HAT530P

DUAL POWER ATS CONTROLLER

COMMUNICATION PROTOCOL

SMARTGEN (ZHENGZHOU) TECHNOLOGY CO., LTD

Chinese trademark

English trademark

SmartGen — make your generator ***smart***

SmartGen Technology Co., Ltd.
No.28 Jinsuo Road
Zhengzhou City
Henan Province
P. R. China
Tel: +86-371-67988888/67981888/67992951
+86-371-67981000(overseas)
Fax: +86-371-67992952
Web: www.smartgen.com.cn
www.smartgen.cn
Email: sales@smartgen.cn

Table 1 Software Version

Date	Version	Content
2021-01-27	1.0	Original release.

CONTENT

1 OVERVIEW

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 etc.), and DCS or third-party monitoring system which is compatible with MODBUS. 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 communication loops should follow the master-slave mode. In this way, data can be transferred between a master (e.g. PC) and 32 slaves.

— The master will initialize all information transferred by the device on the communication loop.

— No communication can start from slaves.

— In 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 10 bits (stop bit: 1) or 11 bits (stop bit: 2) between master and slave.

Table 2 Data Frame Format

Item	Bits
Start Bit	1-bit
Data Bit	8-bit
Parity Bit	None
Stop Bit	1-bit, 2-bit can be set
Baud Rate	9600bps

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 3 Information Frame Format

Initiating Structure	Address Code	Function Code	Data Field	CRC	End Structure
Delay (equivalent to 4 bytes)	1 byte 8-bit	1byte 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, 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). This 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 4 ModBus Partial Function Codes

Function Code	Definition	Operation
03H	Read Registers	Read single or multiple register data
05H	Place Single Coil	Place single coil
06H	Write Single Register	Write a 16-bit binary number to register

4.4.2 03H READ REGISTERS

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

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

4.4.3 05H PLACE SINGLE COIL

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

4.4.4 06H WRITE SINGLE REGISTER

With this command master can store single data to bit registers in the device. 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 CORRESPONDING DATA FIELD FORMAT TO FUNCTION CODE 03H

Table 5 Master Request

Data Sequence	Data Signification	Byte Count
1	Starting Address	2
2	Read Register Numbers	2

Table 6 Slave Response

Data Sequence	Data Signification	Byte Count
1	Loopback Byte Count	1
2	N Register Data	N

4.5.3 CORRESPONDING DATA FIELD FORMAT TO FUNCTION CODE 05H

Table 7 Master Request

Data Sequence	Data Signification	Byte Count
1	Coil Address	2
2	Forced Single Coil Value	2

Table 8 Slave Response

Data Sequence	Data Signification	Byte Count
1	Coil address	2
2	Single Coil Value	2

4.5.4 CORRESPONDING DATA FIELD FORMAT TO FUNCTION CODE 06H

Table 9 Master Request

Data Sequence	Data Signification	Byte Count
1	Register Address	2
2	Register Value (2 bytes)	2

Table 10 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 imperceptible changes 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 formats are same: address code, function code, data field 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 recalculate whether the CRC code of the received information is the same as that received. 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 code value.

CRC-16 Code Calculation Procedure:

a) Place a 16-bit CRC register as FFFF hex;

b) 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;

c) Shift the CRC register one bit to the right, and fill MSB with 0. Examine the moved-out bit;

d) If LSB was 0: repeat Step 3 (another shift);

e) If LSB was 1: Exclusive OR the CRC register with A001 hex;

f) Repeat Step 3 and 4 until 8 shifts have been performed. When this is done, a complete 8–bit data are processed;

g) Repeat Step 2 to 5 for the next data processing;

h) The final CRC register value is the CRC code. Low-order 8-bit data is transmitted first and high-order 8-bit data is at the last.

⚠**NOTE: The calculation 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 (each data is 2 bytes).

Table 11 Data Address

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

Table 12 Function Code 03H Master Request Example

Request	Bytes	Example (Hex)	
Slave Address	1	01	Send to slave 01
Function Code	1	03	Read point register
Starting Address	2	00 26	Starting address is 0026H
Read Number	2	00 03	Read 3 data (total 6 bytes)
CRC Code	2	E4 00	CRC code which calculated by master

Table 13 Function Code 03H Slave Response Example

Response	Bytes	Example (Hex)	
Slave Address	1	01	Respond slave address 01
Function Code	1	03	Read point register
Read Bytes	1	06	3 data (total 6 bytes)
Point 1 Data	2	00 14	The contents of address 0026H
Point 2 Data	2	00 14	The contents of address 0027H
Point 3 Data	2	00 05	The contents of address 0028H
CRC Code	2	91 71	CRC code which calculated by slave

4.7.2 FUNCTION CODE 05H

Slave address is 01 and starting address is 1 coil of 0002H, place 0002H unit as 1.

Table 14 Coil Data Address

Address	Data (Hex)
0000	0
0001	1
0002	0

⚠**NOTE:** FF00 hex coil is forced to 1 and 0000H is forced to 0. Other values are illegal and will not affect the coil.

Table 15 Function Code 05H Master Request Example

Request	Bytes	Example (Hex)
Slave Address	1	01 Send slave address 01
Function Code	1	05 Forced coil
Starting Address	2	00 Starting address is 0000H 00
Data	2	FF Place coil as 1 00
CRC Code	2	CD CRC code which calculated by master FB

Table 16 Function Code 05H Slave Response Example

Response	Bytes	Example (Hex)
Slave Address	1	01 Respond slave address 01
Function Code	1	05 Forced coil
Starting Address	2	00 Starting address is 0000H 00
Data	2	FF Place coil as 1 00
CRC Code	2	CD CRC code which calculated by master FB

4.7.3 FUNCTION CODE 06H

Slave address is 01 and place the 1 point content of starting address 00E3H as 0002H.

Table 17 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 E3	Starting address is 00E3H
Data	2	00 02	Place 1 point data (total 2 bytes)
CRC Code	2	F9 FD	CRC code which calculated by master

Table 18 Function Code 06H Slave Response Example

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

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 19 Error Code Format of Slave Response (CRC excluded)

Type	Bytes
Address Code	1 byte
Function Code	1 byte (MSB is 1)
Error Code	1 byte
CRC Code	2 bytes

Error Function 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

5.1 FUNCTION CODE 03H MAPPING DATA FIELD

Table 20 Function Code 03H Mapping Data Field

Address	Item	Description	Bytes
0000	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
0001	1# Switch Status	0 Open 1 Close	1bit
	1# Normal	0 Abnormal 1 Normal	1bit
	2# Switch Status	0 Open 1 Close	1bit
	2# Normal	0 Abnormal 1 Normal	1bit
	Auto/Manual	0 Auto 1 Manual	1bit
	1# Master	1 for active	1bit
	2# Master	1 for active	1bit
	Generator Start Output Status	0 Not Start 1 Start	1bit
	Force Open Input Port Status	1 for active	1bit
	Auto Transfer/Restore Status	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
0002	1# Abnormal	1 for active	1bit
	1# Over Voltage	1 for active	1bit
	1# Under Voltage	1 for active	1bit
	1# Loss of Phase	1 for active	1bit
	1# Reverse Phase Sequence	1 for active	1bit

Address	Item	Description	Bytes
	1# Over Frequency	1 for active	1bit
	1# Under Frequency	1 for active	1bit
	Reserved		1bit
	2# Abnormal	1 for active	1bit
	2# Over Voltage	1 for active	1bit
	2# Under Voltage	1 for active	1bit
	2# Loss of Phase	1 for active	1bit
	2# Reverse Phase Sequence	1 for active	1bit
	2# Over Frequency	1 for active	1bit
	2# Under Frequency	1 for active	1bit
	Reserved		1bit
0003	LO Output Port Status	1 for active	1bit
	NO Output Port Status	1 for active	1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
	Reserved		1bit
0004	Reserved		2Bytes
0005	1# UAB	Signed	2Bytes
0006	1# UBC	Signed	2Bytes
0007	1# UCA	Signed	2Bytes
0008	1# UA	Signed	2Bytes
0009	1# UB	Signed	2Bytes
0010	1# UC	Signed	2Bytes
0011	1# UA Phase	Signed	2Bytes
0012	1# UB Phase	Signed	2Bytes
0013	1# UC Phase	Signed	2Bytes
0014	1# Frequency	Signed (*10)	2Bytes
0015	Reserved		2Bytes
0016	2# UAB	Signed	2Bytes
0017	2# UBC	Signed	2Bytes
0018	2# UCA	Signed	2Bytes
0019	2# UA	Signed	2Bytes

Address	Item	Description	Bytes
0020	2# UB	Signed	2Bytes
0021	2# UC	Signed	2Bytes
0022	2# UA Phase	Signed	2Bytes
0023	2# UB Phase	Signed	2Bytes
0024	2# UC Phase	Signed	2Bytes
0025	2# Frequency	Signed (*10)	2Bytes
0026	Reserved		2Bytes
0027	1# Voltage Normal Delay Time (s)	Unsigned	2Bytes
0028	2# Voltage Normal Delay Time (s)	Unsigned	2Bytes
0029	ATS Status	ATS Status Table	2Bytes
0030	ATS Status Delay Value	Signed	2Bytes
0031	1# Status	AC 1# Status Table	2Bytes
0032	1# Status Delay Value	Signed	2Bytes
0033	2# Status	AC 2# Status Table	2Bytes
0034	2# Status Delay Value	Signed	2Bytes
0035	Controller Model	Signed	2Bytes
0036	Software Version	Signed (*10)	2Bytes
0037	Hardware Version	Signed (*10)	2Bytes
0038	Issue Year	Only save last two digits of year	2Bytes
0039	Issue Month	Signed	2Bytes
0040	Issue Day	Signed	2Bytes
0041	Reserved		2Bytes
0042	Reserved		2Bytes

5.2 FUNCTION CODE 05H MAPPING DATA FIELD

Table 21 Function Code 05H Mapping Data Field

Address	Item	Description
0000	Remote 1# Close Key	Switch set to 1 active
0001	Remote Open Key	Switch set to 1 active
0002	Remote 2# Close Key	Switch set to 1 active
0003	Reserved	Switch set to 1 active
0004	Remote Manual/Auto Key	Switch set to 1 active
0005	Reserved	Switch set to 1 active

5.3 ATS STATUS TABLE

Table 22 ATS Status Table

No.	Content	Description
1	1# Closing	
2	1# Opening	
3	2# Closing	
4	2# Opening	
5	Transfer Delay	
0xff	End	

5.4 AC 1# STATUS TABLE

Table 23 AC 1# Status Table

No.	Content	Description
0	1# Normal	
1	1# Normal Delay	
2	1# Abnormal	
3	1# Abnormal Delay	

5.5 AC 2# STATUS TABLE

Table 24 AC 2# Status Table

No.	Content	Description
0	2# Normal	
1	2# Normal Delay	
2	2# Abnormal	
3	2# Abnormal Delay	