Modbus Actions and Status Channels on Modbus2PAK-Server

Modbus Channel	Type	Command/ Status	Description
1	Action 1)	1	Start test (changes status from 0 to 1)
		2	Stop test (changes status from any value to 0)
		3	*Not used*
		4	Start recording (changes status from 1 to 3)
		5	Stop recording (changes status from 3 to 1)
		6	*Not used*
		7	*Not used*
		8	*Not used*
		9	*Not used*
2	Status	0	Stopped (test not running)
		1	Started (test running, no recording)
		2	*Not used*
		3	Recording (test running, recording)
		99	*Not used*
		100	Error (for details, see Modbus register 10)
3	Status	0..1..0..1	Heartbeat (toggles each second) The heartbeat only beats when the Modbus2PAK-Server has control on the main process it must control (the PAK 5.8 software)
4	Status	number	*Not used*
5	Status	number	ID of measurement (value corresponds to the counter value in measurement definition window of PAK5.8 as used with the variable $(INC) for the naming convention of datasets; number of digits must be preconfigured in PAK5.8 – e.g. to allow a counting from 000 to 999)
6	Status	number	Remaining free space on the disk used to store measurements (in MB; an estimated value is indicated)
7	Status	number	Measurement duration in seconds
8	Input	[xxxxx]	Test No. as provided from external system (5 digits)
9	Input	[xx]	Project No. as provided from external system (2 digits)
10	Status	0	No error
		1	***Not used***
		2	***Not used***
		3	Error: Main process not responding. Restart of process necessary. Remark: At the same time the heart beat stopps due to the lost control over the main process.
		other	Internal error code

Remarks:

1) The **channel** value will be reset to zero immediately after it is accepted by the Modbus2PAK-Server. Attention: A reset of this channel to zero is <u>not</u> an indication for the completeness of the processing of the command – the **status** channel no. 2 must be used for this!!
   If no reset of the channel value occurs, the desired command is not known or can not be

processed in the current status. No error will be indicated in this case, but the command is silently ignored.

2) Modbus **channel** as numbered above utilize two Modbus holding registers per channel. To address the correct holding registers, use this formula: n = (2*x-1), with x being the channel number, and n being the first of the two registers.

3) **Our Modbus "slave" (=server) implementation utilizes Big Endianness when storing data. Data type is Floating Point**.