ST10F276E, ST10F275E, ST10F273E
Errata sheet

16-bit MCUs with
512/768/832 KByte Flash and 36/36/68 KByte RAM memories

Introduction

This errata sheet describes all the functional and electrical problems known in the Cxx silicon versions of the ST10F276E, ST10F275E and ST10F273E.

Note: *The ST10F275E and ST10F273E are commercial products based on the same silicon as the ST10F276E. Therefore the present document is valid for all the three devices.*

The major revision of the device can be read in the IDCHIP register (@F07Ch) which is set to 1143h for the Cxx versions.

1 Functional problems' summary

1.1 Functional's problem summary for the CAA version

Table 1. Functional problems of the ST10F276E, F275E and F273E CAA version

Functional problem	Short description
ADC.1	ADC accuracy degradation
ADC.2	Injected conversion stalling the ADC
BUS.9	Spurious BREQ pulse in slave mode during external bus arbitration phase
C-CAN.1	Concurrent transmission requests in DAR-mode
C-CAN.2	Disabling of transmission request
FLASH.1	Read While Write not supported
FLASH.4	Flash access time
FLASH.6	P2.0 activity and Flash operations
FLASH.7	XFlash access and wait-state
PORT.1	Bit protection not implemented on the CAPCOM IOs of ports P2, P7 and P8
PWRDN.1	Execution of PWRDN Instruction
PWRDN.2	High current consumption in Power Down Mode
RESET.1	Software and Watchdog reset malfunction
XASC.1	XASC Receive Overrun Error Flag not set
XBUS.3	Corruption of XPWM register on read access
XPWM.1	Consecutive write accesses to the XPWM module not performed
XSSC.1	XSSC Receive Error Flag not set

1.2 Functional's problem summary for the CEA version

Table 2. Functional problems of the ST10F276E, F275E and F273E CEA version

Functional problem	Short description
ADC.2	Injected conversion stalling the ADC
BUS.9	Spurious BREQ pulse in slave mode during external bus arbitration phase
C-CAN.1	Concurrent transmission requests in DAR-mode
C-CAN.2	Disabling of transmission request
FLASH.1	Read While Write not supported
PWRDN.1	Execution of PWRDN Instruction
XASC.1	XASC Receive Overrun Error Flag not set
XSSC.1	XSSC Receive Error Flag not set

1.3 Functional's problem summary for the CEG version

Table 3. Functional problems of the ST10F276E, F275E and F273E CEG version

Functional problem	Short description
ADC.2	Injected conversion stalling the ADC
BUS.9	Spurious BREQ pulse in slave mode during external bus arbitration phase
C-CAN.1	Concurrent transmission requests in DAR-mode
C-CAN.2	Disabling of transmission request
FLASH.1	Read While Write not supported
PWRDN.1	Execution of PWRDN Instruction

2 Functional problems description

2.1 ADC.1 - ADC accuracy degradation

On specific analog input values an accuracy degradation of several LSBs may be seen on the digital output value of the implemented ADC. In particular this occurs if the input voltage matches the internal comparator thresholds, which is when a roll over from the least significant bits (i.e. 0x1Fh -> 0x20h or vice versa) occurs.

The maximum deviation in accuracy is ± 250mV over the entire conversion range.

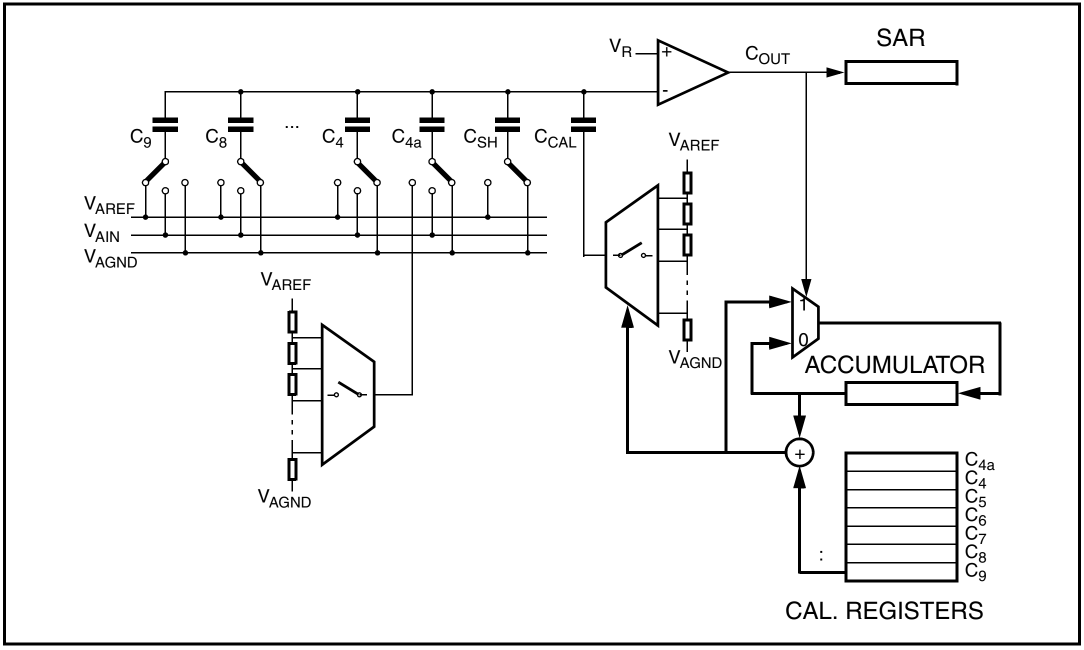
Additional information:

The failure is linked to the implemented “self calibrating circuitry” which is summarized in the *Figure 1: Calibration mechanism (during bit 9 comparison phase)*.

The maximum correction applied by the calibration is 250mV.

When the analog input voltage is close to one of the thresholds of the comparator (by less than 1mV), then the amplifier sees a differential voltage very close to zero. As a result, the output of the comparator will be fluctuating between the values “0” or “1”. Because of the parallel use of the comparator output and of the fluctuating value, different values may be latched in the SAR (Successive Approximation Register) and the Calibration section. As a result a bad calibration mechanism is applied to the successive bit

Figure 1. Calibration mechanism (during bit 9 comparison phase)



Application hints - containment strategy:

Measurements on different pieces and over several billions conversions have been made. The results are showing an error rate of 1/600,000.

Workaround:

It is not possible to define a workaround generic for all applications.

2.2 ADC.2 - Injected conversion stalling the ADC

Description

Whenever a new Injection request is issued before the ADDAT2 register has been read by the CPU (i.e. the result of the previous injection request was not read), the ADC is stalled and no further conversions are performed.

Workarounds - Recovery actions

The following actions allows to unlock the ADC module:

1. Read the ADDAT2 register twice at the end of every injected conversion (also prevents stall condition to occur)
2. Disable then re-enable the Wait For Read mode

Application conditions

In an application, all the following conditions are needed to reach this state:

- Injection requests are hardware triggered (via CapCom CC31)
- The result of injected conversion are read via a PEC transfer (prevents from reading twice ADDAT2 by software)
- A high level task is disabling the PEC transfer for a long time (2 analog conversion + time between 2 injection requests)

Therefore ensuring, at application level, that no task can disable interrupts for a time such than 2 injection requests can be issued before a read operation is performed prevents the locking situation to occur.

Detailed analysis

Channel Injection Mode allows the conversion of a specific analog channel (also while the ADC is running in a continuous or auto scan mode) without changing the current operating mode.
The following main points need to be highlighted:

- Wait for ADDAT Read Mode is needed in order for the ADC Channel Injection mode to properly operate
- At the end of the Injected conversion the data will be available in the alternate result register ADDAT2 and a Channel Injection Complete Interrupt request will be generated (ADEIR Flag)
- If the temporary data register used for ADDAT2 Read Mode is full, the respective next conversion (standard or Injected) will be suspended. The temporary register can hold data for ADDAT (from a standard Conversion) or for ADDAT2 (from an injected conversion)

If the temporary data register used for ADDAT2 Read Mode is full and a new Injection request is issued then the new converted value is stored into a temporary data register until the previous one is read from ADDAT2.

For a correct functionality as soon as ADDAT2 register is read then the last converted value should be moved from temporary register to ADDAT2 and the ADEINT interrupt request should be issued. This would allow the CPU to read the last converted value. See *Figure 2*.

In the real situation as soon as ADDAT2 register is read then the last converted value is correctly moved from temporary register to ADDAT2 but **the ADEINT interrupt request is not received by the Interrupt Controller (see *Figure 3*).** As a consequence the CPU/PEC will never know that a new converted value is ready to be read in ADDAT2 register and therefore at the following injection request the ADC will fill the temporary register again (without generating any ADEINT interrupt request) and then will stall the ADC for any further conversion. The ADC will stay in the “wait for read ADDAT2 register” condition forever.

Figure 2. ADC injection theoretical operation

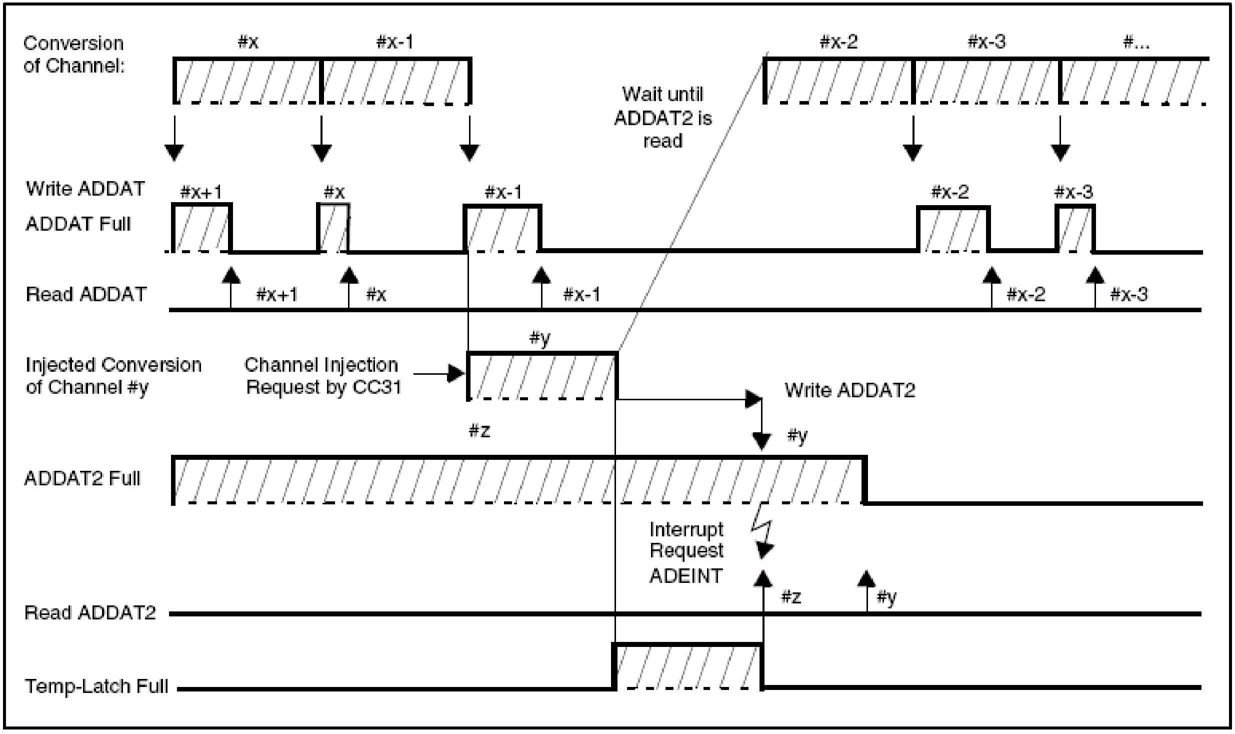
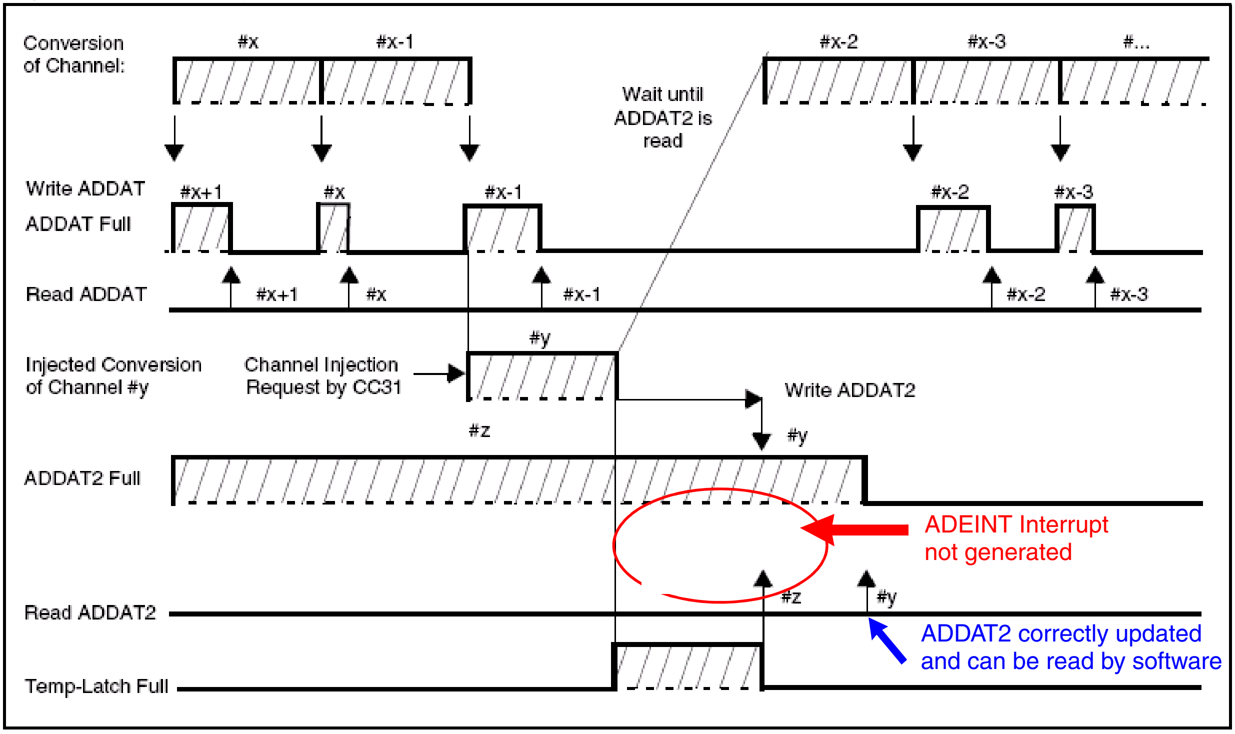


Figure 3. ADC injection actual operation


Conversion of Channel:
#x
#x-1
Wait until ADDAT2 is read
#x-2
#x-3
#...
Write ADDAT
ADDAT Full
#x+1
#x
#x-1
#x-2
#x-3
Read ADDAT
#x+1
#x
#x-1
#x-2
#x-3
#y
Injected Conversion of Channel #y
Channel Injection Request by CC31
Write ADDAT2
#z
#y
ADDAT2 Full
ADEINT Interrupt not generated
#z
#y
Read ADDAT2
ADDAT2 correctly updated and can be read by software
Temp-Latch Full

2.3 BUS.9 - Spurious BREQ pulse in slave mode during external bus arbitration phase

When using external bus arbitration via HOLD-function with the ST10F276 configured as a slave, sporadic bus errors may occurs.

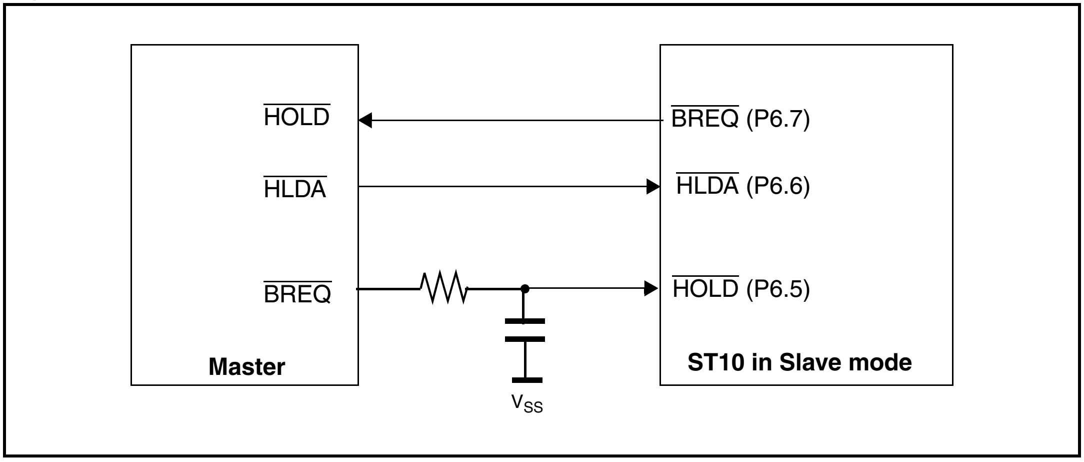
After the slave has been granted the bus, the slave deactivates BREQ sporadically for a short time, even though the bus access of the slave has not been completed. The master then starts its own bus access, leading to a bus conflict between master and slave.

Workaround:

In order not to produce any spurious BREQ pulse during a Slave External Bus Arbitration Phase, it is necessary to guarantee that the distance between the HOLDA assertion (Bus Acknowledge from Master device) and the following HOLD falling edge (Bus Request from Master) is greater than three clock cycles.

This can be implemented by delaying the HOLD signal with a RC circuit as shown in the figure below.

Figure 4. ST10 in slave mode



2.4 C-CAN.1 - Concurrent transmission requests in DAR-mode

Description:

When the C_CAN is configured to work in DAR-mode (Disable Automatic Retransmission) and the host requests the transmission of several messages at the same time, only two of these messages will be transmitted. For all other requested transmit messages, the TxRqst bits will be reset, but no transmission will be started, NewDat and IntPnd will be left unchanged. For the two messages that are transmitted, the TxRqst and NewDat bits will be reset and, if enabled by TxIE, IntPnd will be set.

The normal operation mode (DAR = 0) is not affected by this phenomenon.

Workaround:

The DAR-mode is intended to support time triggered operation (TTCAN level 1), where a message may be transmitted only in its dedicated time window. It would be an error to request the transmission of several messages at the same time. So no workaround is necessary for TTCAN applications.

The progress of a requested transmission can be monitored by checking the message object's TxRqst and NewDat [optionally IntPnd] bits. In DAR-Mode, a message that was requested to be transmitted, but that was not transmitted either because the transmission was disturbed or because the transmission was not started needs to be requested again.

2.5 C-CAN.2 - Disabling of the transmission request

Description:

When the host disables the pending transmission request of the message object with the lowest priority (number 32 in the default implementation) in the short time window where the message handler state machine has prepared the transmission of the message, but before the transmission has actually started, then the transmission request of this message object may remain stuck at disabled even if the host immediately re-enables it. Reading the transmission request bit of this message object will not show that it is stuck at disabled. This can only happen when this message object is the only one with a pending transmission request.

If the transmission request is stuck at disabled, it will be re-enabled by the first activity detected on the CAN bus or by setting the transmission request of any other message object.

The other message objects are not affected by this phenomenon.

Workaround:

Generally, it is not necessary to disable the transmission request of a message object. If the message object is to be used for another message, it is sufficient to prepare the new content for this message object in the CPU interface register (Identifier, DLC, Data, with TxRqst and NewDat [optionally TxIE] bits) and to transfer this content into the message object. The new content will be transmitted at the next opportunity, not influenced by a possibly ongoing transmission of the previous content of the same message object.

2.6 FLASH.1 - Read while Write not supported

The read while write functionality is not supported with this silicon version: a write operation in one of the four banks prevents subsequent reading from the Flash (whatever XFlash or IFlash) till it is completed. The reason of this limitation has been found and will be corrected in a future silicon version.

Workaround:

Flash programming / reprogramming routines must be executed from RAM.

2.7 FLASH.4 - Flash access time

Preliminary characterization data are showing limitations on Flash access time for CPU clock frequency above 40MHz. This can result in erroneous data read from the IFlash or the XFlash for operand accesses or in illegal opcode for opcode accesses.

Workaround:

To have functional parts in the whole -40 / 125 degree Celsius range, limit the CPU clock frequency to 40MHz.

There is no workaround for higher CPU clock frequencies.

2.8 FLASH.6 - P2.0 activity and Flash operations

Activity on pin P2.0 could cause failures during the Flash fetch, read or erase operations. The source of the activity can be external hardware or internal pin activity. This applies when P2.0 is:

- configured in input and connected to a dynamic signal,
- configured in output compare mode or software controlled output.

Workaround:

For fetch and read operations:

- When P2.0 is configured as input, connect it to ground.
- When P2.0 is configured as output:
  - for CPU frequency up to 40MHz, the minimum time between two transitions of the signal must be 90 us;
  - for CPU frequency higher than 40MHz, use P2.0 in static way.

For erase operations: use the pin P2.0 in a static way (input or output).

2.9 FLASH.7 - XFlash access and wait-state

Preliminary characterization data are showing limitations on XFlash access time for CPU clock frequency above 35MHz with 0 wait-state (see register XFICR programming).

Workaround:

For CPU frequencies above 35MHz, configure XFICR to have 1 wait-state.

Note that the default value of the XFICR after reset is such that 15 wait-states are used to access the XFlash.

2.10 PORT.1 - Bit protection not implemented on the CAPCOM IOs of ports P2, P7 and P8

The ST10F276-5-3E provides protected bits. These bits can be modified by both the on-chip hardware and the software. This is the case of the Port registers when Alternate functions are in use. The protection ensures that these bits are not modified when the software accesses to other bits of the register.

The Capture Compare Unit (simply CAPCOM) can make a port pin toggle automatically via compare match. If the toggling event occurs during a software modify or write-back operation using any bit manipulation instruction (BSET/CLR, BFLDH/L, BAND,...), performed on another bit of the same port (same data register Px), then the toggling event is overwritten by the write-back.

These bits are the ones related to Port Data registers where the CAPCOM output alternate functions are mapped, see *Table 4*.

Table 4. List of the bits affected by the PORT.1 problem

Port register	Affected bit	Alternate function
Port2	P2.0 to P2.15	CC0IO to CC15IO
Port7	P7.4 to P7.7	CC28IO to CC32IO
Port8	P8.0 to P8.7	CC16IO to CC23IO

Workaround 1: Application mapping

Map the application IOs in order not to mix outputs controlled by software on one of those ports when a Compare output is used.

Workaround 2: Software toggling of the Compare output

Instead of using the hardware toggling of the bit, generate an interrupt at each Compare output match. Then inside the interrupt routine a safe port bit manipulation can be performed.

2.11 PWRDN.1 - Execution of PWRDN instruction

When instruction PWRDN is executed while pin $\overline{\text{NMI}}$ is at a high level (if PWDCFG bit is cleared in SYSCON register) or while at least one of the port P2 pins used to exit from Power Down mode (if PWDCFG bit is set in SYSCON register) is at the active level, Power Down mode is not entered, and the PWRDN instruction is ignored.

However, under the conditions described below, the PWRDN instruction is not ignored, and no further instructions are fetched from external memory, i.e. the CPU is in a quasi-idle state.

This problem only occurs in the following situations:

a) The instructions following the PWRDN instruction are located in an external memory, and a **multiplexed bus** configuration **with memory tristate waitstate** (bit MTTCx = 0) is used.

Or

b) The instruction preceding the PWRDN instruction **writes** to external memory or an XPeripheral (XRAM, CAN, etc.), and the instructions following the PWRDN instruction are located in external memory. In this case, the problem occurs for any bus configuration.

Note: The on-chip peripherals are still working correctly, in particular the Watchdog Timer, if not disabled, resets the device upon an overflow. Interrupts and PEC transfers, however, cannot be processed. In case $\overline{\text{NMI}}$ is asserted low while the device is in this quasi-idle state, Power Down mode is entered.

No problem occurs if the $\overline{\text{NMI}}$ pin is low (if PWDCFG = 0) or if all P2 pins used to exit from Power Down mode are at inactive level (if PWDCFG = 1): the chip normally enters Power Down mode.

Workaround:

Ensure that no instruction that writes to external memory or to an XPeripheral precedes the PWRDN instruction, otherwise insert a NOP instruction in front of PWRDN. When a multiplexed bus with memory tristate wait state is used, the PWRDN instruction must be executed from internal RAM or XRAM.

2.12 PWRDN.2 - High current consumption in Power Down mode

When the PWRDN instruction is executed from the Flash (whatever XFlash or IFlash), the Flash is not correctly switched off, leading to a current consumption above the specifications. Moreover, as the XRAM2 and the XFlash are sharing the same XBus chip select, the same behavior occurs when executing the instruction from XRAM2.

Workaround:

Execute the PWRDN instruction from IRAM or XRAM1.

2.13 RESET.1 - Software and watchdog reset malfunction

The Software reset instruction (SRST) and the Watchdog reset may not be correctly recognized by the Flash controller. As a consequence the Flash controller maintains the ST10 under reset leading to a dead-lock situation.

Workaround:

If the SRST instruction or the watchdog reset are used in the application then the SYSCON register must be configured to activate the bidirectional reset feature and allow the Flash Controller to detect a low level on the $\overline{\text{RSTIN}}$ pin.

Provided that the RPD pin level will remain high for the whole reset duration, the reset will not turn into a hardware reset and the flags in the WDTCON register will be set as before.

To achieve a correct behavior the following constraints must be considered:

1. The circuitry on $\overline{\text{RSTIN}}$ pin must be compatible with the use of the bidirectional reset.
2. $\overline{\text{RSTIN}}$ discharge time must be shorter than the internal Reset duration of 512 CPU clock cycles.
3. RPD pin discharge time must be longer than the time the $\overline{\text{RSTIN}}$ pin is seen at a low level by the ST10F276.
4. $\overline{\text{RSTIN}}$ pulse (low level active) duration must be longer than the PLL Lock Time, in case an unlock occurred.

Application example:

The following values for RC circuitries on the RPD and $\overline{\text{RSTIN}}$ pins have been simulated and are matching the previous constraints.

RPD RC circuit:$\overline{\text{RSTIN}}$ RC circuit:

R = 215 kohm	R = 215 kohm (pull-up to V_{DD})
C = 100 nF	C = 10 nF (between $\overline{\text{RSTIN}}$ and V_{SS})

Impact on the Alternate Bootstrap Mode:

The Alternate Boot Mode is affected by this functional problem. When the User key is correctly programmed into the Flash, a Software Reset instruction is executed in order to restart the ST10F276 in normal mode and to run the application software.

2.14 XASC.1 - XASC receive overrun error flag not set

In the ST10's XASC, data reception is double buffered, so that reception of a second character may begin before the previously received data has been read out of the receive buffer register (XS1RBUF). The overrun error flag (S1OE, bit 10 of XS1CON register) and an error interrupt request flag will be set when the receive buffer has not been read out before the completion of the second reception.

In the condition where the XS1RBUF register content is read out at the completion of the second reception, the XS1RBUF register content is updated and the new data is read out without the flags being set.

Workaround:

The real time behavior of the application must be checked. The polling rate of the XASC Receive Interrupt Request flag or the priority of the XASC Receive Interrupt must be increased in order to ensure the reading of the XS1RBUF before the end of the next transfer.

2.15 XBUS.3 - Corruption of XPWM register on read access

If, fetching from IFlash, a write to XRAM is performed and followed immediately (two MOV instructions in sequence) by a read from a XPWM register, then the XPWM register is corrupted (spurious write) with the content of the data previously written in XRAM.

Workaround:

Insert a NOP instruction before all read accesses to the XPWM module.

2.16 XPWM.1 - Consecutive write accesses to the XPWM module

When both following conditions are met:

a) code is executed from IFlash

AND

b) two (or more) consecutive write operations are executed with XPWM registers as target

Then only the first write operation is performed: the others are not performed.

Workaround:

Add four NOP instructions between two write accesses to XPWM registers.

2.17 XSSC.1 - XSSC receive error flag not set

In the following conditions:

- 2 data (Data_A and Data_B) are consecutively received on the XSSC
- the XSSC Receive Buffer (XSSCRB) is read (by the CPU) at the end of the reception of the second data

Then the XSSCRB register content is updated with Data_B by the XSSC module. The Data_A is lost and the Receive Error Flag is not set in the XSSCCON register. As the reception of the Data_B sets the XSSC Receive Interrupt Flag, the CPU will read Data_B as the second data received.

Therefore Data_B will be read twice with no error flag.

Additional information:

The item is valid in both master and slave modes.

This is impacting the application when transmissions are made consecutively without stopping the clock as described below.

To initiate the clock, a dummy transmission is made by the master: i.e. the application code writes to the XSSCTB (transmit buffer) and a clock is generated, data is received along with this clock.

When several data are to be received, generally the application uses the XSSC Transmit Interrupt Request flag. As soon as the XSSCTB has been transferred to the shift register, the XSSC Transmit Interrupt Request flag is set to indicate that the XSSCTB can be reloaded again. This gives the fastest transmission as a continuous clock is provided to the slave. The application checks the XSSC Receive Interrupt Request flag to read the received value and a new reception is on going during the effective reading of XSSCRB register.

Workaround:

When the ST10's XSSC is in master mode, do not use continuous transfer sequences. Instead wait for the complete reception of each data, using the XSSC Receive Interrupt Request flag before starting a new reception.

When the ST10's XSSC is in slave mode (or in master mode), the real time behavior of the application must be checked. The polling rate of the XSSC Receive Interrupt Request flag or the priority of the XSSC Interrupt must be increased in order to ensure the reading of the XSSCRB before the end of the next transfer.

3 Documentation update - Modifications of features

Previous references:	ST10F276 Preliminary Data, Revision 1.2, January 2005
	ST10F276 User Manual, Release 1.6, January 2005
New references:	ST10F276E Data sheet, Revision 1, May 2006
	ST10F273E Data sheet, Revision 1, May 2006
	ST10F276 User Manual, Release 1, March 2007

3.1 Flash commands timings updated

Characterization and validation results were showing that the typical programming and erasing times for the Flash are longer than the original specified values. This point was formerly described as FLASH.3 functional problem.

Specification change:

Timings for erase and programing are now fully specified in the devices' data sheets.

3.2 Main Voltage regulator must be OFF in Power Down

Characterization and validation results show that the main voltage regulator must be switched off during Power Down to meet power consumption specification and avoid possible CPU wake-up. This point was formerly described as PWRDN.3 functional problem.

Specification change:

The main voltage regulator must be turned off in Power Down by setting bit VREGOFF in XMISC register (bit XMISC.3). Setting this bit will automatically turn-off the main voltage regulator after the Power Down instruction is executed and will turn it up again when leaving Power Down mode.

3.3 DC and AC parameters modifications

The following parameters were formerly highlighted as out of the target specifications. They have been re-specified in the revision 1.2 of the Preliminary Data.

V_{HYS} (Input Hysteresis in TTL threshold): this parameter is re-specified to 400mV instead of 500mV.

I_{SB1} (Stand-by mode current with RTC off, Oscillator off and V_{DD} off): for temperature within 105 / 125 degree Celsius this parameter is re-specified to 500uA instead of 200uA.

VCO frequency range: the VCO frequency range have been re-specified to 64-128MHz instead of 40-128MHz.

4 Deviations from DC/AC specification

DC Parameters

The devices are in line with the specifications.

AC Timings

One limitation have been found:

- **T_{LOCK}**: PLL Lock-in time for x10 multiplication factor is 300us instead of 250us.

5 Revision history

Table 5. Revision history

Date	Revision	Description of changes
March 2004	1	First Issue.
September 2004	2	Added FLASH.7 and I_{OV2} items. Modified FLASH.1 and FLASH.6 items. Added AC deviation on parameter T_{LOCK} in section Deviations from DC/AC specifications. Applied latest “Document Format Standards”.
September 2004	3	Added XSSC.1 and XASC.1 items.
October 2004	4	Added ADC.1 item. Added information for the CDA silicon version. Created Section 4 - Documentation Update. Moved items FLASH.3 and PWRDN.3 to Section 4.
February 2005	5	Updated to include CEA and CEG versions. Updated Section 4: reference documents have new versions. Updated revision history table for Revision 4 to highlight Section 4 creation.
July 2005	6	Added Section 5: DC/AC deviations with T_{LOCK} limitation.
05-July-2006	7	Added item *BUS.9 - Spurious BREQ pulse in slave mode during external bus arbitration phase* Added support of the ST10F275E and ST10F273E devices. Updated *Chapter 4: Deviations from DC/AC specification* according to final specification
04-Sept-2007	8	Added items: – *ADC.2 - Injected conversion stalling the ADC* – *C-CAN.1 - Concurrent transmission requests in DAR-mode* – *C-CAN.2 - Disabling of the transmission request* Re-introduce item *FLASH.1 - Read while Write not supported* for all devices

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -
Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com