Distribution list:

name	company	rev.	date	notes
various	Newave	0.1	17.1.03	discussion basis
various				as per separate document

Modification History:

rev.	nr.	page	ch.	modification	status[1]
0.1	1	-	-	new document	P
1.0	2	-	-	various. First official release	P
1.1	3	2	2.4	Additional note	P
	4	2	2.5	“Get Next Event” assigned to different XID	P
	5	6	3.1.3	idem	P
	6	4	3.1.2	Correction to the max number of events returned in 1 msg	P
	7	5	3.1.3	Bit b7 “Maintenance bypass is on” remove (double)	P
	8	5	3.1.3	Bit b12. Appropriate naming	P
	9	5	3.1.3	Bit b13. Appropriate naming	P
	10	5	3.1.3	Bits b16, b17, b18: breaker open status means “1”	P
	11	5	3.1.3	Bits b20,b21 deplaced to b22,b23	P
	12	5	3.1.3	Bits b19..b21: 3 additional Breaker introduced	P
	13	3	2.7	New protocol command for password protection	P
	14	7	3.1.3	idem	P
	15	6	3.1.3	Password protection for “UPS Commands”	P
	16	6		Password protection for “Execute UPS Function”	P
1.2	17	6	3.1.3	UPS Status bits. Reassigned bits 23..26	P
	18	6	3.1.3	UPS status bits. Bits 7,15,23,31 fixed to 1	P
	19	6/7	3.1.3	UPS all alarms. Reassignment and extension to auxiliary xid’s 202, 202 and 401	P
	20	8	3.1.3	New xid 3024, for a set of measurement values	P
1.3	21	8	3.1.3	New measures added to xid 3024 and values delimited by comma	P
1.4	22	7	3.1.3	UpsAllAlarms: xid 607 was considered twice → Bits b8, b9 and b10 has been reassigned (all of them are anyway not used/defined by the UPS)	P
	23	-	-	The note that “special care should apply in order to avoid the char 3Eh (‘>’)” has been eliminated from various places. Indeed the special char is never generated, because all numerical values are ASCII coded before being transmitted.	P
1.5	24	6	3.1.3	Bit b21 of UPS Status bits: the name has been corrected	P
	25	8	3.1.3	Get Next Events: modification in the description: “the readout is initiated by a **get** command” (not by a write command)	P

[1] RFC=Request for comments; T=Temporary; F=Final; P=Published (definitive)

	26	7,8	3.1.3	Some text added to XIDs 645 and 3024	P
	27	8	3.1.3	Mixed Measurements (XID 3024): Load% is now in units of 0.1% (value multiplied by 10)	P
1.6				11.11.2003	P
	28	6	3.1.3	Note to UPS_Status bit 17	P
1.7				October 2005	-
	29	8	3.1	XID3024, output voltage A-N is xid410 (and not 420)	P
1.8				June 2006	-
	30	9	3.1.3	New commands (XID0170 and XID0171) for the configuration of the UPS and UPS Interface Boards.	RFC
	31	9	3.1.3	New command 3025 for retrieving the status of the relays	RFC
	32	10	3.1.3	New command 112 (similar to 102, but for internal use)	RFC
	33	10	3.1.1	New XID105, for requesting the unique UPS ID	RFC

1 Scope of the document

The introduction of new parameters into the protocol is required in order to know more details about the UPS. This document describes the newly introduced items. Particularly the definition of a new “status word” reflecting the status of the elements of the UPS, the access to the UPS-internal event buffer (a kind of log-file) and the capability of controlling the UPS operating mode.

2 New Protocol Commands

2.1 Status Word

The status word is introduced as new item into the Group GID=30, using a long type (32 bits). We choose for this purpose the item “Misc Long #1” having **XID=3017**, which is renamed to **UPS Status bits**. The status of each element of the UPS, which is normally on or off, is assigned to a specific bit of the status word, with 0 meaning “off” and 1 meaning “on”. On the same item we find also some specific status information regarding the operating mode of the UPS and its input/output lines (1 meaning that the sentence is true). The item is read-only.

Notes: instead of using a bit-wise assignment approach, one alternative would have been to introduce for every UPS element a specific item (XID). The great disadvantage is of course the amount of communication needed to determine the whole status of the UPS. The status information is the basis for fast reacting and for fast graphical display refreshing. As a such it will be requested frequently, more often than other items. Consequently the solution to assign all information to a single long-item is optimal and considers the requirement for fast processing and lower communication bandwidth.

2.2 Available Status bits

The previous parameter, "Ups Staus bits" could be modified during lifetime. It is therefore important to have knowledge on which bits are supported and which not, in order to avoid false alarm triggering or similar effects. Every external application has therefore to use this information to mask out invalid bits from the status word and has never to react to all bits declared as invalid.

The item "**Available Status bits**" is introduced with **XID=3018**. The parameter is firmly defined by the UPS and is read-only.

2.3 UPS Commands

For management purposes it is important to have total control over the UPS. As an example think at the implementation of the energy management, where it is required to switch off/on the Inverters of a parallel system. A series of commands are therefore implemented through the additional parameter "**UPS Command**", with **XID=806**

2.4 Ups Alarm Info

The alarm group is currently a collection of 41 parameters. In order to know if there are newly generated alarms on the UPS, all parameters has to be inquired, requiring bandwidth and <u>time</u>. Not to mention if it is a parallel UPS system. In order to lean the communication requirements, all 41 parameters are collected in a single newly created parameter: "**UPS All Alarms**", with **XID=645**.
NOTE: The introduction of this command could be avoided if the UPS supported Group-Request Messages (GR<id>).

2.5 UPS Event Buffer

The UPS keeps track of various events in a so called "event buffer", which has a limited dimension and is cyclic (when it is full, a new event will overwrite the oldest one). The event buffer is very crucial for servicing. A new protocol parameter "**Get Next Event**" with **XID=3021** ensures the access to this event buffer.

2.6 Generic UPS Function

This command allows to execute some special UPS functions, provided by the UPS-Firmware. The caller has to specify on ***Fnum*** which function has to be executed. As for any function call, it is possible to pass calling-parameters (via "InputValues") and get back results (via "OutputValues"), using the data-field. In order to be very generic the data-field is a string. This functionality is implemented by means of the new protocol item "**Execute UPS Function**", with **XID=3023**. Normally it is used for setup/config, calibration or debugging purposes.
The meaning of every function is strictly UPS dependent and can freely

be defined by the firmware developers. Any detail is therefore beyond the scope of this document.

NOTE: Every get message (G) has the data-field consisting of the <xid> specification, followed by a string holding the function number *Fnum* and all requested calling-parameters. The length of he data-field is subsequently variable, depending on the function-number.

2.7 Password protection

In order to protect the UPS from unauthorized access for critical operations, a password protection scheme is introduced. To be resistant against message spying the implementation is based on a scrambling approach, as follows:

1. the requester (e.g. the management application) notifies to the UPS it would like to deactivate protection (sending a Get-Message).
2. the ups gives back to the requester a randomly generated value
3. the requester modifies this random value based on a scrambling algorithm and uses this generated value as password.

The scrambling algorithm, which should be kept secret, is known by the authorized requester and by the UPS so that the firmer can generate the password and the latter is able to verify it for correctness.

Password protection is implemented by means of the new protocol item "**Password**", with **XID=3022**.

2.8 Selected measurement values

To accelerate the refreshing of sensitive data, some special measurement values are grouped in a single command "**Mixed Measures**", with **xid=3024**

2.9 Various

Additional Protocol Commands are introduced, as they are needed.

3 Specifications

3.1.1 Timestamp

The timestamp in normally assigned to each event to identify the precise moment it has been generated. The protocol has to support two different timestamp formats, in consideration that not each UPS has a real-time clock.

3.1.1.1 Timeticks (running counter)

MSB=0

MSB	B3							B2								B1								B0							LSB
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	n09	b08	b07	b06	b05	b04	b03	b02	b01	b00
0	counter value																														

The counting is relative to the ups installation date, which has to be known externally. Resolution is 200 ms. The counter value can cover a time span of 13.62 years.

3.1.1.2 Real time clock

MSB=1

MSB	B3							B2								B1								B0							LSB
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	n09	b08	b07	b06	b05	b04	b03	b02	b01	b00
1	year offset					month				day					p	hour				minutes						seconds					

where:

year offset	[0..31]	=	the year relative to 2000. To have the absolute year, add 2000 to the 'year offset'.
month	[1..12]	=	the month
day	[1..31]	=	the day
p	[0,1]	=	am (0) / pm (1) identifier. Midnight is specified as 00:00 am Noon is specified as 00:00 pm
hour	[0..11]	=	the am respectively pm hour
minutes	[0..59]	=	the minutes
seconds	[0..59]	=	the seconds

3.1.2 Access to the Event Buffer

The UPS returns the events stored in the buffer, in chronological order (oldest firstly). The parameter ***evLastTime*** and ***evLastCode*** permits to selectively request just part of the buffer, especially to request only new events, that is, events which has not yet been seen by the requester. To achieve this, the values from the last known event will be assigned to *evLastTime* and *evLastCode*.
The number of events returned in a single packet is limited to 5.
When ***evNRet=0*** then no more (new) events are stored in the Buffer. This is an indication that all events have been read out.
A separate chapter summarizes the meaning for each ***EventCode***, which is UPS dependent.
The event-buffer in the UPS is not cleared when the events are requested. The UPS buffer is circular, meaning that when it is full, a new event will overwrite the oldest one. The length of the event buffer may vary from UPS to UPS.

All data which is transferred by the Get- and Data-Messages are ASCII coded hex values (in brackets the number of required bytes).
Variables for the Get-Message (G):

evNReq (1) the maximum number of events which should be returned [1..5].

evLastTime (8) the Timestamp of the last already known event, which should be skipped. If the specified Timestamp doesn't exist in the Buffer, then the event having the closest higher Timestamp value has to be returned as first. As a direct consequence: evLastTime=0000’0000 means "from the beginning" (no event has to be skipped).

evLastCode (4) the Code of the last already known event, which should be skipped. If the specified code doesn't exist in the Buffer, then the event having the closest higher Timestamp value, relative to evLastTime, has to be returned.

Variables in the Data-Message (A):

nRetEvents (1) the number of Events returned in the reply packet [0..5].

evCode (4) the event code, in numerical format.

evStatus (8) the status of the UPS when the event was generated. Format is specified on the "upsStatus" command. If the UPS is not able to deliver status information, then ffff’ffffh is returned.

evTime (8) the Timestamp, when the event was generated.

3.1.3 New Protocol Commands

UPS Status bits	long	0 to 2^{32}-1	Bit by bit status of the UPS **(1=true)**	3017	r	yes
	mask: bit b0	0000’0001h	Rectifier Mains Input ok			
	bit b1	0000’0002h	Bypass Mains Input ok			
	bit b2	0000’0004h	- (unused. Should be 0)			
	bit b3	0000’0008h	- (unused. Should be 0)			
	bit b4	0000’0010h	Rectifier is on			
	bit b5	0000’0020h	Inverter is on			
	bit b6	0000’0040h	Bypass is on			
	bit b7	0000’0080h	- (unused. Should be 1)			
	bit b8	0000’0100h	Booster is on			
	bit b9	0000’0200h	- (unused. Should be 0)			
	bit b10	0000’0400h	- (unused. Should be 0)			
	bit b11	0000’0800h	- (unused. Should be 0)			
	bit b12	0000’1000h	Battery is charging			
	bit b13	0000’2000h	Battery is discharging			
	bit b14	0000’4000h	Battery is Low (→ introduce shutdown)			
	bit b15	0000’8000h	- (unused. Should be 1)			
	bit b16	0001’0000h	Output Breaker IA2 is open			
	bit b17	0002’0000h	Maintenance Byp. Breaker IA1 is open.[2]			
	bit b18	0004’0000h	Inverter Breaker K2 is open			
	bit b19	0008’0000h	Battery Breaker is open			
	bit b20	0010’0000h	Input Rectifier Breaker is open			
	bit b21	0020’0000h	Input Bypass Breaker is open			
	bit b22	0040’0000h	Load is supplied (Inv, Byp or Maint. Byp.)			
	bit b23	0080’0000h	- (unused. Should be 1)			
	bit b24	0100’0000h	UPS Output is Off (Inv or Bypass)			
	bit b25	0200’0000h	Eco mode is on			
	bit b26	0400’0000h	Energy management mode is on			
	bit b27	0800’0000h	Global Parallel values are returned			
	bit b28	1000’0000h	New alarm (alarm buzzer is on)			
	bit b29	2000’0000h	General Alarm (alarm indication is on)			
	bit b30	4000’0000h	- (unused. Should be 0)			
	bit b31	8000’0000h	- (unused. Should be 1)			
Available Status bits	long	0 to 2^{32}-1	For every bit which is supported by the UPS a 1 is returned, in the same bit-position.	3018	r	yes

[2] Note: when a UPS does not implement the Maintenance Bypass Breaker, then it is important to fix bit17=1

UPS Command	short	0 to 2^{16}-1	Access to the operating mode of the UPS. Execution of commands (write operation). Bitwise parameter. Each bit is assigned to a specific command.. The MSB specify if the command means "on" or "off". By reading the parameter it is possible to check which command is supported by the UPS (1 means that the command is implemented). This protocol command is password-protected with level=5.	806	r/w	yes
	mask: bit b0	0000'0001h	Alarm Reset (Buzzer off)			
	bit b1	0000'0002h	- (unused. Should be 0)			
	bit b2	0000'0004h	- (unused. Should be 0)			
	bit b3	0000'0008h	- (unused. Should be 0)			
	bit b4	0000'0010h	Inverter On/Off			
	bit b5	0000'0020h	Rectifier On/Off			
	bit b6	0000'0040h	Bypass On/Off			
	bit b7	0000'0080h	- (unused. Should be 0)			
	bit b8	0000'0100h	Eco Mode On/Off			
	bit b9	0000'0200h	Energy Management On/Off			
	bit b10	0000'0400h	- (unused. Should be 0)			
	bit b11	0000'0800h	- (unused. Should be 0)			
	bit b12	0000'1000h	Global Parallel values On/Off			
	bit b13	0000'2000h	- (unused. Should be 0)			
	bit b14	0000'4000h	- (unused. Should be 0)			
	bit b15	0000'8000h	Execute On (1) or Off (0)			

Ups All Alarms	string	14	Bit collection of all alarms using 7 bytes, which are coded from hex to ASCII and assigned to the string. Byte 6 appears first in the string.	645	r	yes
	mask: bit b0	Byte 0 bit 0	xid 601 Gen. Alarm 1 ….			
	bit b1	1	xid 602			
	bit b2	2	xid 603			
	bit b3	3	xid 604			
	bit b4	4	xid 605			
	bit b5	5	xid 606			
	bit b6	6	xid 607			
	bit b7	7	Should be 1			
	bit b8	Byte 1 bit 0	xid 608			
	bit b9	1	xid 609			
	bit b10	2	xid 610			
	bit b11	3	xid 611			
	bit b12	4	xid 612			
	bit b13	5	xid 613			
	bit b14	6	xid 614			
	bit b15	7	Should be 1			
	bit b16	Byte 2 bit 0	xid 615			
	bit b17	1	xid 616 ….. 16			
	bit b18	2	xid 617 Temperature			
	bit b19	3	xid 618 Input Bad			
	bit b20	4	xid 619 Output Bad			
	bit b21	5	xid 620 Overload			
	bit b22	6	xid 621 Bypass Bad			
	bit b23	7	Should be 1			
	bit b24	Byte 3 bit 0	xid 622 Output Off			
	bit b25	1	xid 623 UPS Shutdown			
	bit b26	2	xid 624 Charger Failure			
	bit b27	3	xid 625 System Off			
	bit b28	4	xid 626 Fan Failure			
	bit b29	5	xid 627 Fuse Failure			
	bit b30	6	xid 628 General Fault			
	bit b31	7	Should be 1			
	bit b32	Byte 4 bit 0	xid 629 Awaiting Power			
	bit b33	1	xid 630 Shutd. Pending			
	bit b34	2	xid 631 Shutd.Imminent			

	bit b35	3	xid 632 Ambient Temp.			
	bit b36	4	xid 633 Inverter Temp.			
	bit b37	5	xid 634 Bypass Temp.			
	bit b38	6	xid 635 Battery Temp.			
	bit b39	7	Should be 1			
	bit b40	Byte 5 bit 0	xid 636 Inverter Failure			
	bit b41	1	xid 637 Emrgncy Pwr Off			
	bit b42	2	xid 638 Batt. Grnding Err.			
	bit b43	3	xid 639 Batt. Brkr Open			
	bit b44	4	xid 640 Sync. Error			
	bit b45	5	xid 641 Loss of redudancy			
	bit b46	6	xid 3009 Generator			
	bit b47	7	Should be 1			
	bit b48	Byte 6 bit 0	xid 201 bit 0 Battery Condition			
	bit b49	1	xid 201 bit 1 "			
	bit b50	2	xid 202 bit 0 Battery Status			
	bit b51	3	xid 202 bit 1 "			
	bit b52	4	xid 401 bit 0 Output Source			
	bit b53	5	xid 401 bit 1 "			
	bit b54	6	xid 401 bit 2 "			
	bit b55	7	Should be 1			

Get Next Events	string	17 (Get-Message) 1 to 101 (Data-Msg)	Return a sequence of events from the UPS event-buffer in chronological order (oldest firstly). The readout is initiated by a get command which specifies the last event that has already been seen (and consequently has to be skipped). Refer to the specific description	3021	r	yes
	data format for the Get-Message (G): in round brackets the number of chars xid (4), evNReq (1), evLastTime (8), evLastCode (4) data format for the Data-Message (A) evNRet (1), [evCode (4), evStatus (8), evTime (8), [evCode …..]]					
Password	string	0 to 9	Handshake for authorization request. 1. Request (Get) a random value from the device. 2. Retrieve the random value from the Data-Message (A), scramble it and write (Set) the result back to the device, specifying an access-level too. The authorization ends setting the AccessLevel=0 or after a timeout of 30 minutes. Access levels are specified by the manufacturer or by specific protocol commands. RandomValue and Password are 32 bit long and are transmitted as ASCII coded hex values.	3022	r/w	yes
	Data format for the Data message (A): RandomValue (8) Data format for the Set message (X): AccessLevel (1), Password (8)					

Execute UPS Function	string	0 to 117	execute a UPS intrinsic firmware procedure, which is specified by the number Fnum (01..99). The input of the Procedure is specified by the optional parameters in InputValues (comma delimited values) and the result of the procedure, if any, is returned by the parameters in OutputValues (comma delimited values). The Set-Message is used when the specific function delivers no results. This protocol command is password-protected, with level=9. Deviations may apply for single functions.	3023	r/w	yes
	data format for the Get (G) and Set (X) messages: xid (4), Fnum (2), InputValues (0..111) data format for the Data-Message (A): OutputValues (0..117)					
Mixed Measures	string	variable length	A set of measurement values is returned. Each of them is an ASCII coded decimal value, with a variable length. Following xid's are transmitted, delimited by commas, in the given sequential order . Please note that XIDs 408, 414 and 420 are multiplied by 10 205, 206, 309, 315, 321, 410, 416, 422, 406, 412, 418, 408*, 414*, 420*	3024	r	yes
	205, 206: Estimated Minutes, Estimated Charge, 309, 315, 321: Input Voltage A to Neutral, B, C, 410,416,422: Output Voltage A to Neutral, B, C, 406, 412, 418: Output Current A, B, C 408*, 414*, 420*: Output Load A, B, C in units of 0.1% (multiplied by 10)					

Relay Status	string	variable length	Get the status of all the Relays located on the 8142 Interface-	3025	r	yes
	Board. The status is returned as a sequence of 1's and 0's, one for each relay. All inputs follows the "I:" specifier, meanwhile all outputs appear after the "O:" specifier. It is not know in advance how many inputs and outputs are available. Conseuently the length of the string is variable.					
	data format for the **Data**-Message (A): I: bbbbbbb….bb O: bbbbb…bbb NOTES: - each bit **b**, may assume the value 0 or 1 - spaces are included to enhance readability but are not transmitted.					

Configure Board	string	13…101 (Set-Msg)	Configuration parameters for a particular UPS Interface Board, identified by a 1 digit Board Type identificator (0..9).	0170	w	yes
	The number of configuration parameters is passed on the command. The configuration parameters "Param1…ParamN" are separated by commas. The Board Type may be useful when the configuration command is broadcasted.					
	data format for the **Set**-Message (X): xid0170 (4), BoardType (1), npar(2), Param1 [**,** Param2 [**,** …. Param10]] <u>Example</u> for the set-message: "0170 0 02 123456,78" configures the Board Type=0 with two parameters. The first parameter value is 123456, and the second value is 78. NOTE: spaces are included to enhance readability but are not transmitted.					

UPS Slot ID	string	4 (Get-Msg)	Return the physical position of the UPS Module inside the frame. Value in range [1..5]	0171	r	yes

UpsModel2	string	4	Same as xid102, but used by the interface board to discover the modules on the **same** cabinet. Not intended for external usage.	0112	r	yes

Unique UPS ID	string	11 or variable	Return a 7 digit code, which uniquely identifies the UPS (module). It is a 5 digit code assigned sequentially, followed by the 2 digit UPS Address [01..15]. The string may optionally be preceded by any alphanumerical string, terminating with an 'X'	0105	r	yes