Application Report
SLAA293–April 2006

Using Two MSP430F4xx Devices to Drive Additional LCD Segments

Lane Westlund MSP430 Applications

ABSTRACT

This application report describes a technique that uses two MSP430F4xx devices to expand the total number of segments available for driving an LCD. In addition to driving additional LCD segments, other benefits include increased available RAM, flash, and I/O lines. All of these benefits can be realized while keeping the overall system current consumption below 8 µA.

Contents

List of Figures

1 Introduction

The maximum number of segments that can be driven by a single MSP430 device is currently 160. In some cases it is necessary for an MSP430 to interface to an LCD with more than this number of segments. External LCD drivers are one possibility, but may not be low-power. Using a second MSP430 achieves the same goal, but brings additional benefits such as ultralow-power, dual processors, additional peripherals, and additional memory.

2 MSP430x4xx LCD Driver Information

The MSP430 on-chip LCD controller directly drives LCD displays by creating the ac segment and common voltage signals automatically. The MSP430 LCD controller can support static, 2-multiplexer (mux), 3-mux, and 4-mux LCDs. The LCD controller generates these signals through the use of a timing generator which is sourced from the basic timer. In 4-mux mode, each segment pin drives 4 LCD segments. This is done by combining the signal from the segment pin and the common lines at the point of the LCD segment. The resultant RMS value of the combined signals causes the segment to be either on or off. More information on the LCD driver can be found in the MSP430x4xx User's Guide (SLAU056).

3 System Overview

In this system, two MSP430 devices are used to drive one LCD. This is accomplished by having one master MSP430 drive the LCD using its common and segment lines. The slave device connects to the display only through its segment lines. It is the task of the master device to signal the slave to ensure that both devices are synchronized and transmitting the correct data to be displayed.

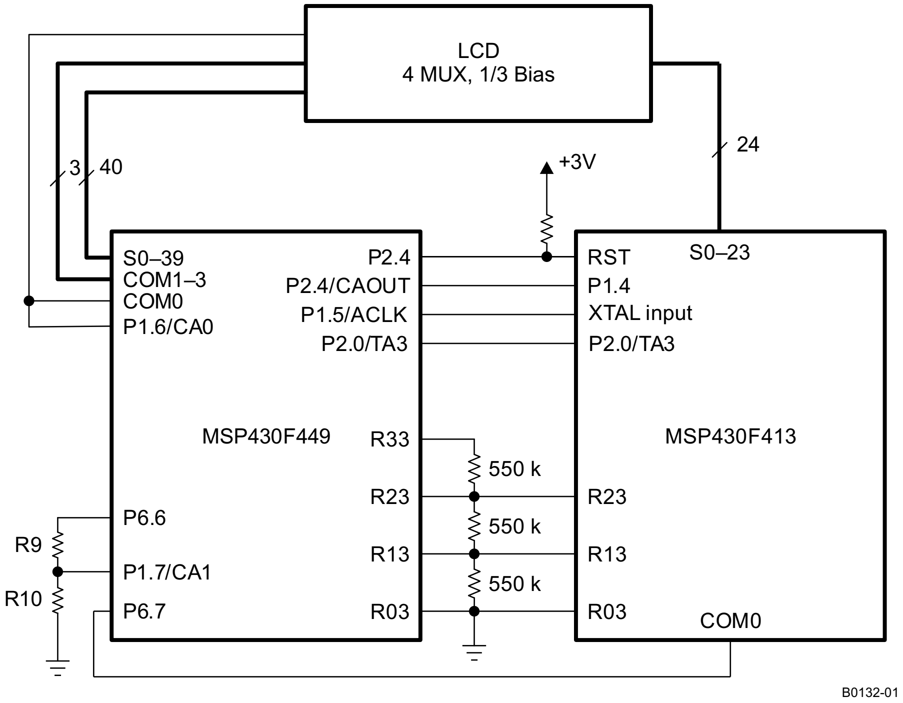

Figure 1. System Overview

For certain LCD segments, the COM signals from the master are combined with the segment signals from the slave. Synchronization is important because any discrepancy between these signals could cause a dc buildup which can damage an LCD screen. This means, in effect, that although only one MSP430 COM signal goes to the LCD, the COM lines of both devices must be synchronized and switching on identical ACLK transitions.

To accomplish this, the master must do four things:

- Ensure both master and slave are using the same ACLK for their LCD drivers
- Ensure its LCD driver state machine is in a known state
- Reset the slave and ensure the state of the slave LCD driver
- Signal to the slave device that the master has started

After the master has done these things, the slave must simply accomplish the following:

- Receive the signal from the master, and start its LCD driver so that the COM signals of both devices are synchronized
- Receive communication from the master describing which data to display

4 Synchronizing the Clocks

A key factor for enabling LCD controller synchronization between two MSP430s is the clock source driving the LCD controller of each device. It is important that both devices share a clock in order that LCD controller transitions happen on identical edges. The master first polls for LFOF, and then waits for approximately 2 seconds to ensure that the clock has completely stabilized. After the crystal has stabilized, it then enables its ACLK out pin to begin outputting ACLK to the slave.

The slave device receives this signal at its XIN and XOUT pins through the circuit shown in Figure 2. This circuit enables the slave to remain in LF mode and receive the square wave. This method is more susceptible to interference from noise, and for that reason should be shielded from outside signals. Typically, the recommended practice for such a problem is to simply drive a 0–3-V square wave into XIN with the device in HF mode. However, using HF mode in such a way leads to higher current consumption. By using the method outlined in Figure 2, the slave 4xx device is able to input a 32.768-kHz square wave while still remaining in LF mode for low-power reasons.

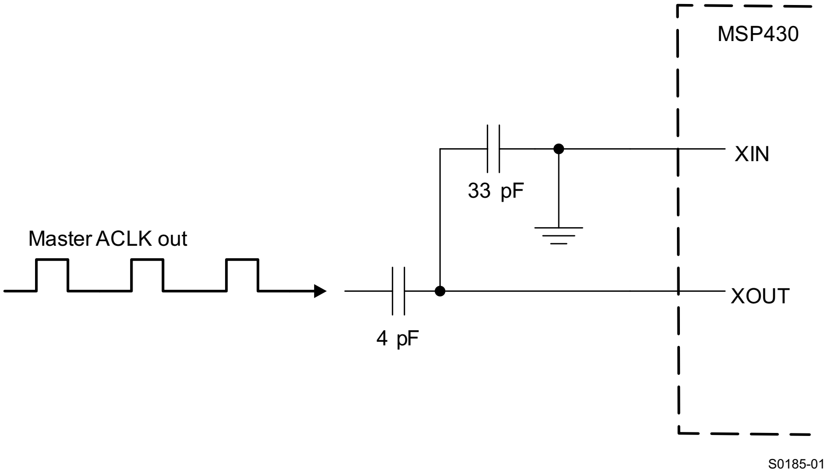

Figure 2. Slave ACLK Input Schematic

Any discrepancy in the crystal startup could cause the master and slave LCD drivers to be off by an ACLK cycle. Additional error could also be inserted from other interference. As with any sensitive analog system, best-practice guidelines should be followed; the capacitors should be placed near to the pins, and the clock lines should be shielded from outside interference.

5 LCD Deterministic Startup

The MSP430 LCD driver state machine can start up in a non-deterministic manner in two ways: following a toggling of the $\overline{\text{RST}}$ line, or if the LCD module was started with the basic timer already running.

As mentioned previously, the basic timer drives the LCD state machine. At certain counts, as determined by the divider used, the basic timer causes the LCD driver to change state and output a different voltage level on the COM lines. The basic timer can be at any count during the time the LCD state machine is started. Because of this, the first voltage driven by the LCD state machine is a nondeterminate shorter interval. If however, the basic timer count is set to zero immediately before starting the LCD driver, then the first voltage seen is a fixed number of basic timer ticks. This number is shorter than the subsequent voltages, but it is a deterministic value, so it can be timed against.

When a device is reset by driving the $\overline{\text{RST}}$ line low, it is possible for the LCD module to start nondeterministically. This becomes important for the slave device, as the master must be able to reset it. After a reset, if the LCD module has not been fully reset, the COM0 line is driven high instead of staying low as it does after the device is powered on. If this is the case, subsequent resets ensure the resetting of the LCD module and cause it to be in the correct state.

For this reason, the master measures the slave COM0 output on P6.7 and resets the slave while COM0 is high. It is important to note that P6.7 was chosen specifically because it can be routed to analog function. After the slave is correctly running, its COM0 drives intermediate voltages into this pin. If this were done to a regular I/O pin, indeterminate voltage would be driven into a digital I/O, increasing current consumption. It is acceptable, however, to drive such signals into an analog pin as is documented in this application report.

6 Signaling the Slave

In order for the slave to correctly start its LCD controller, it must receive a signal from the master. This signal by the master must be at a precise time within the master LCD frame, so the slave can wait exactly the correct amount of time before starting its LCD module. To accomplish this, the master measures its own COM0 line with its comparator module. The reference voltage is input from a resistor-divider. These resistors are powered by the master port pin (P6.6) in order that they can be turned off to conserve power. The ratio between these resistors should be 1 to 5 so that the comparator is only triggered by the highest voltage of the COM0 frame. In this way, the comparator output can be used as a signal to the slave. When this signal is high, the master has just begun the first high voltage of its LCD frame.

As mentioned previously, the master's first high signal in the LCD frame is somewhat shorter than subsequent frames. Due to this, the master only enables its comparator output after the first comparator interrupt has been triggered.

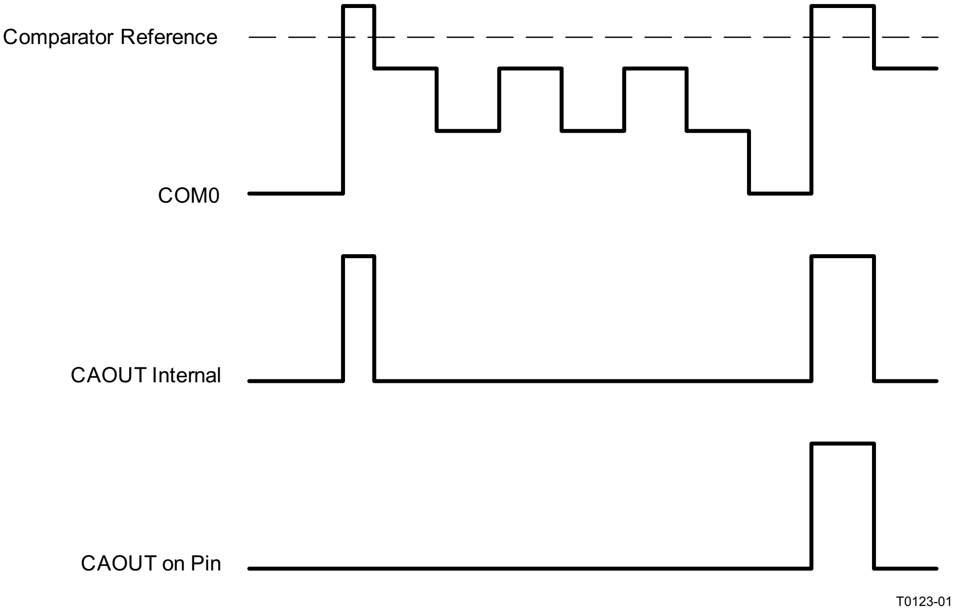

Figure 3. Master Comparator Output

It would be possible to trigger the slave from the first CAOUT signal. However, the slave would then need to time two different start-up delays, one for when the master has just started, and another for any subsequent resynchronizations based on an already running master. For the slave, it is easier always to wait a certain number of ACLK cycles after the signal before starting its LCD module. For this reason, the shorter initial high voltage is not used for synchronization.

7 Slave Startup

When the slave starts, it performs some standard initialization and then goes directly into a low-power mode, waiting for an interrupt. The slaves LCD module and basic timer are initialized; however, the basic timer is stopped by setting the BTHOLD bit. This prevents the basic timer from incrementing, so that it remains at a known value after zero is written to it. At this time, it can be reset by the master in order to reset the LCD module as described previously. If the master has determined the slave LCD is in the correct state, it signals the slave as described previously. The slave receives this signal as an interrupt on pin P1.4.

When the slave receives the signal from the master, it means the master has just reached a certain known state in its LCD frame sequence. Because the master is in a known state and the slave is in a known state, the slave can wait a fixed number of ACLK cycles and start its LCD module with both master and slave synchronized. This is accomplished by initializing Timer_A in the port ISR. The slave sets Timer_A to interrupt after a fixed number of ACLK cycles, and then the slave returns to LPM3. In the Timer_A ISR, the slave simply clears the BTHOLD bit. Clearing this bit starts the basic timer, which starts the LCD module. Because all timing in this sequence is deterministic, and based off the same clocks between master and slave, this procedure results in synchronized LCD state machines.

8 Master–Slave Communication

After both LCD controllers have been synchronized, the slave can operate as normal and write values to its LCDMEM. Naturally, the values displayed should be coordinated with the master. In this application report, a custom UART protocol has been developed which allows the master to write to memory locations in the slave, including LCDMEM locations.

Any method of communication between two MSP430s can be used to exchange this information. For this application report, master-to-slave 9600-baud UART communication using Timer_A was chosen.

9 User Interaction

The zip file accompanying this application report contains master and slave source files. Also included is a schematic describing an evaluation board with all necessary connections and external hardware. The supplied code is designed to run on this board. When run, the master code waits before synchronizing the LCD. While waiting, the master listens for a high-to-low transition caused by depressing a button connected to P1.0. Pressing this button causes the master to begin the synchronization sequence and enter the first display state. Subsequent button presses cycle the master through the following display states:

1. All off – All segments are off.
2. Single-step – Each device cycles through its LCD memory, turning on one bit at a time.
3. All on – All segments are turned on.
4. Step all – Master and slave cycle through various number sequences to simulate actual data being displayed.

10 Software Flow Chart

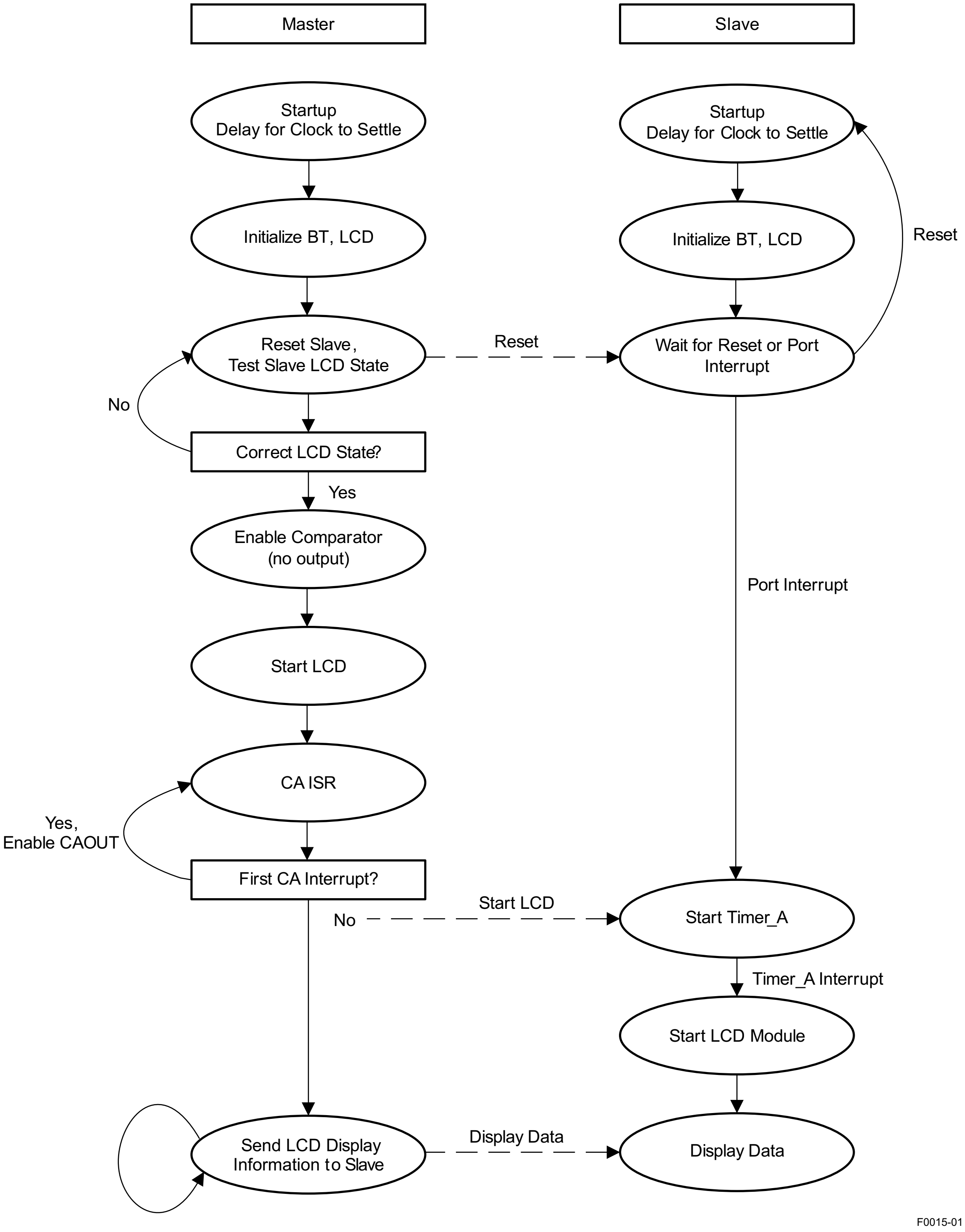

Figure 4. Software Flow Chart

11 References

1. *MSP430x4xx Family User's Guide* (SLAU056)
2. *MSP430x41x Mixed Signal Microcontroller* data sheet (SLAS340)
3. *MSP430x43x, MSP430x44x, Mixed Signal Microcontroller* data sheet (SLAS344)
4. Softbaugh SBLCDA2 Specification, Softbaugh

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

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

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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