Table of Contents

1. ABSTRACT

This user manual describes the functionality and characteristics of the AL5887 RGB LED driver using demo board which is an I2C/SPI bus controlled, 36 channel, constant current LED driver. This user manual includes hardware and software setup instructions, schematic diagram, bill of materials, printed-circuit board layout drawings and demo board images.

2. INTRODUCTION

This demo board characterizes the features of AL 5887 RGB LED driver. The main goal is to exercise vivid LED effects by communicating through I2C/SPI. This demo board has an additional feature of providing supply to the LEDs using power bank connected through USB Type-C (J350) connector and supply voltage to led driver through micro USB connector(J407).

Figure 2: Image of Demo Board

3. DESCRIPTION

The demo board consists of the following major components:

3.1. AL5887 RGB LED driver

The AL5887 is a 36 channel RGB LED driver with integrated color mixing and brightness control. This driver is comprised of 36 programmable LED current channels each with internal 12-bit PWM for color and brightness control through SPI or I2C digital interface. This is ideal for up to 12 RGB LED modules lighting applications with 3 programmable banks (A B C) for software control of each color. The global output current of all 36 channels can be set up by an

external resistor. Each channel current can digitally be configured up to 70mA under the thermal limitation of the package.

Features of the AL5887 are controlled via programmable SPI/I2C digital interface. Supports 400Khz I2C and 2MHz SPI interface. Using a dedicated INT_SEL pin, SPI/I2C protocol can be selected. The AL5887 has a 30 kHz, 12-Bit PWM generator for each channel, as well as independent color mixing and brightness control registers for each RGB module to enable vivid LED effects with zero audible noise. Provision for connecting up to 4 devices using two external hardware address pins. Ultra low quiescent current with four modes of operation (shutdown, standby, normal and power save mode).

3.2. FT4222H USB to I2C/SPI Interface

FT4222H is a high speed USB to Quad-SPI/I 2C interface Device Controller. This requires an external Crystal (12 MHz) for the internal PLL to operate. This contains SPI/ I 2C configurable interfaces. The SPI interface can be configured in master mode with single, dual, or quad bits' data with transfer or in slave mode with single bit data width transfer. The I 2C interface can be configured in master or slave mode.

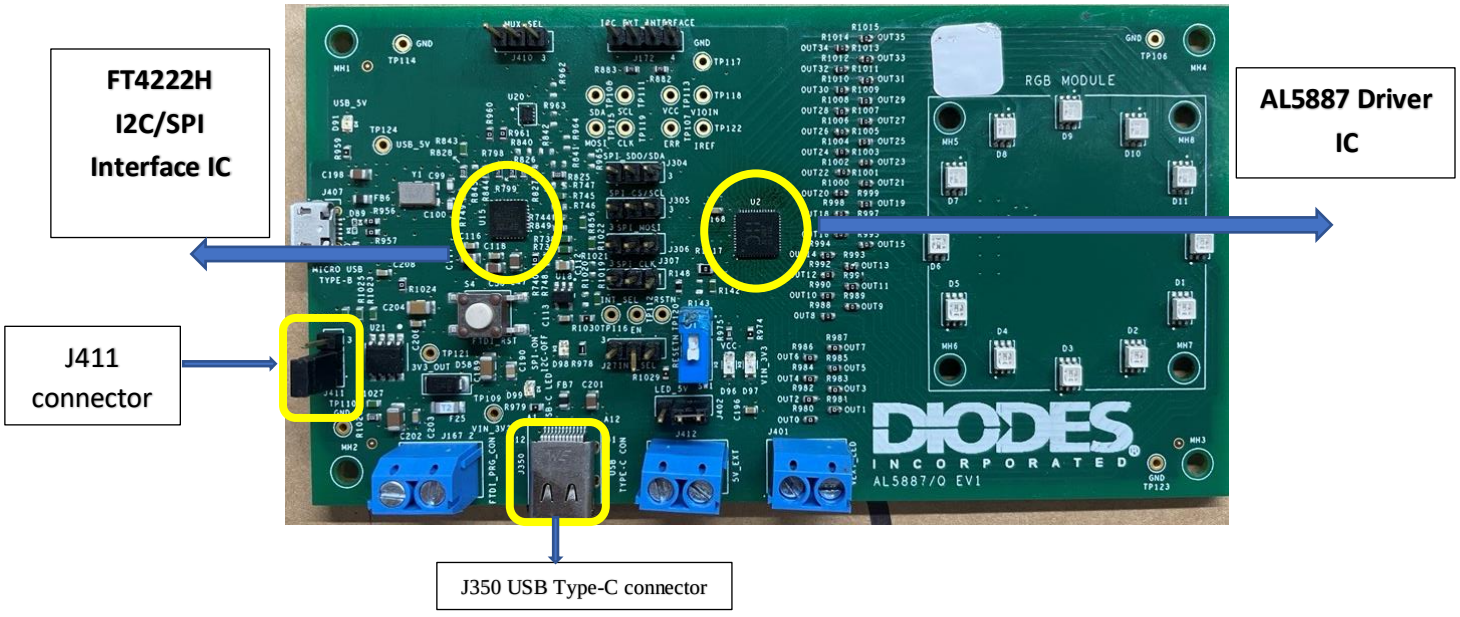
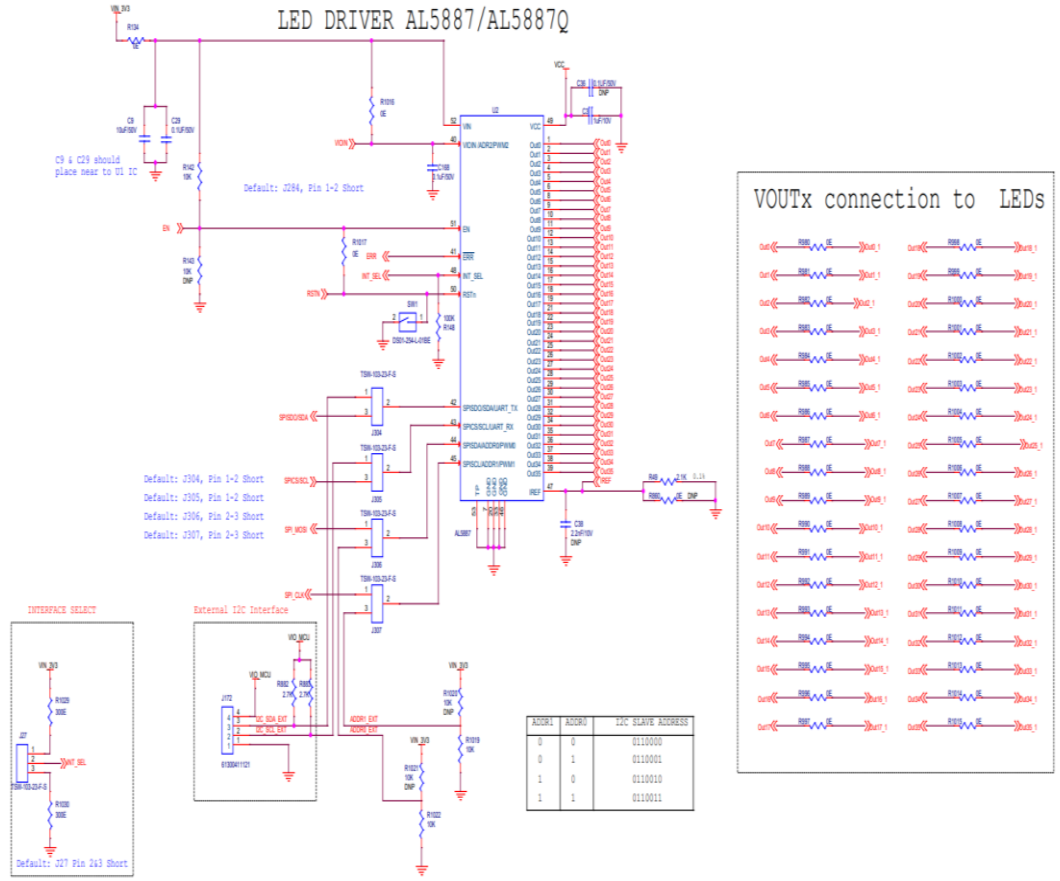


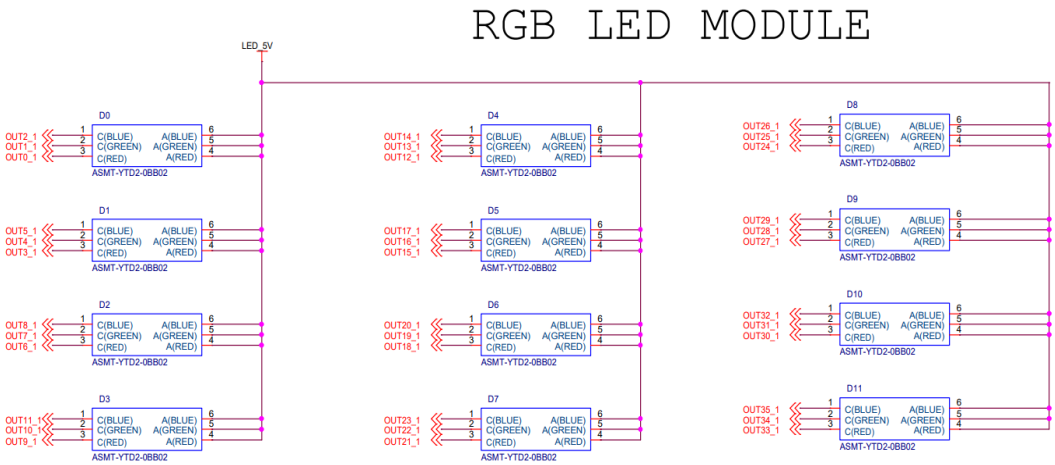
Figure 3: Image of demo board representing AL5887 IC and FT4222H IC

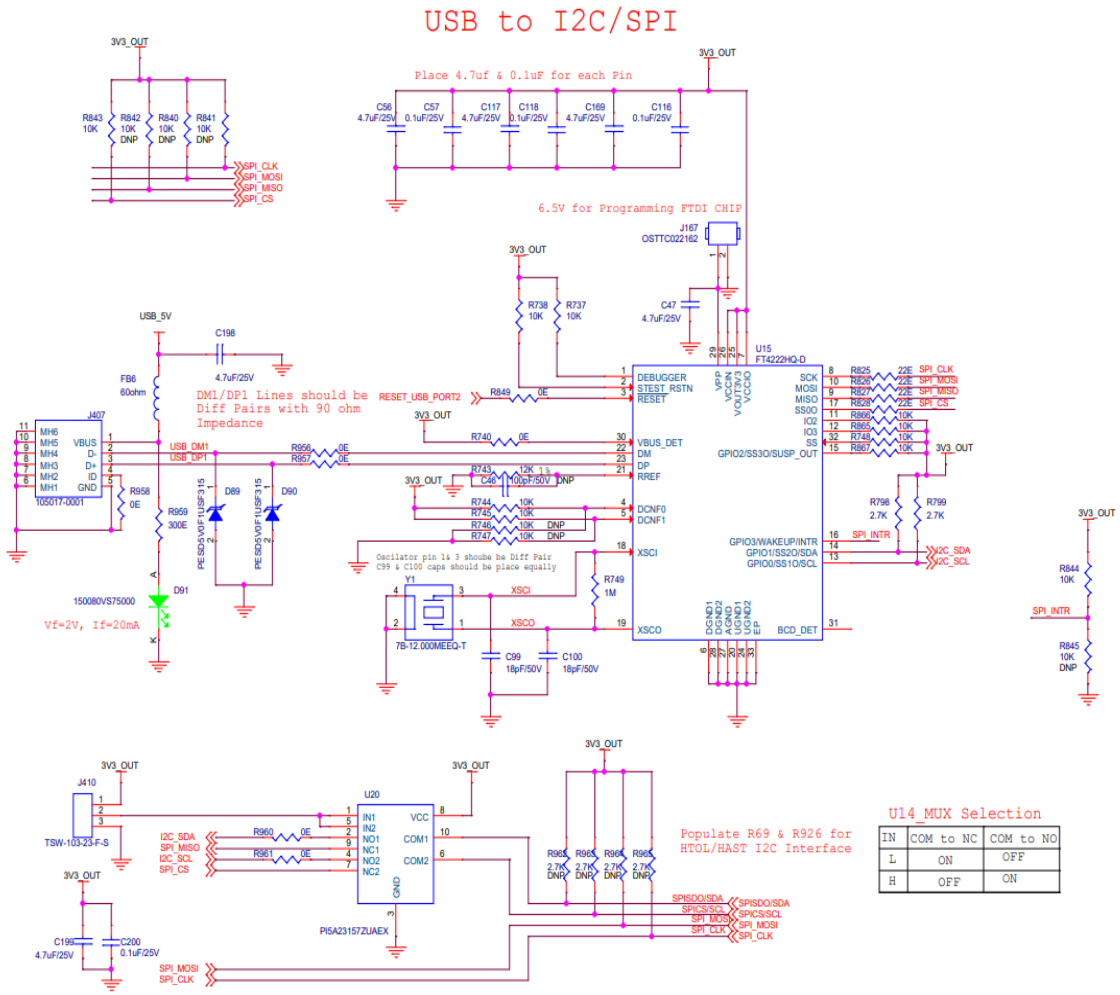
The demo board has an additional feature i.e., the demo board can powered up through USB to micro USB Type-B connector(USB_VBUS) connected at J407 connector and short 2-3 pins of the J411 connector. The LED can be powered up by conneting battery bank through USB Type-C connector(J350) and short 2-3 pins of the J402 connector. For this setup refer jumper settings given below in table3.

4. SCHEMATICS

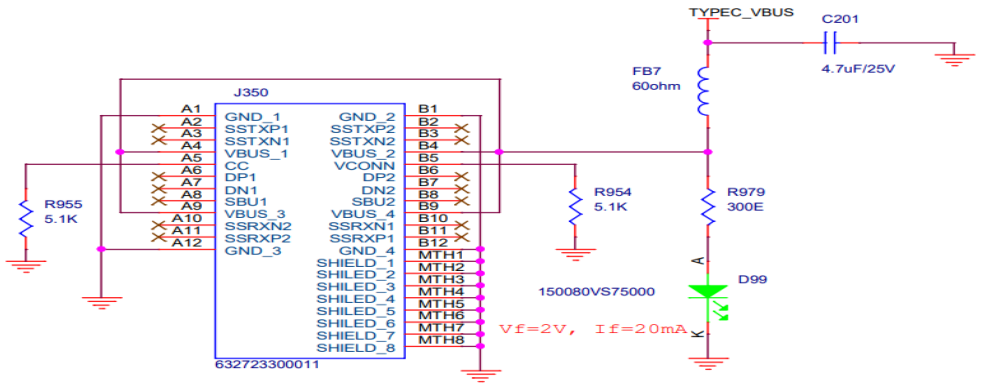
The schematic of the AL5887 RGB LED driver demo board is as shown below.

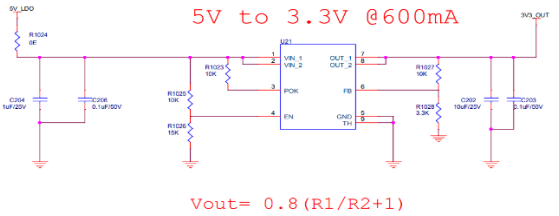


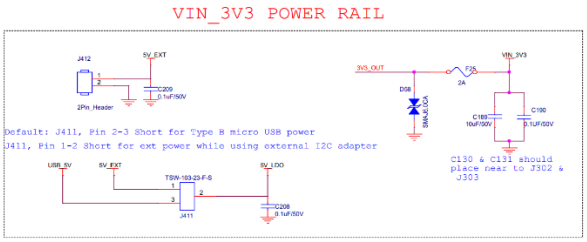


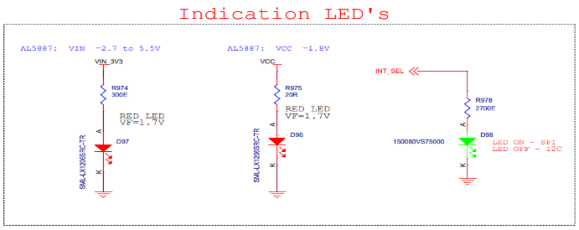


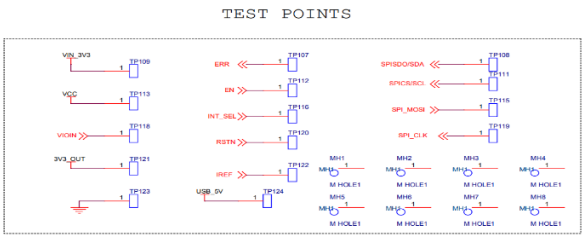
TYPE C: BATTERY BANK INPUT











5. BILL OF MATERIAL

The components used in the demo board are listed below with their part numbers.

DESIGNATOR	VALUE	PART NUMBER	REMARKS
C3	1uF/10V	C0805C105K8RACTU	
C9,C189	10uF/50V	GMC31X7R106K50NT	
C29,C190	0.1UF/50V	08055C104KAT4A	
C36	0.1UF/50V	08055C104KAT4A	DNP
C38	2.2nF/10V	8.85012E+11	DNP
C46	100pF/50V	CL21C101JBANNNC	DNP
C47,C56,C117,C169,C198,	4.7uF/25V	CGA4J1X7R1E475M125AE	
C199,C201			
C57,C116,C118,C200	0.1uF/25V	CL10B104MB8NNNC	
C99,C100	18pF/50V	06035A180JAT4A	
C112	1uF/10V	GRM21BR71A105KA01L	
C113	10nF/50V	06035C103KAT4A	
C168,C196,C203,C206,C207,	0.1uF/50V	CL10B104MB8NNNC	
C208,C209			
C202	10uF/25V	CL31B106KAHVPNE	
C204	1uF/25V	C1206C105M3RAC7800	
D1,D2,D3,D4,D5,D6,D7,D8,	ASMT-YTD2-0BB02	ASMT-YTD2-0BB02	
D9,D10,D11,D0			
D58	SMAJ6.0CA	SMAJ6.0CA	
D89,D90	PESD5V0F1USF315	PESD5V0F1USF315	
D91,D98,D99	150080VS75000	150080VS75000	
D96,D97	SML-LX1206SRC-TR	SML-LX1206SRC-TR	
FB6,FB7	60ohm	BLM21PG600SH1D	
F25	2A	0685T2000-01	
J27,J304,J305,J306,J307,	TSW-103-23-F-S	TSW-103-23-F-S	
J402,J410,J411			
J167	OSTTC022162	OSTTC022162	
J172	61300411121	61300411121	
J350	6.32723E+11	6.32723E+11	
J401,J412	2Pin_Header	OSTTC022162	
J407	105017-0001	105017-0001	
MH1,MH2,MH3,MH4,MH5,MH6,	M HOLE1		not neccesary
MH7,MH8			

R49	2.1K	RQ73C1J2K1BTD	
R134,R1016,R1017	0E	RMCF0805ZT0R00	
R142,R737,R744,R745	10K	RC0603FR-7W10KL	
,R748			
,R843,R844,R856,			
R865,R866,R1019,R1020,			
R1021,R1022,R1023,R1025,			
R1027			
R148	100K	RCA0603100KFKEAHP	
R738,R867	10K	RC0603FR-7W10KL	
R740,R849,R956,R957,	0E	RC0603JR-070RL	
R958,R960,R961,R980,R981,			
R982,R983,R984,R985,R986,			
R987,R988,R989,R990,R991,			
R992,R993,R994,R995,R996,			
R997,R998,R999,R1000,			
R1001,R1002,R1003,R1004,			
R1005,R1006,R1007,R1008,			
R1009,R1010,R1011,R1012,			
R1013,R1014,R1015,R1024			
R743	12K	RT0603DRD0712KL	
R749	1M	RN73H1JTTD1004B25	
R798,R799,R882,R883,	2.7K	RMCF0603FT2K70	
R825,R826,R827,R828	22E	CRGCQ0603F22R	
R954,R955	5.1K	RMCF0603JT5K10	
R959,R974,R979,R1029,	300E	ESR03EZPJ301	
R1030			
R975	20R	RNCP0603FTD20R0	
R978	2700E	ERJ-PB3D2701V	
R1026	15K	SG73S1JTTD1502F	
R1028	3.3K	CRGH0603F3K3	
SW1	DS01-254-L-01BE	DS01-254-L-01BE	
S4	FSM2JSMAATR	FSM2JSMAATR	
TP106,TP110,TP114,TP117	5124	5124	not neccesary
TP107,TP108,TP109,TP111,	TEST POINT		not neccesary
TP112,TP113,TP115,TP116,			
TP118,TP119,TP120,TP121,			
TP122,TP123,TP124			
U2	AL5887	AL5887	
U15	FT4222HQ-D	FT4222HQ-D	
U18	TLV840MADL29DBVR	TLV840MADL29DBVR	
U20	PI5A23157ZUAEX	PI5A23157ZUAEX	
U21	AP7165-SPG-13	AP7165-SPG-13	
Y1	7B-12.000MEEQ-T	7B-12.000MEEQ-T	
R143,R1020,R1021,R840,R841,R842,R746,R747,R845	10K	RC0603FR-7W10KL	DNP
R860	0E	RC0603JR-070RL	DNP
R962,R963,R964,R965	2.7K	RMCF0603FT2K70	DNP

6. SETUP REQUIREMENTS

The user will get following contents along with the demo board:

- Demo board.
- Demo board user manual.
- Emulator Software.
- Emulator software user manual.

6.1.Software:

Emulator Software:

Emulator is a standalone application developed using LabVIEW 2019 to support AL5887 LED driver testing. Features of AL5887 are controlled via SPI/I2C interface. Minimum system requirements for execution of Emulator software are given below

Item No.	Description	Specification/Requirement
1.	OS	Windows 7/10
2.	RAM	4GB or above
3.	Disk space	250MB approx..

Refer emulator software user manual for installing emulator software in PC/Laptop.

6.2.Hardware setup:

- We need DC power supply with 5V/5A, as the max. current consumed when all the LEDs with full brightness is approx. 1.5A.
- PC/Laptop in which Emulator software is installed.
- A USB to Micro USB Type-B connecting cable from PC/Laptop to Demo board.
- Arrange the setup as shown in the figure below.

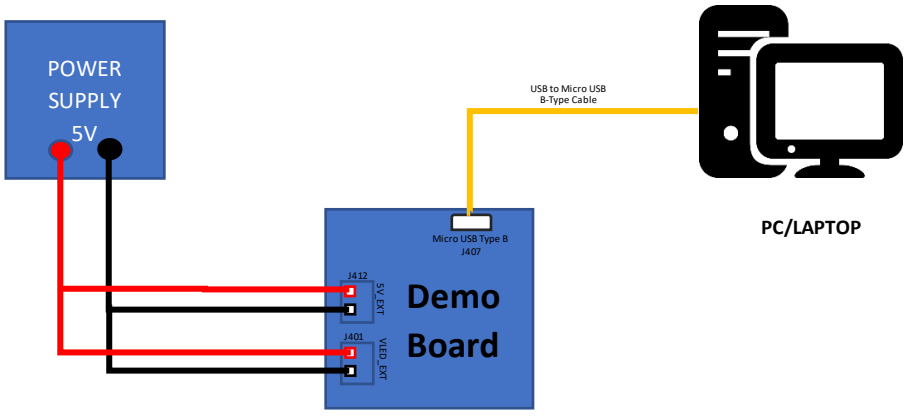


Figure 4: Basic hardware setup representation

HARDWARE SETUP:

7. Connections

- Connect V_{IN} (5V) to J412 (5V_Ext) connector. Ensure Vin less than 5.5V
- Connect V_{LED} (5V) to J401 (VEXT_LED) connector or a Battery bank via Type C Connector(J350). Place appropriate jumper at J402.
- Place the jumpers as per the **default** jumper settings given in the Table.1 below to communicate via I2C communication.

CONNECTOR	PURPOSE	DEFAULT CONNECTION with I2C and External Supply to RGB LEDs
J304	SPISDO/SDA	PIN 2-3 SHORT
J305	SPICS/SCL	PIN 2-3 SHORT
J306	SPI_MOSI	PIN 1-2 SHORT
J307	SPI_CLK	PIN 1-2 SHORT
J27	INTERFACE SELECT	PIN 2-3 SHORT: I2C
J402	LED_5V POWER RAIL	PIN 2-3 SHORT: VEXT_LED
J411	VIN_3V3 POWER RAIL	PIN 1-2 SHORT: 5V_EXT
J410	MUX SELECTION	PIN 1-2 SHORT: I2C

Table 1: Default Jumper Settings

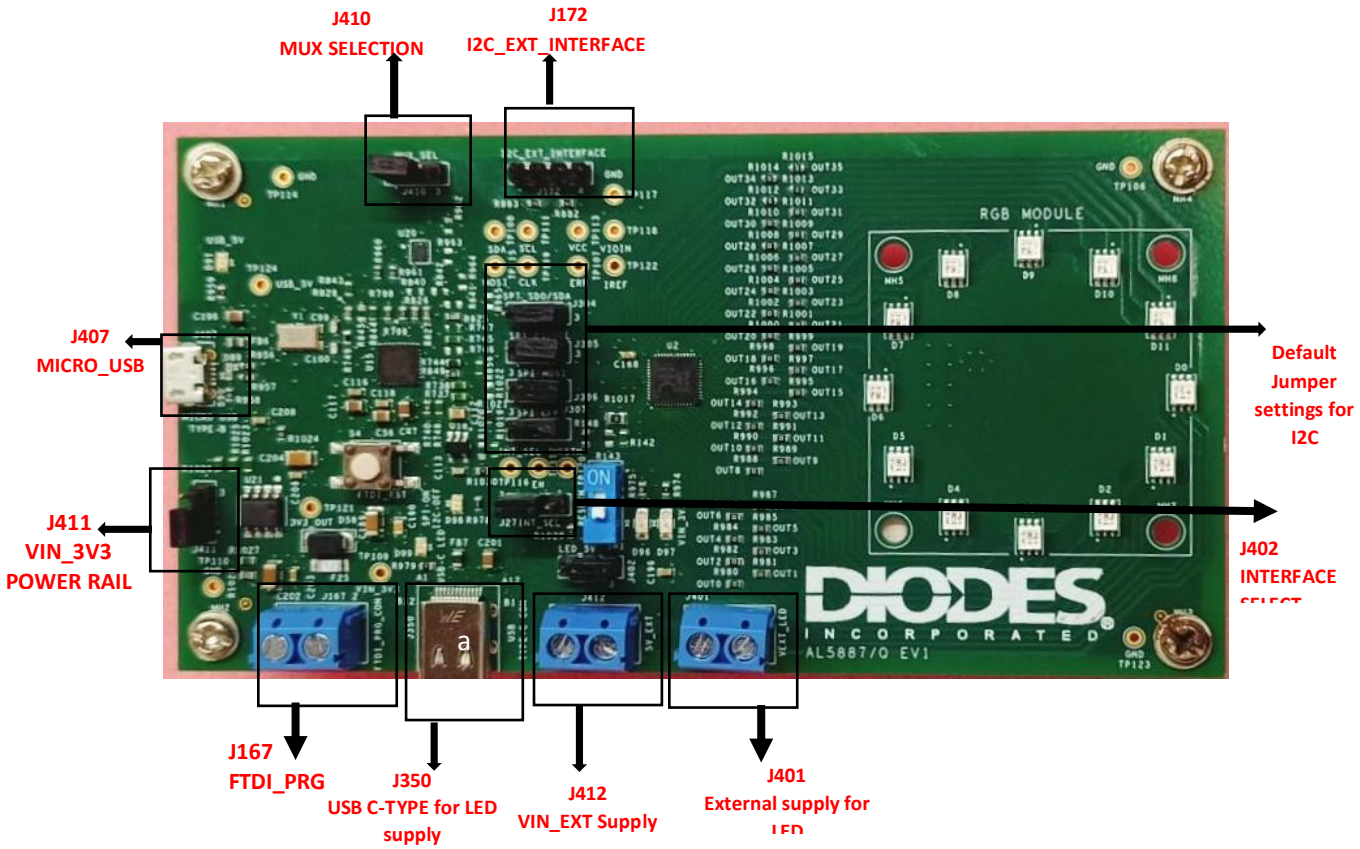


Figure 5: Demo board representing all the Connectors and Jumpers.

- The slave address selection resistor table is given below. Solder / De-solder resistors to change the configuration

ADDR1	ADDR0	I2C SLAVE ADDRESS	RESISTOR ARRANGEMENT		Remarks
			MOUNT	REMOVE	
0	0	0110000	R1019, R1022	R1020, R1021	**Default**
0	1	0110001	R1019, R1021	R1020, R1022	
1	0	0110010	R1020, R1022	R1019, R1021	
1	1	0110011	R1020, R1021	R1019, R1022	

Table 2: **Address selection table**

- Connect USB to Micro USB connector at J407 to establish communication (I2C/SPI) from GUI.
- To communicate with AL5887 using external I2C bus (not from GUI), refer jumper setting table (Table .3)

8. Check list before turning ON the board

- Ensure that V_{IN} given at J412 is less than 5.5V to avoid device damage.
- Ensure LED External supply V_{LED} given at J401 is not exceeding 5V.
- Make sure the jumper settings are changed as given in Table 3 and for default settings refer Table 1.
- Check the RESETN switch (SW1) is in OFF condition.

9. Turn ON Procedure

1. Turn on the External 5V power supply given at connector J412 and check for the power on indication at LED D97(RED LED).
2. Check LED D96 lit on the board which indicates AL5887 LDO output supply.
3. Power ON LED Supply (Either External 5V supply on J401 or Battery Bank).
4. Open GUI in Desktop/Laptop and run the exe file shared along with this manual.
5. Connect USB to Micro USB connector between desktop/Laptop to Demo Board Connector (J407).
6. Operate LEDs from GUI using the controls given. Refer emulator software manual shared along with this manual.

10. Jumper Settings

Connect the jumpers as per user requirement and connection table is given below.

CONNECTOR	PURPOSE	CONNECTION
J304	I2C_SDA_EXT	PIN 1-2 SHORT
	SPISDO/SDA	PIN 2-3 SHORT
J305	I2C_SCL_EXT	PIN 1-2 SHORT
	SPICS/SCL	PIN 2-3 SHORT
J306	SPI_MOSI	PIN 1-2 SHORT
	ADDR0_EXT	PIN 2-3 SHORT
J307	SPI_CLK	PIN 1-2 SHORT
	ADDR1_EXT	PIN 2-3 SHORT
J27	INTERFACE SELECT	PIN 1-2 SHORT: SPI
		PIN 2-3 SHORT: I2C
J402	LED_5V POWER RAIL	PIN 1-2 SHORT: TYPEC_VBUS
		PIN 2-3 SHORT: VEXT_LED
J411	VIN_3V3 POWER RAIL	PIN 1-2 SHORT: 5V_EXT
		PIN 2-3 SHORT: USB_5V
J410	MUX SELECTION	PIN 1-2 SHORT: I2C
		PIN 2-3 SHORT: SPI

Table 3: **Jumper connection table.**

11. BOARD LAYOUT

The following images are the top layer and bottom layer gerber images of demo board.

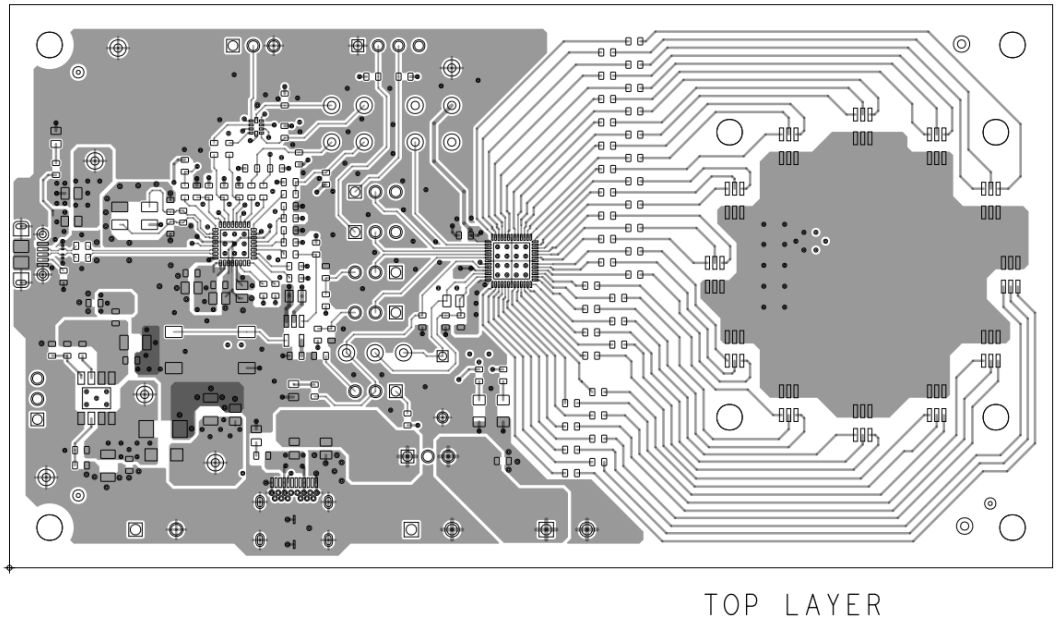


Figure 5: TOP LAYER

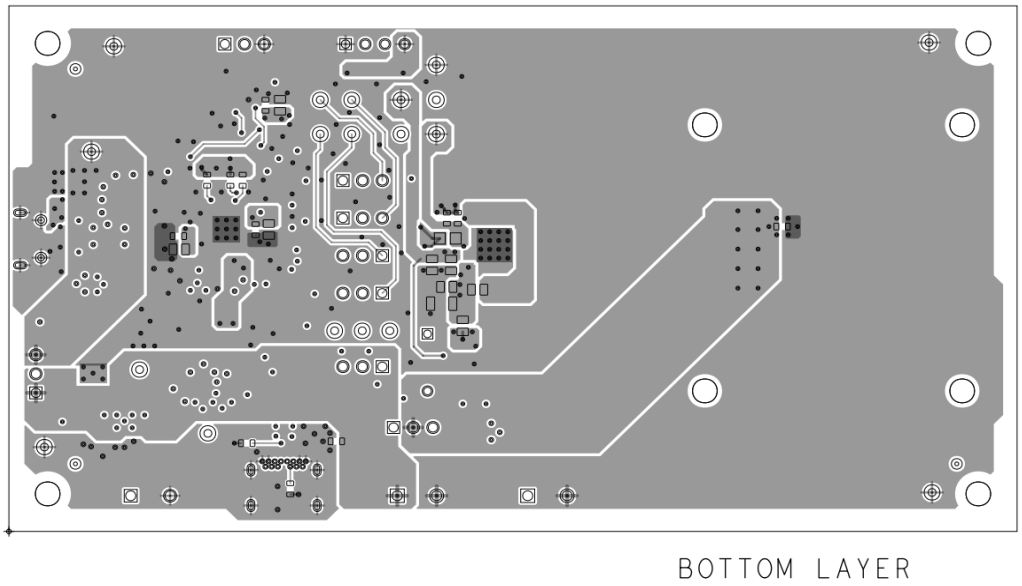


Figure 6: BOTTOM LAYER

12. BOARD IMAGES

The following images are top view, bottom view of the demo board and few LED effects performed using demo board.

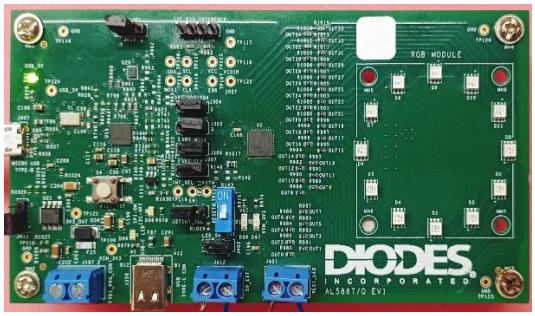


Figure 7: TOP VIEW

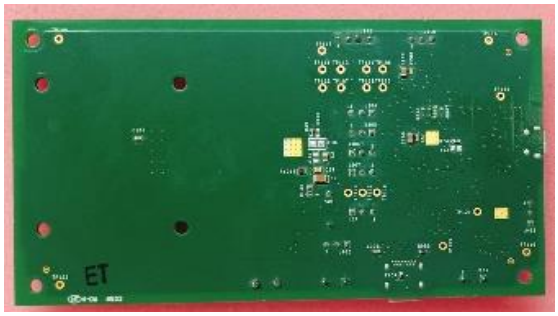


Figure 8: BOTTOM VIEW

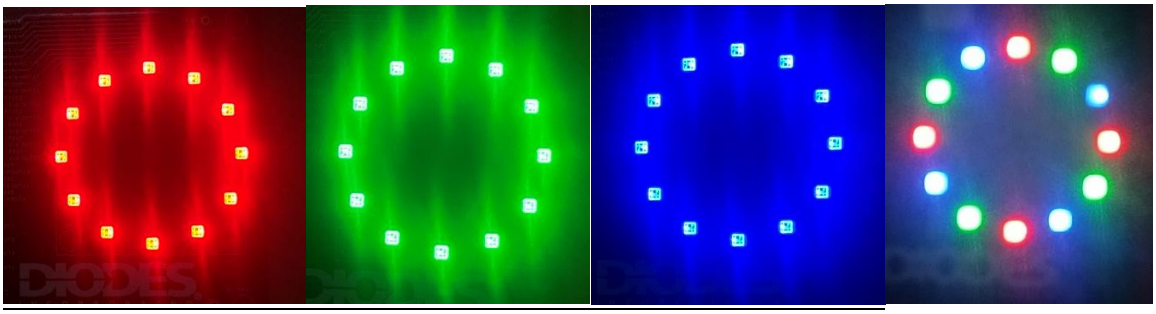


Figure9: Shows various LED effects, starting from right, a. RED, b. GREEN, c. BLUE, d. mixture of RGB

13. EXTERNAL I2C INTERFACE USING DEMO BOARD:

Using demo board with external I2C interface, the way to communicate is as shown below.

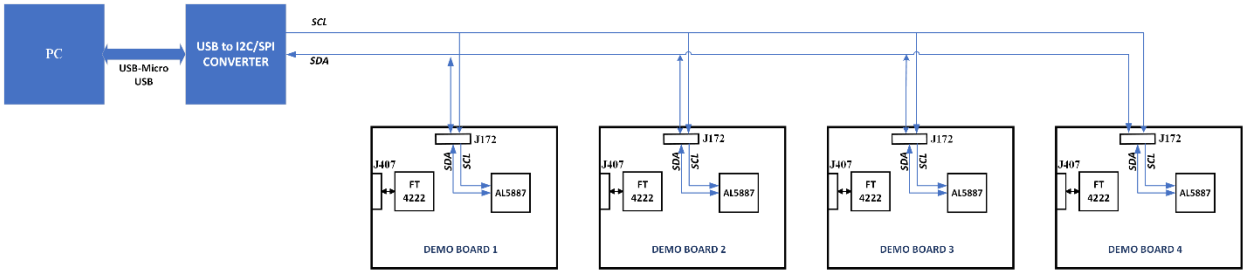


Figure10: PC to Demo board.

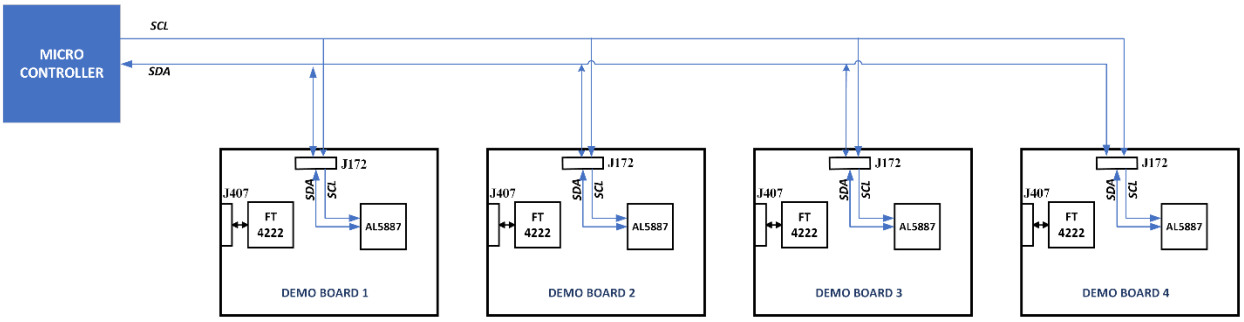


Figure11: Micro controller to Demo board

Follow the steps mention below to set jumper settings while using external I2C interface,

Step 1: Set the following jumper settings.

CONNECTOR	PURPOSE	CONNECTION
J27	INT_SEL	PIN 2-3 SHIORT
J304	I2C_SDA_EXT	PIN 1-2 SHORT
J305	I2C_SCL_EXT	PIN 1-2 SHORT
J410	MUX_SELECTION	PIN 1-2 SHORT: FOR I2C

Step 2: Set slave address, refer Table 2 to set address.
Step 3: Connect demo board to PC or Micro controller as shown in figure10 &11 at J172 connector and the connections are as follows,

CONNECTOR	PIN	PURPOSE
J172	2	I2C_SDA_EXT
J172	3	I2C_SCL_EXT
J172	4	GND

14. TROUBLESHOOT:

- Current Limit:

  If board is drawing excess current, remove the resistor R134, then check for the cause.

- Verify the following test points to ensure the required voltage levels.

i) VIN_3V3 – 3.3V (TP 109)

ii) EN – 3.3V (TP 112)

iii) VCC – 1.8V (TP 113)

iv) 3V3_V_{OUT} – 3.3V (TP 121)

v) RSTn – 3.3V (TP 120)

vi) V_{IOIN} – 3.3V (TP 118)

vii) GND – 0V (TP 123)

viii) ERR – (TP 107)

ix) INT_SEL – 3.3V (TP 116)

x) IREF – 0.7V (TP 122)

- To verify the communication, reconnect the cable. Probe oscilloscope on the following test points,
    1. SPISDO / SDA – TP108
    2. SPICS / SCL – TP111
    3. SPIMOSI – TP115
    4. SPICLK – TP119
- If LEDs are not glowing, then check the LED supply and also check LED behavior using multi meter.
- For open and short, read the register data using GUI and check for corresponding fault bit value. If any fault bit is set high. Then check the corresponding channel is open / short.

15. Additional Information:

15.1 Device functional modes:

1. **Normal mode:** The AL5887 device enters the NORMAL mode when Chip_EN (register) = 1. ICC is 5 mA (typical). The voltage at IREF test point (TP122) should be 696mV – 704mV.
2. **Power save mode:** Automatic power-save mode is enabled when register bit Power_Save_EN = 1 (default) and all the LEDs are off (both color and brightness registers = 00H) for a duration of >30 ms. Almost all analog blocks are powered down in power-save mode. If any I2C/SPI command to the device occurs, the AL5887 device returns to NORMAL mode.

3. **Shutdown mode**: The device enters into SHUTDOWN mode from all states on VIN power down or when EN = Low >25ms. I_{CC} is < 1 µA (max).
4. **Standby mode**: The device enters the STANDBY mode when Chip_EN (register bit) = 0. In this mode, all the OUTx are shut down, but the registers retain the data and keep it available via I2C/SPI. STANDBY is the low-power-consumption mode, when all circuit functions are disabled. ICC is 15 µA (maximum).
5. **Thermal shutdown mode:** The device automatically enters the THERMAL SHUTDOWN mode when the junction temperature exceeds 160°C (typical). In this mode, all the OUTx outputs are shut down. If the junction temperature decreases below 150°C (typical), the device returns to the NORMAL mode.

15.2 Current Setting for all channels:

The maximum global output current for all 36 channels can be adjusted by the external resistor, RSET, as described below.

*IMAX = KIREF * VIREF / RSET * [(Max_Current_Option/4) +(3/4)]* ……..(1)

where, IMAX = Channel average current, Color Register=FF, Brightness Register=FF VREF= 0.7 V, RSET = External dimming resistor (2.1kΩ recommended), Max_Current_Option = 1 (default), KIREF = 21 + (N * 3), is the current multiplication factor which can be programmed using 6-bit global dimming register G5:G0 (Address = 66H), which is analog dimming register and N is the decimal equivalent of G5 G4 G3 G2 G1 G0.

For example, if all global dimming register bits are 0, the N will be decimal equivalent of 100000 which is 32. Hence, KIREF = 21 + (32 * 3) = 117.

using equation (1) above, for global dimming register setting of 000000H and Max_Current_Option = 1, below is the table that shows IMAX variation with respect to the RSET.

R_{SET}(KΩ)	I_{MAX}(mA)	K_{IREF}
2.1(Recommended)	39	117
14.7	5.57	117
36.5	2.24	117

Table 4: IMAX variation w.r.to Rset

Similarly, the below table shows IMAX range using global dimming at different RSET values

$R_{SET}(K\Omega)$	$I_{MAX}(mA)$		
	MIN	DEFAULT	MAX
2.1(Recommended)	7	39	70
14.7	1	5.57	10
36.5	0.4	2.24	4

Table 5: IMAX range using global dimming at different Rset values.

For more information on current setting refer data sheet.

15.3 Brightness and Color register:

The below table shows various values of brightness and color register w.r.t duty cycle.

Duty Cycle	Brightness Register(Hex)	Color Register(Hex)
10	2D	92
20	5F	8A
30	75	A9
40	81	CC
50	96	DC
60	B8	D6
70	C0	EF
80	EF	DC
90	F3	F3
100	FF	FF

Table 6: Brightness and Color Register Values.

Emulator Software

Emulator is a standalone application developed using LabVIEW 2019 to support AL5887 LED driver testing. Features of AL5887 are controlled via SPI/I2C interface.

Minimum system requirements for execution of Emulator software are given below

#	Description	Specification/Requirement
1	OS	Windows 7/10
2	RAM	4GB or above
3	Required Disk space	250 MB approx.

Note: Requires Admin Rights for Installation and has to be run as Administrator.

16 Installation instructions:

1. Download the latest Emulator software package from the Diodes Server.
2. Unzip Emulator_Installer.zip.

   The extracted folder will have the following files and folders.

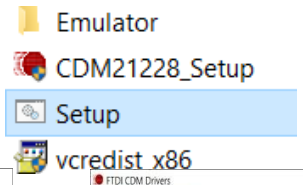


Figure 12: Emulator Installer Folder

Figure 13: FTDI Driver installation

3. Click the **Setup.bat** and follow the on-screen instructions to complete the installation of FTDI Drivers.

4. Next **Microsoft Visual C++ 2010 x86 Redistributable** installation will begin automatically. Follow the on-screen instructions to complete the installation.

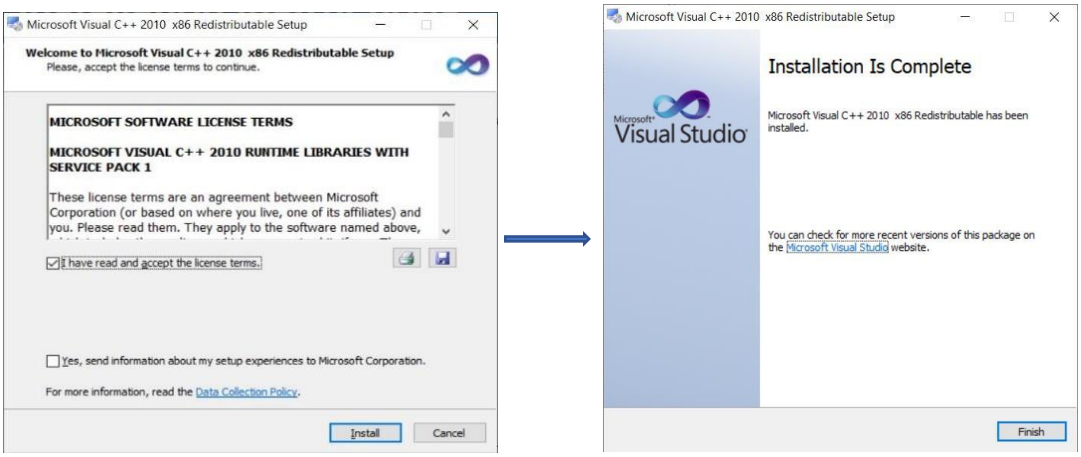


Figure 14: Microsoft Visual C++ 2010 x 86 Redistributable Installation

5. Emulator setup will be installed following Microsoft Visual C++ 2010 x86 Redistributable. Follow on screeninstructions and complete the installation and Restart the system when prompted

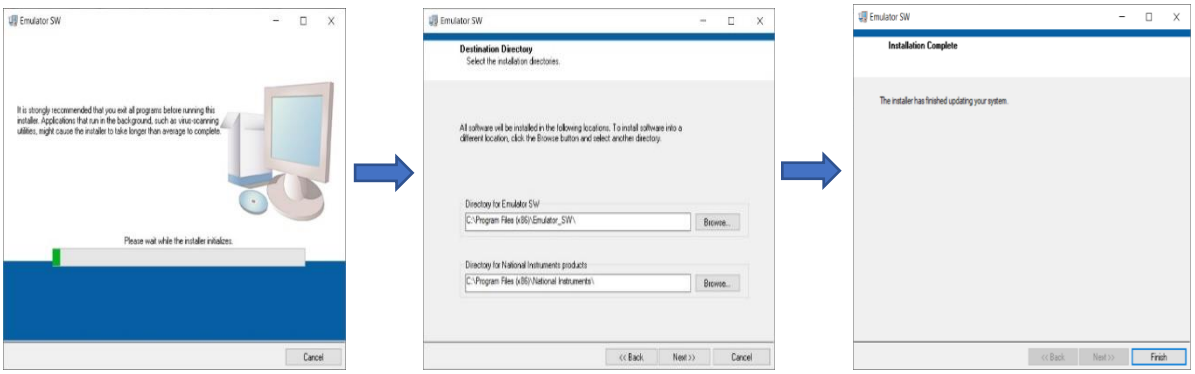


Figure 15: Microsoft Visual C++ 2010 x 86 Redistributable Installation

17 Getting started:

Run the Emulator software **Emulator_SW_V2.7.exe**

a) From **Start menu** by selecting the below icon.

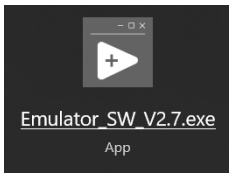


Figure 16: Emulator SW EXE Icon

b) From C:\Program Files (x86)\Emulator_SW\ Emulator_SW_V2.7.exe.

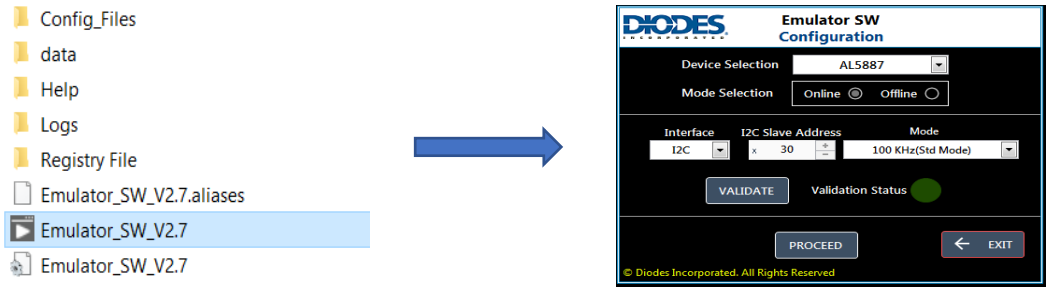


Figure 17: Emulator SW Execution

Note: It is recommended to run the executable in Administrator Mode.

18 Software Features – Configuration Screen

Emulator SW has two major modules namely **Configuration Screen** and **Emulator Screen**.

1. **Configuration Screen** allows users to
   - Select the Device under test and configure its Interface parameters.
   - Perform Device Validation based on device and Interface Configuration.
   - Mode Selection - Either Online or Offline Mode.

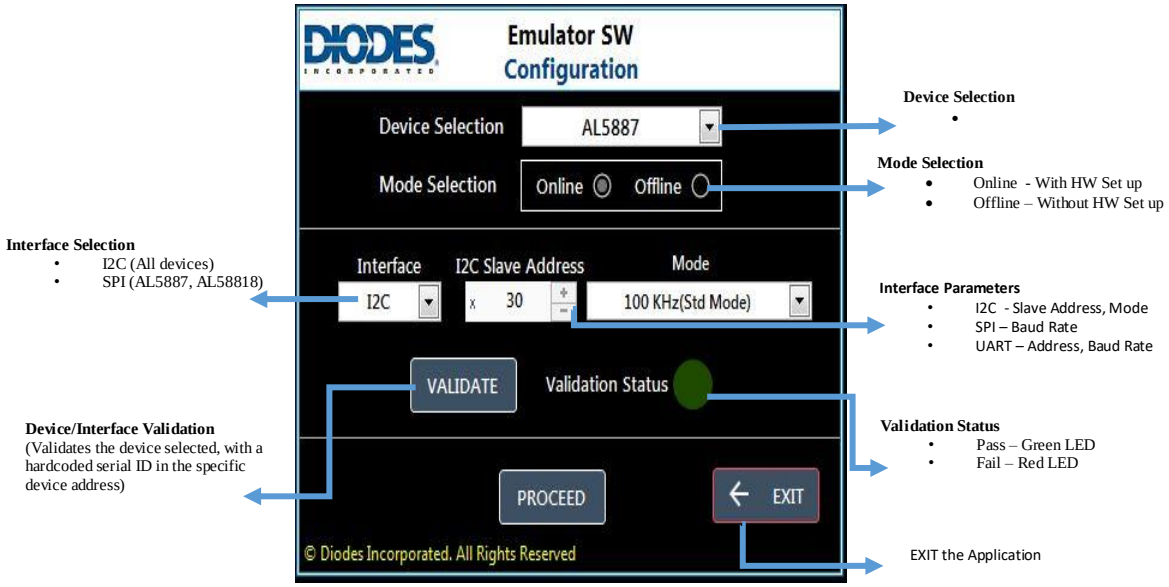


Figure 18: Emulator configuration screen

2. **Emulator Screen** has the following major features
   - Display the parameters configured in CONFIGURATION Screen.
   - LED Test Tab to display

- The Device Connection Status.
- The LED Annunciation in 2 different modes
    1. RGB Modules and
    2. Individual LEDs.

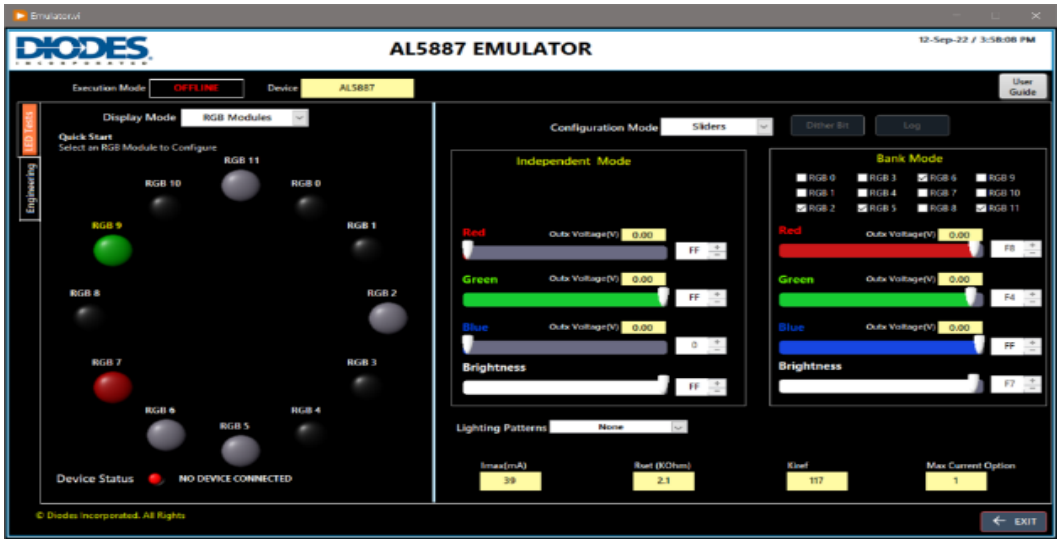


Figure 19: RGB Modules

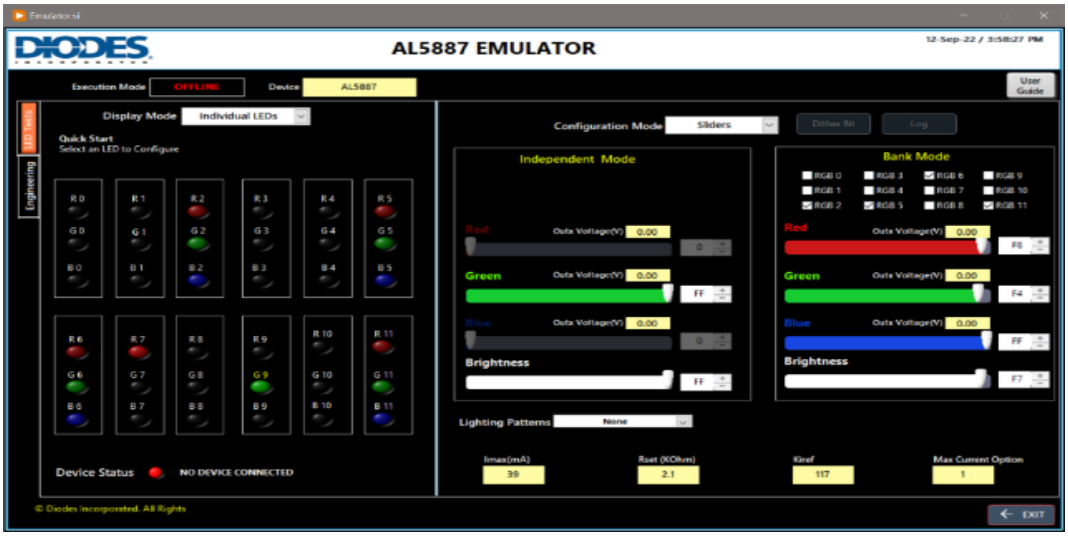


Figure20: *Individual LEDs*

- Configure the LEDs to adjust the Color and Intensity in 2 different modes
    1. Sliders (Independent Mode and Bank Mode).
    2. Registers.
- Display following Lighting Patterns in RGB Modules Annunciation Mode
    1. Breathing effect

2. Mono color chasing effect
3. Dual color chasing effect and
4. Multi-color chasing effect.

- Engineering Tab to guide user on current settings and to Compute the OutX Voltage.

19 Execution Instructions – Configuration Screen

1. Click the **Emulator_SW_V2.7.exe** to run the Emulator Application.
2. In the Configuration Screen, Select **AL5887** under Device Selection.
3. If AL5887 chip and test hardware are available, select Mode as **Online.** Else, if hardware is not available, then skip to step 6.
4. Select the **Interface** (I2C/SPI), **Address**, **Mode** based on the hardware in use.
5. Once Configuration is complete, Select **VALIDATE** Button.

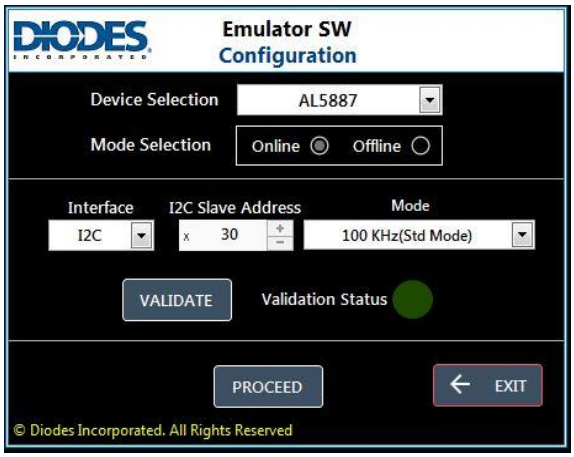


Figure 21 : Emulator Configuration Screen

Emulator SW verifies specific bits in the AL5887 Device Register.

- If values match, SW displays the validation status as PASS and navigates to the Emulator Screen.
- If values do not match, SW displays the validation Status as FAIL and alerts user with a message as given below.

Note: If Validation Failed, user shall choose among one of the below options

- User shall select MODIFY in the pop-up and re-visit step 4.
- User shall select PROCEED to continue using Emulator SW in OFFLINE Mode.
- User shall select ABORT to Quit the Application & to verify the Hardware.

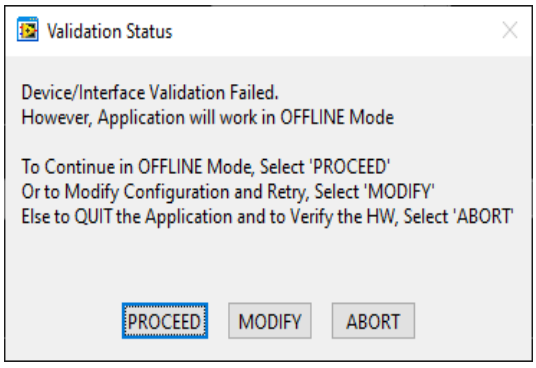


Figure 22: Validation status

6. If hardware is not available and if user prefers to use the SW in Offline Mode, select Mode as **OFFLINE.**

Figure 23: **Offline Mode Selection**

7. Select PROCEED button to navigate/access the **Emulator Screen.**

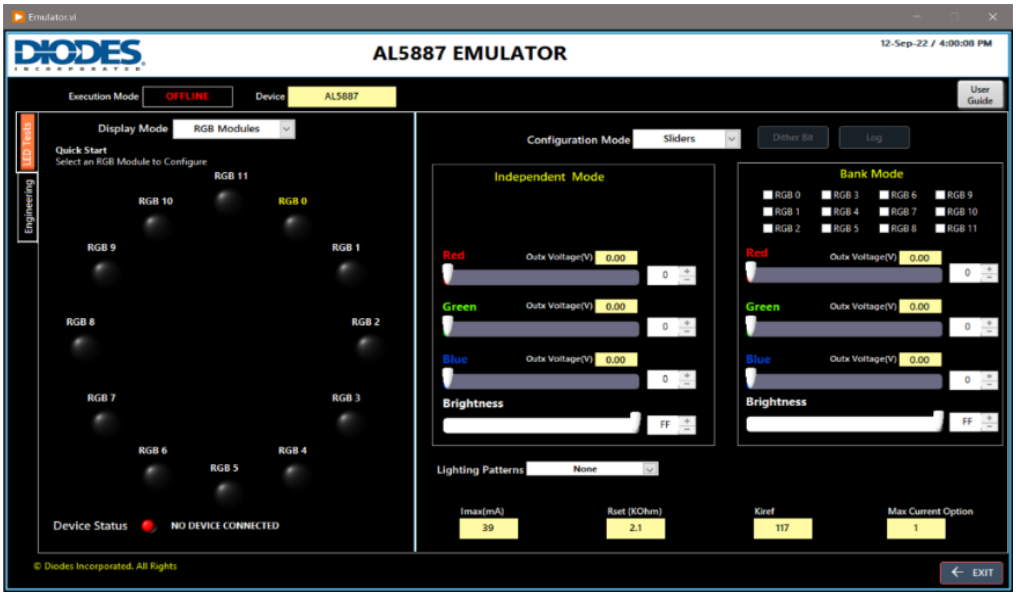


Figure 24: **Emulator Screen in Offline Mode**

20 LED Tests

20.1 Configuring LEDs in RGB Modules & Independent Mode

1. Select Display Mode as RGB Modules.

2. Select an RGB Module (Example: RGB8) and Adjust the Red/Green/Blue Sliders in Independent Mode to make adjustments to the colour and Intensity

Figure 25: **Configuring LEDs in RGB Modules & Independent Mode**

3. Each RGB module can be configured to different Red/Blue/Green settings in independent Mode.
   For instance, RGB1 to RED, RGB2 to GREEN and RGB8 to combination of RED+GREEN+BLUE and so on.

20.2 LED Tests – Configuring LEDs in RGB Modules & Bank Mode

1. Select Display Mode as **RGB Modules.**
2. Enable One/More Banks to enable RGBs in **Bank Mode.**
3. Move the Red/Green/Blue Slides under Bank Mode to adjust the colour and Intensity of the RGB Modules selected in Bank Mode.

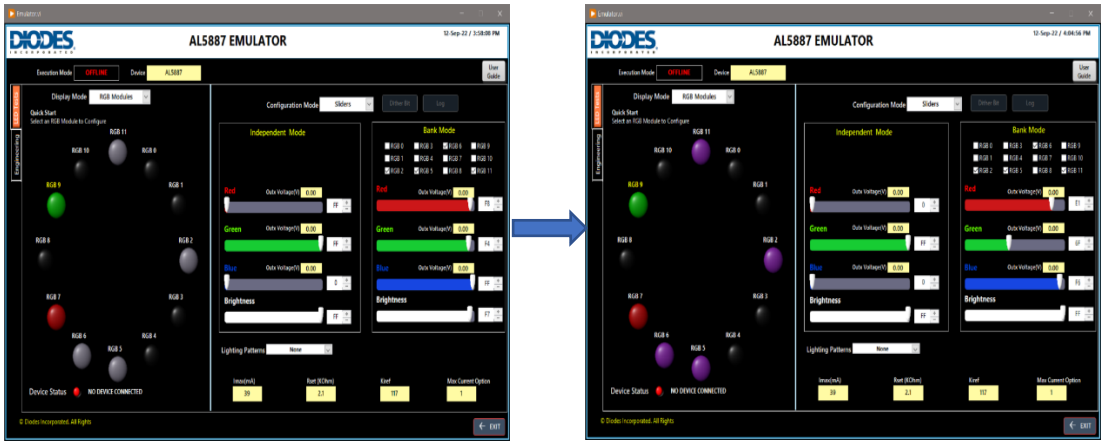


Figure 26: **Configuring LEDs in RGB Modules & Bank Mode**

All RGBs selected in Bank Mode can be adjusted only to same color adjustments.

20.3 LED Tests – Configuring LEDs in Individual LEDs & Independent Mode

1. Select Display Mode as **Individual LEDs.**
2. Select an **Individual LED** (Example G2). Only Green Slider will be enabled in

Independent mode while Red and Blue will be greyed Out.

3. Adjust the Green Slider to vary the colour and Intensity of G2 LED.

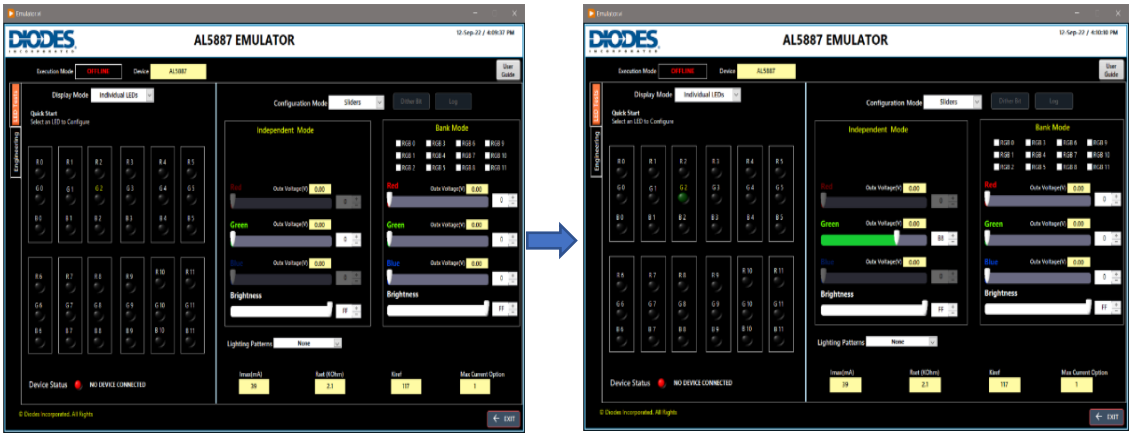


Figure 27: **Configuring LEDs in Individual LEDs & Independent Mode**

Each LED can be configured to different Red/Blue/Green settings in **independent Mode.**

20.4 LED Tests – Configuring LEDs in Individual LEDs & Bank Mode

1. Select Display Mode as **Individual LEDs.**
2. Enable One/More Banks to enable RGBs in **Bank Mode.**
3. Move the Red/Green/Blue Slides under Bank Mode to adjust the colour and Intensity of the RGB Modules selected in Bank Mode.

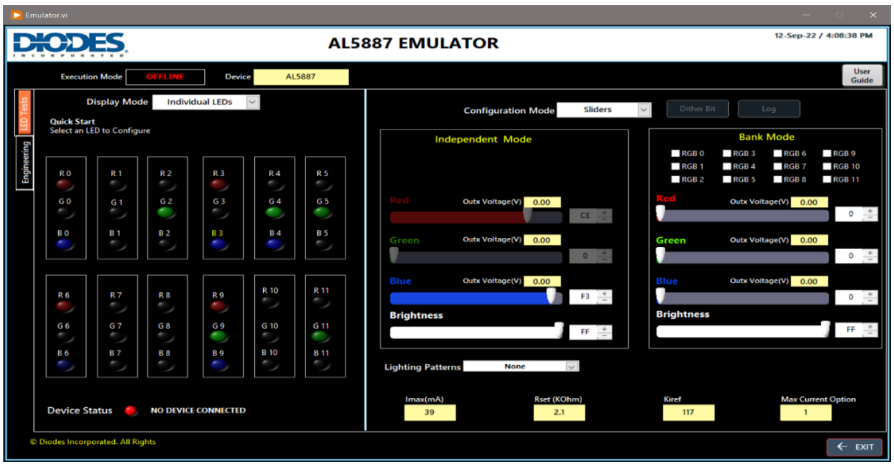


Figure 28: **Configuring LEDs in Individual LEDs & Bank Mode**

All LEDs selected in Bank Mode can be adjusted only to same color adjustments.

20.5 LED Tests – Notes on Display Mode

1. Adjustments made to LEDs in RGB Modules Mode will be reflected in Individual LEDs mode and vice versa.

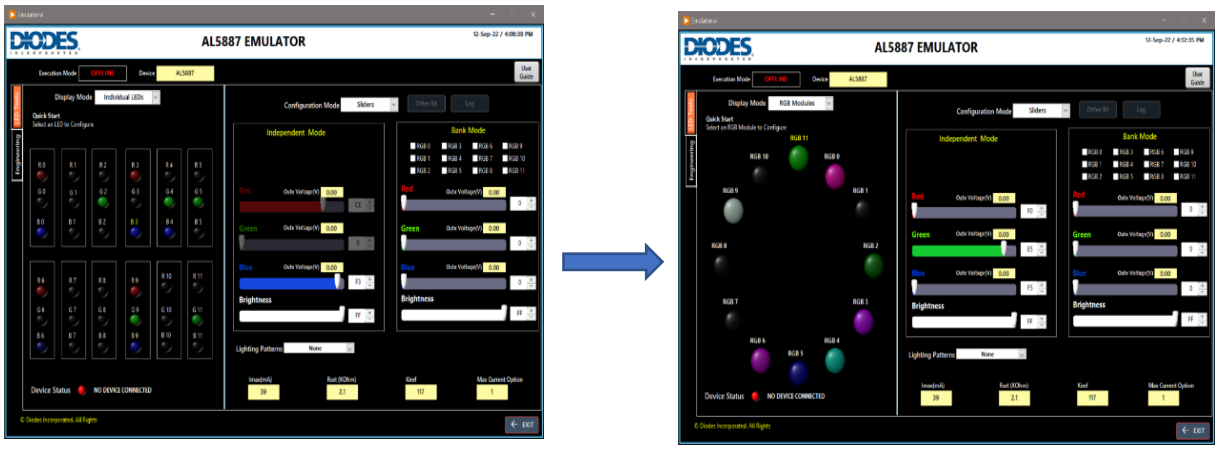


Figure 29: **LED Changes in Individual LEDs & RGB Modules**

20.6 LED Tests – Lighting Patterns

Emulator SW displays following Lighting Patterns in RGB Modules Annunciation Mode

- Breathing effect – Displays LEDs in Red color with a breathing effect.
- Mono color chasing effect – Displays LEDs in one color that cycles from RGB0-RGB11.
- Dual color chasing effect – Displays LEDs in two colors that cycles from RGB0-RGB11.
- Multi-color chasing effect – Displays LEDs in multiple colors that cycles from RGB0-RGB11.
- None – LED lighting Pattern is disabled.

Except in breathing effect, the colors keep changing during each cycle.
Select a lighting pattern from the drop down to simulate the pattern.

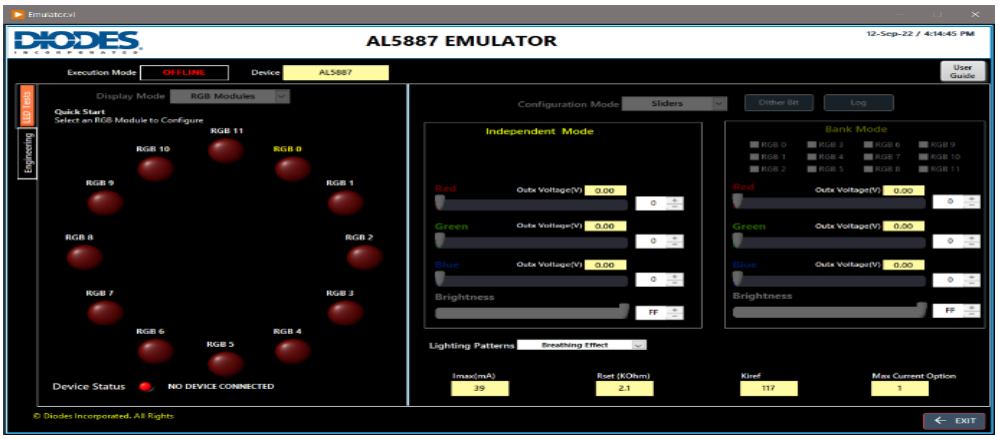


Figure 30: **Breathing Effect**

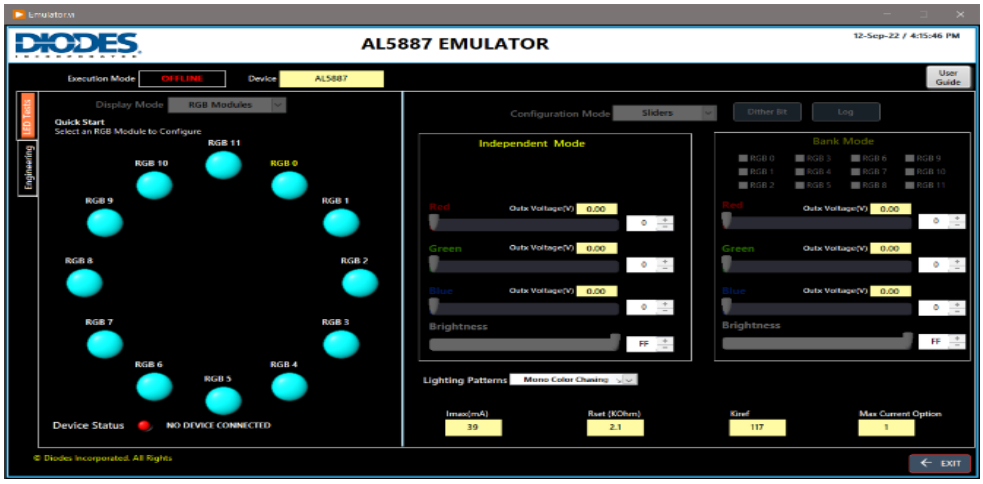


Figure 316 : Mono Colour Chasing

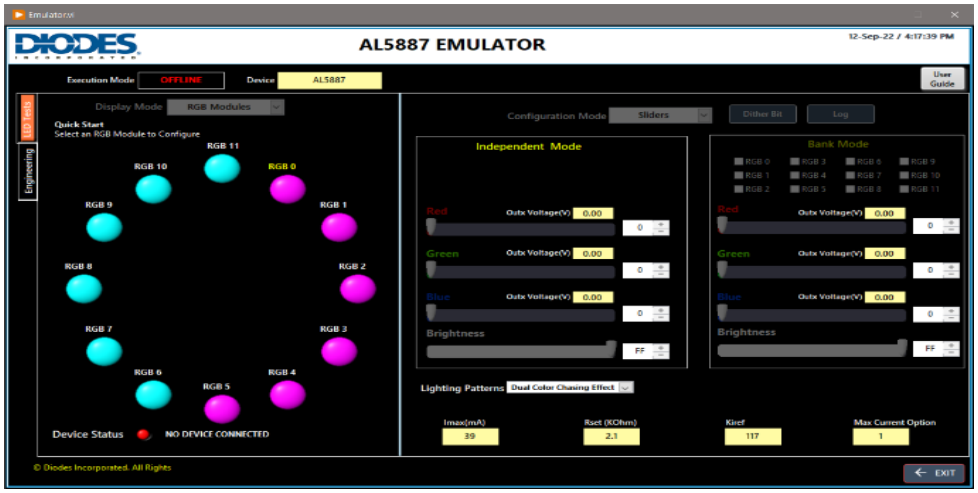


Figure 32: Dual Colour Chasing

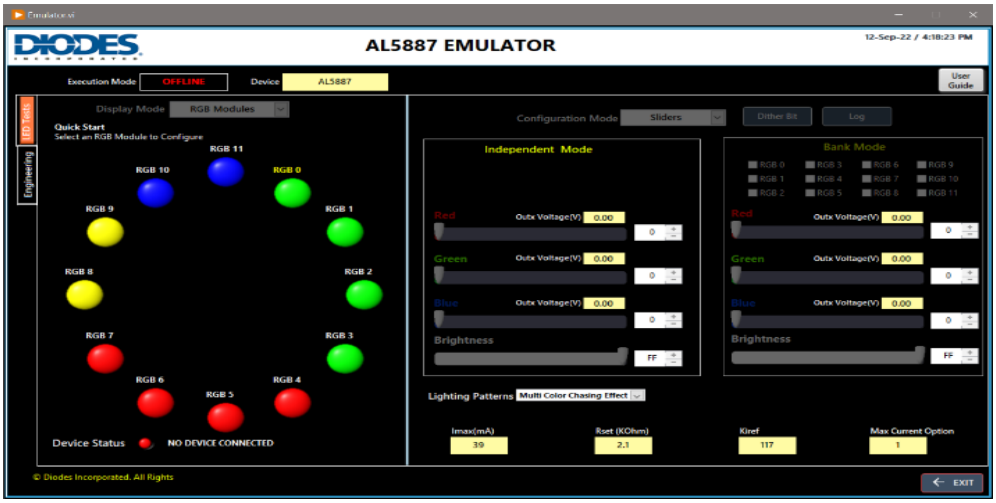


Figure 33: Multi-Colour Chasing

20.7 LED Tests – Configuring LEDs using Register Configuration Mode:

1. Select Configuration Mode as **Registers.**
2. Modify the Register Value(s) in the Table either bits D0-D7 or Hex data.

(For Instance: LED_CONFIG1 (0x03). This register enables/disables BANK mode of RGB8 to RGB11. Let us here enable RGB10 and RGB11).

3. When values of a register are modified, the applicable Register(s) will be highlighted in Blue colour indicating the value change. But these values are yet to be applied to the device and GUI.

LED_CONFIG1	0x3					1	1	0	0	C

4. Select **WRITE REGISTRY** to apply the values to the AL5887 and GUI. Once applied, the highlight is disabled.

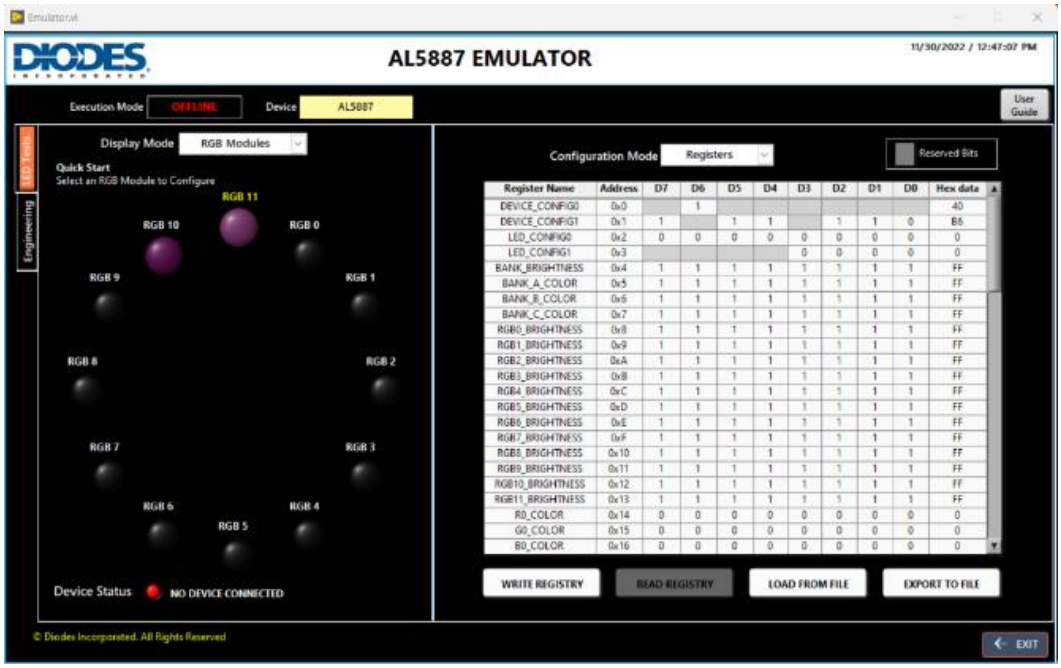


Figure 34: **Register Configuration Mode**

5.

6.

Note:

- Reserved Bits are greyed out in the Register Table.
- Please refer to datasheet for register and bit details.

20.8 LED Tests – Other Features in Register Configuration Mode:

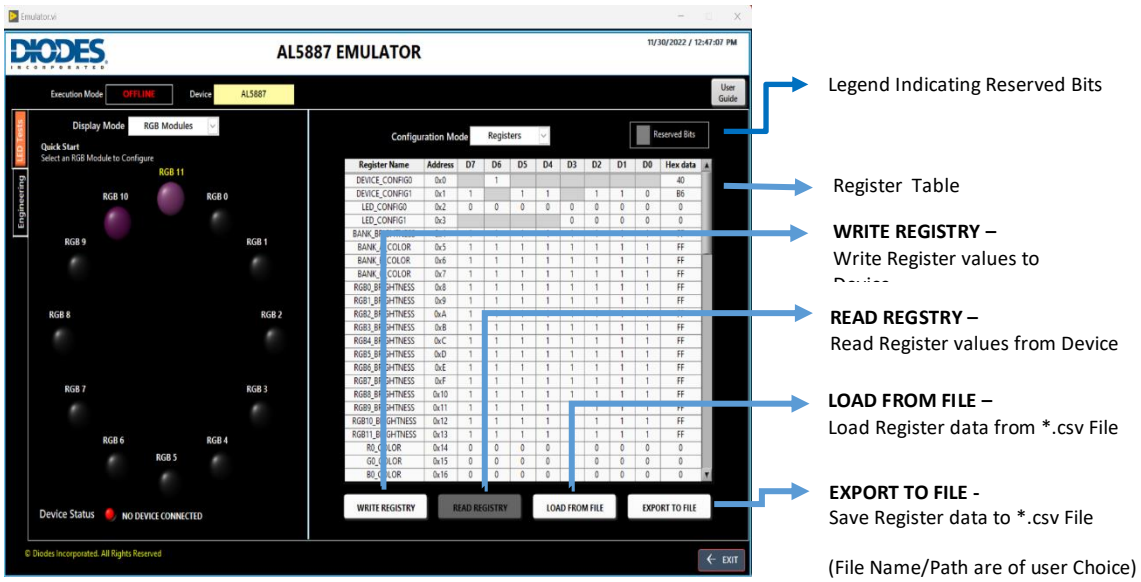


Figure 35: **Register Configuration Mode Features**

21 Engineering Tab –

21.1 Features:

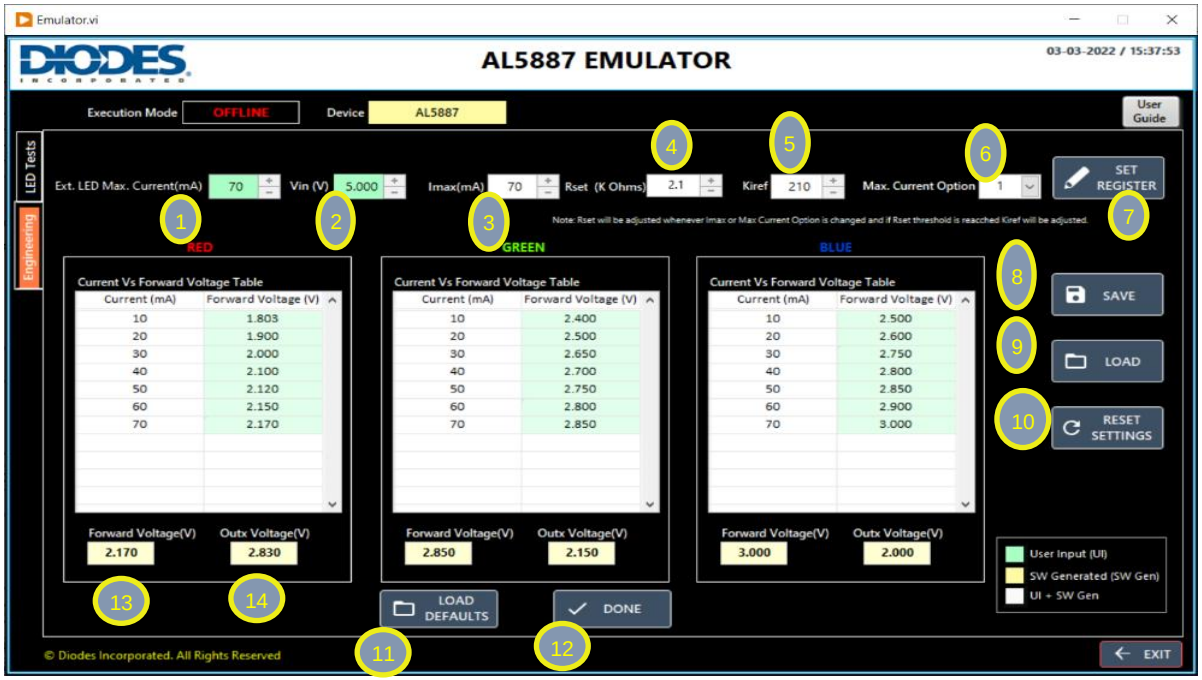


Figure 36: **Engineering Tab Features**

Engineering Tab – Serves as a guide in computing the Current configuration and Outx Voltage.

1. **Ext.LED Max Current(mA)**. Recommended Value <= 70 mA
   SW Alerts user when value entered is > 70 mA.
2. **Input Voltage** Vin(V) – Default is 5 V.
3. **Imax Value** – The channel LED current amplitude when PWM control is turned on.
4. **External Dimming Resistor**/Rset. (2.1K to 36K discrete values based on standard resistor chart).
5. **Kiref - Current Multiplication Factor**.
   Default value is 117. Acceptable values 21-210.
6. **Max. Current Option** – Either 0 (3/4th of -) or 1.
7. **SET REGISTER**– Write/Update Global Dimming Register and Max Current Option.
8. **SAVE** – Save Configuration to *.ini File of User Choice.
9. **LOAD** – Load Configuration from *.ini File of User Choice.
10. **RESET SETTINGS** – Resets all fields to initial condition .
11. **LOAD DEFAULTS** – Updates Forward Voltage Value as 1 in Current Vs Forward Voltage Table.
12. **DONE** – Enables User Input and SW Generated parameters.
13. **Forward Voltage** – Computed based on interpolation from Current Vs Forward Voltage table defined by the use.
14. **Outx Voltage** – computed as Vin – Forward Voltage.
    This is auto-updated by SW whenever Imax is adjusted .

Formula to calculate current settings : IMAX = KIREF * VIREF /RSET * [(Max_Current_Option/4)+(3/4)] .

21.2 Engineering Tab – Using Default File to populate the parameters:

1. Select **Engineering** Tab.
   Note : Ext LED Max.Current(mA) and LOAD Button will be enabled by default.
2. Select **LOAD** button and choose ‘default.ini’ file.

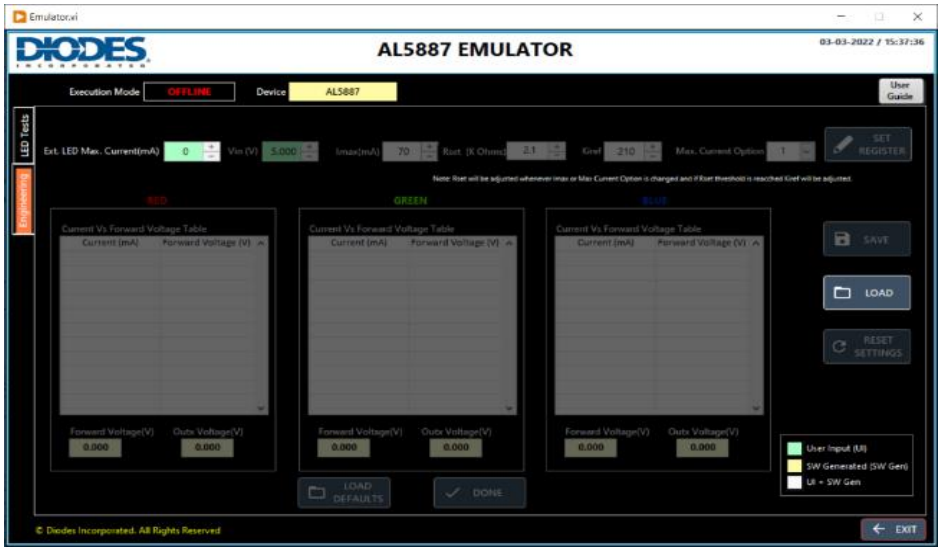


Figure 37: **Engineering Tab - default screen**

3. Emulator will display a pop-up indicating ‘Configuration Uploaded’.
4. All the fields will be auto-populated by the Emulator as shown below.

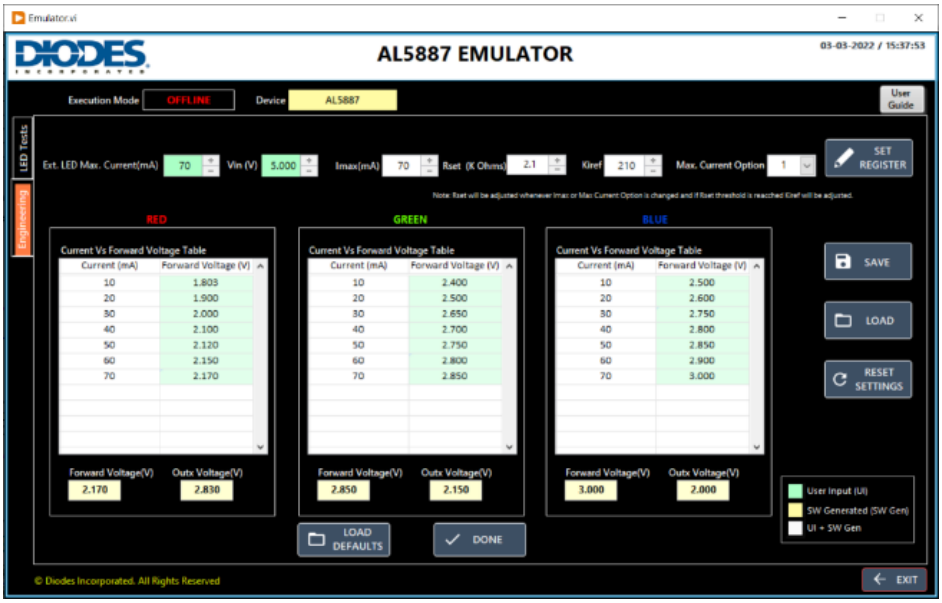


Figure 38: **Engineering Tab - default values loaded**

Note:

- By default, once file is loaded,

  Imax, Rset, Kiref, Max Current are set to 70, 2.1, 210, 1 respectively. Also, Forward Voltage and Outx Voltages are calculated for current Imax values.

- Now, User shall make modifications to the current parameters to check how the changes impact other associated parameters.
- SET REGISTER will update Global Dimming Bits (Kiref) and Max Current Option.
- Register Table in LED Tests

  a) Brightness of selected LEDs in LED Tests and

  b) AL5887 HW if connected in Online Mode.

21.3 Engineering Tab – Instructions to compute Outx Voltage:

1. Select **Engineering** Tab.

   Note : EXT LED Max.Current (mA) and LOAD Button will be enabled by default.

2. Enter the **EXT LED Max.Current** Value (To be updated based on datasheet and is recommended to be less than 70 mA).

When entered, multiple fields are enabled automatically as in the screenshot below.

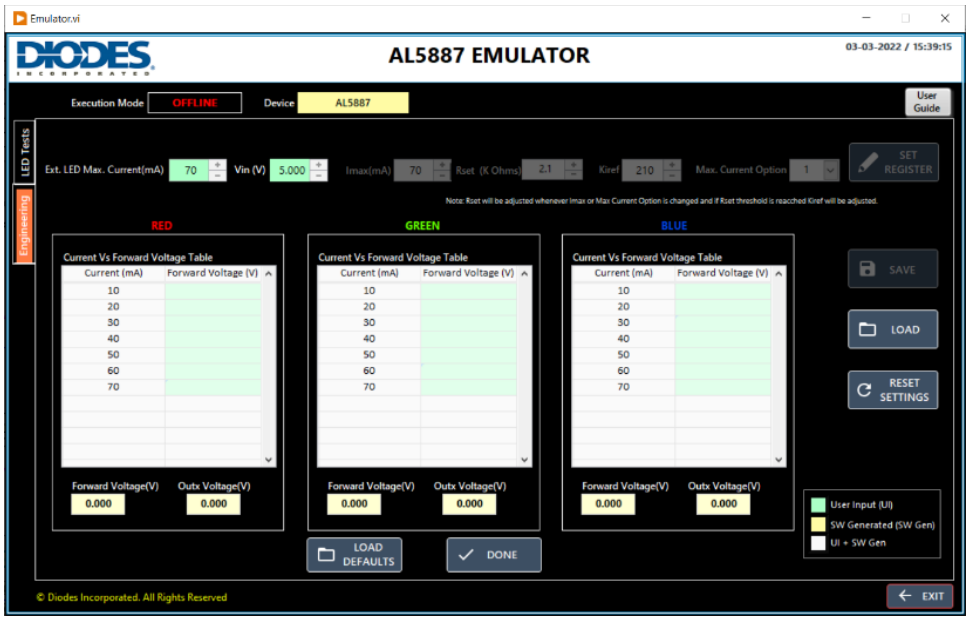


Figure 39: **Engineering Tab, LED Current Updated**

3. Update Vin if the values used are different from the default 5V.

4. Manually Enter the Forward Voltage(s) for each current Value based on real LED I-V characteristics for RED, GREEN and BLUE Table.

Note:

- **Load defaults** button auto-populates the blank table fields with a default value 1.0V.
- **Reset settings** will reset all the fields to Step #1

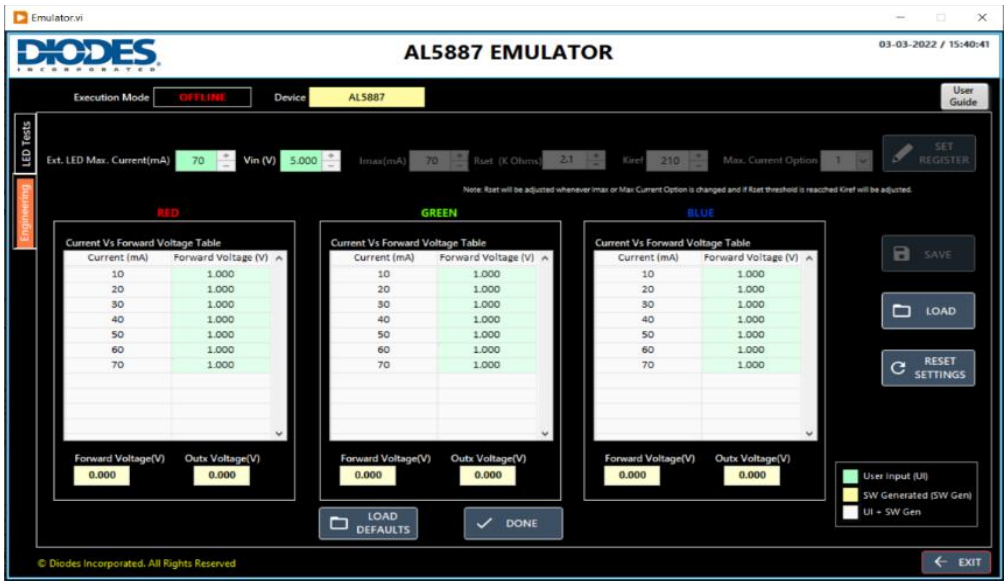


Figure 40: **Engineering Tab, default Forward voltages loaded**

5. Once the Current Vs Forward Voltage Table values have been entered, select **DONE** button.

Note: Current Configuration Parameters will be disabled until DONE is selected.

6. All the fields will be enabled now.

Figure 41: **Engineering Tab, all the fields enabled**

Note:

- Imax will be set to a value equivalent EXT LED Max.Current Max or to a max 70 mA by default. Rset and Kiref will be adjusted based on the Imax Value. Max. Current Option will be set to 0 by default.
- Now, User shall make modifications to the current parameters to check how the changes impact other associated parameters.
    - SET REGISTER will update Global Dimming Bits (K_{iref}) and Max Current Option in

    a) Register Table in LED Tests

    b) Brightness of selected LEDs in LED Tests and

    c) AL5887 HW if connected in Online Mode.

7. Select **SAVE** and provide a new file name to save the current configuration for future use.

22 Micro Controller setup – Using the Arduino board as an example

The AL5887 Demo board can easily be connected to a µC of your choice. Below is an example of an Arduino setup.

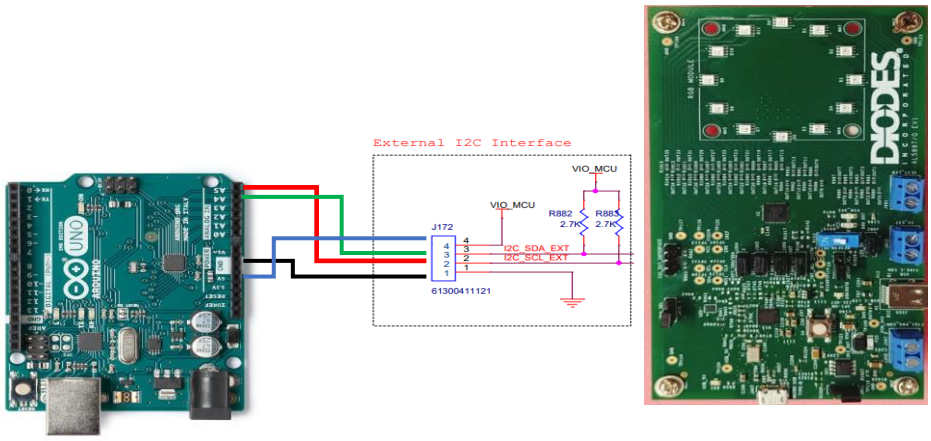


Figure 42: Wiring diagram from Arduino board to Demo board through external I2C interface

Signal	Arduino UNO R3	AL5887 DEMO
SCL	A5	J172 pin 2
SDA	A4	J172 pin 3
VIO	5V	J172 pin 4
GND	GND	J172 pin 1

Table 7: Arduino I2C connections

22.3 Example code:

```
/*    Author: Diodes INC   */
/*      Date: 4/1/2023      */
/* Company: Diodes Incorporated */
/*******************************************************************************/
#include<Wire.h>
#define I2C_Addr 0x30 // I2C address
// main function that allows us to communicate with the chip
void writeByte(uint8_t deviceAddress, uint8_t registerAddress, uint8_t registerData) {
  Wire.beginTransmission(deviceAddress);  // sends device address and starts communication
  Wire.write(registerAddress);          // sends register address
  Wire.write(registerData);             // sends register data
  Wire.endTransmission();               // stops communication
}
// put your setup code here, to run once:
void setup() {
  Wire.begin();
  Wire.setClock(200000); // set I2C to run at 200kHz
```

```
  initialize();
}

// put your main code here, to run repeatedly:
void loop() {
  mode1(); // change this to whatever mode is desired
}
void initialize() { // setup the board
  writeByte(I2C_Addr, 0x00, 0x40); // write a 1 to CHIP_EN
  writeByte(I2C_Addr, 0x38, 0xFF); // write a 1 to CHIP_EN
  writeByte(I2C_Addr, 0x00, 0x40); // write a 1 to CHIP_EN
  for (uint8_t i = 0x08; i <= 0x13; i++) { // start at first brightness register and go to the last
    writeByte(I2C_Addr, i, 0x80); // write all brightness to half
  }
  for (uint8_t i = 0x14; i <= 0x37; i++) { // start at first color register and go to the last
    writeByte(I2C_Addr, i, 0x00); // write all color to 0
  }
}
/*****************************************************************************/
/* Spin and Fade */
void mode1() {
  for (uint8_t i = 0; i < 12 ; i++) {
    writeByte(I2C_Addr, 0x14 + i * 3, 0x80); // write half color to red leds
    delay(100); // add a 0.1s delay before turning on the next LED
  }
  for (uint8_t j = 0; j < 12; j++) {
    writeByte(I2C_Addr, 0x16 + j * 3, 0x80); // write half color to blue leds
    delay(100);
  }
  for (uint8_t k = 0; k < 12; k++) {
    writeByte(I2C_Addr, 0x15 + k * 3, 0x80); // write half color to green leds
    delay(100);
  }
  for (uint8_t m = 1; m <= 16; m++) {
    uint8_t brightness = 128 - m * 8; // start decreasing the brightness
    for (uint8_t n = 0x08; n <= 0x13; n++) {
      writeByte(I2C_Addr, n, brightness);// write updated brightness to brightness registers
    }
    delay(100);
  }
  for (uint8_t p = 0x14; p <= 0x37; p++) {
    writeByte(I2C_Addr, p, 0x00);
  }
  for (uint8_t q = 0x08; q <= 0x13; q++) {
    writeByte(I2C_Addr, q, 0x80);
  }
  delay(500);
```

```
}
```