TRANSPORTATION SYSTEMS GROUP
MASK SET ERRATA AND INFORMATION SHEET
Part: HC912B32.09H91F.A Mask Set:
Report Generated: Aug 06, 99 02:00

HC912B32.09H91F.A Modules
Current Module Revision
ATD8B8C.2.8
BDLC.1.4
BDM256.3.0
BKP.2.2
CORNER.2.3
CPU.3.3
EE768.2.2
FEE32KB2K.1.4
INT.4.0
IOB.2.3
LIM_B32.3.2
MEBI_B32.3.1
MMI_B32.2.1
MSI1C1P.2.2
PADS_B32.3.1
PWM8B4C.2.3
RAM1K.1.6
ROC.3.2
TIM16B8C.2.3
VDD.3.2
VDDA.3.2
VDDX.3.2
VFP.3.1
VPP.3.1
VREF.1.1
VSS.3.2
VSSA.3.2
VSSX.2.2
WCR.2.0
XTAL.2.3

HC12_AR_301 **Customer Erratum** **HC912B32.09H91F.A**

DESCRIPTION:
The present STOP Idd test limit is set to a higher limit on the H91F and J54E mask sets which is greater than the expected specification limit of:10uA -40C to 85C25uA -40C to 105C50uA -40C to 125C

WORKAROUND:
None

HC12_AR_456 **Customer Erratum** **HC912B32.09H91F.A**

DESCRIPTION:
Flash requires 250nsec delay for wakeup from wait mode with FEESWAI=1. If the operating bus frequency is greater than 4MHz, the Flash can not be used when recovering from WAIT mode when the FEESWAI bit is equal to ’1’.

WORKAROUND:
If your interrupt vectors are located in the Flash array, do not set the FEESWAI bit in Wait mode.

HC12_AR_374 **Customer Erratum** **ATD8B8C.2.8**

DESCRIPTION:
ATD status bits and conversion counter are not reset properly when writing any ATD register while an active A/D conversion sequence is in the process of completion.

WORKAROUND:
When starting a new sequence, perform two writes to control registers 4/5 in quick secession. If the first write occurs when the status bit/conversion counter is not reset, the second write will correct ATD operation.

HC12_AR_311 **Customer Information** **ATD8B8C.2.8**

DESCRIPTION:
The (VRH-VRL)/2 internal reference conversion result may be $7F, $80 or $81.

WORKAROUND:
If the (VRH-VRL)/2 internal reference is used (perhaps for system diagnostics), expected pass result may be $7F, $80 or $81.

HC12_AR_312 **Customer Erratum** **BDLC.1.4**

DESCRIPTION:
After initiating a transmission, if the BDLC is put into Analog Loopback Status mode (ALOOP=’1’) in the middle of that transmission, there is a possibility that the BDLC will lock-up and continually stream out it’s last data bit as a pulse train. This causes a J1850 network lock-up.

WORKAROUND:
To avoid this condition, do not toggle the ALOOP bit during a transmission.If you enter into this lock-up condition, put the MCU into STOP or WAIT mode or initiate a system reset.

HC12_AR_493 **Customer Erratum** **BDLC.1.4**

DESCRIPTION:
To transmit a message using the BDLC, the user writes the first byte of the message to be transmitted into the BDLC data register (BDR). This will initiate the transmission process at the beginning of the next idle bus state.An invalid symbol being received by the BDLC clears any byte that had been previously written to the BDR. This will inhibit the transmission process until the user writes another byte to the BDR. The following scenario describes an event sequence that would prevent the user from knowing that an invalid symbol was received and that the BDR had been cleared.

WORKAROUND:
A two level strategy has been developed that positively signals the need to restart the transmission of a message. The first level looks for the special case of reception of an illegal symbol with a byte pending transmission in the BDLC data register, as described above. The second level uses a transmit watchdog timer to spot any case of a transmission not occurring within a maximum amount of time.1a) If an illegal symbol interrupt occurs with a byte pending transmission in the BDR, reload the BDR with the first byte of that message to restart transmission.

HC12_AR_519 **Customer Erratum** **BDLC.1.4**

DESCRIPTION:
CRC error in received message does not prevent IFR transmission.

WORKAROUND:
In response to the CRC Error interrupt ($18 in BSVR), immediately write an $FF byte into the BDR, and then clear the lower four bits in BCR2 (TSIFR, TMIFR0, TMIFR1, TEOD). This will cancel the IFR transmission.

HC12_AR_520 Customer Erratum BDLC.1.4

DESCRIPTION:
When programmed to transmit a single byte Type 2 IFR (byte loaded in the BDR and TSIFR bit set in BCR2), the BDLC will attempt to transmit a single IFR byte following the EOD of the message currently being received. If the byte to be transmitted loses arbitration, the BDLC will continue to retry transmission until it is successful or an error occurs or the TEOD bit is set.

WORKAROUND:
(Short form) When the first RXIFR interrupt occurs the requester message has been received without errors (invalid symbol or CRC). When the BDLC receives back correctly the IFR byte is was trying to transmit, then response by this node is complete and the requester message received can be passed to the application layer. Ignore any invalid symbol error that occurs after the first RXIFR interrupt, since transmission of IFRs are not retried.

HC12_AR_463 Customer Erratum BKP.2.2

DESCRIPTION:
Breakpoint Address and Data registers are properly reset only upon Power on Reset.

WORKAROUND:
To insure BRKAH, BRKAL, BRKDH and BRKDL registers have the correct default values, always clear each immediately after reset.

HC12_AR_287 Customer Erratum CPU.3.3

DESCRIPTION:
The ETBL instruction will provide an incorrect result under EITHER of the following 2 conditions:1) Y2 less than Y1 and C-bit = 0 before ETBL instruction executed2) Y2 greater than Y1 and C-bit = 1 before ETBL instruction executed except for the cases where (B * (Y2 - Y1)) = 0 (i.e. B=0 or Y2=Y1)

WORKAROUND:
If a borrow is needed from the equation, then set C-bit=1 before executing the ETBL instruction; if a borrow is not needed than clear C-bit=0 before executing the ETBL instruction. Example: LDxy (idx); initialize index register to point to the start point LDD (Y2); get value of Y2 into ACCDCPD (Y1); D-M, Y2-Y1, This updates C-bit LDAB (B); accumulator B initialized with ratio ETBL (idx); Perform instruction

HC12_AR_288 **Customer Erratum** **CPU.3.3**

DESCRIPTION:
If the REV or REVW instructions are interrupted while processing a rules list, the results from the instructions may be incorrect. The Condition Code Register and Index Register Y (weight pointer for REVW) may be incorrect when the stacking occurs for the interrupt. The REV and REVW instructions produce the wrong result after returning from the interrupt because the V-bit in the CC register and IY registers (REVW) may not hold the same state as prior to the interrupt.

WORKAROUND:
Disable the interrupts prior to using the REV or REVW instruction (which could increase the interrupt latency). If a non-maskable interrupt has been enabled, there is no workaround.

HC12_AR_469 **Customer Information** **FEE32KB2K.1.4**

DESCRIPTION:
Flash can not be programmed reliably at high temperatures.

WORKAROUND:
To enhance Flash programming reliability at elevated temperatures, the Flash bulk erase pulse time (Tepulse) should be reduced from 90-110ms to 5-10ms.

HC12_AR_528 **Customer Erratum** **INT.4.0**

DESCRIPTION:
When using an edge sensitive IRQ signal to trigger an interrupt service routine and the IRQ is not released during servicing, the part will not enter stop mode if the following executed instruction is STOP.

WORKAROUND:
When IRQ is set to be edge sensitive, release pin before executing STOP instruction.

HC12_AR_441 **Customer Erratum** **INT.4.0**

DESCRIPTION:
There is a cycle at the beginning of executing a BDM instruction that is susceptible to being interrupted directly after the background has executed. Since the interrupt source is masked, the interrupt vector that is requested is $FFF6. The BDM firmware at $FFF6 points to the routine at

$FF24. The interrupt was stacked when it was taken, but the BDM routines do not un-stack (no RTI). When you return from BDM the stack pointer is pointing to the wrong place. Avoid using TRACE during cycles that allow interrupts to become unmasked (i.e. TRACE of a CLI if interrupts are pending). Caution should a

WORKAROUND:
None.

HC12_AR_527 Customer Information INT.4.0

DESCRIPTION:
If the source of an interrupt is taken away by disabling the interrupt without setting the I mask bit in the CCR, an SWI interrupt may be fetched instead of the vector for the interrupt source that was disabled.

WORKAROUND:
Before disabling an interrupt using a local interrupt control bit, set the I mask bit in the CCR.

HC12_AR_510 Customer Erratum IOB.2.3

DESCRIPTION:
Expanded mode operation causes an increase in bus driver shoot through current pin leakage which can cause inaccurate A/D conversions.

WORKAROUND:
Enabling reduced drive on ports A, B, and E while the HC12 is in expanded mode reduces the amount of VDD/VSS noise caused by bus driver shoot through current. Therefore, the A/D accuracy meets specified limits.

HC12_AR_306 Customer Erratum MEBI_B32.3.1

DESCRIPTION:
In narrow expanded modes, addresses on Port B are not held past the multiplexed address hold time. Port B switches to high impedance input during the data cycle and is not held throughout the cycle as on the HC11.

WORKAROUND:
A 16 bit address latch is required in all expanded modes.

HC12_AR_359 **Customer Erratum** **PWM8B4C.2.3**

DESCRIPTION:
PWM boundary cases in Center Mode are not according to specification1. When PWDTYx is greater than PWPERx, the pin is stuck in the opposite value of the specification (i.e. the state opposite the PPOLx value).2. When PWDTYx = PWPERx, the pin will change value when the counter reaches the new duty value.3. When the PWM channel is enabled, and the period and duty registers are changed such that PWPERx = $00 and the new PWDTYx value is between zero and the old PWPERx value, the pin will change to the opposite value of the specification.

WORKAROUND:
1. If PWDTYx value is greater than the PWPERx value, change the PPOL bit.
2. No workaround available.
3. Update the PWDTYx and PWPERx registers when the associated channel is disabled; or for the boundary case to be correct when PWPERx = $00, PWDTYx should also be equal to $00.

HC12_AR_363 **Customer Erratum** **PWM8B4C.2.3**

DESCRIPTION:
In center or left aligned modes, when operating in 8-bit or 16-bit configurations, the PWM pins associated with channels 1 and 3 may have an incorrect value when enabled or after disable/enable sequence. If DTY register is written when the channel is enabled, the new duty value will not transferred from buffer to register when the channel is disabled.

WORKAROUND:
While channels 1 or 3 are disabled, write to the associated PWPER and PWDTY register.

HC12_AR_364 **Customer Erratum** **PWM8B4C.2.3**

DESCRIPTION:
In left or center aligned modes with 8 or 16 bit configurations, when a PWM channel is enabled with PWDTY=$00, (therefore the associated waveform has a duty of $01), the associated port pin awakes with the wrong polarity.When operating in left aligned mode, the polarity will correct itself on the next PWM cycle.When operating in center aligned mode, the polarity will correct itself after the associated PWM counter advances from $00 to $01.

WORKAROUND:
None available.

HC12_AR_315 **Customer Erratum** **PWM8B4C.2.3**

DESCRIPTION:
Output waveform polarity may ‘flip’ when writing PWDTY register to $00 (or $0000 in 16-bit configuration) in CENTER mode. Problem occurs when writing PWDTY register with associated channel enabled or disabled.

WORKAROUND:
Program PWDTY=$FF (or $FFFF) instead of PWDTY=$00 (or $0000) to achieve “zero-duty”.

HC12_AR_332 **Customer Erratum** **PWM8B4C.2.3**

DESCRIPTION:
When PWDTY = $FF and PWPERx = $00, the pin value is opposite the value in the POLx bit.This activity is not according to the HC12 PWM documentation which states “If PWPERx = $00 & POLx = 0, then the output is always low. If PWPERx = $00 & POLx = 1, then the output is always high.”

WORKAROUND:
When programming PWPERx = $00 to get a boundary case, program PWDTYx <> $FF.

HC12_AR_358 **Customer Erratum** **PWM8B4C.2.3**

DESCRIPTION:
A write to any PWM register followed immediately by a read of a PWM counter register (PWCNTx) may cause a false counter reset.Note: If the PWM register written is a PWM counter register the counter may reset after both the write and read accesses.

WORKAROUND:
Add NOP instruction between the write and the counter-read operations.

HC12_AR_243 **Customer Erratum** **PWM8B4C.2.3**

DESCRIPTION:
16-bit concatenated PWM output works only if both PWM bits are enabled. For example, if you try to use a 16-bit PWM output on PP0(2), you will not get a PWM output with only PWEN0(2) set. You will, however, get a PWM output with both PWEN bits 0(2) and 1(3) set. Since PWEN1(3) must be set, you lose the GP I/O capability on PP1(3).

WORKAROUND:
To use 16-bit concatenated PWM, set both PWENx of the desired pair to ’1’.There is no workaround for the loss of GP I/O on the low order channel.

HC12_AR_331 Customer Erratum PWM8B4C.2.3

DESCRIPTION:
The PWM channel output becomes active on the PWM pin immediately after assertion of the associated PWENx bit. As a result, the start of the first PWM period may be truncated by up to one channel source clock period.This activity is not according to the HC12 PWM documentation which states “There is an edge-synchronizing gate circuit to guarantee that the clock will only be enabled or disabled at an edge.”

WORKAROUND:
None available.

HC12_AR_333 Customer Information PWM8B4C.2.3

DESCRIPTION:
HC11 code for the PWM module is not directly portable to the HC12.The PWM Concatenation (16-bit) mode implementation is not compatible with the HC11 PWM.When operating in Concatenation mode, the HC11 PWM channel output pin and clock source are both controlled by the “low-order” byte.i.e. if concatenate channels 1(3) & 2(4) the output pin is channel 2(4) and the clock source for the concatenated counter is the clock source for channel 2(4).When operating in Concatenation mode, the HC12 PWM channel output pin is the pin associated with the “high-order” byte and the clock source is controlled by the “low-order” byte.i.e. if concatenate 01(23), then pin 0(2) is the output and channel 1(2) controls the clock source.

WORKAROUND:
Modify system configuration based on HC12 specification information.

HC12_AR_360 Customer Information PWM8B4C.2.3

DESCRIPTION:
Write to PWM counter under the following conditions may result in opposite polarity on pin for one PWM cycle:1. Value of counter is greater than the value of duty register.2. The PWM channel is enabled.

WORKAROUND:
Only write PWM counter when associated channel is disabled (PWENx=0)

HC12_AR_148 **Customer Information** **PWM8B4C.2.3**

DESCRIPTION:
HC11 code for the PWM module is not directly portable to the HC12.

WORKAROUND:
Modify all HC11 duty register values to be the desired duty value - 1 on the HC12.

HC12_AR_327 **Customer Information** **PWM8B4C.2.3**

DESCRIPTION:
The PWM scaled clock (S0,S1) equations are incorrect in the HC12 documentation. The incorrect equations are: Clock S0 = A / 2 * PWSCAL0Clock S1 = B / 2 * PWSCAL1

WORKAROUND:
The correct PWM scale clock (S0,S1) equations are: Clock S0 = A / 2 * (PWSCAL0 + 1)Clock S1 = B / 2 * (PWSCAL1 + 1)Therefore, programming $FF is full scale divide of 256.

HC12_AR_328 **Customer Information** **PWM8B4C.2.3**

DESCRIPTION:
HC11 code for the PWM module is not directly portable to the HC12.The PWM scaled clock (S0,S1) equations are not compatible with the HC11 PWM.The HC11 PWM scaled clock equations are: Clock S0 = A / 2 * PWSCAL0Clock S1 = B / 2 * PWSCAL1The HC12 PWM scaled clock equations are:
Clock S0 = A / 2 * (PWSCAL0 + 1)Clock S1 = B / 2 * (PWSCAL1 + 1)

WORKAROUND:
Modify all HC12 PWM scaled clock (S0,S1) values to use the following equations: Clock S0 = A / 2 * (PWSCAL0 + 1)Clock S1 = B / 2 * (PWSCAL1 + 1)Therefore, programming $FF is full scale divide of 256.

HC12_AR_329 **Customer Information** **PWM8B4C.2.3**

DESCRIPTION:
The PWM Center-Aligned Mode Duty Cycle equations are incorrect in the HC12 documentation. The incorrect equations are: Center-Aligned-Output Mode: (center=1)Duty cycle = ((PWPERx - PWDTYx) / (PWPERx + 1)) X 100% for Polarity = 1 Duty cycle = ((PWDTYx + 1) / (PWPERx + 1)) X 100% for Polarity = 0

WORKAROUND:
The correct PWM duty cycle equations are: Duty cycle = [(PWPERx - PWDTYx) / (PWPERx)] X 100% for Polarity = 0Duty cycle = {(PWDTYx) / (PWPERx)] X 100% for Polarity = 1

HC12_AR_330 **Customer Information** **PWM8B4C.2.3**

DESCRIPTION:
The PWM Period equations are incorrect in the HC12 documentation. The incorrect equations are: Period = (Channel-Clock-Period / (PWPER + 1)) for center = 0Period = (Channel-Clock-Period / (2 * (PWPER + 1))) for center = 1

WORKAROUND:
The correct PWM Period equations are: Period = [Channel-Clock-Period * (PWPER + 1)] for center = 0Period - [Channel-Clock-Period * (2 * PWPER)] for center = 1

HC12_AR_198 **Customer Information** **PWM8B4C.2.3**

DESCRIPTION:
When the PWM counters are cleared by a write and then the PWM channels are enabled, sometimes the PWM channels are idle for one PWM period, then resume normal operation. From extensive testing, it appears that this idle period occurs only when the PWM channels are disabled while the PWM counters are between the duty cycle count (PWDTYx) and the period count (PWPERx).

WORKAROUND:
None.

HC12_AR_472 **Customer Erratum** **ROC.3.2**

DESCRIPTION:
When the device exits WAIT mode, it does not return to the correct location within the routine if the stack is positioned in external memory and if the stretch bits have been enabled to lengthen the clock.

WORKAROUND:
To overcome this problem, locate the stack in internal RAM resources and/or clear the stretch bits to prevent clock stretching.

HC12_AR_504 **Customer Erratum** **ROC.3.2**

DESCRIPTION:
When a clock monitor reset occurs, the crystal may start-up with no delay period. As a result, the MCU attempts to fetch reset vectors before the crystal has fully stabilized.

WORKAROUND:
When clock monitor failure has occurred, hold the reset signal until the crystal has reached stable oscillation.

HC12_AR_400 **Customer Information** **ROC.3.2**

DESCRIPTION:
When coming out of STOP by using reset, the crystal start-up delay time-out does not occur. This means the MCU may be attempting to run before the crystal has become stable.

WORKAROUND:
When coming out of STOP with reset hold the reset signal until the crystal has reached stable oscillation.

HC12_AR_285 **Customer Erratum** **TIM16B8C.2.3**

DESCRIPTION:
The output compare flags cannot be cleared until the timer increments past the matching count. This will cause a problem when using a timer prescaler value larger than divide-by-8. The result is the output compare interrupt is asserted continuously until the timer increments past the matching count.The same result will occur when an external clock source is used to drive the timer through the pulse accumulator pin.

WORKAROUND:
It takes at least 11 cycles to get into an interrupt service routine. If the effective prescale value is greater than divide-by-8, then the user must account for the number of cycles it takes to increment the timer counter past the match point before clearing the associated output compare flag and exiting the interrupt service routine.

HC12_AR_256 **Customer Erratum** **VFP.3.1**

DESCRIPTION:
Vfp (FLASH programming voltage) pin does not meet ESD protection specifications.

WORKAROUND:
Use extra care when handling parts.

HC12_AR_257 **Customer Erratum** **VPP.3.1**

DESCRIPTION:
Vpp pin does not meet ESD expectations.

WORKAROUND:
Use extra care when handling parts.

HC12_AR_291 **Customer Erratum** **WCR.2.0**

DESCRIPTION:
When an I-type interrupt occurs at the same time as an RTI interrupt, the program address at $FFC0 will be fetched.

WORKAROUND:
Point the $FFC0 vector to a return from interrupt (RTI) instruction. This will get the program returned to start processing the proper interrupt as quickly as possible.

HC12_AR_502 **Customer Erratum** **WCR.2.0**

DESCRIPTION:
When a clock monitor reset occurs, the crystal may start-up with no delay period. As a result, the MCU attempts to fetch reset vectors before the crystal has fully stabilized.

WORKAROUND:
When a clock monitor failure has occurred, hold the reset signal until the crystal has reached stable oscillation. Since it is not generally possible to detect the cause of reset with external circuitry, you could design your external reset so it always holds the reset pin low long enough to allow the oscillator to stabilize.

HC12_AR_504 **Customer Erratum** **WCR.2.0**

DESCRIPTION:
When a clock monitor reset occurs, the crystal may start-up with no delay period. As a result, the MCU attempts to fetch reset vectors before the crystal has fully stabilized.

WORKAROUND:
When clock monitor failure has occurred, hold the reset signal until the crystal has reached stable oscillation.

Last updated: Wednesday, 18-Aug-1999 16:02:07 CDT