Active Errata List

- **Timer 2 (Baud Rate Generator Mode) – Long Start Time**
- **UART – RB8 Lost with JBC on SCON Register**
- **CAN – CANCONCH Harmless Corruption**
- **ADC – Interrupt During Idle Conversion**
- **Flash/EEPROM – First Read After Load Disturbed**
- **CAN – Sporadic Errors**
- **C51 Core – Bad Exit of Power-down in X2 Mode**
- **Timer0/1 – Extra interrupt**
- **Timer1 - Mode1 Does Not Generate Baud Rate Generator for UART**
- **Transmission after a 3 bit Intermessage**

Errata History

Lot Number	Errata List
A00470	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
A01013 and above	**5, 6, 7, 8, 9, 10, 11, 12, 13, 14**

Errata Description

1. EEPROM – EEPROM Data Cannot be Accessed from the Boot Area (FM1)

EEPROM Data cannot be read or written from the boot area (FM1).

Workaround

None.

2. XRAM – P3.7 and P3.6 Signals are Generated

When an overflow access is detected on the XRAM during a Movx instruction, negative pulses on the P3.7 and P3.6 signals are generated.

Workaround

None.

3. Double IT on External Falling Edge On INT1 or INT0 in X2 Mode

When CPU is in X2 Mode and Timer 1 or Timer 0 in X1 Mode (CKCON = 0x7F), IEx flag is not cleared by hardware after a service interrupt. In this case, the CPU executes the ISR a second time.

Workaround

The work around is to clear IEx bit during the Interrupt subroutine.

```
INT1_ISR :                    ; Interrupt sub routine
                  CLR IE1
            ....
```

4. Timer 2 (Baud Rate Generator Mode) – No IT When TF2 is Set by Software

When Timer 2 is used in baud rate generator mode, setting TF2 does not generate an interrupt.

Workaround

None.

CAN Microcontrollers

T89C51CC02
T89C51CC02UA
T89C51CC02CA

Errata Sheet

4160F–CAN–05/06

5. Timer 2 (Baud Rate Generator Mode) – Long Start Time

When Timer 2 is used as baud rate generator, TH2 is not loaded with RACP2H at the beginning, then UART is not operational before almost 10,000 machine cycles.

Workaround

In the software add an initialization of TH2 and TL2, with the value of RCAP2H and RCAP2L.

6. UART - RB8 Lost With JBC on SCON Register

When using the JBC instruction on any bit of SCON register, if RB8 changes from 1 to 0, the 0 bit can be lost.

Workaround

After each use clear RB8.

7. CAN – CANCONCH Harmless Corruption

When a stuff error occurs during a CAN frame transmission on DPRAM write access, the CONCH1, CONCH0 bits in CANCONCH are corrupted. This corruption has no effect on the correct behavior of the Transmit channel.

Workaround

No workaround required, re-writing CANCONCH to start a new message takes care of the corruption.

8. ADC - Interrupt Controller/ADC Idle Mode/Loops In High Priority Interrupt

The problem occurs during an A/D conversion in idle mode, if a hardware interrupt occurs followed by a second interrupt with higher priority before the end of the A/D conversion. If the above configuration occurs, the highest priority interrupt is served immediately after the A/D conversion. At the end of the highest priority interrupt service, the processor will not serve the hardware reset interrupt pending. It will also not serve any new interrupt requests with a priority lower than the high level priority last served.

Workaround

Disable all interrupts (Interrupt Global Enable Bit) before starting an A/D conversion in idle mode, then re-enable all interrupts immediately after.

9. Flash/EEPROM – First Read After Load Disturbed

In the "In-Application Programming" mode from the Flash, if the User software application load the Column Latch Area prior to call the programming sequence in the CAN Bootloader.

The "Read after load" issue leads to a wrong Opcode Fetch during the column latch load sequence.

Workaround

Update of the Flash API Library. A NOP instruction has to be inserted after the load instruction.

```
MOVX @DPTR,A ;Load Column latches
NOP ; ADDED INSTRUCTION
```

10. CAN – Sporadic Errors

When BRP = 0 or when BRP > 0 and SMP = 0, the CAN controller may de synchronize and send one error frame to ask for the retransmission of the incoming frame, even though it had no error.

This is likely to occur with BRP = 0 or after long inter frame periods without synchronization (low bus load). The CAN macro can still properly synchronize on frames following the error.

Workaround

Setting BRP greater than 0 in CANBT1 and SMP equals 1 in CANBT3 allows re-synchronization with the majority vote, and thus fixes the issue. The sampling point might have to be slightly advanced for the majority vote to take place within the bit. Therefore, at maximum speed of 1Mbit/s, the sampling point should be at less than 80% (e.g. 75%) for XTAL = 16MHz or less than 85% (e.g. 80%) for XTAL = 20 MHz.

11. C51 Core – Bad Exit of Power-down in X2 Mode

When exiting power-down mode by interrupt while CPU is in X2 mode, it leads to bad execution of the first instruction run when CPU restarts.

Workaround

Set the CPU in X1 mode directly before entering power-down mode.

12. Timer0/1 – Extra Interrupt

When the Timer0 is in X1 mode and Timer1 in X2 mode and vice versa, extra interrupt may randomly occurred for Timer0 or Timer1.

Workaround

Use the same mode for the two timers.

13. Transmission after a 3 bit Intermessage

If a Transmit Message Object (MOB) is enabled while the CAN bus is busy with an on going message, the transmitter will wait for the 3-bit Intermission before starting its transmission. This is in full agreement with the CAN recommendation.

If the transmitter lost arbitration against another node, two conditions can occur :

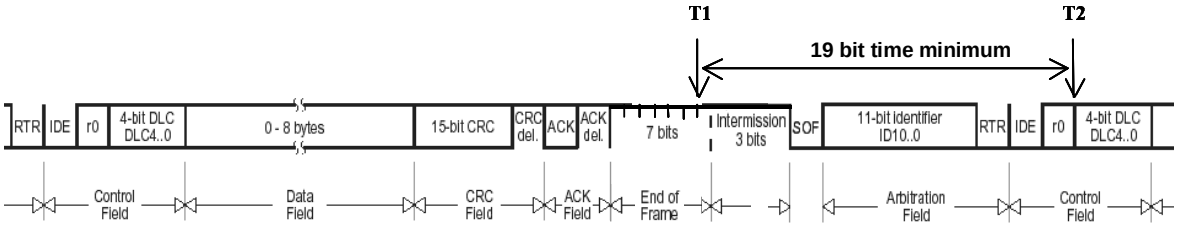
1. At least one Receive MOB of the chip are programmed to accept the incoming message. In this case, the transmitter will wait for the next 3-bit Intermission to retry its transmission.

2. No Receive MOB of the chip are programmed to accept the incoming message. In this case the transmitter will wait for a 4-bit Intermission to retry its transmission. In this case, any other CAN nodes ready to transmit after a 3-bit Intermission will start transmit before the chip transmitter, even if their messages have lower priority IDs.

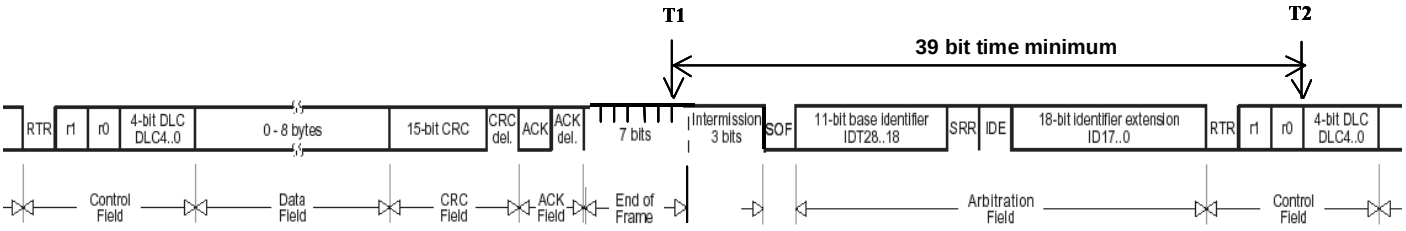
Workaround

Always have a Receive MOB enabled ready to accept any incoming messages. Thanks to the implementation of the CAN interface, a Receive MOB must be enable at latest, before the 1st bit of the DLC field. The Receive MOB status register is written (RXOK if message OK) immediately after the 6th-bit of the End Of Frame Field. This will leave in CAN2.0A mode a minimum 19-bit time delay to respond to the end of message interrupt (RXOK) and re-enable the Receive MOB before the start of the DLC field of the next incoming message. This minimum delay will be 39-bit time in CAN2.0B. See CAN2.0A CAN2.0B frame timings below.

CAN2.0A



CAN2.0B



Workaround implementation

The workaround is to have the last MOb (MOb3) as "spy" enabled all the time; it is the MOb of lowest priority : If a Mob other than MOb3 is programmed in receive mode and its acceptance filter matches with the incoming message ID, this Mob will take the message. MOb3 will only take messages than no other MObs will have accepted. MOb3 will need to be re-enabled fast enough to manage back to back frames. The deadline to do this is the beginning of DLC slot of incoming frames as explained above.

Minimum code to insert in CAN interrupt routine:

```
__interrupt void can_int_handler(void)
{
if ((CANSIT & 0x08) == 0x08 )          /* MOb3 interrupt  (SIT3=1) */
  {
  CANPAGE = (0x03);              /* select MOb3 */
  CANSTCH = 0x00;                /* reset MOb3 status */
  CANCONCH = 0x88;         /* reception enable */
  }
.........
.........
}
```

14. Timer1 – in Mode 1 Does Not Generate Baud Rate to UART

The timer1, when used as a baud rate generator in mode 1 (16 bits counter) for low baud rates, does not generate baud rate to UART.

Workaround

No.

Active UART Bootloader Errata List

- **Timer 2 and UART Are Not De-activated**
- **Watchdog and Flash API Starting the Bootloader Execution**
- **Autobaud False Start Bit Detection**
- **Flash API "__api_wr_code_page" with 0 Data in Length Parameter Field**

UART Bootloader Errata History

Version Number	Errata List
1.2	1, 2, 3, 4

UART Bootloader Errata Description

1. Timer 2 and UART Are Not De-activated

When the bootloader receives the command "Start Application" (LJMP 0), the Timer 2 and the UART are not de-activated.

Workaround

The application must have in its setup function a reset of Timer 2 and UART.

```
mov SCON, #00h
mov T2CON, #00h
mov RCAP2L, #00h
mov RCAP2H, #00h
mov TL2, #00h
mov TH2, #00h
```

2. Watchdog and Flash API Starting the Bootloader Execution

When an application call "__api_start_bootloader" or"__api_start_isp" routines while the watchdog is enabled, when the watchdog overflow it will restart the application instead of the bootloader

Workaround

Set BLJB(=1) before calling the __api_start_bootloader or __api_start_isp if the watchdog is used.

3. Autobaud False Start Bit Detection

UART autobaud sequence does not work on some special UARTs.

Some laptops have the UART TX line set to 0 when unused (COM port closed), this results in a false baud rate calculation on the 'U' character.

The autobaud sequence checkes for a '0' state (not a falling edge) on the Rx line of the UART microcontroller to detect the 'start' bit of the 'U' synchro character.

As this line is '0' by default when COM port is closed, the autobaud routine starts its baudrate calculation at the opening sequence of the UART.

Workaround

A 'Special Sync' can be used with 'FLIP' software.

In this case, the open port event and the 'U' sent are dissociated. The user must first open his COM port with the 'connect' button, then reset its hardware and finally push the 'sync' button.

4. Flash API "__api_wr_code_page" with 0 Data in Length Parameter Field

When the flash api "__api_wr_code_page" is called with the field nb_data equal 0 then 255 data are wrote in flash.

Workaround

Include a test on nb_data before executed __api_wr_code_page routine.

Active CAN Bootloader Errata List

- **The CAN is Not De-activated**
- **Watchdog and Flash API Starting the Bootloader Execution**
- **Flash API "__api_wr_code_page" with 0 Data in Length Parameter Field**
- **Problem to program a hex file less than 16 bytes**
- **Unexpected echo after start application command**
- **Start application with security set**

CAN Bootloader Errata History

Version Number	Errata List
1.0.2	1, 2, 3, 4, 5, 6

CAN Bootloader Errata Description

1. The CAN is Not De-activated

When the bootloader receives the command "Start Application" (LJMP 0), the CAN is not de-activated.

Workaround

The application must have in its setup function a reset of CAN macro.

```
mov CANGCON, #00h
```

2. Watchdog and Flash API Starting the Bootloader Execution

When an application call "__api_start_bootloader" or"__api_start_isp" routines while the watchdog is enabled, when the watchdog overflow it will restart the application instead of the bootloader

Workaround

Set BLJB(=1) before calling the __api_start_bootloader or __api_start_isp if the watchdog is used.

3. Flash API "__api_wr_code_page" with 0 Data in Length Parameter Field

When the flash api "__api_wr_code_page" is called with the field nb_data equal 0 then 255 data are wrote in flash.

Workaround

Include a test on nb_data before executed __api_wr_code_page routine.

4. Problem to Program a Hex File Less than 16 Bytes

When we try to program a hex file with a size size less than 16 bytes, some troubles appear depending of the start address.

Workaround

Program with a range address higher than 16 bytes.

5. Unexpected echo after start application command

The CAN bootloader returns an unexpected echo after a start application command (with reset).

Workaround

Don't manage this return echo.

6. Start application with software security set

The start application with or without reset doesn't work if the software security are set.

Workaround

Use the SSB API to secure the device in the application.

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan

9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

Printed on recycled paper.

4160F–CAN–05/06