

UPS R&D department

Serial protocol: 2400 Baud, No Parity, 8 Data bits, 1 Stop bit

Packet format: STX data length ... data ... Checksum (ASCII 0x02) byte byte byte

The whole packet has a maximum length of 256 bytes.

All the numeric data are little-endian, i.e. are sent LSB first.

STX indicates the start of the packet.

Data length is the number of bytes of data + the check byte.

The "check" byte is computed as the sum modulus 256 of all the bytes in the packet except the STX and the check byte itself.

In case of reception of an invalid checksum or a partial packet, no answer will be supplied.

In case of an invalid command the answer will be:

byte	=	STX	[0x02]
byte	=	data length	[0x04]
byte	=	received command	[]
byte	=	'K'	[0x4B]
byte	=	'0'	[0x6F]
byte	=	Checksum	[]

In order to flush the receiver buffer of UPS, send 255 bytes of NUL (ASCII 0) without expecting any answer.

The serial communication is half duplex: while the UPS is sending an answer to a pending packet, the receiver is disabled.

Applies to...

ID code	Config	family	model
# 1	1	HF Line	HF Line (1 board)
# 1	2	HF Line	HF Line (2 boards)
# 1	3	HF Line	HF Line (3 boards)
# 1	4	HF Line	HF Line (4 boards)
# 2	1	ECO Network	ECO Network 750/1000
# 2	2	ECO Network	ECO Network 1050/1500
# 2	3	ECO Network	ECO Network 1500/2000
# 2	4	ECO Network	ECO Network 1800/2500
# 2	5	ECO Network	ECO Network 2100/3000
# 3	1	ECO 3xx	ECO 308
# 3	2	ECO 3xx	ECO 311
# 4	4	HF Line / 2	HF Line (4 boards) / 2
# 4	5	HF Line / 2	HF Line (5 boards) / 2
# 4	6	HF Line / 2	HF Line (6 boards) / 2
# 4	7	HF Line / 2	HF Line (7 boards) / 2
# 4	8	HF Line / 2	HF Line (8 boards) / 2
# 5	1	HF Millennium	HF 810
# 5	2	HF Millennium	HF 820
# 6	1	HF TOP Line	HF 910
# 6	2	HF TOP Line	HF 920
# 6	3	HF TOP Line	HF 930
# 6	4	HF TOP Line	HF 940
# 7	4	HF TOP Line / 2	HF 940 / 2
# 7	5	HF TOP Line / 2	HF 950 / 2
# 7	6	HF TOP Line / 2	HF 960 / 2
# 7	7	HF TOP Line / 2	HF 970 / 2
# 7	8	HF TOP Line / 2	HF 980 / 2
# 8	1	ECO Network 5xx	ECO 508
# 8	2	ECO Network 5xx	ECO 511
# 8	3	ECO Network 5xx	ECO 516
# 8	4	ECO Network 5xx	ECO 519
# 8	5	ECO Network 5xx	ECO 522
# 9	1	ECO 3xx	ECO 305 / Harviot 530 SX
# 9	2	ECO 3xx	ORDINATORE DUE
# 9	3	ECO 3xx	Harviot 730 SX
# 10	1	ECO Interactive SX	ECO 308 SX / ORDINATORE
# 10	2	ECO Interactive SX	ECO 311 SX
# 11	1	ally HF	ally HF 800
# 11	2	ally HF	ally HF 1600
# 12	1	ally HF	ally HF 1000
# 12	2	ally HF	ally HF 2000
# 13	1	ally HF	ally HF 1250
# 13	2	ally HF	ally HF 2500
# 14	1	Megaline	Megaline 1250
# 14	2	Megaline	Megaline 2500
# 14	3	Megaline	Megaline 3750
# 14	4	Megaline	Megaline 5000
# 15	4	Megaline / 2	Megaline 5000 / 2
# 15	5	Megaline / 2	Megaline 6250 / 2
# 15	6	Megaline / 2	Megaline 7500 / 2
# 15	7	Megaline / 2	Megaline 8750 / 2
# 15	8	Megaline / 2	Megaline 10000 / 2
# 20	1	DHEA	DHEA 1000
# 21	1	DHEA	DHEA 1500

Common commands

# 0	UPS info (read): Request c Answer:	ommand: byte byte byte word byte byte char	= 0 Command (replica of the request command) ID code (family code - see model table) Config (number and/or type of modules - see mo Maximum active power (W) Firmware version Firmware subversion [112] Serial number	del table)
#1	Output data (read Request c Answer:		 = 1 Command (replica of the request command) Active power (W) Voltage (V) Current (Arms*10) Peak current (A*10) 	<pre>{ -1: overrange } { -2: not available } { -1: overrange } { -2: not available } { -1: overrange } { -2: not available } { -1: overrange } { -2: not available }</pre>
# 2	Input data (read): Request c Answer:		= 2 Command (replica of the request command) Active power (W) Voltage (V) Current (Arms*10) Peak current (A*10)	<pre>{ -1: overrange } { -2: not available } { -1: overrange } { -2: not available } { -1: overrange } { -2: not available } { -1: overrange } { -2: not available } }</pre>
#3	UPS status (read Request c Answer:	,	= 3 Command (replica of the request command) Status 0: Running on mains power 1: Running on battery power 2: Battery reserve 3: Bypass engaged 4: Manual bypass engaged Fault 0: All right 1: Overload 2: Overheat 3: Hardware Fault 4: Battery charger failure (overcharging) 5: Replace batteries UPS Temperature in °C + 128	{ i.e T = n-128 °C } { 0: not available }

# 4	Battery data (rea Request c Answer:		= 4 Command (replica of the request of Actual value (V*10) Reserve Threshold (V*10) Exhaust Threshold (V*10)	ommand)		
#5	History data (read Request c Answer:		= 5 Command (replica of the request of UPS Total Run Time (s) Inverter Total Run Time (s) Inverter Interventions Battery Full Discharges Stabiliser or Bypass Interventions Overheatings	ommand)		ot available } ot available }
#6	Scheduling (read Request c Answer:		= 6 Command (replica of the request of Remaining time to shutdown (s) Programmed time to restart (s)	ommand)		o shutdown } 1: no restart }
#7	Event list (most re Request of Answer:	ommand: byte byte byte	<pre>(read): = 7 Command (replica of the request of Event Absolute Counter (roll over ength; 1 byte for each event; the list 1: Odd Turn Off 2: Memory Error! 3: Battery Limit 4: Battery Charger 5: Overload 6: Long Overload 7: Neutral Wrong 8: Neutral Wrong while running 9: Modules Number 10: Programmed battery time ei 11: Programmed reserve time ei 12: Earth Fault 13: Load Waiting 14: HV Bus Runaway 15: Output DC Level 16: Bad Wiring 17: Hardware Fault: Inverter 19: Hardware Fault: Inverter 19: Hardware Fault: Inverter 21: Hardware Fault: Inverter 23: Hardware Fault: Inverter 23: Hardware Fault: 24: Hardware Fault: Inverter 25: Hardware Fault: 26: Hardware Fault: Inverter 27: Hardware Fault: 28: Hardware Fault: Inverter</pre>	at 256) can be em	Booster Booster Booster Booster	Overheat Overheat Overheat Overheat

			29: Hardware Fault: 30: Hardware Fault: Inverter 31: Hardware Fault: 32: Hardware Fault: Inverter 33: Hardware Fault: Battery charg 34: Hardware Fault: Overheat & I 35: Output Plug Removed		Booster Booster Booster Booster narger	Overheat Overheat Overheat Overheat
# 8	Times on battery (re Request com by Answer:	nmand:	= 8			
	b		Command (replica of the request cor Max time on battery (s)			edly limited }
	in	nteger N	Max time after battery reserve (s)	{ 0:	not purpos	ot available } edly limited } ot available }
	b	oyte r	not 0 -> Autorestart after battery dep	leted ena	•	
#9		nmand:	= 9			
	b	byte r	Command (replica of the request cor not 0 -> Neutral sense enabled not 0 -> Ignore while running	nmand)		
# 10	Scheduling (set):					
	lo Io	oyte = ongint F	= 10 Remaining time to shutdown(s) Programmed time to restart(s)			o shutdown } = no restart }
	lo lo	ongint F ongint F	Command (replica of the request cor Remaining time to shutdown (s) Programmed time to restart (s)		{ -1 :	o shutdown } = no restart }
	N.B.: If a sele and reports t		e is longer than the maximum allowed answer.	d the UP	S uses the	maximum
# 11	b	nmand: oyte = oyte r	= 11 not 0 -> Neutral sense enabled not 0 -> Ignore while running			

Answer:

wer.		
	byte	Command (replica of the request command)
	byte	not 0 -> Neutral sense enabled
	byte	not 0 -> Ignore while running

12 Times on battery (set):

Request co	ommand:		
	byte	= 12	
	word	Max time on battery (s)	{ 0 = not purposedly limited }
	word	Max time after battery reserve (s)	{ 0 = not purposedly limited }
	byte	not 0 -> Autorestart after battery deplet	ted enabled
Answer:			
	byte	Command (replica of the request comr	mand)
	word	Max time on battery (s)	{ 0 = not purposedly limited }
		2	{ -2: not available }
	word	Max time after battery reserve (s)	$\{0 = not purposedly limited\}$
		,	{ -2: not available }
	l	wet O Automotion often bettem dealer	4 a d a m a b l a d

byte not 0 -> Autorestart after battery depleted enabled N.B.: If a selected time is longer than the maximum allowed the UPS uses the maximum and reports this in the answer.

13 Buzzer mute (set):

Request command: byte = 13

	Dyte	= 15
	byte	0 -> not muted
		1 -> muted
		2 -> read current status
Answer:		
	byte	Command (replica of the request command)
	byte	0 -> not muted
		1 -> muted
		a dia akta dia wakee wate daa waxee ala waxee a a didia w

N.B.: Buzzer can be disabled only up to the next alarm condition.

14 Battery test (set):

Request command:

byte

- byte = 14
 - 0 -> perform the battery test
 - 1 -> force the UPS to work on battery power
 - 2 -> restore the UPS standard mode (on mains power)
- Answer (at the end of the battery test if it's required):
 - Command (replica of the request command) byte byte
 - 0 -> Generic ok
 - 1 -> Battery charge: 20 %
 - 2 -> Battery charge: 40 %
 - 3 -> Battery charge: 60 %
 - 4 -> Battery charge: 80 %
 - 5 -> Battery charge: 100 %
 - 254 -> Batteries must be replaced
 - 255 -> Test impossible

"ECO" specific commands

19 Stabiliser level (read): Request command: byte = 19 Answer: Command (replica of the request command) byte Stabiliser set level (1..3) byte # 20 Stabiliser level (set): Request command: = 20 byte byte Stabiliser set level (1..3) Answer: Command (replica of the request command) byte Stabiliser set level (1..3) byte N.B.: If the level is outside the allowed range the UPS keeps the previous level.

"HF" specific commands

21 No load threshold (set): Request command: byte = 21 word No load threshold (W) Answer: Command (replica of the request command) byte No load threshold (W) word N.B.: If the threshold is above or below the allowed range the UPS uses the maximum or minimum value respectively. This will be reported in the answer. # 22 No load threshold (read): Request command: byte = 22 Answer: byte Command (replica of the request command) word No load threshold (W) # 23 Output voltage control (set): Request command: byte = 23 byte Output set level (V) byte not 0 -> Output 60 Hz not 0 -> PLL inhibited byte byte not 0 -> Extended PLL lock range Dip detector speed: byte 1 -> fast 2 -> standard 3 -> slow Answer: bvte Command (replica of the request command) byte Output set level (%) (for internal use only) byte not 0 -> Output 60 Hz not 0 -> PLL inhibited byte byte not 0 -> Extended PLL lock range Dip detector speed: byte 1 -> fast 2 -> standard 3 -> slow N.B.: If the level is above or below the allowed range, the UPS uses the maximum or minimum value respectively. This will be reported in the answer. # 24 Output voltage control (read): Request command: byte = 24 Answer:

Command (replica of the request command) byte Output set level (%) (for internal use only) byte not 0 -> Output 60 Hz byte not 0 -> PLL inhibited byte byte not 0 -> Extended PLL lock range byte Dip detector speed: 1 -> fast

2 -> standard

3 -> slow

# 25	Bypass (set):						
		command:					
		byte	= 25				
		byte	not 0 -> Bypass e	enabled			
		byte	not 0 -> Bypass i				
		byte	not 0 -> Off-line r				
		byte	not 0 -> Load Wa	aiting Mode	e enabled		
	Answer:		• • • •				
		byte	Command (replic		equest com	mand)	
		byte	not 0 -> Bypass e				
		byte	not 0 -> Bypass i				
		byte	not 0 -> Off-line r				
	NR·Loo	byte d Waiting N	not 0 -> Load Wa lode, Off-line mode			oon not ho or	ablad if the
	bypass is	-			eu bypass (an not be er	
# 26	Bypass (read):						
	Request	command:	= 26				
	Answer:	byte	= 20				
	Answer.	byte	Command (replic	a of the re	equest com	mand)	
		byte	not 0 -> Bypass e		iquoor oonn	nana)	
		byte	not 0 -> Bypass i				
		byte	not 0 -> Off-line r		peration		
		byte	not 0 -> Load Wa	aiting Mod	e enabled		
# 07			al\.				
# 27	Hardware status		d):				
	Request	command: byte	= 27				
	Answer:	byte	- 21				
	Answer.	byte	Command (replic	a of the re	equest com	mand)	
		shortint	Number of faulty		94666 66111		2 = not available }
		byte	Hardware Fault:			· · ·	
		,	0: Unknown o	r none			
			1: Inverter				
			2:	P.F.C.			
			3: Inverter	P.F.C.			
			4:		Booster		
			5: Inverter		Booster		
			6:	P.F.C.	Booster		
			7: Inverter	P.F.C.	Booster	Overheat	
			8: 0: Invertor			Overheat Overheat	
			9: Inverter 10:	P.F.C.		Overheat	
			11: Inverter	P.F.C.		Overheat	
			12:	1.1.0.	Booster	Overheat	
			13: Inverter		Booster	Overheat	
			14:	P.F.C.	Booster	Overheat	
			15: Inverter	P.F.C.	Booster	Overheat	
			16: Battery ch	arger			
			17: Overheat	& Battery	charger		
# 30	A/D average valu	les (read).					
00							

Request command: byte Answer:

= 30 byte byte byte byte byte Command (replica of the request command) Vin A/D lin A/D Vout A/D

Bus voltages (read): # 31

Request command:

byte

= 31

Answer:

Command (replica of the request command) byte word Positive bus voltage Negative bus voltage (absolute value) word

32 Battery thresholds (set):

Request	command:		
-	byte	= 32	
	word	Reserve Threshold (V*10)	{ 0 = automatic }
	word	Exhaust Threshold (V*10)	{ 0 = automatic }
	byte	not 0 -> Perform an automatic battery test after turr power present	on with mains
Answer:			
	byte word	Command (replica of the request command) Reserve Threshold (V*10)	{ 0 = automatic }

byte	Command (replica of the request command)	
word	Reserve Threshold (V*10)	{ 0 = automatic }
word	Exhaust Threshold (V*10)	{ 0 = automatic }
byte	not 0 -> Perform an automatic battery test af	ter turn on with mains
	power present	

N.B.: If a threshold is above or below the allowed range the UPS uses the maximum or minimum value respectively.

If Reserve Threshold < Exhaust Threshold then Reserve Threshold = Exhaust Threshold. This will be reported in the answer.

Exception: selecting a level of 0 V for at least one of the thresholds means using for both the default levels that are load dependent (automatic mode).

33 Battery thresholds mode (HF only) (read):

Request c	command:	
	byte	= 33
Answer:		
	byte	Command (replica of the request command)
	byte	not 0 -> Automatic battery thresholds
	byte	not 0 -> Perform an automatic battery test after turn on with mains
	-	power present

"ALLY HF" specific commands

# 21	No load threshold same as		ic command (# 21)
# 22	No load threshold same as		ic command (# 22)
# 23	Output voltage co same as	· · ·	ic command (# 23)
# 24	Output voltage co same as): ic command (# 24)
# 25	Bypass (set): same as	HF specifi	ic command (# 25)
# 26	Bypass (read): same as	HF specifi	ic command (# 26)
# 27	Hardware status same as		d): i c command (# 27)
# 30	A/D average valu same as		ic command (# 30)
# 31	Bus voltages (rea same as		ic command (# 31)
# 32	Battery threshold same as		ic command (# 32)
# 33	Battery threshold same as		ad): i c command (# 33)
# 36	Line frequency a Request c Answer:		set): = 36 0 -> Line frequency autoselect disabled 1 -> Line frequency autoselect enabled 2 -> Read line frequency autoselect
		byte byte	Command (replica of the request command) 0 -> Line frequency autoselect disabled 1 -> Line frequency autoselect enabled

"MEGALINE" specific commands

# 21	No load threshold (set): same as HF specifi	ic command (# 21)		
# 22	No load threshold (read): same as HF specifi	ic command (# 22)		
# 23	Output voltage control (set): same as HF specifi			
# 24	Output voltage control (read same as HF specifi			
# 25	Bypass (set): same as HF specifi	ic command (# 25)		
# 26	Bypass (read): same as HF specifi	ic command (# 26)		
# 27	Hardware status report (rea same as HF specifi			
# 31	Bus voltages (read): same as HF specifi	ic command (# 31)		
# 32	Battery thresholds (set): same as HF specifi	ic command (# 32)		
# 33	Battery thresholds mode (re same as HF specifi			
# 36	Line frequency autoselect (set): same as ally HF specific command (# 36)			
# 37	Battery state of charge (read	d):		
	Request command: byte	= 37		
	Answer:			
	byte	Command (replica of the request command) 0 -> Data are valid		
	byte	1 -> Data are valid 1 -> Data are not valid (battery discharge learning required)		
	word	255 -> Impossible (classic management is selected)		
	word byte	Remaining time to battery exhaust (s) Battery remaining charge (%)		
# 38	Batteny discharge learning (sot).		
# 30	Battery discharge learning (Request command:	sel).		
	byte	= 38		
	byte	0 -> Start battery discharge learning 1 -> Read battery discharge learning status		
		2 -> Abort battery discharge learning		
	Answer: byte	Command (replica of the request command)		
	byte	Battery discharge learning status		
		0: Running 1: Not running		
		2: Ended		
		255: Impossible (UPS on battery power or classic management is selected)		

# 39	Battery autonomy		ent (read / set):
	Request co		22
		byte	= 39
		byte	0 -> Set classic management
			1 -> Set S.O.C. management 2 -> Read actual status
	Answer:		2 -> Neau actual status
	/ (10/001.	byte	Command (replica of the request command)
		byte	Actual battery autonomy management
		,	0: Classic management
			1: S.O.C. management
	Detter see en se tie		
# 40	Battery reserve time (set): Request command:		
	Request co		= 40
		byte word	Reserve time (m)
	Answer:	woru	
		byte	Command (replica of the request command)
		word	Actual reserve time (m)
	D <i>u u</i>	<i>(</i> 1)	
# 41	Battery reserve time (read): Request command:		
	Request of	byte	= 41
	Answer:	Dyte	- +1
	/	byte	Command (replica of the request command)
		word	Actual reserve time (m)
ш л л		ntral (ant).	
# 44	Output voltage co		
	Request co	byte	= 44
		byte	Output set level (V)
		byte	not 0 -> Output 60 Hz
		byte	not 0 -> PLL inhibited
		byte	not 0 -> Extended PLL lock range
		byte	Dip detector speed:
		.,	1 -> fast
			2 -> standard
			3 -> slow
	Answer:		
		byte	Command (replica of the request command)
		byte	Output set level (V)
		byte	not 0 -> Output 60 Hz
		byte	not 0 -> PLL inhibited
		byte	not 0 -> Extended PLL lock range
		byte	Dip detector speed:
			1 -> fast 2 -> standard
			2 -> standard 3 -> slow
	N.B.: If the	level is ab	ove or below the allowed range, the UPS uses the maxi
		- I	attach.

N.B.: If the level is above or below the allowed range, the UPS uses the maximum or minimum value respectively. This will be reported in the answer.

Output voltage control (read): # 45 Request command: byte = 45 Answer: Command (replica of the request command) byte byte Output set level (V) not 0 -> Output 60 Hz byte not 0 -> PLL inhibited byte byte not 0 -> Extended PLL lock range byte Dip detector speed: 1 -> fast 2 -> standard 3 -> slow # 46 Single event read (read): Request command: byte = 46= 0 -> Reset (answer will contain most recent event) byte = 1 -> Autoincrement Answer: byte Command (replica of the request command) byte Time Stamp - Year (BCD packed) Time Stamp - Month (BCD packed) byte byte Time Stamp - Day (BCD packed) byte Time Stamp - Hour (BCD packed) byte Time Stamp - Min (BCD packed) byte Event Code - LSB 0 = Regular Turn Off 1 = Irregular turn Off 2 = Memory Error! 3 = Low Battery Turn Off 4 = Battery Runaway 5 = Overload Turn Off 6 = Long Overload 7 = Neutral Wrong Turn Off 8 = Neutral Wrong while running 9 = Modules Number Error 10 = Programmed battery time expired 11 = Programmed reserve time expired 12 = Earth Fault Turn Off 13 = Load Waiting Turn Off 14 = HV Bus Runaway 15 = Output DC Level 16 = Bad Wiring 17 = Hardware Fault: Unknown 18 = Hardware Fault: Inverter 19 = Hardware Fault: P.F.C. 20 = Hardware Fault: Inverter P.F.C. 21 = Hardware Fault: Booster 22 = Hardware Fault: Inverter **Booster** 23 = Hardware Fault: P.F.C. **Booster** 24 = Hardware Fault: Inverter P.F.C. **Booster** 25 = Hardware Fault: Overheat 26 = Hardware Fault: Inverter Overheat 27 = Hardware Fault: P.F.C. Overheat 28 = Hardware Fault: Inverter P.F.C. Overheat 29 = Hardware Fault: Booster Overheat 30 = Hardware Fault: Overheat Inverter Booster 31 = Hardware Fault: P.F.C. Booster Overheat 32 = Hardware Fault: P.F.C. Inverter Booster Overheat 33 = Hardware Fault: Battery charger 34 = Hardware Fault: **Overheat & Battery charger** 35 = Output Plug Removed

36 = Mains Power: voltage very high

			 37 = Watchdog reset 38 = Battery Startup failed 39 = Battery Calibration aborted by 40 = Mains Power: SAG 41 = Mains Power: SWELL 42 = Mains Power: BROWNOUT 43 = Mains Power: SPIKE 	user
			44 = Mains Power: Harmonic distor45 = Neutral Voltage: SWELL	tion
		h. da	46 = Neutral Voltage: SPIKE	
		byte byte	Event Code - extended byte 1 Event Code - extended byte 2	
		byte	Event Memory Position	
# 48	Date/Time (set):			
	Request c			
		byte	= 48	
		byte	Year (BCD packed)	
		byte byte	Month (BCD packed) Day (BCD packed)	
		byte	Hour (BCD packed)	
		byte	Min (BCD packed)	
		byte	Sec (BCD packed)	
	A	byte	Day of Week	{ 0 = Sunday; 1 = Monday; }
	Answer:	byte	Command (replica of the request cor	mmand)
		byte	Year (BCD packed)	linana)
		byte	Month (BCD packed)	
		byte	Day (BCD packed)	
		byte	Hour (BCD packed)	
		byte	Min (BCD packed)	
		byte	Sec (BCD packed)	(0 - Sunday: 1 - Manday:)
		byte	Day of Week	{ 0 = Sunday; 1 = Monday; }
# 49	Date/Time (read):	:		
# 49	Date/Time (read): Request co			
# 49	Request co		= 49	
# 49	· · ·	ommand: byte		
# 49	Request co	ommand: byte byte	Command (replica of the request cor	nmand)
# 49	Request co	ommand: byte byte byte	Command (replica of the request cor Year (BCD packed)	nmand)
# 49	Request co	ommand: byte byte	Command (replica of the request cor	nmand)
# 49	Request co	ommand: byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed)	nmand)
# 49	Request co	ommand: byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed)	nmand)
# 49	Request co	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed)	
# 49	Request co	ommand: byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed)	mmand) { 0 = Sunday; 1 = Monday; }
# 49 # 52	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed)	
	Request co	byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week	
	Request co Answer: Calendar schedul	byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week	{ 0 = Sunday; 1 = Monday; }
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimension)	{ 0 = Sunday; 1 = Monday; } sion-1)
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimension Month (BCD packed)	{ 0 = Sunday; 1 = Monday; … } sion-1) { 0xAA = wildcard }
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimension Month (BCD packed) Day (BCD packed)	{ 0 = Sunday; 1 = Monday; } sion-1)
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimens Month (BCD packed) Day (BCD packed) Day of Week { 0 = Sunday; 1 = Hour (BCD packed)	{ 0 = Sunday; 1 = Monday; … } sion-1) { 0xAA = wildcard } { 0xAA = wildcard }
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimens Month (BCD packed) Day (BCD packed) Day of Week { 0 = Sunday; 1 = Hour (BCD packed) Min (BCD packed)	{ 0 = Sunday; 1 = Monday; … } sion-1) { 0xAA = wildcard } { 0xAA = wildcard }
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimens Month (BCD packed) Day (BCD packed) Day of Week { 0 = Sunday; 1 = Hour (BCD packed) Min (BCD packed) Min (BCD packed) Action	{ 0 = Sunday; 1 = Monday; … } sion-1) { 0xAA = wildcard } { 0xAA = wildcard }
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimens Month (BCD packed) Day (BCD packed) Day (BCD packed) Day of Week { 0 = Sunday; 1 = Hour (BCD packed) Min (BCD packed) Min (BCD packed) Action 0: No Action – Delete Step 1: Turn Off	{ 0 = Sunday; 1 = Monday; … } sion-1) { 0xAA = wildcard } { 0xAA = wildcard }
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimens Month (BCD packed) Day (BCD packed) Day of Week { 0 = Sunday; 1 = Hour (BCD packed) Min (BCD packed) Min (BCD packed) Action 0: No Action – Delete Step 1: Turn Off 2: Turn On	{ 0 = Sunday; 1 = Monday; … } sion-1) { 0xAA = wildcard } { 0xAA = wildcard }
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimens Month (BCD packed) Day (BCD packed) Day (BCD packed) Day of Week { 0 = Sunday; 1 = Hour (BCD packed) Min (BCD packed) Min (BCD packed) Action 0: No Action – Delete Step 1: Turn Off	{ 0 = Sunday; 1 = Monday; … } sion-1) { 0xAA = wildcard } { 0xAA = wildcard }
	Request co Answer: Calendar schedul	ommand: byte byte byte byte byte byte byte byte	Command (replica of the request cor Year (BCD packed) Month (BCD packed) Day (BCD packed) Hour (BCD packed) Min (BCD packed) Sec (BCD packed) Day of Week = 52 Step index (0step_memory_dimens Month (BCD packed) Day of Week {0 = Sunday; 1 = Hour (BCD packed) Min (BCD packed) Min (BCD packed) Min (BCD packed) Action 0: No Action – Delete Step 1: Turn Off 2: Turn On 3: Battery Calibration	{ 0 = Sunday; 1 = Monday; } sion-1) { 0xAA = wildcard } { 0xAA = wildcard } = Monday; ; 0xAA = wildcard }

byte Step index (0..step_memory_dimension-1) byte Month (BCD packed) { 0xAA = wildcard } byte Day (BCD packed) { 0xAA = wildcard } { 0 = Sunday; 1 = Monday; ...; 0xAA = wildcard } byte Day of Week byte Hour (BCD packed) byte Min (BCD packed) byte Action 0: No Action 1: Turn Off 2: Turn On 3: Battery Calibration 4: Battery Test 255: Step index out of range # 53 Calendar scheduling (read): Request command: byte = 53 byte Step index (0..step_memory_dimension-1) Answer: byte Command (replica of the request command) byte Step index byte Month (BCD packed) $\{ 0xAA = wildcard \}$ byte Day (BCD packed) { 0xAA = wildcard } byte Day of Week { 0 = Sunday; 1 = Monday; ...; 0xAA = wildcard } byte Hour (BCD packed) byte Min (BCD packed) byte Action 0: No Action 1: Turn Off 2: Turn On 3: Battery Calibration 4: Battery Test 255: Step index out of range # 54 Calendar scheduling enable (read / set): Request command: = 54 byte 0 -> Disable byte 1 -> Enable 2 -> Read actual status Answer: byte Command (replica of the request command) 0 -> Disabled byte 1 -> Enabled

DHEA specific commands

# 21	No load threshold (set): same as HF specific command (#21)
# 22	No load threshold (read): same as HF specific command (# 22)
# 23	Output voltage control (set): same as HF specific command (#23)
# 24	Output voltage control (read): same as HF specific command (#24)
# 25	Bypass (set): same as HF specific command (# 25)
# 26	Bypass (read): same as HF specific command (# 26)
# 27	Hardware status report (read): same as HF specific command (#27)
# 31	Bus voltages (read): same as HF specific command (# 31)
# 32	Battery thresholds (set): same as HF specific command (# 32)
# 33	Battery thresholds mode (read): same as HF specific command (# 33)
# 36	Line frequency autoselect (set): same as ally HF specific command (# 36)
# 37	Battery state of charge (read): same as ally HF specific command (# 37)
# 38	Battery discharge learning (set): same as ally HF specific command (# 38)
# 39	Battery autonomy management (read / set): same as ally HF specific command (# 39)
# 40	Battery reserve time (set): same as ALLY HF specific command (# 40)
# 41	Battery reserve time (read): same as ALLY HF specific command (# 41)
# 44	Output voltage control (set): same as MEGALINE specific command (#44)
# 45	Output voltage control (read): same as MEGALINE specific command (#45)
# 46	Single event read (read): same as MEGALINE specific command (#46)
# 48	Date/Time (set):

same as MEGALINE specific command (#48)

# 49	Date/Time (read): same as	MEGALINI	E specific command (# 49)	
# 52	Calendar scheduling (set): same as MEGALINE specific command (# 52)			
# 53	Calendar scheduling (read): same as MEGALINE specific command (# 53)			
# 54	Calendar scheduling enable (read/set): same as MEGALINE specific command (# 54)			
# 55	Auxiliary output (s Request co Answer:		= 55 AUX OUT 1 max time on battery (m) AUX OUT 2 max time on battery (m) AUX OUT 1 OFF on battery reserve {1 => OFF on battery reserve} AUX OUT 2 OFF on battery reserve {1 => OFF on battery reserve} Command (replica of the request command) AUX OUT 1 max time on battery (m) AUX OUT 1 max time on battery (m) AUX OUT 2 max time on battery (m) AUX OUT 1 OFF on battery reserve AUX OUT 2 OFF on battery reserve	
# 56	Auxiliary output (re Request co Answer:	,	= 56 Command (replica of the request command) AUX OUT 1 max time on battery (m) AUX OUT 2 max time on battery (m) AUX OUT 1 OFF on battery reserve AUX OUT 2 OFF on battery reserve	

-