Table of Contents

Abstract

This user manual describes the functionality and characteristics of the AL5887/Q RGB LED driver using the 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 diagrams, bill of materials, printed circuit board layout drawings, and demo board images.

Introduction

This demo board characterizes the features of AL5887/Q RGB LED driver. The main goal is to exercise vivid LED effects by communicating through I2C/SPI.

Demo Board Features

1. This demo board has an additional feature of providing supply to the LEDs by using a power bank connected through the USB Type-C (J14) connector, and supply voltage to the LED driver through the micro USB connector (J5).
2. The user can manipulate on-board LEDs or connect their own LEDs using connector J15.
3. The user can communicate with theLED driver through the external I2C/SPI interface connector J2.

Figure 1: Demo Board

General Description

The demo board consists of the following major components:

AL5887/Q RGB LED Driver

The AL5887/Q 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 an internal 12-bit PWM for color and brightness control through an SPI or I2C digital interface. This is ideal for lighting applications with up to 12 RGB LED modules and 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/Q are controlled via a programmable SPI/I2C digital interface. The device supports 400Khz I2C and 2MHz SPI interfaces. Using a dedicated INT_SEL pin, SPI/I2C protocol can be selected. The AL5887 has a 30kHz, 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. The device can also connect up to 4 devices using two external hardware address pins, and features ultra-low quiescent current with four modes of operation (shutdown, standby, normal, and power save mode).

FT4222 USB to I2C/SPI Interface

The FT4222H is a high-speed USB to Quad-SPI/I 2C interface device controller. This requires an external crystal (12MHz) for the internal PLL to operate. This contains SPI/ I2C configurable interfaces. The SPI interface can be configured in master mode with a single-, dual-, or quad-bit data widths for transfer, or in slave mode with a single-bit data width. The I2C interface can be configured in master or slave mode.

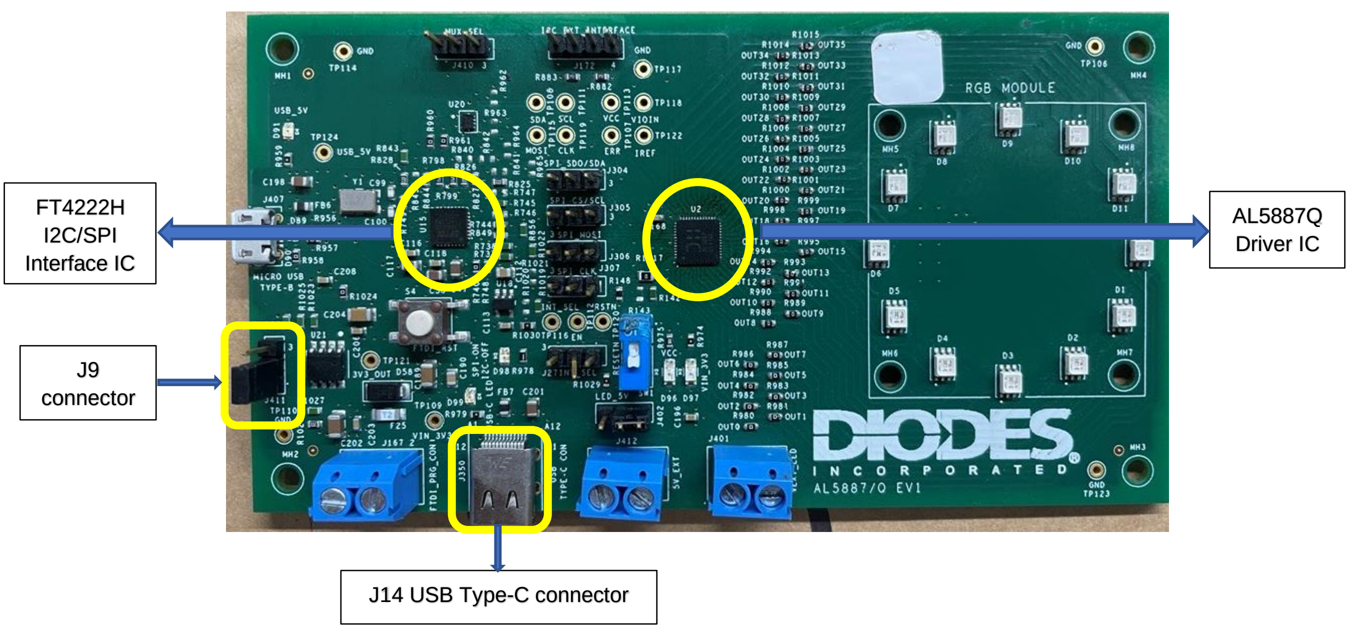
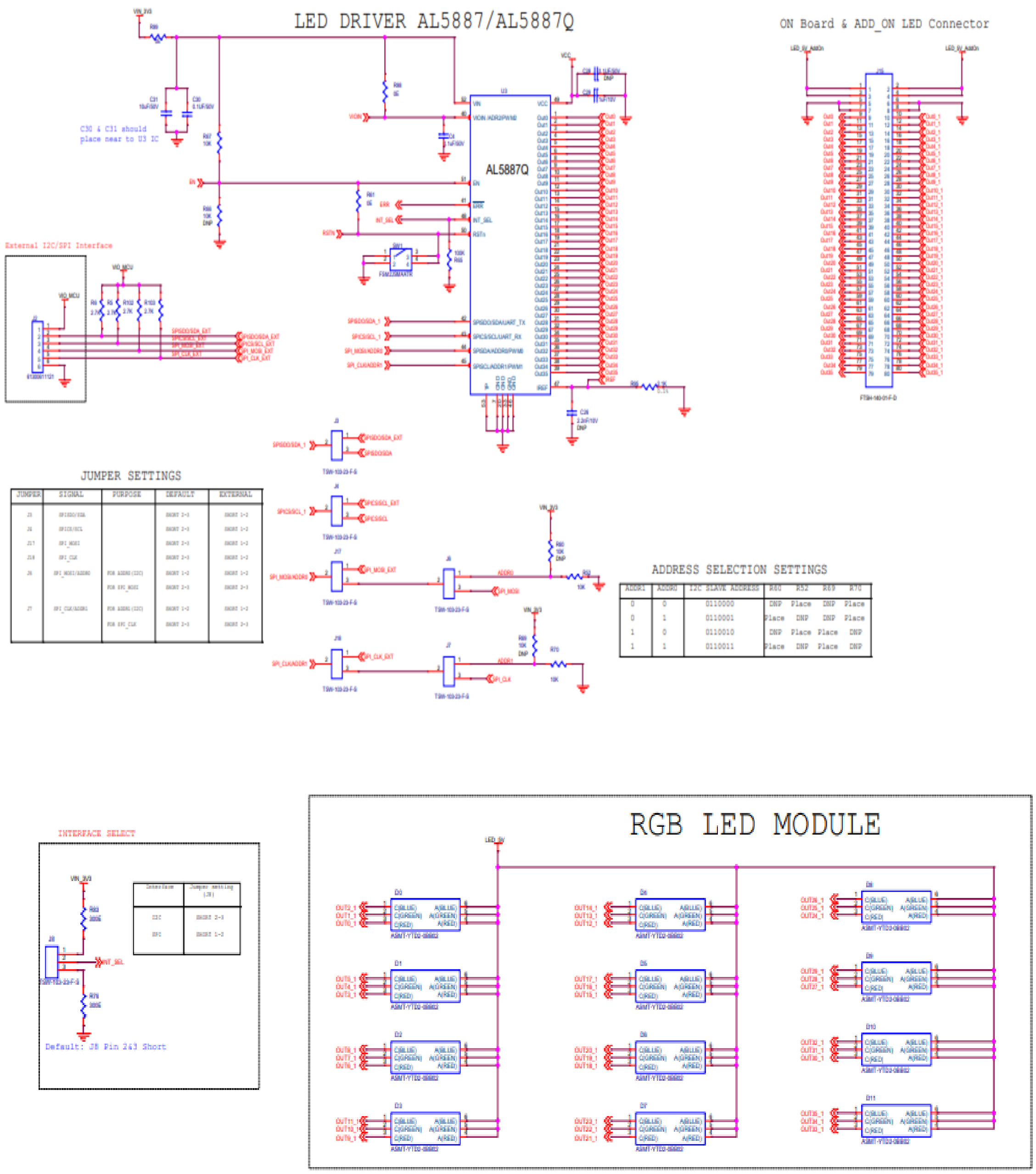


Figure 2: Demo Board with AL5887Q IC and FT4222H IC

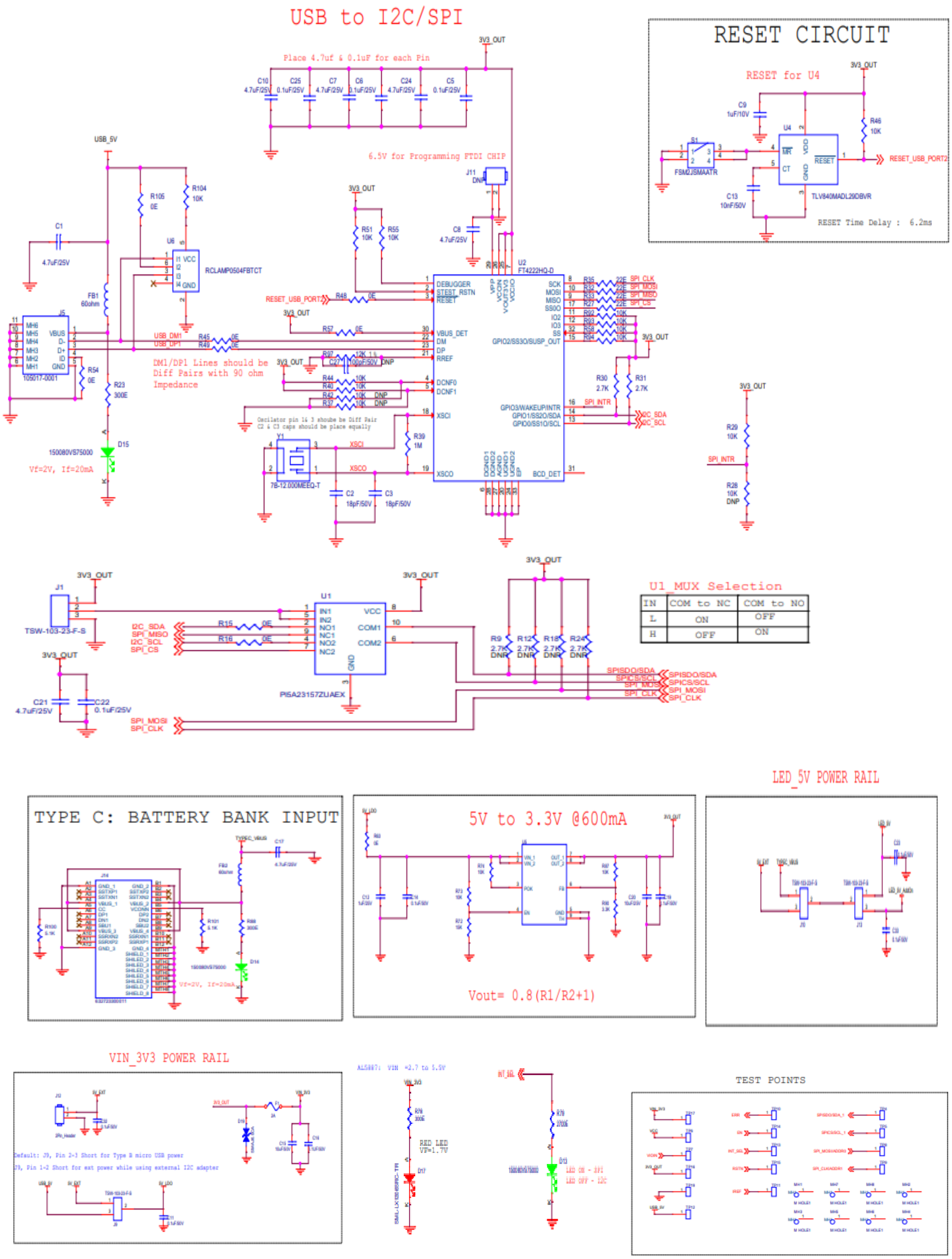
The demo board can be powered up through a USB to micro USB Type-B connector (USB_VBUS) connected at the J5 connector and short 2-3 pins of the J9 connector. The LED can be powered up by conneting the battery bank through a USB Type-C connector (J14) and short 2-3 pins of the J9 connector. For this setup, please refer to the jumper settings below in Table 3.

Schematics

The schematic of the AL5887Q RGB LED driver demo board is shown below.



Schematics (continued)



Bill of Materials

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

Item	Reference Designator	Part Number	Manufacturer Part Number
1	C1, C7, C8, C10, C17, C21, C24	4.7µF/25V	CGA4J1X7R1E475M125AE
2	C2, C3	18pF/50V	06035A180JAT4A
3	C4, C11, C14, C18, C19, C23, C32	0.1µF/50V	CL10B104MB8NNNC
4	C5, C6, C22, C25	0.1µF/25V	CL10B104MB8NNNC
5	C9	1µF/10V	GRM21BR71A105KA01L
6	C12	1µF/25V	C1206C105M3RAC7800
7	C13	10nF/50V	06035C103KAT4A
8	C15, C31	10µF/50V	GMC31X7R106K50NT
9	C16, C30	0.1µF/50V	08055C104KAT4A
10	C20	10µF/25V	CL31B106KAHVPNE
11	C26	2.2nF/10V	885012207009
12	C27	100pF/50V	CL21C101JBANNNC
13	C28	0.1µF/50V	08055C104KAT4A
14	C29	1µF/10V	C0805C105K8RACTU
15	D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D0	ASMT-YTD2-0BB02	ASMT-YTD2-0BB02
16	D12, D18	PESD5V0F1USF315	PESD5V0F1USF315
17	D13, D14, D15	150080VS75000	150080VS75000
18	D16, D17	SML-LX1206SRC-TR	SML-LX1206SRC-TR
19	D19	SMAJ6.0CA	SMAJ6.0CA
20	FB1, FB2	60Ω	BLM21PG600SH1D
21	F1	2A	0685T2000-01
22	J1, J3, J4, J6, J7, J8, J9, J10	TSW-103-23-F-S	TSW-103-23-F-S
23	J2	61300411121	61300411121
24	J5	Micro USB Type B	105017-0001
25	J11, J12, J13	Terminal Block 2P	OSTTC022162
26	J14	USB Type C	632723300011
27	MH1, MH2, MH3, MH4, MH5, MH6, MH7, MH8	M HOLE1	-
28	R1, R2, R3, R4, R7, R8, R10, R11, R13, R14, R15, R16, R17, R22, R25, R34, R36, R38, R41, R43, R45, R47, R48, R49, R50, R53, R54, R56, R57, R59, R62, R63, R64, R68, R71, R75, R80, R81, R82, R84, R85, R86, R89, R91, R96	0E	RC0603JR-070RL
29	R5, R6, R9, R12, R18, R24, R30, R31	2.7K	RMCF0603FT2K70
31	R23, R76, R78, R83, R88	300E	ESR03EZPJ301
32	R27, R32, R33, R35	22E	CRGCQ0603F22R
33	R39	1M	RN73H1JTTD1004B25
34	R51, R94	10K	RC0603FR-7W10KL
35	R61, R98, R99	0E	RMCF0805ZT0R00
36	R65	100K	RCA0603100KFKEAHP
37	R72	15K	SG73S1JTTD1502F
38	R77	20R	RNCP0603FTD20R0
39	R79	2700E	ERJ-PB3D2701V
40	R90	3.3K	CRGH0603F3K3
41	R95	2.1K	RQ73C1J2K1BTD
42	R97	12K	RT0603DRD0712KL
43	R100, R101	5.1K	RMCF0603JT5K10
44	SW1	DS01-254-L-01BE	DS01-254-L-01BE
45	S1	FSM2JSMAATR	FSM2JSMAATR
46	TP1, TP2, TP3, TP18	TEST POINT	-
47	TP4, TP5, TP6, TP7, TP8, TP9, TP10, TP11, TP12, TP13, TP14, TP15, TP16, TP17, TP19	TEST POINT	-
48	U1	PI5A23157ZUAEX	PI5A23157ZUAEX
49	U2	FT4222HQ-D	FT4222HQ-D
50	U3	AL5887/Q	AL5887/Q
51	U4	TLV840MADL29DBVR	TLV840MADL29DBVR
52	U5	AP7165-SPG-13	AP7165-SPG-13
53	Y1	7B-12.000MEEQ-T	7B-12.000MEEQ-T

Table 1: BOM of AL5887/Q Demo Board.

Setup Requirements

The following contents are included with the demo board:

- Demo board user manual
- Emulator software
- Emulator software user manual

Software

The emulator is a standalone application developed using LabVIEW 2019 to support AL5887 LED driver testing. The AL5887's features are controlled via the SPI/I2C interface. Minimum system requirements for the emulator software are given below

Item Number	Description	Specification/Requirements
1.	OS	Windows 7/10
2.	RAM	4GB or above
3.	Disk space	250MB approx..

Table 2: Software Configuration Requirements

Please refer to the emulator software user manual for installing emulator software on your PC/laptop.

Hardware

- DC power supply with maximum 5V/5A current consumed when all LEDs at full brightness are approximately 1.5A
- PC/laptop with emulator software installed
- A USB to micro USB Type-B connecting cable from the PC/laptop to the demo board
- Arrange the setup as shown in the figure below

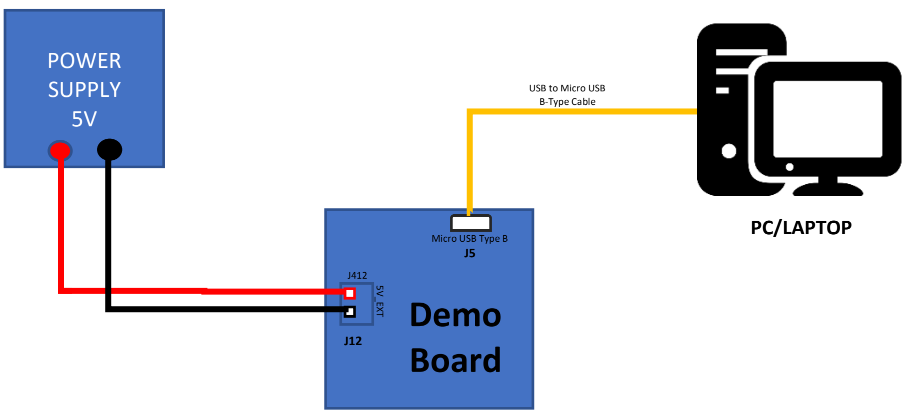


Figure 3: Basic hardware setup representation

Hardware Setup

Connections

1. Connect the V_{IN} (5V) at the J12 (5V_Ext) connector. Ensure the Vin is between 2.7V - 5.5V.
2. Place the jumpers as per the **default** jumper settings given in Table 1 below to communicate via I2C communication and power up.

CONNECTOR	PURPOSE	**DEFAULT CONNECTION** with I2C and External Supply to RGB LEDs
J1 J3	MUX SELECTION SPISDO/SDA	PIN 1-2 SHORT: I2C Pin 2-3: Short
J4	SPICS/SCL	Pin 2-3: Short
J6	ADDR0	Pin 1-2 Short: I2C Address (ADDR0)
J7 J8 J9	ADDR1 INTERFACE SELECT 5V_Ext / USB_5V	Pin 1-2 Short: I2C Address (ADDR1) Pin 2-3 Short: for I2C PIN 1-2 SHORT: 5V_Ext
J10	5V_Ext / Type-C_VBUS	PIN 2-3 SHORT: 5V_Ext
J13	On-board/add-on board LED supply	Pin 1-2 Short: for On board LED supply

Table 3: Default Jumper Settings

Hardware Setup (continued)

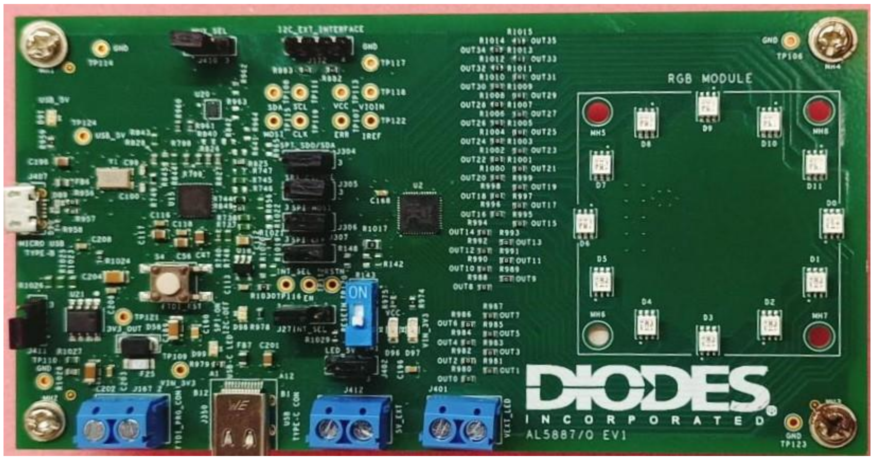


Figure 4: Demo board with all Connectors and Jumpers

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

ADDR1	ADDR0	I2C SLAVE ADDRESS	RESISTOR ARRANGEMENT		Remarks
			MOUNT	REMOVE	-
0	0	011 0000	R52, R70	R60, R69	**Default**
0	1	011 0001	R60, R70	R52, R69	-
1	0	011 0010	R52, R69	R60, R70	-
1	1	011 0011	R60, R69	R52, R70	-

Table 4: Address Selection Table

To communicate with the LED driver using an external I2C/SPI bus (not from GUI), please refer to the jumper settings in Table 5.

Check List Before Turning ON the Board

1. Ensure that the V_{IN} given at J12 is less than 5.5V to avoid device damage.
2. Ensure J9, J10, & J13 jumper settings are as per default settings (or as per requirements).

Turn ON Procedure

1. Turn on the external 5V power supply given at connector J12 and check for the power on indication at LED D17 (RED LED).
2. Open GUI on the PC/laptop and run the .exe file provided with the manual.
3. Connect USB to micro USB connector between the PC/laptop to the demo board connector (J5).
4. Operate LEDs from GUI using the controls given. Refer to emulator software manual provided with the manual.

Jumper Settings

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

CONNECTOR	PURPOSE	CONNECTION
J1	MUX_Selection	Pin 1-2 SHORT: For I2C
		Pin 2-3 SHORT: For SPI
J3	SPISDO/SDA_External	Pin 1-2 SHORT
	SPISDO/SDA_Internal	Pin 2-3 SHORT
J4	SPICS/SCL_External	Pin 1-2 SHORT
	SPICS/SCL_Internal	Pin 2-3 SHORT
J6	ADDR0	Pin 1-2 SHORT
	SPI_MOSI	Pin 2-3 SHORT
J7	ADDR1	Pin 1-2 SHORT
	SPI_CLK	Pin 2-3 SHORT
J8	Interface Select	Pin 1-2 SHORT: For SPI
		Pin 2-3 SHORT: For I2C
J9	Power Rail	Pin 1-2 SHORT: 5V_EXT
		Pin 2-3 SHORT: USB_5V
J10	LED Power Rail	Pin 1-2 SHORT: Type-C VBUS
J13	LED Supply	Pin 2-3 SHORT: 5V_Ext Pin 1-2 SHORT: Supply for on-board LEDS Pin 2-3 SHORT: Supply for add-on board LEDS

Table 5: Jumper Connection Table

Board Layout

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

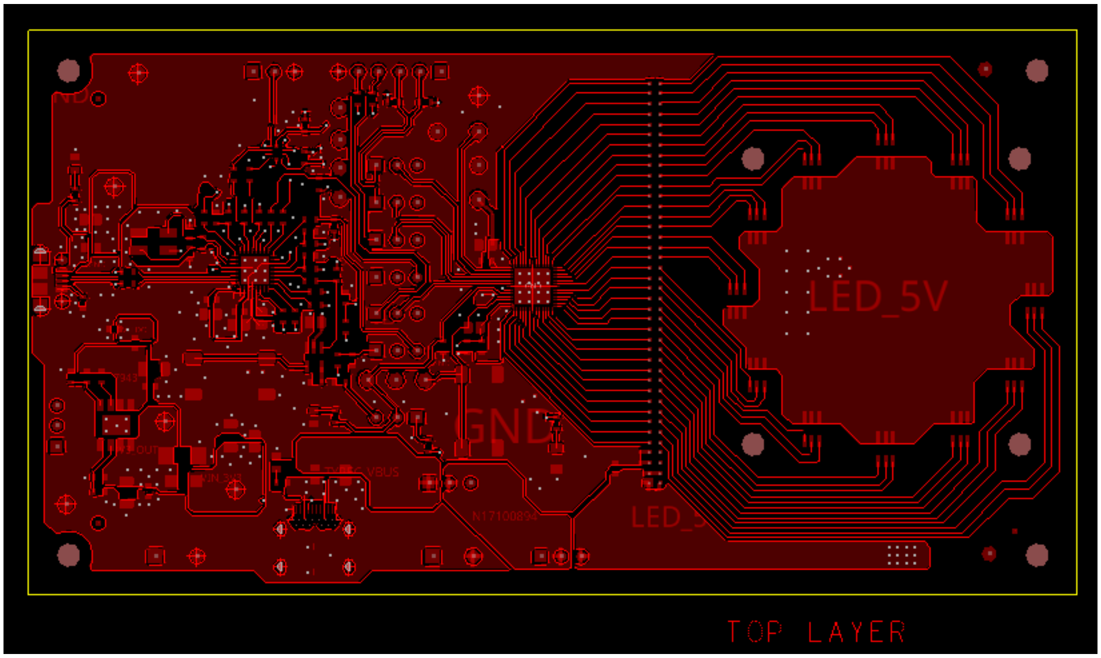


Figure 5: Top Layer

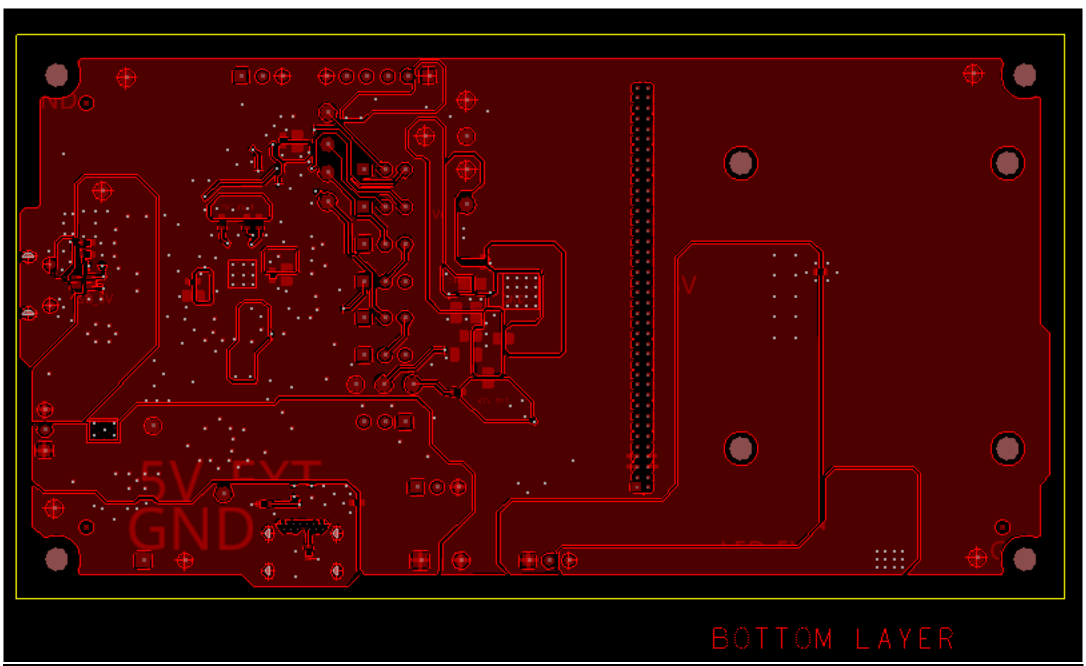


Figure 6: Bottom Layer

Board Images

The following images are the top and bottom views of the demo board with a few LED effects.

Figure 7: Top View

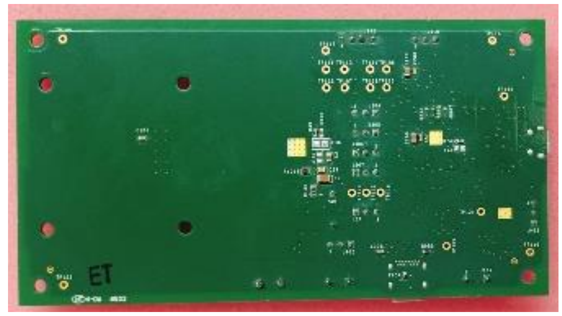


Figure 8: Bottom View

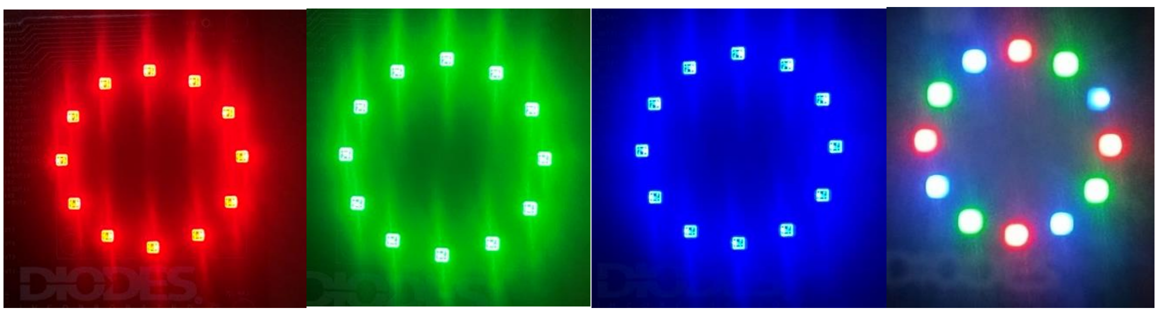


Figure 9: Shows various LED effects; a. Red, b. Green, c. Blue, d. RGB mixture

External I2C/SPI Interface Using Demo Board

Shown below is the demo board communicating with an external I2C/SPI interface.

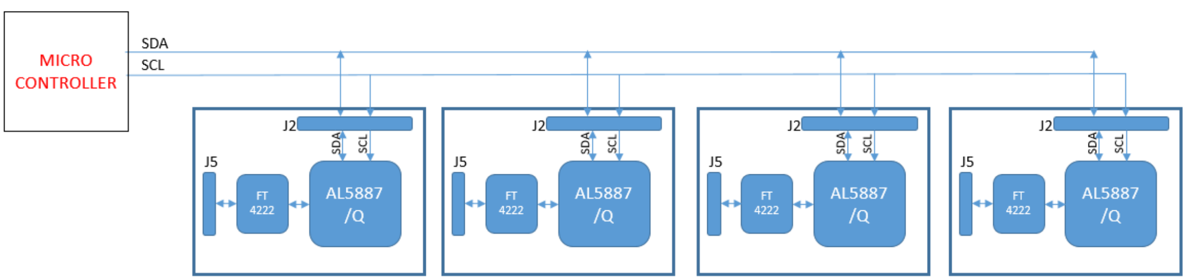


Figure 10: Micro Controller to Demo Board using I2C

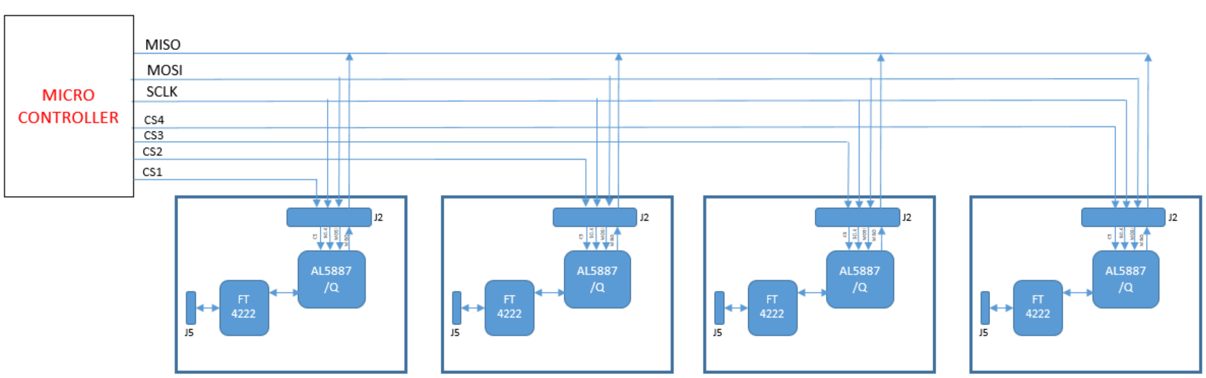


Figure 11: Micro Controller to Demo Board using SPI

External I2C/SPI Interface Using Demo Board (continued)

Step 1: Set the following jumper settings as per the requirements.

For I2C, ensure the address setting. Please refer to the address selection information in Table 4.

CONNECTOR	PURPOSE	CONNECTION
J8	Interface Select	Pin 1-2 Short: For SPI
		Pin 2-3 Short: For I2C
J3	SPISDO/SDA_External	Pin 1-2 Short
J4	SPICS/SCL_External	Pin 1-2 Short
J6	SPI_MOSI/ADDR0	Pin 1-2 Short: External SPI_MOSI Pin 2-3 Short: ADDR0
J7	SPI_CLK/ADDR1	Pin 1-2 Short: External SPI_CLK Pin 2-3 Short: ADDR1

Table 6: Jumper settings when Communicating through an External I2C/SPI Interface

Step 2: Set slave address. Please refer to Table 2 to set the address.

Step 3: Connect demo board to PC or micro controller as shown in the above figures using the external I2C/SPI interface connector J2. The connections of the J2 connector are as follows:

CONNECTOR	PIN	PURPOSE
J2	1	SPISDO/SDA_EXT
J2	2	SPICS/SCL_EXT
J2	3	SPI_MOSI_EXT
J2	4	SPI_CLK_EXT

Table 7: External I2C/SPI Interface Pin Configuration.

Troubleshoot

Current Limit

If the board is drawing excess current, remove the resistor R95, then check for the cause.

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

VIN_3V3 – 3.3V (TP 17)

EN – 3.3V (TP 14)

VCC – 1.8V (TP 6)

3V3_V_{OUT} – 3.3V (TP 16)

RSTn – 3.3V (TP 15)

V_{IOIN} – 3.3V (TP 7)

GND – 0V (TP 2)

ERR – (TP 10)

INT_SEL – 3.3V (TP 13)

IREF – 0.7V (TP 11)

To verify communication, reconnect the cable. Probe using an oscilloscope on the following test points.

SPISDO / SDA – TP 4

SPICS / SCL – TP 5

SPIMOSI – TP 8

SPICLK – TP 9

If the LEDs are not glowing, then check the LED supply and behavior using a 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, check if the corresponding channel is open or short.

Additional Information

Device Functional Modes

Normal mode

The AL5887/Q device enters the NORMAL mode when Chip_EN (register) = 1. ICC is 5 mA (typical). The voltage at IREF test point (TP11) should be 696mV – 704mV.

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.

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).

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).

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.

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), which is the current multiplication factor that can be programmed using the 6-bit global dimming register G5:G0 (Address = 66H). It is also the analog dimming register, where 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 the decimal equivalent of 100000, which is 32. Hence, KIREF = 21 + (32 * 3) = 117.

Using equation (1) above, with a global dimming register setting of 000000 and Max_Current_Option = 1, the below table 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 8: IMAX variation w.r.to Rset

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

R_{SET}(KΩ)	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 9: IMAX range using global dimming at different Rset values.

For more information on current settings, please refer to the datasheet.

Brightness and Color Register

The below table shows various values of brightness and color registers with 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 10: Brightness and Color Register Values.

Emulator Software

The emulator is a standalone application developed using LabVIEW 2019 to support AL5887 LED driver testing. The AL5887's features are controlled via the SPI/I2C interface.

Minimum system requirements for the 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 must be run as the administrator.

Installation instructions

1. Download the latest Emulator software package from the the Diodes server.
2. Unzip Emulator_Installer.zip.
3. The extracted folder will have the following files and folders.

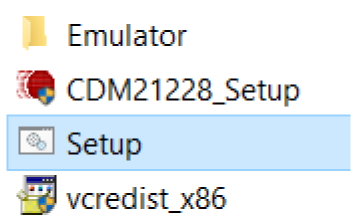


Figure 12: Emulator Installer folder

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

Figure 13: FTDI Driver installation

Emulator Software (continued)

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

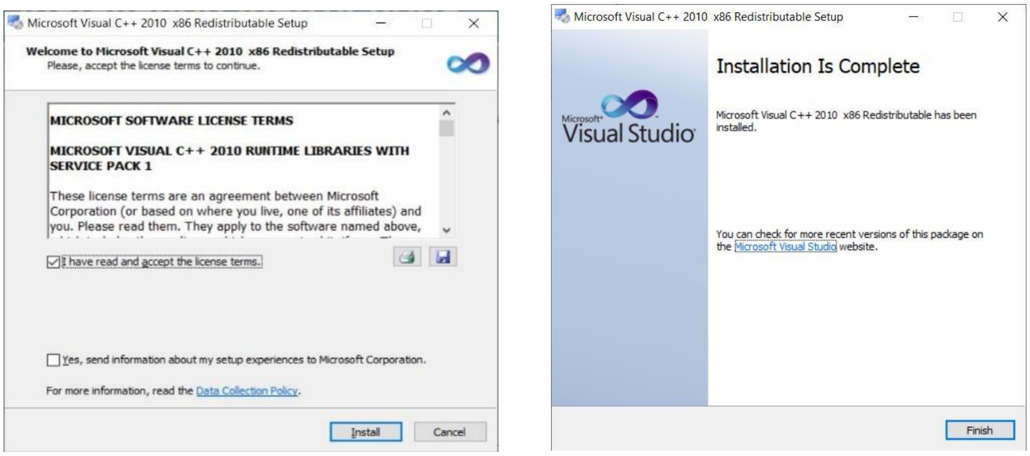


Figure 14: Microsoft Visual C++ 2010 x 86 Redistributable Installation

The emulator setup will be installed following Microsoft Visual C++ 2010 x86 Redistributable. Follow the on-screen instructions and complete the installation. Restart the system when prompted

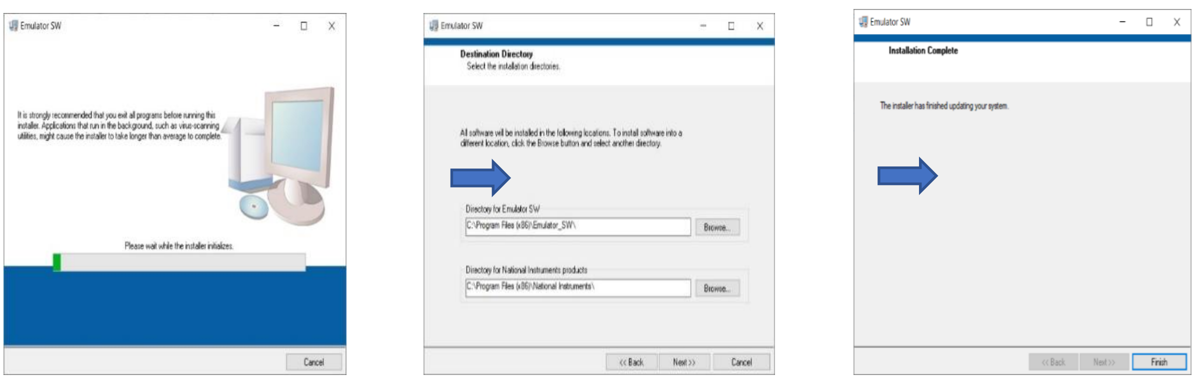


Figure 15: Microsoft Visual C++ 2010 x 86 Redistributable Installation

Getting started

Run the emulator software **Emulator_SW_V2.7.exe** (C:\Program Files (x86)\Emulator_SW\ Emulator_SW_V2.7.exe).

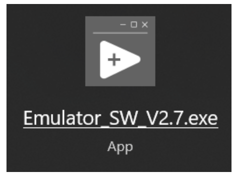


Figure 16: Emulator SW EXE Icon

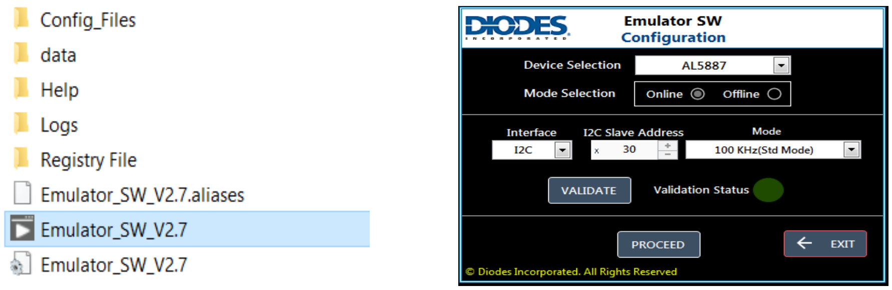


Figure 17: Emulator SW Execution

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

Software Features: Configuration Screen

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

The **Configuration Screen** allows users to:

1. Select the device and test/configure its interface parameters.
2. Perform device validation based on the device and interface configuration.
3. Mode selection to either online or offline mode.

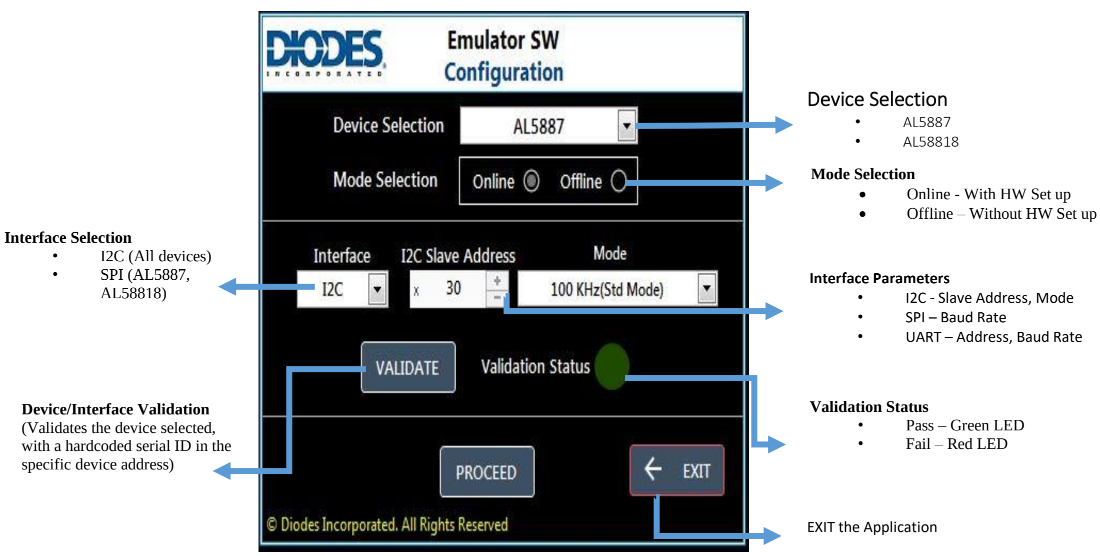


Figure 18: Emulator Configuration Screen

The **Emulator Screen** allows users to:

1. View the parameters configured in the CONFIGURATION Screen.
2. Use an LED Test Tab for support
3. View the Device Connection Status.
4. View the LED Annunciation in two different modes: RGB Modules and Individual LEDs.

Software Features: Configuration Screen (continued)

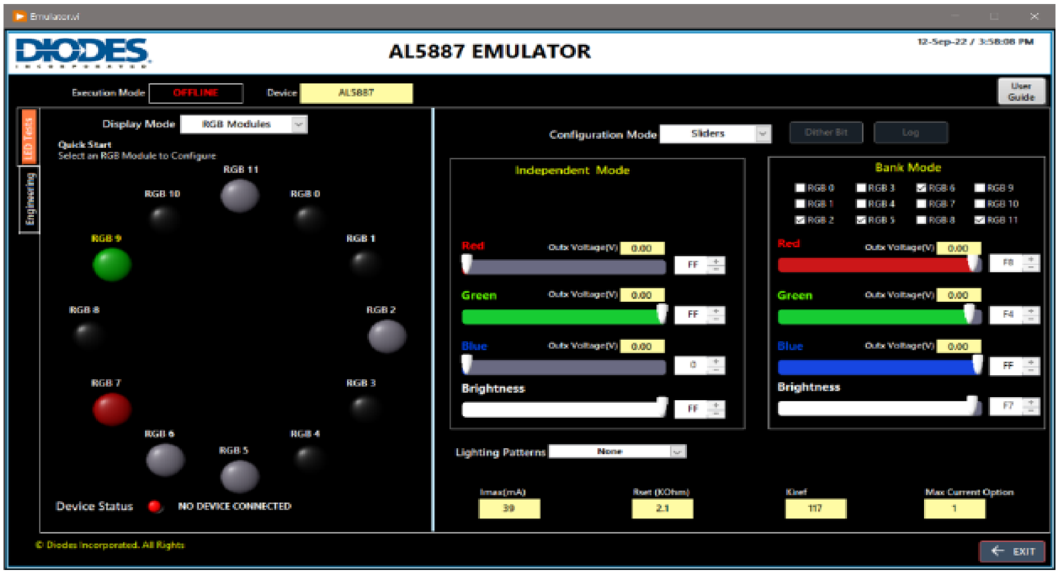


Figure 19: RGB Modules

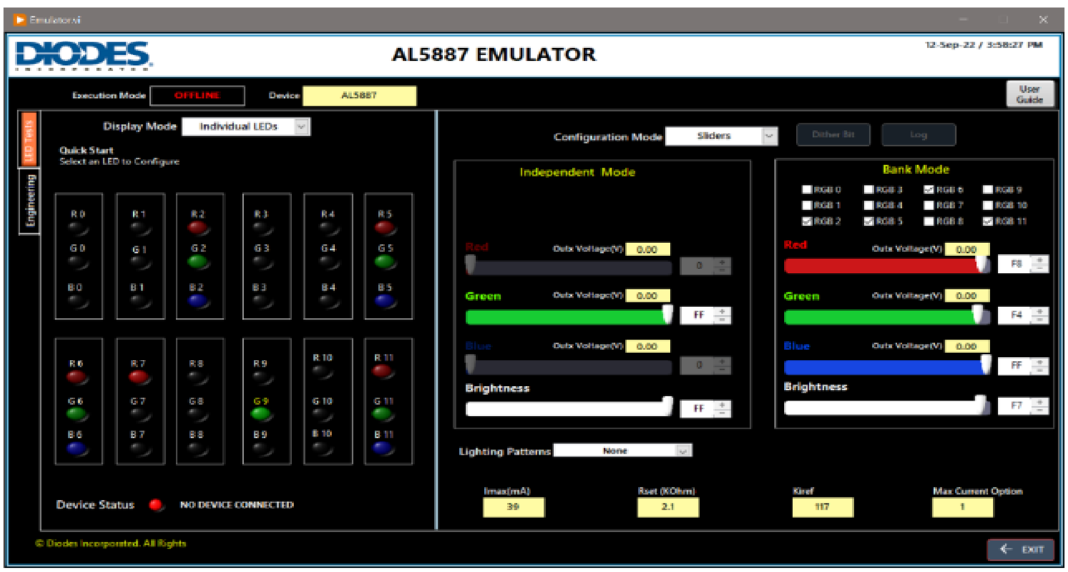


Figure 20: Individual LEDs

Configure the LEDs to adjust color and intensity in two different settings:

1. Sliders (Independent Mode and Bank Mode)
2. Registers.

Configure the following lighting patterns in the RGB module's annunciation mode:

1. Breathing effects
2. Mono color chasing effects
3. Dual color chasing effects
4. Multi-color chasing effects

The Engineering Tab can assist the user on viewing current settings and to compute the OutX voltage.

Execution Instructions: Configuration Screen

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

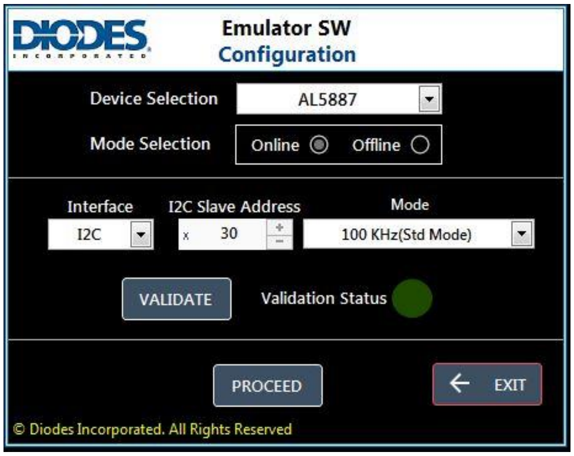


Figure 21: Emulator Configuration Screen

Emulator SW verifies specific bits in the AL5887 Device Register.

1. If the values match, SW displays the validation status as PASS and navigates to the Emulator Screen.
2. If the values do not match, SW displays the validation Status as FAIL and alerts the user with a message as seen in Figure 22.

Note: If Validation fails, the user can choose one of the below options:

1. The user can select MODIFY in the pop-up and re-visit step 4.
2. The user can select PROCEED to continue using Emulator SW in OFFLINE Mode.
3. The user can select ABORT to quit the application and verify the hardware.

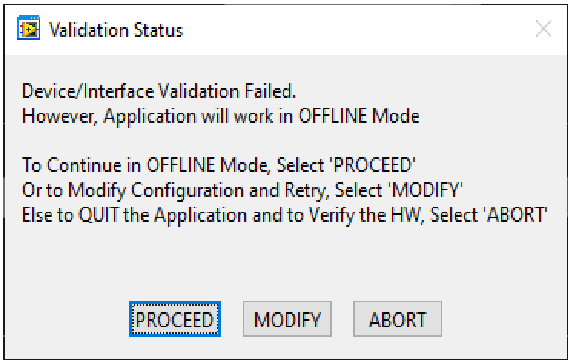


Figure 22: Validation status

Execution Instructions: Configuration Screen (continued)

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

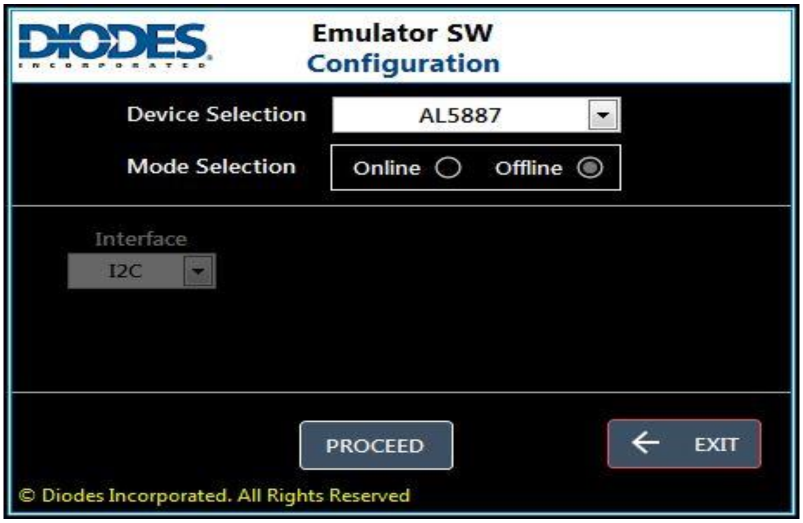


Figure 23: Offline Mode Selection

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

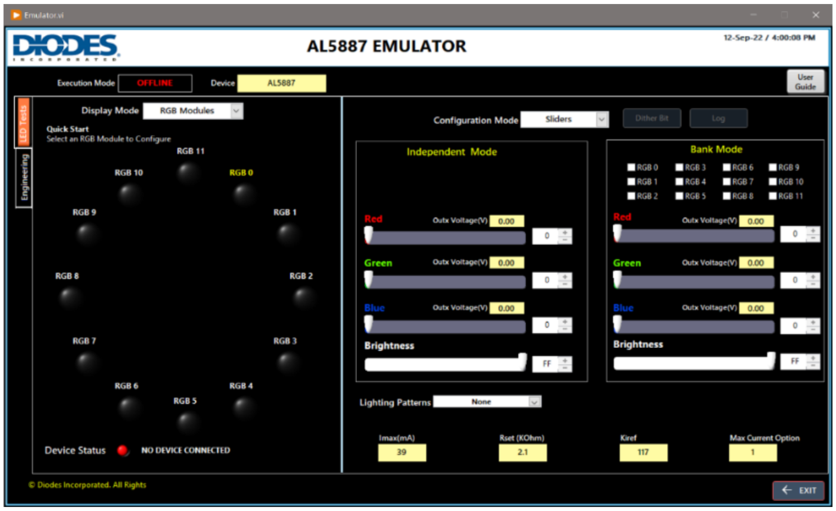


Figure 24: Emulator Screen in Offline Mode

LED Tests

Configuring LEDs in RGB Modules and 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 color and intensity.

Figure 25: Configuring LEDs in RGB Modules and 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 a combination of RED+GREEN+BLUE, and so on.

Configuring LEDs in RGB Modules and Bank Mode

1. Select Display Mode as **RGB Modules.**
2. Enable one or more banks to enable RGBs in **Bank Mode.**
3. Move the Red/Green/Blue Slides under Bank Mode to adjust the color and intensity of the RGB Modules selected in Bank Mode.

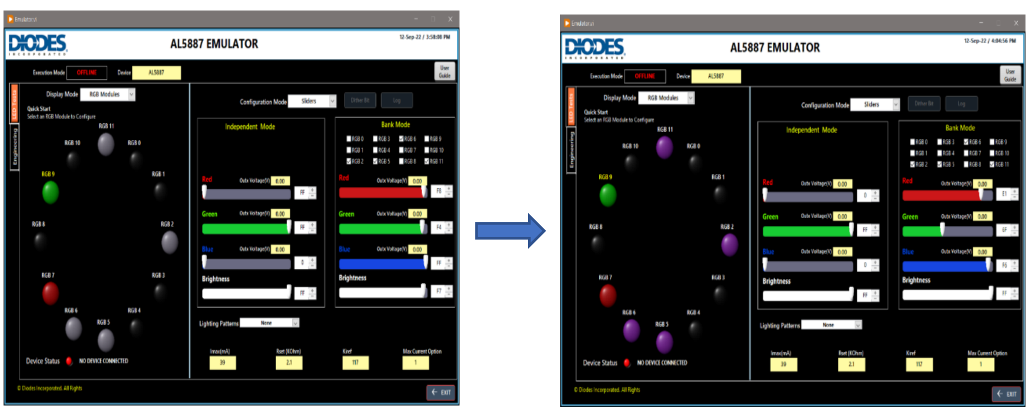


Figure 26: Configuring LEDs in RGB Modules and Bank Mode

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

LED Tests (continued)

Configuring LEDs in Individual LEDs and Independent Mode

1. Select Display Mode as **Individual LEDs.**
2. Select an **Individual LED** (example G2). Only the green slider is enabled in Independent Mode, while red and blue are grayed out.
3. Adjust the Green slider to vary the color and intensity of the G2 LED.

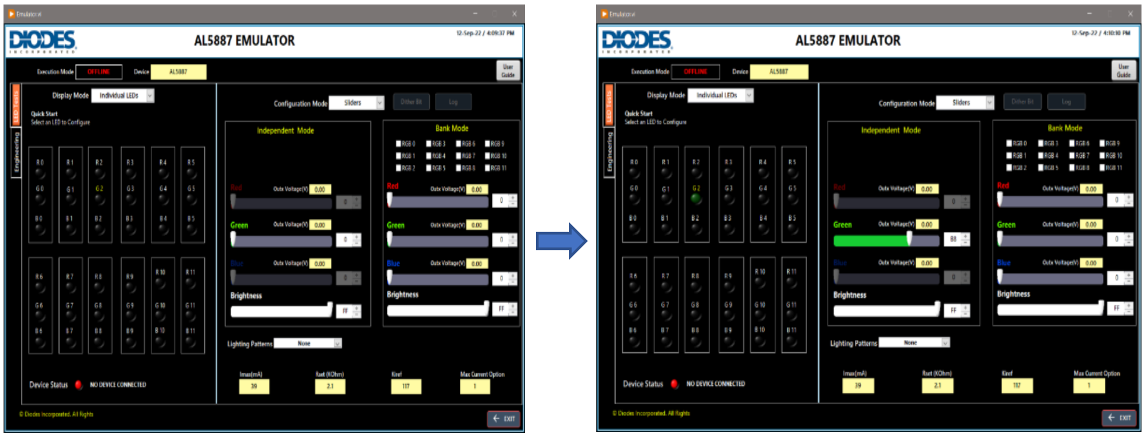


Figure 27: Configuring LEDs in Individual LEDs and Independent Mode

Each LED can be configured to different red/blue/green settings in **Independent Mode**.

Configuring LEDs in Individual LEDs and Bank Mode

1. Select Display Mode as **Individual LEDs.**
2. Enable one or more banks to enable RGBs in **Bank Mode.**
3. Move the red/green/blue sliders under Bank Mode to adjust the color and intensity of the RGB Modules selected in Bank Mode.

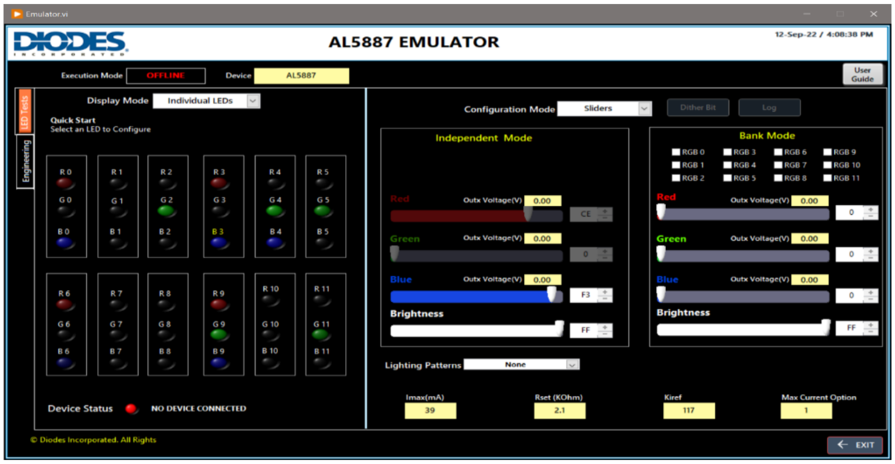


Figure 28: Configuring LEDs in Individual LEDs and Bank Mode

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

LED Tests (continued)

Notes on Display Mode

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

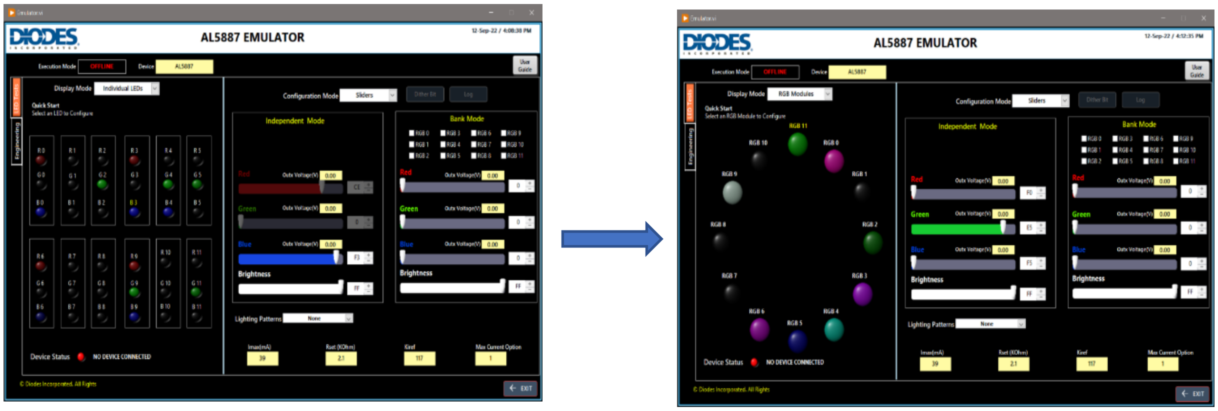


Figure 29: LED Changes in Individual LEDs and RGB Modules

Lighting Patterns

Emulator SW displays the following lighting patterns in RGB Modules Annunciation Mode

1. Breathing effect: Displays LEDs in red color with a breathing effect.
2. Mono color chasing effect: Displays LEDs in one color that cycles from RGB0-RGB11.
3. Dual color chasing effect: Displays LEDs in two colors that cycle from RGB0-RGB11.
4. Multi-color chasing effect: Displays LEDs in multiple colors that cycle from RGB0-RGB11.
5. None: LED lighting pattern is disabled.

Colors will continue to change during each cycle, except for the breathing effect.

Select a lighting pattern from the dropdown menu to simulate the pattern.

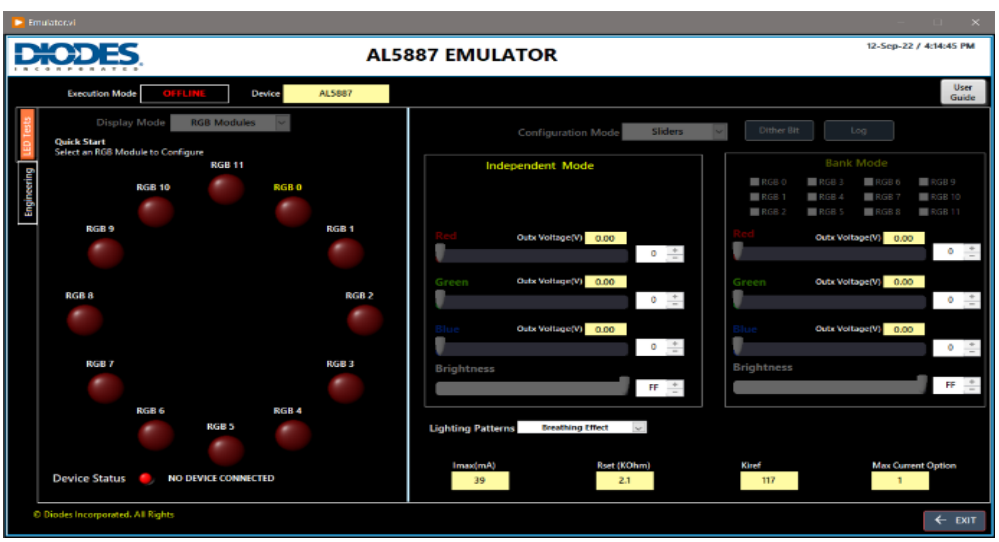


Figure 30: Breathing Effect

Lighting Patterns (continued)

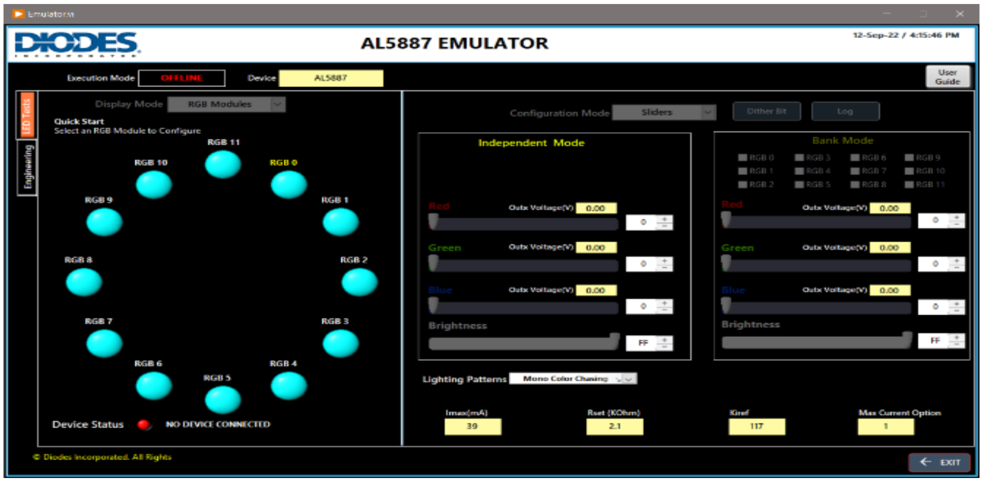


Figure 31: Mono Colour Chasing

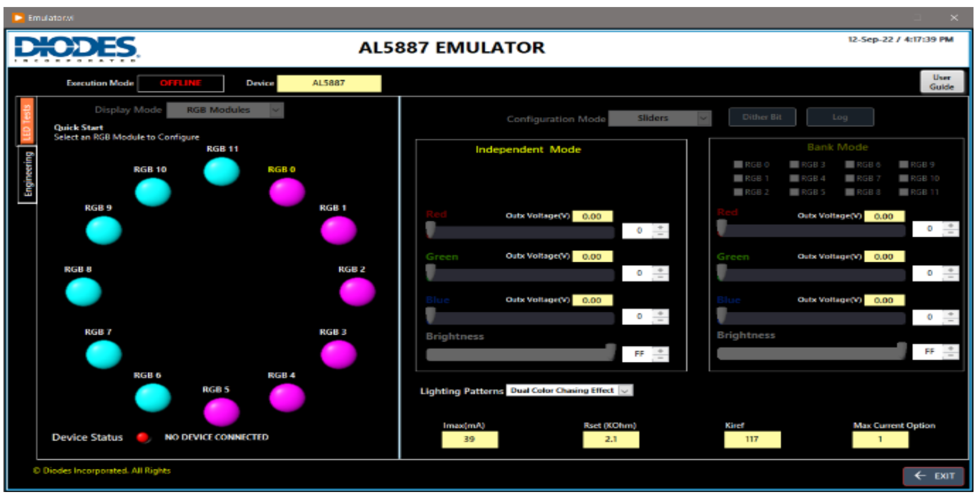


Figure 32: Dual Colour Chasing

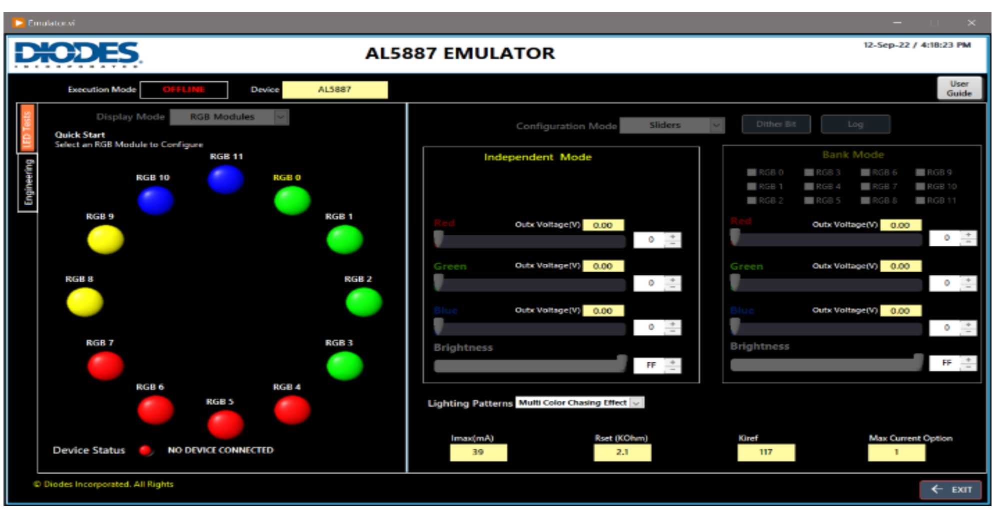


Figure 33: Multi-Colour Chasing

Configuring LEDs using Register Configuration Mode

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

(For example: LED_CONFIG1 (0x03). This register enables/disables BANK mode of RGB8 to RGB11. In this scenario, RGB10 and RGB11 are enabled).

3. When values of a register are modified, the applicable Register(s) will be highlighted in a blue color, indicating the value change. These values are not yet applied to the device and GUI.

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

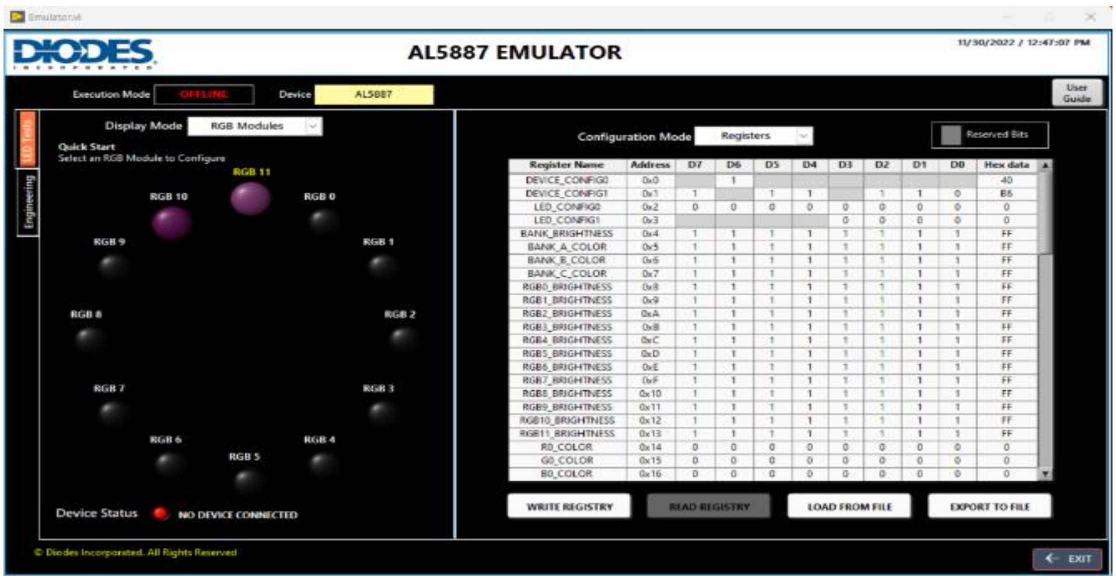


Figure 34: Register Configuration Mode

Note: Reserved bits are grayed out in the Register Table. Please refer to datasheet for further register and bit details.

Other Features in Register Configuration Mode

Configuration mode also offers other features, as seen below.

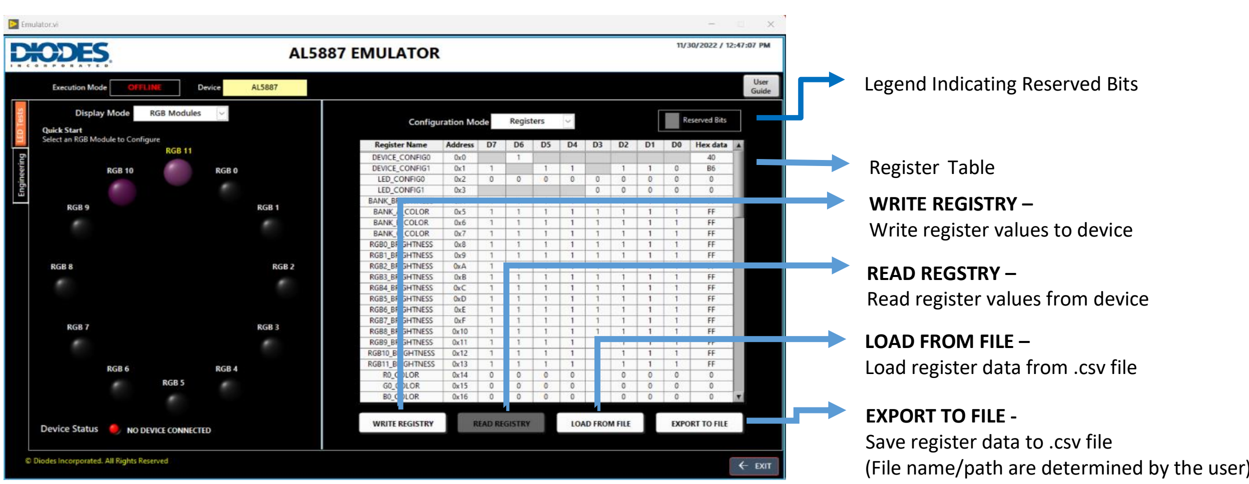


Figure 35: Register Configuration Mode Features

Engineering Tab

Features

Serves as a guide in computing the current configuration and Outx voltage.

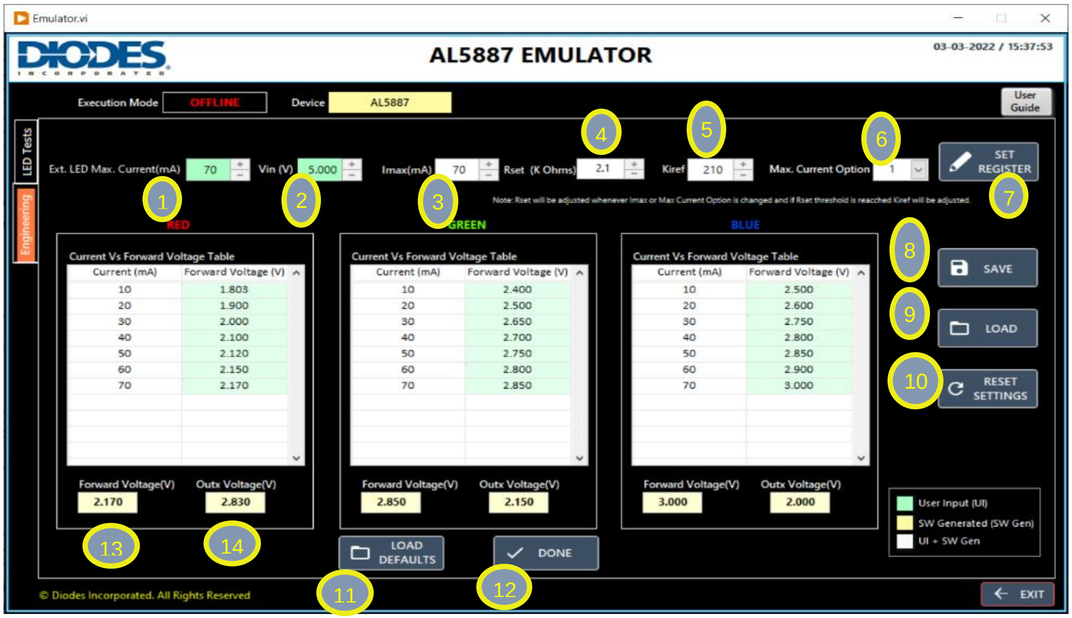


Figure 36: Engineering Tab Features

1. **Ext.LED Max Current(mA)**: Recommended Value <= 70mA.
   SW Alerts user when value entered is > 70mA.
2. **Input Voltage**: Vin(V) – Default is 5V.
3. **Imax Value**: 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 are 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 user’s choice of *.ini file.
9. **LOAD**: Load configuration from user’s choice of *.ini file.
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 automatically updated by the SW whenever the Imax is adjusted.

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

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.

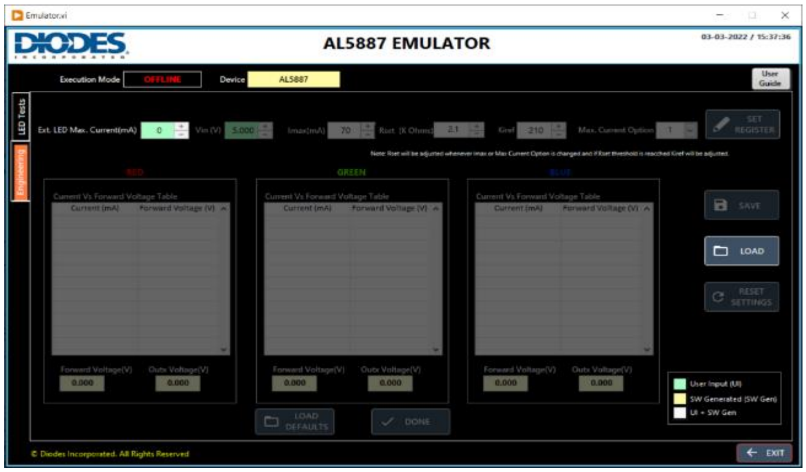


Figure 37: Engineering Tab Default Screen

3. The emulator will display a pop-up indicating “Configuration Uploaded”.
4. All fields will be auto-populated by the emulator as shown below.

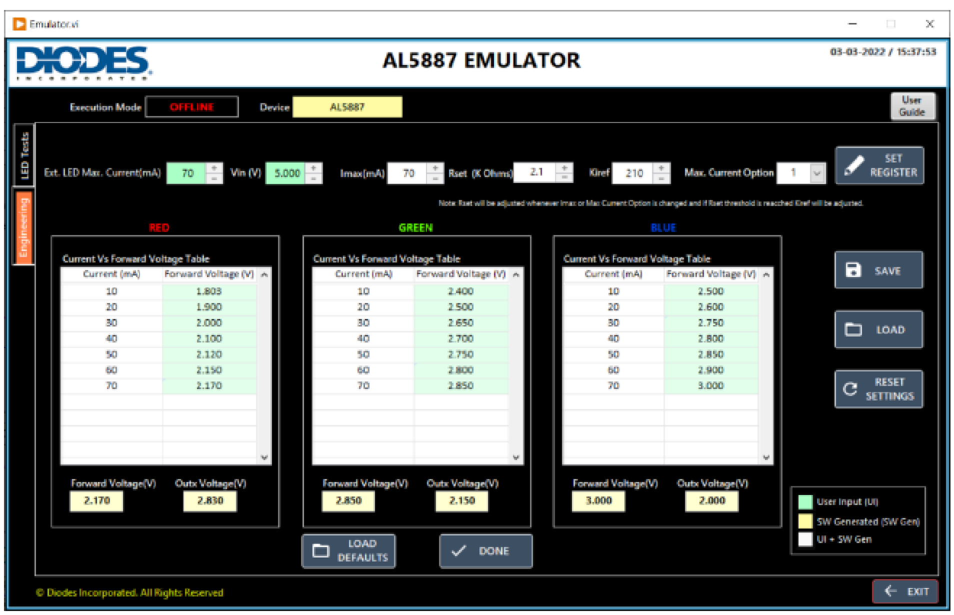


Figure 38: Default Values Loaded

Note:

1. By default, once the file is loaded, the parameter settings are as follows.
   Imax = 70, Rset = 2.1, Kiref = 210, Max current = 1. Forward voltage and Outx voltages are calculated for current Imax values.
2. Now, the user can make modifications to the current parameters to check how the changes impact other associated parameters.
3. SET REGISTER will update the Global Dimming Bits (Kiref) and max. current option.
4. Please refer to the Register Table.
   - Brightness of selected LEDs
   - AL5887 HW if connected in Online Mode.

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 (recommended to be less than 70mA, to be updated based on the datasheet). When entered, multiple fields are enabled automatically as in the screenshot below.

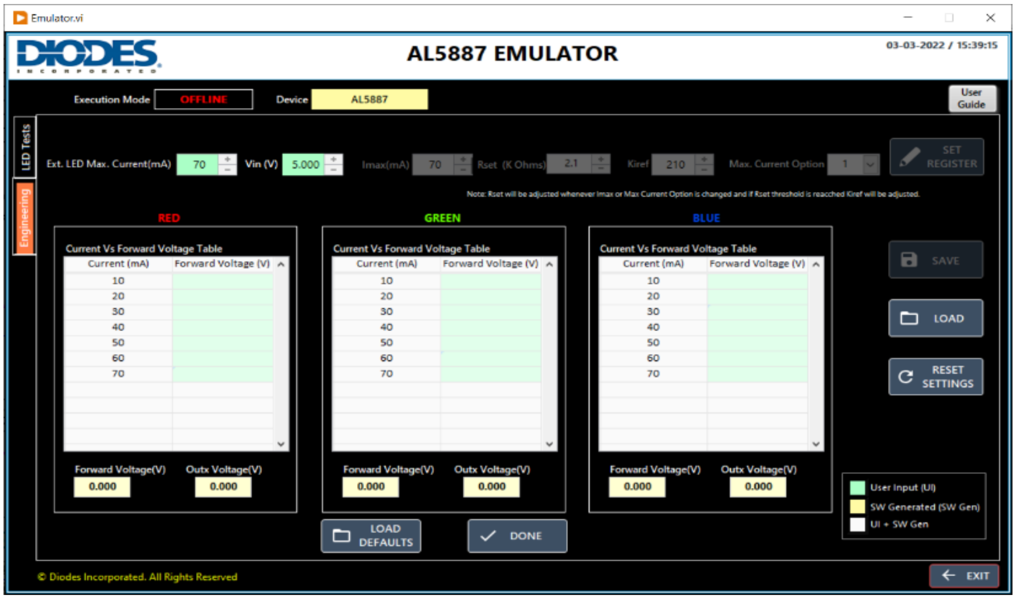


Figure 39: LED Current Updated

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

Figure 40: Default Forward Voltages Loaded

5. Once the current vs. forward voltage table values have been entered, select the **DONE** button.
   Note: Current configuration parameters will be disabled until DONE is selected.
6. All the fields will now be enabled.

Instructions to Compute Outx Voltage

Figure 41: All 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
  - Register Table in **LED Tests**
  - Brightness of selected LEDs in **LED Tests**
  - AL5887 HW if connected in Online Mode.

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

Micro Controller Setup – Using the Arduino Board as an Example

The AL5887QEV1 demo board can be easily connected to a µC of your choice. Below is an example of an Arduino setup**.**

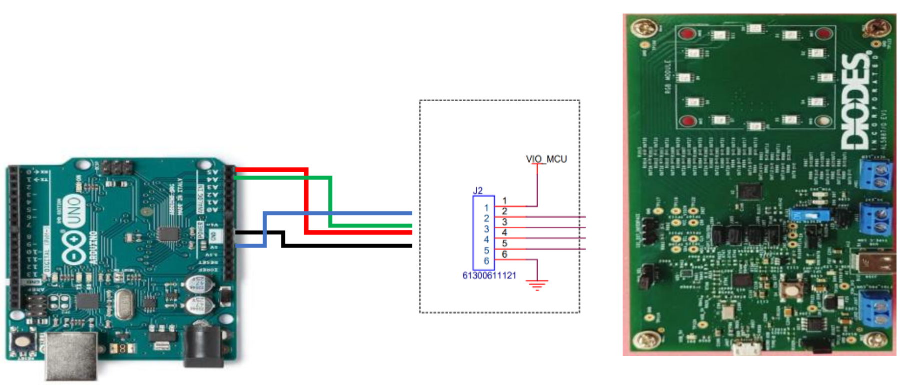


Figure 42: Wiring Diagram from Arduino Board to Demo Board through External I2C Interface

Signal	**Arduino UNO R3**	**AL5887Q DEMO**
SCL	A5	J2 pin 3
SDA	A4	J2 pin 2
VIO	5V	J2 pin 1
GND	GND	J2 pin 4

Table 2: Arduino I2C Connections

Example Code

```
/****************************************************************************/
/*     Author: Diodes INC    */
/*       Date: 10/15/2024      */
/* 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);
}
```

IMPORTANT NOTICE

1. DIODES INCORPORATED (Diodes) AND ITS SUBSIDIARIES MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARDS TO ANY INFORMATION CONTAINED IN THIS DOCUMENT, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION).

2. The Information contained herein is for informational purpose only and is provided only to illustrate the operation of Diodes' products described herein and application examples. Diodes does not assume any liability arising out of the application or use of this document or any product described herein. This document is intended for skilled and technically trained engineering customers and users who design with Diodes' products. Diodes' products may be used to facilitate safety-related applications; however, in all instances customers and users are responsible for (a) selecting the appropriate Diodes products for their applications, (b) evaluating the suitability of Diodes' products for their intended applications, (c) ensuring their applications, which incorporate Diodes' products, comply the applicable legal and regulatory requirements as well as safety and functional-safety related standards, and (d) ensuring they design with appropriate safeguards (including testing, validation, quality control techniques, redundancy, malfunction prevention, and appropriate treatment for aging degradation) to minimize the risks associated with their applications.

3. Diodes assumes no liability for any application-related information, support, assistance or feedback that may be provided by Diodes from time to time. Any customer or user of this document or products described herein will assume all risks and liabilities associated with such use, and will hold Diodes and all companies whose products are represented herein or on Diodes' websites, harmless against all damages and liabilities.

4. Products described herein may be covered by one or more United States, international or foreign patents and pending patent applications. Product names and markings noted herein may also be covered by one or more United States, international or foreign trademarks and trademark applications. Diodes does not convey any license under any of its intellectual property rights or the rights of any third parties (including third parties whose products and services may be described in this document or on Diodes' website) under this document.

5. Diodes' products are provided subject to Diodes' Standard Terms and Conditions of Sale (https://www.diodes.com/about/company/terms-and-conditions/terms-and-conditions-of-sales/) or other applicable terms. This document does not alter or expand the applicable warranties provided by Diodes. Diodes does not warrant or accept any liability whatsoever in respect of any products purchased through unauthorized sales channel.

6. Diodes' products and technology may not be used for or incorporated into any products or systems whose manufacture, use or sale is prohibited under any applicable laws and regulations. Should customers or users use Diodes' products in contravention of any applicable laws or regulations, or for any unintended or unauthorized application, customers and users will (a) be solely responsible for any damages, losses or penalties arising in connection therewith or as a result thereof, and (b) indemnify and hold Diodes and its representatives and agents harmless against any and all claims, damages, expenses, and attorney fees arising out of, directly or indirectly, any claim relating to any noncompliance with the applicable laws and regulations, as well as any unintended or unauthorized application.

7. While efforts have been made to ensure the information contained in this document is accurate, complete and current, it may contain technical inaccuracies, omissions and typographical errors. Diodes does not warrant that information contained in this document is error-free and Diodes is under no obligation to update or otherwise correct this information. Notwithstanding the foregoing, Diodes reserves the right to make modifications, enhancements, improvements, corrections or other changes without further notice to this document and any product described herein. This document is written in English but may be translated into multiple languages for reference. Only the English version of this document is the final and determinative format released by Diodes.

8. Any unauthorized copying, modification, distribution, transmission, display or other use of this document (or any portion hereof) is prohibited. Diodes assumes no responsibility for any losses incurred by the customers or users or any third parties arising from any such unauthorized use.

9. This Notice may be periodically updated with the most recent version available at https://www.diodes.com/about/company/terms-and-conditions/important-notice

www.diodes.com

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Diodes Incorporated:

AL5887QEV1