



# PD-LED-3

Version 3.1 - November 12, 2018

## 1 General Driver Board Description

The P-ROC Driver Boards are used to control connected devices by turning on or off power to the devices in response to commands from a P-ROC. The boards receive commands over an RS-485 serial bus, which allows many boards to be chained together and used simultaneously. There are multiple types of Driver Boards that can be used to make up a chain including: Master, Power Driver 16, Power Driver Matrix 8x8, and PD-LED-2.

## 2 PD-LED-3 Overview

### 2.1 LEDs

The PD-LED-2 has 84 individual drive pins and can therefore control up to 84 individual LEDs in parallel or up to 682 individually addressable serially-chained RGB LEDs. When controlling LEDs in parallel, each drive pin operates independently, based on commands sent to the board from an external RS-485 bus master. The commands are interpreted by a local logic chip, which then turns the individual drive pins on or off as instructed. To control individually addressable serially-chained RGB LEDs, a few of the drive pins can be configured to drive the data for up to 6 LED chains.

To control the LEDs, there are 2048 color registers. For the parallel-controlled drive pins, color registers 0-83 each control a drive pin. When serial chains are being driven, any or all of the 2048 color registers can drive LEDs in the serial chain(s). The PD-LED-2 supports driving up to three individual WS281x LED chains and up to three individual LPD880x LED chains. Each of the chains can be configured to use consecutive sets of the LED color registers, of which there are 2048. Parallel-driven LEDs and serially chained LEDs can be controlled simultaneously. Note - when using the PD-LED-2 to drive LED serial chains, the serial chains must be powered externally. The PD-LED-2 should not be used to supply power to the chain(s).

The PD-LED-2 can be programmed to set the intensity level of an LED to one of 256 levels. Level

0 is off, meaning the board will drive the pin low (ground). Level 255 is on, meaning the board will drive the pin high (3.3v). All other levels result in the board driving the pin on and off repeatedly, where the on and off times are different, depending on the level. High level values have long on times and short off times, and low level values value have short on times and long off times. The resulting PWM (pulse width modulation) determines the intensity of the LED.

The board can also be programmed to fade the intensity level from one level to another. When fading, the board will increment or decrement the drive intensity level at a programmable rate, one level at a time until the final level is reached.

### **RGB (Red-Green-Blue) LEDs:**

Since RGB LEDs are essentially 3 LEDs in one package, this board can control up to 28 RGB LEDs (84 / 3 = 28) in parallel or 682 individually addressable RGB LEDs connected through serial chains. 256 levels per LED results in a possible color depth for RGB LEDs of 16M colors (256\*256\*256).

### **Power / Current:**

Each LED output is limited to 22mA of current. For designs requiring more current to be delivered to one or more LEDs, use the PD-LED-2 board's LED outputs to turn on/off transistors or FETs, through which more current can be supplied from a power source.

## **2.2 Motors**

Some of the PD-LED-3 drive pins can be used to control servos and stepper motors. Note - the pins used for servo and/or stepper control are shared with the serial RGB string control. So system designers will need to choose which features they want to use in their systems.

### **2.2.1 Servos**

There are 12 servo-control-capable drive pins on the PD-LED-3, and each can be connected to a separate servo. Just like when using the drive pins to control LEDs, the outputs are 3.3v and limited to 22mA current. The servos need to have external connections to power.

### **2.2.2 Stepper Motors**

The PD-LED-3 can work in external stepper motor controllers, like the StepStick or EasyDriver, to control up to 2 stepper motors. 3 drive pins are used for each stepper motor. One is for pulsing the motor, one is for motor direction, and the other is to enable the motor controller's sleep function.

# 3 Programming

Configuring the PD-LED-2 and changing the colors of the LEDs its controlling require writing to registers. There are a set of direct registers and a set of indirect registers. Indirect registers are programmed by writing to the direct data and address registers. Refer to the Direct Registers and Indirect Registers sections for a listing of each register or register range.

LED colors are changed by directly writing the color registers. LEDs can automatically fade to another color by writing a fade rate followed by the destination color.

For serial LED chain control, the serial chain control logic must be configured. First the full range of LEDs that are mapped to serial chains is programmed into a serial string manager. The serial string manager can also be configured with timing parameters for WS281x-type serial chains. (Note - The PD-LED-2 can be configured to control either WS2811 or WS2812 serial strings at one time. Even multiple WS2811 or multiple WS2812 serial strings can be driven at once, but both timings (WS2811 and WS2812) can not be used at the same time). Next, each of three WS281x serial chain controllers and each of three LPD880x serial chain controllers is configured with the range of LEDs they should drive. Finally each of the six serial chain controllers is individually enabled.

## 3.1 Changing an LED's color

To change the color of an individual LED:

1. Write the lower 8 bits of the LED's index number to the Address direct register [7:0].
2. Write the upper 8 bits of the LED's index number to the Address direct register [15:8].
3. Write the desired intensity level to the Color direct register.

## 3.2 Fading an LED's color

To fade an LED from its current intensity level to another:

1. Write the lower 8 bits of the fade rate to the Fade Rate [7:0] direct register.
2. Write the upper 8 bits of the fade rate to the Fade Rate [15:8] direct register.
3. Write the lower 8 bits of the LED's index number to the Address direct register [7:0].
4. Write the upper 8 bits of the LED's index number to the Address direct register [15:8].
5. Write the desired intensity level to the Fade Destination Color direct register.

## 3.3 Enabling a Serial Chain

There are six enable bits in the Serial Chain Control Register, which is accessed through the indirect register bus. Each bit enables one of the Serial Chain Controllers, of which there are three for WS281x chains and three for LPD8806 chains. When a chain is enabled, the corresponding pins (See Connectors section) are used for driving the serial chain rather than driving parallel LEDs.

To enable one or more serial LED chains:

1. Write the address of the Serial Chain Enable Register into the Data direct register.
2. Set the desired enable bits in the Address direct register [7:0].

### 3.4 Configuring the Serial Chain Manager

The Serial Chain Manager determines which of the PD-LED-2's LEDs are available to be driven onto serial chains. There are two sub-managers in the Serial Chain Manager. The first is configured with a starting LED number and an ending LED number. The second is configured with just a starting number, as the second sub-manager is assumed to make available the same number of LEDs as the first sub-manager. Note - making LEDs available to LED chains doesn't require them to be driven onto chains, but only LEDs that are made available to LED chains can be driven onto chains.

The Serial Chain Manager works by retrieving the LED color information for a sequence of LEDs and then making that data available to Serial Chain Controllers so they can drive the data onto the serial chain pins. All Serial Chain Controllers are updated at the exact same speed, and that speed is determined by the configuration of the Serial Chain Manager. The more LEDs that the Serial Chain Manager services and the slower the timing parameters are configured, the longer it takes to update each Serial Chain Controller. Upon finishing an update cycle, the Serial Chain Manager immediately starts the next cycle.

By using both sub-managers, the time it takes to service all of the LEDs can be halved. When using both sub-managers, two LED values are retrieved at the same time (one by sub-manager A and one by sub-manager B). Note, the Serial Chain Controllers can only service data from one sub manager at a time. So using both sub-managers usually means there are at least two serial chains.

Indirect registers used to configure the Serial Chain Manager include:

<b>Serial String A Start</b> - Sub-Manager A's first LED
<b>Serial String A End</b> - Sub-Manager A's last LED
<b>Serial String B Start</b> - Sub-Manager B's first LED
<b>WS281x Low Bit Time</b> - The total number of 32 MHz clock cycles to drive the data signal high for a low bit. (Default is 13 for WS2812)
<b>WS281x High Bit Time</b> - The total number of 32 MHz clock cycles to drive the data signal high for a high bit. (Default is 24 for WS2812)
<b>WS281x End Bit Time</b> - The total number 32 MHz clock cycles to drive the data signal high and then low for a single data bit. (Default is 40 for WS2812)
<b>WS281x Reset Bit Time</b> - The total number of 32 MHz clock cycles to drive the data bit low to complete the update cycle for the entire chain. (Default is 1603 for WS2812)

As a timing example, assume that 500 RGB LEDs are driven onto LED chains, and the chains use the default WS2812 timing. The total time it takes to update the entire chain is  $((500 \text{ RGBs} * 3 \text{ (for 1500 individual LEDs)} * 40 \text{ clock cycles}) + 1603) * (1 / 32 \text{ MHz}) = 1.925\text{ms}$ .

To configure the Serial Chain Manager:

1. Write the address of the desired indirect register into the Data direct register.
2. Write the desired data into the Address direct registers.

### 3.5 Configuring the Serial Chain Controllers

There are six Serial Chain Controllers, each of which connects directly to FPGA pins (1 pin for WS281x, 2

pins for LPD880x). Each Controller is configured with the first and last LED numbers it should onto the connected chain. The data for all of the LEDs between the first and last LED numbers, inclusive, will be driven each cycle. The cycle update frequency depends on the configuration of the Serial Chain Manager.

To configure a Serial Chain Controller:

3. Write the address of the desired indirect register into the Data direct register.
4. Write the desired data into the Address direct registers.

## **3.6**

## 4 Connectors

<b>J1</b>	<b>Logic Power</b>	<b>Required</b>
<b>2-pin Molex: 0.156" spacing</b>		
1	5V	I
2	Ground	I

<b>J2</b>	<b>Serial Data In</b>	<b>Required</b>
<b>3-pin Molex: 0.100" spacing</b>		
1	Serial+	I
2	Serial-	I
3	Ground	I

<b>J3</b>	<b>Serial Data Out</b>	<b>Not Required</b>
<b>3-pin Molex: 0.100" spacing</b>		
1	Serial+	O
2	Serial-	O
3	Ground	O

<b>J5</b>	<b>LED Drivers</b>				
<b>20-pin Header: 0.100" spacing</b>					
1	3.3V	O	2	LED 0	O
3	LED 1	O	4	LED 2	O
5	Ground	O	6	3.3V	O
7	LED 3	O	8	LED 4	O
9	LED 5	O	10	Ground	O
11	3.3V	O	12	LED 6	O
13	LED 7	O	14	LED 8	O
15	Ground	O	16	3.3V	O
17	LED 9	O	18	LED 10	O
19	LED 11	O	20	Ground	O

<b>J9</b>	<b>LED Drivers</b>				
<b>20-pin Header: 0.100" spacing</b>					
1	3.3V	O	2	LED 12	O

3	LED 13	0	4	LED 14	0
5	Ground	0	6	3.3V	0
7	LED 15	0	8	LED 16	0
9	LED 17	0	10	Ground	0
11	3.3V	0	12	LED 18	0
13	LED 19	0	14	LED 20	0
15	Ground	0	16	3.3V	0
17	LED 21	0	18	LED 22	0
19	LED 23	0	20	Ground	0

<b>J6</b>	<b>LED Drivers</b>				
<b>20-pin Header: 0.100" spacing</b>					
1	3.3V	0	2	LED 24	0
3	LED 25	0	4	LED 26	0
5	Ground	0	6	3.3V	0
7	LED 27	0	8	LED 28	0
9	LED 29	0	10	Ground	0
11	3.3V	0	12	LED 30	0
13	LED 31	0	14	LED 32	0
15	Ground	0	16	3.3V	0
17	LED 33	0	18	LED 34	0
19	LED 35	0	20	Ground	0

<b>J10</b>	<b>LED Drivers</b>				
<b>20-pin Header: 0.100" spacing</b>					
1	3.3V	0	2	LED 36	0
3	LED 37	0	4	LED 38	0
5	Ground	0	6	3.3V	0
7	LED 39	0	8	LED 40	0
9	LED 41	0	10	Ground	0
11	3.3V	0	12	LED 42	0
13	LED 43	0	14	LED 44	0
15	Ground	0	16	3.3V	0
17	LED 45	0	18	LED 46	0
19	LED 47	0	20	Ground	0

<b>J7</b>	<b>LED Drivers</b>				
<b>20-pin Header: 0.100" spacing</b>					
1	3.3V	0	2	LED 48	0
3	LED 49	0	4	LED 50	0
5	Ground	0	6	3.3V	0

7	LED 51	0	8	LED 52	0
9	LED 53	0	10	Ground	0
11	3.3V	0	12	LED 54	0
13	LED 55	0	14	LED 56	0
15	Ground	0	16	3.3V	0
17	LED 57	0	18	LED 58	0
19	LED 59	0	20	Ground	0

<b>J11</b>	<b>LED Drivers</b>				
<b>20-pin Header: 0.100" spacing</b>					
1	3.3V	0	2	LED 60	0
3	LED 61	0	4	LED 62	0
5	Ground	0	6	3.3V	0
7	LED 63	0	8	LED 64	0
9	LED 65	0	10	Ground	0
11	3.3V	0	12	LED 66	0
13	LED 67	0	14	LED 68	0
15	Ground	0	16	3.3V	0
17	LED 69	0	18	LED 70	0
19	LED 71	0	20	Ground	0

<b>J8</b>	<b>LED Drivers</b>				
<b>20-pin Header: 0.100" spacing</b>					
1	3.3V	0	2	LED / Servo 0	0
3	LED 73 / Servo 1	0	4	LED 74 / Servo 2	0
5	Ground	0	6	3.3V	0
7	LED 75 / Servo 3 / Stepper 1 Sleep / LPD880x Serial Clock 2	0	8	LED 76 / Servo 4 / Stepper 1 Pulse / LPD880x Serial Data 2	0
9	LED 77 / Servo 5 / Stepper 1 Direction / LPD880x Serial Clock 1	0	10	Ground	0
11	3.3V	0	12	LED 78 / Servo 6 / Stepper 0 Sleep LPD880x Serial Data 1	0
13	LED 79 / Servo 7 / Stepper 0 Pulse / LPD880x Serial Clock 0	0	14	LED 80 / Servo 8 / Stepper 0 Direction / LPD880x Serial Data 0	0
15	Ground	0	16	3.3V	0
17	LED 81 / Servo 9 / WS281x Serial Data 2	0	18	LED 82 / Servo 10 / WS281x Serial Data 1	0
19	LED 83 / Servo 11 / WS281x Serial Data 0	0	20	Ground	0



## 5 Status LEDs

<b>LED</b>	<b>Meaning</b>
D1	3.3V
D2	5.0V
D3	Watchdog expired
D4	Ready

# 6 Dipswitches

Switch	Meaning*
1	Address bit 0
2	Address bit 1
3	Address bit 2
4	Address bit 3
5	Address bit 4
6	LED reset value
7	Watchdog disable
8	Terminate serial bus

\* On=1, Off=0

## 6.1 Serial Chain Termination

The last board in the physical chain (not necessarily the highest address) needs to be set to terminate the serial chain. This is done by setting dipswitch 8 on.

## 6.2 Board Address

In order to receive the correct data from the bus master, the board address needs to be set appropriately. The following table describes how to use the dipswitches to set the board address:

Dipswitch	Meaning
1	Address bit 0*
2	Address bit 1*
3	Address bit 2*
4	Address bit 3*
5	Address bit 4*

\*On=1, Off=0

# 7 Direct Registers

Serial Bus Register Address	Serial Bus Data
0	Address [7:0]
1	Color
2	Fade Destination Color
3	Fade Rate [7:0]*
4	Fade Rate [15:8]*
5	RESERVED
6	Address [15:8]
7	Data

Note - Registers are all 8-bits wide.

\* Fade Rates have 4ms resolutions (1 = 4ms, 2 = 8ms... 256 = 1s)

# 8 Indirect Registers

Note 1 - To access indirect registers, first write the direct address registers, and then write the direct data register. Writing the direct data register is what causes the write to complete.

Note 2 - The meaning of the Direct Data Register and Direct Address Register are counter-intuitive. The contents of the Direct Data Register serve as the function address, and the contents of the Direct Address Register serve as the function data.

Direct Data Register	Direct Address Register
0	Control
1	Serial String A Start
2	Serial String A End
3	Serial String B Start
4	WS281x Low Bit Time
5	WS281x High Bit Time
6	WS281x End Bit Time
7	WS281x Reset Bit Time
8	WS281x[0] First Address
9	WS281x[0] Last Address
10	WS281x[1] First Address
11	WS281x[1] Last Address
12	WS281x[2] First Address
13	WS281x[2] Last Address
15:14	RESERVED
16	LPD880x[0] First Address
17	LPD880x[0] Last Address
18	LPD880x[1] First Address
19	LPD880x[1] Last Address
20	LPD880x[2] First Address / Servo Enables
21	LPD880x[2] Last Address / Servo Pulse Count
22	Stepper Ticks Per Half Period
23	Stepper Data 0
24	Stepper Data 1
25	Stepper Data 2

## 8.1 Control Register

Data Bit	Description
0	use_ws281x[0]

1	use_ws281x[1]
2	use_ws281x[2]
3	use_lpd880x[0]
4	use_lpd880x[1]
5	use_lpd880x[2]
7:6	RESERVED
8	use_stepper[0]
9	use_stepper[1]
15:10	RESERVED

## 8.2 Servo Enables Register

Data Bit	Description
0	Enable output 72 as a servo controller
1	Enable output 73 as a servo controller
2	Enable output 74 as a servo controller
3	Enable output 75 as a servo controller
4	Enable output 76 as a servo controller
5	Enable output 77 as a servo controller
6	Enable output 78 as a servo controller
7	Enable output 79 as a servo controller
8	Enable output 80 as a servo controller
9	Enable output 81 as a servo controller
10	Enable output 82 as a servo controller
11	Enable output 83 as a servo controller

## 8.3 Servo Pulse Count

Data Bit	Description
9:0	Servo Pulse Count
15:10	RESERVED

Servo Pulse Count: Number of 32 MHz clock pulses for a 100% duty cycle servo output.

## 8.4 Stepper Ticks Per Half Period

Data Bit	Description
15:0	Stepper Ticks Per Half Period

Stepper Ticks Per Half Period: Number of 32 MHz clock pulses for half of a stepper pulse period. A full period defines the time to drive the stepper pulse signal high and then low again at 50% duty cycle.

## 8.5 Stepper Data [0-2]

Data Bit	Description
14:0	Pulse Count
15	Direction

# PDB Protocol

The PDB Protocol describes the format of data being received on the Data input pin (J8: pin-1). This is also the format of the serial data sent out on J10.

Serial Bus Protocol (PDB Protocol)	
Bit	Field
Byte 0	{0,0,Board Address[5:0]}
<between bytes>	1-5 IDLE clock cycles
Byte 1	{0,0,0,0,0,Command[2:0]}:  0: Read 1: Write 2: RESERVED 3: RESERVED 4: RESERVED 5: RESERVED 6: RESERVED 7: Clear all registers on board
<between bytes>	1-5 IDLE clock cycles
Byte 2	{0,0,Register Address[5:0]}:  0: Bank A 1: Bank B
<between bytes>	1-5 IDLE clock cycles
Byte 3	{0,0,0,0,Bank data bits[7:4]}
<between bytes>	1-5 IDLE clock cycles
Byte 4	{0,0,0,0,Bank data bits[3:0]}

Each byte is sent MSB first.

A transaction consists of 5 data bytes. There must be at least 10 IDLE cycles between transactions. Between each byte of a transaction there must be between 1 and 5 IDLE cycles.

The data signal must be high during all IDLE cycles.

Data must be sent at 8 MHz, 125ns per bit.

The Bank data bits in Bytes 3 and 4 are all active high. A '1' will turn the corresponding circuit on, and a '0' will turn the corresponding circuit off.

## 9 Hardware Installation:

- Mount the board using the 4 mounting holes.
- Connect a 5V supply to J1.
- Connect J2 to the previous board in your chain using a 2-wire cable. If the boards are separated by more than a few feet, a shielded & twisted pair is recommended. For short runs, any 2-wire cable should suffice.
- Connect LEDs to the appropriate pins of J5-J11.