

Communication with the LED PWM driver

For code *rgbv2-102.asm*

Packet Format and Description

The LED PWM driver uses a 9600bps serial interface to receive RGB data and commands from the control device.

Since communication is entirely unidirectional, that is the LED PWM drivers only receive data, and each device can have a unique address, multiple devices can be connected to a single serial bus and controlled individually.

Serial data is transferred asynchronously using a frame format of 1 start bit, 8 data bits and 1 stop bit. Data is transferred using a 6 frame fixed length packet. The packet format is shown below.

| Frame 1 | Frame 2 | Frame 3 | Frame 4 | Frame 5 | Frame 6 |
|-------------|----------|----------|----------|----------|----------|
| Start Frame | Payload1 | Payload2 | Payload3 | Payload4 | Checksum |

Start Frame

This must always be set to 0x81 (128)

Checksum

This frame must contain the two's complement of the modulo 256 sum of the previous 5 frames. Packets received with an incorrect checksum will be discarded.

The checksum is computed as shown in the examples below.

Checksum computation

PIC 14-bit core code

| | | |
|----------------------------------|--------------------|--------------------------|
| Frame1 + | <code>clrf</code> | <code>checksum</code> |
| Frame2 + | <code>movf</code> | <code>frame0, W</code> |
| Frame3 + | <code>addwf</code> | <code>checksum, F</code> |
| Frame4 + | <code>movf</code> | <code>frame1, W</code> |
| Frame5 + | <code>addwf</code> | <code>checksum, F</code> |
| Total | <code>movf</code> | <code>frame2, W</code> |
| | <code>addwf</code> | <code>checksum, F</code> |
| Checksum = 256 – (total mod 256) | <code>movf</code> | <code>frame3, W</code> |
| | <code>addwf</code> | <code>checksum, F</code> |
| | <code>movf</code> | <code>frame4, W</code> |
| | <code>addwf</code> | <code>checksum, F</code> |
| | <code>comf</code> | <code>checksum, F</code> |
| | <code>incf</code> | <code>checksum, F</code> |

Payload1

This frame determines how the device will interpret the whole packet

| <i>Value</i> | <i>Function</i> |
|-------------------------|---|
| 0x00 to 0x7F (0 to 127) | Single device (unicast) address |
| 0x80 (128) | Multicast group 1. Addresses 0x00 to 0x0F |
| 0x90 (144) | Multicast group 2. Addresses 0x10 to 0x1F |
| 0xA0 (160) | Multicast group 3. Addresses 0x20 to 0x2F |
| 0xB0 (176) | Multicast group 4. Addresses 0x30 to 0x3F |
| 0xC0 (192) | Multicast group 5. Addresses 0x40 to 0x4F |
| 0xD0 (208) | Multicast group 6. Addresses 0x50 to 0x5F |
| 0xE0 (224) | Multicast group 7. Addresses 0x60 to 0x6F |
| 0xF0 (240) | Multicast group 8. Addresses 0x70 to 0x7F |
| 0xFE (254) | Command packet |
| 0xFF (255) | Broadcast |

Unicast

Each device will have a unique address between 0x00 and 0x7F which allows up to 128 individually controlled devices on a single bus. A packet sent with a unicast address will only be processed by a device with that address.

Multicast

There are 8 multicast groups. A packet sent to a multicast address will be processed by any devices whose unicast address falls within the group. This allows data to be sent to up to 16 devices with a single packet rather than having to send 16 individual packets to the unicast addresses.

Broadcast

Packets sent to the broadcast address are processed by all devices on the bus regardless of their unicast address.

Command

A Payload1 frame with the value 0xFE (254) indicates the packet contains a Command Operation. All devices will respond to a Command packet regardless of their individual unicast address.

Non Command Packets

Packets sent with an address (unicast, multicast or broadcast) in the Payload1 frame should contain RGB data in Payload frames 2,3 and 4.

| Payload frame | Function | Value |
|---------------|-----------|------------------------------------|
| 1 | Address | Any unicast, muticast or broadcast |
| 2 | Red PWM | 0x00 to 0xFF (0 - 255) |
| 3 | Green PWM | 0x00 to 0xFF (0 - 255) |
| 4 | Blue PWM | 0x00 to 0xFF (0 - 255) |

A value of 0x00 is fully off (0% PWM duty cycle)

A value of 0xFF is fully on (99% PWM duty cycle)

The maximum duty cycle is 99% or an on ratio of 255/256.

The PWM refresh rate varies and will be between ~145Hz when receiving and processing packets and ~170Hz when idling (no serial data being received)

Command Functions

Packets received where the Payload1 frame contains the value 0xFE (254) are interpreted as a command. It is important to understand that a command packet will affect all devices on the bus regardless of their address.

| <i>Payload frame</i> | <i>Function</i> | <i>Value</i> |
|----------------------|-----------------|------------------------------|
| 1 | Command | 0xFE (254) |
| 2 | Operation | 0x01, 0x02, 0x04, 0x08, 0x10 |
| 3 | ComData1 | 0x00 to 0xFF (0 -255) |
| 4 | ComData2 | 0x00 to 0xFF (0 -255) |

The Command Operation is specified by the value in Payload2.

The operations are:

- Memory Transfer
- Memory Select
- Write Mode
- Reset
- Address Change

Sending a command packet with a value in the Payload 2 (Operation) frame that is not a value specified below may cause unexpected results.

Memory Transfer. Payload2 = 0x01 (1)

RGB data in the specified memory is transferred to the LED PWM driver. All devices respond to this command allowing synchronous updates to the LED PWM driver on all devices.

| <i>Payload frame</i> | <i>Function</i> | <i>Value</i> |
|----------------------|-----------------|-----------------------|
| 1 | Command | 0xFE (254) |
| 2 | MemTransfer | 0x01 (1) |
| 3 | Memory # | 0x00 to 0x0E (0 – 14) |
| 4 | ComData2 | Not used - Don't care |

Memory Select. Payload2 = 0x02 (2)

The specified memory is made active. When the device is in *Write Memory* mode, all RGB data sent to the device by unicast, muticast or broadcast will be written into the active memory location.

| <i>Payload frame</i> | <i>Function</i> | <i>Value</i> |
|----------------------|-----------------|-----------------------|
| 1 | Command | 0xFE (254) |
| 2 | MemSelect | 0x02 (2) |
| 3 | Memory # | 0x00 to 0x0E (0-14) |
| 4 | ComData2 | Not used – Don't care |

Write Mode. Payload2 = 0x04 (4)

The controller transfers received RGB data to the LED PWM driver in one of two ways.

1. Direct mode
2. Memory mode

In *Write Direct* mode RGB data is written directly to the LED PWM driver.

In *Write Memory* mode RGB data is written into the active memory only. It is not sent to the LED PWM driver. In order for the RGB data in the memory to be displayed by the LED PWM driver a *Memory Transfer* command must be issued.

| <i>Payload frame</i> | <i>Function</i> | <i>Value</i> |
|----------------------|-----------------|-----------------------|
| 1 | Command | 0xFE (254) |
| 2 | WriteMode | 0x04 (4) |
| 3 | ModeSelect | 0x00 to 0xFF (0 -255) |
| 4 | ComData2 | Not used - Don't Care |

ModeSelect = 0x00 (0) Write Direct
ModeSelect = 0x01 to 0xFF (1-255) Write Memory

The *Memory Select* and *Write Mode* operations are 'sticky'. The selected memory or write mode remains effective until another command is sent that changes it.

The *Memory Select* operation will set the active memory even if the device is in *Write Direct* mode however to write to the active memory it must be in *Write Memory* mode.

Soft Reset . Payload2 = 0x08 (8)

Following this command the device will.

- Reload its address from non-Volatile memory
- Set all RGB memories to 0x00 (0)
- Set the active memory to 0x00 (0)
- Set Write Mode to Direct

| <i>Payload frame</i> | <i>Function</i> | <i>Value</i> |
|----------------------|-----------------|-----------------------|
| 1 | Command | 0xFE (254) |
| 2 | SoftReset | 0x08 (8) |
| 3 | ComData1 | Not used - Don't Care |
| 4 | ComData2 | Not used - Don't Care |

Note: Any packet sent less than 500µS after sending a reset command may be missed by the device while it is resetting.

Address Change. Payload2 = 0x10 (16)

This command allows the current unicast address of the device to be changed to a new address.

The new address is saved to the non-volatile EEPROM memory and becomes effective immediately following the command.

| <i>Payload frame</i> | <i>Function</i> | <i>Value</i> |
|----------------------|-----------------|-------------------------------------|
| 1 | Command | 0xFE (254) |
| 2 | AddressChange | 0x10 (16) |
| 3 | NewAddress | 0x00 to 0x7F (0 – 127) |
| 4 | CurrentAddress | 0x00 to 0x7F (0- 127) or 0xFF (255) |

When an address change packet is received the device compares its address to the Current Address in Payload4. If it matches the device will update itself to use the New Address specified in Payload3. If it does not match the command is ignored.

If the Current Address value is set to 0xFF (255), all devices will accept the update and change their address to the value in New Address.

This 'any device' change is not intended for use when multiple devices are attached to the bus since it will assign the same address to all devices that receive it. The intended purpose is to assign a known address to a device when, for any reason, its current address is unknown. It is therefore recommended that this command only be issued when a single device is attached to the bus.

If the old and new addresses are the same, it will rewrite the address to non-volatile EEPROM memory.

RGB Memory operation

The device has 15 memories. RGB data can be written to any of the memories and then recalled for display by the LED PWM driver.

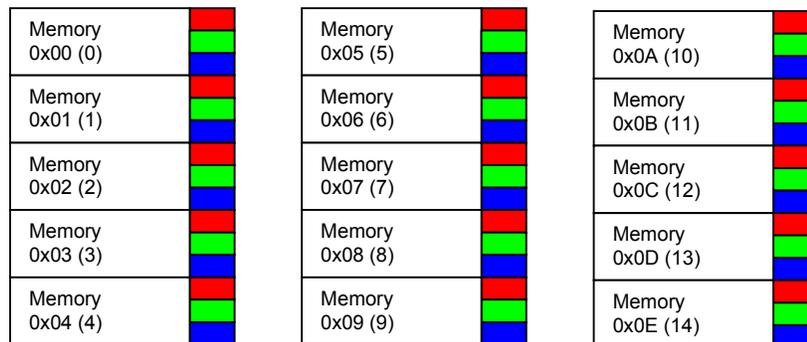
If the device is set to *Write Memory* mode, RGB data received by the device is only written to the active memory.

The active memory is selected by sending a *Memory Select* command packet. All devices on the serial bus will set their active memory to the value specified in the command.

To transfer the contents of any memory to the RGB PWM driver requires a *Memory Transfer* command to be sent containing the memory number to transfer.

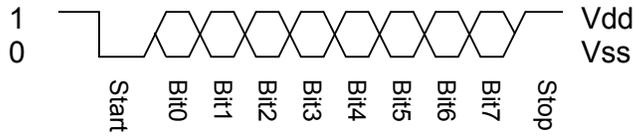
All devices receive and respond to a *Memory Transfer* command. This allows many devices to change the LED PWM data synchronously by sending a single packet. By preloading the 15 memories of many devices, it is then only necessary to send repeated *Memory Transfer* commands to create a wide range of sequences and effects.

When the device is in *Write Direct* mode the *Memory Transfer* command can still be used to transfer previously loaded RGB data from the memory to the LED PWM driver.



Serial frame format

Serial data should be sent using the following frame format.



Note: RS232 uses inverted voltages to represent the logic levels
0 = +3 to + 12V
1 = -3 to - 12V

Power-on Reset

When the device is first powered up it will:

- Load its address from non-Volatile memory
- Set all RGB memories to 0x00 (0)
- Set the active memory to 0x00 (0)
- Set Write Mode to Direct

Allow 500mS for the device to initialise before sending serial data to it.