DPM86xx Series Power Supply
Simple Communication Protocol

Autor and Version

Alois Hockenschlohe, Version 221103

Introduction and overview

The power supplies of the DPM86xx series provide two communication protocols: the classical modbus protocol and a proprietary protocol called “Simple Communication Protocol”. This summary describes the “Simple Communication Protocol” as far as it is possible for the author. The information given here is based on various documents available in the internet and on the experiences of the author. Corrections and improvements are welcome. This document is “work in progress” - please watch out for the latest version. Don’t rely too much on any of the information presented in this document. You have been warned. :-)

Many thanks goes to MSe, which helped me to improve this document and did some final work.

The “Simple Communication Protocol”

The “Simple Communication Protocol” is a straight forward, point-to-point, client-server, command-response protocol based on a serial connection. It is used to write and read numerical values between 0 and 65535 from/to the power supply.

Various tests have shown that the protocol has been implemented rather poorly by the manufacturer. The power supply lacks of appropriate safety mechanisms such as range checks and sanity checks of the passed parameters.

Frames

Four different frames are known:

	command (client → power supply)	response (power supply → client)
write a value	write command	result

	command (client → power supply)	response (power supply → client)
read a value	read command	value

Write command frame

start symbol	address	access	function	field separator	value	feedback	end sequence
“:”	01..99	“w”	00..99	“=”	0..65535	“,” or “.” or “,,”	”\n”

Field name	Comment/description
start symbol	Each command starts with a “:”.
address	Multiple power supplies can be controlled from a single instance. For this reason each power supply has a assigned address between 01 (default) and 99, which can be configured via its physical interface. A command will be processed by a power supply if this field has the same value as its address, otherwise it will be silently ignored.
access	Access “w” indicates a write access.
function	A power supply provides various state information and accepts various commands via so-called functions. A function is addressed via a function number, which is a 2-digit number between 00 and 99. For each read command addressing a valid function number, the power supply provides a response. Addressing a function above 99 will be silently ignored. Only a subset of the functions has been documented by the manufacturer. More detailed information can be found in the section “Functions”.
field separator	The character “=” separates the field “function” from the field “value”.
value	The provided value will be handed over to the function specified in the “function” field. Various functions expect various data types. More detailed information can be found in the section “Data Types”.
feedback	The “feedback” field indicates if the power supply should response immediately (“.” or “,,”) or if the feedback should be queued (“,”). The queued responses will be flushed if the feedback field has the value “.” or “,,”.
end sequence	The official documentation states: “Each command is terminated by the string ‘\r\n’ (this is actually a return character followed by a newline character in ASCII, hexadecimal representation is 0x0d, 0x0a).” In contradiction to this statement it seems as if a command should be terminated just by a single ‘\n’ (0x0a), because the end documented sequence ‘\r\n’ caused delays and errors.

Result frame

start symbol	address	status	last reply	end sequence
“:”	01...99	“ok”	“.”	”\n”

field name	comment/description
start symbol	See section “Write command format”.
address	See section “Write command format”.
status	“ok” indicates that the incoming write command passed some very basic checks and has been processed by the power supply. It does not (!) indicate that the command was successful in any way.
last reply	Responses can be queued by the power supply until they are requested by the corresponding value in the field “feedback” of a command frame. Multiple responses are delivered in a single response. The responses are separated from each other by the end code. The last response is marked by an additional “.”.
end sequence	See section “Write command format”.

Read command frame

start symbol	address	access	function	field separator	nr. of additional functions to read	feedback	end sequence
“:”	01...99	“r”	00...99	“=”	0...99	“,” or “.” or “,,”	”\n”

field name	comment/description
start symbol	See section “Write command format”.
address	See section “Write command format”.
access	Access “r” indicates a read access.
function	See section “Write command format”.
field separator	See section “Write command format”.
nr. of additional functions to read	One or more consecutive functions can be read with a single read command. The field “additional functions to read” indicates how many additional functions will be read. The value ‘00’ indicates that only the function ‘function number’ will be read. A value of 01 and above will read the function ‘function number’ and the specified number of consecutive functions.
feedback	See section “Write command format”.
end sequence	See section “Write command format”.

Value frame

start symbol	address	access	function	field separator	value	last reply	end sequence
“:”	01...99	“r”	00...99	“=”	0...65335	“.”	”\n”

field name	comment/description
start symbol	See section “Write command format”.
address	See section “Write command format”.
access	Access “r” indicates a response to a read command.
function	The function number of the corresponding read command.
field separator	See section “Write command format”.
value	Response to the read command. See section “Data Types”.
last reply	See section “Status frame”.
end sequence	See section “Status frame”.

Data Types

The field “value” (found in the write and the response frame) transports various data types from and to the power supply.

The values handed over are often processed by the power supply without proper range and sanity checks. This leads to meaningless and/or random response frames to valid requests.

This behaviour stays on until the power supply is rebooted (power off, power on after a few seconds of waiting period). In rare cases the power supply even had to be restored to factory defaults.

A program which uses the Simple Communication Protocol should implement strict sanity checks on any parameter handed over and monitor the behaviour of the power supply closely.

data type	description	range	example
voltage	Voltage (Volt). Four-digit integer, first two digits positions interpreted as before, last two digits interpreted as position after the decimal point.	0...6000 (0..60 V)	1234 → 12.34 V
current	Current (Ampere). Four-digit integer, first digit interpreted as the position before, last three digits interpreted as the positions after the decimal point	DPM-8605: 0..5000 (0..5A) DPM-8608: 0..8000 (0..8A) DPM-8616: 0..16000 (0..16A) DPM-8624: 0..24000 (0..24A) Hint: Maximum value can be read via function 1.	1234 → 1.234 A
boolean	Boolean (true/false)	0 / 00 or 1 / 01	0 → off
temperature	Temperature (Celsius). Two-digit-integer.	0..100 (?)	23 → 23° C
slot	Address of memory slots	1..12	5

Functions

Documented Functions

function	R	W	description	data type	comment/hint
00	X		Read the max output voltage (V). Will always be 6000 (60V).	voltage	6000 → 60V
01	X		Read the max output current (A).	current	5000 → 5A → DPM-8605, 8000 → 8A → DPM-8608 16000 → 16A → DPM-8616, 24000 → 24A → DPM 8624
10	X	X	Read/set output voltage target (V).	voltage	1234 → 12,34 V
11	X	X	Read/set output current target (A).	current	1234 → 1,234 A
12	X	X	Read/set output status.	boolean	0 / 00 → off, 1 / 01 → on
20		X	Set output voltage and current target (V,A).	voltage,current	“1234,5678” → 12,34 V and 5,678 A
30	X		Read current output voltage (V).	voltage	1234 → 12,34 V
31	X		Read current output current (A).	current	1234 → 1,234 A
32	X	X	Read/set constant mode (current or voltage).	boolean	0 → constant voltage, 1 → constant current
33	X		Read the temperature of the power supply (C).	temperature	23 → 23° C

Undocumented but (partly) known functions

The following functions are not part of the official documentation from the manufacturer:

function	R	W	description	data type	comment/hint
21		X	This function provides four different features: • Save the actual configured current and voltage to a slot. • Set the defined current and voltage as the lower or upper limit. • Remove both limits (upper and lower limit).	slot	Slot 1..9: Save current and voltage to the slot Slot 10: Set upper limit. Slot 11: Set lower limit. Slot 12: Remove upper and lower limit.
22		X	Set current and voltage from a memory slot.	slot	Slot 1..9: index of the slot
40	X		Only a little is known about this function: • Can be set via the SET menu between settings “8-CH” and “L-Ui”.	voltage (?)	
41	X		Only a little is known about this function: • Can be set via the SET menu (L-Ui). • Can not be manipulated via a write frame. • Seems to be something related to voltage and function 42. • Initial value is “0”.	voltage (?)	
42	X		Only a little is known about this function: • Can be set via the SET menu (H-Ui). • Can not be manipulated via a write frame. • Seems to be something related to voltage and function 41. • Initial value is “6000”	voltage (?)	

Unknown (or unused) functions

Only ten out of 100 function numbers are documented by the manufacturer. The purpose of four more functions are (partly) known at the moment. A read scan performed against all 100 function numbers brings up the following picture:

function	documented	undocumented		
		(partly) known functions	nothing known yet	
			read commandreturns a “0”	read command returns other values than “0”
00-01	X			
02				X (static: 8)
03				X (static: 11)
04-09			X	
10-12	X			
13-19			X	
20	X			
21-22		X		
23-29			X	
30-33	X			
34-37				X (minor fluctuations)
38-40			X	
40		X		
41-42		X		
43-49			X	
50				X (static: 1528)
51				X (static: 4476)
52				X (static: 1648)
53				X (static: 4233)
54				X (static: 2101)
55				X (static: 3920)
56				X (static: 2251)
57				X (static: 4931)
58-70			X	
71-72				X
73-99			X	
	10	4	70	16