Mask Set Errata 1

PC9S12DP256 Microcontroller Unit

INTRODUCTION

This errata provides information applicable to the following MCU mask set devices:

- 0K36N mask of the PC9S12DP256

MCU DEVICE MASK SET IDENTIFICATION

The mask set is identified by a four-character code consisting of a letter, two numerical digits, and a letter, for example F74B. Slight variations to the mask set identification code may result in an optional numerical digit preceding the standard four-character code, for example 0F74B.

MCU DEVICE DATE CODES

Device markings indicate the week of manufacture and the mask set used. The data is coded as four numerical digits where the first two digits indicate the year and the last two digits indicate the work week. The date code “9115” would indicate the 15th week of the year 1991.

MCU DEVICE PART NUMBER PREFIXES

Some MCU samples and devices are marked with an SC, PC, ZC or XC prefix. An SC, PC or ZC prefix denotes special/custom device. An XC prefix denotes device is tested but is not fully characterized or qualified over the full range of normal manufacturing process variations. After full characterization and qualification, devices will be marked with the MC prefix.

ERRATA SYSTEM TRACKING NUMBERS

MUCTS00xxx is the tracking number for device errata. It can be used with the mask set and date code to identify a specific errata to a Motorola representative.

When contacting a Motorola representative for assistance, please have the MCU device mask set and date code information available.

MSE Published Date: March 8, 2001

ERRATA SUMMARY

Errata Number	Module affected	Brief description	Workaround available?	First Issued
MUCTS00152	FLASH	Security register (FSEC) is writable in special modes	No	Rev. 1
MUCTS00155	PWM8b8c	PWM interrupt enable gates the flag	No	Rev. 1(1)
MUCTS00163	ATD	ATD analog channel selection is flipped.	Yes	Rev. 1
MUCTS00164	BDM	BDM word reads in expanded narrow mode fail	Yes	Rev. 1
MUCTS00165	BDM	BDM word writes in expanded narrow mode fail	Yes	Rev. 1
MUCTS00168	PortA / PortB	Pull-down resistors instead of pull-ups	Yes	Rev. 1
MUCTS00169	Ports A, B, E and K	Pull-up resistors selected even when outputs.	Yes	Rev. 1
MUCTS00171	Device ID register	Not correctly programmed.	Yes	Rev. 1
MUCTS00172	IIC	IIC bit rates are inconsistent with existing IIC modules in two cases	Yes	Rev. 1
MUCTS00174	PLL	Lock bit setting and reset is not immediately reflected after register write	Yes	Rev. 1
MUCTS00176	Core	Security can be broken by using expanded modes	Yes	Rev. 1
MUCTS00177	BDM	BKGD pin works with reduced drive instead of full drive	Yes	Rev. 1
MUCTS00182	BDM	If BDM steals cycles during CALL or RTC instruction PPAGE register can get corrupted	Yes	Rev. 1
MUCTS00183	CRG	Switching to self-clock mode is not always possible if oscillation is stopped	No	Rev. 1
MUCTS00184	CRG	Self-clock mode is not entered at low frequency inputs (<20KHz)	No	Rev. 1
MUCTS00185	FLASH	Mass-erase of protected Flash block possible	No	Rev. 1
MUCTS00186	EEPROM	Mass-erase of protected EEPROM possible	No	Rev. 1
MUCTS00187	RTI	RTI does not generate interrupts if SYSWAI = 1	Yes	Rev. 1
MUCTS00188	PLL	PLL fails to power down in WAIT mode with PLLWAI = 1	Yes	Rev. 1 (1)
MUCTS00189	SPI	Changing the CPHA bit in MASTER mode before transmission is completed locks SPI	Yes	Rev. 1
MUCTS00190	ATD	ATD draws excess STOP current	No	Rev. 1
MUCTS00191	Ports H, J, P	Missing Interrupts in STOP mode	Yes	Rev. 2
MUCTS00202	SPI	SPI freeze with baud rate prescaler>=16 and CPHA=1	Yes	Rev. 2
MUCTS00204	SPI	SPIF Flag clearing problems	No	Rev. 2
MUCTS00209	BDM	BDM BACKGROUND command fails during CPU WAIT mode	No	Rev. 3
MUCTS00210	BDM	BDM loses synchronization when using PLL	Yes	Rev. 3
MUCTS00213	MEBI	E clock stretched high 1 cycle when writing IVIS	No	Rev. 4
MUCTS00214	IIC	IBIF/IBAL reset strategy not compatible with existing HC12	No	Rev. 4
MUCTS00239	ECT	Forced OC & successful OC occurring at the same time	No	Rev. 4
MUCTS00240	ECT	TOF flag getting generated at wrong place	No	Rev. 4

Errata Number	Module affected	Brief description	Workaround available?	First Issued
MUCts00242	CRG	CRG switches to Self-Clock Mode due to long Oscillator startup times	No	Rev. 4
MUCts00243	CRG	oscillator start-up problem	No	Rev. 4
MUCts00244	CRG	Oscillator input is noise sensitive	Yes	Rev. 4
MUCts00283	FLASH	Flash CCIF & CBEIF flag NOT SET during STOP	Yes	Rev. 5
MUCts00284	EEPROM	EEPROM CCIF & CBEIF flag NOT SET during STOP	Yes	Rev. 5
MUCts00285	FLASH	Flash 0 FAIL flag NOT SET on illegal command	No	Rev. 5
MUCts00287	EEPROM	Flash 0 FAIL flag NOT SET on illegal command	No	Rev. 5
MUCts00288	FLASH	Flash FCMD register is overwritten when CBEIF =0	Yes	Rev. 5
MUCts00289	EEPROM	EEPROM ECMD register is overwritten when CBEIF =0	Yes	Rev. 5
MUCts00293	SPI	No de-assertion of SS between data if CPHA=0	Yes	Rev. 5
MUCts00302	SPI	Improper SPTEF handshake in slave mode	Yes	Rev. 5
MUCts00309	ATD	ATD accuracy out of specification	No	Rev. 5
MUCts00310	ECT	Toggle on overflow error	No	Rev. 6
MUCts00318	msCAN	Double interrupts on status change	Yes	Rev. 6
MUCtS00324	MEBI	LSTRB/TAGLO pin driver contention	No	Rev. 7
MUCtS00325	MEBI/INT	Interrupt missed switching into STOP/WAIT	No	Rev. 7
MUCtS00326	PIM	Glitches not filtered correctly	No	Rev. 7
MUCtS00329	FLASH	FCLKDIV writable several times in NORMAL mode	No	Rev. 7
MUCtS00330	EEPROM	ECLKDIV writable several times in NORMAL mode	No	Rev. 7
MUCtS00331	BDLC	TEOD,IMSG,RX4XE,TSIFR and TMIFR[1:0] do not clear	Yes	Rev. 7
MUCtS00339	FLASH	Flash array read generates ACCERR during command sequence	No	Rev. 7
MUCtS00362	CRG/BDM	COP generates reset while BDM active	Yes	Rev. 7

1. Wording has changed from Revision 1.

As long as not noted otherwise all bugs will be fixed with the next revision of the silicon.

SECURITY FSEC WRITEABLE MUCTS00152

The FSEC register can be written in special modes. This allows the security of the MCU to be disabled easily.

Work-around None

PWM MUCTS00155

If the PWM interrupt enable bit is cleared (PWMIE = 0), the PWM interrupt flag PWMIF never gets set. This is not in line with the specification.

Work-around Avoid disabling the PWM interrupt locally.

ATD CHANNEL SELECTION MUCTS00163

On the ATD, the selection of analog channels is flipped, I.e. AN07 is selected by the input multiplexer instead of AN00.

ATD Converter	Result Register	Input being converted
0	0	AN07
0	1	AN06
0	2	AN05
0	3	AN04
0	4	AN03
0	5	AN02
0	6	AN01
0	7	AN00
1	0	AN15
1	1	AN14
1	2	AN13
1	3	AN12
1	4	AN11
1	5	AN10
1	6	AN09
1	7	AN08

The main impact is if not all 8 channels are converted, i.e. the conversion length is <8. In this case, the inputs must be re-routed on the PC-board.

The alternative is to always sample all 8 channels, at the expense of an overall increase in conversion.

Work-around If all 8 channels are converted (SC8,SC4,SC2,SC1) = 0, the software must pick up the conversion result from a different result register to that detailed in the Technical Data Book.

BDM NARROW EXPANDED MODE WORD READ MUCTS00164

BDM word read of narrow external space does not get proper data to the BDM. The second byte retrieved appears in both high and low bytes. Note that internal reads, wide external reads and narrow external byte reads all work properly.

Work-around Always use byte operations when using BDM to access narrow external space.

BDM NARROW EXPANDED MODE WORD WRITE MUCTS00165

BDM word write of narrow external space at addresses $FF00 to $FFFF does not complete the write properly. In order to do this correctly, the BDM ROM must be removed from the map. The problem occurs because the access is broken into two cycles: the first cycle properly removes the BDM from the map, but the second does not. The second byte is therefore not written to external space. Note that internal writes, wide external writes and narrow external byte writes all work properly.

Work-around Always use byte operations when using BDM to access narrow external space.

PORTA AND PORTB PULL-UPS MUCTS00168

Instead of pull-up resistors, pull-down resistors are enabled on PortA and PortB when the corresponding enable bits (PUPAE and PUPBE) are set in the PUCR register.

Work-around If the pull-up feature is not used, there is no impact.

If pull-ups are required, external pull-ups must be chosen and the PUPAE, PUPBE bits should not be set.

PORT A, B, E AND K PULL-UPS MUCTS00169

Pull-up resistors are selected on Ports A, B, E and K even if the pins are configured as outputs (DDR = 1). This leads to excess current. For Ports E and K, the pull-ups are turned on after reset. For Ports A and B, the pull-ups are turned off out of reset as specified.

Work-around Use internal pull-ups only if the entire port is configured as input.

If all pins of a port are used either as inputs or outputs, but not a mixture of both, there is no problem.

If on PortE or PortK pins used as outputs, the corresponding pull-up enable bit must be turned off.

If used as inputs, no problem exists.

In mixed applications (some pins used as input, others used as outputs) the pull-ups should be turned off and external pull-ups used instead.

DEVICE ID REGISTER MUCTS00171

The Device ID register is not correctly programmed. Locations $1A and $1B contain $1256 instead of $0000 as specified.

Work-around Use $1256 if a compare for the PC9S12DP256 mask set K36N is required.

The Data Book will be updated accordingly.

IIC BIT RATES MUCTS00172

IIC bus clock rate values not compatible with other Motorola IIC implementation.

	MC9S12DP256	MC68HC912DG128
Scale Register	**scl2tap**	**scl2tap**
IBC[5:3] = 000	5	4
IBC[5:3] = 001	6	4

Work-around Depending on the bit rate, another combination of pre-scaler values could be possible.

PLL LOCK DELAY MUCTS00174

Due to internal synchronization between different clock domains, the setting and clearing of the LOCK flag is delayed by 5 PLL clocks and 5 IP bus clocks maximum. This delay occurs after writing to the SYNR or after the REFDV register failing to clear the lock bit immediately.

Work-around Add a 5 PLL and 5 IP bus clock delay following a write to the SYNR or REFDV registers before the polling of the LOCK bit.

SECURITY MUCTS00176

With the part in a secured state, the following sequence can be executed to defeat security.

1. Reset into special expanded mode (ST)
2. Load code into RAM
3. Jump to RAM
4. Execute Code from RAM:

This writes $00 or $80 to the mode register, configuring the part for single chip mode. Memory reads are now possible.

Work-around None

BKGD REDUCED DRIVE MUCTS00177

Reduced drive strength on BKGD pin, drive strength enable signal tied to low.

Work-around None, however minimizing the load on the BKGD pin circumvents the problem in most cases.

BDM / CALL INSTRUCTION MUCTS00182

If the BDM is forced to steal a bus cycle from the CPU one cycle before a CALL instruction starts, and the last cycle of the instruction prior to the CALL is a write, the write will be performed to the PPAGE register rather than the desired location.

Similarly, if the BDM is forced to steal a bus cycle during the “U” cycle of the RTC instruction, the unstuck read will not return the correct data, leading to an incorrect return address being used.

Work-around Insert two consecutive “NOP” instructions before any CALL or RTC instructions. This might not be possible using High Level Languages.

The probability is however very low.

SELF-CLOCK MODE “CLOCK SWITCH” MUCTS00183

In external oscillator mode (PE7 = 0 during reset) a constant high on EXTAL prevents switching into self-clock mode.
In crystal mode (PE7 = 1 during reset) a constant low on EXTAL prevents switching into self-clock mode.

Work-around None

SELF-CLOCK MODE WITH LOW INPUT FREQUENCIES MUCTS00184

At low frequencies (<20KHz) the system does not detect loss of oscillation and therefore does not enter self-clock mode or clock failure reset.

Work-around None

MASS ERASE OF PROTECTED FLASH MUCTS00185

Mass-Erase commands are not prevented when any protection is active on a Flash block. Protection violations are detected by the incoming address only, so a Mass-Erase can be issued by initiating a command sequence with an array write to a non-protected sector of a flash block.

Work-around None

MASS ERASE OF PROTECTED EEPROM MUCTS00186

Mass-Erase commands are not prevented when any protection is active on the EEPROM. Protection violations are detected by the incoming address only, so a Mass-Erase can be issued by initiating a command sequence with an array write to a non-protected sector of the EEPROM.

Work-around None

RTI INTERRUPT IN WAIT MUCTS00187

RTI fails to generate interrupt when in WAIT mode with SYSWAI=1.

Work-around Do not set SYSWAI bit. This will result in an increased current consumption.

PLL FAILS TO POWER DOWN IN WAIT MODE MUCTS00188

PLL fails to deselect and power down when entering WAIT mode with PLLWAI bit set.

When the part enters WAIT mode with the PLLWAI bit set and the PLL on, the PLL should be automatically powered down and deselected as the system clock source.

Currently at PLL frequencies above (1.7 x EXTAL) clock the PLL is not powering down upon entering WAIT mode.

Work-around This behavior leads to an increased power consumption in WAIT mode. A possible workaround is to deselect the PLL (PLLSEL = 0) and turn it off (PLLON = 0) before going into WAIT.

SPI CPHA CHANGE MUCTS00189

Changing the CPHA bit before an SPI master transmission is completed, i.e. before the SS line gets de-asserted leads to additional pulses on the SCK line. Disabling the SPI (SPE = 0) before an SPI master transmission is completed results in a SPI lock situation.

Work-around Several options exist here:

Avoid changing the CPHA bit, check if CPOL does the job.

Wait for SS line de-asserted (read the value using the PTIS register).

ATD STOP CURRENT MUCTS00190

The ATD draws excess STOP current of around 100µA per module in STOP mode.

Work-around None

MISSING INTERRUPTS IN STOP MODE MUCTS00191

No interrupt and therefore no exit from STOP is generated if the STOP mode is entered with an active level on the input line. This applies to all Port H, J, and P interrupt inputs.

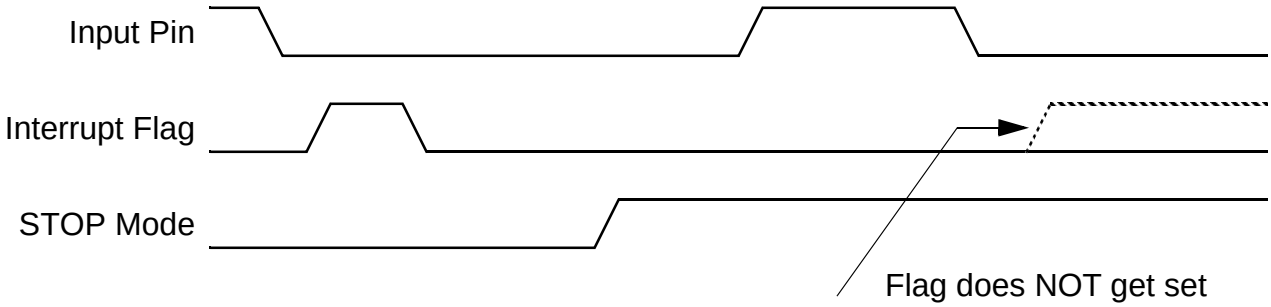

Figure 1 Example shows falling edge sensitivity

Work-around Verify by software input level before going into STOP, in case of active level do not go into STOP mode.

SPI FREEZE, BAUD RATE PRESCALER >=16 AND CHAFF = 1 MUCTS00202

Operating in master mode with CPHA=1 and baud rate prescalers greater than or equal to 16, SPI freezes if data register is written immediately after SPTEF is set.

Work-around Don’t use baud rate prescaler>=16 with polled I/O or baud rate preschooler >=40 with IRQ driven I/O in conjunction with CPHA=1.

SPIF CLEARING PROBLEMS MUCTS00204

The write access to the data register clears the SPIF flag in the current implementation and specification. If the transmit interrupt handler reads SPISR (with SPIF set) and writes to the SPI data register, this clears SPTEF (the desired outcome) and clears SPIF simultaneously (the undesired outcome). This is a potential race condition between the write and the read path on the SPI.

Work-around

Don't use the SPTEF feature.

This behavior will be fixed in the next revision by clearing the SPIF flag only by reading and not by writing. The flag name will also change to SPRF (SPI Receiver Full) like on the HC08 to better reflect the purpose of this flag.

BDM BACKGROUND COMMAND DURING CUP WAIT MODE MUCTS00209

The BDM hardware command BACKGROUND should not be used while the CPU is in wait mode.

Work-around

None available.

BDM LOSES SYNC WHEN USING PLL MUCTS00210

When using the BDM constant clock source, i.e. CLKSW=0, with the PLL engaged, PLLSEL=1, the BDM can lose communication with the host system, if the bus clock frequency is equal to the crystal frequency.

Work-around

When using the PLL and the constant rate BDM clock, the bus clock frequency should be a minimum of 2 times faster than the crystal frequency.

This behavior will not be fixed in future Silicon revisions.

E CLOCK STRETCHED HIGH 1 CYCLE WHEN WRITING IVIS MUCTS00213

External E clock on PE4 stretches high for 1 cycle when the IVIS bit is cleared in the MEBI MODE register ($000B). Writing this register is an internal access. Internal accesses with IVIS=0 should stretch the external E clock low for all internal accesses.

Work-around

None available.

IBIF/IBAL RESET NOT COMPATIBLE WITH EXISTING HC12 MUCTS00214

The IIC flag reset strategy for IBIF/IBAL flags is not compatible with other existing HC12 implementations. On PC9S12DP256 the arbitration lost flag IBAL and the interrupt flag IBIF must be cleared by software by writing a low to it.
e.g. on MC68HC912DG128 these flags must be cleared by software by writing a one to them.

Work-around None available

FORCED & SUCCESSFUL OC OCCUR AT THE SAME TIME MUCTS00239

If forced output compare on any timer channel occurs at the same time as the successful output compare then forced output compare action will not take precedence. The interrupt flag will be set.

Work-around None available

TOF FLAG GETTING GENERATED AT WRONG PLACE MUCTS00240

If TC7=$FFFF and TCRE=1, TOF will be set when TCNT is reset from $FFFF to $0000”

Work-around None available

CRG SWITCHES TO SELF-CLOCK MODE DUE TO LONG OSCILLATOR STARTUP TIMES MUCTS00242

CRG initial time-out does not provide enough time for the oscillator to start properly. Measurements of the oscillator startup times show a minimum startup time of 10ms for crystal resonators. This is more than the time-out provided by the CRG which is 1.5ms to 8ms depending on the minimum VCO frequency. This potentially causes the MCU to start in Self-Clock Mode after power-up.

Work-around None available

OSCILLATOR START-UP PROBLEM MUCTS00243

The oscillator start-up time with crystal quartz greater or equal 4MHz can be in the 100ms range. The oscillator does not start with quartz below 4MHz.

Work-around None available

OSCILLATOR INPUT IS NOISE SENSITIVE MUCTS00244

Oscillator input is noise sensitive. Especially at low oscillator frequencies (< 4 MHz) erroneous CPU behavior may occur.

Work-around

- Use ceramic resonator with a frequency of 4MHz or above.
- Use an external oscillator and operate the part in noncritical mode.

FLASH CCIF & CBEIF FLAG NOT SET DURING STOP MUCTS00283

If the CPU executes a STOP instruction while a program or erase algorithm is being executed on the flash, the algorithm is immediately stopped but the CCIF flag will not set to 1. Additionally, if a second command has been pipelined (CBEIF=0) the CBEIF flag will not set to 1. As a result, the command state machine hangs and prevents subsequent programming and erasure of any of the 4 flash blocks until a reset occurs.

Work-around It is not, in any case, recommended to allow STOP to execute while an algorithm is running on the flash as the resulting state of the array will be unknown.

EEPROM CCIF & CBEIF FLAG NOT SET DURING STOP MUCTS00284

If the CPU executes a STOP instruction while a program or erase algorithm is being executed on the EEPROM, the algorithm is immediately stopped but the CCIF flag will not set to 1. Additionally, if a second command has been pipelined (CBEIF=0) the CBEIF flag will not set to 1. As a result, the command state machine hangs and prevents subsequent programming and erasure of any of the 4 flash blocks until a reset occurs.

Work-around It is not, in any case, recommended to allow STOP to execute while an algorithm is running on the EEPROM as the resulting state of the array will be unknown.

FLASH 0 FAIL FLAG NOT SET ON ILLEGAL COMMAND MUCTS00285

FAIL flag does not set in FSTAT register for flash 0 when an illegal command is written in special modes.

Work-around None

EEPROM FAIL FLAG NOT SET ON ILLEGAL COMMAND MUCTS00287

FAIL flag does not set in ESTAT register for EEPROM when an illegal command is written in special modes.

Work-around None

FLASH FCMD REGISTER IS OVERWRITTEN WHEN CBEIF =0 MUCTS00288

It is possible to write to the FCMD register while CBEIF=0. If the command written to FCMD differs from the pending command then the pending command will not be executed.

Work-around It is always recommended to test that the CBEIF flag is set in the FSTAT register before executing another command sequence.

EEPROM ECMD REGISTER IS OVERWRITTEN WHEN CBEIF =0 MUCTS00289

It is possible to write to the ECMD register while CBEIF=0. If the command written to ECMD differs from the pending command then the pending command will not be executed.

Work-around It is always recommended to test that the CBEIF flag is set in the ESTAT register before executing another command sequence.

SPI: NO DE-ASSERTION OF SS BETWEEN DATA IF CPHA=0 MUCTS00293

If the SPI is configured as master with CPHA=0 and new data is written immediately after SPTEF is set, the slave select line (SS) between data transmissions is not de-asserted.

Work-around Write new data after SPIF is set.

SPI: IMPROPER SPTEF HANDSHAKE IN SLAVE MODE MUCTS00302

If the SPI is configured as slave, SPTEF is not cleared if transmit data register is full. If the SPI is operating in SLAVE mode, SPTEF is always read as'1'. New data is immediately written to shift register and no queue feature is available.

Work-around Do not use interrupt routines based on SPTEF Flag.

ATD ACCURACY OUT OF SPECIFICATION MUCTS00309

The ATD accuracy under worst case conditions is +-5counts in 10-Bit mode. Specified are +-2 counts.

Work-around None

ECT TOGGLE ON OVERFLOW ERROR MUCTS00310

Toggle on timer overflow doesn't work if a prescaler value greater than 1 is engaged.

Work-around None

MSCAN DOUBLE INTERRUPTS ON STATUS CHANGE MUCTS00318

MSCAN generates “RxOK” CSCIF interrupt before a RxWrn/TxWrn interrupt when returning from the RxErr/TxErr state to the RxWrn/TxWrn state.

When receiving corrupted messages and reaching the RxERR state, all appears fine. When the first good message is received after that, REC resets to a value between 119 and 127. But instead of generating one single transition interrupt on CSCIF when returning to the RxWrn state, the MSCAN enters the RxOK state before the RxWRN state. This causes two CSCIF interrupts. The same problem exists for the TxErr/TxWrn scenario.

Work-around Carefully analyze status bits in the interrupt routine and account for the fact that a single transition may cause two interrupts in the software.

LSTRB/TAGLO PIN DRIVER CONTENTION MUCTS00324

A pin drive conflict exists between the LSTRB output signal and the TAGLO input signal while the external clock is being stretched. This conflict occurs because the output buffer enable signal for the LSTRB output is derived from an internal, unstretched clock.

Work-around None

INTERRUPT MISSED SWITCHING INTO STOP/WAIT MUCTS00325

An interrupt that occurs after the last clock edge immediately proceeding the assertion of cpu_stop or cpu_wait can be missed. The stop and wait signals are used to switch a multiplexer, which selects between level sensitive interrupts and edge sensitive interrupts. Edge sensitive interrupts are used while in stop or wait, since the clocks will be halted during this condition, thus disabling the FF used by the normal level sensitive interrupt logic. This causes a partial dead cycle where an interrupt can be missed.

Work-around None

PIM: GLITCHES NOT FILTERED CORRECTLY MUCTS00326

If a glitch with a length of t1=(0.25... 0.50)*t_wakeup is followed by a second glitch with a minimum length of t2=t_wakeup-t1 an interrupt will be generated.

Work-around None

FCLKDIV WRITABLE SEVERAL TIMES IN NORMAL MODE MUCTS00329

If the FCLKDIV register is written in special mode and then the mode is changed to NORMAL mode, the FCLKDIV register can be written more than once.

Work-around None

ECLKDIV WRITABLE SEVERAL TIMES IN NORMAL MODE MUCTS00330

If the ECLKDIV register is written in special mode and then the mode is changed to NORMAL mode, the ECLKDIV register can be written more than once.

Work-around None

TEOD, IMSG, RX$XE, TSIFR AND TMIFR[1:0] DO NOT CLEAR MUCTS00331

The following BDLC control register bits have an automatic clearing mechanism: TEOD, IMSG, RX4XE, TSIFR and TMIFR[1:0]. For a detailed description of the different clearing conditions please refer to the BDLC Control register description. The automatic clearing of the TEOD, IMSG, RX4XE, TSIFR and TMIFR[1:0] bits doesn’t work. These control bits stay set until the control bits are cleared by the programmer.

The automatic clearing for the TEOD, IMSG, RX4XE, TSIFR and TMIFR[1:0] bits is implemented as a synchronous reset. The TEOD, IMSG, RX4XE, TSIFR and TMIFR[1:0] bits get only a clock edge when the BDLC register space is accessed. This means the control bits get cleared automatically only when BDLC is accessed.

Work-around Read or write to any of the registers in the BDLC address space to update the flags mentioned above.

FLASH ARRAY READ GENERATES ACCERR DURING COMMAND SEQUENCE MUCTS00339

During command sequence entry, an array access to any Flash block will generate an access error. The ACCERR flag will set in the FSTAT register associated with the flash block which is being programmed or erased. The ACCERR will prevent the program or erase operation from starting. As a result, it is not possible to execute code in one flash block to program or erase another flash block. After a valid array write, the instruction to write to the FCMD register must be fetched from the flash. This fetch generates a flash array select which in turn triggers the ACCERR.

Work-around Do not program or erase out of Flash

COP GENERATES RESET DURING BDM ACTIVE MUCTS00362

If COP is enabled and BDM gets active, the COP will cause a system reset. Previous HC12 derivatives had a bit called RSBCK bit which is not available in the CRG. This bit, when set, stops the COP and RTI counters when entering active BDM mode. Although this is not a bug, since the part behaves as specified, it is severe enough to be tracked here.

Work-around Debug with disabled COP.

Additional mask set errata can be found on the World Wide Web at http://www.mcu.motsps.com/documentation/index.html