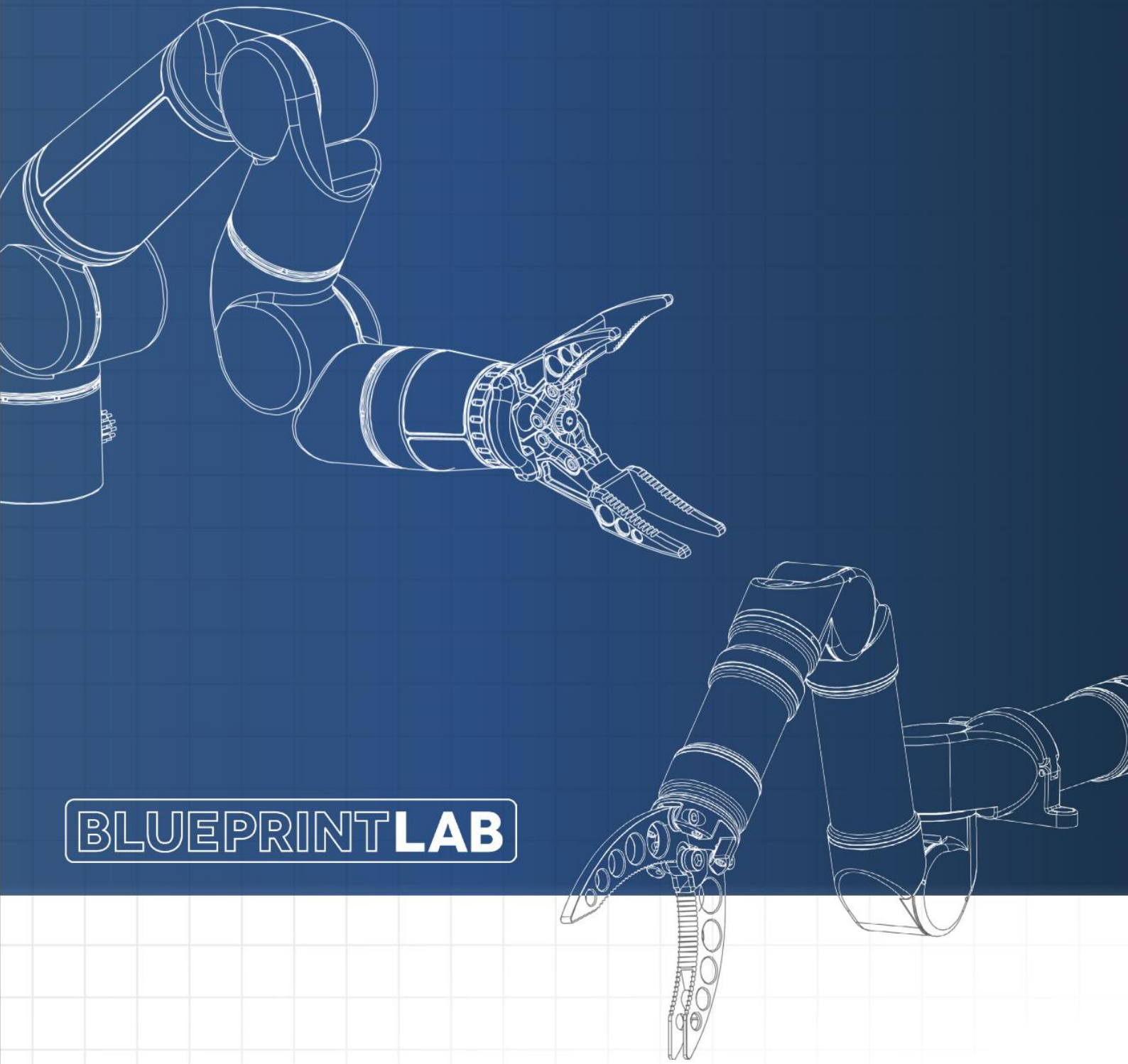

BLUEPRINTLAB

Reach System Communication Protocol

VERSION 1.9.0

LITE (No Kinematics)

Contact Blueprint Lab for Full Version

tel: +61 (2) 9519 7651 | email: info@blueprintlab.com

www.blueprintlab.com

1 Contents

2 OVERVIEW

Reach System is equipped with sophisticated electronics enabling it to perform a variety of functions without the need for external logic or feedback. Communication with the device is made via a full duplex RS232 connection using the Reach Communication Protocol laid out below. The protocol is designed to be expandable for multi axis configurations and robust in dealing with bad connections.

3 Packet Structure

The Packet structure is as follows

Table 1 Packet Structure

OVERHEAD BYTE	DATA	PACKET_ID	DEVICE_ID	LENGTH	8BIT_CRC	0x00
1	1:LENGTH-4	LENGTH-3	LENGTH-2	LENGTH-1	LENGTH	
COBS Byte	DATA	FOOTER				TERMINATOR

OVERHEAD BYTE: Based on an implementation of COBS (\https://en.wikipedia.org/wiki/Consistent_Overhead_Byte_Stuffing)

DATA: This is the data 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 254 bytes including the footer.

PACKET_ID: This is 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: This identifies to which device the packet is intended. Within one Reach System product there may be any number of actuators or peripherals. Certain device IDs are used to send commands to multiple devices where as others ensure only a single device will receive the command. The device ID is a single byte between 0x00 and 0xFF. Refer to Table 6 for Device ID assignments

LENGTH: The length of data and the 4 Byte footer in bytes. One is not required to use this in parsing the data as the footer length remains constant. 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: This is an 8 bit polynomial (x^8+x^6+x^3+x^2+1 ; 0x14D or 0xA6 in Koopman notation) CRC 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 4 CRC.

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

4 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 below.

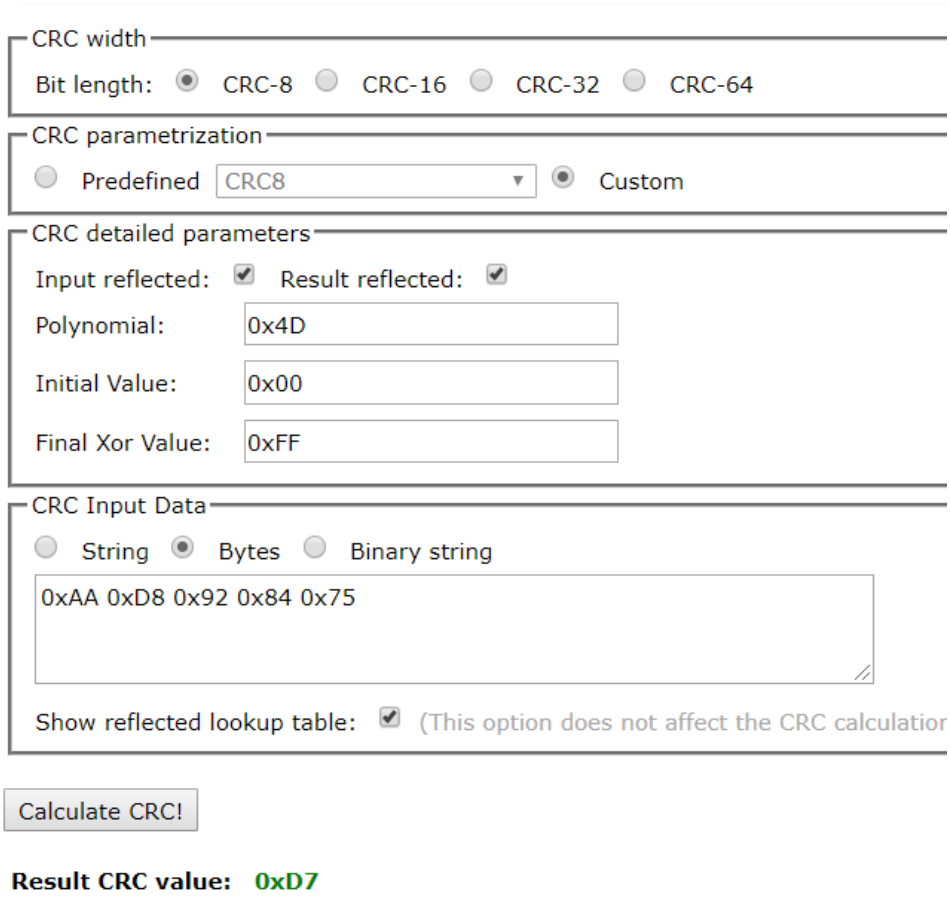

Figure 1: Online CRC calculator: http://www.sunshine2k.de/coding/javascript/crc/crc_js.html

NOTE This implementation demonstrates "MSB first" calculation. CRC calculation may be made more efficient by running calculations "LSB first."

5 Device ID's

Using Manipulators have a unique device ID for each joint, with a specified base. This base device can be the same as the base axis device. The base device processes the kinematics and related settings for the whole manipulator. There is a device id for each function of the manipulator, starting with 0x01 at the jaws incrementing as it propagates down to the base. Table 2 shows this with the Alpha 5 manipulator.
Lower function manipulators (4 function or lower) do not have a relevant specific base id. the appropriate device ID can be found for controlling the full range of Reach products. Some packets are axis or base specific and can only be parsed if sent to the appropriate device.

Manipulators have a unique device ID for each joint, with a specified base. This base device can be the same as the base axis device. The base device processes the kinematics and related settings for the whole manipulator. There is a device id for each function of the manipulator, starting with 0x01 at the jaws incrementing as it propagates down to the base. Table 2 shows this with the Alpha 5 manipulator.
Lower function manipulators (4 function or lower) do not have a relevant specific base id.

Table 2 Alpha 5 Device IDs

Function (from end)	Axis	DEVICE_ID
Function 1	Linear (Jaws Open Close)	0x01
Function 2	Rotate (Rotate End Effector)	0x02
Function 3	Bend (Elbow Bend)	0x03
Function 4	Bend (Shoulder Bend)	0x04

Function 5	Rotate (Base Rotate)	0x05
ALL	Sends to all functions	0xFF

Table 3 Reach System Manipulator Device IDs

Product	Axes Device IDs	BASE DEVICE_ID
Alpha 5	0x01, 0x02, 0x03, 0x04, 0x05	0x05
Bravo 5	0x01, 0x02, 0x03, 0x04, 0x05	0x0E
Bravo 7	0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07	0x0E

Packets can be communicated to all axis devices using the device ID 0xFF using once packet. This is similar for requesting data described in section 6.2.1. where you can request data from all devices using one packet.

6 Packet Listing

The packet listing pertains unique information for each packet receivable and transmittable by Reach System. The information is presented in the following way

Table 3 Packet Listing Meaning

Packet ID	Data	Transmit/Receive	Target Device
The unique packet identifier for every different type of packet.	The expected data to be contained within the packet	Whether the packet can be received, transmitted or both.	The devices this packet can be sent to.

6.1 Control

6.1.1 MODE

Sets the operating mode of the device. When in Standby, receiving a control command will automatically change the mode of operation. When disabled all other control commands will be ignored. The mode packet is useful in establishing in which mode the device is operating without sending a control demand.

Table 4 Operating Modes

Mode	Byte Value	Hex Value
Standby	0	0x00
Disable	1	0x01
Position	2	0x02
Velocity	3	0x03
Current	4	0x04

Table 5 MODE Packet

Packet ID	Data	Transmit/Receive	Target Device
0x01	1 Byte Byte value corresponding to the desired mode.	Transmit/Receive	Axes devices

6.1.2 VELOCITY

When sent sets the velocity setpoint of the actuator. When used with a rotational device it is an angular velocity in radians per second and when it is a linear device it is in mm per second. Demanding a velocity setpoint above the maximum limit will set the velocity to maximum.

When received it is the instantaneous velocity of the device.

Note: Sending any control command takes the device out of its previous mode and puts it into requested mode. It is not possible to be in two modes simultaneously.

Table 6 VELOCITY Packet

Packet ID	Data	Transmit/Receive	Target Device
0x02	1 float (4 bytes) 0 to +-MAX_VELOCITY	Transmit/Receive	Axes devices

6.1.3 POSITION

When sent sets the absolute position setpoint of the actuator. When used with a rotational device it is an angle between zero and 2PI and when it is a linear device it is a distance in mm. If the position setpoint is outside of the limits the command is ignored.

When received it is the instantaneous position of the device.

Table 7 POSITION Packet

Packet ID	Data	Transmit/Receive	Target Device
0x03	1 float (4 bytes) 0 to 2PI	Transmit/Receive	Axes devices

6.1.4 RELATIVE POSITION

When sent sets the relative position of actuator. The actuator will move from its current position the amount specified in the data. When received it is the amount moved since this packet was last sent.

Table 9 RELATIVE POSITION Packet

Packet ID	Data	Transmit/Receive	Target Device
0x0E	1 float (4 bytes) 0 to 2PI	Transmit/Receive	Axes devices

6.1.5 INDEXED POSITION

On first receiving indexed position an offset is created between the indexed position demand received and the current position. New indexed positions packets then move the actuators relative to the initial indexed position.

Indexed Position is similar to Relative Position except it has the advantage of not relying on all packets to be received in order to reach the required position.

Table 10 INDEXED POSITION Packet

Packet ID	Data	Transmit/Receive	Target Device
0x0D	1 float (4 bytes) 0 to 2PI	Receive	Axes devices

6.1.6 CURRENT

When sent sets the current setpoint of the motor windings in mAh. Demanding current that is out of range will set the current to maximum.

When received it is the instantaneous current of the device.

Note: This command should not be required regularly and is intended more as means of monitoring actuator torque.

Table 11 CURRENT PACKET

Packet ID	Data	Transmit/Receive	Target Device
0x05	1 float (4 bytes) 0 to +-MAX_CURRENT	Transmit/Receive	Axes devices

6.1.7 INVERSE KINEMATICS: GLOBAL POSITION

Inverse kinematic manipulators only.

Set the end effector coordinate with respect to the base. Three values representing X Y Z coordinate of the end effector in millimetres. If the demand is outside of the workspace the end effector will move to the closest possible position. Details on kinematic properties for the Alpha 5 can be found here.

Seven Function Manipulators only

Reach System seven function manipulators can move the end effector using X, Y, Z coordinates mentioned above as well as yaw, pitch and roll (YPR) rotation in radians.

Table 12 INVERSE KINEMATICS: GLOBAL POSITION PACKET

Packet ID	Data	Transmit/Receive	Target Device
	Full Version Only		

6.1.8 INVERSE KINEMATICS: GLOBAL VELOCITY

Inverse kinematic manipulators only.

Move the end effector along a prescribed vector given as XYZ velocities in mm per second.

Seven Function Manipulators only

Reach System seven function manipulators can also rotate the end effector in radians per second.

Table 13 INVERSE KINEMATICS: GLOBAL VELOCITY PACKET

Packet ID	Data	Transmit/Receive	Target Device
	Full Version Only		

6.1.9 INVERSE KINEMATICS: LOCAL POSITION

Seven Function Manipulators only

Set the end effector coordinate with respect to the current end effector position. Six values representing X, Y, Z in mm and Y, P, R in radians represent the target position of orientation and location of the end effector relative to the current position.

Table 14 INVERSE KINEMATICS: GLOBAL VELOCITY PACKET

Packet ID	Data	Transmit/Receive	Target Device
	Full Version Only		

6.1.10 INVERSE KINEMATICS: LOCAL VELOCITY

Seven Function Manipulators only

Move the manipulator along a prescribed vector given as an X, Y, Z velocity in mm/second as well as rotating the end effector in Y P R in radians per second. Movement is relative to the current position of the end effector.

Table 15 INVERSE KINEMATICS: GLOBAL VELOCITY PACKET

Packet ID	Data	Transmit/Receive	Target Device
	Full Version Only		

6.2 Monitor

6.2.1 REQUEST

Request a packet ID. 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.

Table 16 REQUEST Packet

Packet ID	Data	Transmit/Receive	Target Device
0x60	1 to 10 Bytes Bytes corresponding to valid Packets IDs	Transmit	Axes Devices

6.2.2 TEMPERATURE

The internal temperature of the device. Useful in monitoring motor winding temperatures under high loads. In cold environments, it is sometimes possible to overload the motor to achieve greater torque.

Table 17 TEMPERATURE Packet

Packet ID	Data	Transmit/Receive	Target Device
0x66	1 float (4 Bytes)	Receive	Axes Devices

	The internal temperature in °C		

6.2.3 VOLTAGE

The supply voltage of the device.

Table 18 VOLTAGE Packet

Packet ID	Data	Transmit/Receive	Axis
0x90	1 float (4 Bytes) The supply voltage in Volts	Receive	Axes Devices

6.2.4 SERIAL NUMBER

The unique Serial Number

Table 19 SERIAL NUMBER packet

Packet ID	Data	Transmit/Receive	Axis
0x61	1 float (4 Bytes) 4 digit Number	Receive	Axes Devices

6.2.5 MODEL NUMBER

Model Number of Device

Table 20 MODEL NUMBER Packet

Packet ID	Data	Transmit/Receive	Axis
0x62	1 float (4 Bytes) 4 Digit Number	Receive	All devices

6.2.6 SOFTWARE VERSION

Software Version of current firmware loaded on Device. Three bytes for major sub major and minor version number.

Table 21 SOFTWARE VERSION Packet

Packet ID	Data	Transmit/Receive	Axis
0x6C	3 Bytes Version Byte1.Byte2.Byte 3	Receive	All devices

6.3 Configure

The packets listed here are used to change the configuration settings of the device.

6.3.1 SAVE CONFIGURATION

When the SAVE packet is sent to a Device_ID target, it initiates SAVE on that Device_ID only.

Table 22 SOFTWARE VERSION Packet

Packet ID	Data	Transmit/Receive	Axis
0x50	None	Receive	All devices

6.3.2 HEARTBEAT FREQUENCY

Sets the frequency of the packets to be sent autonomously from the device. The available frequencies can be found in the Table 23 Heartbeat Frequency Selection

Table 23 Heartbeat Frequency Selection

Frequency	Byte Value	Hex Value
OFF	0	0x00
1 Hz	1	0x01
10 Hz	10	0x0A
50 Hz	50	0x32

Table 24 HEARTBEAT FREQUENCY Packet

Packet ID	Data	Transmit/Receive	Axis
0x92	1 Byte The Byte value corresponding to the required heartbeat frequency	Transmit/Receive	All devices

6.3.3 HEARTBEAT SET

Sets the packets to be sent at the specified heartbeat frequency. Setting the heartbeat frequency to OFF clears all the packets being sent. This is set to 0 (no heartbeat) by default.

Table 25 HEARTBEAT SET Packet

Packet ID	Data	Transmit/Receive	Axis
0x91	Up to 10 bytes pertaining to the required packets IDs to be sent	Transmit/Receive	All devices

6.3.4 POSITION LIMITS

Configures custom minimum and maximum position limits. These will not override the factory limits. In a rotary device, it is an angle in radians and on a linear device it is a position in mm.

Table 26 POSITION LIMIT

Packet ID	Data	Transmit/Receive	Axis
0x10	Two floats the first corresponding to the minimum position and the second to the maximum position	Transmit/Receive	Axes Devices

6.3.5 VELOCITY LIMITS

Configures the maximum velocity. These will not override the factory limits. In a rotary device it is an angular rate in radians/second and on a linear device it is a position in mm/second.

Table 27 VELOCITY LIMIT Packet

Packet ID	Data	Transmit/Receive	Axis

0x11	One float corresponding to the maximum velocity	Transmit/Receive	Axes Devices

6.3.6 CURRENT LIMITS

Configures the maximum current. These will not override the factory limits. It is a value in mAh.

Table 28 CURRENT LIMIT Packet

Packet ID	Data	Transmit/Receive	Axis
0x12	One float corresponding to the maximum current	Transmit/Receive	Axes Devices

6.3.7 WORKSPACE RESTRICTIONS

Kinematic Manipulators Only.

Sets workspace restrictions. Workspace restrictions can be set using cylinders or box obstacles and defining them around the environment of the manipulator.

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. The

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 neglect these restrictions.

The workspace restrictions are automatically saved to flash. Up to four box obstacles and four cylinder obstacles can be saved. These are set using the packet ID's laid out below.

Workspace	Packet ID
BOX OBSTACLE 1	
BOX OBSTACLE 2	
BOX OBSTACLE 3	
BOX OBSTACLE 4	
CYLINDER OBSTACLE 1	
CYLINDER OBSTACLE 2	
CYLINDER OBSTACLE 3	
CYLINDER OBSTACLE 4	

Table 29 BOX OBSTACLE Packet

Packet ID	Data	Transmit/Receive	Axis

Table 30 CYLINDER OBSTACLE Packet

Packet ID	Data	Transmit/Receive	Axis

7 Upgrading Firmware

The firmware is upgraded using STM32 bootloader specifications. These can either be integrated directly into a custom interface or run externally via STM's freely available *Flash Loader Demonstrator.*

Bootloader mode is entered when the following packet is received.

Table 31 BOOTLOADER Packet

Packet ID	Data	Transmit/Receive	Axis
0xFF	None	Transmit	Axis devices

To exit bootloader mode a device restart is required.

Documentation:
http://www.st.com/content/ccc/resource/technical/document/application_note/b9/9b/16/3a/12/1e/40/0c/CD00167594.pdf/files/CD00167594.pdf/jcr:content/translations/en.CD00167594.pdf

Alpha 5 Kinematics and Dynamic Properties: https://github.com/blueprint-lab/Blueprint_Lab_Software/raw/283c448a497ca70d2ee8a49aec40a7b90569c846/Documentation/Reach-System-1-Kinematic-and-Dynamic-Properties.pdf

Software: http://www.st.com/en/development-tools/flasher-stm32.html

8 REVISION HISTORY

Name	Revision	Date	Change
Paul Phillips	V1.1.0	1/9/17	Made separate Document
Paul Phillips	V1.2.0	6/9/17	Changed packet ID of position and velocity packets. Added CRC Polynomial.
Paul Phillips	V1.3.0	20/2/18	Added R5M packets
Paul Phillips	V1.4.0	15/8/18	Added further Reach 5 Mini Packets
Jean-Luc Stevens	V1.5.0	16/7/19	Updated Heartbeat Frequency Packet ID
Paul Phillips	V1.6.0	24/10/19	Added model number, version and serial number packets
Jean-Luc Stevens	V1.7.0	5/12/19	Added CRC section
Paul Phillips	V1.8.0	24/1/20	Added relative positions and index position
John Sumskas	V1.9.0	7/8/2020	Added Local Coordinate/ Velocity. Updated Packet Ids to reflect current system. Added Cylinder Obstacles Updated Packed IDS to be up to date.