Baoding Shenchen Precision Pump Co., Ltd.
Add: No. 103, Building 2, Pivot Industrial Park, Fuxing East Raod, Baoding, China.
Tel: 0086-312-6780681 Fax: 0086-312-6780636
Website: www.good-pump.com Email: amy@good-pump.com

# Lab V Series MODBUS Communication Protocol

**Note: The hexadecimal numbers are expressed by 'XXXXH' or 'XXH' in the below description.**

## 1. MODBUS-RTU standard communication format

This communication use MODBUS RTU mode, message frame as below:

| Slave address | Function code | Data area | CRC Check (Cyclic Redundancy Check) | |
|---|---|---|---|---|
| 1 Byte | 1 Byte | 0 or up to 252 bytes | 2 Bytes | |
| | | | CRC low | CRC high |

(1) **Slave address**: Host control peristaltic pump address No. The pump address No. should not be same when they are in the same 485 line. The address No. range is 1~32, 0 means broadcast.

(2) **Function code**: The protocol use 2 common function codes which defined by MODBUS protocol.

**03H**: Read holding registers

**06H**: Write single register

**10H**: Write multiple registers

**02H**: Read discrete inputs (Read bits of data )

**05H**: Write single bit to register

(3) **Data area**: The detailed information command that the peristaltic pumps need to follow, such as start/stop, change direction, increase/decrease speed..and so on.

(4) **CRC check**: CRC code is 2 bytes, 16 check codes. Use CRC-16(which used in American binary synchronous system).

Polynomial: G(X)=X16+X15+X2+1.

**CRC check C language code please refer to Appendix 1**.

## 2. Communication Setting

**(1) Communication boudrate: 1200, 2400, 4800, 9600 optional**

**(2) Byte structure:** 1 start bit + 8 data bits +1 parity bit + 1 stop bit

**(3) Bit serial sending order:** The least significant big(**LSB**)….. The most significant bit (**MSB**)

| **Start** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **Check** | **Stop** |
|---|---|---|---|---|---|---|---|---|---|---|

**(4) Data transferring format:**

**Integer (2 bytes):**

Data: The second byte The first byte

Send: The second byte The first byte

For example: 1234H send 12H 34H

**Long integer and Float (4 bytes):**

Data: The fourth byte The third byte The second byte The first byte

Send: The fourth byte The third byte The second byte The first byte
For example: 12345678H send 12H 34H 56H 78H

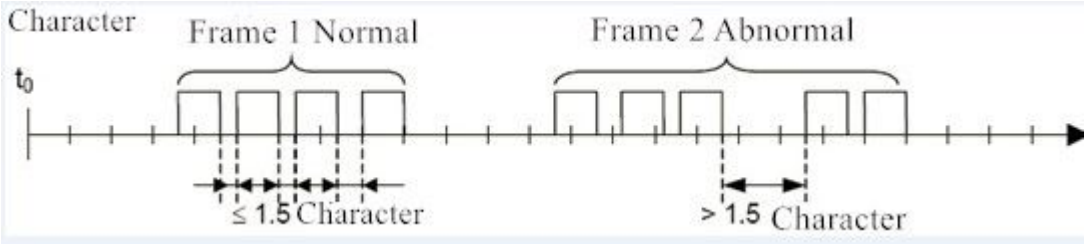
## 3. MODBUS Message RTU Frame Format

| ← MODBUS Message → | | | |
|---|---|---|---|
| Address | Function Code | Data | CRC Check Code |
| 8 Bits | 8 Bits | N x 8 Bits | 16 Bits |

| Start |
|---|
| ⩾3.5 characters |

| End |
|---|
| ⩾3.5 characters |

**The entire message frame must be sent in a continuous stream of characters. If the idle space between two characters is greater than 1.5 character times, the message frame is considered as incomplete, should be discarded by receiving node. As below:**




## 4. Abnormal reaction

When host sending request data, slave receiving data abnormal, it should have abnormal reaction. If the address code sent from host is wrong, there is no this address code between slaves or the data received by slave is wrong when CRC check, no abnormal code return, the host should have super reaction process.

**Function code area**: Abnormal reaction function code is normal reaction function code +80H.

**Data area**: Return abnormal code, define as below.

**Chart 1**: Abnormal code definition

| Code | Name | Meaning |
|---|---|---|
| 01H | Illegal function code | The function code received by peristaltic pump except 03H/06H/10H. |
| 02H | Illegal data address | This abnormal code means the register address is not allowed data which received by peristaltic pump. |

| | | |
|---|---|---|
| 03H | Illegal data value | Written data does not meet the operating range. |
| 06H | Slave(peristaltic pump) busy | The current state of the peristaltic pump conflict with the command received, unable to complete the command. |

**Peristaltic Pump only receive MODBUS command with the Main Interface, other interface do not receive message.**

## 5. Holding register address and content

### Basic Parameters Setting

| Address（Decimal） | Name | Range | Data Type |
|---|---|---|---|
| 1000 | Pump Head | Relative datas refer to Chart 1 | unsigned short int（2 Bytes） |
| 1001 | Tubing Size | Relative datas refer to Chart 1 | unsigned short int（2 Bytes） |
| 1002 | Motor Speed | 0.1-600rpm | float （4 Bytes） |
| 1004 | Flow Rate | 0.1-99999 mL | float （4 Bytes） |
| 1007 | Back Suction Angle | 0-360° | unsigned short int（2 Bytes） |
| 1008 | Start/Stop Control | 1：Start 0：Stop | unsigned short int（2 Bytes） |
| 1009 | Direction Control | 1：Clockwise 0：Anticlockwise | unsigned short int（2 Bytes） |
| 1010 | Full Speed Running | 1：Start full speed 0：Stop full speed | unsigned short int（2 Bytes） |
| 1015 | Set Flow Volume | 0-99999 mL | Float （4 Bytes） |
| 1018 | Working Time | 0.1-9999（s） | Float （4 Bytes） |
| 1020 | Working Mode | 0: Transferring 1: Fixed volume measurement 2: Fixed time and volme | unsigned short int（2 Bytes） |
| 1021 | Pause Time | 0.1-9999（s） | Float (4 Bytes) |
| 1023 | Copy Numbers | 0-9999 次．0 means infinite | unsigned short int (2 Bytes) |

**Note**: ① When working mode is transferring, set up register 1015 and 1018 is invalid.

② When working mode is Fixed volume measurement, set up register 1018 invalid.

③ When working mode is Fixed time and volume, set up register 1002 and 1004 invalid.

④ Please set up the register datas refer to the chart, it can not receive one order with setting up multiple registers.

### Calibration Parameters Setting Up

| Address（Decimal） | Name | Range | Data Type |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 2001 | Testing time | 0.5-9999s | unsigned short int（2 Bytes ） |
| 2002 | Start test | 1：Start 0：Stop | unsigned short int（2 Bytes ） |
| 2003 | Actual volume | 0-9999 mL | float （4 Bytes） |
| 2005 | Restore Defaults | 1：Restore calibration | unsigned short int（2 Bytes） |
| 2006 | Micro Adjustment | 1：Increase 0: Decrease | unsigned short int（2 Bytes） |

**Chart 1** Pump Head & Tubing No.

| Pump Head Name | Pump Head | Tubing Size | Tubing Specific |
|---|---|---|---|
| YZ1515x | 0 | 13 | 13# |
| | | 14 | 14# |
| | | 19 | 19# |
| | | 16 | 16# |
| | | 25 | 25# |
| | | 17 | 17# |
| | | 18 | 18# |
| YZ2515x | 1 | 15 | 15# |
| | | 24 | 24# |
| 2*YZ1515x | 2 | 13 | 13# |
| | | 14 | 14# |
| | | 19 | 19# |
| | | 16 | 16# |
| | | 25 | 25# |
| | | 17 | 17# |
| | | 18 | 18# |
| 2*YZ2515x | 3 | 15 | 15# |
| | | 24 | 24# |
| MCn(10) | 4 | 101 | 1*1 |
| | | 102 | 2*1 |
| | | 103 | 2.4*0.8 |
| | | 104 | 2.79*0.9 |
| | | 105 | 3*1 |
| MCn(6) | 5 | 101 | 1*1 |
| | | 102 | 2*1 |
| | | 103 | 2.4*0.8 |
| | | 104 | 2.79*0.9 |
| | | 105 | 3*1 |
| DZ25-3L | 6 | 15 | 15# |

| | | | |
|---|---|---|---|
| | | 24 | 24# |
| | | 35 | 35# |
| | | 36 | 36# |
| SN15 | 7 | 14 | 14# |
| | | 16 | 16# |
| SN25 | 8 | 24 | 24# |

## Appendix 1——CRC Check C Language Code

### CRC generation process:

1. Put one 16 bits register into hexadecimal FFFF( all 1), we call it CRC register.
2. Make the first 8 bytes with 16 CRC register low bytes XOR, the result put in CRC register.
3. Move CRC register 1 bit to right, MSB zeroing. Extraction and detection of LSB.
4. (If LSB is 0): Repeat Step 3 ( another shift).

   (If LDB is 1): XOR register for CRC polynomial value 0xA001 (1010 0000 0000 0001).
5. Repeat Step 3 and 4, until finish 8 shifts. After finish this operation, will finish the complete operation for 8 Bytes.
6. Repeat Step 2 to Step 5 for the next Bytes in message. Continue this operation till all the message be deal with finished.
7. The final content in CRC register is CRC value.
8. When put CRC value in message, high and low Bytes must be exchanged.