



# Angel - JSON Interface

## Interface Description Document

Project name : *Angel A1/A1p LED Fixture*  
Document type : *Interface Description Document*  
Document name : *IDD JSON Interface.docx*  
Version number : *1.3*

<b>1. Introduction</b>	<b>3</b>
1.1. Purpose of the document	3
1.2. Audience	3
<b>2. Protocol flow</b>	<b>4</b>
2.1. Discovery	4
2.2. Login	5
<b>3. Variables and constants</b>	<b>6</b>
3.1. Enumerations	6
3.1.1.EN_ANGEL_CC	6
3.1.2.EN_ANGEL_DEVICE	7
<b>4. API</b>	<b>8</b>
4.1. System	10
SystemGetName	10
SystemSetName	11
SystemSetDefaultName	12
SystemGetDevices	13
SystemGetTime	14
SystemSetTime	15
SystemLogin	16
SystemSetNewPassword	17
4.3. Device	18
DeviceGetInfo	18
DeviceFactoryReset	19
4.4. Program	20
ProgramGetList	20
ProgramGet	21
ProgramGetChannel	22
ProgramSet	23
ProgramSetChannel	24
ProgramSetDefault	25
ProgramGetCurrent	26
ProgramSetCurrent	27
4.5. Event	28
EventGetList	28
EventGet	29
EventSet	30
EventSetDefault	31
EventGetCurrent	32
EventSetCurrent	33
EventStopCurrent	34
4.6. Channel	35
GetChannelList	35
4.7. Demo	36
DemoModeStart	36
DemoModeStop	37
DemoProgramSetSpeed	38
DemoChannelIntensitiesSet	39
DemoChannelIntensitiesReset	40
4.8. Update	41
FirmwareUpdate	41
4.8.2.FirmwareUpdateStatus	42
FirmwareUpdateApply	43

# 1. Introduction

## 1.1. Purpose of the document

This document covers the interface description of the JSON Interface. The JSON Interface module will be designed using the protocol specified in this document.

## 1.2. Audience

The audience this document is intended for:

Audience	Purpose
System architect	For review
SW developers	Input for implementation

Table 5: Audience

## 2. Protocol flow

### 2.1. Discovery

Each of the Angle A1/A1p Master light fixtures will send out a periodic Alive message. This message is a standard UDP broadcast message supported by the WiFi module, it contains information such as the device IP address as well as the Network name. The Application can listen to the UDP port to discover the devices present in the network. It can now display and connect to them based on IP or network name. The network name of the device will reside in the acDeviceID variable.

```
#define UDP_ALIVE_TIMEOUT ( 2 * SECOND )
#define UDP_ALIVE_PORT ( 64888 )

#define ALIVE_SIZE_MAC ( 6 )
#define ALIVE_SIZE_TIME (14)
#define ALIVE_SIZE_VERSION (28)
#define ALIVE_SIZE_DEV_ID (31)
#define ALIVE_SIZE_SENSORS (17)

typedef struct UTIL_PACKED _ST_ALIVE
{
    u_int8 aubyMAC[ALIVE_SIZE_MAC];
    u_int8 ubyChannel;
    u_int8 ubyRSSI;
    u_int16 usPort;
    u_int32 ulRTC;
    u_int16 usBatt;
    u_int16 usGPIO;
    char acTime[ALIVE_SIZE_TIME];
    char acVersion[ALIVE_SIZE_VERSION];
    char acDeviceID[ALIVE_SIZE_DEV_ID];
    u_int16 usBootTime;
    u_int8 aubySensors[ALIVE_SIZE_SENSORS];
}ST_ALIVE;
```

## 2.2. Login

The login procedure will be done via the JSON interface "SystemLogin" is passed the password in the clear. This call will when it succeeds provide an session ID, which needs to be added to the HTTP header each subsequent call to the JSON web interface.

Login command:

```
POST /command.html HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Content-Length: 51
SessionID: 0
Origin: http://127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 ... Chrome/
33.0.1750.146 Safari/537.36
Content-type: text/x-www-form-urlencoded
Accept: */*
Referer: http://127.0.0.1/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: nl,en-US;q=0.8,en;q=0.6,de;q=0.4

{ "method": "SystemLogin", "params": ["Admin123"] }
```

Subsequent call:

```
POST /command.html HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Content-Length: 40
SessionID: 41
Origin: http://127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 ...
Content-type: text/x-www-form-urlencoded
Accept: */*
Referer: http://127.0.0.1/program.html
Accept-Encoding: gzip, deflate, sdch
Accept-Language: nl,en-US;q=0.8,en;q=0.6,de;q=0.4

{ "method": "ProgramGet", "params": [] }
```

Insert HTTP header in JavaScript:

```
function SendComCallb(jsonstring, callback)
{
  var xp=new XMLHttpRequest();
  xp.onreadystatechange=function()
  {
    if (xp.readyState==4 && xp.status==200)
    {
      callback( JSON.parse( xp.responseText ) );
    }
  };
  xp.open("POST","command.html",true);
  xp.setRequestHeader("Content-type","text/x-www-form-urlencoded");
  xp.setRequestHeader("SessionID", SesId);
  xp.send(jsonstring);
}
```

### 3. Variables and constants

#### 3.1. Enumerations

##### 3.1.1.EN\_ANGEL\_CC

Completion code for function calls.

**Elements:**

Element name	#	Description
eANGEL_CC_OK	0	Everything went OK
eANGEL_CC_DONE	1	Done with a task
eANGEL_CC_FAILURE	2	Error detected
eANGEL_CC_INVALID_PARAMETER	3	Parameter error detected
eANGEL_CC_ALREADY_REGISTERED	4	Item is already registerd
eANGEL_CC_NOT_FOUND	5	Item not found
eANGEL_CC_BUFFER_FULL	6	Buffer is full
eANGEL_CC_INVALID_VERSION	7	Version check error
eANGEL_CC_CRC_ERROR	8	CRC check failed
eANGEL_CC_INVALID_TARGET	9	Invalid target
eANGEL_CC_OUT_OF_RESOURCES	10	Busy we cannot handle any requests anymore
eANGEL_CC_EMPTY	11	List is empty
eANGEL_CC_INVALID_CONTEXT	12	Wrong context for function call
eANGEL_CC_INVALID_PASSWORD	13	Password check failed
eANGEL_CC_NOT_LOGGED_IN	14	Not logged in
eANGEL_CC_WRITE_PROTECTED	15	Write protected
eANGEL_CC_NOT_CHANGED	16	Not changed
eANGEL_CC_OUT_OF_BOUNDS	17	Memory out of bounds

### 3.1.2.EN\_ANGEL\_DEVICE

Device ID's

**Elements:**

Element name	#	Description
eANGEL_DEVICE_A1	0	Device id for angel a1
eANGEL_DEVICE_A1PRO	1	Device id for angel a1 PRO
eANGEL_DEVICE_A0	2	Device id for angel a0
eANGEL_DEVICE_A0PRO	3	Device id for angel a0 PRO

## 4. API

The command and control can be done by sending JSON encoded commands to the webserver, which will send JSON encoded responses. (JSON-RPC look-a-like)

Function name	Description
<i>System</i>	
SystemGetName	Returns the name of the angel network
SystemSetName	Sets the name of the angel network
SystemSetDefaultName	Sets the network default name
SystemGetDevices	Returns a list off all the devices in the network
SystemGetTime	Returns the angel network time
SystemSetTime	Sets the angel network time
SystemLogin	Login to network
SystemSetNewPassword	Sets new login password
<i>Device</i>	
DeviceGetInfo	Returns device info: Type, device ID, Temperature,Uptime, Fan level, version info, RF info.
DeviceFactoryReset	Perform a factory reset to all devices or a specific device
DeviceRemove	Remove a device from the network
<i>Program</i>	
ProgramGetList	Returns list of all pre-programmed and user programs
ProgramGet	Get program name and affect
ProgramGetChannel	Get given channel info for given program
ProgramSet	Save new program name parameters
ProgramSetChannel	Save new channel parameters for a given program
ProgramSetDefault	Sets a user program parameters to default
ProgramGetCurrent	Returns the ID of the current program
ProgramSetCurrent	Set new program
<i>Event</i>	
EventGetList	Returns a list of all pre-programmed and user events
EventGet	Gets detailed event info of a specific event
EventSet	Saves event parameters
EventSetDefault	Set all event parameters to default
EventGetCurrent	Returns current event info, or error when no event is selected
EventSetCurrent	Sets a new event.
EventStopCurrent	Stops the event



Function name	Description
<i>Channel</i>	
<i>ChannelGetList</i>	Get the list of supported channels
<i>Demo</i>	
<i>DemoModeStart</i>	Go into demo mode to set program speed or to drive the LED channels manually
<i>DemoModeStop</i>	Go out of demo mode and restore previously running program
<i>DemoProgramSetSpeed</i>	Set the program speed to demo a program
<i>DemoChannellntensitiesSet</i>	Set the intensities of the Channels manually
<i>DemoChannellntensitiesReset</i>	Reset the manually set channels
<i>Update</i>	
<i>FirmwareUpdate</i>	Upload part of an update file
<i>FirmwareUpdateStatus</i>	Get status of firmware update broadcast to slaves
<i>FirmwareUpdateApply</i>	Apply the firmware update (will reboot the fixtures)

## 4.1. System

System specific API commands.

### SystemGetName

#### Description

Gets the name of the system.

**Command:** { "method": "SystemGetName" }

#### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0,"AquaNet"] }

#### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	string	Network name	AquaNet

## SystemSetName

### Description

Sets the name of the system

**Command:** { "method": "SystemSetName", "params": ["AquaNet"] }

### Params:

#	Type	Description	Example
0	string	Network name to set	AquaNet

**Response:** { "result": [0] }

### Results

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## SystemSetDefaultName

### Description

Sets the name of the system to default

**Command:** { "method": "SystemSetDefaultName" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0] }

### Results

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## SystemGetDevices

### Description

Gets all devices in the systems network.

**Command:** { "method": "SystemGetDevices" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [ 0 , 2, [ 0, 005A421A, 1 ], .. ] }

### Results

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	u_int8	Nr. Off nodes in the network. 2..n	2
2..n	Node info array	Network node information	-

### Node info array

#	Type	Description	Example
0	u_int8	Node number	0
1	u_int32	Device ID	005A421A
2	bool	Device still active in network?	True

## SystemGetTime

### Description

Get time used by the system

**Command:** { "method": "SystemGetTime" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0,20,22,24] }

### Results

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	u_int8	Hours	20
2	u_int8	Minutes	22
3	u_int8	Seconds	24

## SystemSetTime

### Description

Set time used by the system

**Command:** { "method": "SystemSetTime", "params": [20,22,24] }

### Params:

#	Type	Description	Example
0	u_int8	Hours	20
1	u_int8	Minutes	22
2	u_int8	Seconds	24

**Response:** { "result": [0] }

### Results

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## SystemLogin

### Description

Perform a system login

**Command:** { "method": "SystemLogin", "params": ["Admin123"] }

### Params:

#	Type	Description	Example
0	string	Password	Admin123

**Response:** { "result": [0, 1234] }

### Results

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	Session ID	Session ID, needs to be inserted in the http header	1234



## SystemSetNewPassword

### Description

Set new network password

**Command:** { "method": "SystemSetNewPassword", "params": ["Admin123"] }

### Params:

#	Type	Description	Example
0	String	Password	Admin123

**Response:** { "result": [0] }

### Results

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

### 4.3. Device

Device specific API commands.

#### DeviceGetInfo

##### Description

Get the device info from a node in the RF network

**Command:** { "method": "DeviceGetInfo", "params": [1] }

##### Params:

#	Type	Description	Example
0	U_int8	Node number of the device	1

**Response:** { "result": [0,0, 005A421A, 25, 2080210, 200, "V0004 L002 M001", 80 ] }

##### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	EN_ANGEL_DEVICE	Device type id	eANGEL_DEVICE_A1
2	u_int32	Device ID	005A421A
3	int16	Temperature in degrees Celsius	25
4	u_int32	Device uptime in seconds	2080210
5	u_int16	Fan level in pro milles	200
6	string	Version info	V0004 L002 M001
7	u_int8	RF signal quality in percentage	80

## DeviceFactoryReset

### Description

Perform a factory reset.

**Command:** { "method": "DeviceFactoryReset", "params": [1] }

### Params:

#	Type	Description	Example
0	u_int8	Node number of the device, if 0 all devices in the network will be resetted	1

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## DeviceRemove

### Description

Perform a factory reset.

**Command:** { "method": "DeviceRemove", "params": [2] }

### Params:

#	Type	Description	Example
0	u_int8	Node number of the device that needs to be removed	2

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## 4.4. Program

Program specific API commands.

### ProgramGetList

#### Description

Gets a list of all the pre-programmed programs and user defined programs. Pre-programmed program numbers are numbered from 1-99, user programs 101 – 199.

**Command:** { "method": "ProgramGetList" }

#### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0, 11, [0,"Hawaiian Luau"], ..] }

#### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	u_int8	Nr of programs in list	11
2..n	Program info	Program info	

#### Program info:

#	Type	Description	Example
0	u_int8	Program number	0
1	string	Program name	Hawaiian Luau

## ProgramGet

### Description

Get detailed information of a program.

**Command:** { "method": "ProgramGet", "params": [0] }

### Params:

#	Type	Description	Example
0	u_int8	Program number	1

**Response:** { "result": [0, "Hawaiian Luau", [1,2,60,40]] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	string	Name of program	Hawaiian Luau
2...n		Affect info	

### Affect info:

#	Type	Description	Example
0	Bool	On / Off	1
1	u_int8	Probability	2
3	u_int8	Cloud cover	60
4	u_int8	Lightning	40

## ProgramGetChannel

### Description

Get detailed information of a program.

**Command:** { "method": "ProgramGetChannel", "params": [1,1] }

### Params:

#	Type	Description	Example
0	U_int8	Program number	1
1	U_int8	Channel number	1

**Response:** { "result": [0, 12, [ 842, 2 ],...] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
2	u_int16	Nr of used transitions	12
3..n	Transition info	Channel info	

### Transition info:

#	Type	Description	Example
0	u_int16	Minute of day	842
1	u_int16	Intensity	2

## ProgramSet

### Description

Saves new program parameters. For Pre-defined programs only the Affect settings can be changed.

**Command:** { "method": "ProgramSet", "params": [101, "Test program" , [1,2,60,40]] }

### Params:

#	Type	Description	Example
0	u_int8	Program number	101
1	string	Name of program	Test program
2	Affect array	Affect info	[1,2,60,40]

### Affect info:

#	Type	Description	Example
0	Bool	On / Off	1
1	u_int8	Probability	2
3	u_int8	Cloud cover	60
4	u_int8	Lightning	40

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## ProgramSetChannel

### Description

Saves new program parameters. Pre-defined programs cannot be changed, so the program number must be greater than 99 otherwise the call result will be a failure.

**Command:** { "method": "ProgramSetChannel", "params": [101,0,12,[[ 912, 60 ],...]] }

### Params:

#	Type	Description	Example
0	u_int8	Program number	101
1	EN_ANGEL_CHANNEL	Channel ID	0
2	u_int16	Nr of used transitions	12
3..n	Transition info	Transition info	

### Transition info:

#	Type	Description	Example
0	u_int16	Minute of day	912
1	u_int16	Intensity	60

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK



## ProgramSetDefault

### Description

Sets a new program, using the program number

**Command:** { "method": "ProgramSetDefault", "params": [105]}

### Params:

#	Type	Description	Example
1	u_int8	Program number	105

**Response:** { "result": [0]}

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## ProgramGetCurrent

### Description

Returns the program number of the current program

**Command:** { "method": "ProgramGetCurrent" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0, 1] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	u_int8	Program number	1

## ProgramSetCurrent

### Description

Sets a new program, using the program number

**Command:** { "method": "ProgramSetCurrent", "params": [2]}

### Params:

#	Type	Description	Example
1	U_int8	Program number	2

**Response:** { "result": [0]}

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## 4.5. Event

Event specific API commands.

### EventGetList

#### Description

Gets a list of all the pre-programmed event and user defined events. Pre-defined events have an event number between 1-99. User events have a number between 101-199.

**Command:** { "method": "EventGetList" }

#### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0, 15, [0,"Dinner party"], ..] }

#### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	u_int8	Nr of events in list	15
2..n	Event info	Event info	

#### Event info:

#	Type	Description	Example
0	u_int8	Event number	0
1	string	Event name	Dinner party

## EventGet

### Description

Get detailed information of an event

**Command:** { "method": "EventGet", "params": [5] }

### Params:

#	Type	Description	Example
0	u_int8	Event number	5

**Response:** { "result": [0, "Dinner party", 7, [90, ..] ] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	string	Name of event	Dinner party
2	u_int8	Nr. Of channels	7
3			

### Channel intensity:

#	Type	Description	Example
0	u_int16	Channel intensity	90

## EventSet

### Description

Saves event parameters. For pre-defined events only the Affect settings can be changed.

**Command:** {"method": "EventSet", "params": [105, "Test event", [1,2,60,40],7,[90,...]]}

### Params:

#	Type	Description	Example
0	u_int8	Event number	105
1	String	Name of event	Test event
2	Affect array	Affect setting	[1,2,60,40]
3	u_int8	Nr. Of channels	7
4	u_int16 array	Channel intensity	[90,...]

### Affect info:

#	Type	Description	Example
0	Bool	On / Off	1
1	u_int8	Probability	2
3	u_int8	Cloud cover	60
4	u_int8	Lightning	40

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## EventSetDefault

### Description

Set user event parameters to default

**Command:** { "method": "EventSetDefault", "params": [104] }

### Params:

#	Type	Description	Example
0	u_int8	Event number	104

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## EventGetCurrent

### Description

Returns the event ID of the current event. If no event is set, eANGEL\_CC\_EMPTY is returned.

**Command:** { "method": "EventGetCurrent" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0, 5, 282, 360 ] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	u_int8	Event number	5
2	u_int32	Event run time left in seconds	282
3	u_int32	Initial time set of event in seconds	360



## EventSetCurrent

### Description

Sets a new event.

**Command:** { "method": "EventSetCurrent", "params": [5,180] }

### Params:

#	Type	Description	Example
0	u_int8	Event number	5
1	u_int32	Length of event in seconds	180

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## EventStopCurrent

### Description

Stops the running event

**Command:** { "method": "EventStopCurrent" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## 4.6. Channel

### GetChannelList

#### Description

Gets a list of all the channels.

**Command:** { "method": "ChannelGetList" }

#### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0, 6, [0,"Intensity"], ..] }

#### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK
1	u_int8	Nr of channels in list	6
2..n	Channel info	Channel info	

#### Channel info:

#	Type	Description	Example
0	u_int8	Channel number	0
1	string	Channel name	Intensity

## 4.7. Demo

### DemoModeStart

#### Description

Start demo mode to test run a program or direct control over the LED levels

**Command:** { "method": "DemoModeStart" }

#### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0] }

#### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## DemoModeStop

### Description

Stops the demo mode and go back to running program or event.

**Command:** { "method": " DemoModeStop" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## DemoProgramSetSpeed

### Description

Set the speed of the current program, this will disable the storm settings for the program temporarily.

**Command:** { "method": "DemoProgramSetSpeed", "params": [5] }

### Params:

#	Type	Description	Example
0	u_int16	Program speed ( 1 – 800 )	5

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## DemoChannelIntensitiesSet

### Description

Set the intensities of the LED channels to given values to preview settings. These settings will persist until the demo mode is left, a subsequent call to *DemoChannelIntensitiesSet* with different settings or a *DemoChannelIntensitiesReset*.

**Command:** { "method": "DemoChannelIntensitiesSet", "params": [[0,90],...]}

### Params:

#	Type	Description	Example
0	u_int16 array	Channel intensity	[[0,90],...]

### Channel intensity:

#	Type	Description	Example
0	u_int8	Channel number	0
1	string	Channel name	90

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

## DemoChannelIntensitiesReset

### Description

Reset the previously set channel intensities and go back to the running program or event.

**Command:** { "method": "DemoChannelIntensitiesReset" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK



## 4.8. Update

### FirmwareUpdate

#### Description

Send a set of lines from a download file to the Master for processing, the download file will be iHEX format which contains ASCII lines of data. When this function returns done this master will start the firmware broadcast, status of this broadcast can be retrieved with the *FirmwareUpdateStatus* function.

**Command:** { "method": "FirmwareUdate", "params": [0, 10, 120, [":09384939...",...]] }

#### Params:

#	Type	Description	Example
0	u_int16	Line index in file	0
1	u_int16	Number of lines in this message	10
2	u_int16	Total number of lines for this download file	120
3	String array	Array containing the lines from the downloaf file	[":09384939...",...]]

**Response:** { "result": [0] }

#### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

#### Success:

eANGEL\_CC\_OK: Received data correct, download not finished yet

eANGEL\_CC\_DONE: Received data correct, download ready

#### failures:

eANGEL\_CC\_INVALID\_VERSION: Version is same or older or does not match with data or firmware version.

eANGEL\_CC\_CRC\_ERROR: The download CRC check failed

eANGEL\_CC\_INVALID\_TARGET: The download is not for this target

eANGEL\_CC\_OUT\_OF\_BOUNDS: Address does not match the internal flash map

## 4.8.2.FirmwareUpdateStatus

### Description

This function returns the status of firmware broadcast to the slaves.

**Command:** {"method": "FirmwareUpdateStatus"}

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

### Success:

eANGEL\_CC\_OK: Received data correct, download not finished yet

eANGEL\_CC\_DONE: Received data correct, download ready

### failures:

eANGEL\_CC\_FAILURE: Master was unable to send to all slaves

eANGEL\_CC\_INVALID\_CONTEXT: No download running

## FirmwareUpdateApply

### Description

Apply an uploaded firmware

**Command:** { "method": " FirmwareUpdateApply" }

### Params:

#	Type	Description	Example
-	-	-	-

**Response:** { "result": [0] }

### Results:

#	Type	Description	Example
0	EN_ANGEL_CC	Completion code, Result of function call	eANGEL_CC_OK

### Success:

eANGEL\_CC\_OK: Update will be applied

### failures:

eANGEL\_CC\_FAILURE: Upload or broadcast not ready yet

eANGEL\_CC\_NOT\_FOUND: No update pending