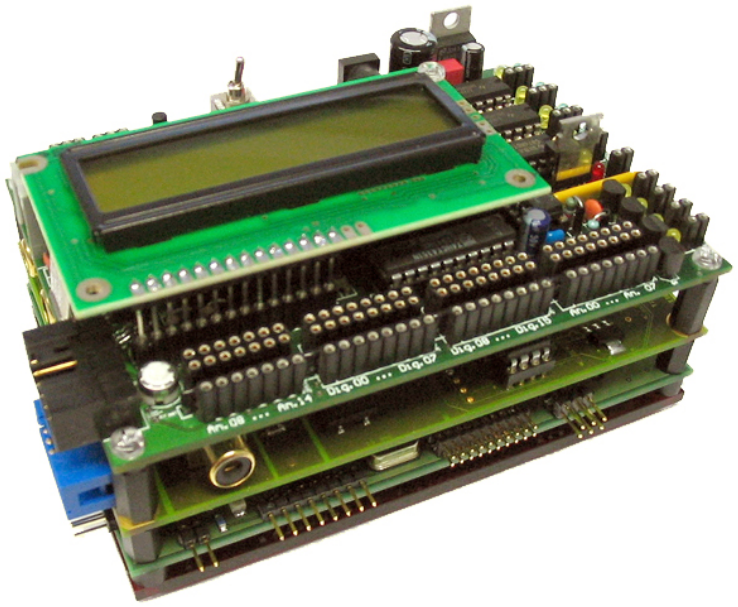
Getting Started with RVISON

Version 0.95



Fachhochschule Brandenburg, Fachbereich Informatik und Medien

Inhaltsverzeichnis

Einleitung

Herzlichen Glückwunsch zum Erwerb Ihres RVISION-Moduls.
Der RVISON ist eine Konfigurationsvariante der RCBUE-Architektur [rcu04]. Er ist in der Lage, Sensordaten inklusive Kameradaten zu verarbeiten und in Aktor-Signale umzuwandeln. Einsatzgebiet sind kleine, autonome Systeme (z.B. Roboter, solargestützte Applikationen oder Portables), die eine Bildverarbeitung benötigen.

Die vor Ihnen liegende Anleitung soll Ihnen den Einstieg in die Programmierung erleichtern - aber bitte denken Sie daran, das RVISION einen Forschungs- und Entwicklungsplattform ist und damit einigen Einarbeitungsaufwand verlangt. Wir möchten Sie bitten, diese Anleitung gut durchzulesen, da Sie viele Ihrer Fragen beantworten wird.
Für weitere Informationen möchten wir Sie auf die offizielle Homepage des RCUBE-Projekts aufmerksam machen. Sie finden Sie unter

`http://ots.fh-brandenburg.de/rcube.`

Sollten trotzdem Fragen offenbleiben, so wenden Sie sich bitte an Ihren Händler oder schreiben Sie uns eine eMail unter `rvision-support@fh-brandenburg.de`.

1 Übersicht über RVISION

1.1 Aufbau

RVISION besteht aus 3 Platinen:

- AKSEN-Board,
- CPU-Board und
- VIO-Board.

Das CPU-Board und das VIO-Board zusammen bilden das sogenannte VIO-Modul. Das VIO-Modul und das AKSEN-Board sind über einen CAN-Bus miteinander verbunden. Details zur Hardware für das AKSEN-Board sind in [KI-04], für das VIO-Board in [Sch02a] und [KS02] nachzulesen. Das CPU-Modul ist ein angepasstes LART-Board, dessen Dokumentation unter [lar04] zu finden ist. Die Erweiterungen sind minimal, so dass, wenn nicht anders erwähnt, von einem LART ausgegangen werden kann.
Unterschiede des CPU-Moduls zum Original-LART:

- 8MB statt 4MB Flash-Speicher
- CAN-Schnittstelle
- Formfaktor 12x8 cm statt 10x7.5 cm
- andere Stromversorgung

Der RVISION ist als Stack aufgebaut, wobei sich das AKSEN-Board oben befinden sollte, da es die meisten Peripherie-Anschlüsse sowie ein LCD-Display besitzt. Es ist ohne weiteres möglich, weitere Module mit dem RVISION zu verbinden, z.B. weitere AKSEN-Boards. Dazu muss nur ein CAN-Kabel mit entsprechend vielen Steckern vorliegen.

1.2 Stromversorgung

Die Stromversorgung des RVISION erfolgt zentral mit dem mitgelieferten Kabel über den Strom-Connector des AKSEN-Boards. Das CPU-Board erhält seine Stromversorgung über ein 2-adriges Adapter-Kabel von einem Anschluss des AKSEN-Boards. Die Spannung am CPU-Modul muss zwischen 5 und 9V betragen. Zum Anschluss des Adapterkabels gibt es drei Varianten:

1. **low current system**
   Anschluss an einen beliebigen analogen oder digitalen Port des AKSEN. Diese Ports stellen auf der inneren Seite stabilisierte 5V zur Verfügung, allerdings ist der Strom durch den Spannungsregler des AKSEN auf 1A begrenzt. Für den normalen Betrieb des RVISION genügt das, aber wenn viele aktive Sensoren am AKSEN angeschlossen sind, kann der Spannungswandler des AKSEN sehr heiß werden. Wenn also der Gesamtstrom eines RVISION in die Nähe von 1A gelangt, ist diese Variante nicht zu verwenden, da das CPU-Modul bei ungenügender Spannung einen Reset erfährt.

2. **mobile robot**
   Anschluss mit dem Adapterkabel an die ungeregelte Spannung am Servo-Port des AKSEN. Bei dieser Variante liegt die Versorgungsspannung des AKSEN direkt am CPU-Modul an und unterliegt nur der Strombegrenzung der Spannungsquelle. Dies ist die empfohlene Variante für mobile Roboter, allerdings sind die obengenannten Spannungsgrenzen des CPU-Modul einzuhalten.

3. **brain on demand**
   Anschluss an einen Motorport. Hierzu ist ein extra-Kabel notwendig und es besteht die Möglichkeit für das AKSEN das VIO-Modul bei Bedarf zuzuschalten. Die Motor-PWM ist dabei natürlich auf maximal zu stellen, damit des CPU-Modul nicht mit einer gepulsten Spannung versorgt wird. Für die Spannung gilt dasselbe wie in Variante 2.
   Achtung - bei dieser Variante ist auf die richtige Polung zu achten, deshalb unbedingt vor Anschluss ans CPU-Modul mit einem Meßgerät prüfen und die Lage des Steckers im Motorport fixieren. Bei Verpolung zerstört sich eine Schutzdiode des CPU-Moduls und/oder der Motortreiber des AKSEN.

Die Stromversorgung des VIO-Boards erfolgt über den High-Density-Connector zwischen VIO-Board und CPU-Board und ist damit systemintern.

1.3 Anschlüsse

Die Anschlüsse des AKSEN-Boards sind im [KI-04] detailiert beschrieben.

1.3.1 CPU-Modul

Das CPU-Board besitzt folgende Anschlüsse (siehe auch Abb. 1.1, die Pinbelegung finden Sie in den Schaltplänen im Anhang unter 5.2):

1. **Stromversorgung**
   4 poliger Stecker, verpolungssichere Belegung, außen Masse, innen Vcc

2. **CAN-Stecker**
   10 poliger Stecker, verpolungssicher durch Wanne. Bitte beachten Sie bei der Anfertigung eigener CAN-Kabel, dass der CAN-Bus am Ende des Kabels auf beiden Seiten durch

jeweils 120Ω terminiert werden sollte ([Esc02]). Beim RVISION wird nur CAN-Stecker des VIO-Boards verwendet.

3. **RESET-Pins**
   Durch Kurzschließen dieser beiden Pins wird das CPU-Modul zurückgesetzt

4. **JTAG-Interface**
   Dieses Interface dient zum erstmaligen Flashen des Urladers `blob` auf das CPU-Modul. Da sich bei Ihrem CPU-Modul eine aktuelle Version des `blob` im Flash befindet, wird dieses Interface im normalen Betrieb nicht benötigt. Es sei denn, der `blob` wurde beschädigt, dann benötigen Sie das auf der CD enthaltene Programm `jflash-amd`.

5. **2 × RS2322**
   Hier können Sie das beiliegende serielle Kabel anschliessen oder einen Adapter zu einem Bluetooth-Modul. Achtung, dieser Steckverbinder ist nicht verpolungssicher, deshalb ist beim Anschluß des Kabel auf die Kabelbeschriftung zu achten.

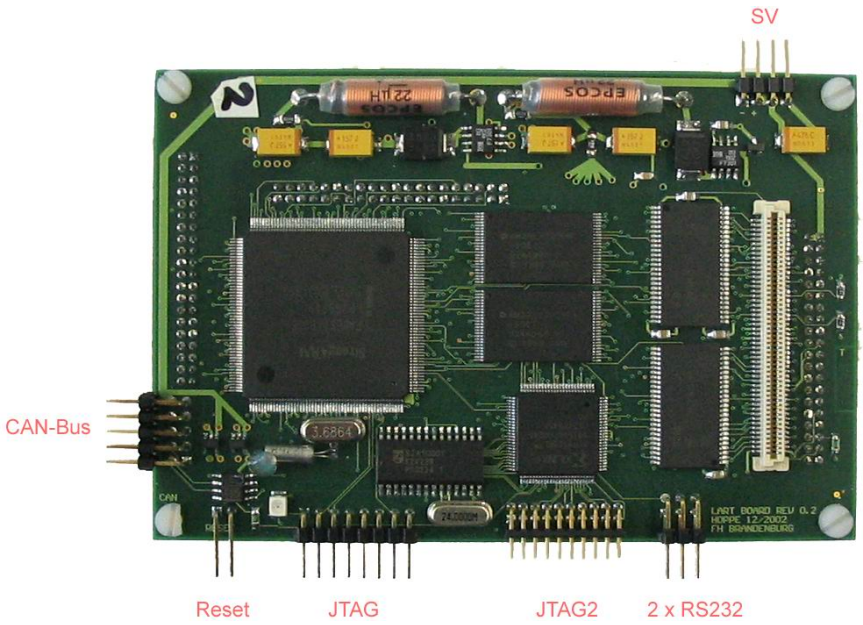


Abbildung 1.1: Anschlüsse des CPU-Moduls

1.3.2 VIO-Board

Das VIO-Board besitzt folgende Anschlüsse (siehe auch Abb. 1.2):

1. **CAN-Stecker**
   10 poliger Stecker, verpolungssicher durch Wanne. Bitte beachten Sie bei der Anfertigung eigener CAN-Kabel, dass der CAN-Bus am Ende des Kabels auf beiden Seiten durch jeweils 120Ω terminiert werden sollte.

2. **4 × VideoInput**
   Die 4 Cinch-Buchsen können zum Anschluss von PAL-Videoquellen verwendet werden. Beim Anschluss von mehreren Videoquellen gleichzeitig kann nur jeweils eine bearbeitet werden. Das Umschalten zwischen zwei Quellen benötigt ca. 40 Millisekunden. Standardmäßig ist die Quelle 0 gewählt.

3. **Video-Ausgang (Monitor)**
   An den Video-Ausgang kann ein Anzeigegerät angeschlossen werden. Dies empfiehlt sich insbesondere bei der Entwicklung eigener Bildverarbeitungs-Applikationen. Es können hier alle Geräte verwendet werden, die mit Standard-PAL-Signalen arbeiten können, z.B. TV-Geräte, VCR, Framegrabber, Videofunksender, Beamer oder Video-Eingänge von Grafikkarten.

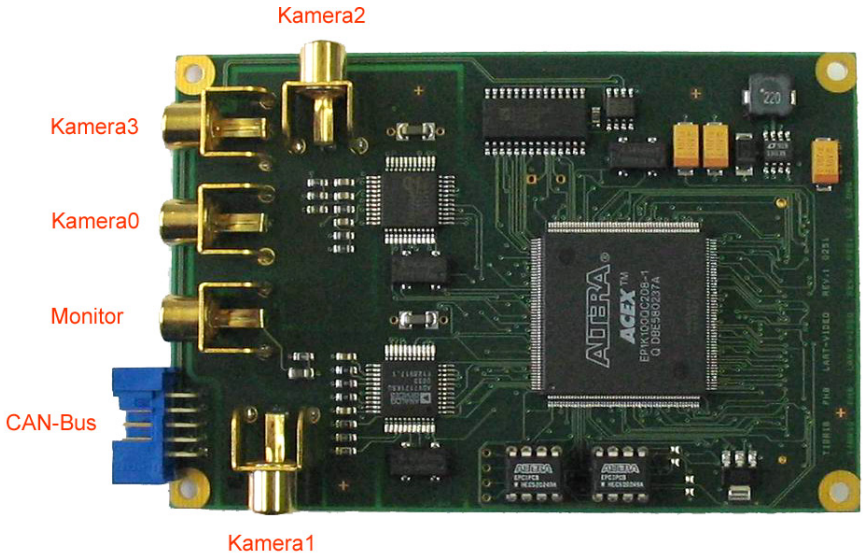


Abbildung 1.2: Anschlüsse des VIO-Boards

1.3.3 C-Cam8

Die im Lieferumfang enthaltene CMOS-Kamera ([Con04]) benötigt zur Stromversorgung $5V \pm 0.5V$. Daher ist der entsprechende Stecker an einen digitalen oder analogen Port des AKSEN-Boards anzuschließen. Der Cinchstecker kann zum Testen der Kamera an einen Monitor oder an einen Kameraeingang des VIO-Moduls angeschlossen werden. Der kameraseitige Stecker ist nicht verpolungssicher, hier ist die farbliche Markierung und die Beschriftung an der Kamerarückseite zu beachten:

- rot auf PWR
- schwarz auf GND

- weiß auf CVO

Damit ergibt sich folgende Standardverkabelung für den RVISION (Abb. 1.3):

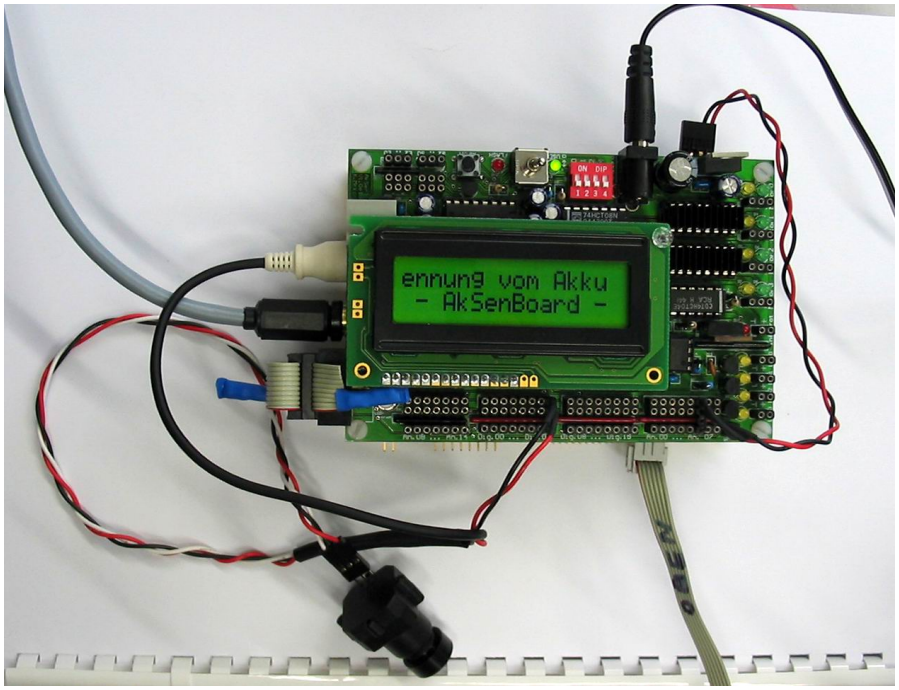


Abbildung 1.3: Standardverkabelung eines RVISION

2 Installation

2.1 Allgemeines

Applikationen für den RVISION bestehen in der Regel aus 2 Programmen: eines für das AKSEN-Board und eines für das CPU-Modul. Es handelt sich damit um eine verteilte Anwendung. Die Installation der AKSEN-Umgebung kann unter Windows oder Linux vorgenommen werden und ist im AKSEN-Handbuch beschrieben.

Wir empfehlen für RVISION-Projekte als Basis-System ein Debian-Woody, andere Linuxe sind ebenfalls möglich. Im weiteren Installationsprozeß gehen wir von einer funktionierenden AKSEN-Installation unter Linux aus, so daß ein RVISION-Projekt mit seinem Makefiles gleichzeitig die CPU-Modul- wie auch die AKSEN-Programme übersetzen kann. **Wenn Sie die AKSEN-Umgebung noch nicht installiert haben, unterbrechen Sie an dieser Stelle und installieren die AKSEN-Umgebung unter Linux, wie im AKSEN-Handbuch auf der AKSEN-CD beschrieben.**

Zu beachten ist, dass RVISION-Entwickler alle Entwicklungschritte einschließlich Programmtest als normaler Nutzer durchführen können, aber zum permanenten Speichern eines Programms auf dem CPU-Modul mittels Erstellen einer Ramdisk root-Rechte benutzen müssen!

2.2 Cross-Compiler

Die folgenden Schritte müssen als 'root' vorgenommen werden. Die komplette Crosscompilerumgebung findet sich auf RVISION-CD im Verzeichnis `setup` oder auf der LART-Seite im Netz unter `http://www.lart.tudelft.nl/lartware/compile-tools/cross-2.95.3.tar.bz2`

```
# mkdir /usr/local/arm
# cd /usr/local/arm
# wget http://www.lart.tudelft.nl/lartware/compile-tools/cross-2.95.3.tar.bz2
# bzip2 -d cross-2.95.3.tar.bz2
# tar -xvf cross-2.95.3.tar
```

Zur Bequemlichkeit wird der Programm-Pfad des Cross-Compilers zur PATH-Variable aller Nutzer hinzugefügt:

```
vmkilab1:~# echo "export PATH=/usr/local/arm/2.95.3/bin:\$PATH" >> /etc/profile
```

Testen des Compilers in einer Login-Shell oder einem neuen xterm:

```
boersch@vmkilab1:~$ arm-linux-gcc -v
Reading specs from /usr/local/arm/2.95.3/lib/gcc-lib/arm-linux/2.95.3/specs
gcc version 2.95.3 20010315 (release)
```

Setzen der Eigentümer des Cross-Compilers:

```
chown -R root /usr/local/arm
chgrp -R staff /usr/local/arm
```

2.3 Einrichten einer seriellen Verbindung zum CPU-Modul

Zur seriellen Verbindung mit dem CPU-Modul wird das Programm `minicom`, dazu bitte das entsprechende Programm-Paket (minicom, lsrzs) installieren. Das CPU-Modul verbindet sich auf seiner Schnittstelle 1 mit 9600, auf der Schnittstelle 2 mit 115000 Baud. Auf der RVSION-CD finden Sie unter `/setup/minicom` zwei vorbereitete Ressource-Dateien für minicom, die Sie in das Minicom-Verzeichnis (unter Debian /etc/minicom) kopieren und auf Ihre konkreten Schnittstellen anpassen können.

Tip: Beim Suchen nach den seriellen Schnittstellen hilft das Programm `setserial`
Root sollte die seriellen Schnittstellen freigeben:

```
# chmod 777 /dev/ttyS*
```

Test als beliebiger Nutzer: `minicom lart0` stellt Verbindung zum CPU-Modul her.

2.4 Einrichten der Sourcen und Beispiele

Auf der RVISION-CD befinden sich Beispielapplikationen, Kernelsourcen, Ramdisks und einige kleine Tools im Unterverzeichnis `/rcube` und gepackt in `/rcube.tar.gz`. Dieses Verzeichnis sollte als `/usr/local/rcube` ins System eingefügt werden. Dazu entpacken Sie die Datei `/rcube.tar.gz` in `/usr/local`:

```
# cd /usr/local
# tar -xvzf /pfad_zur_CD/rcube.tar.gz
```

Zur Bequemlichkeit wird der Programm-Pfad der Tools zur PATH-Variable aller Nutzer hinzugefügt:

```
vmkilab1:~# echo "export PATH=/usr/local/rcube/bin:\$PATH" >> /etc/profile
```

3 Applikationsentwicklung

3.1 Kompilieren der Beispiele

In `/usr/local/rcube/beispiele` finden Sie einige Beispielapplikationen. Die Bezeichnung weist auf verwendete Funktionalitäten hin, dabei bedeutet

- C = CPU-Board
- CA = CPU-Board und AKSEN
- CV = CPU,VIO
- CVA = CPU, VIO, AKSEN

Alle Beispiele lassen sich durch ein

```
# make clean && make
```

im beispiel-Verzeichnis übersetzen. Die Beispiele können als Rahmen für eigene Applikationen verwendet werden und sind als Projektverzeichnis frei beweglich, Sie können also z.B. das Beispiel `CV_xvtmmap` in Ihr Home-Verzeichnis kopieren und dort verändern. Zum Inhalt der Beispiele gehört jeweils eine kleine README-Datei zur Erklärung. Die Beispiele sind teilweise Studentenarbeiten und nicht immer perfekt. Eine Übersicht über die Beispiele der CD wird in Tabelle 3.1 gegeben.

3.2 Laden von Programmen

Der Upload von Programmen erfolgt zweckmäßigerwiese über die schnellere der beiden seriellen Schnittstellen, also `lart1`. Dazu öffnen Sie ein Minicom auf dieser Schnittstelle:

```
# minicom lart1

Welcome to minicom 1.83.1

OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Nov 21 2001, 00:35:58.

Press CTRL-A Z for help on special keys

lart #
```

Beispiel	Erläuterung
C_helloworld	Einfaches helloworld zum ersten Kompilieren und Download
CV_v4l2	Zugriff auf Bildeingabe und -Ausgabe (näher erläutert in [Sch02a])
CV_xvtmmap	Schnelle Bildverarbeitungdemo als Grundlage für eigene CV_Applikationen
C_sja1000	Erster Zugriff auf CAN-Controller SJA1000 (dazu Manual [sja00],[sja97]
CVA_helloworld	Applikationsrahmen für CVA_Applikationen (Doku dazu [Her04]
CVA_CatAndDog	Eine CVA_Anwendung, die in einem Overhead-Bild handgezeichnete Katzen und Hunde klassifiziert und das Ergebnis mit einem Servo-Motor anzeigt ([BHLM03])
CVA_R2D0	Komplexe CVA_Applikation, die den energieautonomen Roboter mithilfe einer behaviobasiert Architekutr steuert und per Kamera an seine Energie-station andocken laesst [Ahl03] menzel)

Tabelle 3.1: Beispiele der CD: Vorschlag zur Reihenfolge der Einarbeitung

Zum Senden von Files aus Minicom drücken Sie `Ctrl-A S` und wählen `zmodem`. In der folgenden Liste könnnen Sie mit den Cursor-Tasten navigieren und mit 2-maligem `Space` die Verzeichnisse wechseln. Wählen Sie mit `Space` die auf das CPU-Modul zu ladende Datei und drücken Sie `ENTER`. Die heruntergeladenen Files befinden sich nur im RAM des CPU-Boards, sind also beim nächsten Reset wieder verschwunden.

Tip: Es lassen sich nur Dateien laden, die nicht auf dem CPU-Board vorhanden sind, also ggf. vorher löschen.

3.3 Arbeiten mit der Ramdisk

Auf Ihrem CPU-Board befindet sich eine Ramdisk im Flash-Speicher. Wenn Programme permanent auf dem Modul verfügbar sein sollen, müssen diese in die Ramdisk aufgenommen werden. Sie finden die Ramdisks unter `/usr/local/rcube/ramdisk`. Hier befinden sich 2 Pfade: `src` und `image`. In `src` liegen die ausgepackten Verzeichnisbäume, in `image` die Ramdisks fertig zum Upload auf das CPU-Modul.

3.3.1 Erstellen und Auspacken

Zur Konvertierung von `src` nach `image` und umgekehrt können die Scripte `erd` ('Extract Ramdisk') und `crd` ('Create Ramdisk') genutzt werden (siehe Abb. 3.1). Diese sind weitgehend intuitiv bedienbar. Die beiden Programme müssen als Nutzer root verwendet werden, da der `src`-Zweig einer Ramdisk naturgemäß viele Files enthält, die root gehören.

Sie können also eine bestehende Ramdisk mit `erd` auspacken, in den `src`-Zweig dieser Ramdisk Files hineinkopieren, dann die Ramdisk mit `crd` packen. Details dieses Prozesses finden

sich in den Scripten und unter [WW02]

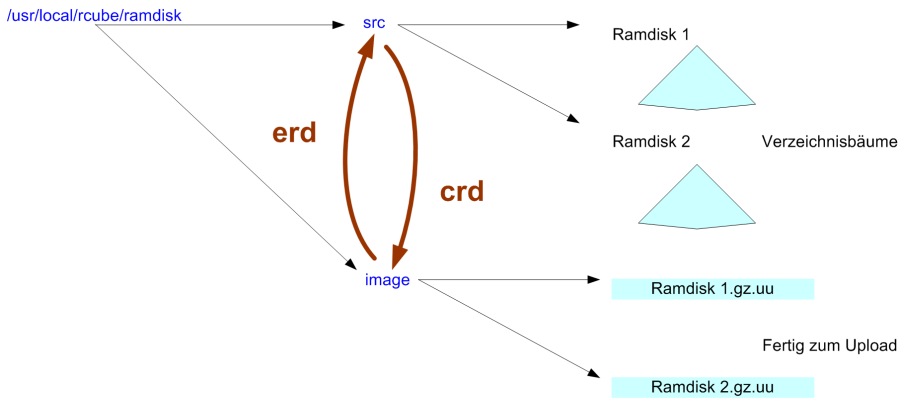


Abbildung 3.1: Arbeiten mit Ramdisks

3.3.2 Upload von Ramdisk und Kernel

Zum Laden einer Ramdisk auf das CPU-Modul wird das Tool `lartload` benutzt, das ausführbare Programm sollte bei korrekter Installation im Pfad liegen, die Quelltexte befinden sich auf der CD.

Beispiel zum Flashen einer Ramdisk:

```
lartload -rf /usr/local/rcube/ramdisk/image/rvision-1.0.gz.uu
```

Mit diesem Tool lassen sich ebenfalls neue Kernel flashen. Nähere Informationen zum Kompilieren eigener Kernel finden Sie in [WW02] und [Sch02b].

3.4 Automatischer Start

Beim Booten des CPU-Moduls werden standardmäßig zwei Login-Prozesse `getty` auf den seriellen Schnittstellen gestartet. Wenn einer dieser Prozesse durch ein eigenes Programm ersetzt wird, so wird bei jedem Booten dieses Applikationsprogramm gestartet. Zusätzlich wird das Programm bei jedem Programm-Absturz sofort neu gestartet.

Vorgehensweise: Wir wollen das Programm `/usr/bin/robotest` automatisch starten lassen. Dazu ist auf der Ramdisk in der Datei `/etc/inittab` die Zeile

```
null::respawn:/sbin/getty -L ttySA1 115200 vt100 -n
```

zu ersetzen durch

```
null::respawn:/sbin/getty -L ttySA1 115200 vt100 -n -l /bin/logmein
```

Als zweites ist ein Shell-Script `/bin/logmein` mit folgendem Inhalt zu erstellen:

```
#!/bin/sh
echo -e "Welcome to Larting"
/bin/sh  -c /usr/bin/robotest
```

Das Script als ausführbar kennzeichnen:

```
chmod 755 /bin/logmein
```

Dieser Ansatz wurd beispielsweise in der Ramdisk CatDog umgesetzt.

3.5 Qt-Bibliothek

Durch die Kompatibilität des implementierten Framebuffers ist es möglich, Qt-Programme für das CPU-Board zu programmieren. Qt ist eine C++-GUI ähnlich Motif. Einzelheiten und Beispielprogramme (z.B. `zollstock`) sind in [Sch02b] und auf der CD zu finden.

4 Weiterentwicklung RVISION

4.1 Kernel

Normalerweise ist es nicht notwendig, neue Kernel zu kompilieren. Falls doch, befinden sich die Sourcen des verwendeten Kernels unter `/usr/local/rcube/kernel/src`. Die von Frank Schwanz geschrieben Kernelmodule zur Ansteuerung des VIO-Boards ([Sch02a], citelartvio-soft) finden Sie in der CD zu seiner Diplomarbeit auf der RVISION-CD unter `/doc/lartvio`.

4.2 blob

Der blob unterscheidet sich vom blob der LART-Webseite durch einen veränderten Zugriff auf den Flash-Speicher, da das CPU-Board andere Flash-Bausteine verwendet als das LART. Sie finden den Quelltext des blob auf der CD.

4.3 Persistente Daten

In der aktuellen Version des RVISION geht erworbenes Erfahrungswissen der Applikation bei einem Reset verloren. Deshalb wird momentan ein Erweiterungsboard für die RCUBE-Architektur entwickelt, auf dem Applikationsdaten über einen Stromausfall hinaus in CF-Karten abgelegt werden können.

Probleme

1. In sehr seltenen Fällen benötigt ein RVISION nach längerer stromloser Periode eine Warmlaufzeit von einigen Minuten. Dies äußert sich in einem Einfrieren beim Zugriff auf das /dev/video. Dieses Phänomen wurde sehr selten beobachtet und ist im Moment nicht erklärbar.

5 Anhang

5.1 Inhalt RVISION-CD

Die CD enthält die komplette Software, die zum Programmieren des CPU-Moduls und des VIO-Boards benötigt wird.

Im Folgenden ist der Inhalt der CD kurz beschrieben:

`doc`	Handbuch
`doc/r2d0`	Beschreibung des energieautonomen Roboters R2D0
`doc/r2d1`	Studentenarbeit: ein Applikationsrahmen für typische CVA_Applikationen
`doc/vio-board`	Dokumentation des VIO-Boards (LARTVIO)
`doc/sja1000`	Dokumentation des CAN-Controlles SJA1000
`setup`	Cross-Compiler für RVISION für Linux (x86)
`setup/minicom`	Minicom-Profile
`rcube`	Dieses Verzeichnis sollte nach `/usr/local/` kopiert werden oder die Datei `rcube.tar.gz` dorthin auspacken
`rcube/bin`	Binäre Tools (erd, crd, lartload etc.)
`rcube/ramdisk`	Verzeichnis für Bäume und Images der Ramdisks
`rcube/beispiele`	Beispiel-Applikationen
`rcube/kernel`	Verzeichnis für Kernel-Sourcen und Images
`rcube/tools`	Quelltexte einiger Tools

5.2 Schaltplan CPU-Board

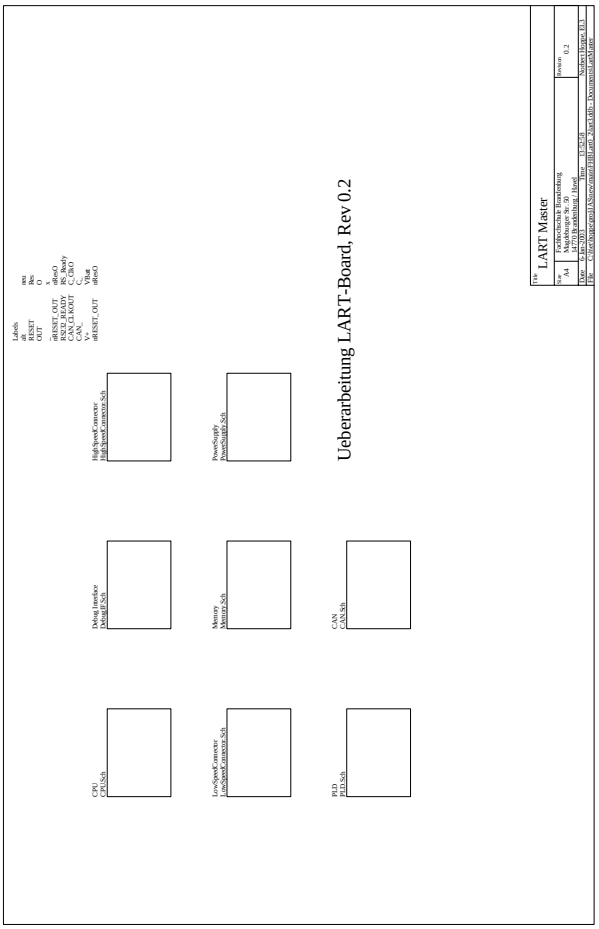


Abbildung 5.1: Schaltplan des CPU-Moduls 1/9

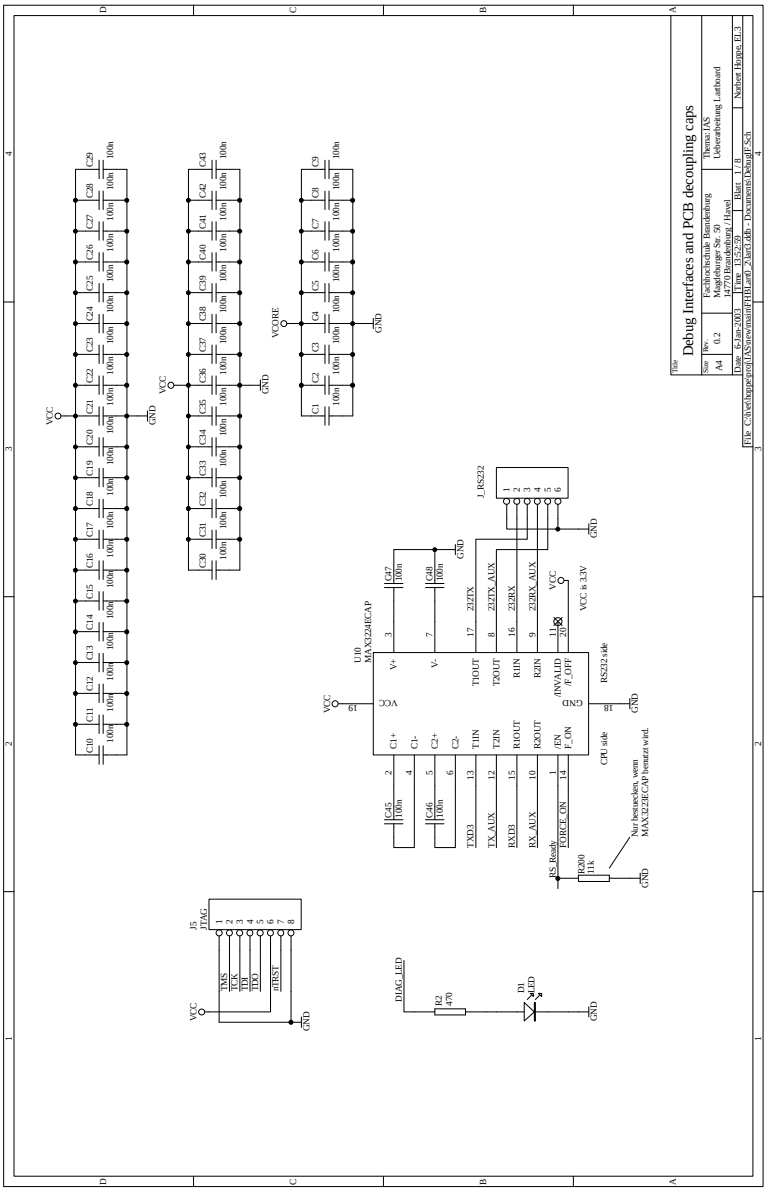


Abbildung 5.2: Schaltplan des CPU-Moduls 2/9

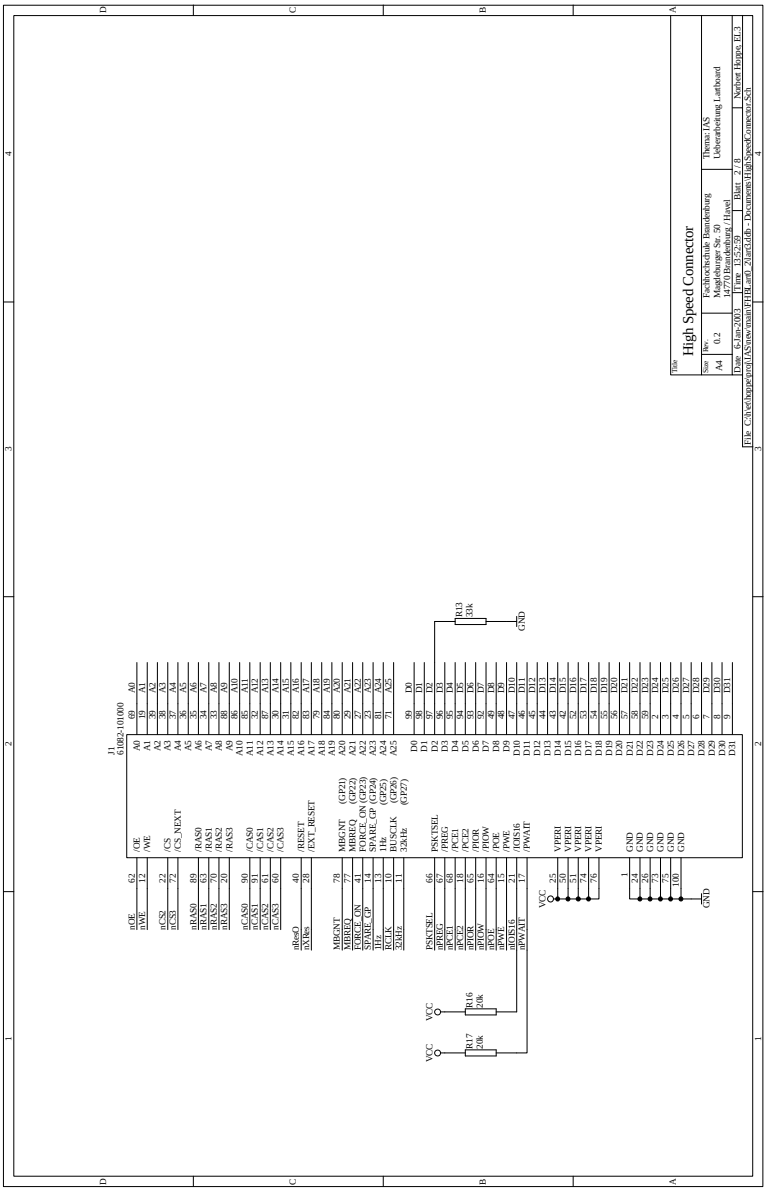


Abbildung 5.3: Schaltplan des CPU-Moduls 3/9

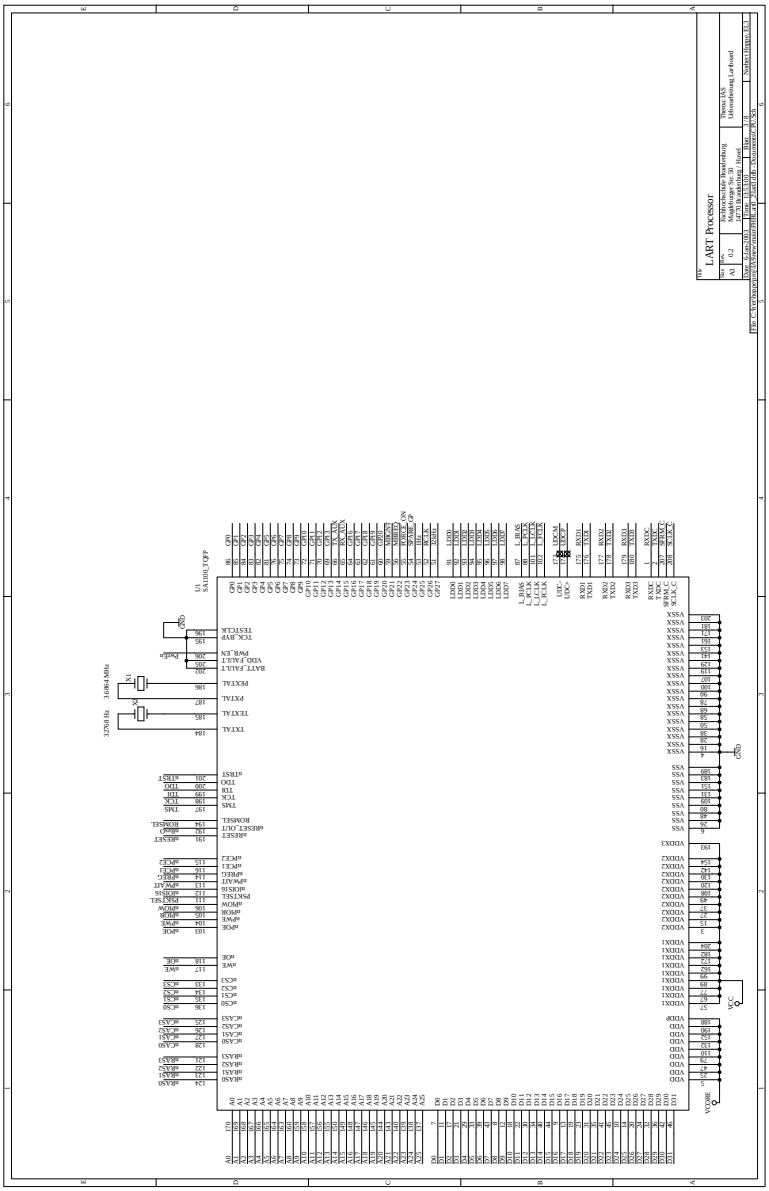


Abbildung 5.4: Schaltplan des CPU-Moduls 4/9

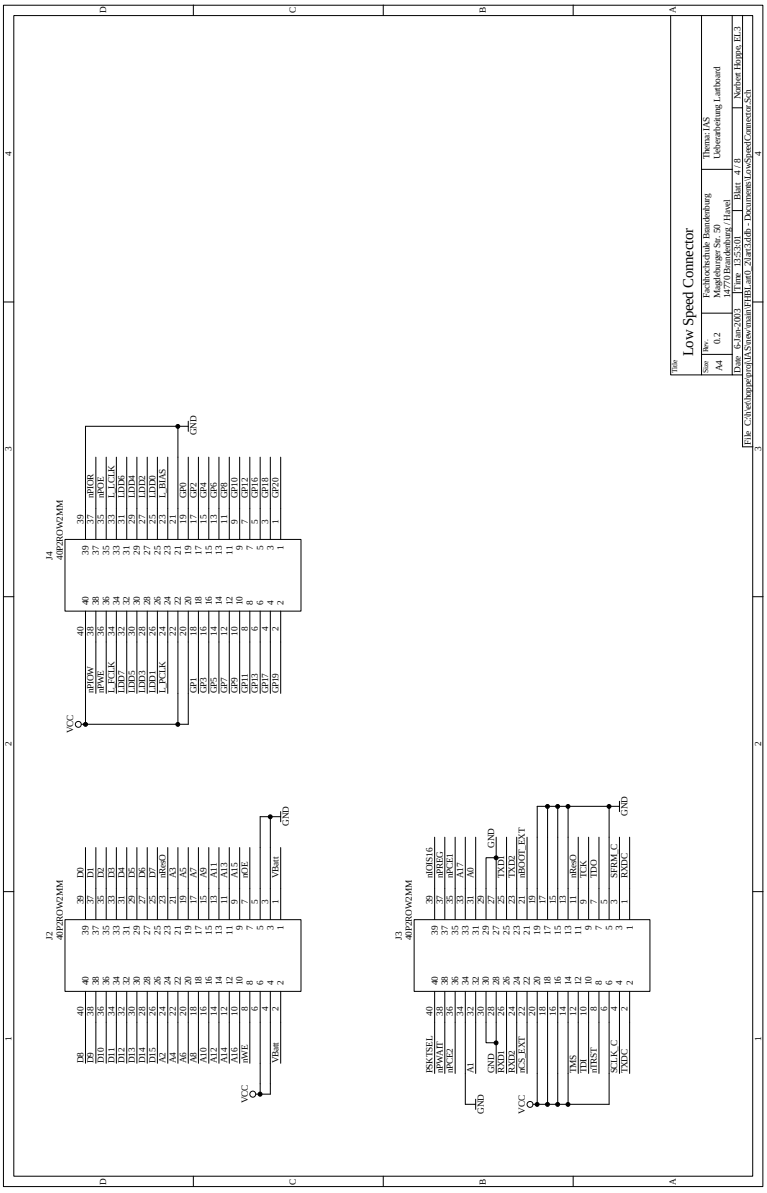


Abbildung 5.5: Schaltplan des CPU-Moduls 5/9

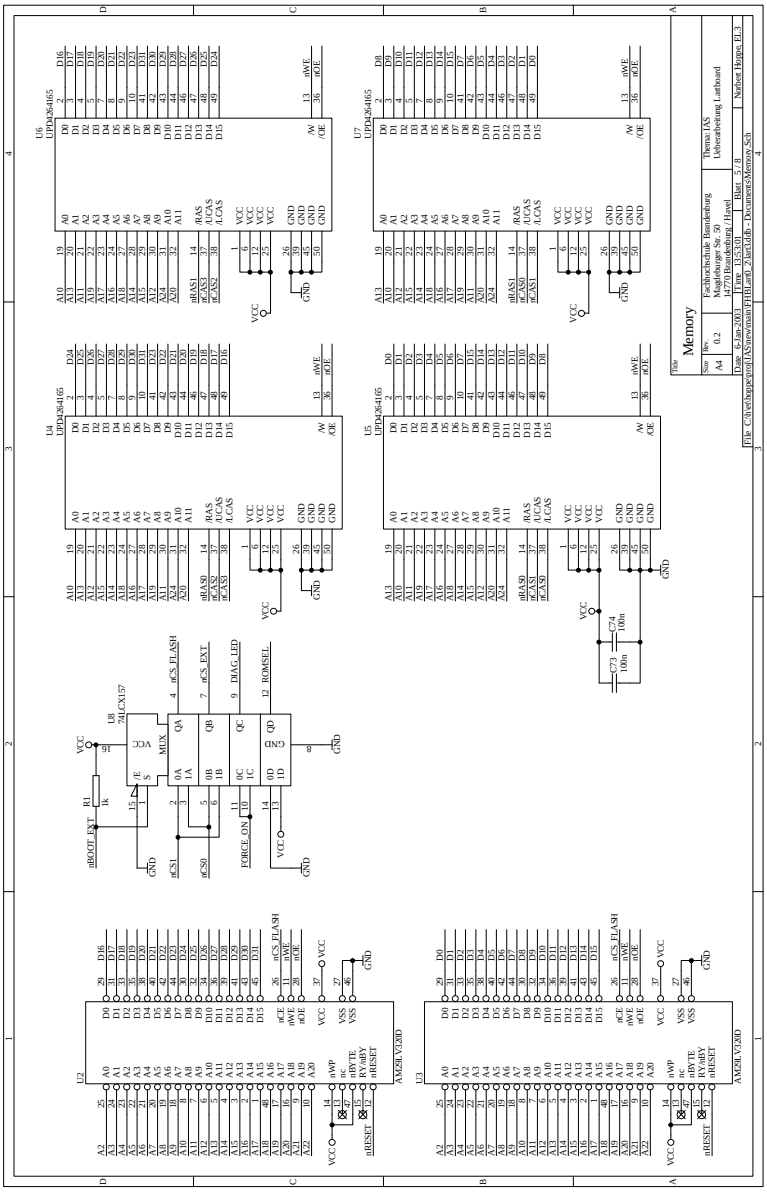


Abbildung 5.6: Schaltplan des CPU-Moduls 6/9

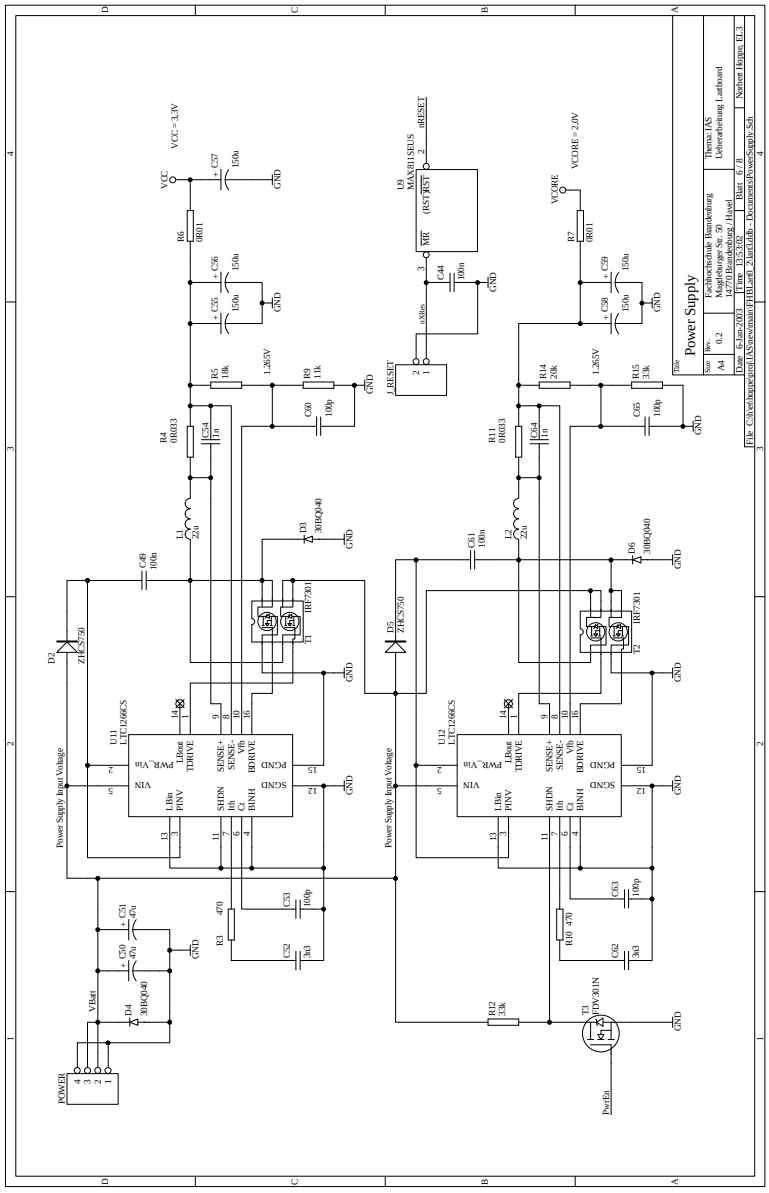


Abbildung 5.7: Schaltplan des CPU-Moduls 7/9

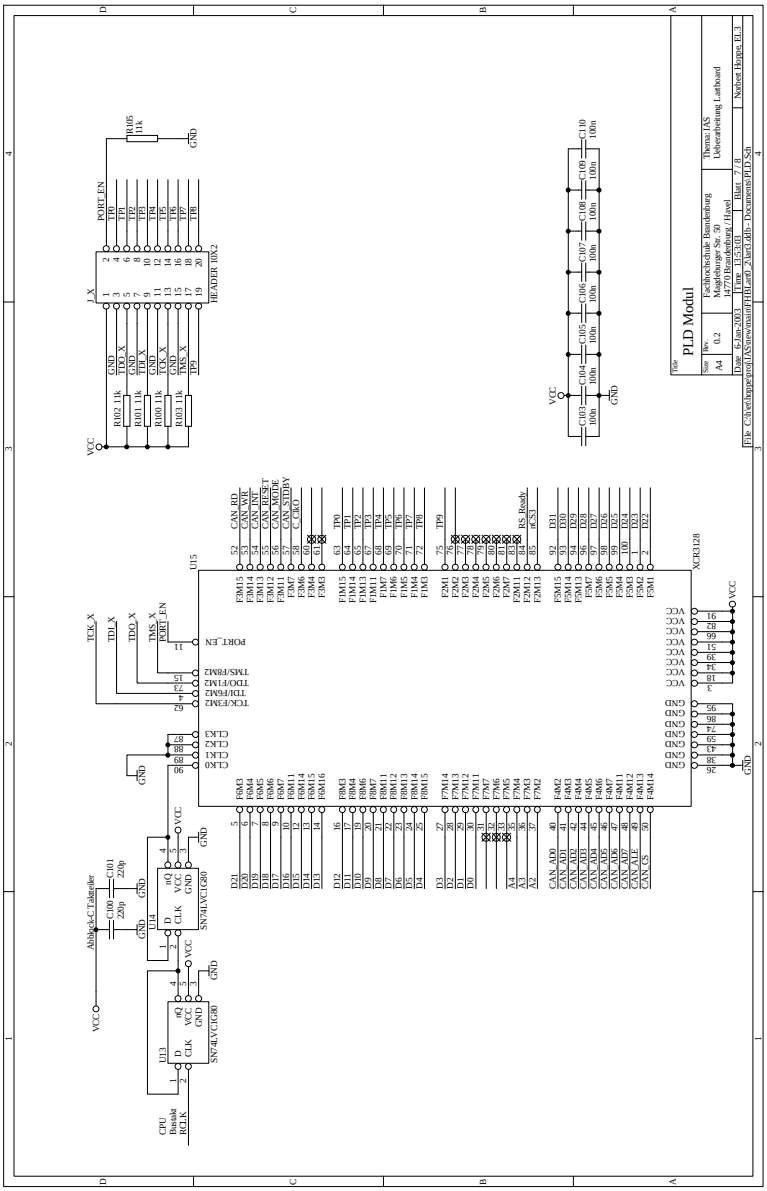


Abbildung 5.8: Schaltplan des CPU-Moduls 8/9

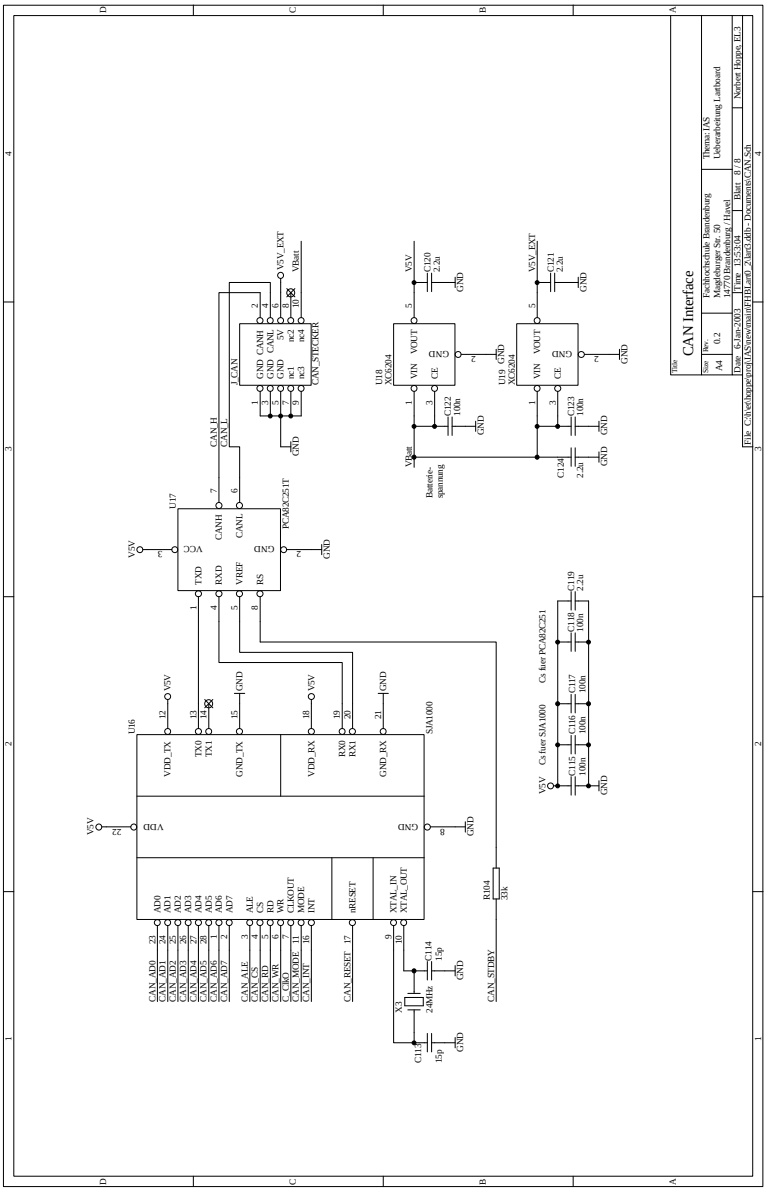


Abbildung 5.9: Schaltplan des CPU-Moduls 9/9

5.3 Schaltplan VIO-Board

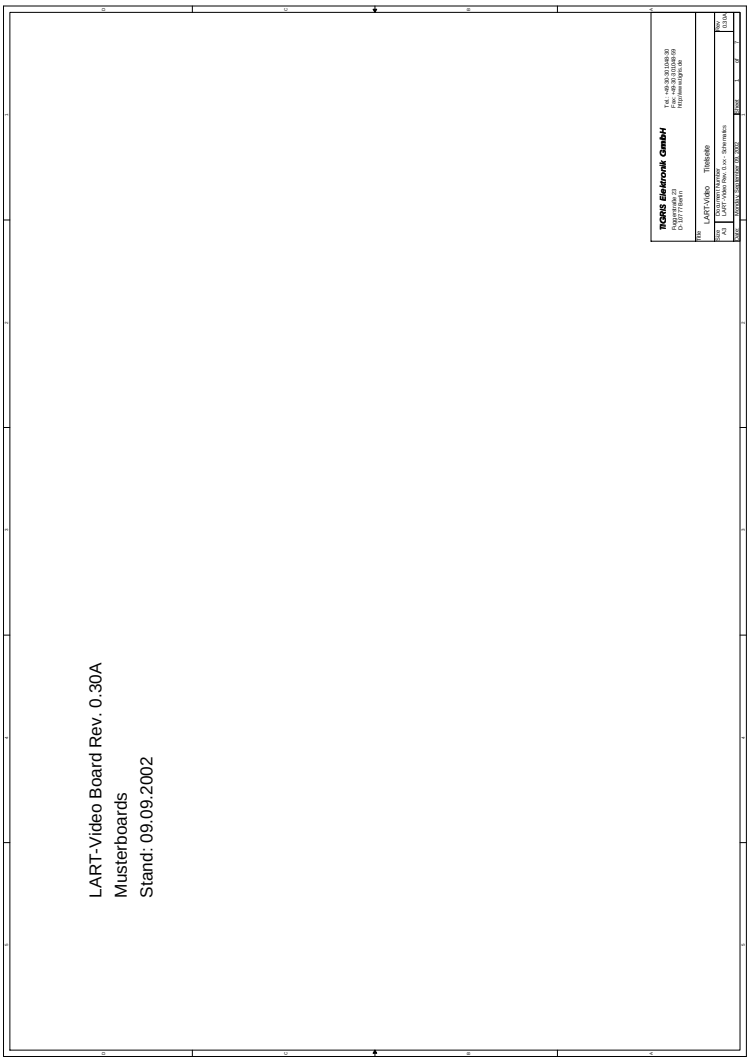


Abbildung 5.10: Schaltplan des VIO-Boards 1/7

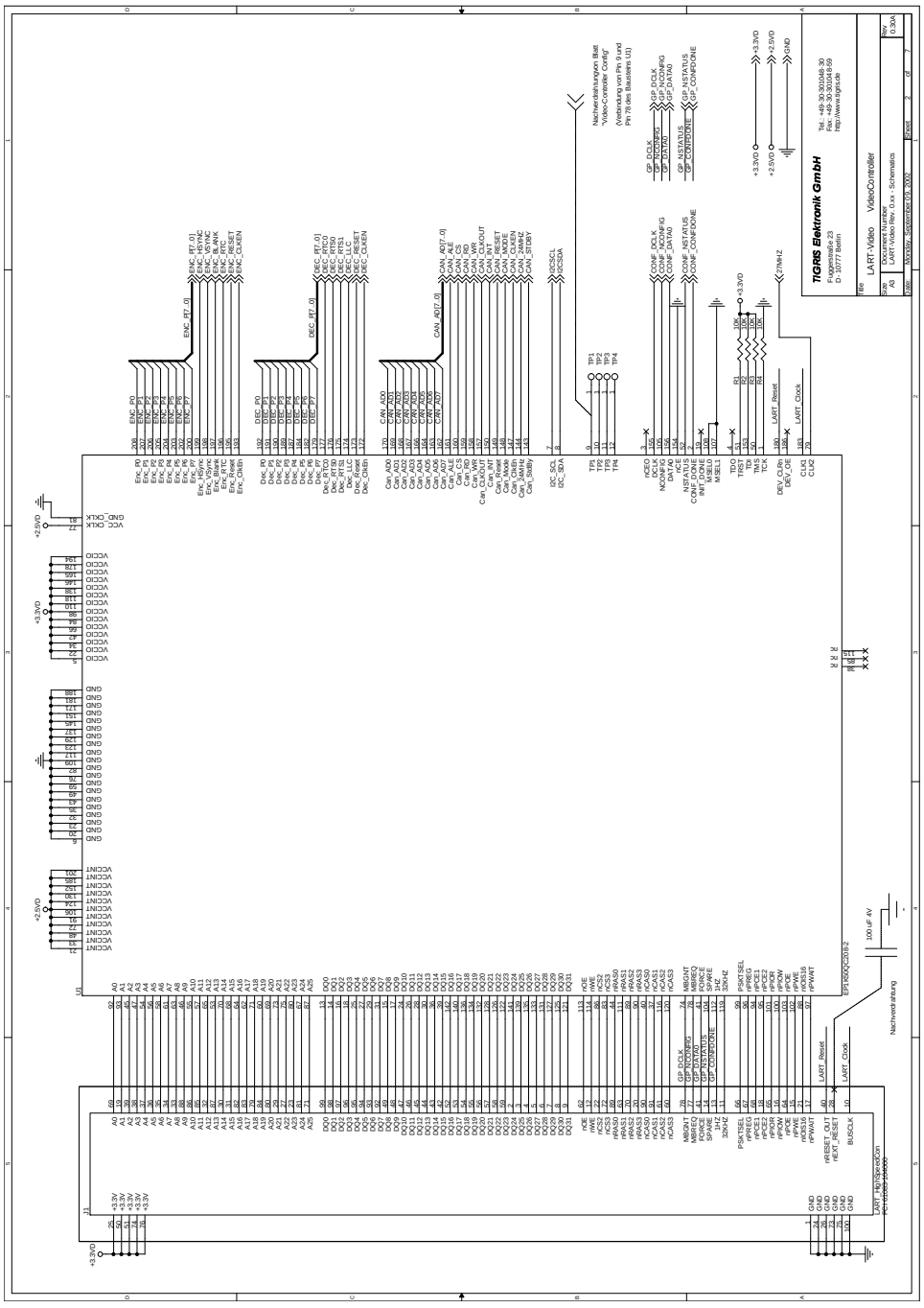


Abbildung 5.11: Schaltplan des VIO-Boards 2/7

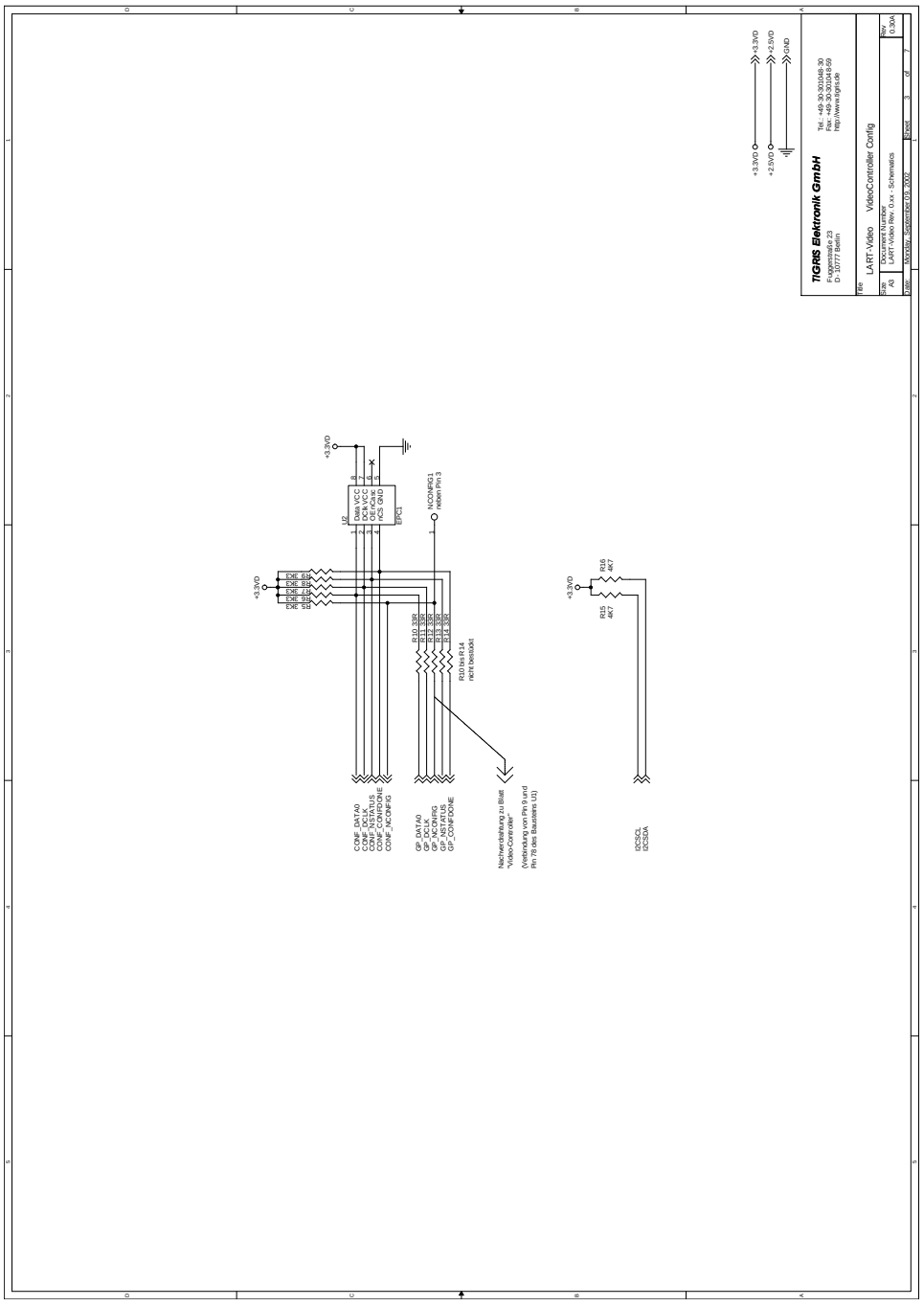


Abbildung 5.12: Schaltplan des VIO-Boards 3/7

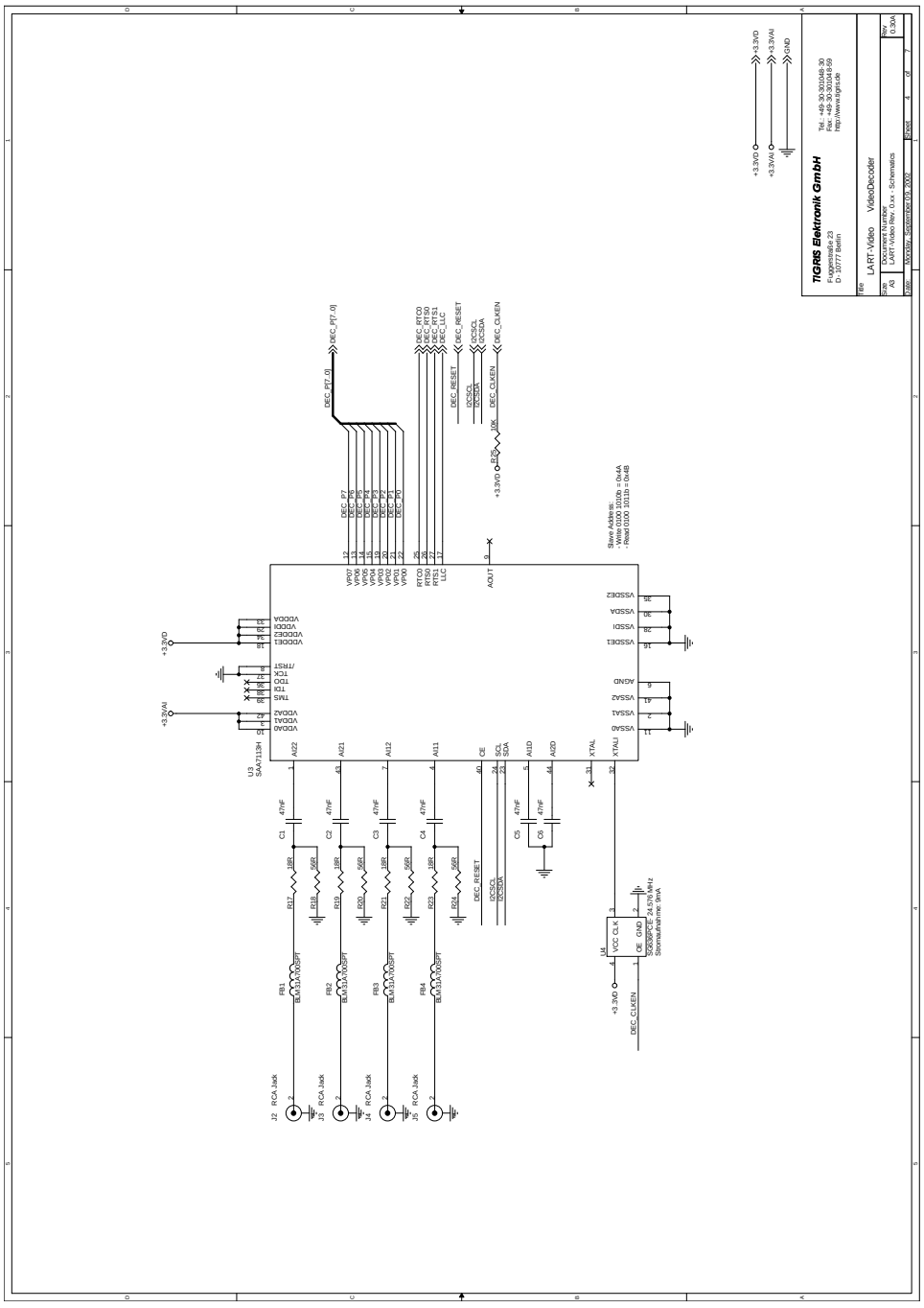


Abbildung 5.13: Schaltplan des VIO-Boards 4/7

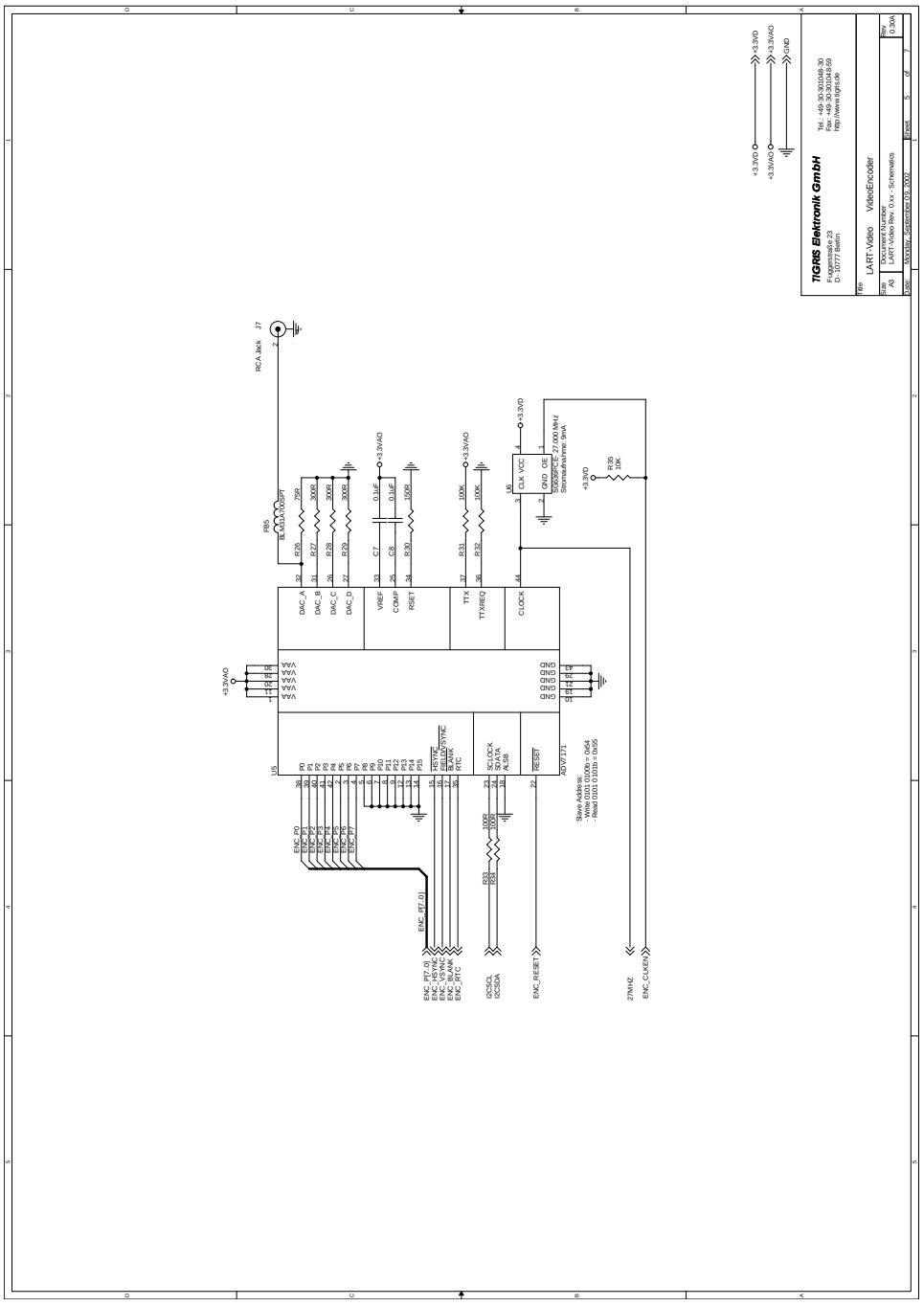


Abbildung 5.14: Schaltplan des VIO-Boards 5/7

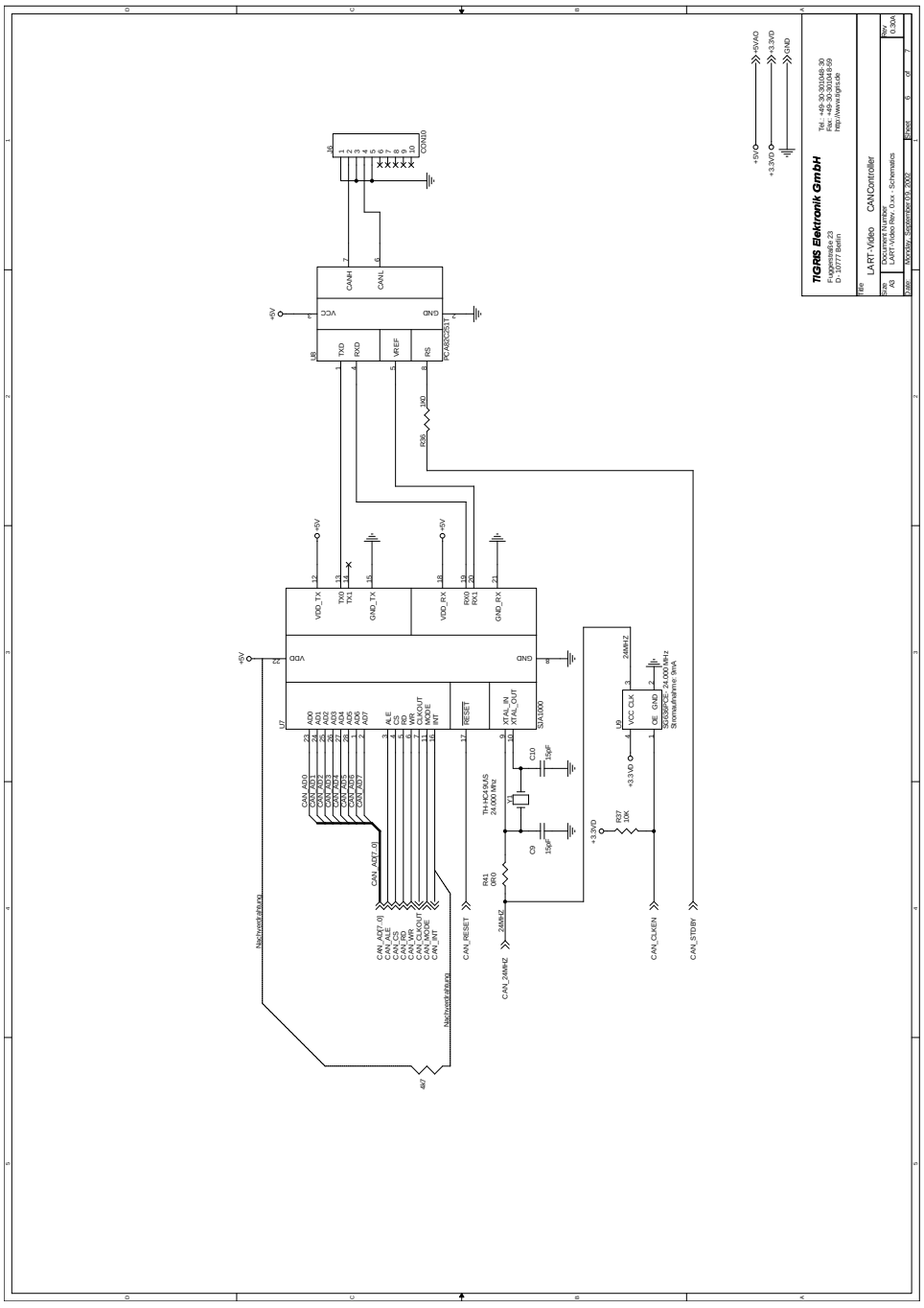


Abbildung 5.15: Schaltplan des VIO-Boards 6/7

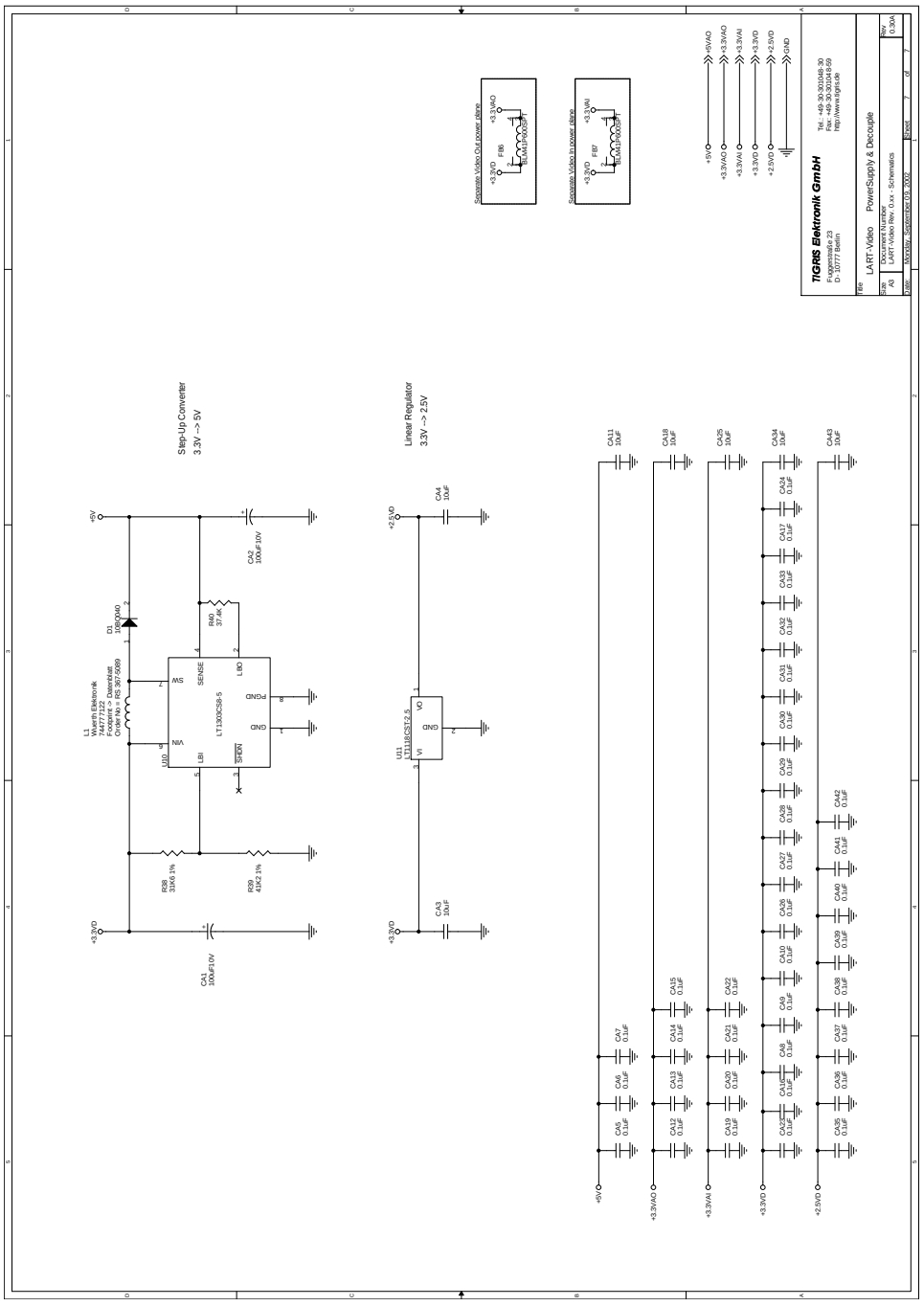


Abbildung 5.16: Schaltplan des VIO-Boards 7/7

Literaturverzeichnis

[Ahl03] Martin Ahlborg. R2D0 (Die Bildverarbeitung). Projektarbeit, Fachhochschule Brandenburg, December 2003. als PDF auf der CD.

[BHLM03] I. Boersch, J. Heinsohn, H. Loose, and K.-U. Mrkor. RCUBE A Platform For Intelligent Autonomous Systems. In IEEE ICIT 2003 Proceedings, International Conference on Industrial Technology, Maribor, Slovenia, December 2003.

[Con04] Conrad Electronics. C-Cam8 Miniature Color Camera, 2004. als PDF auf der CD.

[Esc02] Konrad Eschberger. Controller Area Network. Carl Hauser Verlag, München, 2002. ISBN 3-446-21776-2.

[FJ96] A. M. Flynn and J. L. Jones. Mobile Roboter. Addison-Wesley, 1996.

[Her04] Mathias Herkt. Applikationsrahmen für ein AKSEN-VIO-System, Version 1.0. Projektarbeit, Fachhochschule Brandenburg, January 2004. als PDF auf der CD.

[KI-04] KI-Labor der Fachhochschule Brandenburg, Brandenburg / Havel. Nutzerhandbuch AKSEN-Board, 2004.

[KS02] G. Kuhlmann and F. Schwanz. LART-Video Hardware & Programmable Logic Manual. Tigris Elektronik GmbH, 2002. als PDF auf der CD.

[lar04] The LART Pages, 2004. http://www.lart.tudelft.nl/.

[rcu04] RCUBE Webseite, 2004. http://ots.fh-brandenburg.de/rcube.

[Sch02a] Frank Schwanz. Konzeption und Implementierung einer Videodigitalisierung und Videoausgabe unter Embedded Linux. Diplomarbeit, Fachhochschule Brandenburg, November 2002. als PDF auf der CD.

[Sch02b] Frank Schwanz. LARTVIO Softwareinstallation und Programmierung. Tigris Elektronik GmbH, 2002. als PDF auf der CD.

[sja97] APPLICATION NOTE SJA1000, 1997. als PDF auf der CD.

[sja00] Datenblatt zum SJA1000 Stand-alone CAN Controller, 2000. als PDF auf der CD.

[UNE02] ECONOMIC COMMISSION FOR EUROPE UNECE. World Robotics - Statistics, Market Analysis, Forecasts, Case Studies and Profitability of Robot Investment. Co-authored by the International Federation of Robotics, 2002.

[WW02] Wookey and Paul Webb. Guide to ARMLinux for Developers. Aleph One Limited, 2002. teilweise online unter http://www.aleph1.co.uk/armlinux/thebook.html.