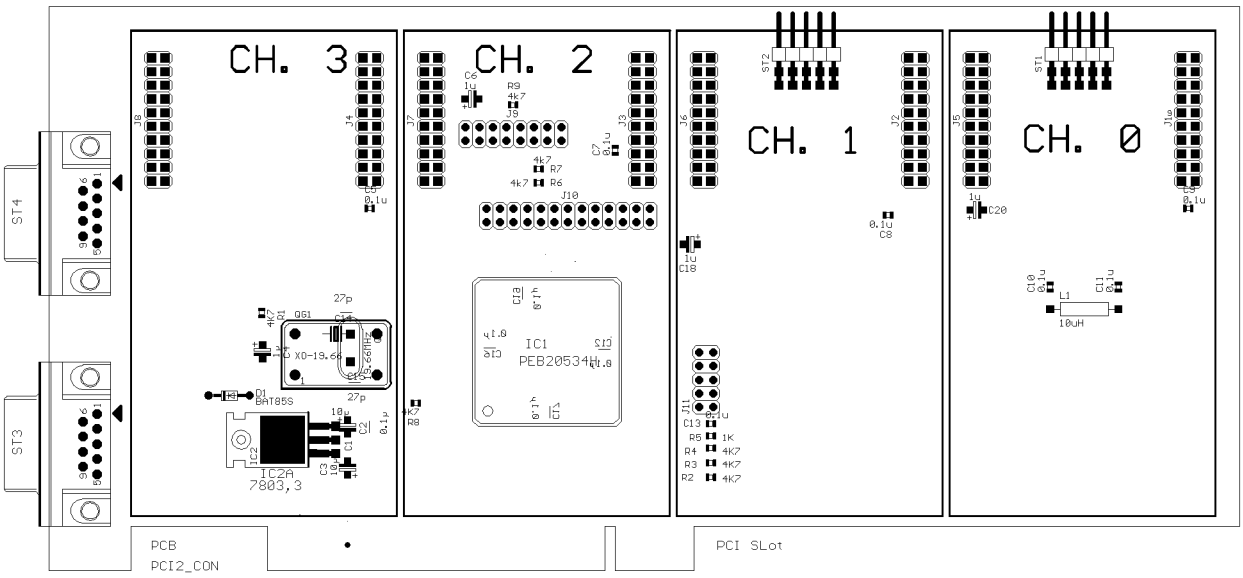
PCISCC

High Speed HDLC slotcard for PCI bus



Hardware Documentation
Stand: 06/01

BayCom

Important note:

Ideas, texts, drawings and schematics in this manual are copyright protected. Distribution and publication and commercial exploitation are prohibited. Misuse will be legally prosectuted.

Copyrights for schematics and board are held by Jens David, DG1KJD and Johannes Kneip, DG3RBU.

Liability disclaimer: The authors cannot be held liable for correctness of the diagrams and descriptions in this manual. When operating this product, all legal regulations need to be observed.

This project is the result of a cooperation between

Jens David, DG1KJD

Johannes Kneip, DG3RBU

Steffen Köhler, DH1DM

Special thanks to

Alex Kurpiers, DL8AAU

Martin Liebeck, DL4ZX for their cooperation in the first field tests.

Introduction

The trend to increasing data rates on amateur radio got a big boost with the allocation of wide band user access frequencies in the 70cm band in Germany. Data rates of several hundred kilobits per second came into reality. This also opens up completely new applications, like digital speech transmission or HTML based content presentation, integrating text and graphics.

The existing ISA based USCC card is employed at several hundred network nodes and user stations. However, due to a lack of buffering in the Z8530 SCC chips, the maximum data rate with this hardware is around 38400 bit/s. Also, many new PCs do not longer feature an ISA slot. This called for the development of new, powerful hardware.

In cooperation with Jens David, DG1KJD (Hardware, Linux drivers) and Steffen Köhler, DH1DM (DOS and Windows drivers) we have developed the PCISCC, a PCI based successor to the USCC card. This card will allow High-Speed packet radio up to the Megabit range.
We have tried to preserve the investments of the radio amateurs into their network infrastructure. Thus, the PCISCC allows the use of the same piggyback modems as on the USCC>4. With the FPGA based piggy back modem by HB9JNX a new high speed modem is available which has been designed especially for the PCISCC.

PCISCC schematics

The schematic diagram on the next page shows the principal design of the PCISCC board. The PCISCC bases on the Infineon DSCC-4 communication controller (PEB 20534H). This chip achieves its high throughput by a number of measures: It features 10 built in DMA controllers and is capable to process a data rate of up to 52 Mbit/s per serial channel. It features four channels, which can be configured independently from each other.
Each channel consists of:

- A clock generation and regeneration block, which provides up to 13 different clock modes and a clock regeneration PLL for data rates up to 2 Mbit/s.
- A Decoder block supporting NRZ, NRZI, FM0, FM1 and Manchester coding.
- A bit processing unit supporting synchronous and asynchronous data transmission (e.g. RS-232 or HDLC), with CRC, Preamble generator and programmable bit stuffing.

The PEB20534 (IC1) includes a full PCI interface, thus only very few peripherals are required. However, the chip operates on 3,3V instead of the usual 5V. This auxiliary voltage is gained by IC2 (7803,3 or LM2937). D1 is a latch-up protection. Oscillator or crystal (both can be used alternatively) are used to generate external clocks. A baudrate crystal with 19.66 Mhz is used, but can be changed to any other convenient frequency. Please note that the division factors of the software drivers

then need to be changed accordingly.

The digital interfaces of the four channels are lead to four modem-disconnect pinheads (J1-J3). They feature the DF9IC standard layout and connect to the piggyback modems.

JP9 and JP10 are connected to a general purpose bus output of the DSCC chips. They can be used to configure modems or perform other switching functions. However, this is not yet supported by the current drivers. JP11 carries the JTAG Test and debug signals used for driver development and debugging.

Component list

IC1	PEB20534	R1	4k7
IC2	7803.3	R2	4k7
		R3	4k7
D1	BAT85S	R4	4k7
		R5	1k
Q1 o. XO1	19.66 MHz Crystal o. Osc.	R6	4k7
		R7	4k7
St1	Pinhead 2*5, 90deg.	R8	4k7
St2	Pinhead 2*5, 90deg.	R9	4k7
St3	DB-9, male		
St4	DB-9, male	C1	not assembled
		C2	10uF Tantalum
J1	Pinhead 2*10 w	C3	10uF Tantalum
J2	Pinhead 2*10 w	C4	1uF Tantalum
J3	Pinhead 2*10 w	C5	100nF
J4	Pinhead 2*10 w	C6	1uF
J5	Pinhead 2*10 w	C7	100nF
J6	Pinhead 2*10 w	C8	100nF
J7	Pinhead 2*10 w	C9	100nF
J8	Pinhead 2*10 w	C10	1uF Tantalum
J9	Pinhead 2*8	C11	1uF Tantalum
J10	Pinhead 2*13	C12	100nF
J11	Pinhead 2*5	C13	100nF
		C14	27pF
L1	Drossel 10uH	C15	27pF
		C16	100nF
		C17	100nF
		C18	1uF Tantalum
		C19	100nF
		C20	1uF Tantalum

C12, C16, C17 and C19 are located on the solder side of the board.

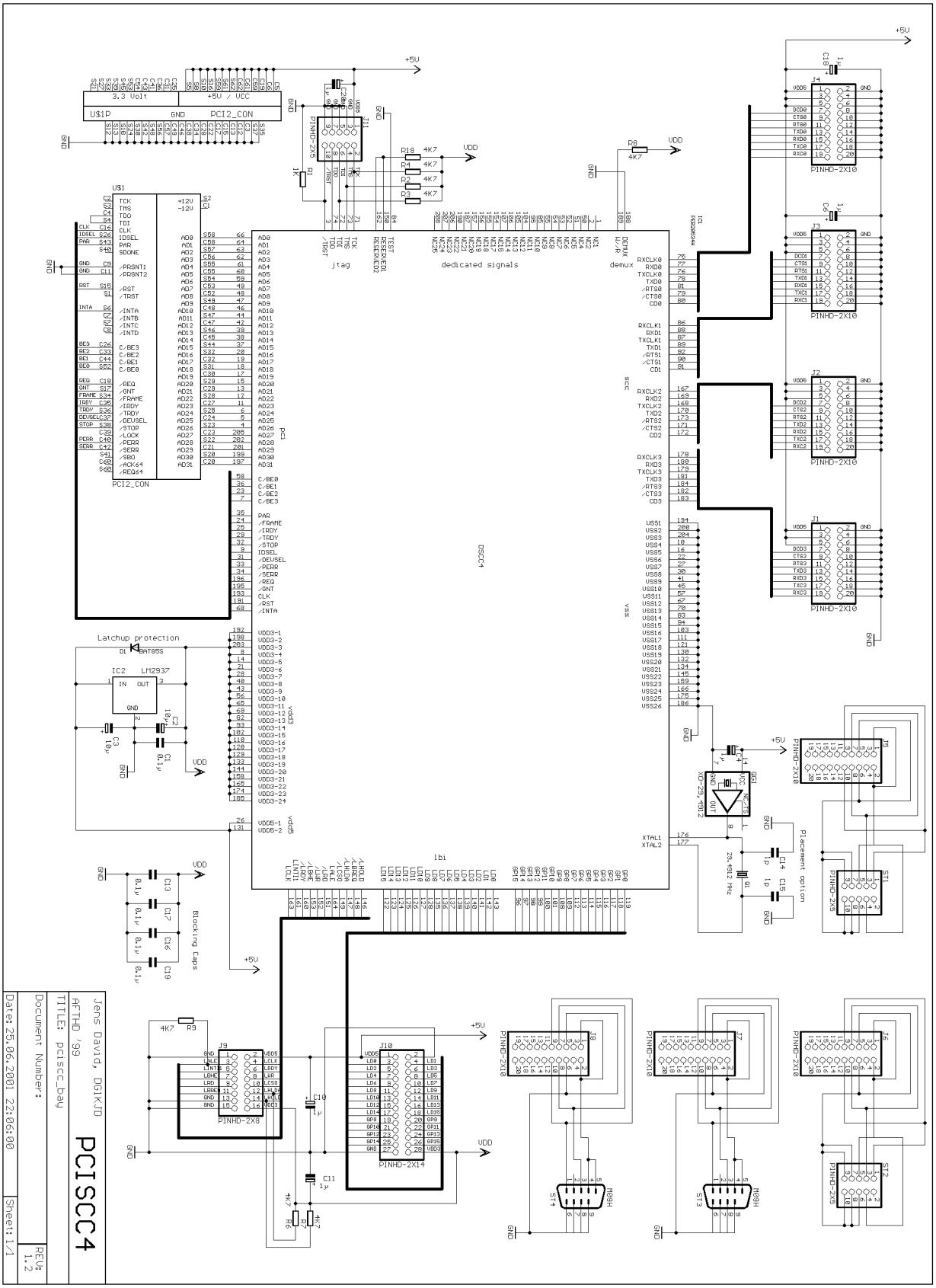


Fig 1: PCISCC-Schematics

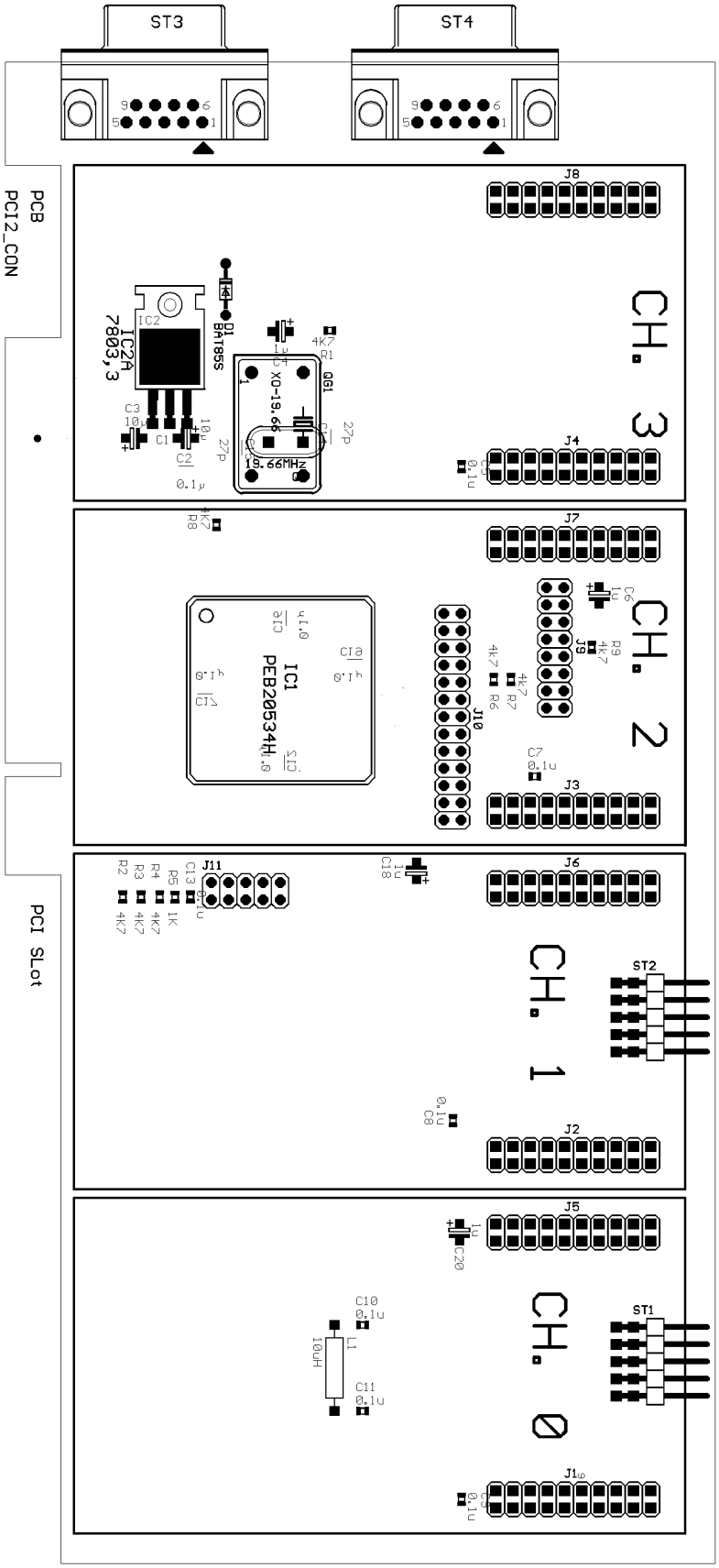


Fig 2. shows the assembly diagram.

Connector layout

Pin 1 is marked on the assembly diagram (Fig. 2).

St 1, St2 (Modem-Af Connectors channels 0 und 1):

1	TX Nf	2	n.c.
3	PTT	4	RX Nf
5	RX Nf	6	GND
7	GND	8	GND
9	GND	10	n.c.

St1 and St2 allow the use of a standard RS-232 extension cable (10p Pinhead to 9p Sub-D male) to achieve the same connector layout as on St3 and St4.

St 3, St4 (Modem-Af connectors channel 2 und 3):

1	TX Nf	2	n.c.
3	PTT	4	RX Nf
5	RX Nf	6	GND
7	GND	8	GND
9	GND		

Modem-Disconnect, JP1-JP4:

1	RXCLOCK	2	GND
3	TXCLOCK	4	GND
5	RXDATA	6	GND
7	TXDATA	8	GND
9	RTS	10	GND
11	CTS	12	GND
13	DCD	14	GND
15	n.c.	16	GND
17	VCC (5V)	18	GND
19	VCC (5V)	20	GND

Modem-Disconnect Af, JP5-JP8:

1	RX Nf	2	RX Nf
3	n.c.	4	n.c.
5	GND	6	GND
7	PTT	8	PTT
9	TX Nf	10	TX Nf
11	n.c.	12	n.c.
13	n.c.	14	n.c.
15	n.c.	16	n.c.
17	n.c.	18	n.c.
19	n.c.	20	n.c.

General Purpose Ports und Debugging (JP9-11)

JP9:

1	GND	2	VDD5
3	LALE	4	LCLK
5	LINTI1	6	LRDY
7	LBHE	8	LWR
9	LRD	10	LCS0
11	LBREQ	12	LHLDA
13	GND	14	LHOLD
15	GND	16	VDD3

JP10:

1	VDD5	2	GND
3	LD0	4	LD1
5	LD2	6	LD3
7	LD4	8	LD5
9	LD6	10	LD7
11	LD8	12	LD9
13	LD10	14	LD11
15	LD12	16	LD13
17	LD14	18	LD15
19	GP8	20	GP9
21	GP10	22	GP11
23	GP12	24	GP13
25	GP14	26	GP15
27	GND	28	VDD3

JP11:

1	VDD5	2	TCK
3	GND	4	TMS
5	GND	6	TDI
7	GND	8	TDO
9	GND	10	TRST

Getting started

The PCISCC is delivered ready for use. The general purpose connectors JP9 and 10 are assembled. If you want to install a modem with high components on modem channel 2, you will need to cut the pins of these two connectors with a cutter. If you want to use JP9 or JP10 simultaneously, please either do not use channel 2 or solder the wires to JP9 and 10 on the back of the PCB. Please note the different order of

channels compared to the USCC>4 (see Fig. 2). Channel 3 is closest to the bracket, while channel 0 is on the outer right side of the board.

Plug the PCISCC into a vacant PCI slot of your computer. Please clean the slot carefully before use. Corroded slot connectors have been the most frequent reason for non-working PCISCC cards during the field tests.

Close the computer case and switch on the computer again. Most BIOS versions list their PCI devices on boot up. This list should now feature a new “Network Controller”. This is the first proof that the card is working correctly.

Operation of the PCISCC card with FlexNet under DOS

This chapter explains the configuration and use of the PCISCC card with the FlexNet software for a true DOS Operating system (no DOS window unter Windows!)

1. Install the FlexNet software by copying all files from the directory DOS/FLEXBIN on the CD-ROM into an appropriate directory on your hard disk. The newest version of the FlexNet software is available from the Internet. (on http://www.afthd.tu-darmstadt.de/~flexnet/). The driver for the PCISCC card is named SCCDRV.EXE.
2. Power up your PC and then start the FlexNet software by typing in the commands printed in `Courier:`

```
FLEXNET          (starts Flexnet)
SCCDRV           (no Parameters)
FLEX             (finish driver load)
FSET MODE 0 76800trzd
                 (set modes for channel 0, repeat with correct parameters for
                 channels 1-3)

BCT              Start application (here the BayCom Terminal, on CD-ROM this can
                  be found in the directory DOS/BCT)
```

Now the divided Screen of the BayCom Terminal should show up. The modem can be tested easily by entering a connect command (:C <destination call> <channel number>) in the upper window or by sending out a few test packets (e.g. on monitor screen; change to monitor screen by pressing F10, then enter correct channel (e.g. :K 0), then delete the : on first cursor position and press ENTER. If the PTT LED on the modem lights up, the principal function of the card should be given.

IMPORTANT NOTE: SCCDRV.EXE cannot be operated in the DOS window of a Windows application, a true DOS operating system is required. The driver is not EMM386 compatibel.

For further explanation of the operation of FlexNet: See FlexNet documentation on the CD-ROM.

Operating the PCISCC with FlexNet32 on Windows NT

The currently available driver is only suited for Windows NT. Other Windows operating systems are not supported.

The CD-ROM only contains the drivers for the PCISCC card, but not the FlexNet32 software or a suitable terminal. Actual versions of these need to be downloaded from the internet.

1. Download FlexNet32 for windows. The actual version is available from: http://www.afthd.tu-darmstadt.de/~flexnet/
2. If you need a terminal program, we would recommend using PAXON. This software is available from: www.paxon.de. Of course also other FlexNet32 compatible applications can be used. Install the terminal according to the PAXOn documentation, before proceeding with the driver installation.
3. Extract the FlexNet package into any suitable directory on your hard disk. Copy the PCISCC drivers from the WIN32 directory of the CD-ROM into the same directory. (files: PCISCC.DLL, PCISCC.SYS). If you cannot see these files in your windows explorer, you need to select “Show System Files” in your EXPLORER options.
4. Start the Flexnet control Center FLEXCTRL.
5. Select sub menue TOOLs and entry PARAMETER. Select the first available channel by double clicking and select “NEW DRIVER” by clicking with the right mouse button. A new window appears. Select “PCISCC” by doubleclicking. If this entry is not present, the drivers were not copied correctly on item 3. (Remind that the drivers need to be in the same directory as the flexnet software). The four SCC channels are now being added in the parameter window.
6. The channel parameters are now being configured by highlighting a channel with the left mouse button and then selecting EDIT with the right button. A normal TCM3105 modem needs to be set to 1200Bd. When using the BayCom FSK modems, the switches NRZ and external TX and RX clock need to be activated. When all settings are correct, the channel is displayed as “activated” in the parameter window (instead of being crossed out).
7. Start the PAXON terminal. Packet operation may begin now.

Operating the PCISCC under Linux

For the PCISCC, DG1KJD has also developed a Linux driver. However, this is not officially supported by BayCom. Installation and use are not that easy, so we would recommend this option for Linux experts only. The driver and more detailed informations are available from the authors home page: http://www.afthd.tu-darmstadt.de/~dg1kjd/.
The Linux driver is only supported by Linux versions 2.2.x, the Kernel needs to be patched, too.

1. After inserting the card and booting, check availability of the card: `cat /proc/pci`. This displays a table with all PCI devices. A slotcard with ID 110a:2102 should be visible, this is the PCISCC.
2. AX.25 support and PCISCC needs to be configured in the kernel. The kernel

needs to be recompiled.
3. Install new Kernel and reboot.
4. `insmod ax25` and `insmod pciscc4`. This adds AX.25 protocol stack and PCISCC drivers.
5. By using `ifconfig –a` all channels can be displayed, output should include dscc[0..3].
6. Driver contains a module named SETPCISCC. This needs to be compiled and installed (`install ax25-tools`)
7. SETPCI allows setting of channel parameters, e.g. `setpciscc –i dscc0 –d txd=192` sets TX-Delay for a 19k2-channel to 10ms. See DG1KJD homepage for detailed explanation.
8. `ifconfig dscc0 hs ax25 dg1kjd-8 mtu 256 up` configures channel 0 with a callsign (in this example: dg1kjd-8) and maximum packet length.
9. Axparms and call sets a route and establishes connection.

Enjoy operation of the PCISCC card!

BayCom

Order Information

Ordering of BayCom goods

All BayCom goods are available via the address provided below. Please ask for actual catalogue and price list.

Order Information:

4900 PCISCC Assembled Unit

Technical Information and Repair Service

Your PCISCC comes with a one year-warranty. Mishandling and external overvoltage (like lightning strokes) are not protected by warranty. We are offering a repair service for all our kits and assembled units.
Please contact Johannes Kneip on baycom@baycom.de before sending items in for repair.
Call ++49 5105 585050 or send a fax to ++49 5105 585060 for any other assistance you need.

BayCom GmbH
Bert–Brecht–Weg 28
D–30890 Barsinghausen
GERMANY

baycom@baycom.de

Your authorized
BayCom–Dealer: