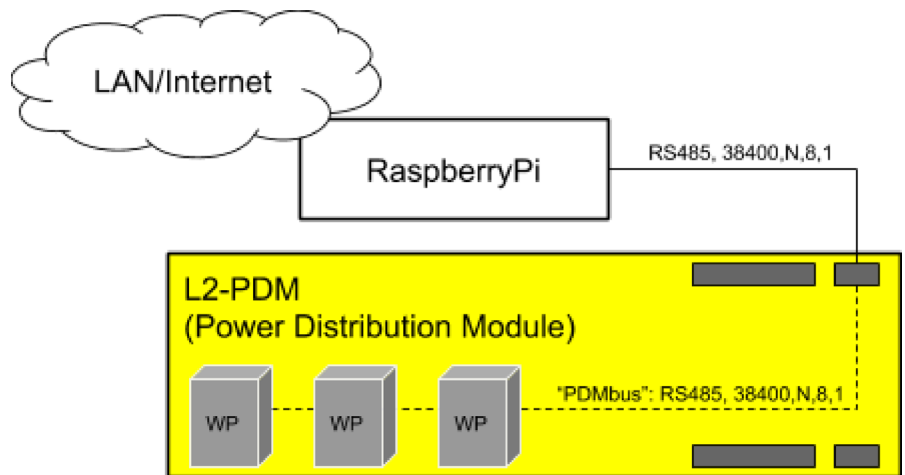


## Introduction

LumenCache LibRE uses RS485 serial communication to control and monitor lighting Zone Control Modules (L2-DM “Dimming Module”). Lighting zones can be commanded from 0% (OFF) to 100% (ON full) using the **PDMbus** serial protocol. Up to 239 devices can communicate on the PDMbus and are addressed by their ID# (1..239).



## ID#'s

ID#s can be 1..239 and should be assigned by the gateway without sequential gaps starting with ID#3. ID#1 is reserved for new or unassigned devices. ID#2 is reserved for the Gateway, leaving ID#s 3..249 for other devices. There are situations where the same ID# may be assigned to more than one device on the PDMbus. For instance, when two L2-DM must respond to the same ID#, one is set to Mode 5 (Default) and another is set to Mode 7 “Silent Follower”. If more than one device have the same ID#, they will both receive commands from the gateway but will both transmit replies at the same time, stepping on each other resulting in garbage response. The silent follower will simply follow the commands being sent to the ID# they both share but only the other device will send replies. The device EEPROM stores the ID# so it should not be changed often as the EEPROM is rated for around 100,000 writes. It can be read an unlimited number of times.

## Hardware Serial Number

Each device has a unique 20 digit serial number. It can be used to locate devices and assign ID#s.

## Communication Parameters

The PDMbus operates at 38400baud, No Parity, 8 bits, No stop bit. RS485. Only one device can talk on the bus at time. Most commands will produce a response within 200ms except the Hail Unassigned Devices command which can require up to 2.5 seconds for a reply.

## Command Structure

All messages are human readable and start with a square bracket “[”, open parenthesis “(”, or open curly brace “{”. All messages end in the matching ], ), or }. Square brackets are used for commands. Parentheses are value replies or status replies from devices, and curly braces are configuration strings from the devices. All commands contain an ID# followed by a comma-separated list. The brace type and number of parameters can be used to determine the exact message type.

For example: A SwitchPuck with ID #3 can set its Output Value to 50% ON with the following command string:

Gateway sends: [3,128] set ID# 3 to 50% (128 out of 255)

The SwitchPuck will immediately reply with the new Output Value setting for SwitchPuck ID #3:

SwitchPuck replies: (3,128) ID# 3 Output Value is at 128, or 50% ON

Gateway sends: [3,256] Code 256 means “send current output value” (for ID#3)

SwitchPuck replies: (3,128) ID# 3 Output Value is at 128, or 50% ON

## Initializing a system and gathering configurations

When a Gateway is first attached to a system, it should survey the attached SwitchPucks and gather their configuration and current Output value settings. There are several commands in the protocol to support this.

All SwitchPucks ship from the factory with ID #0. Requesting the current value from ID #0 in a system with several new SwitchPucks attached will result in garbled reply as they all respond simultaneously to the request, because they all have the same ID of 0.

The Gateway should first request the current Output value from ID numbers 1 through 240 and set up a table containing all the ID'd SwitchPucks. The SwitchPuck will reply within 25ms so the gateway can complete all 240 requests in under 7 seconds.

Host sends:	[2,256]	ask ID# 2 for its current Output level
SwitchPuck replies:	(2,0)	ID# 2 replies level is at 0
Host sends:	[3,256]	ask ID# 3 for its current Output level
SwitchPuck replies:	(3,128)	ID# 3 replies level is at 128
Host sends:	[4,256]	ask ID# 4 for its current Output level
SwitchPuck replies:	(4,0)	ID# 4 replies level is at 0
Host sends:	[5,256]	ask ID# 5 for its current Output level
(SwitchPuck does not reply within 25ms so there is NO ID #5)		
Host sends:	[6,256]	ask ID# 6 for its current Output level
SwitchPuck replies:	(6,0)	ID# 6 replies level is at 0
...etc...to ID 240		

An alternate command to the Output Value request is to request the Hardware Serial Number from each ID. This method requires a little more time but will allow you to build the Unique Keys of the system from the unique Hardware Serial Numbers.

Host sends:	[2,259]	259 is the command to request the Serial Number from ID # 2
SwitchPuck replies:	{2,5042384E333269063716}	

Another alternative is to gather the hardware and software attributes from each device in one step by sending the Get Config command [ID,258]. The Scene and Keypad Button configurations are not included in this message.

Host sends:	[6,258]	258 is the command to request the full configuration from ID # 6
SwitchPuck replies:	{6,2,3,200304.00,505136394231690C193E,5,1,2,30,220,255,0,1,0,0}	

<i>ID</i>	6
<i>Hardware Type</i>	2
<i>Hardware Version</i>	3
<i>Firmware Version</i>	200304.00
<i>Hardware Serial Number</i>	505136394231690C193E
<i>Mode</i>	5
<i>Dimming Curve</i>	1
<i>PWM Frequency</i>	2
<i>Minimum Output PWM</i>	30
<i>Maximum Output PWM</i>	220
<i>Resume Level</i>	255
<i>Ramp Duration</i>	0
<i>Motion Sensor Enable</i>	1
<i>Mode 6 Alternate Action</i>	0
<i>Inverted Output</i>	0

### Finding new SwitchPucks with Unassigned ID numbers (ID = 0)

After the table of assigned IDs is created, the gateway should search for any unassigned IDs. There is a special HAIL function that only ONE unassigned ID SwitchPuck will reply to first, cancelling the others with the same ID from replying. The response string contains the Hardware Serial Number of the SwitchPuck that replied. This can then be used to assign an ID # above 0 to the SwitchPuck.

Host sends: [253,0] 253 is the special HAIL ID address. The 0 is the ID number to hail.

A SwitchPuck replies: {0,2,3,200304.00,505136394231690C193E,5,1,2,30,220,255,0,1,0,0}

The HAIL command can take up to 2.5 seconds to complete. If no reply after 2.5 seconds then there are no unassigned SwitchPucks with that ID. New SwitchPucks are all ID #0.

Be careful not to assign more than one SwitchPuck with the same ID, they are hard to find unless you know the Hardware Serial Number. If more than one SwitchPuck is assigned the same ID, this command sequence can help de-tangle them. You'll need to re-assign every ID that replies to a hail and re-try the ID after it is reassigned (and should be unused).

### Assigning IDs to SwitchPucks using their Hardware Serial Number

To assign an ID to a SwitchPuck, add 100,000 to the ID to be assigned, followed by the hardware serial number. Valid command range is 100001 to 100240.

Host sends: [100008,5042384E333269063716] assign ID #8 to x Serial Number

SwitchPuck replies: {8,5042384E333269063716} (this is the same reply as a 259 command)

## All SwitchPuck Reply Formats

Format	Example
Used	Description
Value	(id,value)
Values. Reply to [id,256]	id=0..240, value=0..255
<b>Config:A</b>	<b>{6,2,3,200304.00,505136394231690C193E,5,1,2,30,220,255,0,1,0,0}</b>
EEPROM configurations. Reply to [id,258] and setting change commands	id=0..240, hardware_type=1..255 hardware_version=0..255 firmware_version=<string:yyymmdd.vv> hardware_serial_number=<hex:20 digits> mode=0..11 (see spec sheet) dimming_curve=1=exponential, 2=linear, 3=steep exponential pwm_frequency=1=60Hz, 2=122Hz, 3=576Hz, 4=1kHz, 5=2kHz minimum_output_pwm=1..255 (low range of pwm when output is set to 1) maximum_output_pwm=1..255 (high range of pwm when output is set to 255. >min) resume_level=1..255 (default resume level) ramp_duration=0..255 (1/10th seconds to reach new level) motion_sensor_enable=0=off, 1=enabled, 2>manual on, auto off mode_6_alternate_actions=0=Off/Resume/Full/Off, 1=Off/Resume/Off, 2=Off/Resume/Full/Resume/Off. interted_output=0=normal, 1=inverted pwm
<b>Config:B</b>	<b>{1,5,1,2,1,255,255,0,1,0,0}</b>
Settings. Reply to [id,257]	id=0..240 mode=0..11 (see spec sheet) dimming_curve=1=exponential, 2=linear, 3=steep exponential pwm_frequency=1=60Hz, 2=122Hz, 3=576Hz, 4=1kHz, 5=2kHz minimum_output_pwm=1..255 (low range of pwm when output is set to 1) maximum_output_pwm=1..255 (high range of pwm when output is set to 255. >min) resume_level=1..255 (default resume level) ramp_duration=0..255 (1/10th seconds to reach new level) motion_sensor_enable=0=off, 1=enabled, 2>manual on, auto off mode_6_alternate_actions=0=Off/Resume/Full/Off, 1=Off/Resume/Off, 2=Off/Resume/Full/Resume/Off. interted_output=0=normal, 1=inverted pwm
<b>Config:C</b>	<b>{1,505136394231690C193E}</b>
Serial number. Reply to [id,259]	id=0..240, hardware_serial_number=<hex:20 digits>
<b>Config:D</b>	<b>{1,3,4,603,604,605,606,251,252}</b>
Keypad buttons. Reply to [id,260]	id=0..240 button1=0..240zone OR 601..664scene(1..64) OR 251="UP" OR 252="DOWN"
<b>Config:E</b>	<b>{1,2,-1,-1}</b> ID1, Scene2, not participating <b>{1,3,255,20}</b> ID1, Scene3, goto 255 level in 20/10th seconds (2 seconds)
Scene configuration. Reply to [id,260+scene#]	id=0..240 scene=1..64 level=-1=not participating in scene OR 0..255=level target duration=-1=not participating in scene OR 0..255=1/10ths of seconds

## Command Quick Reference

Queries			
Function	Send:	Example:	LibRE Reply:
Request current output value for ID	[ID,256]	[2,256]	Value String
Request firmware settings	[ID,257]	[2,257]	Config:B
Request hardware & Serial Number	[ID,258]	[2,258]	Config:A
Request Serial Number only	[ID,259]	[2,259]	Config:C
Request keypad mode button settings	[ID,260]	[2,260]	Config:D
Hail first module with ID number x	[253,1]	[253,1]	Config:A

Set levels			
Function	Send:	Example:	LibRE Reply:
Set zone output level	[ID,VALUE]	[2,255]	(2,255)
Replay a scene	[ID,600+scene]	[3,601]	(3,601)
Set scene participants to "Off"	[ID,900+scene]	[3,901]	(3,901)
Ramp to target level over duration	[ID,4dddvvv]	[3,4100255]	
Stop a long ramp	[ID,270]	[3,270]	

Set EEPROM (default values upon reboot)			
Function	Send:	Example:	LibRE Reply:
Assign ID number	[100000+ID,serial]	[100008,5042 384E3332690 63716]	Config:A
Set and store the Mode	[ID,500+mode]	[3,505]	
Set ramp curve	[ID,300+curve]	[3,301]	
Set PWM frequency	[ID,400+freq]	[3,402]	
Set PWM Minimum	[ID,1000+min]	[3,1001]	
Set PWM Maximum	[ID,4000+max]	[3,4255]	
Set restore level	[ID,2000+restore]	[3,2128]	
Set switch ramp duration	[ID,3000+rate]	[3,3020]	
<b>KeyPad Mode Commands:</b>			
Keypad: Assign ID to a button	[ID,6bddd]		b=1..8 ddd=ID 1..250, 251=up, 252=dn, 0=disable
Keypad: Assign Scene to a button	[ID,7bsss]		b=1..8 sss=1..64 scene
Keypad: Request Config	[ID,260]	[2,260]	{2,32,33,601,602,251,252,0,0}
Scene: Store level setting	[ID,1ssdddvvv]	[3,01020255]	
Scene: Clear a scene from the device	[ID,700+scene]	[3,701]	
Scene: Clear all scenes in the device	[ID,700]	[3,700]	

**Other volatile settings (resets to default upon reboot)**

<b>Function</b>	<b>Send:</b>	<b>Example:</b>	<b>LibRE Reply:</b>
Disable Motion sensor	[ID,800]	[3,800]	
Automatic ON/OFF occupancy mode	[ID,801]	[3,801]	
Manual ON via switch, Automatic OFF when unoccupied	[ID,802]	[3,802]	
Mode 6 Sequence A	[ID,810]	[3,810]	Sets mode 6: Off, Resume, Full, Off
Mode 6 Sequence B	[ID,811]	[3,811]	Sets mode 6: Off, Resume, Off.
Mode 6 Sequence C (not implemented)	[ID,812]	[3,812]	Sets mode 6: Off,Resume,Full,Resume,Off
Normal PWM output (non-inverted)	[ID,820]	[3,820]	
Invert PWM output (L1-compatible)	[ID,821]	[3,821]	