bdiCW

BDM interface for CodeWarrior™ Debugger

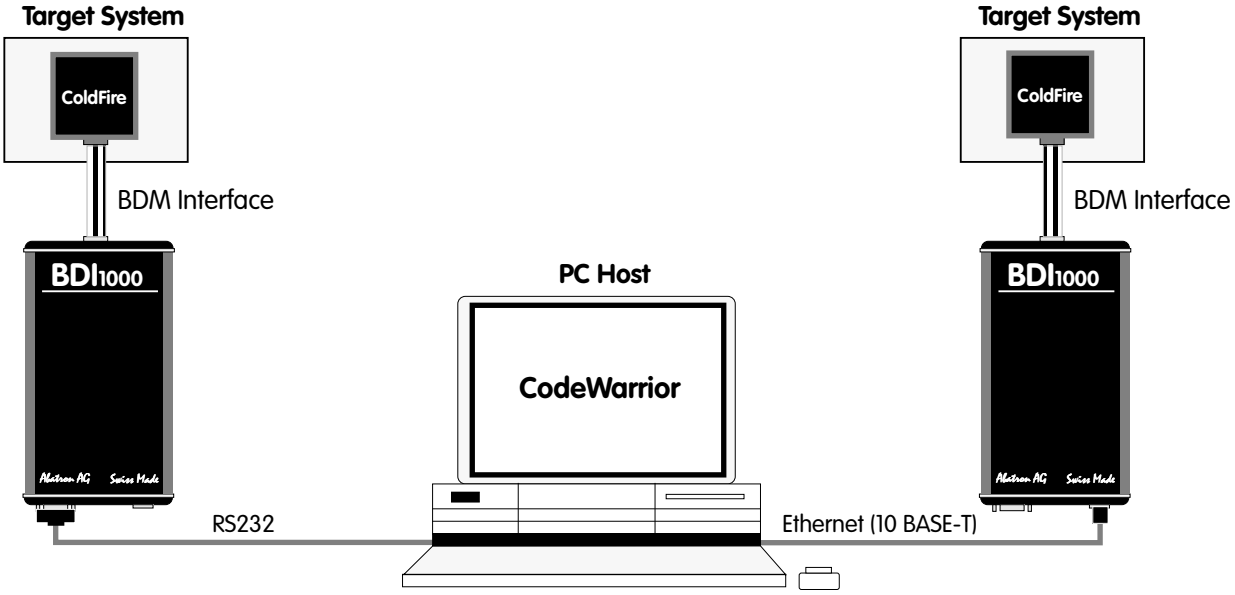
ColdFire

User Manual

Manual Version 1.04 for BDI1000

Appendices

1 Introduction



The BDI1000 adds Background Debug Mode features to the CodeWarrior debugger environment from Metrowerks. With the BDI1000, 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 BDI1000 so it works with the XRAY debugger.

1.1 BDI1000

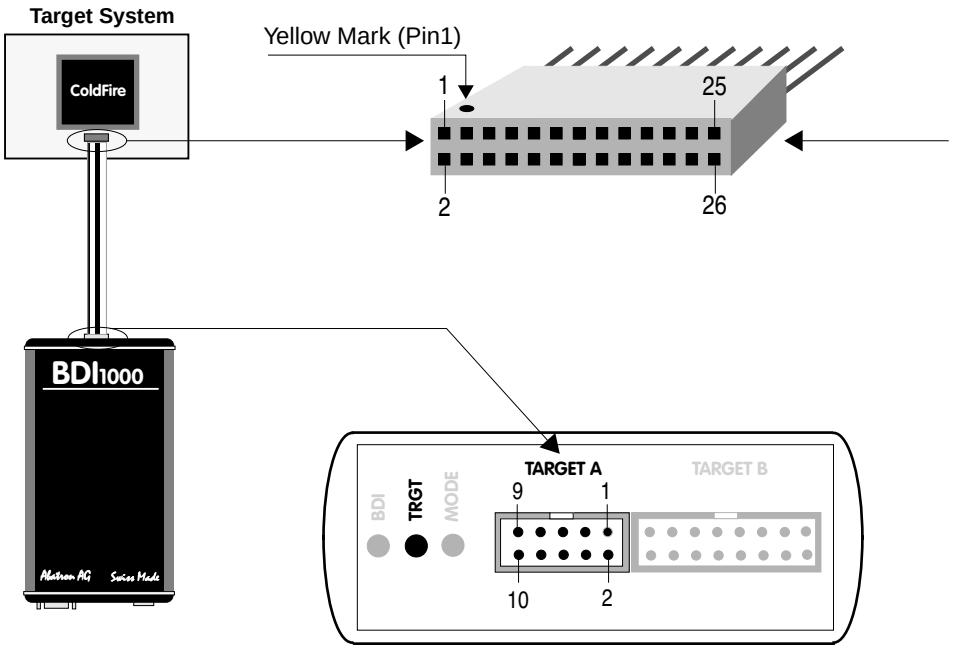
The BDI1000 is a processor system in a small box. It implements the interface between the BDM pins of the target CPU and a 10Base-T Ethernet / RS232 connector. BDI1000 is powered by a MC68331, 256Kbyte RAM and a flash memory of 512Kbyte. As a result of consistent implementation of lasted technology, the BDI1000 is optimally prepared for further enhancements. The firmware and the programmable logic of the BDI1000 can be updated by the user with a simple Windows based configuration program. The BDI1000 supports target system voltages from 2.7 up to 5 Volts.

2 Installation

2.1 Connecting the BDI1000 to Target

The enclosed cable to the target system is designed for the Motorola recommended 26-pin Berg connector. In case where the target system has an appropriate connector, the cable can be directly connected. The pin assignment is in accordance with the Motorola specification.

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

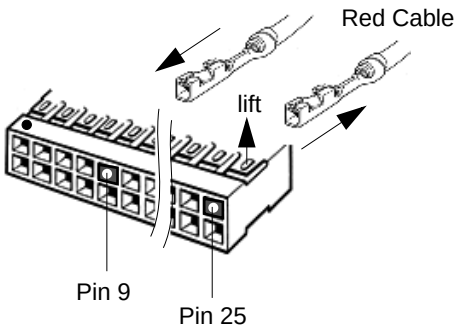
Target Connector

1 - NOT USED
2 - BKPT
3 - GROUND
4 - DSCLK
5 - GROUND
6 - NOT USED
7 - RESET
8 - DSI
9 - Vccio **
10 - DSO
11 - NOT USED
12 - NOT USED
13 - NOT USED
14 - NOT USED
15 - NOT USED
16 - NOT USED
17 - NOT USED
18 - NOT USED
19 - NOT USED
20 - NOT USED
21 - NOT USED
22 - NOT USED
23 - NOT USED
24 - CLK_CPU*
25 - Vcore **
26 - TEA

* only needed for V2 cores
** see note

The target CPU clock is only needed for older V2 cores (MCF5204, MCF5206(e) and MCF5272) because DSCLK and DSI have to change synchronous with the CPU clock. For all other cores, BDM communication can be asynchronous to the CPU clock and therefore this signal is not needed.

Important note for older target cables:
The target cables delivered before October 2004 expect the target BDM reference voltage (Vccio) at pin 25 of the BDM connector. For target boards where Vccio is only routed to pin 9, remove the contact/wire from housing pin 25 and insert it into pin 9.



BDI MAIN / TARGET A Connector Signals:

Pin	Name	Describtion
1	CLK_CPU	**PROCESSOR CLOCK** For some cores, this clock signal is used to synchonize the BDM signals DSCLK and DSI.
2	$\overline{\text{BKPT}}$	**BREAKPOINT** BKPT is an active-low signal that signals a hardware breakpoint for the ColdFire core. It is used to force the ColdFire core to enter debug mode.
3+5	GROUND	**SYSTEM GROUND**
4	DSCLK	**DEVELOPMENT SERIAL CLOCK** For background debug mode, serial shift clock to the MCU.
6	$\overline{\text{TEA}}$ (optional)	**TRANSFER ERROR ACKNOWLEDGE (currently not implemented)** Active-low open-drain signal, used to abort a bus cycle. This signal may be helpful for ColdFire devices which has no built-in bus monitor (e.g. MCF5307). The BDI is able to terminate an invalid memory access. Otherwise BDM communication may hang until a reset is applied.
7	$\overline{\text{RESET}}$	**RESET** Active-low open-drain signal, used to force a system reset.
8	DSI	**DATA SERIAL IN** For background debug mode, serial data input signal to the MCU.
9	Vcc Target	**TARGET POWER** This input to the BDI1000 is used to detect if the target is powered up.
10	DSO	**DATA SERIAL OUT** For background debug mode, serial data output from the MCU.

Pin 1 is only use for some older V2 cores (MCF5204, MCF5206(e) and MCF5272). For all other cores this signal is not used.

Custom designs with ColdFire cores:
In order to use Abatron BDM interfaces, it is not neccessary to use the space consuming 26-pin Berg connector. If a 10-pin connector is used with the above signals, a simple 10-wire flat cable can be used. Also the CPU pins normally used for PST0..PST3 and DDATA0...DDATA3 can be assigned the alternate function (e.g. parallel port pin in a MCF5206 design).
For more information about the PSTx and DDATAx signals, see the appropriate ColdFire user's manual.

2.1.1 Changing Target Processor Type

Before you can use the BDI1000 with an other target processor type (e.g. ColdFire <--> PPC), a new setup has to be done (see chapter 2.6 «Configuration»). During this process the target cable must be disconnected from the target system. The BDI1000 needs to be supplied **between 2.5V and 5V** via the POWER connector. For more information see chapter 2.2.1 «External Power Supply».

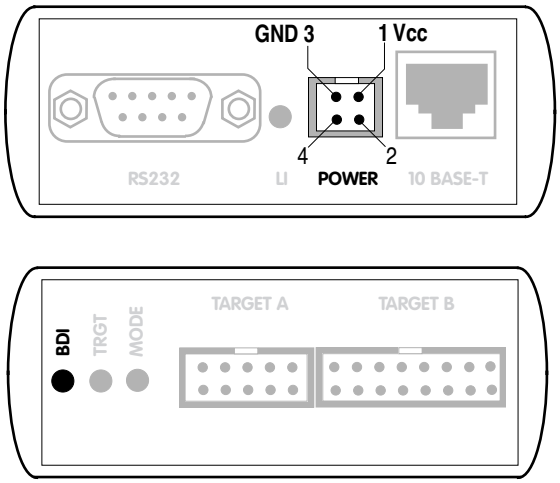
To avoid data line conflicts, the BDI1000 must be disconnected from the target system while programming the logic for an other target CPU.

2.2 Connecting the BDI1000 to Power Supply

2.2.1 External Power Supply

The BDI1000 needs to be supplied **between 2.5V and 5V** via the POWER connector. The available power supply from Abatron (option) or the enclosed power cable can be directly connected. In order to ensure reliable operation of the BDI1000, keep the power supply cable as short as possible.

For error-free operation, the power supply to the BDI1000 must be between 2.5V and 5V DC. **The maximal tolerable supply voltage is 5.25 VDC. Any higher voltage or a wrong polarity might destroy the electronics.**



POWER Connector

1 - Vcc (+2.5 ... +5V)
2 - VccTGT
3 - GROUND
4 - NOT USED

The green LED «BDI» marked light up when power (2.5 – 5V) is connected to the BDI1000

Please switch on the system in the following sequence:

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

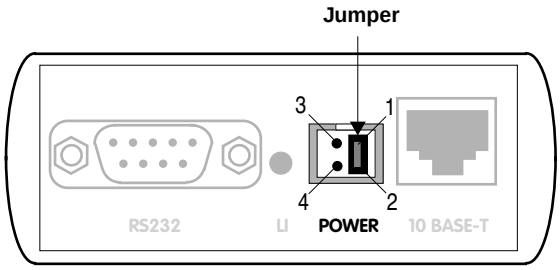
2.2.2 Power Supply from Target System

The BDI1000 needs to be supplied between 2.5V and 5V via TARGET A connector. This mode can only be used when the target system runs **between 2.5V and 5V** and the pin «Vcc Target» is able to deliver a current up to:

- 900mA@2.5Vcc Target
- 700mA@3.3Vcc Target
- 450mA@5.0Vcc Target

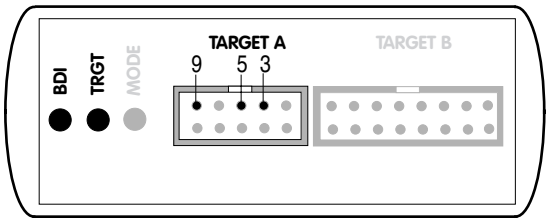
For pin description and layout see chapter 2.1 «Connecting the BDI1000 to Target». Insert the enclosed Jumper as shown in figure below. **Please ensure that the jumper is inserted correctly**.

For error-free operation, the power supply to the BDI1000 must be between 2.5V and 5V DC. **The maximal tolerable supply voltage is 5.25 VDC. Any higher voltage or a wrong polarity might destroy the electronics.**



POWER Connetcor

1 - Vcc BDI1000 (+2.5 ... +5V)
2 - Vcc Target (+2.5 ... +5V)
3 - GROUND
4 - NOT USED



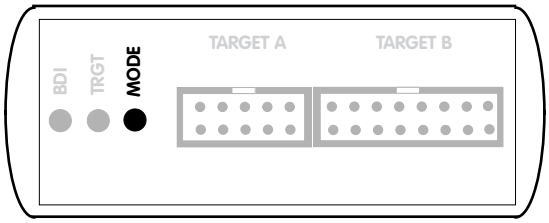
The green LEDs «BDI» and «TRGT» marked light up when target is powered up and the jumper is inserted correctly

Target A Connector

1 - CLK_CPU
2 - BKPT
3 - GROUND
4 - DSCLK
5 - GROUND
6 - TEA
7 - RESET
8 - DSI
9 - Vcc Target
10 - DSO

2.3 Status LED «MODE»

The built in LED indicates the following BDI states:

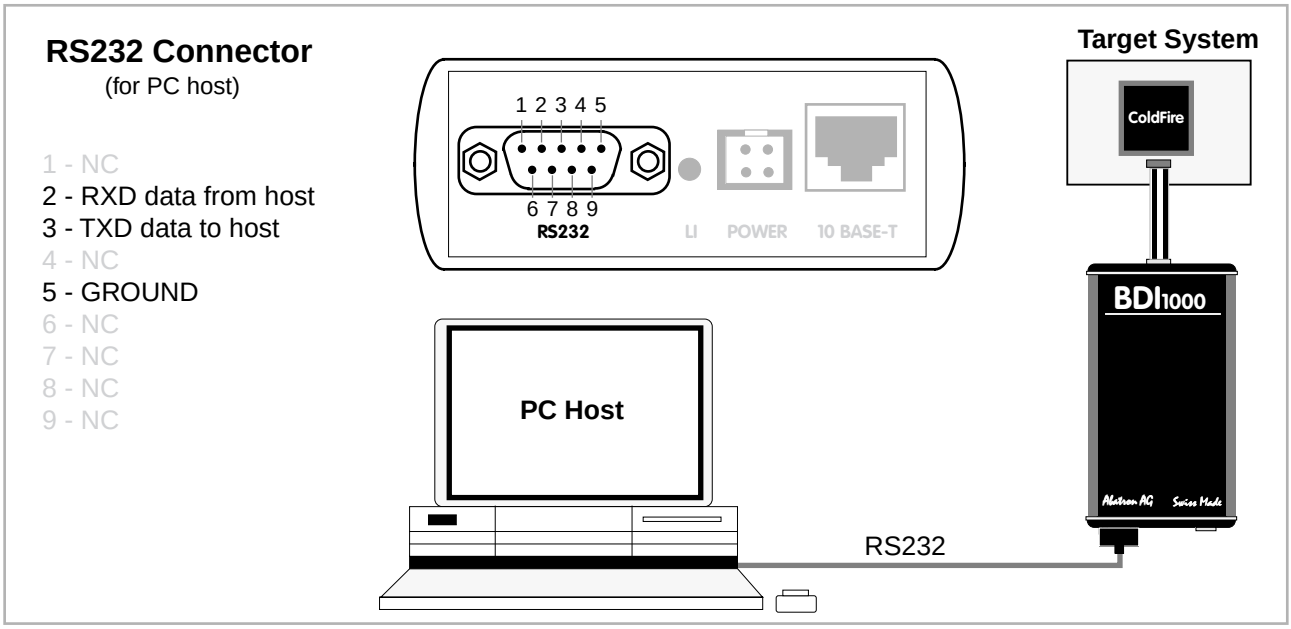


MODE LED	BDI STATES
OFF	The BDI is ready for use, the firmware is already loaded.
ON	The power supply for the BDI1000 is < 2.5VDC.
BLINK	The BDI «loader mode» is active (an invalid firmware is loaded or loading firmware is active).

2.4 Connecting the BDI1000 to Host

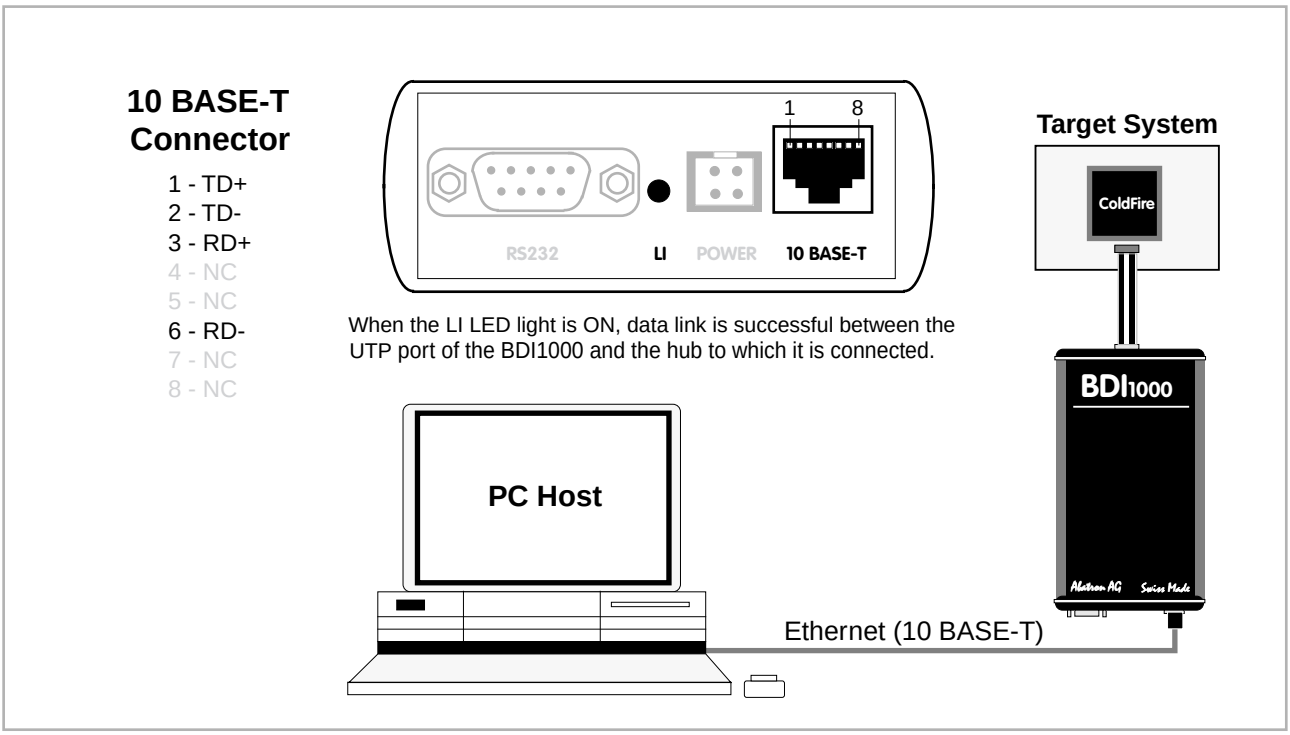
2.4.1 Serial line communication

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).



2.4.2 Ethernet communication

The BDI1000 has a built-in 10 BASE-T Ethernet interface (see figure below). Connect an UTP (Unshilded Twisted Pair) cable to the BD1000. For thin Ethernet coaxial networks you can connect a commercially available media converter (BNC --> 10 BASE-T) between your network and the BDI1000. Contact your network administrator if you have questions about the network.



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:

b10mcf.exe	Configuration program
b10mcf.hlp	Helpfile for the configuration program
b10mcf.cnt	Help contents file
b10mcffw.xxx	Firmware for BDI1000 for ColdFire targets
cf2jed10.xxx	JEDEC file for the BDI logic device (for old V2 cores)
cf3jed10.xxx	JEDEC file for the BDI logic device
bdiifc32.dll	BDI 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 b10mcf.exe configuration program.

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*

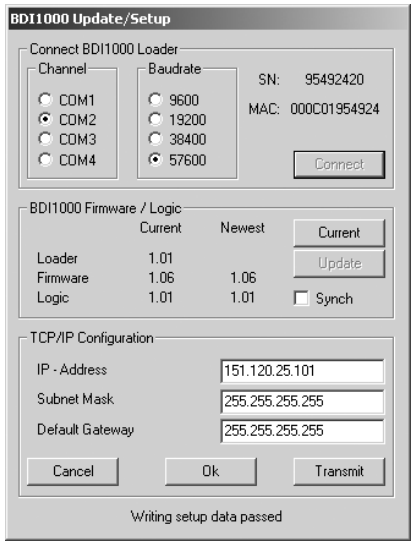
For information about the dialogs and menus use the help system (F1).

2.6.1 BDI1000 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 BDI1000 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 «BDI1000 Update/Setup»

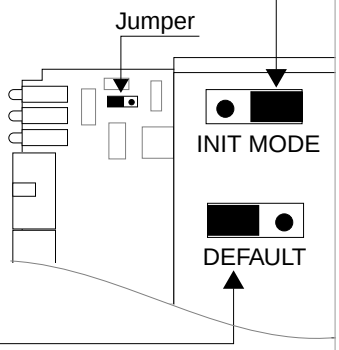
The following options allow you to check or update the BDI firmware and logic and to set the network parameters:

Channel	Select the communication port where the BDI1000 is connected during this setup session.
Baudrate	Select the baudrate used to communicate with the BDI1000 loader during this setup session.
Connect	Click on this button to establish a connection with the BDI1000 loader. Once connected, the BDI1000 remains in loader mode until it is restarted or this dialog box is closed.
Current	Press this button to read back the current loaded BDI1000 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 BDI1000 flash memory / programmable logic.

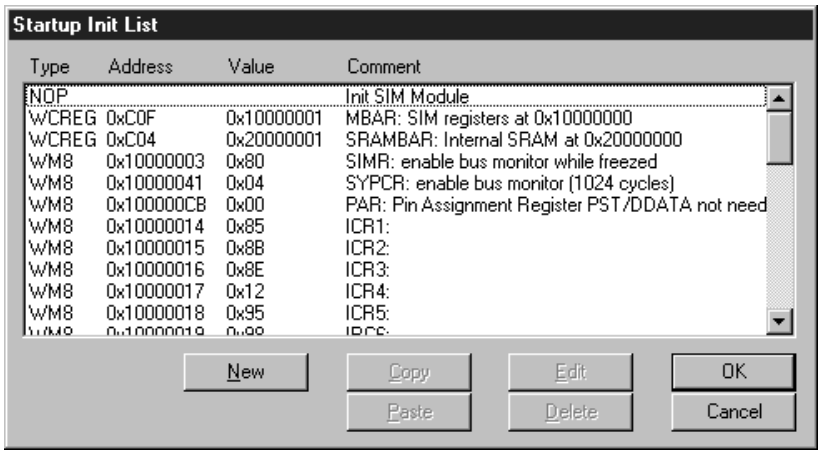
Synch	Because some older V2 ColdFire cores (e.g. MCF5272) need synchronous signals at the BDM interface, the BDI uses a different logic for this cores. Check this box only if your target is a MCF5204, MCF5206, MCF5206e or MCF5272. The BDI automatically selects the correct JEDEC file. Make sure that there is the correct logic loaded for the target CPU you are using.
IP Address	Enter the IP address for the BDI1000. Use the following format: xxx.xxx.xxx.xxx e.g.151.120.25.101 Ask your network administrator for assigning an IP address to this BDI1000. Every BDI1000 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 BDI1000 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 BDI1000 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.

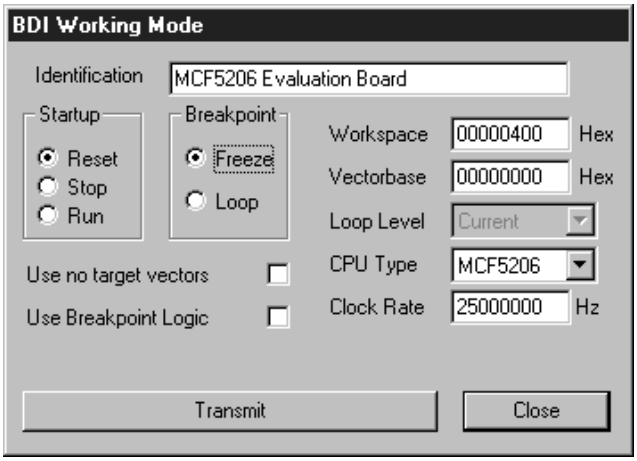
Special BDI Configuration Registers:

In order to change some special configuration parameters of the BDI, the WCREG entry in the init list is used. Control register numbers greater than 0x8000 are used to set BDI internal registers:

8001 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.

8002 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 1ms.

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 processed. The target processor may be frozen (FREEZED option) or may be set to loop in an exception procedure (LOOP option) while the application software is halted.
Workspace	In all configurations except when «Use no target vectors» is activated, BDI needs some target memory space. Enter here the start address of this memory area. A maximum of 512 bytes is needed.
Vector base	The BDI needs to know where the vector table is located. Enter here the start address of the vector table. This address is automatically loaded into the VBR register at startup time. The application should not change the VBR unless «Use no target vectors» is selected.
Loop Level	Selects the priority level (interrupt priority mask) the application uses when halted in LOOP mode. A level of 7 disables all interrupts when the application is halted. The value CURRENT(default) means, the application loops with the level currently active at the point where it is stopped.
CPU Type	Select the appropriate CPU type.
CPU Clock Rate ...	Enter the clock rate which the target CPU runs, after BDI has worked through the init list. BDI selects the BDM communication speed based on this parameter. If this parameter selects a CPU clock rate that is higher than the real clock, BDM communication may fail. When selecting a clock rate slower than possible, BDM communication still works but not as fast as possible.

Use no target vectors	Check this switch if the BDI should not use any RAM or vectors in the target system. This option is only enabled when FREEZED is selected as breakpoint mode. This mode is suitable for testing new hardware or debugging custom exception routines.
Use Breakpoint Logic	This switch defines how instruction breakpoints are implemented. When not checked (default), instruction breakpoints are set as re-quested by the debugger (Software or Hardware breakpoints). When checked, the BDI uses always hardware breakpoints. This is useful when the attached debugger does not support hardware break-points.

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 and BKPT are activated on the target system.
- RESET is deactivated and the target system changes to background debug mode.
- The BDI works through the initialization list and writes to the corresponding addresses.
- Depending on the break mode, the necessary vectors are set and help code is written into the RAM on the target system.

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 and sets the stackpointer. 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 or, if, for example, separate hardware needs to be initialized immediately.

In this mode the following steps are executed by the BDI after system reset or power up:

- RESET and BKPT are activated on the target system.
- RESET is deactivated and the target system changes to background debug mode.
- The target CPU is started (the target starts at the address fetched when reading the start vector at address 0).
- The target system is working through the application code.
- After 2 seconds, BKPT is activated and the target system changes to background debug mode.
- The BDI works through the initialization list and writes the corresponding addresses.
- Depending on the break mode, the necessary vectors are set and support code is written into the RAM on the target system.

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 by the BDI as soon as the debugger connects to the BDI.
In this mode, the following steps are executed by the BDI after system reset or power up:

- RESET and BKPT are activated on the target system.
- RESET is deactivated and the target system changes to background debug mode.
- The target CPU is started (the target starts at the address fetched when reading the start vector at address 0).
- The target system is executing the application code.
- The application runs until the debugger stops the execution.
- BKPT is activated on the target system, and the target system changes to background debug mode.
- Depending on the break mode, the necessary vectors are set and help code is written into the RAM on the target system.

4.2 Breakpoint Mode

The use of software breakpoints is only possible if the application code is stored in RAM (not in ROM) on the target system. Depending on the selected breakpoint mode, breakpoint and single step functions are implemented total differently.

4.2.1 Breakpoint Mode FREEZED

In this mode breakpoints are implemented by replacing application code with the HALT instruction. All the time the application is halted (i.e. caused by a breakpoint) the target processor remains freezed.
Single step is implemented by setting the appropriate bits (SSM, IPI) in the Debug Configuration/Status register and starting the CPU. Interrupts are ignored and debug mode is reentered after executing exact one instruction.

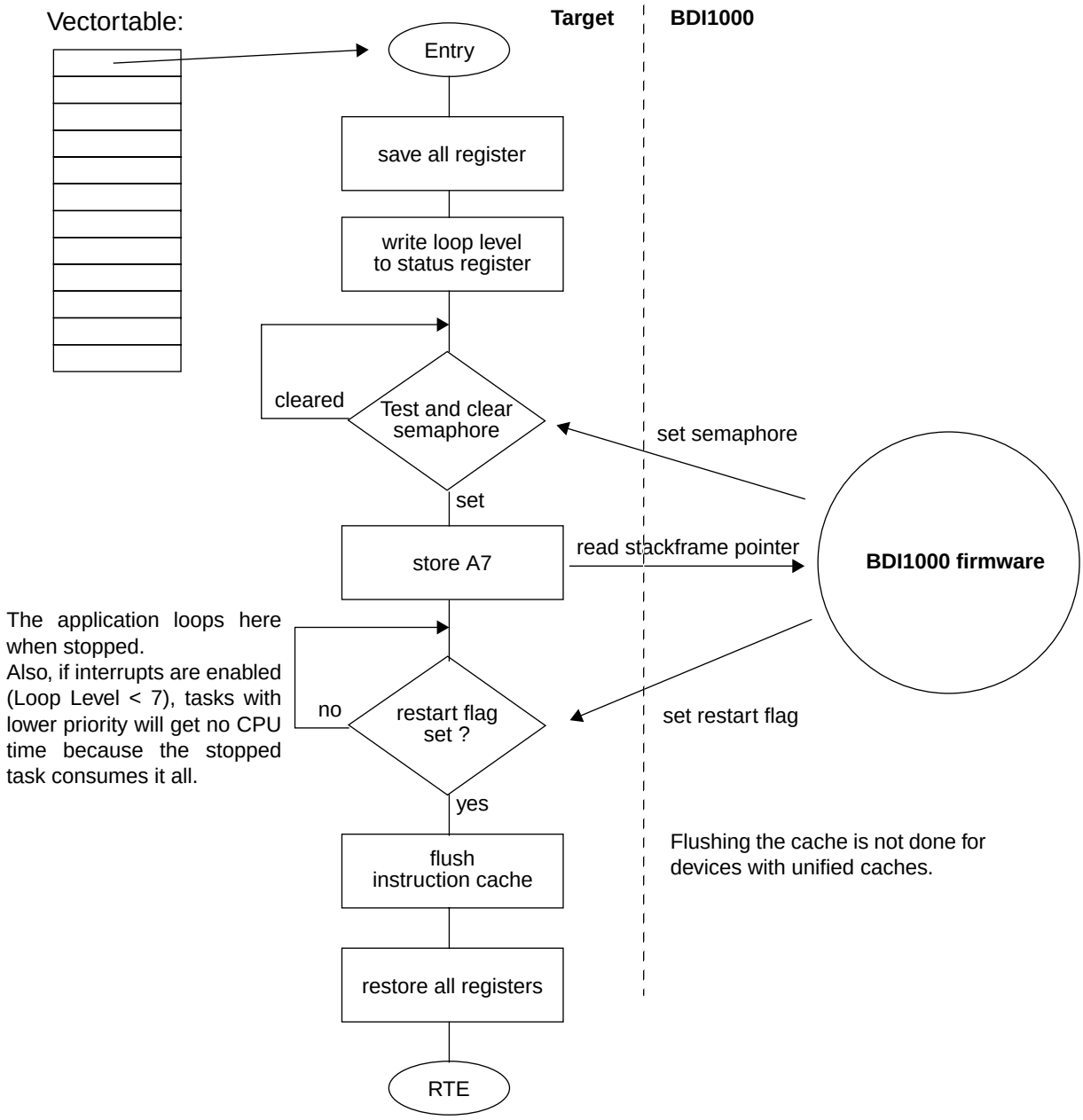
4.2.2 Breakpoint Mode LOOP

In this mode breakpoints are implemented by replacing application code with a ILLEGAL instruction. A stopped application loops within an exception procedure. The target processor is never freezed. The priority level used when looping in the exception procedure can be defined using the option «Loop Level». If you want to stop the hole application use Loop Level 7. If only the current task should be stopped, use Loop Level CURRENT.

Single step is implemented by setting the Trace bit in the processor status register. So a single step steps always over the current instruction. If interrupts are pending, they are served first without stopping the target processor.

The Loop mode is suitable when debugging real-time applications which can not be freezed, because external interrupt must be handled even when an application task is stopped at a breakpoint.

The following diagram shows the used universal exception procedure for ColdFire targets. May be this helps you to understand how BDI works in LOOP mode.



The ColdFire is never freezed because it is possible to access target memory via the BDM interface while the CPU is running.

4.3 Workspace

Depending on the working mode, the BDI needs some RAM in the target system. The following table shows how much RAM is used in the different modes.

Mode	Workspace (Bytes)	Remark
RESET/FREEZED	4	needed to trap the exceptions
STOP/FREEZED	4	needed to trap the exceptions
RUN/FREEZED	4	needed to trap the exceptions
RESET/LOOP	512	used for standard exception procedure and the initial (supervisor) stack
STOP/LOOP	256	used for standard exception procedure
RUN/LOOP	256	used for standard exception procedure

Vectors in RUN mode:

When RUN mode is selected, only the vectors 0... 24 are initialized when the application is halted for the first time .

5 Specifications

Operating Voltage Limiting	2.5 ... 5.25VDC
Power Supply Current (max)	900mA@2.5V 700mA@3.3V 450mA@5.0V
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	5.5 Mbit/s (BDM) 12 Mbit/s (JTAG)
Supported target voltage	2.7 ... 5 VDC
Operating Temperature	+ 5 °C ... +60 °C
Storage Temperature	-20 °C ... +65 °C
Relative Humidity (noncondensing)	<90 %rF
Size	160 x 85 x 35 mm
Weight (without cables)	280 g
Electromagnetic Compatibility (EMC)	EN 50081-2, EN 50082-2

Specifications subject to change without notice

6 Environmental notice

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

7 Declaration of Conformity (CE)

CE

Declaration of Conformity

This declaration is valid for the following product:

Type of device: **BDM/JTAG Interface**
Product name: **BDI1000**

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:

EN50081-2
EN50082-2

This declaration of conformity is based on the test report no. QNL-E079-05-9-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, November 2, 1999

8 Warranty

ABATRON Switzerland warrants the physical diskette, cable, BDI1000 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, BDI1000 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, BDI1000 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 (2.5 VDC ... 5 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 (BDI1000 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 BDI1000 is not connected or not correctly connected to the network (LAN cable or media converter)
- An incorrect IP address was entered (BDI1000 configuration)

B Maintenance

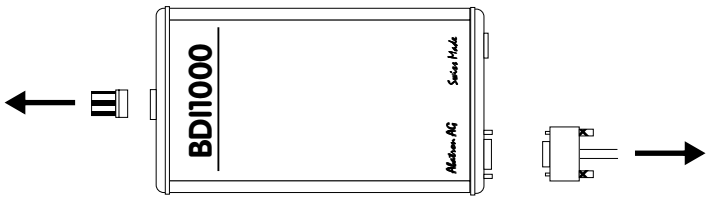
The BDI needs no special maintenance. Clean the housing with a mild detergent only. Solvents such as gasoline may damage it.

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

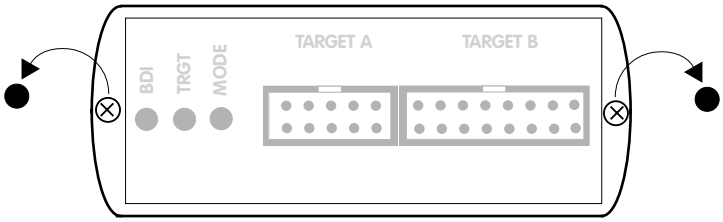
1.1 Unplug the cables



2

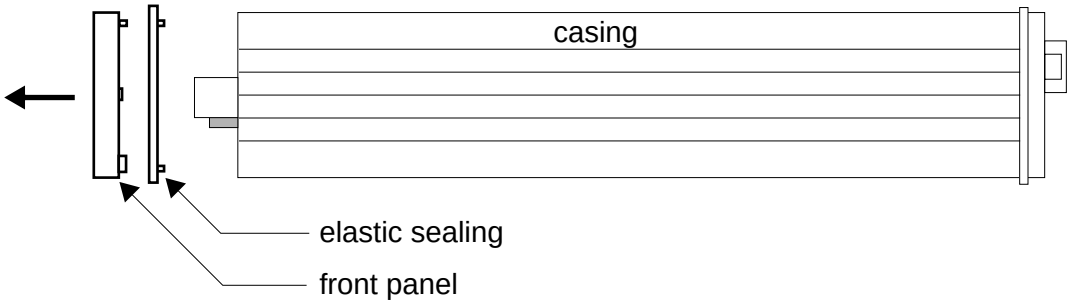
2.1 Remove the two plastic caps that cover the screws on target front side (e.g. with a small knife)

2.2 Remove the two screws that hold the front panel



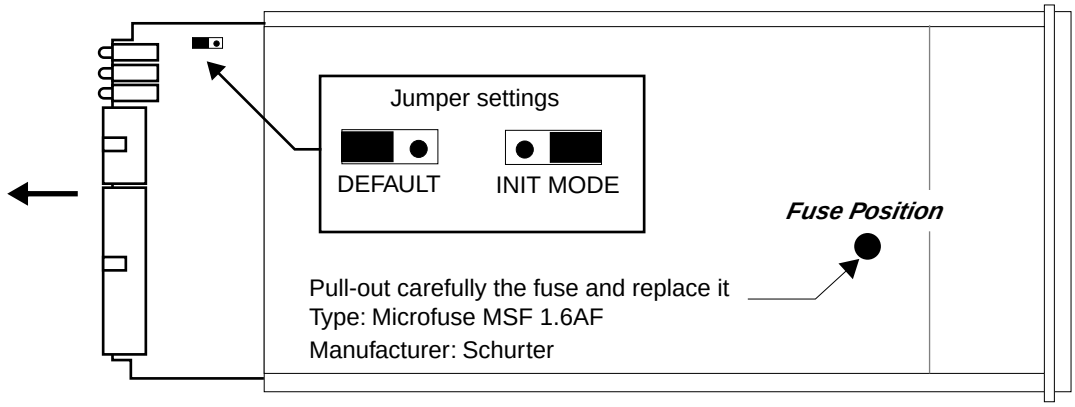
3

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



4

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

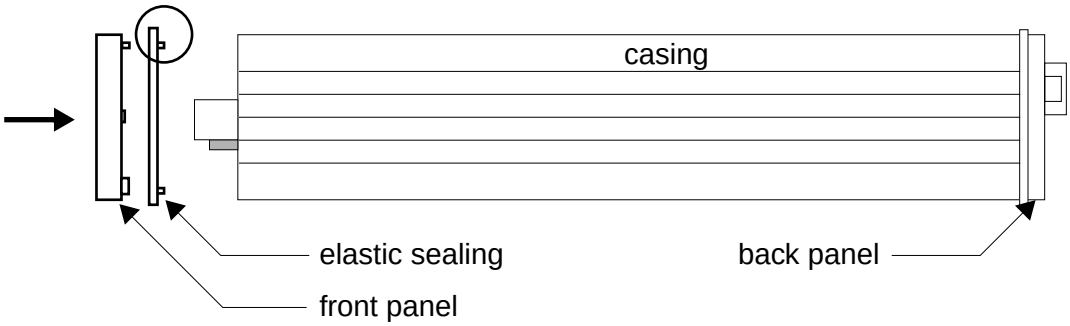


5

Reinstallation

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

5.2 Push carefully the front panel and the blue 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.

