Advatek PixLite Configuration Protocol V8

Contents

Document Changelog

Document Date	Description
26 Feb 2020	Initial release.

Overview

This document describes the packets used to communicate between PixLite devices and a configuration software. The protocol described is version 8. In future firmware releases, the protocol version is subject to change without notice, unless an agreement otherwise has been made. For an up to date list of the protocol running on any specific controller types and firmware versions, contact Advatek Lighting.

When a PixLite is polled by an older configuration software (running at least protocol V5), it will send a Poll Reply that is backwards compatible with the configuration software's protocol version. No other packets are backwards compatible. This allows an out-dated configuration software to still discover the controller and potentially advise the user to update their configuration software.

Communication Mechanism

The PixLite controller status and configuration data are read and set using a small range of UDP packets. Due to the nature of UDP, considerations should be made for more reliable communication at the configuration application level.

All communication is made on UDP port 49150.

All multi byte integer fields are in big endian order (most significant byte first).

Packet Structure

As of protocol version 5, many fields now have variable lengths. Where fixed lengths are used, these are defined in each packet.

All packets in both directions begin with the following structure:

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Must match: 'A' 'd' 'v' 'a' 't' 'e' 'c' 'h' 0x00
OpCode	UINT16	Refer to Table 1. This field represents the type of packet to follow.
ProtVer	UINT8	This document describes ProtVer: 8

When processing packets, discard the packet if your software does not recognise the values of these three fields. Likewise, when sending a packet to a controller, assign the values of the three fields consistent with the protocol version you are implementing.

OpCodes

The following OpCodes are currently in use which describe the nature of the packet being sent or received:

Table 1 - OpCodes

Name	Value	Description
OpPoll	0x0001	A request for controllers to identify themselves.
OpPollReply	0x0002	Reply from a controller to a poll (includes configuration data).
OpConfig	0x0005	New configuration data being directed at a controller.
OpBootload	0x0006	Send the controller into bootloader mode.
OpNetwork	0x0007	Change network settings only.
OpTestSet	0x0008	Set the test mode.
OpTestAnnounce	0x0009	Announce the current test mode.
OpVisualIdent	0x000a	A broadcast to request the controller of the designated MAC to identify its location by flashing its status LED.

Discovery and Configuration

Poll Packet

The poll packet is broadcast to the whole network by the configuration software and is used to query for online devices. There are no extra parameters other than the initial three fields that are common to all packets. Using the OpCode “OpPoll” will be all that is required to get the devices to respond.

This packet can be broadcast to the zero network broadcast address of 255.255.255.255. Devices will reply to that broadcast address also (rather than unicasting back to the configuration software directly). This enables discovery of controllers with IP addresses outside the range of the device running the configuration software. If the PixLite controller’s reported IP address is outside the subnet of the configuration software, it can be updated via the broadcast mechanism using the Network Configuration Packet (which is described later).

The structure of this packet is as follows:

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Value: ‘A’ ‘d’ ‘v’ ‘a’ ‘t’ ‘e’ ‘c’ ‘h’ 0x00
OpCode	UINT16	Refer to Table 1. Value: OpPoll
ProtVer	UINT8	8

Poll Reply Packet

Upon receiving a poll packet, the controller will reply with a “PollReply” packet which contains all its configuration and status information. The controllers will report back to the zero network broadcast address of 255.255.255.255 (rather than unicasting back to the configuration software directly), to allow discovery across subnets.

The structure of this packet is as follows:

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Value: ‘A’ ‘d’ ‘v’ ‘a’ ‘t’ ‘e’ ‘c’ ‘h’ 0x00
OpCode	UINT16	Refer to Table 1. Value: OpPollReply

ProtVer	UINT8	8
CurrentProtVer	UINT8	The current protocol version running on this PixLite. *If the device reports back a version higher than 7, then it is running a protocol version greater than this document describes. You can still process the remainder of this packet, but you should not attempt to send other packets to the device. You could indicate to the user that your configuration software needs to be updated to communicate with this device.*
MAC[6]	UINT8	MAC Address.
LENGTH_MODEL	UINT8	Length of model array. Used in variable length array to follow.
Model[LENGTH_MODEL]	UINT8	Device's display/model name E.G. "PixLite 16 MkII".
HwRev	UINT8	Hardware Version * 10. Examples: Hardware V1.0 = 10 Hardware V2.1 = 21
MinAssistantVer[3]	UINT8	Minimum Advatek Assistant version required to understand this version of the protocol. Example: Array value [2, 3, 7] = Advatek Assistant V2.3.7 or higher required. *This field remains but is not useful for 3rd party processing. Ignore the values here and use the field "CurrentProtVer" to determine if your configuration software is out-dated.*
LENGTH_FW	UINT8	Length of the firmware array. Used in variable length array to follow.
Firmware[LENGTH_FW]	UINT8	Array of 20 characters representing the firmware version/name. Padded with 0x00.
Brand	UINT8	OEM code. *This field is used by Advatek Assistant software but does not need to be interpreted by 3rd party configuration software.*
CurrentIP[4]	UINT8	Current IP address.
CurrentSM[4]	UINT8	Current subnet mask.
DHCP	UINT8	0 = OFF 1 = ON
StaticIP[4]	UINT8	Static IP address (DHCP must be OFF for this to be used).
StaticSM[4]	UINT8	Static subnet mask (DHCP must be OFF for this to be used).
Protocol	UINT8	0 = sACN (E1.31) 1 = ArtNet
HoldLastFrame	UINT8	0 = Timeout as per sACN / Art-Net specification on loss of universe 1 = Hold last data received indefinitely
SimpleConfig	UINT8	0 = Simple patch 1 = Advanced patch *This field is used by Advatek Assistant software but does not need to be interpreted by 3rd party configuration software.*
MaxPixPerOutput	UINT16	Maximum number of pixels any output can drive.
NUM_OUTPUTS	UINT8	Number of outputs (including possible expanded mode outputs). Used in variable length arrays to follow.

OutputPixels[NUM_OUTPUTS]	UINT16	Number of pixels (for each output).
OutputUniv[NUM_OUTPUTS]	UINT16	Start universe (for each output).
OutputChan [NUM_OUTPUTS]	UINT16	Start channel (for each output).
OutputNull[NUM_OUTPUTS]	UINT8	Null pixels (for each output).
OutputZig[NUM_OUTPUTS]	UINT16	Zig zag (for each output).
OutputReverse[NUM_OUTPUTS]	UINT8	Reverse enabled (for each output). 0 = Normal 1 = Reversed
OutputColOrder[NUM_OUTPUTS]	UINT8	RGB/RGBW order (for each output). Refer to Table 2.
OutputGrouping[NUM_OUTPUTS]	UINT16	Pixel grouping (for each output).
OutputBrightness[NUM_OUTPUTS]	UINT8	Brightness limit for each output (as %). 1 = 1% brightness 100 = 100% brightness
NUM_DMX_OUT	UINT8	Number of DMX outputs. Used in variable length arrays to follow.
ProtocolsOnDmxOut	UNIT8	Protocols that can be routed to the DMX outputs, specified by the bit positions: Bit 0 -> Art-Net (‘0’ not possible, ‘1’ possible) Bit 1-> sACN(‘0’ not possible, ‘1’ possible) Bits 2…7 -> Reserved
DmxOutOn[NUM_DMX_OUT]	UINT8	DMX output enabled (for each output). 0 = OFF 1 = ON
DmxOutUniv[NUM_DMX_OUT]	UINT16	DMX output universe number (for each output).
NUM_DRIVERS	UINT8	Number of pixel drivers. Used in variable length arrays to follow.
DRIVER_NAME_LENGTH	UINT8	Length of driver description string. Used in variable length arrays to follow.
DriverType[NUM_DRIVERS]	UINT8	RGB / RGBW capability (for each driver). 0 = RGB only 1 = RGBW only 2 = Either
DriverSpeed[NUM_DRIVERS]	UINT8	Speed capability (for each driver). 0 = N/A (single fixed speed) 1 = Slow only (data only driver, fixed at the slow speed) 2 = Fast only (data only driver, fixed at the fast speed) 3 = Either slow or fast (selectable between 2 speeds) 4 = Adjustable clock speed
DriverExpandable[NUM_DRIVERS]	UINT8	Expanded mode capability (for each driver). 0 = Normal mode only 1 = Capable of expanded mode
DriverNames[NUM_DRIVERS][DRIVER_NAME_LENGTH]	UINT8	Array of characters representing the driver name/description.
CurrentDriver	UINT8	Current driver selection. Range: 0 to NUM_DRIVERS - 1.
CurrentDriverType	UINT8	0 = RGB 1 = RGBW
CurrentDriverSpeed	UINT8	Refer to Table 3.
CurrentDriverExpanded	UINT8	0 = Normal mode 1 = Expanded mode (or condensed mode)

Gamma[4]	UINT8	Red, Green, Blue and White channel gamma correction * 10. The values in these fields only apply to driver chips with > 8 bit resolution. Array order: Red, Green, Blue, White. Range: 10 to 30. Example values: 10 = Gamma 1.0, 22 = Gamma 2.2
Nickname[40]	UINT8	Array of 40 characters representing the user configurable name of this controller. Padded with 0x00.
Temperature	UINT16	Temperature * 10 in Degrees Celsius. If no temperature sensor is present, will be set to 0xFFFF. Example value: 105 = 10.5 Degrees C
MaxTargetTemp	UINT8	Temperature in Degrees Celsius at which the fan (if equipped) should be fully turned on. Before this point, the controller will slowly ramp the fan up to try and keep the environment below the MaxTargetTemp value.
NUM_BANKS	UINT8	Number of power input banks on controller. Used in variable length array to follow.
VoltageBanks[NUM_BANKS]	UINT16	Voltage of the bank input (for each bank) * 10. Example value: 302 = 30.2V
TestMode	UINT8	Refer to Table 4.
TestCols[4]	UINT8	If a test mode that uses a colour value is set, these parameters specify the color used for the test. Order: Red, Green, Blue, White Range: 0 – 255 Current test modes that expect the color to be set are: • Set Color • Single Pixel
TestOutputNum	UINT8	The output number that the test mode is being applied to. 0 = All outputs.
TestPixelNum	UINT16	The physical pixel number that is being lit in single pixel test mode (includes null pixels). 0 = All pixels.

The OutputColOrder field referenced in the PollReply and Config packets is defined as:

Table 2 – RGB / RGBW Orders

Value	RGB / RGBW Order	Value	RGB / RGBW Order
0	R-G-B / R-G-B-W	12	R-W-G-B
1	R-B-G / R-B-G-W	13	R-W-B-G
2	G-R-B / G-R-B-W	14	G-W-R-B
3	B-R-G / B-R-G-W	15	B-W-R-G
4	G-B-R / G-B-R-W	16	G-W-B-R
5	B-G-R / B-G-R-W	17	B-W-G-R
6	R-G-W-B	18	W-R-G-B
7	R-B-W-G	19	W-R-B-G
8	G-R-W-B	20	W-G-R-B
9	B-R-W-G	21	W-B-R-G
10	G-B-W-R	22	W-G-B-R
11	B-G-W-R	23	W-B-G-R

The CurrentDriverSpeed field referenced in the PollReply and Config packets is defined as:

Table 3 – Driver Speeds

Value	Data Only Driver Speed	Clocked Driver Speed
0	Slow	0.4 MHz
1	Fast	0.6 MHz
2		0.8 MHz
3		1.0 MHz
4		1.2 MHz
5		1.4 MHz
6		1.6 MHz
7		1.8 MHz
8		2.0 MHz
9		2.2 MHz
10		2.5 MHz
11		2.9 MHz

Configuration Packet

To update the configuration of the controller, the configuration software will send a packet with the OpCode field set to the “Config” value. There are variable length arrays in the packet that are the same dimension as the arrays from the PollReply packet. In every instance of a variable length field, the same dimensions received in the PollReply packet must be used.

The structure of this packet is as follows:

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Value: ‘A’ ‘d’ ‘v’ ‘a’ ‘t’ ‘e’ ‘c’ ‘h’ 0x00
OpCode	UINT16	Refer to Table 1. Value: OpConfig
ProtVer	UINT8	8
MAC[6]	UINT8	MAC Address. Must match the destined controller or the remainder of packet will not be processed.

DHCP	UINT8	0 = OFF 1 = ON
StaticIP[4]	UINT8	Static IP address (DHCP must be OFF for this to be used).
StaticSM[4]	UINT8	Static subnet mask (DHCP must be OFF for this to be used).
Protocol	UINT8	0 = sACN (E1.31) 1 = ArtNet
HoldLastFrame	UINT8	0 = Timeout as per sACN / Art-Net specification on loss of universe 1 = Hold last data received indefinitely
SimpleConfig	UINT8	0 = Simple patch 1 = Advanced patch *This field is used by Advatek Assistant software but does not need to be interpreted by 3rd party configuration software.*
OutputPixels[NUM_OUTPUTS]	UINT16	Number of pixels (for each output).
OutputUniv[NUM_OUTPUTS]	UINT16	Start universe (for each output).
OutputChan [NUM_OUTPUTS]	UINT16	Start channel (for each output).
OutputNull[NUM_OUTPUTS]	UINT8	Null pixels (for each output).
OutputZig[NUM_OUTPUTS]	UINT16	Zig zag (for each output).
OutputReverse[NUM_OUTPUTS]	UINT8	Reverse enabled (for each output). 0 = Normal 1 = Reversed
OutputColOrder[NUM_OUTPUTS]	UINT8	RGB/RGBW order (for each output). Refer to Table 2.
OutputGrouping[NUM_OUTPUTS]	UINT16	Pixel grouping (for each output).
OutputBrightness[NUM_OUTPUTS]	UINT8	Brightness limit for each output (as %). 1 = 1% brightness 100 = 100% brightness
DmxOutOn[NUM_DMX_OUT]	UINT8	DMX output enabled (for each output). 0 = OFF 1 = ON
DmxOutUniv[NUM_DMX_OUT]	UINT16	DMX output universe number (for each output).
CurrentDriver	UINT8	Current driver selection. Range: 0 to NUM_DRIVERS - 1.
CurrentDriverType	UINT8	0 = RGB 1 = RGBW
CurrentDriverSpeed	UINT8	Refer to Table 3.
CurrentDriverExpanded	UINT8	0 = Normal mode 1 = Expanded mode (or condensed mode)
Gamma[4]	UINT8	Red, Green, Blue and White channel gamma correction * 10. The values in these fields only apply to driver chips with > 8 bit resolution. Array order: Red, Green, Blue, White. Range: 10 to 30. Example values: 10 = Gamma 1.0, 22 = Gamma 2.2
Nickname[40]	UINT8	Array of 40 characters representing the user configurable name of this controller. Padded with 0x00.
MaxTargetTemp	UINT8	Temperature in Degrees Celsius at which the fan (if equipped) should be fully turned on. Before this point, the controller will slowly ramp the fan up to try and keep the environment below the MaxTargetTemp value.

Network Configuration Packet

This packet can be used to change the network settings of a controller. The controller checks the MAC address contained and drops the packet if it doesn't match its own MAC. This enables the configuration software to broadcast these packets to 255.255.255.255 so that it can reach controllers that are on a different subnet to the configuration tool configuring it. In combination with a broadcast Poll packet, the config software can use this packet discover and resolve controllers that start up with an incompatible network configuration.

The structure of this packet is as follows:

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Value: 'A' 'd' 'v' 'a' 't' 'e' 'c' 'h' 0x00
OpCode	UINT16	Refer to Table 1. Value: OpNetwork
ProtVer	UINT8	8
MAC[6]	UINT8	MAC Address. Must match the destined controller or the remainder of packet will not be processed.
DHCP	UINT8	0 = OFF 1 = ON
StaticIP[4]	UINT8	Static IP address (DHCP must be OFF for this to be used).
StaticSM[4]	UINT8	Static subnet mask (DHCP must be OFF for this to be used).

Test Modes

The controller's built-in test modes can be activated remotely (as opposed to the physical button combinations). Using the Set Test Mode packet, there is an extra test mode available which allows you to set the color of the outputs to an arbitrary value.

The possible test modes are:

Table 4 – Test Mode Values

Value	Test Mode
0	None (Live Control Data)
1	RGBW Cycle
2	Red
3	Green
4	Blue
5	White
6	Set Color
7	Color Fade
8	Single Pixel

Set Test Mode Packet

This packet is used to activate a test mode or return the controller to live operation.

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Value: 'A' 'd' 'v' 'a' 't' 'e' 'c' 'h' 0x00
OpCode	UINT16	Refer to Table 1. Value: OpTestSet
ProtVer	UINT8	8
MAC[6]	UINT8	MAC Address. Must match the destined controller or the remainder of packet will not be processed.
TestMode	UINT8	Refer to Table 4.
TestCols[4]	UINT8	If a test mode that expects a colour value is being set, these parameters specify the color used for the test. Order: Red, Green, Blue, White Range: 0 – 255 Current test modes that expect the color to be set are: • Set Color • Single Pixel
TestOutputNum	UINT8	The output number that the test mode should apply to. 0 = All outputs.
TestPixelNum	UINT16	The physical pixel number that should be lit (includes null pixels). 0 = All pixels. Only the TestMode of Single Pixel may use a non-zero TestPixelNum.

Test Announce Packet

The controller will broadcast this packet anytime the test mode changes. This can be used to update the 'current test mode' displayed in software to the user and to confirm that test mode changes made remotely have been activated.

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Value: 'A' 'd' 'v' 'a' 't' 'e' 'c' 'h' 0x00
OpCode	UINT16	Refer to Table 1. Value: OpTestAnnounce
ProtVer	UINT8	8
MAC[6]	UINT8	MAC Address.
IP[4]	UINT8	Current IP Address.
TestMode	UINT8	Refer to Table 4.
TestCols[4]	UINT8	If a test mode that uses a colour value is set, these parameters specify the color used for the test. Order: Red, Green, Blue, White Range: 0 – 255 Current test modes that expect the color to be set are: • Set Color • Single Pixel
TestOutputNum	UINT8	The output number that the test mode is being applied to. 0 = All outputs.

TestPixelNum	UINT16	The physical pixel number that is being lit (includes null pixels). 0 = All pixels.

Firmware Updates

The firmware update process has two main parts. The first is to put the controller into bootloader mode. This can be done by the user physically by holding down the "Factory IP" button while powering the controller up, however that is primarily available as a backup/emergency/troubleshooting function. The controller will also always use the default IP of 192.168.0.50 in this "Forced Bootloader" mode.

The normal process would be for the configuration software to put it into bootloader mode remotely and then send the firmware update. In this situation the controller uses the IP address it is configured for and already operating on, not the factory default.

If a controller is sent into bootloader (either with the button or remotely) and then a poll packet is sent, the controller will respond with "Bootloader" in its firmware version. This way the configuration software can see a controller in this mode and tell it needs to be sent firmware rather than trying to configure it.

Bootloader Trigger

To remotely send the device into bootloader mode you use the Bootload OpCode in combination with the target MAC Address. Upon receiving a packet matching this, the controller will restart itself immediately in bootloader mode. It will remain in bootloader mode indefinitely until new firmware is received or the device is power cycled.

Bootloader trigger packet structure:

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Value: 'A' 'd' 'v' 'a' 't' 'e' 'c' 'h' 0x00
OpCode	UINT16	Refer to Table 1. Value: OpBootload
ProtVer	UINT8	8
MAC[6]	UINT8	MAC Address. Must match the destined controller.

Bootload Update

Once the controller is in bootloader mode, the firmware file must be sent. This is done over a TCP connection to the controller. The TCP port used is the same as the UDP port (TCP Port 49150).

The process is as follows:

1. Send the string "Firmware PixLite 16" (regardless of the actual device's model name), simply to indicate to the bootloader that the message following is intended for this purpose.
2. Pause for 2 seconds.
3. Send the raw data straight from the firmware update file.

Once the data is sent you can expect a delay of a few seconds while the controller writes this permanently into its flash memory. Following this the controller will restart itself into normal operating mode, executing the new firmware. Configuration is preserved across firmware updates.

Visual Identify Packet

This packet is to be used to identify a controller's physical location by flashing a special pattern on its status LED. Only one controller may indicate its location at a time. The Visual Identify packet is to be sent to the IPv4 broadcast address. It specifies the controller's MAC address and the duration it is to flash its status LED.

On receiving a Visual Identify packet bearing its MAC address, the controller starts flashing the visual-identify pattern its status LED.

On receiving a Visual Identify packet bearing any other MAC address OR when its duration has lapsed, the controller will stop flashing the visual-identify pattern on its status LED.

The visual-identify pattern is dit-dit-dit-gap, where dit is 200ms ON, 100ms OFF, and gap is 1000ms OFF.

The structure of this packet is as follows:

Name	Size	Description
ID[9]	UINT8	Array of 9 characters (including null terminator). Value: 'A' 'd' 'v' 'a' 't' 'e' 'c' 'h' 0x00
OpCode	UINT16	Refer to Table 1. Value: OpVisualIdent
ProtVer	UINT8	8
MAC[6]	UINT8	MAC Address. Must match the destined controller or the remainder of packet will not be processed. Setting this to all zeros is a valid way stop the identify flash pattern on any controller (since the MAC will not match the controller's).
Duration	UINT8	How much time the status LED is to flash. 0 = indefinite. 1 to 255 = duration in seconds.

Change History

The following is a summary of the changes between protocol V8 and protocol V7.

Set Test Mode AND Test Announce AND Poll Reply Packets

1. The "TestParameters" field is renamed to "TestCols" for clarity, since there are other parameters in this packet now.
2. The new field "TestOutputNum" has been added to allow any test mode to be set either on an individual output or all outputs.
3. A new test mode "Single Pixel" has been added which allows turning on a single physical pixel on one or all outputs. This test mode will turn on null pixels too and ignores the grouping, reversing and zig zag parameters. It is meant as a physical check/counter of individual pixels.
4. The new field "TestPixelNum" has been added to operate the "Single Pixel" test mode. This is to be coded 0 (all pixels) for all other test modes.

Identify Location Packet

1. This new packet is added to provide means for physically locating a controller on the network by a flash sequence on its status LED.

The following is a summary of the changes between protocol V7 and protocol V6.

Poll Reply Packet

1. New parameter ProtocolsOnDmxOut is added to Poll Reply packet to dictate which Ethernet protocols can be routed to a DMX port.

The following is a summary of the changes between protocol V6 and protocol V5.

Poll Reply Packet

1. When a PixLite is polled by an older configuration software (running at least protocol V5), it will send a Poll Reply that is backwards compatible with the configuration software's protocol version. No other packets are backwards compatible. This allows an out-dated configuration software to still discover the controller and potentially advise the user to update their configuration software.
2. The Poll Reply now includes the field "CurrentProtVer" which allows the configuration software to more easily identify that it is out-dated.
3. The field "MinAssistantVer" remains but is not useful for 3rd party processing. Ignore the values here and use the field "CurrentProtVer" to determine if your configuration software is out-dated.
4. The field "LENGTH_FW" has been added and controls the now variable length firmware array.
5. The "Gamma" array field has been increased to size 4. This allows support of gamma correctable RGBW drivers by adding a white channel correction setting.
6. The "TestParameters" array field has been added so that configuration software can detect the current test mode details when in "Set Color" test mode. Previously this field was available in the "Test Announce" packet only. The "TestParameters" field is increased from 3 to 4 elements now, to allow test mode control of the white channel in RGBW drivers.

Configuration Packet

1. The "Gamma" array field has been increased to size 4. This allows support of gamma correctable RGBW drivers by adding a white channel correction setting.

Set Test Mode Packet

1. The "TestParameters" field is increased from 3 to 4 elements now, to allow test mode control of the white channel in RGBW drivers.

Test Announce Packet

1. The "TestParameters" field is increased from 3 to 4 elements now, to allow test mode control of the white channel in RGBW drivers.