

# DH-3 Protocol

V1.0



## Contents

DH-3 Protocol.....	1
Introduction.....	2
Communication logic.....	2
Protocol format.....	3
Command Overview.....	4
Detailed command.....	4
Initialization.....	4
Force.....	5
Position.....	5
Feedback.....	6
CAN ID.....	7
Firmware Version.....	7
CAN Baud Rate.....	8

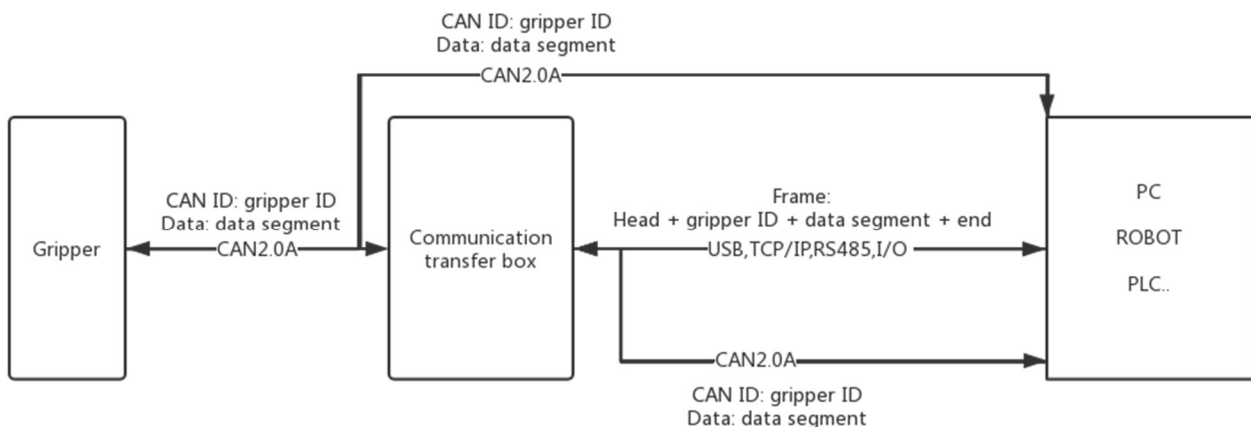
## Introduction

The DH-3 adaptive Robot hand use aviation connector to connect the CAN network. It supports the CAN protocols version 2.0A.

For environments without a CAN network, we have provided a communication transfer box to transfer other interfaces (like USB, TCP/IP, RS485, I/O) to CAN2.0A.

You can also connect it directly to the CAN network without the communication transfer box.

The communication system block diagram is as follows:



## Communication logic

- Successfully receive feedback:** The gripper will return the same data after successfully receiving the command.
- Initialization success automatic feedback:** After the initialization is successful, the success flag will be returned automatically.
- Initialization can be interrupted:** The initialization process can be interrupted by a new initialization command, and start a new initialization process. It is recommended to check the relevant flags to avoid frequent interruptions.
- Position command can be interrupted:** The moving process can be interrupted by a new position command, and gripper will move to the new position. It is recommended to check the relevant flags to avoid frequent interruptions.
- Guaranteed setting successfully:** The setting command (CAN ID, CAN Baud rate, I/O Mode) will return after the setting is successful, so gripper will not return the same data immediately.
- Command interval:** It is recommended that the send interval between the command and the command is above 20 milliseconds.
- Drop detection:** The diameter of the object needs to be greater than 5mm.

# Protocol format

(For TCP/IP, USB, RS485)

All external communication interfaces (except CAN and I/O interfaces) send commands to the communication transfer box in the following format (a total of 14 bytes).

Frame header	Gripper ID	Data Segment (8 Bytes)					Frame End
		Function register	Sub-Function register	Read/Write	Reserve	Data	
4 Bytes (0xFFFEFDFC)	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte (0x00)	4 Bytes	1 Byte (0xFB)

(For CAN2.0A)

The CAN ID is the gripper ID. The 8 bytes data is as follows:

Function register	Sub-Function register	Read/Write	Reserve	Data
1 Byte	1 Byte	1 Byte	1 Byte (0x00)	4 Bytes

The command consists of four parts: frame header, Gripper ID, data segment, and the end of the frame. the frame header and the end of the frame are fixed.

1. **Frame header:** The command starts with 0xFFFEFDFC, and the communication transfer box will recognize this field to determine the start of the command.
2. **Gripper ID:** The Gripper ID in the command is the actual CAN ID of the gripper (The default is 1), range : 0-255. (The "Gripper ID" in the following example is the default value of 1)
3. **Data segment:** It is the actual command, the data segment is also the CAN data segment, **when using the CAN interface as the communication interface, the command has 8 bytes, and there is no need to add a frame header and end.**

**Function register:** It is used to identify the main function of the command

**Sub-Function register:** It is used to identify the use function of the command

**Read/Write:** Only 0x00 and 0x01 are allowed, 0x00 indicates that the command is a read command, and 0x01 indicates that the command is a write command.

**Reserve:** unused, default is 0x00

**Data:** 32-bits signed integer, value range: 0x00000000 – 0xFFFFFFFF, **little endian** mode, For example, 1 = 01 00 00 00, -1 = FF FF FF FF.

(The data has no effect during the read operation.)

4. **Frame End:** The command starts with 0xFB and the communication transfer box will recognize this field to determine the end of the command.

example: Initialization --- FF FE FD FC 01 08 02 01 00 00 00 00 00 FB

# Command Overview

FUNCTION	FUNCTION REGISTER	SUB-FUNCTION REGISTER	REMARK
INITIALIZE	0x08	0x01-0x02	Initialize related commands
FORCE	0x05	0x02-0x03	Read/ Write open/close Force
POSTION	0x06	0x02	Read/ Write Position
ROTATION	0x07	0x02	Read/ Write Rotation
FEEDBACK	0x0F	0x01-0x02	Read current status
CAN ID	0x12	0x01	Read/ Write Gripper CAN ID
VERSION	0x13	0x01	Read Current Gripper firmware version
CAN BAUDRATE	0x14	0x01	Read/ Write Gripper CAN Baud Rate

## Detailed command

### Initialization

Function register	Sub-Function register	Read/Write	Reserve	Data	Function
0x08	0x01	0x00/0x01	0x00	Integer	Read/Write Initialization feedback
	0x02	0x00/0x01		Integer	Read/Write

When the Sub-Function register value is 0x01, the function of this command is to read and write whether feedback after finish initialization.

Example:

Set Initialization feedback:

Send : FF FE FD FC 01 08 01 01 00 00 00 00 00 FB

Receive : FF FE FD FC 01 08 01 01 00 A5 00 00 00 FB

Read whether feedback :

Send : FF FE FD FC 01 08 01 00 00 00 00 00 00 FB

Receive : FF FE FD FC 01 08 01 00 00 A5 00 00 00 FB (YES)

OR:

Receive : FF FE FD FC 01 08 01 00 00 00 00 00 00 FB (NO)

When the Sub-Function register value is 0x02, the function of this command is to initialize gripper or read whether finish initialization.

initialization :

Send : FF FE FD FC 01 08 02 01 00 00 00 00 00 FB

Receive : FF FE FD FC 01 08 02 01 00 00 00 00 00 FB

After set Initialization feedback:

Receive : FF FE FD FC 01 08 02 00 00 00 00 00 00 FB

read whether finish initialization :

Send : FF FE FD FC 01 08 02 00 00 00 00 00 00 FB

Receive : FF FE FD FC 01 08 02 00 00 01 00 00 00 FB (Finished)

OR:

Receive : FF FE FD FC 01 08 02 00 00 00 00 00 00 FB (Not finished)

## Force

Function register	Sub-Function register	Read/Write	Reserve	Data	Function
0x05	0x02/0x03	0x00/0x01	0x00	Integer	Read/Set Open/Close Force

The function of this command is to read and write the gripper Force (internal and external grip):

Data range:20-100 (14 00 00 00 – 64 00 00 00)

Example: (0x1E = 30)

Set 30% internal grip force:

Send : FF FE FD FC 01 05 02 01 00 1E 00 00 00 FB

Receive : FF FE FD FC 01 05 02 01 00 1E 00 00 00 FB

Read current internal grip force:

Send : FF FE FD FC 01 05 02 00 00 00 00 00 00 FB

Receive : FF FE FD FC 01 05 02 00 00 1E 00 00 00 FB

## Position

Function register	Sub-Function register	Read/Write	Reserve	Data	Function
0x06	0x02	0x00/0x01	0x00	Integer	Read/Write Position

The function of this command is to read and write the gripper position:

Data range: 0-100 (00 00 00 00 – 64 00 00 00)

Example: (0x3C = 60)

Set 60% position:

Send : FF FE FD FC 01 06 02 01 00 3C 00 00 00 FB

Receive : FF FE FD FC 01 06 02 01 00 3C 00 00 00 FB

Read current position:

Send : FF FE FD FC 01 06 02 00 00 00 00 00 00 FB

Receive : FF FE FD FC 01 06 02 00 00 3C 00 00 00 FB

## Rotation

Function register	Sub-Function register	Read/Write	Reserve	Data	Function
0x07	0x02	0x00/0x01	0x00	Integer	Read/Write Rotation

The function of this command is to read and write the gripper rotation:

Data range: 0-100 (00 00 00 00 – 64 00 00 00)

Example: (0x43 = 67% =  $67 \times 90^\circ = 60^\circ$  )

Set 67% position:

Send : FF FE FD FC 01 07 02 01 00 43 00 00 00 FB

Receive : FF FE FD FC 01 07 02 01 00 43 00 00 00 FB

Read current position:

Send : FF FE FD FC 01 07 02 00 00 00 00 00 00 FB

Receive : FF FE FD FC 01 07 02 00 00 43 00 00 00 FB

## Feedback

Function register	Sub-Function register	Read/Write	Reserve	Data	Function
0x0F	0x01	0x00	0x00	Integer	Read current status
	0x02	0x00	0x00	Integer	Read current rotation status

The function of this command is to read and write the gripper current status.

00 00 00 00 : default or moving

02 00 00 00 : Arrive position/rotation but not catch object

03 00 00 00 : Catch the object but not arrive position/rotation

Example:

Read current status:

Send : FF FE FD FC 01 0F 01 00 00 00 00 00 00 FB

Receive default : FF FE FD FC 01 0F 01 00 00 00 00 00 00 FB

OR:

Arrive position : FF FE FD FC 01 0F 01 00 02 00 00 00 00 FB

OR:

Catch the object : FF FE FD FC 01 0F 01 00 03 00 00 00 00 FB

Read current rotation status:

Send : FF FE FD FC 01 0F 02 00 00 00 00 00 00 FB

Receive default : FF FE FD FC 01 0F 02 00 00 00 00 00 00 FB

OR:

Arrive rotation : FF FE FD FC 01 0F 02 00 02 00 00 00 00 FB

OR:

Catch the object : FF FE FD FC 01 0F 02 00 03 00 00 00 00 FB

## CAN ID

Function register	Sub-Function register	Read/Write	Reserve	Data	Function
0x12	0x01	0x00/0x01	0x00	Integer	Read/Set Gripper CAN ID

The function of this command is to read and write the CAN ID. (default: ID = 1)

After the CAN ID has been set successfully, the gripper must be reboot.

Data Range: 1-255 (01 00 00 00 – FF 00 00 00)

When user don't know or forget gripper CAN ID, user can use ID 0 to read or set gripper CAN ID.

Example:

Set CAN ID to 2:

Send : FF FE FD FC 01 12 01 01 00 02 00 00 00 FB (when ID=1, set ID to 2)

Receive : FF FE FD FC 01 12 01 01 00 02 00 00 00 FB

OR: (when you don't know current ID)

Send : FF FE FD FC 00 12 01 01 00 02 00 00 00 FB (use ID=0, set ID to 2)

Receive : FF FE FD FC 00 12 01 01 00 02 00 00 00 FB

Read CAN ID:

Send : FF FE FD FC 02 12 01 00 00 00 00 00 00 FB (when ID=2, read ID)

Receive : FF FE FD FC 02 12 01 00 00 02 00 00 00 FB

OR: (when you don't know current ID)

Send : FF FE FD FC 00 12 01 01 00 00 00 00 00 FB (use ID=0 to read ID)

Receive : FF FE FD FC 00 12 01 01 00 02 00 00 00 FB

## Firmware Version

Function register	Sub-Function register	Read/Write	Reserve	Data	Function
0x13	0x01	0x00	0x00	0x00000000	Read gripper firmware version

The function of this command is to read gripper current firmware version.

Example:

Read firmware version:

Send FF FE FD FC 01 13 01 00 00 00 00 00 00 FB

Receive FF FE FD FC 01 13 01 00 00 00 02 01 04 FB

# CAN Baud Rate

Function register	Sub-Function register	Read/Write	Reserve	Data	Function
0x14	0x01	0x00/0x01	0x00	Integer	Read/Set Gripper CAN Baud Rate

The function of this command is to read and write the CAN baud rate.

After the CAN baud rate has been set successfully, the gripper must be reboot.

Data range: 0-5 (00 00 00 00 – 05 00 00 00)

Table: Baud Rate

Index	Baud Rate
0	500Kbps
1	400Kbps
2	250Kbps
3	200Kbps
4	125Kbps
5	100Kbps

Example:

Set CAN bps to 250K :

Send : FF FE FD FC 01 14 01 01 00 02 00 00 00 FB

Receive : FF FE FD FC 01 14 01 01 00 02 00 00 00 FB

Read CAN bps (Receive 2) :

Send : FF FE FD FC 01 14 01 00 00 00 00 00 00 FB

Receive : FF FE FD FC 01 14 01 00 00 02 00 00 00 FB