http://www.bintec.de

RELEASE NOTE BIANCA/BRICK-XL

September 3,1997

New System Software: *Release 4.5 Revision 5*

This document describes the new features, enhancements, bugfixes, and changes to the BIANCA / BRICK-XL System Software since Release 4.4 Revision 8.

New in R 4.5.5

Changes in Previous Revisions

Upgrading System Software

1. Retrieve the current system software image from BinTec's HTTP server at http://www.bintec.de.

2. With this image you can upgrade the BIANCA/BRICK-XL with the **update** command from the SNMP shell via a remote host (i.e. using telnet, minipad, or isdnlogin) or by using the **BOOTmonitor** if you are logged in directly on the console.

   Information on using the BOOTmonitor can be found in the *BRICK-XL User's Guide* under *Firmware Upgrades*.

3. Once you've installed Release 4.5 Revision 5 you may want to retrieve the latest documentation (in Adobe's PDF format) which is also available from BinTec's FTP server noted above.

   Note: When upgrading system software, it is also recommended that you use the most current versions of *BRICKware for Windows* and *UNIXTools*. Both can be retrieved from BinTec's FTP server.

What’s New in Revision 5

4.5 Revision 5: Released: 03.09.97

Feature:

ISDN / V.110

BIANCA/BRICK-XL now fully supports asynchronous bit rate adaptation according to V.110 on the CM-BRI, CM-2BRI and CM-2UP0 communication modules.

Asynchronous bit rate adaptation is often used in communication with terminal adapters and for connecting to GSM networks from the ISDN. The BIANCA/BRICK-XL is fully compatible with the V.110 standard.

☞ Please note that depending on the bit rate you can have a different number of V.110 connections at the same time.

Bit Rate	max. # of channels
38,400 bps	4
19,200 bps	8
9,600 bps	15
4,800 bps and lower	20

Configuration

V.110 support can be configured individually for each partner in the *biboPPPLayer1Protocol* variable of the ***biboPPPTable*** (in **Setup Tool** this is the *Layer 1 Protocol* setting in the [*WAN Partner*][*EDIT*][*Advanced Settings*] menu).

v110_1200	V.110 bit rate adaptation (asynchronous 1200 baud, 8,N,1)
v110_2400	V.110 bit rate adaptation

	(asynchronous 2400 baud, 8,N,1)
v110_4800	V.110 bit rate adaptation (asynchronous 4800 baud, 8,N,1)
v110_9600	V.110 bit rate adaptation (asynchronous 9600 baud, 8,N,1)
v110_19200	V.110 bit rate adaptation (asynchronous 19200 baud, 8,N,1)
v110_38400	V.110 bit rate adaptation (asynchronous 38400 baud, 8,N,1)

For outgoing ISDN calls the Layer1Protocol setting is signalled via the D channel.

Reception of Incoming Calls

For ISDN partners that can be identified by the ISDN Calling Party's Number (i.e. outband), the *Layer1Protocol* settings will be adjusted according to their [*WAN Partner*] / ***biboPPPTable*** entry (see previous section).

If the caller can not be identified by his calling number, identification must be performed "inband" using PPP. The setting of the *Layer1Protocol* for incoming calls can then be performed in two different ways.

- In most cases you can specify a generic ppp entry for the ISDN number you wish to accept incoming PPP and V.110 calls on (from the SNMP shell this is done in the ***isdnDispatchTable***, setting *Item* to **ppp** and *LocalNumber* to the ISDN number for routing; in Setup Tool go to the slot of your ISDN interface, and then in the [*Incoming Call Answering* >] menu add an entry with *Item* set to **PPP (routing)** and *Number* set to the local ISDN number for routing).

  The Layer 1 parameters are then automatically taken from the D channel signalling elements of the incoming call.

- In some cases these signalling elements are not transmitted correctly (e.g. through certain switching stations and PABXs). You can then setup a different dispatch entry with a different ISDN number for each V.110 speed you wish to support (as described above). The following values for *Item* are supported:

ppp_v110_1200
ppp_v110_2400
ppp_v110_4800
ppp_v110_9600
ppp_v110_19200
ppp_v110_38400

Enhancement:

HTTP password / Setup Tool

The separate HTTP password (see notes to Rel. 4.5.1 beta) can now also be configured from the Setup Tool. It's the *HTTP Server Password* entry in the [*System*] menu.

Bugfixes

CAPI

- With the introduction of the ***capiMultiControllerTable*** there occured an error. When a CAPI application requested a connection to CAPI controller *n*, the reply contained a PLCI with controller number *n+1*, which was not accepted by some applications.
  The PLCI now contains the correct controller number *n*.
- The system rebooted, when it received incorrectly coded DATA_B3_REQ messages from an RCAPI application, because the message length was not checked.
  This is no longer the case.
- The system also rebooted, when CAPI2_RESET_B3_REQ messages contained an empty NCPI field.
  This bug was fixed.

Modem

The stability of the modem software was significantly improved.

In particular the following problems have been solved:

- The system rebooted in irregular intervals.
- Single modem CPUs occasionally stopped dead. In this case the corresponding *State* entry in the **modem status** listing remained fixed at either **connected**, **called**, or **calling**, and the modem was no longer available for any calls.

If you want to update your FML-8MOD modem modules to the current software revision, you will have to download the file **csm_455.csm** from the BRICK-XL section of our WWW server, and install it with the *modem update* command, as described in the separate FML-8MOD release note.

PPP

- Encapsulation x25_ppp: The appropriate ***biboPPPLinkTable*** entries will now be deleted after closing a dialup connection. These entries were not deleted in Rel. 4.5.3.
- *OperStatus* in the ***IfTable***: In releases 4.5.1 to 4.5.3 this state was immediately changed from **blocked** to **dormant**, ignoring the *biboPPPBlockTime*, which resulted in constant connection setup retries.
  The *biboPPPBlockTime* is now handled correctly.
- (Re)loading configuration files: If the ***IfTable*** contains entries where *AdminStatus*=**dialup**, this state will now be changed to *AdminStatus*=**up** when loading the file.
  In previous releases the *AdminStatus* remained **dialup**, effectively blocking outgoing connections on these interfaces after a reload.

SNMP

- If no IPX license was installed, SNMP used up more and more memory.
  SNMP now correctly releases all memory areas it no long-

er needs, and additionally makes a »garbage collection« of the dynamically allocated memory areas from time to time. This is indicated by a special Syslog Message.

X.25

- Closed User Groups / minipad: All Closed User Group parameters of the minipad program now also accept values from 100 to 1000.

Changes in previous Releases

4.5 Revision 3: Released: 01.08.97

Features:

CAPI

CAPI Modem connection support

Both CAPI 1.1 and CAPI 2.0 now support connections using any modem built-in. At the moment the parameters of modem profile 1 from the ***mdmProfileTable*** (also accessible via the [*MODEM*] menu of the Setup Tool) are used.

Full support of CAPI modem parameters (B2 Configuration for B2 protocol 7, NCPI for B3 protocol 7) will be available in an upcoming release.

New capiMultiControllerTable

A new table, the ***capiMultiControllerTable***, was added to the CAPI group to enable the use of CAPI with different ISDN controllers at the same time.

This table contains mappings between controller numbers used by CAPI applications and the ISDN stacks available on the BRICK (i.e., the *Number* field of the ***isdnStkTable***). The Version field specifies whether an entry applies to a **capi11** or **capi20** application.

If no CAPI 1.1 entry is defined, CAPI 1.1 applications are assigned *isdnStkNumber* **n** where n is the controller number requested by the application.

If no CAPI 2.0 entry is defined, CAPI 2.0 applications are assigned *isdnStkNumber* **n-1** where n is the controller number requested by the application.

Creating entries: Entries are created by assigning a value to the *capiControllerNumber* object.

Deleting entries: An entry can be removed by assigning the value **delete** to its *capiControllerVersion* object.

The fields of the table have the following meaning:

Number The controller number requested by the CAPI application.

StkMask This binary number defines the ISDN stack(s) to use for the specified CAPI 1.1 or CAPI 2.0 applications. Each bit corresponds to one entry (stack) in the ***isdnStkTable***, the rightmost bit selects entry 0, the next bit selects entry 1, and so forth. For example, **Number=1 StkMask=0b1101 Version=capi11** means: allow CAPI 1.1 applications requesting ISDN controller 1 to use ISDN stacks 0, 2and 3.

Version Specifies which CAPI applications (version 1.1, or 2.0) this entry applies to.
Set this field to **delete** to delete this entry.

PPP

New x25_ppp_opt encapsulation

There is a new encapsulation type, *x25_ppp_opt*, which can be used in the ***biboPPPTable***. It provides a special case of the *x25_ppp* encapsulation.

This encapsulation enables your BRICK to determine whether an incoming call is an X.25 call or a PPP call even if no outband authentication (by CLID) is possible. This is done by scanning the first incoming data packet.

Dial-in partners which can not be authenticated outband (CLID) will then be given an X.25 connection via ISDN, or optionally a PPP connection, when they can be authenticated inband by using CHAP or PAP.

⚠ Concurrent use of X.25 and PPP encapsulation is not possible.

Note that you will need one WAN partner definition for X.25, where the *x25_ppp_opt* encapsulation is selected, and one or more for PPP connections (authentication via PAP, CHAP or RADIUS).

New field Layer2Mode

The variable Layer2Mode was added to the ***biboPPPTable***.

This object specifies the layer 2 mode to be used for a connection. It is only relevant, if the Encapsulation involves a LAPB protocol, this is the case for *x25*, *x25_ppp*, *x25_ppp_opt*, *ip_lapb*, *lapb*, *x31_bchan*, *x75_ppp*, *x75btx_ppp*, and *x25_nosig*.

The default value of this object is *auto*.

For dialup connections, the layer 2 mode will then be DTE on the calling side and DCE on the called side.

For leased lines the layer 2 mode is set at lower layers (for example *isdnChType* in the ***isdnChTable***). When this object is set to *dte* or *dce*, the layer 2 mode will always be DTE or DCE, regardless of the call direction or the settings at the lower layer.

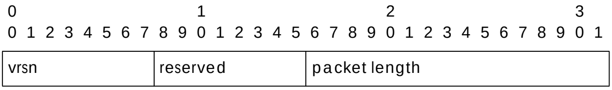
PPP Modem connections

PPP now also accepts modem connections signalled as »ISDN speech 3.1 kHz«. These capabilities are signalled by some PBXs with analog ports.

X.25 / TCP-TP0 Bridge

Enhancements for RFC 1006 support

- The maximum packet length is now 65535 bytes including the header instead of just 8191 bytes.
- The transport packet header can now contain additional information for special X.25 packet types.



When transmitting packets where the *q bit is set* (=1), *vrsn* has to be set to **255** (instead of the default value 3).
For X.25 *reset request/indication* packets the *vrsn* field is set to **255** and the *reserved* field is set to **1**.
For X.25 *reset confirmation* packets the *vrsn* field is set to **255** and the *reserved* field is set to **2**.

A received X.25 *reset indication* packet contains 8 octets of data. The 4th octet contains the X.25 cause code and the 8th octet contains the X.25 diagnostic code. Data contained in transmitted reset packets is ignored.
Reset confirmation packets contain no additional data, so the packet length of such packets is always set to 4.

Enhancements for RFC 1086 Support

X.25 on your BRICK now also supports extended addresses of address type=4 (standard addresses have type=3). Type 4 allows for variable length address fields and facilities.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+---------------+---------------+
| address type                  |X.25 addr. length| X.25 address|
+-------------------------------+---------------+               |
|       ...             ...             ...             ...     |
|       ...             ...             ...             ...     |
|       ...             ...             ...             ...     |
|       ...             ...             ...     +---------------+
|       ...             ...             ...     |Protocol ID length|
+-----------------------------------------------+---------------+
|Protocol ID            ...             ...             ...     |
+---------------+-----------------------------------------------+
|CDUF length    |Call User Data field   ...             ...     |
+---------------+                                               |
|       ...             ...             ...             ...     |
|       ...             ...             ...             ...     |
|       ...             ...             ...             ...     |
+---------------+-----------------------------------------------+
|Facility length|X.25 Facilities        ...             ...     |
+---------------+                                               |
|       ...             ...             ...             ...     |
+---------------------------------------------------------------+
```

The address fields have the following meaning:

address type (2 octets)

A binary-encoded value in network order indicating the address type. The value 4 is used for extended X.25 addressing of this format.

X.25 addr. length (1 octet)

A binary-encoded value in network order indicating how many octets of the X.25 address there are.

X.25 address

The ASCII-encoded value of the X.25 address. Maximum length of 55 bytes allowed.

Protocol ID length (1 octet)

A binary-encoded value indicating the number of protocol ID octets there are.

Protocol ID

Meaningful at the remote system.

CUDF length (1 octet)

A binary-encoded value indicating the number of User Data octets there are.

Call User Data field

Meaningful at the remote system.

X.25 Facility Length (1 octet)

A binary-encoded value indicating the number of X.25 Facility octets there are.

X.25 Facilities

Meaningful at the remote system.

Closed User Group support

The Closed User Group (CUG) feature of X.25 is now supported. For this purpose the three variables *Cug*, *CugOutgoing*, and *CugBilateral* were added to both the ***x25RouteTable*** and the ***x25RewriteTable***. The fields have the following meaning:

Cug

Closed User Group. Connections are only possible to DTEs in the same CUG. *Cug* can take values of -1 (*default*) if you do *not* want to use this feature, and 0-9999 for closed user groups.

In the ***x25RewriteTable*** the value -2 is also possible, meaning that the *Cug* field of a call request packet is cleared on rewriting.

CugOutgoing

Closed User Group with outgoing access. Connections are also possible to DTEs not in the same CUG. *CugOutgoing* can take values of -1 (*default*) if you do *not* want to use this feature, and 0-9999 for closed user groups.

In the ***x25RewriteTable*** the value -2 is also possible, meaning that the *CugOutgoing* field of a call request packet is cleared on rewriting.

CugBilateral

Bilateral Closed User Group. Connections are only possible between exactly two DTEs. *CugBilateral* can take values of -1 (*default*) if you do *not* want to use this feature, and 0-9999 for closed user groups.

In the ***x25RewriteTable*** the value -2 is also possible, meaning that the *CugBilateral* field of a call request packet is cleared on rewriting.

New minipad options

To support the Closed User Group feature, three call options were added to the *minipad* application available on your BRICK.

-*c ‹cug›* Closed user group.
Possible values for ‹*cug*›: 0-9999.

-*o ‹outgocug›*

Closed user group with outgoing access.
Possible values for ‹*outgocug*›: 0-9999.

-*b ‹bcug›*

Bilateral Closed user group.
Possible values for ‹*bcug*›: 0-9999.

Miscellaneous

- The *RewritingRule* in the ***x25RewriteTable*** can now take values of up to 999999 instead of 9999.
- The **debug x.25** command now outputs extensive information concerning layer 2 and layer 3 events on the existing links.

Bugfixes:

BootP

- The forwarding of BootP messages now also works with the CM-TR Token Ring module.

CAPI

- When receiving a fax message the memory area allocated for that purpose was not returned to the system properly, resulting in less and less available user memory.
  This bug has been fixed.

- When you used the autoconfiguration feature of the ISDN interface (*Autoconfig*=**on** in the ***isdnIfTable***) your CAPI configuration in the ***capiConfigTable*** was overwritten on each system start.
  This is no longer the case.

HTTP

- The graphics on the BRICK's HTTP status page were sometimes not displayed correctly. They are now displayed correctly.

IP

- You could not establish TCP connections to Windows PCs running Version 2.0 of the OnNet32 TCP/IP stack software of Ftp Software, Inc.
  This problem no longer occurs.

ISDN / CM-EBRI

- Incoming fax messages were sometimes not received correctly. This behaviour has been improved.

Modem

- The default modem profile (profile 1) was not set correctly for the isdnlogin service.

- Characters were lost when using low data rates (up to 9600 bps).

These bugs have been fixed.

PPP

- Leased lines: When changing or deleting bundle configurations in conjunction with leased lines the system occasionally rebooted. After creating a new bundle it was necessary to reset the system to actually use it.
  These problems have been fixed.
- Multilink PPP: The *ifOutOctets* counter in the ***ifTable*** now displays the correct number of transmitted bytes.
- SPVs (»Semi-permanente Verbindungen« of the german 1TR6 ISDN protocol) are now fully supported.
- VJHC: The Van Jacobson Header Compression now also works correctly in connection with Windows PCs.

V.42bis on Leased Lines

- When you used V.42bis data compression on a leased line, and the leased line failed, it could happen that after the leased line was back up the data connection could not be reestablished.
  This is no longer the case.

X.25

- Encapsulation x25 / x25_ppp: When using the minipad application in connection with this encapsulation the system sometimes rebooted after closing the minipad session.
  This problem was fixed.
- RFC 1086 Support: The Protocol ID Length field of X.25 packet headers is now correctly evaluated. This means that you can set the Protocol ID Length field to 0 and use the following 16 bytes for Call User Data.

4.5 Revision 1 beta: Released: 18.07.97

Features:

New mdmProfileTable for Modem Support

The new ***mdmProfileTable*** in the modem group contains eight modem profiles which can be configured to fit different modem speeds and types. The ***isdnDispatchTable*** and ***biboPPPTable*** were also slightly modified to support modems.

For a detailed description of these changes please refer to the separate FML-8MOD release note.

Support for Modem Configuration in Setup Tool

The Setup Tool has also been enhanced to support modem configuration.

There is a new [*MODEM*] menu, where you can configure the eight modem profiles mentioned above.

The [*Incoming Call Answering*] menu has changed its appearance. It now contains a list of incoming number entries, instead of offering just a rigid mask with checkboxes.

The [*WAN Partner*] [*Advanced Settings*] menu now sports a new *Layer 1 Profile* entry.

For a detailed description of these changes please refer to the separate FML-8MOD release note.

ISDN / DDI

Direct Dialling In (DDI) is now supported for point-to-point ISDN accesses (Anlagenanschluß in Germany).

This is especially useful when you use S_{2M} modules in connection with the new FML-8MOD modem modules.

With DDI the BRICK can collect the digits of the called party number until a matching entry in the ***isdnDispatchTable*** is found. When the call is initiated from an analog device the digits usually arrive one by one.

☞ You have to set the *Mode* in the ***isdnDispatchTable*** to **left_to_right** (2) to enable DDI.

For an explanation of how to configure this feature from the Setup Tool please refer to the separate FML-8MOD release note.

New separate HTTP password

The ***bintecsec*** table now also contains a *biboAdmHttpPassword*, which has to be entered when you want to view system tables from the http status page of your BRICK. The http password defaults to **bintec**.

⚠ After entering this password, you can view all system tables apart from the ***bintecsec*** table from a WWW browser; therefore you should change the password from its default value (this has to be done from the SNMP shell at the moment), because otherwise anybody knowing the **bintec** password can spy out your system configuration.

☞ Some WWW browsers require that you enter a user name, before accepting the http password.
If this is the case, please use **http** as a user name.

New sysPCMTable for S_{2M} BusMaster Priority Settings

You will only need this table if you installed more than one S_{2M} module in your BRICK, and one of these modules is not connected to the ISDN.

You can then use the *MasterPrio* variable to specify the PCM highway bus master (set it to a value >0).

⚠ Under normal circumstances you will *never* need to modify this table, everything will run correctly automatically.

New Fan control and status tables

There are five new entries in the ***sysConfig*** table, which show the status of the different fans in your BRICK-XL and determine the operating mode of the fans.

The *sysConfigFanControl* variable can take the following two values:

Value	Description
auto	The fan speed will be set to *high* if the CPU temperature exceeds 50 degrees centigrade, and will be set back to *low* when the temperature drops to below 40 degrees centigrade again.
high	The fans always run at high speed.

☞ If you have FML-8MOD modem modules installed in your BRICK-XL, we recommend that you set *sysConfigFanControl* to **high**.

The *sysConfigFan1* to *sysConfigFan4* variables show the state of the fans in your BRICK.

Fans 1 and 2 are the ones in the power supply, fan 3 is near the CPU, and fan 4 will only be present if you have modem modules installed. The variables will either be **on**—fan is operating normally—or **off**—fan is disconnected or non-functional. If you do not have modems installed in your BRICK, *sysConfigFan4* will alway be **off**.

☞ This will only work with hardware revisions 1.3 and later.

Bugfixes

Ethernet

Several problems regarding the ethernet driver were fixed.

- Occasionaly duplicate packets were produced.
- The sender rarely ran into a blocked state.
- Under certain circumstances the ethernet response got very slow (e.g. ping reply times of up to 1000ms).

All these problems no longer occur.