bdiRDI

JTAG debug interface for RDI compatible debuggers

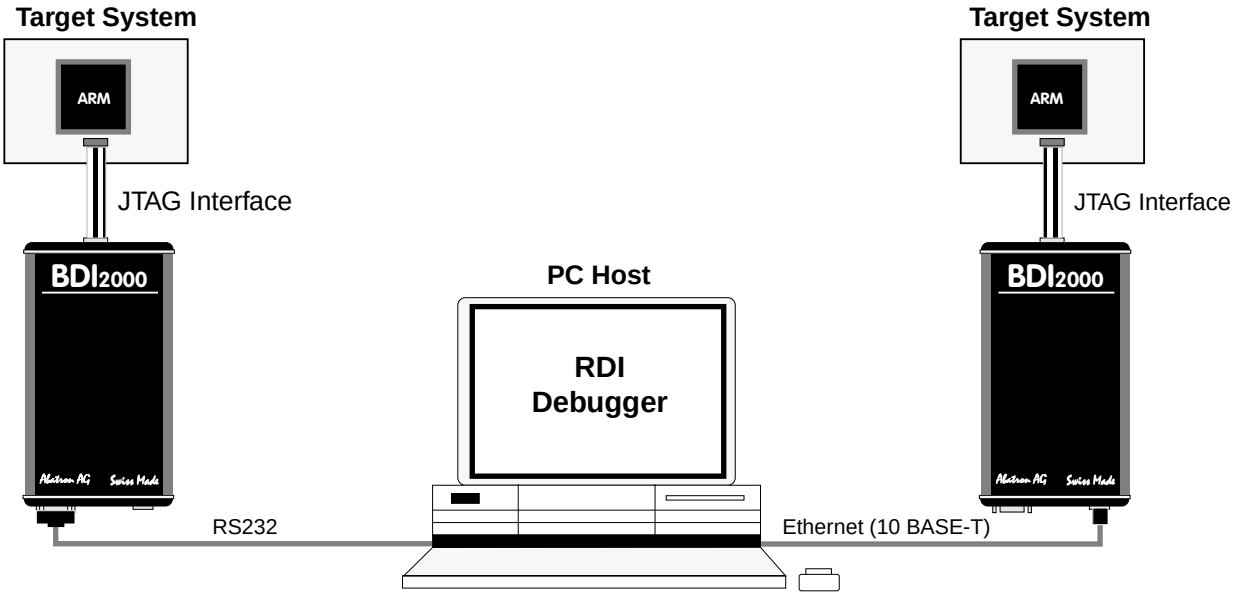
ARM11

User Manual

Manual Version 1.00 for BDI2000

Appendices

1 Introduction



The BDI2000 adds JTAG based debugging to RDI compatible debuggers (e.g. AXD from ARM Ldt). With the BDI2000, you control and monitor the microcontroller solely through the stable on-chip debugging services. You won't waste time and target resources with a software ROM monitor, and you eliminate the cabling problems typical of ICE's. This combination runs even when the target system crashes and allows developers to continue investigating the cause of the crash. A RS232 interface with a maximum of 115 kBaud and a 10Base-T Ethernet interface is available for the host interface. The configuration software is used to update the firmware and to configure the BDI2000 so it works with the RDI compatible debugger.

1.1 BDI2000

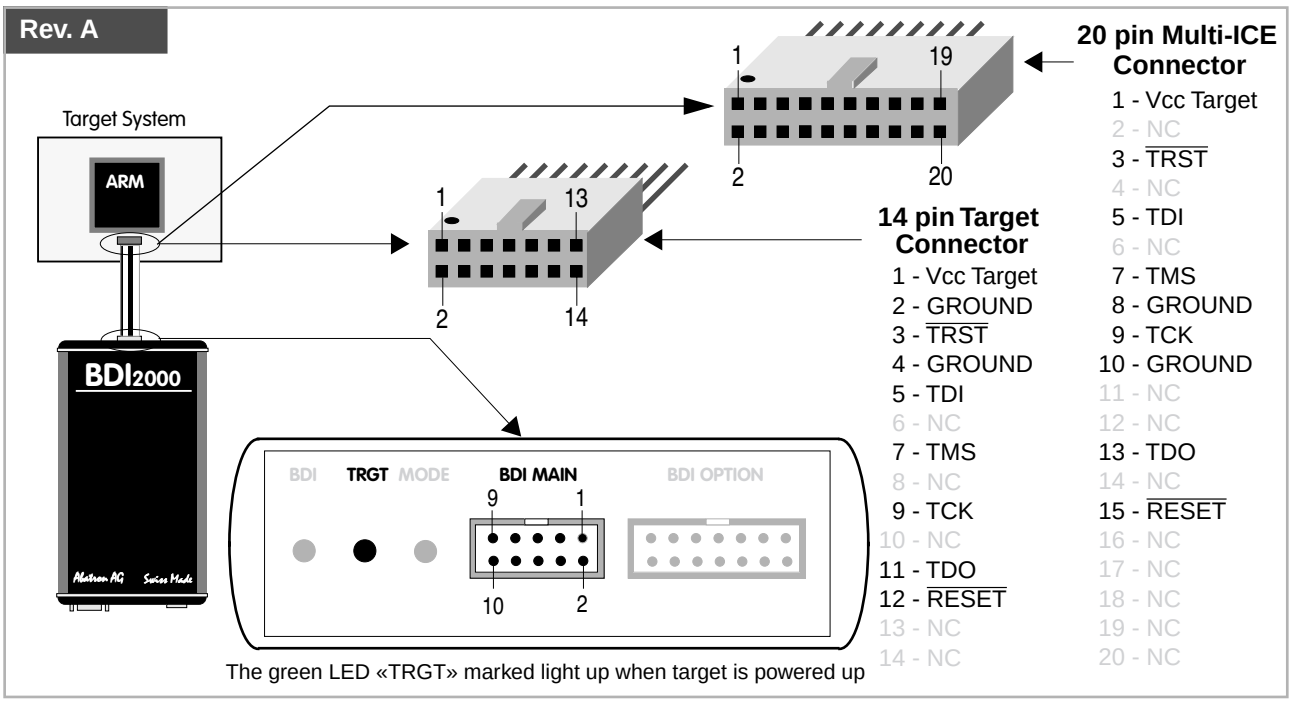
The BDI2000 is a processor system in a small box. It implements the interface between the BDM/JTAG pins of the target CPU and a 10Base-T Ethernet / RS232 connector. BDI2000 is powered by a MC68360, 512Kbyte RAM and a flash memory of 1024Kbyte. As a result of consistent implementation of lasted technology, the BDI2000 is optimally prepared for further enhancements. The firmware and the programmable logic of the BDI2000 can be updated by the user with a simple Windows based configuration program. The BDI2000 supports 1.8 – 5.0 Volts target systems (3.0 – 5.0 Volts target systems with Rev. A/B).

2 Installation

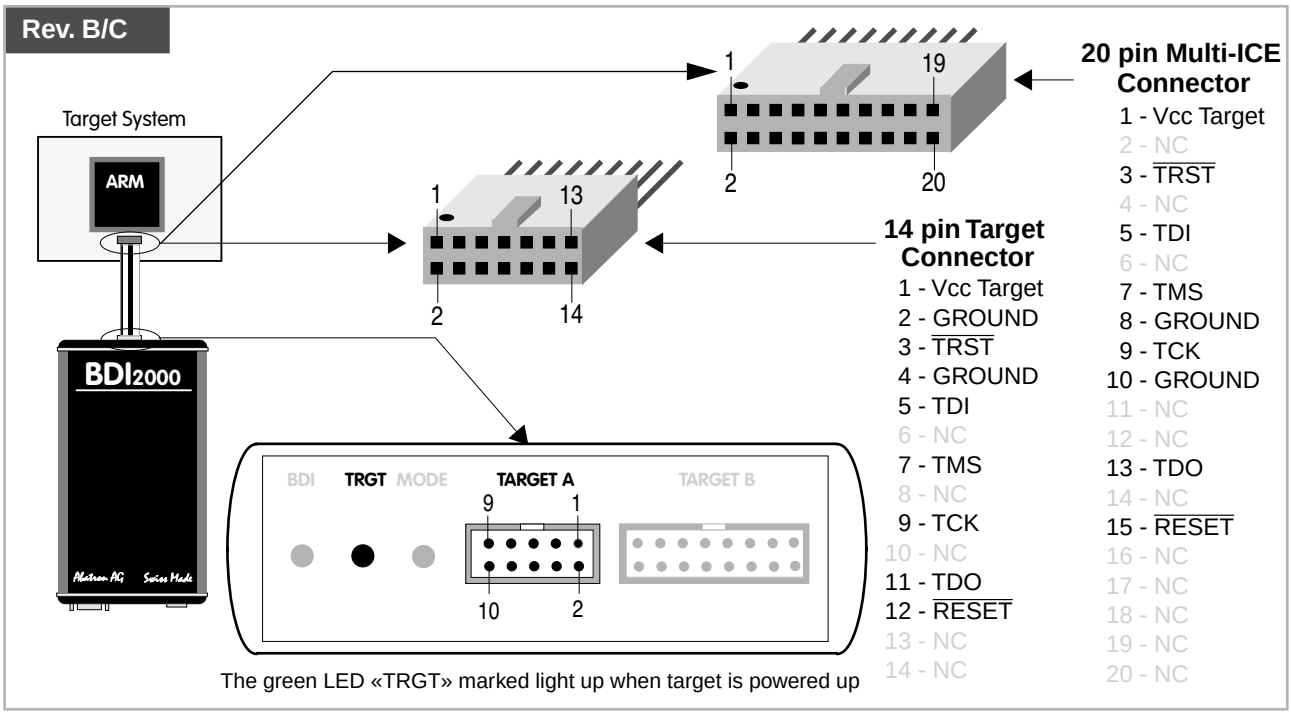
2.1 Connecting the BDI2000 to Target

The enclosed cables to the target system are designed for the ARM Development Boards. In case where the target system has the same connector layout, the cable (14 pin or 20 pin) can be directly connected.

In order to ensure reliable operation of the BDI (EMC, runtimes, etc.) the target cable length must not exceed 20 cm (8").



The green LED «TRGT» marked light up when target is powered up



The green LED «TRGT» marked light up when target is powered up

For BDI MAIN / TARGET A connector signals see table on next page.

BDI MAIN / TARGET A Connector Signals

Pin	Name	Describtion
1	reserved	This pin is currently not used.
2	$\overline{\text{TRST}}$	**JTAG Test Reset** This open-drain / push-pull output of the BDI2000 resets the JTAG TAP controller on the target. Default driver type is open-drain.
3+5	GND	**System Ground**
4	TCK	**JTAG Test Clock** This output of the BDI2000 connects to the target TCK line.
6	TMS	**JTAG Test Mode Select** This output of the BDI2000 connects to the target TMS line.
7	$\overline{\text{RESET}}$	This open collector output of the BDI2000 is used to reset the target system.
8	TDI	**JTAG Test Data In** This output of the BDI2000 connects to the target TDI line.
9	Vcc Target	**1.8 – 5.0V:** This is the target reference voltage. It indicates that the target has power and it is also used to create the logic-level reference for the input comparators. It also controls the output logic levels to the target. It is normally fed from Vdd I/O on the target board. **3.0 – 5.0V with Rev. A/B :** This input to the BDI2000 is used to detect if the target is powered up. If there is a current limiting resistor between this pin and the target Vdd, it should be 100 Ohm or less.
10	TDO	**JTAG Test Data Out** This input to the BDI2000 connects to the target TDO line.

The BDI2000 works also with targets which have no dedicated $\overline{\text{TRST}}$ pin. For this kind of targets, the BDI cannot force the target to debug mode immediately after reset. The target always begins execution of application code until the BDI has finished programming the Debug Control Register.

2.1.1 Changing Target Processor Type

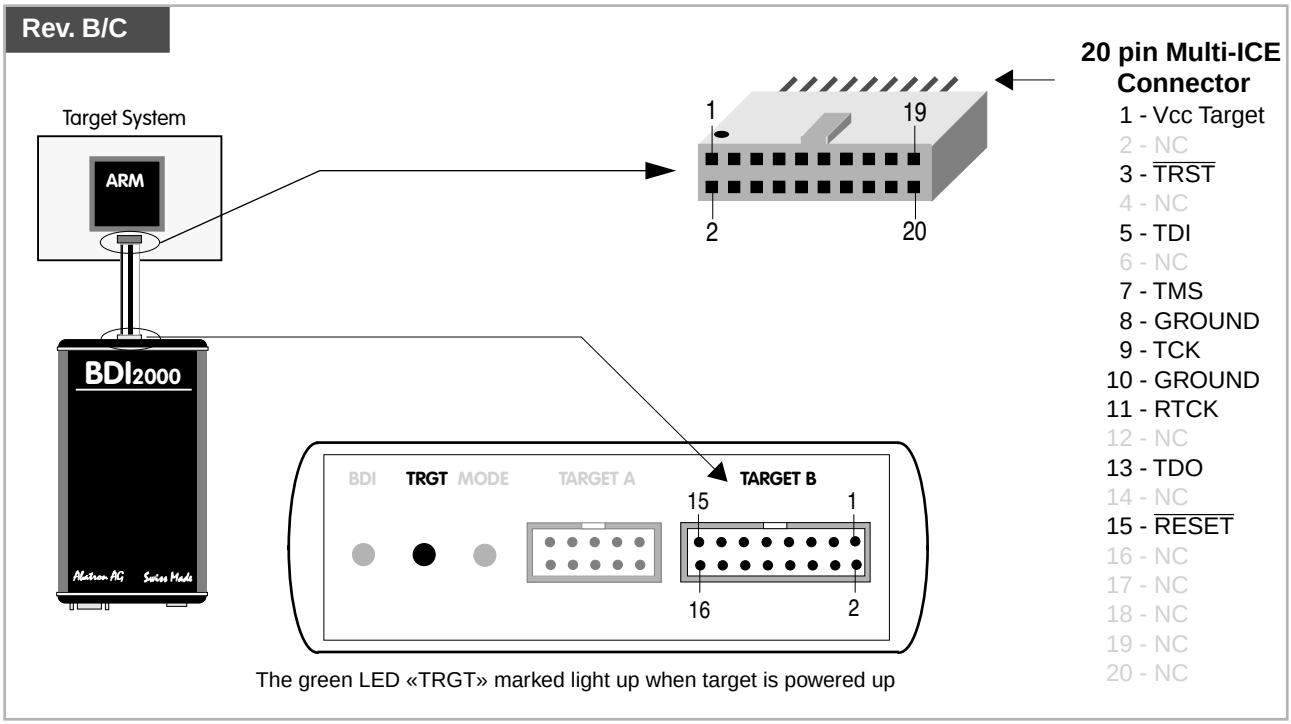
Before you can use the BDI2000 with an other target processor type (e.g. ARM <--> PPC), a new setup has to be done (see Appendix A). During this process the target cable must be disconnected from the target system. The BDI2000 needs to be supplied with 5 Volts via the BDI OPTION connector (Rev. A) or via the POWER connector (Rev. B/C). For more information see chapter 2.2.1 «External Power Supply».

To avoid data line conflicts, the BDI2000 must be disconnected from the target system while programming the logic for an other target CPU.

2.1.2 Adaptive Clocking

Adaptive clocking is a feature which ensures that the BDI2000 never loses synchronization with the target device, whatever the target clock speed is. To achieve this, BDI2000 uses two signals TCK and RTCK. When adaptive clocking is selected, BDI2000 issues a TCK signal and waits for the Returned TCK (RTCK) to come back. BDI2000 does not progress to the next TCK until RTCK is received. For more information about adaptive clocking see ARM documentation.

Note:
Adaptive clocking is only supported with BDI2000 Rev.B/C and a special target cable. This special cable can be ordered separately from Abatron.



The green LED «TRGT» marked light up when target is powered up

For TARGET B connector signals see table on next page.

BDI TARGET B Connector Signals:

Pin	Name	Describtion
1	TDO	**JTAG Test Data Out** This input to the BDI2000 connects to the target TDO line.
2	reserved	
3	TDI	**JTAG Test Data In** This output of the BDI2000 connects to the target TDI line.
4	reserved	
5	RTCK	**Returned JTAG Test Clock** This input to the BDI2000 connects to the target RTCK line.
6	Vcc Target	**1.8 – 5.0V:** This is the target reference voltage. It indicates that the target has power and it is also used to create the logic-level reference for the input comparators. It also controls the output logic levels to the target. It is normally fed from Vdd I/O on the target board. **3.0 – 5.0V with Rev. A/B :** This input to the BDI2000 is used to detect if the target is powered up. If there is a current limiting resistor between this pin and the target Vdd, it should be 100 Ohm or less.
7	TCK	**JTAG Test Clock** This output of the BDI2000 connects to the target TCK line.
8	$\overline{\text{TRST}}$	**JTAG Test Reset** This open-drain / push-pull output of the BDI2000 resets the JTAG TAP controller on the target. Default driver type is open-drain.
9	TMS	**JTAG Test Mode Select** This output of the BDI2000 connects to the target TMS line.
10	reserved	
11	reserved	
12	GROUND	**System Ground**
13	$\overline{\text{RESET}}$	**System Reset** This open collector output of the BDI2000 is used to reset the target system.
14	reseved	
15	reseved	
16	GROUND	**System Ground**

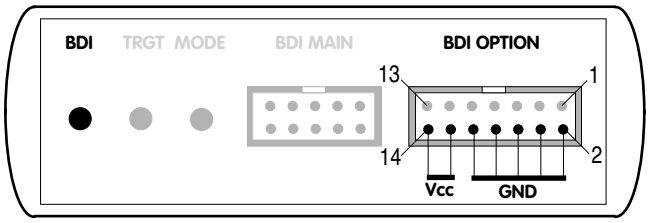
2.2 Connecting the BDI2000 to Power Supply

2.2.1 External Power Supply

The BDI2000 needs to be supplied with 5 Volts (max. 1A) via the BDI OPTION connector (Rev. A) or via POWER connector (Rev. B/C). The available power supply from Abatron (option) or the enclosed power cable can be directly connected. In order to ensure reliable operation of the BDI2000, keep the power supply cable as short as possible.

For error-free operation, the power supply to the BDI2000 must be between 4.75V and 5.25V DC. **The maximal tolerable supply voltage is 5.25 VDC. Any higher voltage or a wrong polarity might destroy the electronics.**

Rev. A

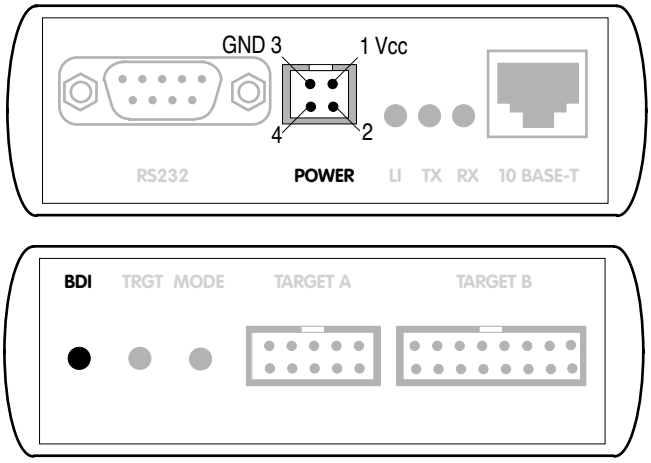


The green LED «BDI» marked light up when 5V power is connected to the BDI2000

BDI OPTION Connector

1 - NOT USED
2 - GROUND
3 - NOT USED
4 - GROUND
5 - BREAK-ENA
6 - GROUND
7 - NOT USED
8 - GROUND
9 - BREAK-A
10 - GROUND
11 - BREAK-B
12 - Vcc (+5V)
13 - Vcc Target (+5V)
14 - Vcc (+5V)

Rev. B/C



The green LED «BDI» marked light up when 5V power is connected to the BDI2000

POWER Connector

1 - Vcc (+5V)
2 - VccTGT
3 - GROUND
4 - NOT USED

Please switch on the system in the following sequence:

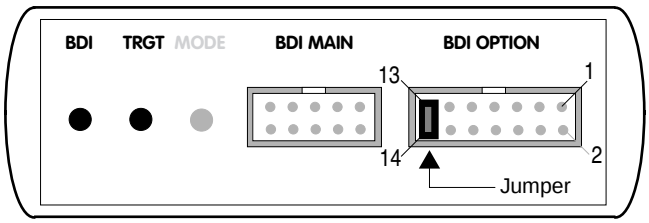
- 1 --> external power supply
- 2 --> target system

2.2.2 Power Supply from Target System

The BDI2000 needs to be supplied with 5 Volts (max. 1A) via BDI MAIN target connector (Rev. A) or via TARGET A connector (Rev. B/C). This mode can only be used when the target system runs with 5V and the pin «Vcc Target» is able to deliver a current up to 1A@5V. For pin description and layout see chapter 2.1 «Connecting the BDI2000 to Target». Insert the enclosed Jumper as shown in figure below. **Please ensure that the jumper is inserted correctly**.

For error-free operation, the power supply to the BDI2000 must be between 4.75V and 5.25V DC. **The maximal tolerable supply voltage is 5.25 VDC. Any higher voltage or a wrong polarity might destroy the electronics.**

Rev. A

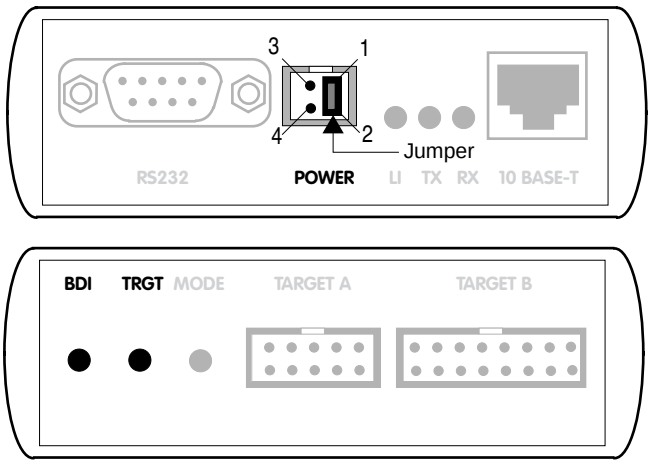


The green LEDs «BDI» and «TRGT» marked light up when target is powered up and the jumper is inserted correctly

BDI OPTION Connector

1 - NOT USED
2 - GROUND
3 - NOT USED
4 - GROUND
5 - BREAK-ENA
6 - GROUND
7 - NOT USED
8 - GROUND
9 - BREAK-A
10 - GROUND
11 - BREAK-B
12 - Vcc (+5V)
13 - Vcc Target (+5V)
14 - Vcc BDI2000 (+5V)

Rev. B/C



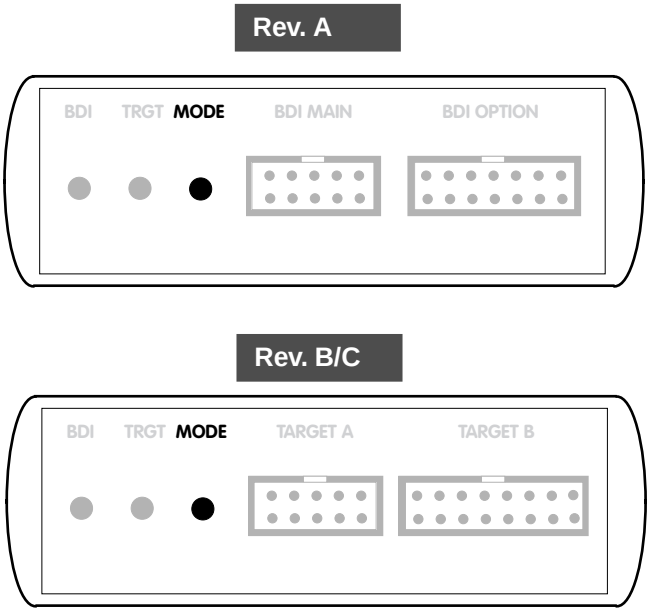
The green LEDs «BDI» and «TRGT» marked light up when target is powered up and the jumper is inserted correctly

POWER Connector

1 - Vcc BDI2000 (+5V)
2 - Vcc Target (+5V)
3 - GROUND
4 - NOT USED

2.3 Status LED «MODE»

The built in LED indicates the following BDI states:



MODE LED	BDI STATES
OFF	The BDI is ready for use, the firmware is already loaded.
ON	The power supply for the BDI2000 is < 4.75VDC.
BLINK	The BDI «loader mode» is active (an invalid firmware is loaded or loading firmware is active).

2.4 Connecting the BDI2000 to Host

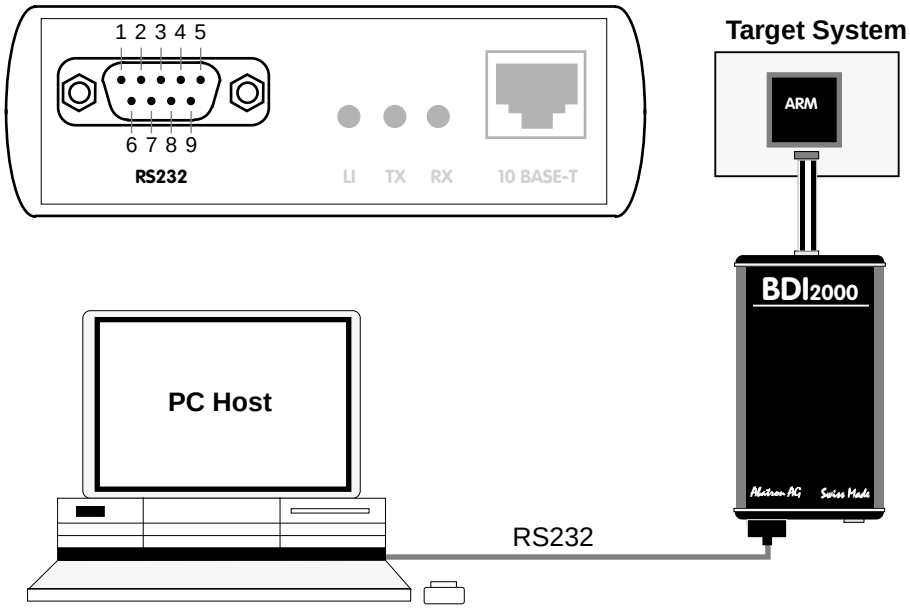
2.4.1 Serial line communication

The host is connected to the BDI through the serial interface (COM1...COM4). The communication cable between BDI and Host is a serial cable (RXD / TXD are crossed). There is the same connector pinout for the BDI and for the Host side (Refer to Figure below).

Rev. A

RS232 Connector
(for PC host)

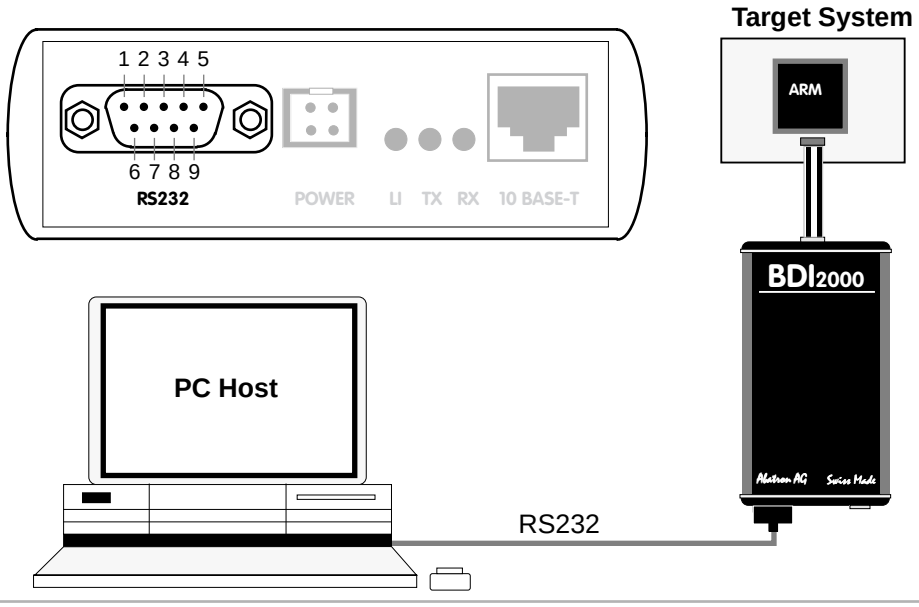
1 - NC
2 - RXD data from host
3 - TXD data to host
4 - NC
5 - GROUND
6 - NC
7 - NC
8 - NC
9 - NC



Rev. B/C

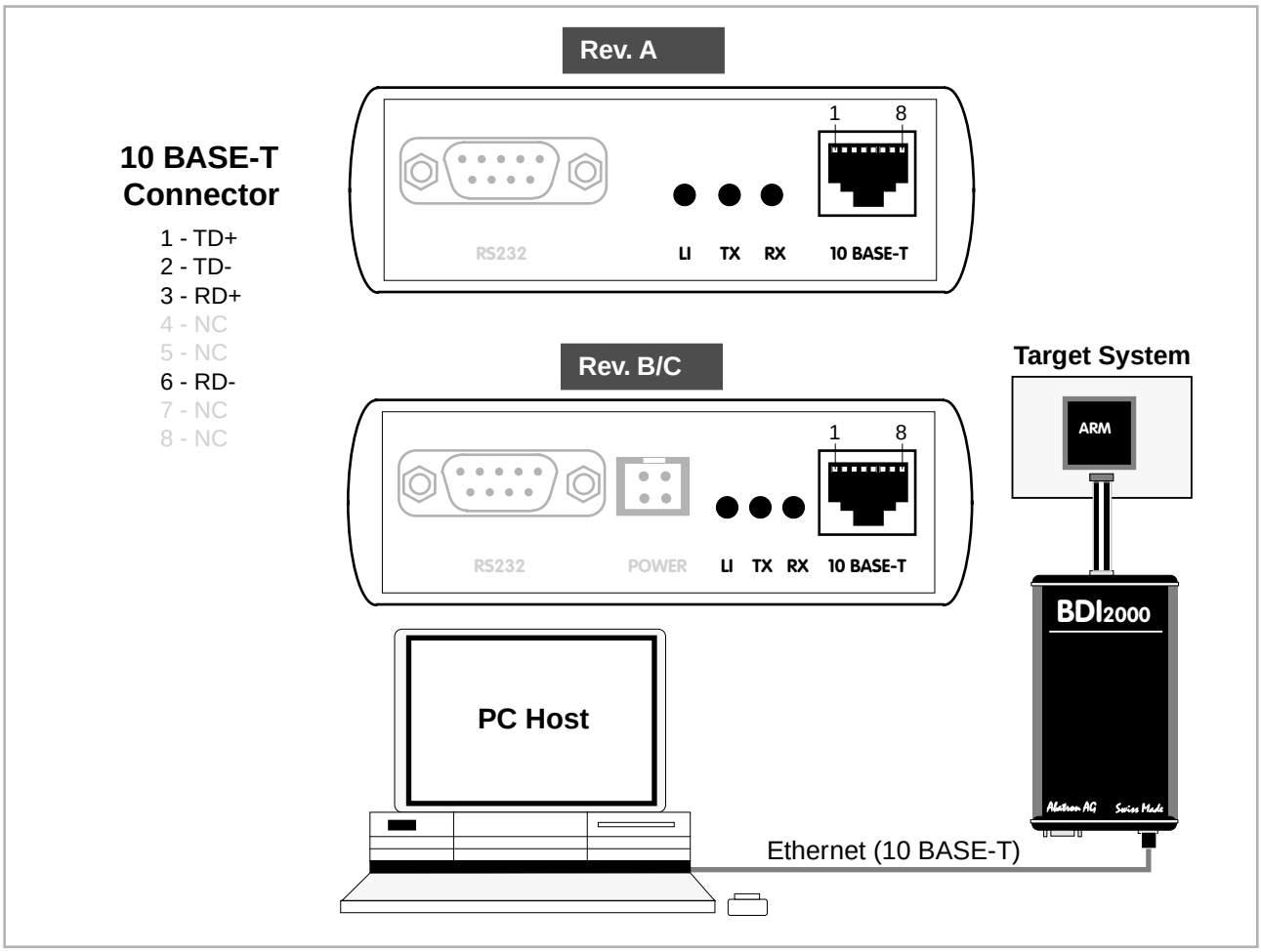
RS232 Connector
(for PC host)

1 - NC
2 - RXD data from host
3 - TXD data to host
4 - NC
5 - GROUND
6 - NC
7 - NC
8 - NC
9 - NC



2.4.2 Ethernet communication

The BDI2000 has a built-in 10 BASE-T Ethernet interface (see figure below). Connect an UTP (Unshilded Twisted Pair) cable to the BD2000. For thin Ethernet coaxial networks you can connect a commercially available media converter (BNC-->10 BASE-T) between your network and the BDI2000. Contact your network administrator if you have questions about the network.



The following explains the meanings of the built-in LED lights:

LED	Name	Description
LI	Link	When this LED light is ON, data link is successful between the UTP port of the BDI2000 and the hub to which it is connected.
TX	Transmit	When this LED light BLINKS, data is being transmitted through the UTP port of the BDI2000
RX	Receive	When this LED light BLINKS, data is being received through the UTP port of the BDI2000

2.5 Installation of the Configuration Software

On the enclosed diskette you will find the BDI configuration software and the firmware required for the BDI. Copy all these files to a directory on your hard disk.

The following files are on the diskette:

b20a11.exe	Configuration program
b20a11.hlp	Helpfile for the configuration program
b20a11fw.xxx	Firmware for BDI2000 for ARM targets
armjed20.xxx	JEDEC file for the BDI2000 (Rev. A/B) logic device programming
armjed21.xxx	JEDEC file for the BDI2000 (Rev. C) logic device programming
bdiifc32.dll	BDI Interface DLL for configuration program
bdirdi.dll	RDI Interface DLL
*.bdi	Configuration Examples

Example of an installation process:

- Copy the entire contents of the enclosed diskette into a directory on the hard disk.
- You may create a new shortcut to the b20a11.exe configuration program.
- The RDI interface DLL has to be copied to the appropriate debugger directory

2.6 Configuration

Before you can use the BDI together with the debugger, the BDI must be configured. Use the *SETUP* menu and follow the steps listed below:

- Load or update the firmware / logic, store IP address --> *Firmware*
- Set the communication parameters between Host and BDI --> *Communication*
- Setup an initialization list for the target processor --> *Initlist*
- Select the working mode --> *Mode*
- Transmit the configuration to the BDI --> *Mode Transmit*

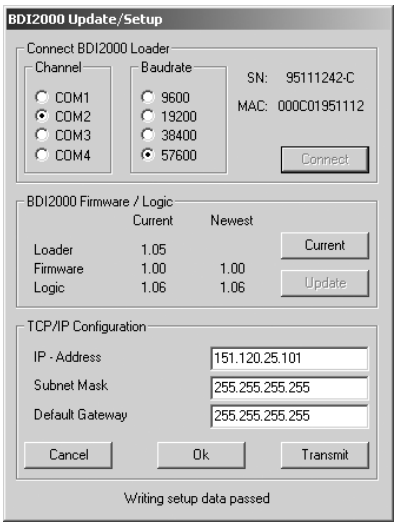
For information about the dialogs and menus use the help system (F1).

2.6.1 BDI2000 Setup/Update

First make sure that the BDI is properly connected (see Chapter 2.1 to 2.4). The BDI must be connected via RS232 to the Windows host.

To avoid data line conflicts, the BDI2000 must be disconnected from the target system while programming the logic for an other target CPU (see Chapter 2.1.1).

The following dialogbox is used to check or update the BDI firmware and logic and to set the network parameters.



dialog box «BDI2000 Update/Setup»

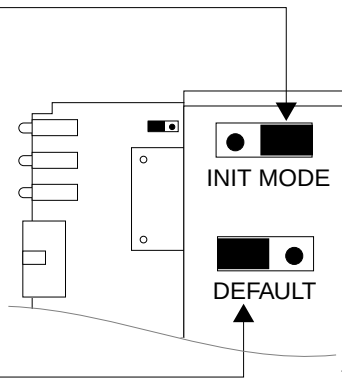
The following options allow you to check or update the BDI firmware and logic and to set the network parameters:

Channel	Select the communication port where the BDI2000 is connected during this setup session.
Baudrate	Select the baudrate used to communicate with the BDI2000 loader during this setup session.

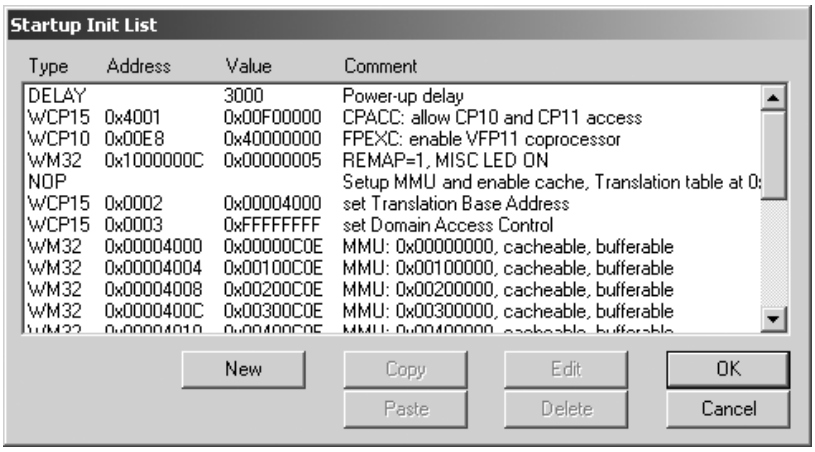
Connect	Click on this button to establish a connection with the BDI2000 loader. Once connected, the BDI2000 remains in loader mode until it is restarted or this dialog box is closed.
Current	Press this button to read back the current loaded BDI2000 software and logic versions. The current loader, firmware and logic version will be displayed.
Update	This button is only active if there is a newer firmware or logic version present in the execution directory of the BDI setup software. Press this button to write the new firmware and/or logic into the BDI2000 flash memory / programmable logic.
IP Address	Enter the IP address for the BDI2000. Use the following format: xxx.xxx.xxx.xxxe.g.151.120.25.101 Ask your network administrator for assigning an IP address to this BDI2000. Every BDI2000 in your network needs a different IP address.
Subnet Mask	Enter the subnet mask of the network where the BDI is connected to. Use the following format: xxx.xxx.xxx.xxxe.g.255.255.255.0 A subnet mask of 255.255.255.255 disables the gateway feature. Ask your network administrator for the correct subnet mask.
Default Gateway	Enter the IP address of the default gateway. Ask your network administrator for the correct gateway IP address. If the gateway feature is disabled, you may enter 255.255.255.255 or any other value..
Transmit	Click on this button to store the network configuration in the BDI2000 flash memory.

In rare instances you may not be able to load the firmware in spite of a correctly connected BDI (error of the previous firmware in the flash memory). **Before carrying out the following procedure, check the possibilities in Appendix «Troubleshooting»**. In case you do not have any success with the tips there, do the following:

- Switch OFF the power supply for the BDI and open the unit as described in Appendix «Maintenance»
- Place the jumper in the **«INIT MODE»** position
- Connect the power cable or target cable if the BDI is powered from target system
- Switch ON the power supply for the BDI again and wait until the LED «MODE» blinks fast
- Turn the power supply OFF again
- Return the jumper to the **«DEFAULT»** position
- Reassemble the unit as described in Appendix «Maintenance»



3 Init List



dialog box «Startup Init List»

In order to prepare the target for debugging, you can define an Initialization List. This list is stored in the Flash memory of the BDI2000 and worked through every time the target comes out of reset. Use it to get the target operational after a reset. The memory system is usually initialized through this list. After processing the init list, the RAM used to download the application must be accessible.

Use on-line help (F1) and the supplied configuration examples on the distribution disk to get more information about the init list.

CPx register number:
The register number is used to build the appropriate MCR or MRC instruction.

```
+-----+-+-------+-----+-+-------+
|opc_2|0|  CRm  |opc_1|0|  nbr  |
+-----+-+-------+-----+-+-------+
```

CP15 : ID register (CRn = 0, opcode_2 = 0)	0x0000
CP15 : Cache Type (CRn = 0, opcode_2 = 1)	0x2000
CP15 : Invalidate I cache line (CRn = 7, opcode_2 = 1, CRm = 5)	0x2507

Special BDI Configuration Registers:
In order to change some special configuration parameters of the BDI, the GPR entry in the init list is used. Normal ARM GPR's covers a range from 0 to 15. Other GPR's are used to set BDI internal registers:

8005 This entry in the init list allows to change the JTAG clock frequency. This is useful if you have to start with a slow JTAG clock out of reset but after some initialization (e.g. PLL setup) you can use a faster clock. As an example see AT91EB55 setup. The value you enter selects the following JTAG frequency:

```
0 = adaptive
1 =  16 MHz            6 = 200 kHz
2 =   8 MHz            7 = 100 kHz
3 =   4 MHz            8 =  50 kHz
4 =   1 MHz            9 =  20 kHz
5 = 500 kHz           10 =  10 kHz
```

8006 This entry in the init list allows to define a delay time (in ms) the BDI inserts between releasing the reset line and starting communicating with the target. This delay is necessary when a target needs some wake-up time after a reset.

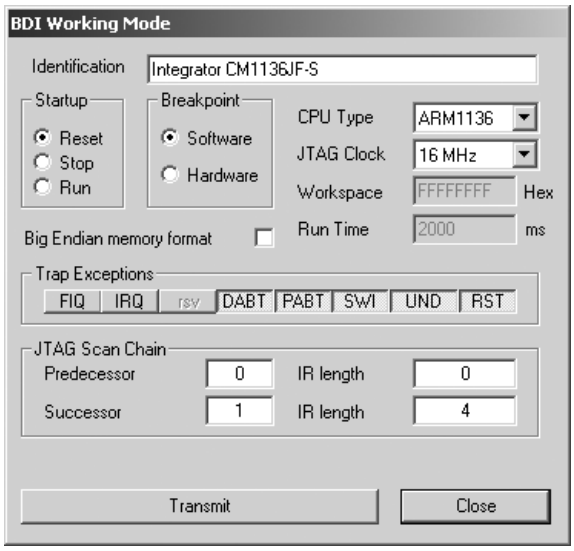
8007 By default, the BDI asserts the RESET signal during reset processing. After writing zero to this special register, the BDI no longer drives RESET low. This may be useful in some special cases.

8008 This entry in the init list allows to define a time (in ms) the BDI asserts the hardware reset signal. By default the reset signal is asserted for about 3 ms.

8009 By default, the TRST signal is driven with an open-drain driver by the BDI. Write a 1 to this special BDI register if the TRST signal should be driven with a push-pull driver.

8010 When using adaptive clocking, write a 1 to this special BDI register if the ARM core is clocked with a frequency slower that 6 MHz.

4 BDI working modes



dialog box «BDI Working Mode»

With this dialog box you can define how the BDI interacts with the target system.

Identification	Enter a text to identify this setup. This text can be read by the debugger with the appropriate Command.
Startup	Startup mode defines how the BDI interacts with the target processor after reset or power up. The options RESET, STOP or RUN can be selected.
Breakpoint	Breakpoint mode defines how breakpoints are implemented. When Software is selected (default), Breakpoints are set by replacing program code in target memory. When Hardware is selected, the built-in breakpoint logic of the target CPU is used to implement Breakpoints. In this mode, only up to 6 Breakpoints are available. This mode may be used when debugging code stored in a ROM.
CPU Type	Select the CPU type of the target system.
Workspace	not used yet
Run Time	When startup mode STOP is selected, this option allows to set the run time after reset in milliseconds until the target CPU is stopped. Values from 100 (0.1 sec) till 32000 (32 sec) are accepted.
Trap Exceptions	Select the exceptions that should lead to debug mode entry instead of entering the normal exception handler. The ARM vector catch register is used for this.
Big Endian...	Check this switch if the target memory uses Big Endian format.
JTAG Clock	This option allows to select the used JTAG clock rate (adaptive needs a special cable from Abatron, please ask for it).

JTAG Scan Chain	The BDI can also handle systems with multiple devices connected to the JTAG scan chain. In order to put the other devices into BYPASS mode and to count for the additional bypass registers, the BDI needs some information about the scan chain layout. Enter the number and total instruction register (IR) length of the devices present before the ARM chip (Predecessor). Enter the appropriate information also for the devices following the ARM chip (Successor).
Transmit	Click on this button to send the initialization list and the working mode to the BDI. This is normally the last step done before the BDI can be used with the debugging system.

4.1 Startup Mode

Startup mode defines how the BDI interacts with the target system after a reset or power up sequence.

4.1.1 Startup mode RESET

In this mode no ROM is required on the target system. The necessary initialization is done by the BDI with the programmed init list. The following steps are executed by the BDI after system reset or system power up:

- RESET is activated on the target system.
- RESET is deactivated and the target is forced into debug mode.
- The BDI works through the initialization list and writes to the corresponding addresses.

The RESET mode is the standard working mode. Other modes are used in special cases (i.e. applications in ROM, special requirements on the reset sequence...).

4.1.2 Startup Mode STOP

In this mode the initialization code is in a ROM on the target system. The code in this ROM handles base initialization. At the end of the code, the initialization program enters an endless loop until it is interrupted by the BDI. This mode is intended for special requirements on the reset sequence (e.g. loading a RAM based programmable logic device).
In this mode the following steps are executed by the BDI after system reset or power up:

- RESET is activated on the target system.
- RESET is deactivated and the target starts executing application code.
- After a delay of RUNTIME seconds, the target is forced into debug mode.
- The BDI works through the initialization list and writes the corresponding addresses.

4.1.3 Startup mode RUN

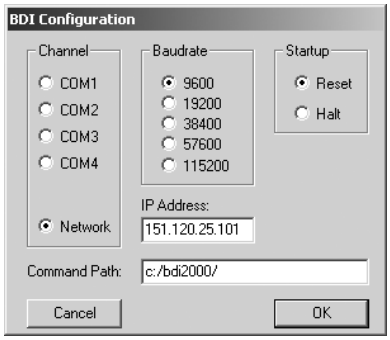
This mode is used to debug applications which are already stored in ROM. The application is started normally and is stopped when the debugger is started.
In this mode, the following steps are executed by the BDI after system reset or power up:

- RESET is activated on the target system.
- RESET is deactivated and the target starts executing application code.
- The application runs until it is stopped by the debugger.

5 Working with RDI Debuggers

5.1 Configuration

The debugger manual describes how to install a RDI target connection. When you configure the RDI connection, the following dialog box will be displayed. Enter the appropriate communication parameters and if necessary the path to the BDI command files.



The Startup mode defines how the BDI interacts with the target when the debugger starts. If Reset is selected (default), the BDI forces a hardware reset of the target when the debugger attaches. The Halt mode is useful when you would like to connect to a running target. The BDI simply stops the target when the debugger attaches. This way you can investigate the current status of the target.

You may use a name instead of a number as IP address but then the appropriate entry has to be present in the "hosts" file.

Do not use "\" for the command path, use "/" instead. Also some debuggers limit the size of a configuration string. Therefore use a short name for the command path.

5.2 BDI Direct Commands

For special functions (mainly for flash programming) the BDI supports so called «Direct Commands». This commands can be entered in a command file (e.g. PRELOAD.CMD). This Direct Commands are not interpreted by the debugger but directly sent to the BDI. After processing the command the result is displayed in the debugger's Command Line Window.

Direct Commands are ASCII - Strings with the following structure:

<Object>.<Action> [<ParName>=<ParValue>]...

Example:

flash.erase addr=0x02800000

All names are case insensitive. Parameter values are numbers or strings. Numeric parameters can be entered as decimal (e.g. 700) or as hexadecimal (0x80000) values.

5.2.1 Target.Reset

This direct command executes a real physical reset of the target system.

5.2.2 Flash.Setup

In order to support loading into flash memory, the BDI needs some information about the used flash devices. Before any other flash related command can be used, this direct command must be executed.

Syntax:

flash.setup type=am29f size=0x80000 bus=32 workspace=0x1000

type	This parameter defines the type of flash used. It is used to select the correct programming algorithm. The following flash types are supported: AM29F, AM29BX8, AM29BX16, I28BX8, I28BX16, AT49, AT49X8, AT49X16 STRATAX8, STRATAX16, MIRROR, MORRORX8, MIRRORX16, I28BX32, AM29DX16, AM29DX32
size	The size of **one** flash chip in bytes (e.g. AM29F010 = 0x20000). This value is used to calculate the starting address of the current flash memory bank.
bus	The width of the memory bus that leads to the flash chips. Do not enter the width of the flash chip itself. The parameter TYPE carries the information about the number of data lines connected to one flash chip. For example, enter 16 if you are using two AM29F010 to build a 16bit flash memory bank.
workspace	If a workspace is defined, the BDI uses a faster programming algorithm that run out of RAM on the target system. Otherwise, the algorithm is processed within the BDI. The workspace is used for a 1kByte data buffer and to store the algorithm code. There must be at least 2kBytes of RAM available for this purpose.

5.2.3 Flash.Erase

This command allows to erase one flash sector, block or chip.

Syntax:

flash.erase addr=0x02800000 mode=chip

addr	The start address of the flash sector to erase.
mode	This parameter defines the erase mode. Not all flash types support CHIP erase. The following modes are accepted: CHIP, BLOCK and SECTOR (default is sector erase)

5.2.4 Flash.Load

This command enables loading to flash memory. If the address of a data block is within the given flash range, the BDI automatically uses the appropriate programming algorithm. This command must be executed before downloading is started.

Syntax:

flash.load addr=0x02800000 size=0x200000

addr	The start address of the flash memory
size	The size of the flash memory

5.2.5 Flash.Idle

This command disables loading to flash memory.

Syntax:

flash.idle

5.3 Download to Flash Memory

The BDI supports download and debugging of code that runs out of flash memory. To automate the process of downloading to flash memory, the BDI looks for two command files in the working directory.

PRELOAD.CMD	This command file is executed just before download begins
POSTLOAD.CMD	This command file is executed after download is terminated.

Following is an example used to download into the flash memory of the PID7T board (socket U12 with a AMD flash Am29F010).

PRELOAD.CMD:
```
;Reset target
target.reset
;
;Define used flash memory: AM29F010
flash.setup type=am29f size=0x20000 bus=8 workspace=0x00001000
;
;Erase sector 0 to 7 of flash memory bank
flash.erase addr=0x04000000
flash.erase addr=0x04004000
flash.erase addr=0x04004000
flash.erase addr=0x04008000
flash.erase addr=0x0400C000
flash.erase addr=0x04010000
flash.erase addr=0x04014000
flash.erase addr=0x04018000
flash.erase addr=0x0401C000
;
;Enable loading into flash
flash.load addr=0x04000000 size=0x200000
```

POSTLOAD.CMD:
```
flash.idle
```

Note:
Some Intel flash chips (e.g. 28F800C3, 28F160C3, 28F320C3) power-up with all blocks in locked state. In order to erase/program those flash chips, use the init list to unlock the appropriate blocks.

```
WM16    0xFFF00000    0x0060        unlock block 0
WM16    0xFFF00000    0x00D0
WM16    0xFFF10000    0x0060        unlock block 1
WM16    0xFFF10000    0x00D0
        ....
WM16    0xFFF00000    0xFFFF        select read mode
```


Supported Flash Memories:
There are currently 3 standard flash algorithm supported. The AMD, Intel and Atmel AT49 algorithm. Almost all currently available flash memories can be programmed with one of this algorithm. The flash type selects the appropriate algorithm and gives additional information about the used flash.

For 8bit only flash: AM29F (MIRROR), I28BX8, AT49

For 8/16 bit flash in 8bit mode: AM29BX8 (MIRRORX8), I28BX8 (STRATAX8), AT49X8

For 8/16 bit flash in 16bit mode: AM29BX16 (MIRRORX16), I28BX16 (STRATAX16), AT49X16

For 16bit only flash: AM29BX16, I28BX16, AT49X16

For 16/32 bit flash in 16bit mode: AM29DX16

For 16/32 bit flash in 32bit mode: AM29DX32

For 32bit only flash: I28BX32

The AMD and AT49 algorithm are almost the same. The only difference is, that the AT49 algorithm does not check for the AMD status bit 5 (Exceeded Timing Limits).
Only the AMD and AT49 algorithm support chip erase. Block erase is only supported with the AT49 algorithm. If the algorithm does not support the selected mode, sector erase is performed. If the chip does not support the selected mode, erasing will fail. The erase command sequence is different only in the 6th write cycle. Depending on the selected mode, the following data is written in this cycle (see also flash data sheets): 0x10 for chip erase, 0x30 for sector erase, 0x50 for block erase.
To speed up programming of Intel Strata Flash and AMD MirrorBit Flash, an additional algorithm is implemented that makes use of the write buffer. This algorithm needs a workspace, otherwise the standard Intel/AMD algorithm is used.

The following table shows some examples:

Flash	x 8	x 16	x 32	Chipsize
Am29F010	AM29F	-	-	0x020000
Am29F800B	AM29BX8	AM29BX16	-	0x100000
Am29DL323C	AM29BX8	AM29BX16	-	0x400000
Am29PDL128G	-	AM29DX16	AM29DX32	0x01000000
Intel 28F032B3	I28BX8	-	-	0x400000
Intel 28F640J3A	STRATAX8	STRATAX16	-	0x800000
Intel 28F320C3	-	I28BX16	-	0x400000
AT49BV040	AT49	-	-	0x080000
AT49BV1614	AT49X8	AT49X16	-	0x200000
M58BW016BT	-	-	I28BX32	0x200000
SST39VF160	-	AT49X16	-	0x200000
Am29LV320M	MIRRORX8	MIRRORX16	-	0x400000

6 Telnet Interface

A Telnet server is integrated within the BDI that can be accessed when the BDI is connected via ethernet to the host. It may help to investigate problems and allows access to target resources that can not directly be accessed by the debugger.

The following commands are available:

```
"MD    [<address>] [<count>]  display target memory as word (32bit)",
"MDH   [<address>] [<count>]  display target memory as half word (16bit)",
"MDB   [<address>] [<count>]  display target memory as byte (8bit)",
"MM    <addr> <value> [<cnt>] modify word(s) (32bit) in target memory",
"MMH   <addr> <value> [<cnt>] modify half word(s) (16bit) in target memory",
"MMB   <addr> <value> [<cnt>] modify byte(s) (8bit) in target memory",
"MT    <addr> <count>         memory test",
"MC    [<address>] [<count>]  calculates a checksum over a memory range",
"MV                           verifies the last calculated checksum",
"RD                           display general purpose register",
"RDALL                        display all ARM registers ",
"RDCP  [<cp>] <number>        display CP register, default is CP15",
"RDFP                         display floating point register",
"RM    <nbr>  <value>         modify general purpose register",
"RMCP  [<cp>] <number><value> modify CP register, default is CP15",
"DTLB <from> [<to>]           display Data Micro TLB entries",
"ITLB <from> [<to>]           display Inst Micro TLB entries",
"LTLB <from> [<to>]           display Lockable Main TLB entries",
"ATLB <from> [<to>]           display Set-Associative Main TLB entries",
"DTAG  <from> [<to>]          display L1 Data Cache Tag(s) ",
"ITAG  <from> [<to>]          display L1 Inst Cache Tag(s) ",
"RESET                        reset the target system",
"GO    [<pc>]                 set PC and start target system",
"TI    [<pc>]                 single step an instruction",
"HALT                         force target to enter debug mode",
"BI  <addr>                   set instruction breakpoint",
"CI  [<id>]                   clear instruction breakpoint(s)",
"BD  [R|W] <addr>             set data watchpoint (32bit access)",
"BDH [R|W] <addr>             set data watchpoint (16bit access)",
"BDB [R|W] <addr>             set data watchpoint ( 8bit access)",
"CD [<id>]                    clear data watchpoint(s)",
"INFO                         display information about the current state",
"DCMD <direct command>        execute a BDI direct command (see manual)",
"HELP                         display command list",
"QUIT                         terminate the Telnet session"
```


7 Specifications

Operating Voltage Limiting	5 VDC ± 0.25 V
Power Supply Current	typ. 500 mA max. 1000 mA
RS232 Interface: Baud Rates	9’600,19’200, 38’400, 57’600,115’200
Data Bits	8
Parity Bits	none
Stop Bits	1
Network Interface	10 BASE-T
Serial Transfer Rate between BDI and Target	up to 16 Mbit/s
Supported target voltage	1.8 – 5.0 V (3.0 – 5.0 V with Rev. A/B)
Operating Temperature	+ 5 °C ... +60 °C
Storage Temperature	-20 °C ... +65 °C
Relative Humidity (noncondensing)	<90 %rF
Size	190 x 110 x 35 mm
Weight (without cables)	420 g
Host Cable length (RS232)	2.5 m

Specifications subject to change without notice

8 Environmental notice

Disposal of the equipment must be carried out at a designated disposal site.

9 Declaration of Conformity (CE)

CE

DECLARATION OF CONFORMITY

This declaration is valid for following product:

Type of device: BDM/JTAG Interface
Product name: BDI2000

The signing authorities state, that the above mentioned equipment meets the requirements for emission and immunity according to

EMC Directive 89/336/EEC

The evaluation procedure of conformity was assured according to the following standards:

EN 50081-2
EN 50082-2

This declaration of conformity is based on the test report no. QNL-E853-05-8-a of QUINEL, Zug, accredited according to EN 45001.

Manufacturer:

ABATRON AG
Stöckenstrasse 4
CH-6221 Rickenbach

Authority:

Max Vock
Marketing Director

Ruedi Dummermuth
Technical Director

Rickenbach, May 30, 1998

10 Warranty

ABATRON Switzerland warrants the physical diskette, cable, BDI2000 and physical documentation to be free of defects in materials and workmanship for a period of 24 months following the date of purchase when used under normal conditions.

In the event of notification within the warranty period of defects in material or workmanship, ABATRON will replace defective diskette, cable, BDI2000 or documentation. The remedy for breach of this warranty shall be limited to replacement and shall not encompass any other damages, including but not limited loss of profit, special, incidental, consequential, or other similar claims.
ABATRON Switzerland specifically disclaims all other warranties- expressed or implied, including but not limited to implied warranties of merchantability and fitness for particular purposes - with respect to defects in the diskette, cable, BDI2000 and documentation, and the program license granted herein, including without limitation the operation of the program with respect to any particular application, use, or purposes. In no event shall ABATRON be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential, or other damages.

Failure in handling which leads to defects are not covered under this warranty. The warranty is void under any self-made repair operation except exchanging the fuse.

Appendices

A Troubleshooting

Problem
The firmware can not be loaded.

Possible reasons

- The BDI is not correctly connected with the target system (see chapter 2).
- The power supply of the target system is switched off or not in operating range (4.75 VDC ... 5.25 VDC) --> MODE LED is OFF or RED
- The built in fuse is damaged --> MODE LED is OFF
- The BDI is not correctly connected with the Host (see chapter 2).
- A wrong communication port (Com 1...Com 4) is selected.

Problem
No working with the target system (loading firmware is ok).

Possible reasons

- Wrong pin assignment (BDM/JTAG connector) of the target system (see chapter 2).
- Target system initialization is not correctly --> enter an appropriate target initialization list.
- An incorrect IP address was entered (BDI2000 configuration)
- BDM/JTAG signals from the target system are not correctly (short-circuit, break, ...).
- The target system is damaged.

Problem
Network processes do not function (loading the firmware was successful)

Possible reasons

- The BDI2000 is not connected or not correctly connected to the network (LAN cable or media converter)
- An incorrect IP address was entered (BDI2000 configuration)

B Maintenance

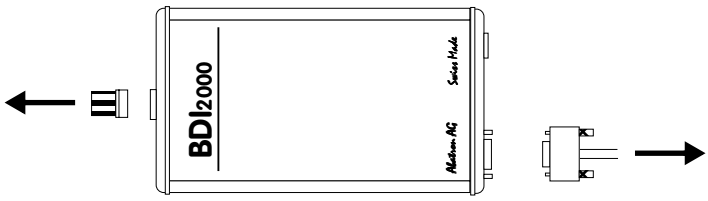
The BDI needs no special maintenance. Clean the housing with a mild detergent only. Solvents such as gasoline may damage it.

If the BDI is connected correctly and it is still not responding, then the built in fuse might be damaged (in cases where the device was used with wrong supply voltage or wrong polarity). To exchange the fuse or to perform special initialization, please proceed according to the following steps:

Observe precautions for handling (Electrostatic sensitive device)
Unplug the cables before opening the cover.
Use exact fuse replacement (Microfuse MSF 1.6 AF).

1

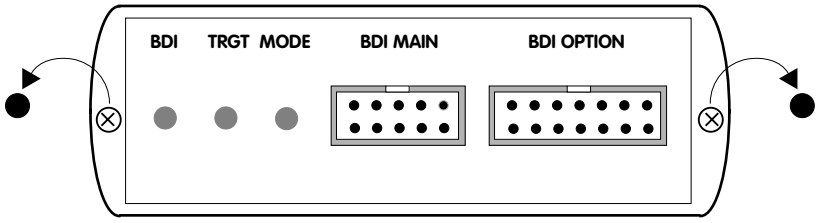
1.1 Unplug the cables



2

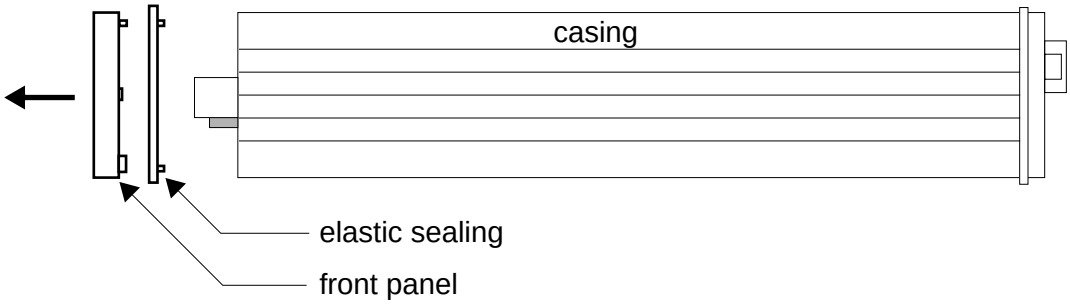
2.1 Remove the two plastic caps that cover the screws on target front side (e.g. with a small knife)

2.2 Remove the two screws that hold the front panel



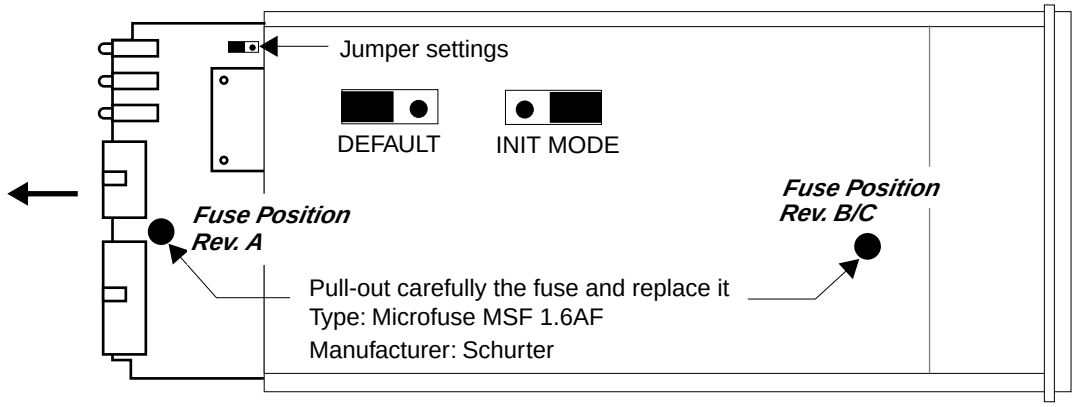
3

3.1 While holding the casing, remove the front panel and the red elastig sealing



4

4.1 While holding the casing, slide carefully the print in position as shown in figure below

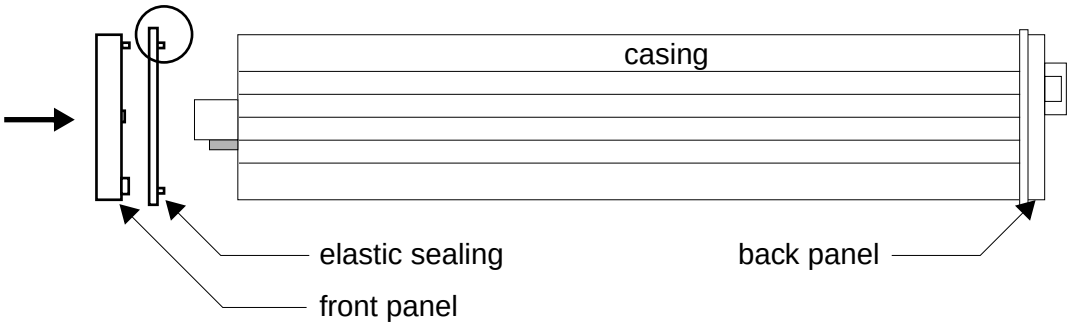


5

Reinstallation

5.1 Slide back carefully the print. Check that the LEDs align with the holes in the back panel.

5.2 Push carefully the front panel and the red elastig sealing on the casing. Check that the LEDs align with the holes in the front panel and that the position of the sealing is as shown in the figure below.



5.3 Mount the screws (do not overtighten it)

5.4 Mount the two plastic caps that cover the screws

5.5 Plug the cables

Observe precautions for handling (Electrostatic sensitive device)
Unplug the cables before opening the cover.
Use exact fuse replacement (Microfuse MSF 1.6 AF).

C Trademarks

All trademarks are property of their respective holders.

