Progwhiz.com

Robotics Server API

Rev 1.8

Table of Contents

Introduction

The following document outlines the command packets compatible with the Progwhiz Robotics Server Module. This API will enable software developers to create applications on various Operating Systems and Devices to interact with the following electronic peripherals:

- ✓ Servos
- ✓ Switches
- ✓ Sensors

API Definition

The API is a bi-directional protocol. This protocol supports sending commands to the server as well as receiving feedback from the server, the classes are defined into the following groups:

API Sending

- Fetch Configuration Settings
- Normal Send to Channel/Port
- Activate/Close Switch
- De-Activate/Open Switch
- Play Internal Script
- Stop Internal Script
- Play External Script
- Stop External Script

API Receiving

- Sensor Feedback
- Configuration Settings

API Message Descriptions

The following descriptions will show the format of each command and the meaning and value sets of each parameter. Please note the messages are passed as textual strings. The rules are as follows:

- Brackets [ ] denotes a single Byte passed as a single character
- No spaces allowed
- Single digit Values shown once not in Brackets [ ] are each a single byte
- (for e.g. 20 is really two(2) bytes which are 50 and 48 and the 20 is a textual representation)

Fetch Configuration Settings

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 2Bytes + 2Bytes + 1Byte + 2Bytes + 1Byte + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = 24
Data2 = 0
Data3 = 0

Example

1. rr1-24--0--0-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	24	--	0	--	0	-ss1

Fetch Camera Image

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 2Bytes + 2Bytes + 1Byte + 2Bytes + 1Byte + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = 29
Data2 = 0
Data3 = 0

Example

1. rr1-29--0--0-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	29	--	0	--	0	-ss1

Normal Send to Channel/Port

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1-2Bytes + 2Bytes + 1-4Bytes + 2Bytes + 2Bytes + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = Channel# (0 to 23)
Data2 = 0 to 2000
Data3 = 0

Example

1. rr1-0--60--0-ss1
2. rr1-2--72--0-ss1
3. rr1-14--9--0-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	60	--	0	-ss1

Example 2 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	2	--	72	--	0	-ss1

Example 3 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	14	--	9	--	0	-ss1

Activate/Close Switch

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1-2Bytes + 2Bytes + 1Byte + 2Bytes + 1Byte + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = Channel# (0 to 23)
Data2 = 1 (Turn On)
Data3 = 0

Example

1. rr1-0--1--0-ss1
2. rr1-2--1--0-ss1
3. rr1-14--1--0-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	1	--	0	-ss1

Example 2 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	2	--	1	--	0	-ss1

Example 3 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	14	--	1	--	0	-ss1

De-Activate/Open Switch

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1-2Bytes + 2Bytes + 1Byte + 2Bytes + 1Byte + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = Channel# (0 to 23)
Data2 = 0 (Turn Off)
Data3 = 0

Example

1. rr1-0--0--0-ss1
2. rr1-2--0--0-ss1
3. rr1-14--0--0-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	0	-ss1

Example 2 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	2	--	0	--	0	-ss1

Example 3 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	14	--	0	--	0	-ss1

Play Internal Script

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1Byte + 2Bytes + 1-4Bytes + 2Bytes + 1-3Bytes + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = 0
Data2 = 0
Data3 = Script# (11 to 266)

Example

1. rr1-0--0--11-ss1
2. rr1-0--0--23-ss1
3. rr1-0--0--120-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	11	-ss1

Example 2 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	23	-ss1

Example 3 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	120	-ss1

Stop Internal Script

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1-2Bytes + 2Bytes + 1-4Bytes + 2Bytes + 1-3Bytes + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = 0
Data2 = 0
Data3 = 267

Example

1. rr1-0--0--267-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	267	-ss1

Play External Script

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1Byte + 2Bytes + 1-4Bytes + 2Bytes + 1-3Bytes + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = 0
Data2 = 0
Data3 = Script# (270 to 275)

Example

1. rr1-0--0--270-ss1
2. rr1-0--0--272-ss1
3. rr1-0--0--275-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	270	-ss1

Example 2 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	272	-ss1

Example 3 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	275	-ss1

Stop External Script

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1-2Bytes + 2Bytes + 1-4Bytes + 2Bytes + 1-3Bytes + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = 0
Data2 = 0
Data3 = 269

Example

1. rr1-0--0--269-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	269	-ss1

Sensor Request

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1-2Bytes + 2Bytes + 1-3Bytes + 2Bytes + 1Byte + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = Channel# (0 to 23)
Data2 = 0
Data3 = 0

Example

1. rr1-0--0--0-ss1
2. rr1-2--0--0-ss1
3. rr1-3--0--0-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	0	-ss1

Example 2 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	2	--	0	--	0	-ss1

Example 3 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	3	--	0	--	0	-ss1

Servo Position Request

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1-2Bytes + 2Bytes + 1-3Bytes + 2Bytes + 1Byte + 4Bytes

Header = rr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = Channel# (0 to 23)
Data2 = 0
Data3 = 2

Example

1. rr1-0--0--2-ss1
2. rr1-2--0--2-ss1
3. rr1-3--0--2-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	0	--	0	--	2	-ss1

Example 2 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	2	--	0	--	2	-ss1

Example 3 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
rr1-	3	--	0	--	2	-ss1

Sensor/Servo Response

Format

Description: Header + Data1 + Separator + Data2 + Separator + Data3 + Trailer
Layout: 4Bytes + 1-2Bytes + 2Bytes + 1-3Bytes + 2Bytes + 1Byte + 4Bytes

Header = cr1-
Trailer = -ss1
Separator = -- (Double Dash)

Data1 = Channel# (0 to 23)
Data2 = Sensor/Servo Data (0 to 1024, 0 to 8000 respectively)
Data3 = 0

N.B. Servo Position is Servo Data / 4

Example

4. cr1-0--0--0-ss1
5. cr1-2--100--0-ss1
6. cr1-2--220--0-ss1

Example 1 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
cr1-	0	--	0	--	0	-ss1

Example 2 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
cr1-	2	--	100	--	0	-ss1

Example 3 Structure

Header	Data1	Separator	Data2	Separator	Data3	Trailer
cr1-	2	--	220	--	0	-ss1

Configuration Settings

Format

Description: Header + Data1 + (Data2.0 + Data2.1 + Data2.2 …. Data2.N-2 + Data2.N-1 + Data2.N + Trailer
Layout: 4Bytes + 1Byte + (1Byte + 1Byte + 1Byte …) + 4Bytes

Header = sr1-
Trailer = -ss1

Data1 = Channel Max# (0 to 23)
Data2.0 = Unit Type (0 - Servo, 1 - Switch, 2 - Sensor)
Data2.1 = Not Used (Def = 0)
Data2.2 = Switch Type (0 or 1)

Example

1. sr1-[6]000000000000000000-ss1
2. sr1-[12]001001001001001001001001001001001001-ss1
3. sr1-[12]200200200200200200200200200200200200-ss1
4. sr1-[12]101101101101101101101101101101101101-ss1

Example1 – 6 Servos
Example2 – 12 Servos
Example3 – 12 Sensors
Example3 – 12 Switches

Example 1 Structure

Header	S1	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	Trailer
sr1-	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-ss1

Data1 = S1
Data2.0 = D1
Data2.1 = D2 etc

Set Configuration Settings

Format

Description: Header + Data1 + (Data2.0 + Data2.1 + Data2.2 …. Data2.N-2 + Data2.N-1 + Data2.N + Trailer
Layout: 4Bytes + 1Byte + (1Byte + 1Byte + 1Byte …) + (Chksum Byte) + 1Byte + 4Bytes

Header = sr1-
Trailer = -ss1

Data1 = 25
Data2.0 = Dev# (0 to 127, Default=12)
Data2.1 = Port# (0 to 23)
Data2.2 = Unit Type (0 - Servo, 1 - Switch, 2 - Sensor)

Example

1. sr1-25--[12][0][0] [12][1][0] [12][2][0] [12][3][0] [12][4][0] [12][5][0][72]--0-ss1

Example1 – Device 12 with 6 Servos (using Ports 0 to 5)

Example 1 Structure

Header	S1	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	Trailer
sr1-	25	12	0	0	12	1	0	12	2	0	12	3	0	12	4	0	12	5	0	72	0	-ss1

Data1 = S1
Data2.0 = D1
Data2.1 = D2
Data2.2 = D3

Functional Flow

There are two (2) methods to establish connections with the API:

- **VPN Client / Server**
- **Winsock Client / Server**

VPN Client

1) Connect to Server (IP, Port and Password configured properly)
2) Request Server Configuration Settings
3) Receive Configuration Settings if connection was **validated**

Winsock Client

1) Connect to Server (IP, Port and Password configured properly)
2) Wait for ***'sendchkpass'*** message from Server
3) Respond with ***<password>***
4) Receive ***'readytogo'*** if valid password or will respond with ***'Failed Validation'***
5) Request Server Configuration Settings if response was ***'readytogo'***
6) Receive Configuration Settings if invalid password will respond with ***'Failed Validation'***