



SmartGen
ideas for power

HAT162

ATS CONTROLLER

COMMUNICATION PROTOCOL

SmartGen

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

All rights reserved. No part of this publication may be reproduced in any material form (including photocopying or storing in any medium by electronic means or other) without the written permission of the copyright holder.

Applications for the copyright holder's written permission to reproduce any part of this publication should be addressed to SmartGen Technology at the address above.

Any reference to trademarked product names used within this publication is owned by their respective companies.

SmartGen Technology reserves the right to change the contents of this document without prior notice.

Table 1 - Software Version

Date	Version	Note
2018-04-20	1.0	Original release.

CONTENT

1.	INTRODUCTION.....	4
2.	MODBUS BASIC RULES	4
3.	DATA FRAME FORMAT.....	4
4.	COMMUNICATION PROTOCOL.....	4
4.1	ILLUSTRATION	4
4.2	INFORMATION FRAME FORMAT	5
4.3	ADDRESS CODE	5
4.4	FUNCTION CODE.....	5
4.4.1	ILLUSTRATION.....	5
4.4.2	01H READ COILS.....	5
4.4.3	03H READ HOLDING REGISTERS	5
4.4.4	05H FORCE SINGLE COIL	6
4.5	DATA FIELD.....	6
4.5.1	ILLUSTRATION.....	6
4.5.2	FUNCTION 01H –READ COILS	6
4.5.3	FUNCTION 03H –READ HOLDING REGISTERS	6
4.5.4	FUNCTION 05H –FORCE SINGLE COIL.....	6
4.6	ERROR CHECK CODE (CRC).....	7
4.7	EXAMPLES OF INFORMATION FRAME FORMAT	8
4.7.1	FUNCTION CODE 01H	8
4.7.2	FUNCTION CODE 03H	9
4.7.3	FUNCTION CODE 05H	10
4.8	ERROR HANDLING	11
5.	APPENDIX: ADDRESS AND DATA	12
5.1	FUNCTION CODE 03H MAP DATA FIELD	12
5.2	FUNCTION CODE 05H MAP DATA FIELD	15
5.3	ATS STATUS TABLE	15
5.4	1# AC STATUS TABLE.....	16
5.5	2# AC STATUS TABLE.....	16

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

- All RS232 communication loops should follow the master-slave mode. If so, data can be transferred between a master (e.g. PC) and 32 slaves.
- The master will initialize all messages sent from RS232 communication coil of the device.
- No communication can start from slaves.
- In RS232 communication loop, all communication should be transmitted in “information frame”.
- 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 11-bit serial data stream.

Table 2 - Data frame format

Item	Description
Start bit	1-bit
Data bit	8-bit
Parity bit	No parity
Stop bit	2-bit can be set.

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

Table 3 – 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 0-255). Single device address range is 1–247; 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). HAT162 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.

Table 4 - ModBus Partial Function Codes

Function code	Definition	Operation
01H	Read Coils	Reads the contents of coils
03H	Read Holding Registers	Reads the contents of holding registers
05H	Force Single Coil	Forces a single coil to either ON or OFF.

4.4.2 01H READ COILS

Master can read all coils (like switch on/off, fault information, auto or manual status and so on) of the device via function code 01 communication command.

4.4.3 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.

4.4.4 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.

4.5 DATA FIELD

4.5.1 ILLUSTRATION

Data field varies with different function codes.

4.5.2 FUNCTION 01H –READ COILS

Request:

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

Response:

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

4.5.3 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

4.5.4 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

4.6 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 01H

Slave address is 00 and read starting address is 20H of 0000H (decimalism32).

Request

Request	Bytes	Example (Hex)
Slave address	1	01 Send to the slave 01
Function code	1	01 Read coils
Starting address	2	00 Starting address is 0000 00
No. of Points	2	00 Read 28 coils 1C
CRC code	2	3D CRC code which calculated by PC. C3

Response

Response	Bytes	Example (Hex)
Slave address	1	01 Respond to the slave 01
Function code	1	01 Read coils
Read count	1	04 Respond coils: 28 coils (total 4bytes)
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

Discrete value 07-00 expressed as 30H of hexadecimal and 00110000 of binary, 07 is in MSB and 00 is in LSB. 07-00 discrete value status are: OFF – OFF – ON – ON – OFF – OFF – OFF – OFF.

4.7.2 FUNCTION CODE 03H

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

Address	Data(Hex)
0026	0014
0027	0014
0028	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 0026 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

4.7.3 FUNCTION CODE 05H

Read coil for slave address is 01 and starting address is 1 discrete 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 0000H 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 0000 00
Data	2	00 Set coil as 1 FF
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 0000 00
Data	2	00 Set coil as 1 FF
CRC code	2	CD CRC code which calculated by slave. FB

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. APPENDIX: ADDRESS AND DATA

5.1 FUNCTION CODE 03H MAP DATA FIELD

Address	Items	Description	
0000H	Reserved	1 for active(LSB)	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active (MSB)	1bit
0001H	1#Breaker Status	0: open 1: Close	1bit
	1# Normal	0: Abnormal 1:Normal	1bit
	2# Breaker Status	0: open 1: Close	1bit
	2# Normal	0: Abnormal 1:Normal	1bit
	Auto/Manual	0: Auto 1: Manual	1bit
	Prior Transfer to 1# Breaker	1 for active	1bit
	Prior Transfer to 2# Breaker	1 for active	1bit
	Generator Start Status Output	0: Standby 1: Start	1bit
	Reserved	1 for active	1bit
	1# Breaker Close Feedback Signal Input	1 for active	1bit
	2# Breaker Close Feedback Signal Input	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
0002H	1# Abnormal	1 for active	1bit
	1# Overvoltage	1 for active	1bit
	1# Undervoltage	1 for active	1bit
	1# Loss of Phase	1 for active	1bit
	1# Phase Rotation	1 for active	1bit
	1# Overfrequency	1 for active	1bit
	1# Underfrequency	1 for active	1bit
	Reserved	1 for active	1bit



Address	Items	Description	
	2# Abnormal	1 for active	1bit
	2# Overvoltage	1 for active	1bit
	2# Undervoltage	1 for active	1bit
	2# Loss of Phase	1 for active	1bit
	2# Phase Rotation	1 for active	1bit
	2# Overfrequency	1 for active	1bit
	2# Underfrequency	1 for active	1bit
	Reserved	1 for active	1bit
0003H	1# Close Relay Output	1 for output	1bit
	2# Close Relay Output	1 for output	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Programmable Output 1 Status	1 for output	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
0004H	1# Transfer Fail Alarm	1 for active	1bit
	1# Close Fail	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	2# Transfer Fail Alarm	1 for active	1bit
	2# Close Fail	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
	Reserved	1 for active	1bit
0005H	1#Uab		
0006H	1#Ubc		
0007H	1#Uca		

Address	Items	Description
0008H	1#Ua	
0009H	1#Ub	
000AH	1#Uc	
000BH	Reserved	
000CH	Reserved	
000DH	Reserved	
000EH	1# Frequency	Frequency*10
000FH	Reserved	
0010H	2#Uab	
0011H	2#Ubc	
0012H	2#Uca	
0013H	2#Ua	
0014H	2#Ub	
0015H	2#Uc	
0016H	Reserved	
0017H	Reserved	
0018H	Reserved	
0019H	2# Frequency	Frequency *10
001AH	Reserved	
001BH	Reserved	
001CH	Reserved	
001DH	ATS Status (<i>ATS STATUS TABLE</i>)	
001EH	ATS Status Delay Value	
001FH	1# Status (<i>1# AC STATUS</i>)	
0020H	1# Status Delay Value	
0021H	2# Status (<i>1# AC STATUS</i>)	
0022H	2# Status Delay Value	
0023H	Controller Model	162: HAT162
0024H	Controller Software Version	
0025H	Controller Hardware Version	
0026H	Release Time (Year)	
0027H	Release Time (Month)	
0028H	Release Time (Day)	
0029H	Reserved	
002AH	AC System	0: 3 Phase, 4 Wire; 1: Single Phase, 2 Wire; 2: 3 Phase, 3 Wire; 3: 2 Phase, 3 Wire;
002BH	1#Ua Phase	
002CH	1#Ub Phase	
002DH	1#Uc Phase	
002EH	2#Ua Phase	
002FH	2#Ub Phase	

Address	Items	Description
0030H	2#Uc Phase	
0031H	Reserved	
0032H	Reserved	
0033H	Reserved	
0034H	Reserved	
0035H	Reserved	
0036H	Reserved	
0037H	Reserved	
0038H	Reserved	
0039H	Reserved	
003AH	Reserved	
003BH	Reserved	
003CH	Reserved	
003DH	Reserved	
003EH	Reserved	
003FH	Reserved	
0040H	Reserved	

5.2 FUNCTION CODE 05H MAP DATA FIELD

Discrete Values		
Address	Item	Description
0000H	Remote Close 1# Breaker	1 for active (0xFF00 for active)
0001H	Reserved	1 for active (0xFF00 for active)
0002H	Remote Close 2# Breaker	1 for active (0xFF00 for active)
0003H	Reserved	1 for active (0xFF00 for active)
0004H	Auto/Manual Switch Key	1 for active (0xFF00 for active)
0005H	Manual Test Run Key	1 for active (0xFF00 for active)

5.3 ATS STATUS TABLE

No.	Content	Description
0x01	1# Breaker is closing	
0x02	1# Breaker is opening	
0x03	2# Breaker is closing	
0x04	2# Breaker is opening	
0x05	Transfer Interval Delay	
0x06	1# Breaker is closing again	
0x07	1# Breaker is opening again	
0x08	2# Breaker is closing again	
0x09	2# Breaker is opening again	
0x0a	Transfer Delay Expired	
0xff	End of process	

5.4 1# AC STATUS TABLE

No.	Content	Description
0	1# Normal	No delay values will be displayed in this state.
1	1# Normal Delay	
2	1# Abnormal	No delay values will be displayed in this state.
3	1# Abnormal Delay	

5.5 2# AC STATUS TABLE

No.	Content	Description
0	2# Normal	No delay values will be displayed in this state.
1	2# Normal Delay	
2	2# Abnormal	No delay values will be displayed in this state.
3	2# Abnormal Delay	

