

THIS SPEC IS OBSOLETE

Spec No: 001-25499

Spec Title: CAPACITIVE SENSING - POWER AND SLEEP CONSIDERATIONS AN2360

Sunset Owner: Matthew Leung (MATT)

Replaced by: NONE

Capacitive Sensing - Power and Sleep Considerations

AN2360

Author: Mark Lee
Associated Project: No
Associated Part Family: CY8C21x34
GET FREE SAMPLES HERE
Software Version: PSoC® Designer™ 4.2 Service Pack 3
Associated Application Notes: None

OBSOLETE

Application Notes Abstract

This Application Note describes power consumption and sleep considerations with PSoC CapSense.

Introduction

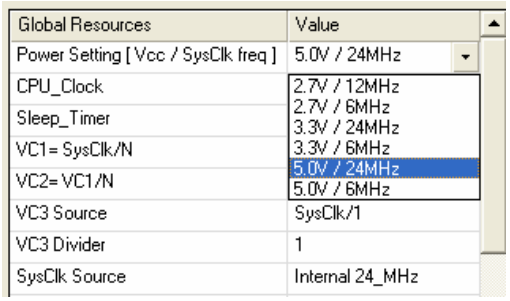
In battery-powered applications, lower current levels translate into longer battery life. With this in mind, a system designer should adjust system parameters so that specifications are met with the minimum active current.

In PSoC capacitive sensing applications, active current can be minimized through a number of programmable settings including the CPU clock speed and sleep mode. This Application Note shows how to minimize active current and incorporate sleep mode into button sensing applications.

PSoC Designer Global Resources Related to Power Consumption

The supply voltage and SysClk settings both affect power consumption. Both are set using the Global Resources window in the Design Editor Interconnect View of PSoC Designer, as shown in Figure 1.

Figure 1. V_{dd} and SysClk Global Resources



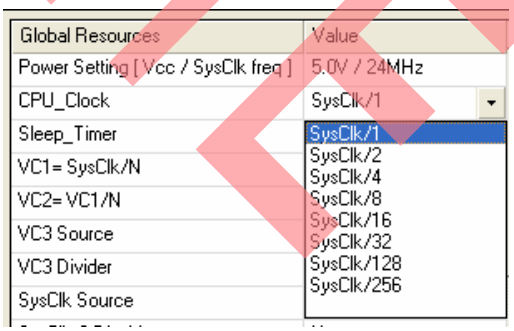
The supply voltage setting selects a trim value for the internal main oscillator. There is a small variation in the frequency of SysClk over V_{DD}. The trim values make the SysClk frequency accurate at each of the three V_{DD} settings.

Normal operation of CapSense buttons can be achieved for any of these settings, but the button scan rate for a SysClk of 6 MHz may be too slow if there are a large number of sensors. The 6 MHz setting will have the lowest power consumption, so for systems with large number of buttons, a balance needs to be found between power consumption and scan rate.

Operation of CapSense sliders is more computationally intensive than buttons. The CPU speed must be kept high to process the slider data. For this reason, 24 MHz and 12 MHz are the only recommended settings of the SysClk for sliders.

The CPU_Clock speed is set using the Global Resources window in the Design Editor Interconnect View of PSoC Designer, as shown in Figure 2. It is a function of SysClk/N, where N = 1,2,4,…,256. To prevent timing issues, especially I^2C-interrupt timing issues, it is recommended that the CPU clock speed only be set to SysClk/1 or SysClk/2.

Figure 2. CPU_Clock Speed Global Resource



SysClk*2 is an optional internal clock derived from SysClk using a frequency doubler. If the project does not require SysClk*2, it is recommended that this function be disabled to save power. Turning off SysClk*2 can save almost 1 mA of supply current with a 5V Vdd.

Power Consumption without CSR User Module

Similar to a standby current, the amount of current required to power the PSoC without any user modules (UMs) placed can be measured. This base level of current can be compared the current required with the user modules to estimate the power required for each function. The average power consumed by a PSoC device is:

$$AveragePowerConsumed = V_{DD} * I_D \quad \text{Equation 1}$$

- V_{DD} = PSoC supply voltage, 2.7-5.0VDC
- I_D = average PSoC supply voltage

Table 1 lists typical values for the supply current for a CY8C21434 PSoC device without any user modules placed. The source for this measurement consists of an infinite loop that keeps the CPU busy toggling a pin. All GPIO pins are set to High Z drive mode to minimize current paths into and out of the device.

Table 1. Typical Supply Current Without CSR UM

V_{DD} (V)	SysClk (MHz)	I_D (mA)	
		SysClk/1	SysClk/2
5.0	24	7.5	5.9
5.0	6	2.7	2.2
3.3	24	4.9	3.7
3.3	6	1.8	1.5
2.7	12	2.2	1.9
2.7	6	1.5	1.3

I_D has one pulsed component linked directly to SysClk. This current is associated with the comparator and the 16-bit counter of the CSR User Module. I_D has another pulsed component linked directly with the CPU clock. This current is associated with CPU processes including CSR UM API function calls and high-level communication of CapSense data. I_D also has a constant component associated with DC bias of some sections of the device. This current is only a function of V_{DD}. These three components of I_D are combined into a single equation that fits the data in Table 1.

$$I_D = (V_{DD} - V_0) * \left(\frac{SysClk}{R1'} + \frac{SysClk/N}{R2'} + \frac{1}{R3} \right) + I_0 \quad \text{Equation 2}$$

- V_0 = 0.38 volts
- I_0 = 0.62 mA
- R1’ = 43.5 ohms*MHz
- R2’ = 27.8 ohms*MHz
- R3 = 13.3 ohms
- N = 1,2,4,8,16,32,64,128,256

Power Consumption with CSR User Module

Adding the CSR User Module to the project increases the supply current, as shown in Table 2. This data is for a project that scans a single button with ScanSpeed set to 3 (a single oscillator cycle). Increasing ScanSpeed to 255 increases the current by only 100 µA, due mainly to the additional switching losses of running the counter for a longer duration.

Table 2. Typical Supply Current With CSR UM, Sleep Mode not Enabled

V_{DD} (V)	SysClk (MHz)	I_D (mA)	
		SysClk/1	SysClk/2
5.0	24	9.3	7.4
5.0	6	3.5	3.0
3.3	24	5.6	4.4
3.3	6	2.2	1.9
2.7	12	2.6	2.2
2.7	6	1.8	1.6

In the PSoC device, sensors are scanned one at a time since the relaxation oscillator and counter are a shared resource for all CapSense measurements. The power required to scan a single button is the same for all sensors. To determine how much power is consumed by a CapSense project with multiple sensors, one must determine how much time is spent scanning the sensors, and how much time is spent doing other tasks, including entering sleep mode. From this set of conditions, the average current can be found. For example, for the simple case of the project that involves only button scans and sleep, the average current is found using Equation (3).

$$AverageCurrent = \frac{N_b * t_{scan} * I_{scan} + t_{sleep} * I_{sleep}}{N_b * t_{scan} + t_{sleep}} \quad \textbf{(3)}$$

- tscan = scan time for a single button
- Nb = number of sensors scanned
- Iscan = average current during a button scan
- t_{sleep} = sleep mode duration
- I_{sleep} = sleep mode current

Typical current values for the button scan operation are listed in Table 2. These values are for continuous scanning of the sensors with sleep mode disabled.

OBSOLETE

Comparing the measured current with and without CapSense, the additional current required for CapSense is listed in Table 3.

Table 3. Additional Current Required for CSR UM (ΔI_D = Difference Between Table 1 and Table 2)

V_{DD} (V)	SysClk (MHz)	ΔI_D (mA)	
		SysClk/1	SysClk/2
5.0	24	1.8	1.5
5.0	6	0.8	0.8
3.3	24	1.2	0.7
3.3	6	0.4	0.4
2.7	12	0.4	0.3
2.7	6	0.3	0.3

Enabling Sleep Mode

Enabling sleep mode requires only two lines of code. One line enables the sleep timer interrupt.

```
INT_MSK0 |= INT_MSK0_SLEEP;
```

The other line puts PSoC into sleep mode.

```
M8C_Sleep;
```

The program listed in Appendix A shows how these two lines of code are used in a simple demonstration project of sleep. The PSoC is configured for one CapSense button, one tick indicator for the start and end of sleep, and High Z GPIO drive mode for all unused IO pins. CapSense is set to period mode, and as a result, the effects of a finger on a button can be seen by monitoring the tick counter with a scope. A finger on the button will stretch out the time between ticks.

Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of power on reset (POR) and low voltage detection (LVD) events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically for about 60 µs at the Sleep System Duty Cycle, which is independent of the sleep interval and typically higher. When the No Buzz bit in the OSC_CR0 register is set, the Sleep System Duty Cycle value is overwritten and the bandgap circuit is forced to be on during sleep. For projects using sleep mode with the CSR User Module, the No Buzz bit can be set to '1' so that the bandgap reference is always on. The following line of code is used to set this bit.

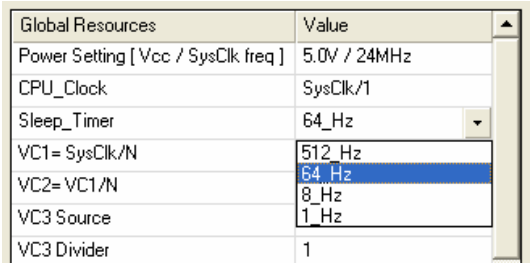
```
OSC_CR0 &= 0b11111111 ;
```

Setting the No Buzz bit, bit 5 of the OSC_CR0 register, is optional. The trade off is faster wake-up from sleep mode versus slightly higher sleep mode current.

Reducing Power Consumption with Sleep Mode

Sleep mode lowers the average current. Sleep mode current is no greater than 5 µA with a 3.3V supply. As shown in the previous section, the longer the sleep mode duration, the lower the average current. The Sleep_Timer frequency is set using the Global Resources window in the Design Editor Interconnect View of PSoC Designer, as shown in Figure 3.

Figure 3. Sleep Timer Global Resource



To sleep for a multiple of one of the standard sleep mode intervals, include consecutive instances of the M8C_Sleep command in the firmware. For example:

```
M8C_Sleep; M8C_Sleep; M8C_Sleep;
```

This command will cause a sleep interval of 46.8 ms if the Sleep_Timer rate is set to 64 Hz (3*15.6 ms).

Table 4 lists some typical average currents for a CapSense project. The No Buzz bit is set to '0'. The button scan time is 100 µs. The supply voltage is 3.3V and SysClk is 24 MHz. In this case, the SysClk/1 and SysClk/2 translate into a CPU speed of 24 MHz and 12 MHz.

Table 4. Typical Supply Current as Function of Sleep Time, CPU Speed and Button Number

Button #	Sleep Time (ms)	I_D (mA)	
		24 MHz CPU	12 MHZ CPU
1	15.6	0.04	0.03
2	15.6	0.08	0.06
4	15.6	0.15	0.12
8	15.6	0.28	0.22
1	125	0.01	0.01
2	125	0.01	0.01
4	125	0.02	0.02
8	125	0.04	0.03

OBSOLETE

Summary

Sleep mode is an effective method for extending battery life in PSoC applications with CapSense. Two important system parameters that affect battery life are the supply voltage and the CPU speed. Design equations are presented that quantify the effect of these two parameters on the supply current. The number of sensors scanned influences the average current in sleep mode. More buttons translate into longer scan rates or higher average current. Through minor firmware adjustments, the system designer can find the right balance between scan rate and current. The ability of PSoC to adapt to changing requirements through firmware is a big asset to engineers designing CapSense systems for long battery life.

OBSOLETE

Appendix. Sleep Mode Example Firmware Listing

```
//--- Sleep Mode example, main.c, for CY8C21434 PSoC
#include <m8c.h>        // part specific constants and macros
#include "PSoCAPI.h"    // PSoC API definitions for all User Modules
void main()
{
  INT_MSK0 |= INT_MSK0_SLEEP ;

//P1[1] toggles to indicate start and end of sleep
//P0[4] has capsense button
//all other ports set to Hi-Z to minimize current
  PRT3DM0=0b00000000; PRT3DM1=0b11111111; PRT3DM2=0b11111111; PRT3DR=0b00000000;
  PRT2DM0=0b00000000; PRT2DM1=0b11111111; PRT2DM2=0b11111111; PRT2DR=0b00000000;
  PRT1DM0=0b00000010; PRT1DM1=0b11111101; PRT1DM2=0b11111101; PRT1DR=0b00000000;
  PRT0DM0=0b00010000; PRT0DM1=0b11101111; PRT0DM2=0b11101111; PRT0DR=0b00000000;

  CSR_1_Start(); CSR_1_SetDacCurrent(0x10,0); CSR_1_SetScanSpeed(3);
  M8C_EnableGInt;

  while(1)
  {
// Scan only switch 0
    CSR_1_StartScan(0,1,0);
// stall until the button 0 scan are finished
    while(!(CSR_1_GetScanStatus() & CSR_1_SCAN_SET_COMPLETE));

//mark start of sleep
    PRT1DR=0b00000010; PRT1DR=0b00000000;
    M8C_Sleep;
//mark end of sleep
    PRT1DR=0b00000010; PRT1DR=0b00000000;
  }
}
```

OBSOLETE

About the Author

Name: Mark Lee
Title: Senior Application Engineer
Contact: olr@cypress.com

OBSOLETE

In March of 2007, Cypress recataloged all of its Application Notes using a new documentation number and revision code. This new documentation number and revision code (001-xxxxx, beginning with rev. **), located in the footer of the document, will be used in all subsequent revisions.

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip," PSoC Designer, and PSoC Express are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/

© Cypress Semiconductor Corporation, 2006-2010. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.