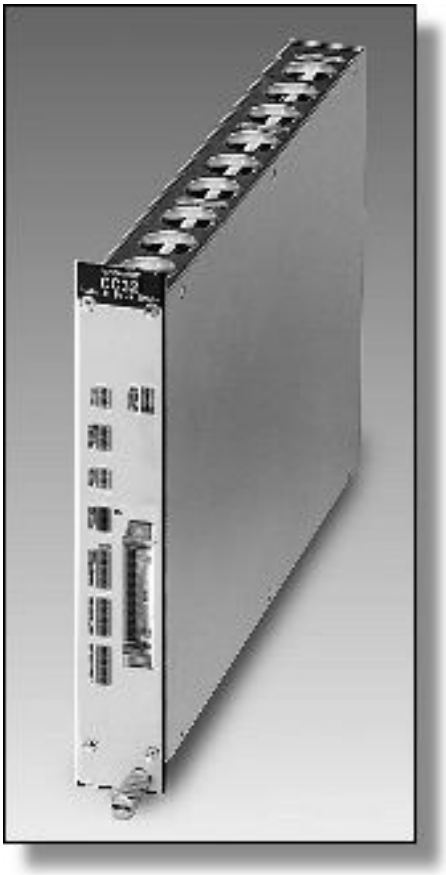


VME – to – CAMAC

CC32 CAMAC Crate Controller

with VC32 VME interface

User Manual

General Remarks

The only purpose of this manual is a description of the product. It must not be interpreted a declaration of conformity for this product including the product and software.

W-Ie-Ne-R revises this product and manual without notice. Differences of the description in manual and product are possible.

W-Ie-Ne-R excludes completely any liability for loss of profits, loss of business, loss of use or data, interrupt of business, or for indirect, special incidental, or consequential damages of any kind, even if **W-Ie-Ne-R** has been advises of the possibility of such damages arising from any defect or error in this manual or product.

Any use of the product which may influence health of human beings requires the express written permission of **W-Ie-Ne-R**.

Products mentioned in this manual are mentioned for identification purposes only. Product names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies.

CC32, VMEMM, PCIADA and VC32 (VMEADA) are designed by ARW Elektronik, Germany.

Table of contents:

Legend:

1 VME-to-CAMAC SYSTEM: General description

With the help of the VME-CAMAC system which consists of CAMAC crate controller CC32 and the VME interface VC32 (VMEADA) CAMAC bus systems can be linked into the VME environment. For a fast and efficient CAMAC control and data read-out the system supports 16-bit and 32-bit wide data transfers.

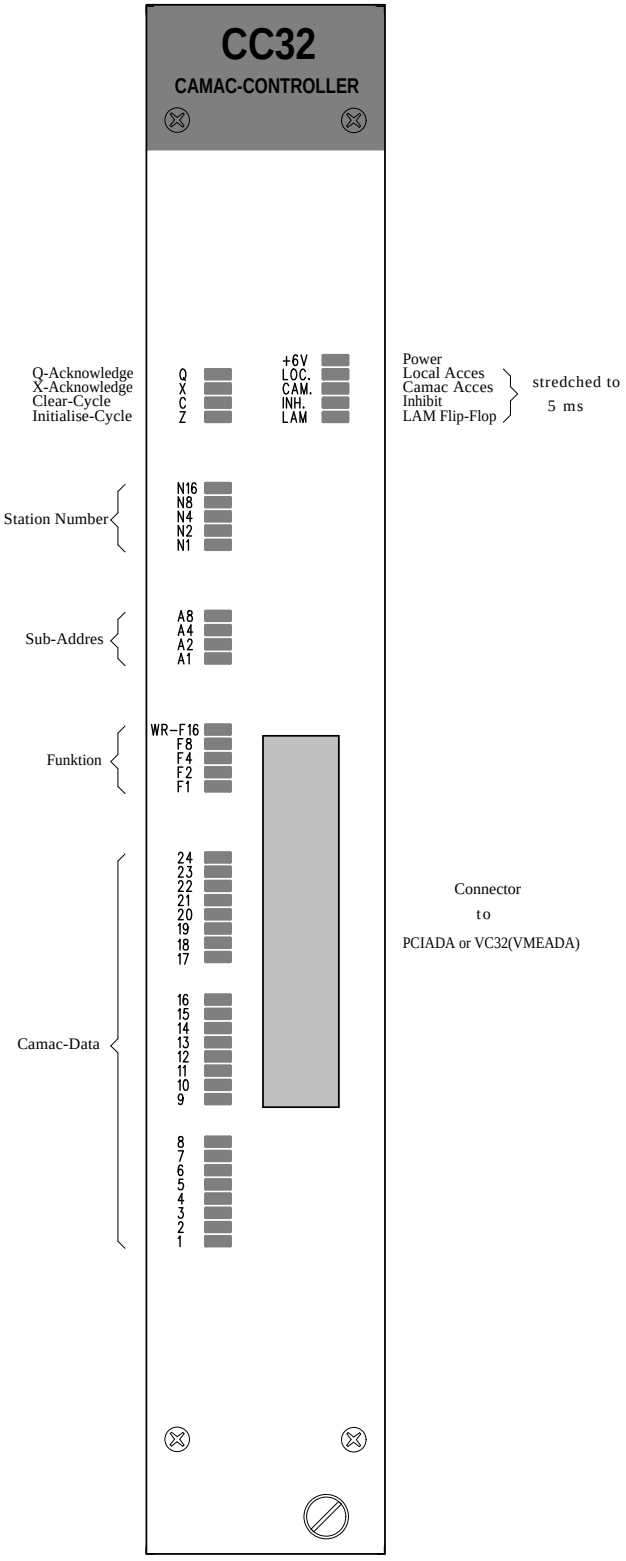
1.1 Summary VC32 (VMEADA)

- 6U x 160mm standard VME card size
- VME-slave module
- Support of A24 / D16 as well as D32 VME-bus transfers
- **32kB** memory area for addressing / direct data mapping to CC32
- supports one maskable interrupt source from CC32
- Interrupt registers for vector, priority and interrupt mode (ROAK or RORA)
- „Auto-read" function
- differential bus driver and receiver for fast and reliable data transfer
- Only 5V DC-power required in VME crate

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 VC32
- FASTCAMAC Level 1
- LED-display for:
  Power +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 VME-bus operation due to CAMAC crate on / off changes (if interrupt disabled), any VME SYSRES activates the CC32 RESET

1.3 Front Panel



1.4 Installation VC32 and CC32

1.4.1 ATTENTION

Observe precautions for handling:

- **Electrostatic device!** Handle only at static safe workstations. Do not touch electronic components or wiring.
- The CAMAC and the VME crate have to be on the same electric potential. Different potentials can result in unexpected currents between the CC32 and VC32, 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 the VC32 into a VME crate under power. **Switch off the VME crate first before inserting or removing any VME module!** For safety reasons the crate should be disconnected from AC mains.

1.4.2 Installation

1. Check the VC32 jumpers and set them according to the required functionality (see **1.5.1 VC32 base-address and Address Modifier**).
2. Check the CC32 jumpers and set them according to the required functionality (see **1.5.2 Module number**).
3. Turn off the VME crate and any peripheral equipment. Remove the power cable.
4. Carefully plug in the VC32 (VMEADA) into a free VME slot. After the card is firmly in, secure its front panel with 2 screws.
5. 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.
6. Attach one end of the 50-pin cable to the CC32 connector and the other side to the VC32 card.
7. Switch on the CAMAC and the VME crate.

1.5 VC32 and CC32 hardware settings

1.5.1 VC32 base address and AM

The VC32 interface can be accessed in the standard A24 SLAVE mode using the 0x39 or 0x3D address modifiers.

For CC32 control and data transfer to/from CAMAC a 32k memory window is used. The base address for this memory segment is defined with jumpers JA13 to JA23.

JA23	JA22	JA21	JA20	JA19	JA18	JA17	JA16	JA15	JA14	JA13	Jumper
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	VME-address bits
I	:	I	:	I	:	I	:	I	X	X	0x550000-0x557fff (default)

I = jumper is installed
: = jumper is not installed

Please check that the jumper J301 is in the „32k" position to define the 32kB window size (8kB window for VME-to-VME systems used). It is possible to check the jumper J301 setting by reading the VC32 status register.

1.5.2 Crate number

Each VC32 interface can control one CAMAC crate equipped with a CC32 controller. Using more than one VC32 cards in one VME crate to control multiple CAMAC crates is possible and requires to differ between the connected CC32 controllers. For this purpose a crate number can be defined by jumper settings on the CC32 board. Please see the location of the jumper array on the component scheme (see 3.17).

Remark: If using multiple CC32 controllers with the same crate number connected to VC32 cards in one VME-bus system it is not possible to differ between the CAMAC controllers and to access them. This is only applicable if your software uses a CAMAC crate search algorithm.

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 VME-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

The following table presents typical access times (DS0 to DTACK /***1**)

Access type	AR-time / ns	RD-time / ns
VME to VMEADA Status D16	200	200
VME to CC32 internal D16	450	500
VME to CC32 internal D32	500	600
VME to CC32 Dataway RW16	450 ***2**	950 *(750)* ***2**
VME to CC32 Dataway RW24 (D32)	500 ***2**	1050 *(850)* ***2**
VME to CC32 Dataway FCL1 R24 (D32)	-	600
VME to CC32 Dataway + AutoRead R24 (D32)	-	125 ***2,3**
VME to CC32 Dataway FCL1 + AutoRead R24 (D32)	-	125 ***4**

All values in brackets are determined with optional CAMAC cycle tuning (Cycle-Tune bits = 11)

***1** The realization of a VME-bus cycle (DSx and Dtack = high) depends on the used VME CPU, a typical value is about 250ns.

***2** Any access to the CC32 requires that a previous dataway-cycle is finished

***3** Minimal repetition time 1,1µs + 125ns

***4** Minimal repetition time about 650ns + 125ns

Using a 200 MHz PowerPC CPU the following results have been obtained:

- Loop-time DS0 with DTACK only approximately 650ns
- CC32 in FASTCAMAC Level 1 mode read cycle approximately 1.15 us
- CC32 in FASTCAMAC Level 1 mode combined with VC32 AUTORAEAD read cycle approximately 800ns.

2 VC32 (VMEADA) INTERFACE CARD

2.1 General description and function

The data transfer to/from CAMAC and for CC32 control is done using a 32kB memory segment on the VMEbus. All CAMAC Read or Write operations are "NAF-coded", i.e. the VC32 translates the address lines A2 … A14 into NAF-bits for the CC32. In case of write operations the data long word (D0 … D32) or word (D0 … D16) are automatically transmitted simultaneously to the CC32. Immediately after the data have been passed to the CC32 the VMEbus cycle is finished by the VC32 with setting DTACK = low.

In case of a read operation the data are transmitted by the CC32 automatically, i.e. after receiving the NAF code requesting the data a data long word (D0 … D32) or word (D0 … D16) is automatically submitted to the VC32 interface. If the read-operation is a CAMAC dataway based one (read-out of a CAMAC module) then the data have to be sent to the CC32 via a CAMAC cycle first. The data are stored in the VC32 memory and immediately passed to the VME-bus. The VME bus cycle is ended by setting DTACK = low.

2.2 Auto-Read Function

To increase the read-out performance the VC32 has a build in Auto-Read function (block mode read) which has to be enabled / disabled in the Status and Control Register **SCR**.

Enabling the Auto-Read mode by setting the **SCR** Auto-Read-bit the first read-operation will be performed as a standard read-operation with NAF-code transmission. After this cycle the VC32 automatically starts to read again data at the same memory address (NAF) from the CC32. Every read operation will cause the CC32 to provide the data already for the next read-call until the Auto-Read cycle is stopped. This mode can reduce the read-out time per cycle up to 500ns.

To stop the Auto-Read-cycle a write-operation has to be performed. It is recommended to write to the status and control register for this purpose. Please note, that the last data submitted by the CC32 CAMAC crate controller will be lost. Further any change of the VME address (corresponding to NAF) as well as switch between word and long word will be neglected until the Auto-Read cycle is terminated.

To change the mode to non-auto-read the Auto-Read bit in the **SCR** has to be set to 0 else any new read-operation will start automatically the next Auto-Read cycle.

2.3 VC32 Status and Control Register SCR

The 16-bit Status and Control Register **SCR** can be used for defining the VC32 configuration / settings. Reading this register returns the actual configuration. A read or write from/to VC32-SCR doesn't need access to CC32.

The SCR address is: **Base address + 0x0c** (NAF-Notation = **Base address + N0*A0*F3)**

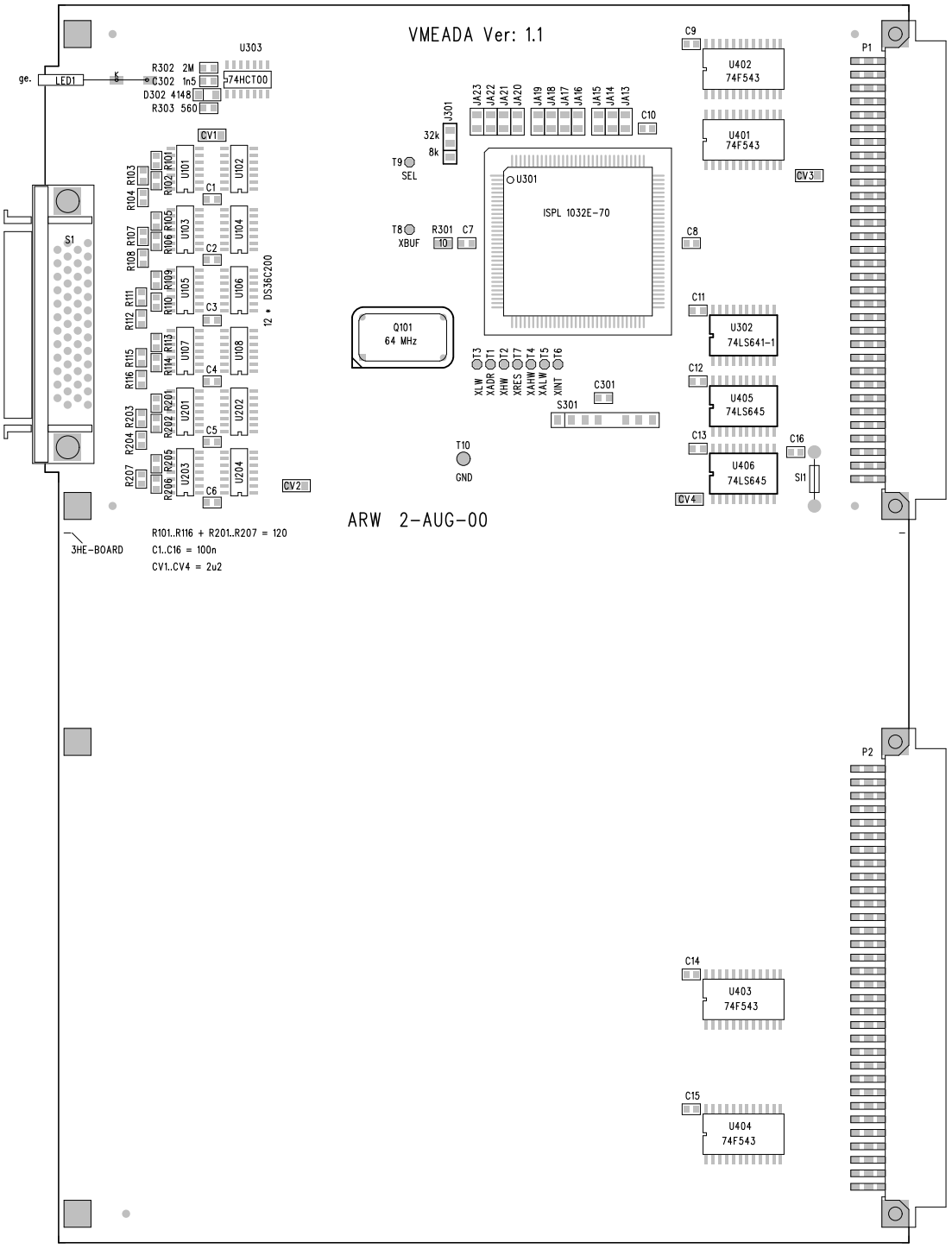
VC32 SCR – bit assignment (word read/write)

Bit		RD	WR	after Init
15	1 = CC32-LAM-FF is set / 0 = CC32-LAM-FF is not set	yes	no	x
14	1 = CC32-OK / 0 = CC32 not connected or Power off	yes	no	x
13	1 = J301>32k-Size for CC32 / 0 = J301>8k-Size for VMEMM	yes	no	x
12	1 = Auto Read on / 0 = Auto Read off	yes	yes	0
11	1 = ROAK-Mode / 0 = RORA-Mode / see VME bus spec.	yes	yes	0
10..8	VME-Interrupt-Priority / 0 = disable Interrupt / 1-7 enable VME-Interrupt 1..7 (e.g. VC32 generates a interrupt 1..7 when LAM-FF is set)	yes	yes	0
7..0	VME-Interrupt-Vector / User defined Vector	yes	yes	0

2.4 Power Consumption

Voltage	Max, current	Power
+5V	ca. 0,7 A	Ca. 3,5W

2.5 Component location VC32



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 VC32 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 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 VC32 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 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 VMEbus standard address space (Address Modifier 0x3D or 0x39).

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 3.4.)

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. Normally they are terminated by a VME-Timeout from VME-Master. 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-RegC** D15..D00 >> N24..N17	**CYCLE-TUNE-RegC** D15..D00 << N24..N17
N30*A1*Fx	Word	**CYCLE-TUNE-RegB** D15..D00 >> N16-N9	**CYCLE-TUNE-RegB** D15..D00 << N16..N9
N30*A0*Fx	Word	**CYCLE-TUNE-RegA** D15..D00 >> N8..N1	**CYCLE-TUNE-RegA** 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..D02 = 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 << Module-Number D11..D08 << FPGA-Revision D15..D12 << Module-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-Flip-Flop) if LAM request

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 are 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 Register-C for station N24..N17 (Write/Read Word)

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

N30*A0*Fx Register-A 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 RegC	N20-1	N20-0	N19-1	N18-0	N18-1	N18-0	N17-1	N16-0
Cycle-tune RegB	N12-1	N12-0	N19-1	N11-0	N18-1	N10-0	N9-1	N9-0
Cycle-tune RegA	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 RegC	N24-1	N24-0	N23-1	N23-0	N22-1	N22-0	N21-1	N21-0
Cycle-tune RegB	N17-1	N16-0	N^5-1	N15-0	N14-1	N14-0	N13-1	N13-0
Cycle-tune RegA	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)

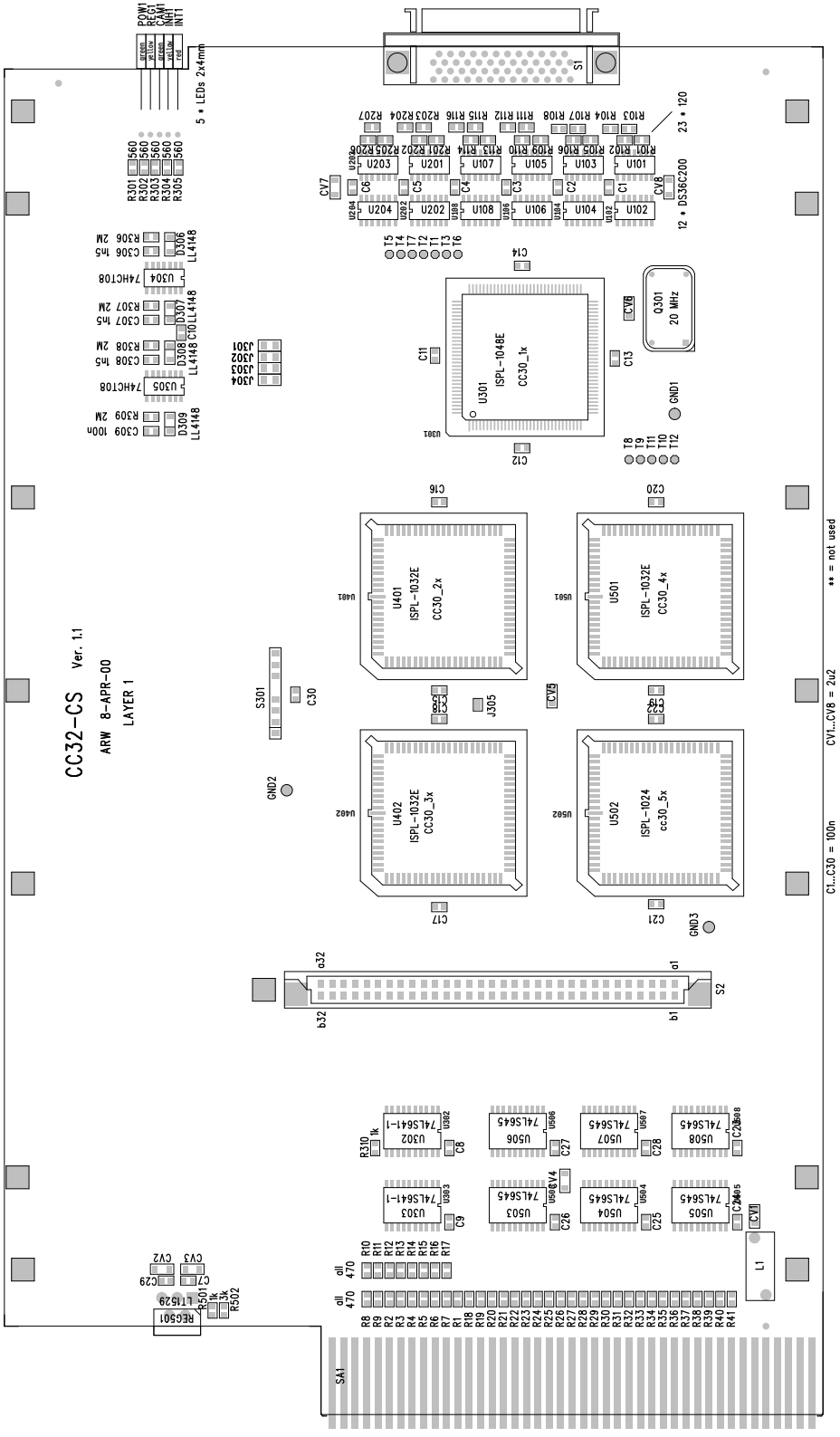
Resetting the CC32 initializes the following registers:

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

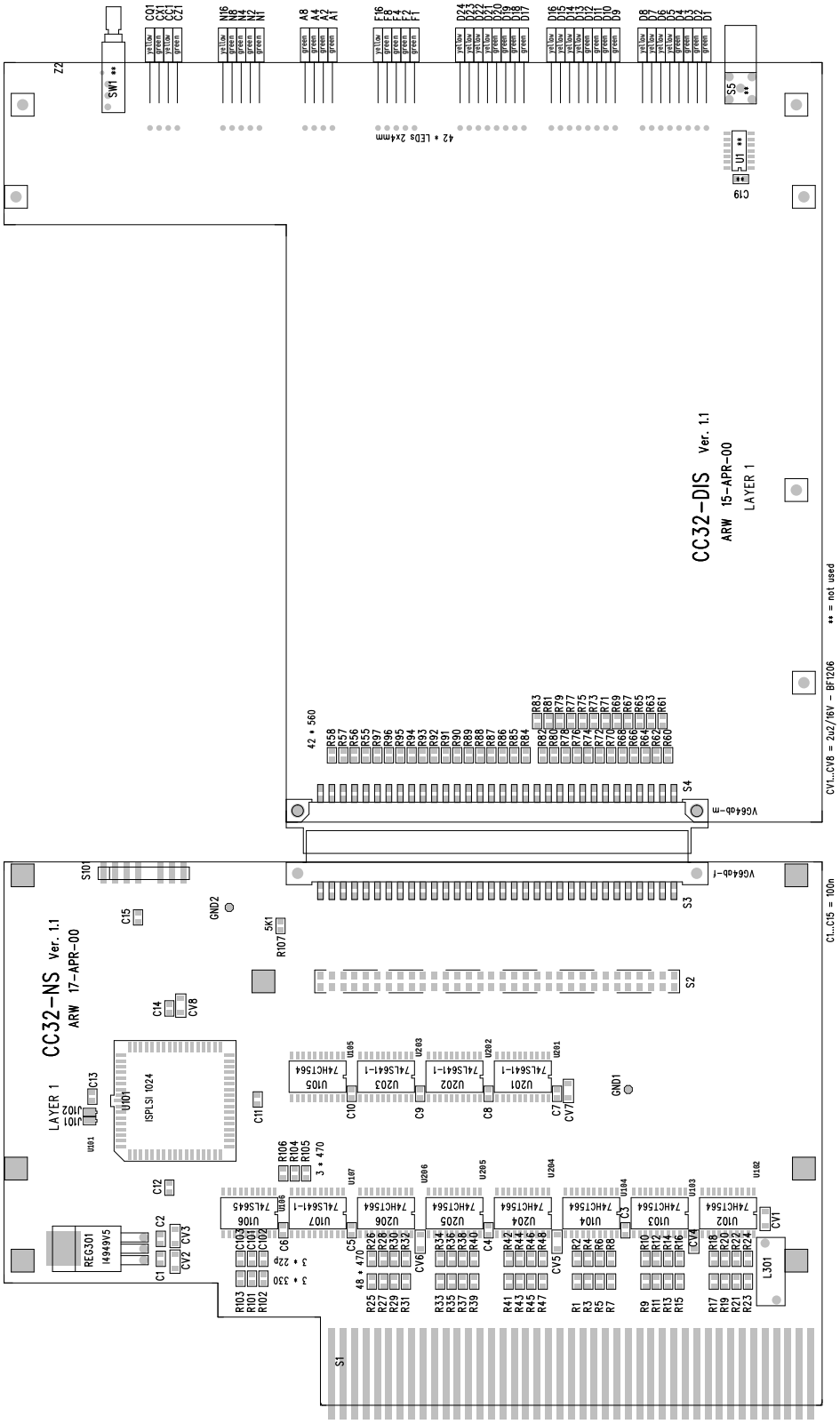
3.15 Power consumption

Voltage	Max, 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 CC32 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