***5LIA0: EVC Interfaces***

Raspberry JSON RPC

# Introduction

This template, modified in MS Word 2007 and saved as a “Word 97-2003 Document” for the PC, provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout a conference proceedings.

# Methods

## RPC methods

**“JSONRPC.Version”** will return the version of the JSON RPC. It does not need any parameters. The answer contains the version in the result field

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:1,   **"method"**:"JSONRPC.Version" } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:1,   **"result"**:"0.1" } |

**“JSONRPC.GetMethods”** returns the methods that are supported by the current version of the RPC. It does not require any parameters. The server’s response will contain all supported methods in an array.

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:2,   **"method"**:"JSONRPC.GetMethods" } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:2,   **"result"**:[       "JSONRPC.GetVersion",     "JSONRPC.GetMethods",     "System.SetMode",     "..." ] } |

## System Control Methods

“**System.SetMode**” sets the operating mode of the system. The server will respond that the setting was successful.

Params: 1. System.SetMode ( “autonomous”, “manual” )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:3,   **"method"**:"System.SetMode",   **"params"**{      **"mode"**:"autonomous"} } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:3,   **"result"**:"OK" } |

“**System.SetVerbose**” sets the verbosity level. The system sends out notifications about various parts of the system. With this function it is possible to subscribe to each of these parts.

Params: 1. System.VerbosityList ( “mode”,”motorSpeed”,”turretAngle”,”turretmissile”,”laser” )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:4,   **"method"**:"System.SetVerbose",   **"params"**:[     "mode",     "motorSpeed",     "turretAngle" ] } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:4,   **"result"**:"OK" } |

“**System.GetMode**” gets the operating mode of the system. The method takes no parameters and the server will respond with a System.OperatingMode (“autonomous”, “manual”).

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:5,   **"method"**:"System.GetMode"} |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:5,   **"result"**:"autonomous" } |

“**System.SendUART**” sends a string of data over the bus. The server will respond that the setting was successful.

Params: 1. System.uartDataType ( “hex”, “string” )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:6,   **"method"**:"System.SendUART",   **"params"**:{      **"dataType"**:"hex"     **"data"**:"5AFFA5" } } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:6,   **"result"**:"OK" } |

## Moter Control Methods

“**Moter.SetMotor**” sets the speed of one or two motors. The motors can be set by setting the speeds of the motors individually. If an optional parameter is not supplied, the system will not change the speed of that motor. Note that this method only works when in automatic mode.

Params: 1. Optional.Motor.Speed “left” ( [-100,100] )

2. Optional.Motor.Speed “right” ( [-100,100] )

Example 1:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:7,   **"method"**:"Motor.SetMotor",   **"params"**:{       **"left"**:"-6",     **"right"**:50 } } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:7,   **"result"**:"OK" } |

Example 2:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:8,   **"method"**:"Motor.SetMotor",   **"params"**:{       **"type"**:"speed",     **"left"**:"50" } } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:8,   **"result"**:"OK" } |

“**Moter.GetMotorSpeed**” returns the angle and the speed of the motors. The method does not take parameters.

Returns: 1. Motor.Speed “left” ( [-255,255] )

2. Motor.Speed “right” ( [-255,255] )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:9,   **"method"**:"Motor.GetMotorSpeed" } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:9,   **"result"**:[50,50] } |

## Turret Control Methods

“**Turret.SetAngle**” sets the horizontal and vertical angle of the turret. The angle can bet set for one or both the horizontal and vertical angle. If an optional parameter is not supplied, the system will not change the angle of that motor.

Params: 1. Optional.Turret.Angle “horizontal” ( [-90,90] )

3. Optional.Turret.VAngle “vertical” ( [0,90] )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:10,   **"method"**:"Turret.SetAngle",   **"params"**:{      **"hoirzontal"**:"-6",     **"vertical"**:50 } } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:10,   **"result"**:"OK" } |

“**Turret.GetAngle**” returns both the horizontal and vertical angle of the turret. The method does not take parameters.

Returns: 1. Turret.Angle “horizontal” ( [-90,90] )

3. Turret.VAngle “vertical” ( [0,90] )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:11,   **"method"**:"Turret.GetAngle" } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:11,   **"result"**: [0,45] } |

“**Turret.FireMissile**” tries to fire a single or all missiles from one or both turrets, dependent on the parameters.

Returns: 1. Turret.TurretID “turret” ( 1,2,12 )

3. Turret.amount “amount” ( “one”, “all” )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:12,   **"method"**:"Turret.FireMissile"  **"params"**:{      **"turret"**:1,     **"amount"**:"one" } } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:12,   **"result"**: "OK" } |

“**Turret.SetLaser**” turns the laser on or off. The angle can bet set for one or both the horizontal and vertical angle. If an optional parameter is not supplied, the system will not change the angle of that motor.

Params: 1. bool ( true, false )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:13,   **"method"**:"Turret.SetLaser",   **"params"**:{       **"on"**:true } } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:13,   **"result"**:"OK" } |

“**Turret.GetLaser**” returns if the laser is on or off. The method does not take any parameters.

Returns: 1. bool ( true, false )

Example:

|  |  |
| --- | --- |
| 🡪 | { **"jsonrpc"**:"2.0",   **"id"**:14,   **"method"**:"Turret.GetLaser" } |
| 🡨 | { **"jsonrpc"**:"2.0",   **"id"**:14,   **"result"**:true } |