Freescale Semiconductor
Mask Set Errata

MAC7202MSE1
Rev. 8, 06/2008

MAC7200 Microcontroller Device Mask Set Errata

This document identifies implementation differences (summarized in Table 1) between specific MAC7200 family microcontroller mask sets and the functional descriptions contained in the *MAC7200 Reference Manual* (MAC7200RM).

1 Introduction

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

- 0M34A mask of MAC7202, MAC7212, MAC7242
- 1M34A mask of MAC7202, MAC7212, MAC7242
- 0M84D mask of MAC7202, MAC7212, MAC7242
- 1M84D mask of MAC7202, MAC7212, MAC7242
- 0M19G mask of MAC7201, MAC7211, MAC7241

When contacting a Freescale representative for assistance, please have the MCU device mask set and date code information (described below) available.

1.1 MCU Device Part Number Prefixes

All MAC7200 family devices are marked with a PAC, MAC or SAC prefix. These prefixes denote the following:

PAC	Devices that have been tested, but are not fully characterized or qualified over the full range of normal manufacturing process variations.
MAC	Fully characterized and qualified standard devices.
SAC	Fully characterized and qualified special or custom devices.

1.2 MCU Device Mask Set Identification

The mask set of a device is identified by a four-character code consisting of a letter, two numerical digits, and a letter, for example L38Y. Slight variations to the mask set identification code may result in an optional numerical digit preceding the standard four-character code, for example 0M34A.

1.3 MCU Device Date Codes

In addition to the part number and mask set markings, each device is marked to indicate the week of manufacture. The date is coded as four numerical digits where the first two digits indicate the year and the last two digits indicate the work week. For example, the date code “0412” indicates that the device was manufactured during the 12th week of the year 2004.

1.4 Errata System Tracking Numbers

MUCts0*xxxx* and PSxxxx are tracking numbers for MAC7200 family device errata. An errata number can be used with the mask set and date code to identify a specific errata to a Freescale representative.

2 Errata Summary

Table 1. Summary of MAC72*x*2 Mask Set Errata

Errata Number	Brief Description	Module(s) Affected	Mask Set Affected				
			0M34A	1M34A	0M84D	1M84D	0M19G
MUCts01793	IPM/IPWM Modes, Value Read From UCAn May Be Incorrect	eMIOS	Yes	Yes	—	—	—
MUCts01831	Deactivating A Receive MB May Corrupt Another Active Receive MB	FlexCAN	Yes	Yes	—	—	—
MUCts02515	ATD Not Reset Correctly Under Specific Conditions	ATD	Yes	—	—	—	—
MUCts02523	FlexCAN Transmit Buffers May Freeze or Indicate Missing Frame	FlexCAN	Yes	Yes	—	—	—
MUCts02605, MUCts03041	ATD Predischarge Feature Non-functional	ATD	Yes	Yes	—	—	—
MUCts03021, MUCts03540	Changing DSPI CTARs Between Frames in Continuous PCS Mode Causes Error	DSPI	Yes	Yes	Yes	Yes	Yes
MUCts03024	Power-On-Reset Does Not Properly Reset the Flash	Flash	Yes	Yes	—	—	—
MUCts03027	Flash Smart Erase Does Not Work Correctly	Flash	Yes	—	—	—	—
MUCts03071, MUCts03072	eMIOS Comparators A and B Enabled by Writing to A2 and B2 in DAOC Mode	eMIOS	Yes	Yes	—	—	—
MUCts03073, MUCts03074	Writes to eMIOS MCR Register in IPM and IPWM Modes May Cause Incoherent Reads	eMIOS	Yes	Yes	—	—	—
MUCts03075	Flash Smart Erase Disabled	Flash	—	Yes	—	—	—
MUCts03132	eMIOS B Not Updated Correctly With 8-bit Accesses	eMIOS	—	—	Yes	Yes	—
MUCts03240	Debug Mode “Convert Then Freeze” Fails	ATD	—	—	Yes	—	—
MUCts03347	Expanded Mode Requires External Clock	PIM	—	—	Yes	Yes	—

Table 1. Summary of MAC72*x*2 Mask Set Errata (continued)

Errata Number	Brief Description	Module(s) Affected	Mask Set Affected				
			0M34A	1M34A	0M84D	1M84D	0M19G
MUCts03458	Conversion Reset Implementation Does Not Comply with the Reference Manual	ATD	—	Yes	—	—	—
MUCts03533, MUCts03534	Large Blocks Limited to 10,000 Program/Erase Cycles	Flash	—	—	Yes	Yes	Yes
MUCts03562	Minor Rev-ID not incremented	mcu_ mac7202	—	—	—	Yes	—
PS4414	Corrupt ID may be sent in early-SOF condition	FlexCan	Yes	Yes	Yes	Yes	Yes
MUCts03958	Low pulse on LIN RX line may prevent assertion of transmit data ready flag TXRDY	eSCI	Yes	Yes	Yes	Yes	Yes
MUCts03989	LIN fast bit error detection causes incorrect LIN reception	eSCI	Yes	Yes	Yes	Yes	Yes
MUCts03990	Automatic reset of LIN state machine causes incorrect transmission	eSCI	Yes	Yes	Yes	Yes	Yes
MUCts03991	LIN slave timeout flag STO not asserted if CRC is received too late	eSCI	Yes	Yes	Yes	Yes	Yes

3 Errata Details

This section provides a detailed description of each errata and a description of a possible workaround, where appropriate.

3.1 MUCts01793 — IPM/IPWM Modes, Value Read From UCA*n* May Be Incorrect

Description

When reading the UCA*n* register in Input Pulse Width Measurement (IPWM) or Input Period Measurement (IPM) modes, if the IPS bus cycle starts on the same clock cycle as an A2 capture, the data read will not be coherent with the one at the next UCB*n* read.

- In IPWM mode, data read from UCB*n* will be greater than UCA*n* (UCB*n* minus UCA*n* will be the pulse width measurement of the polarity opposite that defined by EDPOL).
- In IPM mode, data read from UCA*n* and UCB*n* will be the same.

The expected scenario is that UCA*n* will be greater than UCB*n* for both modes. Note that coherency is guaranteed in a sequence of several measurements only if the combined UCA*n* / UCB*n* reads for each new measurement are performed after the correspondent new flag event.

Workaround

After reading UCA*n* and UCB*n*, if UCA*n* is not greater than UCB*n*, discard this pulse measurement and read both registers again in the usual order: first read UCA*n*, then read UCB*n*.

3.2 MUCts01831 — Deactivating A Receive MB May Corrupt Another Active Receive MB

Description

Deactivating a FlexCAN receive message buffer (MB) may cause corruption of another active receive MB if the following sequence occurs.

1. A receive MB is locked via reading the Control/Status word, and has a pending message in the temporary receive serial message buffer (SMB).
2. A message is received that matches a second receive MB, and is queued in the second SMB.
3. The first MB is unlocked during the time between the CRC field and the 6th bit of EOF.
4. The second MB is deactivated within 9 system clock cycles of the first MB being unlocked, resulting in corruption of the first MB.

Workaround

Do not write to the Control/Status word after initializing a receive MB, and use the IFLAG status bit to determine reception of a new frame, as the Control/Status field will always indicate FULL or OVERRUN after receiving the first frame.

If a write (deactivation) is required to the Control/Status field of an active receive MB, a delay of 25 CAN bit times plus 9 system clock cycles between unlocking one MB and deactivating another MB will avoid corruption, however frames may still be lost.

3.3 MUCts02515 — ATD Not Reset Correctly Under Specific Conditions

Description

When there are conversions ongoing and a write to the ATDTRIGCTL, ATDETRIGCH, ATDPRE or ATDMODE register occurs it is possible that the ATD is not reset correctly.

Workaround

Do not write to the ATDTRIGCTL, ATDETRIGCH, ATDPRE or ATDMODE registers when there are conversions ongoing.

3.4 MUCts02523 — FlexCAN Transmit Buffers May Freeze or Indicate Missing Frame

Description

If a received frame is serviced during reception of a 2nd frame identified for that same MB (message buffer) and a new Tx (transmit) frame is initiated during this time also, the Tx MB can become frozen and will not transmit while the bus is idle. The MB remains frozen until a new frame appears on the bus. If the new frame is a received frame, the frozen MB is released and will arbitrate for external transmission. If the new frame is a transmitted frame from another Tx MB, the frozen MB changes its C/S (control status word) and IFLAG to indicate that transmission has occurred although no frame was actually transmitted. The frozen MB occurs if lock, unlock and initiate Tx events all occur at specific times during reception of two frames. The timing of the lock event affects the timing window of the unlock event as follows:

Situation A) Rx MB is locked during the 2nd frame. A frozen Tx MB occurs if all three events occur:

1. A new transmission is initiated by writing its C/S sometime between CRC3 (3rd bit of CRC field) and EOF7 (7th bit of end of frame) of the 2nd frame.
2. The Rx MB is locked by reading its C/S word sometime after EOF6 of 1st frame and before EOF6 of 2nd frame. Note that the order of events 1) and 2) can be reversed.
3. The Rx MB is unlocked between EOF7 and intermission at end of the 2nd frame.

Notice in this situation that if the lock / unlock combination happen close together, the lock must have been just before EOF6 of the 2nd frame and therefore the system is very close to having an overrun condition due to the delayed handling of received frames.

Situation B) Rx MB was locked before EOF6 of the first frame i.e. before its IFLAG is set. This is a less likely situation but provides a larger window for the unlock event. A frozen Tx MB occurs if all three events occur:

1. The Rx MB is locked by reading its C/S word before EOF6 of the 1st frame.
2. A new transmission is initiated by writing its C/S word sometime between CRC3 and EOF7 of the 2nd frame.
3. The Rx MB is unlocked between CRC3 and intermission at end of the 2nd frame. Note that the order of events 2) and 3) can be reversed.

Notice in this situation that if the unlock occurs after EOF6, the first frame would be lost and the 2nd frame would be moved to the Rx MB due to the delayed handling of received frames.

In addition, there is an alternative set of events that can cause the frozen/missing Tx as described for situation C) below.

Situation C) Rx unlocked during bus idle.

1. an Rx MB is locked before EOF6 of an incoming frame with matching ID and remains locked at least until intermission. This situation would usually occur only if the received frame was serviced after reception of a second frame.
2. An internal arbitration period is triggered by writing a C/S field of an MB.
3. the locked Rx MB is unlocked within two internal arbitration periods (defined below) of ii) above.
4. 0xC is written to the C/S field of a Tx MB within these same two arbitration periods. This step is optional if a 0xC was written in ii) above.

Two internal arbitration periods are calculated as:

$$\frac{(2 \times \text{number of MBs}) + 16}{f_{\text{IPS}}} = t_{\text{ARB}} \qquad \textit{Eqn. 1}$$

The number of MBs can be reduced by writing to CAN_MCR[MAXMB]. Bus clocks are the high frequency bus clocks regardless of CAN_CR[CLK_SRC] setting.

Additional Notes:

- The received frames can be transmitted from the same node, but they must be received into an Rx MB.
- When the frozen Tx MB's IFLAG becomes set, an interrupt will occur if enabled.
- The timestamp of the missing Tx will be set to the same timestamp value as the last reception before it was frozen.
- If the user software locks the Rx MB before a frame is received, situation A can occur with a single received frame.
- The issue does not occur if there were any additional pending Tx MBs before CRC3.
- If multiple Tx MBs are initiated within the CRC3/EOF7 window (situation A and B) or two internal arbitration windows (situation C), they all become frozen.

Workaround

If received frames can be handled (lock/unlocked) before EOF6 of the next frame, situations A and C are avoided. If they are handled before CRC3, or lock times are below 23 CAN bit times, situation B is avoided.

If these conditions cannot be guaranteed by the existing system design, situations A and B are avoided by inserting a delay of at least 28 CAN bit times between initiating a transmission and unlocking an Rx MB and vice versa. Typically a system will use a mechanism to selectively add the necessary delay. For example, software might use a global variable to record an external timer value (the FlexCAN timer can't be used, as that would unlock) when initiating a new Tx or unlocking an Rx, and then add the required delay before performing the second action.

Situation C can also be avoided by inserting a delay of at least two internal arbitration periods between writing 0xC and unlocking the locked Rx MB.

3.5 MUCts02605, MUCts03041 — ATD Predischarge Feature Non-functional

Description

The predischarge feature (ATDMODE.PRE = 1) is supposed to discharge the sampling capacitor before executing the actual conversion. The conversion flow in this case is:

1. Discharge the sample capacitor with VRL (discharge phase),
2. Charge the sample capacitor with the voltage on the channel specified by the command word (sample phase),
3. Convert the sampled voltage to a digital value (conversion phase).

Due to a problem in the predischarge feature the ATD will do this:

1. Sample VRL (sample phase),
2. Convert the sampled voltage to a digital value (conversion phase).

Therefore the result of a conversion using the predischarge feature will always show a conversion result for VRL of 0x000 (assuming right justified unsigned, 12 bit).

As the predischarge phase is skipped, the conversion length is reduced by two ATD clock cycles, which is the number of ATD clock cycles the predischarging would need.

Workaround

There is a workaround available for this problem:

1. Set ATDMODE.PRE = 1. The predischarge feature is enabled
2. Start a conversion. This will cause the sample capacitor to be predischarged (sample phase).
3. After the sample phase finished set ATDMODE.PRE = 0. This will terminate the current conversion and disable the predischarge feature.
4. Restart the same conversion that was started in step 2.

5. Wait for the end of the conversion

Steps 1 to 5 can be repeated.

Unfortunately, there is no indicator which indicates that the sample phase finished. This time must be measured, for example by using one of the timers in the PIT module.

3.6 MUCts03021, MUCts03540 — Changing DSPI CTARs Between Frames in Continuous PCS Mode Causes Error

Description

In continuous operation (CONT = 1) under some conditions the command word associated with the data frame is not always executed properly.

Erroneous data could be transmitted if multiple CTARs are used with different frame sizes while using the Continuous Peripheral Chip Select mode (DSPI*x*_PUSHR[CONT] = 1), if the Clock Phase is set to capture the DSPI data on the leading edge of SCK and change the data on the following edge (DSPIx_CTARn[CPHA] = 0) to create a virtual frame greater than 16 bits long.

For example, if an attempt is made to transmit a 12-bit frame and a 16-bit frame without negating PCS, two 12-bit frames are transferred. This has been observed in simulations where CPHA = 0. The two frames are transmitted correctly if CPHA = 1.

Workaround

When CPHA = 0 and continuous PCS mode is used, extended length (> 16 bits) frames may be created only by using two frames of equal size. This means that certain frame sizes cannot be constructed (prime numbers > 16).

3.7 MUCts03024 — Power-On-Reset Does Not Properly Reset the Flash

Description

The Flash reset sequence starts when POR is released. However, the 3.3 V supply is not yet at a sufficient level at that time for the Flash to operate correctly. Hence the Flash reset is not executed properly.

Workaround

It is possible to avoid the problem by supplying the device voltages externally (bypassing the internal VREG) and ramping up the voltages so that the 3.3 V level is stable before POR is deasserted.

An alternative is to issue an external reset after the POR sequence is enabled, which initializes the Flash again.

3.8 MUCts03027 — Flash Smart Erase Does Not Work Correctly

Description

The Flash can not be erased reliably; thus it is possible that a device can only be programmed once.

Workaround

There is no workaround available.

3.9 MUCts03071, MUCts03072 — eMIOS Comparators A and B Enabled by Writing to A2 and B2 in DAOC Mode

Description

When the eMIOS is in Double Action Output Compare (DAOC) mode, comparators A and B are enabled by the transfer of A2 to A1, and B2 to B1. However, writes to A2 and B2 before the transfers occur will also enable the comparators.

The main impact of this issue is that Output Update Disable (OU*n*) bit in the eMIOS Output Update Disable Register (eMIOS_OUDR) can not be used to cause both enables to occur at the same time. If the software sets OU*n* to disable the transfers and afterwards writes to A2/B2, the comparators will then be enabled. Then, if the timebase matches A1 or B1 (which have not yet been updated), the match will be recognized. The expected behavior is that the comparators should not be enabled until software clears OU*n* and the comparators are enabled simultaneously.

Workaround

When writing to the A2/B2 registers between the time that OU*n* is set and then cleared, software should ensure that the timebase does not match the old A1/B1 values.

3.10 MUCts03073, MUCts03074 — Writes to eMIOS MCR Register in IPM and IPWM Modes May Cause Incoherent Reads

Description

If a write to the eMIOS module control register (MCR) occurs between read A and read B in IPM and IPWM modes, the pair A, B may not be coherent. The write to MTSC is incorrectly re-enabling the B2 to B1 transfer expected to occur after read B.

Workaround

Proceed as usual for a coherent access: read A, the read B. Software should not write to the MCR in the middle of the coherent access in IPM and IPWM modes.

3.11 MUCts03075 — Flash Smart Erase Disabled

Description

The Smart erase feature has been disabled to avoid problems with programming and erasing of the prototypes.

Workaround

None.

3.12 MUCts03132 — eMIOS B Not Updated Correctly With 8-bit Accesses

Description

An 8-bit write to [15:8] of the unified channel register B does not update the register value.

Workaround

Use 32-bit accesses to write to B register.

3.13 MUCts03240 — Debug Mode “Convert Then Freeze” Fails

Description

The ATD may not be able to complete a conversion correctly after leaving Debug mode “convert then freeze”. This mode is entered when ATDMODE.DEBUG == 2'b10 is set and the MCU enters Debug mode.

The intended ATD behavior in this mode is as follows. The ATD should execute a conversion and freeze after completion. The ATD should then wait until either Debug mode is exited or a new conversion is started.

Instead, the conversion may not terminate or may yield an incorrect result.

Workaround

Two workarounds are possible:

1. Replace Debug mode “convert then freeze” with Debug mode “freeze immediately”.
   This workaround avoids the defective Debug mode. In this mode (ATDMODE.DEBUG == 2'b11) the conversion is frozen immediately when the MCU entered Debug mode. Please note that it is not possible to change the ATDMODE register without terminating the current conversion.
2. Use Debug mode “convert then freeze”, but restart the conversion after leaving Debug mode.
   This workaround requires a “conversion reset” after a conversion finished and the ATD entered Debug mode “convert then freeze”. This reset can be generated e.g. by writing ATDCW.CWCM with 2'b00. Afterwards the conversion sequence can be restarted by writing a new command word to ATDCW.

3.14 MUCts03347 — Expanded Mode Requires External Clock

Description

If the device is powered up in expanded mode, the oscillator mode is not latched correctly. Instead, the oscillator is always configured to expect an external clock. If that external clock is not present, the device will start up in self-clock mode.

Workaround

The following workarounds are possible:

1. Supply the device with an external clock.
2. Power the device up in single chip mode, then reset it into expanded mode; this will cause the oscillator mode to be latched correctly.
3. Use the device in single-chip mode.
4. Operate the device in self-clock mode.

3.15 MUCts03458 — Conversion Reset Implementation Does Not Comply with the Reference Manual

Description

The specified behavior for “conversion reset” is that a DMA request will be asserted only when the previously completed conversion was either “continuous conversion” or “wait for trigger”.

The actual behavior is that a conversion reset will always cause the DMA to request a new command. This causes a compatibility problem, as it does not match the behavior of the production masks.

Workaround

The application should avoid the command sequence: “convert then pause” followed by “conversion reset”.

After a “convert then pause” command, another valid conversion command (wait for trigger, continuous conversion, “convert then pause”) should be used to re-start DMA requests. The command “conversion reset” must be avoided for this purpose.

The command “conversion reset” should be used only to abort a running conversion.

3.16 MUCts03533, MUCts03534 — Large Blocks Limited to 10,000 Program/Erase Cycles

Description

The electrical specification for Program/Erase cycling on large Flash blocks (all 128K blocks - Middle Address Space [MAS] blocks M0 and M1, plus High Address Space [HAS] blocks H0 to H11) has been changed to 10,000 PE cycles minimum. The small blocks (16K, 48K, and 64K — Low Address Space [LAS] blocks L0-L5) are still specified as 100,000 PE cycles minimum.

The data retention specification all blocks is still 20 years for blocks cycled fewer than 1000 times and 5 years for blocks cycled 1001 to 100,000 cycles (10,000 for large blocks).

Workaround

Use only the small blocks for EEPROM emulation (LAS L0-L5). Do not use blocks MAS M0/M1 or HAS H0 to H11 for EEPROM emulation requiring greater than 10,000 Program/Erase cycles. Refer to the latest device electrical specifications (Data Sheet), dated June 2006 or later.

3.17 MUCts03562 — Minor Rev-ID not incremented

Description

For mask 1M84D the mask number was not incremented, so the 1M84D mask can not easily be distinguished from the 0M84D mask.

Workaround

The only way to readily distinguish the devices is to check the printed mask number on the package

3.18 PS4414 — Corrupt ID may be sent in early-SOF condition

Description

This erratum is not relevant in a typical CAN network, with oscillator tolerances inside the specified limits, because an early start of frame condition (early-SOF) should not occur.

An early-SOF may only be a problem if the oscillators in the network operate at opposite ends of the tolerance range (maximum 1.58%), which could lead to a cumulated phase error after 10 bit-times larger than phase segment 2.

A corrupt ID will be sent out if a transmit message buffer is identified for transmission during INTERMISSION, and an early-SOF condition is entered due to a dominant bit being sampled during bit 3 of INTERMISSION.

The message sent will be taken from the newly set up transmit buffer (Tx MB), with the exception of the 1st 8 ID bits, which are taken from the previously selected Tx MB.

The CRC is correctly calculated on the resulting bit stream so that receiving nodes will validate the message.

The early-SOF condition is detailed in the Bosch CAN Specification Version 2.0 Part B, Section 3.2.5 INTERFRAME SPACING - INTERMISSION.

Workaround

1. Configure Tx MBs during FREEZE mode, or
2. Out of FREEZE mode, configure Tx MBs during bus idle:
   — For networks with low traffic, determine Bus Idle status by reading the Idle bit of the Error and Status register (CANx_ESR[IDLE]).
   — For networks with high traffic, configure Tx MBs after the 3rd bit of intermission, and before the third bit of the CRC field from the next transmission.

3.19 MUCts03958 — Low pulse on LIN RX line may prevent assertion of transmit data ready flag TXRDY

Description

If the eSCI module is in LIN mode and receives a low pulse on the RX line while transmitting a frame header or a stop bit, the eSCI internal LIN master and slave tasks may be stopped, and the transmit data ready flag LINSTAT1[TXRDY] will never be asserted again.

Each of the following scenarios of the RX low pulse may provoke this erroneous behavior.

1. The low pulse begins less than 12 bits before the start of the break character, and ends at least 21 bits and at most 30 bits after the start of the break character

2. The low pulse begins at least 13 bits and at most 14 bits after the start of the break character, and ends at least 22 bits after the start of the break character.
3. The low pulse begins during the stop bit and has a duration of at least 2 bits.

Workaround

The application should use a timer, external to the eSCI module, that is started when a LIN frame transmission is started, with an expiration time programmed to the expected duration of the transmission. The timer is stopped when the eSCI module signals the end of the LIN frame transmission.

If this timer expires, the eSCI tasks have been stopped internally. In this case, the application should clear the SCICR2[TE] and SCICR2[RE] bits to disable the transmitter and receiver, and should set the LINCTRL1[LRES] bit, then reset the eSCI internal LIN slave and master tasks.

To resume LIN frame data transmission, the application should enable the transmitter and receiver, and clear the LINCTRL1[LRES] bit to enable the LIN slave and master tasks.

3.20 MUCts03989 — LIN fast bit error detection causes incorrect LIN reception

Description

When using the eSCI module in LIN mode, if the fast bit error detection is enabled in the SCI Control Register 3 (SCICR3[FBR]), and a bit distortion occurs on the RX line, the eSCI module may process the bit error signal incorrectly. This may cause unexpected transmission and reception behavior of the eSCI module.

Workaround

The application should disable the fast bit error detection by setting the SCICR3[FBR] bit to 0.

3.21 MUCts03990 — Automatic reset of LIN state machine causes incorrect transmission

Description

When the LIN FSM of the eSCI module performs an automatic reset due to an bus error condition, it can happen that the application is no longer synchronized to the LIN FSM. As a result, the eSCI considers payload data provided by the application via the LINTX register as LIN frame header data, and generates unexpected LIN frames.

Workaround

The application should disable the automatic LIN FSM reset functionality by setting LDBG bit in the LINCTRL1 register to 1.

3.22 MUCts03991 — LIN slave timeout flag STO not asserted if CRC is received too late

Description

The eSCI module will not assert the Slave-Timeout Flag STO in the LINSTAT1 register if the checksum field of on RX frame is received later than the time given in the LINTX[TO] field of the LIN RX frame control header. If the checksum field of on RX frame is not received at all, the STO flag will not be set, and the LIN FSM will wait forever.

Workaround

The application should use a timer, external to the eSCI module, that is started when a LIN RX frame transmission is started, with an expiration time programmed to the expected duration of the reception. The timer is stopped when the eSCI module signals the end of the LIN frame reception.

If this timer expires, the eSCI has encountered the slave timeout condition. In this case, the application should clear the SCICR2[TE] and SCICR2[RE] bits to disable the transmitter and receiver, and should set the LINCTRL1[LRES] bit, then reset the eSCI internal LIN slave and master tasks.

To resume LIN frame data transmission, the application should enable the transmitter and receiver, and clear the LINCTRL1[LRES] bit to enable the LIN slave and master tasks.

4 Revision History

Rev. No.	Date	Description of Changes
5	28 Sep 2006	Added MUCts03533. Added MUCts03540 (same as MUCts03021). Added revision history section.
6	6 Feb 2007	Changed “MAC72x2” to “MAC7200”. Rewrote introductory paragraph to reflect changes in device status and other documentation. Removed references to MAC7222 and MAC7252. Added 0M19G mask set. Added MUCts03534 (same as MUCts03533). Added MUCts03562. Added PS4414 Added PS4726
7	10 May 2007	Removed PS4726
8	20 Jun 2008	Added MUCts03958, MUCts03989, MUCts03990, MUCts03991

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

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

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

MAC7202MSE1
Rev. 8
06/2008