SynQor 3-Phase PFC Terminal Commands

The SynQor 3-Phase PFC contains a serial port to facilitate system monitoring.
Commands are sent to request individual data, the PFC responds to each command individually.
The PFC's external serial port IO pins are at 3.3V logic levels, not RS232 levels.
See the device datasheet for IO pin level & polarity details.
Commercial data converters are available to translate these levels to RS232 or USB.
The serial port uses 9600 baud, 8 data bits, no parity, and one stop bit. Commands can be sent, and output viewed, with any standard terminal emulator. The interface is case sensitive. To view user text, set the terminal emulator for local echo.
Some data are writeable; written values are, however, volatile and will revert to default upon power cycling.
A pause of >5 seconds between any two comand characters will timeout/reset the command parser. No response is generated.
This timeout value is user settable - see Hung Parser Timeout below.
The protocol supports communication to/from individual modules on a shared serial bus via a unique 'network address'.
(See the Enumerate command, available in models featuring a START SYNC control pin)
In this document, the symbol ↙ will be used to represent ascii 13 (cr) OR ascii 10 (lf) OR both (they're all equivalent).

Value Formats

Data supplied to, and returned from, the PFC can be communicated in several formats, broadly divided into strings and numeric values. While strings are a unique type, numeric data can be further divided into serveral interchangeable formats.

Strings	**Up to 64 ascii characters, and "enclosed in double quotes". A null string is "". Strings can contain unprintable and control characters which can, of course, affect the appearance of any terminal emulator receiving them.**			
Numerical Formats	**Fmt**	**Numeric Range**	**Description**	
	char	**0 to 127**	**An ascii character, preceeded by single quote character ' to remove ambiguity. Responses using char format will always contain the ' prefix.**	
	hex	**0x0 to 0xFFFF**	**0x and then zero to four ascii hex characters**	**Hex, U int, and int formats have no internal units conversion**
	U int	**0 to 65535**	**Unsigned decimal numeral, optionally preceeded by 'u'**	
	int	**-32768 to 32767**	**Signed decimal numeral, specified by '+' or '-' prefix**	
	mks	**±###.#**	**Decimal numerals *containing a decimal point* are interpreted as real MKS values and are internally converted to/from machine units. MKS values are naturally signed. MKS format is recommended whenever referring to a real-life datum.**	

Command Formats

Both data read and data write commands are supported

Data Write Command Syntax

{Net Addr}	{Access Address}	{Action Specifier}	{Write Value}	{Terminator}
m	0-127	=	"String" or 16-bit numerical	(cr) or (lf) or both, ↙

Net Addr: **The network address character, which takes a default value of 'm on power-up. 'A thru 'z is a single unit's address, while '@ is interpreted as all units. To avoid collision, the response to '@ commands is automatically muted.**
See the Enumerate command to automatically assign unique network addresses for multi-unit applications.

Access Addr: **The address to write, see table below.**
Addresses can be specified in char, hex, or int formats, see Value Formats below.

Action Specifier: **The = character specifies a data write command.**
Only select addresses are writeable.

Write Value: **Values to write can be supplied in the formats described below;**
The supplied format will determine the format of the response datum.

Terminator: **a carriage return character (cr) = ascii 13, OR a linefeed character (lf) = ascii 10, OR both ↙.**

SynQor 3-Phase PFC Terminal Commands

Data Read Command Syntax

{Net Addr}	{Access Address}	{Action Specifier}	[Format]	{Terminator}
m	0-127	?	optional value of 0	(cr) or (lf) or both, ↙

Net Addr: The network address character, which takes a default value of 'm on power-up.
while valid, the '@ address mutes responses to avoid collision; it's not useful for read operations.
See the Enumerate command to automatically assign unique network addresses for multi-unit applications.

Access Addr: The address to read, see table below.
Addresses can be specified in char, hex, or int formats, see Value Formats below.

Action Specifier: The ? character specifies a data read command.

Format: The numerical response datum format of a read command can be controlled by supplying the command with an argument value of 0. The format of the 0 specifies the desired return datum format (see table below). In the absence of a 0 argument, the datum is returned in its natural or mks format.

Terminator: a carriage return character (cr) = ascii 13, OR a linefeed character (lf) = ascii 10, OR both ↙.

Examples of controlling the return datum format of read commands

m46?↙	No value supplied, return datum in default (mks or natural) format
m46?0.↙ m46?.↙	Value of 0 supplied in mks, return in mks/natural format
m46?0↙ m46?u0↙ m46?u↙	Unsigned value of 0 supplied, return as unsigned int.
m46?+0↙ m46?+↙	Signed value of 0 supplied, return as signed int.
m46?0x↙ m46?x↙	Value of 0 supplied in hex, return in hex.

Response Format

Immediately after reception of its terminator, each valid command generates a response.
Invalid commands do not generate a response.

Response Syntax

{Net addr}	{Access Address}	{Is Now}	{Value}	{Terminator}
m	0-127	=	"String" or 16-bit numerical	↙

Net Addr: The PFCs echoes its network address, 'm in the example above.

Access Address: Echo of the address specifier in the same format as supplied.
Char addresses will be prefixed by '.

Is Now: The character '='

Value: The address's current value.
See above for descriptions of the return datum format to both read and write commands.
Char format will always contain leading ', hex format will always contain 0x and four ascii hex value characters.

Terminator: ↙ concludes the response

SynQor 3-Phase PFC Terminal Commands

Addresses

The following table contains a summary of the addressesthat contain useful data.

Access Address hex,	dec,	or char	Name	Read / Write	Natural Format	Natural Units
0x0	0		ID String	R	string	
0x1	1		Part Number	R	string	
0x2	2		Serial Number	R	string	
0xC	12		Build Revision	R	hex	
0xD	13		Code Revision	R	string	
0x2E	46	.	Code Revision	R	hex	Rev<<8 | Subrev
0x21	33	!	Net Address	R	quoted character	
0x22	34	"	Hung Parser Timeout	R/W	##.#	seconds
0x23	35	#	Powered Days	R/W	int	days
0x24	36	$	Powered Hours	R/W	int	hours
0x25	37	%	Powered Minutes	R/W	int	minutes
0x26	38	&	Powered Seconds	R/W	##.#	seconds
0x41	65	A	Auto Command String	R/W	string	
0x61	97	a	Auto Command Interval	R/W	##.#	seconds
0x45	69	E	Enumerate Network Address	R/W	int	
0x46	70	F	AC Input Line Frequency	R	##.#	Hz
0x53	83	S	AC Input Voltage	R	###.#	Volts
0x73	115	s	AC Input Voltage Ripple/Imbalance	R	###.#	Volts
0x76	118	v	Midbus (Buck Stage) Output Voltage	R	###.#	Volts
0x69	105	i	Buck Stage Output Current	R	#.##	Amperes
0x70	112	p	Buck Stage Output Power	R	####	Watts
0x49	73	I	Boost Stage Input Current	R	#.##	Amperes
0x50	80	P	Boost Stage Input Power	R	int	Watts
0x56	86	V	Boost Output Voltage	R	###.#	Volts
0x54	84	T	PCB Temperature	R	##.#	°C
0x4D	77	M	Machine State	R	int	
0x4E	78	N	PFC ENA Pin Override	R/W	int	
0x6E	110	n	Battle Short Mode Enable	R/W	int	
0x63	99	c	AC Line Status	R	hex	
0c65	101	e	Stop Status	R	hex	
0x77	119	w	BS Warnings	R	hex	

Address Details

Address:	hex	decimal	char
	0x0	0	
Name:	**ID String**		
Description:	The unit's ID string "Synqor 3PH PFC"		
Response units:	string		
Read/Write:	Read Only		

Example Cmd:	**Typical Response:**
m0x0?↙	m0x0000= "Synqor 3PH PFC"↙
m0?↙	m0= "Synqor 3PH PFC"↙

Address:	hex	decimal	char
	0x1	1	
Name:	**Part Number**		
Description:	The part number "MPFC-115-3PH-270-FP" or "MPFC-115-3PH-270P-FP"		
Response units:	string		
Read/Write:	Read Only		

Example Cmd:	**Typical Response:**
m0x1?↙	m0x0001= "MPFC-115-3PH-270-FP"↙
m1?↙	m1= "MPFC-115-3PH-270-FP"↙

Address:	hex	decimal	char
	0x2	2	
Name:	**Serial Number**		
Description:	The unit's serial number, factory assigned in sequential order		
Response units:	String		
Read/Write:	Read Only		

Example Cmd:	**Typical Response:**
m0x2?↙	m0x0002= "S17139017"↙
m2?↙	m2= "S17139017"↙

Address:	hex	decimal	char
	0xC	12	
Name:	**Build Revision**		
Description:	The unit's Build Revision		
Natural Response units:	hex		
Read/Write:	Read Only		

Example Cmd:	**Typical Response:**
m0xC?↙	m0x000C= 0x0000↙
m12?↙	m12= 0x0000↙

Address:	hex	decimal	char	
	0xD	13		
Name:	**Code Revision**			as a string
Description:	The unit's Code Revision			
Response units:	string			
Read/Write:	Read Only			

Example Cmd:	**Typical Response:**
m0xD?↙	m0x000D= "Rev 5, v26.6"↙
m13?↙	m13= "Rev 5, v26.6"↙

Address:	hex	decimal	char	
	0x2E	46	.	
Name:	**Code Revision**			as a number
Description:	The unit's Code Revision			
Natural Response units:	hex			
Read/Write:	Read Only			
Discussion:	Code Revision is returned as MainRev<<8 | Subrev			

Example Cmd:	**Typical Response:**	**Interpretation:**	
m.?↙	m'.= 0x1A06↙	0x1A00 = 26<<8 | 6	Code Rev is 26.6
m.?0↙	m'.= 6662↙	6662 = 26*256 + 6	Code Rev is 26.6

Address:	hex	decimal	char	
	0x21	33	!	
Name:	**Net Address**			
Description:	Returns unit's Network Address as a quoted character			
Natural Response units:	quoted character, 'm by default			
Read/Write:	Read Only			

Example Cmd:	**Typical Response:**
m0x21?↙	m0x0021= 'm↙
m33?↙	m33= 'm↙
m!?↙	m'!= 'm↙

---

	hex	decimal	char
Address:	0x22	34	"

Name: **Hung Parser Timeout**

Description: The Hung Parser Timeout

Natural Response units: decimal seconds

Read/Write: Read/Write

Discussion: A delay of >N.N seconds between any two command characters will reset the command parser.
This paramter controls the timeout value. It is both readable and writeable.
Supplied in hex or int format, the units are integer 1/16ths of second.
Supplied in MKS format, the units are decimal seconds.

Example Cmd:	Typical Responses:	Interpretation:
m0x22?↙	m0x0022= 5.0↙	Parser timeout is 5.0 seconds (default)
m"=60↙	m'"= 60↙	(Note int units supplied) Parser timeout set to 60/16 = 3.75 secs.
m"=?↙	m'"= 3.75↙	Confirm 3.75 sec timeout in MKS units.
m34=60.↙	m34= 60.0↙	(Note MKS units supplied) Parser timout re-set to one minute

---

	hex	decimal	char	Response units	
Powered Days	0x23	35	#	int 0-65535	Read/Write
Powered Hours	0x24	36	$	int 0-23	Read/Write
Powered Minutes	0x25	37	%	int 0-59	Read/Write
Powered Seconds	0x26	38	&	decimal 0-59.9	Read/Write

Discussion: Togethe
They

Example Cmds:	Typical Responses:	Interpretation:
m35?↙	m35= 329↙	Unit has been powered for 329 days
m36?↙	m36= 3↙	and 3 hours.
m35=0↙	m35= 0↙	Days variable reset to 0.
m36=0↙	m36= 0↙	hours variable reset to 0.

---

	hex	decimal	char
Address:	0x41	65	A

Name: **Auto Command String**

Description: The Auto Command String will have the PFC send the sama data repeatedly and automatically.

Response units: string

Read/Write: Read/Write

Discussion: The AutoCommandString is written to contain the exact commands to read the desired data. The string's contents are copied directly to the command parser input at an interval given by the Auto Command Interval, see below. This command should only be used with a single unit to avoid response data

Example Cmd:	Typical Responses:	Interpretation:
mA?↙	m'A= ""↙	The Auto Command String is empty.
mA="mS?↙mF?↙mP?↙"↙	m'A= "mS?↙ms?↙mF?↙mP?↙"↙	Write the auto command string with: Read AC Input Voltage, Read AC Line Frequency, Read Boost Stage Power.

Address:	hex	decimal	char
	0x61	97	a
Name:	**Auto Command Interval**		
Description:	Controls how often the Auto Command String is executed.		
Response units:	decimal seconds		
Read/Write:	Read/Write		
Discussion:	A value of 0 (default) turns OFF Auto Command execution		
	The range of active values is from 1 to 65535 1/16ths of a second, a bit over an hour.		
	If the value is supplied in hex or int formats, the units are 1/16ths of a second		
	If the value is supplied in MKS format, the units are decimal seconds.		

Example Cmd:	Typical Responses:	Interpretation:
ma?↙	m'a= 0↙	Auto Command is OFF
ma=0.5↙	m'a= 0.5↙	Execute Auto Command String every half second

---

Address:	hex	decimal	char
	0x45	69	E
Name:	**Enumerate Network Address**		
Description:	Enumeration automatically sets the network address of each unit in a multi-unit application.		
Write Value Argument:	The value of the **first** adddress to be auto-assigned (when after = sign, char format must include leading ').		
Read Response value:	The number of sequential addresses just assigned.		
Read Response units:	int		
Read/Write:	R/W		
Discussion:	At power-up, network address for each unit is reset to 'm. While that's sufficient for single-unit application a response collision would occur if multiple units shared the default 'm address.		
	Upon receiving an Enumerate command, all addressed units will communicate over their interconnected StartSync line and assign themselves unique sequential network addresses.		
	Note that the units must be disabled to communicate over StartSync.		
	Enumerated addresses are volatile; they will revert to the default 'm whenever power is cycled.		

Example Cmd:	Typical Responses:	Interpretation:
@E='A↙	none (response muted for global addr @)	Enumerate network addresses, begin assignments at 'A
AE?↙	A'E=5↙	Unit A, how many addrs were assigned. Answer=5
Host now knows that there are 5 units on the network; their addresses are A, B, C, D, & E.		
Host can now communcate with them individually using any other command		
A0x2?↙	A0x0002= "S17139017"↙	unit A: what is your serial number?
B0x2?↙	B0x0002= "S15930713"↙	unit B: what is your serial number?
C0x2?↙	C0x0002= "S17166150"↙	unit C: what is your serial number?
D0x2?↙	D0x0002= "S15922316"↙	unit D: what is your serial number?
E0x2?↙	E0x0002= "S15922317"↙	unit E: what is your serial number?

Address:	hex	decimal	char
	0x46	70	F
Name:	**AC Input Line Frequency**		
Description:	The AC Line Frequency		
Natural Response units:	decimal Hz		
Read/Write:	Read Only		
Discussion:	The accuracy of this measurement is limited by that of the internal timebase, which has a ±2% rating. Negative values indicate CBA rotation		

Example Cmd:	Typical Responses:	Interpretation:
m0x46?↙	m0x0046= 59.9↙	60 Hz
m70?↙	m70= 400.2↙	400 Hz
mF?↙	m'F= -60.1↙	60 Hz, CBA

Address:	hex	decimal	char
	0x53	83	S
Name:	**AC Input Voltage**		
Description:	The AC		
Natural Response units:	decimal Volts		
Read/Write:	Read Only		
Discussion:	The value is the peak L-N voltage; divide by √2 to obtain rms.		

Example Cmd:	Typical Response:	Interpretation:
mS?↙	m'S= 169.2↙	The L-N AC input voltage is 169 Vpk = 115 Vrms

---

Address:	hex	decimal	char
	0x73	115	s
Name:	**AC Input Voltage Ripple/Imbalance**		
Description:	The AC Input Voltage Ripple/Imbalance		
Natural Response units:	decimal pk-pk Volts.		
Read/Write:	Read Only		
Discussion:	Unequal line amplitudes and non-ideal phase angles appear as ripple in the 3-phase line voltage.		

Example Cmd:	Typical Response:	Interpretation:
ms?↙	m's= 2.2↙	There's 2.2V p-p of imbalance in the L-N voltages.

Address:	hex	decimal	char
	0x76	118	v
Name:	**Midbus (Buck Stage) Output Voltage**		
Description:	The +Midbus output voltage		
Natural Response units:	decimal Volts DC		
Read/Write:	Read Only		
Discussion:			

Example Cmd:	Typical Response:	Interpretation:
mv?↙	m'v= 205.1↙	The Midbus output voltage is 205.1 VDC

Address:	hex	decimal	char
	0x69	105	i
Name:	**Buck Stage Output Current**		
Description:	The Buck-Stage output current		
Natural Response units:	decimal Amperes DC		
Read/Write:	Read Only		
Discussion:			

Example Cmd:	Typical Response:	Interpretation:
mi?↙	m'i= 3.21↙	The buck stage is delivering 3.21 Adc (out)

Address:	hex	decimal	char
	0x70	112	p
Name:	**Buck Stage Output Power**		
Description:	Returns the Buck-Stage output power		
Natural Response units:	Watts		
Read/Write:	Read Only		
Discussion:			

Example Cmd:	Typical Response:	Interpretation:
mp?↙	m'p= 658↙	The buck stage is processing 658 W

Address:	hex	decimal	char
	0x49	73	I
Name:	**Boost Stage Input Current**		
Description:	The Boost-Stage **input** current (for approximate output current, multiply by 0.985*Vmidbus/Vout)		
Natural Response units:	decimal Amperes DC		
Read/Write:	Read Only		
Discussion:			

Example Cmd:	Typical Response:	Interpretation:
mI?↙	m'I= 3.2↙	The boost stage is also handling 3.2 A (in)

Address:	hex	decimal	char
	0x50	80	P
Name:	**Boost Stage Input Power**		
Description:	The Boost-Stage input power		
Natural Response units:	Watts		
Read/Write:	Read Only		
Discussion:			

Example Cmd:	Typical Response:	Interpretation:
mP?↙	m'P= 650↙	The boost stage is also drawing 650 W in.

Address:	hex	decimal	char
	0x56	86	V
Name:	**Boost Output Voltage**		
Description:	The Boost-Stage output voltage		
Response units:	decimal Volts DC		
Read/Write:	Read Only		
Discussion:			

Example Cmd:	Typical Response:	Interpretation:
mV?↙	m'V= 260.2↙	The boost stage is producing 260.2 VDC.

Address:	hex	decimal	char
	0x54	84	T
Name:	**PCB Temperature**		
Description:	The unit's internal temperature		
Natural Response units:	signed decimal °C		
Read/Write:	Read Only		
Discussion:			

Example Cmd:	Typical Response:	Interpretation:
mT?↙	m'T= 63.5↙	The PFC is at about 63 °C internally

Address:	hex	decimal	char
	0x4D	77	M
Name:	**Machine State**		
Description:	The present operational state number of the PFC		
Responses:	1	Waiting for all conditions to be ready to start	
	2	Starting	
	3	Running	
	4	Stopping	
	5	Waiting for auto-restart timeout	
Natural Response units:	int		
Read/Write:	Read Only		
Discussion:			

Example Cmd:	Typical Response:	Interpretation:
mM?↙	m'M= 3↙	The PFC is Running

Address:	hex	decimal	char
	0x4E	78	N
Name:	**PFC ENA Pin Override**		
Description:	Allows overriding the PFC ENA pin via serial interface		
Response units:	int		
Read/Write:	Read / Write		
Discussion:	value of 10 (0xA) allows unit operation to follow PFC ENA pin [default]		
	writing value of 11 (0xB) overrides PFC ENA pin and forces the unit ON		
	writing a value of 12 (0xC) overrides PFC ENA pin and forces the unit OFF		

Example Cmd:	Typical Response:	Interpretation:
mN?↙	m'N= 0x000A↙	unit m is obeying PFC ENA pin
@N=11↙	none (response muted for global addr @)	Everyone: Ignore PFC ENA pin value and turn ON
mN=12↙	m'N= 12↙	unit m, Ignore the PFC ENA pin value and turn OFF.

Address:	hex	decimal	char
	0x6E	110	n
Name:	**Battle Short Mode Enable**		
Description:	Allows engaging Battle Short mode via serial interface		
Natural Response units:	hex		
Read/Write:	Read/Write		
Discussion:	value of 10 (0xA) allows Battle Short mode to follow BS Pin [default]		
	writing value of 11 (0xB) turns on Battle short mode regardless of BS Pin value		

Example Cmd:	Typical Response:	Interpretation:
mn?↙	m'n= 0x000A↙	unit m is obeying BS pin
@n=11↙	none (response muted for global addr @)	Everyone: Battle Short Mode = On
mn=10↙	m'n= 10↙	unit m, obey BS pin again

Address:	hex	decimal	char
	0x63	99	c
Name:	**AC Line Status**		
Description:	Status		
Responses:	bit 0	AC Line Voltage has excessive imbalance	
	bit 1	AC Line Frequency is outside normal ranges	
	bit 2	AC Line Voltage is <80 Vrms L-N	
	bit 3	AC Line Voltage is >145 Vrms L-N	
Natural Response units:	hex		
Read/Write:	Read Only		
Discussion:	Any bit set indicates an out-of-range condition for the AC Line		

Example Cmd:	Typical Response:	Interpretation:
mc?↙	m'c= 0x0004↙	AC Line voltage is low

Address:	hex	decimal	char
	0c65	101	e
Name:	**Stop Status**		
Description:	Status bits that indicate a no-operate condition		
Responses:	bit 0	NA	
	bit 1	Over Temperature	
	bit 2	Midbus short timeout	
	bit 3	Low bias suppply output voltage	
	bit 4	NA	
	bit 5	PFC ENA	
	bit 6	NA	
	bit 7	Fast Overcurrent detect	
	bit 8	NA	
	bit 9	Low bias supply input voltage	
Natural Response units:	hex		
Read/Write:	Read Only		
Discussion:	Any bit set indictates why the unit shut down (or will not start).		

Example Cmd:	**Typical Response:**	**Interpretation:**
`mc?↙`	`m'c= 0x0020↙`	PFC ENA set to OFF

Address:	hex	decimal	char
	0x77	119	w
Name:	**BS Warnings**		
Description:	Status bits associated with the BATTLE SHORT pin (warnings)		
Responses:	bit 0	NA	
	bit 1	Warn Over Temperature	
	bit 2	NA	
	bit 3	NA	
	bit 4	Warn AC Line Imbalance.	
Natural Response units:	hex		
Read/Write:	Read Only		
Discussion:	These flags indicate the reason the BATTLE SHORT pin internal pull-down transistor has been turned off		

Example Cmd:	**Typical Response:**	**Interpretation:**
`mw?↙`	`m'w= 0x0002↙`	Battle Short Warning is due to temperature