

Digital Addressable Lighting Interface (DALI) Implementation Using MSP430 Value Line Microcontrollers

MSP430

ABSTRACT

The Digital Addressable Lighting Interface (DALI) was defined in IEC 60929 and has been updated in IEC 62386. One of the main reasons for this update was the inclusion of the LED device type.

This application uses the existing TPS62260LED-338 EVM and a level translation board to implement a DALI LED device type control gear. The microcontroller found on the TPS62260LED-338 EVM is the MSP430F2131. The MSP430F2131 performs the communication with the CPU, while the timer resources are used for controlling the fade rate and the LED intensity.

This application report has been expanded to support MSP430 value line devices. For more information, see [Appendix B](#).

Related code and additional information are available for download from www.ti.com/lit/zip/slaa422.

Contents

1	Introduction	2
2	DALI Implementation with the MSP430F2131	2
3	Clock Configuration	6
4	DALI Software Functions	6
5	References	8
Appendix A	Command Sets	9
Appendix B	DALI Implementation With the MSP430G2xx2 Value Line Devices	11

List of Figures

1	Manchester Encoding	2
2	Schematic of DALI Interface Boards	2
3	Logarithmic Nature of Increasing Intensity	3
4	LED Driver Circuit.....	5
5	Software File Structure	11
6	DALI Demo Using MSP430G2xx2 Value Line Device.....	12

List of Tables

1	Timer_A3 PWM Configurations	5
2	Flash Variables and Offset in Information Memory.....	8
3	Supported Commands	9
4	Unsupported Commands	10
5	Supported Special Commands	10
6	dali_demo_hw.h Hardware Definitions	11
7	Pinout Differences Between MSP430F2131 and MSP430G2xx2 Implementations	12

1 Introduction

The DALI protocol is a half-duplex digital communication composed of forward and backward frames. Forward frames consist of one start bit, one address byte, one data byte, and two stop bits. The backward frame (the response after reception of a query or memory command in the forward frame) consists of one start bit, one data byte, and two stop bits.

DALI uses Manchester encoding. The bi-phase levels are depicted in [Figure 1](#). The voltage of the interface power supply can vary from 11.5 V to 22.5 V per the standard.

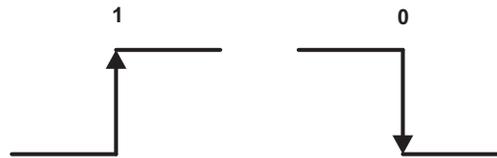


Figure 1. Manchester Encoding

2 DALI Implementation with the MSP430F2131

The MSP430F2131 provides two timers, WDT+ and Timer_A3. The Timer_A3 updates the PWMs that drive the LEDs. The WDT+ is used to fade the LED intensity per the DALI standard and manage the special communication requirements of configuration commands.

2.1 Hardware Interface

Opto-isolators translate signals between the DALI physical layer and the MSP430 as shown in [Figure 2](#).

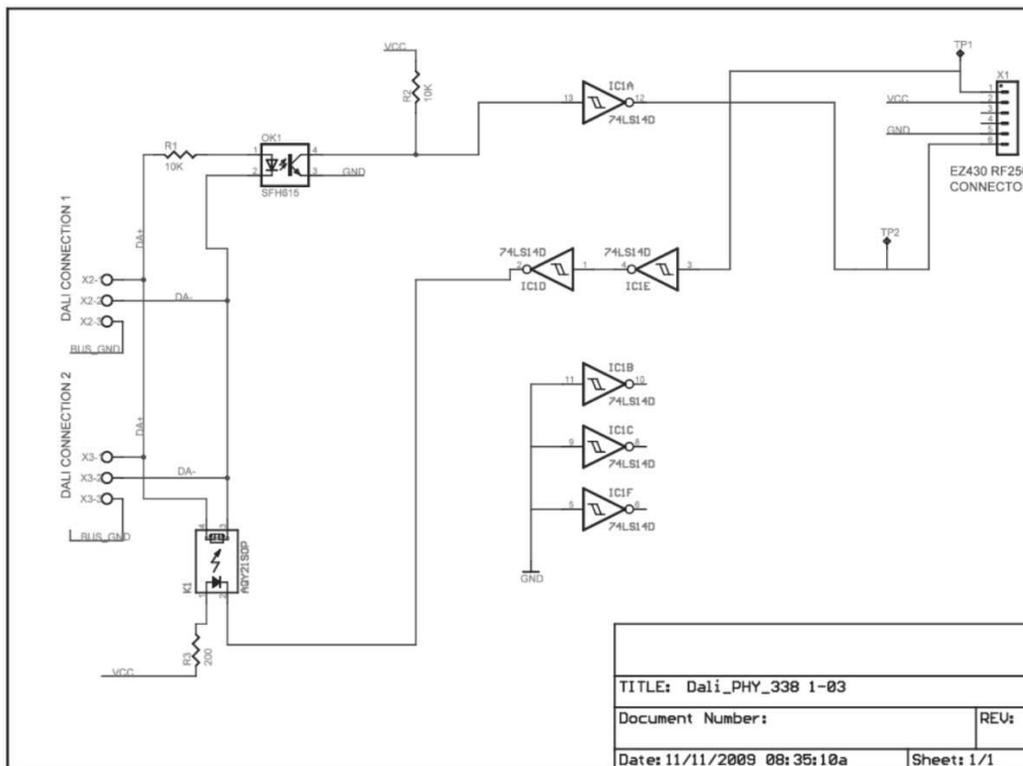


Figure 2. Schematic of DALI Interface Boards

2.2 Lighting Control

Depending upon the application, the lighting can be controlled with either analog or digital modules within the MSP430. In this example, the lighting application is controlled with a PWM generated from the Timer_A3 module within the MSP430. A simple look-up table of 255 PWM values is stored in memory. The values in this table represent the 254 arc power levels defined in the standard.

2.2.1 Logarithmic Intensities

The DALI standard defines the variable Actual Level as one byte in RAM that represents the current output level of the control gear ⁽¹⁾. The standard also defines a logarithmic relationship between the different levels from 1 to 254. 1 represents 0.1% illumination and 254 represents 100% illumination. The difference between successive levels is a constant 2.8%. The standard also defines a "physical minimum level". With an update rate of 1.6 kHz, the physical minimum level that still provided a 'good' minimum level for both the red and green LEDs is 90. For the PWM driving the LED, this relates to about a 1.17% duty cycle.

Figure 3 shows PWM values for the timer and the logarithmic relationship.

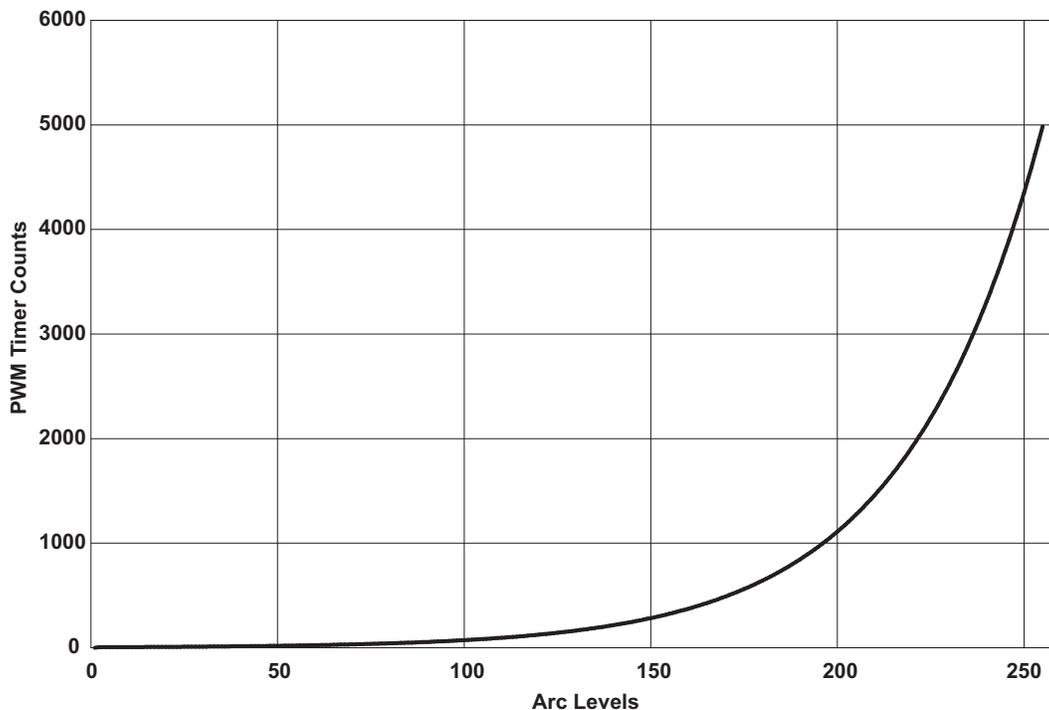


Figure 3. Logarithmic Nature of Increasing Intensity

⁽¹⁾ In the event of a fault or lamp failure, the Actual Level represents the virtual power level.

2.2.2 Timer_A3 Module Example

The values shown in [Figure 3](#) are stored in the flash array 'LED'. The array is indexed with the actual_level variable and applied to the Timer_A3 module when an update is requested.

The Timer_A3 module is configured to run without CPU intervention. DALI commands that request a change to the lighting intensity or the fade routine exercise a call back function that enables a Timer_A3 ISR. When the service request occurs, the Timer_A3 PWMs are updated, and then the ISR is disabled.

```

/** PWM Timer Initialization *****/
P1OUT &= ~BIT0;           // Turn off LEDs
TACTL = TASSEL_2+ID_0+MC_0+TACLK; // Timer clock = SMCLK = 8MHz
TACCR0=5000;             // Define 1.6Khz frequency
TACCR1=LED[power_on_level]; // This power level will be overwritten
TACCR2=LED[power_on_level]; // by the reception of a power command
                               // within 0.6 seconds. After this time
                               // the device will automatically go to
                               // the power on level if no command is
                               // received.
TACCTL1 = CM_0+CCIS_2+OUTMOD_3; // Set on TACCRx, Reset on TACCR0
TACCTL2 = CM_0+CCIS_2+OUTMOD_3;
/** *****/

TACTL |= MC_1; // start Timer_A (up mode)
__enable_interrupt(); // enable maskable interrupts

TI_DALI_Transaction_Loop();
/* should never return */
while(1);
} //End Main

void TI_DALI_Update_Callback()
{
    TACCTL1 |= CCIE;
}

void TI_DALI_Idle_Callback(void)
{
    __no_operation();
}

/** *****/
/* Timer_A Interrupt Service Routine: */
/** *****/
#pragma vector=TIMER1_VECTOR
__interrupt void ISR_TimerA(void)
{
    if(actual_level==OFF)
    {
        /* turn off LEDs */
        P1OUT &= ~BIT0;
    }
    else
    {
        //--- PWM signal generation
        P1OUT |= BIT0;
        TACCR1=LED[actual_level]; //
        TACCR2=LED[actual_level]; //
    }
    TACCTL1 &= ~CCIE;
}
    
```

The update rate of the LED PWM is a function of the Timer_A3 clock source and the count period. In this application example, the timer period is 5000 counts. With an SMCLK of 8 MHz, the PWM frequency is 1.6 kHz. One of the implications of choosing 5000 counts is the inability to maintain the 2.8% step size across all 254 arc power levels. Table 1 shows the valid ranges for different PWM periods.

Table 1. Timer_A3 PWM Configurations

Clock Source (MHz)	Period (Counts)	PWM (kHz)	Valid Range Supporting 2.8% Step Size
8	5000	1.6	72 to 255
8	10000	0.8	45 to 255
8	20000	0.4	24 to 255

2.2.3 LED Driver Hardware

A portion of the TPS62260LED-338 schematic is shown in Figure 4. A more complete description of the EVM can be found in reference 1 (see Section 5).

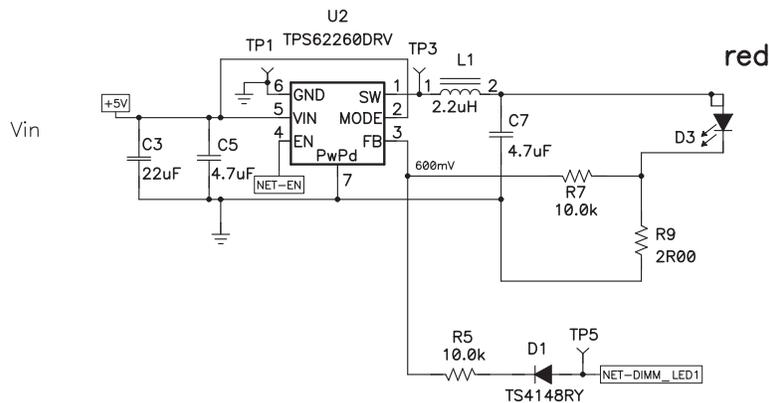


Figure 4. LED Driver Circuit

3 Clock Configuration

The system clock configuration supports the DALI communication, the DALI defined fade control for the control gear, the various timers required for DALI compliance, and the flash controller operation. The key variables that can be used to configure the clock system are MCLK_KHZ, SMCLK_KHZ, WDT_CYCLES, and FADE_INTERVAL.

```

/** Application Definitions *****/
#define PHYS_MIN_LEVEL    90
#define MCLK_KHZ          8000      // Application MHZ frequency
#define SMCLK_KHZ         8000

#define WDT_CYCLES        8192

// #define DIV_SMCLK_32768 (WDT_MDLY_32) /* SMCLK/32768 */
#define FADE_INTERVAL     (WDT_MDLY_8) /* SMCLK/8192 */
// #define DIV_SMCLK_512  (WDT_MDLY_0_5) /* SMCLK/512 */
// #define DIV_SMCLK_64   (WDT_MDLY_0_064) /* SMCLK/64 */
    
```

3.1 DALI Communication

The Manchester encoding/decoding is performed by the CPU. The bit timing definitions are based upon the selection of the CPU (MCLK) frequency.

3.2 Fade Control

The DALI standard defines 15 different fade times and rates that control how long the LED takes to change from the current power level to the target power level. The watchdog timer plus (WDT+) interval mode accomplishes the fade implementation. The fastest fade rate is 358 steps/second, while the shortest fade time (fast fade time) is 25 ms. To achieve the 25-ms fade time across 254 levels, the WDT+ interval needs to be 98 μ s and run at a frequency of 10.16 MHz ⁽¹⁾.

In this application example, the WDT+ interval is approximately 1 ms (976 Hz \approx 8 MHz / 8192). This supports all of the possible fade times and fade rates but supports only the fast fade times 11 through 27.

4 DALI Software Functions

4.1 TI_DALI_Init()

Inputs: FWKEY (this is the password key for the flash controller module)

Outputs: None

The initialization function has four tasks: configure the GPIO to support the DALI interface, maintain the DALI variables stored in Flash, initialize the RAM variables (from the values stored in Flash), and configure the WDT+.

4.2 TI_DALI_Transaction_Loop()

Inputs: None

Outputs: None

This function initiates the monitoring of the DALI bus and does not return to the main (calling) application.

⁽¹⁾ This frequency is not possible with the WDT+ interval mode, because the smallest interval is 64 cycles, which requires an input clock of 650 MHz. As an alternative, the WDT+ can be replaced with a Timer_A module.

4.3 *TI_DALI_Update_Callback()*

Inputs: None

Outputs: None

The update callback function can be used by the main application to determine if a DALI command has requested a change in the arc power level or if the fade routine has requested an arc power level change.

In this application, the update to the timer pwm is done by simply enabling a timer ISR.

```
void TI_DALI_Update_Callback()
{
    TACCTL1 |= CCIE;
}
```

4.4 *TI_DALI_Idle_Callback()*

Inputs: None

Outputs: None

The DALI standard requires that the time between two forward frames is greater than $22 \times TE$ or 9.17 ms. After a DALI command has been processed there is an approximate idle time of 7 ms in which the processor can service other tasks. The WDT+ (interval mode) is still running during this time, and the global variable `idle_time` is used to monitor the time in 1-ms increments.

The DALI standard also requires that the time between a forward frame and a backward frame (response from the control gear) be between $7 \times TE$ and $22 \times TE$. If the tasks within the call back function are less than the required $7 \times TE$, then the `TI_Dali_Transaction_Loop` function provides the appropriate delay before transmitting the response.

4.5 *TI_DALI_Flash_Update()*

Inputs: FWKEY (this is the password key for the flash controller module)

Outputs: None

This function copies the required variables from RAM into the information memory space (flash). This function is intended to be called only once before the device stops operating (for example, in a power-down or power-loss situation). [Table 2](#) shows the values that are stored in flash memory. This table also shows how segments B, C, and D are divided into two 32-byte sections. Upon power up, the MSP430 retrieves the values from information memory and place them into RAM variables. As mentioned previously, the `TI_DALI_Init()` function maintains the information memory segments used and establishes the address pointer where data is to be written.

Table 2. Flash Variables and Offset in Information Memory

Name	Offset
Power On Level	[0]
System Failure Level	[1]
Minimum Level	[2]
Maximum Level	[3]
Fade Rate	[4]
Fade Time	[5]
Short Address	[6]
Group 0 through 7	[7]
Group 8 through 15	[8]
Scene 0 through 15	[9-24]
Random Address	[25-27]
Fast Fade Time	[28]
Failure Status	[29]
Operating Mode	[30]
Dimming Curve	[31]

The clock of the flash controller is set to 333 kHz and 32 flash byte writes are performed. Assuming that each write requires 30 flash clock cycles, then the cumulative programming time is 2.88 ms ($32 \times 30 / 333 \text{ kHz}$). In the event of a power loss, the MSP430 must first detect the power loss (with the Comparator_A+ module, for example), and the hardware on the board must maintain a V_{CC} of 2.2 V for at least 3 ms ⁽¹⁾.

Additionally, the segments are divided into two spaces and, therefore, two updates occur to any one segment before it is erased. The cumulative programming time for each segment is 4.5 ms before an erase of the segment is performed. The cumulative programming time does not exceed the 10 ms of the MSP430F2xx family.

The typical flash endurance of the MSP430 is 10^5 cycles. Dividing the information memory space into six equal parts across three segments allows for six times the number of write/erase cycles. The flash_update_request variable is provided to indicate if any of the nonvolatile variables have been updated and a flash update is required.

5 References

1. TPS62260LED-338, Three-Color LED Driver Evaluation Module ([SLVU240](#))
2. MSP430x2xx Family User's Guide ([SLAU144](#))
3. IEC 62383-102
4. IEC 62383-107

⁽¹⁾ $(3 \text{ ms}) / (2.9 - 2.2); I_{CC} = 2.5 \text{ mA} (f_{DCO} = 8 \text{ MHz}); C = i \times (\Delta t / \Delta V) = 10.7 \text{ }\mu\text{F}$

Appendix A Command Sets

A.1 Supported Command Set

Table 3. Supported Commands

Command Number	Description
0	Off
1	Up (fade up for 200 ms at Fade Rate)
2	Down (fade down for 200 ms at Fade Rate)
3	Step Up (increment arc power level)
4	Step Down (decrement arc power level)
5	Recall Max Level
6	Recall Min Level
7	Step Down and Off (decrement arc power level if at min level turn off)
8	On and Step Up (increment arc power level if off then turn on)
9	Enable DAPC Sequence (fade rate is controlled directly)
10-15	Reserved
16-31	Go to Scene (0-15)
32	Reset
33	Store the Actual Level in the DTR
34-41	Reserved
42-47	Store the DTR as Max Level – Fade Rate
48-63	Reserved
64-79	Store the DTR as Scene (0-15)
80-95	Remove from Scene (0-15)
96-111	Add to Group (0-15)
112-127	Remove from Group (0-15)
128	Store the DTR as Short Address
130-143	Reserved
144-155	Query Commands
158-159	Reserved
160-165	Query Commands Continued
166-175	Reserved
176-196	Query Commands Continued
197	Read Memory Location
198-223	Reserved
228	Store DTR as Fast Fade Time
229-236	Reserved
237-241	Query Commands Continued
252-255	Query Commands Continued

A.2 *Unsupported Command Set*

This code example does not support the bank 1 memory map described within the standard. Therefore, commands 129 and 275 are not supported. The memory bank 1 option and write command requires the use of the flash controller, which does not support the minimum time between DALI forward frames (the erase and write time for a segment exceed the minimum data transmission time which would cause the control gear to possibly miss commands).

Commands 224 through 227 and 242 through 251 are not required, but the command 240 must indicate that these features are not supported.

Table 4. Unsupported Commands

Command Number	Description
129	Enable Write Memory
224	Reference System Power
225	Enable Current Protector
226	Disable Current Protector
227	Select Dimming Curve
242-251	Query Commands (Open, Short, Change in Load, Thermal Overload, Thermal Shutdown, Measurement fail, Current Protector Active or Enabled)

A.3 *Supported Special Command Set*

Table 5. Supported Special Commands

Command Number	Description	Supported
256	Terminate	Yes
257	Data Transfer Register (DTR)	Yes
258	Initialize	Yes
259	Randomize	Yes
260	Compare	Yes
261	Withdraw	Yes
262-263	Reserved	
264-266	Search Address H,M,L	Yes
267	Program Short Address	Yes
268	Verify Short Address	Yes
269	Query Short Address	Yes
270	Physical Selection	Yes
271	Reserved	
272	Enable Device Type 6	Yes
273	Data Transfer Register 1	Yes
274	Data Transfer Register 2	Yes
275	Write Memory Location	No
276-383	Reserved	

Appendix B DALI Implementation With the MSP430G2xx2 Value Line Devices

This application report has been expanded to support the MSP430 value line devices. Given the high compatibility between MSP430 devices, the software migration from MSP430F2131 to MSP430G2xx2 is very simple.

Software was modified by adding the file `dali_demo_hw.h`, which provides an additional level of hardware abstraction and allows for more modularity and an easier migration between platforms. [Table 6](#) lists the definitions that are included in this file.

Table 6. dali_demo_hw.h Hardware Definitions

Function	Relevant Definitions
TPS62260 Enable pin	TPS62260_ENABLE_PxOUT TPS62260_ENABLE_BIT
PWM1 Timer pin	PWM1_BIT
PWM2 Timer pin	PWM2_BIT
DALI RX pin	DALI_RX_PxIN DALI_RX_PxIES DALI_RX_PxIFG DALI_RX_BIT DALI TX pin DALI_TX_BIT
GPIO Initialization	GPIO_INIT()
Timer driving PWMs	TIMER_VECTOR TAxCCR0 TAxCCR1 TAxCCR2 TAxCCTL1 TAxCCTL2 TAxCTL
Unused (dummy) interrupt vectors	DUMMY_VECTORS

The software package (<http://www.ti.com/lit/zip/slaa422>) has been modified to include not only the source code for the project, but also IAR and CCS project files. [Figure 5](#) shows the file structure of the provided software:

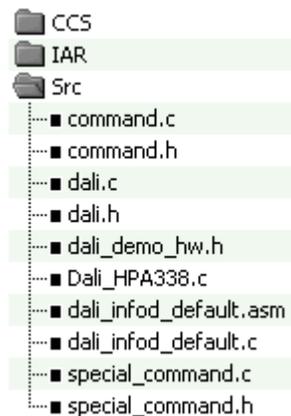


Figure 5. Software File Structure

On the Hardware side, DALI can be easily implemented and demonstrated using the Launchpad MSP-EXP430G2 board, and the same hardware interface shown in [Section 2.1](#).

P1.6 which also has the functionality of TimerA0.1 and is connected to an LED in the Launchpad, can be used to demonstrate light control capabilities, or if desired, an external hardware can be connected to the device.

[Table 7](#) shows the hardware connection differences between the MSP430F2131 using TPS62260LED-338 EVM and the MSP430G2xx2 using MSP-EXP430G2:

Table 7. Pinout Differences Between MSP430F2131 and MSP430G2xx2 Implementations

Function	MSP430F2131	MSP430G2xx2
TPS62260 Enable	P1.0	P1.1
PWM1	P1.2/TA1	P1.6/TA0.1
PWM2	P1.3/TA2	P1.4/TA0.2
DALI RX	P2.0	P2.0
DALI TX	P2.1	P2.1

[Figure 6](#) shows the implementation of a demo using the MSP-EXP430G2 controlling the TPS62260LED-338 EVM LEDs (with previous removal of MSP430F2131) and communicating through the DALI bus using the DALI PHY board described in [Section 2.1](#).

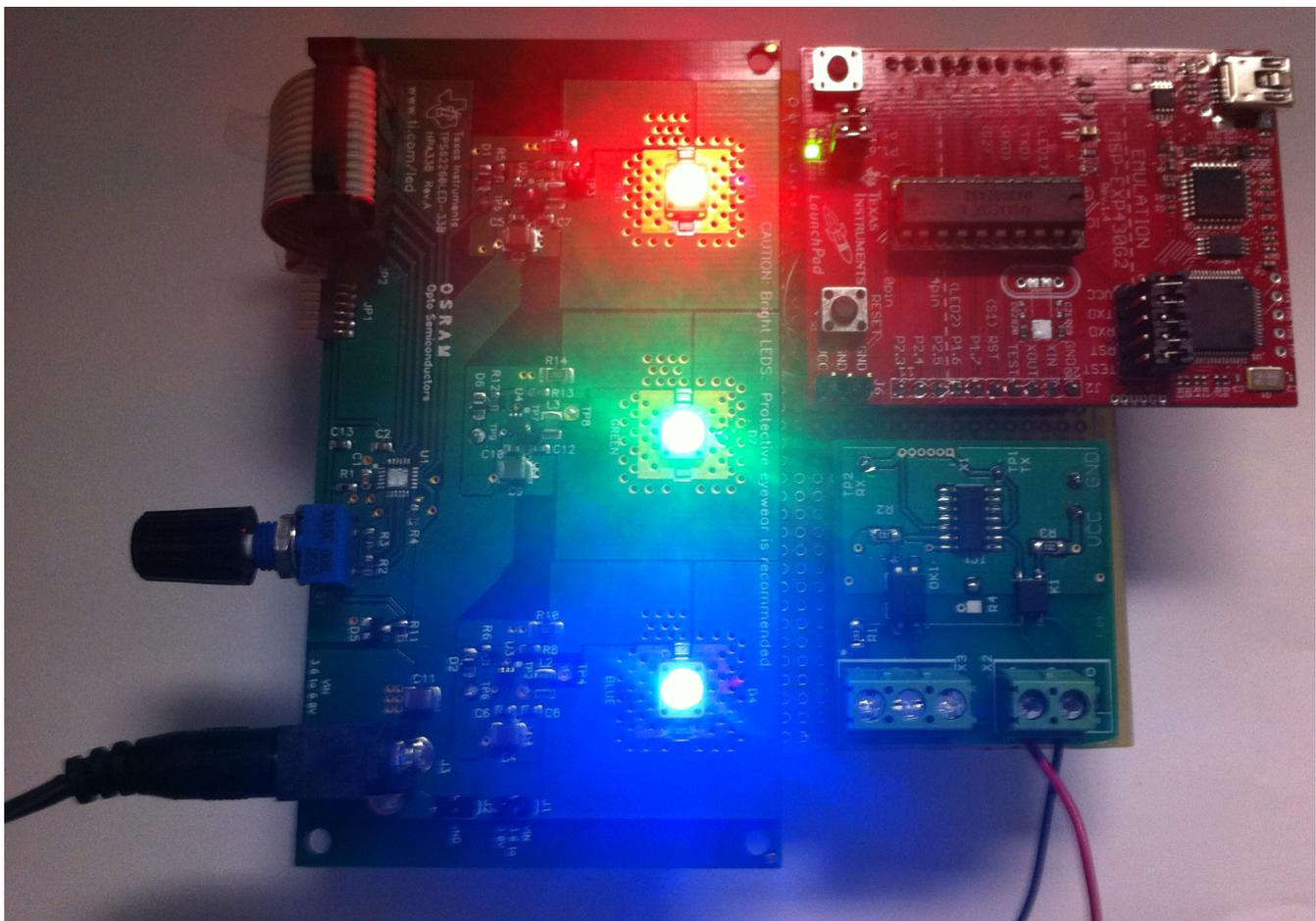


Figure 6. DALI Demo Using MSP430G2xx2 Value Line Device

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components which meet ISO/TS16949 requirements, mainly for automotive use. Components which have not been so designated are neither designed nor intended for automotive use; and TI will not be responsible for any failure of such components to meet such requirements.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com