
Ultamation Universal Device Protocol

Revision: 1.01
Date: 29 May 2020

Outline

This document defines a universal protocol for simple control of Ultamation's control device nodes. Although the function of these nodes will vary considerably, we have defined a standard low level protocol that will be common across all devices, with device specific functionality wrapped within the generic transport.

Packet Structure

The protocol uses a simple ASCII payload sent over either TCP or UDP to the port defined through the configuration tool (default: 13255 for both UDP and TCP).

For UDP, packets should be sent as single datagrams.

Each device uses a "Profile" which defines what the various addresses mean for that particular devices. Some addresses are common to all devices but these tend to be for configuration and are not publicly documented.

The packet is formed as follows:

Field	Purpose	Description
1	Operation	This is a single ASCII character G = Get data S = Set data F = Feedback (a response to a GET)
2	Address	This is a device specific address represented in hexadecimal as 4 characters (generally '0000')
3	Parameter	This is a device specific parameter which is address specific. Represented in hexadecimal as 4 characters (e.g. '100A')
4	Length	This is the length (in characters) of the payload, represented in hexadecimal as 4 characters
5	Payload	The address/parameter specific payload data, with each byte represented as a 2-character hexadecimal ASCII value

Each field is separated by a single COLON ':' character.

There is no terminating character.

Examples

Taking the UC-DMX-200 as an example:

Example 1

Say we wish to set the value of an RGB fixture on channel 77 (decimal) to dark red (128, 0, 0).

Referring to the UC-DMX-200 Protocol Profile we can see that individual channel data can be written to using address 0x0000 (hexadecimal), and parameters 0x0001 to 0x0200 (hexadecimal) (1-512 decimal), followed by a number of bytes of channel data.

So the packet will look like this:

S:0000:004D:0006:800000

Breaking this down, we have:

S - SET

0000 - The device address

004D - The Channel (77 decimal => 4D hex)

0006 - The payload length (6 characters)

800000 - The RGB data in hex (128 = 80, 0 = 00, 0 = 00)

Example 2

Say we wish to fill DMX channels 100 to 111 (say we have 4 RGB fixtures in channel sequence) with the repeating sequence (RGB) of 0, 64, 255 (A greeny-blue).

Referring to the UC-DMX-200 Protocol Profile we can see that this would use the special parameter 0x1000.

S:0000:1000:000E:0064006F0030FF

Breaking this down, we have:

S - SET

0000 - The device address

1000 - The fill function (see profile sheet)

000E - The payload length (14 characters) - 4 for the start address, 4 for the end address and then 6 for the colour pattern

0064 - The start channel address (100 decimal => 64 hex)

006F - The end channel address (111 decimal => 6F hex)

0030FF - The colour fill (0 => 00, 64 => 40, 255 => FF)

This will result in the channels being set, from channel 100 as:
0, 64, 255, 0, 64, 255, 0, 64, 255, 0, 64, 255