F90 Series

Projector Control Protocol Manual
Setup and Commands

601-0421

R601-0421/00
28/09/2016

BARCO

Visibly yours

Barco Fredrikstad AS
Habornveien 53, N-1630 Gamle Fredrikstad, Norway
Phone: +47 6930 4550
Fax: +47 6930 4580
Support: Support.fre@barco.com
Visit us at the web: www.barco.com

Printed in Norway

1. SETTING UP LAN COMMUNICATION

1.1 Input description

Input

Set up parameters

Description	Value
IP Version	IPv4
DHCP	On
IP Adress	-
Subnet Mask	-
Default Gateway	-
TCP Port	9090

About Setting Up LAN

The projector is shipped with a set of default settings, these are as follows:

Before you connect the projector to your LAN make sure that the IP settings are set correct according to your LAN configuration.

IP settings can be changed from OSD. It can be set to automatic (DHCP - Dynamic Host Configuration Protocol) or manual. In manual mode the IP address, Subnet mask and Gateway must be applied by the user.

When enabling DHCP please allow up to a minute for the projector to receive IP settings from the DHCP server. The IP address will be updated and shown in the OSD.

To renew an IP address, select “apply” in the OSD.

Connect the projector to the network using the Ethernet connector (RJ45).

2. SETTING UP RS232 COMMUNICATION

Overview

- Connect to the projector
- RS232 Communication Parameters

2.1 Connect to the projector

About

Connect the projector and host using a standard serial cable with 9-pin female to the host, and 9-pin male to the projector. Pin 2 connects to pin 2, pin 3 connects to pin 3 and pin 5 connects to pin 5.

2.2 RS232 Communication Parameters

About

Table 1 shows supported RS232 settings:

Parameter	Data
Baud Rate	19200
Parity	None
Databits	8
Stopbits	1
Flowcontrol	None

Table 2-1
RS232 parameters

3. COMMUNICATION PROTOCOL

This section describes how to use the communication protocol to control projectors remotely.

When the projects are connected to either RS232 or LAN you can control the projectors through this UTF-8 based protocol.

The communication protocol is based on JSON-RPC 2.0. Through this protocol external clients can access the internal API of the projector. It uses JSON (RFC 4627) as data format.

JSON-RPC 2.0 specification can be found here: http://www.jsonrpc.org/specification

3.1 Overview

Overview

The projector supports execute, read and write operations. See a protocol example below.

--> indicates data sent to projector

<-- indicates data received from the projector

```
-->
{
  "jsonrpc": "2.0",
  "method":"property.get",
  "params": {"property": "objectname.propertyname"},
  "id": 4
}

<--
{
  "jsonrpc": "2.0",
  "result": 100,
  "id": 4
}
```

“Method” and “params” are named using dot notation in lowercase format (Javascript-like notation). Please note starting “{“ and ending “}” must be included.

Table 2 shows the members of a JSON request.

Property	Type	Comments
jsonrpc	string	Must be exactly "2.0"
method	string	A string containing the name of the method to be invoked.
params	string	Structured value that holds the parameter values to be used during the invocation of the method.
id	string or	Request id. Can be any number or string. To be used to
	number	identify response. If id is omitted, the command will be executed without any response.

Table 3-1
JSON-RPC Parameters

3.2 Examples

Below are some examples for the different type of commands

3.3 Write

Figure 1shows how to set laser power (illumination.sources.laser.power) to 59%.

```
-->
```

```
{
  "jsonrpc": "2.0",
  "method": "property.set",
  "params": {
    "property": "illumination.sources.laser.power", "value": 59.0
  },
  "id": 1
}

<--
{
  "jsonrpc":"2.0",
  "id":1,
  "result":true
}
```

3.4 Read

Figure 2 shows how to read laser power status from the projector.

```
1

       -->
{
  "jsonrpc": "2.0",
  "method": "property.get",
  "params": {"property": "illumination.sources.laser.power"},
  "id": 5
}
                <--
{
  "jsonrpc":"2.0",
  "id":5,
  "result":59.0
}
```

Figure below shows how to read multiple parameters from the projector. In this example laser power (illumination.sources.laser.power) and laser power state (illumination.state) are requested as an example. See the table at the end of this document for available commands.

```
-->
{
  "jsonrpc": "2.0",
  "method":"property.get",
  "params": {
    "property":
      ["illumination.sources.laser.power",
       "illumination.state"]
  },
  "id": 5
}
<--
{
  "jsonrpc":"2.0",
  "id":5,
  "result":{
    "illumination.sources.laser.power":59.0,
    "illumination.state":"On"
  }
}
```

Figure 3 shows how to read network information (multiple entries).

```
-->
{
  "jsonrpc": "2.0",
  "method": "property.get",
  "params": {
    "property": [
      "network.device.lan.configuration",
      "network.device.lan.devicetype",
      "network.device.lan.hwaddress",
      "network.device.lan.ip4config",
      "network.device.lan.ip4configmanual",
      "network.device.lan.state"
    ]
  },
  "id": 36
}

<--
```

1. Figure 2

```
{
  "jsonrpc":"2.0",
  "id":36,
  "result":{
    "network.device.lan.configuration":"AUTO",
    "network.device.lan.devicetype":"WIRED",
    "network.device.lan.hwaddress":"00:0D:0A:01:64:39",
    "network.device.lan.ip4config":{
      "Address":"10.163.8.235",
      "Mask":"255.255.252.0",
      "Gateway":"10.163.8.1",
      "NameServers":"10.163.224.11,10.163.224.12,10.197.192.11"},
      "network.device.lan.ip4configmanual":{
        "Address":"192.168.1.100",
        "Mask":"255.255.255.0",
        "Gateway":"192.168.1.1",
        "NameServers":""},
      "network.device.lan.state":"CONNECTED"}
}
```


3.5 Execute

Figure 4 shows how to execute “power on” command.

```
-->
{
  "jsonrpc": "2.0",
  "method": "system.poweron",
  "params": {},
  "id": 12
}

<--
{
  "jsonrpc": "2.0",
  "id": 12,
  "result": null
}
```

3.6 Error Response

If an error occurs when sending JSON commands the response will include an error. Figure 5 shows and example where the method name is wrong. The error code received is code 1 - Method not found. See Table for possible error codes.

```
-->
{
  "jsonrpc": "2.0",
  "method": "property.gest",
  "params": {
    "property": "image.brightness"
  },
  "id": 0
}

<--
{
  "jsonrpc":"2.0",
  "id":0,
  "error":{"code":32601, "message":"Method not found: property.gest"}
}
```

Error Codes

Code	Message	Meaning
-32700	Parse error	Invalid JSON was received by the server. An error occurred on the server while parsing the JSON text.
-32600	Invalid Request	The JSON sent is not a valid Request object.
-32601	Method not found	The method does not exist / is not available.
-32602	Invalid params	Invalid method parameter(s).

Code	Message	Meaning
-32603	Internal error	Internal JSON-RPC error.
-32000 to -32099	Server error	Reserved for implementation-defined server-errors.

3.7 available Commands

Available Commands

See appendix for a table of commands.

PRODUCT/PROJECT NAME

DOCUMENT

F90 JSON commands

PRODUCT/PROJECT NUMBER

-

REVISION HISTORY

REV.	CREATED		APPROVED		PAGES	DESCRIPTION
	DATE	SIGN	DATE	SIGN		
1.0	071016					Initial document
2.0	270117	GO				Updated for stage 1

	Command		Method	Params {}	Description	
Power	Power on	Turn On	"system.poweron"			
	Power off	Turn Off	"system.poweroff"			
	State	Read	"property.get"	"property": system.state"	Value: Ready On	

Source	Select source	Read	"property.get"	"property": "image.window.main.source"	Value: "DVI 1" "DVI 2" "DisplayPort 1" "DisplayPort 2" "Dual DVI" "Dual DisplayPort" "HDMI" "HDBaseT" "SDI"	
		Write	"property.set"	"property": "image.window.main.source", "value": <value>		
	Color space	Read	"property.get"	"property": "image.connector.<connector>.colorspace"	Connector: displayport1 displayport2 dvi1 dvi2 hdmi hdbaset sdi	Value: "Auto" "RGB" "REC709" "REC601"
		Write	"property.set"	"property": "image.connector.<connector>.colorspace", "value": <value>		
	Signal range	Read	"property.get"	"property": "image.connector.<connector>.signalrange "	Connector: displayport1 displayport2 dvi1 dvi2 hdmi hdbaset sdi	Value: "Auto" "0-255" "16-235"
		Write	"property.set"	"property": "image.connector.<connector>.signalrange ", "value": <value>		

Image	Brightness	Read	"property.get"	"property": "property": "image.brightness"	Value: <-1 - 1>	
		Write	"property.set"	"property": "image.brightness", "value": <value>		
	Contrast	Read	"property.get"	"property": "image.contrast"	Value: <0 - 2>	
		Write	"property.set"	"property": "image.contrast", "value": <value>		
	Saturation	Read	"property.get"	"property": "image.saturation"	Value: <0 - 2>	
		Write	"property.set"	"property": "image.saturation", "value": <value>		
	P7 RealColor Mode	Read	"property.get"	"property": "image.color.p7.custom.mode"	Value: "Native" "CustomRGB" "CustomRGBCMY" "CustomWHITE"	
		Write	"property.set"	"property": " image.color.p7.custom.mode ", "value": <value>		
	P7 RealColor White mode	Read	"property.get"	"property": "image.color.p7.custom.whitemode"	Value: "Temperature" "Coordinates"	
		Write	"property.set"	"property": " image.color.p7.custom.whitemode", "value": <value>		
	P7 RealColor White temperature	Read	"property.get"	"property": "image.color.p7.custom.whitetemperature"	Value: <3200 – 13000>	
		Write	"property.set"	"property": " image.color.p7.custom.whitetemperature", "value": <value>		
	P7 RealColor White Coordinates (x,y,gain)	Read	"property.get"	"property": ["image.color.p7.custom.whitex", "image.color.p7.custom.whitey", "image.color.p7.custom.whitegain"]	Value: <0.0 – 1.0>	
		Write	"property.set"	"property": "image.color.p7.custom. whitex ", "value": <value>		

	P7 RealColor Custom Color Coordinates (x,y,gain)	Read	"property.get"	"property": ["image.color.p7.custom.redx", "image.color.p7.custom.redy", "image.color.p7.custom.redgain", "image.color.p7.custom.bluex", "image.color.p7.custom.bluey", "image.color.p7.custom.bluegain", "image.color.p7.custom.greenx", "image.color.p7.custom.greeny", "image.color.p7.custom.greengain", "image.color.p7.custom.yellowx", "image.color.p7.custom.yellowy", "image.color.p7.custom.yellowgain", "image.color.p7.custom.cyanx", "image.color.p7.custom.cyany", "image.color.p7.custom.cyangain", "image.color.p7.custom.magentax", "image.color.p7.custom.magentay", "image.color.p7.custom.magentagain"]	Value: <0.0 – 1.0>	
		Write	"property.set"	"property": "image.color.p7.custom.redx", "value": <value>		
	P7 RealColor Native Coordinates (x,y,luminace)	Read	"property.get"	"property": "image.color.p7.native.rgbw"	Values: red green blue white x,y and lumens	
	P7 RealColor Reset to Native	Reset	"image.color.p7.custom.resettonative"			
	Output Resolution	Read	"property.get"	"property":"image.resolution.resolution"	Values: "WQXGA", "4K-UHD"	
		Write	"property.set"	"property":"image.resolution.resolution", "value":"<value>"		

Installation	Lens shift	Up	"optics.lensshift.vertical.stepreverse"	"steps": <value>	Value: Steps	
		Down	"optics.lensshift.vertical.stepforward"	"steps": <value>		
		Left	"optics.lensshift.horizontal.stepreverse"	"steps": < value >		
		Right	"optics.lensshift.horizontal.stepforward"	"steps": < value >		
	Shutter	Read	"property.get"	"property": "optics.shutter.target"	Value: "Open" "Closed"	
		Write	"property.set"	"property": "optics.shutter.target", "value": <value>		
	Zoom	In	"optics.zoom.stepforward"	"steps": < value >	Value: Steps	
		Out	"optics.zoom. stepreverse"	"steps": < value >		
	Focus	In	"optics.focus. stepforward"	"steps": < value >	Value: Steps	
		Out	"optics.focus.stepreverse"	"steps": < value >		
	Orientation	Read	"property.get"	"property": "image.orientation"	Value: "DESKTOP_FRONT" "DESKTOP_REAR" "CEILING_FRONT" "CEILING_REAR"	
		Write	"property.set"	"property": "image.orientation", "value": <value>		
	Warp screen size height	Read	"property.get"	"property": "image.processing.warp.fourcorners.screenheight"	Value: Screen height	
		Write	"property.set"	"property": "image.processing.warp.fourcorners.screenheight", "value": <value>		
	Warp screen size witdth	Read	"property.get"	"property": "image.processing.warp.fourcorners.screenwidth"	Value: Screen width	
		Write	"property.set"	"property": "image.processing.warp.fourcorners.screenwidth", "value": <value>		
	Warp enable	Read	"property.get"	"property": "image.processing.warp.fourcorners.enable"	Value: true false	

	Write	"property.set"	"property": "image.processing.warp.fourcorners.enable", "value": <value>		
Warp corner bottom left	Read	"property.get"	"property": "image.processing.warp.fourcorners.bottomleft"	Value: x position y position	

	Write	"property.set"	"property":
			"image.processing.warp.fourcorners.bottomleft",
			"value": {
			"x": <x position>,
			"y": <y position>
			}		

Warp corner bottom right	Read	"property.get"	"property": "image.processing.warp.fourcorners.bottomright"	Value: x position y position	

	Write	"property.set"	"property":
			"image.processing.warp.fourcorners.bottomright",
			"value": {
			"x": <x position>,
			"y": <y position>
			}		

Warp corner top left	Read	"property.get"	"property": "image.processing.warp.fourcorners.topleft"	Value: x position y position	

	Write	"property.set"	"property":
			"image.processing.warp.fourcorners.topleft",
			"value": {
			"x": <x position>,
			"y": <y position>
			}		

Warp corner Top right	Read	"property.get"	"property": "image.processing.warp.fourcorners.topright"	Value: x position y position	

		Write	"property.set"	"property":
				"image.processing.warp.fourcorners.topright",
				"value": {
				"x": <x position>,
				"y": <y position>
				}		

	Illumination	Read	"property.get"	"property": "illumination.sources.laser.power"	Value: Laser power <0-100> %	
		Write	"property.set"	"property": "illumination.sources.laser.power", "value": <value>		

System settings	Communication		"property.get"	"property": ["network.device.lan.configuration", "network.device.lan.devicetype", "network.device.lan.hwaddress", "network.device.lan.ip4config", "network.device.lan.ip4configmanual", "network.device.lan.state"]		
	Show selected test pattern	Read	"property.get"	"property": "image.testpattern.show"	Value true false	
		Write	"property.set"	"property": "image.testpattern.show", "value": <value>		
	Select P7 patterns	Read	"property.get"	"property": "image.testpattern.selected"	Value	
					"UNCORRECTED_RED"	
					"UNCORRECTED_GREEN"	
					"UNCORRECTED_BLUE"	
					"UNCORRECTED_WHITE"	
					"BLACK"	
		Write	"property.set"	"property": "image.testpattern.selected", "value": <value>		
	Select test pattern	Read	"property.get"	"property": "image.testpattern.selected"	Value	
					"internal:Focus Black"	
					"internal:Ansi Lumen"	
					"internal:Scenergix"	
					"internal:Checker Board"	
					"internal:Focus"	
					"internal:Cross Hatch"	
					"internal:Monoscope"	
					"internal:Outline"	
					"internal:Color Bars"	
		Write	"property.set"	"property": "image.testpattern.selected", "value": <value>		
	Standby ECO enable	Read	"property.get"	"property": "system.eco.enable"	Value:	

		Write	"property.set"	"property": "system.eco.enable", "value": <value>	true false	
Status	Product		"property.get"	"property": ["system.articlenumber", "system.serialnumber", "system.colorwheel", "system.firmwareversion", "system.modelname"]		
	Network info Configuration Devicetype HW address (MAC) IP4 Config IP4 Config manual Connection state		"property.get"	"property": ["network.device.lan.configuration", "network.device.lan.devicetype", "network.device.lan.hwaddress", "network.device.lan.ip4config", "network.device.lan.ip4configmanual", "network.device.lan.state"]		
	Environment temp		"property.get"	"property": "environment.temperature.inlet.observableoutput"		