

How to Use Low Power Modes on the Newest S08 Devices

by: **César Manzanarez and Paul Tian**
TICS (Technical Information and Commercial Support)

Contents

1 Introduction

The power consumption of devices and the implications around designing for low power are common topics nowadays.

Freescale offers solutions with energy efficient products that can help you achieve the power consumption needed for your application.

This application note will help you achieve the available power consumption modes on the S08MM and S08LL devices.

The Flexis MM series is ideal for portable medical devices and ultra-low-power medical designs.

The S08LL family is ideal for long battery life in consumer, industrial, and personal health care applications

| | | |
|---|--|----|
| 1 | Introduction..... | 1 |
| 2 | Low Power Overview and Best Practices..... | 1 |
| 3 | System Clock Generation..... | 2 |
| 4 | Modes of Operation..... | 5 |
| 5 | How to Enter Power Modes..... | 9 |
| 6 | Conclusions..... | 14 |
| 7 | References..... | 14 |

2 Low Power Overview and Best Practices

There are many reasons why a designer of an embedded system increasingly strives to stay within tight power consumption budgets.

One reason—money, having a portable device with poor battery life can be the reason for losing in the market against more power efficient competitors.

System Clock Generation

There are other applications where money is not the issue, like in medical devices that are implanted in a patient's body. These devices need to run for years on battery life; replacing the battery means taking the patient into surgery, in this scenario an efficient power solution is critical.

A low power design must take into consideration the trade-off between power consumption and performance, and use every possible feature provided by the device to accomplish the best results.

Both software and hardware designers should be involved in a power efficient design, and estimate the application's power needs early in the planning stage to avoid any late redesigns.

Tips and tricks to consider when designing for low power:

- Set the pins to a known state. For this family of devices—Never leave floating pins. Floating pins affect the overall power consumption
- Select and configure the desired clock mode. Keep in mind that higher core frequency means higher power consumption.
- Higher operating temperature increases power consumption.
- Disable clock to unused modules

3 System Clock Generation

There are two different system clock generation modules that are used, depending on the device family selected.

The S08MM family contains a Multipurpose Clock Generator module (MCG), whereas the S08LL family contains an Internal Clock Source (ICS) module.

Clock module comparison — The MCG clock module has been designed with high-end applications in mind, whereas the ICS is intended for smaller, lower power, and cost-sensitive applications. The MCG is the only module to contain a PLL.

Depending on application requirements, power can be reduced by selecting the best system-clock-generation option.

For more information about the system clock generation modules refer to the device's Reference Manual.

3.1 Multipurpose clock generator (MCG)

The multipurpose clock generator (MCG) module provides several clock source choices for this device.

The module contains a frequency-locked loop (FLL) and a phase-locked loop (PLL) that are controllable by an internal or an external reference clock. The module can select either of the FLL or PLL clocks or either of the internal or external reference clocks as a source for the MCU system clock. The selected clock source is passed through a reduced bus divider that allows a lower output clock frequency to be derived.

For USB operation on the MC9S08MM128 series, the MCG must be configured for PLL engaged external (PEE) mode to achieve a MCGOUT frequency of 48 MHz.

NOTE

MCGPLLSCLK is not used on the MC9S08MM128 series.

The MC9S08MM128 series devices are unique, because they support a Time Of Day module which includes a dedicated oscillator. The TOD oscillator can also be used as the reference clock in the MCG.

Table 1. MCG configuration considerations

| | Clock-Reference Source (Internal) | Clock-Reference Source (External) |
|--------------------|--|---|
| FLL Engaged | <p>FLL engaged-internal reference (FEI)</p> <p>Default mode out of reset (fBus = 8 MHz)</p> <p>2 MHz < fBus < 24 MHz</p> <p>Medium power (Higher than FBI or FBE)</p> <p>Medium clock accuracy (precision trimming of internal reference allows 0.2% resolution and 2% deviation over temperature and voltage)</p> <p>Lowest system cost (no external components required)</p> | <p>FLL engaged-external reference (FEE)</p> <p>2 MHz < fBus < 24 MHz</p> <p>Medium power (approximately the same as FEI depending on range selected)</p> <p>Good clock accuracy</p> <p>Medium/High system cost (crystal, resonator or external clock source required)</p> |
| FLL Bypassed | <p>FLL bypassed-internal reference (FBI)</p> <p>Low power (less than FEI or FEE)</p> <p>Medium clock accuracy (precision trimming of internal reference allows 0.2% resolution and 2% deviation over temperature and voltage)</p> <p>Lowest system cost (no external components required)</p> | <p>FLL bypassed-external clock (FBE)</p> <p>Lowest power of FLL modes (less than FEI, FEE and FBI)</p> <p>Highest clock accuracy</p> <p>Medium/High system cost (Crystal, resonator or external clock source required)</p> |
| PLL Engaged | NA | <p>PLL engaged-external clock (PEE)</p> <p>2MHz<=fBus <= 24 MHz</p> <p>Higher power than all FLL modes</p> <p>Highest clock accuracy</p> <p>Medium/High system cost (Crystal, resonator or external clock source required)</p> |
| PLL Bypassed | NA | <p>PLL bypassed-external clock (PBE)</p> <p>Higher power than all FLL modes</p> <p>Higher clock accuracy than FEE, but less than FBE or PEE</p> <p>Medium/High system cost (Crystal, resonator or external clock source required)</p> |
| Bypassed Low Power | <p>Bypassed low power internal reference (BLPI)</p> <p>Same as FBI, but with FLL and PLL disabled for lower power consumption.</p> | <p>Bypassed low power external reference (BLPE)</p> <p>Same as FBE and PBE, but with FLL and PLL disabled for lower power consumption.</p> <p>Optimal for switching between FBE and PBE</p> |

NOTE

The MCG enters Stop mode whenever the MCU enters Stop mode. The MCG returns to the state that was active before the MCU entered Stop mode, unless a reset occurs while in Stop mode.

ICS modes of operation — There are seven modes of operation for the ICS:

- FEI
- FEE
- FBI
- FBILP
- FBE
- FBELP
- Stop

For more details on each operation mode please refer to Chapter “Internal Clock Source” in the device’s reference manual

4 Modes of Operation

The operating modes described in this chapter apply for the MC9S08MM and MC9S08LL series.

The available modes of operation are the following:

- Active background mode for code development
- Bootloader mode
- Run mode
- LPRUN mode
- Wait mode
- LPWAIT mode
- Stop3 mode
- Stop2 mode

This chapter will discuss all modes of operation relevant to the topic, that is all modes except the first two listed above (active background mode and bootloader mode)

NOTE

For a full description of the operating modes refer to the device’s Reference Manual, search for the chapter “Modes of operation”

This a brief overview of each operating mode.

4.1 Run mode

Run is the normal operating mode. This mode is selected after any internal reset including LVD and when both the BKGD/MS and BLMS pins are high, after a POR exit or a BDC forced reset. In this mode, the CPU executes code from the internal memory with execution beginning at the address fetched from the memory at 0xFFFFE:0xFFFF after reset.

4.2 Low-Power Run mode (LPRun)

In the low-power run mode, the on-chip voltage regulator goes into into a standby state. This state uses the minimum power consumption necessary for the CPU function. Power consumption is reduced by disabling the clocks to all unused peripherals by clearing the corresponding bits in the SCGC1 and SCGC2 registers.

NOTE

Important considerations

- BLPE is the selected clock mode for the MCG
- The bus frequency is less than 125 kHz

Modes of Operation

- Flash programming and erasing is not allowed
- If enabled, the TOD must be configured as an external clock source or the LPO oscillator

Exit sources

- Any interrupt request or reset

4.3 Wait mode

The Wait mode is entered by executing a WAIT instruction. After execution of the WAIT instruction, the CPU enters a low-power state in which it is not clocked. The I bit in the condition code register (CCR) is cleared when the CPU enters wait mode, enabling interrupts. When an interrupt request occurs, the CPU exits wait mode and resumes processing, it begins with the stacking operations and leads to the interrupt service routine.

Exit sources

- Any interrupt request or reset

4.4 Low-Power Wait mode (LPWait)

The low-power wait mode is entered by executing a WAIT instruction while the MCU is in Low-Power Run mode. In the LPWait mode, the on-chip voltage regulator remains in a standby state (as in the LPRun mode). This state uses the minimum power consumption necessary for modules to maintain functionality. Power consumption is reduced by disabling the clocks to all unused peripherals by clearing the corresponding bits in the SCGC register.

The same restrictions on the LPRun mode apply to LPWait mode.

NOTE

The LPRun mode must be entered first in order to enter LPWait mode.

Exit sources:

- Any interrupt request or reset

4.5 Stop modes

One of two stop modes is entered after execution of a STOP instruction when STOPE in SOPT1 is set.

The selected mode is entered following the execution of a STOP instruction depending on the settings selected as shown in Table 2.

Table 2. Stop Mode Selection

| STOPE | ENBDM | LVDE | LVDSE | PPDC | Stop mode |
|-------|-------|---------------------|-------|------|--|
| 0 | x | x | | x | Stop modes disabled; illegal opcode reset if Stop instruction executed |
| 1 | 1 | x | | x | Stop 3 with BDM enabled |
| 1 | 0 | Both bits must be 1 | | x | Stop 3 with voltage regulator active |
| 1 | 0 | Either bit 0 | | 0 | Stop 3 |
| 1 | 0 | Either bit 0 | | 1 | Stop 2 |

4.6 Stop2 mode

Most of the MCU internal circuitry is powered off in stop2, with the exception of the RAM. Upon entering stop2, all I/O pin control signals are latched so that the pins retain their states during stop2.

NOTE

This is the lowest power consumption mode

Exit sources:

- Asserting reset pin or external IRQ pin.

4.7 Stop3 mode

In this mode the states of all of the internal registers and logic, RAM contents, and I/O pin states are maintained.

Exit sources

- Asserting reset
- An interrupt from any of the following sources — TOD interrupt, USB resume interrupt, LVD, ADC, IRQ, KBI, SCI, and ACMP

4.8 Overview of enabled peripherals

Depending on the power mode selected some peripherals and modules are enabled or disabled. Refer to Table 3 for available components in Stop mode.

Table 3. Enabled peripherals in Stop Modes

| Peripheral | Mode | |
|-------------------------|---------------|----------------------------|
| | Stop2 | Stop3 |
| CPU | Off | Standby |
| RAM | Standby | Standby |
| Flash | Off | Standby |
| Parallel port registers | Off | Standby |
| ADC | Off | Optionally On ¹ |
| PRACMP | Off | Optionally On ² |
| MCG | Off | Optionally On ³ |
| IIC | | Standby |
| DAC | | Standby |
| VREF | Optionally On | Optionally On ⁴ |
| SCIx | Off | Standby |
| SPIx | Off | Standby |
| OPAMPx | Off | On |
| PDB | Off | Standby |
| TRIAMPx | Off | On |

Table continues on the next page...

Table 3. Enabled peripherals in Stop Modes (continued)

| Peripheral | Mode | |
|---------------------------|---------------|----------------------------|
| | Stop2 | Stop3 |
| TOD | Optionally On | Optionally On |
| CRC | Off | Standby |
| CMT | Off | Standby |
| TPMx | Off | Standby |
| System voltage regulator | Off | Standby |
| XOSC2 | Off | Optionally On ⁵ |
| XOSC1 | Optionally On | Optionally On ⁶ |
| I/O pins | States Held | States Held |
| USB (SIE and transceiver) | Off | Optionally On ⁷ |
| USB 3.3 V regulator | Off | Standby |
| USB RAM | Standby | Standby |

1. Requires the asynchronous ADC clock and LVD to be enabled, else in standby.
2. If ACGBS in ACPSC is set, LVD must be enabled, else in standby.
3. IRCLKEN and IREFSTEN set in MCGC1, else in standby.
4. PS[3:0] in SC does not equal 0 before entering stop, else off.
5. ERCLKEN and EREFSTEN set in MCGC2, else in standby. High-frequency range (RANGE in MCGC2 set) requires the LVD to also be enabled in stop3.
6. ERCLKEN and EREFSTEN set in MCGC2, else in standby. High-frequency range (RANGE in MCGC2 set) requires the LVD to also be enabled in stop3.
7. USBEN in CTL is set and USBPHYEN in USBCTL0 is set, else off.

For the rest of the power modes all peripherals are enabled. Refer to Table 4 for the characteristics of the rest of the power modes.

Table 4. Peripherals in Run and Wait modes

| Modes of Operation | CPU and Peripheral Clocks |
|---|---|
| Run mode | on. MCG in any mode |
| LPRUN mode | Low frequency required. MCG in BLPE mode only |
| Wait mode — Assumes WAIT instruction executed | CPU clock is off, peripheral clocks on. MCG state same as RUN mode. |
| LPWAIT mode — Assumes WAIT instruction executed | CPU clock is off, peripheral clocks at low speed. MCG in BLPE mode |

4.9 Allowed power mode transitions

Figure 2 illustrates mode state transitions allowed between the power modes. RESET must be asserted low or a TOD interrupt must occur to exit STOP2. Interrupts suffice for the other STOP and WAIT modes.

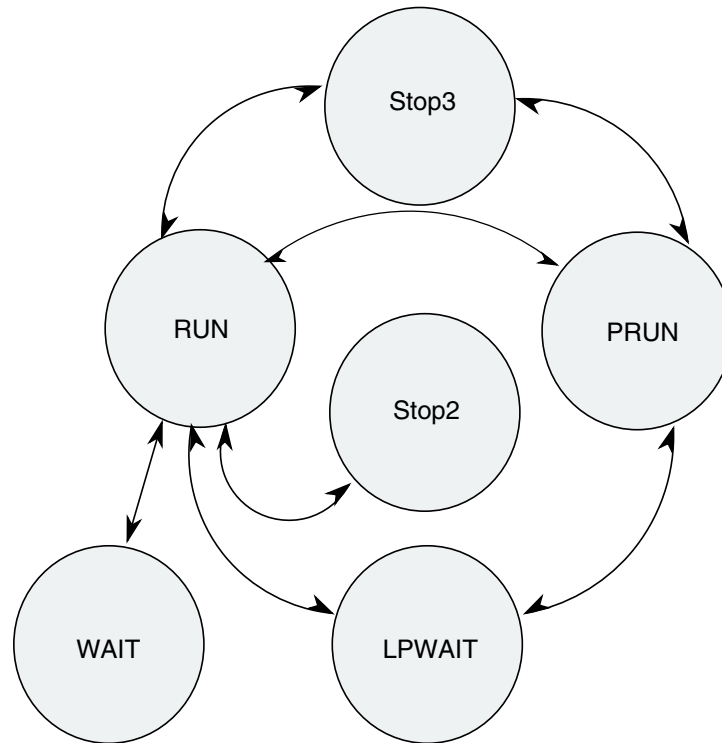


Figure 2. Allowed power mode transitions

5 How to Enter Power Modes

This chapter reviews the necessary conditions to enter the available power saving modes as well as a guide to enter each power saving mode on the Low Power Template project “LPTemplate_S08MM128” and “LPTemplate_MC9S08LL16.”

5.1 Low-power Run mode (LPRun)

MC9S08MM128 typical consumption — 20 uA

MC9S08LL16 typical consumption — 21 uA

All modules are off, voltage regulator in standby, and running from external clock at 16.384 KHz bus frequency.

Necessary conditions:

1. Initialize all GPIOs to a known state, so no pins are left floating.
2. Set the system clock to bypasses Low Power External reference (BLPE mode) The bus frequency should be less than 125 kHz.
3. Disable clocks to all unused modules, Power consumption is reduced by disabling the clocks to all unused peripherals (leave only TOD peripheral if needed)
4. Disable low voltage detect function This is done by clearing either LVDE or LVDSE bit in SPMSC1 register
5. Set LPR bit in SPMSC2 register

LPTemplate project settings:

1. Enable LPRun macro and disable all others

```

/*Available Power Modes*/
/*Note: To enter LPWait you also need to enable LPRun*/
  
```

How to Enter Power Modes

```
#define LPRun //Used for LPRun mode, Must select vfnSetBLPE mode
//define LPWait //Used for LPWait mode, Must select vfnSetBLPE mode
//define Wait //Used for Wait mode
//define STOP //Used for Stop modes
//define STOP2 //Only enable when using STOP2
```

2. Enable proper clock mode (BLPE)

```
/*Clock Modes*/
```

```
void vfnSetFBE(void); //Eternal 16 MHz crystal, 8 MHz bus clock
void vfnSetFEI(void); //Internal references 1 MHz bus clock
void vfnSetBLPE(void); //External 32.768 KHz crystal, 16.384 KHz bus clock
```

```
#define SetClockMode vfnSetBLPE()
```

3. Load project

NOTE

To measure power consumption make sure the device is running in stand-alone mode.

5.2 Wait mode

MC9S08MM128 typical consumption — 1 mA

MC9S08LL16 typical consumption — 0.8 mA

(All modules off, running from internal clock at 1 MHz bus frequency)

Necessary conditions:

1. Initialize all GPIOs to a known state so that no pins are left floating
2. Set the system clock desired
3. Disable clocks to all unused modules

Initialize Time of Day (TOD) module if desired

4. Disable clocks to all unused modules

Power consumption is most reduced by disabling the clocks to all unused peripherals (leave only TOD peripheral if needed)

5. Execute WAIT instruction

LPTemplate project settings:

1. Enable WAIT macro and disable all others

```
/*Available Power Modes*/
/*Note: To enter LPWait you also need to enable LPRun*/

//define LPRun //Used for LPRun mode, MUST select vfnSetBLP mode
//define LPWait //Used for LPWait mode, MUST select vfnSetBLP mode
#define WAIT //Used for Wait mode
//define STOP // Used for Stop modes
//define STOP2 //Only enabled when using STOP2
```

2. Enable desired clock mode FEI

```
/*Clock Modes*/
```

```
void vfnSetFBE(void); //External 16MHz Crystal, 8MHz bus clock
void vfnSetFEI(void); //Internal Reference 1MHz bus clock
void vfnSetBLPE(void); //External 32.768 KHz crystal, 16.384 KHz bus clock
```

```
#define SetClockMode vfnSetFEI()
```

With the above settings the MCU will enter WAIT mode and will exit every second due to the TOD Interrupt.

3. Load project

NOTE

To measure power consumption make sure the device is running in stand- alone mode.

5.3 Low-power Wait mode (LPWait)

MC9S08MM128 typical consumption — 10 uA

MC9S08LL16 typical consumption — 1.3 uA

All modules running from the external clock at 16.384 KHz bus frequency.

Necessary conditions:

1. Initialize all GPIOs to a known state so that no pins are left floating.
2. Set the system clock to Bypasses Low Power External reference (BLPE mode) The bus frequency should be less than 125 kHz.
3. Disable clocks to all unused modules Initialize Time of Day (TOD) module if desired Power consumption is most reduced by disabling the clocks to all unused peripherals (leave only TOD peripheral if needed)
4. Disable Low Voltage Detect function This is done by clearing either LVDE or LVDSE bit in SPMSC1 register
5. Set LPR bit in SPMSC2 register

NOTE

Up to this point LPRun is entered.

6. Execute WAIT instruction LPTemplate project settings:

1. Enable LPRun and LPWait macros; disable all others

```

/*Available Power Modes*/
/*Note: To enter LPWait you also need to enable LPRun*/

#define LPRun           //Used for LPRun mode, Must select vfnSetBLPE mode
#define LPWait         //Used for LPWait mode, Must select vfnSetBLPE mode
//#define Wait         //Used for Wait mode
//#define STOP         //Used for Stop modes
//#define STOP2        //Only enable when using STOP2
    
```

2. Enable proper clock mode (BLPE)

```

/*Clock Modes*/

void vfnSetFBE(void); //Eternal 16 MHz crystal, 8 MHz bus clock
void vfnSetFEI(void); //Internal references 1 MHz bus clock
void vfnSetBLPE(void); //External 32.768 KHz crystal, 16.384 KHz bus clock

#define SetClockMode vfnSetBLPE()
    
```

3. Load project

NOTE

To measure power consumption make sure the device is running in stand- alone mode.

5.4 Stop2 mode

MC9S08MM128 typical consumption — 0.39 uA

How to Enter Power Modes

MC9S08LL16 typical consumption — 0.35 uA

Necessary conditions:

1. Initialize all GPIOs to a known state so that no pins are floating
2. Set the system clock desired
3. Disable clocks to all unused modules. Power consumption reduced by disabling the clocks to all unused peripherals (leave only TOD peripheral if needed)
4. Enable STOP mode by setting SOPT1_STOPE bit
5. Disable Low Voltage Detect function. This is done by clearing either the LVDE or LVDSE bit in the SPMSC1 register
6. Set PPDC and PPDE bits in the SPMSC2 register to indicate the Stop2 power mode is enabled and enable the partial power down feature.
7. Execute STOP instruction

LP Template project settings:

1. Enable STOP and STOP2 macros; disable all others

```
/*Available Power Modes*/
/*Note: To enter LPWait you also need to enable LPRun*/

// #define LPRun //Used for LPRun mode, MUST select vfnSetBLPEmode
// #define LPWait //Used for LPWait mode, MUST select vfnSetBLPEmode

// #define Wait //Used for Wait mode
#define STOP //Used for STOP mode
#define STOP2 //Only enabled when using STOP2
```

2. Enable proper clock mode (FEI)

```
/*Clock Modes*/

void vfnSetFBE (void); //External 16MHz, 8MHz bus clock
void vfnSetFEI (void); //Internal Reference 1MHz bus clock
void vfnSetBLPE (void); //External 32.768 KHz crystal, bus clock 16.384KHz
clock
#define SetClockMode vfnSetFEI ()
```

3. Select Stop Mode 2

```
/*Stop mode selection*/

void vfnStop3 (void);
void vfnStop2 (void);

#define SetSystemOptionsRegisters void vfnStop2 ()
```

4. Load project

NOTE

To measure power consumption make sure the device is running in stand-alone mode.

5.5 Stop3 Mode

MC9S08MM128 typical consumption: 0.55 uA

MC9S08LL16 typical consumption: 0.40 uA

Necessary conditions:

1. Initialize all GPIOs to a known state so that no pins are left floating
2. Set the system clock desired
3. Disable clocks to all unused modules Power consumption is most reduced by disabling the clocks to all unused peripherals (leave only TOD peripheral if needed)
4. Enable STOP mode by setting SOPT1_STOPE bit

5. Disable Low Voltage Detect function This is done by clearing either LVDE or LVDSE bit in SPMSC1 register
6. Make sure PPDC bit is zero in SPMSC2 register to indicate Stop3 power mode enabled.
7. Execute STOP instruction

LPTemplate project settings:

1. Enable STOP macro; disable all others

```

*/Available Power Modes*/
/*Note: To enter LPWait you also need to enable LPRun*/

// #define LPRun           //Used for LPRun mode, MUST select vfnSetBLPE mode
// #define LPWait         //Used for LPWait mode, MUST select vfnSetBLPE mode
// #define Wait           //Used for Wait mode
#define STOP             //Used for Stop modes
// #define STOP2         //Only enabled when using STOP2

```

2. Enable desired clock mode FEI

```

/*Clock Modes*/

void vfnSetFBE (void); //External 16MHz Crystal, 8MHz busclock
void vfnSetFEI (void); //Internal Reference 1MHz bus clock
void vfnSetBLPE (void); //External 32.768, 16.384 KHz bus clock

#define SetClockMode vfnSetFEI()

```

3. Select Stop Mode 3

```

/*Stop mode selection*/

void vfnStop3 (void); void vfnStop2 (void);

#define SetsystemOptionsregisters vfnStop3()

```

4. Load project

NOTE

To measure power consumption make sure the device is running in stand-alone mode.

For all modes:

NOTE

If TOD wakeup is not desired comment InitializeTOD line in the main function and disable all modules on the vfnDisableModules function.

Main Function

```

#ifdef WAIT

/*GPIO Default Initialization*/
vfnGPIOInit();

/*Configure the selected clock source (macro) */
SelectClockMode;

/*Configure TOD clock source and prescaler*/
//InitializeTOD; //Comment this line if TOD wakeup is not desired

/*Disable clock to unused modules*/
vfnDisableModules();

```

vfnDisableModules

```

void vfnDisableModules(void)
{
/*Use the Gating Registers to disable the clock to the
used peripherals*/

```

Conclusions

```
SCGC1 = 0x00;
SCGC2 = 0x00; //Measure current consumption with all clocks disabled
//SCGC2 = SCGC2_TOD_MASK; //For TOD wakeup leave TOD clock enabled
SCGC3 = 0x00;
}
```

6 Conclusions

When designing for low power you must take into consideration both hardware and software, both come into play to design a successful power efficient application.

To achieve the lowest power consumption make sure to set the GPIO status accordingly, based on your board and application, such as pull-up resistors, triode, and so on.

Remember the basics — Set pins to a known state , higher core frequency and higher temperature both translate to higher power consumption, disable clocks to unused modules.

7 References

The chapters in this application note contain in summary the most important details.

For more details on the topics for this document refer to the following:

- Information about power consumption — Refer to the device datasheet Search for “Supply Current Characteristics”
- Visualize current adder per module enabled — Refer to table “Stop Mode Adders”
- Information and details about power modes — Refer to the device Reference Manual Search for Chapter 3 “Modes of Operation”

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2012 Freescale Semiconductor, Inc.