Freescale Semiconductor
Application Note

Document Number: AN3387
Rev. 0, 10/2007

HCS08 Automotive Low-Power Modes

by: Matt Ruff
Automotive Systems Engineering
Austin, TX

1 Introduction

Microcontroller (MCU) based electronic-control units (ECU) are rapidly replacing mechanical and passive electronic systems in automobiles. Each ECU node demands current from the electrical system even when the engine is not running, leaving the battery as the only source of power. ECU designers face the challenge of adding more functionality to the vehicle while keeping the total key-off current demand below the fixed limit of the battery's available energy.

This application note focuses on using the S08DZ MCU to achieve low power consumption. The S08DZ is a member of the high-performance HCS08 family of 8-bit microcontrollers. Common family features include a 40 MHz HCS08 CPU, an enhanced instruction set, controller area network (CAN), and a background-debug controller (BDC) that provides an easy interface for in-system, real-time debugging. See the device data sheet, Freescale document MC9S08DZ60, for a more

Contents

freescale semiconductor

complete description of part features. Always refer to the data sheet for the most current specification (http://www.freescale.com).

The S08DZ MCU has additional features for achieving low power consumption. These features provide effective flexibility for the user and can provide ideal conditions for many applications.

Dynamically changing the operating mode of the microcontroller can achieve the lowest net power consumption. For example, when the microcontroller needs to perform a computationally intensive task, the MCU can be run as fast as possible for the shortest possible time. Then, when the ECU is not needed, an extremely low current sleep mode can be entered, where as much of the silicon is shut off as possible. Clock modules offer options to change clock modes and clock frequencies allowing selection of the best power consumption for different tasks. Stop and wait modes allow further current savings by shutting down modules inside the MCU. The figure below shows how this strategy might be implemented.

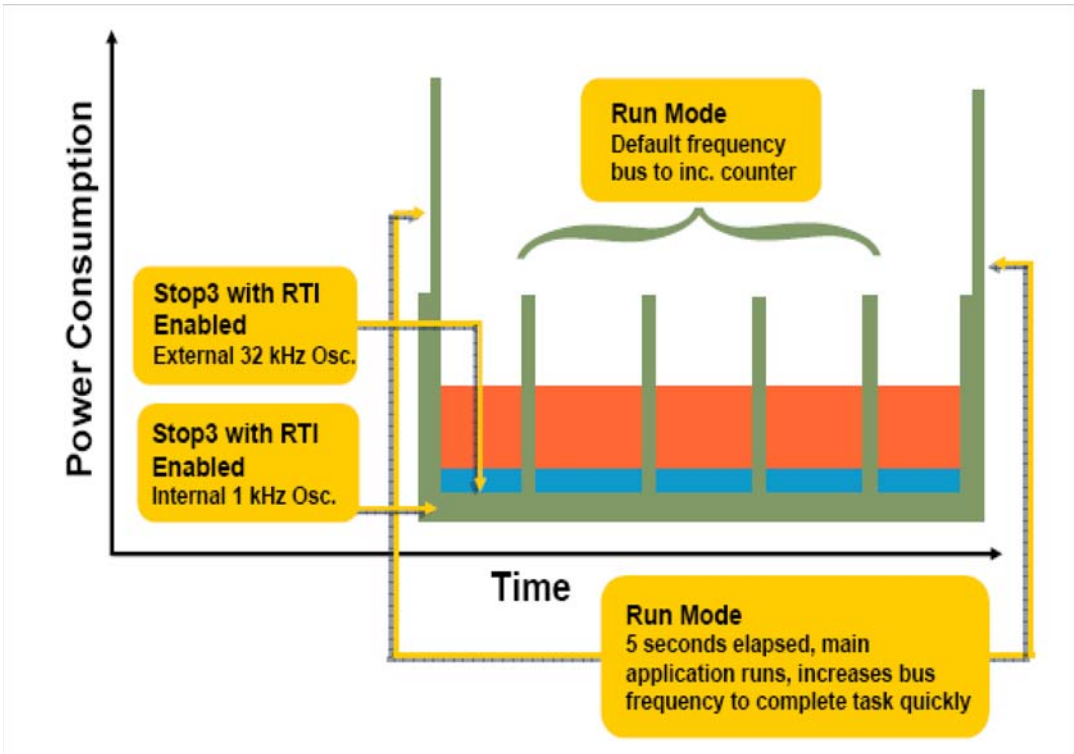

Figure 1.

2 System Clock Generation

The S08DZ features a multi-clock generator (MCG) module capable of using an external (crystal, resonator, or square wave) or an internal source to generate the system clock. Also, in the MCG, frequency-locked loop (FLL) or phase-locked loop (PLL) stages can boost the external or internal clock source to a higher frequency. The S08DZ can use a low-range (32 kHz to 100 kHz) or high-range (1 MHz to 16 MHz) crystal or resonator. Upon any system startup (from stop or reset), the MCU uses the low-power internal clock source, that eliminates a long startup time. Depending on application

requirements, power can be reduced by selecting the best system-clock-generation option. Table 1 shows configuration considerations among clock modes. For more information about the S08DZ clock options, refer to the device data sheet MC9S08DZ60.

Table 1. MCG Configuration Considerations

	Clock-Reference Source (Internal)	Clock-Reference Source (External)
FLL Engaged	FLL engaged-internal reference (FEI) Default mode out of reset (f_{Bus} = 8 MHz) 2 MHz < f_{Bus} < 20 MHz Medium power (Higher than FBI or FBE) Medium clock accuracy (After IRG is trimmed) Lowest system cost (no external components required)	FLL engaged-external reference (FEE) 2 MHz < f_{Bus} < 20 MHz Medium power (approximately the same as FEI depending on range select) Good clock accuracy Medium/High system cost (crystal, resonator or external clock source required)
FLL Bypassed	FLL bypassed-internal reference (FBI) 2 kHz < f_{Bus} < 20 kHz. Low power (less than FEI or FEE) Medium clock accuracy (after IRG is trimmed) Lowest system cost (no external components required)	FLL bypassed-external clock (FBE) f_{Bus} range <= 2.5 MHz when crystal or resonator is used. Lowest power of non-LP modes Highest clock accuracy Medium/High system cost (Crystal, resonator or external clock source required)
PLL Engaged	NA	PLL engaged-external clock (PEE) f_{Bus} <= 20 MHz Higher power than all FLL modes Highest clock accuracy Medium/High system cost (Crystal, resonator or external clock source required)
PLL Bypassed	NA	PLL bypassed-external clock (PBE) f_{Bus} <= 8 MHz when crystal or resonator is used, up to 20 MHz when using an external clock. Higher power than all FLL modes Higher clock accuracy than FEE, but less than FBE or PBE Medium/High system cost (Crystal, resonator or external clock source required)
Bypassed Low Power	Bypassed low power internal reference (BLPI) Same as FBI, but with FLL and PLL disabled for lower power consumption.	Bypassed low power external reference (BLPE) Same as FBE and PBE, but with FLL and PLL disabled for lower power consumption. Optimal for switching between FBE and PBE

3 Operation Modes

After reset, operation mode is a run mode in which the CPU is active and peripherals can be enabled. By executing a WAIT instruction, the MCU enters wait mode. In wait mode, power is reduced because the CPU is not clocked. To reduce power consumption further, stop mode can be used. When a STOP instruction executes, one of two stop modes are entered. Stop2 and stop3 each provide different levels of operation that reduce power consumption. Table 2 describes stop mode behaviors.

Table 2. Stop Mode Behaviors

Mode	CPU, digital peripherals, flash	RAM	ICG	ADC	Pin interrupt	Regulator	I/O Pins	RTC
Stop2	Off	Standby	Off	Disabled	Off	Standby	States held	Optionally on
Stop3	Standby	Standby	Standby[1]	Optionally on	Optionally on	Standby	States held	Optionally on

1. Crystal oscillator can be configured to run in stop3. Please see the ICG registers.

3.1 Real-Time Counter (RTC)

The RTC can be used to exit stop2 or stop3. In stop3, it can be configured to use an external or one of two internal references (LPO or IRCLK). In stop3, using an internal reference reduces power consumption further than using the external reference. In stop2, only the LPO clock can be used. The RTC module can be configured to achieve a variety of real-time interrupt periods. The 1 kHz reference has a tolerance of about ±30%; therefore, the wakeup times are approximate when the RTC uses the LPO clock.

3.2 Low-Voltage Detect (LVD)

The S08DZ MCU can enable or disable low-voltage detection when in stop3 mode. If low voltage detection is enabled in stop, only stop3 can be entered. If the LVDSE bit in SPMSC1 is set, then upon execution of a STOP instruction, stop3 is entered, regardless of the state of the PDC and PPDC bits in SPMSC2.

3.3 Operating Voltage Ranges

The S08DZ MCU is specified to operate from 5.5 V down to 2.7 V

3.4 Internal Voltage Regulator

The S08DZ uses an internal voltage regulator to provide about 2.4 V for the internal power supplies to the CPU and most peripherals. The regulator is always on when the MCU is in run and wait modes. In stop2 and stop3 modes, the regulator enters a state that results in looser regulation, thereby saving power.

4 Description of Low-Power Modes

4.1 Stop Modes

With the introduction of the HCS08 family of MCUs, two new forms of stop mode were introduced, resulting in three stop modes: stop1, stop2, and stop3. Stop3 is functionally equivalent to a stop mode on the HC08 MCUs, although stop1 and stop2 are new lower power modes. The S08DZ and other 5 V HCS08 MCUs do not implement stop1 mode.

4.2 Stop2

Stop2 provides lower standby currents than stop3. Stop2 is a partial power-down mode in which the internal voltage regulator enters a loose regulation mode, thereby reducing the current consumption by reducing the regulator power output.

In stop2, the RAM remains powered, and the states of all the I/O pins are latched in their state prior to entering stop2. Pins configured as inputs remain inputs, and output pins drive the last known state. However, the voltage regulator powers all other peripherals, and they are powered down and cannot be used such as LVD and pin interrupts. The ADC is also turned off and cannot be used.

Although the I/O pins retain their state in stop2, all registers are powered down. To preserve the values of any register such as SCI, timer, or port data, copy them into RAM before entering stop2.

Exit stop2 by asserting the $\overline{\text{RESET}}$ pin or IRQ pin low. The IRQ is active low in this mode, regardless of its configuration before entering stop2. The IRQ pin must be enabled prior to entering stop2 mode.

In addition to the $\overline{\text{RESET}}$ or IRQ pin, in stop2, the RTC can be enabled and used for wakeup without depending on an external input. However, only the internal 1 kHz LPO oscillator can be the clock source for the RTC in stop2. When the RTC event occurs, stop2 is exited as if a POR occurred.

Exiting stop2 results in the registers resetting to their POR values with the following exception. The PPDF bit in the SPMSC2 register is set, and the I/O pins remain latched in their current state until a logic 1 is written to the PPDACK bit in SPMSC2. The PPDF bit can be used as a flag to branch to a stop2 recovery routine. To maintain the current state of the I/O pins, copy the saved register values in RAM back into their respective locations before writing the PPDACK bit. Any un-restored register reverts to its POR value and any corresponding I/O pins also revert to their POR state. Upon stop2 recovery, normal peripheral operation does not begin until the PPDACK is written because the I/O is latched.

4.3 Stop3

Stop3 in HCS08 family of devices is functionally equivalent to the stop mode on HC08 family MCUs. All I/O pins are latched in the state they were in prior to executing the stop command. In stop3, several options are available which are not available in the other stop modes. Stop3 is the only stop mode where LVD protection can be enabled during stop. In fact, if the LVDSE bit in the SPMSC2 register is set, only stop3 can be entered.

Also, the OSCSTEN bit can be set so that the external crystal oscillator is enabled but the FLL, PLL, and the clock to the rest of the MCU are off. The OSCSTEN option can be used to avoid long oscillator startup times. This also allows the RTI to use an external clock source as a reference for the real-time interrupts. For time-critical applications using the external reference provides for precise RTC intervals.

Exit from stop3 can be less intrusive than the exit of stop2. If an interrupt source such as IRQ, pin interrupts, or RTC is used to exit stop3, the MCU services the interrupt and then continues operation at the instruction following the stop instruction. It is not necessary to initialize peripherals after exiting stop3. Stop3 can also be exited by asserting the $\overline{\text{RESET}}$ pin. In this case, the MCU fetches the reset vector, and registers and peripherals are placed in their reset state.

4.4 Wait

Wait mode consumes less power than run mode. In this mode, clocks to the CPU are turned off to reduce power. All other peripherals can be enabled in wait. In this mode, any interrupt can be used to exit wait. A common application is executing a WAIT command and then waiting for an SCI or SPI interrupt so operation continues. After exit from wait via an interrupt, the MCU services the interrupt then continues operation at the instruction that follows the WAIT command.

4.5 Using the Low-Power Modes

To enter either stop mode, in the system options register (SOPT), the stop-mode-enable bit (STOPE) must be set to a logic 1. This register is a write-once after any reset, so ensure to configure the other options in the same write. If the STOPE bit is clear and an attempt to execute a STOP instruction is made, the instruction is treated as an illegal opcode and a reset is forced.

In the system-power-management-status-and-control-2 register (SPMSC2), the partial-power-down control (PPDC) determines which stop mode is entered when a STOP instruction executes.

In addition, to be able to use stop2 mode, the LVDSE bit in SPMSC1 must be cleared. If this bit is not cleared, only stop3 can be entered.

Table 3 summarizes the source of exit and condition upon exit for each of the stop modes.

Table 3. Stop-Mode Selection and Source-of-Exit

Mode	PPDC (in SPMC2)	Source of Exit	Condition Upon Exit
Stop2	1	IRQ, reset, or RTC	POR (PPDF bit set in SPMSCR)
Stop3	0	IRQ, reset, RTC, pin interrupts, ADC, MSCAN, SCI	If reset is used, then POR; else, normal operation continues from the interrupt vector

4.6 Using Stop2

When the PPDC bit a logic 1, stop2 is entered upon execution of the STOP instruction. Stop2 results in lower current consumption than stop3. The RAM is kept powered on to maintain its values, and the I/O pins are latched in their current state.

Considerations to ensure proper operation when using stop2:

- The IRQ pin must be enabled or pulled up externally.
- The LVD must be disabled in stop (LVDSE = 0).
- If using the RTC, only the LPO clock source functions in stop2.
- The EREFSTEN bit has no effect in stop2. This clock reference is always powered down.
- Only the RAM remains powered; all other I/O registers are reset upon wakeup.
- The PPDF flag must always be cleared before the I/O pins can be modified from their stop2 entry state.

The IRQ pin must be enabled by writing to the IRQ pin-enable bit (IRQPE) in the IRQ status-and-control (IRQSC) register. Failure to do this results in the MCU waking from stop2 immediately after entering stop

unless an external pullup is placed on the IRQ pin. The IRQ interrupt does not need to be enabled (IRQIE bit in IRQSC).

The $\overline{\text{RESET}}$ pin automatically configures as a wakeup pin for stop2. No software or external pullups are necessary.

If the LVD module is enabled in stop mode, stop2 cannot be used. Attempting to enter stop2 with the LVD enabled in stop results in the MCU entering stop3 mode instead.

When using the RTC module in stop2 as a wakeup source, the LPO clock source must be used because the external clock source does not remain powered in stop2.

If the external oscillator is enabled in stop mode (the EREFSTEN bit in ICG control register 1) and stop2 is entered, this bit is ignored and the clocks power down.

The stop instruction must be enabled in the SOPT register, and the PPDC bit in the SPMSC2 register must be set to a logic 1.

The peripherals not already mentioned do not require any special handling because they automatically power down upon entry to stop2.

Stop2 is best suited for situations where the lowest possible power consumption is required, but RAM contents and I/O states must be maintained. Because the RTC module can run in stop2, the MCU can also wake up without external input.

Upon waking from stop2, the MCU starts up as if a POR occurred. However, PPDF in the SPMSC2 register can be used to indicate the MCU woke up from stop2 instead of a standard POR.

By using PPDF and PPDACK, the user code can save any desired register values into RAM before entering stop2 and restore these values after waking up. If the port registers are saved and restored before the PPDACK is written to a logic 1, then the I/O states are preserved. Any port pin not reconfigured to its latched stop2 state reverts to its reset state. Also, any peripheral not reconfigured to its pre-stop2 state reverts to its reset state.

Typical code execution sequence for stop2 entry and exit:

```
; Constant declarations


IRQSCinit:      equ     $10                     ; enable the IRQ pin
SOPT1init:      equ     $A0                     ; enable COP and STOP
RTCSCinit:      equ     $1F                     ; enable int and select 1 sec timeout
SPMSC2init:     equ     $01                     ; PPDC set => stop2
SPMSC2st2:      equ     $07                     ; PPDACK, PPDC set
PPDFmask:       equ     $08                     ; mask for PPDF bit in SPMSC2 reg
…
1) System initialization after reset
Start:          lda     SPMSC2                  ; Check if coming from stop2
                and     #PPDFmask
                bne     Stop2rec                ; If so, branch to recovery code
                lda     #SOPT1init              ; Else, treat as normal POR
                sta     SOPT1                   ; init the System Options
                lda     #SPMSC2init
```

```
                sta     SPMSC2                  ; init the SPMSC2 reg
                mov     #IRQSCinit,IRQSC        ; init the IRQ pin
…
2) Entering stop with RTC enabled
                jsr     SaveRegs
                lda     #RTCSCinit              ; Enable RTC module
                sta     SRTISC
                stop
…
3) After RTC times out, a POR executes and code restarts at reset vector but this time PPDF
are set
Start:          lda     SPMSC2
                and     #PPDFmask
                bne     Stop2rec
…
Stop2rec:       jsr     LoadRegs
                lda     #SPMSC2st2
                sta     SPMSC2
                bra     Main
…
; Begin Main code execution
Main:
```

The constant SPMSC2st2 also sets the PPDC bit to logic 1. This is because this bit is write-once. Failure to set this bit to 1 in this write results in the next STOP instruction entering stop3 mode instead of stop2. You can enable stop3 instead of stop2 at this point.

The POR results in the system bus clock being driven by an internal 8 MHz clock, and stop recovery occurs fairly quickly, allowing for rapid code execution to restore registers.

4.7 Using Stop3

Stop3 mode does not lead to the lowest possible IDDs but is versatile and the least intrusive of all the stop modes. Stop3 is entered as long as the PPDC bit in SPMSC2 is 0. Also, if LVD is enabled in stop or entry into background debug mode is enabled (ENBDM bit in BDCSCR is set), the only stop3 can be entered. When the ENBDM bit is set and a stop instruction is executed, the system clocks to the background debug logic remain active so background debug communication remains possible.

Stop3 must be used when you depend on an easy exit from stop mode. Stop recovery time is typically around 100 µs when using the internal clock or the PLL. For applications using the PLL to boost a reference frequency, stop3 has the advantage of preserving previous DCO settings when recovering from stop3 with an interrupt. This means that upon stop recovery, the DCO is set up with the system clock configuration predefined.

Unlike stop2 mode, if stop3 is exited with an interrupt, there is no need for any initialization or reconfiguration. When the interrupt occurs, the CPU begins processing with the stacking operations leading to the interrupt-service routine. Upon the RTI command of the interrupt service routine, the CPU resumes at the instruction immediately following the stop command.

5 Demonstration Software

```
Freescale HC08-Assembler
(c) Copyright Freescale 1987-2006

 Abs. Rel.   Loc    Obj. code   Source line
 ---- ----   ------ ---------   -----------
    1    1                      ;*********************************************************************
    2    2                      ;*      Copyright (c) Freescale 2006
    3    3                      ;*********************************************************************
    4    4                      ;*File name:       Automotive Low Power Modes.mcp Current Release Level:  1.0
    5    5                      ;*Last Edit Date:  14-Dec-06         Classification:           ES
    6    6                      ;*
    7    7                      ;*Include Files:   MC9S08DZ60.inc  MC9S08DZ60 MCU definitions
    8    8                      ;*Assembler:       CodeWarrior for HC08 V5.1
    9    9                      ;*Target Device:   MC9S08DZ60
   10   10                      ;*Documentation:   Automotive S08 Low Power Modes AN3387
   11   11                      ;*********************************************************************
   12   12                      ;* Author:             Donnie Garcia
   13   13                      ;* First Release:      06-May-03
   14   14                      ;*
   15   15                      ;* Update History:
   16   16                      ;*
   17   17                      ;* Rev     Date         Author  Description of Change
   18   18                      ;* ------  ---------    ------  ------------------------------------
   19   19                    ;*   1.0   Dec 4, 2006  MWR     Conversion of AN2493 SW to DZ60 Demo Board
   20   20                      ;*
   21   21                      ;*********************************************************************
   22   22                      ;*********************************************************************
   23   23                      ;* This code is used along with EVB9S08DZ60 board to demonstrate
   24   24                      ;* Stop Modes
   25   25                      ;* For Measurement purposes all headers/jumpers (Except the Power_Sel
   26   26                      ;* jumper) were removed from the demo board
   27   27                      ;* PTC0 LED on the demo board flashes at the interrupt rate
   28   28                      ;* When using Stop2 in order to re-establish BDM connection
   29   29                      ;* PTG0/BKGD must be held low on power up, then released.
   30   30                      ;*********************************************************************
   31   31                      ;*       StopSelect and WakeSelect are used to configure the code
   32   32                     ;*
   33   33                      ;* To test Stop2 RTC
   34   34                      ;*  StopSelect = %00000010 WakeSelect = %00000010
   35   35                      ;*
   36   36                    ;* To test Stop3

   37   37                      ;*  StopSelect = %00000100 WakeSelect = %00000001
   38   38                ;*

   39   39                    ;* To test Stop3 RTC Internal

   40   40                      ;*  StopSelect = %00000100 WakeSelect = %00000010
   41   41                ;*

   42   42                    ;* To test Stop3 RTC External

   43   43                      ;*  StopSelect = %00000100 WakeSelect = %00000100
   44   44                      ;*********************************************************************
   45   45                       include "derivative.inc"
 4339 4289i                             ENDIF
 4375   46
 4376   47                       ; export symbols
 4377   48                      ;
 4378   49                                 XDEF _Startup
 4379   50                                 ABSENTRY _Startup
 4380   51
 4381   52                      ;*********************************************************************
 4382   53                      ;SELECT STOP MODE AND WAKE UP SOURCE HERE
 4383   54                      ;
 4384   55          0000 0004   StopSelect:  equ     %00000100 ;Select Stop Mode Here
 4385   56                      ;                          ||
```

```
4386   57                     ;                          ||
4387   58                     ;                          |+--- Stop2 Mode selected
4388   59                     ;                          +----- Stop3 Mode Selected
4389   60                     ;If more than 1 mode is selected the lowest stop mode is set
4390   61                     ;If no selection is made stop3 is chosen
4391   62
4392   63          0000 0002  WakeSelect:  equ     %00000010 ;Select Method of wake up (Stop2,3)
4393   64                     ;                          |||
4394   65                     ;                          ||+-- Pin Interrupt wake selected (For Stop3)
4395   66                     ;                          ||+-- RTC Internal wake selected (LPO)
4396   67                     ;                          +----- RTC external wake Selected (For Stop3)
4397   68                     ;If No selection is made Pin Interrupt is selected
4398   69                     ;*****************************************************************
4399   70
4400   71
4401   72                     ;IMPORTANT REGISTER INITS
4402   73                     ;
4403   74          0000 0014  initSPMSC1:  equ     %00010100 ;Disable LVD in stop
```

```
4404   75                      ;                    ||||||
4405   76                      ;                    |||||+---LVDE ---- Enable LVD
4406   77                      ;                    ||||+----LVDSE --- Disable LVD in stop
4407   78                      ;                    |||+-----LVDRE --- Enable LVD reset protection
4408   79                      ;                    ||+------LVWIE
4409   80                      ;                    |+-------LVWACK
4410   81                      ;                    +--------LVWF
4411   82
4412   83
4413   84          0000 0000   initSPMSC2:  equ     %00000000 ;This register sets stop mode
4414   85                      ;                       |||| |
4415   86                      ;                       |||| +-PPDC
4416   87                      ;                       |||+---PPDACK
4417   88                      ;                       ||+----PPDF
4418   89                      ;                       |+-----LVWV
4419   90                      ;                       +------LVDV
4420   91                      ;
4421   92
4422   93          0000 0023   initSOPT1:   equ     %00100011 ;COP and STOP enable controls
4423   94                      ;                        |||   ||
4424   95                      ;                        |||   |+-RSTPE --- Reset pin enabled
4425   96                      ;                        |||   +--BKGDPE -- BKGD pin enabled
4426   97                      ;                        ||+------STOPE --- STOP allowed
4427   98                      ;                        |+-------COPT0 --- COP disabled
4428   99                      ;                        +--------COPT1 /
4429  100
4430  101
4431  102          0000 0004   initMCGC1:   equ     %00000100 ;Clock Generator Control 1
4432  103                      ;                        ||||||||
4433  104                      ;                        |||||||+-IRFSTEN - int ref enabled in STOP
4434  105                      ;                        ||||||+--IRCLKEN - enable int ref clock
4435  106                      ;                        |||||+---IREFS --- osc on in stop mode
4436  107                      ;                        ||||+----RDIV0 \
4437  108                      ;                        |||+-----RDIV1 --- reference divider
4438  109                      ;                        ||+------RDIV2 /
4439  110                      ;                        |+-------CLKS0 --- clock source select
4440  111                      ;                        +--------CLKS1 /
4441  112
4442  113          0000 0037   initMCGC2:   equ     %00110111 ;Clock Generator Control 2
4443  114                      ;                        ||||||||
4444  115                      ;                        |||||||+-ERFSTEN - ext ref enable in STOP
4445  116                      ;                        ||||||+--ERCLKEN - ext ref enable
4446  117                      ;                        |||||+---EREFS --- ext osc select
4447  118                      ;                        ||||+----LP ------ low power select
4448  119                      ;                        |||+-----HGO ----- high gain osc
4449  120                      ;                        ||+------RANGE --- frequency range of ext osc
4450  121                      ;                        |+-------BDIV0 --- bus freq divider
4451  122                      ;                        +--------BDIV1 /
4452  123
4453  124
4454  125          0000 0000   LED          equ     0
4455  126                      ;*****************************************************************
4456  127                                  org     Z_RAMStart
4457  128
4458  129  a000080             StopSet:      RMB     1         ;Used to select stop mode
4459  130  a000081             WakeSet:      RMB     1         ;Used to select Wait Mode
4460  131  a000082             PTCD_STORE:   RMB     1         ;Used to store PTFC
4461  132
4462  133
4463  134                                  org     ROMStart
4464  135
4465  136                      _Startup:
4466  137
4467  138  a001900 A6 23                   lda     #initSOPT1
4468  139  a001902 C7 1802                 sta     SOPT1      ;Disable COP and enable STOP
4469  140
4470  141  a001905 A6 14                   lda     #initSPMSC1
4471  142  a001907 C7 1809                 sta     SPMSC1    ;Disable LVD in stop
4472  143
4473  144  a00190A C6 180A                 lda     SPMSC2    ;how did we get here?
4474  145  a00190D A4 08                   and     #mSPMSC2_PPDF    ;was it a wake-up from STOP2?
```

```
4475  146  a00190F 26 12                 bne     Stop2Recovery ;If = 0 was normal reset
4476  147
4477  148                    ;This begins the path of a normal reset (Not stop2 recovery)
4478  149
4479  150                    INIT:
4480  151
4481  152                    ;*********************************************************************
4482  153                    ;FIRST setup SPMSC2 to to the proper stop mode
4483  154  a001911 A6 04                 lda     #StopSelect
4484  155  a001913 B7 80                 sta     StopSet
4485  156
4486  157  a001915 02 80 02              brset   1,StopSet,Set_Stop2
4487  158
4488  159                    Set_Stop3:
4489  160  a001918 20 09                 bra     StopSelectDone ;Reset state of SPMSC2 selects stop3
4490  161
4491  162                    Set_Stop2:
4492  163  a00191A B6 00                 lda     initSPMSC2 ;enable stop2
4493  164  a00191C AA 01                 ora     #mSPMSC2_PPDC
4494  165  a00191E C7 180A               sta     SPMSC2
4495  166
4496  167
4497  168  a001921 20 00                 bra     StopSelectDone
4498  169
4499  170
4500  171                    StopSelectDone:
4501  172
4502  173                    ;*********************************************************************
4503  174                    Stop2Recovery:              ;Initialize before PDACK
4504  175                    ;*********************************************************************
4505  176                    ;Now set up the selected wakeup source
4506  177  a001923 A6 02                 lda     #WakeSelect
4507  178  a001925 B7 81                 sta     WakeSet
4508  179  a001927 00 81 08              brset   0,WakeSet,InitPinI
4509  180  a00192A 02 81 26              brset   1,WakeSet,InitRTCint
4510  181  a00192D 04 81 29              brset   2,WakeSet,InitRTCext
4511  182  a001930 20 31                 bra     WakeSelectDone
4512  183
4513  184                    InitPinI:
4514  185
4515  186  a001932 A6 00                 lda     #0
4516  187  a001934 C7 1846               sta     PTAES          ;Select low level for interrupt
4517  188
4518  189  a001937 C6 1840               lda     PTAPE
4519  190  a00193A AA 10                 ora     #mPTAPE_PTAPE4  ;Enable PTA4 pullup
4520  191  a00193C C7 1840               sta     PTAPE
4521  192
4522  193  a00193F C6 1845               lda     PTAPS
4523  194  a001942 AA 10                 ora     #mPTAPS_PTAPS4  ;Enable PTA4 interrupt
4524  195  a001944 C7 1845               sta     PTAPS
4525  196
4526  197  a001947 B6 04                 lda     mPTASC_PTAACK  ;Clear Pending Pin Interrupts
4527  198  a001949 C7 1844               sta     PTASC
4528  199
4529  200  a00194C BA 02                 ora     mPTASC_PTAIE   ;Enable PortA Pin Interrupts
4530  201  a00194E C7 1844               sta     PTASC
4531  202
4532  203  a001951 20 10                 bra     WakeSelectDone
4533  204                    InitRTCint
4534  205  a001953 A6 17                 lda     #$17      ;Enable LPO RTC Interrupts 1s timeout
4535  206  a001955 B7 6C                 sta     RTCSC
4536  207  a001957 20 0A                 bra     WakeSelectDone
4537  208                    InitRTCext
4538  209  a001959 A6 3F                 lda     #$3F      ;External clock bit set
4539  210                                          ;Enable RTC Interrupts long timeout
4540  211  a00195B B7 6C                 sta     RTCSC
4541  212  a00195D 6E 37 49              mov     #initMCGC2,MCGC2 ;sets up external clock option
4542  213  a001960 6E 04 48              mov     #initMCGC1,MCGC1 ;8MHz ext xtal on demo board
4543  214                    WakeSelectDone
4544  215
4545  216                    ;*********************************************************************
```

```
4546  217                     ;Initialize all I/O to achieve Low Power
4547  218                     Init_IO
4548  219
4549  220  a001963 6E 10 1C                mov     #mIRQSC_IRQPE,IRQSC ;pull-up and enable IRQ
4550  221                     ;Make All unused I/O Outputs Driving low
4551  222  a001966 6E EF 01                mov     #$EF,PTADD ;ADDR
4552  223  a001969 6E FF 03                mov     #$ff,PTBDD ;BDDR
4553  224  a00196C 6E FF 05                mov     #$ff,PTCDD ;CDDR
4554  225  a00196F 6E FF 07                mov     #$ff,PTDDD ;DDDR
4555  226  a001972 6E FF 09                mov     #$ff,PTEDD ;EDDR
4556  227  a001975 6E FF 0B                mov     #$ff,PTFDD ;FDDR
4557  228  a001978 6E FF 0D                mov     #$ff,PTGDD ;GDDR
4558  229
4559  230  a00197B 6E 00 00                mov     #$00,PTAD ;ADR
4560  231  a00197E 6E 00 02                mov     #$00,PTBD ;BDR
4561  232  a001981 6E 00 04                mov     #$00,PTCD ;CDR
4562  233  a001984 6E 00 06                mov     #$00,PTDD ;DDR
4563  234  a001987 6E 00 08                mov     #$00,PTED ;EDR
4564  235  a00198A 6E 01 0A                mov     #$01,PTFD ;FDR
4565  236  a00198D 6E 00 0C                mov     #$00,PTGD ;GDR
4566  237
4567  238  a001990 C6 180A                 lda     SPMSC2    ;how did we get here?
4568  239  a001993 A4 08                   and     #mSPMSC2_PPDF    ;was it a wake-up from STOP2?
4569  240  a001995 27 13                   beq     MainLoop
4570  241
4571  242  a001997 4E 82 04                mov     PTCD_STORE,PTCD ;Replace PTC with stored info
4572  243  a00199A C6 180A                 lda     SPMSC2    ;acknowledge Stop2 recoverey
4573  244  a00199D AA 05                   ora     #(mSPMSC2_PPDACK|mSPMSC2_PPDC)
4574  245  a00199F C7 180A                 sta     SPMSC2
4575  246
4576  247  a0019A2 B6 04                   lda     PTCD      ;Toggle LED here for Stop2
4577  248  a0019A4 A8 01                   eor     #mPTCD_PTCD0
4578  249  a0019A6 B7 04                   sta     PTCD
4579  250  a0019A8 2E FE                   bil     *         ;Wait while IRQ is low (Debounce)
4580  251
4581  252                     MainLoop
4582  253  a0019AA B6 04                   lda     PTCD
4583  254  a0019AC B7 82                   sta     PTCD_STORE ;Store PTF state into RAM
4584  255  a0019AE 8E                      stop
4585  256  a0019AF 20 F9                   bra     MainLoop
4586  257
4587  258                     ;***********Interrupt Service Routines*****************************
4588  259                     kbi_isr
4589  260                     ;             bset    KBACK,KBISC
4590  261  a0019B1 C6 1844                 lda     PTASC     ;Acknowledge Pin Interrupt
4591  262  a0019B4 AA 04                   ora     #mPTASC_PTAACK
4592  263  a0019B6 C7 1844                 sta     PTASC
4593  264
4594  265  a0019B9 B6 04                   lda     PTCD      ;Toggle LED Here
4595  266  a0019BB A8 01                   eor     #mPTCD_PTCD0
4596  267  a0019BD B7 04                   sta     PTCD
4597  268  a0019BF 80                      rti
4598  269                     rti_isr
4599  270  a0019C0 B6 6C                   lda     RTCSC
4600  271  a0019C2 AA 80                   ora     #mRTCSC_RTIF
4601  272  a0019C4 B7 6C                   sta     RTCSC    ;Acknowledge RTC Interrupt
4602  273  a0019C6 B6 04                   lda     PTCD      ;Toggle LED Here
4603  274  a0019C8 A8 01                   eor     #mPTCD_PTCD0
4604  275  a0019CA B7 04                   sta     PTCD
4605  276  a0019CC 80                      rti
4606  277
4607  278                     ;***********Vectors*************************************************
4608  279                                 org     Vrtc
4609  280  a00FFCC 19C0                   fdb     rti_isr
4610  281                                 org     Vport
4611  282  a00FFD2 19B1                   fdb     kbi_isr
4612  283                                 org     Vreset
4613  284  a00FFFE 1900                   DC.W    _Startup          ; Reset
```

THIS PAGE IS INTENTIONALLY BLANK

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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)
support@freescale.com

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 Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3387
Rev. 0
10/2007

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant 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™
semiconductor