DPH5005 CNC Power Communication Protocol V1.2

Communication protocol overview:

The communication protocol for this device is specifically MODBUS RTU. This can be used via RS232, RS485 or BT serial interface.

Note, This product only supports MODBUS RTU Function Codes 0x03 (03H), 0x06 (06H), 0x10 (10H)

II. Communication protocol described:
Message transmission is asynchronous, Modbus-RTU is in 11-bit bytes.

(For a full description of the general MODBUS RTU Protocol see http://modbus.org/docs/PI_MBUS_300.pdf)
Modbus is a generic protocol. The implementation of the location and function of registers is specific to the designer of this device however the format of the command structure for the Modbus Communication is fixed and as detailed in the above referenced document. Below is a brief introduction to the Standard Modbus RTU protocol and the registers and Modbus RTU Function Codes used in this specific device.

Data Format:

Serial data format	10 (binary system)
Start bit	1
Data bits	8
Parity bit	No
Stop bit	1

Data frame structure:

Data-frame interval	Address code	Function code	Data area	CRC Check
3.5 Bytes more	1 Bytes	1 Bytes	N Bytes	2 Bytes

Before sending data, the bus silent period (no data transmition time) is more than 3.5 character, (eg: Baud rate is 9600, the time is 5ms).
RTU messages start with a silent interval of at least 3.5 character time periods.Following the last transmitted character, a similar interval of at least 3.5 character times marks the end of the message. A new message can begin after this interval.
The entire message frame must be transmitted as a continuous stream. If a silent interval of more than 3.5 character times occurs before completion of the frame, the receiving device flushes the incomplete message and assumes that the next byte will be the address field of a new message.
Similarly, if a new message begins earlier than 3.5 character times following a previous message, then the receiving device will consider it to be a continuation of the previous message. This will set an error, as the value in the final CRC field will not be valid for the combined messages.

1.1 Address Field:

The address field of a message frame contains eight bits, from 1 to 255. This byte shows the slaver of setting address by user will receive the message form main. Every slaver must have unique address field, and only the salver fits the address field, it can respond loopback message. When slaver send the message back, the loopback data will begin with its own address filed. The address field sent by the main is the address filed the salver received, and the loopback address filed is the loopback slaver filed address. The corresponding address filed show where it come from.

1.2 Function Field:
The Function Code field tells the addressed slave what function to perform. This device only supports the functions 0x03 0x06 0x10.

Function code	Define	Operations (binary)
0x03	Read Holding Register	Read one or more registers in the slave.
0x06	Write single register	Writes binary data to a single register.
0x10	Write multiple registers	Writes values into a sequence of holding registers.

1.3 Data area:

The data bytes contain any additional information that the slave will need to perform the function. For example, function code 03 will request the slave to read holding registers and respond with their contents. The data field must contain the information telling the slave which register to start at and how many registers to read.

In the Response, the data bytes contain the data collected from the slave, such as register values or status. If an error occurs, the function code is modified to indicate that the response is an error response, and the data bytes contain a code that describes the error.

The 3 MODBUS RTU Function Codes used by this device: (ONLY these three codes)

0x03 Read Holding Register(s):

Address code	Function code	Register starting address	Number of register addresses n (1 ~ 32)	CRC Check
1 Bytes	1 Bytes	2 Bytes	2 Bytes	2 Bytes

0x03 Read Holding Register(s) function returns the following format.

Address code	Function code	Returns the number of bytes 2*n	Register data	CRC Check
1 Bytes	1 Bytes	1 Bytes	2*n Bytes	2 Bytes

0x06 Write Single Register.

Address code	Function code	Register address	Register data	CRC Check
1 Bytes	1 Bytes	2 Bytes	2 Bytes	2 Bytes

0x06 Write Single Register function returns the following format.

Address code	Function code	Register address	Register data	CRC Check
1 Bytes	1 Bytes	2 Bytes	2 Bytes	2 Bytes

0x10 Write Multiple Registers.

Address code	Function code	Register starting address	Number of register addresses n (1 ~ 32)	Number of bytes written 2*n	Register data	CRC Check code
1 Bytes	1 Bytes	2 Bytes	2 Bytes	1 Bytes	2*n Bytes	2 Bytes

0x10 Function: Returns the following format.

Address code	Function code	Register starting address	Number of register address n (1 ~ 32)	CRC Check code
1 Bytes	1 Bytes	2 Bytes	2 Bytes	2 Bytes

Register Map for this device.

Function	Description	Number of bytes	Decimal places	UNIT	Read/Write	Register address
U-SET	Voltage setting	2	2	V	R/W	0000H
I-SET	Current setting	2	3	A	R/W	0001H
UOUT	Output voltage display value	2	2	V	R	0002H
IOUT	Output current display value	2	2	A	R	0003H
POWER	Output power display value	2	1 or 2	W	R	0004H
UIN	Input voltage display value	2	2	V	R	0005H
LOCK	Key lock	2	0	-	R/W	0006H
PROTECT	Protection status	2	0	-	R	0007H
CV/CC	Constant Voltage / Constant Current status	2	0	-	R	0008H
ONOFF	Switch output state	2	0	-	R/W	0009H
B_LED	Backlight brightness level	2	0	-	R/W	000AH
MODEL	Product model	2	0	-	R	000BH
VERSON	Firmware Version	2	0	-	R	000CH

EXTRACT_M	Shortcut to bring up the required data set	2	0	-	R/W	0023H
U-SET	Voltage settings	2	2	V	R/W	0050H
I-SET	Current setting	2	3	A	R/W	0051H
S-OVP	Over-voltage protection value	2	2	V	R/W	0052H
S-OCP	Over-current protection value	2	3	A	R/W	0053H
S-OPP	Over power protection	2	1	W	R/W	0054H
B-LED	Backlight brightness levels	2	0	-	R/W	0055H
M-PRE	Memory Preset Number	2	0	-	R/W	0056H
S-INI	Power output switch	2	2	-	R/W	0057H

NOTE :

1. This product has user programmable 'Data Groups' which each store preset user values. The groups are named M0-M9 giving a total of 10 'Data Groups'. Each data group has a subset of 8 parameters at number 10-17.

And The 'M0' Data Group is the power-on default data group. M1 and M2 is the shortcut bring up data group. M3-M9 are ordinary storage data groups. Each Data Group (M0-M9) start address is calculated by the following formula: 0050H + (Data group number * 0010H).

For example, M3 Data Group has the starting address as follows: 0050H + (3 * 0010H) = 0080H.

2. Key lock function reading and writing value are 0 and 1, 0 represents not lock, 1 represents lock

3, Protection status reading value are 0-3, 0 represents good running, 1 represents OVP, 2 represents OCP, 3 represents OPP.

4, Constant voltage and constant current reading value are 0-1, 0 represents CV, 1 represents CV.

5, Control output function reading and writing value are 0-1, 0 represents close, 1 represents open.

6, Level of backlight rank reading and writing value is 0-5, 0 represents the darkest, 5 represents the brightest.

7, shortcut bringing up data groups function writing value are 0-9, after writing, all corresponding data group are extracted.

1.4 Error-checking:

In RTU mode, messages include an error–checking field that is based on a Cyclical Redundancy Check (CRC) method. The CRC field checks the contents of the entire message. It is applied regardless of any parity check method used for the individual characters of the message.

CRC Calculation method:

The Cyclical Redundancy Check (CRC) field is two bytes, containing a 16-bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message.

The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

The CRC is started by first preloading a 16-bit register to all 1's. Then a process begins of applying successive eight-bit bytes of the message to the current contents of the register. Only the eight bits of data in each character are used for generating the CRC. Start and stop bits, and the parity bit, do not apply to the CRC.

During generation of the CRC, each eight-bit character is Exclusive ORed with the register contents. The result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position.

The LSB is extracted and examined. If the LSB was a 1 then an Exclusive OR (XOR) of the register with a pre-set, fixed value takes place. If the LSB was a 0, no Exclusive OR (XOR) takes place.

This process is repeated until eight shifts have been performed. After the last (eighth) shift, the next eight-bit character is exclusive ORed with the register's current value, and the process repeats for eight more shifts as described above. The final contents of the register, after all the characters of the message have been applied, is the CRC value.

Three examples of communication using the three available MODBUS Functions Codes supported on this device. (0x03) (0x06) and (0x10)

1 : Read the displayed output voltage and current value. (Using Function 0x03)

Message format sent from Host:

Host sends	Number of bytes	Information sent	Notes
Slave address	1	01	From host to slave address 01H
Function code	1	03	Read holding register(s)
Register starting address	2	0002H	Register starting address
Number of registers to read	2	0002H	A total of 2 registers (4 byte = 2 words)
CRC Checksum:	2	65CBH	CRC Checksum from Host:

For example if the displayed values are 05.00V , 5.000A , The Slave return response would be:

Slave responds	Number of bytes	The information returned	Notes
Slave address	1	01	From slave address 01H
Function code	1	03	Read holding register
Number of bytes read	1	04	A total of 4 Bytes
The contents of 1st register read	2	01F4H	Output voltage display values
The contents of 2nd register read	2	1388H	Output current display value
CRC Checksum:	2	B76BH	CRC Checksum from Slave

2 : Host to set the voltage to 24.00V. (Using Function 0x06)

Message format sent from host:

Host sends	Number of bytes	Information sent	Notes
Slave address	1	01H	From host to slave address 01H
Function code	1	06H	Write single register
Register address	2	0000H	Register address
The value to write to the register	2	0960H	Set the output voltage
CRC Checksum	2	8FB2H	CRC Checksum from host

Message format returned from Slave:

Slave responds	Number of bytes	The information returned	Notes
Slave address	1	01H	From slave address 01H
Function code	1	06H	Write single register
Register address	2	0000H	The Register address
The contents of the register read	2	0960H	Value read
CRC Checksum	2	8FB2H	CRC Checksum from slave

3 : Host to set the voltage to 24.00V , The current 1.500A. (Using Function 0x10 (decimal 16))
Message format sent from HOST:

Host sends	Number of bytes	Information sent	Notes
Slave address	1	01H	From host to slave address 01H
Function code	1	10H	Write multiple registers
Register starting address	2	0000H	Register the starting address
Quantity of registers to write to	2	0002H	2 registers
Number of bytes to write	1	04H	4 Bytes = 2 words = 2 registers
Contents of 1st register	2	0960H	Set the output voltage
Contents of 2nd register	2	05DCH	Set the output current
CRC Code:	2	F2E4H	CRC Checksum from Host:

Message format returned from Slave

From machine responses	Number of bytes	The information returned	Notes
Slave address	1	01H	From Slave address 01H
Function code	1	10H	Write multiple registers
Register the starting address	2	0000H	Register starting address
Number of registers written	2	0002H	2 registers written to
CRC Code:	2	41C8H	CRC Checksum from Slave: