PCI-CAMAC

CC32 CAMAC Crate Controller

with PCI Interface

User Manual

Table of contents:

Legend:

1   PCI-CAMAC SYSTEM: General description

With the help of the PCI-CAMAC system which consists of CAMAC crate controller CC32 and
the PCI card PCIADA, users of the CAMAC bus profit of the technical success, which takes place
in the PC development. For a fast and efficient CAMAC control and data read-out the system
supports 16-bit and 32-bit wide data transfers.
A huge variety of software is available for a PC. In parallel it's performance was constantly
improved. Modern operating systems were developed which turn the PC into a powerful
workstation. On the other hand the number of UNIX workstations which are equipped with PCI
increases strongly. The PCI-CAMAC system can be used in this environment too.
Drivers and programming tools for PCI-CAMAC are provided for different operating systems and
programming languages.
Drivers for Windows-95/98 and Windows-NT offers an easy access to CAMAC data way. For C
and Turbo Pascal users the Pascal and C-libraries provide easy routines to operate CAMAC
modules.
Working with the LINUX driver all advantages of UNIX multi user and multi tasking operating
system can be used for CAMAC operations.
The PCIADA interface card supports 16 and 32 – bit PCI bus slave access. It provides
programmable interrupt generation on PC. PCIADA is also compatible to the PCI-VME system.
To link CAMAC systems equipped with the CC32 into a VME based data acquisition a VME slave
module VC32(VMEADA) is available, which can replace the PCIADA in this case.

1.1   Summary PCIADA / CPCIADA

- PCI-interface based on standard PCI chip **PLX9050** produced by *PLX-Technology*

- support of (8), 16 u. 32 bit PCI-bus slave accesses

- direct data mapping into the target address area with automatic „low-big-endian" conversion.

- use of **32K** memory segment below the 1MB limit to be accessible for MS-DOS applications

- supports interrupt source from CC32 (maskable LAM) and local „timeout" interrupt source

- differential bus driver and receiver for fast and reliable data transfer

1.2   Summary CC32 CAMAC crate controller

- transparent D16 and D24 (D32) CAMAC data way access

- 32K NAF coding / addressing

- 24 bit programmable LAM-mask register

- LAM-interrupt transfer to PCIADA

- FASTCAMAC Level 1

- LED-display for:
  +6V
  CAMAC-access, local CC32-access, INHIBIT and LAM. (stretched to 5 ms)
   CAMAC data way display for: Q, X, C, Z, N1...16, A1…8, F1…16 and Data1…24.

- CAMAC-cycle tuning (Busy to S1) for each station, range 300ns and 200ns, S1/S2=100ns

- Broadcast CAMAC-WRITE and broadcast-mask register

- no interference of PC operation due to CAMAC crate on / off changes (if interrupt disabled)

1.3 Front Panel

CC32

CAMAC-CONTROLLER

Q-Acknowledge — Q
X-Acknowledge — X
Clear-Cycle — C
Initialise-Cycle — Z

+6V — Power
LOC. — Local Acces
CAM. — Camac Acces
INH. — Inhibit
LAM — LAM Flip-Flop

stredched to
5 ms

N16
N8
Station Number — N4
N2
N1

A8
Sub-Addres — A4
A2
A1

WR–F16
F8
Funktion — F4
F2
F1

24
23
22
21
20
19
18
17

Connector
to
PCIADA or VC32(VMEADA)

16
15
14
13
Camac-Data — 12
11
10
9

8
7
6
5
4
3
2
1
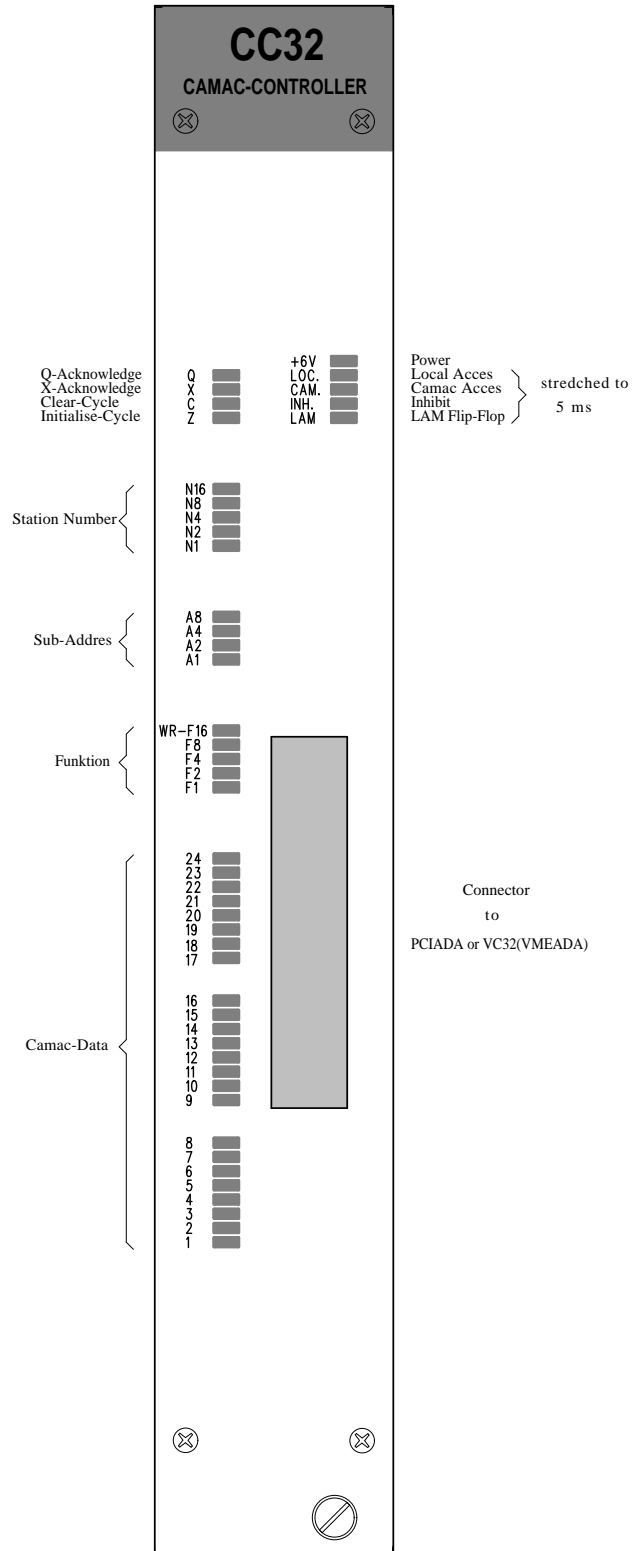
1.4   Installation PCIADA and CC32

1.4.1   ATTENTION

Observe precautions for handling:

- **<u>Electrostatic device!</u>** Handle only at static safe work stations. Do not touch electronic components or wiring.

- The CAMAC crate as well as the used PC have to be on the same electric potential. Different potentials can result in unexpected currents between the CC32 and PCIADA which can destroy the units.

- Do not plug the CC32 into a CAMAC crate under power. **Switch off the CAMAC crate first before inserting or removing any CAMAC module!** For safety reasons the crate should be disconnected from AC mains.

- Do not plug in the PCIADA into a PC under power. First switch off the computer and disconnect from AC mains.

1.4.2   Installation

1. Check the CC32 jumpers and set them according to the required functionality (see **1.5. Module number**). There are no user-definable settings on the PCIADA-card.

2. Turn off the PC and any peripheral equipment. Remove the power cable .

3. Remove the computer outside cover and locate a free PCI-expansion slot.

4. Carefully slide the PCIADA card into the PCI slot. After the card is firmly in secure it's fastening tab to the PC chassis with a screw.

5. Replace the PC cover and power cord.

6. Switch off the CAMAC crate and remove the power cord. Plug in the CC32 on the far right slots (normally slot 24 & 25) and secure it with the front panel screw.

7. Attach one end of the 50-pin cable to the CC32 connector and the other side to the PCIADA card.

8. Switch on the CAMAC crate and the PC. The Plug-and-Play Bios should automatically recognize the card.

9. Install the PCIADA/CC32 driver from the attached CD-ROM as described within the driver manual for your operating system.

10. To work with DOS based programs accessing the PCI-CAMAC system add or modify the EMM386.EXE line in the CONGIG.SYS (as shown on the CD-ROM at \DOS\PASCAL\Readme.txt).

11. Test the installation with the software delivered on CD-ROM.

1.5   Crate number

Using more than one PCIADA cards in one PC to control multiple CAMAC crates requires to differ between the connected CC32 controllers. For this purpose a crate number can be defined by jumper setting on the CC32 board. Please see the location of the jumper array on the component scheme (see 3.17).

Attention: If using multiple CC32 controllers with the same crate number connected to PCIADA cards in one PC it is not possible to differ between the CAMAC controllers and to access them. This can result in unexpected effects.

The actual crate number setting can be determined by software reading the **CC32-Status.**

J304	J303	J302	J301	Function
x	x	x	x	Crate number for multiple PCI-CC32 installations
I	I	I	:	(factory prepared setting, crate number = 1)

   I   =   jumper is installed
   :   =   jumper is not installed

1.6   Access times and performance

Typical access times based on a Pentium 133MHz (including reading of program-code):

Access type	WR-time / us	RD-time / us
PCI to CC32 internal D16	0,6	1,0
PCI to CC32 internal D32	0,65	1,5
PCI to CC32 Dataway D16	1,4 (1,2)	1,8 (1,6)
PCI to CC32 Dataway D32 (R/W24)	1,4 (1,2)	2,0 (1,8)
PCI to CC32 Dataway FAST LEVEL1 D32 (D24)	-	1,5

Values in brackets (..) are obtained with Cycle-Tune bits = 11

2   PCIADA INTERFACE CARD

2.1   32 Kbytes Address range

Attention: The CAMAC version of PCIADA to be used with the CAMAC controller CC32 is configured to use a 32 Kbytes address area instead of the original 8K for use with VMEMM. Thus instead of address bits A0 and A1 the higher address bits A13 u. A14 are transmitted. The CAMAC version of PCIADA is labeled with a "**C"** on the EEPROM.

If necessary it is possible to re-program the EEPROM. This can be done by the user via the computer where the PCIADA is plugged in.

All instructions are given on the enclosed CD-ROM at WORKAROUND.TXT

2.2   General description and function

Attention: It is assumed that the PC is equipped with a self-configuring PCI-BIOS ("Plug-and-Play Bios ") which automatically recognizes the card.

To guarantee a maximum PCI compatibility and performance the PCIADA is based on the PCI-bus target- interface-Chip **PLX9050** produced by *PLX-Technology*, which manages the PCI communication.

At booting time necessary settings are configured by PCI auto setup. No jumpers have to be set on the interface card. All PCIADA information such as memory and IRQ requirements are stored in an EEPROM. Some IDs can be read out from PCR:

During PCI setup PCIADA calls for three different memory areas:

1. 54 byte Local Configuration Register (LCR) of the I/O area. Different control status register and base addresses are available in the LCR. *Note*: Some computer architectures do not provide register mappings in the I/O area. Please refer to LCR in memory area.
2. 32 Kbytes of the memory area for direct access to local CC32 address space and transparent CAMAC data way access

2.3   Major PC-Register (PCR)

The following ID's are used to identify the connected hardware. These values are determined by the PCI-Bios and software drivers:

ID	**PCIADA Defined value for CC32**	PCIADA Defined value for VMEMM (ref. only)
Vendor ID	0x10B5	0x10B5
Device ID	0x2258	0x9050
Subsystem Vendor ID	0x9050	0x9050
Subsystem Device ID	0x2258	0x1167

2.4 Major LC-Registers (LCR)

According to the PCI specification the Base addresses for the LC-Register and CC32 memory areas are automatically allocated during booting of the computer. They can be obtained via BIOS from the PLX chip. Please refer to the software section and the PCI9050 data sheet on the CD-ROM. (see CD-ROM ..PCIADA\Chipdocu\9050-1ds.pdf)

There are no additional registers to be programmed by the user.

2.4.1 Interrupt Control/Status Register

INTERRUPT-1

Interrupt requests, initiated by CC32 (LAM-FF), will set INTERRUPT-1 status bit to 1. For this purpose the interrupt request has to be enabled.

INTERRUPT-2

The INTERRUPT-2 status bit will be activated by local PCIADA events. The following sources can cause **INTERRUPT-2:**

1. Access without connection between CC32 and PCIADA (no cable connection)
2. Access to CAMAC crate which is not switched on (CC32 not working)
3. CC32 access has not yet been enabled. The user bit I/O 2 has to be set to enable the access to the CAMAC crate controller.

INTCSR; LCR-Baseadr + 0x4c (by word-access)

byte		RD	WR	after Init
0	Local Interrupt 1 enable: 1 = enable / 0 = disable (source = CC32)	yes	yes	1
1	Local Interrupt 1 polarity : 1 = active high / 0 = active low	yes	yes	0
2	Local Interrupt 1 status: 1 = active. 0 = not active	yes	no	0
3	Local Interrupt 2 enable: 1 = enable / 0 = disable (source = PCIADA)	yes	yes	1
4	Local Interrupt 2 polarity: 1 = active high / 0 = active low	yes	yes	0
5	Local Interrupt 2 status: 1 = active. 0 = not active	yes	no	0
6	PCI Interrupt enable: 1 = enable / 0 = disable (source = global)	yes	yes	0
7	Software Interrupt: 1 = generate Interrupt	yes	yes	0
15..8	not used	yes	no	00000000 b

2.4.2 User I/O Register / Reset Interrupt

USERI/O2:

The **User I/O Register** (**CNTRL**) is divided into two parts which are summarized in the following. The register USER I/O2:
1. locks CC32 during boot time to prevent any access of the Operating System.
2. resets interrupt 2 if it is deleted.

USER I/O3:

Register USER I/O3 monitors the status of the CAMAC crate and the cable connection, i.e. it shows the switched on CAMAC crate and the proper cable connection.

CNTRL; LCR-Baseadr + 0x50 (by word-access)

byte		RD	WR	after Init
5..0	do not modify !	yes	yes	000100 b
6	USER I/O2 Type, must always be 0	yes	yes	0
7	USER I/O2 Direction, must always be 1 = output	yes	yes	1
8	USER I/O2 output, 0 = disable access to CC32, 1 = enable access	yes	yes	0
9	USER I/O3 Type, must always be 0	yes	yes	0
10	USER I/O3 Direction, must always be 0 = input	yes	yes	0
11	USER I/O3 Input, 0 = CC32 failed , 1 = CC32 OK	yes	no	0
15..12	do not modify	yes	yes	0100 b

Write 0x4184 for enable access to CC32
Write 0x4084 for disable access to CC32

2.5 Access-Timeout

An **Access Timeout** is implemented in the PCIADA interface, which is currently set to about 35 µs. Reaching this time-out the current transaction will be terminated and an INTERRUPT-2 initiated. If the CC32 is switched off or disconnected the same INTERRUPT-2 is activated.

2.6 Interrupts

The PCI-CAMAC system can generate interrupts on the PC caused by two different interrupt sources. PCIADA can activate a Timeout Interrupt which should be routed via software to the interrupt vector number 1. All CC32 generated interrupts should be routed via software to the interrupt vector number 2.

Interrupt Source	Vector No.
PCIADA Interrupt (Timeout)	1
CC32 Interrupt (LAM-FF)	2

2.7   Power Consumption

Voltage	Max. current	Power
+5V	ca. 0,5 A	Ca. 2,5W

2.8   PCIADA Layout
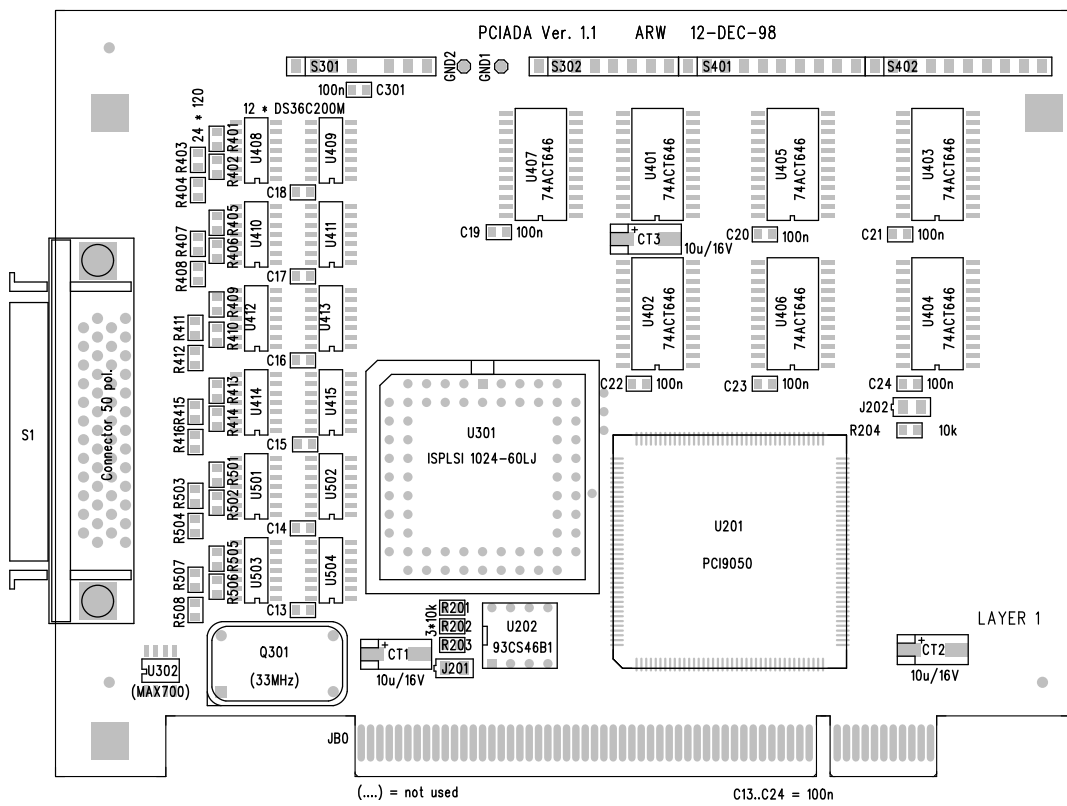
3   CC32 CONTROLLER

3.1   Special Features

3.1.1   FASTCAMAC basic Level 1

The CC32 CAMAC crate controller supports the FAST CAMAC (see DOE/SC-0002 or http://www.yale.edu/fastCAMAC) BASIC level 1 using multiple S1 strobes to increase the data transfer speed (theoretical max. 7.5Mbytes/s). As defined within the FAST CAMAC specification the function code $F = 5$ is used to read data from a module supporting this mode. Getting the first data set in this mode the controller continues automatically to read the following one to have it available without any delay for the next data request from the computer. Thus it is possible to read data with the maximum transfer rate between CC32 and PCIADA which saves about 400ns per read-cycle. This FAST CAMAC level 1 read via $F = 5$ is stopped if the Q-response is missing.

The FAST CAMAC level 1 cycle can be interrupted by another F –command. In this case the new command is executed correctly, however the data which have been already buffered in the CC32 are lost.

3.1.2   CAMAC-Cycle-Tuning

For optimized timing it is possible to adjust the CAMAC cycle time (time between begin of BUSY = active to S1) for each individual CAMAC station C-Station via software. Possible values are 200ns, 300ns and 400ns (default). In addition the width of the S1 and S2 strobe signals can be set optionally to 100ns.

3.1.3   DATAWAY-DISPLAY

To extend the functionality of the CC32 crate controller it is equipped with an integrated CAMAC data way display. This allows to monitor the activity in the CAMAC crate and is a helpful tool to locate faults in the system. The internal data registers (data and control bits) used for this purpose can be accessed also in CC32 controllers without display. The CC32-LED card which is internally plugged onto the CC32 normal station shows the following signals with color LED's:

- Station number N1, N2, N4, N8 and N16
- Sub-address A1, A2, A4 and A8
- Function F1, F2, F4, F8 and F16
- Data 1 – 24 (shared for R1...R24 and W1...W24
- Q, X, (Q and X response)
- C, Z (Clear) and Z (Initialize)
- I (Inhibit)
- Local and CAMAC cycle
- LAM (Look-at-me request from station)
- 6V power line

Please note that the N LED is also responding on local CC32 commands.

3.1.4 CC32 Normal-Station

To allow an easy test of the controller and PCIADA to CC32 connection the following test functions are implemented in the CC32 Normal-Station. The station number **Nn** corresponds to the left one of the two CAMAC slots occupied by the CAMAC controller. As given in 1.4.2 the CC32 has to be plugged in into the most right slots of the CAMAC crate (normally slot 24 and 25).

(**Nn** = CC32 Control-Station – **1**)

Write

Nn * A0 * F16 data $<>$ 5	generate Q and X
Nn * A0 * F16 data = 5	generate Q ,X and LAM (LAM 200ns active)
Nn * A1 * F16 data = 0..15	load test counter, generate Q and X

Read

Nn * A0 * F0 data = 0	generate Q and X

Read in Fast CAMAC Level 1 mode

Nn * A1 * F5 data = 0	decrement test counter, generate X and Q only if test counter content > 0

3.2 NAF Commands and Addressing

A 32 Kbytes memory window is used to access the CC32 and to perform CAMAC operations. This 32 Kbytes area is mapped into the PCI address space. The position (e.g. Basic address) within the PCI address space is dynamically allocated.

For CAMAC commands the **N, A** and **F** numbers are coded into the Address bits *A14 ... A2*. Thus these bits have to be understood as NAF bits. Also local calls are performed as NAF commands. Only word and long word accesses are possible to the CC32 (see 3.3.)

3.2.1 3.2.1 NAF bit coding

32K address CC32	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
CAMAC-Function-bit	N16	N8	N4	N2	N1	A4	A3	A2	A0	F8	F4	F2	F1	-	-

3.2.2 NAF calculation

The F16 bit is automatically defined by the kind of operation, i.e. it is not considered in the NAF code calculation!

WRITE to CC32 defines automatically **F16 = 1**.
READ from CC32 defines automatically **F16 = 0**.

The different address offsets for N, A and F into the 32 Kbytes memory window can be calculated as shown below:

Pascal: **NAF := N shl 10 + A shl 6 + (F AND \$f) shl 2;**
C / C++: **#define MAKE_CC32_OFFSET(N,A,F) ((N<<10) + (A<<6) + ((F & 0xf)<<2))**

To reduce the time required by the software for coding the NAF address it can be helpful to define constants for these values in the user program. This can increase the data rates. Further it is recommended to set the F16-bit to 0 before calculating the address as shown above to avoid an overlap with the A1 bit.

3.3   CC32 Address map

The 32 Kbytes memory window can be accessed only by word and long word calls. byte calls are not processed and answered by the CC32. They are terminated by a TIME-OUT signal from the PCIADA card. All word and long word calls are accepted by the CC32. Please note in case of a long word (Lword) access to an address specified for word only :

Long word-Read from CC32 word address: D31..D16 equals to D15..D00.

Long word-Write to CC32 word address: only D15..D00 will be transferred. D31..D16 are ignored.

All CC32-commands described following are given within the CAMAC NAF-notation. This includes internal CC32 commands. The gray marked cells in the next table indicate operations to the CAMAC stations via the CAMAC data way. All the other described commands are special functions of the CC32 controller.

NAF	Access	WR-Function / F16-bit=1	RD-Function / F16-bit=0
N31*A0*Fx	Word	**CC32 RESET**	**-**
N30*A2*Fx	Word	**CYCLE-TUNE-HIGH** D15..D00 >> N24..N17	**CYCLE-TUNE-HIGH** D15..D00 << N24..N17
N30*A1*Fx	Word	**CYCLE-TUNE-MID** D15..D00 >> N16-N9	**CYCLE-TUNE-MID** D15..D00 << N16-N9
N30*A0*Fx	Word	**CYCLE-TUNE-LOW** D15..D00 >> N8..N1	**CYCLE-TUNE-LOW** D15..D00 << N8..N1
N29*A0*Fx	Lword	**-**	**LED-Status** D23..D00 << LED24..LED1 D27..D24 << C,Z,CT1,CT0 D31..D28 << Q,X,INH,LAM-FF
N28*A4*Fx	Lword	-	**LAM-BUS** D23..D00 << LAM24..LAM1 D31..D24 is equal LAM-MASK
N28*A3*Fx	Lword	-	**LAM-NOT = LAMn & !LMASKn** D23..D00 << NOT24..NOT1 D31..D24 is equal LAM-MASK
N28*A2*Fx	Lword	-	**LAM-AND = LAMn & LMASKn** D23..D00 << AND24..AND1 D31..D24 is equal LAM-MASK
N28*A1*Fx	Lword	**LAM-MASK** D23..D00 >> LMASK23-LMASK0 D31..D24 = xx	**LAM-MASK** D23..D00 << LMASK24..LMASK1 D27..D24 = 0 D28 = LAM-BUS-OR D29 = LAM-NOT-OR D30 = LAM-AND-OR D31 = LAM-FF
N28*A0*Fx	Word	**LAM_FF reset** D15..D00 = xx	**LAM-FF Status** D00 = 1 LAM-FF set D00 = 0 LAM-FF not set D15..D01 = 0

N27*A0*Fx N27*A1*Fx	Word	**INHIBIT on** **INHIBIT off** D31..D00 = xx	**INHIBIT Status** D00 = INHIBIT on = 1 D01 = INHIBIT Dataway on = 0 D15..D01 = 0
N26*A0*Fx	Lword	**Broadcast-MASK** D23..D00 >> BMASK24..BMASK1	**Broadcast-MASK** D23..D00 << BMASK24..BMASK1 D24..D31 << 0
N25*Ax*Fx	Lword	**Broadcast-WR=allN & BMASKn** D23..D00 >>W1..W24	-
N1-24*Ax*Fx	Lword	**CAMAC-DATAWAY WRITE *1** D23..D00 >> W24..W1 *3	**CAMAC-DATAWAY READ *1** D23..D00 << R23..R00 D29..D24 = 0 D31,D30 Q,X
N1-24*Ax*Fx	Word	**CAMAC-DATAWAY WRITE *1** D00..D15 > W1..W16 *3	**CAMAC-DATAWAY READ *1** D00..D15 < R1..R16 *4
N0*A0*Fx N0*A1*Fx N0*A2*Fx N0*A3*Fx	Word	**CAMAC C *2** **CAMAC Z *2** **CAMAC C + INHIBIT off *2** **CAMAC Z + INHIBIT on *2** D15..D00 = xx	**CC32-STATUS** D03..D00 << Q,X,INH,LAM-FF D07..D04 << Modul-Number D11..D08 << FPGA-Revision D15..D12 << Modul-Type 1000b

*1   Standard CAMAC -Access
*2   Standard CAMAC -Access without S1
*3   no W-Data on CAMAC -Dataway when F8-bit is active
*4   if test Q- or X-Status then use Lword-Access

3.4   CC32-Status

N0*A0*Fx   (Read Word)

This register contains the CC32 configuration and status, including the CAMAC status lines Q, X, I and LAM. The module type identification (bit 12 .. 15) and module number (bit 4 … 7) can be used to identify the CC32.

CC32-Status (word read access only)

Bit		RD	WR	after Init
15..12	Module type identification, 1000b for CC32 (0001b VMEMM)	yes	no	1000b
11..8	FPGA-Revision	yes	no	xxxx
7..4	Module number, Coding of Jumpers J304..J301	yes	no	Jumpers
3	Q – Response	yes	no	x
2	X – Response	yes	no	x
1	State of Inhibit-Flip-Flop	yes	no	0
0	State of LAM-Flip-Flop	yes	no	0

3.5 CC32-C,Z,Inhibit,LAM-FF

N0*A0*Fx	= **C** (CAMAC Clear)	(Write Word)
N0*A1*Fx	= **Z** (CAMAC Initialize)	(Write Word)
N0*A2*Fx	= **C** + **I**nhibit off	(Write Word)
N0*A3*Fx	= **Z** + **I**nhibit on	(Write Word)
N27*A0*Fx	= **I**nhibit on	(Write Word)
N27*A1*Fx	= **I**nhibit off	(Write Word)
N28*A0*Fx	= **LAM**-FF reset	(Write Word)

3.6 Broadcast-Mask-Register

N26*A0*Fx   (Read/Write Lword)

Allows to enable / disable CAMAC stations for broadcast write commands. All stations (N1 … N24) with Broadcast-Mask-bit = 1 are enabled.

Data	D23	D22	D21..D3	D2	D1	D0
Broadcast-Mask for:	N24	N23	N22..N4	N3	N2	N1

3.7 Broadcast CAMAC-Write

N25*Ax*Fx   (CAMAC-Write-Cycle Word or Lword)

Broadcast write command, has to be performed with a correct A(x), F(x) and W-Data CAMAC command. Stations N1..N24 are active for this write operation if the corresponding Broadcast-Mask-bit is 1.

3.8 LAM-Mask-Register

N28*A1*Fx   (Read/Write Lword)

All stations (with station number N) with enabled Broadcast-Mask-bit (=1) can generate a LAM-FF in the CC32 controller which can yield in an interrupt on the PCIADA card.

Data	D23	D22	D21..D3	D2	D1	D0
Enable LAM from Station	N24	N23	N22..N4	N3	N2	N1

The negative edge of the LAM-signal arriving from any station is only transmitted to the LAM-Flip-Flop if the corresponding LAM-Mask-bit is active (=1). The LAM-FF stays on the active level until a reset command N28*A0*F16 occurs.

The following status bits can be used to get more detailed information about the LAM conditions:

D28 = 1 (LAM-BUS-OR) if at least one LAM is pending

D29 = 1 (LAM-NOT-OR) if at least one LAM is pending from disabled stations (LAM-Maskbit =0)

D30 = 1 (LAM-AND-OR) if at least one LAM is pending from enabled stations (LAM-Maskbit =1)

D31 = 1 (LAM-FF) if LAM request (LAM FLIP-FLOP).

3.9 LAM-AND-Status

N28*A2*Fx   (Read Lword)

Dxx = 1 if LAM = active and LAM-Maskbit = 1.

Data	D23	D22	D21..D3	D2	D1	D0
LAM status & LMASK	N24	N23	N22..N4	N3	N2	N1

(bits D28..D31 as described in **3.8)**

3.10 LAM-NOT-Status

N28*A3*Fx   (Read Lword)

Dxx = 1 if LAM = active and LAM-Maskbit = 0.

Data	D23	D22	D21..D3	D2	D1	D0
LAM Status & not LMASK	N24	N23	N22..N4	N3	N2	N1

(bits D28..D31 as described in **3.8)**

3.11 LAM-BUS-Status

N28*A4*Fx   (Read Lword)

Dxx = 1 if LAM = active.

LAM-BUS	D23	D22	D21..D3	D2	D1	D0
LAM Status CAMAC-Bus	N24	N23	N22..N4	N3	N2	N1

(bits D28..D31 as described in **3.8)**

3.12 LED-Status

N29*A0*Fx   (Read Lword)

This function can be used to read back the information of the optional dataway display. If no dataway display is installed the function can be used to determine the CAMAC data and status signals of the last CAMAC dataway operation. The LED24..LED1 bits correspond to the write (W) or read (R) data of the last CAMAC cycle.

Data D24..D29 used for CC32 functional tests.

Data	D23	D22	D21..D3	D2	D1	D0
LED-Status	LED 24	LED 23	LED22..LED3	LED 3	LED 2	LED 1

Data	D31	D30	D29	D28	D27	D26	D25	D24
LED-Status	Q	X	Inhibit	LAM-FF	CT1	CT0	Z	C

3.13 CAMAC Cycle-Tune-Register

N30*A2*Fx High-Register for station N24..N17 (Write/Read Word)

N30*A1*Fx Mid-Register for station N16..N9 (Write/Read Word)

N30*A0*Fx Low-Register for station N8..N1 (Write/Read Word)

For optimized timing it is possible to adjust the CAMAC cycle time (time between begin of BUSY = active to negative edge of S1 strobe signal) for each individual CAMAC station. Possible values are 200ns, 300ns and 400ns (default). In addition the width of the S1 and S2 strobe signals can be set optionally to the shorter value of 100ns.

For each station this is done by defining the 2-bit CT1 and CT0 registers. These registers are in the following named Nx-1 and Nx-0 to consider the station number.

Nx-1,Nx-0 = 00 > 400ns CAMAC-Standard

Nx-1,Nx-0 = 01 > 300ns

Nx-1,Nx-0 = 10 > 200ns

Nx-1,Nx-0 = 11 > 200ns / S1 and S2 = 100ns

Register map:

DATA	D07	D06	D05	D04	D04	D02	D01	D00
Cycle-tune high	N20-1	N20-0	N19-1	N18-0	N18-1	N18-0	N17-1	N16-0
Cycle-tune mid	N12-1	N12-0	N19-1	N11-0	N18-1	N10-0	N9-1	N9-0
Cycle-tune low	N4-1	N4-0	N3-1	N3-0	N2-1	N2-0	N1-1	N1-0

DATA	D15	D14	D13	D12	D11	D10	D09	D08
Cycle-tune high	N24-1	N24-0	N23-1	N23-0	N22-1	N22-0	N21-1	N21-0
Cycle-tune mid	N17-1	N16-0	N^5-1	N15-0	N14-1	N14-0	N13-1	N13-0
Cycle-tune low	N8-1	N8-0	N7-1	N7-0	N6-1	N6-0	N5-1	N5-0

Attention: These options do not confirm to the CAMAC standard. They can be used to improve the data transfer and / or the communication with CAMAC modules. It has to be tested by the user which CAMAC module can be used for different CAMAC cycle timing.

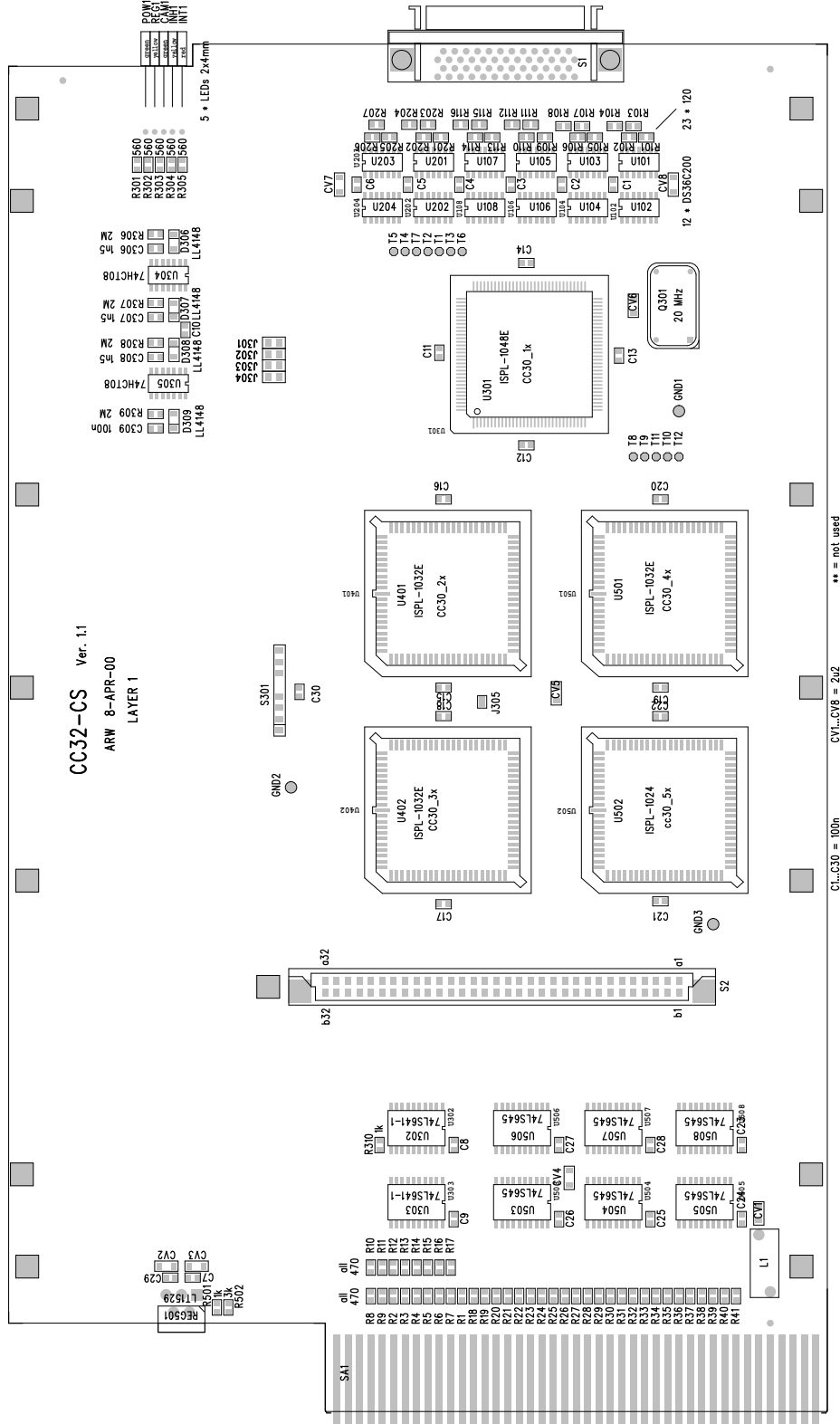## 3.14 CC32-Reset

N31*A0*Fx (Write Word)

Resetting the CC32 initializes the following registers:

- Inhibit-FF
- LAM-FF
- BROADCAST-MASK-, and LAM-MASK-REGISTER
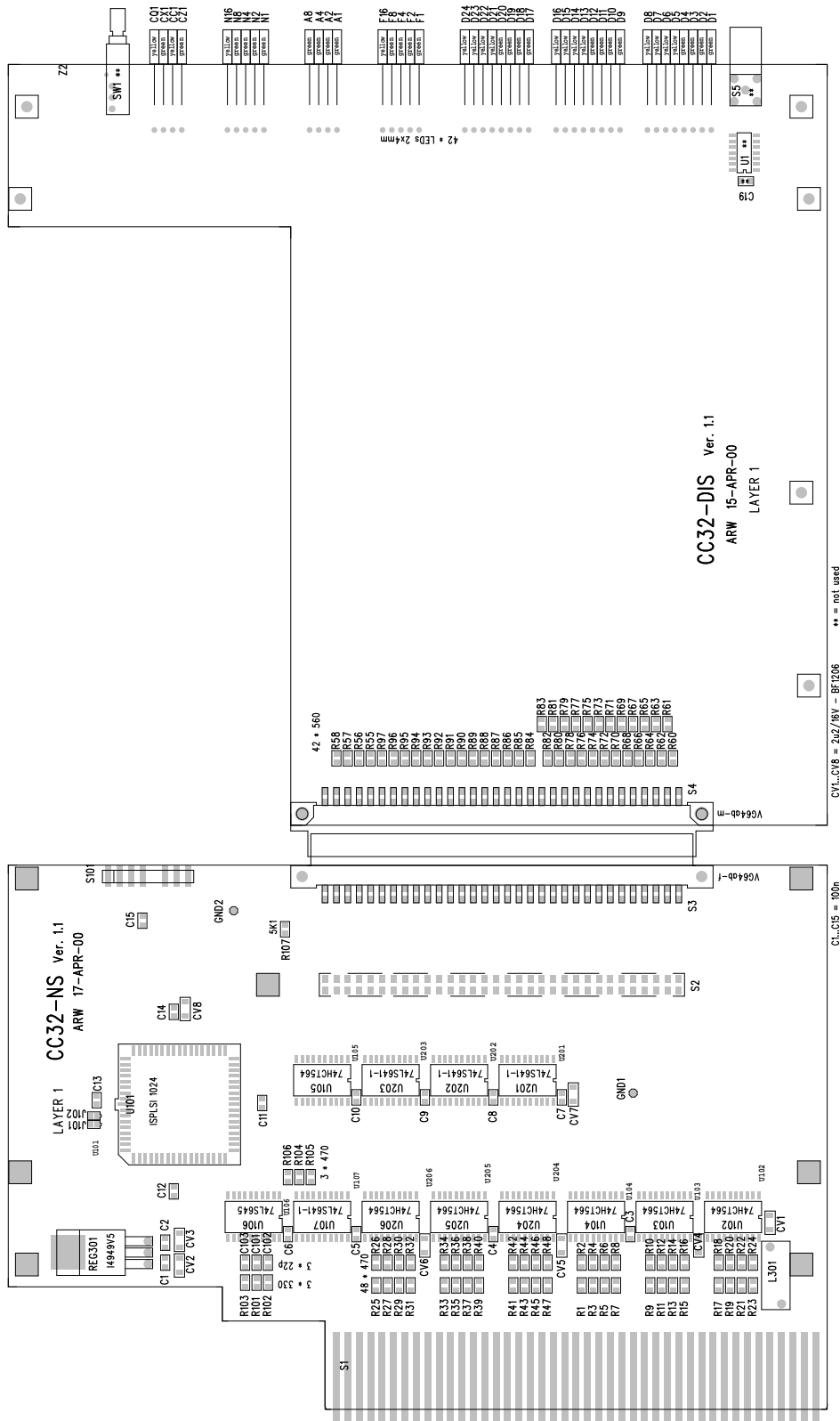- CYCLE-TUNE-REGISTER

3.15 Power consumption

Voltage	Current	Power
+6V	1,7 A	10,2 W

3.16 Component location CC32 Control-Station

3.17   Component location CC32 Normal-Station

3.18 CAMAC data way connector pin assignment

CC32 pin assignment

Normal-Station

Sig.Top	Nr.	Sig.Bott.
	1	B
	2	F16
	3	F8
	4	F4
	5	F2
X	6	F1
I	7	A8
C	8	A4
N	9	A2
L	10	A1
S1	11	Z
S2	12	Q
W24	13	W23
W22	14	W21
W20	15	W19
W18	16	W17
W16	17	W15
W14	18	W13
W12	19	W11
W10	20	W9
W8	21	W7
W6	22	W5
W4	23	W3
W2	24	W1
R24	25	R23
R22	26	R21
R20	27	R19
R18	28	R17
R16	29	R15
R14	30	R13
R12	31	R11
R10	32	R9
R8	33	R7
R6	34	R5
R4	35	R3
R2	36	R1
	37	
	38	
	39	
	40	
	41	
	42	+6
Gnd	43	Gnd

Contol-Station

Sig.Top	Nr.	Sig.Bott.
	1	B
	2	F16
	3	F8
	4	F4
	5	F2
X	6	F1
I	7	A8
C	8	A4
	9	A2
	10	A1
S1	11	Z
S2	12	Q
L24	13	N24
L23	14	N23
L22	15	N22
L21	16	N21
L20	17	N20
L19	18	N19
L18	19	N18
L17	20	N17
L16	21	N16
L15	22	N15
L14	23	N14
L13	24	N13
L12	25	N12
L11	26	N11
L10	27	N10
L9	28	N9
L8	29	N8
L7	30	N7
L6	31	N6
L5	32	N5
L4	33	N4
L3	34	N3
L2	35	N2
L1	36	N1
	37	
	38	
	39	
	40	
	41	
	42	+6
Gnd	43	Gnd