LOCAL DIMMING 24×4 MATRIX LED DRIVER EVALUATION BOARD GUIDE

DESCRIPTION

IS32FL3749 is a matrix LED driver with 24 high voltage (16V) constant current channels. It supports from one to four power scan to become a 24×n (n=1~4) matrix LED driver. Each LED can be pulse width modulated (PWM) with maximum 16-bit precision for smooth LED brightness control. In addition, each LED can be controlled by an 8-bit output current control register (Dot correction, current scale, SL), which allows fine tuning the current for rich RGB color mixing, e.g., a pure white color LED application. The maximum output current of each channel is designed to be 60mA, which can be adjusted by three 8-bit global control registers group (one group for R for channels 3×I, one group for G for channels 3×I+1, and one group for B for channels 3×I+2, where I= 0 to 7). Proprietary algorithms are used in IS32FL3749 to minimize power bus noise caused by passive components on the power bus such as MLCC decoupling capacitor. All registers can be programmed via VSB (video series bus, up to 33MHz) or SPI (up to 33MHz) bus.

IS32FL3749 can be turned off with minimum current consumption by either pulling the SDB pin low or by using the software shutdown feature. It internally generates 4.8V VOUT to power the internal logic operation, which can also be external powered from 3V to 5.5V.

IS32FL3749 is available in eTQFP-48 (7mm × 7mm) package and can work over temperature range from -40°C to +125°C.

QUICK START

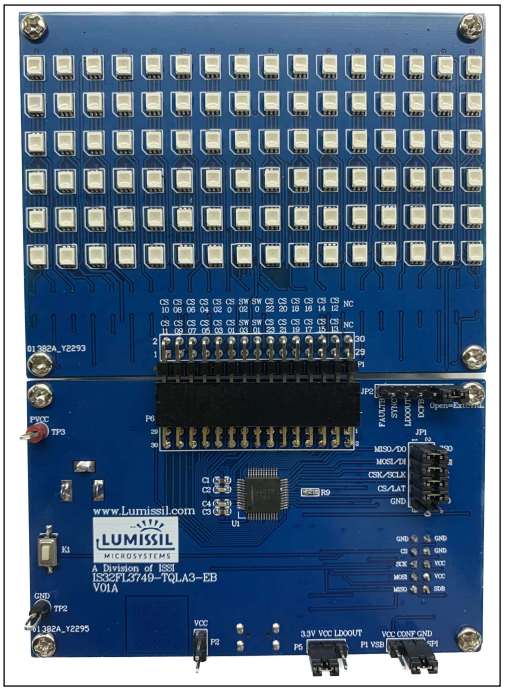

Figure 1: Photo of IS32FL3749 Evaluation Board

FEATURES

- Support 24 constant current channels, 60mA/ch
- Tolerate up to 20V, nominal operation voltage between 4.5V to 16V, multiple LED's can be connected in series
- Built-in LDO to generate 4.8V supply for internal logic (option to float it and use external power)
- Interface
  - VSB (video serial bus, 33MHz)
  - SPI (33MHz)
- Reset register reset all the registers to default
- Support 8-bit, 16-bit, 8+4-bit dithering and 8+8-bit dithering PWM mode
- Built-in Dot correction: 8-bit/dot
- 8-bit × 3 global current adjustment
- 4 groups delay to minimize the power ripple
- Channel to channel timing skew (one sys-clock skew to reduce transient noise)
- Support bi-directional data output via DI
- PSW short protection
- Spread spectrum
- Programmable detection of open/short, detected LED and store detected LED information in registers for ease of manufacturing/debugging
- Low standby and sleep mode current
- For matrix scanning operation
  - Built-in de-ghosting circuit
  - Reduced inactive LED reverse bias to improve LED reliability
- Over temperature protection
- Operating temperature: -40°C to 125°C
- AEC-Q100 qualification in progress
- eTQFP-48 (7mm × 7mm) package
- RoHS & Halogen-Free Compliance
- TSCA Compliance

RECOMMENDED EQUIPMENT

- 12V, ≥1A power supply

ABSOLUTE MAXIMUM RATINGS

- VIN+, ≤15V power supply

Caution: EVB is designed for 12V application, higher than 12V will cause extra-heat on the IC and if VIN (TP3 or CON2) is higher than 15V, the IS32FL3749 will be too hot and enter thermal shutdown mode, if VIN (TP3 or CON2) exceeds the conditions listed above, the board may be damaged.

PROCEDURE

The IS32FL3749 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 the ground terminal of the power supply to the GND and the positive terminal to the TP3 (PVCC), or connect the DC power to the connector (CON2).
2) Connect '5' and '6' in JP2 (Internal Control) to enable the control of board MCU (default status).
3) Connect '3.3V' and 'VCC' in P5.
4) SPI MODE:
   Connect 'MISO/DO' to 'MISO' in JP1.
   Connect 'MOSI/DI' to 'MOSI' in JP1.
   Connect 'CSK/SCLK' to 'SCK' in JP1.
   Connect 'CS/LAT' to 'CS' in JP1.
   Connect 'GND' and 'CONF' in P1.
5) VSB MODE:
   Connect 'MISO/DO' to 'MISO' in JP1.
   Connect 'MOSI/DI' to 'MOSI' in JP1.
   Connect 'CSK/SCLK' to 'SCK' in JP1.
   Connect 'CS/LAT' to 'CS' in JP1.
   Connect 'VCC' and 'CONF' in P1.
6) Turn on the power supply and pay attention to the supply current. If the current exceeds 1A, please check for circuit fault.
7) Enter the desired mode of display by toggling the MODE button (K1).

ORDERING INFORMATION

Part No.	Temperature Range	Package
IS32FL3749-TQLA3-EB	-40°C to +125°C (Industrial)	eTQFP-48, 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.

EVALUATION BOARD OPERATION

The IS32FL3749 evaluation board has 6 display modes. Press MODE button (K1) to switch configurations.

1) (Default mode) Rainbow Mode
2) Color Change Mode
3) Red Mode
4) Green Mode
5) Blue Mode
6) White Mode

Note: IS32FL3749 solely controls the FxLED function on the evaluation board.

SOFTWARE SUPPORT

JP2 default setting is 'IntCTRL' (Internal Control). If it is set to 'ExtCTRL' (External Control), the on-board MCU will configure all the IO pins to high impedance mode and enter sleep mode. The SPI pins and SDB pin are also set to High Impedance. External SPI and SDB signals can be connected to JP1 and TP1 to control the IS32FL3749 LED driver.

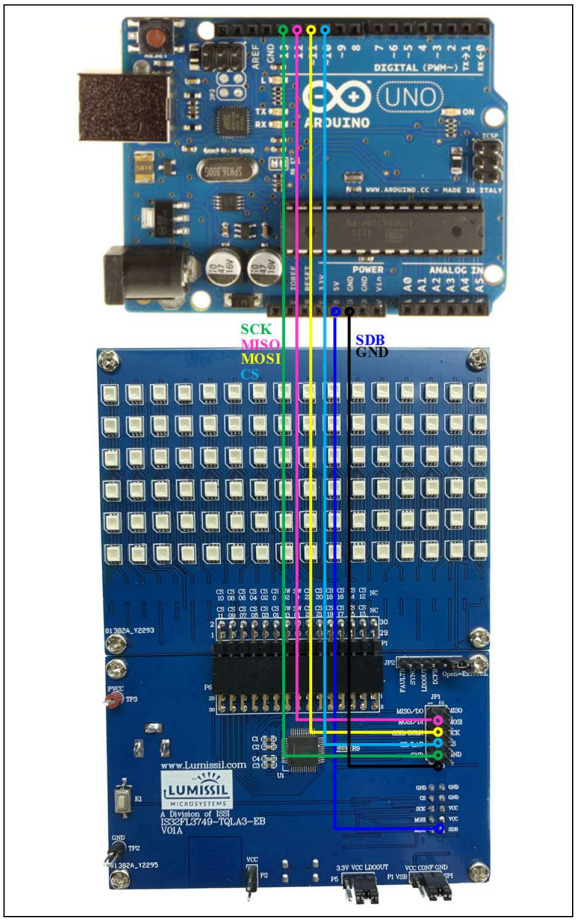

Figure 2: Arduino UNO Connected to Evaluation Board

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 IS32FL3749 test firmware (sketch) from the ISSI website http://www.lumissil.com/products/led-driver/fxled.

1) Open '5' and '6' and JP2 to set to 'ExtCTRL' (External Control).
2) Connect 'VCC' and 'LDOOUT' in P5.
3) Remove MISO/MOSI/SCK/CS Jumper Cap in JP1.
4) Connect 6 pins from Arduino board to IS32FL3749 EVB:
   a) Arduino GND to IS32FL3749 EVB GND.
   b) Arduino SDB (5V) to IS32FL3749 EVB SDB in TP1.
   c) Arduino SCK (13) to IS32FL3749 EVB CSK/SCLK in JP1.
   d) Arduino MISO (12) to IS32FL3749 EVB MISO/DO in JP1.
   e) Arduino MOSI (11) to IS32FL3749 EVB MISO/DO in JP1.
   f) Arduino CS (10) to IS32FL3749 EVB CS/LAT in JP1.
   g) If Arduino use 3.3V MCU VCC, connect 3.3V to IS32FL3749 EVB SDB, if Arduino use 5.0V MCU VCC, connect 5.0V to EVB SDB. (Arduino UNO's VCC (VOH) is 5.0V, so SDB=5.0V)
5) Run the Arduino code for desired mode setting by Arduino code.

Please refer to the datasheet to get more information about IS32FL3749.

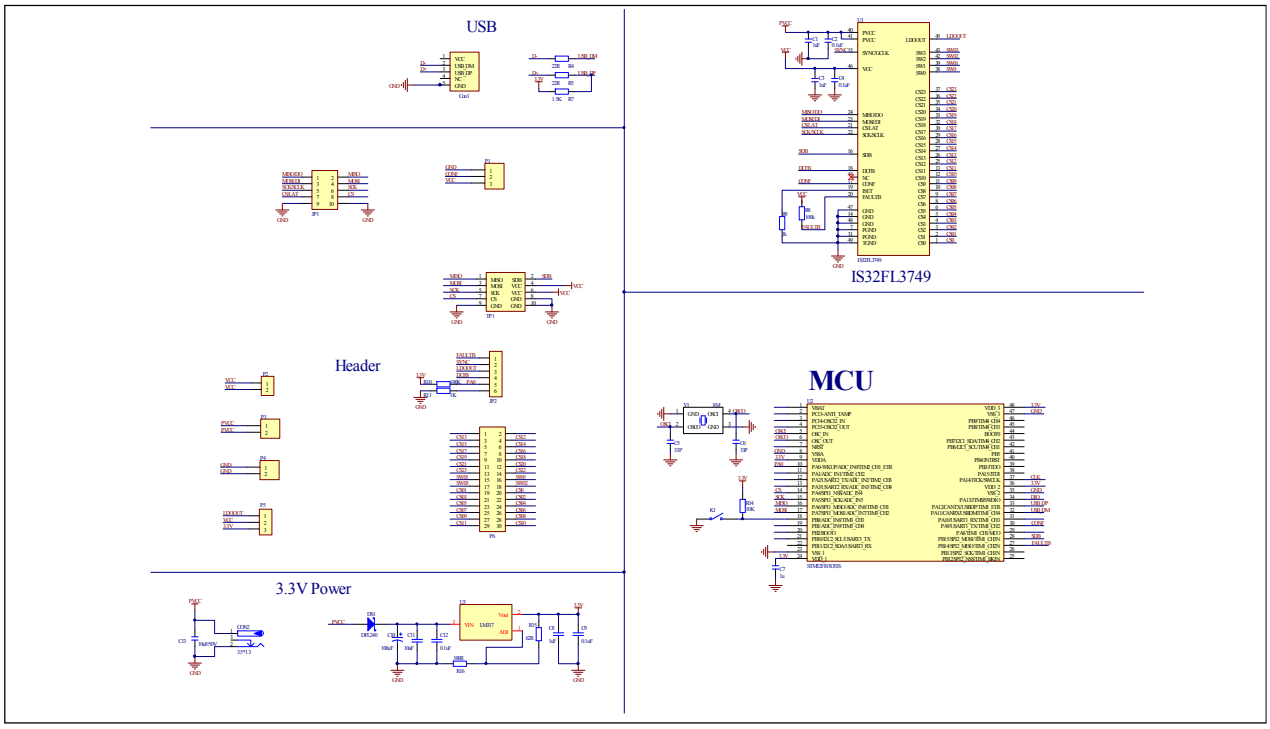

Figure 3: IS32FL3749 EVB Schematic

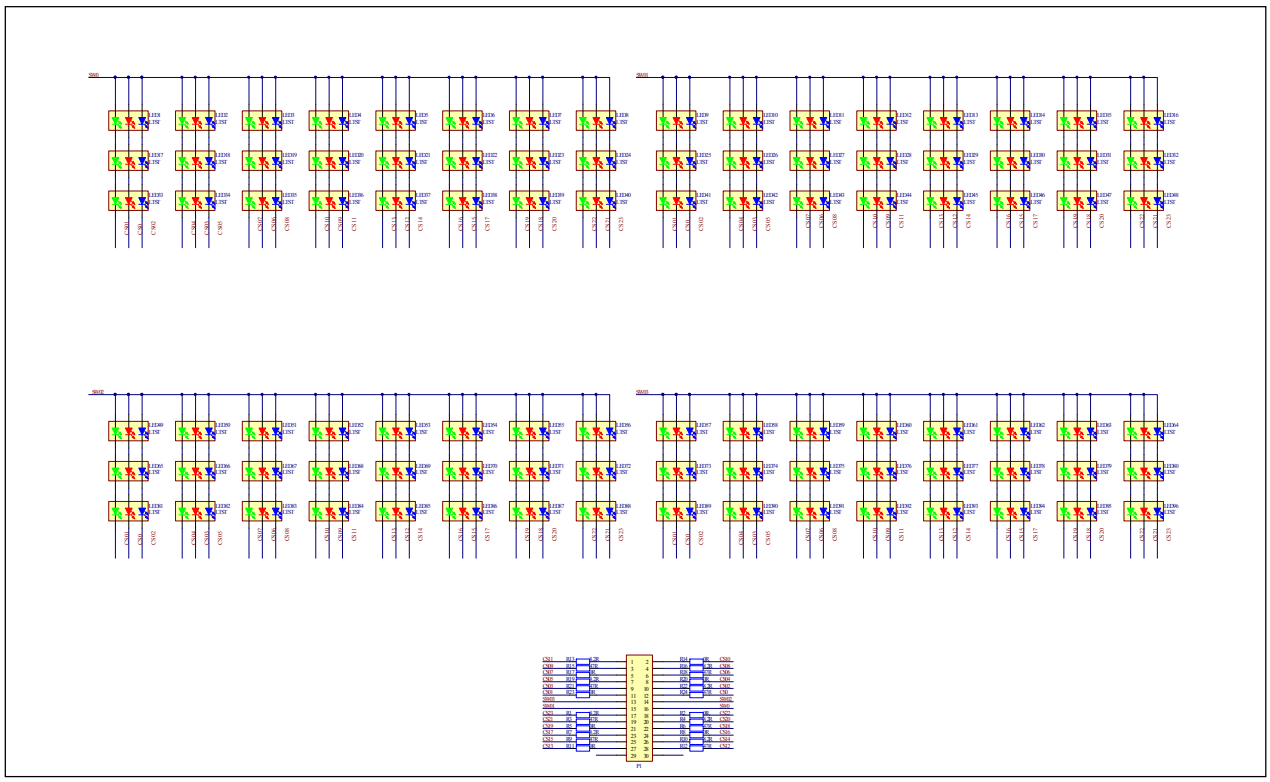

Figure 4: FxLED ARRAY 3×16×2 Schematic

BILL OF MATERIALS

IS32FL3749

Name	Symbol	Description	Qty	Supplier	Part No.
LED Driver	U1	Matrix LED Driver	1	LUMISSIL	IS32FL3749
MCU	U2	Microcontroller	1	STM	STM32F103C8T6
LDO	U3	Low-dropout Regulator	1	ONSEMI	LM317
Crystal	Y1	Crystal, 8MHz	1	JB	HC-49S
Diode	DS1	Diode, SMD	1	DIODES	DFLS240
Resistor	R4,R5	RES,22R,1/10W,±5%,SMD	2	Yageo	RC0603JR-0722RL
Resistor	R7	RES,1.5k,1/10W,±5%,SMD	2	Yageo	RC0603JR-072KL
Resistor	R8,R10	RES,100k,1/10W,±5%,SMD	2	Yageo	RC0603JR-07100KL
Resistor	R9,R11	RES,1k,1/10W,±5%,SMD	2	Yageo	RC0603JR-071KL
Resistor	R14	RES,10k,1/10W,±5%,SMD	1	Yageo	RC0603JR-0710KL
Resistor	R15	RES,62R,1/10W,±5%,SMD	1	Yageo	RC0603JR-07130RL
Resistor	R16	RES,100R,1/10W,±5%,SMD	1	Yageo	RC0603JR-07390RL
Capacitor	C1,C3,C7	CAP, 1μF,16V,±10%,SMD	3	Yageo	CC0603KRX7R7BB105
Capacitor	C2,C4,C9	CAP,100nF,16V,±20%,SMD	3	Yageo	CC0603MRX7R7BB104
Capacitor	C5,C6	CAP,33pF,50V,±5%,SMD	2	Yageo	CQ0603JRNPO9BN330
Capacitor	C8	CAP,1μF,16V,±10%,SMD	1	Yageo	AC0805KFX7R7BB105
Capacitor	C10	CAP,100μF,35V,±20%,SMD	1	Lelon	VEU101M1VTR-0810
Capacitor	C11,C13	CAP,10μF, 50V,±10%,SMD	2	Yageo	CC1206KKX5R9BB106
Capacitor	C12	CAP,100nF,25V,±10%,SMD	1	Yageo	AC1206KRX7R8BB104
Button	K1	Button, SMD	1		

Bill of Materials, refer to Figure 3 above.

FxLED ARRAY 3×16×2

Name	Symbol	Description	Qty	Supplier	Part No.
LED	LED1~LED96	RGB LED	96	EVERLIGHT	19-337/R6GHBHC-A01/2T
Resistor	R1,R4,R7…	RES,8.2R,1/4W,±5%,SMD	8	Yageo	RC1206JR-108R2L
Resistor	R2,R5,R8…	RES,0R,1/4W,±5%,SMD	8	Yageo	RC1206JR-070RL
Resistor	R3,R6,R9…	RES,47R,1/4W,±5%,SMD	8	Yageo	RC1206JR-1047RL

Bill of Materials, refer to Figure 4 above.

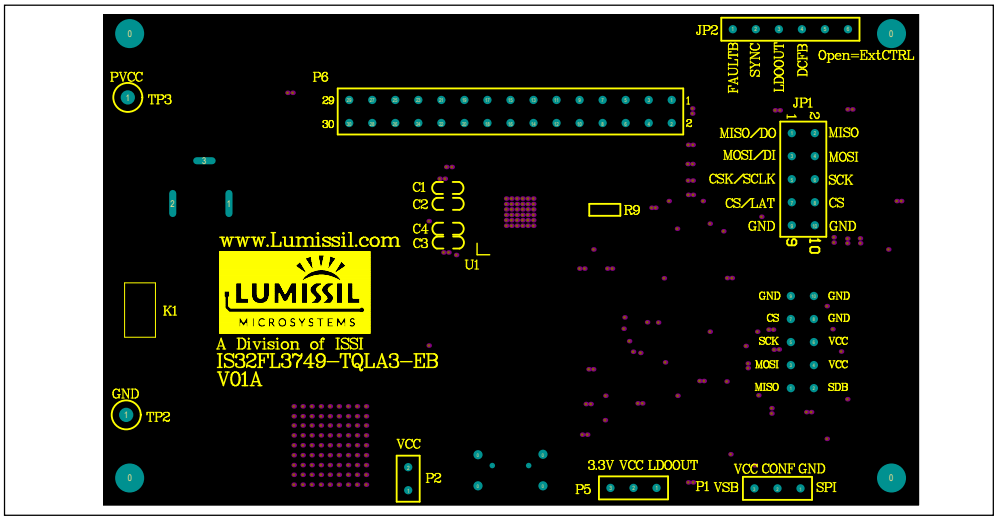

Figure 5: Board Component Placement Guide - Top Layer

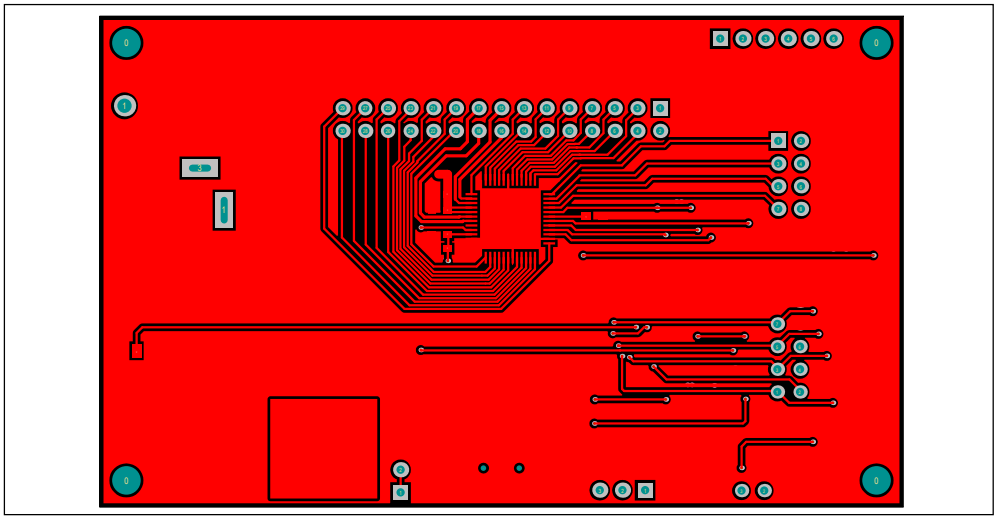

Figure 6: Board PCB Layout - Top Layer

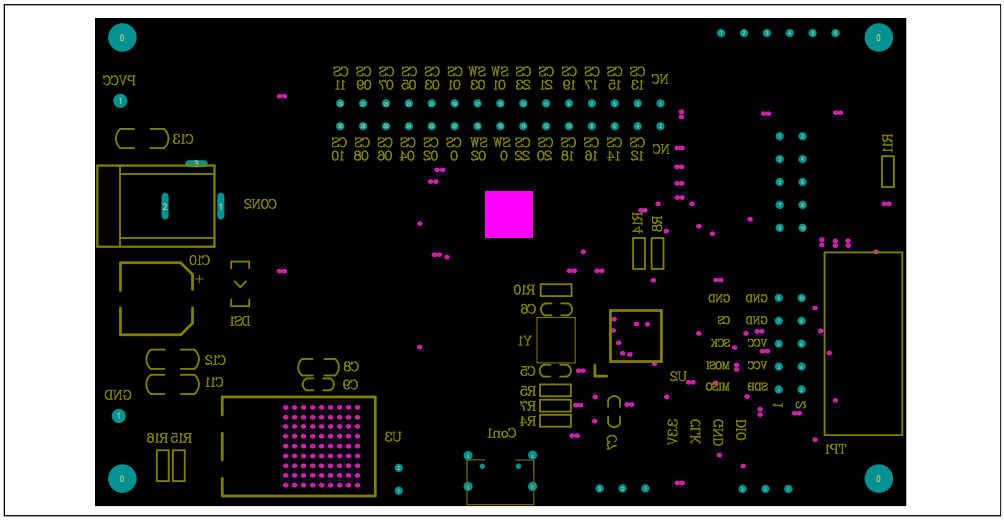

Figure 7: Board Component Placement Guide - Bottom Layer

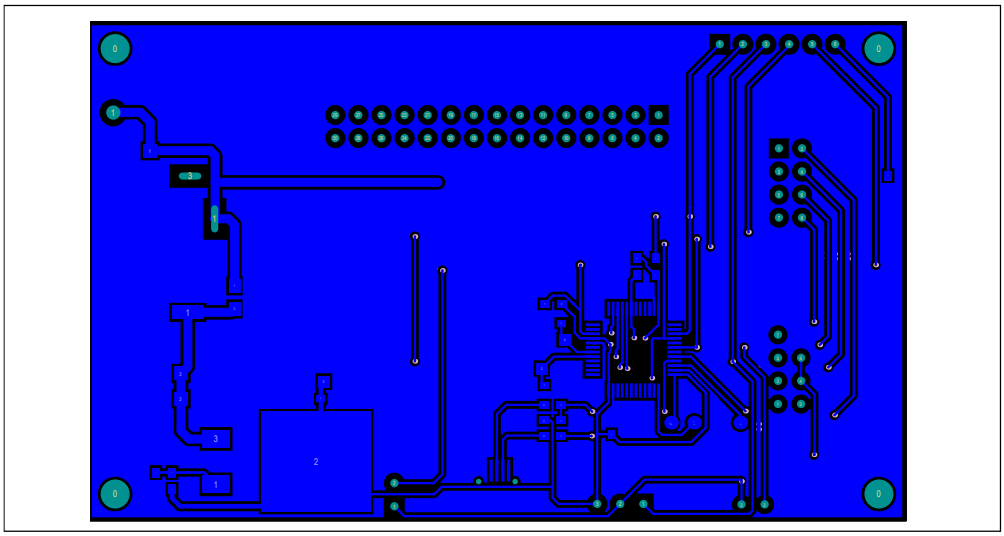

Figure 8: Board PCB Layout - Bottom Layer

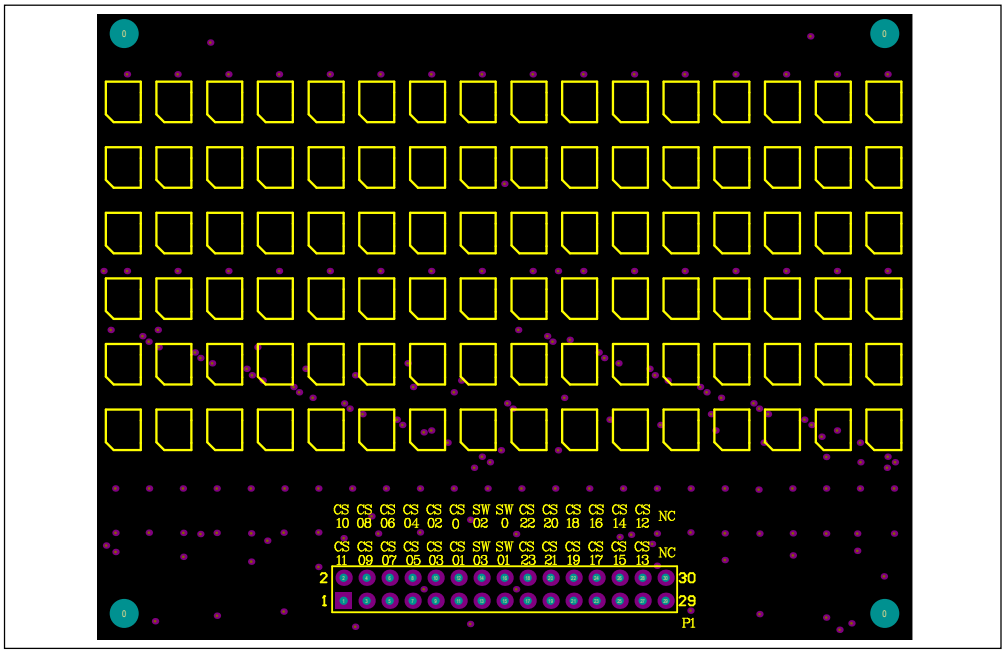

Figure 9: Board Component Placement Guide - Bottom Layer

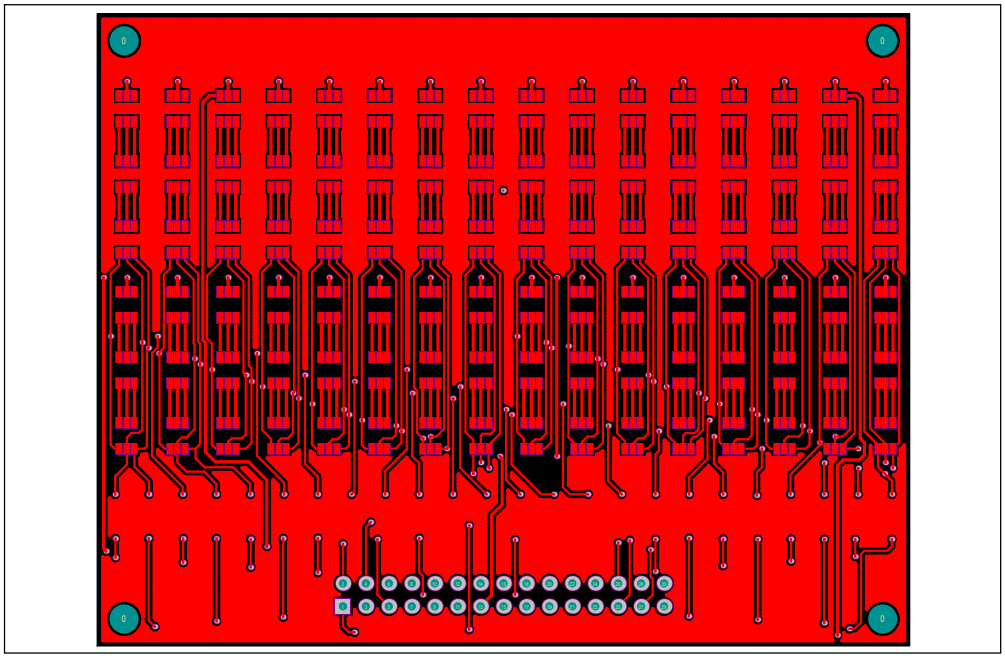

Figure 10: Board PCB Layout - Bottom Layer

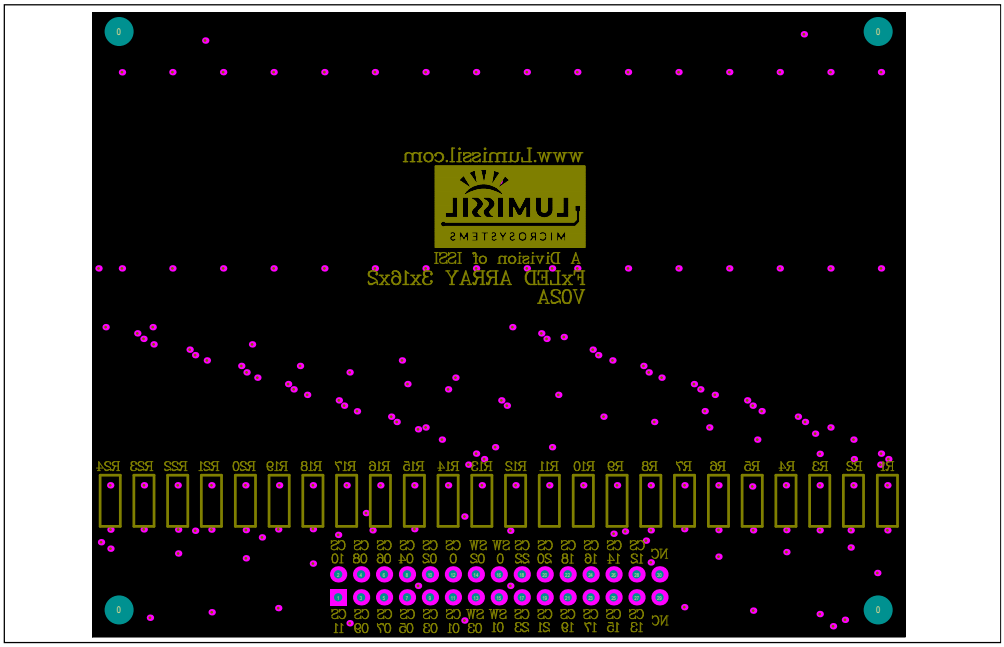

Figure 11: Board Component Placement Guide - Bottom Layer

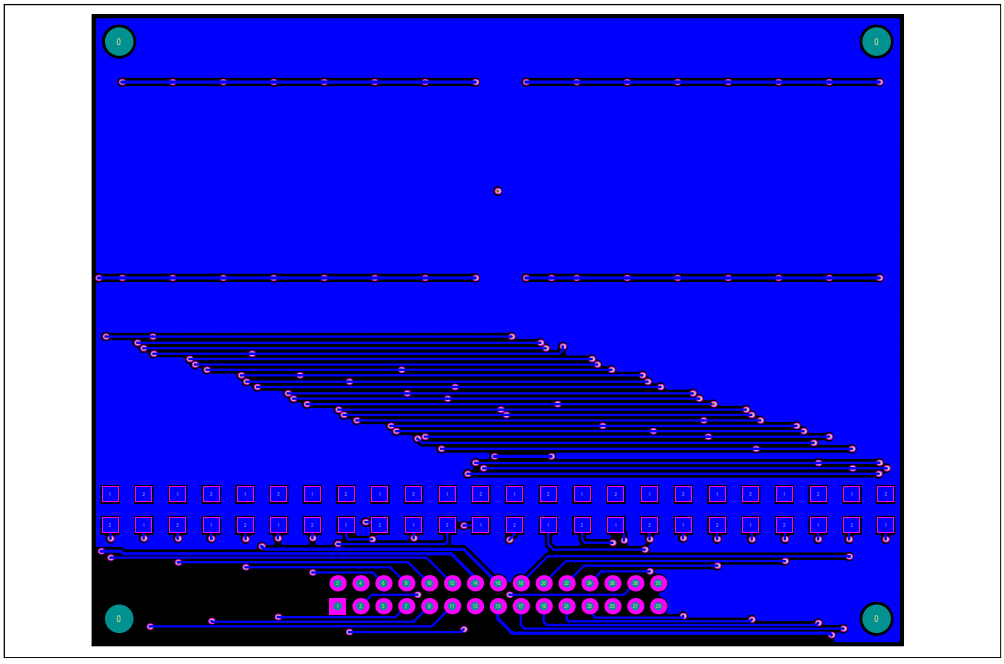

Figure 12: Board PCB Layout - Bottom Layer

REVISION HISTORY

Revision	Detail Information	Data
A	Initial Release	2022.11.04

APPENDIX Ⅰ: IS32FL3749 Arduino Test Code V01A

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

uint8_t PWM_data[48];
uint8_t SL_data[24];
uint8_t FC1_data[6]={0x80,0x00,0x00,0x00,0x00,0x80};
uint8_t FC0_data[6]={0x00,0x3B,0xFF,0xFF,0xFE,0x02};
const int slaveSelectPin = 10;

uint16_t PWM_256_table[]=
{
   0x0000,0x0001,0x0003,0x0006,0x000B,0x0010,0x0017,0x001F,
   0x0029,0x0034,0x0040,0x004E,0x005D,0x006D,0x007F,0x0093,
   0x00A8,0x00BE,0x00D6,0x00F0,0x010B,0x0128,0x0146,0x0166,
   0x0188,0x01AB,0x01D0,0x01F7,0x0220,0x024A,0x0276,0x02A3,
   0x02D2,0x0303,0x0336,0x036B,0x03A1,0x03D9,0x0413,0x044F,
   0x048D,0x04CC,0x050E,0x0551,0x0596,0x05DD,0x0625,0x0670,
   0x06BC,0x070B,0x075B,0x07AD,0x0801,0x0858,0x08B0,0x0909,
   0x0965,0x09C3,0x0A23,0x0A85,0x0AE9,0x0B4E,0x0BB6,0x0C20,
   0x0C8B,0x0CF9,0x0D69,0x0DDB,0x0E4E,0x0EC4,0x0F3C,0x0FB6,
   0x1032,0x10B0,0x1130,0x11B2,0x1236,0x12BC,0x1345,0x13CF,
   0x145C,0x14EA,0x157B,0x160E,0x16A3,0x173A,0x17D3,0x186E,
   0x190C,0x19AB,0x1A4D,0x1AF1,0x1B97,0x1C3F,0x1CE9,0x1D96,
   0x1E44,0x1EF5,0x1FA8,0x205E,0x2115,0x21CE,0x228A,0x2348,
   0x2408,0x24CB,0x258F,0x2656,0x271F,0x27EB,0x28B8,0x2988,
   0x2A5A,0x2B2E,0x2C04,0x2CDD,0x2DB8,0x2E95,0x2F75,0x3056,
   0x313A,0x3221,0x3309,0x33F4,0x34E1,0x35D1,0x36C2,0x37B6,
   0x38AD,0x39A5,0x3AA0,0x3B9D,0x3C9D,0x3D9F,0x3EA3,0x3FA9,
   0x40B2,0x41BD,0x42CB,0x43DB,0x44ED,0x4601,0x4718,0x4831,
   0x494D,0x4A6B,0x4B8B,0x4CAE,0x4DD3,0x4EFA,0x5024,0x5150,
   0x527E,0x53AF,0x54E2,0x5618,0x5750,0x588A,0x59C7,0x5B06,
   0x5C48,0x5D8C,0x5ED2,0x601B,0x6167,0x62B4,0x6404,0x6557,
   0x66AC,0x6803,0x695D,0x6AB9,0x6C18,0x6D79,0x6EDD,0x7043,
   0x71AB,0x7316,0x7483,0x75F3,0x7765,0x78DA,0x7A51,0x7BCB,
   0x7D47,0x7EC6,0x8047,0x81CB,0x8351,0x84D9,0x8664,0x87F2,
   0x8982,0x8B14,0x8CA9,0x8E41,0x8FDB,0x9177,0x9316,0x94B8,
   0x965C,0x9803,0x99AC,0x9B57,0x9D05,0x9EB6,0xA069,0xA21F,
   0xA3D7,0xA592,0xA74F,0xA90F,0xAAD2,0xAC97,0xAE5E,0xB028,
   0xB1F5,0xB3C4,0xB596,0xB76A,0xB941,0xBB1A,0xBCF6,0xBED4,
   0xC0B6,0xC299,0xC47F,0xC668,0xC854,0xCA42,0xCC32,0xCE25,
   0xD01B,0xD213,0xD40E,0xD60C,0xD80C,0xDA0F,0xDC14,0xDE1C,
   0xE026,0xE234,0xE443,0xE656,0xE86B,0xEA82,0xEC9C,0xEEB9,
   0xF0D9,0xF2FB,0xF51F,0xF747,0xF971,0xFB9D,0xFDCD,0xFFFF,

   0xFFFF,0xFDCD,0xFB9D,0xF971,0xF747,0xF51F,0xF2FB,0xF0D9,
   0xEEB9,0xEC9C,0xEA82,0xE86B,0xE656,0xE443,0xE234,0xE026,
   0xDE1C,0xDC14,0xDA0F,0xD80C,0xD60C,0xD40E,0xD213,0xD01B,
   0xCE25,0xCC32,0xCA42,0xC854,0xC668,0xC47F,0xC299,0xC0B6,
   0xBED4,0xBCF6,0xBB1A,0xB941,0xB76A,0xB596,0xB3C4,0xB1F5,
   0xB028,0xAE5E,0xAC97,0xAAD2,0xA90F,0xA74F,0xA592,0xA3D7,
   0xA21F,0xA069,0x9EB6,0x9D05,0x9B57,0x99AC,0x9803,0x965C,
   0x94B8,0x9316,0x9177,0x8FDB,0x8E41,0x8CA9,0x8B14,0x8982,
   0x87F2,0x8664,0x84D9,0x8351,0x81CB,0x8047,0x7EC6,0x7D47,
   0x7BCB,0x7A51,0x78DA,0x7765,0x75F3,0x7483,0x7316,0x71AB,
   0x7043,0x6EDD,0x6D79,0x6C18,0x6AB9,0x695D,0x6803,0x66AC,
   0x6557,0x6404,0x62B4,0x6167,0x601B,0x5ED2,0x5D8C,0x5C48,
   0x5B06,0x59C7,0x588A,0x5750,0x5618,0x54E2,0x53AF,0x527E,
   0x5150,0x5024,0x4EFA,0x4DD3,0x4CAE,0x4B8B,0x4A6B,0x494D,
   0x4831,0x4718,0x4601,0x44ED,0x43DB,0x42CB,0x41BD,0x40B2,
   0x3FA9,0x3EA3,0x3D9F,0x3C9D,0x3B9D,0x3AA0,0x39A5,0x38AD,
   0x37B6,0x36C2,0x35D1,0x34E1,0x33F4,0x3309,0x3221,0x313A,
   0x3056,0x2F75,0x2E95,0x2DB8,0x2CDD,0x2C04,0x2B2E,0x2A5A,
   0x2988,0x28B8,0x27EB,0x271F,0x2656,0x258F,0x24CB,0x2408,
   0x2348,0x228A,0x21CE,0x2115,0x205E,0x1FA8,0x1EF5,0x1E44,
   0x1D96,0x1CE9,0x1C3F,0x1B97,0x1AF1,0x1A4D,0x19AB,0x190C,
   0x186E,0x17D3,0x173A,0x16A3,0x160E,0x157B,0x14EA,0x145C,
   0x13CF,0x1345,0x12BC,0x1236,0x11B2,0x1130,0x10B0,0x1032,
   0x0FB6,0x0F3C,0x0EC4,0x0E4E,0x0DDB,0x0D69,0x0CF9,0x0C8B,
   0x0C20,0x0BB6,0x0B4E,0x0AE9,0x0A85,0x0A23,0x09C3,0x0965,
   0x0909,0x08B0,0x0858,0x0801,0x07AD,0x075B,0x070B,0x06BC,
   0x0670,0x0625,0x05DD,0x0596,0x0551,0x050E,0x04CC,0x048D,
   0x044F,0x0413,0x03D9,0x03A1,0x036B,0x0336,0x0303,0x02D2,
```

```
  0x02A3,0x0276,0x024A,0x0220,0x01F7,0x01D0,0x01AB,0x0188,
  0x0166,0x0146,0x0128,0x010B,0x00F0,0x00D6,0x00BE,0x00A8,
  0x0093,0x007F,0x006D,0x005D,0x004E,0x0040,0x0034,0x0029,
  0x001F,0x0017,0x0010,0x000B,0x0006,0x0003,0x0001,0x0000,

};

void setup()
{
  // put your setup code here, to run once:
  // set the slaveSelectPin as an output:
  pinMode (slaveSelectPin, OUTPUT);
  // initialize SPI:
  SPI.begin();
  SPI.beginTransaction(SPISettings(20000000, MSBFIRST, SPI_MODE0));
  //SPI.setClockDivider(SPI_CLOCK_DIV4);
  SPI.setDataMode(3);
  Init3749();
}

void loop() {
  // put your main code here, to run repeatedly:
  IS32FL3749_mode1();
}

void SPI_Write_Byte(uint8_t SendByte)
{
  digitalWrite(slaveSelectPin, LOW);     // take the SS pin low to select the chip:
  SPI.transfer(SendByte);
  digitalWrite(slaveSelectPin, HIGH);     // take the SS pin high to de-select the chip:
}

void SPI_WriteBuffer(uint8_t* pBuffer, int length_dat, uint8_t Dev_Add)
{
  digitalWrite(slaveSelectPin, LOW);     // take the SS pin low to select the chip:
  SPI.transfer(Dev_Add);     //    send in the address and value via SPI:
  while(length_dat--)
  {
    SPI.transfer(*pBuffer);
    pBuffer++;
  }
  digitalWrite(slaveSelectPin, HIGH);     // take the SS pin high to de-select the chip:
}


// SET the Global Current Control Register in FC0//
void GCC_Control(uint8_t igcc)
{
  FC0_data[1] = ((igcc&0X80)>>7)|(FC0_data[1]&0xFE);
  FC0_data[2] = ((igcc&0X7F)<<1)|(FC0_data[2]&0X01);
  FC0_data[2] = ((igcc&0X80)>>7)|(FC0_data[2]&0xFE);
  FC0_data[3] = ((igcc&0X7F)<<1)|(FC0_data[3]&0X01);
  FC0_data[3] = ((igcc&0X80)>>7)|(FC0_data[3]&0xFE);
  FC0_data[4] = ((igcc&0X7F)<<1)|(FC0_data[4]&0X01);
  SPI_WriteBuffer(FC0_data,6,0x20);//FC0
}

// SET the PWM Register//
void PWM_set(uint8_t ir,uint8_t ir2, uint8_t ig,uint8_t ig2, uint8_t ib, uint8_t ib2)

{
  uint8_t i;
  for(i=0;i<48;i+=6)
  {
    PWM_data[i] = ib;//Bule PWM_H
    PWM_data[i+1] = ib2;//Bule PWM_L
    PWM_data[i+2] = ig;//Green PWM_H
    PWM_data[i+3] = ig2;//Green PWM_L
    PWM_data[i+4] = ir;//Red PWM_H
    PWM_data[i+5] = ir2;//Red PWM_L
  }
  SPI_WriteBuffer(PWM_data,48,0x60);//SW0 PWM
  SPI_WriteBuffer(PWM_data,48,0x62);//SW1 PWM
  SPI_WriteBuffer(PWM_data,48,0x64);//SW2 PWM
  SPI_WriteBuffer(PWM_data,48,0x66);//SW3 PWM
```

```
    SPI_Write_Byte(0x08);//update PWM
}

// SET the Scaling Register//
void SL_set(uint8_t ir, uint8_t ig, uint8_t ib)
{
    uint8_t i;
    for(i=0;i<24;i+=3)
    {
        SL_data[i] = ib;
        SL_data[i+1] = ig;
        SL_data[i+2] = ir;
    }
    SPI_WriteBuffer(SL_data,24,0x40);//SW0 DC
    SPI_WriteBuffer(SL_data,24,0x42);//SW1 DC
    SPI_WriteBuffer(SL_data,24,0x44);//SW2 DC
    SPI_WriteBuffer(SL_data,24,0x46);//SW3 DC
}

void FL3749_FC0_set(uint8_t byte6,uint8_t byte5,uint8_t byte4,uint8_t byte3,uint8_t byte2,uint8_t byte1)
{
    FC0_data[0] = byte6;//BIT41-48
    FC0_data[1] = byte5;
    FC0_data[2] = byte4;
    FC0_data[3] = byte3;
    FC0_data[4] = byte2;
    FC0_data[5] = byte1;//BIT0-7

    SPI_WriteBuffer(FC0_data,6,0x20);//FC0
}
void FL3749_FC1_set(uint8_t byte6,uint8_t byte5,uint8_t byte4,uint8_t byte3,uint8_t byte2,uint8_t byte1)
{
    FC1_data[0] = byte6;//BIT41-48
    FC1_data[1] = byte5;
    FC1_data[2] = byte4;
    FC1_data[3] = byte3;
    FC1_data[4] = byte2;
    FC1_data[5] = byte1;//BIT0-7

    SPI_WriteBuffer(FC1_data,6,0x22);//FC1
}

void Init3749(void)
{
    GCC_Control(0xFF);
    PWM_set(0X00,0X00,0X00,0X00,0X00,0X00);
    SL_set(0x80,0xFF,0xFF);// white balance
    FL3749_FC1_set(0x80,0x00,0x01,0x11,0xC0,0x80);//
}

void IS32FL3749_mode1(void)//
{
    int i;
    SL_set(0x55,0x55,0x55);
    while(1)
    {
    for(i=0;i<512;i++)
    {
        PWM_set(PWM_256_table[i]>>8,PWM_256_table[i],PWM_256_table[i]>>8,PWM_256_table[i],PWM_256_table[i]>>8,PWM_256_table[i]);
        //LED light up and down slowly
        delay(5);
    }
    }

}
```