DIN PIXIE – OSC API

SPI Pixel Strip/Dot controller: Control up to 4 universe of LED pixels over two outputs.

Document Version:	9
Last Updated:	20.Oct.2023

Change Summary

Revision	Change
9	Added support for custom Protocol, and updated render.
8	Added DMX1 only and DMX2 only personalities.
7	Added support for RGBW LED strips.
6	Added option to blackout LED strips on loss of input data.

Purpose

This document specifies the interface requirements for PC-based application programs to use the ENTTEC DIN PIXIE board to drive RGB or RGBW LED strips.

Introduction

The board has one USB port, 2 DMX input ports and 2 output ports. Each output port controls the intensity of the LEDs in the connected LED strip.

The LED strips can be updated by sending DMX or USB LED strip update messages to the board, either via DMX or USB. When a USB message is received, the DMX inputs will be disabled for 3 seconds, so that USB messages will lock out DMX messages.

The board will have a personality setting, stored in non-volatile memory, to select how the DMX inputs will be mapped to the LED strip outputs, as shown below.

Personality	Output 1	Output 2
Personality 1 – Basic Combined	**DMX1, DMX2**	**DMX1, DMX2**
Personality 2 Basic Straight Through	DMX1, DMX1	DMX2, DMX2
Personality 3 DMX1 Only	Part of DMX1	Part of DMX1
Personality 4 – DMX2 Only	Part of DMX2	Part of DMX2

When the personality is basic combined, the first DMX port will drive the first universe of the pixel output, and the second DMX port will drive the second universe of the pixel output. Output 2 will repeat the same data of Output 1, to avoid blank pixels on the second LED strip output.

When the personality is basic straight through, the first DMX port will drive Output 1, and the second DMX port will drive Output 2.

When the personality is DMX1 Only, selectable numbers of pixels from the first DMX port will drive Output 1 and Output 2. Any data on the second DMX port will be ignored.

When the personality is DMX2 Only, selectable numbers of pixels from the second DMX port will drive Output 1 and Output 2. Any data on the first DMX port will be ignored.

RGB DMX Message Format

The full DMX LED strip update message format is shown below. If the DMX message has less than 512 slots, the RGB data for missing slots will be zero.

Size In Bytes	Description
3	R1, G1, B1. Eight-bit intensity of first Red, Green and Blue LEDs.
3	R2, G2, B2. Eight-bit intensity of second Red, Green and Blue LEDs.
...	
3	R170, G170, B170. Eight-bit intensity of last Red, Green and Blue LEDs.
2	Unused for RGB LED strips.

RGBW DMX Message Format

The full DMX LED strip update message format is shown below. If the DMX message has less than 512 slots, the RGBW data for missing slots will be zero.

Size In Bytes	Description
4	R1, G1, B1, W1. Eight-bit intensity of first Red, Green, Blue and White LEDs.
4	R2, G2, B2, W2. Eight-bit intensity of second Red, Green, Blue and White LEDs.
...	
4	R128, G128, B128, W128. Eight-bit intensity of last Red, Green, Blue and White LEDs.
4	R1, G1, B1, W1. Eight-bit intensity of first Red, Green, Blue and White LEDs.

Limitations

To avoid loss of data, the LED strip update message rate should not exceed 200 messages per second. This means that all LED strip outputs can conservatively be updated at least 40 times per second.

Application Message Format

The PC-based application program communicates with the board via the FTDI driver. The format of USB messages is the same as for the USB Pro. Unless otherwise specified, all multi-byte integer fields will be stored in little endian order.

Application Messages

1. Reprogram Firmware Request (Label=1, no data)

This message requests the board firmware to run the board bootstrap to enable reprogramming of the board firmware.

2. Program Flash Page Request (Label=2)

This message programs one Flash page of the board firmware. The Flash pages must be programmed in order from the first to the last Flash page, with the contents of the firmware binary file.

Size in Bytes	Description
64	One page of firmware binary file.

3. Program Flash Page Reply (Label=2)

The board sends this message to the PC on completion of the Program Flash Page request.

Size in Bytes	Description
4	Success character array, set to 'TRUE' if firmware page was programmed successfully, set to 'FALS' if firmware page programming failed.

4. Get Config Request (Label=3, no data)

This message requests the non-volatile board configuration data.

5. Get Config Reply (Label=3)

The board sends this message to the PC in response to the Get Config request.

Size in Bytes	Description
1	Firmware version LSB. Valid range is 0 to 255.
1	Firmware version MSB. Valid range is 0 to 255.
1	Board Personality (see introduction). 0=Basic combined, 1=Basic straight through, 2=DMX1 only, 3=DMX2 only.
1	LED strip pixel group size. The LED strip pixels will be driven in consecutive groups, of the specified group size. Valid range is 1 to 170 inclusive.

	This only applies when Board Personality is not Basic combined.
1	Output pixel order. This applies to slots in DMX, USB and show frames. For a RGB LED strip: 0=123, 1=132, 2=213, 3=231, 4=312, 5=321. For a RGBW LED strip: 6=1234, 7=1243, 8=1324, 9=1342, 10=1423, 11=1432, 12=2134, 13=2143, 14=2314, 15=2341, 16=2413, 17=2431, 18=3124, 19=3142, 20=3214, 21=3241, 22=3412, 23=3421, 24=4123, 25=4132, 26=4213, 27=4231, 28=4312, 29=4321.
1	Type of LED strip. This selects the protocol used to control the LED strips. 0 = WS2811, 1 = WS2812, 2 = WS2812B, 3 = APA-104, 4 = AHL003 (9PDOT) with 8-bit data.
1	Start the stored show, on loss of DMX. 0 = No, 1 = Yes.
2	DMX1 start address. For RGB LED strips, valid range is 0 to 507. For RGBW LED strips, valid range is 0 to 508.
2	DMX2 start address. For RGB LED strips, valid range is 0 to 507. For RGBW LED strips, valid range is 0 to 508.
1	Blackout all LED strips, on loss of DMX and USB LED strip update messages for 30 seconds. 0 = No, 1 = Yes.
2	DMX pixel count for output 1. This only applies when Board Personality is DMX1 only or DMX2 only.
2	DMX pixel count for output 2. This only applies when Board Personality is DMX1 only or DMX2 only.
1	Custom protocol enabled. 0 = No, 1 = Yes.
2	Custom time 0 high
2	Custom time 1 high
2	Custom period time
4	Custom reset time
2	Number of LED strip update messages that have been dropped. If this is not zero, the LED strip update message rate is too high.

6. Set Config Request (Label=4)

This message sets the non-volatile board configuration data, which is preserved when the board loses power.

Size in Bytes	Description
1	Board Personality (see introduction). 0=Basic combined,

	1=Basic straight through, 2=DMX1 only, 3=DMX2 only.
1	LED strip pixel group size. The LED strip pixels will be driven in consecutive groups, of the specified group size. Valid range is 1 to 170 inclusive. This only applies when Board Personality is Basic straight through. For example, setting this parameter to 2 would drive the LED strip pixels in pairs.
1	Output pixel order. This applies to slots in DMX, USB and show frames. For a RGB LED strip: 0=123, 1=132, 2=213, 3=231, 4=312, 5=321. For a RGBW LED strip: 6=1234, 7=1243, 8=1324, 9=1342, 10=1423, 11=1432, 12=2134, 13=2143, 14=2314, 15=2341, 16=2413, 17=2431, 18=3124, 19=3142, 20=3214, 21=3241, 22=3412, 23=3421, 24=4123, 25=4132, 26=4213, 27=4231, 28=4312, 29=4321.
1	Type of LED strip. This selects the protocol used to control the LED strips. 0 = WS2811, 1 = WS2812, 2 = WS2812B, 3 = APA-104, 4 = AHL003 (9PDOT) with 8-bit data.
1	Start the stored show, on loss of DMX. 0 = No, 1 = Yes.
2	DMX1 start address. For RGB LED strips, valid range is 0 to 507. For RGBW LED strips, valid range is 0 to 508.
2	DMX2 start address. For RGB LED strips, valid range is 0 to 507. For RGBW LED strips, valid range is 0 to 508.
1	Blackout all LED strips, on loss of USB LED strip update messages for 30 seconds. 0 = No, 1 = Yes.
2	DMX pixel count for output 1. This only applies when Board Personality is DMX1 only or DMX2 only.
2	DMX pixel count for output 2. This only applies when Board Personality is DMX1 only or DMX2 only.
1	Custom protocol enabled. 0 = No, 1 = Yes.
2	Custom time 0 high
2	Custom time 1 high
2	Custom period time
4	Custom reset time

7. Get Board Serial Number Request (Label=10, no data)

This message requests the board serial number, which should be the same as that printed on the case.

8. Get Board Serial Number Reply (Label=10)

The board sends this message to the PC in response to the Get Board Serial Number request.

Size in Bytes	Description
4	BCD serial number, with LSB stored at lowest address.

9. LED Strip Update Request (Label=6)

This message outputs data to 1 of the 4 LED strip outputs.

If a show is playing, this request, or reception of a DMX frame, will stop the show from playing.

The full message for RGB LED strips is shown below. If the message has fewer slots, the last byte must contain the LED strip output control field, and the RGB data for missing slots will be zero.

Size in Bytes	Description
3	R1, G1, B1. 8-bit intensity of first Red, Green and Blue LEDs.
3	R2, G2, B2. 8-it intensity of second Red, Green and Blue LEDs.
...	
3	R170, G170, B170. 8-bit intensity of the last Red, Green and Blue LEDs.
2	Unused.
1	LED strip output control. Bits 3-2: LED strip output config, see table below. Bits 1-0: LED strip buffer number, see table below.

The full message for RGBW LED strips is shown below. If the message has less slots, the last byte must contain the LED strip output control field, and the RGBW data for missing slots will be zero.

Size in Bytes	Description
4	R1, G1, B1, W1. 8-bit intensity of first Red, Green, Blue and White LEDs.
4	R2, G2, B2, W2. 8-bit intensity of second Red, Green, Blue and White LEDs.
...	
4	R128, G128, B128, W128. 8-bit intensity of the last Red, Green, Blue and White LEDs.
1	LED strip output control. Bits 1: LED strip output Bits 0: LED strip section

10. Show Query Request (Label=7, no data)

This message requests the size of the non-volatile show storage memory.

11. Show Query Reply (Label=7)

Size in Bytes	Description
4	Size of the non-volatile show storage memory in bytes.
1	Memory busy status. 0: Erase or write has completed. 1: Erase or write is in progress.

12. Show Block Erase Request (Label=8)

This message erases one block of the non-volatile show storage memory so that all bytes are set to hex FF. The block size is 64 kbytes for all blocks, except for the first block which may be smaller. The show storage memory must be erased once before a new show can be written to the show storage memory.

After the PC issues a show erase request, the PC will issue show query requests, to detect when the erase has finished. This is necessary to prevent the PC from issuing another show erase request or show write request while the previous erase or write request is still in progress, which could cause a loss of API messages.

Size in Bytes	Description
4	Subcommand character array, set to 'ERAS'.
2	Block number to erase. Valid range is 0 to (number of show memory blocks – 1), where the number of show memory blocks is the integer (size from show query reply + hex FFFF) / hex 10000.

13. Show Sector Erase Request (Label=8)

This message erases one sector of the non-volatile show storage memory so that all bytes are set to hex FF. The sector size is 4 kbytes. The show storage memory must be erased once before a new show can be written to the show storage memory.

After the PC issues a show erase request, the PC should issue show query requests, to detect when the erase has finished. This is necessary to prevent the PC from issuing another show erase request or show write request while the previous erase or write request is still in progress, which could cause a loss of API messages.

Size in Bytes	Description
4	Subcommand character array, set to 'ERSE'.
2	Sector number to erase. Valid range is 0 to (number of show memory sectors – 1), where the number of show memory sectors is the integer (size from show query reply) / hex 1000.

14. Show Write Request (Label=8)

This message writes data to the non-volatile show storage memory. The show storage memory must be erased once before a new show can be written to the show storage memory.

After the PC issues a show write request, the PC should issue show query requests, to detect when the write has finished. This is necessary to prevent the PC from issuing another show erase request or show write request while the previous erase or write request is still in progress, which could cause a loss of API messages.

See the Show File Format section for a description of the show format.

Size in Bytes	Description
4	Subcommand character array, set to 'WRIT'.
4	Address. Valid range is 0 to (size from show query reply – number of data bytes to write).
1 to 256	Data to write.

15. Show Read Request (Label=9)

This message reads data from the non-volatile show storage memory.

Size in Bytes	Description
4	Address. Valid range is 0 to (size from show query reply – number of data bytes to read).
2	Number of bytes to read, with LSB stored at low address. Valid range is 1 to 256.

16. Show Read Reply (Label=9)

The board sends this message to the PC in response to the Show Read request. See the Show File Format section for a description of the show format.

Size in Bytes	Description
4	Address from show read request.
1 to 256	Data read.

17. Start Show Request (Label=8)

This message starts the replay of the stored show if the CRC of the stored show header record is good.

Size in Bytes	Description
4	Subcommand character array, set to 'STAR'.

18. Stop Show Request (Label=8)

This message stops the replay of the stored show.

Size in Bytes	Description
4	Subcommand character array, set to 'STOP'.

19.1Query Hardware Version Request (Label=14, no data)

This message requests the hardware version number of the board. The request is supported whether the firmware is valid.

20.Query Hardware Version Reply (Label=14)

The board sends this message to the PC in response to the Query Hardware Version request.

Size in Bytes	Description
1	Hardware version. Valid value is hex 30 for the Pixel Strip Driver.

Show File Format

This section specifies the format of the single show file that can be stored and replayed by the board. The show file consists of the following consecutive sections:

- Header record.
- Scene records.

The scene records contain frame records. Each frame record has the same fields as the LED Strip Update Request, except that the LED strip output config field is excluded.

1. Header Record

This message requests the board firmware to run the board bootstrap to enable reprogramming of the board firmware.

Size in Bytes	Description
4	File format version, set to 'PSA1'.
128	Show name text to identify the file. This field is not used by the board.
4	Number of frame records, with LSB stored at the low address. Minimum valid value is 1.
2	Number of bytes per frame record. Valid range is 2 to 513.
1	Bits 3-1: LED strip output config, where 0 = output 1 enabled with 1 universe, 1 = outputs 1 & 2 enabled with 2 universes per output, 2 = output 1 enabled with 2 universes. 4 = output 1 & 2 enabled with 1 universe per output, Bit 0: Play show at power on, where 0=disabled, 1=enabled.
1	Delay before looping, in seconds. Valid range is 0 to 255.
2	Loop count, with LSB stored at low address. Valid range is 0 to 65534. The show will play (Loop count + 1) times. 65535 is loop forever.
112	Unused, set to hex FF.
2	CRC is calculated over the rest of the header record, with LSB stored at low address.

The CRC is calculated using the XMODEM CRC polynomial (hex 1021), and an initial CRC value of 0.

The CRC is used to detect corruption of the record. If the CRC is bad, the show will not play.

2. Scene Record

Each scene record specifies when to play a group of 1 to 4 frames in the show, as well as the RGB or RGBW data to output for one scene of the show. The size of each frame record is specified in the header record.

The frames in the show must be stored in the scene record in increasing LED strip buffer number order.

Size in Bytes	Description
4	Bits 31-30: Number of frames in this scene record - 1. Bits 29-0: Time in milliseconds, with LSB stored at a low address.
2 to 513	Frame record for the first frame in the scene.
0 or 2 to 513	Frame record for the second frame in the scene, if present.
0 or 2 to 513	Frame record for the third frame in the scene, if present.

0 or 2 to 513	Frame record for the fourth frame in the scene, if present.
2	CRC is calculated over rest of scene record, with LSB stored at low address.

The CRC is calculated using the XMODEM CRC polynomial (hex 1021), and an initial CRC value of 0.

The CRC is used to detect corruption of the record. If the CRC is bad, the show will stop playing when the bad scene record is read.