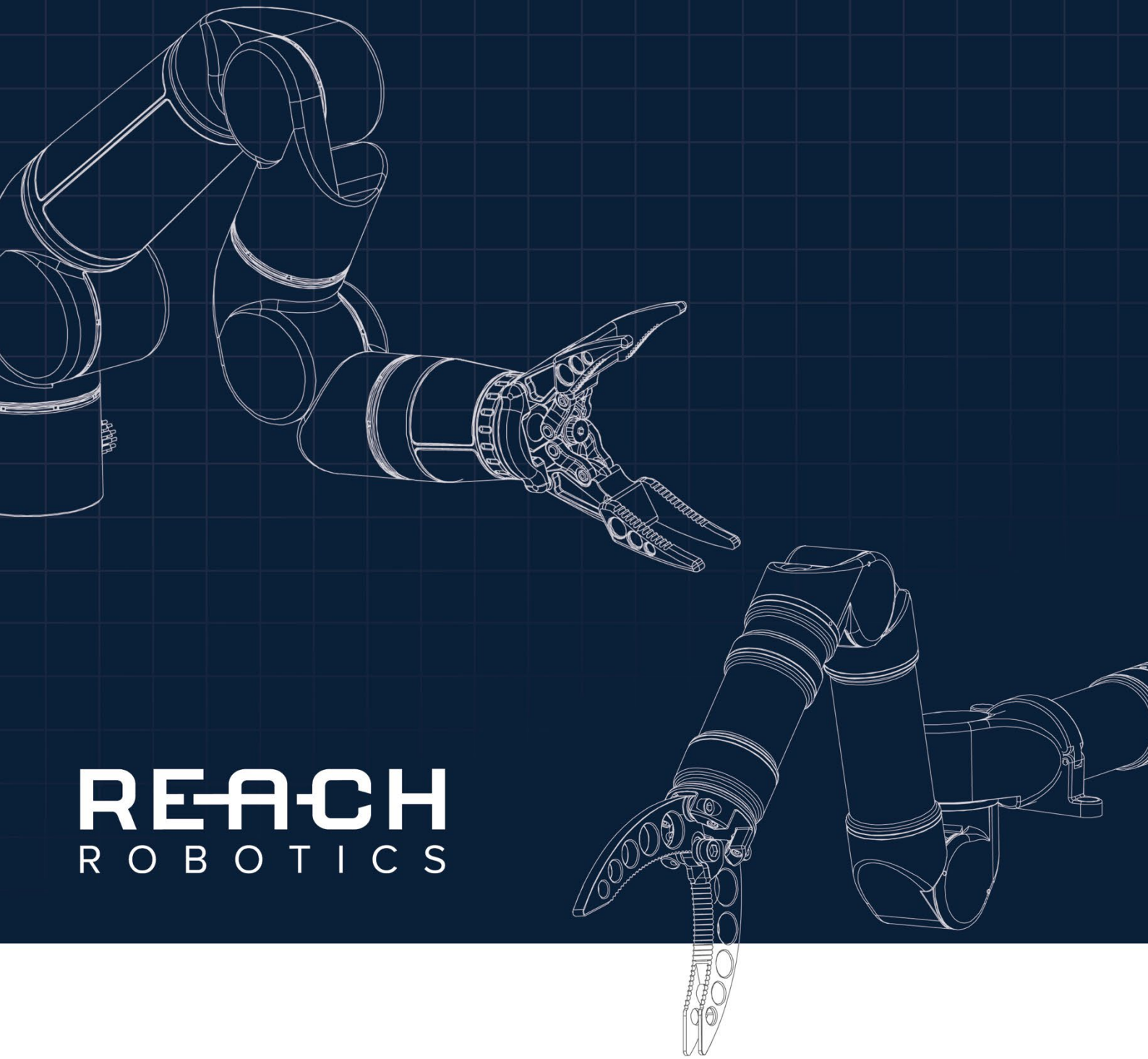

REACH ROBOTICS

REACH SYSTEM COMMUNICATION PROTOCOL

VERSION V1.12.1

info@reachrobotics.com | +61 (2) 9519 7651

reachrobotics.com

CONTENTS

1 INTRODUCTION

This document is designed to help users of Reach Robotics products integrate these into their control systems. It assumes access to the Reach Robotics Software Development Kit (SDK). Please contact sales@reachrobotics.com if you do not have access to the SDK.

Reach Robotics website and Help Centre
The resources tab on our website holds all our documentation as well as FAQs, knowledge articles, and downloads.

Reach Robotics contacts
If you can't find what you're looking for on our website or Help Centre, please get in touch with Reach Robotics Support at support@reachrobotics.com. You can also get in touch with us at sales@reachrobotics.com (for all sales enquiries), or info@reachrobotics.com (for any other enquiries).

Feedback
If anything in our manuals, FAQs or knowledge articles is out-of-date, poorly explained, or erroneous, please don't hesitate to let us know. We always appreciate the opportunity to improve our documentation for the benefit of all users.

2 PACKET STRUCTURE

OVERHEAD BYTE: used by the constant overhead byte stuffing (COBS) algorithm (see INFO below).

DATA: information contained within the packet. Different packets expect different data types. In all cases they are sent as individual bytes, and it is up to the encoding and decoding software to parse them accordingly. Packets cannot be larger than 64 bytes including the footer.

PACKET_ID: a unique identifier to tell the parsing software what the DATA refers to and how it should be interpreted. The packet ID is a single byte ranging from 0x00 to 0xFF.

DEVICE_ID: identifies the target device. Within one Reach System product there may be several actuators or peripherals with different device IDs. The device ID is a single byte between 0x00 and 0xFF. Refer to Section 4 for Device ID assignments.

LENGTH: total packet length (before COBS). Note: It is useful in checking for incomplete packets and in cases where communication is unreliable. After byte stuffing the packet is one byte larger than LENGTH.

8BIT_CRC: an 8-bit polynomial CRC (x^8+x^6+x^3+x^2+1 ; 0x14D or 0xA6 in Koopman notation) conducted over the full packet, excluding the terminator byte. This implementation is reflected, processing the least-significant bit first; the initial CRC value is 0xFF and the final value is exclusive-or'd with 0xFF. Refer to Section 3 for more information.

0x00: The terminating byte signifies the end of the packet stream. Note: byte stuffing ensures no other 0x00 appear in the data stream.


INFO

Separate packets are delimited with zero bytes. Using constant overhead byte stuffing (COBS) guarantees the message will not contain any zero bytes, so the end of packet is easily identified.

Table 1 Packet structure

	OVERHEAD BYTE	DATA	PACKET_ID	DEVICE_ID	LENGTH	8BIT_ CRC	0x00
Position	1	1:LENGTH-4	LENGTH-3	LENGTH-2	LENGTH-1	LENGTH	
Type	COBS byte	DATA	FOOTER				TERMINATOR
Example	09	9E EF 83 40	03	01	08	B8	00

3 CYCLIC REDUNDANCY CHECK (CRC)

CRC implementation in Reach Serial Protocol is as follows:

- Polynomial used for CRC table generation is 0x4D.
- Initial CRC value is 0x00.
- Final XOR value is 0xFF.
- Input data is reflected/reversed.
- Result is reflected/reversed (before final XOR step).

An example implementation is shown in Figure 1.

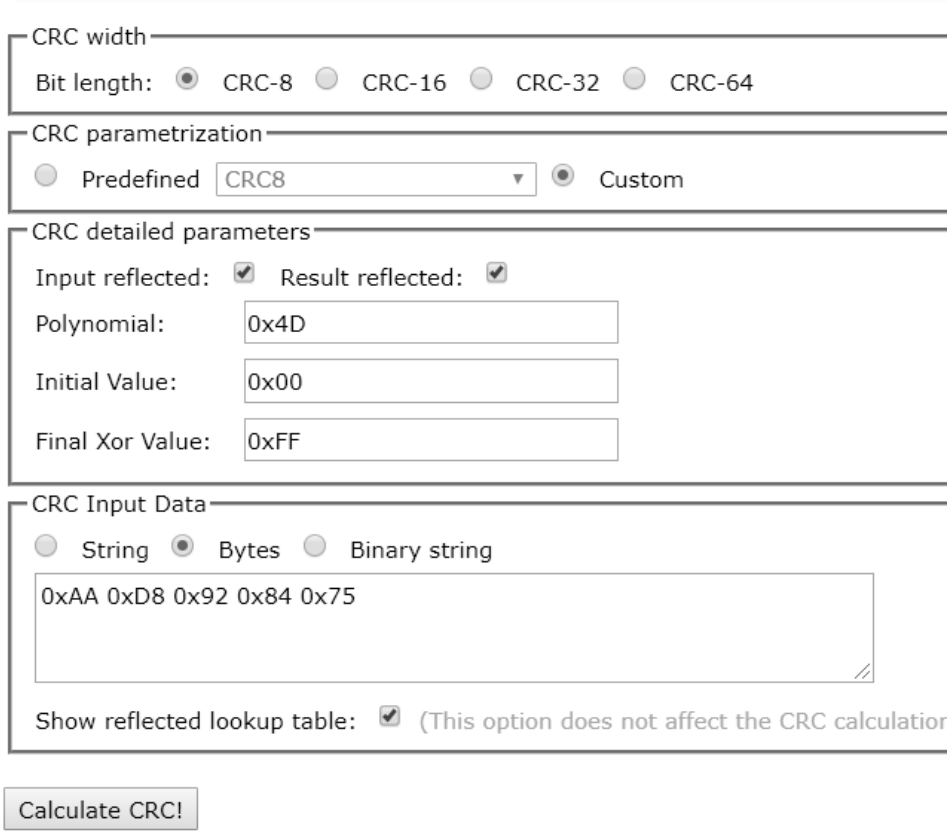

Figure 1: Screenshot from an online CRC calculator (access here).

4 DEVICE IDS

Reach Robotics products have a unique device ID for each embedded device in the product. Some embedded devices include:

- actuators,
- communication routers, and
- computers.

In general, actuators are given device IDs 0x01-0x07 as conveyed in Table 2.

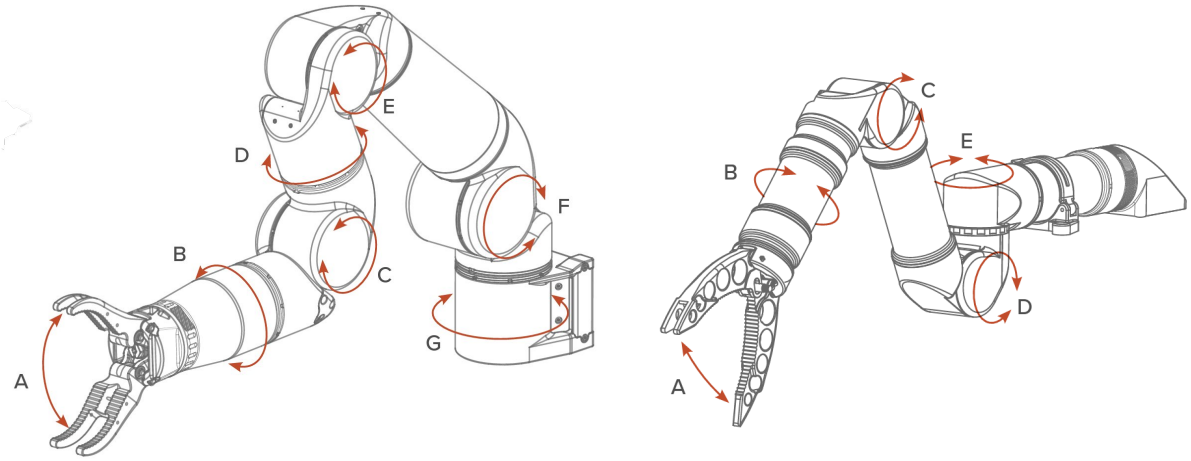

Figure 2: Axis names for 7- and 5-function manipulators.

Table 2 Actuator Device IDs

Function	Axis	DEVICE_ID
Function 1	A	0x01
Function 2	B	0x02
Function 3	C	0x03
Function 4	D	0x04
Function 5	E	0x05
Function 6	F	0x06
Function 7	G	0x07

Products with more than four functions require a compute device for advanced functionality such as inverse kinematics and collision avoidance. For Alpha manipulators, the base device is the final device in the chain (0x05). Advanced functionality for other manipulator lines is provided by a base computer with Device ID 0x0E. Additionally, devices with Ethernet support are also fitted with a communications router with Device ID 0xD. The base Device IDs for Reach Alpha and Bravo products are summarised below.

Table 3 Reach System Manipulator Device IDs

Product	Axes Device IDs	Base/Compute DEVICE_ID
Alpha 5	0x01, 0x02, 0x03, 0x04, 0x05	0x05
Bravo 5	0x01, 0x02, 0x03, 0x04, 0x05	0xD (base router), 0x0E (Compute)
Bravo 7	0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07	0xD (base router), 0x0E (Compute)

Packets can be communicated to all devices simultaneously using the device ID 0xFF.

WARNING

Care should be taken when requesting from 0xFF whilst using a RS-485 connection. This is because all devices will attempt to reply at the same time, resulting in packet loss due to bus collisions.

5 PACKET LISTING

Each listing in this section describes unique information for each packet in the Reach Robotics protocol. Each packet within the protocol is either an 8-bit unsigned integer or 32-bit floating point number.

5.1 Request

REQUEST

Request data from packet IDs. On receiving the command, the device will send the packet corresponding to the packet IDs in the data field. Up to 10 packets can be requested in a single packet request. Only packets capable of being transmitted can be requested. For example, transmitting a request for Velocity, Position and Current (0x02, 0x03, 0x05) in a single request packet will return three separate packets for Velocity data, Position data, and Current data.

Transmit: a list of packet IDs**.**
Receive: returns packet data for each packet ID.

Table 4 REQUEST Packet

Packet ID	Data	Transmit/Receive	Target Device
0x60	1:10 x bytes – Packet IDs	Transmit	All Devices

HEARTBEAT: PACKETS

Sets the packets to be sent at the specified HEARTBEAT: FREQUENCY. There are up to 10 packet allocations which correspond to requested packet IDs. All other values should be set to 0 if no packet is requested for that allocation. For example, to request position, velocity, and current, the data field of
HEARTBEAT: PACKETS would be [3, 2, 5, 0, 0, 0, 0, 0, 0, 0].


INFO

The HEARTBEAT SET packet cannot be saved; this needs to be re-set every power-cycle.

Transmit: desired packets to be sent at heartbeat frequency.
Receive: returns packet IDs currently set.

Table 5 HEARTBEAT: PACKETS Packet

Packet ID	Data	Transmit/Receive	Target Device
0x91	10 x bytes – Packet IDs	Transmit/Receive	All devices

HEARTBEAT: FREQUENCY

Sets the frequency of the HEARTBEAT: PACKETS to be sent periodically from the device. The byte value corresponds to the heartbeat frequency [Hz] (1-255). A value of 0 will disable the heartbeat. This value is automatically saved to the device configuration when sent.

Transmit: desired heartbeat frequency.
Receive: returns frequency currently set.

Table 6 HEARTBEAT: FREQUENCY Packet

Packet ID	Data	Transmit/Receive	Target Device
0x92	1 x byte – Frequency [Hz]	Transmit/Receive	All devices

5.2 Control

MODE

Request or set a device's operating mode. Receiving a control packet will automatically change the mode of operation. However, when in Disabled or Passive mode all control packets will be ignored.

Transmit: sets the operating mode of the device.
Receive: returns the current operating mode.

Table 7 Operating Modes

Mode	Hex Value
Standby	0x00
Disable	0x01
Position	0x02
Velocity	0x03
Current	0x04
Indexed Relative Position	0x13
Position Preset	0x14
Zero Velocity	0x15
Kinematic Position (Base Frame)	0x17
Kinematics Velocity (Base Frame)	0x18
Kinematics Velocity (End-effector Frame)	0x1A
Position Velocity	0x1C
Position Hold	0x1D
Passive	0x26

Table 8 MODE Packet

Packet ID	Data	Transmit/Receive	Target Device
0x01	1 x byte – Mode	Transmit/Receive	Axes devices

VELOCITY

Control an actuator's velocity. Demanding a velocity setpoint above the maximum limit will set the velocity to maximum.

Transmit: sets the velocity setpoint of the actuator in rad/s (rotate) and mm/s (linear).
Receive: returns the instantaneous velocity of the device.

Table 9 VELOCITY Packet

Packet ID	Data	Transmit/Receive	Target Device
0x02	1 x float – Velocity [rad/s]/[mm/s]	Transmit/Receive	Axes devices

POSITION

Control an actuator's position**.** Positions outside of the actuator's position limits are ignored.

Transmit: sets the absolute position setpoint of the actuator in rad (rotate) and mm (linear).
Receive: returns the instantaneous position of the device.

Table 10 POSITION Packet

Packet ID	Data	Transmit/Receive	Target Device
0x03	1 x float – Position [rad]/[mm]	Transmit/Receive	Axes devices

INDEXED RELATIVE POSITION

Control an actuator's position relative to a position index. The position index is captured from the manipulator's current position when transitioning to the Indexed Relative Position MODE.

Transmit: sets the relative position of actuator with respect to the position index.
Receive: returns the position relative to the position index.

Table 11 INDEXED RELATIVE POSITION Packet

Packet ID	Data	Transmit/Receive	Target Device
0x0D	1 x float – Position [rad] (wrapped between 0 and 2π)	Receive	Axes devices

CURRENT

Control an actuator's current in mA. Demanding current that is greater than the maximum limit will set the current to the maximum.


WARNING

Current control should be used with extreme care. Transitioning to current control MODE may cause the manipulator to suddenly drop to the ground since controllers are no longer maintaining position/velocity.

Transmit: sets the quadrature current setpoint of the motor in mA.
Receive: returns the instantaneous current of the device.

Table 12 CURRENT Packet

Packet ID	Data	Transmit/Receive	Target Device
0x05	1 x float – Current [mA]	Transmit/Receive	Axes devices

INVERSE KINEMATICS: GLOBAL POSITION

Control a manipulator's end-effector position with respect to its base coordinate frame. Translational position is given as X, Y, Z coordinates [mm]. Rotational position is given as Yaw, Pitch, Roll Euler angles [rad].


INFO

Inverse kinematics is not available for manipulators with less than 5 functions. Orbit controls (YPR) are not available for manipulators with less than 7 functions (values will be ignored). Additionally, inverse kinematics is disabled for standard manipulator packages.

Transmit: sets the end-effector pose with respect to the base frame.
Receive: returns the current global end-effector pose.

Table 13 INVERSE KINEMATICS: GLOBAL POSITION PACKET

Packet ID	Data	Transmit/Receive	Target Device
0xA1	6 x floats – [X, Y, Z, Y, P, R] Coordinates X, Y, Z [mm] Euler angles Y, P, R [rad]	Transmit/Receive	Base

INVERSE KINEMATICS: GLOBAL VELOCITY

Control a manipulator's end-effector velocity with respect to its base coordinate frame.

INFO

Inverse kinematics is not available for manipulators with less than 5 functions. Orbit controls are not available for manipulators with less than 7 functions (values will be ignored). Additionally, inverse kinematics is disabled for standard manipulator packages.

Transmit: sets the end-effector translational (X, Y, Z) and rotational (RX, RY, RZ) velocity with respect to the manipulator base frame. Rotational velocity is given as a rotation vector.
Receive: No data returned.

Table 14 INVERSE KINEMATICS: GLOBAL VELOCITY PACKET

Packet ID	Data	Transmit/Receive	Target Device
0xA2	6 x floats - [X, Y, Z, RX, RY, RZ] Translational velocity X, Y, Z [mm/s] Rotation rates RX, RY, RZ [rad/sec]	Transmit	Base

INVERSE KINEMATICS: LOCAL VELOCITY

Control a manipulator's end-effector velocity with respect to the end-effector's coordinate frame.

INFO

Local inverse kinematics is not available for manipulators with less than 7 functions. Additionally, inverse kinematics is disabled for standard manipulator packages.

Transmit: sets the end-effector translational (X, Y, Z) and rotational (RX, RY, RZ) velocity with respect to the manipulator end-effector frame. Rotational velocity is given as a rotation vector.
Receive: No data returned.

Table 15 INVERSE KINEMATICS: LOCAL VELOCITY PACKET

Packet ID	Data	Transmit/Receive	Target Device
0xCB	6 x floats - [X, Y, Z, RX, RY, RZ] Translational velocity X, Y, Z [mm/s] Rotation rates RX, RY, RZ [rad/sec]	Transmit	Base

INVERSE KINEMATICS: GLOBAL VELOCITY (LOCAL ROLL)

Control a manipulator's end-effector velocity with respect to the local coordinate frame (X), and base coordinate frame (Y, Z). This control mode is most intuitive when using manual controllers such as gamepad or space mouse controllers.

INFO

This packet is only available for Reach X manipulators.

Transmit: sets the end-effector translational (X, Y, Z) and rotational (RX, RY, RZ) velocity with respect to the manipulator end-effector/base frames. Rotational velocity is given as a rotation vector.
Receive: No data returned.

Table 16 INVERSE KINEMATICS: LOCAL VELOCITY PACKET (Local Roll)

Packet ID	Data	Transmit/Receive	Target Device
0xF4	6 x floats - [X, Y, Z, RX, RY, RZ] Translational velocity X, Y, Z [mm/s] Rotation rates RX, RY, RZ [rad/sec]	Transmit	Base

FORCE TORQUE SENSOR READING

Request force (N) and torque (Nm) from the force torque sensor.

Force torque sensor readings are only available for systems with an attached force torque sensor. If a force torque sensor is not attached to the system, no information will be returned.

Transmit: send 6 x float (0.0) to tare the sensor readings
Receive: returns the instantaneous force torque sensor reading with tared offset

Table 17 FORCE TORQUE SENSOR READING Packet

Packet ID	Data	Transmit/Receive	Target Device
0xD8	6 x floats – [FX, FY, FZ, TX, TY, TZ] Force FX, FY, FZ [N] Torque TX, TY, TZ [Nm]	Transmit/Receive	Base

5.3 Preset Positions

For manipulators with more than 4 functions, the target device is the base. For manipulators with 4 functions or less, presets must be set per actuator.

POSITION PRESET: GO

Go to a position preset. There are 4 allocations for position presets given by index [0:3].

This packet must be streamed at >10 Hz to continue driving to the preset position.

Transmit: sets position preset index to execute.
Receive: No data returned.

Table 18 POSITION PRESET: GO PACKET

Packet ID	Data	Transmit/Receive	Target Device
0x55	1 x byte - Preset index [0:3]	Transmit	Base

POSITION PRESET: CAPTURE

Save current position as a position preset for the selected index. There are 4 allocations for position presets given by index [0:3].

For manipulators with 5+ functions, the positions preset capture packet be sent to the Base device. For manipulators with 4 functions or fewer, the position preset capture packet must be sent to the individual axes one-by-one (the 0xFF device ID is not valid).

Transmit: sets position preset index to save.
Receive: No data returned.

Table 19 POSITION PRESET: CAPTURE PACKET

Packet ID	Data	Transmit/Receive	Target Device
0x56	1 x byte - Preset index [0:3]	Transmit	Base/Axes devices

POSITION PRESET: SET <INDEX>

Set a position preset. Allows the user to specify joint positions for a preset pose. There are 4 allocations for position presets given by packet ID [0x57-0x5A]. Each position corresponds to 1 axis of the target device starting from device ID 0x01.

Transmit: sets the preset positions for devices 0x01-0x08 respectively (see Section 4). Unused positions should be set to 0.
Receive: returns the current preset positions for devices 0x01-0x08 respectively.

Table 20 POSITION PRESET: SET PACKET

Packet ID	Data	Transmit/Receive	Target Device
0x57-0x5A	8 x floats – [A, B, C, D, E, F, G, H] Position [rad]	Transmit/Receive	Base

POSITION PRESET: NAME <INDEX>

Name a position preset. Allows the user to apply an ASCII name to a position preset. The text can be read by a user interface to identify the preset by name, e.g. “STOW” or “DEPLOY”. There are 4 allocations for position presets given by packet IDs 0x5B-0x5E. If the ASCII name is fewer than 8 bytes, the remaining bytes should be padded with 0x00.

Transmit: sets the position preset name in ASCII text.
Receive: returns the position preset name in ASCII text.

Table 21 POSITION PRESET: NAME PACKET

Packet ID	Data	Transmit/Receive	Target Device
0x5B-0x5E	8 x bytes – Preset name	Transmit/Receive	Base

5.4 Configure

The packets listed here are used to change the configuration settings of the device. All packets in Section 5.3 can be saved using the SAVE CONFIGURATION command if the value is required to persist over a system power cycle; otherwise they will be lost when the manipulator is powered down.

SAVE CONFIGURATION

Saves the current configuration to the files on the device. The SAVE CONFIGURATION packet should be sent individually to each joint.

Transmit: saves the current configuration to the device.
Receive: no data returned.

Table 22 SAVE CONFIGURATION Packet

Packet ID	Data	Transmit/Receive	Target Device
0x50	1 x byte – 0x00	Transmit	All devices

POSITION LIMITS

Configures custom minimum and maximum position limits. For a rotate device, the value is an angular position [rad] and on a linear device it is a linear position [mm]. These will not override the factory limits.

Transmit: sets max/min position limits.
Receive: returns current max/min position limits.

Table 23 POSITION LIMIT

Packet ID	Data	Transmit/Receive	Target Device
0x10	2 x floats – [max, min] Position [mm]/[rad]	Transmit/Receive	Axes Devices

VELOCITY LIMITS

Configures custom maximum and minimum velocity limits. For a rotate device, the value is an angular velocity [rad/s] and on a linear device it is a linear velocity [mm/s]. These will not override the factory limits.

Transmit: sets max/min velocity limits.
Receive: returns current max/min velocity limits.

Table 24 VELOCITY LIMIT Packet

Packet ID	Data	Transmit/Receive	Target Device
0x11	2 x floats – [max, min] Velocity [mm/s]/[rad/s]	Transmit/Receive	Axes Devices

CURRENT LIMITS

Configures custom maximum and minimum current limits [mA]. These will not override the factory limits.

Transmit: sets max/min current limits.
Receive: returns max/min current limits.

Table 25 CURRENT LIMIT Packet

Packet ID	Data	Transmit/Receive	Target Device
0x12	2 x floats – [max, min] Current [mA]	Transmit/Receive	Axes Devices

WORKSPACE RESTRICTIONS

Workspace restrictions can be set by defining box or cylinder obstacles in the environment around the manipulator, relative to the global frame. Box obstacles can be defined using an array that represents the coordinates of two diagonally opposed corners of the rectangle given as [X_1 Y_1 Z_1 X_2 Y_2 Z_2]. Cylinder obstacles can be defined using an array that represents the coordinates of the ends of the cylinder as well as a radius, given as [X_1 Y_1 Z_1 X_2 Y_2 Z_2 R] (see Figure 3). The defined obstacles represent regions into which the manipulator will not enter. Workplace Restrictions, such as obstacles and self-collision detection, are only utilised when operating in Velocity, Position, or a Kinematics control mode. Operating in Current Mode or Open-Loop Mode will ignore these restrictions.

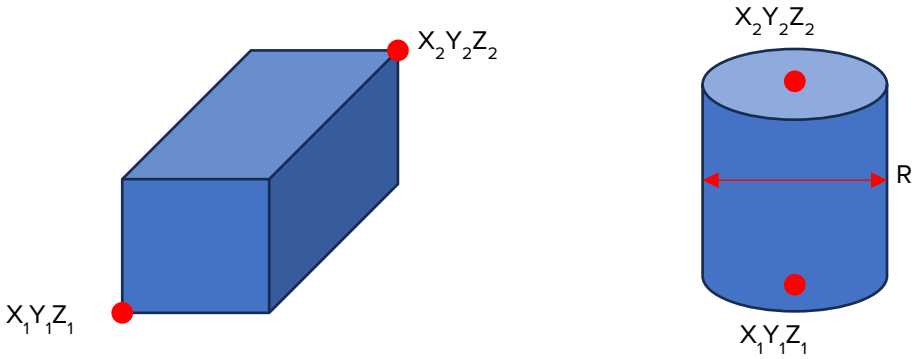

Figure 3: Visualisation of how box and cylinder obstacles are defined in the manipulator environment.

The workspace restrictions are automatically saved when they are sent to the manipulator. Up to four box-obstacles and four-cylinder obstacles can be saved. These are set using the packet IDs laid out in Table 26.

Workspace	Packet ID
BOX OBSTACLE 1	0xA5
BOX OBSTACLE 2	0xA6
BOX OBSTACLE 3	0xA7
BOX OBSTACLE 4	0xA8
CYLINDER OBSTACLE 1	0XAB
CYLINDER OBSTACLE 2	0XAC

CYLINDER OBSTACLE 3	0XAD
CYLINDER OBSTACLE 4	0XAE

Table 26 Obstacle packet IDs


INFO

Workspace restrictions are not available for manipulators with less than 5 functions.

Transmit: define and save an obstacle in the manipulator workspace.
Receive: return defined obstacle saved to a particular obstacle packet.

Table 27 BOX OBSTACLE Packet

Packet ID	Data	Transmit/Receive	Target Device
0xA5-0xA8	6 x floats – [X_1 Y_1 Z_1 X_2 Y_2 Z_2] Box coordinates [mm]	Transmit/Receive	Base

Table 28 CYLINDER OBSTACLE Packet

Packet ID	Data	Transmit/Receive	Target Device
0xAB-0xAE	7 x floats – [X_1 Y_1 Z_1 X_2 Y_2 Z_2, R] Cylinder coordinates [mm]	Transmit/Receive	Base

5.5 Monitor

VOLTAGE

The supply voltage of the device.

Transmit: command will be ignored.
Receive: returns the instantaneous supply voltage.

Table 29 VOLTAGE Packet

Packet ID	Data	Transmit/Receive	Target Device
0x90	1 x float – Supply voltage [V]	Receive	All Devices

INTERNAL HUMIDITY

A measure of internal relative humidity in a joint. Measured in percentage.

Transmit: command will be ignored.
Receive: returns the instantaneous humidity.

Table 30 INTERNAL HUMIDITY Packet

Packet ID	Data	Transmit/Receive	Target Device
0x65	1 x float – Relative humidity [%]	Receive	All Devices

INTERNAL TEMPERATURE

A measure of the internal temperature of the device. Measured in degrees Celsius.

Transmit: command will be ignored.
Receive: returns the instantaneous temperature.

Table 31 INTERNAL TENPERATURE Packet

Packet ID	Data	Transmit/Receive	Target Device
0x66	1 x float – Temperature [°C]	Receive	All Devices

INTERNAL PRESSURE

A measure of internal pressure in a joint. Measured in Bar.

Transmit: command will be ignored.
Receive: returns the instantaneous pressure.

Table 32 INTERNAL PRESSURE Packet

Packet ID	Data	Transmit/Receive	Target Device
0x67	1 x float – Pressure [Bar]	Receive	All Devices

FACTORY CLIMATE

A measure of reference climate parameters at completion of assembly. These values are compared to the current readings to determine if a leak has occurred. If a leak is detected, then the corresponding hardware status flags are set in HARDWARE STATUS FLAGS.

Transmit: command will be ignored.
Receive: returns the factory-set climate values.

Table 33 FACTORY CLIMATE Packet

Packet ID	Data	Transmit/Receive	Target Device
0x28	3 x floats – [T,P,H] Temperature T [°C] Pressure P [Bar] Relative humidity H [%]	Receive	Axes Devices

SOFTWARE VERSION

The software version of the current firmware loaded on the device in the form of three bytes for major, sub-major, and minor version numbers.

Transmit: command will be ignored.
Receive: return version of currently installed firmware.

Table 34 SOFTWARE VERSION Packet

Packet ID	Data	Transmit/Receive	Target Device
0x6C	3 x bytes - [Major, Sub-Major, Minor] Firmware version	Receive	All devices

HARDWARE STATUS FLAGS

A 32-bit list corresponding to flags that are set when hardware errors occur. Hardware status flags will transmit automatically at a frequency of 1 Hz if any errors are set.

Table 35 FACTORY CLIMATE Packet

Packet ID	Data	Transmit/Receive	Target Device
0x68	4 x bytes – [A, B, C, D]	Receive	Axes Devices

Table 36 Hardware STATUS FLAGS

Byte	Bit	Description
A	0x80	FLASH FAILED READ Failed to read from flash
	0x40	HARDWARE OVER HUMIDITY The humidity levels detected are over acceptable factory humidity levels
	0x20	HARDWARE OVER TEMPERATURE Joint temperature is over acceptable temperature levels.
	0x10	COMMS SERIAL ERROR Serial communication errors detected. This may be due to noise, or half duplex communication collisions.
	0x08	COMMS CRC ERROR Communication decoding errors detected. This may be due to noise, or half duplex communication collisions.
	0x04	MOTOR DRIVER FAULT [Alpha only] The motor driver is drawing too much current, or the voltage supply is too low.
	0x02	ENCODER POSITION ERROR Errors found in the joint's position encoder. Absolute position may be incorrect.
	0x01	ENCODER NOT DETECTED Joint's position encoder is not detected.
B	0x80	DEVICE AXIS CONFLICT Detected an incorrect setup in a device's kinematic chain. Device IDs must be in the correct order.
	0x40	MOTOR NOT CONNECTED Detected that the motor is not connected.
	0x20	MOTOR OVER CURRENT The motor is drawing too much current.
	0x10	INNER ENCODER POSITION ERROR Errors found in the inner encoder. Commutation of the joint may be affected.
	0x08	DEVICE ID CONFLICT Detected multiple devices with the same device ID.
	0x04	HARDWARE OVER PRESSURE Pressure levels detected are over the factory levels.
	0x02	MOTOR DRIVER OVER CURRENT AND UNDER VOLTAGE Motor driver is drawing too much current, or the voltage supply is too low.
	0x01	MOTOR DRIVER OVER TEMPERATURE The motor driver temperature is too high.
C	0x80	Unused
	0x40	Unused
	0x20	Unused
	0x10	JAW ZERO REQUIRED The jaws must be calibrated.
	0x08	JOINT SERVICE DUE [Bravo only] One or more of the joints have passed the factory-set service interval.
	0x04	READ PROTECTION ENABLED Updates can no longer be performed. Contact Support.
	0x02	ENCODER FAULT The joint may have developed backlash.

	0x01	Reserved
D	0x80	ENCODER POSITION INVALID [Factory Only]
	0x40	Unused
	0x20	Reserved
	0x10	LOW SUPPLY VOLTAGE The measured supply voltage is too low. Check the power supply is providing sufficient current and/or voltage.
	0x08	INVALID FIRMWARE Invalid 708 or 703 firmware is loaded on the Bravo for the compute module to use.
	0x04	Reserved
	0x02	CANBUS ERROR Errors found whilst communicating on the CANBUS.
	0x01	POSITION REPORT NOT RECEIVED The compute module is unable to get information from the joints.

6 REVISION HISTORY

Name	Revision	Date	Change
Paul Phillips	V1.1.0	01/09/2017	Made separate Document
Paul Phillips	V1.2.0	06/09/2017	Changed packet ID of position and velocity packets. Added CRC Polynomial.
Paul Phillips	V1.3.0	20/02/2018	Added R5M packets
Paul Phillips	V1.4.0	15/08/2018	Added further Reach 5 Mini Packets
Jean-Luc Stevens	V1.5.0	16/07/2019	Updated Heartbeat Frequency Packet ID
Paul Phillips	V1.6.0	24/10/2019	Added model number, version, and serial number packets
Jean-Luc Stevens	V1.7.0	05/12/2019	Added CRC section
Paul Phillips	V1.8.0	24/01/2020	Added relative positions and index position
John Sumskas	V1.9.0	07/08/2020	Added Local Coordinate/Velocity. Updated Packet Ids to reflect current system. Added Cylinder Obstacles Updated Packed IDS to be up to date.
John Sumskas	V1.10.0	22/03/2021	Modified Kinematic Rotation Information.
John Sumskas	V1.11.0	21/06/2021	Added Force Torque Sensor Packet Fixed incorrect information in Limits Added INTERNAL TEMPERATURE, INTERNAL HUMIDITY, INTERNAL PRESSURE, FACTORY CLIMATE, HARDWARE STATUS
John Sumskas	V1.11.1	14/07/2021	Fixed Packet ID Error on TEMPERATURE, FACTORY CLIMATE and HARWARE STATUS
John Sumskas	V1.11.2	03/08/2022	Fixed Table Reference in Section 3
John Sumskas	V1.11.3	12/01/2023	Updated Cover Pages for Reach Robotics
Kyle Mclean	V1.12.0	18/07/2023	Add position preset packets. Update formatting. Remove firmware upgrade section.
Kyle Mclean	V1.12.0	24/07/2023	Add always up frame packet
Kyle Mclean	V1.12.0	05/01/2023	Fix table reference. Refine kinematics description.
Kyle Mclean	V1.12.1	18/01/2023	Layout changes. Added context to multiple commands. Removed redundant package information.