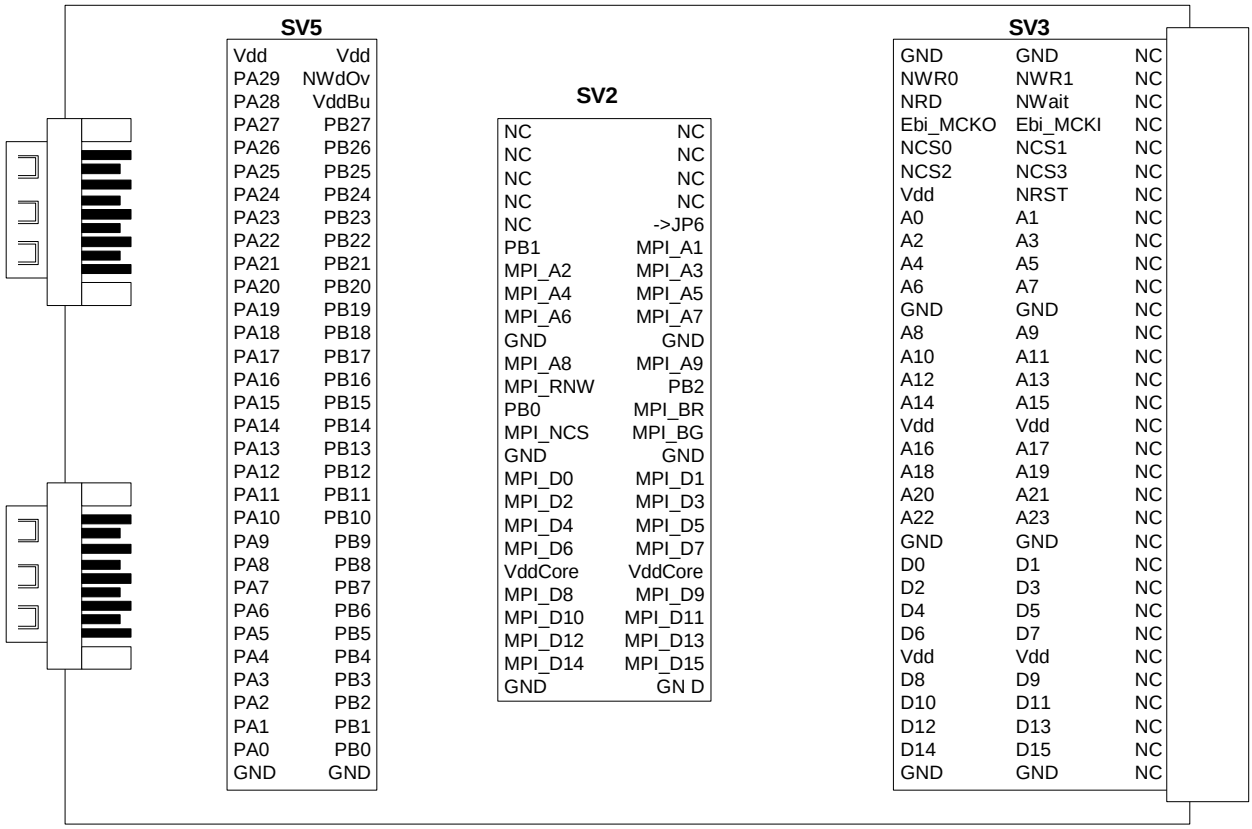
A0-A23	A20F (A20 or $\overline{A20}$ set by JP3)		NRD, NWR0, NWR1, NRST
D0-D15	NCS0, NCS1, NCS2	Vdd and GND	zusätzlich zwei freie Pins

LCD-Connector :

GND , Vdd, Kontrast/NRD (-> JP8), A1, A0, $\overline{NCS3}$, D0..D7 zusätzlich sechs freie Pins

Das Prozessorsystem kann optional über den ARM-Bus versorgt werden.
Dazu sind die Steckbrücken JP1(für Vdd), JP2 (für Vdd_Core) sowie
JP7 (Reset=NRST) zu entfernen.

Belegung EBI-63200:



SV5

Vdd	Vdd
PA29	NWdOv
PA28	VddBu
PA27	PB27
PA26	PB26
PA25	PB25
PA24	PB24
PA23	PB23
PA22	PB22
PA21	PB21
PA20	PB20
PA19	PB19
PA18	PB18
PA17	PB17
PA16	PB16
PA15	PB15
PA14	PB14
PA13	PB13
PA12	PB12
PA11	PB11
PA10	PB10
PA9	PB9
PA8	PB8
PA7	PB7
PA6	PB6
PA5	PB5
PA4	PB4
PA3	PB3
PA2	PB2
PA1	PB1
PA0	PB0
GND	GND

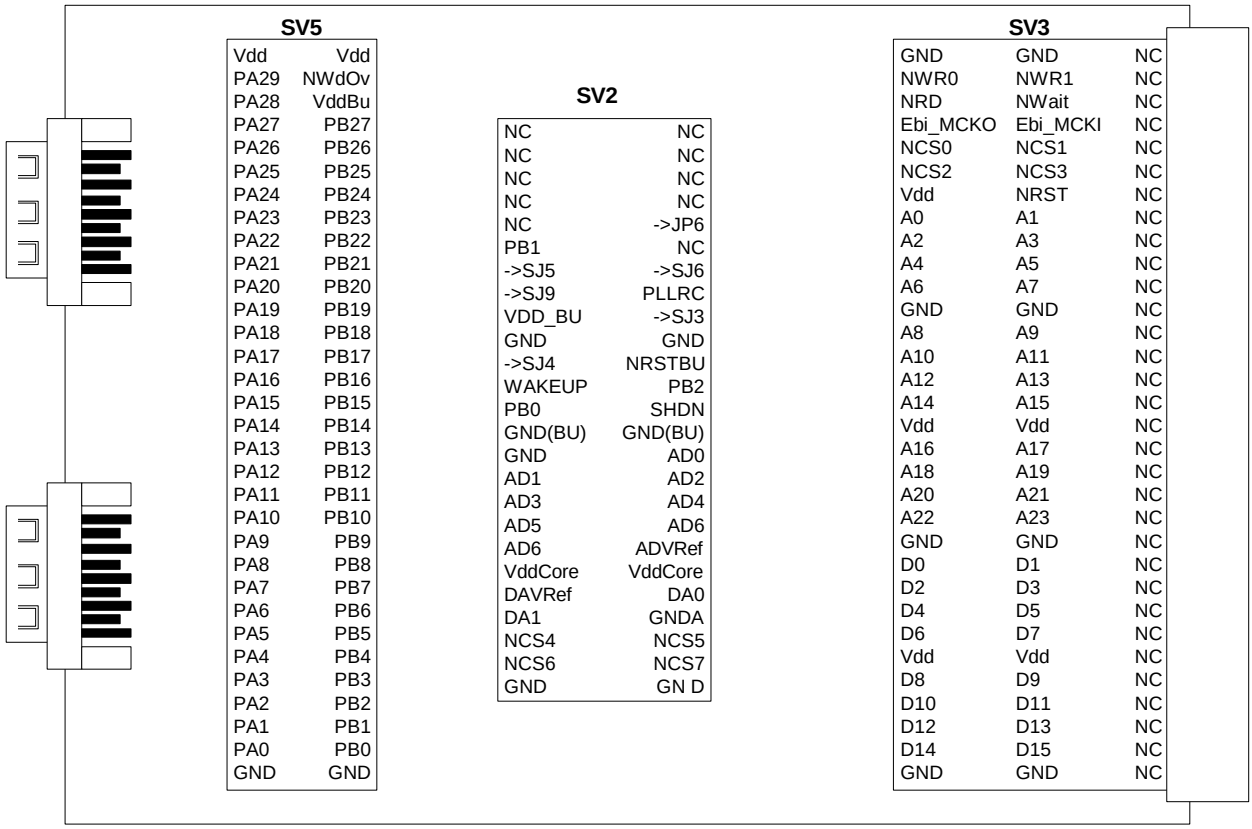
SV2

NC	NC
NC	NC
NC	NC
NC	NC
NC	->JP6
PB1	MPI_A1
MPI_A2	MPI_A3
MPI_A4	MPI_A5
MPI_A6	MPI_A7
GND	GND
MPI_A8	MPI_A9
MPI_RNW	PB2
PB0	MPI_BR
MPI_NCS	MPI_BG
GND	GND
MPI_D0	MPI_D1
MPI_D2	MPI_D3
MPI_D4	MPI_D5
MPI_D6	MPI_D7
VddCore	VddCore
MPI_D8	MPI_D9
MPI_D10	MPI_D11
MPI_D12	MPI_D13
MPI_D14	MPI_D15
GND	GN D

SV3

GND	GND	NC
NWR0	NWR1	NC
NRD	NWait	NC
Ebi_MCKO	Ebi_MCKI	NC
NCS0	NCS1	NC
NCS2	NCS3	NC
Vdd	NRST	NC
A0	A1	NC
A2	A3	NC
A4	A5	NC
A6	A7	NC
GND	GND	NC
A8	A9	NC
A10	A11	NC
A12	A13	NC
A14	A15	NC
Vdd	Vdd	NC
A16	A17	NC
A18	A19	NC
A20	A21	NC
A22	A23	NC
GND	GND	NC
D0	D1	NC
D2	D3	NC
D4	D5	NC
D6	D7	NC
Vdd	Vdd	NC
D8	D9	NC
D10	D11	NC
D12	D13	NC
D14	D15	NC
GND	GND	NC

Belegung EBI-55800:



SV5

Vdd	Vdd
PA29	NWdOv
PA28	VddBu
PA27	PB27
PA26	PB26
PA25	PB25
PA24	PB24
PA23	PB23
PA22	PB22
PA21	PB21
PA20	PB20
PA19	PB19
PA18	PB18
PA17	PB17
PA16	PB16
PA15	PB15
PA14	PB14
PA13	PB13
PA12	PB12
PA11	PB11
PA10	PB10
PA9	PB9
PA8	PB8
PA7	PB7
PA6	PB6
PA5	PB5
PA4	PB4
PA3	PB3
PA2	PB2
PA1	PB1
PA0	PB0
GND	GND

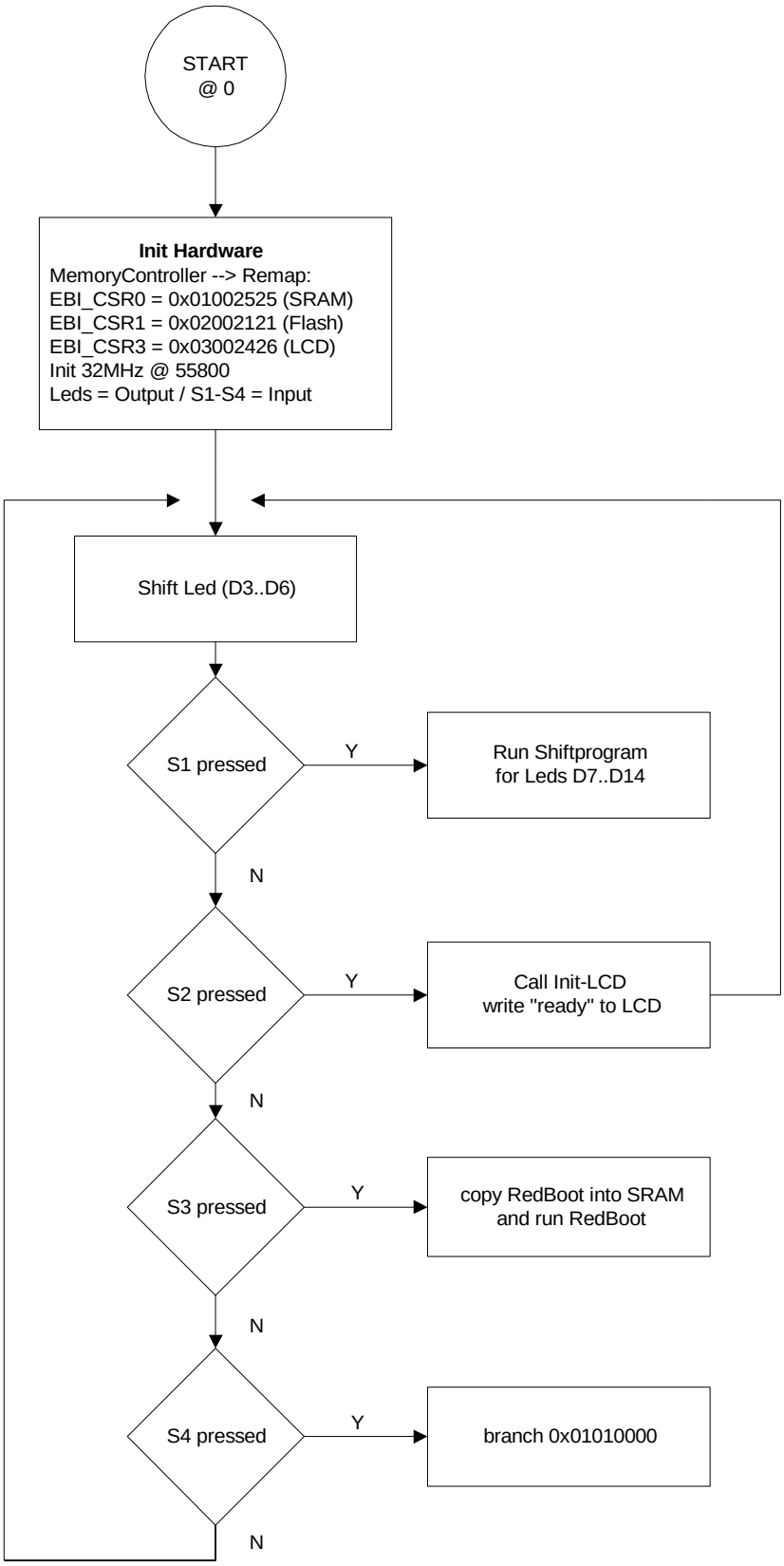
SV2

NC	NC
NC	NC
NC	NC
NC	NC
NC	->JP6
PB1	NC
->SJ5	->SJ6
->SJ9	PLLRC
VDD_BU	->SJ3
GND	GND
->SJ4	NRSTBU
WAKEUP	PB2
PB0	SHDN
GND(BU)	GND(BU)
GND	AD0
AD1	AD2
AD3	AD4
AD5	AD6
AD6	ADVRef
VddCore	VddCore
DAVRef	DA0
DA1	GNDA
NCS4	NCS5
NCS6	NCS7
GND	GN D

SV3

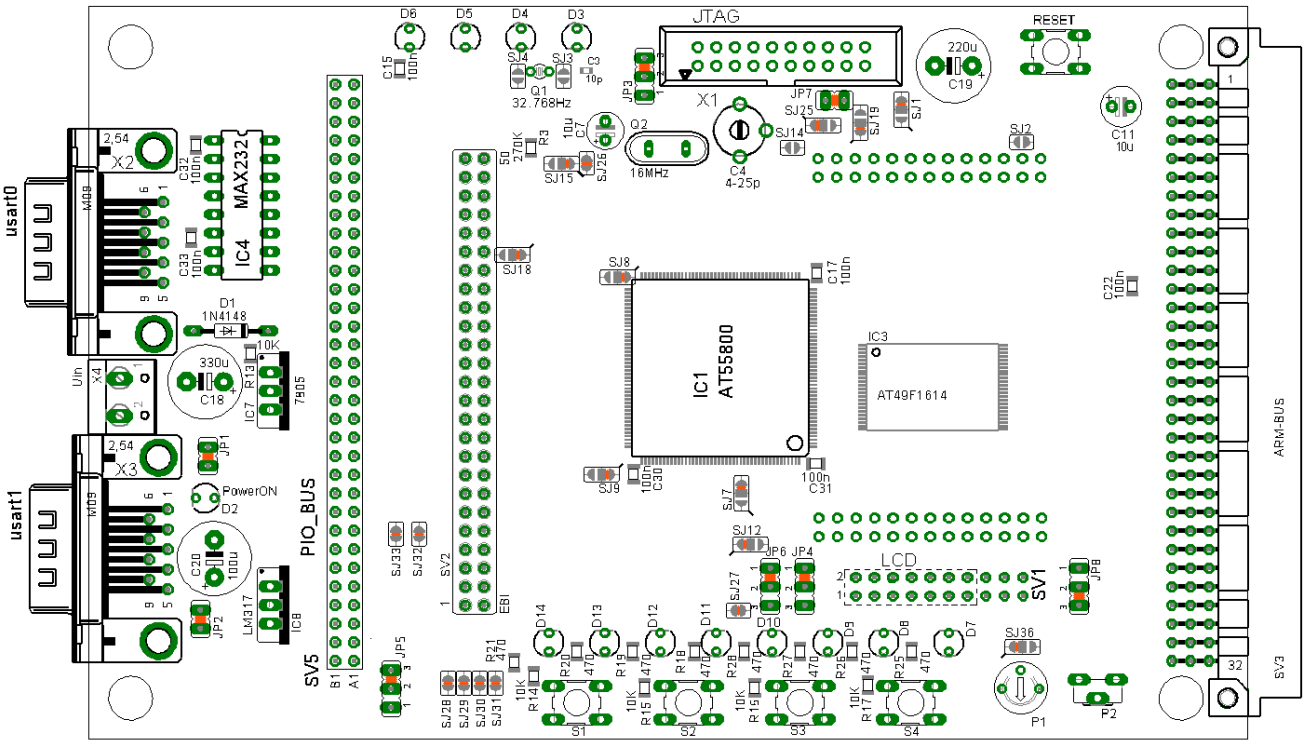
GND	GND	NC
NWR0	NWR1	NC
NRD	NWait	NC
Ebi_MCKO	Ebi_MCKI	NC
NCS0	NCS1	NC
NCS2	NCS3	NC
Vdd	NRST	NC
A0	A1	NC
A2	A3	NC
A4	A5	NC
A6	A7	NC
GND	GND	NC
A8	A9	NC
A10	A11	NC
A12	A13	NC
A14	A15	NC
Vdd	Vdd	NC
A16	A17	NC
A18	A19	NC
A20	A21	NC
A22	A23	NC
GND	GND	NC
D0	D1	NC
D2	D3	NC
D4	D5	NC
D6	D7	NC
Vdd	Vdd	NC
D8	D9	NC
D10	D11	NC
D12	D13	NC
D14	D15	NC
GND	GND	NC

4. Boot-Loader

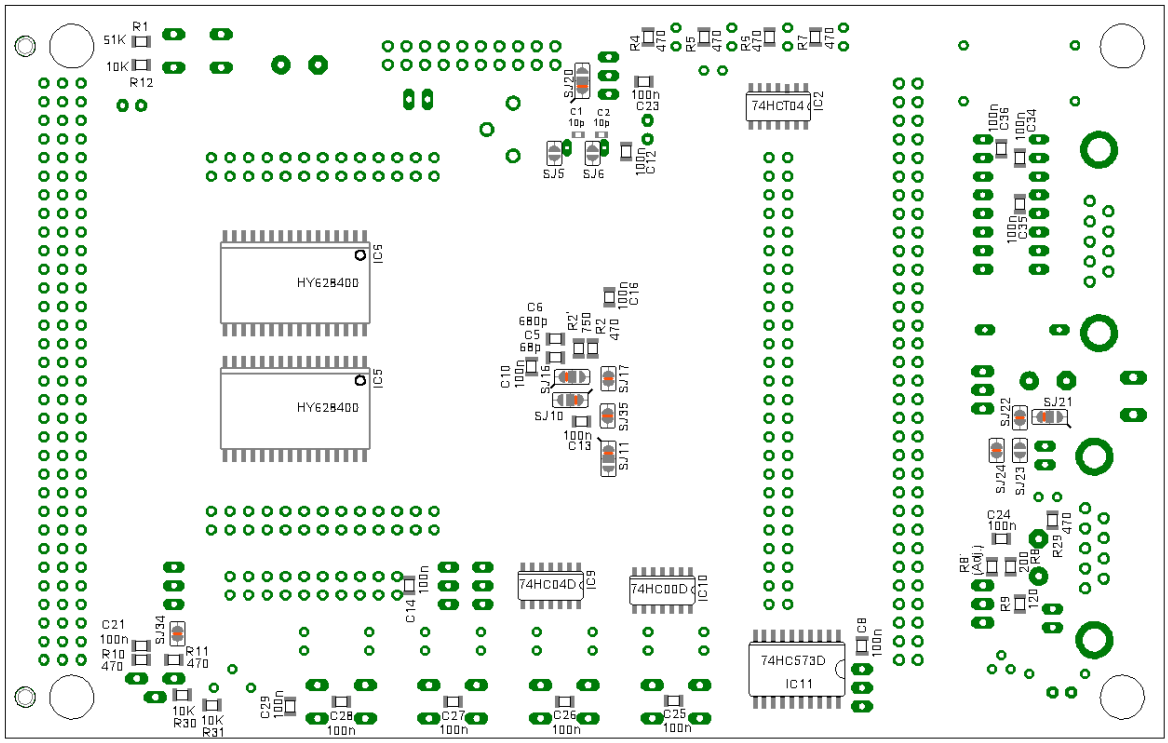


5. Arm-Board (55800)

Top:

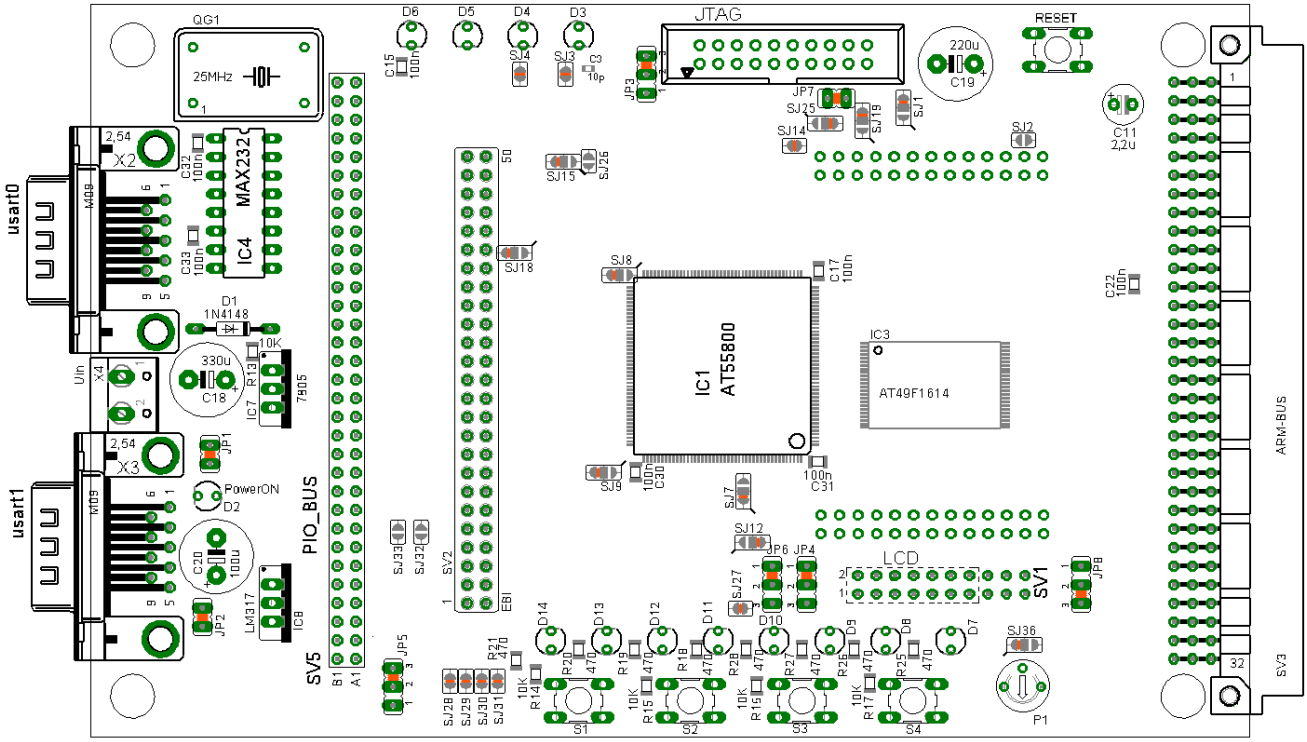


Bottom:

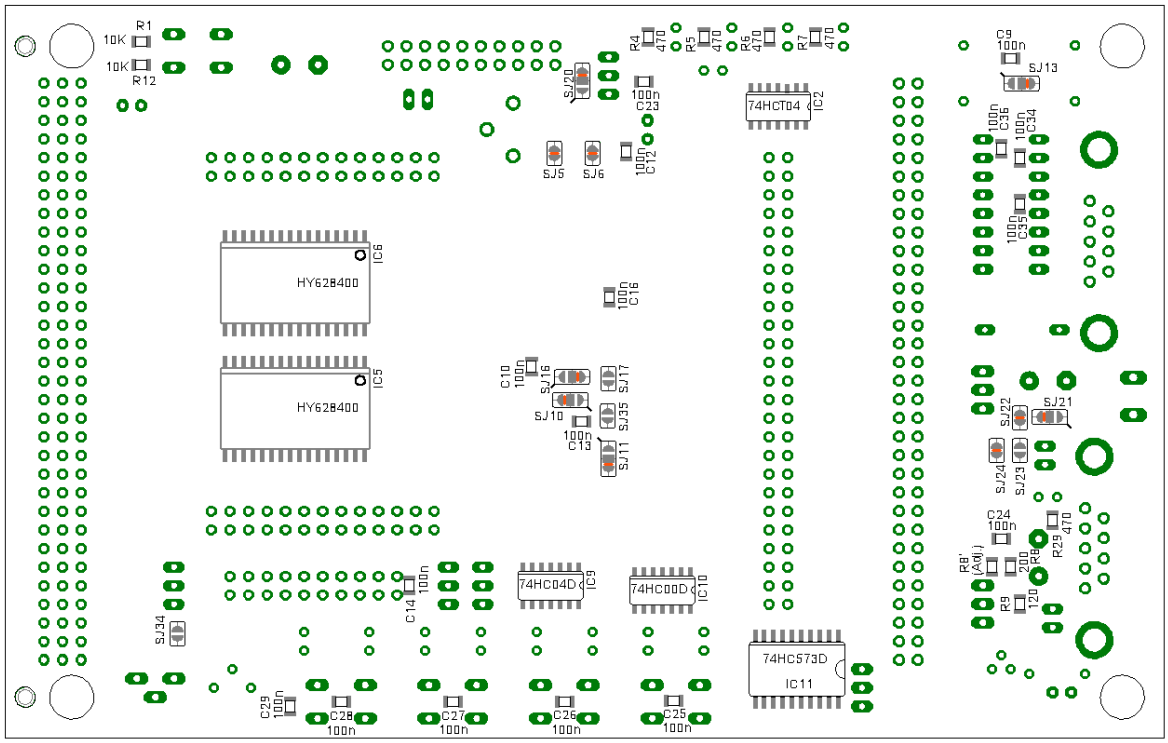


6. Arm-Board (63200)

Top:



Bottom:



7. Bestückung

Part	Value	Device
C1	10p	SMD
C2	10p	SMD
C3	10p	SMD
C4	4-25p	C-TRIMMER
C5	68p	1206
C6	680p	1206
C7	10u	CPOL
C8	100n	1206
C9	100n	1206
C10	100n	1206
C11	2,2 / 10u	CPOL
C12	100n	1206
C13	100n	1206
C14	100n	1206
C15	100n	1206
C16	100n	1206
C17	100n	1206
C18	330u	CPOL
C19	220u	CPOL
C20	100u	CPOL
C21	100n	1206
C22	100n	1206
C23	100n	1206
C24	100n	1206
C25	100n	1206
C26	100n	1206
C27	100n	1206
C28	100n	1206
C29	100n	1206
C30	100n	1206
C31	100n	1206
C32	100n	1206
C33	100n	1206
C34	100n	1206
C35	100n	1206
C36	100n	1206
Q1	32.768Hz	TC26V
Q2	16MHz	XTAL/S
QG1	25MHz	QG5860
T1	(BC547)	TO92
X1	JTAG	057-020-1
X2	Usart0	M09HP
X3	Usart1	M09HP
X4	Uin	W237-02P
PAD1		PAD
PAD2		PAD

Part	Value	Device
D1	1N4148	DO35-10
D2	PowerON	LED3MM
D3		LED3MM
D4		LED3MM
D5		LED3MM
D6		LED3MM
D7		LED3MM
D8		LED3MM
D9		LED3MM
D10		LED3MM
D11		LED3MM
D12		LED3MM
D13		LED3MM
D14		LED3MM
D15	1N4148	SMD
IC1	CPU	TQFP176
IC2	74HC04	SO14
IC3	AT49F1614	TSOP48
IC4	MAX232	DIL16
IC5	HY628400	SOP32
IC6	HY628400	SOP32
IC7	7805	TO220
IC8	LM317	TO220
IC9	74HC04	SO14
IC10	74HC00	SO14
IC11	74HC573	SO20
JP1	Jumper	JP1E
JP2	Jumper	JP1E
JP3	Jumper	JP2E
JP4	Jumper	JP2E
JP5	Jumper	JP2E
JP6	Jumper	JP2E
JP7	Jumper	JP1E
JP8	Jumper	JP2E
P1	10..50K	POTI
P2	10..50K	POTI
RESET	10-XX	B3F-10XX
SV1	LCD	ML20
SV2	EBI	FE25-2
SV3	ARM-BUS	MABC96L
SV4	MEM_EXT	
SV5	PIO_BUS	ML64
S1	10-XX	B3F-10XX
S2	10-XX	B3F-10XX
S3	10-XX	B3F-10XX
S4	10-XX	B3F-10XX

Part	Value	Device
R1	10K / 51K	1206
R2	470	1206
R2'	750	1206
R3	270K	1206
R4	470	1206
R5	470	1206
R6	470	1206
R7	470	1206
R8	200	1206
R8'	(Adj.)	1206
R9	120	1206
R10	470	1206
R11	470	1206
R12	10K	1206
R13	10K	1206
R14	10K	1206
R15	10K	1206
R16	10K	1206
R17	10K	1206
R18	470	1206
R19	470	1206
R20	470	1206
R21	470	1206
R22	(10K)	1206
R23	(10K)	1206
R24	(10R)	0204/7
R25	470	1206
R26	470	1206
R27	470	1206
R28	470	1206
R29	470	1206
R30	10K	1206
R31	10K	1206

Part	Value	Device
SJ1		SJ2W
SJ2	for 49F1604	SJ
SJ3		SJ
SJ4		SJ
SJ5		SJ
SJ6		SJ
SJ7		SJ2W
SJ8		SJ2W
SJ9		SJ2W
SJ10		SJ2W
SJ11		SJ2W
SJ12		SJ2W
SJ13		SJ2W
SJ14		SJ
SJ15		SJ2W
SJ16		SJ2W
SJ17		SJ
SJ18		SJ2W
SJ19		SJ2W
SJ20		SJ2W
SJ21		SJ2W
SJ22		SJ
SJ23		SJ
SJ24		SJ
SJ25		SJ2W
SJ26		SJ
SJ27		SJ
SJ28		SJ
SJ29		SJ
SJ30		SJ
SJ31		SJ
SJ32		SJ
SJ33		SJ
SJ34		SJ
SJ35		SJ
SJ36		SJ2W

8. Jumpereinstellungen

Jumper:

JP1: VDD close: use onboard supply (5V)	 open: use external supply
JP2: VDD_CORE close: use onboard supply (3,2V)	 open: use external supply
JP3: Flash-Page 1-2: inverting A20	 2-3: A20
JP4: Flash-Select 1-2: enabled on NCS0	 2-3: disabled
JP5: LED/Key IC11 1-2: disabled	 2-3: enabled
JP6: LCD Chip-Select 1-2: use NCS3	 2-3: use external CS
JP7: Reset close: use onboard reset	 open: use external reset
JP8: LCD-connector Pin3 1-2: NRD	 2-3: Poti for LCD-contrast

Lötjumper:

SJ1	Flash-Reset	1-2: reset with NRST	2-3: reset disabled
SJ2	Flash-IC	open: for AT49F1614	closed: for AT49F1604
SJ3	EBI_110	open/closed	
SJ4	EBI_111	open/closed	
SJ5	EBI_105	open/closed	
SJ6	EBI_106	open/closed	
SJ7	JTAGSEL	1-2: Boundary-Scan	2-3: JTAG-Debug
SJ8	CPU-Pin104	1-2: for 55800	2-3: for 63200
SJ9	CPU-Pin107	1-2: for 55800	2-3: for 63200
SJ10	CPU-Pin118	1-2: for 55800	2-3: for 63200
SJ11	CPU-Pin117	1-2: for 55800	2-3: for 63200
SJ12	CPU-Pin130	1-2: for 55800	2-3: for 63200
SJ13	VCC QG1	1-2: future use	2-3: default for 63200
SJ14	EBI_MCKO	open/closed	
SJ15	CPU-Pin109	1-2: for 55800	2-3: for 63200
SJ16	CPU-Pin113	1-2: for 55800	2-3: for 63200
SJ17	CPU-Pin113-116	closed: for 55800	open: for 63200
SJ18	CPU-Pin102	1-2: for 55800	2-3: for 63200
SJ19	CPU-Pin160	1-2: for 55800	2-3: for 63200
SJ20	MCKI	1-2: for 55800	2-3: for 63200
SJ21	Uin IC8	1-2: 5V	2-3: Uin
SJ22	IC7	default closed	
SJ23	IC7	future use	
SJ24	IC7	default closed	
SJ25	CPU-Pin103	1-2: for 55800	2-3: for 63200
SJ26	VDD_BU	closed for 55800	
SJ27	LCD-CS	open/closed	
SJ28	S1 PA9 (IRQ0)	open/closed	
SJ29	S2 PB16	open/closed	
SJ30	S3 PB19	open/closed	
SJ31	S4 PB20 (TIOA0)	open/closed	
SJ32	DAVREF	open/closed	
SJ33	ADVREF	open/closed	
SJ34	AD0	open/closed	
SJ35	AD1-DA1	open/closed	
SJ36	CSRAM	1-2: NCS1	2-3: disabled

9. Entwicklungssoftware

Hier folgen einige Hinweise zum Einrichten der Entwicklungssoftware (...unter Win32).

Um unter Windows die GNU-tools einsetzen zu können, muß zuerst eine Linuxumgebung installiert werden.
Z.B. unter http://www.cygwin.com/ kann man sich CYGWIN (=GNU+Cygnus+Windows) herunterladen.
CYGWIN ist eine Portierung von UNIX-tools auf die Windows-plattform.
Je nach Umfang umfaßt das CYGWIN -Setup 45 bis 160MB.

Unter http://www.ocdemon.com/OCDemonWindows.html kann Software heruntergeladen werden,
die das Debuggen per Wiggler, Reaven und mpDemon (ethernet/parallel/serial) unterstützt.
("UNIX" shell emulator product called Cygwin)
http://www.ocdemon.com/Windows98NT/CygwinInstallation.exe (22,2 MB)
(binutils 2.10, gcc 2.95.2,Insight 5.0,Arm7LibRemote)
http://www.ocdemon.com/Windows98NT/ArmElfInstallation.exe (14,6 MB)

Für den GDB kann unter http://sources.redhat.com/insight/ eine grafische Variante heruntergeladen werden.
✍ Insight is a graphical user interface to GDB, the GNU Debugger written in Tcl/Tk.

Benötigt man die Funktionalität eines Betriebssystemes, so empfielht sich der Einsatz von EcOS.

✍ eCos, the embedded Configurable operating system, is an open source real-time operating system for deeply embedded applications.

Unter http://sources.redhat.com/ecos/ findet man umfangreiche Informtionen zu dem Softwareprodukt von Redhat. Hier kann auch die Software heruntergeladen werden.
eCos 1.3.1 for Windows (eCos131.exe - 16MB)
eCos 1.3.1 for Linux (ecos-1.3.1.tar.gz - 12MB)

Per CVS kann die aktuellste Version von EcOS heruntergeladen werden. Der Download per CVS umfaßt etwa 58 MB.

Wenn mit EcOS unter Cygwin entwickelt wird, fehlen gegebenenfalls einige GNU-Tools wie z.B. cat.exe und tail.exe . Liegen einige Tools nicht vor, so kann das Übersetzen der automatisch generierten Files (ecosconfig new eb40 ✍ ecosconfig tree ✍ make) mit einer Fehlermeldung abbrechen,
die nicht sofort auf das Fehlen eines Tools schließen läßt.

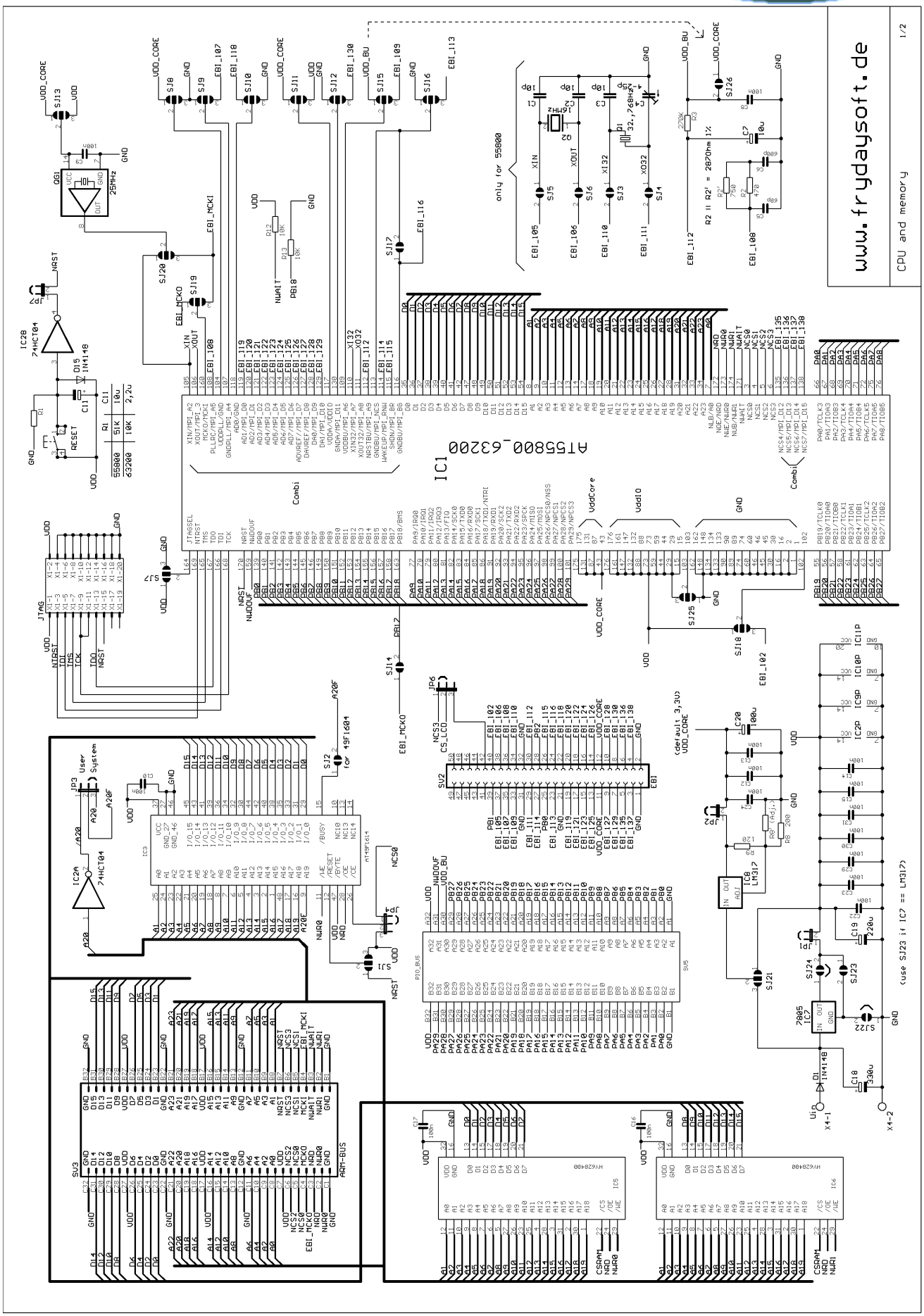
Unter http://neuro2.med.uni-magdeburg.de/~markgraf/mvs/cygwin/ oder http://unxutils.sourceforge.net/
können einige GNU-Tools (portiert nach Win32) heruntergeladen werden.

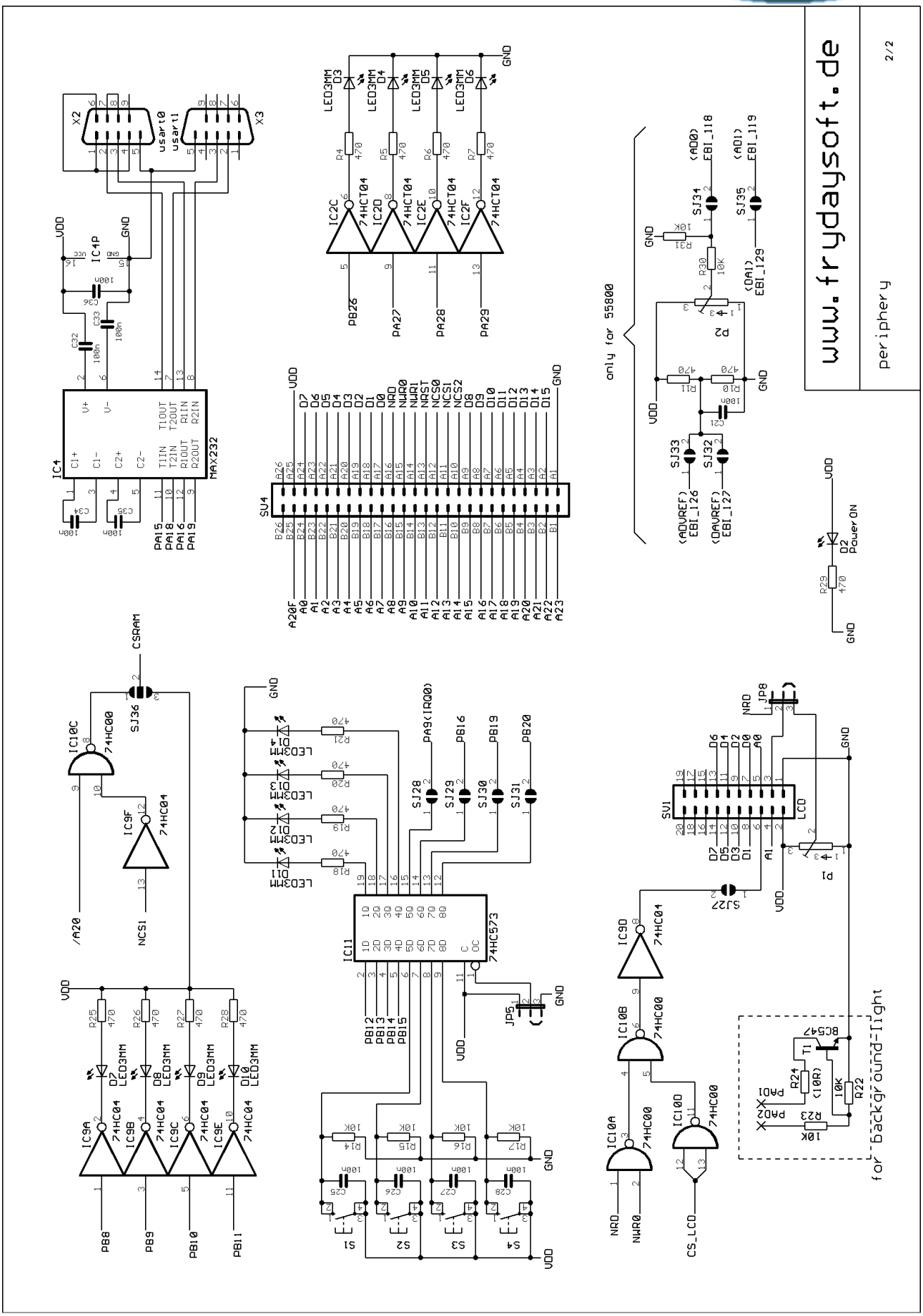
Einrichten der Umgebungsvariablen für EcOS:

Für EcOS müssen einige Einträge gesetzt werden, s.d. beim Übersetzen benötigte Files gefunden werden können. Dazu sollte im HOME-verzeichnis eine Datei namens ".profile" angelegt werden.
Dieses File wird beim Starten der Cygwin-Bash eingelesen. Das File sollte nachfolgende Einträge enthalten, ggf. angepaßt an die Verzeichnisstruktur des PC:

```
PATH=/cygdrive/c/Programme/Red_Hat/eCos/tools/bin:/cygdrive/c/cygwin/usr/local/bin:/usr/local/bin:$PATH
export PATH

ECOS_REPOSITORY=/cygdrive/c/Programme/Red_Hat/eCos/Packages
export ECOS_REPOSITORY
alias ll='ls -la'
```


www.frydaysoft.de
CPU and memory
1/2
AT55800_63200
IC1
only for 55800
ARM-BUS


www.frydaysoft.de
2/2
periphery
only for 55800
for background-light