



MOTION STREAM INTERFACE (MSI) REFERENCE MANUAL

Protocol Version 0

February 2026



Contents

1	Introduction	2
1.1	Intended Use	2
1.2	Product Safety	3
1.3	Limited warranty	3
1.4	Personnel Qualification	3
1.5	Safety-Conscious Working.....	4
2	System Architecture	5
2.1	Overview.....	5
2.2	Control Loop Integration	5
2.3	Timing Model.....	6
3	MSI State Machine	7
3.1	States.....	7
3.2	General Rules	8
4	Network and Communication	9
4.1	UDP Communication Model.....	9
4.2	Network Setup.....	9
5	Data Types and Encoding.....	10
5.1	Numeric Types.....	10
5.2	Time Representation	10
6	Command Set Reference	11
6.1	OPEN.....	12
6.2	ENABLE	13
6.3	POINT.....	14
6.4	DISABLE	15
6.5	CLOSE.....	16
6.6	Feedback Packets	17
7	Examples.....	19
8	Status Codes.....	20
9	Flags.....	22

1 Introduction

The Motion Stream Interface (MSI) is an optional high-performance technology package for Weiss Robotics WPG series servo-grippers. It provides a UDP-based real-time streaming interface that enables external controllers to directly influence gripper motion with deterministic timing.

MSI allows position and force setpoints to be streamed directly into the gripper's internal position and force control loops. Motion trajectories are generated externally and executed by the gripper on a fixed 1 ms internal control raster, enabling precise, responsive, and highly dynamic motion behavior.

This interface is specifically designed for advanced applications that require a high degree of flexibility and real-time responsiveness, including but not limited to:

- Visual servoing
- Advanced control strategies (e.g. impedance or force-adaptive control)
- Teleoperation systems
- AI-based and learning control approaches

Unlike command-based motion interfaces, MSI follows a strictly stream-oriented programming model. The gripper does not generate trajectories internally. Instead, uninterrupted motion execution depends entirely on the continuous delivery of base points by the external controller. While this approach enables sophisticated and adaptive control strategies, it also places responsibility for timing, trajectory generation, and error handling on the external system.

MSI can also be used independently of motion execution as a high-frequency, real-time monitoring interface, providing continuous feedback on gripper state, motion, force, and system status.

This reference manual describes the MSI architecture, communication protocol, timing model, state machine, command set, and feedback mechanism. It serves as a foundation for system integrators and developers to design, implement, and deploy robust real-time gripper applications using the Motion Stream Interface.

1.1 Intended Use

The Motion Stream Interface is primarily intended for research and advanced development applications and must only be used with compatible Weiss Robotics gripper modules. The user is responsible for implementing the external controller that generates motion trajectories and ensures compliance with real-time requirements. Any use outside the scope described in this manual is considered improper and may lead to unsafe operation. Weiss Robotics assumes no liability for damage resulting from such improper use.



Use of the Motion Stream Interface may generate motion profiles that significantly increase mechanical stress on the gripper and can lead to accelerated wear or mechanical damage. Such effects are not covered by the Weiss Robotics warranty..

1.2 Product Safety

When using the Motion Stream Interface, all safety instructions and limitations described in the operating manual of the connected gripper module apply without restriction.



MSI allows direct influence on gripper motion. Incorrect configuration, insufficient timing guarantees, or software errors may result in unintended movements and even permanent damage of your gripper.

1.3 Limited warranty

The Motion Stream Interface (MSI) enables direct, real-time access to the motion behavior of the gripper and allows users to generate highly dynamic, non-standard motion profiles. Depending on the implemented control strategy, MSI can be used to create motion patterns that significantly increase mechanical stress on the gripper, including high accelerations, frequent direction changes, continuous force loading, or operation close to physical limits.

Such operating conditions may lead to **accelerated wear**, reduced service life, or, in extreme cases, **mechanical damage** to the gripper or its components. These effects can occur even when the gripper is operated within nominal electrical limits.

Weiss Robotics does not assume any warranty for damage, excessive wear, or reduced lifetime resulting from the use of the Motion Stream Interface. In particular, no warranty is provided for damages caused by application-specific motion profiles, externally generated trajectories, or control strategies implemented by the user.

The user is solely responsible for ensuring that all motions generated via MSI are suitable for the mechanical design and intended operating conditions of the gripper. This includes appropriate limitation of speed, acceleration, force, duty cycle, and thermal load.

Use of the Motion Stream Interface is therefore at the user's own risk and requires thorough validation and testing before deployment in any productive or safety-relevant environment.

1.4 Personnel Qualification

Installation, commissioning, and operation of applications using MSI may only be performed by trained and qualified personnel with expertise in:

- Industrial automation systems
- Real-time communication
- Control engineering

1.5 Safety-Conscious Working

The following safety rules must always be observed:

- Do not reach into moving parts while the power supply is connected.
- Avoid entering the movement range of the gripper during operation.
- Disconnect the power supply before performing mechanical work.
- Ensure that emergency stop mechanisms are available and functional.

2 System Architecture

2.1 Overview

The Motion Stream Interface integrates an external real-time controller directly into the internal control structure of the gripper. The external controller is responsible for generating motion trajectories and transmitting them to the gripper in the form of streamed base points.

MSI follows a strictly **stream-oriented programming model**. It is not a command-based motion interface and does not generate trajectories internally. Instead, motion execution depends entirely on the continuous delivery of base points by the external controller. Each base point defines a motion segment that is interpolated by the gripper onto its internal 1 ms control raster.

To ensure uninterrupted motion, the external controller must maintain a stable stream of base points and actively manage the internal interpolation buffer. MSI provides continuous feedback, including buffer fill level, operating state, and error flags, which must be monitored by the controller at runtime. Error codes and fault flags must be handled immediately to ensure safe operation.

This programming model enables highly flexible and adaptive motion strategies, but it also places full responsibility for timing, trajectory generation, and error handling on the external controller.

2.2 Control Loop Integration

MSI injects motion points directly into the position and force control loops of the gripper. This allows the external controller to fully define the motion behavior, subject to the physical and safety limits of the hardware.

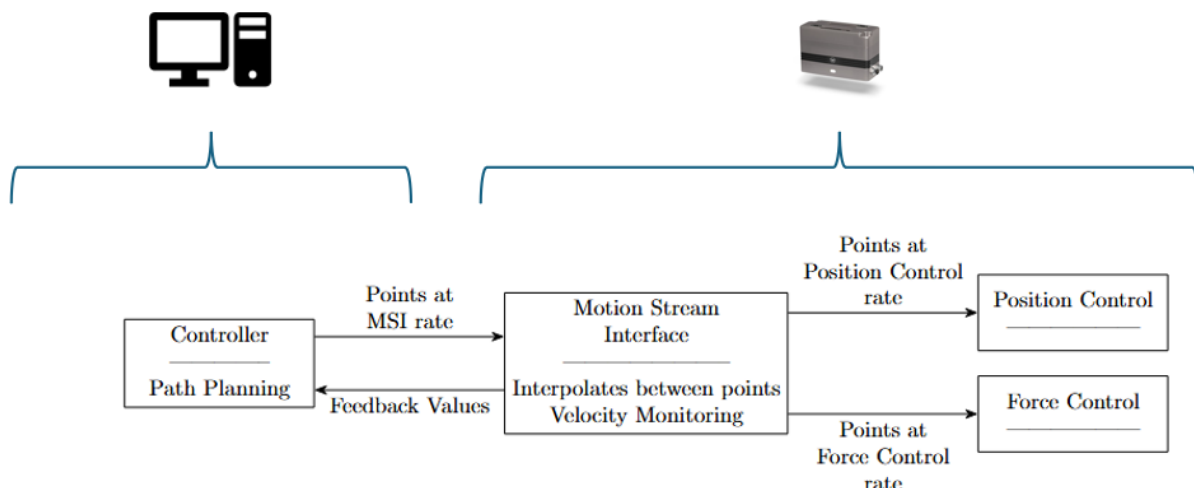


Figure 1: Interaction of the Motion Stream Interface with the control loops and the gripper controller

2.3 Timing Model

MSI operates on a fixed internal control raster of **1 ms**. The external controller does not stream individual control points for each millisecond. Instead, it streams **base points** at a fixed period that is defined when MSI is enabled.

These base points are interpolated by the gripper onto the internal 1 ms controller time grid. The interpolation is performed entirely inside the gripper.

The following rules apply:

- Time is measured in *ticks*, where one tick equals 1 ms.
- Base points are streamed at a fixed period, which is configured at startup.
- Base points must be sent at least at this configured period.
- Each base point defines a segment that is interpolated to 1 ms resolution by the gripper.

MSI maintains an internal buffer for **interpolated points** with a capacity of **500 points**. Since each interpolated point corresponds to 1 ms of motion, the buffer can bridge a maximum interruption of **0.5 seconds**.



If the buffer runs empty, a buffer underrun occurs. Motion execution is then interrupted immediately and a FAULT state is entered.

3 MSI State Machine

The Motion Stream Interface operates as a deterministic state machine. State transitions are triggered by specific MSI commands.

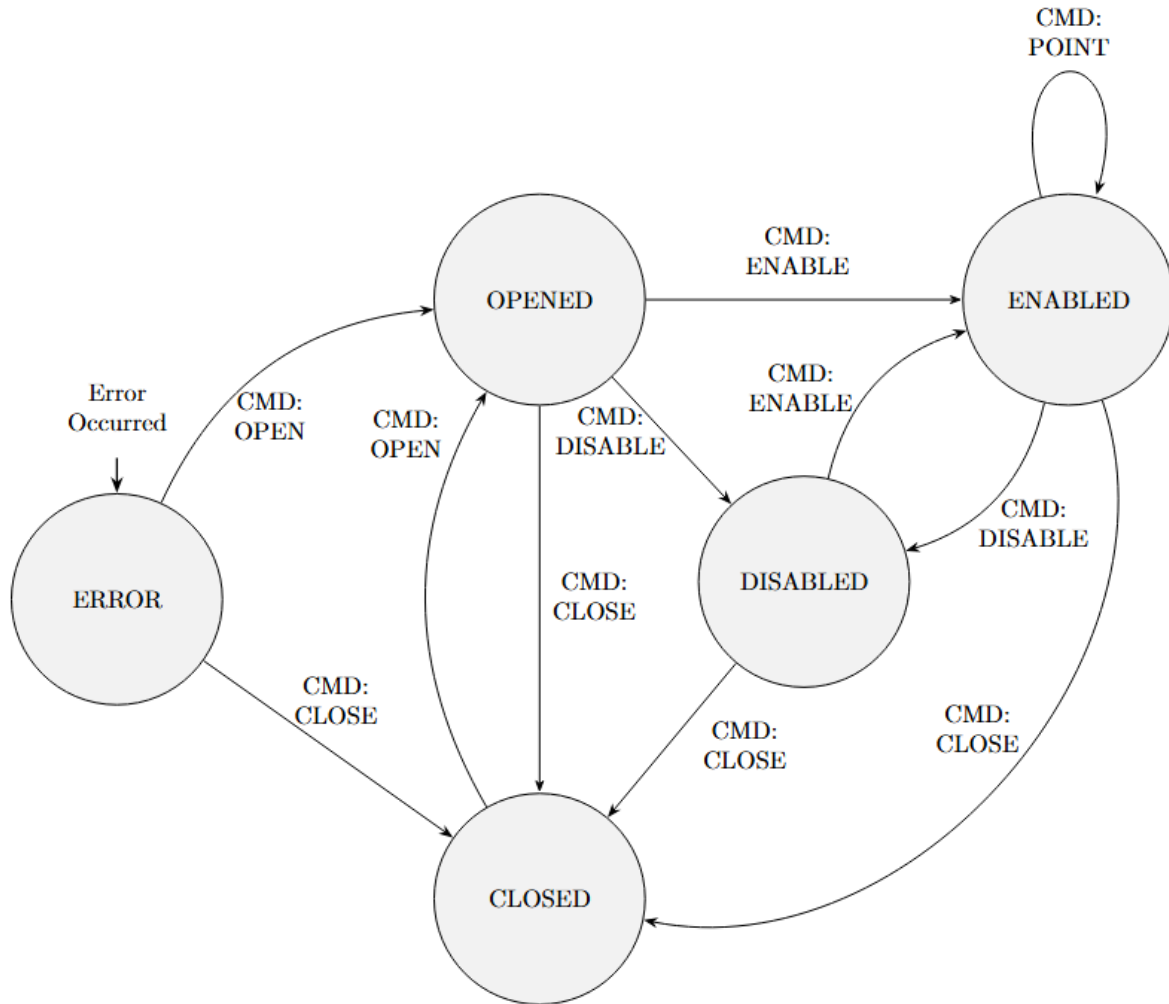


Figure 2: State Machine of the Motion Stream Interface

3.1 States

- **CLOSED:** MSI inactive, no feedback is transmitted.
- **OPENED:** MSI active, feedback is transmitted, gripper can be used normally.
- **ENABLED:** MSI accepts motion points and blocks other motion commands.
- **DISABLED:** MSI active but not executing motion; gripper is free to move.
- **FAULT:** An error occurred and must be resolved before further operation.

3.2 General Rules

- Motion points may only be sent in the **ENABLED** state.
- Feedback transmission is active in all states except **CLOSED**.



Certain state transitions depend on the underlying GRIPLINK device state of the gripper. See the GRIPLINK Unified Command Set Reference Manual for details.

4 Network and Communication

4.1 UDP Communication Model

MSI uses UDP for both command transmission and feedback reception. All packets have a fixed size and a strictly defined layout.

- The gripper listens for MSI commands on UDP port **5005**.
- Feedback packets are sent to the IP address and port configured via the OPEN command.

Strict adherence to packet structure and timing requirements is mandatory. Malformed or delayed packets may be ignored by the gripper.

4.2 Network Setup

The gripper and the host controller must be located within the same network. Each gripper module must be assigned a unique IP address to ensure unambiguous communication. Network latency and jitter directly influence motion quality and can adversely affect the stability and accuracy of real-time motion execution.

5 Data Types and Encoding

All multi-byte data fields are transmitted in **little-endian** byte order.

5.1 Numeric Types

- **float32**: IEEE-754 single-precision floating-point value (4 bytes)
- **uint32**: Unsigned 32-bit integer (4 bytes)
- **uint16**: Unsigned 16-bit integer (2 bytes)
- **uint8**: Unsigned 8-bit integer (1 byte)

5.2 Time Representation

- **tick**: Base time unit, equal to 1 ms

6 Command Set Reference

All MSI commands use a fixed packet size of **32 bytes**. Each command packet is composed of eight consecutive 4-byte words. The first word contains the command opcode and uniquely identifies the requested MSI operation. The remaining words contain command-specific parameters or reserved fields. Reserved fields are currently unused and must be set to zero by the sender to ensure forward compatibility.

MSI commands do not return immediate responses. Instead, the result of command execution is reported asynchronously via the periodic MSI feedback packets. Both the current MSI operating state and the command return status are transmitted in the feedback packet using the fields `opstate` and `statuscode`.

The feedback packets are sent at the configured feedback period and must be continuously monitored by the controller. Correct interpretation of `opstate` and if a state transition did not occur, `statuscode` is essential to detect successful state transitions, invalid commands, or error conditions.

MSI command execution and state transitions can be monitored via the gripper's web-based control interface. When MSI commands are issued, corresponding log entries are displayed in the **GRIPLINK Log** view of the web interface, including command type, parameters, and execution status. This allows users to verify correct command sequencing and diagnose errors during integration and testing. An example of such log output is shown Figure 3.

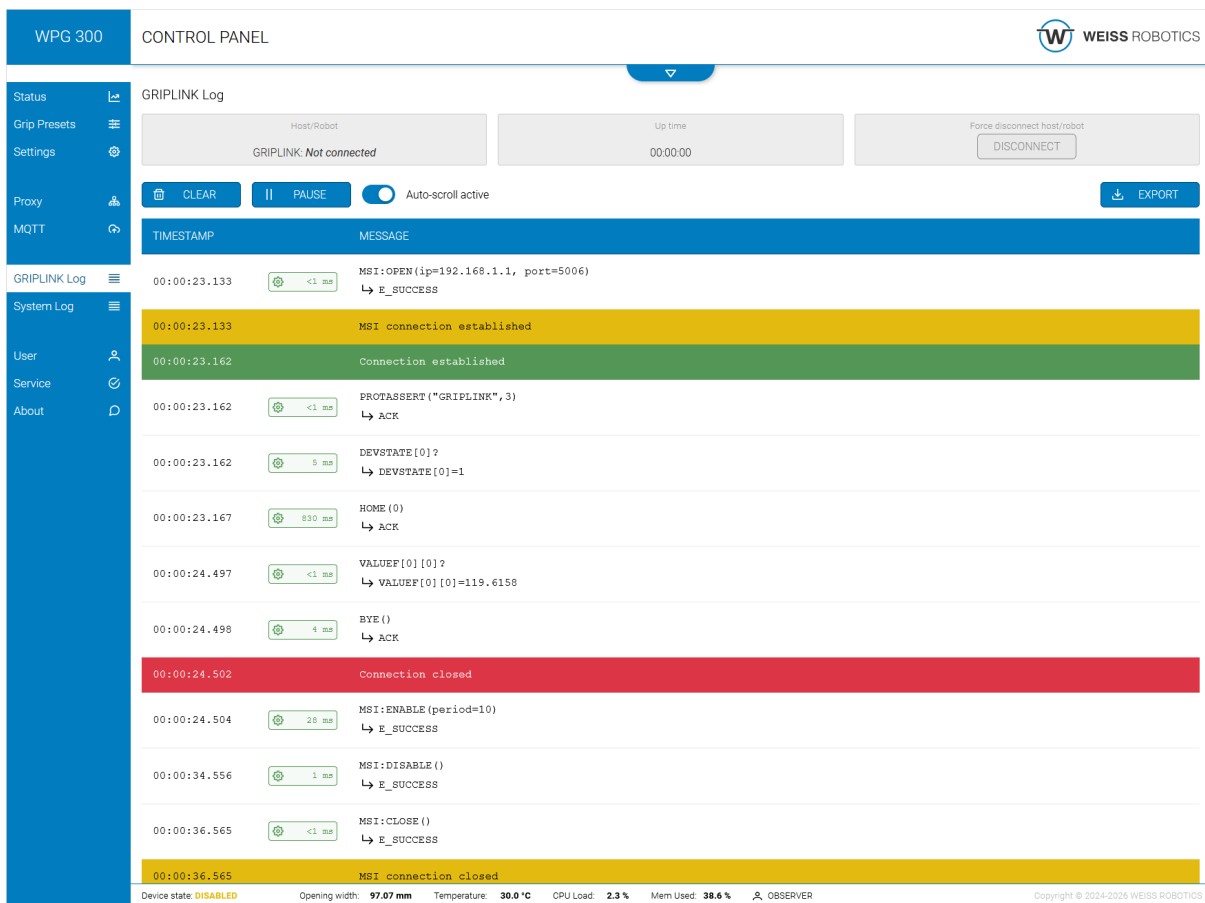


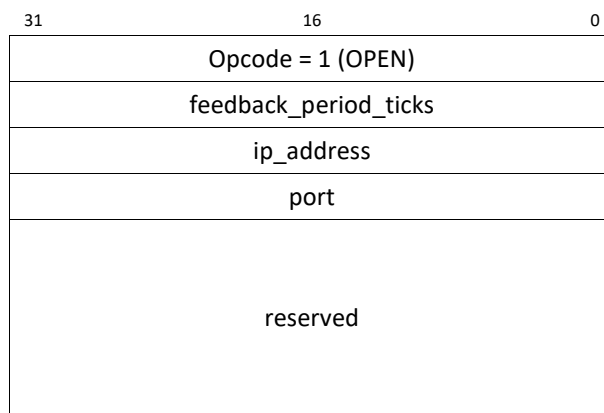
Figure 3: MSI command logging

6.1 OPEN

OPEN configures the Motion Stream Interface and initializes feedback transmission from the gripper to the external controller. It defines the transmission interval of MSI feedback packets as well as the destination IP address and UDP port used for this communication.

After successful execution of the OPEN command, MSI transitions to the OPENED state and immediately starts sending feedback packets to the configured UDP endpoint. This state transition is permitted from any GRIPLINK device state. As a result, MSI can be used as a high-frequency, real-time monitoring interface even without enabling motion execution.

Allowed States: OPEN can be issued in any MSI state.



feedback_period_ticks (uint32)

Specifies the period at which MSI transmits feedback packets, expressed in ticks. One tick corresponds to 1 ms. Minimum value is 2.

ip_addr (uint32)

Destination IP address for MSI feedback packets. The address is encoded as a 32-bit unsigned integer in network byte order and transmitted as four bytes within the command packet.

port (uint32)

Destination UDP port for MSI feedback packets. The port must be set to an available, unused port within the permitted range 1024 to 5004 or 5006 to 65535.

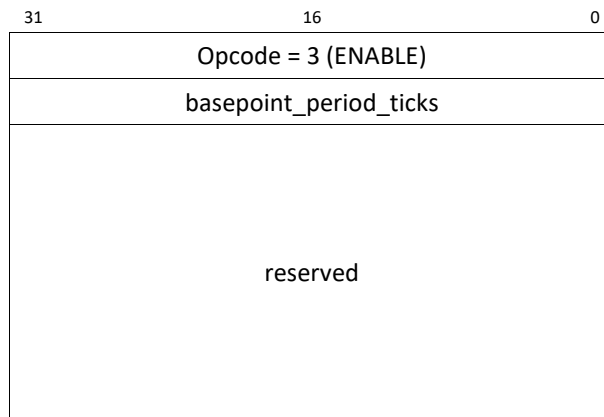
6.2 ENABLE

ENABLE transitions MSI into real-time motion execution mode and prepares the gripper to receive streamed base points. After ENABLE is executed successfully, MSI transitions to the ENABLED state. The gripper holds its current position and blocks other motion commands. MSI expects base points to be received at the configured fixed period. Motion execution begins with the arrival of the first POINT command.

Allowed states: ENABLE can only be issued when MSI is in the OPENED state and the gripper is referenced (i.e. GRIPLINK device state is not “DS_NOT_INITIALIZED”).



If base points are not received at the required rate, a buffer underrun may occur.



basepoint_period_ticks (uint32)

Required reception period for base points, expressed in ticks. Minimum value is 2.

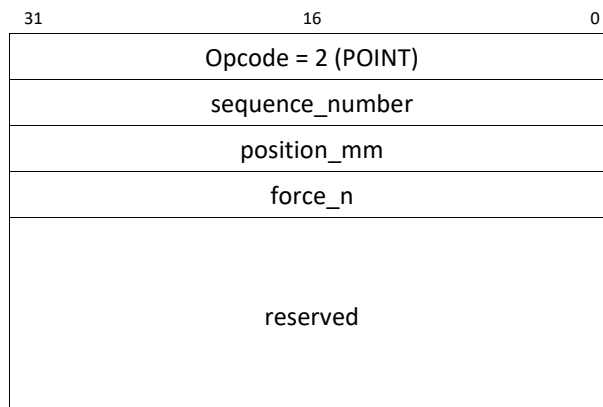
6.3 POINT

Streams a single base point to the MSI interpolation buffer for real-time motion execution. Each POINT command appends one base point to the internal interpolation buffer, which is interpolated by the gripper onto its internal 1 ms control raster. Continuous streaming of base points is required to ensure uninterrupted motion.

Each base point sent with the POINT command expands into `basepoint_period_ticks` interpolated points within the buffer. The interpolation buffer can hold up to 500 interpolated points. To prevent buffer overruns, the controller must continuously monitor the `points_buffered` field in the feedback packet and ensure sufficient free buffer capacity before sending the next base point.

The `sequence_number` is echoed in the feedback packet as an acknowledgment and can be used by the controller to verify correct reception and processing order. To improve robustness against packet loss and network jitter, the MSI implements duplicate suppression based on the sequence number. If a POINT command is received with the same `sequence_number` as the previously accepted POINT command, the packet is silently ignored. This mechanism allows the controller to retransmit base points in case of suspected packet loss without risking unintended duplicate motion segments caused by delayed UDP packets.

Allowed States: POINT can only be issued when MSI is in the ENABLED state.



sequence_number <uint32>

Application-defined sequence number. If two consecutive POINT commands carry the same sequence number, the later packet is silently discarded.

position_mm <float32>

Target gripper position in millimeters.

force_n <float32>

Maximum allowable gripping force in Newtons. The force limit is implemented by limiting the motor current. The gripper internally converts the specified gripping force linearly into the corresponding motor current limit.

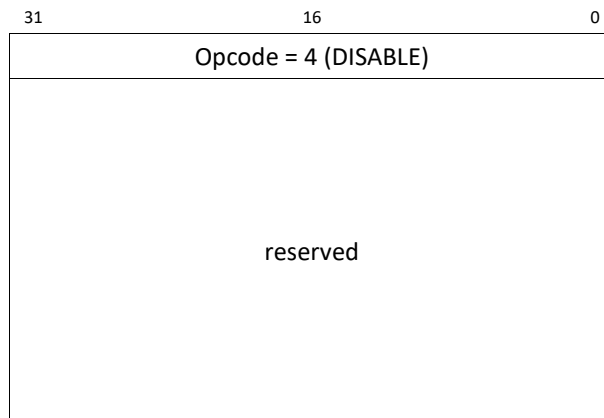
6.4 DISABLE

DISABLE stops real-time motion execution while keeping the Motion Stream Interface active. After DISABLE is executed, MSI transitions to the DISABLED state. No further base points are accepted, the gripper's motor is disabled and finger position control is released. Other motion commands issued e.g. via the GRIPLINK Unified Command Protocol may become available depending on the GRIPLINK device state.



Feedback transmission continues while MSI remains active.

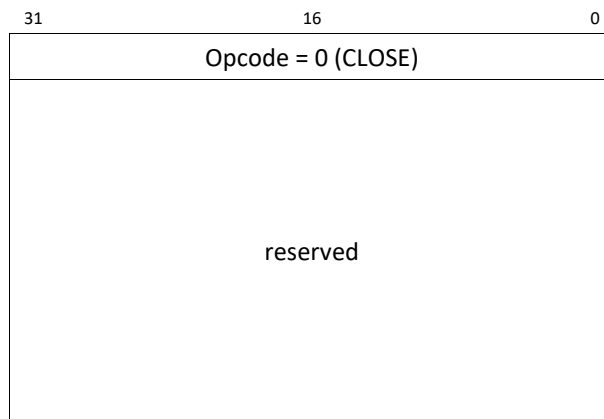
Allowed States: DISABLE can only be issued when MSI is in the ENABLED state.



6.5 CLOSE

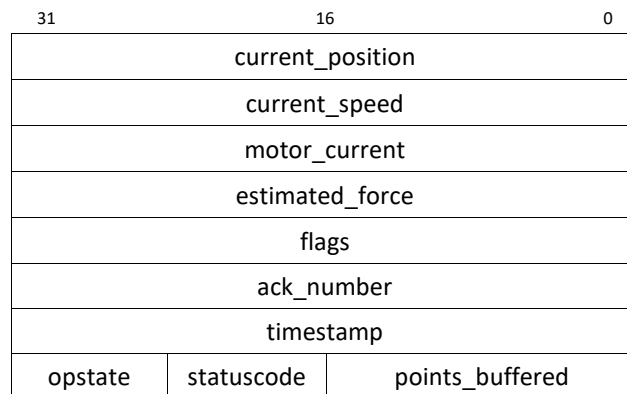
CLOSE deactivates the Motion Stream Interface and terminates feedback transmission. After CLOSE is executed, MSI transitions to the CLOSED state. Feedback transmission stops and no further MSI commands or motion points are processed. The gripper is fully released for normal operation.

Allowed States: CLOSE can be issued in any MSI state.



6.6 Feedback Packets

To enable monitoring of the gripping module and the implementation of basic control loops, continuous feedback from the gripper is required. This feedback is provided through periodic feedback packets transmitted by the gripper interface. Each feedback packet has a fixed length of 32 bytes and contains real-time information describing the current behavior and condition of the gripper. The data includes the finger position and speed, the (mathematically) estimated applied gripping force, status and fault flags, buffering information, and timing data. These values allow the receiving system to track finger motion, evaluate applied force, detect warnings or errors, and verify the correct processing of previously sent commands.



current_pos (float32, mm)

Finger position in millimeters.

current_speed (float32, mm/s)

Finger speed in millimeters per second.

motor_current (float, A)

Motor current in Amperes.

estimated_force (float32, N)

The (mathematically) estimated gripping force that is currently applied to the fingers in Newtons.

Note: estimated_force represents an internally calculated value and is not a direct force measurement.

flags (uint32)

A 32-bit bitmask containing various status and fault flags. See section 9 for a detailed explanation of each flag.

ack_number (uint32)

Echoes the sequence number from the last point command received. This can be used to confirm which base point was processed.

time_stamp (uint32,ticks)

The timestamp in ticks at which the feedback packet was generated and sent.

opstate (uint8)

Indicates the current operating state of the MSI. The operating state can take one of the following values:
0: CLOSED, 1: OPENED, 2: ENABLED, 3: DISABLED, 4: FAULT

statuscode (uint8)

Provides the status code of the last command. See section 8 for status code definitions.

points_buffered (uint16)

Number of interpolated points currently stored in the MSI interpolation buffer. This value is intended to be used for base point flow control.

Note: Each base point sent with the POINT command will inflate into basepoint_period_ticks interpolation points.

7 Examples

The following example programs demonstrate typical usage patterns of the Motion Stream Interface. They are intended as reference implementations and starting points for custom applications. All examples are written in Python and are heavily commented to support understanding and adaptation.

The example package consists of the following files:

griplink_client.py

Provides a TCP client implementation for the Weiss Robotics GRIPLINK protocol. It is used for device management tasks such as homing, enabling, disabling, and querying device state. In the examples, this client is responsible for preparing the gripper for MSI operation and handling non-real-time commands.

msi_client.py

Implements the Motion Stream Interface client. This module encapsulates all MSI-related functionality, including UDP-based command transmission, feedback reception, internal state tracking, buffering, and flag evaluation. It provides a high-level, state-aware API that closely reflects the MSI protocol described in this manual.

main_sin2x.py

Demonstrates streaming of a precomputed motion trajectory to a single gripper. The example generates a sinusoidal-squared position profile, enables MSI, and streams base points at a fixed rate while actively monitoring buffer fill level and error states. It illustrates correct timing behavior, buffer management, and orderly startup and shutdown of MSI.

main_teleoperation.py

Demonstrates a teleoperation scenario involving two grippers. One gripper acts as the master and provides position feedback via MSI, while the second gripper acts as the slave and mirrors the master's motion in real time. The example highlights bidirectional MSI usage, feedback-driven control, and continuous streaming under real-time constraints.



These examples are not intended to be production-ready solutions but serve as a clear and minimal reference for implementing custom real-time controllers based on MSI.

8 Status Codes

After executing a command, the feedback packet returns a status code describing the result of the operation. This status is compatible with GRIPLINK's Unified Command set.

Status code	Identifier	Description
0	E_SUCCESS	No error. Command successfully executed.
1	E_OVERRUN	Data overrun
2	E_RANGE_ERROR	Value out of range
3	E_NOT_AVAILABLE	Function or data not available
4	E_NOT_INITIALIZED	Device not initialized
5	E_TIMEOUT	Timeout
6	E_INSUFFICIENT_RESOURCES	Not enough memory available
7	E_CHECKSUM_ERROR	Checksum error
8	E_ACCESS_DENIED	Access denied
9	E_INVALID_HANDLE	Invalid handle
10	E_INVALID_PARAMETER	Invalid parameter
11	E_INDEX_OUT_OF_BOUNDS	Index out of bounds
12	E_IO_ERROR	Generic I/O error
13	E_READ_ERROR	Read error
14	E_WRITE_ERROR	Write error
15	E_NOT_FOUND	Resource not found
16	E_NOT_OPEN	File or device not open
17	E_EXISTS	Resource already exists
18	E_NO_COMM	Connection error
19	E_STATE_CONFLICT	Invalid state
20	E_NOT_SUPPORTED	Command or function not supported
21	E_INCONSISTENT_DATA	Data inconsistent

22	E_CMD_SYNTAX	Syntax error
23	E_CMD_UNKNOWN	Unknown command
24	E_CMD_ABORTED	Command aborted
25	E_CMD_FAILED	Command failed
26	E_AXIS_BLOCKED	Axis is blocked
27	E_PENDING	Pending action / Not yet finished

9 Flags

The flags field is a 32-bit bitmask encoding status and fault conditions. Only selected bits are currently defined; all others are reserved for future use. Each bit represents a specific status or fault condition. A bit value of 1 indicates that the corresponding condition is active, while a value of 0 indicates that the condition is inactive.

Bit	Function
31	<p>Interpolation buffer underrun (DF_BUFFER_UNDERRUN)</p> <p>Interpolation point buffer underrun occurred. While this is intended at the end of a motion, POINT will raise an error if a new base point was sent while this flag is set.</p>
30 to 23	reserved
22	<p>Motor Temperature Fault (DF_MOTOR_TEMP_FAULT)</p> <p>This bit is set when the motor temperature exceeds the maximum safe limit and typically requires immediate action or shutdown.</p>
21	<p>Undervoltage Fault (DF_UNDERVOLTAGE_FAULT)</p> <p>This bit indicates that the supply voltage has dropped below the allowable operating range.</p>
20	<p>Phase-Current Sensing Fault (DF_IPHASE_SENSING_FAULT)</p> <p>This bit is set when a fault is detected in the motor phase current sensing circuitry.</p>
19	<p>Current Fault (DF_CURRENT_FAULT)</p> <p>This bit indicates that the drive's I²T current limit has been exceeded. The motor has been de-energized to protect it. The flag is reset once the decay time specified by the integrated I²T fuse has elapsed.</p>
18	<p>Service Required (DF_SERVICE_REQUIRED)</p> <p>This bit is set to 1 when maintenance or servicing of the gripper module is required. The condition does not necessarily indicate an immediate fault, but the system should be serviced at the next available opportunity. Set the gripper's operating manual for details.</p>
17	<p>Temperature Fault (DF_TEMP_FAULT)</p> <p>This bit is set to 1 when the measured gripper temperature exceeds a defined fault threshold. The system may continue to operate in this state, but it is highly recommended to immediately stop operation to prevent permanent damage on the device.</p>

16	<p>Temperature Warning (DF_TEMP_WARNING)</p> <p>This bit is set to 1 when the measured gripper temperature exceeds a defined warning threshold. The system may continue to operate in this state, but reduced load or increased monitoring is recommended to prevent overheating or a fault condition.</p>
15 to 1	reserved
0	<p>Referenced (DF_REFERENCED)</p> <p>This bit is set to 1 if the gripper has been successfully referenced (homed).</p>