

39x9 DOTS MATRIX LED DRIVER

DESCRIPTION

The IS31FL3741A is a general purpose 39x9 LED Matrix programmed via an I2C compatible interface. Each LED can be dimmed individually with 8-bit PWM data and 8-bit scaling data which allowing 256 steps of linear PWM dimming and 256 steps of DC current adjustable level.

Additionally each LED open and short state can be detected, IS31FL3741A store the open or short information in Open-Short Registers. The Open-Short Registers allowing MCU to read out via I2C compatible interface. Inform MCU whether there are LEDs open or short and the locations of open or short LEDs.

FEATURES

- Supply voltage range from 2.7V to 5.5V
- 39 Current Sink x 9 SW matrix size: drive up to 351 LEDs or 117 RGBs
- Individual 256 PWM control steps
- Individual 256 DC current steps
- Global 255 current setting
- SDB rising edge reset I2C module
- 29kHz/3.6kHz/1.8kHz/900Hz PWM frequency
- Interrupt and state lookup registers
- Individual open and short error detect function
- De-ghost
- QFN-60 (7mmx7mm) package

QUICK START

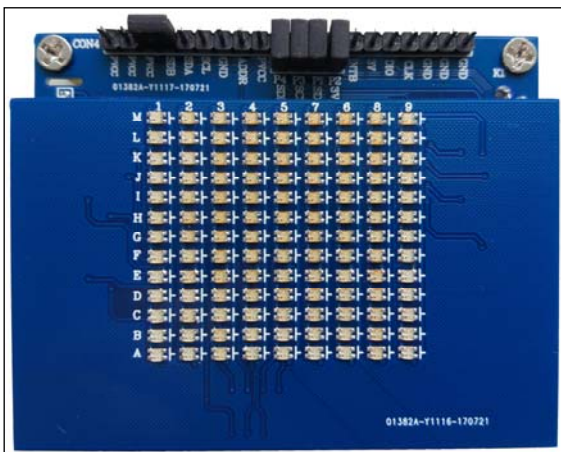


Figure 1: Photo of IS31FL3741A Evaluation Board

RECOMMENDED EQUIPMENT

- 5.0V, 2A power supply

ABSOLUTE MAXIMUM RATINGS

- ≤ 5.5V power supply

Caution: Do not exceed the conditions listed above, otherwise the board will be damaged.

PROCEDURE

The IS31FL3741A evaluation board is fully assembled and tested. Follow the steps listed below to verify board operation.

Caution: Do not turn on the power supply until all connections are completed.

- 1) Connect 3V and P2.
- 2) Connect SDA and P1.
- 3) Connect SCL and P3.
- 4) Connect SDB and P4.
- 5) Connect PVCC and SDB.
- 6) Turn on the power supply/Plug in the Micro USB
Pay attention to the supply current. If the current exceeds 1A, please check for circuit fault.

EVALUATION BOARD OPERATION

The IS31FL3741A evaluation board has three animation display modes. Press K1 to switch configurations.

- 1) Rainbow wave#1
- 2) Rainbow wave#2
- 3) Rain
- 4) Visor

Note: IS31FL3741A solely controls the FxLED function on the evaluation board.

ORDERING INFORMATION

Part No.	Temperature Range	Package
IS31FL3741A-QFLS4-EB	-40°C to +125°C (Industrial)	QFN-60, Lead-free

Table 1: Ordering Information

For pricing, delivery, and ordering information, please contacts Lumissil's analog marketing team at analog@Lumissil.com or (408) 969-6600.

39×9 DOTS MATRIX LED DRIVER

SOFTWARE SUPPORT

P2 and 3V default setting is closed (jumper on). If it is open (no jumper), the on-board MCU will configure its own I2C/SDB/ADDR pins to High Impedance status so an external source can driver the I2C/SDB signals to control the IS31FL3741A LED driver.

The ADDR pin is pulled high via R6 setting the device address of IS31FL3741A to 0x60.

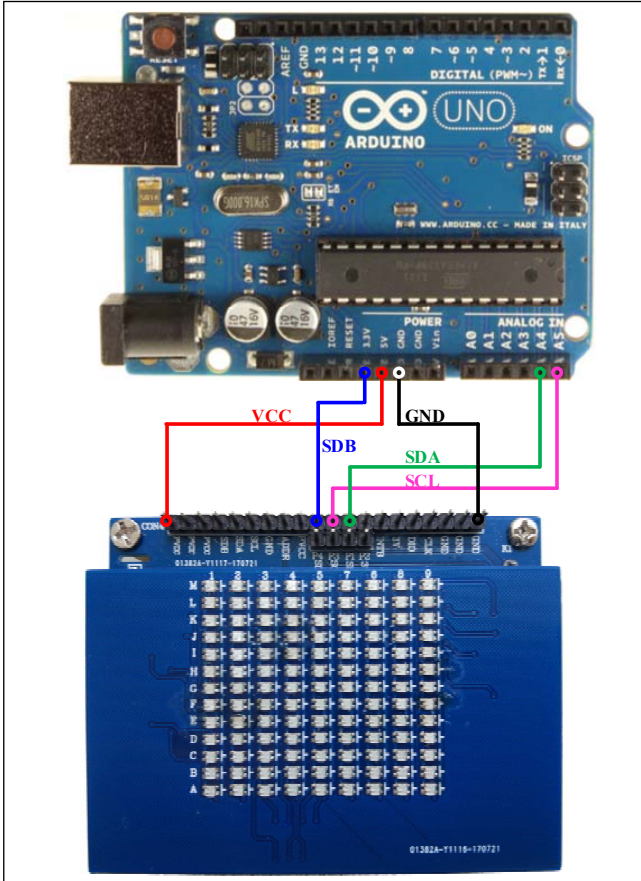


Figure 2: Photo of Arduino UNO connected to Evaluation Board

The steps listed below are an example using the Arduino for external control.

The Arduino hardware consists of an Atmel microcontroller with a bootloader allowing quick firmware updates. First download the latest Arduino Integrated Development Environment IDE (1.6.12 or greater) from www.arduino.cc/en/Main/Software. Also download the Wire.h library from www.arduino.cc/en/reference/wire and verify that pgmspace.h is in the directory ...program Files(x86)/Arduino/hardware/tools/avr/avr/include/avr/. Then download the latest IS31FL3741A test firmware (sketch) from the Lumissil website <http://www.lumissil.com/products/led-driver/fxled>.

- 1) Open P2 and 3V.
- 2) Open PVCC and SDB.
- 3) Connect the 5 pins from Arduino board to IS31FL3741A EVB:
 - a) Arduino 5V pin to IS31FL3741A EVB PVCC.
 - b) Arduino GND to IS31FL3741A EVB GND.
 - c) Arduino SDA (A4) to IS31FL3741A EVB P1.
 - d) Arduino SCL (A5) to IS31FL3741A EVB P3.
 - e) If Arduino use 3.3V MCU VCC, connect 3.3V to IS31FL3741A EVB SDB, if Arduino use 5.0V MCU VCC, connect 5.0V to EVB SDB.
(Arduino UNO MCU VCC is 5V, so SDB can be 5V or 3.3V)
- 4) Use the test code in appendix I or download the test firmware (sketch) from the Lumissil website, a .txt file and copy the code to Arduino IDE, compile and upload to Arduino.
- 5) Run the Arduino code as appendix I.
- 6) In EVB code, the ADDR pin is pulled low via 100k, so the device address is 0x60. When P1~P4 are disconnect, the ADDR pin can be changed by connecting the ADDR as follows:
 - a) ADDR=VCC, device address=0x66.
 - b) ADDR=SCL, device address=0x62.
 - c) ADDR=SDA, device address=0x64.

Please refer to the datasheet to get more information about IS31FL3741A.

39x9 DOTS MATRIX LED DRIVER

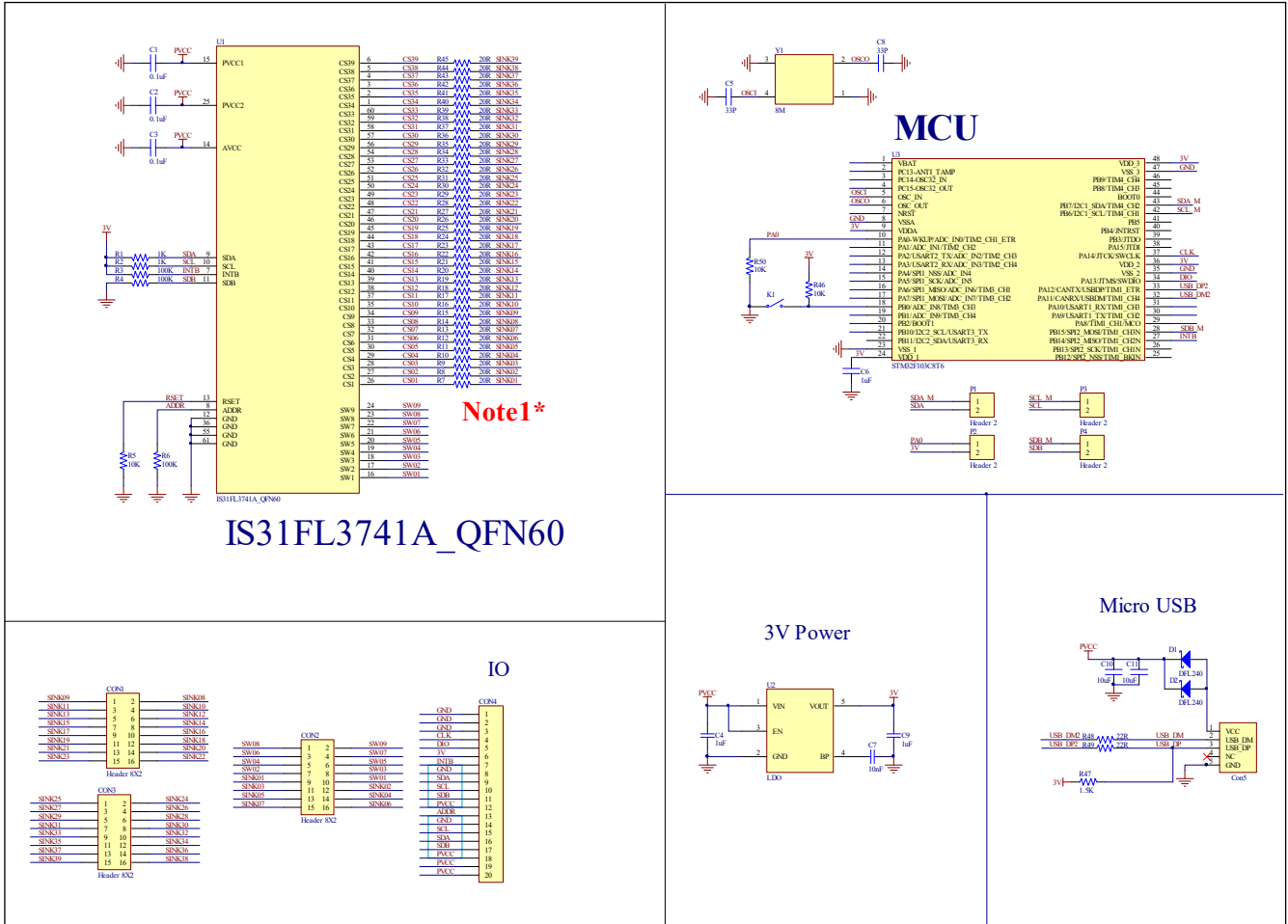


Figure 3: IS31FL3741A Application Schematic

39x9 DOTS MATRIX LED DRIVER



Figure 4: FxLED_9x13_ARRAY Application Schematic

39×9 DOTS MATRIX LED DRIVER

BILL OF MATERIALS

IS31FL3741A

Name	Symbol	Description	Qty	Supplier	Part No.
LED Driver	U1	Matrix LED Driver	1	Lumissil	IS31FL3741A
LDO	U2	Reduced voltage	1	SGMICRO	SGM2019-3.3V
MCU	U3	Microcontroller	1	STM	STM32F103C8T6
Diode	D1,D2	Diode, SMD	2	DIODES	DFLS240
Crystal	Y1	Crystal, 8MHz	1	JB	HC-49S
Resistor	R1,R2	RES,1k,1/10W,±5%,SMD	2	Yageo	RC0603JR-071KL
Resistor	R3,R4,R6	RES,100K,1/10W,±5%,SMD	3	Yageo	RC0603JR-07100KL
Resistor	R5,R46,R50	RES,10k,1/10W,±5%,SMD	3	Yageo	RC0603JR-0710KL
Resistor	R47	RES,1.5k,1/10W,±5%,SMD	1	Yageo	RC0603JR-071K5L
Resistor	R48,R49	RES,22R,1/10W,±5%,SMD	2	Yageo	RC0603JR-0722RL
Resistor	R7,R8, R10,R11 ~ R40, R41, R43, R44	RES,20R,1/10W,±5%,SMD	26	Yageo	RC0603JR-0720RL
Resistor	R9,R12, R15,R18, ~ R42,R45	RES,20R,1/10W,±5%,SMD (Note 1)	13	Yageo	RC0603JR-0720RL
Capacitor	C1,C2,C3	CAP,0.1µF,16V,±20%,SMD	3	Yageo	CC0603MRX7R7BB104
Capacitor	C4,C6,C9	CAP,1µF,16V,±20%,SMD	3	Yageo	CC0603KRX7R7BB105
Capacitor	C5,C8	CAP,33pF,50V,±5%,SMD	2	Yageo	CQ0603JRNPO9BN360
Capacitor	C7	CAP,10nF,16V,±20%,SMD	1	Yageo	CC0603KRX7R7BB103
Capacitor	C10,C11	CAP,10µF,16V, ±20%,SMD	2	Yageo	CC0805MKX5R7BB106
Button	K1	Button SMD	1		

Bill of Materials, refer to Figure 3 above.

FxLED 9x13 ARRAY

Name	Symbol	Description	Qty	Supplier	Part No.
Diode	D1~D117	RGB LED, SMD	117	Everlight	9-237/R6GHBHC-A01/2T

Bill of Materials, refer to Figure 4 above.

Note 1: The value of these resistors on the evaluation board is 20Ω. For PVCC=5V and red LED application, prefer 51Ω for these resistors as shown in datasheet Figure 1.

39x9 DOTS MATRIX LED DRIVER

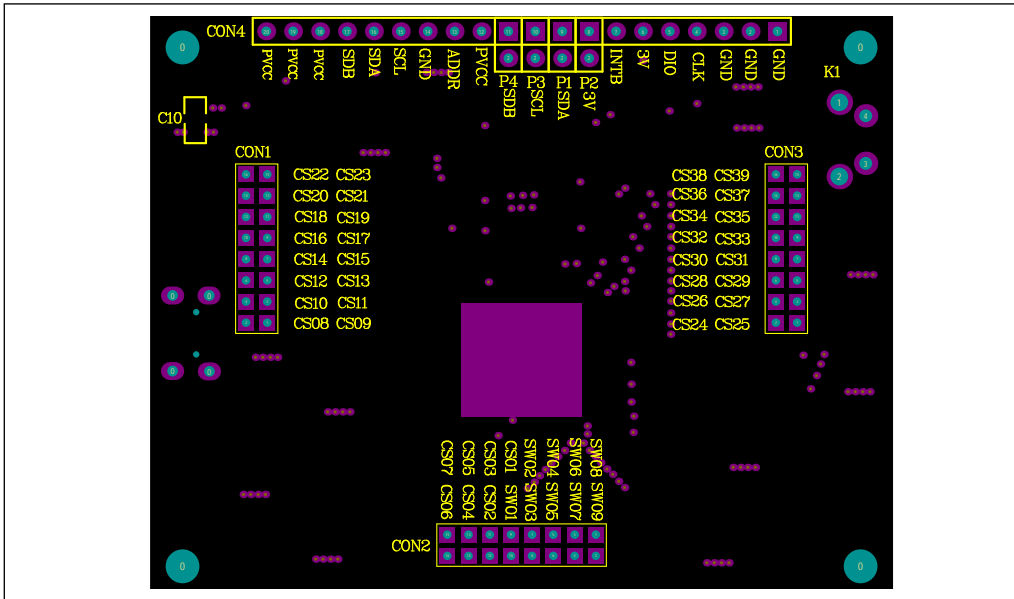


Figure 5: Board Component Placement Guide - Top Layer

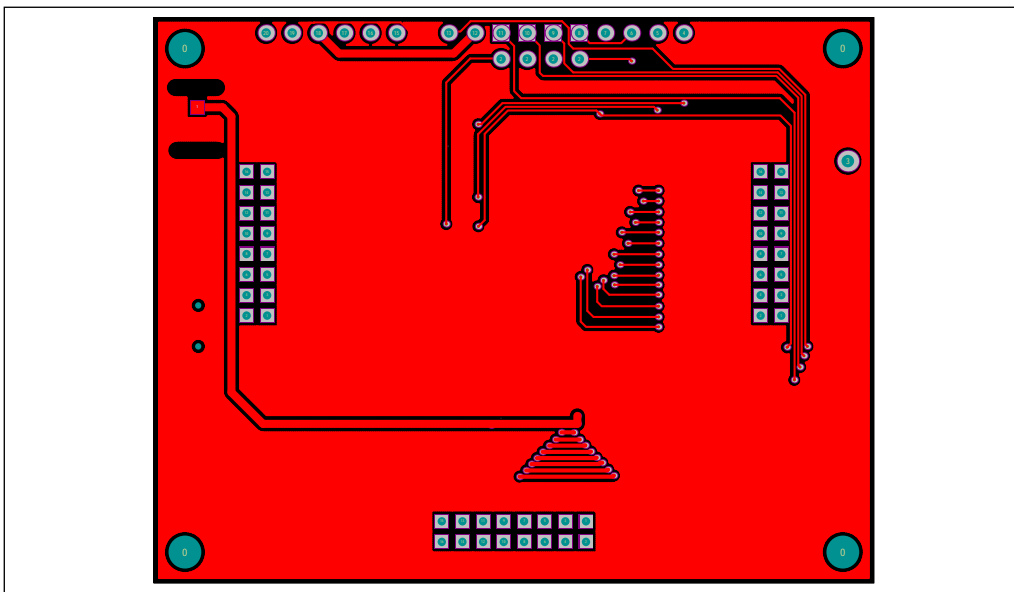


Figure 6: Board PCB Layout - Top Layer

39×9 DOTS MATRIX LED DRIVER

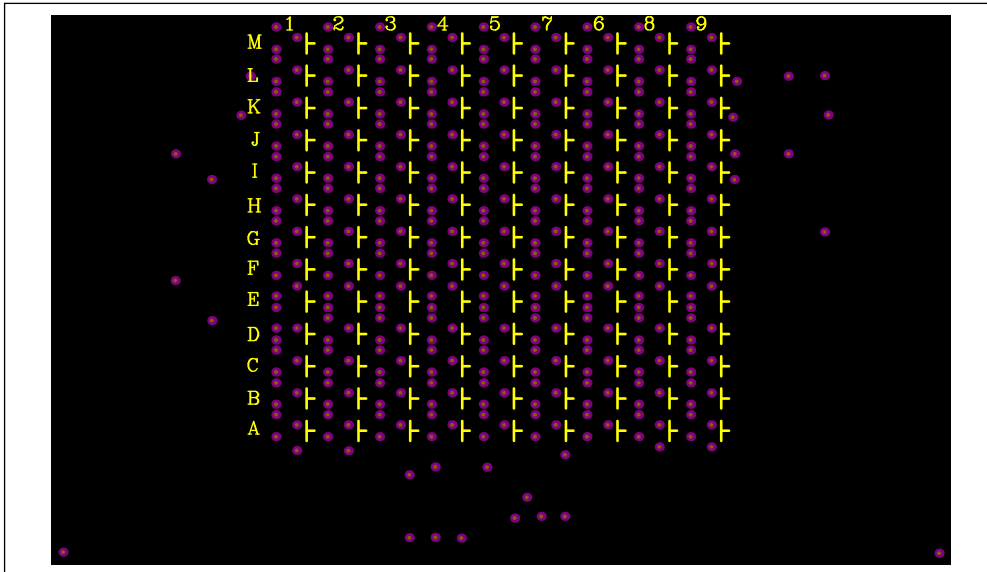


Figure 9: LED Board Component Placement Guide - Top Layer

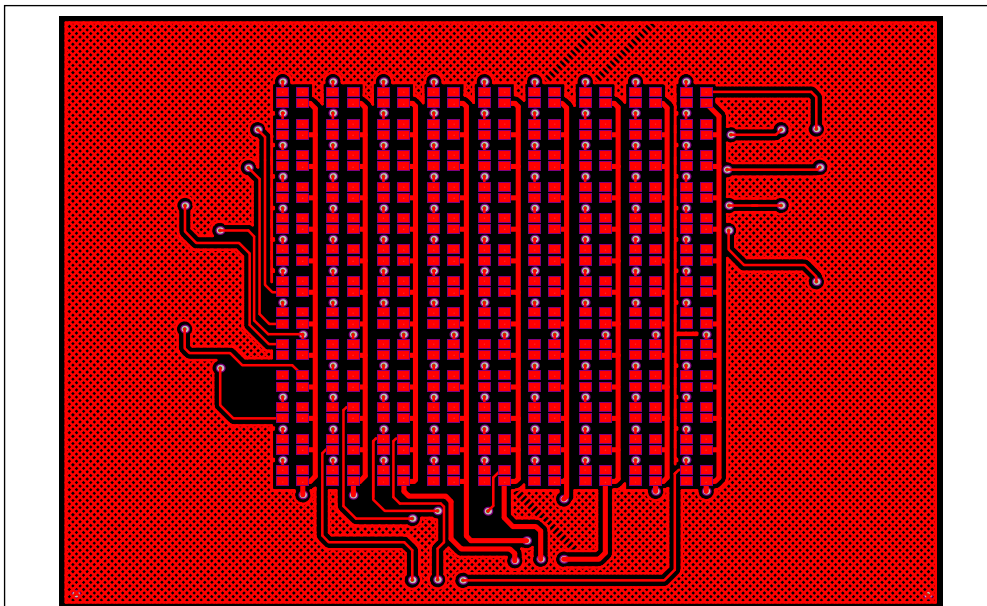


Figure 10: LED Board PCB Layout - Top Layer

39x9 DOTS MATRIX LED DRIVER

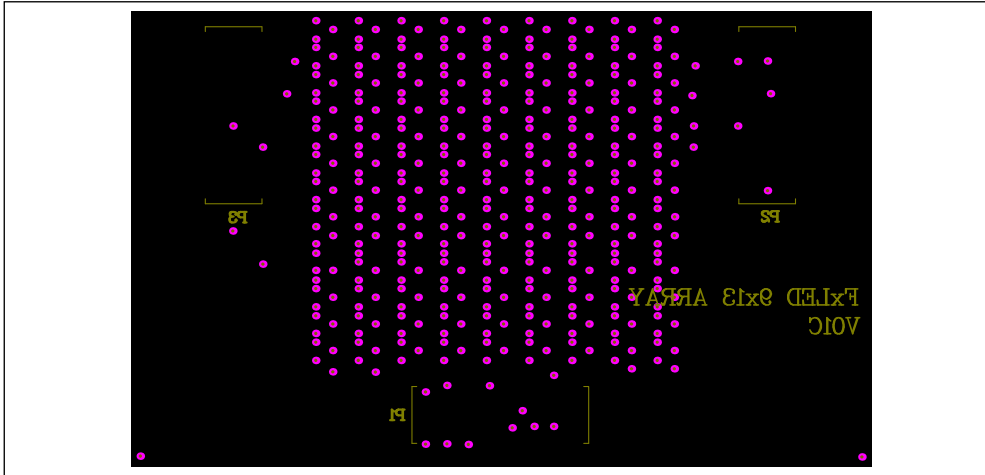


Figure 11: LED Board Component Placement Guide - Bottom Layer

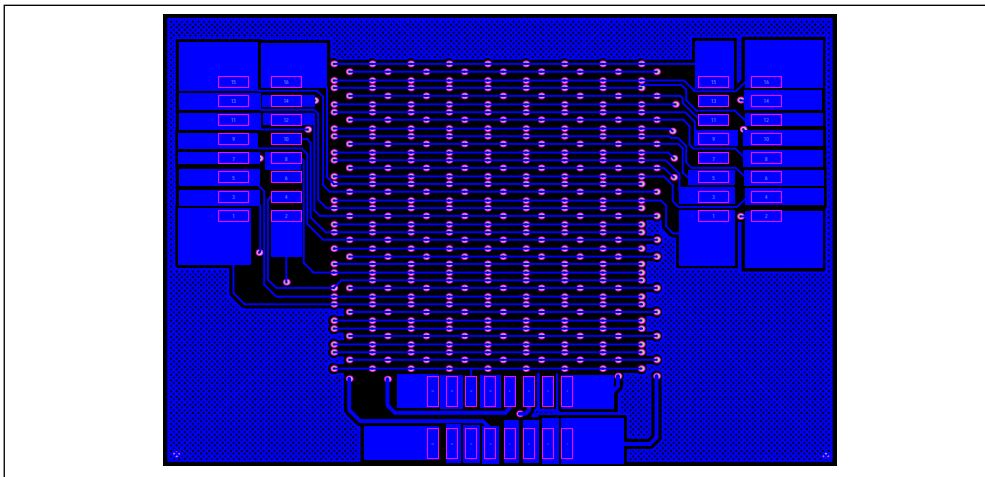


Figure 12: LED Board PCB Layout - Bottom Layer

Copyright © 2021 Lumissil Microsystems. All rights reserved. Lumissil Microsystems reserves the right to make changes to this specification and its products at any time without notice. Lumissil Microsystems assumes no liability arising out of the application or use of any information, products or services described herein. Customers are advised to obtain the latest version of this device specification before relying on any published information and before placing orders for products.

Lumissil Microsystems does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless Lumissil Microsystems receives written assurance to its satisfaction, that:

- a.) the risk of injury or damage has been minimized;
- b.) the user assume all such risks; and
- c.) potential liability of Lumissil Microsystems is adequately protected under the circumstances

39×9 DOTS MATRIX LED DRIVER

REVISION HISTORY

Revision	Detail Information	Date
A	Initial release	2019.09.05
B	Update the Arduino Code	2021.08.31

39×9 DOTS MATRIX LED DRIVER

APPENDIX I : IS31FL3741A Arduino Test Code V01B

```
#include<Wire.h>
#include<avr/pgmspace.h>

#define Addr_GND 0x60
#define R 0xFF
#define G 0x00
#define B 0xFF
uint8_t i,j;

const PROGMEM byte PWM_Gama64[64]=
{
    0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,
    0x08,0x09,0x0b,0x0d,0x0f,0x11,0x13,0x16,
    0x1a,0x1c,0x1d,0x1f,0x22,0x25,0x28,0x2e,
    0x34,0x38,0x3c,0x40,0x44,0x48,0x4b,0x4f,
    0x55,0x5a,0x5f,0x64,0x69,0x6d,0x72,0x77,
    0x7d,0x80,0x88,0x8d,0x94,0x9a,0xa0,0xa7,
    0xac,0xb0,0xb9,0xbf,0xc6,0xcb,0xcf,0xd6,
    0xe1,0xe9,0xed,0xf1,0xf6,0xfa,0xfe,0xff
};

void setup() {
    Wire.begin();
    Wire.setClock(400000);//I2C 400KHz
    Init_3741A(R, G, B);
}

void loop() {
    IS31FL3741A_Test_mode1();//breath mode
}

void IS_IIC_WriteByte(uint8_t Dev_Add,uint8_t Reg_Add,uint8_t Reg_Dat)
{
    Wire.beginTransmission(Dev_Add/2); // transmit to device IS31FL373x
    Wire.write(Reg_Add); // sends regaddress
    Wire.write(Reg_Dat); // sends regaddress
    Wire.endTransmission(); // stop transmitting
}

void Init_3741A(uint8_t Rdata, uint8_t Gdata, uint8_t Bdata)
{
    IS_IIC_WriteByte(Addr_GND,0xfe,0xc5);//unlock
    IS_IIC_WriteByte(Addr_GND,0xfd,0x02);//write page 2
    for(i=2;i<0xB4;i+=3)
```

39x9 DOTS MATRIX LED DRIVER

```

{
  IS_IIC_WriteByte(Addr_GND,i,Rdata);//R LED Scaling
}
for(i=1;i<0xB4;i+=3)
{
  IS_IIC_WriteByte(Addr_GND,i,Gdata);//G LED Scaling
}
for(i=0;i<0xB4;i+=3)
{
  IS_IIC_WriteByte(Addr_GND,i,Bdata);//B LED Scaling
}
IS_IIC_WriteByte(Addr_GND,0xfe,0xc5);//unlock
IS_IIC_WriteByte(Addr_GND,0xfD,0x03);//write page 3
for(i=2;i<0xAB;i+=3)
{
  IS_IIC_WriteByte(Addr_GND,i,Rdata);//R LED Scaling
}
for(i=1;i<0xAB;i+=3)
{
  IS_IIC_WriteByte(Addr_GND,i,Gdata);//G LED Scaling
}
for(i=0;i<0xAB;i+=3)
{
  IS_IIC_WriteByte(Addr_GND,i,Bdata);//B LED Scaling
}

IS_IIC_WriteByte(Addr_GND,0xfe,0xc5);//unlock
IS_IIC_WriteByte(Addr_GND,0xfD,0x00);//write page 0
for(i=0;i<0xB4;i++)
{
  IS_IIC_WriteByte(Addr_GND,i,0x00);//write all PWM set 0x00
}
IS_IIC_WriteByte(Addr_GND,0xfe,0xc5);//unlock
IS_IIC_WriteByte(Addr_GND,0xfD,0x01);//write page 1
for(i=0;i<0xAB;i++)
{
  IS_IIC_WriteByte(Addr_GND,i,0x00);//write all PWM set 0x00
} //init all the PWM data to 0

IS_IIC_WriteByte(Addr_GND,0xfe,0xc5);//unlock
IS_IIC_WriteByte(Addr_GND,0xfD,0x04);//write page 4
IS_IIC_WriteByte(Addr_GND,0x01,0x7F);//global current
IS_IIC_WriteByte(Addr_GND,0x36,0x0B);//PWM frequency setting
IS_IIC_WriteByte(Addr_GND,0x00,0x01);//normal operation
}

```