Application Report
SLAA771–January 2018

Scaling Across the SimpleLink™ MSP432P4 MCU Family

Bob Landers *MSP Applications*

ABSTRACT

The SimpleLink™ MSP432P4x microcontrollers (MCUs) are Arm® Cortex®-M4F MCUs optimized for ultra-low-power performance starting at 80 µA/MHz down to 660 nA in standby mode, with an integrated 16-bit precision analog-to-digital converter (ADC), Floating Point Unit (FPU), and digital signal processor (DSP) extensions. As an optimized wireless host MCU, the MSP432P401M and MSP432P401R devices allow developers to add high-precision analog and memory extension to applications based on SimpleLink wireless connectivity solutions. The introduction of larger memory configurations (up to 2MB in the MSP432P4x1V, MSP432P4x1Y, and MSP432P4x11 devices) to the existing MSP432P4 MCU family opens up new application spaces while retaining the simplicity and code reuse features intrinsic to all SimpleLink devices. This application report highlights the feature differences to consider when moving applications between the different devices within the MSP432P4 family as well as how to best leverage these features.

Contents

Trademarks

1 Introduction

The SimpleLink MSP432™ devices are supported by the SimpleLink software development kit (SDK), a comprehensive ecosystem of tools, software, training and documentation that enable you to get started quickly on new designs. This powerful SDK provides a cohesive and consistent software experience when developing on any device within the SimpleLink family. As application requirements change to require larger memories, the SimpleLink SDK supports 100% code reuse across the MSP432 MCU family, allowing the developer to focus on new features rather than code porting activities.

The MSP432P4x1V/Y/1 devices offer up to 2MB of on-chip flash, which allows more feature-rich designs supporting multiple communication interfaces such as *Bluetooth*® low energy, Wi-Fi®, and Sub-1 GHz all without the requirement for external memory. The dual-bank architecture of the flash system enables robust in-field software updates with little to no downtime to switch to the new firmware version.

In addition to larger memory, the MSP432P4x1V/Y/1 devices bring other new features to the MSP432 line, such as:

- Enhancements to the power management system, including:
  - 7 new wake-up sources from LPM3 low-power mode
  - 4 new wake-up sources from LPM4 low-power mode
  - Higher speeds in the low-frequency low-power clock domains (128 kHz, or four times the performance in low-power mode without paying the latency and power costs of starting up MHz clock sources.)
- Enhancement of the SRAM system both in size (now up to 256KB from 64KB) and a more flexible memory power partition to retain granularity of retention settings with the larger memory.
- Addition of an LCD controller with up to 320-segment control, support for up to 8-wide multiplexing and a low-power animation capability
- Extended temperature operation. A new *T* variant now allows operation up to 105°C junction temperature.

This application report is divided into the following sections:

- System-level differences such as power management
- Changes in the structure of onboard memory
- Peripheral modifications
- Software-level changes

2 System-Level Differences

2.1 Clock System Differences

2.1.1 Operating Mode CPU Frequency Changes

All commercial-temperature (85°C) devices in the MSP432 family have the same CPU frequency capabilities in the active mode. The Industrial temperature (105°C) devices operate only on the lower core voltage VCORE0, and as a result are able to operate only up to frequencies associated with that voltage. If the user application requires porting between 85°C and 105°C devices, the application should take care not to exceed the VCORE0 maximum frequency (24 MHz) on the 105°C device. This delineation can be built into the code at compile time by using the TLV to check the temperature rating of the device and comparing the resulting value to the device IDs found in the respective data sheets as shown below.

```
// from Device Descriptions section, Table 6-89 of MSP432P4111I data sheet
typedef struct
{
    uint32_t  deviceID;
    uint32_t  hwRev;
    uint32_t  bootCodeRev;
    uint32_t  romDriverLibRev;
} SysCtl_A_InfoBlock;

uint32_t     devtemp;      // Device temperature type (1 or 2)
uint_fast8_t tlvLength;    // Length of descriptor in words
SysCtl_A_InfoBlock *devInfo;

SysCtl_A_getTLVInfo(TLV_TAG_DEVINFO, 0, &tlvLength, &devInfo);
devtemp = (devInfo->deviceID & 0x00000F0)>>4;
if(devtemp == 1) {         // Commercial-temp device
}
else if (devtemp == 2) {   // Industrial-temp device
}
else { }                   // else- invalid device id...
```

Table 1. Device IDs for Commercial Temperature (I) and Industrial Temperature (T) Devices

Commercial (I) Devices		High-Temperature (T) Devices	
Device	**Device ID**	**Device**	**Device ID**
MSP432P4111IPZ	0000A010h	MSP432P4111TPZ	0000A020h
MSP432P411YIPZ	0000A012h	MSP432P411YTPZ	0000A022h
MSP432P411VIPZ	0000A016h	MSP432P411VTPZ	0000A026h
MSP432P4011IRGC	0000A019h	MSP432P4011TRGC	0000A029h
MSP432P401YIRGC	0000A01Bh	MSP432P401YTRGC	0000A02Bh
MSP432P401VIRGC	0000A01Fh	MSP432P401VTRGC	0000A02Fh

Table 2. Operating Mode CPU Frequency Comparison of MSP432 Family Devices

Operating Mode	Description	f_{MCLK}			Unit
		MSP432P401R/M	MSP432P4x1V/Y/1		
		85°C	85°C (I)	105°C (T)	
		MAX	MAX	MAX	
AM_LDO_VCORE0	Normal-performance mode with LDO as the active regulator	24	24	24	MHz
AM_LDO_VCORE1	High-performance mode with LDO as the active regulator	48	48	N/A	MHz
AM_DCDC_VCORE0	Normal-performance mode with DC/DC as the active regulator	24	24	24	MHz
AM_DCDC_VCORE1	High-performance mode with DC/DC as the active regulator	48	48	N/A	MHz
AM_LF_VCORE0	Low-frequency mode with LDO as the active regulator	128	128	128	kHz
AM_LF_VCORE1	Low-frequency mode with LDO as the active regulator	128	128	N/A	kHz

2.1.2 Operating Mode Peripheral Frequency Changes

All peripherals in the MSP432 family operate at up to half the maximum CPU frequency available in active modes. The MSP432P4x1V/Y/1I devices have an enhanced performance capability in LPM low-power modes to allow operation of peripherals at up to 128 kHz, instead of 32.768 kHz in the P401M/R devices.

The Industrial temperature (105°C) MSP432P4x1V/Y/1T devices operate only on the lower core voltage VCORE0 and, as a result, are able to operate only up to frequencies associated with that voltage. Do not use the VCORE1 settings with these devices.

Table 3. Operating Mode Peripheral Frequency Comparison of MSP432 Family Devices

Parameter	Operating Mode	Description	Peripheral f_{MAX}			Unit
			P401R/M (85°C)	P4x1V/Y/1I (85°C)	P4x1V/Y/1T (105°C)	
$f_{AM\_LPM0\_VCORE0}$	AM_LDO_VCORE0	Peripheral frequency range in LDO- or DC/DC-based active or LPM0 modes for VCORE0	12	12	12	MHz
	AM_DCDC_VCORE0					
	LPM0_LDO_VCORE0					
	LPM0_DCDC_VCORE0					
$f_{AM\_LPM0\_VCORE1}$	AM_LDO_VCORE1	Peripheral frequency range in LDO- or DC/DC-based active or LPM0 modes for VCORE1	24	24	N/A	MHz
	AM_DCDC_VCORE1					
	LPM0_LDO_VCORE1					
	LPM0_DCDC_VCORE1					
$f_{AM\_LPM0\_LF}$	AM_LF_VCORE0	Peripheral frequency range in low-frequency active or low-frequency LPM0 modes for VCORE0 and VCORE1	128	128	128	kHz
	AM_LF_VCORE1					
	LPM0_LF_VCORE0					
	LPM0_LF_VCORE1					
f_{LPM3}	LPM3_VCORE0	Peripheral frequency in LPM3 mode for VCORE0 and VCORE1	32.768	128	128	kHz
	LPM3_VCORE1				N/A	
f_{LPM4}	LPM4_VCORE0	Peripheral frequency in LPM4 mode for VCORE0and VCORE1	N/A	128	128	kHz
	LPM4_VCORE1				N/A	
$f_{LPM3.5}$	LPM3.5	Peripheral frequency in LPM3.5 mode	32.768	32.768	32.768	kHz

2.1.3 LPM3 and LPM4 Frequency Limits

The MSP432P4xx family of devices have two different implementations for LPM3 and LPM4 depending on the number of peripheral groups implemented inside the devices. The most notable power difference between the MSP432P4xx family devices is that the LPM3 operating frequency for the Watchdog Timer (WDT) and Real-Time-Clock (RTC) is increased to 128 kHz for the MSP432P4x1V/Y/1I and MSP432P4x1V/Y/1T devices compared to 32 kHz for the P401R/M devices. In addition to this higher speed, external low-frequency clocks may also be used along with the other low-frequency internal clocks (LFXT, REFO, VLO). Table 4 lists additional details of the differences in LPM3 and LPM4 behavior between the two sets of devices.

Table 4. Frequency Limit Differences in LPM3 and LPM4 Modes

Component		MSP432P401R/M		MSP432P4x1V/Y/1 [1]		Unit
		LPM3	LPM4	LPM3	LPM4	
f_{max}		32.768	0	128	0	kHz
Low-frequency clocks (LFXT,REFO,VLO)		32.768	Off	128	Off	kHz
High-frequency clocks		Off	Off	Off	Off	kHz
External clocks		Off	Off	128	Off	kHz
Peripheral power						
	RTC	On [2]	Off	On [3]	Off	
	WDT	On [2]	Off	On [3]	Off	
	All other peripherals	Off	Off	On [4]	Off	
	SRAM	Retention [5]		Retention [5]		
	Peripheral registers	Retained		Retained		
	I/O pin states	Latched		Latched		

(1) The industrial temperature devices (MSP432P4x1V/Y/1T) have the same frequency limits in LPM modes as the commerical-temperature devices (MSP432P4x1V/Y/1I).
(2) Low-frequency clock sources only (LFXT, REFO, VLO)
(3) Low-frequency clock sources only (LFXT, REFO, VLO) or low-frequency external clocks
(4) Most but not all peripherals are functional from low-frequency sources.
(5) Only banks and blocks that are enabled for data retention

2.2 Flash Wait-States Differences

The relationship between MCLK frequency and flash wait states is different between the MSP432P4x1V/Y/1x (512KB/1MB/2MB) devices and the MSP432P401M/R (128KB/256KB) devices.

If the application code uses TI Drivers or the MSP432 Driverlib available in the MSP432P4 SDK, these differences have been accounted for, and the same APIs can be used for both device families.

Applications that use register-level code must account for these differences by referring to Table 5 when moving code between these devices.

The larger flash memory sizes available in the MSP432P4x1V/Y/1I and MSP432P4x1V/Y/1T (512KB/1MB/2MB) devices result in changes to the required number of wait-states compared to the earlier P401x (≤256KB) devices. All commercial-temperature (85°C) devices in the MSP432 family can ultimately support the same maximum frequency as shown in Table 5. The Industrial temperature (105°C) devices operate only on the lower core voltage VCORE0, and as a result are able to operate only up to frequencies associated with that voltage. However, the number of flash wait-states is the same between the MSP432P4x1V/Y/1 and MSP432P4x1V/Y/1T devices for a given VCORE voltage (VCORE0).

Table 5. MCLK Frequency and Flash Wait States

Parameter	No. of Flash Wait States	Flash Read Mode	Maximum Supported MCLK Frequency[1][2] (MHz)					
			P401R/M (85°C)		P4x1V/Y/1 (85°C)		P411T (105°C)	
			AM_LDO_ / AM_DCDC_		AM_LDO_ / AM_DCDC_		AM_LDO_ / AM_DCDC_	
			VCORE0	VCORE1	VCORE0	VCORE1	VCORE0	VCORE1
$f_{MAX\_NRM\_FLWAIT0}$	0	Normal read mode	16	24	10	13	10	N/A
$f_{MAX\_NRM\_FLWAIT1}$	1		24	48	21	27	21	
$f_{MAX\_NRM\_FLWAIT2}$	2		N/A	12	24	40	24	
$f_{MAX\_NRM\_FLWAIT3}$	3		N/A	N/A	24	48	N/A	
$f_{MAX\_ORM\_FLWAIT0}$	0	Other read modes[3]	8	12	6	7	6	N/A
$f_{MAX\_ORM\_FLWAIT1}$	1		16	24	12	14	12	
$f_{MAX\_ORM\_FLWAIT2}$	2		24	36	18	21	18	
$f_{MAX\_ORM\_FLWAIT3}$	3		24	48	24	28	24	
$f_{MAX\_ORM\_FLWAIT4}$	4		N/A	N/A	24	35	N/A	
$f_{MAX\_ORM\_FLWAIT5}$	5		N/A	N/A	24	42	N/A	
$f_{MAX\_ORM\_FLWAIT6}$	6		N/A	N/A	24	48	N/A	

(1) Violation of the maximum frequency limitation for a given wait-state configuration results in nondeterministic data or instruction fetches from the flash memory.

(2) In low-frequency active modes, the flash can always be accessed with 0 wait states, because the maximum MCLK frequency is limited to 128 kHz.

(3) *Other read modes* are Read Margin 0, Read Margin 1, Program Verify, and Erase Verify.

2.3 Power Management

2.3.1 LPM3 and LPM4 Wake-up Sources

The MSP432P4x1V/Y/1 devices provide enhanced functionality through the addition of a number of wake-up sources from LPM3 and LPM4 sleep modes. Table 6 lists the wake-up sources available for each of the LPMx low-power modes. The highlighted cells are sources that are available in the MSP432P4x1V/Y/1 devices but are NOT available in the MSP432P401M/R devices.

Table 6. LPM3 and LPM4 Wake-up Sources

Peripheral	LPM0	LPM3	LPM4	LPM3.5	LPM4.5
eUSCI_A	Yes	Yes[1]	Yes[1]		
eUSCI_B	Yes	Yes[1]	Yes[1]		
Timer_A	Yes	Yes[1]	Yes[1]		
Timer32	Yes				
Comparator_E	Yes	Yes[1]	Yes[1]		
ADC14	Yes	Yes[1]			
LCD	Yes[1]	Yes[1]			
AES256	Yes				
DMA	Yes				
Clock System (CS)	Yes	Yes[1]			
Power Control Manager (PCM)	Yes				
FLCTL	Yes				
WDT_A in watchdog mode	Yes				
RTC_C	Yes	Yes		Yes	
WDT_A in interval timer mode	Yes	Yes		Yes	
I/O ports	Yes	Yes	Yes	Yes	Yes
NMI at device pin	Yes	Yes	Yes		
SVSMH in monitor mode	Yes	Yes	Yes		

(1) New for MSP432P4x1V/Y/1

Table 6. LPM3 and LPM4 Wake-up Sources (continued)

Peripheral	LPM0	LPM3	LPM4	LPM3.5	LPM4.5
Debugger power-up request		Yes	Yes	Yes	Yes
Debugger reset request	Yes	Yes	Yes	Yes	Yes
RSTn at device pin	Yes	Yes	Yes	Yes	Yes
SVSMH in supervisor mode	Yes	Yes	Yes	Yes	Yes
Power cycle	Yes	Yes	Yes	Yes	Yes

2.3.2 Power Mode Differences

The power control module (PCM) is responsible for defining the functionality in LPM3, LPM4, and LPMx.5 low-power modes and the sequence of events when transitioning between these modes and active mode. For the MSP432P4x1V/Y/1, the PCM provides modified functionality in low-power modes compared to the MSP432P401M/R. Specifically, for the MSP432P4x1V/Y/1, the ACLK is kept ON by default in deep sleep. Users relying on the device to turn off ACLK automatically (which is the behavior in the MSP432P401M/R devices) will needd to do this explicitly in their own code. Further details of these behavioral differences are provided in Table 7 and in the *Power Control Manager (PCM)* chapter of the MSP432P4xx SimpleLink™ Microcontrollers Technical Reference Manual.

Table 7. LPM Power Mode Behavior Differences Between MSP432P401M/R and MSP432P4x1V/Y/1

Mode	MSP432P401M/R	MSP432P4x1V/Y/1
LPM3	RTC and WDT modules can be functional up to a maximum of 32.768 kHz.	RTC and WDT modules can be functional up to a maximum of 32.768 kHz.
	All other peripherals are kept in state retention power gating.	Most peripherals can operate out of internal or external clock sources up to a maximum of 128 kHz (see data sheet for peripheral group list).
	Only low-frequency clock sources (LFXT, REFO, and VLO) can be active.	Internal low-frequency clock sources (LFXT, REFO, and VLO) or external low-frequency clk sources (max 128 kHz) can be active.
LPM4	All peripherals are kept in state retention power gating.	Analog modules not requiring a clock can remain operational. All other peripherals are kept in state retention power gating.

NOTE: MSP432P401R and MSP432P401M devices only: The application must ensure that the analog modules (ADC14 or COMP_E) are disabled before entering LPM3 or LPM4 modes, if they are configured to use the internal reference. This must be done regardless of the FORCE_LPM_ENTRY bit setting.

The PCM does not perform any clock frequency or condition checks during any active or LPM0 power mode transitions. The application must ensure that the system conditions are met before the transition between these mode transitions. For transitions to LPM3, LPM4, and LPMx.5 modes, the PCM does perform checks on the state of requested clocks prior to enter these low-power states. If an active clock source is present in the system in any of the system clocks, the the application can decide whether to force the LPM3, LPM4, and LPMx.5 entry or not. These two transition modes are:

- *Polite* mode – FORCE_LPM_ENTRY = 0
  - Device will not enter low-power mode if an active clock source is present.
  - User is expected to turn off ACLK, peripherals, and so on before entering deep sleep.
- *Rude* mode – FORCE_LPM_ENTRY = 1
  - Device will enter LPMx regardless of whether an active clock source is present.
  - MSP432P401M/R: Device automatically turns off everything except the backup domain, so ACLK is automatically turned off.
  - MSP432P4x1V/Y/1: ACLK is kept on by default in deep sleep. Application code must turn off ACLK explicitly.

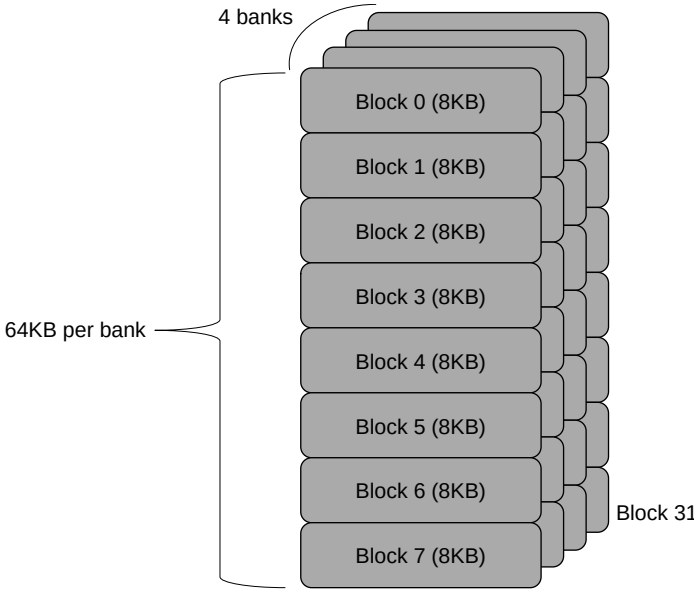
2.3.3 LPM3 Power for High-Temperature Devices

The MSP432P4 devices integrate a Supply Voltage Supervisor and Monitor (SVSMH) to either supervise or monitor the high-side voltage, V_{CC}. This monitor requires the use of a bandgap reference circuit (part of the REF_A module) to accurately set voltage threshold values. For the commercial temperature (85°C) devices (MSP432P401R/M and MSP432P4x1xI) devices, the bandgap can be placed in a low-power sampled mode during LPM3, LPM4, and LPMx.5 operation by setting the PSSCTL0.SVSMHLP = 1. For the industrial-temperature (MSP432P4x1xT) devices, the SVSMHLP is ignored and the bandgap reference is always in static full-performance mode to maintain the voltage references. As a result, the quiescent current and subsequently the LPM3 currents are higher for industrial-temperature devices compared to the commercial devices for any given temperature.

3 Memory System Modifications

3.1 SRAM Bank and Block Organization

The MSP432P401R/M devices have an SRAM structured as 8 banks of memory, each with 1 block of 8KB of SRAM yielding up to 64KB. In the MSP432P4x1V/Y/1 devices, each of the up to 4 SRAM banks has 8 blocks of memory. Figure 1 shows the memory organization of a maximal 256KB configuration.



NOTE: 8 Banks × 8 Blocks × 8KB = 256KB

Figure 1. Maximum MSP432P4x1V/Y/1 SRAM Organization

Table 8. SRAM Organization Comparison

Part Number	Total SRAM (KB)	Banks	Blocks per Bank	Block Size (KB)
MSP432P401My	32	4	1	8
MSP432P401Ry	64	8	1	8
MSP432P4x1Vy	128	2	8	8
MSP432P4x1Yy	256	4	8	8
MSP432P4x11y	256	4	8	8

3.2 SRAM Power Control

In addition to the structural changes to memory, the MSP432P4x1V/Y/1 has per-block, rather than per-bank, granularity for memory retention control in LPM3 and LPM4 sleep modes. Per-bank granularity of SRAM enables in active modes is the same as in the P401R/M.

Table 9. SRAM Bank and Block Organization per Device

Part Number	Total SRAM (KB)	Banks	Blocks per Bank
MSP432P4x11y	256	4	8
MSP432P4x1Yy	256	4	8
MSP432P4x1Vy	128	2	8

NOTE: In all MSP432P4xx devices, Bank0 of SRAM is always enabled and cannot be disabled. Similarly, Block0 (the lowest 8KB) is always retained and cannot be disabled for retention.

3.2.1 Software Changes

A number of System Control (SYSCTL) registers have changed to support the new memory configuration and capabilities. To distinguish this register set from the previous MSP432P401R/M SYSCTL, the MSP432P4x1V/Y/1 devices define a new register, SYSCTL_A, which replaces the existing SYSCTL register.

Table 10. SYSCTL_A Register Changes From SYSCTL

Offset	New or Changed?	Acronym	Register Name
0000h	No	SYS_REBOOT_CTL	Reboot Control Register
0004h	No	SYS_NMI_CTLSTAT	NMI Control and Status Register
0008h	No	SYS_WDTRESET_CTL	Watchdog Reset Control Register
000Ch	Yes	SYS_PERIHALT_CTL	Peripheral Halt Control Register
0010h	No	SYS_SRAM_SIZE	SRAM Size Register
0014h	Yes	SYS_SRAM_NUMBANKS	SRAM Number of Banks Register
0018h	Yes	SYS_SRAM_NUMBLOCKS	SRAM Number of Blocks Register
0020h	Yes	SYS_MAINFLASH_SIZE	Flash Main Memory Size Register
0024h	Yes	SYS_INFOFLASH_SIZE	Flash Information Memory Size Register
0030h	No	SYS_DIO_GLTFLT_CTL	Digital I/O Glitch Filter Control Register
0040h	No	SYS_SECDATA_UNLOCK	IP Protected Secure Zone Data Access Unlock Register
0050h	Yes	SYS_SRAM_BANKEN_CTL0	SRAM Bank Enable Control Register 0
0054h	Yes	SYS_SRAM_BANKEN_CTL1	SRAM Bank Enable Control Register 1
0058h	Yes	SYS_SRAM_BANKEN_CTL2	SRAM Bank Enable Control Register 2
005Ch	Yes	SYS_SRAM_BANKEN_CTL3	SRAM Bank Enable Control Register 3
0070h	Yes	SYS_SRAM_BLKRET_CTL0	SRAM Block Retention Control Register 0
0074h	Yes	SYS_SRAM_BLKRET_CTL1	SRAM Block Retention Control Register 1
0078h	Yes	SYS_SRAM_BLKRET_CTL2	SRAM Block Retention Control Register 2
007Ch	Yes	SYS_SRAM_BLKRET_CTL3	SRAM Block Retention Control Register 3
0090h	No	SYS_SRAM_STAT	SRAM Status Register

3.3 Flash Control Changes

The MSP432P4x1V/Y/1 devices feature additional flash memory over the MSP432P401R/M MCUs. To support this additional memory, certain changes have been made to the flash controller that affect timing and obtainable frequencies for a given flash wait state setting. These changes are shown in Table 5. Additional modifications have been made to the register set, which is now designated FLCTL_A to distinguish it from the FLCTL registers of the MSP432P401R/M. Table 15 lists the flash control APIs for both sets of devices and shows both the naming and functional differences.

4 Peripheral Modifications

4.1 LCD Interface

The MSP432P4x1V/Y/1I and MSP432P4x1V/Y/1T devices add a new LCD interface, LCD_F, to the MSP432 feature set. This controller directly drives segment LCDs by automatically creating the AC segment and common voltage signals. The LCD_F controller supports the following features:

- Low-power mode support
  - Fully functional in LPM0, LF_AM, LF_LPM0 modes
  - Dispay, animation, and blink features in LPM3
- Full multiplexing mode support (static, 2-, 3-, 4-, 5-, 6-, 7-, or 8-mux)
- 8-segment animation with configurable animation frequency
- Individual segment blinking with separate blinking memory for all LCD types
- Automatic signal generation
- Configurable frame frequency
- Flexible pinout options for LCD pins
  - Each LCD pin configurable as SEG or COM pins for maximum flexibility in pinout selection.
  - Alternate ports for selected LCD pins to accommodate differing peripheral usage alongside the LCD interface

Table 11. Differences Among LCD Versions

Features	LCD_B	LCD_C	LCD_E	LCD_F
Supported types of LCDs	Static, 2-, 3-, 4-mux	Static, 2-, 3-, 4-, 5-, 6-, 7-, 8-mux	Static, 2-, 3-, 4-, 5-, 6-, 7-, 8-mux	Static, 2-, 3-, 4-, 5-, 6-, 7-, 8-mux
LCD bias modes	1/2 bias and 1/3 bias	1/2 bias and 1/3 bias	1/3 bias	1/2 bias,1/3 bias, and 1/4 bias
LCD blinking memory	Yes	Yes (supports Static, 2-, 3-, 4-mux modes)	Device specific	Yes (full blink memory supported in all mux modes)
SEG and COM mux	COM fixed	COM fixed	Each LCD drive pin	Each LCD drive pin
External pins	R03, R13, R23, R33	R03, R13, R23, R33	R13, R23, R33, LCDCAP0, LCDCAP1	R03, R13, R23, R33
LPM3.5	Not supported	Not supported	Supported	Not supported
Number of LCD pins	Up to 4 × 46	Up to 4 × 50 or 8 × 46	Up to 4 × 60 or 8 × 56	Up to 4 × 44 or 8 × 40 (device and package specific)
Charge pump	Present	Present	Present	Absent

5 Software Changes

5.1 Moving Projects to the 512KB/1MB/2MB Devices

A number of parameters and settings must be changed to retarget an MSP432P401M/R project to one of the new devices (MSP432P4x1V/Y/1). Ensuring that each of these changes is made can be cumbersome, so TI recommends starting with a new set of project settings by importing the *empty* project for the MSP432P4x1V/Y/1 device and copying source files from the original project into this new one. No code changes are required if using Driverlib or higher levels of abstraction (that is, TI drivers). If code is written at the register level, then the flash and system control (SRAM power management) require changes due to naming differences.

Table 12. Tasks for Moving Projects to the 512KB/1MB/2MB Devices

Source Code Type	Tasks
TI Drivers	No code changes across the entire MSP432P4x family
Driverlib	1. Import the driverlib Empty project for MSP432P4111. 2. Rename the project as needed. 3. Copy source code to the new project. 4. Rename FlashCtl and SysCtl APIs to FlashCtl_A and SysCtl_A. 5. Compile the project. 6. No other code changes needed across the MSP432P4x family.
Register level	1. Import a register-level project (example) for MSP432P4111. 2. Rename the project as needed. 3. Remove functional code from the example. 4. Copy source code over from old project to the new project. 5. Rename FLCTL and SYSCTL register references to FLCTL_A and SYSCTL_A. • FLCTL changes described in Section 3.3. • SYSCTL changes described in Section 3.2.1. 6. Compile the project.

5.2 High-Temperature Considerations for RTOS Settings

The MSP432P4x1V/Y/1I (commercial temperature) and MSP432P4x1V/Y/1T (industrial temperature) devices are functionally identical, and the development environment reports them as having the same family IDs. However, the differences in frequency and voltage settings between the two devices require changes in the underlying power management TI drivers when using an RTOS. These drivers are listed in the linker options, which are selected from within the project properties of the given source project.

Figure 2 shows a typical TI-RTOS project as seen from within Code Composer Studio™ IDE. To access the project properties for this example:

1. Right-click the project name (powerdeepsleep_MSP_EXP432P4111_tirtos_ccs) and select *Properties*.

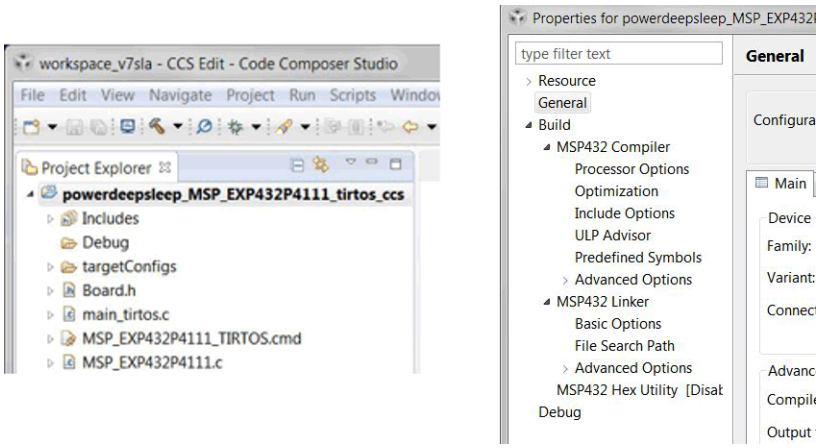


Figure 2. Project Properties

2. From the Properties dialog, click *MSP432 Linker → File Search Path*.
3. The high-temperature-specific power management libraries are specified in the File Search path. The particular driver in question is *drivers_msp432p4x1xi.aem4f* (see Figure 3).

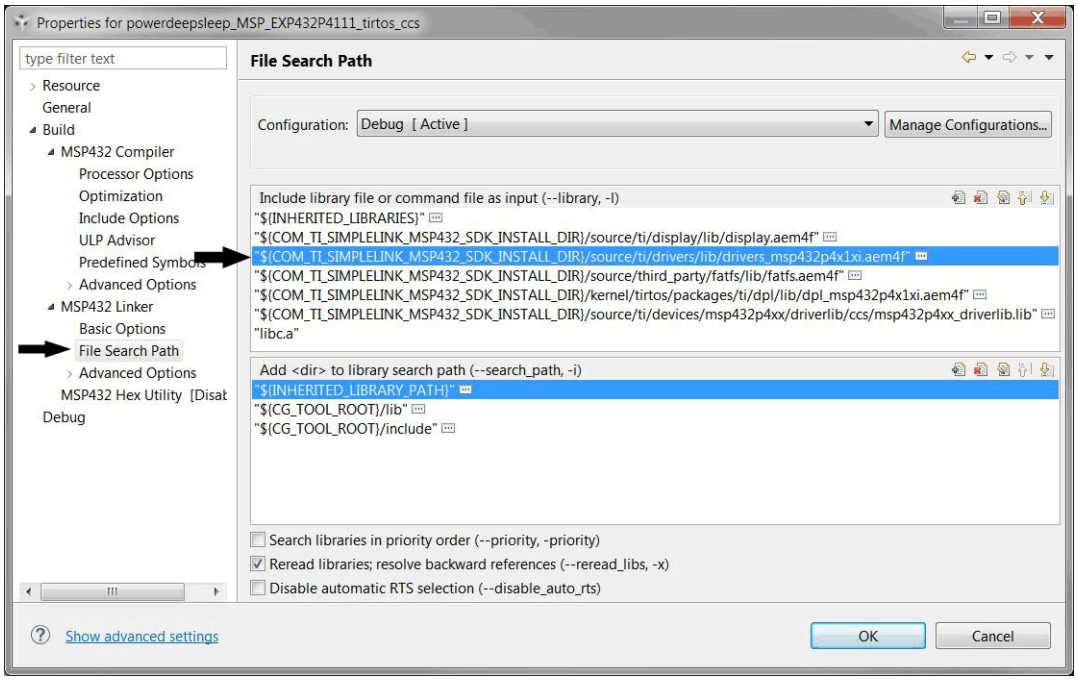


Figure 3. High-Temperature Power Management Libraries

Table 13 lists the appropriate power drivers for each device type (I or T variant).

Table 13. TI Drivers Selection for Power Drivers

File	Compiler	Device	Description
drivers_msp432p401x.aem4f	TI	MSP432P401x	Power drivers for ≤256KB flash devices (85°C)
drivers_msp432p401x.am4fg	GCC		
drivers_msp432p401x.arm4f	IAR		
drivers_msp432p4x1xi.aem4f	TI	MSP432P4x1xI	Power drivers for ≥512KB flash devices (85°C)
drivers_msp432p4x1xi.am4fg	GCC		
drivers_msp432p4x1xi.arm4f	IAR		
drivers_msp432p4xx.aem4f	TI	MSP432P4x1xT	Power drivers for ≥512KB flash devices (105°C)
drivers_msp432p4xx.am4fg	GCC		
drivers_msp432p4xx.arm4f	IAR		

5.3 Driver Library API Changes

As mentioned in Section 5.1, the software differences between the MSP432P401M/R devices and the larger-memory MSP432P4x1V/Y/1 devices depends on the abstraction layer. For applications written using TI drivers, no API changes are visible to the user. For driverlib-level application code, the following changes are apparent:

1. API names

   All flash and system control APIs (*FlashCtl_xxx()* and *SysCtl_xxx()*) have been renamed *SysCtrl_A_xxx* and *FlashCtl_A_xxx* respectively. Some functional changes have also been made as described in the following item, *Include files*.

2. Include files

   No changes needed for #include directives. The driverlib.h #include (shown below) automatically includes the new flash and system control API files if (and only if) the project settings are correct. Following the procedure in Section 5.1 and importing an existing MSP432P4x1V/Y/1 project ensures these settings are correct.

   ```
   #include <ti/devices/msp432p4xx/driverlib/driverlib.h>
   ```

3. In the FlashCtrl_A routines, the names and parameter lists are different. For example:
   `FlashCtl_unprotectSector(FLASH_INFO_MEMORY_SPACE_BANK0,FLASH_SECTOR0);`
   compared to
   `FlashCtl_A_unprotectMemory(uint32_t startAddr, uint32_t endAddr);`

Table 14 and Table 15 highlight the differences in behavior and naming for the flash and system control APIs. For details of behavioral changes, see the DriverLib API guide.

Table 14. SysCtl API Differences

Functional Change?	SysCtl	SysCtl_A
–	uint_least32_t SysCtl_getSRAMSize()	uint_least32_t SysCtl_A_getSRAMSize()
–	uint_least32_t SysCtl_getFlashSize()	uint_least32_t SysCtl_A_getFlashSize()
Yes	N/A	uint_least32_t SysCtl_A_getInfoFlashSize()
–	void SysCtl_rebootDevice()	void SysCtl_A_rebootDevice()
–	void SysCtl_getTLVInfo()	void SysCtl_A_getTLVInfo()
Yes	void SysCtl_enableSRAMBank()	bool SysCtl_A_enableSRAM()
Yes	void SysCtl_disableSRAMBank()	bool SysCtl_A_disableSRAM()
Yes	void SysCtl_enableSRAMBankRetention()	bool SysCtl_A_enableSRAMRetention()
Yes	void SysCtl_disableSRAMBankRetention()	bool SysCtl_A_disableSRAMRetention()
–	void SysCtl_enablePeripheralAtCPUHalt()	void SysCtl_A_enablePeripheralAtCPUHalt()
–	void SysCtl_disablePeripheralAtCPUHalt()	void SysCtl_A_disablePeripheralAtCPUHalt()

Table 14. SysCtl API Differences (continued)

Functional Change?	SysCtl	SysCtl_A
–	void SysCtl_setWDTTimeoutResetType()	void SysCtl_A_setWDTTimeoutResetType()
–	voidSysCtl_setWDTPasswordViolationResetType()	void SysCtl_A_setWDTPasswordViolationResetType()
–	void SysCtl_disableNMISource()	void SysCtl_A_disableNMISource()
–	void SysCtl_enableNMISource()	void SysCtl_A_enableNMISource()
–	uint_fast8_t SysCtl_getNMISourceStatus()	uint_fast8_t SysCtl_A_getNMISourceStatus()
–	void SysCtl_enableGlitchFilter()	void SysCtl_A_enableGlitchFilter()
–	void SysCtl_disableGlitchFilter()	void SysCtl_A_disableGlitchFilter()
–	uint_fast16_t SysCtl_getTempCalibrationConstant()	uint_fast16_t SysCtl_A_getTempCalibrationConstant()

Table 15. FlashCtl API Differences

Function Change?	FlashCtrl()	FlashCtrl_A()
–	void FlashCtl_getMemoryInfo()	void FlashCtl_A_getMemoryInfo()
–	void FlashCtl_enableReadBuffering()	void FlashCtl_A_enableReadBuffering()
–	void FlashCtl_disableReadBuffering()	void FlashCtl_A_disableReadBuffering()
Yes	bool FlashCtl_protectSector()	bool FlashCtl_A_protectMemory()
Yes	bool FlashCtl_unprotectSector()	bool FlashCtl_A_unprotectMemory()
Yes	bool FlashCtl_isSectorProtected()	bool FlashCtl_A_isMemoryRangeProtected()
Yes	NA	bool FlashCtl_A_isMemoryProtected()
–	bool FlashCtl_verifyMemory()	bool FlashCtl_A_verifyMemory()
–	bool FlashCtl_performMassErase()	bool FlashCtl_A_performMassErase()
–	void FlashCtl_initiateMassErase()	void FlashCtl_A_initiateMassErase()
–	bool FlashCtl_eraseSector()	bool FlashCtl_A_eraseSector()
–	bool FlashCtl_programMemory()	bool FlashCtl_A_programMemory()
–	void FlashCtl_setProgramVerification()	void FlashCtl_A_setProgramVerification()
–	void FlashCtl_clearProgramVerification()	void FlashCtl_A_clearProgramVerification()
–	void FlashCtl_enableWordProgramming()	void FlashCtl_A_enableWordProgramming()
–	void FlashCtl_disableWordProgramming()	void FlashCtl_A_disableWordProgramming()
–	uint32_t FlashCtl_isWordProgrammingEnabled()	uint32_t FlashCtl_A_isWordProgrammingEnabled()
–	bool FlashCtl_setReadMode()	bool FlashCtl_A_setReadMode()
–	uint32_t FlashCtl_getReadMode()	uint32_t FlashCtl_A_getReadMode()
–	void FlashCtl_setWaitState()	void FlashCtl_A_setWaitState()
–	uint32_t FlashCtl_getWaitState()	uint32_t FlashCtl_A_getWaitState()
–	void FlashCtl_enableInterrupt()	void FlashCtl_A_enableInterrupt()
–	void FlashCtl_disableInterrupt()	void FlashCtl_A_disableInterrupt()
–	uint32_t FlashCtl_getEnabledInterruptStatus()	uint32_t FlashCtl_A_getEnabledInterruptStatus()
–	uint32_t FlashCtl_getInterruptStatus()	uint32_t FlashCtl_A_getInterruptStatus()
–	void FlashCtl_clearInterruptFlag()	void FlashCtl_A_clearInterruptFlag()
–	void FlashCtl_registerInterrupt()	void FlashCtl_A_registerInterrupt()
–	void FlashCtl_unregisterInterrupt()	void FlashCtl_A_unregisterInterrupt()
–	void FlashCtl_initiateSectorErase()	void FlashCtl_A_initiateSectorErase()
–	uint8_t __FlashCtl_remaskData8Post()	uint8_t __FlashCtl_A_remaskData8Post()
–	uint8_t __FlashCtl_remaskData8Pre()	uint8_t __FlashCtl_A_remaskData8Pre()
–	uint32_t __FlashCtl_remaskData32Post()	uint32_t __FlashCtl_A_remaskData32Post()
–	uint32_t __FlashCtl_remaskData32Pre()	uint32_t __FlashCtl_A_remaskData32Pre()
–	void __FlashCtl_remasKBurstDataPost()	void __FlashCtl_A_remasKBurstDataPost()

Table 15. FlashCtl API Differences (continued)

Function Change?	FlashCtrl()	FlashCtrl_A()
–	void __FlashCtl_remasKBurstDataPre()	void __FlashCtl_A_remasKBurstDataPre()

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