2

Physical layer description of CAN

CAN Protocol Overview

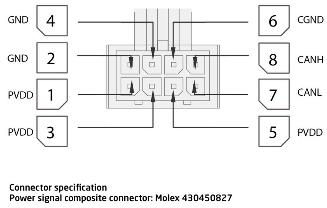
- CAN is an abbreviation of Controller Area Network (hereafter referred to as CAN) and is an ISO international standardized serial communication protocol.
- INNFOS product design is subject to the CAN2.0A protocol standard. This article details the CAN communication protocol format of the company's products and the CAN communication structure of the product.

Comprehensive performance parameters

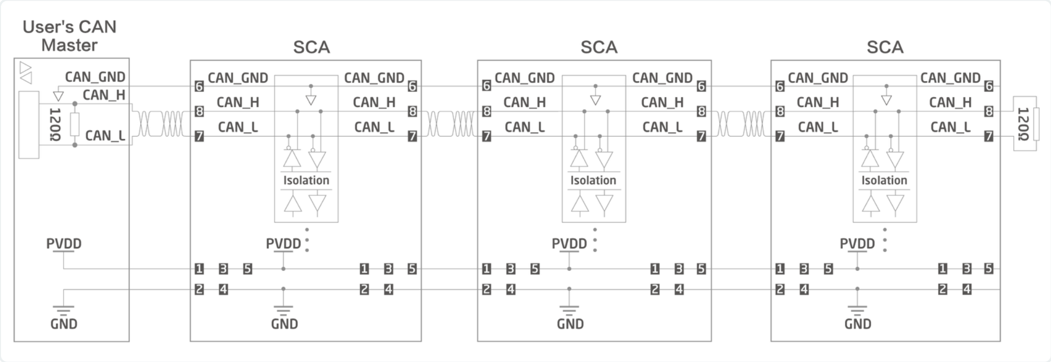
Table 1 Description of the comprehensive performance parameters	
project	Description
Link layer protocol	CAN bus
CAN-ID type	11bit-CAN2.0A
Baud rate	1Mbit/s
Maximum number of sites	63
CAN frame length	0~8 bytes
Application layer CAN frame type	Data frame, remote frame
Terminal matching resistor	120Ω

The baud rate of this communication protocol is 1Mbit/s. For CAN communication, the cable types have little effect on the transmission distance, but the wire diameter is as thick as possible. The maximum number of nodes is 64. The company's products use 0.205mm2 wire diameter.The largest transmission distance is 25m.

Hardware connection introduction of CAN

Table 2 Communication Signal Connector Pin Definitions			
Pin number	definition	description	Terminal pin distribution
1	PVDD	Power supply	 (pages/../img/配线2-2.png)
3	PVDD	Power supply	
5	PVDD	Power supply	
2	GND	Power Ground	
4	GND	Power Ground	
6	CGND	CAN Ground	
7	CANL	CAN communication interface	
8	CANH	CAN communication interface	

CAN communication bus and multi-node connection



(pages/../img/wiring2-3.png)

Fig1 Connection diagram of CAN communication

(pages/../img/wiring2-4.png)

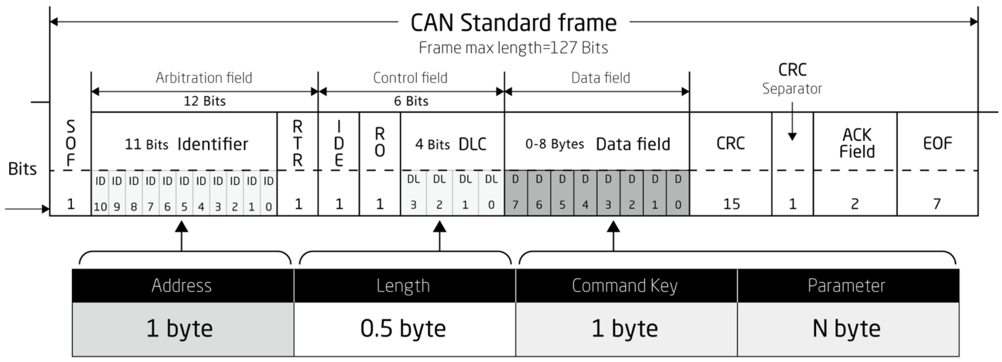
Fig2 Connection diagram of CAN communication

Note: CAN communication is an electrical level signal transmitted by the voltage difference between CAN_H and CAN_L.
If there is no isolation, in some cases, the voltage on the CAN bus can be well above the specification and is likely to damage most of the devices on the entire bus.
The SCA (except for special models) is internally designed with CAN isolation circuit to ensure the safety during communication.

- It is recommended to use shielded twisted pair connection,two 120Ω termination matching resistors should be connected at both ends of the bus in order to prevent signal reflection.
- Use a multimeter to measure the resistance between CANH and CANL to confirm that the resistance at both ends of the field is correctly connected. The normal resistance value should be at about 60Ω (the parallel value of the two resistors).
- The maximum number of attached devices is 64.

communication protocol

communication protocol format



(pages/../img/wiring2-13.png)

Fig 3

Figure 3: Device address corresponding identifier bit, CAN bus standard data frame identifier bit is 11 bits, this protocol uses only 8 of them, just occupies one byte, data length corresponds to DLC, occupies a half byte. The contents of the instruction parameter are also located in data field. The content of the instruction located at front followed by the parameter, the high byte located at front followed by the low byte.The data length is equal to the instruction character plus the parameter content

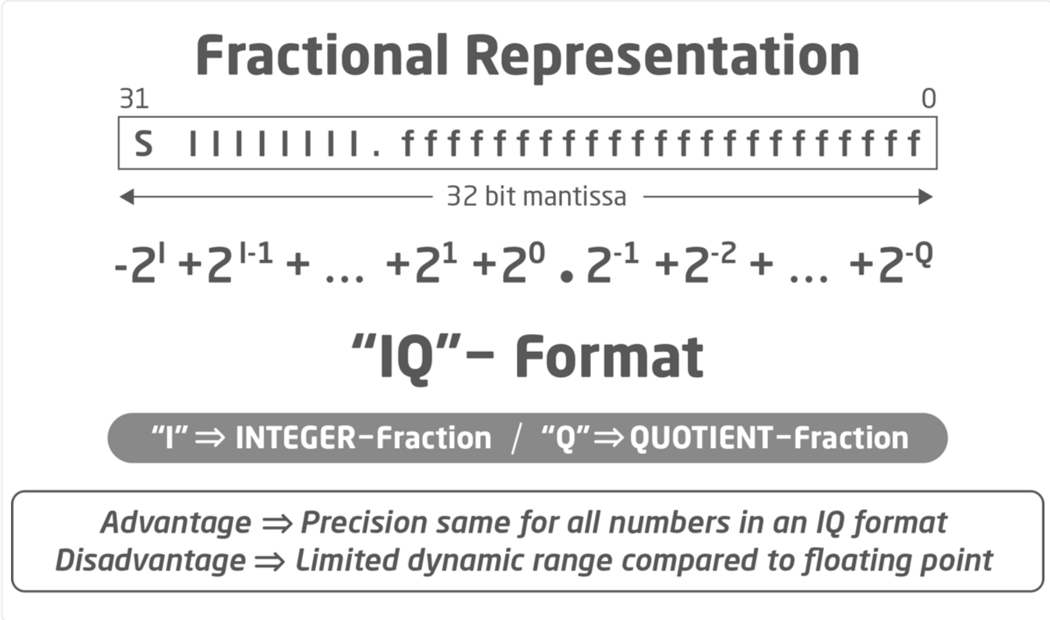
Device Address
A byte of data， identifies the address of the device to communicate with, 0x01 to 0xff are available. 0x00 is the broadcast address.

The length of data
Half a byte of data, identifies the number of specific data to be communicated. The range is 0x00 to 0x0F. The data out of range will not be processed.

Instruction byte
A byte of data， identifies the specific operation performed by the master and the slave. The value ranges from 0x00 to 0xff.

Parameter content
The specific parameter content of an instruction whose length is equal to the data length minus one. Some instructions do not contain specific data and should have a data digit of 1.

IQmath Instruction



(pages/../img/3-1communicationprotocol.png)

Fig 4

Data Type	**Range**		**Resolution/Precision**
	Min	Max	
_iq30	-2	1.999 999 999	0.000 000 001
_iq29	-4	3.999 999 998	0.000 000 002
_iq28	-8	7.999 999 996	0.000 000 004
_iq27	-16	15.999 999 993	0.000 000 007
_iq26	-32	31.999 999 985	0.000 000 015
_iq25	-64	63.999 999 970	0.000 000 030
_iq24	-128	127.999 999 940	0.000 000 060
_iq23	-256	255.999 999 981	0.000 000 119
_iq22	-512	511.999 999 762	0.000 000 238
_iq21	-1024	1023.999 999 523	0.000 000 477
_iq20	-2048	2047.999 999 046	0.000 000 954
_iq19	-4096	4095.999 998 093	0.000 001 907

_iq18	-8192	8191.999 996 185	0.000 003 815
_iq17	-16384	16383.999 992 371	0.000 007 629
_iq16	-32768	32767.999 984 741	0.000 015 259
_iq15	-65536	65535.999 969 482	0.000 030 518
_iq14	-131072	131071.999 938 965	0.000 061 035
_iq13	-262144	262143.999 877 930	0.000 122 070
_iq12	-524288	524287.999 755 859	0.000 244 141
_iq11	-1048576	1048575.999 511 719	0.000 488 281
_iq10	-2097152	2097151.999 023 437	0.000 976 563
_iq9	-4194304	4194303.998 046 875	0.001 953 125
_iq8	-8388608	8388607.996 093 750	0.003 906 250
_iq7	-16777216	16777215.992 187 500	0.007 812 500
_iq6	-33554432	33554431.984 375 000	0.015 625 000
_iq5	-67108864	67108863.968 750 000	0.031 250 000
_iq4	-134217728	134217727.937 500 000	0.062 500 000
_iq3	-268435456	268435455.875	0.125 000 000

		000 000	
_iq2	-536870912	536870911.750 000 000	0.250 000 000
_iq1	-1073741824	1 073741823.500 000 000	0.500 000 000

Note: _iq24 is INNFOS main application.

- In general,the processor we used only supports hardware floating-point arithmetic directly, such as some devices with FPU, or just supports fixed-point arithmetic. In this case, the processing of floating-point numbers needs to be done by the compiler. On devices that support hardware floating-point processing, the quickest way to program floating-point operations is to use floating-point types directly, such as single-precision floats. However, in many cases, limited to cost, material and other factors, when we use only one fixed-point processor, if directly using the float type for floating-point type operations,even a simple floating-point arithmetic will make the compiler to generate a lot of code. It will cause a significantly increase of the program excution time, and the resources occupied will mutiply. This involves the problem of how to efficiently handle floating-point operations on fixed-point processors.
- Since it is a fixed-point processor, it is much more efficient at processing fixed-point numbers, or literal “integers” than it is dealing with floating-point types. So on fixed-point processors, we use fixed-point integers to represent a floating-point number, and specify integer digits and scales to easily convert fixed-point and floating-point numbers. Taking a 32-bit fixed-point number as an example, suppose the conversion factor is Q, that is, the number of decimal places in 32 bits is Q, and the number of integer digits is 31-Q (in the case of signed numbers). The conversion relationship of fixed-point numbers and floating-point numbers is:

references:

C28x_IQmath_Library.pdf (pages/../img/C28x_IQmath_Library.pdf)

IQ-MATH_Library.pdf (pages/../img/IQ-MATH_Library.pdf)

Fixed point number = floating point number × 2^Q

For example, when the floating point number -2.0 is converted to a fixed point number with Q of 24, the result is: fixed point number = -2 × 2^24 = -33554432.

The range of representation of the 32-bit signed number is: -2147483648 to 2147483647. If we convert the maximum value of 2147483647 of the signed fixed point number to a floating point number corresponding to Q, the result is: floating point number 2147483647/2^24=127.999999940.

- See Appendix D for specific methods of IQ value conversion.

CAN communication protocol command application example

Example 1. Write command

Read the current speed value of the actuator motor ID 0x01			
Device address	Data length	CMD	parameter
0x01	0x01	0x05	NO

Device address: 0x01 = Object ID

Data length: 0x01 = Data length

Instruction character: 0x05 = Current speed command read

Parameter content: None = Parameter content sent

Content: 0x05

Answer command			
Device address	Data length	Command Key	CMD
0x01	0x05	0x05	data[3~0]

Return device address: 0x01 = Response object ID

Data length: 0x05 = Data length of the response 5 bits

Instruction character: 0x05 = Acknowledge the current speed command (same as the send command)

Parameter content: 0xXX 0xXX 0xXX 0xXX = Parameter content of the response

Response content :0x05 0xXX 0xXX 0xXX 0xXX

Description: The parameter content data[3~0] is in the high position and the low bit is in the back. For the _IQ24 format. _IQ(-1.0)~_IQ(1.0) represents the reverse speed full scale and forward speed full scale. The full scale is 6000 RPM. If data = _IQ (0.5).
Then it is 0.5*6000=3000RPM.

Example 2. Write command

Set the current position value of the actuator motor ID to 0x01			
Device address	Data length	CMD	parameter
0x01	0x5	0x0A	0x05 0x00 0x00 0x00

Device address: 0x01 = Set object ID
Data length: 0x5 = Data length
Instruction character: 0x0A = Set position
Parameter: 0x05 0x00 0x00 0x00 = Parameter content sent
Send content: 0x0A 0x05 0x00 0x00 0x00

Description: Parameter content data[3 ~0] The high position is in the front and the low position is in the back. For the _IQ24 format. _IQ(-128.0)~_IQ(127.999999940) represents the reverse position value full scale and forward position value full scale. If data=_IQ(5.0), set the current position to 5.

- Respons: no response (special agreement)

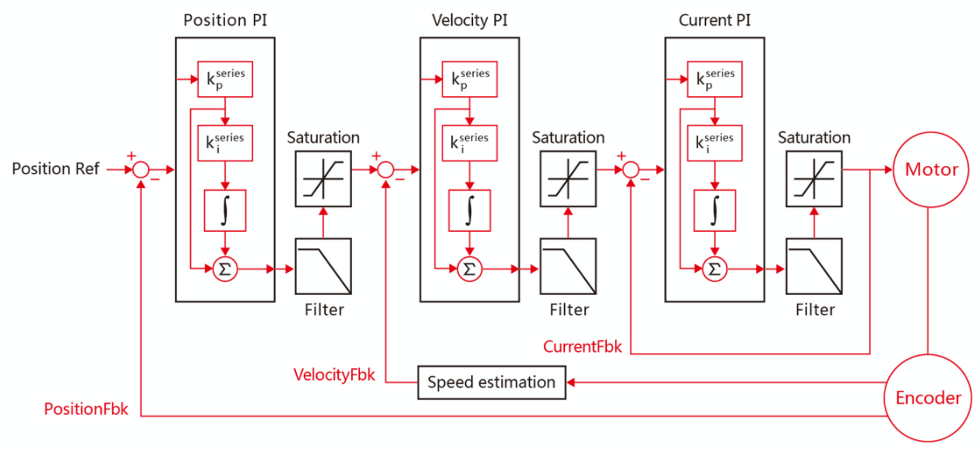
General steps for using the command mode:

(1) SCA enable, command: 0x2A 01;

(2) Select the usage mode, command: current loop 0x07 01, speed loop 0x07 02, position loop 0X07 03, S-position mode 0X07 06, S-speed mode 0X07 07, HOMING mode 0X07 08;

(3) Set the relevant parameters, the instruction refers to the appendix of the manual. For the current value, speed value, position parameter, it is not 0, then it starts, and if it is 0, it stops;

(4) End of use, SCA disable, command: 0X2A 00. The disable command must be sent before the power is turned off, otherwise the zero position may be lost.



(pages/../img/position.jpg)

图5

Note:

- The maximum amplitude (Maximum) set by the host computer is _IQ (1.0), and the minimum amplitude (Minimal) is _IQ (-1.0), which acts as a limiting function (the schematic diagram is shown in Figure 3-2). Example: (Figure: 3-2 uses the output of the position loop through the limiter module as the input of the speed loop. Assuming _IQ(0.5), _IQ(-0.5), the maximum speed of the position loop output should be ±0.5x6000=±3000RPM)
- The upper limit of the proportional integral setting is _IQ (127.999999940), and the lower limit of the set value is _IQ (-128.0), but the set value is adjusted according to the actual operation (schematic diagram: 5)

- The current loop sets the current value range from _IQ(-1.0)~_IQ(1.0). The actual current value is the IQ value multiplied by the full scale. Example: (QDD-PR60 model actuator full-scale current value is 33A, _IQ (0.5) The actual current value is 0.5x33A=16.5A, See the SCAs parameter list)
- The speed loop sets the speed value range from _IQ(-1.0)~_IQ(1.0), and the actual speed value is the IQ value multiplied by the full scale (see). Example: (_IQ (0.5) the actual speed is 0.5 * 6000 = 3000RPM)
- Since the position loop is in _IQ24 format, the forward full scale is _IQ (127.999999940), the reverse full scale is _IQ (-128.0), and the IQ value is the actual value. For example: (_IQ(60.0) is the actual position. 60R, that is, the position where the zero position is rotated forward by 60 turns.)
- Speed loop curve mode and position loop curve mode, you can set the acceleration, deceleration, and relatively smooth to reach your preset speed value and position, which can avoid excessive current during operation, trigger actuator over-current protection or Power supply overcurrent protection.

CAN communication protocol command reference

Read Command

3.3.1.1 Send data 1 byte, return data 2 byte	
Command Name	Read Command
description	This command sends a data length of 1, and its return data with the length of 2
Command byte	See Read Command 1
Data length	1
Data content	No
Command byte（return value）	See Read Command 1
Data length （return value）	2

Lower machine returns data	0x01：Success / enable / normal	Pattern query return data see mode table
	0x00：Failed / disabled / abnormal	

3.3.1.2 Send data 1 byte, return data 3 bytes		
Command name	Read command	
Description	This command sends a data length of 1, and its return data with the length of 3. The parameter in high position shows first. The value is 2^8 times the true value. (special instructions are specified in the instruction list)	
Command character	See Read Command 2	
Data length	1	
Data content	None	
Command character (return value)	See Read Command 2	
Data length (return value)	3	
Lower machine returns data	The data is in iq8 format	Or see the error warning instruction list

3.3.1.3 Send data 1 byte, return data 5 bytes	
Command name	Read command

Description	This command sends a data length of 1, and its return data with the length of 5. the parameter in high position shows first. The value is 2^24 times the true value (special instructions are specified in the instruction list)
Command character	See Read Command 3
Data length	1
Data content	none
Command character (return value)	See Read Command 3
Data length (return value)	5
Lower machine returns data	Data is formed as iq24(special instructions are specified in the instruction list)

Write command

3.3.2.1 Send data 2 byte, return data 2 byte	
Command name	Write command
Description	This command class sends a data length of 2, a return data length is 2, and a byte after the data is sent indicates that the parameter content is to be written. (note: a power-on command should be sent to use and a power-off command should be sent before powering off, otherwise the zero position may be lost.)

<table>
<tr><td>Command character</td><td colspan="2">See Write Command 1</td></tr>
<tr><td>Data length</td><td colspan="2">2</td></tr>
<tr><td rowspan="2">Data content</td><td>0x01：enable/ power on</td><td rowspan="2">The mode setting is mode table</td></tr>
<tr><td>0x00：disable/power off</td></tr>
<tr><td>Data length (return value)</td><td colspan="2">See Write Command 1</td></tr>
<tr><td>Data length (return value)</td><td colspan="2">2</td></tr>
<tr><td rowspan="2">Lower machine returns data</td><td colspan="2">0x01： Success</td></tr>
<tr><td colspan="2">0x00： Failed</td></tr>
</table>

3.3.2.2 Send data 3 byte, return data 2 byte	
Command name	Write command
Description	The data length of this command is 3 bytes, the return data length is 2 bytes, and the sent data 1 byte at last indicate that the parameter content is to be written, and the high parameter shows in front. The value is 2^8 times the true value.
Command character	See Write Command 2
Data length	3
Data content	The value is formed IQ8
	0x00： Disable/power off
Instruction (return value)	See Write Command 2

Data length (return value)	2
IAS return data	0x01：Success
	0x00：Fail

3.3.2.3 Send data 5 byte, return data is 2 byte or less	
Command name	Write Command
Description	The data length of this command is 5 bytes, the return data length is 1 byte, and the sent data 4 bytes at last indicate that the parameter content is to be written. The value is 2^24 times the true value. (Special instructions are specified in the instruction list)
Command byte	See Write Command 3
Data content	5
Data content	The value is formed IQ24 (Special instructions are specified in the instruction list)
Instruction (return value)	See Write Command 3
Data length (return value)	2or0
Lower machine returns data	0x01：Success 0x00：Fail (There are no return data for the three special instructions, which are marked in the instruction list)

3.3.2.4 Send data 1 byte, return data is 2 byte	
Command name	Write Command

Description	This command class sends a data length of 0, and returns a data length of 1
Command byte	See Write Command 4
Data length	1
Data content	None
Command (return value)	See Write Command 4
Data length(return value)	2
Lower machine returns data	0x01： success
	0x00： fail

Appendix A

A.1 Write command code value definition table

Read Command 1:

A.1.1 Read Command 1		
Command key	Definition	Description
0x00	shake hands	The PC sends command, and the slave computer responds to it, indicating that the slave computer is ready to communicate with the host computer. It can also be used as a heartbeat protocol to query the status of the slave in real time.
0x55	inquiry actuator current mode	Read the current mode of the actuator managed by the lower computer.

0xB0	Query the last shutdown state of the actuator	Read the last shutdown state of the actuator, normal / abnormal
0x71	Current loop filter status	Read current loop filter enable/disable for the specified ID actuator.
0x75	Speed loop filter status	Read speed loop filter enable/disable for the specified ID actuator
0x79	Position loop filter status	Read Position loop filter enable/disable for the specified ID actuator
0x2B	Actuator enable/disable	Read enable/disable status for the specified ID actuator

Read Command 2:

>

A.1.2 Read Command 2		
Command key	Definition	Description
0x73	Current loop filter bandwidth	Read the bandwidth (Hz) of the specified ID actuator current loop filter
0x77	Speed loop filter bandwidth	Read the bandwidth (Hz) of the specified ID actuator speed loop filter
0x7B	Position loop filter bandwidth	Read the bandwidth (Hz) of the specified ID actuator Position loop filter
0x6C	Actuator motor protection temperature	Read motor protection temperature of actuator with the specified ID

0x6E	Actuator motor recovery temperature	Read motor recovery temperature of actuator with the specified ID
0x62	Actuator inverter protection temperature	Read inverter protection temperature of actuator with the specified ID
0x64	Actuator inverter recovery temperature	Read inverter recovery temperature of actuator with the specified ID
0xFF	Alarm command (special command)	Lower machine alarm information

Read Command 3:

A.1.3 Read Command 3		
Command key	Definition	Description
0x04	current value	Read the current current value of the specified ID actuator. The current value needs to be multiplied by the current full scale (See the SCAs parameter list). The unit is A.
0x05	Current speed value	Read the current speed value of the specified ID actuator. The true speed value needs to be multiplied by the speed full scale (See the SCAs parameter list) in RPM.
0x06	Current position value	Reads the current position value of the specified ID executor in R
0x17	Speed loop P	Read the current speed loop P of the specified ID actuator
0x18	Speed loop I	Read the current speed loop I of the specified ID actuator

0x19	position loop P	Read the current position loop P of the specified ID executor
0x1A	position loop I	Read the current position loop I of the specified ID executor
0x1C	Max speed of position trapezoidal curve	Reads the maximum speed of the trapezoidal curve of the current position of the specified ID actuator
0x1D	Acceleration of position trapezoidal curve	Reads the maximum acceleration of the trapezoidal curve at the current position of the specified ID actuator
0x1E	Deceleration of position trapezoidal curve	Read the maximum deceleration of the trapezoidal curve of the current position of the specified ID actuator
0x22	Max speed of the speed trapezoidal curve	Reads the maximum speed of the current speed trapezoidal curve of the specified ID actuator
0x23	Acceleration of velocity trapezoidal curve	Reads the maximum acceleration of the current speed trapezoidal curve of the specified ID actuator
0x24	Deceleration of the speed trapezoidal curve	Reads the maximum deceleration of the current speed trapezoidal curve of the specified ID actuator
0x34	Lower limit of current loop output	Read the lower limit of the current current loop output of the specified ID actuator
0x35	Upper limit of current loop output	Read the upper limit of the current current loop output of the specified ID actuator

0x36	Lower limit of the speed loop output	Read the lower limit of the current speed loop output of the specified ID actuator
0x37	Upper limit of speed loop output	Read the upper limit of the current speed loop output of the specified ID actuator
0x38	Lower limit of position loop output	Read the lower limit of the current position loop output of the specified ID actuator
0x39	Upper limit of position loop output	Read the upper limit of the current position loop output of the specified ID actuator
0x85	Upper limit of actuator position	read upper limit of actuator position
0x86	Lower limit of actuator position	Read the lower limit of the position of the execution ID actuator
0x8A	Actuator position offset	Read the position offset value of the specified ID actuator
0x92	The lower limit of the current when the actuator is automatically reset to zero	Read the lower limit of the current when the specified ID actuator is automatically reset to zero
0x93	The upper limit of the current when the actuator is automatically reset to zero	Read the upper limit of the current when the specified ID actuator is automatically reset to zero
0x7F	Stall energy	Reads the stall energy of the specified ID actuator. （The value is

		75.225 times the true value） The heating energy after blocking, the unit is J.

Read Command 4:

A.1.4 Read command 4		
Command byte	Definition	Description
0x94	SCA's current,velocity and position	Read the current,velocity and position SCA with the specified ID. The first 3 bytes equal to the position value. The values are 2^16 times the true values. The 2 bytes in the middle is velocity value. The values are 2^14 times the true values. The last 2 bytes are current value. The values are 2^14 times the true values.

A.2 Write command code value definition table

Write Command 1:

A.2.1 Write Command 1		
Command key	Definition	Description
0x07：	Set the mode of the specified ID executor	Set the current mode of the specified ID executor
0x70：	Current loop filter status	Set the current loop filter enable/disable for the specified ID actuator

0x74：	Speed loop filter status	Set the speed loop filter enable/disable for the specified ID actuator
0x78：	Position loop filter status	Set position loop filter enable/disable for the specified ID actuator
0x2A：	Actuator on/off status	Set the specified ID actuator to power on/off

Write Command 2:

A.2.2 Write Command 2		
Command key	Definition	Description
0x72：	Current loop filter bandwidth	Set the bandwidth (Hz) of the specified ID actuator current loop filter
0x76：	Speed loop filter bandwidth	Set the bandwidth (Hz) of the specified ID actuator speed loop filter
0x7A：	Position loop filter bandwidth	Set the bandwidth (Hz) of the specified ID actuator position loop filter
0x61：	Actuator protection temperature	Set the protection temperature of the specified ID actuator ℃
0x63：	Actuator recovery temperature	Set the recovery temperature of the specified ID actuator ℃

Write Command 3:

A.2.3 Write Command 3		

Command key	Definition	Description
0x08：	Current value	Set the current current value of the specified ID actuator (Note: no return data)
0x09：	Current speed	Sets the current speed value for the specified ID executor. (Note: no return data)
0x0A：	Current value	Sets the current speed value for the specified ID executor. (Note: no return data)
0x10：	Speed loop P	Set the P value of the specified ID actuator speed loop
0x11：	Speed loop I	Set the I value of the specified ID actuator speed loop
0x12：	Position loop P	Set the P value of the specified ID actuator position loop
0x13：	Position loop I	Set the I value of the specified ID actuator position loop
0x1F：	Maximum speed of position trapezoidal curve	Change the maximum speed of the trapezoidal curve of the specified ID actuator position
0x20：	Acceleration of position trapezoidal curve	Change the maximum acceleration of the trapezoidal curve of the specified ID actuator position
0x21：	Deceleration speed of position trapezoidal curve	Change the maximum deceleration of the trapezoidal curve of the specified ID actuator position

0x25：	Maximum speed of the speed trapezoidal curve	Change the maximum speed of the specified ID actuator speed trapezoid
0x26：	Acceleration of velocity trapezoidal curve	Change the acceleration of the specified ID actuator speed trapezoidal curve
0x27：	Deceleration of the speed trapezoidal curve	Change the deceleration of the specified ID actuator speed trapezoid
0x2E：	Lower limit of current loop output	Change the lower limit of the specified ID actuator current loop output
0x2F：	Upper limit of current loop output	Change the upper limit of the specified ID actuator current loop output
0x30：	Lower limit of the speed loop output	Change the lower limit of the specified ID actuator speed loop output
0x31：	Upper limit of speed loop output	Change the upper limit of the specified ID actuator speed loop output
0x32：	Lower limit of position loop output	Change the lower limit of the specified ID actuator position loop output
0x33：	Upper limit of position loop output	Change the upper limit of the specified ID actuator position loop output
0x83：	Upper limit of actuator position	Change the upper limit of the position of the specified ID actuator

0x84：	Lower limit of actuator position	Lower limit of actuator position
0x87：	Lower limit of actuator position	Set the Home value of the specified ID executor
0x89：	Actuator position offset	Set the position offset value of the specified ID actuator
0x90：	Current lower limit of SCA while automatically zeroed	Set the lower limit of the current when the specified ID actuator is automatically reset to zero.
0x91：	Current upper limit of SCA while automatically zeroed	Set the upper limit of the current when the specified ID actuator is automatically reset to zero.
0x7E：	Stall energy	Set the stall energy of the specified ID actuator. (The value is 75.225 times the true value)The heating energy after blocking, the unit is J

Write Command 4:

A.2.4 Write Command 4		
Command key	Definition	Description
0xFE：	Eliminate the alarm of the lower computer	Eliminate the alarm action of the lower position machine. After receiving the command, the lower position machine stops the alarm, otherwise the lower position machine is inoperable.
0x88	Clear Homing	Clear Homing data

	data	
0x0D	Storage parameter	Storage parameter

Appendix B :Mode table

Command key	Command key
0x01	Current mode
0x02	Speed mode
0x03	Position mode
0x06	Position trapezoidal mode (S curve)
0x07	Speed trapezoidal mode (S curve)
0x08	Homing mode

Appendix C： Alarm instruction list

Command character	Command character
0x0001	Overvoltage Error
0x0002	Undervoltage Error
0x0004	Abnormal blocking
0x0008	Overheating Error
0x0010	Read and Write Error
0x0020	Multi-turn count Error
0x0040	Inverter temperature sensor error
0x0080	communication is abnormal
0x0100	Motor tempuature sensor error

0x0200	Step is too big
0x0400	DRV protection Error
other	Device exception
explaination	Multiple errors can be alarmed at the same time. If the return data is 0X05, the error is 0X01 overvoltage abnormality and 0004 blocked abnormality.

Appendix D: Command Sending and IQ Value Conversion Method

- The comment section of the manual indicates that the IQ value in the position mode is the actual value, ranging from -128 to 127. 999999940. At this time, only the corresponding position value needs to be converted into an IQ value to be input into the parameter content. In the speed and current mode, the corresponding parameter value needs to be converted before converting the IQ value. If the current speed value is set to 100 RPM, the current value of the setting needs to be divided by the maximum value, ie 100/6000=0.01666666, and then Then, the IQ conversion is performed by 0.016666666, and the obtained value is the parameter value.
- For example, we need to set the current position to 60R (note the limit of the step response in the position mode. If the difference between the set position and the current position exceeds 1R, it will not respond), first find the corresponding command. The third type of write command (write command 3) in Appendix A indicates that the instruction to set the current position value is 0x0A. After finding the instruction, look for the corresponding transmission format of the instruction. In the "CAN Communication Protocol Command Reference", the 3.3.2.3 subsection corresponds to the third type of write instruction, and the transmission data length is 5, that is, one byte instruction + 4 bytes parameter. content. When the data content is applied to the IQ24 format, the IQ conversion is performed directly on 60, that is, 60*2^24=1006632960, and then unified into hexadecimal (according to the test software), 3C 00 00 00. According to the data frame format description of CAN bus, the instruction parameter should be at the highest position, and the content of the

parameter is after, the content of the instruction we send is 0x 0A 3C 00 00 00, which also corresponds to the length of the data in the description is 5 (bytes). The instruction is sent to this point.

- Correspondingly, if the current or speed mode settings commands need to be sent,the parameter values need to be convert firstly (divided by the corresponding maximum value), to get a number in the range of -1 to 1, and then perform IQ conversion. The steps and methods for sending commands are the same as the position mode. Note that the data format used by each command, if it's in IQ8 format, convert the 2^24 in the formula to 2^8 and then convert it.

Version Change Record

The following table briefly describes the version change record.

Version number	Update time	Change type	Position	Update content
V1.0.5	19.05.14	add	Appendix A	Temperature sensor and inverter temperature sensor read and write commands
		modify	Chapter II Wiring	Modify Chapter 2 Wiring Modify Figure 2-3
		modify	Chapter II Wiring	Modify Chapter 2 Wiring Modify Figure 2-3
		Delete	Appendix D	Delete Appendix D"Model Table"

V1.0.4	18.12.14	add	Appendix E	Command transmission and IQ value conversion method
V1.0.3	18.03.19	add	Chapter III Communication Protocol	More actuator parameter information instructions, actuator temperature information commands, query last shutdown status command, Homing command than the previous version.
		add	Chapter III Communication Protocol	Added storage parameter instruction
		modify	Chapter III Communication Protocol	Revised the layout of the third chapter
		add	Chapter III Communication Protocol	Added storage parameter instruction
V1.0.2	18.01.30	modify	all content	Micro servo changed its

				name to INNFOS actuator
		add	Chapter III Communication Protocol	More alarm commands than the previous version
V1.0.1	17.12.29	modify	Chapter III Communication Protocol	More alarm commands than the previous version
V1.0.0	17.12.15	modify	Chapter II wiring	Updated CAN interface definition
		add	Chapter III Communication Protocol	Added switch command