HP series Servo CAN Communication Technical Guide

Standard and Extended Frames CANopen Master Integration and Status Feedback

Thank you for your interest in the HP series Servo. The existing HP series Servo CAN communication protocol has been validated. It offers standard-frame and extended-frame communication options and supports position control, position feedback, status reading, fault alarms and communication guarding. This guide describes the practical differences between the two options, the configuration of multiple nodes on a shared bus, and the communication parameters needed for integration with an airborne control system.

This guide applies to HP series Servo CAN models that use this protocol. The communication rules below apply across models. For rated voltage, output torque, mechanical speed and actual travel, refer to the product specifications for the relevant model.

1 Differences Between Standard and Extended Frames

Item	CANopen Standard-Frame Option	CANopen Extended-Frame Option
CAN identifier	11 bits; range 0x000–0x7FF	29 bits; range 0x00000000–0x1FFFFFFF
IDE flag	IDE=0	IDE=1
CAN frame type	Classical CAN standard frame; supported by CAN 2.0B controllers	Classical CAN extended frame; CAN 2.0B format
Default bit rate	500 kbit/s	500 kbit/s
Payload per frame	Up to 8 bytes	Up to 8 bytes
Node ID	1–127; 0 is used for broadcast	This protocol also uses node IDs 1–127; the 29-bit identifier provides an extended message identifier space
Master configuration	Configure 11-bit IDs and the corresponding receive filters	Configure 29-bit IDs and the corresponding receive filters
Transmission overhead	Shorter frames for the same amount of data	Longer frames for the same amount of data; more space for identifier allocation

How the Identifier Format Relates to CANopen

The terms 11-bit and 29-bit refer to the length of the identifier in a CAN frame. CANopen defines object access, network management and data exchange. The CANopen predefined identifier allocation scheme is based on 11-bit IDs; 29-bit IDs are also supported, provided that the relevant services and master software support that format. Both HP series Servo communication options use the commands, object indices and data formats specified in this guide.

For example, the parameter request ID for node 0x25 is 0x600+0x25=0x625. Sending this value in a standard frame requires IDE=0; sending it in an extended frame requires IDE=1. These are different frames on the bus. Leading zeros in a hexadecimal display do not change the frame format. A 29-bit extended frame is not equivalent to DroneCAN, ARINC 825 or CANopen FD.

2 CANopen Master Integration and Message Organization

When connecting to a CANopen master, the master interface must support the selected frame format and allow the CAN-ID, DLC, data content and response handling to be configured according to the HP series Servo protocol. Standard-frame mode uses 11-bit CAN-IDs, while extended-frame mode uses 29-bit CAN-IDs. Receive filters must match the corresponding IDE flag.

The HP series Servo interface uses object indices to access position, configuration and status data. The master uses its custom CAN message transmit/receive functions to handle the control and feedback data listed below. Reporting control commands, periodic position reports and alarms use messages defined for the HP series Servo. The master application parses these messages according to the table. Here, n is the servo node ID.

Communication Function	HP series Servo Message Definition	Master Handling
Parameter access	Request ID=0x600+n; response ID=0x580+n; DLC=8	Access objects by index and subindex; receive and match parameter responses
Target position	Object 0x6003:00; DLC=8; 2-byte position value	No success acknowledgment for normal execution; monitor motion through position feedback and receive and handle out-of-range error responses
Periodic position feedback	ID=0x580+n; DLC=4; Low High 00 00	Parse as a 4-byte position report; use DLC to distinguish it from an 8-byte parameter response
Fault alarms	ID=0x380+n; DLC=4	Identify undervoltage, overcurrent and overtemperature from the alarm payload
Communication guarding	ID=0x700+n; master sends RTR; 1-byte response	Send guarding requests periodically; check the response and the alternating most significant bit
Start and stop reporting	ID=0x000+n; DLC=2; 01 00 starts reporting, 02 00 stops reporting	Send HP series Servo reporting control commands to the specified node
Save parameters	Object 0x1010:01; data 73 61 76 65	Send the save command; no response is returned

Receiving Parameter Frames and Periodic Data

When receiving periodic position reports and parameter responses on the same interface, the master identifies each message using CAN-ID, IDE, RTR and DLC together. For example, for 0x580+n, DLC=4 denotes a periodic position report, while DLC=8 denotes a parameter access response or an error response. Multi-byte numerical values are transmitted least significant byte first.

Master Software Configuration

To send commands and read status, configure the CAN frame format, node ID, message identifiers and data formats according to the HP series Servo communication protocol. The receive routine handles reporting control commands, 4-byte position reports and 4-byte alarms as specified in the table, and manages timeouts according to each command's response rules. A command with no response does not return an application-layer success acknowledgment; the CAN data-link-layer ACK mechanism still applies.

Standard and extended frames can coexist on a CAN network that supports both formats. Each servo communicates using its selected frame format. All devices on a mixed-format bus use the same bit rate. Prevent multiple devices from transmitting different data simultaneously with the same frame format and CAN-ID. Coordinate the 0x380+n alarm IDs and 0x580+n position-report IDs with the identifier allocations of other devices.

3 Multi-Servo Configuration and Addressing on a Shared Bus

Node IDs and Default Configuration

Each servo is addressed by a unique node ID in the range 1–127. The factory default is 0x25, and node ID 0 is reserved for broadcast. The master and all servos use the same bit rate. Use standard or extended frames according to the selected communication configuration. The node ID address space and the practical number of devices on the bus are separate considerations: the number of connected devices depends on the transceivers, cabling and bus load.

Purpose	CAN-ID Rule	Example for Node 0x25
Parameter requests and position commands	0x600+n	0x625
Parameter responses and periodic position feedback	0x580+n	0x5A5
Undervoltage, overcurrent and overtemperature alarms	0x380+n	0x3A5
Communication guarding	0x700+n	0x725
Start, stop, prepare and reset	0x000+n	0x025

Multi-Node Configuration Procedure

1 Connect the servos that use the default node ID one at a time. Establish communication using the current bit rate and frame format.
2 Set a unique node ID through 0x3000:00, and set the bus bit rate through 0x3001:00.
3 Send the 0x1010:01 save command, then restart the servo.
4 Read back the configuration using the new node ID and bit rate. Once all nodes have been configured, connect them to the same bus.

The node ID and bit rate take effect after the settings are saved and the servo is restarted. Configuring servos individually prevents multiple devices with the default address from responding at the same time. Send configuration commands to the intended node. Limit broadcast operations to the commands defined as supporting broadcast by the protocol.

Bit Rate Selection

Setting code	00	01	02	03	04	05	06	07
Bit rate in kbit/s	20	50	100	125	250	500	800	1000

The default code is 05, which corresponds to 500 kbit/s. When changing the bit rate of the entire network, update the master and every node consistently, and adjust the controller bit timing accordingly. Use a trunk bus topology, fit termination resistors matched to the cable at both ends of the trunk, and limit the length of the stubs.

Configuration Readback Example

To read the node ID of node 0x25, send ID=0x625, DLC=8, data 40 00 30 00 00 00 00 00. The response is ID=0x5A5, DLC=8, data 4F 00 30 00 X 00 00 00, where X is the node ID. Set IDE to match the standard-frame or extended-frame option.

4 Update Rates Position Feedback and Command Latency

Item	Communication Parameters and Data Definitions
Position reporting period	Default: 50 ms, or 20 Hz. Configurable range: 10–255 ms, corresponding to approximately 100–3.92 Hz. Changes take effect immediately. Save the setting to retain it after restart.
Target position and position feedback	Target object: 0x6003:00; feedback object: 0x6002:00. Values are signed 16-bit two's complement integers in little-endian byte order, with 0.1° per count. The protocol position range is ±100°; the actual range depends on the product and the configured limits.
Position data resolution	The 0.1° value is the communication encoding resolution. Mechanical positioning accuracy and repeatability are specified separately for each model.
Speed information	The protocol directly provides position feedback. The main controller calculates speed from the change between consecutive positions and the actual sampling interval. This is an estimate calculated by the main controller.
Current feedback	Read 0x6005:00. Current is a 2-byte unsigned little-endian integer in units of 10 mA. The communication quantization step is 0.01 A.
Temperature feedback	Read the temperature byte from 0x6005:00. It is a signed 8-bit two's complement value in units of 1 °C. Data range: −40 to +120 °C; measurement accuracy: ±2 °C.
Command and feedback periods	Position command transmission, position reporting and internal servo control have separate timing parameters. The 10 ms value is the shortest configurable position reporting period. The internal control frequency is determined separately by the firmware.
Command latency	Command latency includes master queuing, CAN arbitration and transmission, firmware processing and internal control updates. Mechanical positioning time also depends on the servo model, travel, load and power supply conditions.

Frame Transmission Times at 500 kbit/s

The reference times below are calculated for Classical CAN without bit stuffing or error retransmissions, and include the 3-bit intermission. Actual bus time also includes bit stuffing, arbitration delays and retransmissions. These values are intended for communication budgeting and do not represent the complete mechanical response time.

Message	11-Bit Standard Frame	29-Bit Extended Frame
4-byte periodic position report	Approx. 158 µs	Approx. 198 µs
8-byte command or parameter frame	Approx. 222 µs	Approx. 262 µs
Position read request and response, each with 8 data bytes	Approx. 444 µs	Approx. 524 µs

Multi-Node Traffic Example

Traffic budget example: assume that position commands and position reports for 10 servos are both scheduled at 100 Hz. Each servo receives 1 command with 8 data bytes and sends 1 report with 4 data bytes per cycle. At 500 kbit/s, nominal bus utilization is approximately 38% for standard frames and 46% for extended frames. This is an estimate of message traffic. The complete network budget must also include guarding, status reads, alarms, bit stuffing and traffic from other devices.

The overall command processing rate, end-to-end latency and jitter must be evaluated for the specific model, bus load and control-system schedule. The frame times above exclude firmware processing and mechanical motion. For speed estimation by position differencing, use the actual receive timestamps recorded by the master. The estimate is affected by reporting jitter, position quantization and filtering.

5 Fault Reporting and Communication Reliability

Fault alarms

Alarm Type	ID and Data	Default State and Action
Undervoltage	ID=0x380+n; DLC=4 0A 02 00 00	Disabled by default; enable through the function enable settings. The alarm typically triggers at approximately 70% of the rated operating voltage. The servo loses its operating torque when the alarm triggers. The exact threshold is set for the specific model.
Overcurrent	ID=0x380+n; DLC=4 0A 03 00 00	Enabled by default. The alarm triggers when the configured current limit is reached; the output operates according to the current limiting settings.
Overtemperature	ID=0x380+n; DLC=4 0A 04 00 00	Enabled by default. The alarm triggers when the configured temperature limit is reached; the output operates according to the temperature limiting settings.

The alarm is reported immediately when the fault first occurs. While the fault persists, the alarm is repeated every 10 seconds. Reporting of that alarm stops when the fault clears. Alarms are independent of periodic position reporting. The master must also use communication guarding and status reads to determine whether operation has recovered. An absence of further alarm reports can also result from a loss of communication or insufficient power.

Communication Guarding and the Preset Position on Communication Loss

Communication guarding is disabled by default. Enable it through the guarding extension byte of 0x300F. Once enabled, the master periodically sends an RTR remote frame with ID=0x700+n. The servo replies with the same ID and 1 data byte, whose most significant bit alternates between 0 and 1. The master interface must support RTR. Before changing the enable bit, read the current value and preserve the other function bits.

Set the guarding interval through 0x100C:00; the range is 100–5000 ms. Set the preset position on communication loss through 0x3021:00. If the servo does not receive a valid guarding request within the configured interval, it moves to the preset position. Set the master's guarding request period shorter than the guarding timeout interval, allowing sufficient margin for bus scheduling.

The control system manages position command updates, feedback updates and guarding communication separately. Guarding uses master requests and servo responses. The master uses this mechanism to manage online status and timeout actions. Moving to the preset position on communication loss requires normal servo power and a functioning actuator mechanism.

Read and Write Errors and Status Reading

Parameter read/write errors are returned with ID=0x580+n and DLC=8. The first byte is 80, followed by the relevant index, subindex and error code. Code 0x06020000 indicates that the object does not exist. Code 0x06010002 indicates an attempt to write to a read-only object. A position command does not return a success acknowledgment during normal execution. If the command is outside the configured motion range, the servo does not move and returns an error response.

Read operating current and temperature through 0x6005:00. Object 0x6001:00 provides debug status data. The CAN data link uses CRC, frame checks, acknowledgment and error handling. The application layer uses periodic feedback and communication guarding to manage online status.

Control Input Priority

Object 0x3011:00 sets the priority of the PWM, CAN and RS485 control inputs. Object 0x3012:00 sets the switching wait time: 1–250 counts, with 4 ms per count, corresponding to 4–1000 ms. This is the waiting period when switching control sources. It is defined separately from the transmission or processing time of an individual CAN command.

6 Communication Object Dictionary I

The table below lists communication configuration, calibration and statistics objects. R means read and W means write. Except for the save and restore objects with subindex 01, all objects use subindex 00. Access these objects using the HP series Servo communication commands.

Index and Subindex	Object Name	Access and Data	Units or When Effective
100A:00	Hardware version	R; 4-byte version string	Hardware version identifier
100B:00	Software version	R; 4-byte version string	Software version identifier
100C:00	Guarding interval	R/W; 2 bytes	100–5000 ms
1010:01	Save parameters	W 73 61 76 65	ASCII save; no response
1011:01	Restore factory configuration	W 6C 6F 61 64	ASCII load
2200:00	Position reporting interval	R/W; 1-byte effective value; read response: 4B	10–255 ms; default: 50; takes effect immediately; saved setting retained after restart
3000:00	Node ID	R/W; 1 byte	1–127; default: 0x25; save and restart
3001:00	Bit rate code	R/W; 1 byte	00–07; default: 05 (500 kbit/s); save and restart
3002:00	Midpoint voltage	R/W; 2 bytes	Raw ADC value; save and restart
3003:00	Angle scaling factor	R/W; 2 bytes	Angle calibration parameter; save and restart
3004:00	Accumulated operating time	R; 4 bytes; W to clear	10 min per count
3005:00	Midpoint count	R; 2 bytes; W to clear	Count
3006:00	Power-on count	R; 4 bytes; W to clear	Count
3007:00	Temperature scaling parameter	R/W; 1 byte	Temperature calibration parameter; save and restart
3008:00	Current scaling parameter	R/W; 2 bytes	Current calibration parameter; save and restart
3009:00	Set and read midpoint	W to set; R; 2-byte position value	Midpoint trim: ±20°; takes effect immediately; saved setting retained after restart

Saving Parameters and Applying Configuration

The node ID, bit rate and other parameters that require saving are written to configuration storage when the save command is executed. The node ID and bit rate take effect after restart. Settings that take effect immediately, such as the position reporting interval and midpoint, are retained after restart if saved. Follow the rules for when each command's settings take effect.

Calibration parameters are used to match angle, current and temperature measurements to the product. Application control normally uses the position, current, temperature and operating configuration objects. Set calibration parameters according to the calibration procedure for the relevant product.

7 Communication Object Dictionary II

Index and Subindex	Object Name	Access and Data	Units or Function
300A:00	Negative motion limit	R/W; signed 2-byte value	0.1°; 0 to −100°; save and restart
300B:00	Positive motion limit	R/W; signed 2-byte value	0.1°; 0 to +100°; save and restart
300C:00	Self-test lower position	R/W; 2-byte position value	0.1°; save and restart
300D:00	Self-test upper position	R/W; 2-byte position value	0.1°; save and restart
300F:00	Function enable	R/W; function bit field	Position direction, limits, alarms and guarding; read before writing and preserve the other function bits
3010:00	Torque release	W to execute; response returned	Output torque is 0; sending another position command resumes motion
3011:00	Control source priority and related settings	R/W; 1-byte bit field	PWM/CAN/RS485 priority and inversion
3012:00	Control source switching wait time	R/W; 1 byte	1–250; 4 ms per count
3021:00	Preset position on communication loss	R/W; 2-byte position value	0.1°; used with communication guarding
6001:00	Debug status data	R; 2 bytes	S0: low byte; S1: high byte
6002:00	Actual position	R; signed 2-byte value	0.1°; least significant byte first
6003:00	Target position	W; signed 2-byte value	0.1°; protocol range: ±100°; no success acknowledgment during normal execution
6004:00	Angle ADC value	R; 2 bytes	Raw angle ADC counts
6005:00	Current and temperature	R; 4-byte packed data	Current low byte, current high byte, temperature in two's complement, 00

Data Decoding Examples

Position: multiply the signed 16-bit little-endian integer by 0.1°. Low=F4 and High=01 represent +50.0°; Low=0C and High=FE represent −50.0°. Current: (X1+256×X2)×0.01 A. Decode the temperature byte as a signed 8-bit two's complement value in °C.

Position Command and Feedback Examples

To set node 0x25 to +50.0°, send ID=0x625, DLC=8, data 22 03 60 00 F4 01 00 00. To read the position, send 40 02 60 00 00 00 00 00 with the same ID. The reply uses ID=0x5A5 and data 4B 02 60 00 Low High 00 00. Periodic position feedback also uses ID=0x5A5, with DLC=4 and data Low High 00 00.

Protocol terminology references: CiA descriptions of CANopen and SDO (can-cia.org/can-knowledge/canopen; can-cia.org/can-knowledge/sdo-protocol). Frame structure reference: Bosch CAN Specification 2.0. Use the HP series Servo commands and objects as defined in this guide.