# STM32F10xx8 and STM32F10xxB Errata sheet

## STM32F101x8/B, STM32F102x8/B and STM32F103x8/B medium-density device limitations

## Silicon identification

This errata sheet applies to the revisions B, Z and Y of the STMicroelectronics medium-density STM32F101xx access line and STM32F103xx performance line products, and to revision Y of the STM32F102xx USB access line devices.
These families feature an ARM™ 32-bit Cortex®-M3 core, for which an errata notice is also available (see *Section 1* for details).

The full list of root part numbers is shown in *Table 2*.

The products are identifiable as shown in *Table 1*:

- by the Revision code marked below the Sales Type on the device package
- by the last three digits of the Internal Sales Type printed on the box label

**Table 1. Device identification(1)**

| Sales type | Revision code(2) marked on device |
|---|---|
| STM32F101xxx(3) | “B”, “Z” or “Y” |
| STM32F102xxx(3) | “Y” |
| STM32F103xxx(3) | “B”, “Z” or “Y” |

1. The REV_ID bits in the DBGMCU_IDCODE register show the revision code of the device (see the STM32F10xxx reference manual for details on how to find the revision code).
2. Refer to *Appendix A: Revision code on device marking* for details on how to identify the Revision code on the different packages.
3. Are also concerned all devices with 32 KB of Flash memory that do not have the letter A in their sales type.

**Table 2. Device summary**

| Reference | Part number |
|---|---|
| STM32F101xx | STM32F101C8, STM32F101R8 STM32F101V8, STM32F101T8 |
| | STM32F101RB, STM32F101VB, STM32F101CB |
| STM32F102xx | STM32F102C8, STM32F102R8 |
| | STM32F102CB, STM32F102RB |
| STM32F103xx | STM32F103C8, STM32F103R8 STM32F103V8, STM32F103T8 |
| | STM32F103RB STM32F103VB, STM32F103CB |

# 1 ARM™ 32-bit Cortex®-M3 limitations

An errata notice of the STM32F10xxx core is available from the following web address: *http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.eat0420a/*.
The direct link to the errata notice pdf is:
*http://infocenter.arm.com/help/topic/com.arm.doc.eat0420a/Cortex-M3-Errata-r1p1-v0.2.pdf*.
All the described limitations are minor and related to the revision r1p1-01rel0 of the Cortex-M3 core. *Table 3* summarizes these limitations and their implications on the behavior of medium-density STM32F10xxx devices.

**Table 3. Cortex-M3 core limitations and impact on microcontroller behavior**

| ARM ID | ARM category | ARM summary of errata | Impact on medium-density STM32F10xxx devices |
|---|---|---|---|
| 602117 | Cat 2 | LDRD with base in list may result in incorrect base register when interrupted or faulted | Minor |
| 563915 | Cat 2 | Event register is not set by interrupts and debug | Minor |
| 531064 | impl | SWJ-DP missing POR reset sync | No |
| 511864 | Cat 3 | Cortex-M3 may fetch instructions using incorrect privilege on return from an exception | No |
| 532314 | Cat 3 | DWT CPI counter increments during sleep | No |
| 538714 | Cat 3 | Cortex-M3 TPIU clock domain crossing | No |
| 548721 | Cat 3 | Internal write buffer could be active whilst asleep | No |
| 463763 | Cat 3 | BKPT in debug monitor mode can cause DFSR mismatch | Minor |
| 463764 | Cat 3 | Core may freeze for SLEEPONEXIT single instruction ISR | Minor |
| 463769 | Cat 3 | Unaligned MPU fault during a write may cause the wrong data to be written to a successful first access | No |

## 1.1 Cortex-M3 limitations description for STM32F10xxx medium-density devices

Only the limitations described below have an impact, even though minor, on the implementation of STM32F10xxx medium-density devices.

All the other limitations described in the ARM errata notice (and summarized in *Table 3* above) have no impact and are not related to the implementation of STM32F10xxx medium-density devices (Cortex-M3 r1p1-01rel0).

### 1.1.1 Cortex-M3 LDRD with base in list may result in incorrect base register when interrupted or faulted

#### Description

The Cortex-M3 Core has a limitation when executing an LDRD instruction from the system-bus area, with the base register in a list of the form LDRD Ra, Rb, [Ra, #imm]. The execution may not complete after loading the first destination register due to an interrupt before the second loading completes or due to the second loading getting a bus fault.

#### Workarounds

1. This limitation does not impact the STM32F10xxx code execution when executing from the embedded Flash memory, which is the standard use of the microcontroller.
2. Use the latest compiler releases. As of today, they no longer generate this particular sequence. Moreover, a scanning tool is provided to detect this sequence on previous releases (refer to your preferred compiler provider).

### 1.1.2 Cortex-M3 event register is not set by interrupts and debug

#### Description

When interrupts related to a WFE occur before the WFE is executed, the event register used for WFE wakeup events is not set and the event is missed. Therefore, when the WFE is executed, the core does not wake up from WFE if no other event or interrupt occur.

#### Workaround

Use STM32F10xxx external events instead of interrupts to wake up the core from WFE by configuring an external or internal EXTI line in event mode.

### 1.1.3 Cortex-M3 BKPT in debug monitor mode can cause DFSR mismatch

#### Description

A BKPT may be executed in debug monitor mode. This causes the debug monitor handler to be run. However, the bit 1 in the Debug fault status register (DFSR) at address 0xE000ED30 is not set to indicate that it was originated by a BKPT instruction. This only occurs if an interrupt other than the debug monitor is already being processed just before the BKPT is executed.

#### Workaround

If the DFSR register does not have any bit set when the debug monitor is entered, this means that we must be in this “corner case” and so, that a BKPT instruction was executed in debug monitor mode.

### 1.1.4 Cortex-M3 may freeze for SLEEPONEXIT single instruction ISR

#### Description

If the Cortex-M3 SLEEPONEXIT functionality is used and the concerned interrupt service routine (ISR) contains only a single instruction, the core becomes frozen. This freezing may occur if only one interrupt is active and it is preempted by an interrupt whose handler only contains a single instruction.
However, any new interrupt that causes a preemption would cause the core to become unfrozen and behave correctly again.

#### Workaround

This scenario does not happen in real application systems since all enabled ISRs should at least contain one instruction. Therefore, if an empty ISR is used, then insert an NOP or any other instruction before the exit instruction (BX or BLX).

# 2 STM32F10xxx silicon limitations

*Table 4* gives quick references to all documented limitations.

**Table 4. Summary of silicon limitations**

| Links to silicon limitations | |
|---|---|
| *Section 2.1: Voltage glitch on ADC input 0* | |
| *Section 2.2: Flash memory read after WFI/WFE instruction* | |
| *Section 2.3: Debug registers cannot be read by user software* | |
| *Section 2.4: Alternate function* | *Section 2.4.1: USART1_RTS and CAN_TX* |
| | *Section 2.4.2: SPI1 in slave mode and USART2 in synchronous mode* |
| | *Section 2.4.3: SPI1 in master mode and USART2 in synchronous mode* |
| | *Section 2.4.4: SPI2 in slave mode and USART3 in synchronous mode* |
| | *Section 2.4.5: SPI2 in master mode and USART3 in synchronous mode* |
| | *Section 2.4.6: I2C2 with SPI2 and USART3* |
| | *Section 2.4.7: I2C1 with SPI1 remapped and used in master mode* |
| | *Section 2.4.8: I2C1 and TIM3_CH2 remapped* |
| *Section 2.5: PVD and USB wakeup events* | |
| *Section 2.6: Compatibility issue with latest compiler releases* | |
| *Section 2.7: Boundary scan TAP: wrong pattern sent out after the “capture IR” state* | |
| *Section 2.8: Flash memory BSY bit delay versus STRT bit setting* | |
| *Section 2.9: $I^2C$ peripheral* | *Section 2.9.1: Some software events must be managed before the current byte is being transferred* |
| | *Section 2.9.2: SMBus standard not fully supported* |
| *Section 2.10: General-purpose timers* | *Section 2.10.1: Missing capture flag* |
| | *Section 2.10.2: Overcapture detected too early* |
| | *Section 2.10.3: General-purpose timer: regulation for 100% PWM* |
| *Section 2.11: LSI clock stabilization time* | |

## 2.1 Voltage glitch on ADC input 0

### Description

A low-amplitude voltage glitch may be generated (on ADC input 0) on the PA0 pin, when the ADC is converting with injection trigger. It is generated by internal coupling and synchronized to the beginning and the end of the injection sequence, whatever the channel(s) to be converted.

The glitch amplitude is less than 150 mV with a typical duration of 10 ns (measured with the I/O configured as high-impedance input and left unconnected). If PA0 is used as a digital output, this has no influence on the signal. If PA0 is used has a digital input, it will not be detected as a spurious transition, providing that PA0 is driven with an impedance lower than 5 kΩ This glitch does not have any influence on the remaining port A pin or on the ADC conversion injection results, in single ADC configuration.

When using the ADC in dual mode with injection trigger, and in order to avoid any side effect, it is advised to distribute the analog channels so that Channel 0 is configured as an injected channel.

### Workaround

None.

## 2.2 Flash memory read after WFI/WFE instruction

### Conditions

- Flash prefetch on
- Flash memory timing set to 2 wait states
- FLITF clock stopped in Sleep mode

### Description

If a WFI/WFE instruction is executed during a Flash memory access and the Sleep duration is very short (less than 2 clock cycles), the instruction fetch from the Flash memory may be corrupted on the next wakeup event.

### Workaround

When using the Flash memory with two wait states and prefetch on, the FLITF clock must *not* be stopped during the Sleep mode – the FLITFEN bit in the RCC_AHBENR register must be set (keep the reset value).

## 2.3 Debug registers cannot be read by user software

### Description

The DBGMCU_IDCODE and DBGMCU_CR debug registers are accessible only in debug mode (not accessible by the user software). When these registers are read in user mode, the returned value is 0x00.

### Workaround

None.

## 2.4 Alternate function

In some specific cases, some potential weakness may exist between alternate functions mapped onto the same pin.

### 2.4.1 USART1_RTS and CAN_TX

#### Conditions

- USART1 is clocked
- CAN is not clocked
- I/O port pin PA12 is configured as an alternate function output.

#### Description

Even if CAN_TX is not used, this signal is set by default to 1 if I/O port pin PA12 is configured as an alternate function output.

In this case USART1_RTS cannot be used.

#### Workaround

When USART1_RTS is used, the CAN must be remapped to either another IO configuration when the CAN is used, or to the unused configuration (CAN_REMAP[1:0] set to “01”) when the CAN is not used.

### 2.4.2 SPI1 in slave mode and USART2 in synchronous mode

#### Conditions

- SPI1 and USART2 are clocked
- I/O port pin PA4 is configured as an alternate function output.

#### Description

USART2 cannot be used in synchronous mode (USART2_CK signal), if SPI1 is used in slave mode.

#### Workaround

None.

### 2.4.3 SPI1 in master mode and USART2 in synchronous mode

#### Conditions

- SPI1 and USART2 are clocked
- I/O port pin PA4 is configured as an alternate function output.

#### Description

USART2 cannot be used in synchronous mode (USART2_CK signal) if SPI1 is used in master mode and SP1_NSS is configured in software mode. In this case USART2_CK is not output on the pin.

#### Workaround

In order to output USART2_CK, the SSOE bit in the SPI1_CR2 register must be set to configure the pin in output mode.

### 2.4.4 SPI2 in slave mode and USART3 in synchronous mode

#### Conditions

- SPI2 and USART3 are clocked
- I/O port pin PB12 is configured as an alternate function output.

#### Description

USART3 cannot be used in synchronous mode (USART3_CK signal) if SPI2 is used in slave mode.

#### Workaround

None.

### 2.4.5 SPI2 in master mode and USART3 in synchronous mode

#### Conditions

- SPI2 and USART3 are clocked
- I/O port pin PB12 is configured as an alternate function output.

#### Description

USART3 cannot be used in synchronous mode (USART3_CK signal) if SPI2 is used in master mode and SP2_NSS is configured in software mode. In this case USART3_CK is not output on the pin.

#### Workaround

In order to output USART3_CK, the SSOE bit in the SPI2_CR2 register must be set to configure the pin in output mode,

### 2.4.6 I2C2 with SPI2 and USART3

#### Conditions

- I2C2 and SPI2 are clocked together or I2C2 and USART3 are clocked together
- I/O port pin PB12 is configured as an alternate function output

### Description

- Conflict between the I2C2 SMBALERT signal (even if this function is not used) and SPI2_NSS in output mode.
- Conflict between the I2C2 SMBALERT signal (even if this function is not used) and USART3_CK.
- In these cases the I/O port pin PB12 is set to 1 by default if the I/O alternate function output is selected and I2C2 is clocked.

### Workaround

I2C2 SMBALERT can be used as an output if SPI2 is configured in master mode with NSS in software mode.
I2C2 SMBALERT can be used in input mode if SPI2 is configured in master or slave mode with NSS managed by software.
SPI2 cannot be used in any other configuration when I2C2 is being used.

USART3 must *not* be used in synchronous mode when I2C2 is being used.

## 2.4.7 I2C1 with SPI1 remapped and used in master mode

### Conditions

- I2C1 and SPI1 are clocked.
- SPI1 is remapped.
- I/O port pin PB5 is configured as an alternate function output.

### Description

Conflict between the SPI1 MOSI signal and the I2C1 SMBALERT signal (even if SMBALERT is not used).

### Workaround

Do not use SPI1 remapped in master mode and I2C1 together.

When using SPI1 remapped, the I2C1 clock must be disabled.

## 2.4.8 I2C1 and TIM3_CH2 remapped

### Conditions

- I2C1 and TIM3 are clocked.
- I/O port pin PB5 is configured as an alternate function output.

### Description

Conflict between the TIM3_CH2 signal and the I2C1 SMBALERT signal, (even if SMBALERT is not used).

In these cases the I/O port pin PB5 is set to 1 by default if the I/O alternate function output is selected and I2C1 is clocked. TIM3_CH2 cannot be used in output mode.

### Workaround

To avoid this conflict, TIM3_CH2 can only be used in input mode.

## 2.5 PVD and USB wakeup events

### Description

PVD and USB Wakeup, which are internally linked to EXTI line16 and EXTI line18, respectively, cannot be used as event sources for the Cortex-M3 core. As a consequence, these signals cannot be used to exit the Sleep or the Stop mode (exit WFE).

### Workaround

Use interrupt sources and the WFI instruction if the application must be woken up from the Sleep or the Stop mode by PVD or USB Wakeup.

## 2.6 Compatibility issue with latest compiler releases

### Description

Compilers with improved optimizations for the STM32F10xxx have been recently released on the market. Revisions Z and B of the medium-density STM32F10xxx devices (STM32F10xx8/B) do not support some of the sequences associated with the high-level optimizations done in these compilers. Revision Y is not affected by this limitation.

### Workaround

This behavior is fully deterministic, and should be detected during firmware development or the validation phase. Consequently, systems already developed, validated and delivered to the field with previous silicon revisions are not affected.

For code update of revision Z and B devices already in the field, do not use these new compilers. To date, compilers known to generate these sequences are:

- IAR EWARM rev 5.20 and later
- GNU rev 4.2.3 and later

For new developments associated with these compilers, revision Y of the STM32F10xx8/B must be used.

## 2.7 Boundary scan TAP: wrong pattern sent out after the “capture IR” state

### Description

After the “capture IR” state of the boundary scan TAP, the two least significant bits in the instruction register should be loaded with “01” for them to be shifted out whenever a next instruction is shifted in.
However, the boundary scan TAP shifts out the latest value loaded into the instruction register, which could be “00”, “01”, “10” or “11”.

### Workaround

The data shifted out, after the capture IR state, in the boundary scan flow should therefore be ignored and the software should check not only the two least significant bits (XXX01) but all register bits (XXXXX).

## 2.8 Flash memory BSY bit delay versus STRT bit setting

### Description

When the STRT bit in the Flash memory control register is set (to launch an erase operation), the BSY bit in the Flash memory status register goes high one cycle later.

Therefore, if the FLASH_SR register is read immediately after the FLASH_CR register is written (STRT bit set), the BSY bit is read as 0.

### Workaround

Read the BSY bit at least one cycle after setting the STRT bit.

## 2.9 $I^2C$ peripheral

### 2.9.1 Some software events must be managed before the current byte is being transferred

#### Description

When the EV7, EV7_1, EV6_1, EV2, EV8, and EV3 events are not managed before the current byte is being transferred, problems may be encountered such as receiving an extra byte, reading the same data twice or missing data.

#### Workarounds

When it is not possible to manage the EV7, EV7_1, EV6_1, EV2, EV8, and EV3 events before the current byte transfer and before the acknowledge pulse when changing the ACK control bit, it is recommended to:

1. use the $I^2C$ with DMA in general, except when the Master is receiving a single byte
2. use $I^2C$ interrupts and boost their priorities to the highest one in the application to make them uninterruptible

### 2.9.2 SMBus standard not fully supported

#### Description

The $I^2C$ peripheral is not fully compliant with the SMBus v2.0 standard since It does not support the capability to NACK an invalid byte/command.

#### Workarounds

A higher-level mechanism should be used to verify that a write operation is being performed correctly at the target device, such as:

1. Using the SMBAL pin if supported by the host
2. the alert response address (ARA) protocol
3. the Host notify protocol

## 2.10 General-purpose timers

### 2.10.1 Missing capture flag

#### Description

In capture mode, when a capture occurs while the CCRx register is being read, the capture flag (CCxIF) may be cleared without the overcapture flag (CCxOF) being set. The new data are actually captured in the capture register.

#### Workaround

An external interrupt can be enabled on the capture I/O just before reading the capture register (in the capture interrupt), and disabled just after reading the captured data. Possibly, a missed capture will be detected by the EXTI peripheral.

### 2.10.2 Overcapture detected too early

#### Description

In capture mode, the overcapture flag (CCxOF) can be set even though no data have been lost.

#### Conditions

If a capture occurs while the capture register is being read, an overcapture is detected even though the previously captured data are correctly read and the new data are correctly stored into the capture register.

The system is at the limit of an overcapture but no data are lost.

#### Workaround

None.

### 2.10.3 General-purpose timer: regulation for 100% PWM

#### Description

When the OCREF_CLR functionality is activated, the OCxREF signal becomes de-asserted (and consequently OCx is deasserted / OCxN is asserted) when a high level is applied on the OCREF_CLR signal. The PWM then restarts (output re-enabled) at the next counter overflow.

But if the PWM is configured at 100% (CCxR > ARR), then it does not restart and OCxREF remains de-asserted.

#### Workaround

None.

## 2.11 LSI clock stabilization time

### Description

When the LSIRDY flag is set, the clock may still be out of the specified frequency range ($f_{LSI}$ parameter, see LSI oscillator characteristics in the product datasheet).

### Workaround

To have a fully stabilized clock in the specified range, a software temporization of 100 µs should be added.

# Appendix A Revision code on device marking

*Figure 1*, *Figure 2*, *Figure 3*, *Figure 4* and *Figure 5* show the marking compositions for the LFBGA100, LQFP100, LQFP64, LQFP48 and VFQFPN36 packages, respectively. Only the Additional field containing the Revision code is shown.

**Figure 1. LFBGA100 top package view**

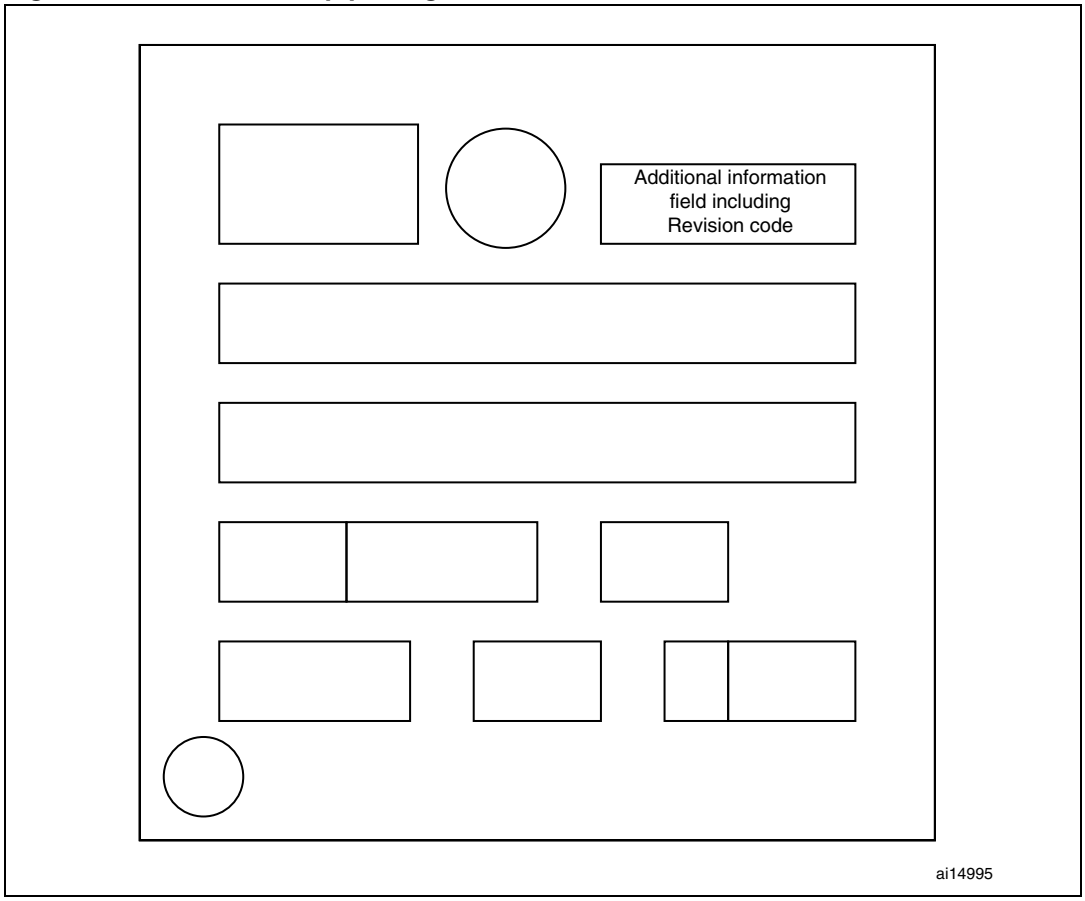


**Figure 2. LQFP100 top package view**

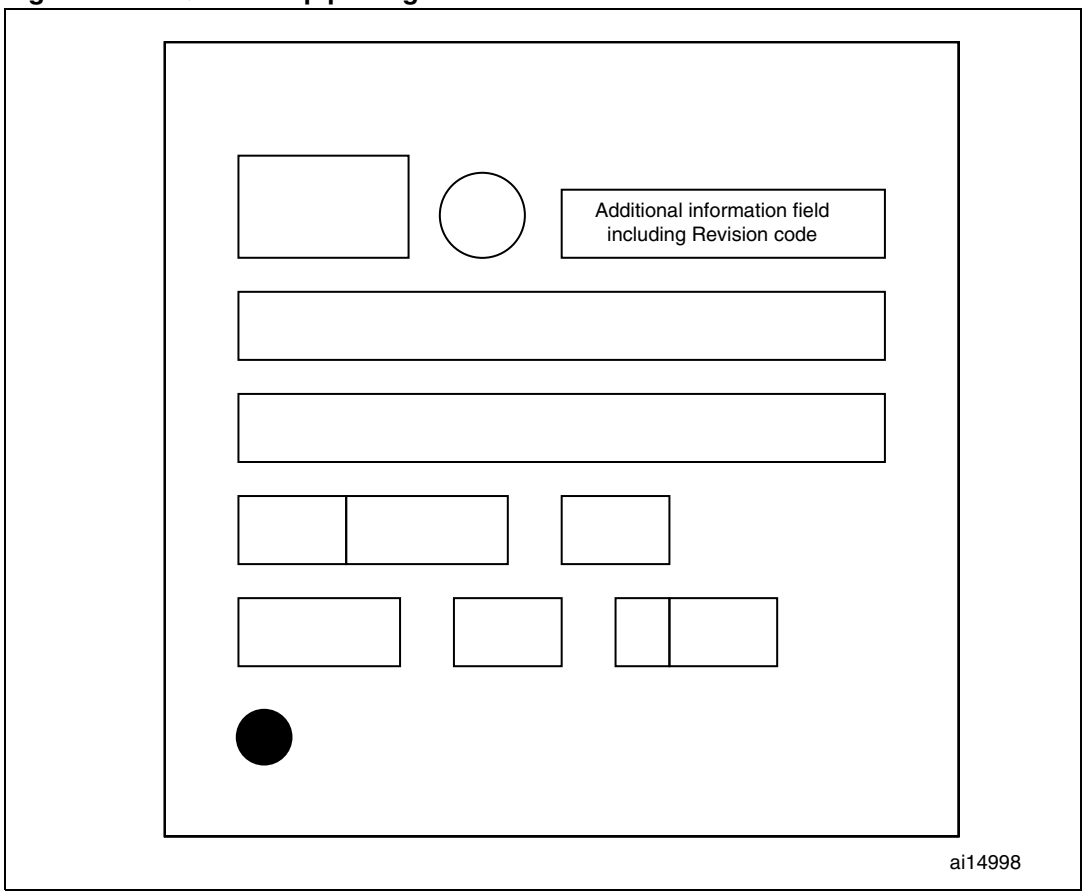


**Figure 3. LQFP64 top package view**

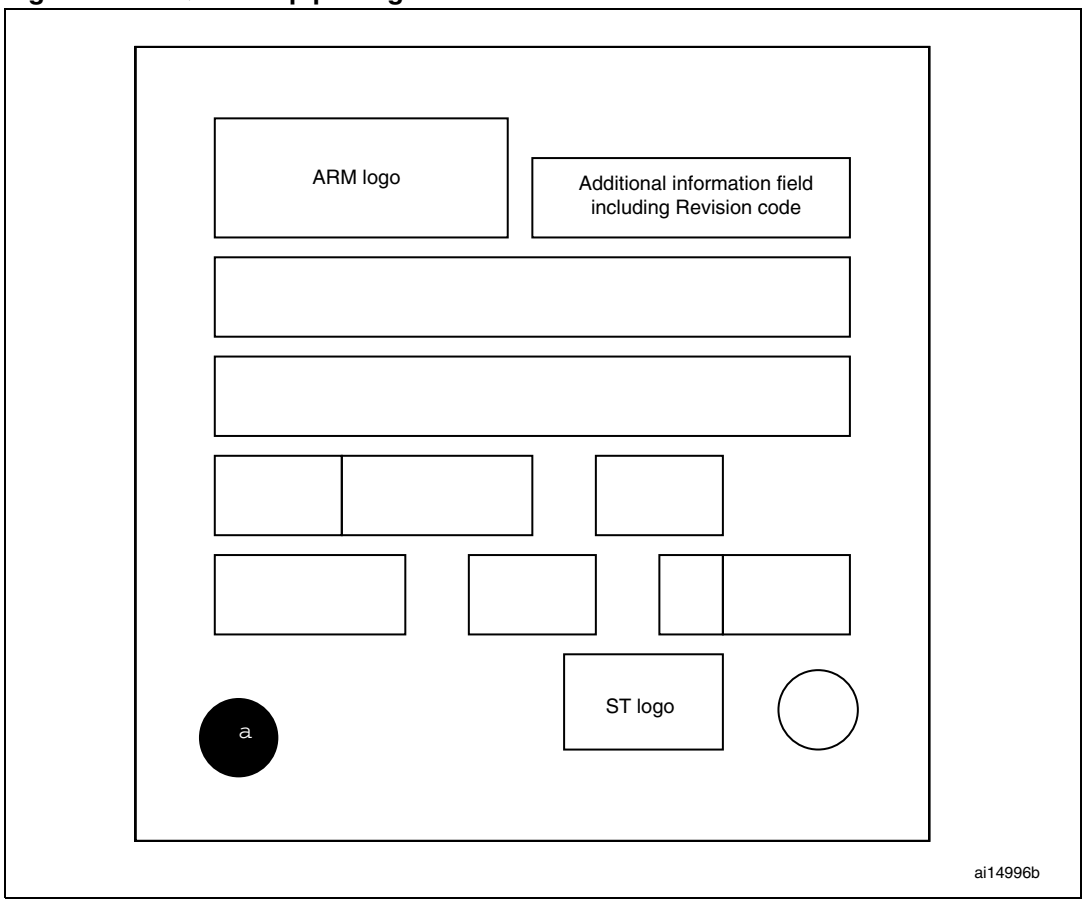


**Figure 4. LQFP48 top package view**

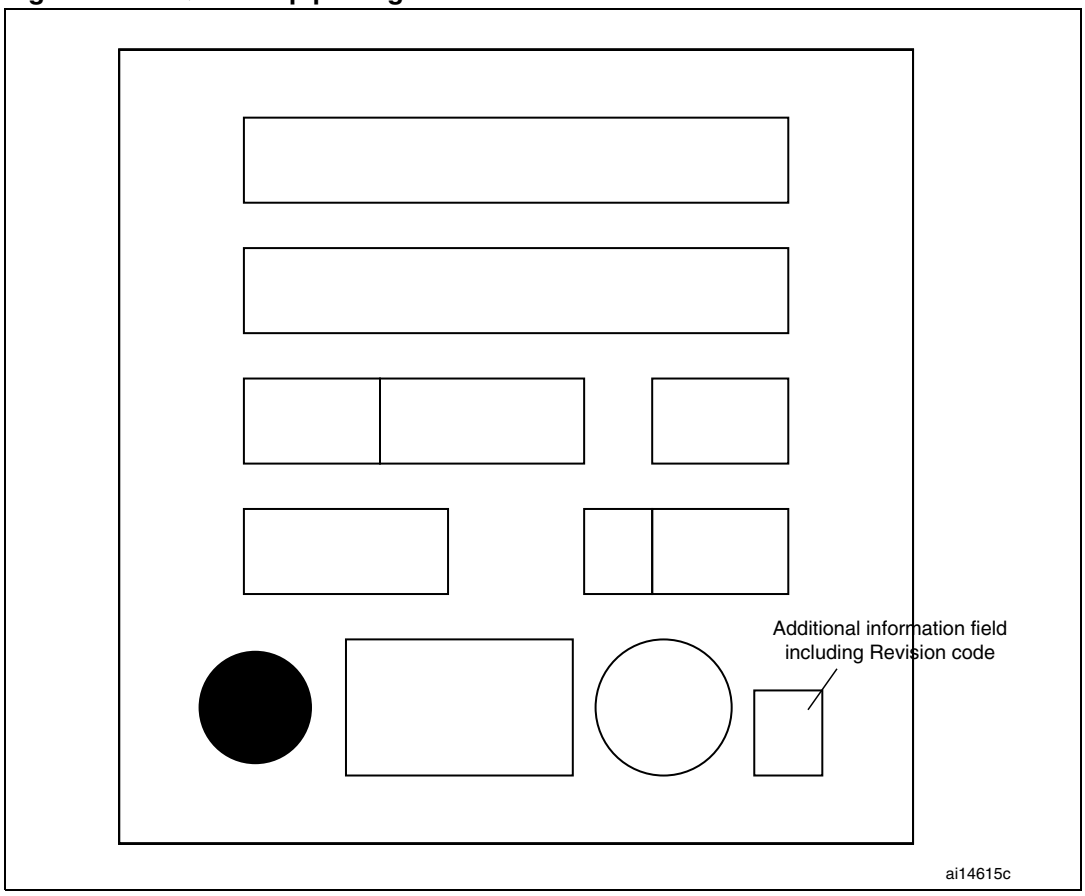
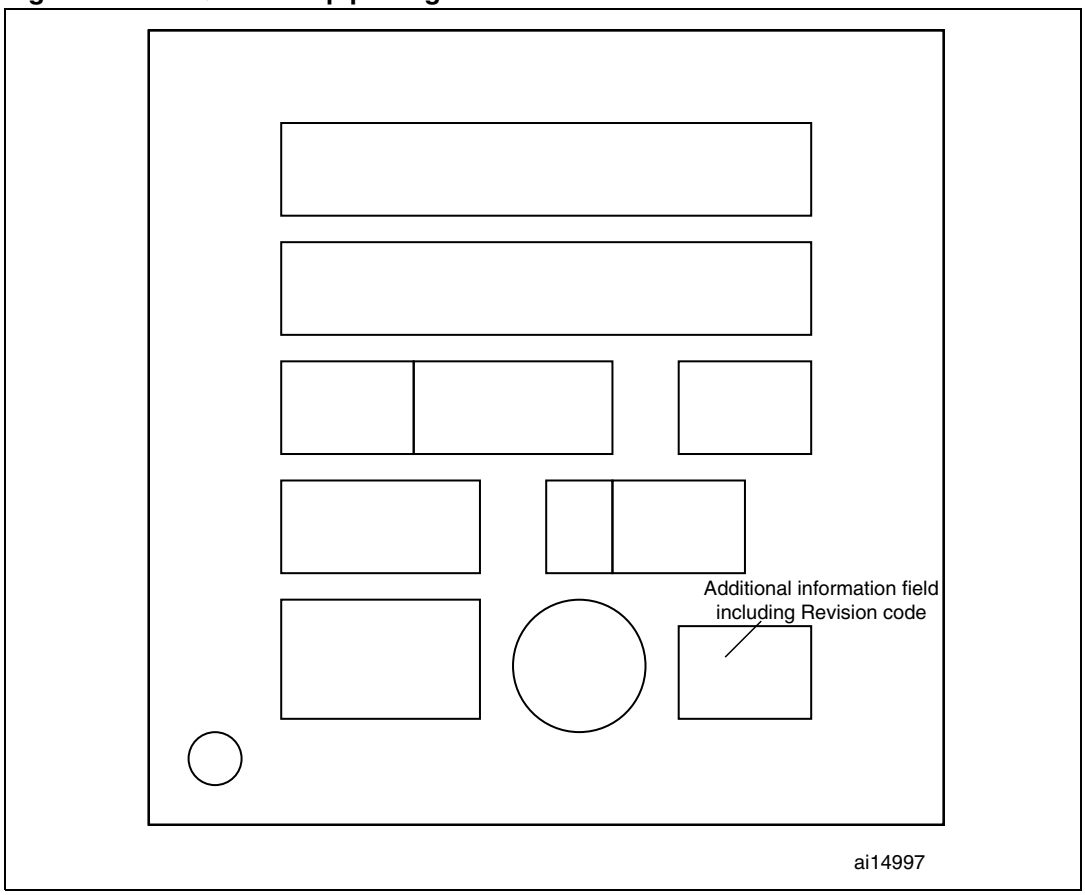


**Figure 5. VFQFPN36 top package view**



# Revision history

**Table 5. Document revision history**

| Date | Revision | Changes |
|---|---|---|
| 28-Mar-2008 | 1 | Initial release. |
| 07-Apr-2008 | 2 | *Section 2.2: Flash memory read after WFI/WFE instruction on page 6* added.<br>Workaround specified in *Section 2.4.1: USART1_RTS and CAN_TX*. |
| 23-May-2008 | 3 | The errata sheet also applies to Revision Y devices.<br>*Section 2.1: PD0 and PD1 use in output mode*, *Section 2.2: ADC auto-injection channel* and *Section 2.3: ADC combined injected simultaneous+interleaved* removed from errata sheet.<br>*Section 2.3: Debug registers cannot be read by user software on page 6* added. Small text changes. |
| 18-Jul-2008 | 4 | *Section 2.5: PVD and USB wakeup events* added. |
| 01-Oct-2008 | 5 | This errata sheet also applies to STM32F102xx medium-density devices. Though medium-density devices with 32 Kbyte of Flash were removed, the errata sheet still applies to devices whose commercial code does not contain an “A”.<br>*Section 2.6: Compatibility issue with latest compiler releases* added.<br>*Figure 1: LFBGA100 top package view* added. *Figure 3: LQFP64 top package view* and *Figure 4: LQFP48 top package view* corrected. |
| 11-Feb-2009 | 6 | *Section 1: ARM™ 32-bit Cortex®-M3 limitations* specified (*Table 3: Cortex-M3 core limitations and impact on microcontroller behavior* added limitations described).<br>Added limitations:<br>– *Boundary scan TAP: wrong pattern sent out after the “capture IR” state*<br>– *Flash memory BSY bit delay versus STRT bit setting*<br>– *$I^2C$ peripheral*<br>– *General-purpose timers*<br>– *LSI clock stabilization time*<br>*Table 4: Summary of silicon limitations on page 5* added. |

**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.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.



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**