《Communication protocol manual》

Menu

1.0 Communication protocol manual OUTLINE

Smart Bus Servo Communication protocol is mainly applicable to Potentiometer series of servos and Magnetic Encoding servo.

Potentiometer series servo adopts TTL level and single bus (a signal line time-sharing multiplexing transmission and receiving data signal) communication connection, physical connection is three lines, including two positive and negative poles of power supply;

Magnetic Encoding series servo adopts ARM 32-bit single-chip computer as the main control core, position induction adopts 360 degree 12-bit precision magnet induction angle scheme, through. The communication level adopts RS485 mode with strong anti-jamming ability. The communication still adopts asynchronous duplex, and the sending and receiving signals are asynchronous processing.

Question-and-answer communication is adopted between the controller and the servo. The controller sends out the instruction package and the servo returns to the response package.

Multiple servos are allowed in a bus control network, so each servo is assigned a unique ID number in the network. The control command issued by the controller contains ID information. Only the servo matching ID number can receive the command completely and return the response information.

The communication mode is serial asynchronous. A frame of data is divided into 1 bit start bit, 8 bit data bit and 1 bit stop bit. There are no parity bits, totally 10 bits.

The difference between Potentiometer series and Magnetic Encoding series communication protocols is that two bytes represent high byte and low byte respectively when some parameters of the memory table are in the range of two bytes. Among them, the parameters of Potentiometer series are in the address of the memory table after the high byte and the low byte after the high byte, while the Magnetic Encoding series is in the low byte after the high byte. In addition, each servo has slightly different functions, so the actual control should refer to the memory table of the specific model.

1.1 Instruction packet

Instruction package format:

initial	ID No.	数据长度	指令	参数	校验和
0XFF 0XFF	ID	Length	Instruction	Parameter1 ...Parameter N	Check Sum

Initial: Continuous receipt of two 0XFFs indicating arrival of data packets。

ID No.: Each servo has an ID number. ID number ranges from 0 to 253, converted to hexadecimal0X00～0XFD。

Broadcast ID: ID No. 254 is a broadcast ID. If the ID number issued by the controller is 254 (0XFE), all the Servos receive instructions, and no response information is returned except PING instructions (multiple servos can not use broadcast PING instructions on the bus).

Data length: equal to the parameter N to be sent plus 2, that is "N + 2"。

Instruction: Packet Operating Function Code, see Instruction Type 1.3。

PARAMETERS: In addition to the additional control information required by the instructions, the parameters support the maximum two-byte parameter to represent a memory value. The byte order refers to the manual memory control table for servo usage (different types of servo have different byte order).

Check sum: Check sum and Check Sum, the calculation method is as follows Check Sum = ~ (ID + Length + Instruction + Parameter1 + ... Parameter N) If the sum in parentheses exceeds 255, the lowest byte will be taken, and "~" means reverse.

1.2 Reply Packet

Reply packet is the servo' s reply to the controller, Reply packet format is below:

initial	ID No	Data Length	current state	Parameter	Check sum
0XFF 0XFF	ID	Length	ERROR	Parameter1 ...Parameter N	Check Sum

The returned response package contains the current status ERROR of the servo.

If the current status of the servo is not normal, it will be reflected through this byte (the meaning of each status is detailed in the manual memory control table). If ERROR is 0, the servo will have no error information.

If the instruction is a read instruction (READ DATA), then Parameter 1... Parameter N is the read information.

1.3 Instruction type

The following instructions are available for Feetech Serial Bus Intelligent servo Communication Protocol:

instruction	function	value	Parameter length
PING（查询）	Query the working status	0x01	0
READ DATA（读）	Query the Characters in the Control Table	0x02	2
WRITE DATA（写）	Write characters into the control table	0x03	≥1
REGWRITE DATA(异步写)	Similar to WRITE DATA, the control character does not act immediately after writing until the ACTION instruction arrives.	0x04	Not less than 2
ACTION（执行异步写）	Actions that trigger REG WRITE writes	0x05	0
SYCNWRITE DATA（同步写）	For simultaneous control of multiple servos	0x83	Not less than 2
RESET（复位）	Reset control table to factory value	0x06	0

1.3.1Query status instruction PING

Function Read the working state of the servo
Length 0X02
Instruction 0X01
Parameter no
The PING command uses the broadcast address, and the steering gear also returns the response information.

Example 1 reads the working state of the steering gear with ID number 1

Instruction frame: FF FF 01 02 01 FB `(sent in hexadecimal)

initial	ID	Effective data length	instruction	Check Sum
0XFF 0XFF	0X01	0X02	0X01	0XFB

Data frame returned: FF FF 01 02 00 FC (hexadecimal display)

initial	ID	Effective data	working	Check

		length	condition	Sum
0XFF 0XFF	0X01	0X02	0X00	0XFC

1.3.2 READ DATA

Function reads data from the servo memory control table

Length * 0X04

Instruction 0X02

Parameter 1. Head address of read-out segment of data

Parameter 2. Length of read data

Example 2 Read the current position of the servo with ID 1 (low byte before, high byte after).

Two bytes are read from address 0X38 in the control table.

Instruction frame: FF FF 01 04 02 38 02 BE (sent in hexadecimal)

initial	ID	Effective data length	instruction	Parameter	Check Sum
0XFF 0XFF	0X01	0X04	0X02	0X38 0X02	0XBE

Data frame returned: FF FF 01 04 00 18 05 DD (hexadecimal display)

initial	ID	Effective data length	working condition	Parameter	Check Sum
0XFF 0XFF	0X01	0X04	0X00	0X18 0X05	0XDD

Read out two byte data: low byte L 0X18 high byte H 0X05

Two-byte synthesis of 16-bit data 0X0518, using decimal representation of the current location of 1304.

1.3.3 WRITE DATA

Function. Write data to the servo memory control table

Length N+3 (N is the parameter length)

Instruction 0X03

Parameter 1. Head address of data write segment

Parameter 2 The first data written

Parameter 3. Second data

Parameter N+1 Number N Data

Example 3 sets an ID of any number to 1.

The address of ID number is 5 in the control table, so write 1 at address 5. The ID of the sending instruction package uses the broadcast ID (0xFE).

Instruction frame: FF FF FE 04 03 05 01 F4 (sent in hexadecimal)

initial	ID	Effective data length	instruction	Parameter	Check Sum
0XFF 0XFF	0XFE	0X04	0X03	0X05 0X01	0XF4

Because broadcasting ID is used to send instructions, there will be no data return. In addition, the memory table EPROM has a protective lock switch, which needs to be turned off before modifying the ID, otherwise the sample ID number will not be saved when power is off. For detailed operation, please refer to the memory table or operation manual of the specific steering gear type.

Example 4 controls the ID1 servo to rotate to 2048 at a speed of 1000 seconds.

In the control table, the first address of the target location is 0X2A, so six consecutive bytes of data are written at the address 0X2A, namely position data 0X0800 (2048), time data 0X0000 (0), speed data 0X03E8 (1000). The ID of the sending instruction package uses a non-broadcast ID (0xFE), so the servo will return to the status package when the instruction is received.

Instruction frame: FF FF 01 09 03 2A 00 08 00 E8 03 D5 (sent in hexadecimal)

Instruction frame: FF FF 01 09 03 2A 00 08 00 E8 03 D5 (sent in hexadecimal)

initial	ID	Effective data length	instruction	Parameter	Check Sum
0XFF 0XFF	0X01	0X09	0X03	0X2A 0X00 0X08 0X00 0X00 0XE8 0X03	0XD5

Data frame returned: FF FF 01 02 00 FC (hexadecimal display)

initial	ID	Effective data length	working condition	Check Sum
0XFF 0XFF	0X01	0X02	0X00	0XFC

The return working state is 0, indicating that the servo has received the instructions correctly and correctly and has begun to execute them.

1.3.4 REG WRITE

The REG WRITE instruction is similar to the WRITE DATA except that the execution time is different. When the REG WRITE instruction frame is received, the received data is stored in the buffer reserve and the Registered Instruction Register is set at 1. When the ACTION instruction is received, the stored instruction is finally executed.

Length N+3 (N is the number of data to be written)

Instruction 0X04

Parameter 1. The header address of the area where the data is to be written

Parameter 2. The first data to be written

Parameter 3. The second data to be written

Parameter N+1 The Nth Data to Write

Example 5 Control ID1 to ID10 servo to rotate to 2048 position at 1000 per second. Only ID in the following instruction package receives instructions on the bus and returns. Other ID numbers are not returned on the bus.。

ID 1 Asynchronous Write Instruction Pack: FF FF 01 09 04 2A 00 08 00 00 E8 03 D4
ID 1 Retun Pack: FF FF 01 02 00 FC
ID 2 Asynchronous Write Instruction Pack: FF FF 02 09 04 2A 00 08 00 00 E8 03 D3
ID 3 Asynchronous Write Instruction Pack: FF FF 03 09 04 2A 00 08 00 00 E8 03 D2
ID 4 Asynchronous Write Instruction Pack: FF FF 04 09 04 2A 00 08 00 00 E8 03 D1
ID 5 Asynchronous Write Instruction Pack: FF FF 05 09 04 2A 00 08 00 00 E8 03 D0
ID 6 Asynchronous Write Instruction Pack: FF FF 06 09 04 2A 00 08 00 00 E8 03 CF
ID 7 Asynchronous Write Instruction Pack: FF FF 07 09 04 2A 00 08 00 00 E8 03 CE
ID 8 Asynchronous Write Instruction Pack: FF FF 08 09 04 2A 00 08 00 00 E8 03 CD
ID 9 Asynchronous Write Instruction Pack: FF FF 09 09 04 2A 00 08 00 00 E8 03 CC
ID10 Asynchronous Write Instruction Pack: FF FF 0A 09 04 2A 00 08 00 00 E8 03 CB

1.3.5 Executing Asynchronous Write Instruction ACTION

Function	trigger REG WRITE instruction
Length	0X02
Instruction	0X05
Parameter	no

ACTION instructions are very useful for controlling multiple servos at the same time.

When controlling multiple servos, the ACTION command enables the first and last servos to perform their respective actions simultaneously without delay.

When the action command is sent to multiple servos, the broadcast ID (0xFE) is used, so no data frame will be returned when the command is sent.

Example 6: After issuing the asynchronous writing instructions that control ID1 to ID10 servo to rotate at 2048 position at a speed of 1000 seconds, the following instruction packages (FF FF FE 02 05 FA) need to be sent when the asynchronous writing instructions need to be executed. All servos on the bus receive this instruction and run the asynchronous writing instruction received before.

1.3.6 SYNC WRITE

Function used to control multiple servos
ID 0XFE
Length (L + 1) * N + 4 (L: Length of data sent to each servo, N: Servo Number)
Instruction 0X83
Parameter 1 Head address of write data
Parameter 2 Length of write data(L)
Parameter 3 First servo Number
Parameter 4 Write the first data of the first servo
Parameter 5 Write the L data of the first servo
...
Parameter L+3 Write the second data of the first servo**Parameter L+4**
The second Servo ID number
Parameter L+5 Write the first data of the second servo
Parameter L+6 Write the second data of the second servo
...
Parameter 2L+4 Write the L data of the second servo
...

Unlike the REG WRITE + ACTION instruction, the real-time performance is higher. A SYNC WRITE instruction can modify the control table contents of multiple servos at one time, while the REG WRITE + ACTION instruction can be implemented step by step. Nevertheless, when using SYNC WRITE instructions, the length of the data written must be the same as the first address of the data saved.

Example7 Writing position 0X0800 time 0X0000 and speed 0X03E8 for ID1-ID4 with four servo header addresses 0X2A (low byte in front, high node in back)。
Instruction frame : FF FF FE 20 83 2A 06 01 00 08 00 00 E8 03 02 00 08 00 00 E8 03 03 00 08 00 00 E8 03 04 00 08 00 00 E8 03 58 (Send in hexadecimal)

initial	ID	Effective data length	instructions	Parameter	Check Sum
0XFF 0XFF	0XFE	0X20	0X83	0X2A 0X06 0X01 0X00 0X08 0X00 0X00 0XE8 0X03 0X02 0X00 0X08 0X00 0X00 0XE8 0X03	0X58

				0X03 0X00 0X08 0X00 0X00 0XE8 0X03 0X04 0X00 0X08 0X00 0X00 0XE8 0X03	

Because broadcasting ID is used to send instructions, no data is returned.。

1.3.7 RESET Instruction

Function Reset the specific data in the memory control table (specific Servo type is used)

length 0X02

Instruction 0X06

Parameter NO

For example reset servo, ID number is 0。

Instruction frame: FF FF 01 02 06 F6 (Send in hexadecimal)

initial	ID	Effective data length	instructions	Check Sum
0XFF 0XFF	0X00	0X02	0X06	0XF7

Returned data frame: FF FF 01 02 00 FC （Send in hexadecimal）

initial	ID	Effective data length	working condition	Check Sum
0XFF 0XFF	0X01	0X02	0X00	0XFC