TELNET PROTOCOL DOCUMENTATION FOR

The BarnMini-22

Revision 1.0

Barnfind Technologies AS
https://www.barnfind.no

Contents

1. Introduction

This document provides technical details on the Telnet protocol implementation for **BarnMini-22**. It describes the command structure, network configurations and a brief summery for all other sections that can be present.

1.1 Purpose

The purpose of this document is to assist developers, system integrators, and IT administrators in configuring and using the Telnet interface for device management and control.

2. Command structure

Every command is terminated with a new line (0x0a). From the client there are a total of 2 types of commands: select section and to write to values/properties.

To select a target section (object). The name of the ***section*** is put inside square brakes.

Example:

[port.1]

- When a valid section has been selected, values can be sent.
- Numeric values are given as is, string values are put inside quotations.
- In newer firmware versions, a value can be deleted if it set to an empty string without any quotations.

Example:

someint=1

somestring="foo"

someoldvar=

Strings can be escaped using the follow codes:

\n - for a newline (0x0a)

\r - for a return carrier (0x0d)

\x00 - for a hexadecimal value where 00 is the designated value

\\ - for a backslash

\t - for a tab (0x09 is also valid, server will always return \x09 notation)

We recommend that all values outside the ASCII range 32-126, \ ' and " to be escaped.

From the server, there are two additional commands. ACK and NAK. These are sent as reply to all commands from the client to verify that the command is valid or not.

The server and client streams are not synchronized, with exception of ACK/NAK being appended to the stream from the server to the client as it receives commands. When the client receives a matching ACK/NAK, the command has been processed, and parameters that are directly influenced by the command should already have been transmitted at this point.

3. Sections

3.1 Section “[barnmini22]”

This section contains identification, model and version information:

[barnmini22]	
firmwareversion="1.1.6"	Running firmware version.
model="BarnMini-22"	Identifies the device model.
serialnumber="86620089"	Device serial number.
PCBVersion=3	PCB version.
sysName="BarnMini-22"	Device name / identifier (writable).
sysContact="Operator Foo"	Contact person responsible for this device (writable).
sysLocation="The office"	Physical location of the system (writable).

3.2 Section "[hardware.ID]":

These sections contain information about how the hardware blocks used by input and output ports are constructed.

Each section is named "[hardware.ID]", where ID is a numeric identifier. Currently, these sections contain only one property: ***name***. “Hardware.1” is always None, describing that no hardware is present at this location.

Example:

```
[hardware.1]
name="None"
```

If this hardware block has parameters, these will be given by sections named "[hardware.ID.parameter.SUBID]" where SUBID is a number counting from 0.

These sections contain two properties: **name** and **syntax**. The syntax property is documented in the appendix.

Examples:

[hardware.1]	
name="None"	No hardware is present at this location.
…	
[hardware.14]	
name="GS12150"	Defines the name of this hardware block.
[hardware.14.parameter.0]	Parameter of hardware.14
name="Rate"	Defines the name of this parameter.
syntax="wE;9=Bypass;0=Auto;8=Auto (SDI only);1=MADI;2=SD-SDI/ASI;3=HD-SDI;4=3G-SDI;5=6G-SDI;6=12G-SDI;7=10G Ethernet"	Syntax/formatting is described in the appendix.
…	

3.3 The Network sections

3.3.1 Section "[network]"

The first section within the network namespace is name "[network]" and currently only contains two read-only properties: **version** and **id**.

- The software version of the network handling software (named “emnema” internally).
- The unique identifier of the frame (the Ethernet MAC address is use for this) .

Example:

[network]	
version="100.1.0"	Specifies the software version of the network handling software being used.
id="80:1f:12:40:87:41"	The MAC address of the network interface/Unique ID of the device. (in addition to Serial Number)

3.3.2 Section "[network.config]"

The “[network.config]” section contains the current configuration file used by the network configuration software and all properties in this section are writeable.

If the client wants to modify this configuration, it should first update all the desired properties (or all properties if needed) before finalizing the changes by setting “commit=1”. This is to avoid the frame reconfiguring its networks settings while they are being uploaded.

Example:

[network.config]	
ipv4.static.addr[0]="192.168.0.95"	Up to 4 static IPv4 addresses can be configured.
ipv4.static.prefix[0]=24	This contains the prefix (also known as the netmask in IPv4: 24=255.255.255.0) for the given static IPv4 address.
…	
ipv4.static.addr[3]="0.0.0.0"	
ipv4.static.prefix[3]=0	
ipv4.static.gw="192.168.0.1"	The gateway if IPv4 is configured to the static.
ipv4.mode=1	Operation mode for IPv4. 0=Disabled, 1=DHCP, 2=Static, 8=LinkLocal.
ipv6.static.addr[0]="2001::2"	Up to 4 static IPv6 addresses can be configured.
ipv6.static.prefix[0]=64	This contains the prefix for the given static IPv6 address.
…	
ipv6.static.addr[3]="::"	
ipv6.static.prefix[3]=0	
ipv6.static.gw="::"	The gateway if IPv6 is configured to the static.
ipv6.mode=12	Operation mode for IPv6. 8=LinkLocal only, 9=DHCP/StateFull, 10=Static,12=StateLess/Router Advertisement.
commit=0	Set this value to 1, in order to commit any changes.

3.3.3 Section "[network.status.ip.ID]"

This section provides the current status regarding IP address for the network interfaces. The current active IP addresses assigned to the frame are readable by the “[network.status.ip.ID]" sections. Each section will contain 3 properties: **dev**, **addr**, and **prefix** (The subnet mask length). Non-used entries will have prefix=0.

Example:

[network.status.ip.0]	
dev="st"	Assigned to the network interface st.
addr="192.168.0.116"	An IPv4 address assigned to this interface. This is the IPv4 address assigned to st (192.168.0.x).
prefix=16	Subnet mask of 255.255.0.0 (/16 means that the upper 16 bits in the subnet mask is set to 1).
[network.status.ip.1]	
dev="st"	Assigned to the network interface st.
addr="FE80::821F:12FF:FE40:8741"	A Link-Local IPv6 address. fe80::/10 addresses are link-local and used only within the local network. These addresses cannot be routed outside the local network.
prefix=64	Current prefix for the IPv6 address.
...	
[network.status.ip.3]	
dev=""	No network interface is assigned.
addr=""	No IP address is assigned.
prefix=0	No subnet mask is assigned.

3.3.4 Section "[network.status.route.ID]"

This section defines the routing table, specifying how network traffic should be directed.
The current routing table assigned to the frame are readable by the "[network.status.route.ID]" sections. Each section will contain four properties: **dev** (Network interface (eth0, etc.).), **dst** (for the destination network or IP address), **dstprefix** (for the subnet prefix length for the

destination.) and **gw** (the IP address of the gateway / next-hop router for forwarding packets). Non-used entries will have dst="".

Example:

[network.status.route.0]	
dev="st"	The network interface assigned is st.
dst="192.168.0.0"	This route covers the 192.168.0.0/16 network.
dstprefix=16	The subnet prefix is /16 (netmask 255.255.0.0), meaning it applies to all IPs in 192.168.0.0/16 network.
gw=""	No gateway assigned.
[network.status.route.1]	
dev="st"	The network interface st is used.
dst="0.0.0.0"	This is the default route (matches all destinations). Any traffic not matching other routes will be sent to gw, 192.168.0.1.
dstprefix=0	No specific subnet mask (applies to all traffic).
gw="192.168.0.1"	The gateway (router) IP is 192.168.0.1.
[network.status.route.2]	
dev="st"	
dst="FE80::"	This is the IPv6 Link-Local network.
dstprefix=64	Covers only fe80::/64 (Link-Local addresses). fe80::/64 is reserved for IPv6 link-local communication. It allows devices on the same physical network to communicate without an external router.
gw=""	No gateway (link-local traffic only).
[network.status.route.3]	
dev=""	No network interface assigned.
dst=""	No destination address assigned.
dstprefix=0	No subnet mask.
gw=""	No gateway assigned.
...	

3.4 Section "[port.ID]"

These sections contain all parameters associated with each physical port in the BarnMini-22.

Each section is named "[port.ID]", where ID starts from 1.

- If the port is not unidirectional, the corresponding input.* and output.* variables will be skipped.
- The word external is if the functionality is built directly into the matrix circuit or not.(To be compatible with BTF1-xx)
- If the port is an SFP port, the SFP.* variables will be included.

Examples:

[port.1]	
name="SFP #1"	The actual name of the port. Read-only.
SFP.present=1	Is an SFP detected in the port? 1=Yes, 2=No
SFP.connector="LC"	What kind of connector does this SFP have. Most common is LC.
SFP.lengths="10000;0;0;0;0;0"	Which cable lengths is this SFP designed for? The 5 numbers represent single mode fiber, multi-mode fiber OM1, OM2, OM3 and copper.
SFP.vendorname="Barnfind Tech"	The vendor of the SFP module.
SFP.partnumber="BTSFP-LX-SM-12G10"	The part number of the SFP module.
SFP.serialnumber="A89T140016"	The serial number of the SFP module.
SFP.productiondate=20181107	The production date of the SFP module (given in yymmdd format).
SFP.productionlot=""	The production lot of the SFP module.
SFP.revision="0"	The revision of the SFP module.
SFP.wavelengthnm=1310	The wavelength of the transmitter given in nm (for DWDM, the digits behind the decimal point are missing).
SFP.bitrate="12000000;600000;12600000"	This will contain 3 numbers separated by (;) if a SFP module is present. The first number will be the nominated designed bitrate given in KHz.

	The second will be the minimum and the third will be the maximum designed bitrates.
input.label="Input port 1"	User-writeable label for the input part of the port.
input.exteq.type=1	Specifies the type of external equalizer used for signal conditioning. (refer to [hardware.ID] for specific details).
input.exteq.parameters=""	User-writeable parameters, lookup in [hardware.ID] for syntax.
input.exteq.signaldetected=2	Signal detection for external equalizer. 0=Not able to detect (SFP module not inserted, or similar), 1=Detected, 2=Not detected.
output.label="Output port 1"	User-writeable label for the output part of the port.
output.extrc.type=14	Specifies the type of external reclocker used for output signals, ensuring a stable signal for downstream devices (refer to [hardware.ID] for details).
output.extrc.locked=2	Indicates if the reclocker is locked. 1=Locked, 2=Not locked.
output.extrc.lockedat=0	What bitrate is this reclocker locked at (given in KHz)
output.extrc.parameters="0"	User-writeable parameters (refer to [hardware.ID] for details).
output.extrc.signaldetected=2	Signal detection for the external reclocker. 0=Not able to detect (e.g., SFP module not inserted, or similar), 1=Detected, 2=Not detected.

3.5 The section for “SFP diagnostics”

SFP modules can have multiple diagnostics entries dynamically allocated. These entries will be located into sections with the following name scheme: “[port.ID.SFP.diag.SUBID]”. Here, the "ID" refers to the port number to which the module belongs, while "SUBID" represents the specific diagnostic entry for the SFP module.

Each section will contain 3 properties: the ***name***, ***value*** and the ***syntax***. (For description of the syntax, please see the appendix.)

If a diagnostics-entry is removed (e.g., when the SFP module is unplugged), the property named ***remove*** will be set to =1 in the affected sections.

Examples:

[port.1.SFP.diag.1]	This list of diagnostics available for a given SFP is dynamic. Each entry will have a different number, in this example we have diagnostic entry “1”
name="Temperature"	The name of the diagnostic entry.
syntax="rI;suffix=C;scale=0.00390625;amin=-2560;amax=23040;wmin=-1280;wmax=20480"	Syntax is described in the appendix. In this example, a read-only integer (rI).
value="7644"	Number will be stored as text, since syntax could allow text too.
remove=0	The diagnostic entry is active and not marked for removal.
[port.1.SFP.diag.9]	
name="TX Disable"	
syntax="wB"	Syntax is described in the appendix. In this example, a writable Boolean (wB) for selecting the SFP TX Disable.
value="2"	
remove=0	
[port.1.SFP.diag.12]	In this example we have diagnostic entry “12”

name="Present"	
syntax="rE;0=Empty;1=MSA;2=non-MSA"	Syntax is described in the appendix. In this example, this read-only enumeration (rE) specifies if an inserted SFP module is Empty (0), MSA-compliant (1), or non-MSA (2)
remove=0	
value="1"	According to syntax above, “MSA” is the current state of the SFP port.

3.6 The Firmware upload section

Firmware can be upgraded using a file upload using the “[firmware.upload]” section described here. Start by selecting this section.

- Then set the size parameter to the expected file upload size.
    - If successful, you will see that both the size and busy parameters will be set.
- If only the busy parameter is set to a non-zero value, it means another client is uploading. You can cancel any upload (including those from other clients) by setting the size parameter to 0.
- When the client has the handle for uploading content (with both size and busy set to non-zero values), data can be uploaded using a property called "chunk." The recommended size for each data packet is up to 64 bytes.

Example:

[firmware.upload]	
size=0	Writeable property. This tells the server the size of the file the client wants to upload. Will only be non-zero for the client that currently has the handle. Set to zero to cancel any clients currently uploading.
busy=0	Set to non-zero value if a client has the upload-handle.
chunk="...."	Write-only property for performing the actual file upload.

4. APPENDIX A: Syntax specifier

Various parts of the server software use a syntax string to define how a parameter should be interpreted. A syntax specifier can consist of multiple components or properties, each separated by a semicolon (;).

The first property will always define the data type and protection. The first character indicates whether it is read-only or writable: r for read-only and w for writable. The remaining characters specify the data type:

- **B** for boolean:

  (0 = unset, 1 = true, 2 = false). A writable boolean will normally fail if you try to set it to 0.
- **I** for integer:

  This may be followed by options to apply scaling, warnings, error limits, and suffixes.

  Example:
  - I;scale=0.1;wmin=10;wmax=100;emin=0;emax=110;min=-100;max=200;offset=10;suffix=mW
- **E** for enumeration:

  The following properties will define the available options.

  Example:1=option1; 2=option2
- **S** for string:

  Possible properties include len=x to limit the string length. The redundancy switch has a special rule when setting autosource-mask-v1. This string is expected to consist of 4 series of digits separated by hyphens (-) sign. Each digit matches a valid input port from the port that has input properties.

  The groups represent:
  - The first group: "Enabled" column (0 = not used, 1 = main, 2 = backup)
  - The second group: "Sensitive to LOS" column (0 = not set, 1 = set)
  - The third group: "Sensitive to analyzer lock"
  - The fourth group: "Sensitive to analyzer errors" column.

Future Data Types:

- **IPv4** for an IPv4 address.
- **IPv4Net** for an IPv4 address with a prefix (e.g., 192.168.0.255/24).
- **IPv6** for an IPv6 address.
- **IPv6Net** for an IPv6 address with a prefix.

- **IP** for an IPv4 or IPv6 address.
- **IPNet** for an IPv4 or IPv6 address with a prefix.