User's Guide

LP589x and TLC698x Sample Code

ABSTRACT

This document describes the preparation and usage of the sample code for LP589x and TLC698x device families when paired with a MSP-EXP430F5529LP. Following the instructions provided for setup, the installed code lights up the LEDs on the EVM.

Table of Contents

List of Figures

List of Tables

Trademarks

LaunchPad™ and Code Composer Studio™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

The sample code showcases the ability to light up the LEDs on the LP5890EVM, LP5891EVM, TLC6983EVM, and TLC6984EVM. Each EVM has its own sample code. However, the only difference is in the file `led_driver.h` to select the used LED driver IC. This helps the user to be able to light up the EVM without any modification to the sample code.

There are two modes in the code: animation and simple test. The animation mode is selected by default. Section 3.3 describes how to change between the modes. In the animation mode two frames are used to scroll left, right, up, and down and to fade in and fade out according to a predefined sequence. The first frame is a Texas Instruments logo of 32x32 RGB pixels and the second frame a rainbow pattern of 48x32 RGB pixels. This means that not always the full frame is shown on the LED display of the EVM. Examples of this can be seen in Section 3.4. It is outside the scope of this document to explain how the frames in the sample code are generated.

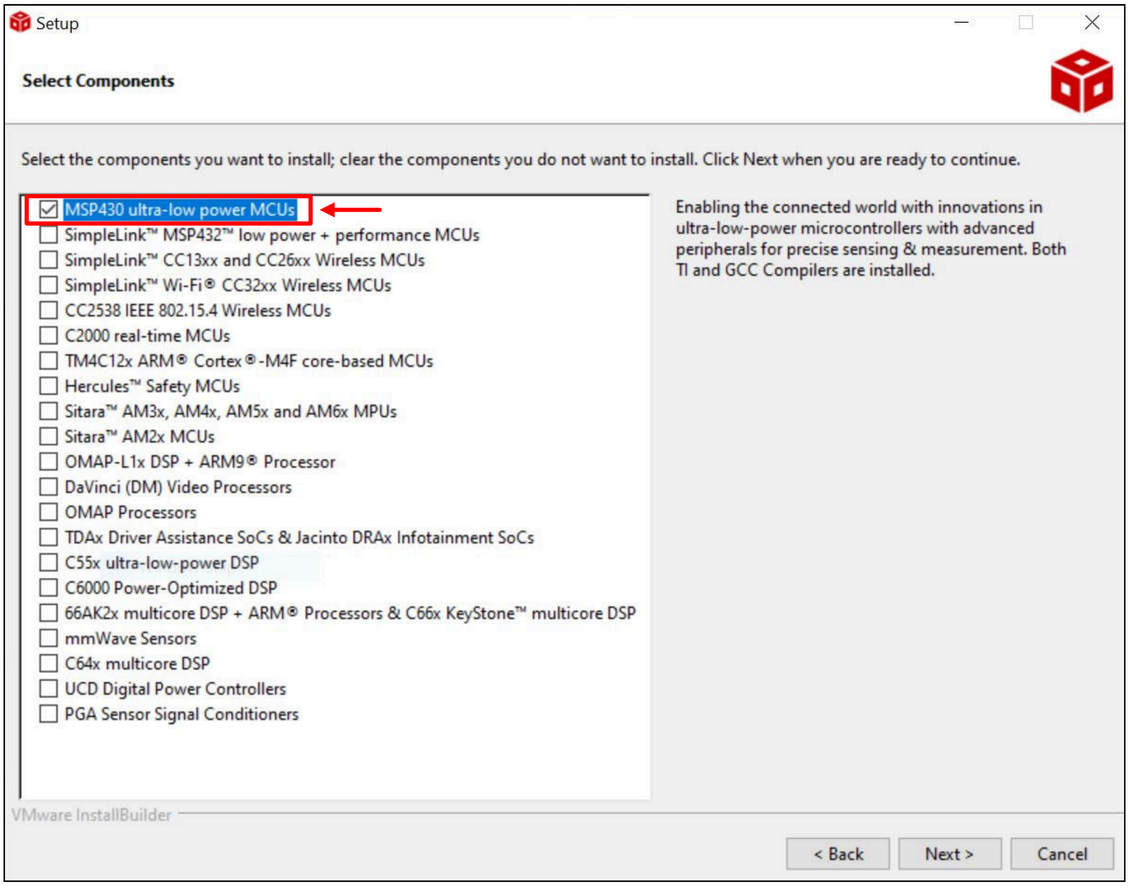
In the simple test mode, the user can use some predefined APIs to light up the LED board, or build custom Continuous Clock Serial Interface (CCSI) commands. The sample code comes with turning on all RGB LEDs which results in a white display.

The predefined APIs automatically adjust to the specified system. More detail about the system specification can be found in Section 3.3.

2 Software Setup

To set up the software for the MSP430F5529 LaunchPad™, please follow these steps (demonstrated in a computer with Windows 10 OS):

1. Download and install Code Composer Studio™.
   a. Download Code Composer Studio integrated development environment (IDE) (Version ≥ 11.1.0).
   b. Follow the installation instructions to install Code Composer Studio. During the installation process, if you choose the "Setup type" to be "Custom Installation", make sure that you select "MSP430 ultra-low power MCUs" in "Select Components", as is marked with red box in Figure 2-1.



Setup

Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ MSP430 ultra-low power MCUs
- ☐ SimpleLink™ MSP432™ low power + performance MCUs
- ☐ SimpleLink™ CC13xx and CC26xx Wireless MCUs
- ☐ SimpleLink™ Wi-Fi® CC32xx Wireless MCUs
- ☐ CC2538 IEEE 802.15.4 Wireless MCUs
- ☐ C2000 real-time MCUs
- ☐ TM4C12x ARM® Cortex®-M4F core-based MCUs
- ☐ Hercules™ Safety MCUs
- ☐ Sitara™ AM3x, AM4x, AM5x and AM6x MPUs
- ☐ Sitara™ AM2x MCUs
- ☐ OMAP-L1x DSP + ARM9® Processor
- ☐ DaVinci (DM) Video Processors
- ☐ OMAP Processors
- ☐ TDAx Driver Assistance SoCs & Jacinto DRAx Infotainment SoCs
- ☐ C55x ultra-low-power DSP
- ☐ C6000 Power-Optimized DSP
- ☐ 66AK2x multicore DSP + ARM® Processors & C66x KeyStone™ multicore DSP
- ☐ mmWave Sensors
- ☐ C64x multicore DSP
- ☐ UCD Digital Power Controllers
- ☐ PGA Sensor Signal Conditioners

Enabling the connected world with innovations in ultra-low-power microcontrollers with advanced peripherals for precise sensing & measurement. Both TI and GCC Compilers are installed.

Figure 2-1. Installation Process of Code Composer Studio

2. Download and import sample code.
   a. The link for each EVM is different. However, the sample code in each link is the same except for the file `led_driver.h` which is setup for the matching EVM.
      i. LP5890EVM: LP5890EVM-SW-F5529
      ii. LP5891EVM: LP5891EVM-SW-F5529
      iii. TLC6983EVM: TLC6983EVM-SW-F5529
      iv. TLC6984EVM: TLC6984EVM-SW-F5529
   b. Importing the Code Composer Studio (CCS) project according to the process provided in the link: Importing a CCS Project.
3. Load the program according to the process provided in the link: Building and Running Your Project.
4. (optional) Download the register map generation tool. This is a handy tool if you want to further configure the registers. For each of the supported devices there is a separate link:
   a. LP5890: LP5890 Registers Map Generation Tool
   b. LP5891: LP5891 Registers Map Generation Tool
   c. TLC6983: TLC6983 Registers Map Generation Tool
   d. TLC6984: TLC6984 Registers Map Generation Tool

3 Sample Code Structure

3.1 Design Parameters

The LED matrix display design parameters used for the different EVMs are listed in Table 3-1.

Table 3-1. Design Parameters EVMs

Design Parameter	LP589x	TLC698x
Display module size	16 × 16 RGB LEDs	32 × 32 RGB LEDs
Frame rate	25 Hz	25 Hz
Refresh rate	3200 Hz	3200 Hz
PWM resolution	16 bits	16 bits
Cascaded devices	1	2
SCLK frequency	12 MHz	6 MHz
GCLK frequency	48 MHz	66 MHz

3.2 Flow Diagram

Figure 3-1 depicts the high level flow in the sample code.

The `FC_settings.h` file can automatically be generated by the Registers Map Generation Tool mentioned in Section 2. Not all register settings are coming from this file. The number of scan lines (field SCAN_NUM in register FC0) and the number of cascaded devices (field CHIP_NUM in register FC0) are coming from the file `system_info.h`. This file is described in more detail in Section 3.3.

The flow diagram also shows the files `frames.c` and `frames.h` which contain the 2 frames used during the animation mode. It is outside the scope of this document to explain how these frames are generated.

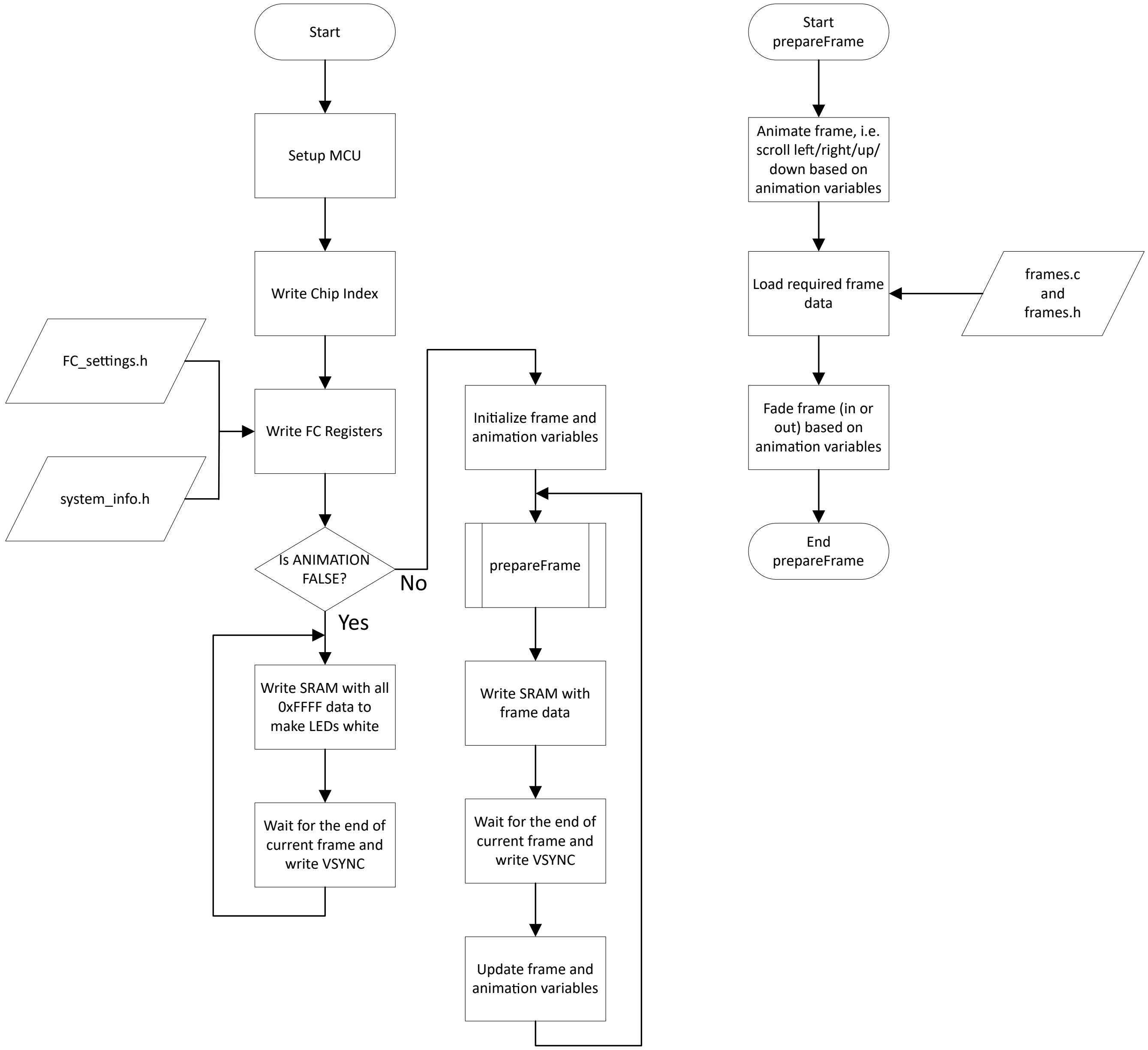

Start
Setup MCU
Write Chip Index
FC_settings.h
system_info.h
Write FC Registers
Is ANIMATION FALSE?
No
Yes
Write SRAM with all 0xFFFF data to make LEDs white
Wait for the end of current frame and write VSYNC
Initialize frame and animation variables
prepareFrame
Write SRAM with frame data
Wait for the end of current frame and write VSYNC
Update frame and animation variables
Start prepareFrame
Animate frame, i.e. scroll left/right/up/down based on animation variables
Load required frame data
frames.c and frames.h
Fade frame (in or out) based on animation variables
End prepareFrame

Figure 3-1. Sample Code Flow Diagram

3.3 System Setup

This section describes how the sample code setups different parameters to identify how the system is built. The first part is the actual used LED driver IC. Within the `led_driver.h` file, the used LED driver IC is selected.

```
#include "LP5891.h"
```

The code supports:

- LP5891
- LP5890
- TLC6984
- TLC6983

A summary about macros and variables that impact the system setup and their location is listed in Table 3-2.

Table 3-2. Summary of Macro and Variable Names Per File

Filename	Macro/Variable name	Description
system_info.h	*ANIMATION*	Selection between animation and simple test modes
	TOTAL_SCAN_LINES	Number of scan lines
	CASCADED_UNITS_CCSI1	Device count in CCSI bus 1
	MONOCHROMATIC	Selection between RGB and single color display
system_info.c	*FRAME_PERIOD*	Interval of VSYNC commands

Within the file `system_info.c` the frame period is specified which determines the frame rate. The frame period is specified in microseconds.

```
const uint16_t FRAME_PERIOD = 40000;      // 40ms = 25 Hz frames-per-second
```

The maximum supported frame period is 65535 microseconds, i.e. the lowest frame rate is 15.3 Hz.

File `system_info.h` includes several system definitions.

```
#define ANIMATION               TRUE
#define MONOCHROMATIC           FALSE
```

Macro *ANIMATION* will determine if the animation or simple test mode is executed.

The EVMs all use RGB LEDs. Therefore, macro *MONOCHROMATIC* is defined as FALSE. The sample code does support systems using single color LEDs, e.g. only red LEDs. In those cases, the macro *MONOCHROMATIC* should be defined as TRUE. This automatically changes the frame data structure, the animation algorithms, and APIs.

The following code block shows macros which impact the register settings.

```
#define TOTAL_SCAN_LINES       16
#define CASCADED_UNITS_CCSI1   1
```

Macro *TOTAL_SCAN_LINES* defines the number of scan lines used in the system and will directly impact the field SCAN_NUM in register FC0.

For the LP5890EVM and LP5891EVM there are 16 scan lines. For the TLC6983EVM and TLC6984EVM there are 32 scan lines.

Macro *CASCADED_UNITS_CCSI1* defines the number of cascaded devices in the system and directly impacts the field CHIP_NUM is register FC0.

For the LP5890EVM and LP5891EVM there is only 1 device cascaded.

When the user cascades more EVMs with the available connectors, this macro will have to be updated.

For the TLC6983EVM and TLC6984EVM there are 2 cascaded devices on one EVM.

3.4 Demo

In this section, examples of the LED demo are presented. Figure 3-2 depicts the LP5891EVM running a demo.

Figure 3-3 depicts the TLC6984EVM running a demo.

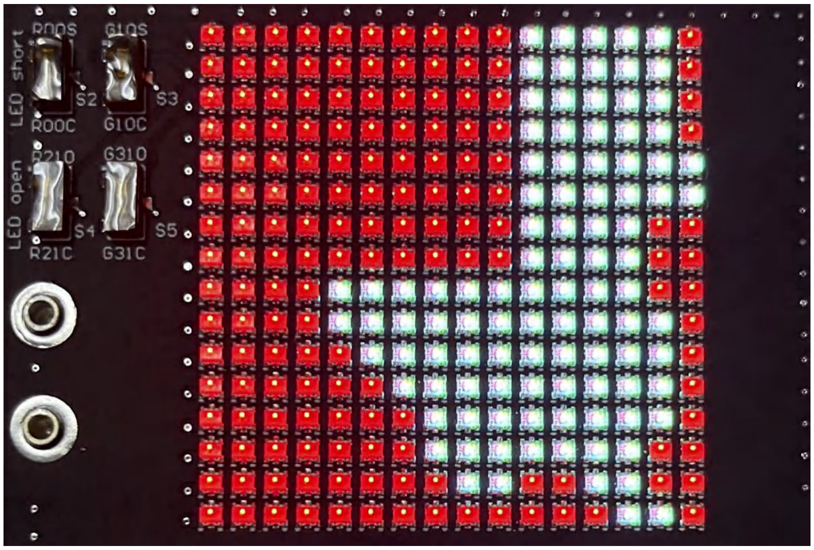


Figure 3-2. LP5891EVM Demo Example

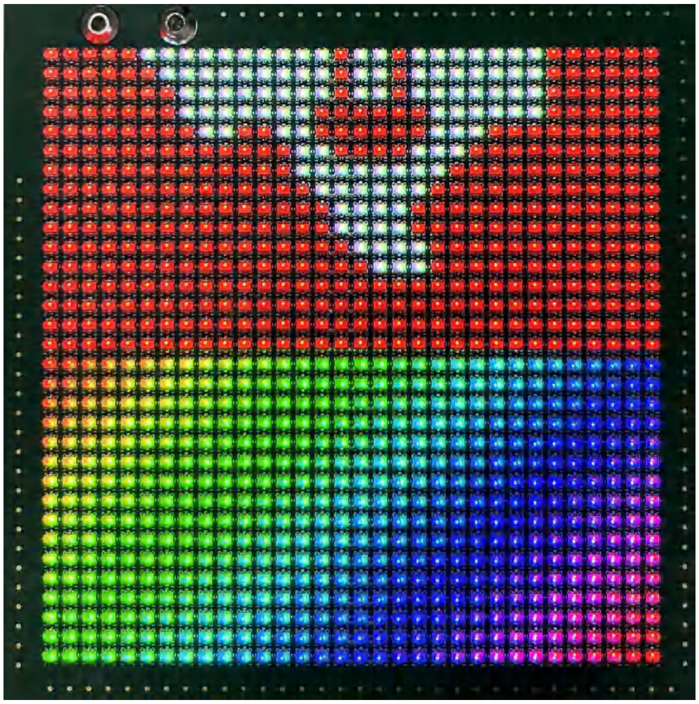


Figure 3-3. TLC6984EVM Demo Example

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

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