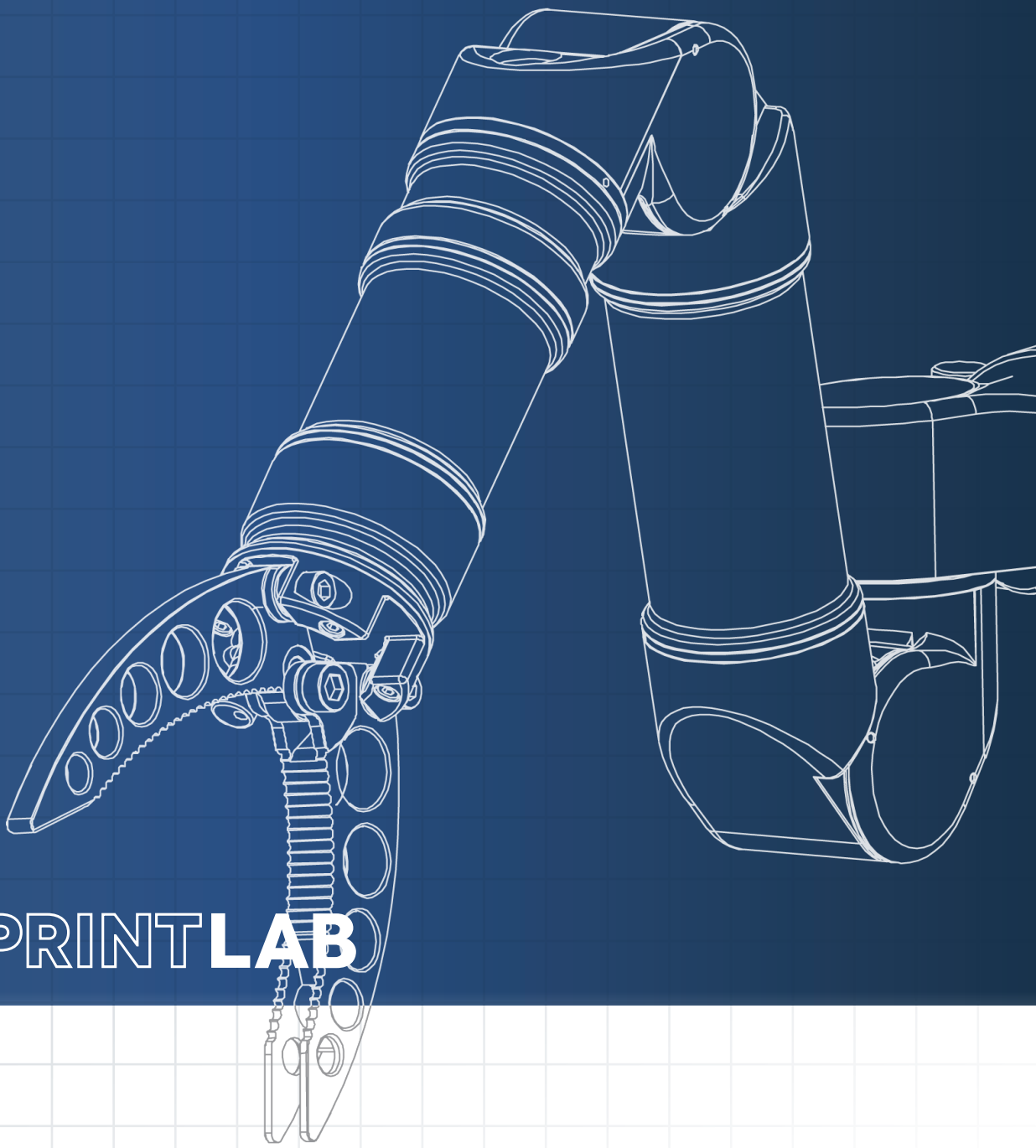

BLUEPRINTLAB

REACH SYSTEM

SERIAL PROTOCOL DOCUMENT

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

www.blueprintlab.com

1 Contents

2 OVERVIEW

RS1 is equipped with sophisticated electronics enabling it to perform a variety of function without the need for external logic or feedback. Communication with the device is made via a full duplex RS232 connection using

the RS1 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 RS1 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 RS1 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.

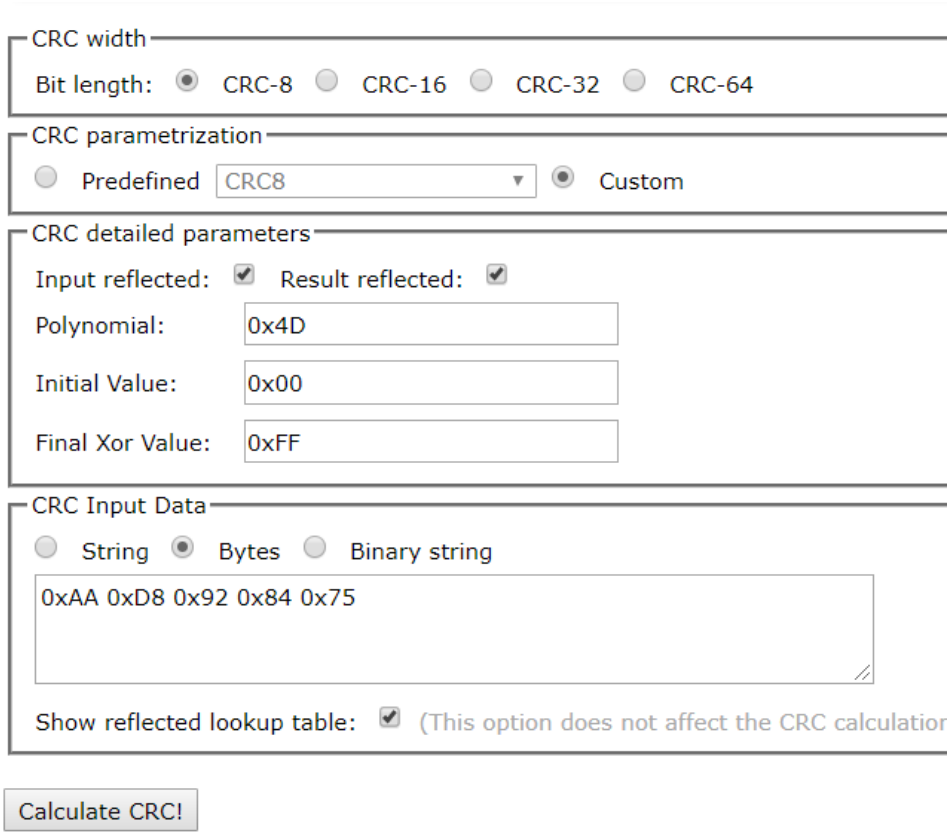

Figure 1: Online CRC calculator: http://www.sunshine2k.de/coding/javascript/crc/crc_js.html

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

5 Device ID’s

Using Table 2 Device IDs the appropriate device ID can be found for controlling the full range of RS1 products. Some commands are axis specific and in the case where a “both” Device ID is received the command will be sent to the axis represented by the first word.

When receiving packets the same is true. This provides a means of monitoring multiple axis with a minimum number of packet IDs.

Table 2 Device IDs

Product	Axis	DEVICE_ID
RS1-1100	Rotate	0x02
RS1-1200	Bend	0x03
RS1-1300	Linear	0x01
RS1-2120	Rotate	0x02
RS1-2120	Bend	0x03
RS1-2120	Both	0xFF
RS1-2130	Rotate	0x02
RS1-2130	Linear	0x01
RS1-2130	Both	0xFF

Table 3 Reach 5 Mini 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

6 Packet Listing

The packet listing pertains unique information for each packet receivable and transmittable by the RS1 Series. The information is presented in the following way

Table 4 Packet Listing Meaning

Packet ID	Data	Transmit/Receive	Axis
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.	Whether the packet can be sent to multiple axis

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 5 Operating Modes

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

Table 6 MODE Packet

Packet ID	Data	Transmit/Receive	Axis
0x01	1 Byte Byte value corresponding to the desired mode.	Transmit/Receive	Single/Both

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 7 VELOCITY Packet

Packet ID	Data	Transmit/Receive	Axis
0x02	1 float (4 bytes) 0 to +-MAX_VELOCITY	Transmit/Receive	Single/Both

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 8 POSITION Packet

Packet ID	Data	Transmit/Receive	Axis
0x03	1 float (4 bytes) 0 to 2PI	Transmit/Receive	Single/Both

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 POSITION Packet

Packet ID	Data	Transmit/Receive	Axis
0x0E	1 float (4 bytes) 0 to 2PI	Transmit/Receive	Single/Both

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 POSITION Packet

Packet ID	Data	Transmit/Receive	Axis
0x0D	1 float (4 bytes) 0 to 2PI	Receive	Single/Both

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	Axis
0x05	1 float (4 bytes) 0 to +-MAX_CURRENT	Transmit/Receive	Single/Both

6.1.7 COORDINATE – Preliminary

Reach Mini Function 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.

Packet ID	Data	Transmit/Receive	Axis
0xA1	3 floats of X Y Z position	Transmit/Receive	Single/Both

6.1.8 VECTOR – Preliminary

Reach 5 Mini Function only.

Move the end effector along a prescribed vector given as XYZ velocities in mm/second. Please see Reach 5 Mini Integration manual for workspace frame description.

Packet ID	Data	Transmit/Receive	Axis
0xA2	3 floats of X Y Z velocities	Transmit/Receive	Single/Both

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 12 REQUEST Packet

Packet ID	Data	Transmit/Receive	Axis
0x60	1 to 10 Bytes Bytes corresponding to valid Packets IDs	Transmit	Single/Both

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 13 TEMPERATURE Packet

Packet ID	Data	Transmit/Receive	Axis
0x65	1 float (4 Bytes) The internal temperature in °C	Receive	Single/Both

6.2.3 VOLTAGE

The supply voltage of the device.

Table 14 VOLTAGE Packet

Packet ID	Data	Transmit/Receive	Axis
0x66	1 float (4 Bytes) The supply voltage in Volts	Receive	Single

6.2.4 SERIAL NUMBER

The unique Serial Number

Table 15 SERIAL NUMBER packet

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

6.2.5 MODEL NUMBER

Model Number of Device

Table 16 MODEL NUMBER Packet

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

6.2.6 SOFTWARE VERSION

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

Table 17 SOFTWARE VERSION Packet

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

6.3 Configure

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

6.3.1 SAVE CONFIGURATION

Saves the current configuration to flash

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 18 Heartbeat Frequency Selection

Table 18 Heartbeat Frequency Selection

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

Table 19 HEARTBEAT FREQUENCY Packet

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

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.

By default, the device streams position, velocity, current and supply voltage of both axes at 50HZ

Table 20 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	Single/Both

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 21 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	Single

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 22 VELOCITY LIMIT Packet

Packet ID	Data	Transmit/Receive	Axis

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

6.3.6 CURRENT LIMITS

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

Table 23 CURRENT LIMIT Packet

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

6.3.7 RECTANGULAR WORKSPACE RESTRICTIONS- Preliminary

Sets workspace restrictions. The value 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. The rectangle represents a region into which the Reach 5 mini will not enter. The zero points is taken from the rotation point of the first axis. Workspace restrictions are utilised only when controlling the arm using a kinematic velocity or position demand.

The workspace restrictions are automatically saved to flash. Up to three rectangular workspace restrictions can be saved. These are set using the packet ID's laid out below.

Workspace	Packet ID
WORKSPACE RECTANGLE 1	0xB1
WORKSPACE RECTANGLE 2	0xB2
WORKSPACE RECTANGLE 3	0xB2

Packet ID	Data	Transmit/Receive	Axis
0xB1-3	6 floats (X_1 Y_1 Z_1 X_2 Y_2 Z_2)	Transmit/Receive	Single

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 24 BOOTLOADER Packet

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

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

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	24.1.19	Added relative positions and index position