# TI Designs: TIDA-01591
# New Human Machine Interface With LED Animation Reference Design

## Description

This reference design is a new human machine interface with LED animation and realizes a vivid lighting pattern on the LED ring with one TLC5955 LED driver.

## Resources

| | |
|---|---|
| TIDA-01591 | Design Folder |
| TLC5955 | Product Folder |
| SimpleLink™ MSP432P401R LaunchPad™ Development Kit | Tool Folder |

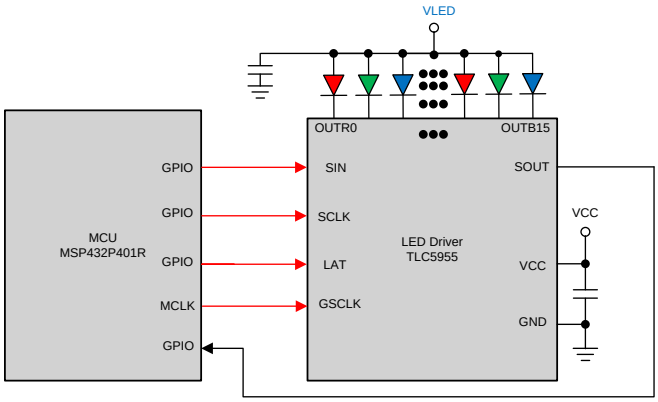
TI E2E™ Community — ASK Our E2E™ Experts

## Features

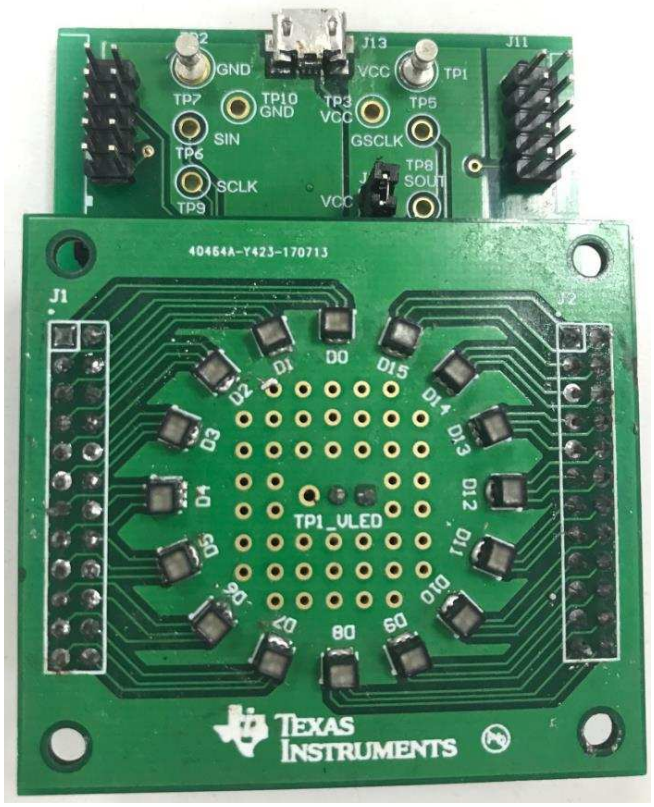
- Vivid Lighting Pattern With LED Ring
- Minimum Device Quantities in Multi RBG LED Modules System

## Applications

- Small Home Appliances
- Major Appliances
- Cooker Hood and Cooking Range
- Power Tools and Garden Tools
- Smart Speakers
- Set-Top Box
- Smart Router






An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

# 1 System Description

This reference design is a new human machine interface (HMI) with light-emitting diode (LED) animation, which realizes the vivid lightings pattern on the LED ring with only one LED driver. In this reference design, TLC5955 linear LED drivers are used to drive 16-pcs RGB LED modules with constant current control. The MSP432P401R LaunchPad sends the control signal to generate the various lighting patterns.

## *1.1 Key System Specifications*

**Table 1. Key System Specifications**

| PARAMETER | SPECIFICATIONS |
|---|---|
| Input voltage range | 3.0 V to 5.5 V |
| Output current | 23.9 mA/channel |
| LED number | 16 RGB |
| LED type | ASMT-YTC2-0AA02 |
| Lighting pattern style | 7 |

# 2 System Overview

## 2.1 Block Diagram

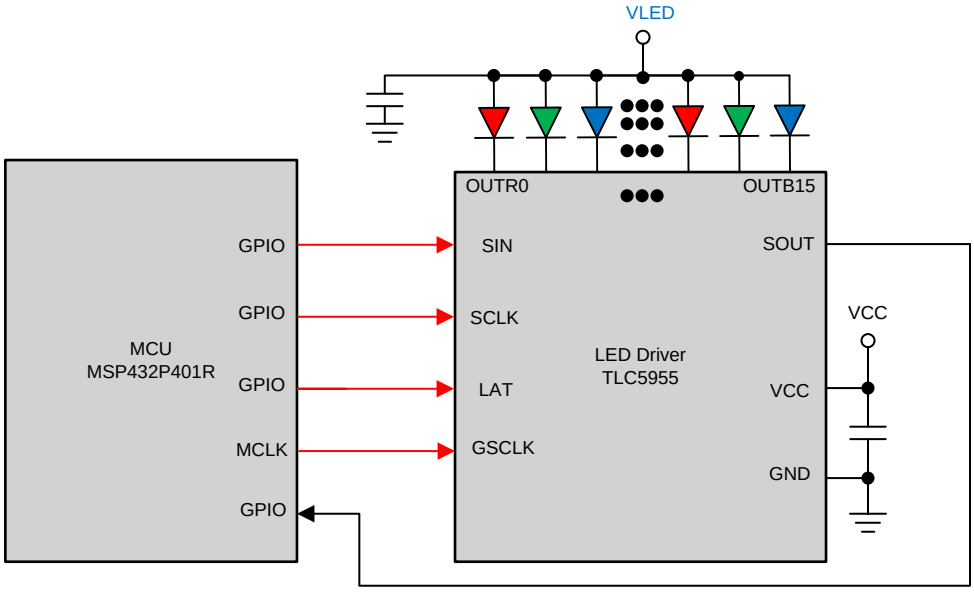





**Figure 1. TIDA-01591 Block Diagram**

## 2.2 Design Considerations

In this reference design, a 48-channel with a 16-bit PWM LED driver (TLC5955) is used to drive a 16-pcs RGB LED module with constant current control and smooth dimming effect. The MSP432P401R LaunchPad sends the control signal to generate the various lighting patterns.

## 2.3 Highlighted Products

The following highlighted products are used in this reference design. The key features for selecting the devices for this reference design are outlined in the following subsections. For the complete details of the highlighted devices, refer to their respective product datasheets.

### 2.3.1 TLC5955—48-Channel, 16-Bit, PWM LED Driver With DC, BC, LED Open-Short Detection, and Internal Current Setting

The TLC5955 is a 48-channel, constant-current sink driver. Each channel has an individually-adjustable, pulse width modulation (PWM), grayscale (GS) brightness control with 65 536 steps and 128 steps of constant-current dot correction (DC). DC adjusts brightness deviation between channels. All channels have a 128-step global brightness control (BC). BC adjusts brightness deviation between the RGB color group. The eight-step maximum current control (MC) selects the maximum output current range for all channels of each color group. GS, DC, BC, and MC data are accessible with a serial interface port. The TLC5955 has two error flags: LED open detection (LOD) and LED short detection (LSD). The error detection results can be read with a serial interface port.

### 2.3.2 SimpleLink™ MSP432P401R LaunchPad™ Development Kit

The SimpleLink MSP432P401R LaunchPad development kit enables one to develop high-performance applications that benefit from low-power operation. This kit features the MSP432P401R LaunchPad, which includes a 48-MHz Arm® Cortex®-M4F, 80-µA/MHz active power and 660-nA RTC operation, 14-bit 1-MSPS differential SAR ADC, and AES256 accelerator. All pins of the MSP-EXP432P401R device are fanned out for easy access. These pins make it easy to plug in 20-pin and 40-pin BoosterPack™ modules that add additional functionality including *Bluetooth*® low energy, Wi-Fi® wireless connectivity, and more.

## 2.4 System Design Theory

One TLC5955 drives 16-pcs RGB LED modules that are designed as LED ring sharp. The MSP432P401R LaunchPad sends the control signal to the TLC5955 to generate the various lighting patterns.

### 2.4.1 System Schematic

Figure 2 and Figure 3 show the schematics for the LED driver board and LED load board, respectively.

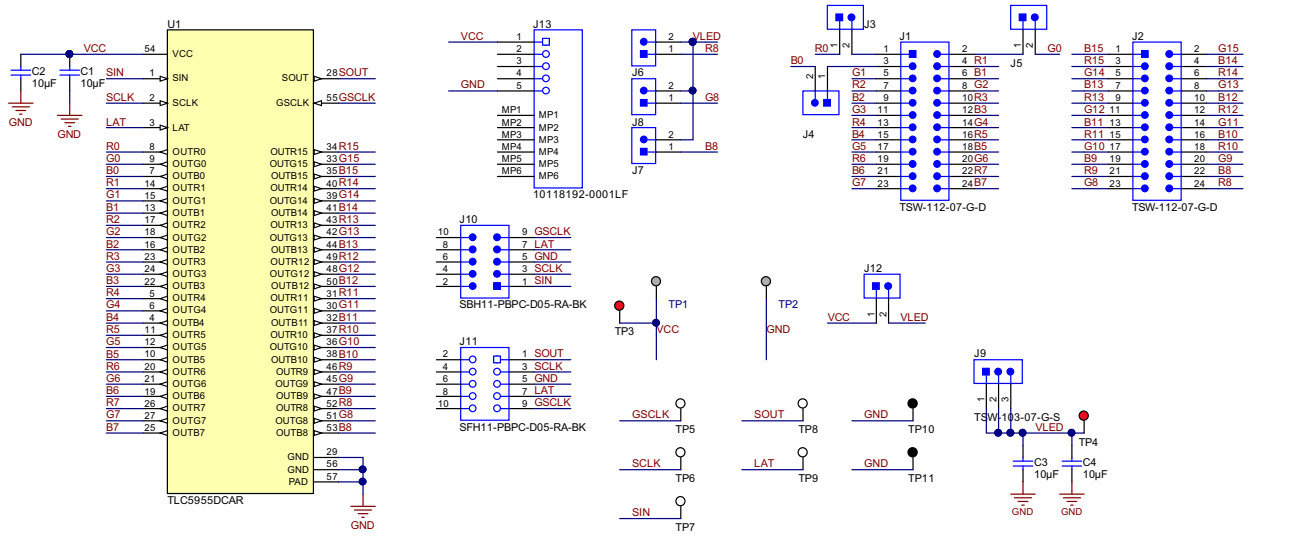





**Figure 2. LED Driver Board Schematic**

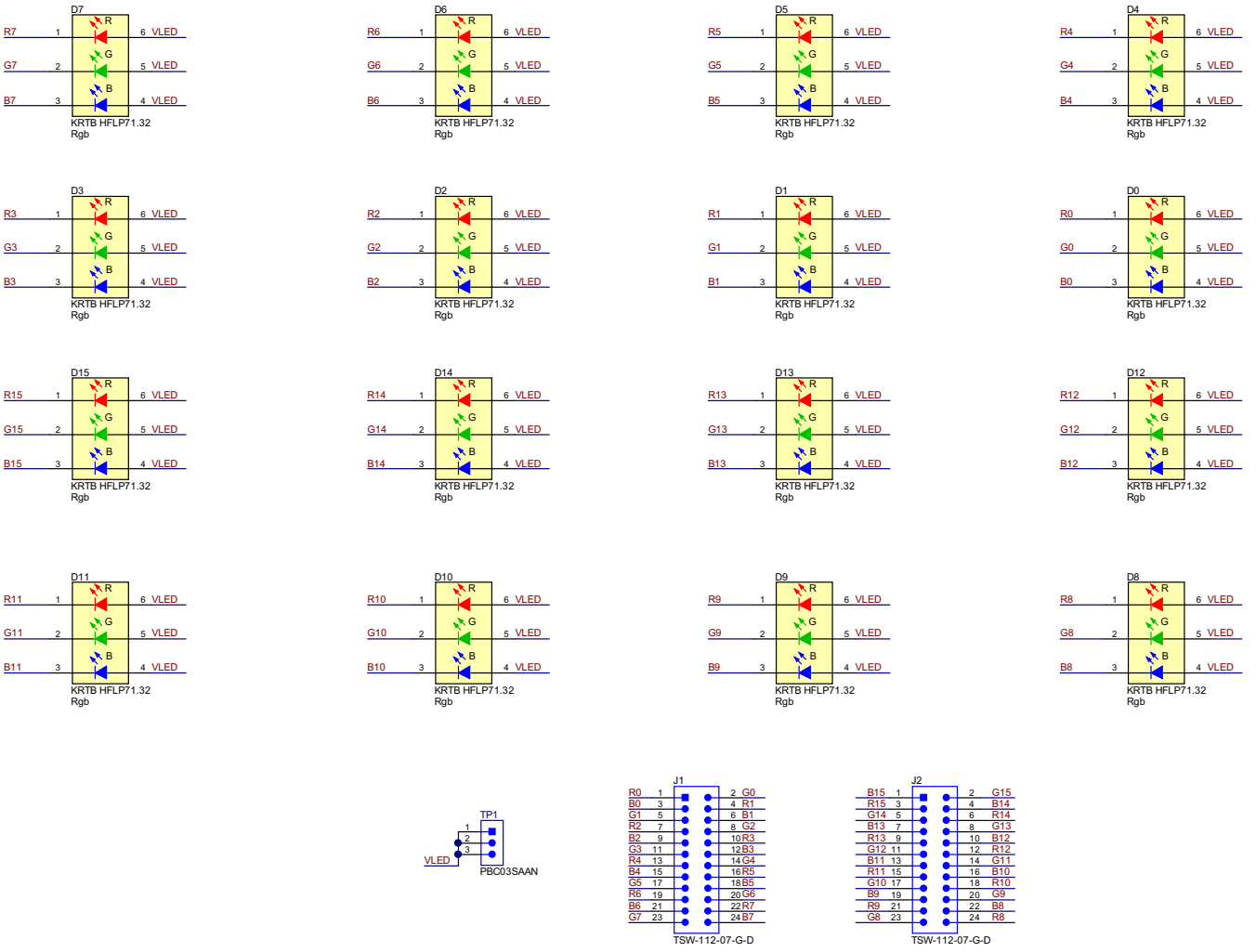





**Figure 3. LED Load Board Schematic**

Table 2 outlines a design example.

**Table 2. Design Performance Specifications**

| PARAMETER | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|
| Input voltage range, $V_{CC}$ | 3 | 3.3 | 5.5 | V |
| LED supply voltage, $V_{LED}$ | 3.3 | 5 | 10 | V |
| Data transfer rate | — | 15 | 25 | MHz |
| LED number | — | — | 48 | — |
| Pattern style | — | — | 7 | — |

**Table 3. MSP432™ LaunchPad™ Pin Connection List**

| MSP432P401R LAUNCHPAD | TLC5955 PCB BOARD |
|---|---|
| P4.3 | GSCLK |
| P3.7 | LAT |
| GND | GND |
| P3.6 | SCLK |
| P3.5 | SIN |
| P5.2 | SOUT |

### 2.4.2 Detailed Design Procedure

To begin the design process, one must decide on a few parameters. The designer must know the following:

- Current of each LED
- LED board layout map
- Lighting pattern

### 2.4.3 Lighting Pattern Design

#### *2.4.3.1 Breathing*

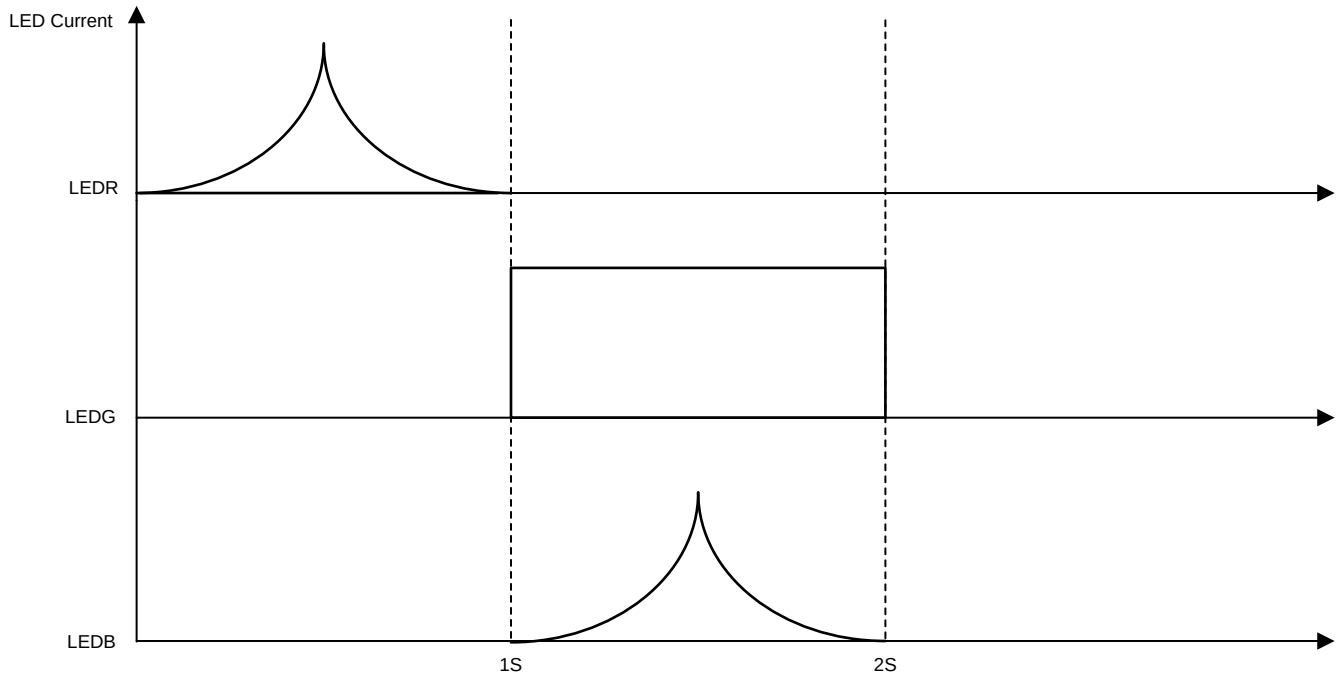



**Figure 4. LED Current Waveform of Breathing**

The sample code is as follows:

```
int GSdata[]={ 0x0000,0x0000,0x0000, //D0 B,G,R
0x0000,0x0000,0x0000, //D1 B,G,R
0x0000,0x0000,0x0000, //D2 B,G,R
0x0000,0x0000,0x0000, //D3 B,G,R
0x0000,0x0000,0x0000, //D4 B,G,R
0x0000,0x0000,0x0000, //D5 B,G,R
0x0000,0x0000,0x0000, //D6 B,G,R
0x0000,0x0000,0x0000, //D7 B,G,R
0x0000,0x0000,0x0000, //D8 B,G,R
```

```
0x0000,0x0000,0x0000, //D9 B,G,R
0x0000,0x0000,0x0000, //D10 B,G,R
0x0000,0x0000,0x0000, //D11 B,G,R
0x0000,0x0000,0x0000, //D12 B,G,R
0x0000,0x0000,0x0000, //D13 B,G,R
0x0000,0x0000,0x0000, //D14 B,G,R
0x0000,0x0000,0x0000, //D15 B,G,R };
//
 int Grey[]={

0,0,0,0,1,1,2,3,4,6,8,10,13,16,19,24,28,33,39,46,53,60,69,78,88,98,110,122,135,149,164,179,196,214
,232,252,273,

295,317,341,366,393,420,449,478,510,542,575,610,647,684,723,764,806,849,894,940,988,1037,1088,1140
,1194,

1250,1307,1366,1427,1489,1553,1619,1686,1756,1827,1900,1975,2051,2130,2210,2293,2377,2463,2552,264
2,2734,

2829,2925,3024,3124,3227,3332,3439,3548,3660,3774,3890,4008,4128,4251,4376,4504,4634,4766,4901,503
8,5177,5319,

5464,5611,5760,5912,6067,6224,6384,6546,6711,6879,7049,7222,7397,7576,7757,7941,8128,8317,8509,870
4,8902,9103,9307,

9514,9723,9936,10151,10370,10591,10816,11043,11274,11507,11744,11984,12227,12473,12722,12975,13230
,

13489,13751,14017,14285,14557,14833,15111,15393,15678,15967,16259,16554,16853,17155,17461,17770,18
083,18399,

18719,19042,19369,19700,20034,20372,20713,21058,21407,21759,22115,22475,22838,23206,23577,23952,24
330,24713,

25099,25489,25884,26282,26683,27089,27499,27913,28330,28752,29178,29608,30041,30479,30921,31367,31
818,32272,

32730,33193,33660,34131,34606,35085,35569,36057,36549,37046,37547,38052,38561,39075,39593,40116,40
643,41175,

41711,42251,42796,43346,43899,44458,45021,45588,46161,46737,47319,47905,48495,49091,49691,50295,50
905,51519,

52138,52761,53390,54023,54661,55303,55951,56604,57261,57923,58590,59262,59939,60621,61308,62000,62
697,63399,64106,64818,65535
                 };// Grey to PWM table
 void LED_Set(int Num,int GS_Red,int GS_Green,int GS_Blue)// Set Num LED's
                 GS
{ GSdata[Num*3]=GS_Red;
GSdata[Num*3+1]=GS_Green;
GSdata[Num*3+2]=GS_Blue; }
void Ini_Dark(void)
{
int i; for(i=0;i<16;i++)
 { LED_Set(i,0,0,0); }
 SendGSData(GSdata);
 }
void Ini_Red(void)
 {
 int i; for(i=0;i<16;i++)
 { LED_Set(i,65535,0,0);//Grey[200] }
 SendGSData(GSdata);
 }
void Ini_Green(void)
 {
 int i; for(i=0;i<16;i++)
```

```
 { LED_Set(i,0,4000,0); }
SendGSData(GSdata);
}
void Ini_Blue(void)
 {
 int i; for(i=0;i<16;i++)
 { LED_Set(i,0,0,4000); }
 SendGSData(GSdata);
 }
 void breath_fresh_Red(void)
 {
 int i=0,j=0; for(i=0;i<240;i++)
 {
for(j=0;j<16;j++)
{ LED_Set(j,Grey[i],0,0); }
 SendGSData(GSdata);
}
 delay_ms(1000);
 for(i=239;i>=0;i--)
 {
for(j=0;j<16;j++)
 { LED_Set(j,Grey[i],0,0); }
 SendGSData(GSdata);
 }
 }
 void breath_fresh_Green(void)
{
 int i=0,j=0; for(i=0;i<240;i++)
{
 for(j=0;j<16;j++)
{ LED_Set(j,0,Grey[i],0); }
 SendGSData(GSdata);
}
 for(i=239;i>=0;i--)
 {
 for(j=0;j<16;j++)
{ LED_Set(j,0,Grey[i],0); }
 SendGSData(GSdata);
 }
 }
void breath_fresh_Blue(void)
 {
 int i=0,j=0; for(i=0;i<240;i++)
 {
for(j=0;j<16;j++)
 { LED_Set(j,0,0,Grey[i]); }
 SendGSData(GSdata);
 }
 for(i=239;i>=0;i--)
 {
 for(j=0;j<16;j++)
 { LED_Set(j,0,0,Grey[i]); }
 SendGSData(GSdata);
 }
 }
void breath_fresh_Yellow(void)
{
 int i=0,j=0; for(i=0;i<240;i++)
{
 for(j=0;j<16;j++)
 { LED_Set(j,Grey[i],Grey[i],0); }
 SendGSData(GSdata);
 }
 for(i=239;i>=0;i--)
{
 for(j=0;j<16;j++)
```

```
 { LED_Set(j,Grey[i],Grey[i],0); }
 SendGSData(GSdata);
 }
 }
 void breath_fresh_Pink(void)
 {
 int i=0,j=0; for(i=0;i<240;i++)
 { for(j=0;j<16;j++)
 { LED_Set(j,Grey[i],0,Grey[i]); }
SendGSData(GSdata);
 }
 for(i=239;i>=0;i--)
 {
 for(j=0;j<16;j++)
 { LED_Set(j,Grey[i],0,Grey[i]); }
SendGSData(GSdata);
 }
 }
 void breath_fresh_Teal(void)
{
 int i=0,j=0; for(i=0;i<240;i++)
 {
 for(j=0;j<16;j++)
{ LED_Set(j,0,Grey[i],Grey[i]); }
SendGSData(GSdata);
 }
 for(i=239;i>=0;i--)
{
 for(j=0;j<16;j++)
 { LED_Set(j,0,Grey[i],Grey[i]); }
 SendGSData(GSdata);
 }
 }
void breath_fresh_White(void)
 {
 int i=0,j=0; for(i=0;i<240;i++)
 {
 for(j=0;j<16;j++)
 { LED_Set(j,Grey[i],Grey[i],Grey[i]); }
SendGSData(GSdata);
}
 for(i=239;i>=0;i--)
{
 for(j=0;j<16;j++)
 { LED_Set(j,Grey[i],Grey[i],Grey[i]); }
SendGSData(GSdata);
 }
 }
void breath_fresh_BreathingWithCorlorChanging(void)
 {
 int i=0,j=0; for(i=0;i<256;i++)
 {
 for(j=0;j<16;j++)
{ LED_Set(j,0,Grey[i],Grey[200]); }
SendGSData(GSdata);
 }
 for(i=255;i>=0;i--)
{
 for(j=0;j<16;j++)
 { LED_Set(j,0,Grey[i],Grey[200]); }
 SendGSData(GSdata);
 }
delay_ms(10); for(i=0;i<256;i++)
 {
for(j=0;j<16;j++)
 { LED_Set(j,Grey[i],0,Grey[200]); }
```

```
SendGSData(GSdata);
}
 for(i=255;i>=0;i--)
{
 for(j=0;j<16;j++)
 { LED_Set(j,Grey[i],0,Grey[200]); }
 SendGSData(GSdata);
 }
delay_ms(10);
 }
```

### 2.4.3.2 Chasing 1

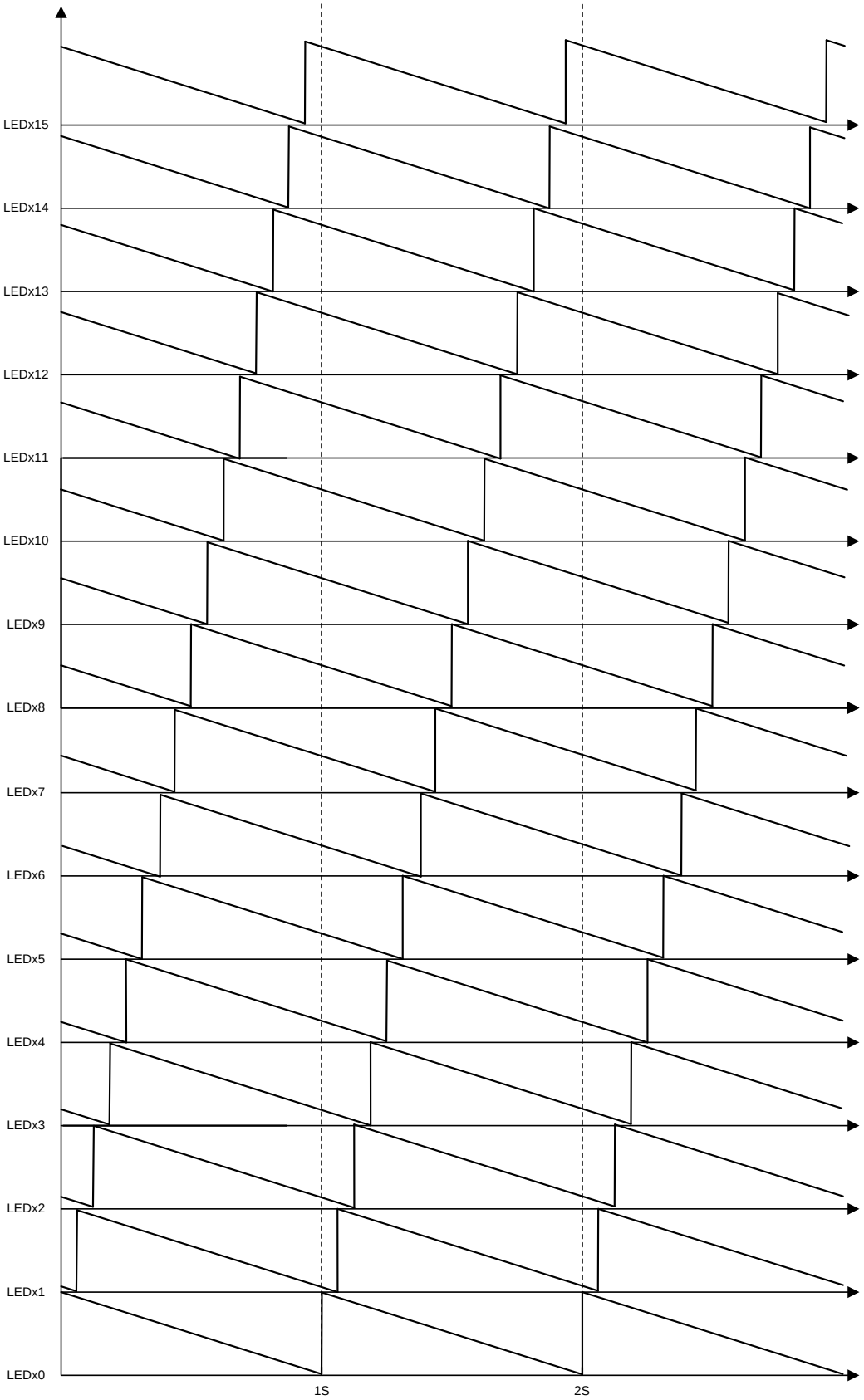



**Figure 5. LED Current Waveform of Single Color Chasing**

The sample code is as follows:

```
void ChasingEffect_Red(void)
{
 int i,j; for(j=0;j<16;j++)
 {
 for(i=0;i<16;i++)
 { LED_Set((i+j)%16,color2pwm[15*i]+1,0,0); }
 SendData(GSdata); delay_ms(25);
 }
 }
}
 void ChasingEffect_Green(void)
 {
int i,j; for
(j=0;j<16;j++)
 {
 for(i=0;i<16;i++)
 { LED_Set((i+j)%16,0,0,color2pwm[15*i]+1); }
 SendData(GSdata); delay_ms(25);
}
 }
 void ChasingEffect_Blue(void)
 {
 int i,j; for(j=0;j<16;j++)
 {
for(i=0;i<16;i++)
 { LED_Set((i+j)%16,0,0,color2pwm[15*i]+1); }
SendData(GSdata); delay_ms(25);
}
}
```

### 2.4.3.3 Chasing 2

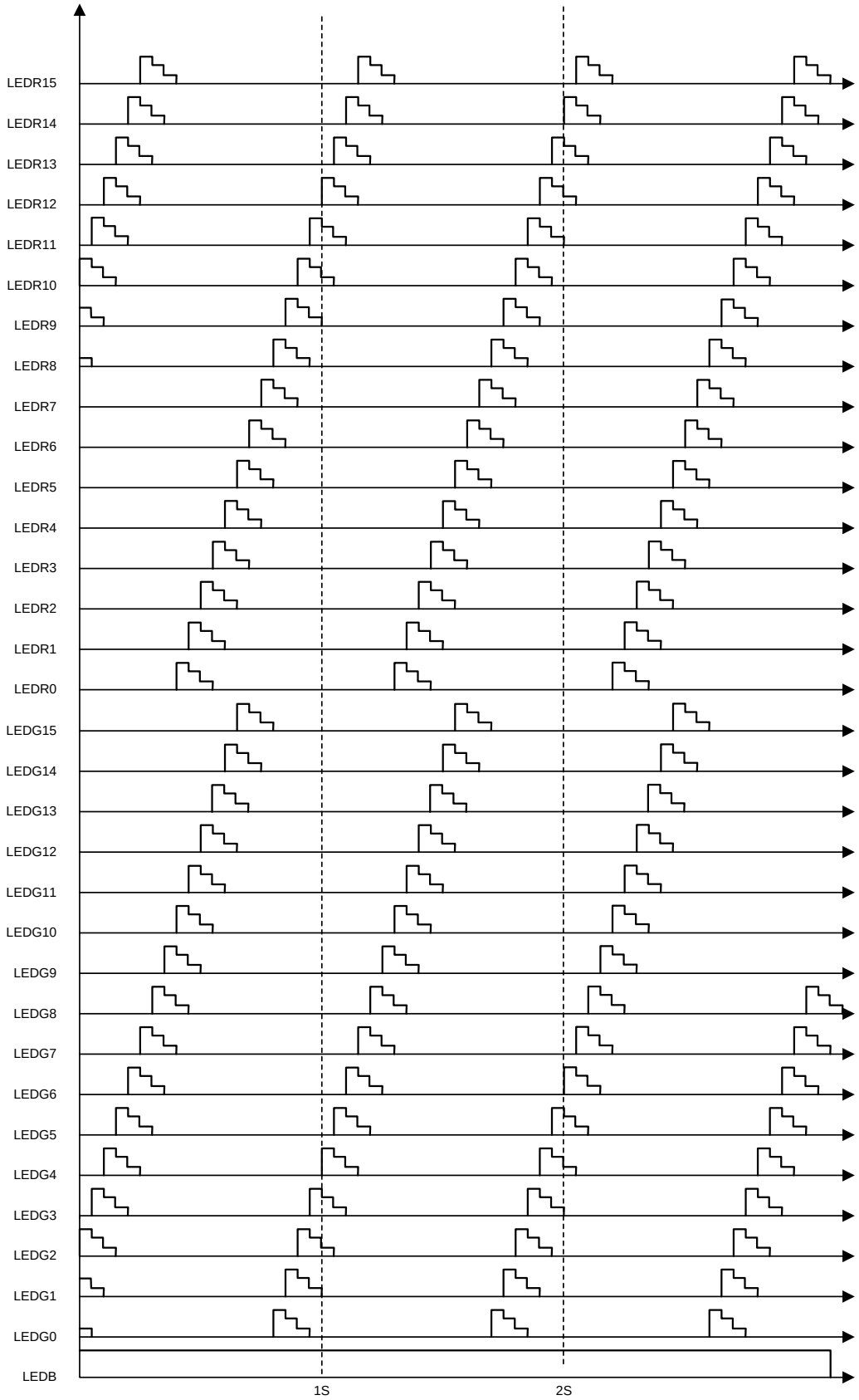



**Figure 6. LED Current Waveform of Multi Color Chasing**

The sample code is as follows:

```
void ChasingEffect_Custom(void)
{
 int i; for(i=0;i<16;i++)
 {
LED_Set(i%16,1034,1034,0);
 LED_Set((i+1)%16,1130,1130,0);
 LED_Set((i+2)%16,1338,1338,0);
LED_Set((i+3)%16,0,0,2000);
LED_Set((i+4)%16,0,0,2000);
 LED_Set((i+5)%16,0,0,2000);
 LED_Set((i+6)%16,0,0,2000);
 LED_Set((i+7)%16,0,0,2000);
LED_Set((i+8)%16,1034,512,256);
 LED_Set((i+9)%16,2048,1024,512);
 LED_Set((i+10)%16,4095,2048,1024);
 LED_Set((i+11)%16,0,0,2000);
 LED_Set((i+12)%16,0,0,2000);
 LED_Set((i+13)%16,0,0,2000);
 LED_Set((i+14)%16,0,0,2000);
 LED_Set((i+15)%16,0,0,2000);
SendData(GSdata);
 delay_ms(50);
 }
}
```

### 2.4.3.4 Chasing 3

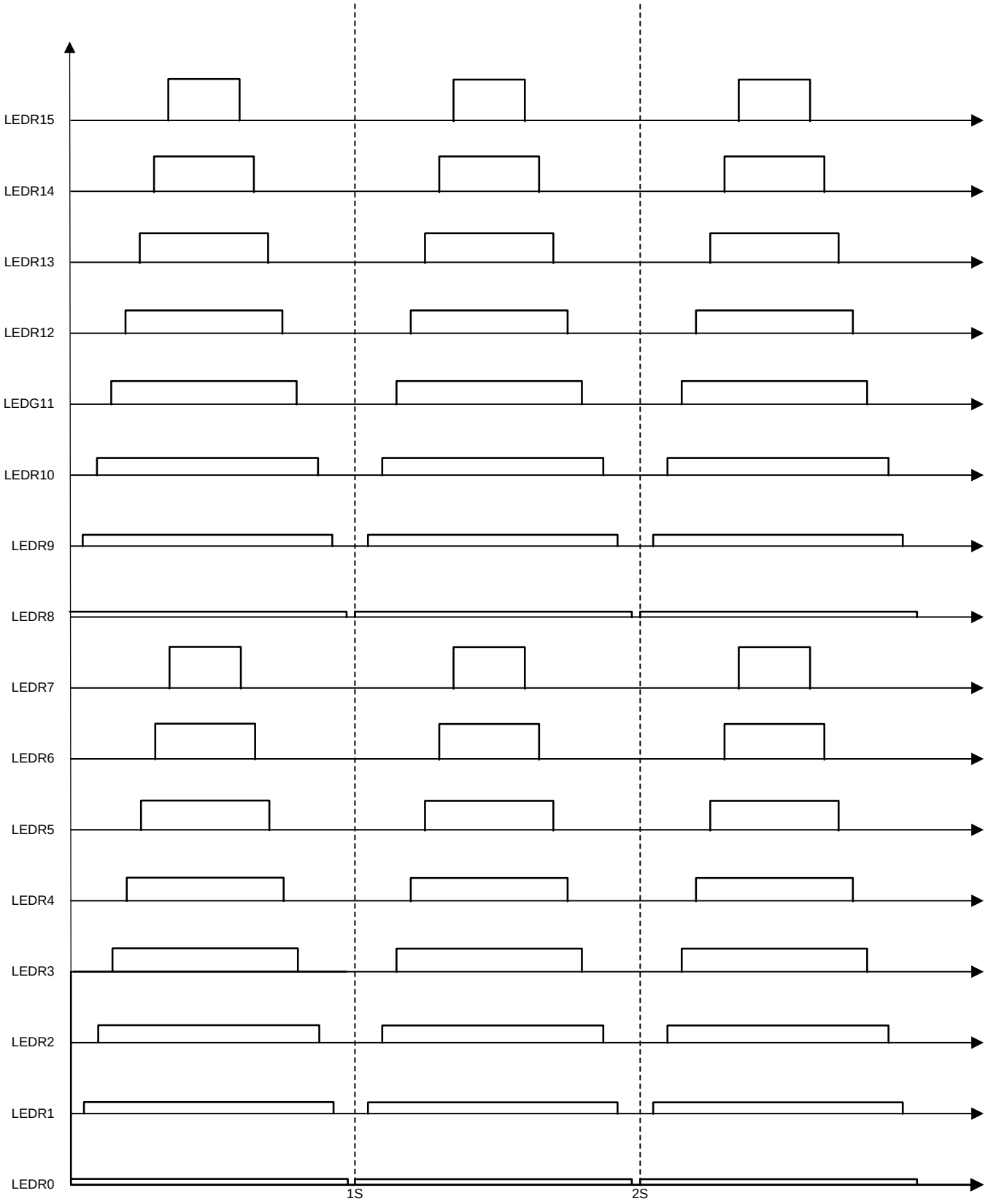



**Figure 7. Current Wave of Custom Chasing**

The sample code is as follows:

```
void ChasingEffectFade_Red(void)
 {
int i; for(i=0;i<16;i++)
 {
LED_Set(i%16,color2pwm[6*(i%16)],0,0);
LED_Set((i+1)%16,color2pwm[8*(i%16)],0,0);
 LED_Set((i+2)%16,color2pwm[10*(i%16)],0,0);
 LED_Set((i+3)%16,color2pwm[12*(i%16)],0,0);
 LED_Set((i+4)%16,color2pwm[14*(i%16)],0,0);
LED_Set((i+5)%16,color2pwm[16*(i%16)],0,0);
LED_Set((i+6)%16,0,0,0);
 LED_Set((i+7)%16,0,0,0);
 LED_Set((i+8)%16,color2pwm[6*(i%16)],0,0);
 LED_Set((i+9)%16,color2pwm[8*(i%16)],0,0);
 LED_Set((i+10)%16,color2pwm[10*(i%16)],0,0);
LED_Set((i+11)%16,color2pwm[12*(i%16)],0,0);
 LED_Set((i+12)%16,color2pwm[14*(i%16)],0,0);
 LED_Set((i+13)%16,color2pwm[16*(i%16)],0,0);
 LED_Set((i+14)%16,0,0,0);
 LED_Set((i+15)%16,0,0,0);
 SendData(GSdata); delay_ms(30);
 }
 for(i=15;i>=0;i--)
 {
LED_Set(i%16,color2pwm[6*(i%16)],0,0);
LED_Set((i+1)%16,color2pwm[8*(i%16)],0,0);
 LED_Set((i+2)%16,color2pwm[10*(i%16)],0,0);
 LED_Set((i+3)%16,color2pwm[12*(i%16)],0,0);
 LED_Set((i+4)%16,color2pwm[14*(i%16)],0,0);
 LED_Set((i+5)%16,color2pwm[16*(i%16)],0,0);
 LED_Set((i+6)%16,0,0,0);
 LED_Set((i+7)%16,0,0,0);
LED_Set((i+8)%16,color2pwm[6*(i%16)],0,0);
 LED_Set((i+9)%16,color2pwm[8*(i%16)],0,0);
LED_Set((i+10)%16,color2pwm[10*(i%16)],0,0);
 LED_Set((i+11)%16,color2pwm[12*(i%16)],0,0);
LED_Set((i+12)%16,color2pwm[14*(i%16)],0,0);
 LED_Set((i+13)%16,color2pwm[16*(i%16)],0,0);
 LED_Set((i+14)%16,0,0,0);
 LED_Set((i+15)%16,0,0,0);
SendData(GSdata); delay_ms(30);
}
 }
```

### 2.4.3.5 Water Drop

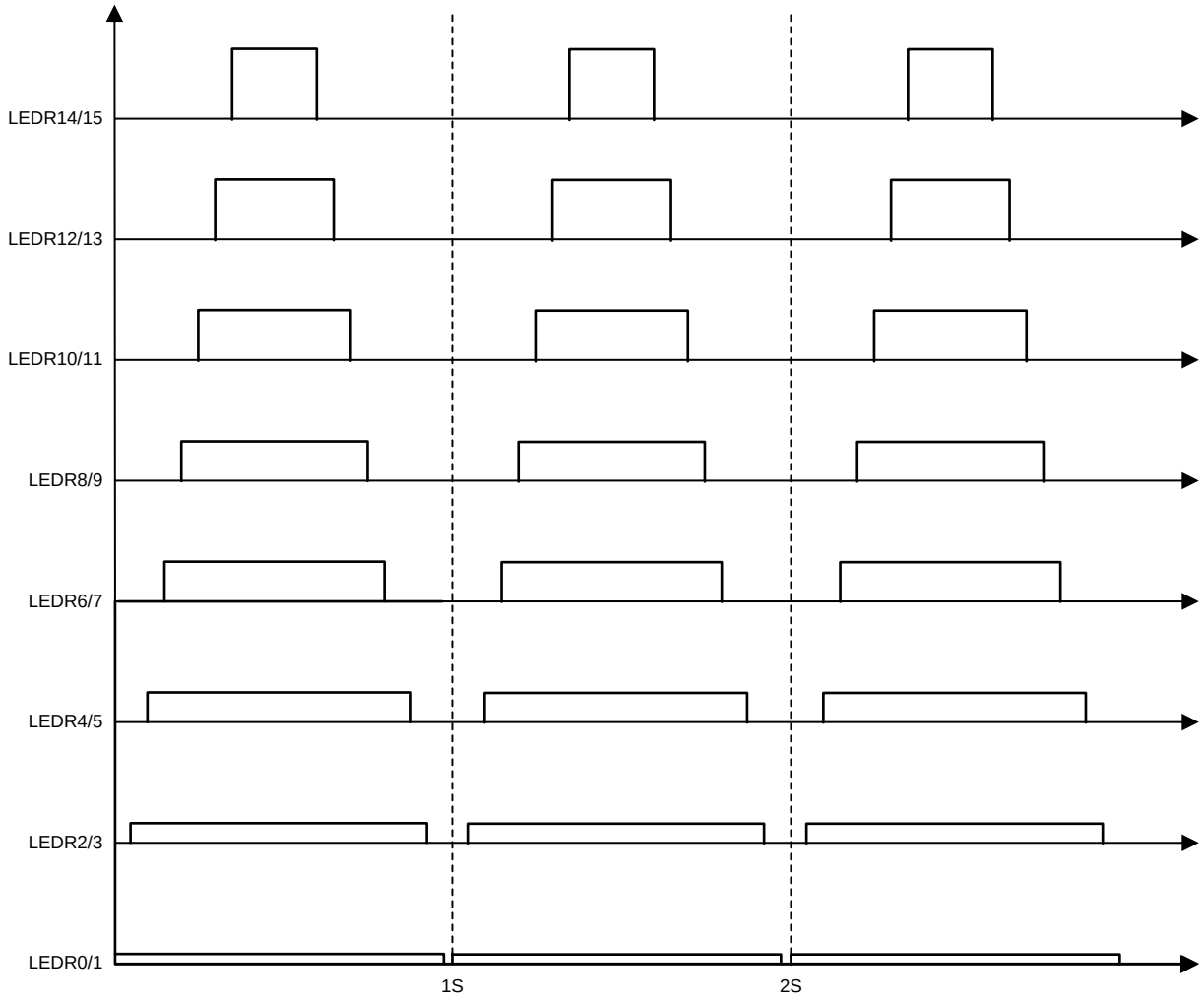



**Figure 8. Current Wave of Water Drop**

The sample code is as follows:

```
void Drop_effect_Red(void)
 {
int i; for (i=0;i<16;i++)
 { LED_Set(i,color2pwm[(i+1)*16-1],0,0); SendData(GSdata); delay_ms(20); }
for(i=15;i>=0;i--)
 { LED_Set(i,0,0,0); SendData(GSdata); delay_ms(20); }
}
```

### 2.4.3.6 Others

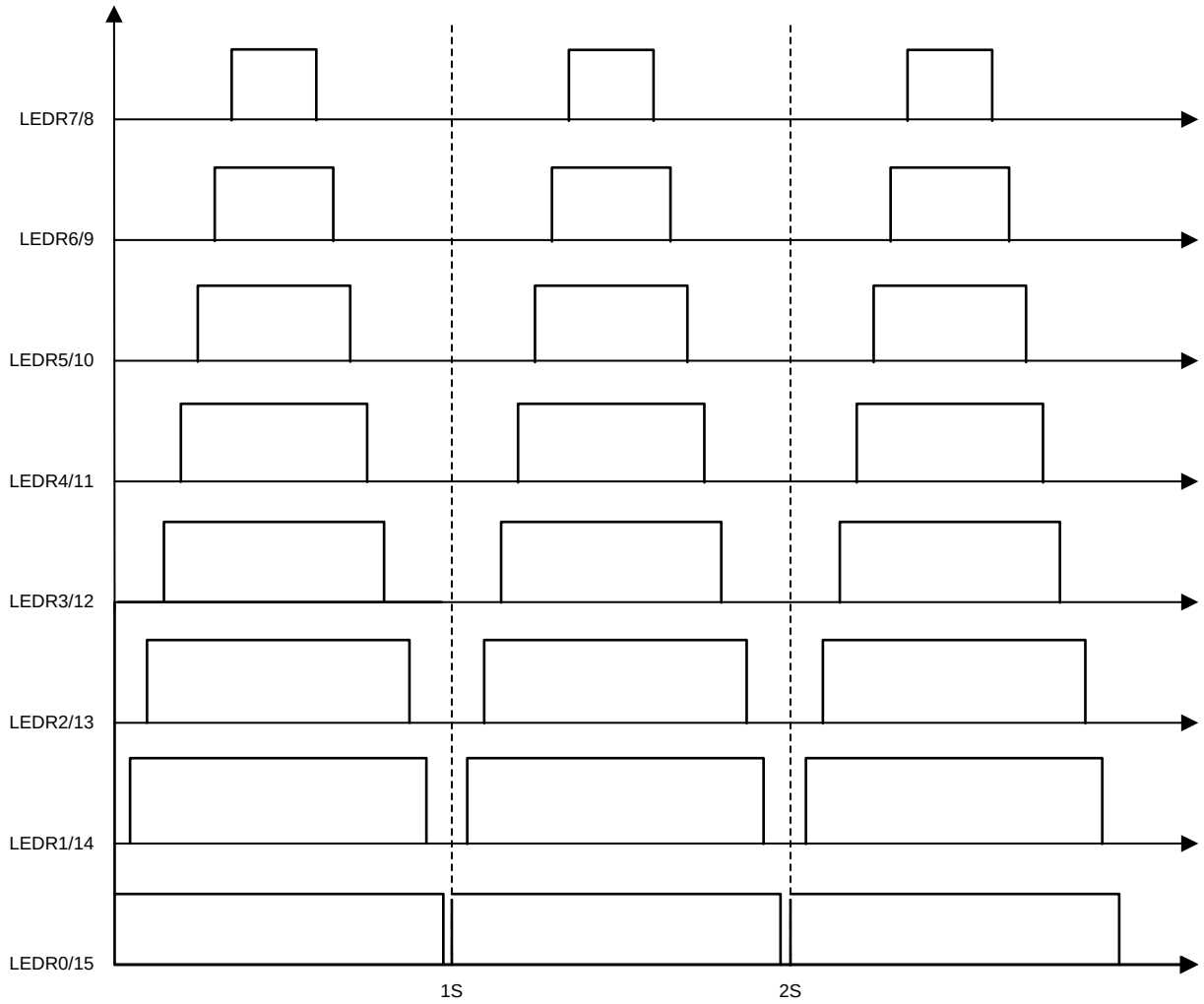



**Figure 9. Current Wave of Open Door**

The sample code is as follows:

```
 void Open_Door_Effect(void)
 {
int i; for(i=0;i<8;i++)
 {
 LED_Set(i,0,color2pwm[250],0);
 LED_Set(15-i,0,color2pwm[250],0);
 SendData(GSdata); delay_ms(20);
 }
 delay_ms(1000);
for(i=7;i>=0;i--)
 {
 LED_Set(i,0,0,0);
LED_Set(15-i,0,0,0);
 SendData(GSdata);
 delay_ms(20);
}
 delay_ms(1000);
}
```

# 3 Hardware, Software, Testing Requirements, and Test Results

## 3.1 Required Hardware and Software

### 3.1.1 Hardware

After download the firmware to the MSP432 LaunchPad through Code Composer Studio™ (CCS), connect a 3.3-V DC supply to the LED board input connector (TP1), plus in the USB port. Then the design board can run the pattern.

### 3.1.2 Software

Install the CCS Integrated Development Environment (IDE) in the PC.

## 3.2 Testing and Results

### 3.2.1 Test Setup

Download the sample code through TI.com. Open the project in CCS and download to the MSP432 LaunchPad.

### 3.2.2 Test Results

All the effects are shown as expected.

# 4 Design Files

## 4.1 Schematics

To download the schematics, see the design files at TIDA-01591.

## 4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-01591.

## 4.3 PCB Layout Recommendations

- Place the decoupling capacitor near the VCC and GND terminals.
- Route the GND pattern as widely as possible for large GND currents. The maximum GND current is approximately 1.53 A.
- Route between the LED cathode side and the device OUTXn as short and straight as possible to reduce wire inductance.
- Connect the PowerPAD™ to the GND layer because the pad is not internally connected to GND and must be connected to a heat sink layer to reduce device temperature.

### 4.3.1 Layout Prints

To download the layer plots, see the design files at TIDA-01591.

## 4.4 Altium Project

To download the Altium project files, see the design files at TIDA-01591.

## 4.5 Gerber Files

To download the Gerber files, see the design files at TIDA-01591.

## 4.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-01591.

# 5 Software Files

To download the software files, see the design files at TIDA-01591.

# 6 Related Documentation

1. Texas Instruments, *TLC5955 48-Channel, 16-Bit, PWM LED Driver with DC, BC, LED Open-Short Detection, and Internal Current Setting Data Sheet*
2. Texas Instruments, *MSP432P401R SimpleLink™ Microcontroller LaunchPad™ Development Kit (MSP-EXP432P401R) User's Guide*

## 6.1 Trademarks

SimpleLink, LaunchPad, E2E, BoosterPack, MSP432, Code Composer Studio, PowerPAD are trademarks of Texas Instruments.
Arm, Cortex are registered trademarks of Arm Limited (or its subsidiaries).
*Bluetooth* is a registered trademark of Bluetooth SIG.
Wi-Fi is a registered trademark of Wi-Fi Alliance.
All other trademarks are the property of their respective owners.

# 7 About the Authors

**XING SU** is an application engineer at Texas Instruments for the LED Driver product group.

**YIJIANG DU** is an application engineer at Texas Instruments for the LED Driver product group.

## IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265