STM32L15xxC/D STM32L162xC/D Errata sheet

STM32L15xxC/D and STM32L162xC/D ultralow power MCU limitations

Silicon identification

This errata sheet applies to revisions A, Z and Y of the STMicroelectronics STM32L15xxC/D and STM32L162xC/D ultralow power products. This family features an ARM™ 32-bit Cortex®-M3 core, for which an errata notice is also available (see *Section 1* for details).

A full list of root part numbers is shown in *Table 1*.

The products can be identified (see *Table 2*) by:

The revision code marked below the sales type on the device package

The last three digits of the internal sales type printed on the box label

Table 1. Device summary

Reference	Part number
STM32L15xxC	STM32L151CC STM32L151QC STM32L151RC STM32L151UC STM32L151VC STM32L151ZC STM32L152CC STM32L152QC STM32L152RC STM32L152UC STM32L152VC STM32L152ZC
STM32L15xxD	STM32L151QD STM32L151RD STM32L151VD STM32L151ZD STM32L152QD STM32L152RD STM32L152VD STM32L152ZD
STM32L162xC	STM32L162VC STM32L162ZC STM32L162QC STM32L162RC
STM32L162xD	STM32L162VD STM32L162ZD STM32L162QD STM32L162RD

Table 2. Device identification(1)

	Sales type	Revision code(2) marked on device
1	STM32L15xxD, STM32L15xQC, STM32L15xZC, STM32L15xRCY, STM32L162xD, STM32L162ZC, STM32L162QC	“A”, “Z” or “Y”
2	STM32L151CC/RC/UC/VC(3), STM32L152CC/UC/RC/VC(3), STM32L162RC, STM32L162VC	“A”

1. The REV_ID bits in the DBGMCU_IDCODE register show the revision code of the device (see the *STM32L151xx, STM32L152xx and STM32L162xx advanced ARM-based 32-bit MCUs* reference manual (RM0038) for details on how to find the revision code).
2. Refer to *Appendix A: Revision and date codes on device marking* for details on how to identify the revision code on the different packages.
3. Except products in WLCSP64 package (i.e. STM32L15xRCY) and sales types ending with “A” (which belong to Line 1 of the table).

Contents

List of tables

List of figures

1 ARM™ 32-bit Cortex®-M3 limitations

An ARM errata notice of the STM32L15xxC/D and STM32L162xC/D core is available from the following web address:
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.eat0420c/index.html.

The direct link to the errata notice pdf is:
http://infocenter.arm.com/help/topic/com.arm.doc.eat0420c/Cortex-M3-Errata-r2p0-v2.pdf

All the described limitations are minor and relate to revision r2p0-00rel0 of the Cortex-M3 core. *Table 3* summarizes these limitations and their implications on the behavior of the STM32L15xxC/D and STM32L162xC/D ultralow power devices.

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

ARM ID	ARM category	ARM summary of errata	Impact on STM32L15xxC/D and STM32L162xC/D ultralow power 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
752419	Cat 2	Interrupted loads to SP can cause erroneous behavior	Minor
740455	Cat 2	SVC and BusFault/MemManage may occur out of order	Minor

1.1 Cortex-M3 limitation description for the STM32L15xxC/D and STM32L162xC/D ultralow power devices

Only the limitations described below have an impact, even though minor, on the implementation of STM32L15xxC/D and STM32L162xC/D ultralow power devices.

All 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 the STM32L15xxC/D and STM32L162xC/D ultralow power devices (Cortex-M3 r2p0-00rel0).

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.

Workaround

1. This limitation does not impact the STM32L15xxC/D and STM32L162xC/D 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, the compilers 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 wake from event (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 a WFE if no other event or interrupt occurs.

Workaround

1. For the following interrupt sources:
   - all external interrupts/events lines (EXTI)
   - PVD output on EXTI line 16 (if VREFINT is enabled only)
   - RTC Alarm on EXTI line 17
   - USB Wake-up on EXTI line 18
   - RTC tamper and timestamp on EXTI line 19
   - RTC Wake-up on EXTI line 20
   - Comparator 1 wake-up on EXTI line 21 (if VREFINT is enabled only)
   - Comparator 2 wake-up on EXTI line 22 (if VREFINT is enabled only)
   - Channel acquisition on EXTI line 23

   use STM32L15x external events instead of interrupts to wake up the core from a WFE by configuring an external or internal EXTI line in event mode.
2. For all other interrupt sources, a timer must be programmed to provide a timeout event and wake-up the core if the event is likely to arrive before the WFE instruction is executed.

1.1.3 Cortex-M3 Interrupted loads to the stack-pointer can cause erroneous behavior

Description

If an interrupt occurs during the data-phase of a single word load to the stack-pointer (SP/R13), erroneous behavior can occur. In all cases, returning from the interrupt will result in the load instruction being executed an additional time. For all instructions performing an update to the base register, the base register will be erroneously updated on each execution, resulting in the stack-pointer being loaded from an incorrect memory location.

The instructions affected by this limitation are the following:

- LDR SP, [Rn],#imm
- LDR SP, [Rn,#imm]!
- LDR SP, [Rn,#imm]
- LDR SP, [Rn]
- LDR SP, [Rn,Rm]

Workaround

As of today, no compiler generates these particular instructions. This limitation can only occur with hand-written assembly code.

Both issues can be solved by replacing the direct load to the stack pointer by an intermediate load to a general-purpose register followed by a move to the stack pointer.

Example:

Replace LDR SP, [R0] by

LDR R2,[R0]

MOV SP,R2

1.1.4 SVC and BusFault/MemManage may occur out of order

Description

If an SVC exception is generated by executing the SVC instruction while the following instruction fetch is faulted, then the MemManage or BusFault handler may be entered even though the faulted instruction which followed the SVC should not have been executed.

Workaround

A workaround is only required if the SVC handler will not return to the return address that has been stacked for the SVC exception and the instruction access after the SVC will fault. If this is the case then padding can be inserted between the SVC and the faulting area of code, for example, by inserting NOP instructions.

2 STM32L15xxC/D and STM32L162xC/D silicon limitations

Table 4 summarizes the fix status for products listed on line 1 in *Table 2: Device identification*.

Table 5 summarizes the fix status for products listed on line 2 in *Table 2: Device identification*.

Legend for *Table 4* and *Table 5*: A = workaround available; N = no workaround available; P = partial workaround available, '-' and grayed = fixed.

Table 4. Summary of silicon limitations 1(1)

Links to silicon limitations		Rev A	Rev Z	Rev Y
Section 2.1: System limitations	*Section 2.1.1: AOP_RANGE bit is mapped on register COMP_CSR(28) instead of register AOP_CSR(28)*	A	-	-
	Section 2.1.2: Missing analog switch on GPIO PC10	N	N	N
	Section 2.1.3: Ports PG0 to PG15 and ports PF0 to PF5 sink current when VIN>VDD	A	-	-
	Section 2.1.4: If Debugger is connected in JTAG mode and JNTRST (PB4) pin configuration is changed, the connection is lost	A	A	A
	Section 2.1.5: Read protection: a mass erase occurs if the RDP register is written with Level0 when Level0 is already set	A	A	A
	Section 2.1.6: In STOP mode, RAMs are not in power down if DMA is enabled	A	-	-
	Section 2.1.7: Invalid read from Data EEPROM after write in Program Flash	A	-	-
	Section 2.1.8: Debugging Stop mode with WFE entry	A	A	A
	Section 2.1.9: Boot from bank2 limitation	A	A	-
	Section 2.1.13: Pull-up on PB7 when configured in analog mode	A	A	A
Section 2.2: LCD peripheral limitations	*Section 2.2.1: High drive resistive network total value too low*	N	-	-
Section 2.3: IWDG peripheral limitation	*Section 2.3.1: RVU and PVU flags are not reset in STOP mode*	A	A	A

Table 4. Summary of silicon limitations 1(1) (continued)

Links to silicon limitations		Rev A	Rev Z	Rev Y
Section 2.4: I2C peripheral limitations	*Section 2.4.1: SMBus standard not fully supported*	A	A	A
	Section 2.4.2: Wrong behavior of I2C peripheral in Master mode after misplaced STOP	A	A	A
	Section 2.4.3: Violation of I2C "setup time for repeated START condition" parameter	A	A	A
	Section 2.4.4: In I2C slave "NOSTRETCH" mode, underrun errors may not be detected and may generate bus errors	A	A	A
Section 2.5: SPI/I2S peripheral limitations	*Section 2.5.1: In I2S slave mode, WS level must to be set by the external master when enabling the I2S*	A	A	A
Section 2.6: USART peripheral limitations	*Section 2.6.1: Idle frame is not detected if receiver clock speed is deviated*	N	N	N
	Section 2.6.2: In full duplex mode, the Parity Error (PE) flag can be cleared by writing the data register	A	A	A
	Section 2.6.3: Parity Error (PE) flag is not set when receiving in Mute mode using address mark detection	N	N	N
	Section 2.6.4: Break frame is transmitted regardless of nCTS input line status	N	N	N
	Section 2.6.5: nRTS signal abnormally driven low after a protocol violation	A	A	A
Section 2.7: SDIO peripheral limitations	*Section 2.7.1: SDIO hardware flow control*	A	A	A
	Section 2.7.2: Wrong CCRCFAIL status after a response without CRC	A	A	A
	Section 2.7.3: No underrun detected and wrong data transmission	A	A	A
	Section 2.7.4: CE-ATA multiple write command and card busy signal management	A	A	A
Section 2.9: FSMC peripheral limitation	*Section 2.9.1: FSMC NOR Flash/PSRAM controller asynchronous access on bank2..4 when bank1 is in synchronous mode (CBURSTRW bit is set)*	A	A	A
	Section 2.9.2: FSMC Synchronous mode and disabled NWAIT signal	A	A	A
	Section 2.9.3: Dummy read cycles inserted when reading synchronous memories	-	-	-

1. This table applies to products listed on line 1 of *Table 2: Device identification*.

Table 5. Summary of silicon limitations 2(1)

Links to silicon limitations		Rev A
Section 2.1: System limitations	*Section 2.1.4: If Debugger is connected in JTAG mode and JNTRST (PB4) pin configuration is changed, the connection is lost*	A
	Section 2.1.5: Read protection: a mass erase occurs if the RDP register is written with Level0 when Level0 is already set	A
	Section 2.1.8: Debugging Stop mode with WFE entry	A
	Section 2.1.9: Boot from bank2 limitation	-
	Section 2.1.10: Range 3 of dynamic voltage scaling cannot be used	A
	Section 2.1.11: The operational amplifier factory trimming value cannot be selected	A
	Section 2.1.12: Electrostatic discharge limits are 1kV (HBM) and 250 V (CDM) instead of 2 kV and 500 V respectively	N
	Section 2.1.14: Data EEPROM cycling limited to 100 kcycles	N
Section 2.3: IWDG peripheral limitation	*Section 2.3.1: RVU and PVU flags are not reset in STOP mode*	A
Section 2.4: I2C peripheral limitations	*Section 2.4.1: SMBus standard not fully supported*	A
	Section 2.4.2: Wrong behavior of I2C peripheral in Master mode after misplaced STOP	A
	Section 2.4.3: Violation of I2C “setup time for repeated START condition” parameter	A
	Section 2.4.4: In I2C slave “NOSTRETCH” mode, underrun errors may not be detected and may generate bus errors	A
Section 2.5: SPI/I2S peripheral limitations	*Section 2.5.1: In I2S slave mode, WS level must to be set by the external master when enabling the I2S*	A
Section 2.6: USART peripheral limitations	*Section 2.6.1: Idle frame is not detected if receiver clock speed is deviated*	N
	Section 2.6.2: In full duplex mode, the Parity Error (PE) flag can be cleared by writing the data register	A
	Section 2.6.3: Parity Error (PE) flag is not set when receiving in Mute mode using address mark detection	N
	Section 2.6.4: Break frame is transmitted regardless of nCTS input line status	N
	Section 2.6.5: nRTS signal abnormally driven low after a protocol violation	A
Section 2.8: ADC peripheral limitation	*Section 2.8.1: ADC accuracy lowered*	N

1. This table applies to products listed on line 2 of *Table 2: Device identification*.

2.1 System limitations

2.1.1 AOP_RANGE bit is mapped on register COMP_CSR(28) instead of register AOP_CSR(28)

Description

The operational amplifier voltage range is controlled by COMP_CSR(28) instead of AOP_CSR(28). The COMP_CSR(28) bit keeps its original function which selects GPIO PC3 as ADC input channel CH13 together with the control of operational amplifier voltage range.

Workaround

When the operational amplifiers are enabled, ADC channel CH13 must not be used.

This limitation is planned to be fixed in the next revision.

2.1.2 Missing analog switch on GPIO PC10

Description

An analog switch is not present on port PC10. As a consequence bit RI_ASMR3(10) has no effect, so PC10 cannot be used for the touch sensing feature.

Workaround

None.

2.1.3 Ports PG0 to PG15 and ports PF0 to PF5 sink current when $V_{IN}>V_{DD}$

Description

Ports PG0 to PG15 and ports PF0 to PF5 exhibit leakage current when a voltage above V_{DD} is applied on them.

Workaround

When current consumption is important, do not apply voltage above V_{DD} on port PG0 to PG15 and port PF0 to PF5.

This limitation is planned to be fixed in the next revision.

2.1.4 If Debugger is connected in JTAG mode and JNTRST (PB4) pin configuration is changed, the connection is lost

Description

PB4 is configured by default in Alternate function mode after Reset.

When the configuration bit changes from Alternate function to Input, Analog or GPIO, the Reset signal connected to the CPU is tied to '0', and forces a Reset on the CPU TAP that stops the Debugger connection, even if the pin itself is pulled up.

Only JTAG mode with 4 wires is impacted. Serial Wire Debug (SWD) mode is not impacted.

Workaround

During the debug phase, when the debugger is connected in JTAG mode is used, I/O port PB4 should not be used by the application. If the application needs to use PB4 even during debug phase, the Debugger should use Serial Wire Debug (SWD) mode to connect.

2.1.5 Read protection: a mass erase occurs if the RDP register is written with Level0 when Level0 is already set

Description

The read protection ranges from the lowest level 0 (no read protection) to level 2 (disable debug/chip read protection). It is always possible to increase the read protection level without any side effects. It is not possible to decrease the protection level from level 2 to a lower level. It is possible to decrease the protection level from level 1 to level 0, but to avoid allowing access to data that were previously protected, a Mass Erase of the data EEPROM, program Flash and backup registers is performed.

The limitation appears when level 0 is requested (writing 0xAA in the Option byte RDP) while the protection Level is already at 0, in this case an unwanted mass erase is performed.

Workaround

Before setting Level0 in the RDP register, check that the current level is not equal to Level0.

- If the current level is not equal to Level0, Level0 can be activated.
- If the current level is Level0 then the RDP register must not be written again with Level0.

2.1.6 In STOP mode, RAMs are not in power down if DMA is enabled

Description

In STOP mode, the RAM memories (Main and USB) are not in power-down when DMA1 and/or DMA2 are enabled in the RCC registers. If DMA1 and/or DMA2 are disabled, the RAMs are switched off. This induces an overconsumption of about 200 nA in STOP mode.

Workaround

Before entering STOP mode, if the DMA1EN and DMA2EN bits in the RCC_AHBENR register are enabled then they have to be reset. When the MCU wakes up from STOP mode, the DMA1EN and DMA2EN bits in the RCC_AHBENR register have to be enabled again (this can be done by the interrupt subroutine).

2.1.7 Invalid read from Data EEPROM after write in Program Flash

Description

When a read from data EEPROM in one bank is performed after a write in the program code, the value of the read might be wrong.

This problem can occur even if the write and the read access are separated by a delay of several CPU clock periods.

Workaround

In this case, perform two read operations from the same address location in data EEPROM, the first one should be discarded, the second one is correct and the value can be used.

2.1.8 Debugging Stop mode with WFE entry

Description

When the Stop debug mode is enabled (DBG_STOP bit set in the DBGMCU_CR register) this allows software debugging during Stop mode. However, if the application software uses the WFE instruction to enter Stop mode, after wakeup some instructions could be missed if the WFE is followed by sequential instructions. This affects only Stop debug mode with WFE entry.

Workaround

To debug Stop mode with WFE entry, the WFE instruction must be inside a dedicated function with 1 instruction (NOP) between the execution of the WFE and the Bx LR.

Example: __asm void _WFE(void) {

WFE

NOP

BX lr }

2.1.9 Boot from bank2 limitation

Description

The condition is either “Read protection level 2 is set and Boot pins are configured to boot from System Memory” or “BFB2 option bit is reset (boot from system memory) and Boot pins are configured to boot from Internal Main Flash”.

In both cases, the Bootloader is executed and checks if there is a valid code in Bank2, then it jumps to it, else if there is a valid code in Bank1 it jumps to it, otherwise it goes to infinite loop. In case there is a valid code in Bank2 or Bank1, the Bootloader initializes the user code's top-of-stack address, then jumps to execute this code.

Two issues might occur:

1. The first issue is that the top-of-stack address that is initialized by the Bootloader is not the value stored by the linker in the first entry of the vector table, but this value incremented by 8. As a consequence, when the user code starts execution, the stack pointer is corrupted. Two behaviors can then occur depending on how the compiler/linker manages the stack memory:
   - The compiler/linker (case of ARM/IAR) places the top of the stack in the first region of the SRAM memory. In this case, the stack corruption overflows the user data located in the adjacent 8-byte memory location.
   - The compiler/linker (case of GNU) places the top of the stack at the top of the SRAM memory. In this case, due to the MSP corruption, the top of the stack is

moved beyond the SRAM address boundaries, and when the first stack push occurs a hard fault is generated.

2. The second issue occurs when the top of the stack is set at the top of the SRAM memory (i.e. the first word of the code area points outside the physical SRAM area). In that case, after reset, neither the code in bank1 nor the code in bank2 is executed.

Workaround

Make sure the compiler/linker used does not place the top of the stack at the top of the SRAM memory.

In the startup file, the user code should force the top-of-stack address before jumping to the main program. This can be done in the “Reset_Handler” routine using the following instructions:

LDR R1,=0x08000000 LDR R0, [R1] msr msp, r0

2.1.10 Range 3 of dynamic voltage scaling cannot be used

Description

Dynamic voltage scaling is a power management technique which consists in increasing or decreasing the voltage used for the digital peripherals (VCORE) according on the circumstances.

Three voltage ranges can be configured. From those three voltage ranges, only Range 1 and Range 2 are functional. The low power/low performance Range 3 must not be selected.

Workaround

Use voltage scaling Range 2 instead of Range 3.

2.1.11 The operational amplifier factory trimming value cannot be selected

Description

The factory trimming values of the operational amplifiers are present but cannot be selected as trimming values.

Workaround

When the operational amplifiers are enabled, the trimming values should be initialized with the preset factory trimming value.

If you want to keep this behavior, you need to copy the factory trimming value to the user trimming values:

Copy content of address 0x1FF80048[15:0] to OPAMP_OTR[15:0].

Copy content of address 0x1FF8004C[3:0] to OPAMP_OTR[19:16].

Copy content of address 0x1FF80050[15:0] to OPAMP_LPOTR[15:0].

Copy content of address 0x1FF80054[3:0] to OPAMP_LPOTR[19:16].

2.1.12 Electrostatic discharge limits are 1kV (HBM) and 250 V (CDM) instead of 2 kV and 500 V respectively

Description

The ESD absolute maximum ratings are limited as follows:

Table 6. ESD absolute maximum ratings

Symbol	Ratings	Conditions	Class	Maximum value(1)	Unit
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	T_A = +25 °C, conforming to JESD22-A114	2	1000	V
$V_{ESD(CDM)}$	Electrostatic discharge voltage (charge device model)	T_A = +25 °C, conforming to JESD22-C101	II	250	

1. Based on characterization results, not tested in production.

Workaround

None.

2.1.13 Pull-up on PB7 when configured in analog mode

Description

PB7 can be used as a Power Voltage Detection Input. To select this, the IO is configured in analog mode through GPIOB_MODER and selected as external input analog voltage for PVD level selection through bits PLS[2:0] in the PWR_CR register.

When PB7 is configured in analog mode but not selected for PVD level selection, then an internal pull-up is visible through PB7.

Workaround

Configuring PB7 to limit power consumption whatever the external connection can be done differently:

1. Keep PB7 in Input mode (default of GPIOB_MODER)
2. Disconnect the schmidt trigger on PB7 by programming the RI_HYSCR1

2.1.14 Data EEPROM cycling limited to 100 kcycles

The Data EEPROM, which usually supports 300 kcycles, supports only 100 kcycles. All the retention parameter are the same (replacing in the electrical characteristics 300 kcycles with 100 kcycles).

Workaround

None.

2.2 LCD peripheral limitations

2.2.1 High drive resistive network total value too low

Description

The value of the high drive resistive network is 60 kΩ instead of 240 kΩ, which leads to higher current consumption when used.

Workaround

None.

2.3 IWDG peripheral limitation

2.3.1 RVU and PVU flags are not reset in STOP mode

Description

The RVU and PVU flags of the IWDG_SR register are set by hardware after a write access to the IWDG_RLR and the IWDG_PR registers, respectively. If the Stop mode is entered immediately after the write access, the RVU and PVU flags are not reset by hardware.

Before performing a second write operation to the IWDG_RLR or the IWDG_PR register, the application software must wait for the RVU or PVU flag to be reset. However, since the RVU/PVU bit is not reset after exiting Stop mode, the software goes into an infinite loop and the independent watchdog (IWDG) generates a reset after the programmed timeout period.

Workaround

Wait until the RVU or PVU flag of the IWDG_SR register are reset before entering Stop mode.

2.4 I^{2}C peripheral limitations

2.4.1 SMBus standard not fully supported

Description

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

Workaround

The following higher-level mechanisms should be used to verify that a write operation is being performed correctly at the target device:

1. The SMBA pin if supported by the host
2. The alert response address (ARA) protocol
3. The host notify protocol

2.4.2 Wrong behavior of I²C peripheral in Master mode after misplaced STOP

The I2C peripheral does not enter Master mode properly if a misplaced STOP is generated on the bus and the START bit is already set in I2C_CR2. In this case the START condition is not correctly generated on the bus and can create bus errors.

Workaround

In the I2C standard, it is not allowed to send a STOP before the full byte is transmitted (8 bits + acknowledge). Other derived protocols like CBUS allow it, but they are not supported by the I²C peripheral.

In case of noisy environment in which unwanted bus errors can occur, it is recommended to reset the I2C peripheral by setting the SWRST bit in the I2C_CR2 control register if a BERR is detected while the START bit is set in I2C_CR2.

No fix is planned for this limitation.

2.4.3 Violation of I²C “setup time for repeated START condition” parameter

Description

In case of a repeated Start, the “setup time for repeated START condition” parameter (named $t_{SU(STA)}$ in the datasheet and Tsu:sta in the I²C specifications) may be slightly violated when the I²C operates in Master Standard mode at a frequency ranging from 88 to 100 kHz. $t_{SU(STA)}$ minimum value may be 4 µs instead of 4.7 µs.

The issue occurs under the following conditions:

1. The I²C peripheral operates in Master Standard mode at a frequency ranging from 88 to 100 kHz (no issue in Fast mode)
2. and the SCL rise time meets one of the following conditions:
   - The slave does not stretch the clock and the SCL rise time is more than 300 ns (the issue cannot occur when the SCL rise time is less than 300 ns).
   - or the slave stretches the clock.

Workaround

Reduce the frequency down to 88 kHz or use the I²C Fast mode if it is supported by the slave.

2.4.4 In I²C slave “NOSTRETCH” mode, underrun errors may not be detected and may generate bus errors

Description

The data valid time ($t_{VD;DAT}$, $t_{VD;ACK}$) described by the I²C specifications may be violated as well as the maximum current data hold time ($t_{HD;DAT}$) under the conditions described below. In addition, if the data register is written too late and close to the SCL rising edge, an error may be generated on the bus: SDA toggles while SCL is high. These violations cannot be detected because the OVR flag is not set (no transmit buffer underrun is detected).

This issue occurs under the following conditions:

1. The I^2C peripheral operates In Slave transmit mode with clock stretching disabled (NOSTRETCH=1)
2. and the application is late to write the DR data register, but not late enough to set the OVR flag (the data register is written before the SCL rising edge).

Workaround

If the master device supports it, use the clock stretching mechanism by programming the bit NOSTRETCH=0 in the I2C_CR1 register.

If the master device does not support it, ensure that the write operation to the data register is performed just after TXE or ADDR events. You can use an interrupt on the TXE or ADDR flag and boost its priority to the higher level or use DMA.

Using the “NOSTRETCH” mode with a slow I^2C bus speed can prevent the application from being late to write the DR register (second condition).

Note: The first data to be transmitted must be written into the data register after the ADDR flag is cleared, and before the next SCL rising edge, so that the time window to write the first data into the data register is less than t_{LOW}.

If this is not possible, a possible workaround can be the following:

1. Clear the ADDR flag
2. Wait for the OVR flag to be set
3. Clear OVR and write the first data.

The time window for writing the next data is then the time to transfer one byte. In that case, the master must discard the first received data.

2.5 SPI/I2S peripheral limitations

2.5.1 In I2S slave mode, WS level must to be set by the external master when enabling the I2S

Description

In slave mode the WS signal level is used only to start the communication. If the I2S (in slave mode) is enabled while the master is already sending the clock and the WS signal level is low (for I2S protocol) or is high (for the LSB or MSB-justified mode), the slave starts communicating data immediately. In this case the master and slave will be desynchronized throughout the whole communication.

Workaround

The I2S peripheral must be enabled when the external master sets the WS line at:

- High level when the I2S protocol is selected.
- Low level when the LSB or MSB-justified mode is selected.

2.6 USART peripheral limitations

2.6.1 Idle frame is not detected if receiver clock speed is deviated

Description

If the USART receives an idle frame followed by a character, and the clock of the transmitter device is faster than the USART receiver clock, the USART receive signal falls too early when receiving the character start bit, with the result that the idle frame is not detected (IDLE flag is not set).

Workaround

None.

2.6.2 In full duplex mode, the Parity Error (PE) flag can be cleared by writing the data register

Description

In full duplex mode, when the Parity Error flag is set by the receiver at the end of a reception, it may be cleared while transmitting by reading the USART_SR register to check the TXE or TC flags and writing data in the data register.

Consequently, the software receiver can read the PE flag as '0' even if a parity error occurred.

Workaround

The Parity Error flag should be checked after the end of reception and before transmission.

2.6.3 Parity Error (PE) flag is not set when receiving in Mute mode using address mark detection

Description

The USART receiver is in Mute mode and is configured to exit the Mute mode using the address mark detection. When the USART receiver recognizes a valid address with a parity error, it exits the Mute mode without setting the Parity Error flag.

Workaround

None.

2.6.4 Break frame is transmitted regardless of nCTS input line status

Description

When CTS hardware flow control is enabled (CTSE = 1) and the Send Break bit (SBK) is set, the transmitter sends a break frame at the end of current transmission regardless of nCTS input line status.

Consequently, if an external receiver device is not ready to accept a frame, the transmitted break frame is lost.

Workaround

None.

2.6.5 nRTS signal abnormally driven low after a protocol violation

Description

When RTS hardware flow control is enabled, the nRTS signal goes high when a data is received. If this data was not read and a new data is sent to the USART (protocol violation), the nRTS signal goes back to low level at the end of this new data.

Consequently, the sender gets the wrong information that the USART is ready to receive further data.

On USART side, an overrun is detected which indicates that some data has been lost.

Workaround

Workarounds are required only if the other USART device violates the communication protocol which is not the case in most applications.

Two workarounds can be used:

- After data reception and before reading in the data in the data register, the software takes over the control of the nRTS signal as a GPIO and holds it high as long as needed. If the USART device is not ready, the software holds the nRTS pin high, and releases it when the device is ready to receive new data.
- The time required by the software to read the received data must always be lower than the duration of the second data reception. For example, this can be ensured by treating all the receptions by DMA mode.

2.7 SDIO peripheral limitations

2.7.1 SDIO hardware flow control

Description

When enabling hardware flow control by setting bit 14 of the SDIO_CLKCR register to '1',

glitches can occur on the SDIOCLK output clock resulting in wrong data being written to the SD/MMC card or to the SDIO device. As a consequence, a CRC error is returned to the SD/SDIO MMC host interface (DCRCFAIL bit set to '1' in SDIO_STA register).

Workaround

There is no workaround. Do not use hardware flow control. Overrun errors (Rx mode) and FIFO underrun (Tx mode) should be managed by the application software.

2.7.2 Wrong CCRCFAIL status after a response without CRC

Description

When a command is followed by a response which does not contain a CRC field, the CRC is calculated anyway. Consequently, after the SDIO command IO_SEND_OP_COND (CMD5), the CCRCFAIL bit of the SDIO_STA register is set.

Workaround

In this case the CCRCFAIL bit in the SDIO_STA register must be ignored by software. CCRCFAIL must be cleared by setting the CCRCFAILC bit in the SDIO_ICR register after reception of the response to the CMD5 command.

2.7.3 No underrun detected and wrong data transmission

Description

When a data transfer from SDIO host to card is on going and the HW Flow Control is disabled (bit 14 of the SDIO_CLKCR is not set) and an underrun condition occurs, the controller may transmit a wrong data without detecting that an underrun condition occurs if the following clock frequencies relationship is verified:
[3 x period(APBClock) + 3 x period(SDIOCLK)] >= (32 / (BusWidth)) x period(SDIO_CK)

Workaround

Ensure that the following clock frequencies relationship is verified:
[3 x period(APBClock) + 3 x period(SDIOCLK)] < (32 / (BusWidth)) x period(SDIO_CK)

by:

- Incrementing the APB frequency,
- or decreasing the transfer bandwidth,
- or reducing SDIO_CK frequency.

2.7.4 CE-ATA multiple write command and card busy signal management

Description

The CE-ATA card may inform the host that it is busy by driving low the SDIO_D0 line, two cycles after the transfer of a write command (RW_MULTIPLE_REGISTER or RW_MULTIPLE_BLOCK). When the card is in busy state, the host must not send any data until the BUSY signal is de-asserted (SDIO_D0 released by the card).
This condition is not respected if the data state machine leaves the IDLE state (Write operation programmed and started DTEN = 1, DTDIR = 0 in SDIO_DCTRL register and TXFIFOE = 0 in SDIO_STA register).

As consequence, the write transfer fails and the data lines are corrupted.

Workaround

After having sent the write command (RW_MULTIPLE_REGISTER or RW_MULTIPLE_BLOCK), the software must check that the card is not busy by polling the BSY bit of the ATA status register using FAST_IO (CMD39) command before enabling the data state machine.

2.8 ADC peripheral limitation

2.8.1 ADC accuracy lowered

Workaround

Some ADC accuracy characteristics have maximum errors increased. See updated table below.

Table 7. ADC accuracy(1)(2)

Symbol	Parameter	Test conditions	Min(3)	Typ	Max(3)	Unit
ET	Total unadjusted error	2.4 V ≤V_{DDA} ≤ 3.6 V 2.4 V ≤V_{REF+} ≤ 3.6 V f_{ADC} = 8 MHz, R_{AIN} = 50 Ω T_A = -40 to 105 °C	-	2	5	LSB
EO	Offset error		-	1	3	
EG	Gain error		-	1.5	3.5	
ED	Differential linearity error		-	1	2	
EL	Integral linearity error		-	1.7	3	
ENOB	Effective number of bits	2.4 V ≤V_{DDA} ≤ 3.6 V V_{DDA} = V_{REF+} f_{ADC} = 16 MHz, R_{AIN} = 50 Ω T_A = -40 to 105 °C 1 kHz ≤F_{input} ≤ 100 kHz	9.2	10	-	bits
SINAD	Signal-to-noise and distortion ratio		57.5	62	-	dB
SNR	Signal-to-noise ratio		57.5	62	-	
THD	Total harmonic distortion		-74	-75	-	
ET	Total unadjusted error	2.4 V ≤V_{DDA} ≤ 3.6 V 1.8 V ≤V_{REF+} ≤ 2.4 V f_{ADC} = 4 MHz, R_{AIN} = 50 Ω T_A = -40 to 105 °C	-	4	7	LSB
EO	Offset error		-	2	5	
EG	Gain error		-	4	6	
ED	Differential linearity error		-	1	2	
EL	Integral linearity error		-	1.5	4	
ET	Total unadjusted error	1.8 V ≤V_{DDA} ≤ 2.4 V 1.8 V ≤V_{REF+} ≤ 2.4 V f_{ADC} = 4 MHz, R_{AIN} = 50 Ω T_A = -40 to 105 °C	-	2	5	LSB
EO	Offset error		-	1	3	
EG	Gain error		-	1.5	2	
ED	Differential linearity error		-	1	2	
EL	Integral linearity error		-	1	3	

1. ADC DC accuracy values are measured after internal calibration.
2. ADC accuracy vs. negative injection current: Injecting a negative current on any analog input pins should be avoided as this significantly reduces the accuracy of the conversion being performed on another analog input. It is recommended to add a Schottky diode (pin to ground) to analog pins which may potentially inject negative currents.
   Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in *Section "I/O current injection characteristics" of relevant datasheet* does not affect the ADC accuracy.
3. Based on characterization, not tested in production.

Workaround

None.

2.9 FSMC peripheral limitation

2.9.1 FSMC NOR Flash/PSRAM controller asynchronous access on bank2..4 when bank1 is in synchronous mode (CBURSTRW bit is set)

Description

If the bank1 of NOR/PSRAM controller is enabled in synchronous write mode (CBURSTRW bit is set), while any other NOR/PSRAM bank 2..4 is enabled in asynchronous mode, there are two issues:

- For the first write access to the asynchronous memory, the byte lane NBL[1:0] is not active (kept high).
- A write access to an asynchronous memory with Extended feature enabled, the system hangs without any fault generation. The two issues occur only when the NOR/PSRAM bank1 is configured in synchronous write mode (CBURSTRW bit is set).

Workaround

If multiple banks are enabled with mixed asynchronous and synchronous write access, use any NOR/PSRAM bank for synchronous write access except bank1.

2.9.2 FSMC Synchronous mode and disabled NWAIT signal

Description

In case of FSMC synchronous mode with the NWAIT signal disabled, if the polarity (WAITPOL in the FSMC_BCRx register)of the NWAIT signal is same as NWAIT input signal level, the system hangs and no fault generated. Only system reset will recover.

Workaround

PD6 (NWAIT signal) must not be connected to AF12 and configure the NWAIT polarity to active high (set WAITPOL bit to 1 in FSMC_BCRx register).

2.9.3 Dummy read cycles inserted when reading synchronous memories

Description

When performing a burst read access to a synchronous memory, two dummy read accesses are performed at the end of the burst cycle whatever the type of AHB burst access. However, the extra data values which are read are not used by the FSMC and there is no functional failure.

Workaround

none.

Appendix A Revision and date codes on device marking

Figure 1, *Figure 2*, *Figure 3*, *Figure 4*, *Figure 5* and *Figure 6* show the marking compositions for the UFBGA132/UFBGA100, LFP144, LQFP100, LQFP64, WLCSP64/63L and LQFP48/UFQFPN48 packages, respectively. The only fields shown are the “additional” field, containing the revision code, and the “year” and “week” fields making up the date code.

Figure 1. UFBGA132/UFBGA100 top package view

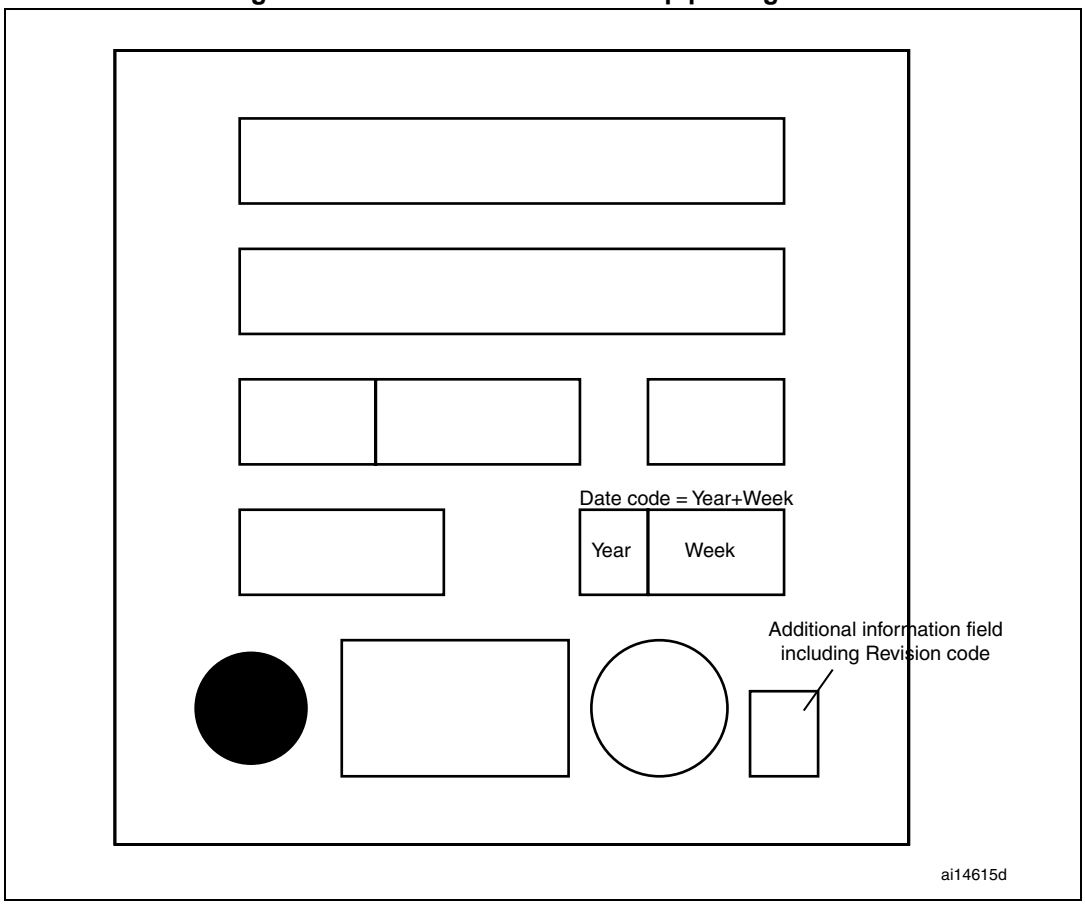


Figure 2. LQFP144 top package view

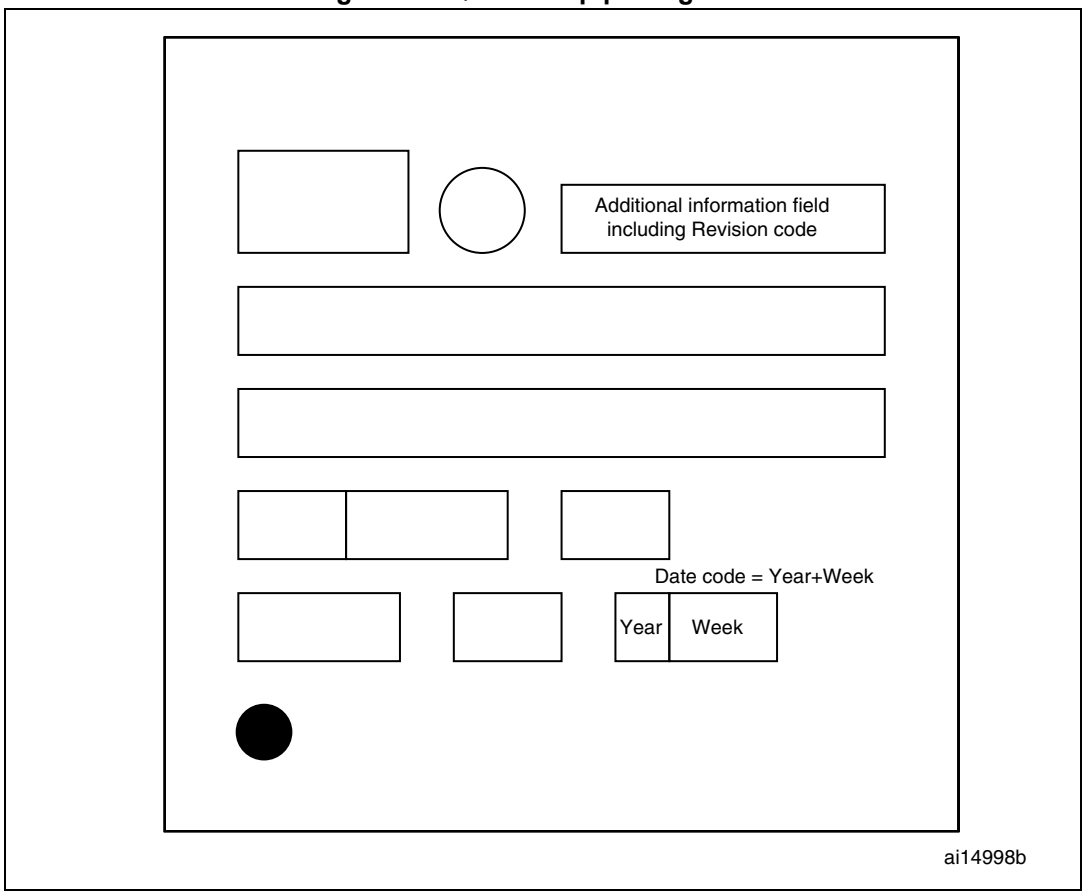


Figure 3. LQFP100 top package view

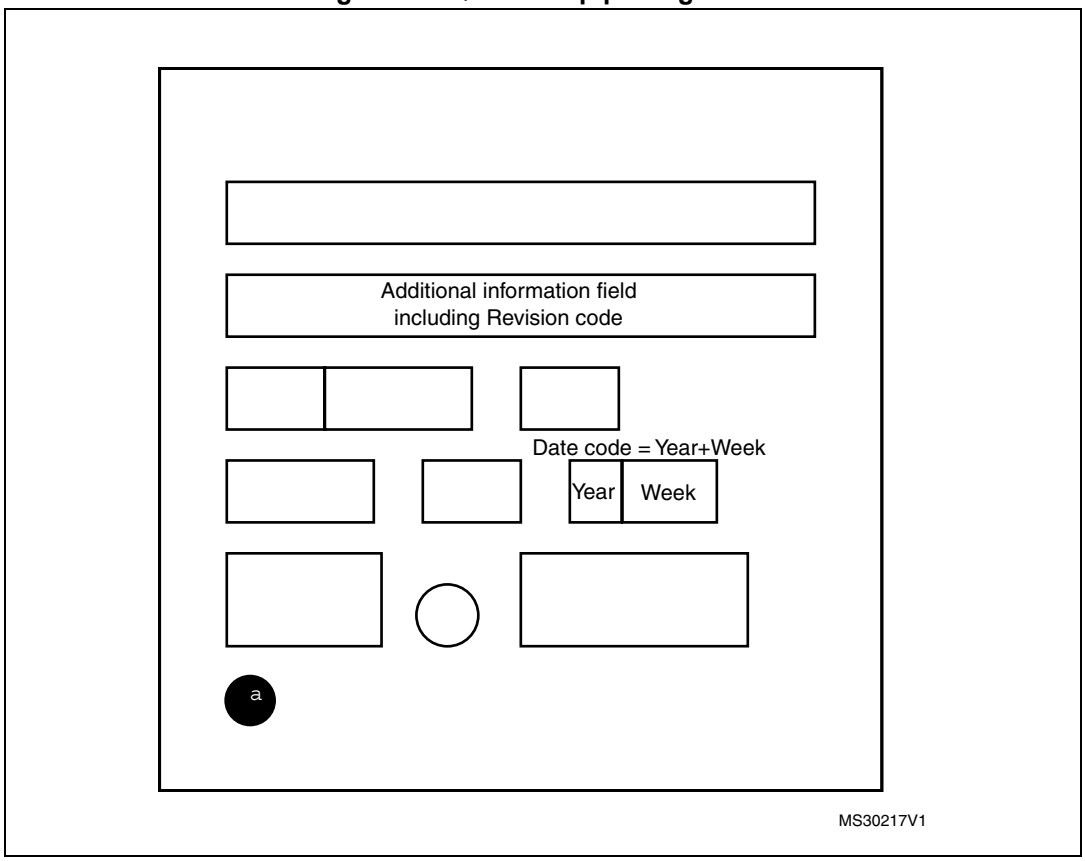


Figure 4. LQFP64 top package view

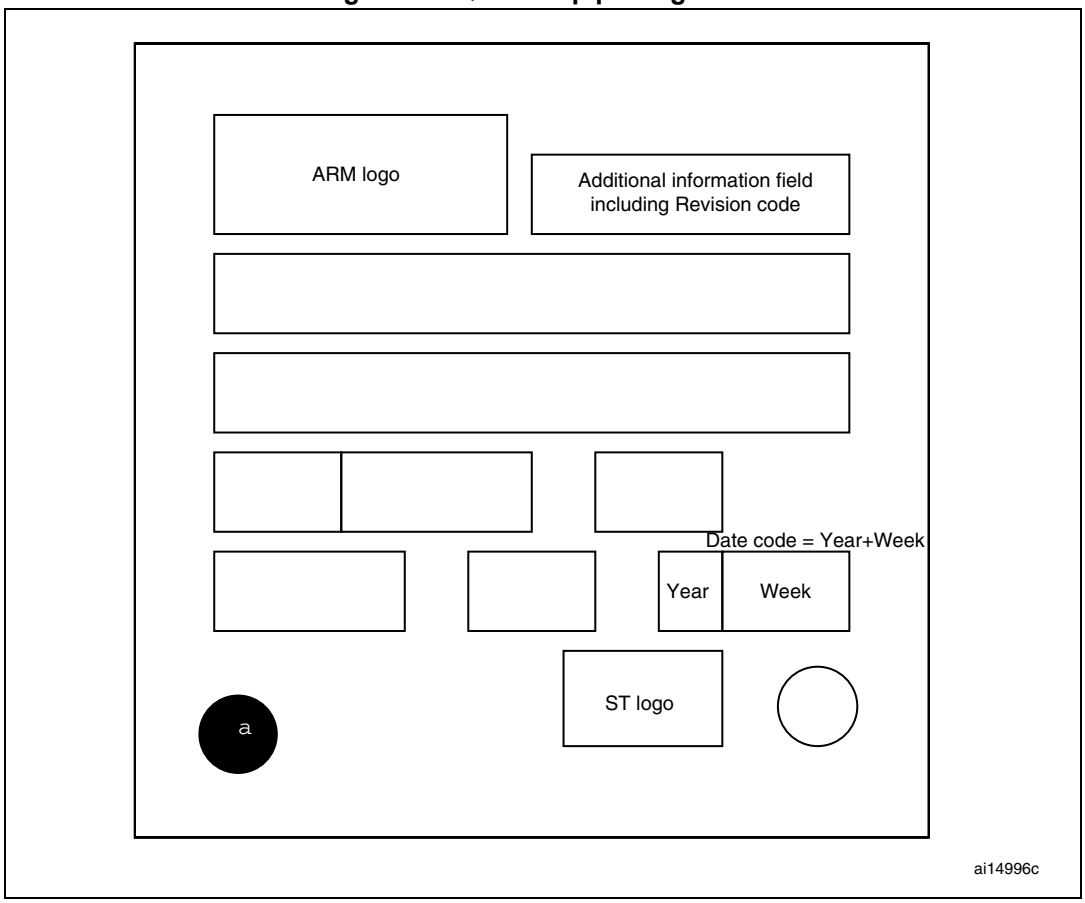


Figure 5. WLCSP64 /WLCSP63L top package view

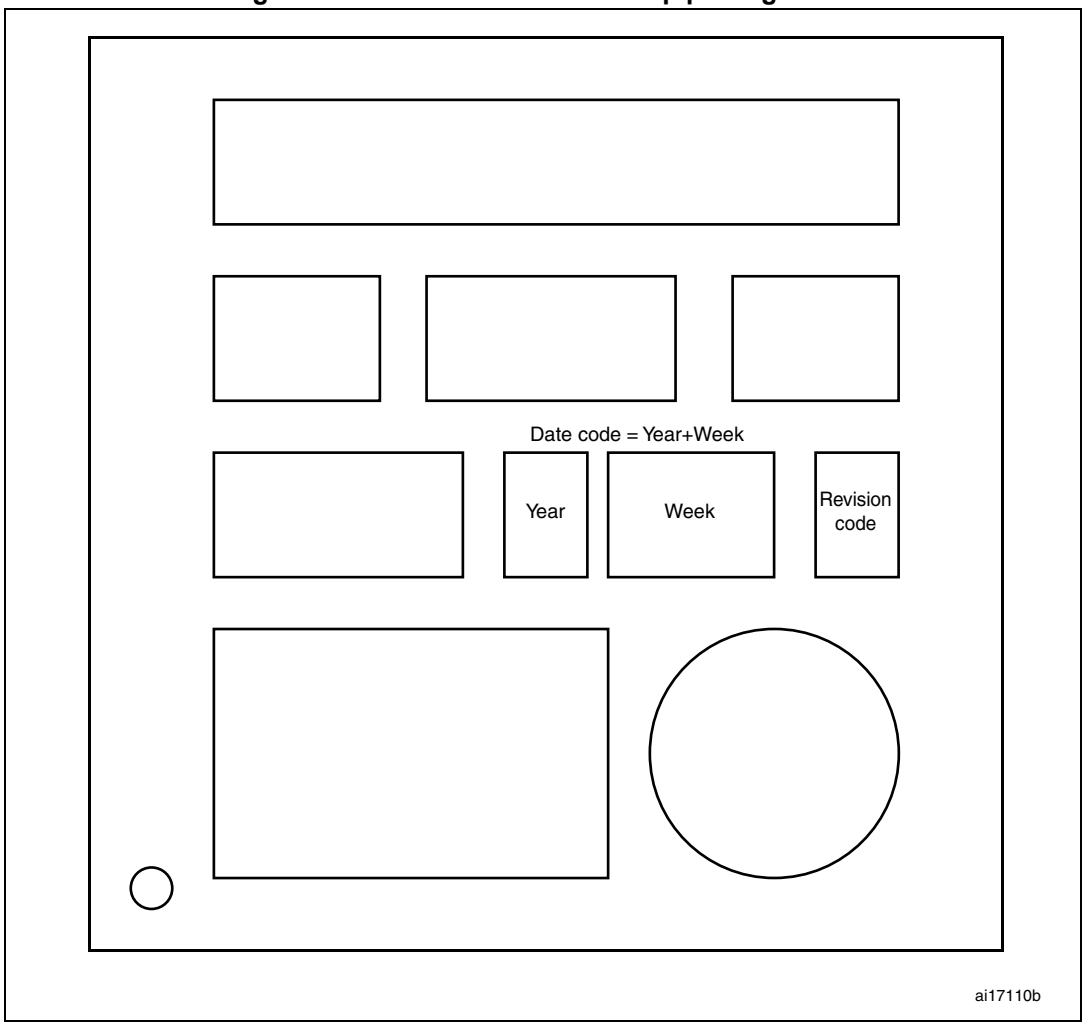
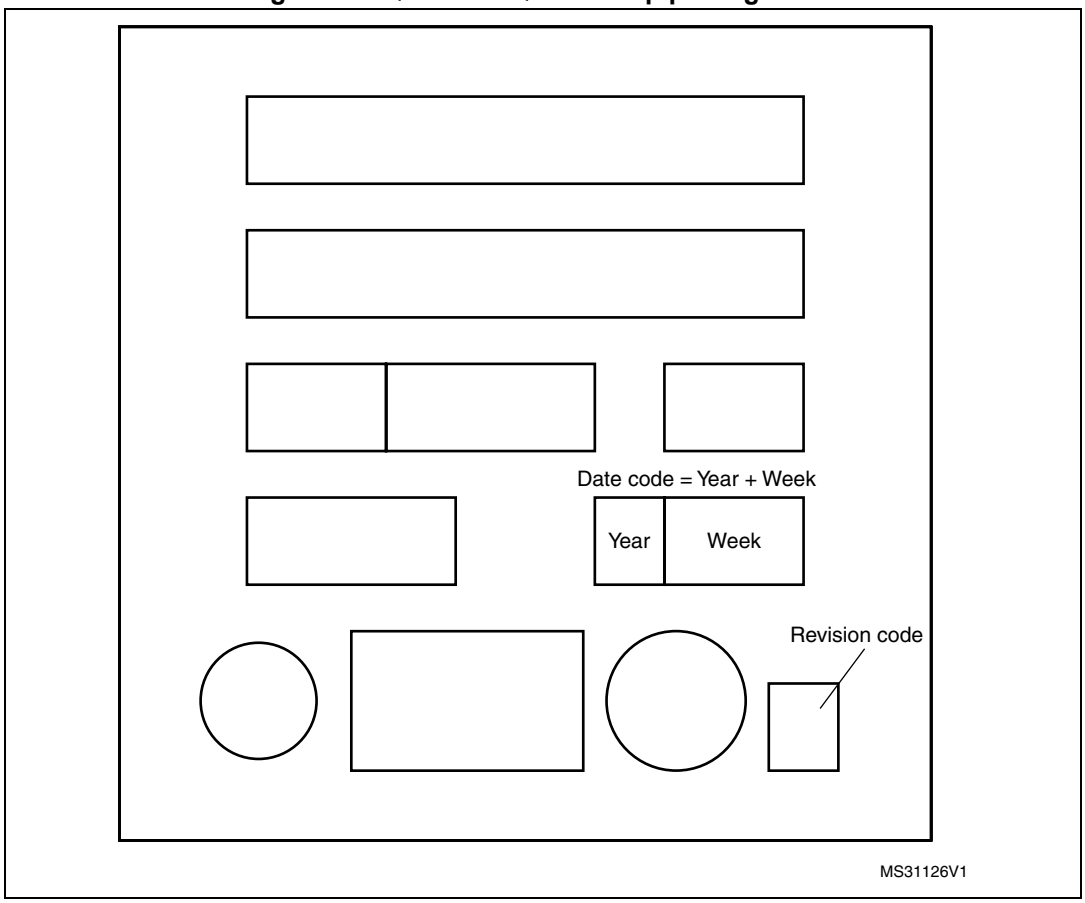


Figure 6. LQFP48/UFQFPN48 top package view



Revision history

Table 8. Document revision history

Date	Revision	Changes
14-Sep-2011	1	Initial release.
18-Jan-2012	2	Updated *Table 4: Summary of silicon limitations* for silicon Rev Z. Added *Section 2.4.1: SMBus standard not fully supported* *Section 2.4.3: Violation of I2C "setup time for repeated START condition" parameter* *Section 2.4.4: In I2C slave "NOSTRETCH" mode, underrun errors may not be detected and may generate bus errors* *Section 2.7.2: Wrong CCRCFAIL status after a response without CRC*
27-Nov-2012	3	Added – Limitations for STM32L162xC/D – *Section 2.1.9: Boot from bank2 limitation* – *Section 2.1.12: Electrostatic discharge limits are 1kV (HBM) and 250 V (CDM) instead of 2 kV and 500 V respectively* – *Section 2.8.1: ADC accuracy lowered* Updated – *Appendix A: Revision and date codes on device marking* Deleted – *Section 2.1.12: Data in Flash size register is not accurate*
26-Feb-2013	4	Added – *Section 1.1.3: Cortex-M3 Interrupted loads to the stack-pointer can cause erroneous behavior* – *Section 1.1.4: SVC and BusFault/MemManage may occur out of order* – *Section 2.1.13: Pull-up on PB7 when configured in analog mode* – *Section 2.1.14: Data EEPROM cycling limited to 100 kcycles* – *Section 2.3.1: RVU and PVU flags are not reset in STOP mode* – *Section 2.6.5: nRTS signal abnormally driven low after a protocol violation* – *Section 2.7.3: No underrun detected and wrong data transmission* – *Section 2.7.4: CE-ATA multiple write command and card busy signal management* – *Section 2.9.1: FSMC NOR Flash/PSRAM controller asynchronous access on bank2..4 when bank1 is in synchronous mode (CBURSTRW bit is set)* – *Section 2.9.2: FSMC Synchronous mode and disabled NWAIT signal* – *Section 2.9.3: Dummy read cycles inserted when reading synchronous memories*

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.

ST PRODUCTS ARE NOT AUTHORIZED FOR USE IN WEAPONS. NOR ARE ST PRODUCTS DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER’S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR “AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL” INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

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.
The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

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

www.st.com