HGM501

GENSET CONTROLLER

COMMUNICATION PROTOCOL

SMARTGEN (ZHENGZHOU) TECHNOLOGY CO., LTD.

Chinese trademark

SmartGen **English trademark**

SmartGen — make your generator ***smart***
SmartGen Technology Co., Ltd.
No.28 Jinsuo Road
Zhengzhou City
Henan Province
P. R. China
Tel: 0086-(0)371-67988888/67981888
0086-(0)371-67991553/67992951
0086-(0)371-67981000(overseas)
Fax: 0086-371-67992952
Web: www.smartgen.com.cn
www.smartgen.cn
Email: sales@smartgen.cn

Table 1 - Version History

Date	Version	Note
2019-11-04	1.0	Original release.

This is applicable for HGM501 Genset Controller.

Symbol illustration for this protocol:

Table 2 - Symbol Illustration

Sign	Remark
▲ **NOTE**	Highlights an essential element of a procedure to ensure correctness;
⚠ **CAUTION!**	Indicates a procedure or practice, which, if not strictly observed, could result in damage or destruction of equipment;
☠ **WARNING**	Indicates a procedure or practice, which could result in injury to personnel or loss of life if not followed correctly.

CONTENT

1. INTRODUCTION

This protocol describes effective transfer of information and data between monitoring system. The monitoring system can be set up if a central PC (or IPC)-based communication master software is added (such as Kingview, Intouch, FIX, Synall etc.).

2. MODBUS BASIC RULES

— All RS232 communication loops should follow the master-slave mode. In this way, data can be transferred between a master (e.g. PC) and 32 slaves.
— Master will make the initialization device to send all information on RS232 communication loops.
— No communication can start from slaves.
— In RS232 communication loop, all communication should be transmitted in "information frame".
— If master or slave receives information frame with unknown command, no response will be given.

3. DATA FRAME FORMAT

Communication is asynchronously transferred by the unit of byte (data frame). Each data frame is a serial data stream of 11 bits between master and slave.

Table 3 - Data Frame Format

Item	Description
Start bit	1-bit
Data bit	8-bit
Parity bit	No parity
Stop bit	2-bit

4. COMMUNICATION PROTOCOL

4.1 ILLUSTRATION

When communication command is sent to the instrument, device who accords with the address code receives the communication command, and removes the address code to read information. If nothing goes wrong, it shall conduct the task, and then send implementation result to the sender. The returned information includes address code, function code of implemented action, data after implemented action, and CRC. If an error occurs, then nothing shall be sent.

4.2 INFORMATION FRAME FORMAT

Table 4 – Information Frame Format

Initiating structure	Address code	Function code	Data field	CRC	End structure
Delay (equivalent to 4 bytes)	1 byte 8-bit	1 byte 8-bit	N bytes N*8-bit	2 bytes 16-bit	Delay (equivalent to 4 bytes)

4.3 ADDRESS CODE

Address code is the first data frame (8-bit) in each transmitted information frame from 0 to 255. Single device address range is 1-247, which means that slave device whose address code is defined by users will receive the information sent by the master. Each slave has a unique address code, and each response begins with its address code. The address code issued by the master means the slave address to be sent to, while address code issued by slave means the responded slave address.

4.4 FUNCTION CODE

4.4.1 ILLUSTRATION

Function code is the second data of each communication transmission. ModBus communication protocol defines function code as 1-255 (01H-0FFH). HGM501 controller uses a part of it. By master request master can tell slave to conduct certain action. By slave response slave can show that it has responded to the master and conducted the action as the function code issued by the slave is the same as the one issued by the master. If the function code MSB is 1 (function code>127), it means slave does not respond, or response has an error.

The following table shows the specific signification and operation of function code.

Table 5 - ModBus Partial Function Codes

Function code	Definition	Operation
01H	Read Coils	Reads single or multiple coils;
03H	Read Registers	Reads single or multiple register data;
05H	Set Single Coil	Set single coil;

4.4.2 01H READ COILS

Master can read all coils in the device by function code 01 (for example: switch close, open, fault, auto status or manual status etc.)

4.4.3 03H READ REGISTERS

With communication command of function code 03H, master can read the numerical registers (all kinds of collected analogue data and pre-set parameter values are stored in the register) inside the instrument. Input register value of 03H mapping data field is 16-bit (2 bytes). So register values read from the instrument are all 2 bytes. For each time maximum readable register values are 125.

Command format of slave response is address code, function code, data field, and CRC code. Data in data field are dual bytes in a group of 2 bytes and high byte is in the front.

4.4.4 05H FORCE SINGLE COIL

With this command master can store single coil data to bit registers (e.g. ATS transfer control). Slave also can respond information to the master with this command.

4.4.5 06H WRITE SINGLE REGISTER

With this command master can store single data to bit registers in the instrument. Register in ModBus communication protocol refers to 16-bit (2 bytes) and high byte is in the front. In this way all points in the device are 2 bytes. Command format is slave address, function code, data field and CRC code.

4.5 DATA FIELD

4.5.1 ILLUSTRATION

Data field varies with different function codes.

4.5.2 FUNCTION 01H DATA FIELD FORMAT

Table 6 - Master Request

Data Sequence	Data Signification	Byte Count
1	Starting address	2
2	Read coil numbers	2

Table 7 - Slave Response

Data Sequence	Data Signification	Byte Count
1	Loopback byte count	1
2	N coil data	1

4.5.3 FUNCTION 03H DATA FIELD FORMAT

Table 8 - Master Request

Data Sequence	Data Signification	Byte Count
1	Starting address	2
2	Read register numbers	2

Table 9 - Slave Response

Data Sequence	Data Signification	Byte Count
1	Loopback byte count	1
2	N register data	N

4.5.4 FUNCTION 05H DATA FIELD FORMAT

Table 10 - Master Request

Data Sequence	Data Signification	Byte Count
1	Coil address	2
2	Force single coil value	2

Table 11 - Slave Response

Data Sequence	Data Signification	Byte Count
1	Coil address	2
2	Single coil value	2

4.5.5 FUNCTION 06H DATA FIELD FORMAT

Table 12 - Master Request

Data Sequence	Data Signification	Byte Count
1	Register address	2
2	Register value (2 bytes)	2

Table 13 - Slave Response

Data Sequence	Data Signification	Byte Count
1	Register address	2
2	Register value (2 bytes)	2

4.6 ERROR CHECK CODE (CRC)

Master or slave can detect whether the received information is wrong or not with CRC. Sometimes due to electric noise or other interference, information will have small change in the transmission process. CRC ensures master or slave does not respond to the wrong information in the transmission process. In this way system safety and efficiency are guaranteed. CRC applies CRC-16 calibration method.

For 2 bytes CRC, low byte is in the front and high byte is in the back.

⚠**NOTE:** All information frame format are same: address code, function code, data area and CRC code.

CRC includes 2 bytes, which is 16-bit binary number. CRC is counted by the sender and placed at the end of the transmitted information. Responded device will count the received information is the same as the information again. If they are different, then it means there is an error.

CRC counting method: first place 16-bit register as 1. Then gradually tackle with 8-bit data information. Only 8-bit of data is used in the process of CRC counting. Start bit and stop bit are not included.

In the process of CRC counting, 8-bit data is Exclusive OR with the register data. The obtained result moves 1 bit to the low byte direction and fill MSB with 0. Check LSB again and if LSB is 1, then make register contents Exclusive OR with the preset values. If LSB is 0, then do not do Exclusive OR counting.

This process is repeated for many times. After the eighth bit move, the next 8-bit shall Exclusive OR with the current register contents. This also repeated for 8 times as the last one. Until all data information is handled, the last register contents are CRC value.

CRC-16 CALCULATION PROCEDURE:

1) Place a 16–bit register as FFFF hex;
2) Make the first 8–bit data Exclusive OR with the low 8-bit of the CRC register, and put the result in the CRC register;
3) Shift the CRC register one bit to the right, and fill MSB with 0. Examine the moved-out bit.
4) If LSB was 0: repeat Step 3 (another shift).
   If LSB was 1: Exclusive OR the CRC register with A001 hex;
5) Repeat Step 3 and 4 until 8 shifts have been performed. When this is done, a complete 8–bit data are processed.
6) Repeat Step 2 to 5 for the next 8–bit data of the message.
7) The final CRC register value is the CRC code. Least Significant Byte is transmitted first and Most Significant Byte is at the last.

⚠**NOTE：** The calculation of CRC code starts from＜slave address＞ for all bytes, excluding <CRC code>.

4.7 EXAMPLES OF INFORMATION FRAME FORMAT

4.7.1 FUNCTION 01H

Table 14 - Function 01H Master Request Example

Request	Bytes	Example (Hex)
Slave address	1	01 Send to slave 01
Function code	1	01 Read coils
Starting address	2	00 Starting address is 0000 00
Count number	2	00 Read 28 coils 1C
CRC code	2	3D CRC code which calculated by PC C3

Table 15 - Function 01H Slave Response Example

Response	Bytes	Example (Hex)
Slave address	1	01 Respond slave address 01
Function code	1	01 Read coil
Read count	1	04 Return coil number; 28 coils (total 4 bytes);
Data 1	1	30 The content of address 07-00;
Data 2	1	00 The content of address 0F-08;
Data 3	1	93 The content of address 17-10;
Data 4	1	0A The content of address 1C-18;
CRC code	2	18 CRC code which calculated by slave. 26

Coil 07-00 can be described by 30H hex, 00110000 decimal; coil 07 is the MSB, while 00 the LSB; Coil 07-00 status: OFF-OFF-ON-ON-OFF-OFF-OFF-OFF.

4.7.2 FUNCTION CODE 03H

Slave address is 01 and start address is 0026H of 3 data.

Table 16 - Data Address Example

Address	Data(Hex)
0026	0014
0028	0014
002A	0005

Table 17 - Function Code 03H Master Request Example

Request	Bytes	Example (Hex)
Slave address	1	01 Send to slave 01
Function code	1	03 Read holding registers
Starting address	2	00 Starting address is 0026 26
Count number	2	00 Read 3 data (total 6 bytes) 03
CRC code	2	E4 CRC code which calculated by PC 00

Table 18 - Function Code 03H Slave Response Example

Response	Bytes	Example (Hex)
Slave address	1	01 Respond slave address 01
Function code	1	03 Read register
Read count	1	06 3 data (total 6 bytes)
Data 1	2	00 The content of address 0026 14
Data 2	2	00 The content of address 0028 14
Data 3	2	00 The content of address 002A 05
CRC code	2	91 CRC code which calculated by slave. 71

4.7.3 FUNCTION CODE 05H

Slave address is 01 and starting address is 0002H of 1 coil. Set 0002 to unit 1.

Table 19 - Coil Data Address Example

Address	Data(Hex)
0000	0
0001	1
0002	0

⚠**NOTE:** FF00 hex coil value is forced to 1 and 000H is forced to 0. Other values are illegal and will not affect the coil.

Table 20 - Function Code 05H Master Request Example

Request	Bytes	Example (Hex)
Slave address	1	01 Send slave address 01
Function code	1	05 Force single coil
Starting address	2	00 Starting address is 0000 00
Data	2	00 Set coil as 1 FF
CRC code	2	8D CRC code which calculated by PC. FA

Table 21 - Function Code 05H Slave Response Example

Slave Response	Bytes	For Example (Hex)
Slave address	1	01 Respond slave address 01
Function code	1	05 Force single coil
Starting address	2	00 Starting address is 0000 00
Data	2	00 Set coil as 1 FF
CRC code	2	8D CRC code which calculated by slave. FA

4.7.4 Function Code 06H

Slave address is 01 and place starting address of 0026H of 1 point as 0014H.

Table 22 - Function Code 06H Master Request Example

Request	Bytes	Example (Hex)
Slave address	1	01 Send slave address 01
Function code	1	06 Write single register
Starting address	2	00 Starting address is 0026H 26
Data	2	00 Place 1 datum (2 bytes in total) 14
CRC code	2	68 CRC code which calculated by PC. 0E

Table 23 - Function Code 06H Slave Response Example

Slave Response	Bytes	For Example (Hex)
Slave address	1	01 Respond slave address 01
Function code	1	06 Write single register
Starting address	2	00 Starting address is 0026H 26
Data	2	00 Place 1 datum (2 bytes in total) 14
CRC code	2	68 CRC code which calculated by slave. 0E

4.8 ERROR HANDLING

When device detects other errors except the CRC code, the slave must send information to the master. The function code MSB is 1, which means the response function code by slave should add 128 based on the function code sent by the master. The following codes show that unexpected errors have occurred.

If CRC error occurs for the information received by the slave, then the device will ignore.

Table 24 - Error Code Format of Slave Response (CRC excluded):

Type	Byte
Address code	1 byte
Function code	1 byte (MSB is 1)
Error code	1 byte
CRC code	2 bytes

Error code:

01 illegal function code

The function code received in the query is not an allowable action for the slave.

02 illegal data address

The data address received in the query is not an allowable address for the slave.

03 illegal data value

A value contained in the query data field is not an allowable value for the slave.

5. ATTACHMENT: ADDRESS AND DATA

Table 25 - Function Code 01 Mapping Data Field 01 01 00 00 00 01

Address	Item	Description
0000H	Genset Start Status	1 for active
0001H	Oil Pressure Low Input Status	0 for active
0002H	Shutdown Status	1 for active
0003H	Reserved	
0004H	Reserved	
0005H	Reserved	
0006H	Reserved	
0007H	Reserved	

Table 26 - Function Code 03H Mapping Data Field

1. Reading register values are 2 bytes and once the most number is 10;

Address	Item & Description
0000H	Gen Voltage
0001H	Active Power
0002H	Gen Frequency
0003H	Battery Voltage
0004H	Gen Current
0005H	Engine Temp.
0006H	Generator Temp.
0007H	Oil Engine Running Count (H) (0-999)
0008H	Software Version
0009H	Engine Temp. Resistance
000AH	Generator Temp. Resistance

2. Register values read are 1 byte, and the most read number at once is 7;

000BH	Auto Protection Status; 0-OFF; 0x80-ON
000CH	Generator Voltage Status; 0 Normal 1 Undervolts 2 Overvolts
000DH	Generator Power Status; 0 Normal 1 Overload
000EH	Generator Frequency Status; 0 Normal 1 Under Frequency 2 Over Frequency
000FH	Battery Voltage Status; 0 Normal 1 Undervolts 2 Overvolts
0010H	Engine Temp. ; 0 Normal 1 High
0011H	Generator Temp.; 0 Normal 1 High

Table 27 - Function Code 03H, 06H Mapping Data Field; All are EEPROM except for notes.

(03 is read by shift address 1000)

(14H) Address	Item & Description	
0000H	Rated Voltage	{110,115,120,130,220,230,240}
0001H	Rated Power (KW)	{0-999} 50
0002H	Rated Frequency (Hz)	{50,60} 50
0003H	Auto Protection Setting	{0,0x80} 0x80
0004H	CT Ratio	{0-999} 50
0005H	Module Address	{1,254} 1
0006H	AC System	{1,2,3,4} 1 1: Single Phase 2: 2 Ph 3 Wire 3: 3 Ph 4 Wire 4: Double Power
0007H	Engine Temp. Curve	{0-Not Used; 1-SGX; 2-SGD; 3-PT100; 4-HGM501-S04} 4
0008H	Generator Temp. Curve	{0-Not Used; 1-SGX; 2-SGD; 3-PT100; 4-HGM501-S04} 0
0009H	Voltage Calibration	746
000AH	Current Calibration	1020
000BH	Power Calibration	1000
000CH	Battery Voltage Calibration	1016
000DH	Engine Temp. Calibration	{500,1500,1000}
000EH	Generator Temp Calibration	{500,1500,1000}
000FH	Engine Temp. Sensor Fix	{-200,200,0}
0010H	Generator Temp. Sensor Fix	{-200,200,0}
0011H	Temporarily used for Phase Fix	70
0012H	Safety On Delay	{0-3600}s 10
0013H	Delay for when power is over 10% set power	30
0014H	Engine Temp. High Threshold	98
0015H	Flexible Input	(0-5) 5 0: Not Used 1: 3P4W Active 2: Reserved 3: Reserved 4: Reserved 5: Generator Temp. Sensor for Multiplex)
0016H	Reserved 6	
0017H	Generator Running Time (H)	0-60
0018H	Generator Running Time (M)	{0-999} 0