KST CAN Bus Servo User Manual

(Standard Frame)

Document modifications			
RN	**Date**	**Author**	**Modifications**
V1.01	2025-02-13	Lanmz	Creation;

Table of Contents

1.Instructions

1.1 Default value

This manual provides instructions only for CAN communication servos (using software version V3.72 as an example). Before operating the servo, ensure that the power supply and communication cables are properly connected. This document assumes the servo is configured with the Standard Frame format, Baud Rate: 500 kbps, and a default Node ID of 0x25 (valid range: 1–127; 0 represents the broadcast address).

1.2 Debugging Device Instructions

The CAN Analyzer (Chuangxin Technology), as shown in the figure below (Figure 1), is used for all debugging and configuration operations described in this manual in conjunction with the CANALYST-II analyzer and its test software. Before proceeding, verify in your computer's Device Manager that the CAN Analyzer driver is correctly installed. (Connect the servo's CANH and CANL wires to the ports marked with red circles in Figure 2.)

Figure 1

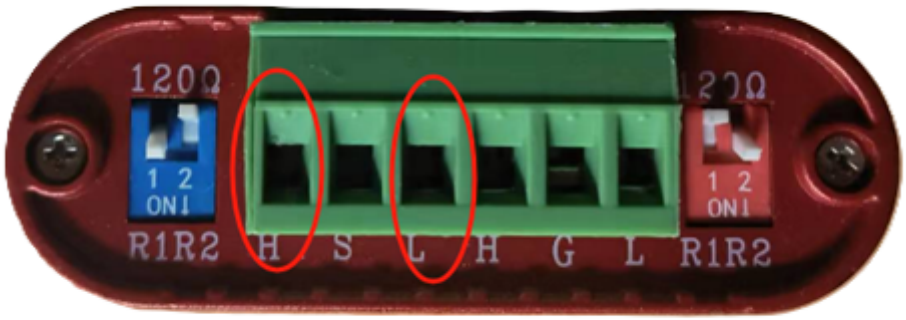

Figure 2

1.3 Command Frame Instructions

Send the corresponding communication commands to the servo via the host computer. Note that the Frame ID (HEX) format is: Command ID + Node ID

Example: The Frame ID for the Start Reporting Command is 0x00000000 + Node ID. If the Node ID of the servo to be configured is 0x25, the Frame ID should be 0x00000025.

A complete data frame consists of the Frame ID and the data payload. For details, refer to the document: **KST Servo CAN Bus Communication Protocol (Standard Frame) V3.72**.

All Frame IDs in subsequent sections assume a default Node ID of 0x25. The Frame ID format and calculation method follow the same rules as described above and will not be reiterated.

2.Servo wiring

2.1 CAN Bus Servo Harness Introduction

The servo has four cables in total for power and communication: DC+, DC-, CAN_H, and CAN_L.

DC+ connects to the positive terminal of the power supply.

DC- connects to the negative terminal of the power supply.

CAN_H connects to the bus's CAN_H line.

CAN_L connects to the bus's CAN_L line.

Note: Termination resistors are not pre-installed inside the servo. Users must install them based on the bus topology and specific application requirements.

2.2 Wiring Diagram See Figure 1.0.1 (taking HS30 servo as an example)

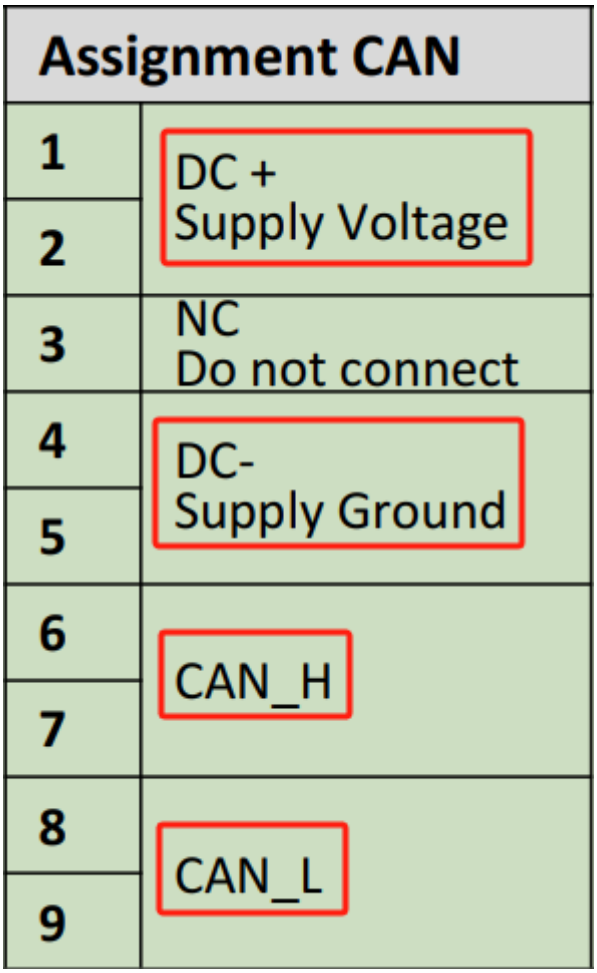

Figure 1.0.1

3.Servo configuration

3.1 Baud Rate Configuration

3.1.1: Send the corresponding command frame (Figure 1.0.2).

3.1.2: X is the baud rate setting byte, and the baud rate setting reference table is shown in Figure 1.0.2.

3.1.3: Receiving the corresponding reply frame indicates a successful setting (Figure 1.0.2).

3.1.4: Send the save command to save the configured baud rate. Note that the lower-level device does not need to respond to this command (Figure 1.0.3).

3.1.5: An example is provided to illustrate how to set the baud rate to 250K and save it (Figures 1.0.4-1.0.5).

Note: The default baud rate is 500K. After any changes, the save command must be used to save the settings, and the changes will only take effect after a restart.

2.6 Baud rate setting command

The default baud rate is 500K. Use the set command to set the baud rate to an optional value in the comparison table.

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	22	01	30	00	X	00	00	00

"X" is the baud rate setting byte. The baud rate setting comparison table is as follows:

X	00	01	02	03	04	05	06	07
Baud	20K	50K	100K	125K	250K	500K	800K	1000K

The following reply indicates that the operation is successful:

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 05 80 +NodeID	8	60	01	30	00	00	00	00	00

The baud rate change must be saved by using the Save command and will not take effect until it is restarted.

Figure 1.0.2

2.10 Save command

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 06 00 +NodeID	8	22	10	10	01	73	61	76	65

Save node number, baud rate, etc. (the lower computer does not need to answer)

Figure 1.0.3

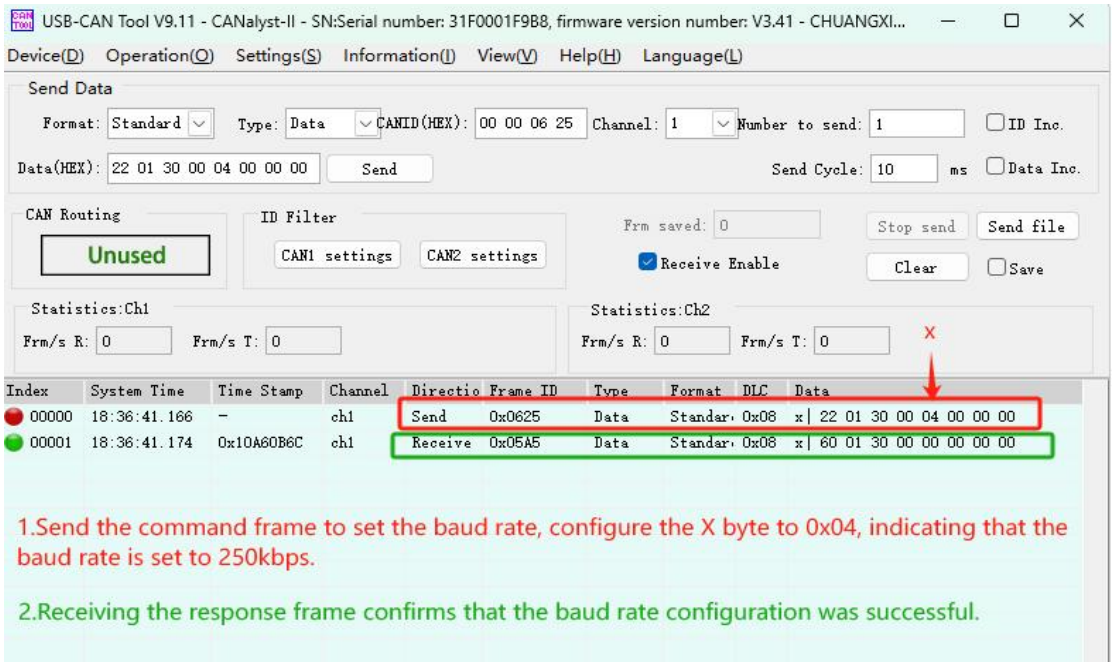

Figure 1.0.4

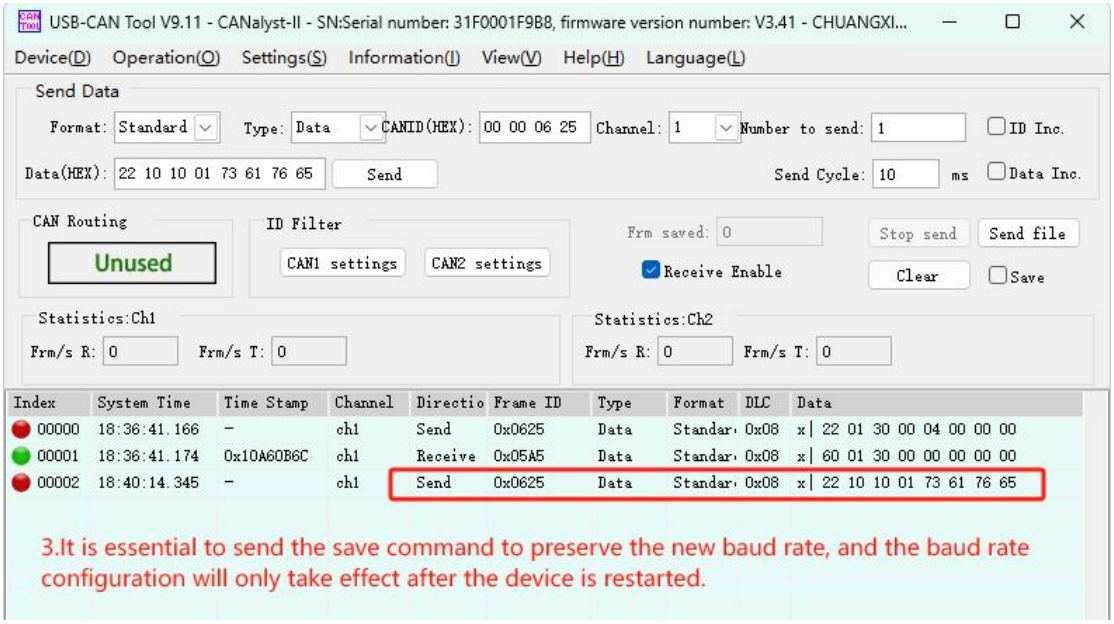

Figure 1.0.5

3.2 Node ID Configuration

3.2.1: Send the corresponding command frame (Figure 1.0.6).

3.2.2: X is the new node ID.

3.2.3: Receiving the corresponding reply frame indicates a successful setting (Figure 1.0.6).

3.2.4: Send the save command to save the configured node ID. Note that the lower-level device does not need to respond to this command.

3.2.5: An example is provided to illustrate how to set the node ID to 0x20 and save it (Figures 1.0.7-1.0.8).

Note: Any changes to the node ID must be saved using the save command, and the changes will only take effect after a restart.

2.7 Node number setting command

The default node number is 0x25.

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	22	00	30	00	X	00	00	00

"X" is the new node number.

The following reply indicates that the operation is successful:

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 05 80 +NodeID	8	60	00	30	00	00	00	00	00

The change of node number must be saved with the Save command and will not take effect until it is restarted.

Figure 1.0.6

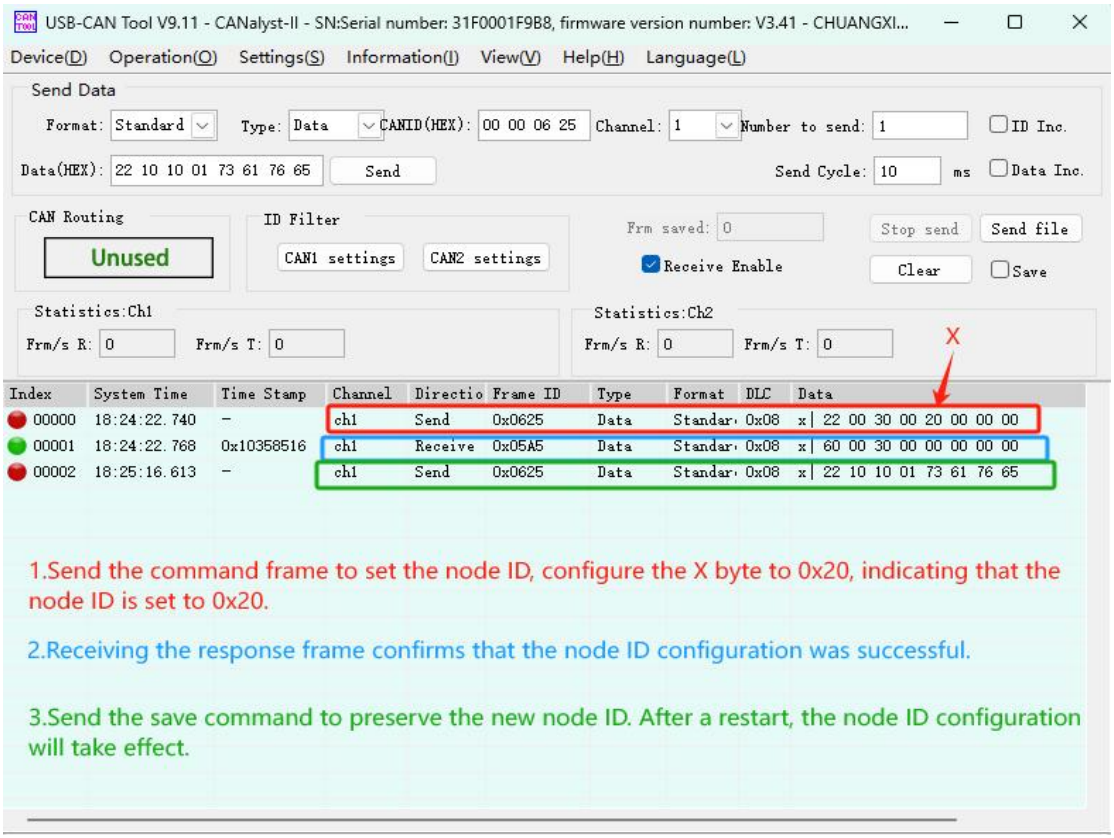

Figure 1.0.7

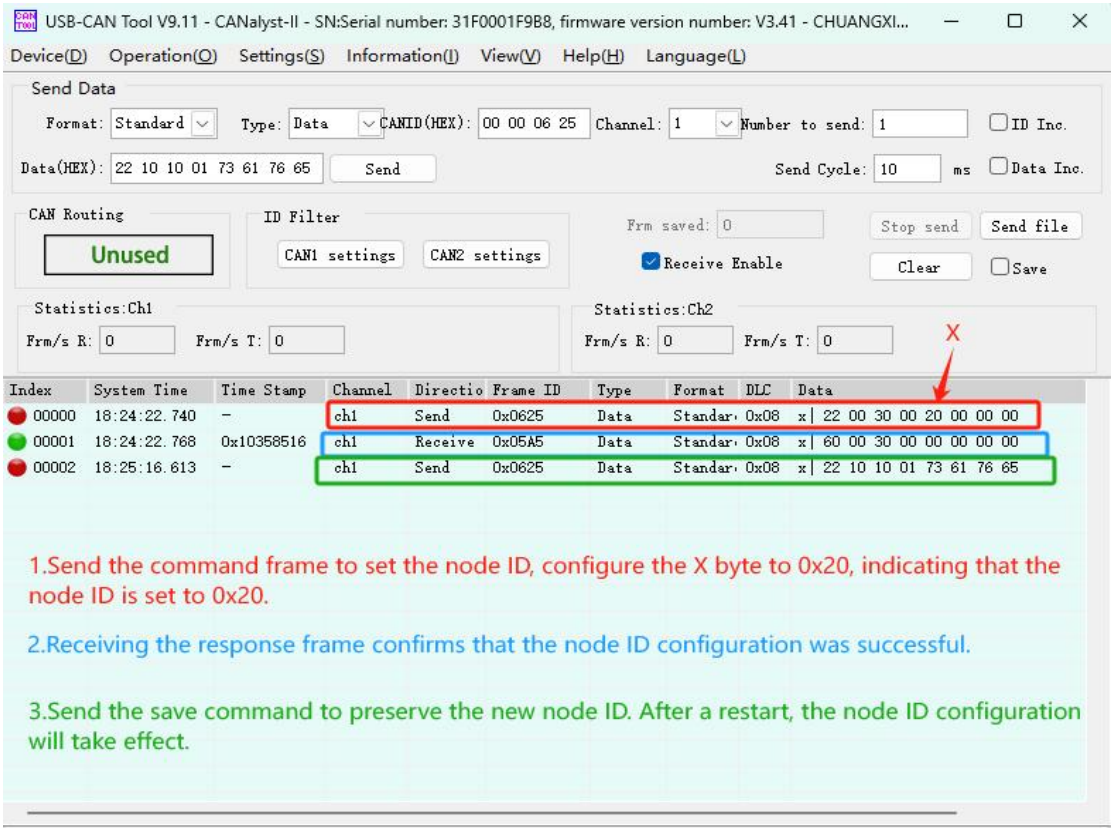

Figure 1.0.8

4.Servo function settings

4.1 Reporting Interval Configuration

4.1.1: Locate the corresponding command frame in the protocol.

4.1.2: The setting range is 10~255ms, where X is the hexadecimal number corresponding to the time interval to be set.

4.1.3: Receiving the corresponding reply frame indicates a successful setting (Figure 1.0.9). Please note the following:

1.After a successful setting, the interval change takes effect immediately.

2.If the change is not saved using the save command, it will be lost after a restart.

4.1.4: An example is provided to illustrate how to set the reporting interval to 100ms and save it (Figures 1.0.10~1.0.11).

2.5 Report interval setting command

The default reporting interval is 50ms. With the setting command, the interval can be set to 10 ~ 255ms.

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	22	00	22	00	X	00	00	00

"X" sets bytes for reporting interval. For example, 0x32 corresponds to 50ms, 0x64 to 100ms, and 0x0A to 10ms.

The following reply indicates that the operation is successful:

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 05 80 +NodeID	8	60	00	22	00	00	00	00	00

After the setting is successful, the interval change will take effect immediately. If the changes are not saved with the Save command, the changes will be lost after restart.

Figure 1.0.9

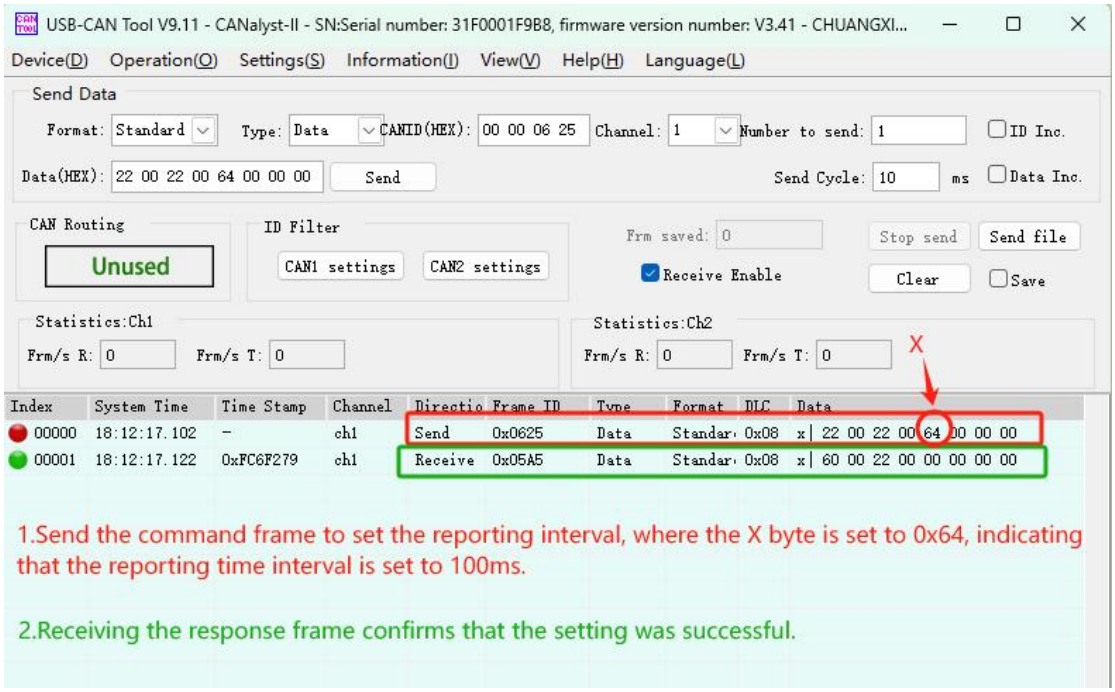

Figure 1.0.10

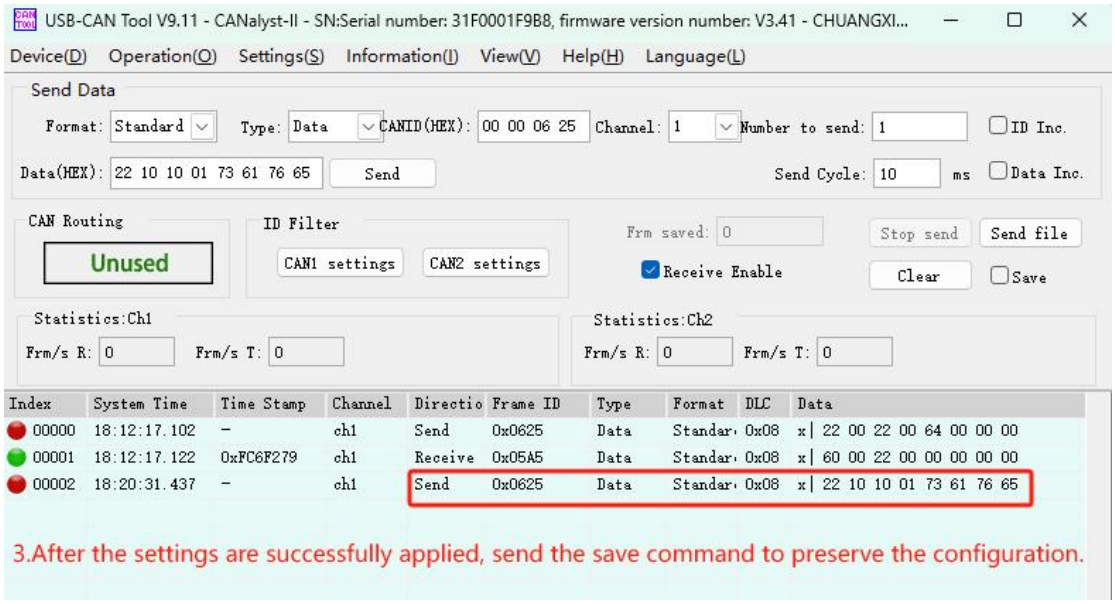

Figure 1.0.11

4.2 Motion Range Configuration

4.2.1: Locate the corresponding command frame in the protocol.

4.2.2: Command description: This function sets the motion range limit within ±100° (positive range: 0~+100°, negative range: 0~-100°, factory default range is ±100°). After the range setting command is successfully sent, the save command must be sent, and the set motion range will take effect after a restart. Once the motion range is successfully set, if the position command sent exceeds the servo's motion range, the servo will not move and will only return an error code in response.

4.2.3: Pay attention to the difference between the command frames for setting the positive range and the negative range (Figures 1.0.12~1.0.13).

4.2.4: The definitions of the High and Low bytes in the command are shown in Figure 1.0.14.

4.2.5: An example is provided to illustrate how to set the reporting interval to ±50° and save it(Figures1.0.15~1.0.16).

2.15.1 Setting negative range

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	22	0A	30	00	Low	High	00	00

Reply to the following command to indicate that the setting is successful

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 05 80 +NodeID	8	60	0A	30	00	00	00	00	00

Figure 1.0.12

2.15.2 Setting positive range

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	22	0B	30	00	Low	High	00	00

Reply to the following command to indicate that the setting is successful

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 05 80 +NodeID	8	60	0B	30	00	00	00	00	00

High and low bytes are the same as that of setting position command.

Figure 1.0.13

The position data precision that can be set is 0.1 °, the range is - 100 ° to + 100 ° and the negative value is expressed by complement code as follows:

Position	-100°	-50°	-10.9°	-0.8°	0°	0.8°	10.9°	50°	100°
High	0xFC	0xFE	0xFF	0xFF	0x00	0x00	0x00	0x01	0x03
Low	0x18	0x0C	0x93	0xF8	0x00	0x08	0x6D	0xF4	0xE8

Figure 1.0.14

Figure 1.0.15

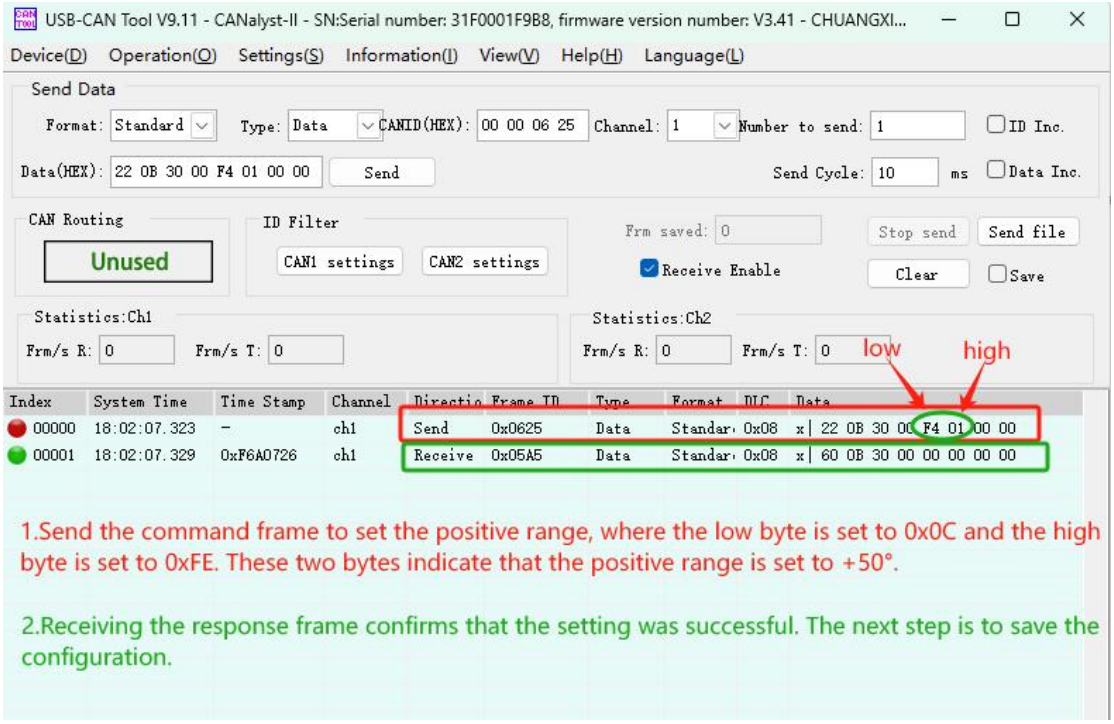

Figure 1.0.16

4.3 Midpoint Configuration

4.3.1: Locate the corresponding command frame in the protocol (Figure 1.0.17).

4.3.2: Command description: The set midpoint command is used to set the current position as the midpoint. This command is used for fine-tuning the midpoint and is limited to a range of ±20°. For example, to set the position at 5° as the midpoint, first send the command to set the position to 5°, and then send the set midpoint command, which will set the position at 5° as the midpoint.

4.3.3: Once the midpoint is successfully set, it takes effect immediately. If the save command is not used to save the setting, the midpoint will revert to its original position after a restart.

4.3.4: An example is provided to illustrate how to set +0.8° as the midpoint (Figure 1.0.18).

2.9 Set midpoint command

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 06 00 +NodeID	8	22	09	30	00	00	00	00	00

Figure 1.0.17

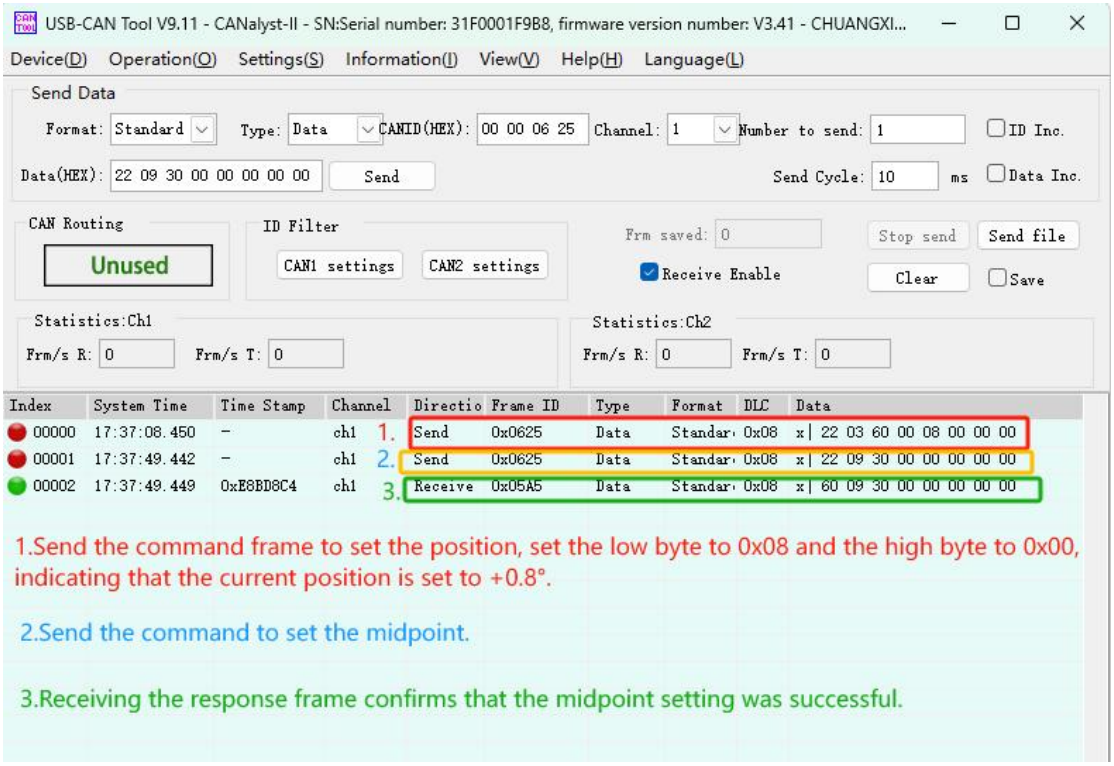

Figure 1.0.18

4.4 Guard Function Configuration

4.4.1: Locate the corresponding command frame in the protocol (Figure 1.0.19).

4.4.2: Function Description: The watchdog function is designed to monitor communication integrity between the host computer and the servo. Its operational principle is as follows: the host computer periodically sends request commands. If the servo detects a command within the preset time interval, communication is deemed normal. If no valid command is detected within the set time, communication is considered abnormal, and the servo will move to a predefined position. When the watchdog interval is set to "0", the watchdog function is disabled. (Default watchdog interval at power-on is "0", indicating the function is inactive).

4.4.3: Watchdog Interval: Represented by two bytes (High and Low), in milliseconds (ms), with a valid range of 100ms–5000ms.

4.4.4: Example: Instructions for setting the watchdog time to 100ms and reading the watchdog interval (Figure 1.0.20).

2.18.1 Read guard interval

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	40	0C	10	00	00	00	00	00

Reply to the following command to indicate that the operation is successful.

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 05 80 +NodeID	8	4B	0C	10	00	Low	High	00	00

The guard time is represented in two bytes, High and Low, with a unit of ms and a range of 100ms-5000ms.

2.18.2 Setting Guardian Interval

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	22	0C	10	00	Low	High	00	00

Low, High have the same definition as the read guard interval.

Reply to the following command to indicate that the setting is successful.

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 05 80 +NodeID	8	60	0C	10	00	00	00	00	00

Figure 1.0.19

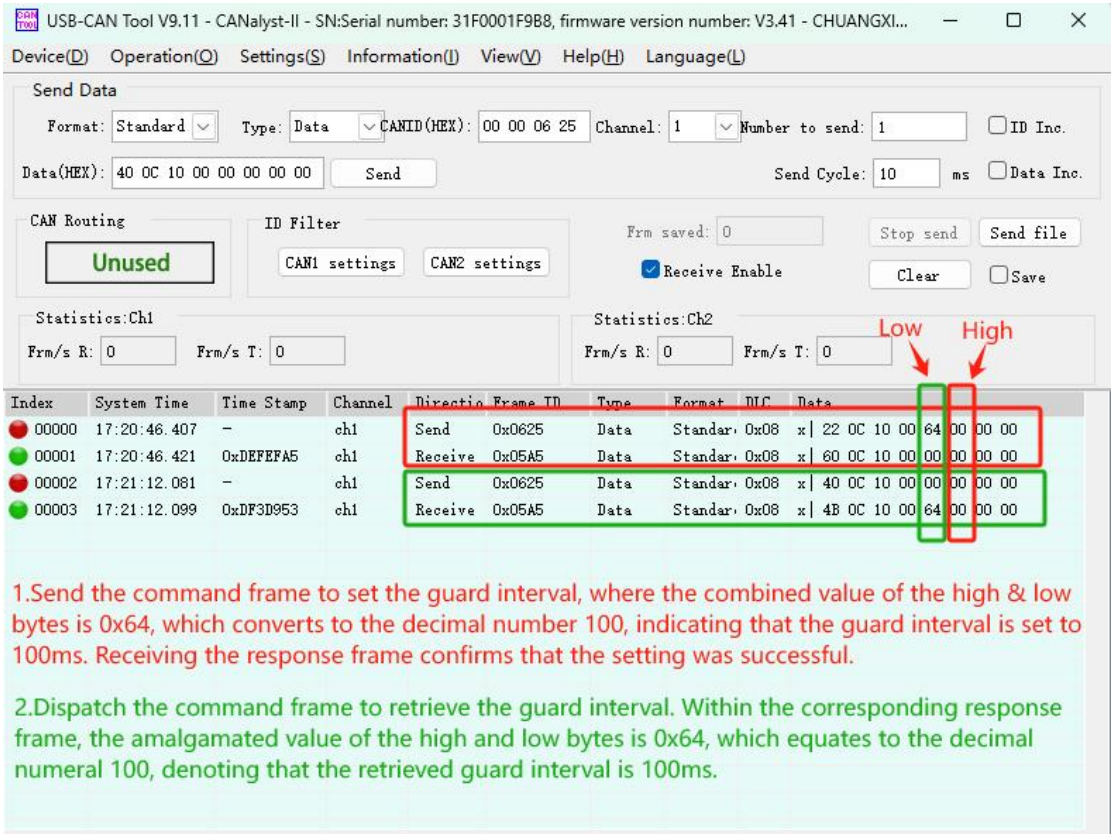

Figure 1.0.20

4.5 Factory Reset

4.5.1: Locate the corresponding command frame in the protocol.

4.5.2: Function Description: This function allows the host computer to send a command to restore the servo's configuration data to factory default settings.

4.5.3: Example: Instructions on how to restore factory configuration (Figure 1.0.21).

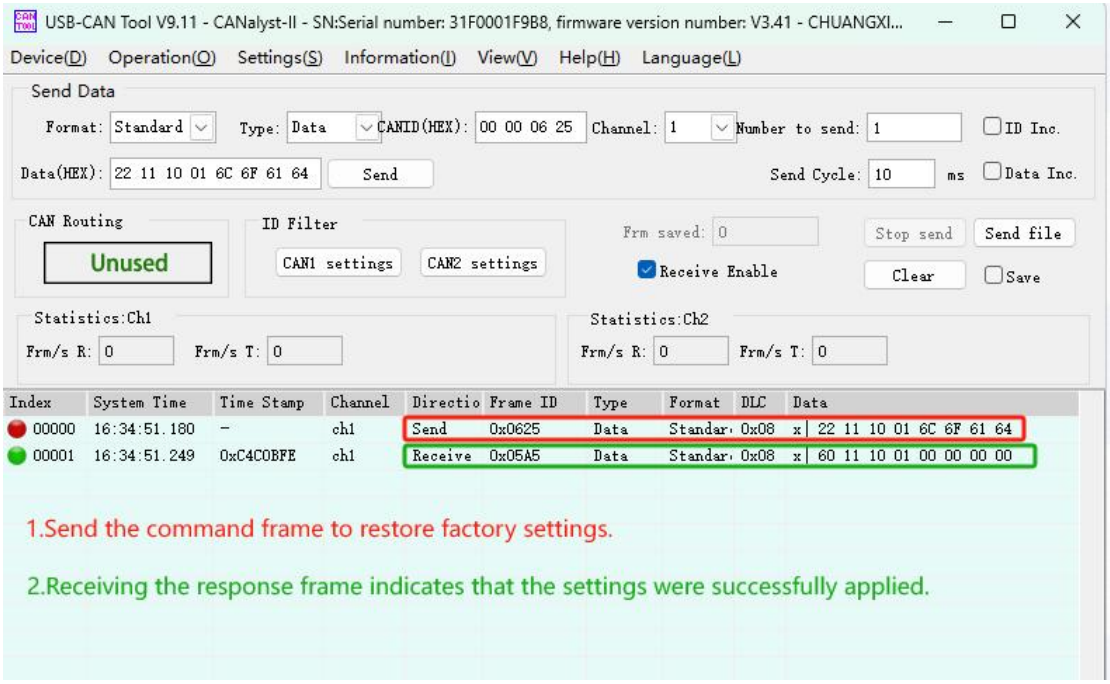

Figure 1.0.21

4.6 Enable Reporting Function

4.6.1: Locate the corresponding command frame in the protocol.

4.6.2: Send the corresponding communication command to the servo via the host computer (Figure 1.0.22).

4.6.3: Data reporting initiates, indicating successful startup. Continuous reception of data frames is now enabled (Figure 1.0.23).

2.1 Start reporting

ID	Length	Byte 1	Byte 2
0x 00 00 00 00 +NodeID	2	01	00

Figure 1.0.22

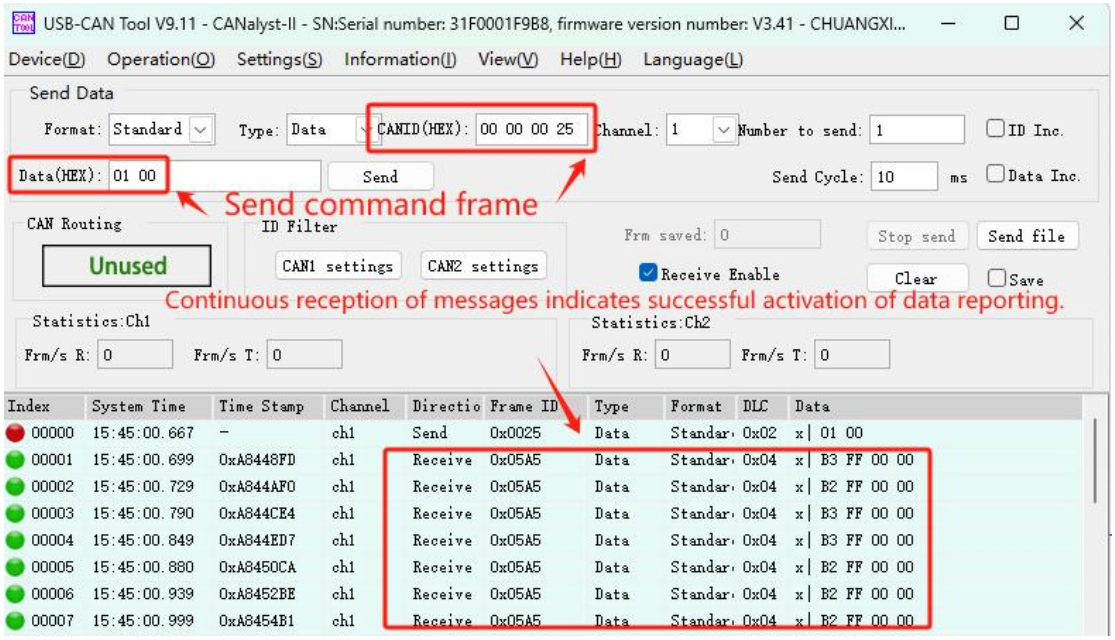

Figure 1.0.23

4.7 Disable Reporting

4.7.1: Locate the corresponding command frame in the protocol.

4.7.2: Send the corresponding communication command to the servo via the host computer (Figure 1.0.24).

4.7.3: Data reporting stops, indicating successful termination (Figure 1.0.25).

2.2 Stop reporting

ID	Length	Byte 1	Byte 2
0x 00 00 00 00 +NodeID	2	02	00

Figure 1.0.24

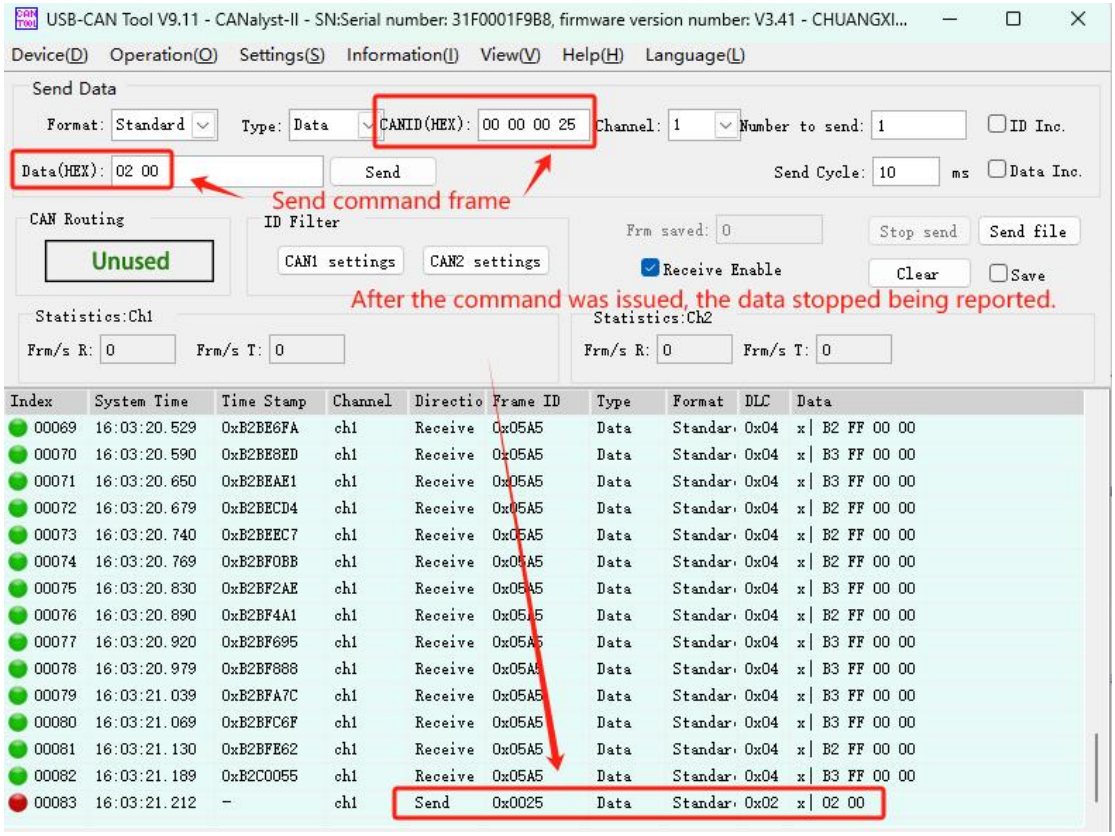

Figure 1.0.25

4.8 Save Command

4.8.1: Send the corresponding command frame.

4.8.2: Function Description: This command is used to save the node ID, baud rate, and other parameters. (No response required from the subordinate device).

4.9 Read Node ID

4.9.1: Send the corresponding command frame.

4.9.2: Receive the corresponding response frame, where X represents the read node ID (Figure 1.0.26).

4.9.3: Example: Instructions on how to read the node ID (Figure 1.0.27).

2.11 Read node number

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 06 00 +NodeID	8	40	00	30	00	X	00	00	00

reply:

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 05 80 +NodeID	8	4F	00	30	00	X	00	00	00

Figure 1.0.26

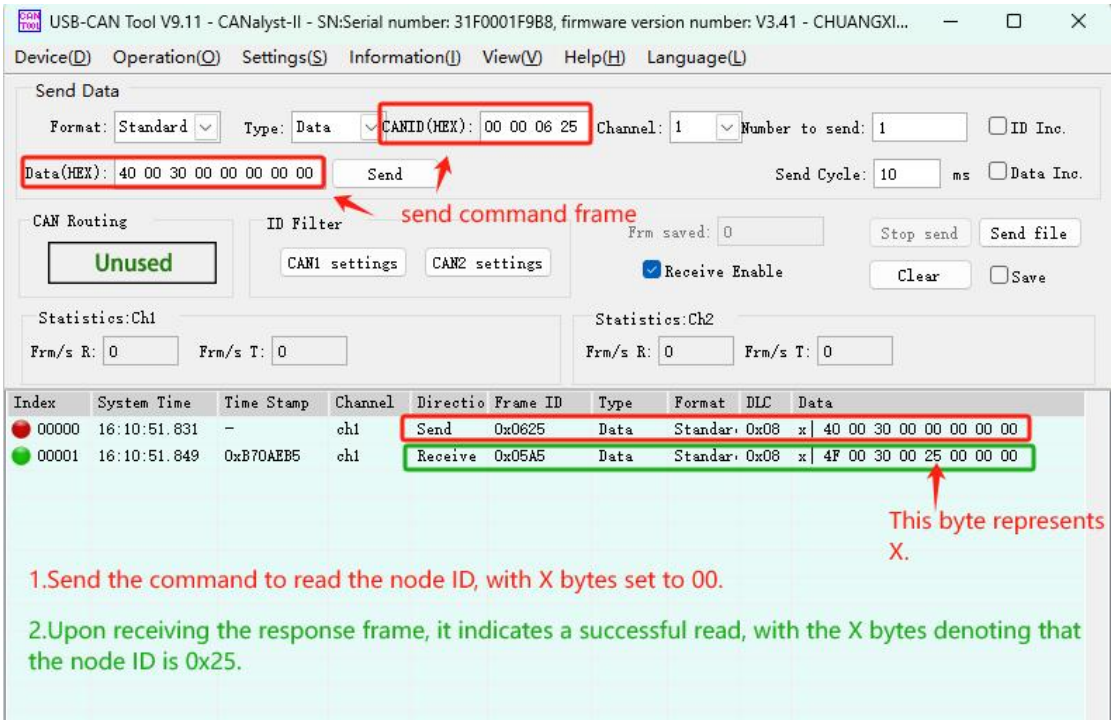

Figure 1.0.27

4.10 Function Enable Configuration

4.10.1: Command Description: This command is used to enable/disable specific system functions. First, read the enablement byte data, then modify only the corresponding bit(s) to activate/deactivate the desired function (other bits remain unchanged). Default state: "1" indicates enabled, "0" indicates disabled.

4.10.2: Read Command and Response Frame: Note that modifying reserved bits may cause servo malfunctions.

4.10.3: Modify Enablement and Response Frame: Ensure to save changes after modification (Figure 1.0.28).

4.10.4: Example: Instructions for enabling the reverse function (Figures 1.0.29–1.0.30).

Logic Reverse Function Description:

This function is implemented by setting the "Reverse" bit in the enablement byte.

When the "Reverse" bit is "0", the position returned by the read command is positive.

When the "Reverse" bit is "1", the position returned is negative.

Example: If the current position is 50°:

With the "Reverse" bit set to "0", the read command returns "50°".

With the "Reverse" bit set to "1", the read command returns "-50°".

Direction Inversion Function Description:

This function is implemented by setting the "Invert" bit in the enablement byte.

When the "Invert" bit is "0", the servo rotates in the default direction.

When the "Invert" bit is "1", the servo rotates in the opposite direction of the default.

The read enable command is as follows:

ID	Length	Byte1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	40	0F	30	00	00	00	00	00

reply:

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 05 80 +NodeID	8	4B	0F	30	00	X1	X2	00	00

X1,X2 is the enable byte of read back, and each bit has the following functions:

Bit	7	6	5	4	3	2	1	0
Function (X1)	X	logic reverse	Low voltage alarm	X	X	X	X	X
Bit	7	6	5	4	3	2	1	0
Function (X2)	X	X	X	X	X	X	X	motion reverse

X: Indicates the reserved bit. If the user rewrites it, the function of the servo will be abnormal;

The rewrite enable command is as follows: X1,X2 is the data written by the user

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 06 00 +NodeID	8	22	0F	30	00	X1	X2	00	00

The following reply indicates that the operation is successful:

ID	Length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x 00 00 05 80 +NodeID	8	60	0F	30	00	00	00	00	00

The rewrite function will take effect immediately after it is successful. If you do not save with the Save command, it will return to the original state after restart.

Figure 1.0.28

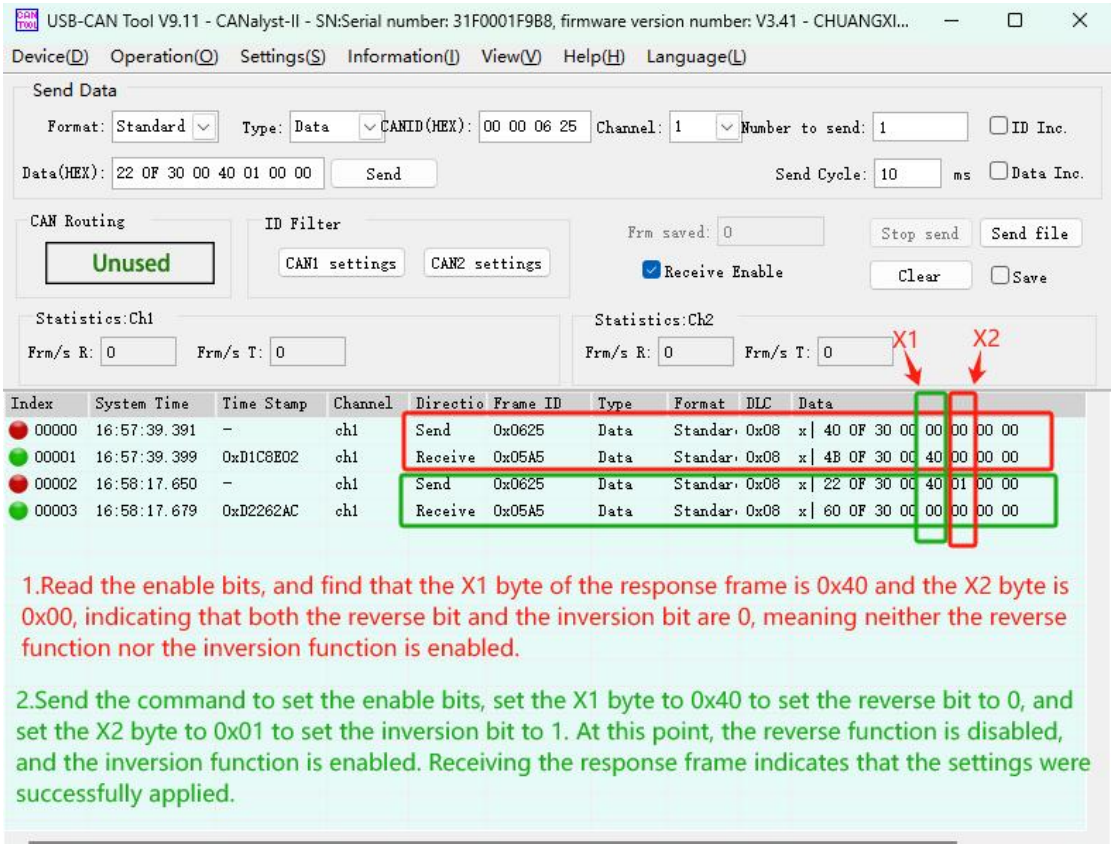

Figure 1.0.29

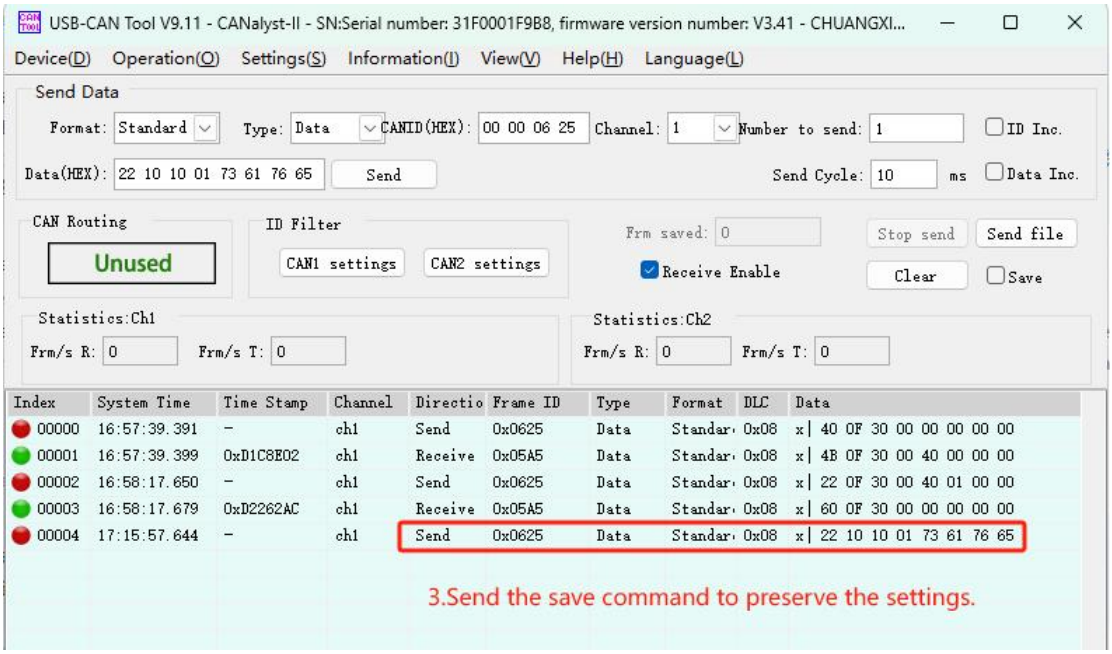

Figure 1.0.30

5. Servo Function Testing

5.1 Set Position

5.1.1: Function Description: As shown in the figure below, note that the position data resolution is 0.1°, with a range of -100° to +100°. Negative values are represented in two's complement (Figure 1.0.31).

5.1.2: Example: Instructions for setting the position to -0.8° and reading the configured position (Figure 1.0.32).

2.8 Position setting command

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 06 00 +NodeID	8	22	03	60	00	Low	High	00	00

(the lower computer does not need to answer)

The position data precision that can be set is 0.1 °, the range is - 100 ° to + 100 ° and the negative value is expressed by complement code as follows:

Position	-100°	-50°	-10.9°	-0.8°	0°	0.8°	10.9°	50°	100°
High	0xFC	0xFE	0xFF	0xFF	0x00	0x00	0x00	0x01	0x03
Low	0x18	0x0C	0x93	0xF8	0x00	0x08	0x6D	0xF4	0xE8

(If the position command exceeds the range, the current device returns an error code)

Figure 1.0.31

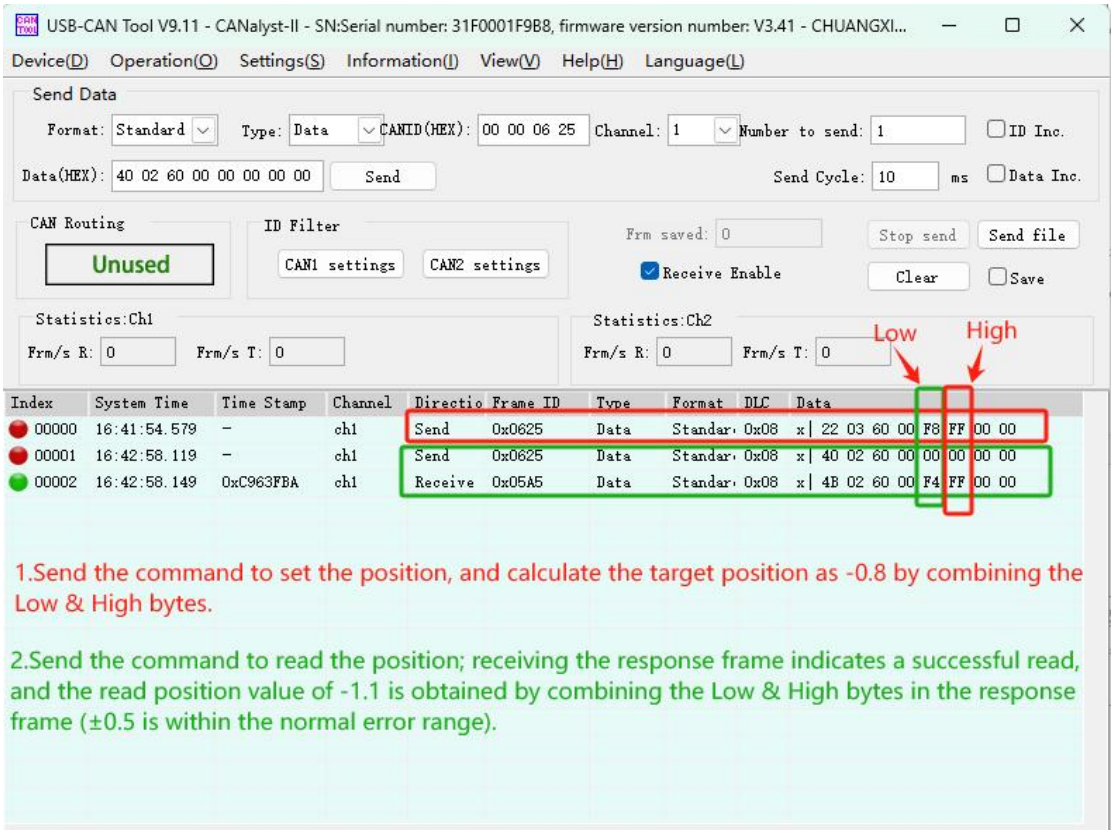

Figure 1.0.32

5.2 Read Position

5.2.1: Send the command frame to read position information (Figure 1.0.33).

5.2.2: Response Confirmation: Receiving the corresponding response frame indicates a successful read. Note that the High and Low bytes for the read position follow the same data structure as those used in position setting commands.

5.2.3: Example: Instructions for reading the current position information (Figure 1.0.34).

2.12 Read location

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 06 00 +NodeID	8	40	02	60	00	00	00	00	00

reply:

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 05 80 +NodeID	8	4B	02	60	00	Low	High	00	00

The read position is the same as the high and low byte definition of the set position.

Figure 1.0.33

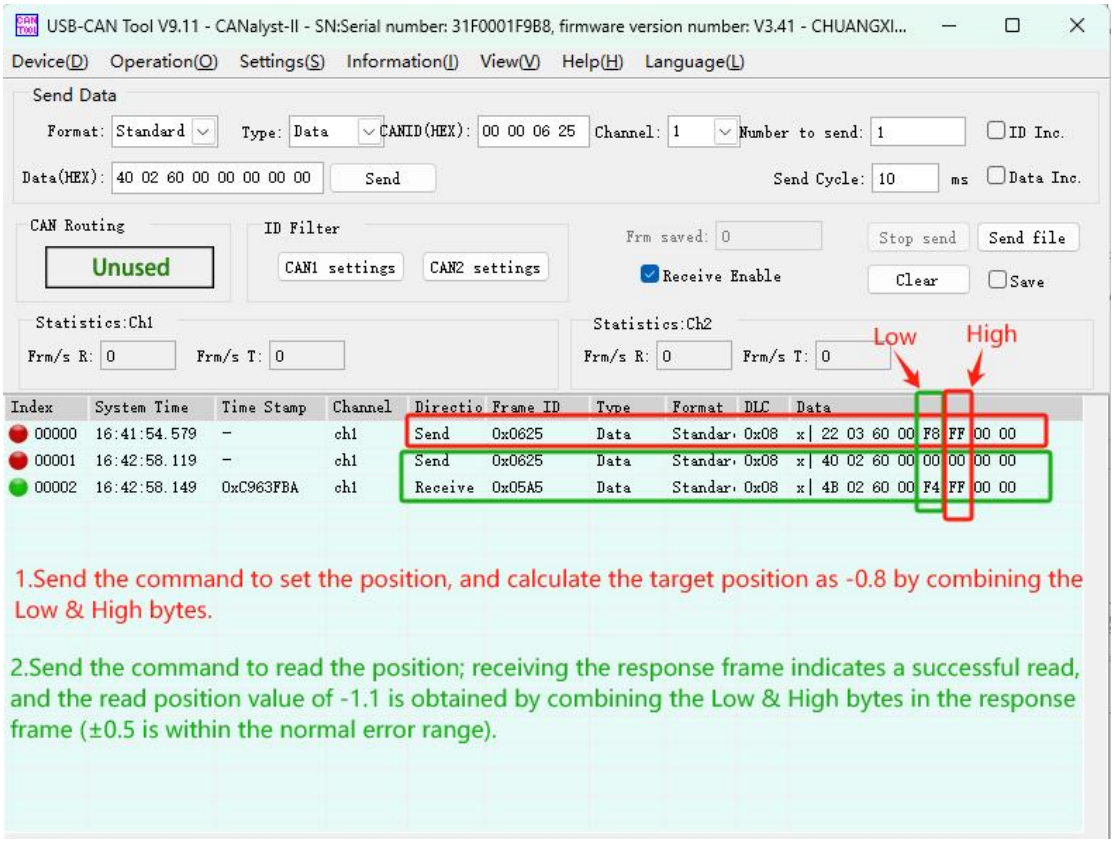

Figure 1.0.34

6. Servo Status Reading

6.1 Current and Temperature Reading

6.1.1: Send the read command and parse the response frame as shown in the figure (Figure 1.0.35).

6.1.2: Example: Instructions for reading current and temperature (Figure 1.0.36).

6.1.3: Based on the data read in Step 2, calculate the results using the formula defined in the command frame:

1.Current: 0x0000 converted to decimal is 0, with a unit of 10mA.

2.Temperature: 0x1E converted to decimal using two's complement calculation results in 30.

2.13 Read status

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 06 00 +NodeID	8	40	05	60	00	00	00	00	00

reply:

ID	Length	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x 00 00 05 80 +NodeID	8	43	05	60	00	X1	X2	X3	00

"X1" and "X2" denote the working current (accuracy ± 5%, or 0.01A) in 10mA, for example:

X2	0x00	0x00	0x00	0x01	0x01
X1	0x32	0x4B	0x64	0x00	0xF4
Current	0.50A	0.75A	1.00A	2.56A	5.00A

"X3" is the working temperature (temperature range - 40 ~ + 120 ℃, accuracy ± 2 ℃), expressed by complement code, for example:

X3	0xD8	0xF6	0x00	0x19	0x78
Temperature	-40℃	-10℃	0℃	25℃	120℃

Figure 1.0.35

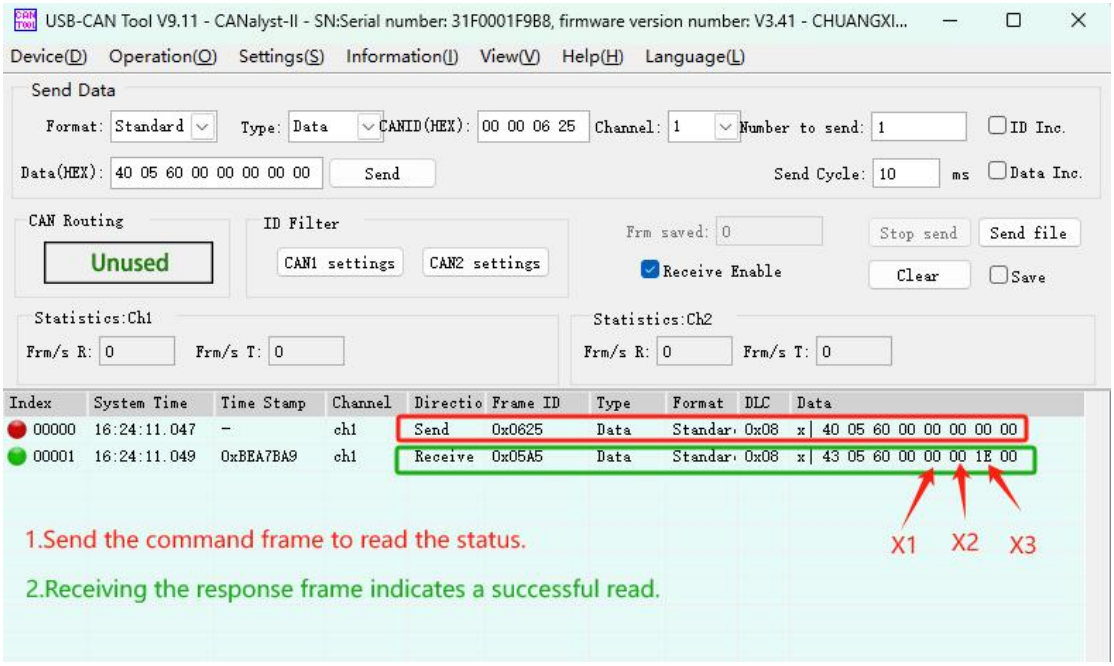

Figure 1.0.36

END