Glossary 1
Introduction 1
I. Ethernet general information 1
II. Power supply general information 2
A. Power supply state 2
B. Command 2
C. Interlocks 2
III. Modbus/TCP 4
A. Overview 4
B. Standard mapping 4
IV. Custom ASCII protocol 5
A. Overview 5
B. Standard commands 5
1. Get local/remote state 5
2. Command 5
3. Get output current 5
4. Get output voltage 5
5. Get output current error 5
6. Get output current reference 6
7. Set output current reference 6
8. Get software interlocks 6
9. Get hardware interlocks 6
10. Get state 6

Glossary

PS Power Supply

Introduction

This document gives general details about the Ethernet interface and then presents the two different interfaces to allow customer to choose which one will be made available: Modbus/TCP or custom ASCII protocol.

I. Ethernet general information

The Ethernet interface enables the machine to be connected to a local network. These are its characteristics:

- Machine controlled in remote mode
- Reading of the machine state in local and remote modes
- Speed: 10 or 100 Mbps
- Connector: RJ45
- Services:
  - Control/command via Modbus /TCP protocol or Telnet client
  - Maximum 2 TCP connections (Modbus/TCP or Telnet)

The power supply IP address can be fixed or determined by a DHCP. The USB interface allows the settings to be made.

II. Power supply general information

A. Power supply state

The power supply state machine can take several values. A variable (UNSIGNED8), which can be read via Ethernet, gives the current position of the state machine.

For instance:

	Name : coState	UNSIGNED8	
Description	This object gives the current position of the general state machine. **Read only.**		
Value	**State**	**Corresponding values**	
	• START sequence	0xFF or 0x01	
	• FAULT	0x80	
	• ACK sequence	0x81 or 0x82 or 0x83 or 0x84	
	• IDLE	0x22	
	• INRUSH sequence	0x24 or 0x25 or 0x26	
	• ON	0x27	
	• STOPPING	0x29	

B. Command

Controlling the power supply remotely can be done by writing a variable. The value written defines the order. Once taken into account, the variable is cleared by software.

For instance:

	Name : coOrder	UNSIGNED16	
Description	This object is used to remotely drive the power supply. Commands are relative to the current state of the power supply. **Write only.**		
Value	• **ACK** **0x0003** **Fault acknowledgement** When in FAULT state, resets interlocks (if possible). • **ON** **0x0011** **ON** Starts INRUSH sequence. • **OFF** **0x0012** **OFF** Stops power supply.		

C. Interlocks

Interlocks are divided into two groups: hardware interlocks and software interlocks.

Software interlocks are generated by software (like a communication watchdog). In this case the firmware opens the main relays and stops the PWM.

On the control unit, a dedicated hardware circuitry with several inputs (normally closed mechanical contacts or analog windowing of measures) is also present. If a hardware interlock occurs (if the mechanical contact linked to magnet temperature is open for instance), main relays are opened and PWM is stopped by hardware. The software reads the status of hardware interlocks.

Depending on control unit type and on the power supplies, up to 17 hardware interlocks and up to 16 software interlocks are present.

The following tables give an **example** of interlocks mapping.
Detailed mapping will be provided after design.

	Name : coSoftFault	UNSIGNED16	
Description	This object gathers all software interlocks. A fault is memorized until acknowledgement. **Read only.**		
Value	Each bit corresponds to an interlock. High for interlock pending. • Bit 0 : EEPROM Board configuration is stored in an EEPROM. An error rises at startup if data read from EEPROM are not valid or if EEPROM in uninitialized. • Bit 1 : State machine Internal software error (timeout or unexpected case). • Bit 2 : CAN watchdog • Bit 3 : Regulation parameter set A wrong regulation parameter set was asked. • Bit 4: Serial/Parallel board error (no PWM ON return). • Bit 6..5: n/a • Bit 7: MEAS1 (current) out of range (software) • Bit 8: MEAS2 (voltage) out of range (software) • Bit 9: MEAS3 out of range (software) • Bit 10: MEAS4 out of range (software) • Bit 11: MEAS_IPRIM (primary current) out of range (software). • Bit 12: Inverter error (inverter in a wrong state) • Bit 12: Bus voltage error (bus voltage too low) • Bit 15..14: n/a		

	Name : coHardFault	UNSIGNED32	
Description	This object gathers all hardware interlocks. A fault is memorized until acknowledgement. **Read only.**		
Value	Each bit corresponds to an interlock. High for interlock pending. • Bit 0: Heatsink • Bit 1: Temperature TM1 • Bit 2: Temperature L1 • Bit 3: Door • Bit 4: Phase • Bit 5: Emergency • Bit 6: Temperature L2 • Bit 7: DCCT • Bit 8: External 1 • Bit 9: External 2 • Bit 10: n/a • Bit 11: n/a • Bit 12: Water • Bit 13: n/a • Bit 14: Overcurrent (MEAS1) • Bit 15: Overvoltage (MEAS2) • Bit 16: Primary current overcurrent (MEAS_IPRIM)		

III. Modbus/TCP

A. Overview

The Modbus/TCP connections constitute the main means of controlling power supply units. To use them effectively, it is necessary to use supervision software on the user side. The port to which the supervisor connects is port 502. The following Modbus/TCP functions are supported:

- Function 3, ReadMultipleRegisters
- Function 4, ReadInputRegisters
- Function 6, WriteSingleRegister
- Function 16, WriteMultipleRegisters

The registers are all defined in a single table in the Modbus/TCP sense. Bit-type objects are not supported.

B. Standard mapping

Register	Description	Details
00 - W	Command	UNSIGNED16. See details.
01 - R	Output current LSB	FLOAT32 Output current [A].
02 - R	Output current MSB	
03 - R	Output voltage LSB	FLOAT32 Output voltage [V].
04 - R	Output voltage MSB	
05 - RW	Current reference LSB	FLOAT32 Current reference [A].
06 - RW	Current reference MSB	
07 - R	Output current error LSB	FLOAT32 Output current error [A].
08 - R	Output current error MSB	
09 - R	Local/Remote	UNSIGNED16. 1: Remote - 0: Local
10 - R	Power supply state	UNSIGNED16. See details.
11 - R	Software interlocks	BITFIELD. See details.
12 - R	Hardware interlocks LSB.	BITFIELD See details.
13 - R	Hardware interlocks MSB.	

IV. Custom ASCII protocol

A. Overview

On the user side, it is necessary to use a Telnet client, such as HyperTerminal or PuTTY. The machine listens on port 23. The Telnet client can send data character by character, or send whole lines. The size of the line must be 40 characters or less. If this number is exceeded, the excess characters will be refused, and the client is informed by the emission of "Bell" characters (ASCII code 7).

A prompt appears to indicate the beginning of a line, in the form of a '>' character followed by a space. While the line is being entered, the console re-emits the characters received. The reception of a CR or LF character tells the console that a complete command line is available. This line is then interpreted and executed. The result of the execution can send messages on several lines, which are ended with a CR+LF sequence (ASCII codes 13, 10). Commands can be entered in either upper or lower case.

Minimum editing possibilities are provided by the console. For this, the following keys are used:

- Escape or Ctrl-C cancels a line while it is being entered
- Backspace deletes the last character entered

Furthermore, Ctrl-D or the command "Q" are used to close a session.

B. Standard commands

1. Get local/remote state

Telnet:	**REM/**	Read Local / Remote State
Coding:	integer	
Example:	**REM/ 1**	Status Remote
	REM/ 0	Status Local

2. Command

Telnet:	**ORD=**	Access to all remote software commands
Coding:	integer	
Example:	**ORD=3**	**ACK** command
	ORD=17	**ON** command, 17 corresponds to 0x11
	ORD=18	**OFF** command, 18 corresponds to 0x12

3. Get output current

Telnet:	**CUR/**	
Coding:	float	
Example:	**CUR/ 1.234**	Read output current, equal to 1.234A

4. Get output voltage

Telnet:	**VLT/**	
Coding:	Float	
Example:	**VLT/ 22.523**	Read output voltage, equal to 22.523V

5. Get output current error

Telnet:	**CER/**	
Coding:	Float	
Example:	**CER/ 0.002**	Read output voltage, equal to 0.002A

6. Get output current reference

Telnet: **REF/**
Coding: Float

Example: **REF/ 5.500** Read current reference, was set to 5.500A

7. Set output current reference

Telnet: **REF=**
Coding: Float

Example: **REF= 18.20** Read current reference, was set to 18.20A

8. Get software interlocks

Telnet: **ITS/** Read Software Interlocks
Coding: hexadecimal

Example: **ITS/ 00000002** Software interlock Bit 1 active ‘State machine’

9. Get hardware interlocks

Telnet: **ITH/** Read Software Interlocks
Coding: hexadecimal

Example: **ITH/ 00000044** Software interlock Bits 3 and 7active

10. Get state

Telnet: **STA/** State machine position
Coding: hexadecimal

Example: **STA/ 00000022** State machine position: 0x22 (IDLE)