Keywords: power modes, transition, EMI, firmware update, power cycling, Teridian, energy meter

APPLICATION NOTE 5268

Firmware for Safe Power Mode Transitions

Dec 13, 2011

Abstract: This document describes two firmware improvements that were applied to the Demo Code version 4.8p for Maxim's Teridian™ 71M6521 family of energy meter ICs. These improvements make transitions between power modes on the 71M6521 more reliable and support proper correction of the real-time clock (RTC) after return to mission mode. This document isolates the changes and describes details so that the updates can be retrofitted into customer firmware, if needed.

Introduction

Trim Fuses

Trim fuses are a form of nonvolatile (NV) memory used in the production of Maxim's Teridian energy metering ICs to adjust analog and digital characteristics of the ICs. After the production stage, the contents of the trim fuses cannot be changed. Some examples are the fuses used to trim the V_{REF} voltage to within ±1mV of 1.195 VDC and the fuses used to trim the V_{BIAS} internal voltage to the target value.

During operation of the IC, the contents of the trim fuses are read and refreshed by hardware on the IC at regular intervals, and the values obtained by the trim fuse read are provided to both the hardware controlled by the fuse values and to the fuse register locations in I/O RAM (see **Figure 1**). In the 71M6521 family, a total of seven individual trim fuse values can be read by first writing any number from 1 to 7 to I/O RAM register 0x20FD (`TRIMSEL`) and then reading the corresponding fuse value from I/O RAM register 0x20FF (`TRIM`).

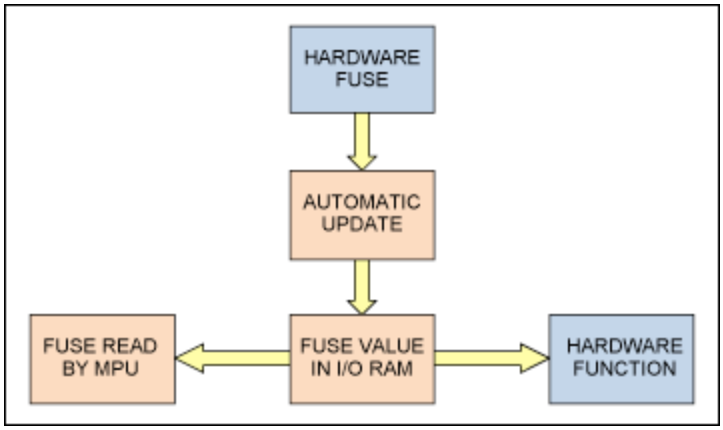

Figure 1. Trim fuse mechanism.

Misread of the Trim Fuses

The 71M6521 ICs support three low-power modes (sleep mode, LCD only mode, and brownout mode). In the low-power modes, regular operating currents are not available to the IC.

Extreme electromagnetic interference (EMI) or severe and repeated disruptions of the supply voltage can sometimes affect the trim read/refresh cycle, resulting in the IC misreading one of its trim fuses. If this occurs, a number of failures can result. The likelihood of potential failures is increased when the power supply rapidly "oscillates" the 71M6521 between mission and brownout modes, which can occur when the AC mains voltage is ramped up or down slowly in conjunction with a "soft" power supply. This oscillation causes the 71M6521 to switch rapidly and repeatedly between regular current consumption (mission mode) and low current consumption (brownout mode). Failures based on trim fuse misreads are extremely rare and can only be seen when large numbers of meters are power-cycled over days or weeks.

IC failures caused by a misread of the trim fuses can cause various symptoms, depending on which trim fuse read is affected. Symptoms include the following:

- Unexpected MPU software execution. Depending on the firmware, the unpredictable software execution can in some instances cause the internal clock to stop, preventing the watchdog timer from resetting the meter.
- Unexpected changes to internal power supply voltages and detection thresholds. This failure could cause the external power supply voltage applied to the V1 pin to be misread, causing an unexpected transition to sleep mode when the wake timer has not been set. In some instances, operating with an improperly initialized internal power supply for an extended period of time can corrupt the RTC.

Eliminating all EMI in a meter is not practical, but firmware can detect and avoid the above issues by testing the trim fuses of the IC when it enters brownout mode.

Another issue addressed by this new demo code version is faulty RTC drift correction. Older codes contained RTC code that integrated long term jitter into the RTC's drift correction after reset.

Application Cases for the Improved Code

The code described in this application note should be used for all versions of the 71M6521, including the 71M6521BE, 71M6521DE, and 71M6521FE. Maxim recommends including the firmware correction in all existing designs that contain batteries, and in all new designs.

Possible Failure Events at Power Mode Transition

Trim fuses are read by the fuse logic in the 71M6521 in a loop. Extensive repeated power mode transitions and severe EMI could occasionally affect the proper function of the trim fuse read. The failure mechanism is as follows:

1. The meter undergoes multiple power transitions or EMI events in rapid succession.
2. A trim fuse misread can occur on a mission-to-brownout mode transition in meters with a new lithium bat-tery. The fact that the battery voltage (typically 3.6 VDC) is different from the regular

operating voltage of the 71M6521 (typically 3.3 VDC) contributes to the failure.

3. Meters equipped with batteries tend to be affected to a larger degree. This is because a meter without a battery is off when no mains voltage is present and experiences a clean power-on-reset (POR) when the mains voltage returns.
4. After the power mode transition, the failure symptoms described above could occur. The exact form of a failure is unpredictable and intermittent, because it depends on the nature of the power disruption or EMI, the firmware of the meter, the battery selection, the power supply design, and on the exact trim fuse that is affected by the misread. Most meter designs exhibit no symptoms, even in extensive qualification.
5. When the meter returns to mission mode, higher operating currents are available, and the trim values self-correct from the fuse refresh. However, the meter state may be corrupted.

The RTC Failure Event

The RTC code adjusts the time to compensate for the drift while the meter was in low-power mode. However, the adjustment is not synchronized to the start of a second, and therefore accumulates jitter into the drift.

Details of Demo Code Revision 4.8p

Testing of the Code

A practical test is to attach new batteries to the meter of demo board, set the display defaults to display a clock value from the RTC, then cycle AC power every 10 seconds for eight meter-weeks. The meters can be tested by briefly stabilizing the power supply to nominal AC voltage. Successful code will resume operation and display a reasonable clock time. Code that passes eight meter-weeks is unlikely to fail in full qualification. New code should be checked at increasing intervals, starting with a half meter-hour.

Code Side Effects

Demo Code Revision 4.8p enables the PLL_OK interrupt very early, before running main(). This assures that a brownout mode transition during firmware initialization will be tested and handled. Other interrupts must be carefully managed to avoid creating spurious failures.

Demo Code Revision 4.8p also keeps a nonzero value in the wake timer at all times. In mission mode, the wake timer value does not count or cause an event unless the IC unexpectedly enters sleep or LCD only mode. In that case, the wake timer will trigger, waking the meter, and recover the IC from the error. This method may affect code that depends on using a zero reset value of the wake timer.

Code Location

The linker should be commanded to put the corrective code in the first 8KB of flash memory. In the Keil® IDE, the linker command can be changed by hovering the mouse over the project workspace, and by then selecting <right click>→options→BL_51 Locate Tab→Code Space. In the demo code, this command puts the segments as low in memory as possible:

?C_C51STARTUP,?PR?_?PLL_ISR?BATMODES_20,?PR?_?BATMODE_CHANGE?BATMODES_20,

<rest of the link commands for code space>

?C_C51STARTUP is the segment that contains the demo code's startup code.

?PR?_?PLL_ISR?BATMODES_20SAFE,?PR?_?BATMODE_CHANGE?BATMODES_20SAFE are the segments needed to run the demo code's PLL_OK interrupt routine.

PLL_OK Interrupt

The following code should be placed in the PLL_OK interrupt routine, which runs when brownout mode is detected. A tested copy of the code is in 4.8p in main\batmodes_20.c

```
    extern void trim_test(void) small reentrant;

    EA = 0;
    CONFIG0 = 0;        // Make sure we are running as fast as we can.
    IFLAGS = (~IE_PLLFALL_) & (~IE_PLLRISE_);  // force an edge to occur

    WAKE = 0x81;        // force a wake timer wake if sleep is forced

    trim_test();        // From start-up code.
```

Startup Code

The following startup code should be used in place of startup.a51. A tested copy of the code is available from Maxim (in the version 4.8p code build, the startup code can be found in util\startup_n_safe.a51). The code listing is provided on the following pages.

When this code first runs in the factory, it copies the trim registers to a location in flash memory. After that, it tests the trim registers. When the test fails, it forces a reread of the trim values by briefly putting the IC to sleep mode. After the wake occurs, the trim fuses are automatically read again. Calling the fuse test during reset assures that even in severe EMI situations, the trim fuses will be reread until they are correct.

```
;/*****************************************************************************
; * This code and information is provided "as is" without warranty of any   *
; * kind, either expressed or implied, including but not limited to the     *
; * implied warranties of merchantability and/or fitness for a particular   *
; * purpose.                                                                 *
; *                                                                          *
; * Copyright (C) 2011 Maxim Integrated Products, Inc. All Rights Reserved. *
; *****************************************************************************/
;//***************************************************************************
;//
;//  DESCRIPTION: 71M652x POWER METER - STARTUP Code.
;//
;//***************************************************************************
;//
;// File: STARTUP_N_SAFE.A51.
;//
$NOMOD51
;------------------------------------------------------------------------------
;  This file is part of the C51 Compiler package
;  Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.
;------------------------------------------------------------------------------
;  STARTUP.A51:  This code is executed after processor reset.
;
;  To translate this file use A51 with the following invocation:
;
```

```
;     A51 STARTUP_N_SAFE.A51
;
;  To link the modified STARTUP_N_SAFE.OBJ file to your application use the
following
;  BL51 invocation:
;
;     BL51 <your object file list>, STARTUP_N_SAFE.OBJ <controls>
;
;------------------------------------------------------------------------------
--
;
;  User-defined Power-On Initialization of Memory
;  With the following EQU statements the initialization of memory
;  at processor reset can be defined:
;
;               ; the absolute start-address of IDATA memory is always 0
IDATALEN        EQU     100H    ; the length of IDATA memory in bytes.
;
XDATASTART      EQU     0H      ; the absolute start-address of XDATA memory
XDATALEN        EQU     000H    ; the length of XDATA memory in bytes.
;
PDATASTART      EQU     0H      ; the absolute start-address of PDATA memory
PDATALEN        EQU     0H      ; the length of PDATA memory in bytes.
;
;  Notes:  The IDATA space overlaps physically the DATA and BIT areas of the
;          8051 CPU. At minimum the memory space occupied from the C51
;          run-time routines must be set to zero.
;------------------------------------------------------------------------------
--
;
;  Reentrant Stack Initialization
;
;  The following EQU statements define the stack pointer for reentrant
;  functions and initialized it:
;
;  Stack Space for reentrant functions in the SMALL model.
IBPSTACK        EQU     1       ; set to 1 if small reentrant is used.
IBPSTACKTOP     EQU     0FFH+1  ; set top of stack to highest location+1.
;
;  Stack Space for reentrant functions in the LARGE model.
XBPSTACK        EQU     0       ; set to 1 if large reentrant is used.
XBPSTACKTOP     EQU     07FFH+1; set top of stack to highest location+1.
;
;  Stack Space for reentrant functions in the COMPACT model.
PBPSTACK        EQU     0       ; set to 1 if compact reentrant is used.
PBPSTACKTOP     EQU     07FFH+1; set top of stack to highest location+1.
;
;------------------------------------------------------------------------------
--
;  Page Definition for Using the Compact Model with 64 KByte xdata RAM
;
;  The following EQU statements define the xdata page used for pdata
;  variables. The EQU PPAGE must conform with the PPAGE control used
;  in the linker invocation.
;
PPAGEENABLE     EQU     1       ; set to 1 if pdata object are used.
;
PPAGE           EQU     7       ; define PPAGE number.
                PUBLIC  PPAGE_SFR
PPAGE_SFR       DATA    0BFH    ; SFR that supplies uppermost address byte
;               (most 8051 variants use P2 as uppermost address byte)
;------------------------------------------------------------------------------
--

; Switch to M6520 when chip is available.

; Standard SFR Symbols
ACC     DATA    0E0H
B       DATA    0F0H
```

```
SP      DATA    81H
DPL     DATA    82H
DPH     DATA    83H
USER1   DATA    90H
DIR1    DATA    91H
FCTRL   DATA    0B2H
IPH     DATA    0B9H
IPL     DATA    0A9H
IRCON   DATA    0C0H
IEN0    DATA    0A8H
IEN1    DATA    0B8H

                NAME    ?C_STARTUP

?C_C51STARTUP   SEGMENT   CODE
?STACK          SEGMENT   IDATA

                RSEG    ?STACK
                DS      1

                EXTRN CODE (?C_START)
                PUBLIC  ?C_STARTUP
                PUBLIC  _?TRIM_TEST

                CSEG    AT      0
?C_STARTUP:     LJMP    STARTUP1
                RSEG    ?C_C51STARTUP
STARTUP1:
                CLR     IEN0^7          ; Disable interrupts
; To enable secure mode, remove the semicolon of the next line
;                ORL     FCTRL,#40H       ; set secure bit
                MOV     0E8h,#0FFh      ; Refresh nonmaskable watchdog.

; Set system interrupt priorities; more frequent are higher priority.
                MOV     IPH,#01DH       ; From code in options_gbl.h, main.c
                MOV     IPL,#00AH       ; From code in options_gbl.h
; Clear PLL_OK interrupt (bit3), and others
                MOV     IRCON,#0
; Enable PLL_OK interrupt
                MOV     DPTR,#2007H     ; Set EX_PLL
                MOV     A,#20H
                MOVX    @DPTR,A
; Enable interrupts
                MOV     IEN1,#08H       ; Enable INT4, the PLL_OK interrupt
                MOV     IEN0,#80H       ; Enable all interrupts

                ACALL   _?TRIM_TEST       ; Test 6521's trims

IF IDATALEN <> 0
                MOV     R0,#IDATALEN - 1
                CLR     A
IDATALOOP:      MOV     @R0,A
                DJNZ    R0,IDATALOOP
ENDIF

IF XDATALEN <> 0
                MOV     DPTR,#XDATASTART
                MOV     R7,#LOW (XDATALEN)
  IF (LOW (XDATALEN)) <> 0
                MOV     R6,#(HIGH (XDATALEN)) +1
  ELSE
                MOV     R6,#HIGH (XDATALEN)
  ENDIF
                CLR     A
XDATALOOP:      MOVX    @DPTR,A
                INC     DPTR
                DJNZ    R7,XDATALOOP
                DJNZ    R6,XDATALOOP
ENDIF
```

```
IF PPAGEENABLE <> 0
                MOV     PPAGE_SFR,#PPAGE
ENDIF

IF PDATALEN <> 0
                MOV     R0,#LOW (PDATASTART)
                MOV     R7,#LOW (PDATALEN)
                CLR     A
PDATALOOP:      MOVX    @R0,A
                INC     R0
                DJNZ    R7,PDATALOOP
ENDIF

IF IBPSTACK <> 0
EXTRN DATA (?C_IBP)
                MOV     ?C_IBP,#LOW IBPSTACKTOP
ENDIF

IF XBPSTACK <> 0
EXTRN DATA (?C_XBP)
                MOV     ?C_XBP,#HIGH XBPSTACKTOP
                MOV     ?C_XBP+1,#LOW XBPSTACKTOP
ENDIF

IF PBPSTACK <> 0
EXTRN DATA (?C_PBP)
                MOV     ?C_PBP,#LOW PBPSTACKTOP
ENDIF

                MOV     SP,#?STACK-1
                LJMP    ?C_START

; Read a trim whose index is in A. Return the trim in A.
                CSEG    AT 100H
FUSE_TABLE:
                DB      0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff

; Trim test. Prefix "_?" tells Keil C it's reentrant.
_?TRIM_TEST:
                PUSH    IEN0
                PUSH    PSW             ; Save registers, so it's reentrant.
                MOV     PSW,00H         ; Select registers before saving them
                PUSH    DPL
                PUSH    DPH
                PUSH    ACC
                MOV     A,R7            ; Save R7 from the current register
set.
                PUSH    ACC
                MOV     A,R6            ; Save R6 from the current register
set.
                PUSH    ACC

                CLR     IEN0.7

; Wait for all fuses to be read once before checking fuses
; Complete fuse read takes 45 xtal clock cycles (1 fuse per xtal clock)
                MOV     R7,#21          ; Wait 21x4 = 48 cycles
loop_inner:     DJNZ    R7,loop_inner   ; 4 cycles (in brownout, 1 cycle/clk)

                mov     DPH, #HIGH(FUSE_TABLE)

fuse_test_frst: ; check if values previously stored
                mov     DPL,#LOW(FUSE_TABLE)
                clr     a
                movc    a,@a+dptr       ; read fuse table
                xrl     a,#55H
                jnz     fuse_save
```

```
                MOV     DPTR,#2003H     ; Check if in brownout mode
                MOVX    A,@DPTR
                JB      ACC.6,PASS      ; Mission mode, so don't check.
fuse_test:
                mov     r7,#8           ; byte counter
fuse_test_lp1:
                mov     DPTR,#FUSE_TABLE    ; load pointer to fuse table
                mov     a,r7
                movc    a,@a+dptr       ; read fuse table
                mov     r6,a

                mov     a,r7

                MOV     DPTR,#20FDH     ; TRIMSEL address
                MOVX    @DPTR,A         ; load reg to read trim
                MOV     DPL,#0FFH       ; TRIM value address

                MOVX    A,@DPTR         ; Get trim value

                cjne    a,06H,FAIL      ; test if flash read and trim ==
                djnz    r7,fuse_test_lp1

                JMP     PASS
fuse_save:
                MOV     DPTR,#2003H     ; Check if in brownout mode
                MOVX    A,@DPTR
                ; If brownout mode, don't save them and enter sleep.
                JNB     ACC.6,FAIL

            ;   mov     84H,#LOW(FUSE_TABLE)
                mov     r7,#8           ; byte counter
fuse_save_lp1:
                mov     a,r7

                MOV     DPTR,#20FDH     ; TRIMSEL address
                MOVX    @DPTR,A         ; load reg to read trim
                MOV     DPL,#0FFH       ; TRIM value address

                MOVX    A,@DPTR         ; Get trim value
                mov     r6,a

                mov     a,r7
                mov     dptr,#FUSE_TABLE
                add     a,DPL
                mov     DPL,a
                mov     a,r6

                orl     0B2H,#01H       ; set to flash write
                movx    @dptr,a         ; write fuse table byte
                djnz    r7,fuse_save_lp1

                mov     a,#55H          ; control byte to mark

                mov     dptr,#FUSE_TABLE
                orl     0B2H,#01H       ; set to flash write
                movx    @dptr,a         ; write fuse table byte

                jmp     PASS

FAIL:
                MOV     DPTR,#20A9H     ; WAKE address
                MOV     A,#0C1H         ; SLEEP forces wake and fuse re-read
                MOVX    @DPTR,A

PASS:           CLR     A               ; Select 0
                MOV     DPTR,#20FDH     ; TRIMSEL address
                MOVX    @DPTR,A         ; clear trim select register.

                POP     ACC
```

```
                MOV     R6,A            ; Restore R7 in this register set
                POP     ACC
                MOV     R7,A            ; Restore R7 in this register set
                POP     ACC
                POP     DPH
                POP     DPL
                POP     PSW             ; Restore registers, so it's
reentrant.
                POP     IEN0
                RET

                END
```

RTC Code

At reset, the code should not adjust the clock to correct for the drift during sleep mode. The code that sets the clock after calculating the drift should be removed.

Related Parts		
71M6521BE	Energy Meter IC	Free Samples
71M6521DE	Energy Meter ICs	Free Samples
71M6521FE	Energy Meter ICs	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 5268: http://www.maximintegrated.com/an5268
APPLICATION NOTE 5268, AN5268, AN 5268, APP5268, Appnote5268, Appnote 5268