M16C/26

Using Brown Out Detection Vdet4

1.0 Abstract

The following article discusses the Brown-Out Detection circuit, Vdet4, of the M30262 MCU. A short sample program to evaluate Vdet4 behavior and verification is provided that will run on the MSV30262 board for. A variable power supply, connected to the board, is required to make the MCU Vcc voltage adjustable for evaluation/verification.

2.0 Introduction

The Renesas M30262 is a 16-bit MCU based on the M16C/60 series CPU core. The MCU features include up to 64K bytes of Flash ROM, 2K bytes of RAM, and 4K bytes of Virtual EEPROM. The peripheral set includes 10-bit A/D, UARTs, Timers, DMA, GPIO, and a voltage detection circuit. The voltage detection circuit has monitoring circuits to check the input voltage of the Vcc pin. These circuits monitor the input voltage at Vdet3 and Vdet4 (see Figure 1). VC26 to VC27 of VCR2 register is used to enable/disable these monitoring circuits (see Figure 2).

This application note discusses Vdet4 monitoring circuit that can be set to detect whether the input voltage is equal to and greater or less than Vdet4 depending on the value of VC13 bit of the VCR1 register. In addition, the Vdet4 monitoring circuit, if enabled, can generate an interrupt.

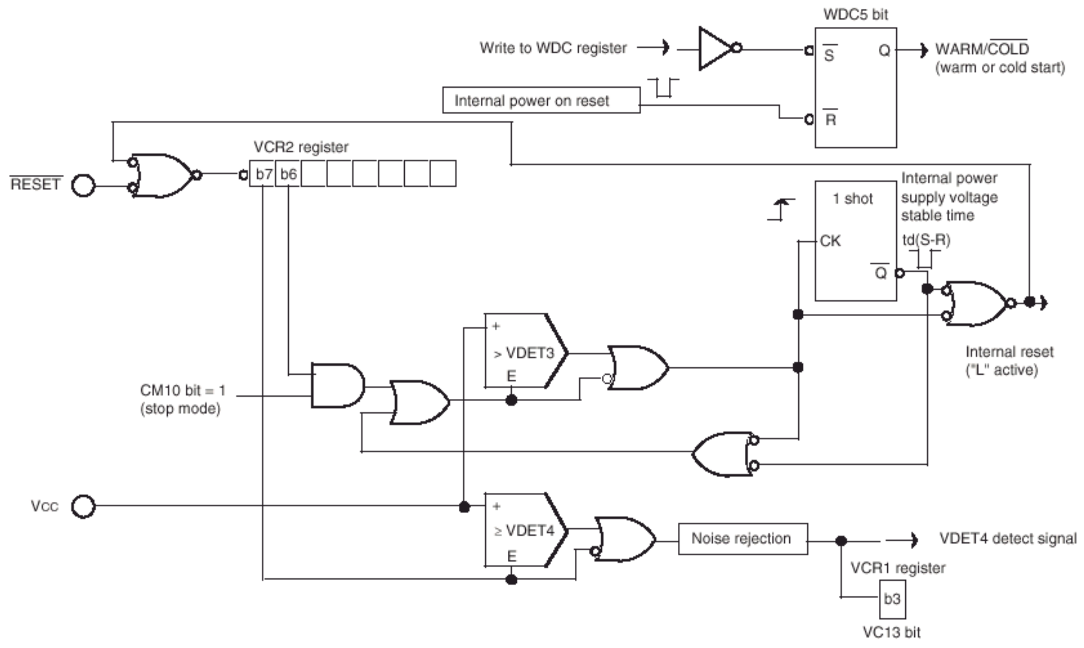
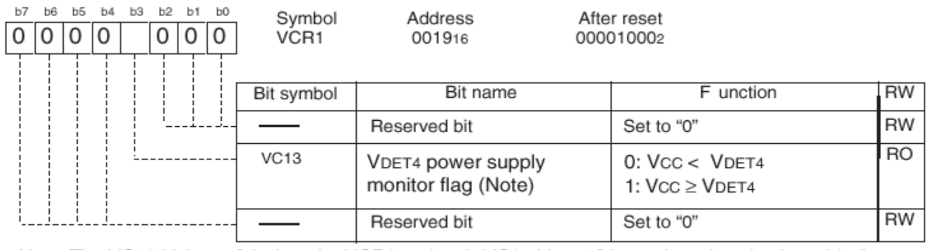


Figure 1 Vdet4 detection interrupt generation block

Power supply detection register 1

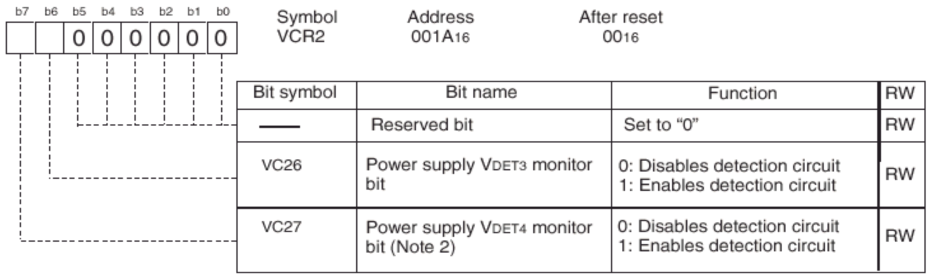


Symbol	Address	After reset
VCR1	0019_{16}	00001000_2

Bit symbol	Bit name	Function	RW
——	Reserved bit	Set to “0”	RW
VC13	VDET4 power supply monitor flag (Note)	0: $V_{CC} < V_{DET4}$ 1: $V_{CC} \geq V_{DET4}$	RO
——	Reserved bit	Set to “0”	RW

Note: The VC13 bit is useful when the VCR2 register's VC27 bit = 1 (VDET4 detection circuit enabled).
The VC13 bit is always 1 ($V_{CC} \geq V_{DET4}$) when the VCR2 register's VC27 bit = 0 (VDET4 detection circuit disabled).

Power supply detection register 2 (Note 1)

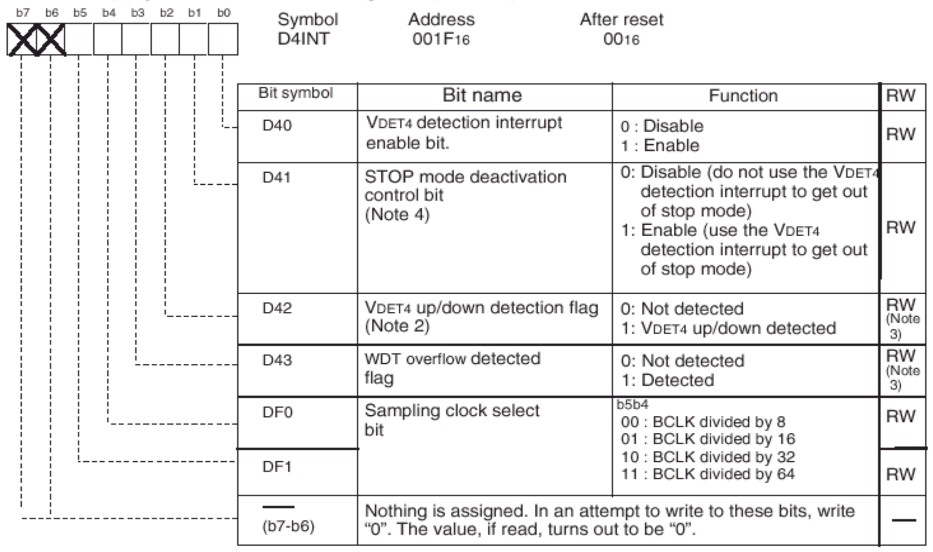


Symbol	Address	After reset
VCR2	$001A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
——	Reserved bit	Set to “0”	RW
VC26	Power supply VDET3 monitor bit	0: Disables detection circuit 1: Enables detection circuit	RW
VC27	Power supply VDET4 monitor bit (Note 2)	0: Disables detection circuit 1: Enables detection circuit	RW

Note 1: Write to this register after the PRCR register's PRC3 bit is set to “1” (write enabled).
Note 2: To use the VCR1 register's VC13 bit or D4INT register's D42 bit, set the VC27 bit to “1” (VDET4 detection circuit enabled).

Power supply VDET4 detection register (Note 1)



Symbol	Address	After reset
D4INT	$001F_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
D40	VDET4 detection interrupt enable bit.	0 : Disable 1 : Enable	RW
D41	STOP mode deactivation control bit (Note 4)	0: Disable (do not use the VDET4 detection interrupt to get out of stop mode) 1: Enable (use the VDET4 detection interrupt to get out of stop mode)	RW
D42	VDET4 up/down detection flag (Note 2)	0: Not detected 1: VDET4 up/down detected	RW (Note 3)
D43	WDT overflow detected flag	0: Not detected 1: Detected	RW (Note 3)
DF0	Sampling clock select bit	b5b4 00 : BCLK divided by 8 01 : BCLK divided by 16 10 : BCLK divided by 32 11 : BCLK divided by 64	RW
DF1			RW
—— (b7-b6)	Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—

Note 1: Write to this register after the PRCR register's PRC3 bit is set to “1” (write enabled).
Note 2: Useful when the VCR2 register's VC27 bit = 1 (VDET4 detection circuit enabled). If the VC27 bit is cleared to 0 (VDET4 detection circuit disabled), the D42 bit is set to 0 (Not detected).
Note 3: This bit is cleared to “0” by writing a “0” in a program. (Writing a “1” has no effect.)
Note 4: If the VDET4 detection interrupt needs to be used to get out of stop mode again after once used for that purpose, reset the D41 bit by writing a 0 and then a 1.

Figure 2 VCR1, VCR2 and D4INT registers

3.0 Vdet4 voltage detection circuit

A Vdet4 detection interrupt request is generated when the input voltage of the Vcc pin rises over or drops under Vdet4 while the D40 bit of the D4INT register is set (D40 = 1). Setting D41 (D41 = 1) enables the Vdet4 detection interrupt. The Vdet4 detection interrupt shares the interrupt vector with the watchdog timer and oscillation stop detection interrupts.

The D42 bit of D4INT register is set to "1" when an input voltage over or under Vdet4 is detected. A Vdet4 detection interrupt is generated when the D42 bit changes state from "0" to "1". The D42 bit needs to be cleared to "0" in the program in order to generate an interrupt for the following detection.

The D41 interrupt enable bit needs to be disabled right after entering the interrupt service routine of Vdet4 and enabled once program is out of the service routine. This step prevents unwanted interrupts from being generated once the program is inside the Vdet4 interrupt service routine. Table 1 shows the conditions under which a Vdet4 detection interrupt request is generated. The Vdet4 detection interrupt generation block diagram is shown in Figure 3.

Table 1 Vdet4 detection interrupt request generation conditions

VDET4 up/down detected	D40 bit	Status of D42 bit	Status of D41 bit	VDET4 detection Interrupt request
Detected	1	0 (Change from 0 to 1)	—	Generated
		—	1(Note)	
Detected	1	1(No change)	0	Not generated
	0	—	—	
Not detected	—			

Note : D41 bit = 1 while in stop mode.

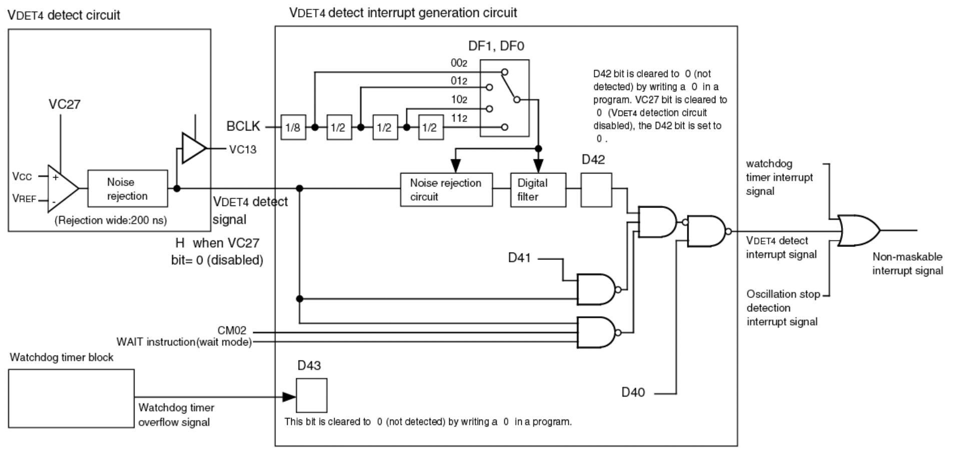

VDET4 detect circuit
VC27
Vcc
VREF
+
-
Noise rejection
(Rejection wide:200 ns)
VC13
BCLK
VDET4 detect signal
H when VC27 bit= 0 (disabled)
VDET4 detect interrupt generation circuit
DF1, DF0
00₂
01₂
10₂
11₂
1/8
1/2
1/2
1/2
D42 bit is cleared to 0 (not detected) by writing a 0 in a program. VC27 bit is cleared to 0 (VDET4 detection circuit disabled), the D42 bit is set to 0 .
Noise rejection circuit
Digital filter
D42
D41
CM02
WAIT instruction(wait mode)
Watchdog timer block
Watchdog timer overflow signal
D43
This bit is cleared to 0 (not detected) by writing a 0 in a program.
D40
watchdog timer interrupt signal
VDET4 detect interrupt signal
Oscillation stop detection interrupt signal
Non-maskable interrupt signal

Figure 3 Vdet4 detection interrupt generation block

4.0 Vdet4 Limitations in STOP and WAIT modes

The behavior of Vdet4 in STOP and WAIT modes is discussed below:

4.1 Vdet4 in STOP mode

Under the following condition (in STOP mode), if the input voltage goes over Vdet4 when the D41 bit is equal to "1", a Vdet4 detection interrupt is immediately generated regardless of the value of D42, which causes the MCU to exit STOP mode.

- VC27 bit of the VCR2 register equal to "1" (Vdet4 detection circuit enabled),
- VC13 bit of the VCR1 register equal to "1" (Vcc >= Vdet4),
- D40 bit of the D4INT register equal to "1" (Vdet4 detection interrupt enabled), and
- CM10 bit of the CM1 register is set to "1" (STOP mode)

Thus, in applications where Vdet4 interrupt is used to exit STOP mode, ensure that the CM10 bit is set when the VC13 is equal to 0 (Vcc < Vdet4).

4.2 Vdet4 in WAIT mode

The limitation of Vdet4 in WAIT mode is that, if a WAIT instruction is executed under the following condition, a Vdet4 interrupt is immediately generated, causing the MCU to exit from WAIT mode.

- VC27 bit of the VCR2 register is equal to "1" (Vdet4 detection circuit enabled),
- VC13 bit of the VCR1 register equal to "1" (Vcc >= Vdet4), and
- D40 bit of the D4INT register equal to "1" (Vdet4 detection interrupt enabled)

Thus, in applications where Vdet4 is used to exit WAIT mode, ensure that VC13 is equal to 0 (Vcc < Vdet4) before entering WAIT mode.

5.0 Implementation Software

5.1 Application code outline

The Vdet4 detection program, written in C and compiled using the KNC30 compiler, will run on the M16C/26 starter kit MSV30262 SKP board. To run the program, follow the steps below.

1. Download the program to the M30262 MCU using the FoUSB Programmer (with USB-ICD).
2. Disconnect USB-ICD from the SKP board. Using a variable voltage power supply, supply 5V to the SKP board using the board's Vcc and GND pins. At this point, the green power LED (D7) should light up.
3. After some time, two more LEDs (D3 and D5) should also light up. If they do not, press the reset switch (S1). The red LED (D3) will be blinking indicating that the MCU is running. The Vcc > Vdet4 condition is indicated with the green LED (D5) lit up.

4. Decrease the Vcc supply voltage gradually and at some point the green LED (D5) will turn OFF indicating that Vcc < Vdet4. The red LED (D3) will still be blinking at this stage indicating that the program is still running.
5. Increase the Vcc supply voltage gradually and at some point the green LED (D5) will light up again indicating that Vcc > Vdet4 once again.
6. To verify Vdet4 behavior in STOP mode, set D41 = ”1” (enable Vdet4 in STOP mode) by pressing switch S3. The yellow LED (D4) will light up momentarily indicating that S3 has been pressed.
7. Decrease the Vcc supply voltage gradually and once Vcc < Vdet4, the green LED (D5) turns OFF but the red LED (D3) keeps on blinking as in step 4 mentioned above.
8. To enter STOP mode, press switch S4. The red LED (D3) will stop blinking and of indicating that the MCU has entered the STOP mode.
9. Increase the Vcc supply voltage gradually and at a point where Vcc > Vdet4, both the red and the green LEDs (D3 and D5) turn ON indicating that the MCU has exited from STOP mode.
10. To verify Vdet4 behavior in STOP mode with a Vcc > Vdet4, press switch S4 to enter STOP mode while Vcc > Vdet4. The red and the green LEDs (D3 and D5) will only turn OFF momentarily indicating that STOP mode has been exited immediately. This occurs because STOP mode cannot be sustained with a Vcc > Vdet4.
11. The Vdet4 behavior in WAIT mode is similar to the Vdet4 behavior in STOP mode and can be verified by repeating steps similar to 6-10 by pressing switch S2 instead of switch 4 to enter WAIT mode.

6.0 Reference

Renesas Technology Corporation Semiconductor Home Page

http://www.renesas.com

E-mail Support

support_apl@renesas.com

7.0 Software source code

```
/******************************************************************************
*
*       File Name: main.c
*
*       Content:   This program demonstrates the usage and bahavior of the Vdet4
*                          voltage detection circuit. The normal execution of a program
*                          loop is simulated by the blinking of the red LED (D3). the
*                          green LED (D5) remains turned ON as long as the supply voltage
*                          Vcc is greater than Vdet4 and turns OFF when Vcc < Vdet4.
*                          The chip may be made to enter stop mode or wait mode by pressing
*                          switch sw2 or sw4 respectively. Switch sw3 is used to reset the
*                          stop mode deactivation control bit d41 of D4INT register.
*                          If the chip is made to enter stop or wait mode when Vcc>Vdet4,
*                           it immediately exits from stop or wait mode and the red and the
*                           green LEDs resume flashing. However, sustained stop or wait mode
*                           may be entered when Vcc<Vdet4 in which case stop or wait mode
*                           is exited when Vcc becomes greater than Vdet4 once again.
*                           Note that sw3 must be pressed (resetting bit d41) after exiting
*                           stop or wait mode each time in order to make the next exit
*                           possible.
*
*       Date:  1-06-2003
*       This program was written to run on the MDECE30262 Board for MSV30262-SKP.
*
*       Copyright 2003 Renesas Technology America, Inc.
*       All rights reserved
*
*==============================================================================
*       $Log:$
*=============================================================================*/

#include "sfr262.h"     // M16C/26 special function register definitions
#pragma INTERRUPT       Vdet4_ISR /* Interrupt Service Routine for Vdet4 */

/* LEDs */
#define red_led                 p7_0
#define yellow_led      p7_1
#define green_led       p7_2

/* SWITCHES */
#define sw2             p10_5
#define sw3             p10_6
#define sw4             p10_7

/*****************************************************************************/
void mcu_init(void);      /* routine that initializes the mcu */
void Vdet4_init(void);    /* routine that initializes Vdet4 */
void Vdet4_ISR(void);     /* Interrupt service routine for Vdet4 */
void enter_stop(void);    /* routine for entering stop mode */
void enter_wait(void);    /* routine for entering wait mode with peripheral clocks stopped */
void reset_d41(void);     /* routine that resets d41 bit for successive stop mode exits */
void blink_redLED(void); /* blink red LED to show continuation of program execution */


int delay_count;          /* a delay counter for blinking red LED*/
int delay_counter;        /* a delay counter for blinking yellow LED */
int i;                    /* a delay counter for stop mode switch */
int flag = 0;             /* Vdet4 ISR entry flag */
/******************************************************************************
Name:   main
Parameters: None
Returns: None
Description:  main program loop and initialization
*****************************************************************************/
main() {
        mcu_init();                     /* initialize MCU */

        Vdet4_init();                   /* initialize Vdet4 */
        asm( "fset i");                 /* enable interrupts */

        while(1){                       /* program loop */
                /*** blink red LED to show continuation of program execution ***/
                blink_redLED();

                if( vc13 == 0 ){                        /* check is Vcc < Vdet4 */
                        green_led     = 1;              /* turn OFF green LED */
                }
                else if( vc13 == 1){    /* check if Vcc > Vdet4 */
                        green_led     = 0;                      /* turn ON green LED */
                }
                /* invoke stop mode by pressing sw2 */
                if (sw2 == 0 && sw3 != 0 && sw4 != 0){ /* only sw2 is pressed */
                        enter_stop();           /* invoke stop mode */

                /** remain in delay loops to avoid sw2 switch bouncing effect **/
                for (delay_count = 0; delay_count<0xffff; delay_count++)
                        ;
                for (delay_count = 0; delay_count<0xffff; delay_count++)
                        ;
                }

                /* reset stop exit bit of D4INT register by pressing sw3 */
                if (sw3 == 0 && sw2 != 0 && sw4 != 0){ /* only sw3 is pressed */

                        reset_d41();            /* reset d41 stop exit bit of D4INT register */
                }

                /* invoke wait mode by pressing sw4*/
                if (sw4 == 0 && sw2 != 0 && sw3 != 0){ /* only sw4 is pressed */
                        enter_wait(); /* enter wait mode with peripheral clocks stopped */
                }
                /** remain in delay loops to avoid sw4 switch bouncing effect **/
                for (delay_count = 0; delay_count<0xffff; delay_count++)
                        ;
                for (delay_count = 0; delay_count<0xffff; delay_count++)
                        ;
                /* re-enable Vdet4 interrupt for next interrupt generation */
                if (flag == 1){/* Entered Vdet_ISR */
                        /* reset flag for detecting next Vdet4 interrupt */
                        flag = 0;
                        /* enable Vdet4 interrupt for next interrupt generation*/
                        prc3 = 1; /* unlock d4int register */
                        d40  = 1; /* enable Vdet4 interrupt */
                        d42  = 0; /* re-initialize up-down detection bit */
                        prc3 = 0; /* lock d4int register */
                }
        }
}
```

```
/*****************************************************************************
Name:   Vdet4_ISR
Parameters: None
Returns: None
Description: This is the service routine for Vdet4 Interrupt. This routine
             will be called whenever Vdet4 interrupt is generated.
*****************************************************************************/
void Vdet4_ISR(void){
        /* disable Vdet4 interrupt */
        prc3 = 1; /* unlock D4INT register */
        d40  = 0; /* disable Vdet4 interrupt to prevent successive unwanted*/
                  /* interrupts to be fired due to the comparator circuit */
        prc3 = 0; /* lock D4INT register */
        flag = 1; /* indicate entry to Vdet4 ISR */
}

/*****************************************************************************
Name:           mcu_init
Parameters:     None
Returns:        None
Description:  Initialization routine for the SKP board.
*****************************************************************************/
void mcu_init(void) {
   /* LED initialization */
        pd7_0 = 1;              /* set LED ports to outputs (connected to LEDs) */
        pd7_1 = 1;
        pd7_2 = 1;
        red_led = 1;            /* turn off all LEDs */
        green_led = 1;
        yellow_led = 1;
        return;
}
/*****************************************************************************
Name:           Vdet4_init
Parameters:     None
Returns:        None
Description:    Initialization routine for the Voltage Detection Circuit 4
*****************************************************************************/
void Vdet4_init(void) {
        prc3 = 1;                       /* unlock vcr2 and d4int registers */
        vc27 = 1;                       /* enable Vdet4 */
        d40  = 1;                       /* enable Vdet4 interrupt detection */
        d41  = 1;                       /* enable Vdet4 detection interrupt to get out of stop
mode */
        prc3 = 0;                       /* lock vcr2 and d4int */
        return;
}
```

```
/*****************************************************************************
Name:           enter_stop
Parameters:     None
Returns:        None
Description:    routine for entering the stop mode. This routine will be called
                when switch sw2 is pressed
*****************************************************************************/
void enter_stop(void){
        /* turn OFF green LED for easy visibility of stop mode being activated*/
        green_led      = 1;

        /* configure clock mode register to enter stop mode */
        prc0 = 1;                             /* unlock clock mode register */
        cm10 = 1;                             /* stop all clocks */
        prc1 = 0;                             /* lock clock mode register */
}
/*****************************************************************************
Name:           enter_wait
Parameters:     None
Returns:        None
Description:    routine for entering the wait mode with peripheral clocks stopped.
                This routine will be called when switch sw4 is pressed
*****************************************************************************/
void enter_wait(void){
        /* stop peripheral clocks */
        prc0 = 1;                     /* unlock CM0 register */
        cm02 = 1;                     /* stop peripheral clocks */
        prc0 = 0;                     /* lock CM0 register */
        /* turn OFF green LED */
        green_led = 1;

        _asm ( "wait" );              /* invoke wait mode */
        return;
}
/*****************************************************************************
Name:           reset_d41
Parameters:     None
Returns:        None
Description:    routine for resetting d41-bit of the D4INT register . This routine
                will be called when switch sw3 is pressed. The yellow LED lights up
                momentarily to indicate that sw3 has been pressed.
*****************************************************************************/
void reset_d41(void){
        /* turn on yellow LED momentarily */
        yellow_led = 0;                       /* turn ON yellow LED */
        /* wait for a while to keep yellow LED turned ON */
        for (delay_counter=0; delay_counter<0xffff; delay_counter++)
                ;
        yellow_led = 1;                       /* turn off yellow LED */

        /* configure D4INT register */
        prc3 = 1;                             /* unlock D4INT register */
        /* reset d41 bit */
        d41  = 0;
        d41  = 1;
        prc3 = 0;                              /* lock D4INT register */
        return;
}
/*****************************************************************************
Name:           blink_redLED
Parameters:     None
Returns:        None
Description:    routine for blinking red LED. This routine is called from the
                main program loop.
*****************************************************************************/
void blink_redLED(void) {
        /* turn ON red LED */
        red_led         = 0;
        /* wait for a while */
        for (delay_count = 0; delay_count<0xffff; delay_count++)
                ;
        /* turn OFF red LED */
        red_led         = 1;
        /* wait for a while */
        for (delay_count = 0; delay_count<0xffff; delay_count++)
                ;
        return;
}
 /*****************************************************************************
Name:           blink_redLED
Parameters:     None
Returns:        None
Description:    routine for blinking red LED. This routine is called from the
                main program loop.
*****************************************************************************/
void blink_redLED(void) {
        /* turn ON red LED */
        red_led         = 0;
        /* wait for a while */
        for (delay_count = 0; delay_count<0xffff; delay_count++)
;
        /* turn OFF red LED */
        red_led         = 1;
        /* wait for a while */
        for (delay_count = 0; delay_count<0xffff; delay_count++)
                ;
        return;
}
```

In order for this program to run properly, Vdet4 interrupt vector in the sect30.inc startup file needs to point to the Vdet4 service routine. This is shown below.

```
;          :
;          :
;=================================================================
; fixed vector section
;-----------------------------------------------------------------
        .org    0fffdch
        .glb    _Vdet4_ISR
UDI:
        .lword  dummy_int
OVER_FLOW:
        .lword  dummy_int
BRKI:
        .lword  dummy_int
ADDRESS_MATCH:
        .lword  dummy_int
SINGLE_STEP:
        .lword  dummy_int
WDT:
        .lword  _Vdet4_ISR
DBC:
        .lword  dummy_int
;          :
;          :
```