MGC120

PETROL 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
Henan Province
P. R. China
Tel: 0086-371-67988888/67981888
0086-371-67991553/67992951
0086-371-67981000(overseas)
Fax: 0086-371-67992952
Web: www.smartgen.com.cn
www.smartgen.cn
Email: sales@smartgen.cn

Software Version

Date	Version	Note
2017-07-17	1.0	Original release.

CONTENT

1. INTRODUCTION

This protocol describes read and write command format of PC serial port and the definition of internal information data for the third-party to develop and use.

MODBUS communication protocol allows the module to transfer information and data effectively with PLC, RTU, SCADA system of international brands (such as, Schneider, Siemens, and Modicon), and DCS or third-party monitoring system compatible with MODBUS. The monitoring system can be set up if only adding central communication master software (such as Kingview，Intouch、FIX、Synal) basing on PC (or IPC).

2. MODBUS BASIC RULES

1) All communication loops should follow the master-slave mode. If so, data can be transferred between a master (e.g. PC) and 32 slaves.
2) The master will initialize all messages sent from communication coil of the device.
3) No communication can start from slaves.
4) In communication loop, all communication should be transmitted in "information frame".
5) If received information frame contains unknown command, no response will be given.

3. DATA FRAME FORMAT

Communication is asynchronously transferred, using byte (data frame) as unit. Between master and slave, every transmitted data frame is 10-bit (stop bit is 1-bit) serial data stream or 11-bit (stop bit is 2-bit).

Data frame format:

Item	Description
Start bit	1-bit
Data bit	8-bit
Parity bit	No parity
Stop bit	1-bit, 2-bit can be set
Transmission Baud Rate	9600bps

4. COMMUNICATION PROTOCOL

4.1 ILLUSTRATION

When communication command is sent to the slave, corresponding slave receives the communication command, then removes address code, and read the information. If no mistakes, it will execute commands, and sends the result back to the master. Response information includes address code, function code, data and error check code (CRC). If an error occurred in receipt of the command, it will send no information.

4.2 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. The device address range is 1–255; this byte shows that the slave defined by users will receive the information sent by the master. Each slave has a unique address code, and responses begin with the address code. A master addresses a slave by placing the slave address in the address field of the message. When the slave sends its response, it places its own address in this address field of the response to let the master know which slave is responding.

4.4 FUNCTION CODE

4.4.1 ILLUSTRATION

This is the second byte of each transmission. ModBus communication protocol defined function code as 1-255 (01H-0FFH). MGC120 controller use part of it. Master sends the request and the slave executes actions according to the function code. If the function code sent by slave is same as that sent by master, it means the response is active. But if the function code MSB is 1 (function code range>127), it means there is no response or response has error.

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

ModBus Partial Function Codes are as follows:

Function code	Definition	Operation
03H	Read Holding Registers	Reads the contents of holding registers.
05H	Force Single Coil	Forces a single coil to either ON or OFF.
06H	Write Single Holding Register	Writing a 16-bit binary number into the holding register.

4.4.2 03H READ HOLDING REGISTERS

With function code 03H command, the master can read the numerical registers inside the device (numerical registers contains various analog and parameter setting values). Input register values of function code 03H mapping data field are 16 bits (2 bytes). So, from the device reads registers values are 2 bytes. Maximum number of readable registers is 125 each time.

The slave received command format is slave address, function code, data field and the CRC code. The data of data field is in double bytes with every two bytes for a group, and high byte is in advance.

4.4.3 05H FORCE SINGLE COIL

Master uses this command to save a single coil data into bit registers in the device (such as ATS transfer control). The slave also uses this function code to feedback information to the master.

4.4.4 06H WRITE SINGLE HOLDING REGISTER

Master uses this command to save a single data into registers in the device. The register in the

ModBus communication protocol is 16-bit (2 bytes) and low-order byte is appended first. Thus all points are 2 bytes. Command format are slave address, function code, data area and CRC code.

4.5 DATA FIELD

4.5.1 ILLUSTRATION

Data field varies with different function codes.

4.5.2 FUNCTION 03H –READ HOLDING REGISTERS

Request:

Data Sequence	Data Signification	Byte Count
1	Starting address	2
2	Read registers	2

Response:

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

4.5.3 FUNCTION 05H –FORCE SINGLE COIL

Request:

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

Response:

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

4.5.4 FUNCTION 06H –FORCE SINGLE COIL

Request:

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

Response:

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

4.6 ERROR CHECK CODE (CRC)

The Error Check Code allows the receiving device to detect a packet that has been corrupted with transmission errors. Sometimes, the transmission information occur imperceptible changes due to electronic noise and other interference and the CRC code ensure the error information does not work to increase the system's safety and efficiency. CRC adapts CRC-16 method of calibration.

When the CRC is appended to the message, the low-order byte is appended first, followed by the high-order byte.

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

The 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 that received in the CRC field. If the two values are not equal, an error will result.

The CRC is started by first preloading a 16–bit register to all 1's. Then a process begins of applying successive 8–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 do not apply to the CRC.

During generation of the CRC, each 8–bit character is exclusive OR with the register contents. Then 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, the register is then exclusive OR with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place.

This process is repeated until eight shifts have been performed. After the last (eighth) shift, the next 8–bit byte is exclusive OR 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 bytes of the message have been applied, is the CRC value.

CRC-16 CALCULATIONPROCEDURE

1) Load a 16–bit register with FFFF hex (all 1's). Call this the CRC register.
2) Exclusive OR the first 8–bit byte of the message with the low–order byte of the CRC register, putting the result in the CRC register.
3) Shift the CRC register one bit to the right (toward the LSB), zero–filling the MSB. Extract and examine the LSB.
4) (If the LSB was 0): Repeat Step 3 (another shift).
   (If the LSB was 1): Exclusive OR the CRC register with the polynomial value A001 hex (1010 0000 0000 0001).
5) Repeat Steps 3 and 4 until 8 shifts have been performed. When this is done, a complete 8–bit byte will have been processed.
6) Repeat Steps 2 through 5 for the next 8–bit byte of the message. Continue doing this until all bytes have been processed.
7) The final contents of the CRC register are the CRC value. Least Significant Byte first. When the 16–bit CRC (two 8–bit bytes) is transmitted in the message, the low-order byte will be transmitted first, followed by the high-order byte.

⚠**Note：The calculating of CRC code starts from＜slave address＞ and except for all bytes of <CRC code>.**

4.7 EXAMPLES OF INFORMATION FRAME FORMAT

4.7.1 FUNCTION CODE 03H

Slave address is 01 and starting address is 3 data of 0026H. (every data is 2 bytes)

Address	Data(Hex)
0026H	0014
0027H	0014
0028H	0005

Request

Request	Bytes	Example (Hex)
Slave address	1	01 Send to the slave 01
Function code	1	03 Read Holding Registers
Starting address	2	00 Starting address is 0026H 26
No. of Points	2	00 Read 3 registers (total 6 bytes) 03
CRC code	2	E4 CRC code which calculated by PC. 00

Response

Response	Bytes	Example (Hex)
Slave address	1	01 Respond to the slave 01
Function code	1	03 Read register
Read count	1	06 3 registers (total 6 bytes)
Data 1	2	00 The content of address 0026 14
Data 2	2	00 The content of address 0027 14
Data 3	2	00 The content of address 0028 05
CRC code	2	91 CRC code which calculated by slave. 71

4.7.2 FUNCTION CODE 05H

Read coil for slave address is 01 and starting address is 1 switch value of 0002H. 0002 unit is 1.

Address	Data(Hex)
0000	0
0001	1
0002	0

⚠ **Note: A value of 00FF hex requests the coil to be ON. A value of 00 0H requests it to be OFF. All other values are illegal and will not affect the coil.**

Request

Request	Bytes	Example (Hex)
Slave address	1	01 Send to the slave 01
Function code	1	05 Force single coil
Starting address	2	00 Starting address for 0000H 00
Data	2	FF Set coil as 1 00
CRC code	2	CD CRC code which calculated by PC. FB

Response

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

4.7.3 FUNCTION CODE 06H

Slave address is 01 and starting address is 1 switch value of 0002H..

Request	Bytes	Example (Hex)
Slave Address	1	01 Respond to the slave 01
Function Code	1	06 Write single register
Starting Address	2	00 Starting address is 00E3H E3
Data	2	00 set one point data（2 bytes totally） 02
CRC Code	2	F9 CRC code which calculated by master FD

Slave Response	Bytes	For Example (Hex)
Slave Address	1	01 Respond to slave address
Function Code	1	06 Write single register
Starting Address	2	00 Starting address is 00E3H E3
Data	2	00 set one point data（2 bytes totally） 02
CRC Code	2	F9 CRC code which calculated by master FD

4.8 ERROR HANDLING

When device detected 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. The following codes show that unexpected errors have occurred.

CRC error received from the master will be ignored by the device.

The frame format of error code that responds by slave is as follows (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. ADDRESS AND DATA

Function code 01H map data field

Address	Item	Description	Bytes
0000H	Common Alarm	1 for active(LSB)	1bit
0001H	Reserved	1 for active	1bit
0002H	Common Shutdown Alarm	1 for active	1bit
0003H	Reserved	1 for active	1bit
0004H	Reserved	1 for active	1bit
0005H	Reserved	1 for active	1bit
0006H	Reserved	1 for active	1bit
0007H	Reserved	1 for active	1bit
0008H	Reserved	1 for active	1bit
0009H	Reserved	1 for active	1bit
000AH	Reserved	1 for active	1bit
000BH	Reserved	1 for active	1bit
000CH	Over Frequency Alarm Shutdown	1 for active	1bit
000DH	Under Frequency Alarm Shutdown	1 for active	1bit
000EH	Over Volt Alarm Shutdown	1 for active	1bit
000FH	Under Volt Alarm Shutdown	1 for active(MSB)	1bit
0010H	Reserved	1 for active(LSB)	1bit
0011H	Fail to start	1 for active	1bit
0012H	Reserved	1 for active	1bit
0013H	Low Oil Pressure Alarm Shutdown	1 for active	1bit
0014H	Reserved	1 for active	1bit
0015H	Reserved	1 for active	1bit
0016H	Reserved	1 for active	1bit
0017H	Reserved	1 for active	1bit
0018H	Reserved	1 for active	1bit
0019H	Reserved	1 for active	1bit
001AH	Reserved	1 for active	1bit
001BH	Reserved	1 for active	1bit
001CH	Reserved	1 for active	1bit
001DH	Reserved	1 for active	1bit
001EH	Reserved	1 for active	1bit
001FH	Reserved	1 for active(MSB)	1bit
0020H	Reserved	1 for active(LSB)	1bit
0021H	Reserved	1 for active	1bit
0022H	Reserved	1 for active	1bit
0023H	Reserved	1 for active	1bit
0024H	Reserved	1 for active	1bit
0025H	Reserved	1 for active	1bit
0026H	Reserved	1 for active	1bit
0027H	Reserved	1 for active	1bit
0028H	Auto Mode	1 for active	1bit

Address	Item	Description	Bytes
0029H	Manual Mode	1 for active	1bit
002AH	Reserved	1 for active	1bit
002BH	Reserved	1 for active	1bit
002CH	Reserved	1 for active	1bit
002DH	Reserved	1 for active	1bit
002EH	Reserved	1 for active	1bit
002FH	Reserved	1 for active(MSB)	1bit
0030H	Reserved	1 for active(LSB)	1bit
0031H	Remote Start Input Status	1 for active	1bit
0032H	Low Oil Pressure Input Status	1 for active	1bit
0033H	Reserved	1 for active	1bit
0034H	Reserved	1 for active	1bit
0035H	Reserved	1 for active	1bit
0036H	Reserved	1 for active	1bit
0037H	Reserved	1 for active	1bit
0038H	Starter Relay Output	1 for active	1bit
0039H	Fuel Relay Output	1 for active	1bit
003AH	Ignition Relay Output	1 for active	1bit
003BH	Programmable Output 1	1 for active	1bit
003CH	Programmable Output 2	1 for active	1bit
003DH	Reserved	1 for active	1bit
003EH	Reserved	1 for active	1bit
003FH	Reserved	1 for active(MSB)	1bit
0040H	Mains Failure	1 for active(LSB)	1bit
0041H	Mains Available	1 for active	1bit
0042H	Mains Over Volt	1 for active	1bit
0043H	Mains Under Volt	1 for active	1bit
0044H	Without Mains	1 for active	1bit
0045H	Reserved	1 for active	1bit
0046H	Reserved	1 for active	1bit
0047H	Reserved	1 for active(MSB)	1bit
0048H	Gen Available	1 for active(LSB)	1bit
0049H	Gen Over Volt	1 for active	1bit
004AH	Gen Under Volt	1 for active	1bit
004BH	Gen Over Frequency	1 for active	1bit
004CH	Gen Under Frequency	1 for active	1bit

Function code 03H map data field

Address	Item & Description	Remark
0000H	Mains UA	Unsigned
0001H	Reserved	Unsigned
0002H	Reserved	Unsigned
0003H	Reserved	Unsigned
0004H	Reserved	Unsigned

Address	Item & Description	Remark
0005H	Reserved	Unsigned
0006H	Mains Frequency	Unsigned (*10)
0007H	Gen UA	Unsigned
0008H	Reserved	Unsigned
0009H	Reserved	Unsigned
000AH	Reserved	Unsigned
000BH	Reserved	Unsigned
000CH	Reserved	Unsigned
000DH	Gen Frequency	Unsigned (*10)
000EH	Reserved	Unsigned
000FH	Reserved	Unsigned
0010H	Reserved	Unsigned
0011H	Temp Value of Temperature Sensor	Unsigned
0012H	Resistance Value of Temperature Sensor	Unsigned (*10)
0013H	Reserved	Unsigned
0014H	Oil Pressure Input Resistance Value	Unsigned (*10)
0015H	Reserved	Unsigned
0016H	Reserved	Unsigned (*10)
0017H	Speed	Unsigned
0018H	Battery Voltage	Unsigned (*10)
0019H	Reserved	Unsigned
001AH	Reserved	Unsigned
001BH	Reserved	Unsigned
001CH	Reserved	Unsigned
001DH	Reserved	Unsigned
001EH	Reserved	Unsigned
001FH	Reserved	Unsigned
0020H	Reserved	Unsigned
0021H	Reserved	Unsigned
0022H	Controller Running Status details to see ***Generator Status Form***	Unsigned
0023H	Controller Running Status Delay	Unsigned
0024H	Auto Running Status: 0 Start 1 Stop 2 Without delay	Unsigned
0025H	Auto Running Delay	Unsigned
0026H	ATS Running Status: 0 Without delay 1 Transfer interval 2 Mains close 3 Open 4 Gen Close	Unsigned
0027H	ATS Running Status Delay	Unsigned
0028H	Mains Status: 0 Normal 1 Abnormal 2 Without delay	Unsigned
0029H	Mains Status Delay	Unsigned
002AH	Reserved	Unsigned
002BH	Engine Total Running Time (hour)	Unsigned（0-9999）
002CH	Engine Total Running Time (minute)	Unsigned（0-9999）

Address	Item & Description	Remark
002DH	Engine Total Running Time (second)	Unsigned（0-9999）
002EH	Start Times (MSB)	Unsigned（0-99）
002FH	Start Times (LSB)	Unsigned（0-9999）
0030H	Reserved	Unsigned
0031H	Reserved	Unsigned
0032H	Software Version	Unsigned (*10)
0033H	Hardware Version	Unsigned (*10)
0034H	Released Time (year)	Unsigned
0035H	Released Time (month)	Unsigned
0036H	Released Time (date)	Unsigned
0037H	Reserved	Unsigned
0038H	Reserved	Unsigned
0039H	Reserved	Unsigned

Function code 05H map data field

Address	Item	Description
0000	Remote Start Button	1 for active
0001	Remote Stop Button	1 for active
0002	Remote Manual /Auto Button	1 for active
0003	Reserved	1 for active
0004	Reserved	1 for active
0005	Reserved	1 for active

Generator Status Form

No.	Content	Description
0	Standby	Without delay in this status
1	Pre-heat	
2	Fuel Output	Without delay in this status
3	Start	
4	Start Interval	
5	Safety On Delay	
6	Start Idle	
7	Warming Up	
8	Waiting for Load	Without delay in this status
9	Normal Running	Without delay in this status
10	Cooling Down	
11	Stop Idle	
12	Energise To Stop	