

# Errata

## DA1469x Errata

### CE0001

#### Abstract

*The document contains the known errata for the DA1469x along with recommended workarounds.*

<b>Product</b>	DA1469x
<b>Silicon Revision</b>	0xAB (Register CHIP_REVISION_REG and CHIP_TEST1_REG)
<b>Datasheet Reference</b>	DA1469x_Datasheet.pdf v2.0
<b>Package(s)</b>	VFBGA86 (86 balls) 6 mm x 6 mm VFBGA100 (100 balls) 5 mm x 5 mm
<b>Issue Date</b>	22-Feb-2022

---

## Contents

<b>Abstract</b> .....	<b>1</b>
<b>Contents</b> .....	<b>2</b>
<b>1 Terms and Definitions</b> .....	<b>7</b>
<b>2 Identifying the Silicon Revision</b> .....	<b>7</b>
<b>3 Workarounds</b> .....	<b>7</b>
<b>4 Errata Summary</b> .....	<b>7</b>
<b>5 Errata Details</b> .....	<b>10</b>
5.1 GP ADC LDO Sampling affects Gain Error.....	10
5.1.1 Effect.....	10
5.1.2 Conditions .....	10
5.1.3 Technical Description .....	10
5.1.4 Workaround .....	10
5.2 Latched RSSI not Retained after DEM_EN is Cleared .....	10
5.2.1 Effect.....	10
5.2.2 Conditions .....	10
5.2.3 Technical Description .....	10
5.2.4 Workaround .....	10
5.3 SPI Needs Proper Reset Sequence .....	10
5.3.1 Effect.....	10
5.3.2 Conditions .....	11
5.3.3 Technical Description .....	11
5.3.4 Workaround .....	11
5.4 CM_CLK_COMP_REG Read Value is Wrong.....	11
5.4.1 Effect.....	11
5.4.2 Conditions .....	11
5.4.3 Technical Description .....	11
5.4.4 Workaround .....	11
5.5 Spike on Input Could Generate a Duplicate KEYBR_IRQ.....	11
5.5.1 Effect.....	11
5.5.2 Conditions .....	11
5.5.3 Technical Description .....	12
5.5.4 Workaround .....	12
5.6 CM_TS1_REG Does Not Implement Byte Select During AHB Write .....	12
5.6.1 Effect.....	12
5.6.2 Conditions .....	12
5.6.3 Technical Description .....	12
5.6.4 Workaround .....	12
5.7 LCD Controller Stride Issue .....	12
5.7.1 Effect.....	12
5.7.2 Conditions .....	12
5.7.3 Technical Description .....	12

## DA1469x Errata

5.7.4	Workaround .....	13
5.8	SRC Startup Fails Sometimes in Auto Mode .....	13
5.8.1	Effect.....	13
5.8.2	Conditions .....	13
5.8.3	Technical Description .....	13
5.8.4	Workaround .....	13
5.9	$\Sigma\Delta$ ADC, gain error related to OSR.....	13
5.9.1	Effect.....	13
5.9.2	Conditions .....	13
5.9.3	Technical Description .....	13
5.9.4	Workaround .....	13
5.10	SRC Generates False Overflows During PDM Mode .....	14
5.10.1	Effect.....	14
5.10.2	Conditions .....	14
5.10.3	Technical Description .....	14
5.10.4	Workaround .....	14
5.11	OTP Leakage from VDD to VDD2 .....	14
5.11.1	Effect.....	14
5.11.2	Conditions .....	14
5.11.3	Technical Description .....	14
5.11.4	Workaround .....	14
5.12	Charger Detect Circuit Erroneous if V30 Setting is not at 3.3 V .....	15
5.12.1	Effect.....	15
5.12.2	Conditions .....	15
5.12.3	Technical Description .....	15
5.12.4	Workaround .....	15
5.13	ADC pstrb[0] not Used for Generating adc_start_i .....	15
5.13.1	Effect.....	15
5.13.2	Conditions .....	15
5.13.3	Technical Description .....	15
5.13.4	Workaround .....	15
5.14	Motor Controller's Read Pointer is Reset by HW Reset Instead of POR.....	15
5.14.1	Effect.....	15
5.14.2	Conditions .....	16
5.14.3	Technical Description .....	16
5.14.4	Workaround .....	16
5.15	$\Sigma\Delta$ ADC, First Conversion is Erroneous after Switching SE <-> DIF .....	16
5.15.1	Effect.....	16
5.15.2	Conditions .....	16
5.15.3	Technical Description .....	16
5.15.4	Workaround .....	16
5.16	LCD_SPI_DC not Aligned Correctly with the LCD_SPI_CLK.....	16
5.16.1	Effect.....	16
5.16.2	Conditions .....	16
5.16.3	Technical Description .....	17

---

**DA1469x Errata**

5.16.4	Workaround .....	17
5.17	SRC Output Underflow for 1.5 MHz PDM Input with High PCM Output Sample Rates .....	17
5.17.1	Effect.....	17
5.17.2	Conditions .....	17
5.17.3	Technical Description .....	17
5.17.4	Workaround .....	17
5.18	LCDC: RGB111-4 when Used in DBI-SPI I/F .....	17
5.18.1	Effect.....	17
5.18.2	Conditions .....	17
5.18.3	Technical Description .....	18
5.18.4	Workaround .....	18
5.19	XTAL32M Oscillator: Lowest Capbank Setting .....	18
5.19.1	Effect.....	18
5.19.2	Conditions .....	18
5.19.3	Technical Description .....	18
5.19.4	Workaround .....	18
5.20	Timer on System or TIM PD Counts +2 Every Time System Wakes up and -2 when Goes to Extend Sleep .....	18
5.20.1	Effect.....	18
5.20.2	Conditions .....	19
5.20.3	Technical Description .....	19
5.20.4	Workaround .....	19
5.21	RFMON will Capture an Extra Word under Specific Conditions .....	19
5.21.1	Effect.....	19
5.21.2	Conditions .....	19
5.21.3	Technical Description .....	19
5.21.4	Workaround .....	19
5.22	CMAC, BS Stop During Sampling Point Calculation .....	19
5.22.1	Effect.....	19
5.22.2	Conditions .....	20
5.22.3	Technical Description .....	20
5.22.4	Workaround .....	20
5.23	Overload on V30 Rail can Cause a System Lock Up .....	20
5.23.1	Effect.....	20
5.23.2	Conditions .....	20
5.23.3	Technical Description .....	20
5.23.4	Workaround .....	20
5.24	CACHE Fetch During a 'Miss' is not Stalling the Sleep Process .....	20
5.24.1	Effect.....	20
5.24.2	Conditions .....	20
5.24.3	Technical Description .....	20
5.24.4	Workaround .....	21
5.25	Charger: I <sub>charge</sub> Increases at Lower VBUS.....	21
5.25.1	Effect.....	21
5.25.2	Conditions .....	21
5.25.3	Technical Description .....	21

## DA1469x Errata

5.25.4	Workaround .....	21
5.26	V18F Switch Resistance too High.....	21
5.26.1	Effect.....	21
5.26.2	Conditions .....	21
5.26.3	Technical Description .....	22
5.26.4	Workaround .....	22
5.27	CMAC, BS DMA doesn't support big endian .....	22
5.27.1	Effect.....	22
5.27.2	Conditions .....	22
5.27.3	Technical Description .....	22
5.27.4	Workaround .....	22
5.28	$\Sigma$ ADC: Insufficient Settling Time when VBAT Scaler is Used .....	22
5.28.1	Effect.....	22
5.28.2	Conditions .....	22
5.28.3	Technical Description .....	22
5.28.4	Workaround .....	22
5.29	LCDC : Wrong Partial Refresh.....	23
5.29.1	Effect.....	23
5.29.2	Conditions .....	23
5.29.3	Technical Description .....	23
5.29.4	Workaround .....	23
5.30	BOD doesn't function at low supply .....	23
5.30.1	Effect.....	23
5.30.2	Conditions .....	23
5.30.3	Technical Description .....	23
5.30.4	Workaround .....	23
5.31	XTAL32M oscillator: Track and Hold timing.....	23
5.31.1	Effect.....	23
5.31.2	Conditions .....	24
5.31.3	Technical Description .....	24
5.31.4	Workaround .....	24
5.32	XTAL32M oscillator: Minimum OFF Time .....	24
5.32.1	Effect.....	24
5.32.2	Conditions .....	24
5.32.3	Technical Description .....	24
5.32.4	Workaround .....	24
5.33	Charger CC-CV Comparator Hysteresis.....	25
5.33.1	Effect.....	25
5.33.2	Conditions .....	25
5.33.3	Technical Description .....	25
5.33.4	Workaround .....	25
5.34	RF Variable Vref Decreases with Low VDDD .....	25
5.34.1	Effect.....	25
5.34.2	Conditions .....	25
5.34.3	Technical Description .....	25

---

DA1469x Errata

5.34.4	Workaround .....	26
5.35	PLL calibration does not work properly .....	26
5.35.1	Effect.....	26
5.35.2	Conditions .....	26
5.35.3	Technical Description .....	26
5.35.4	Workaround .....	26
<b>Revision History</b>	.....	<b>27</b>

## 1 Terms and Definitions

SRC	Sample Rate Converter
ADC	Analog to Digital Converter
LDO	Low Drop-Out Voltage regulator
OTP	One Time Programmable Memory
OSR	Over Sampling Rate
LCDC	LCD Controller
RFMON	RF Monitoring Controller
BS	Bit-Stream Controller
BOD	Brown-Out Detection Controller
CC-CV	Constant Current – Constant Voltage
BLE	Bluetooth Low Energy
RFCU	RF Control Unit
CMAC	Configurable MAC (BLE)
RSSI	Received Signal Strength Indication
CM0+	ARM Cortex® M0+
SDK	Software Development Kit

## 2 Identifying the Silicon Revision

A readback of registers CHIP\_REVISION\_REG and CHIP\_TEST1\_REG will give the device revision information.

## 3 Workarounds

For the cases where complex software workarounds are recommended in form of concept description, please refer to the special section of the SDK release notes that explains which items are addressed and how.

## 4 Errata Summary

**Table 1: Errata Summary**

273	GPADC LDO Sampling affects Gain Error
274	Latched RSSI not retained after DEM_EN is cleared
275	SPI needs proper reset sequence
276	CM_CLK_COMP_REG read value is wrong
241	Spike on input maybe generate a duplicate KEYBR_IRQ
277	CM_TS1_REG does not implement byte select during AHB write
278	LCD controller stride issue

## DA1469x Errata

243	SRC startup fails sometimes in auto mode
279	$\Sigma$ ADC, gain error related to OSR
242	SRC generates false overflows during PDM mode
280	OTP leakage from VDD to VDD2
281	Charger Detect Circuit erroneous if V30 setting not at 3.3V
282	ADC pstrb[0] not used for generating (ds)adc_start_i
283	Motor controller's read pointer is reset by HW reset instead of POR
284	$\Sigma$ ADC, first conversion is erroneous after switching SE <-> DIF
285	LCD_SPI_DC not aligned correctly with the LCD_SPI_CLK
249	SRC Output underflow for 1.5 MHz PDM input with high PCM output sample rates
286	LCDC: RGB111-4 when is used in DBI-SPI I/F
287	XTAL32M Oscillator: Lowest capbank setting
288	Timer on AON or TIM PD counts plus 2 every time system wakes up and minus 2 when goes to extend sleep
289	RFMON will capture an extra word under specific conditions
290	CMAC, BS stop during sampling point calculation
291	Overload on V30 rail can cause a system lock up
292	CACHE fetch during a 'miss' is not stalling the sleep process
293	Charger: Icharge increases at lower VBUS
294	V18F switch resistance too high
295	CMAC, BS DMA doesn't support big endian
296	$\Sigma$ ADC: Insufficient settling time when VBAT scaler is used
297	LCDC: wrong partial refresh
298	BOD doesn't function at low supply
299	XTAL32M oscillator: Track and Hold timing
300	XTAL32M oscillator: Minimum OFF time
302	Charger CC-CV comparator hysteresis
303	RF Variable Vref decreases with low VDDD



304	PLL calibration does not work properly

## 5 Errata Details

### 5.1 GP ADC LDO Sampling affects Gain Error

#### 5.1.1 Effect

Increased gain error.

#### 5.1.2 Conditions

When the internal ADC LDO is put into sample and hold mode.

#### 5.1.3 Technical Description

When the internal ADC LDO is put into sample and hold mode, a small offset is added (~10 mV) which causes an additional gain error to the ADC.

#### 5.1.4 Workaround

Not use LDO sampling. Keep GP\_ADC\_LDO\_ZERO=0.

### 5.2 Latched RSSI not Retained after DEM\_EN is Cleared

#### 5.2.1 Effect

Keeping the RSSI value is needed for two reasons:

1. To provide the RSSI after a Correlation Timeout, which is needed by the Bad Channel detection algorithm.
2. To offload the CMAC FW Field Callback functions, by performing the RSSI read during the CRC check.

This is especially critical for #1, since the algorithm always gets RSSI=0, reducing the detection capability.

#### 5.2.2 Conditions

Always

#### 5.2.3 Technical Description

The latched RSSI value is cleared after DEM\_EN (Enable of Demodulator) is cleared, so it cannot be used by Firmware.

#### 5.2.4 Workaround

Read the latched RSSI and save it to a variable on the spot. For example, while the header is being processed.

### 5.3 SPI Needs Proper Reset Sequence

#### 5.3.1 Effect

SPI block is not operating correctly because the reset/initialization is not set correctly.

**DA1469x Errata**
**5.3.2 Conditions**

If SPI\_ON bit is expected to reset the SPI block.

**5.3.3 Technical Description**

When SPI\_ON=0, SPI does not return to the reset state. SPI\_ON will only turn off the internal SPI clock to reduce power dissipation. However, internal logic needs a clock to be properly reset.

**5.3.4 Workaround**

SPI needs the following commands sequence for proper configuration:

```
SetBits32(SPI_CTRL_REG, SPI_ON, 0);
SetBits32(SPI_CTRL_REG, SPI_RST, 1);
// Update SPI configuration
SetBits32(SPI_CTRL_REG, SPI_ON, 1);
SetBits32(SPI_CTRL_REG, SPI_RST, 0);
SetBits32(SPI_CTRL_REG, SPI_MINT, x); // x: 0 for masked IRQ, 1 for unmasked
IRQ
```

**5.4 CM\_CLK\_COMP\_REG Read Value is Wrong**
**5.4.1 Effect**

Reading register CM\_CLK\_COMP\_REG via CM0+, bit order is mixed.

**5.4.2 Conditions**

Always

**5.4.3 Technical Description**

Incorrect connection of the register port to the AHB bus. When read, this 7-bit wide register will map to the bus as follows:

```
Read_bus_bit[0] <= Register_bit[6]
Read_bus_bit[6:1] <= Register_bit[5:0]
```

**5.4.4 Workaround**

Use the value of this register's bitfields correctly after read.

**5.5 Spike on Input Could Generate a Duplicate KEYBR\_IRQ**
**5.5.1 Effect**

A false spike generated when a button is kept pressed, will generate a duplicate KEYBR\_IRQ.

**5.5.2 Conditions**

Always (if short debounce times are programmed compared the required ones).

**DA1469x Errata**
**5.5.3 Technical Description**

There is a mistake on the digital FSM. It will erroneously go through the KEY\_PRESSED state again, if the key is still pressed after the debounce time has elapsed. Keyboard state machine starts debouncing when it sees a valid edge. After debounce time, it checks if the level is still active, and issues an interrupt.

This happens if the input signal needs a long debounce time, but the user has programmed a short debounce time.

**5.5.4 Workaround**

None

**5.6 CM\_TS1\_REG Does Not Implement Byte Select During AHB Write**
**5.6.1 Effect**

Byte write on this register will affect the complete register (32-bits), not just the required bytes.

**5.6.2 Conditions**

Always

**5.6.3 Technical Description**

During byte write AHB transactions by the CM0+ on this register, bytes not touched will not be protected.

**5.6.4 Workaround**

Always perform 32-bit write access.

**5.7 LCD Controller Stride Issue**
**5.7.1 Effect**

Visible artifacts on the LDC display

**5.7.2 Conditions**

Changing stride lengths while in continuous mode displaying of the LCD controller.

**5.7.3 Technical Description**

Most of the layer registers (LCDC\_LAYER0\_STARTXY\_REG, LCDC\_LAYER0\_SIZEXY\_REG, LCDC\_LAYER0\_BASEADDR\_REG, LCDC\_LAYER0\_RESXY\_REG and LCDC\_LAYER0\_OFFSETX\_REG) are protected from being changed at an inappropriate time. On the contrary, LCDC\_LAYER0\_MODE\_REG and LCDC\_LAYER0\_STRIDE\_REG are not. The protection mechanism applies to the input values only, when a frame is completed, or when there is a write to LCDC\_MODE\_REG.

In case of single frame update, LCDC\_MODE\_REG is written in every frame and as a result, all layer parameters are applied immediately.

On the other hand, in case of continuous update, LCDC\_MODE\_REG is only accessed once to setup the mode, and as a result, the new layer settings are applied with the frame end signal. This results in the stride taking effect in the next output frame, while remaining parameters on the second output frame create a discrepancy and eventually artifacts in the LCD.

**DA1469x Errata**
**5.7.4 Workaround**

Forcefully apply the protected registers by performing a dummy write to the LCDC\_MODE\_REG.

**5.8 SRC Startup Fails Sometimes in Auto Mode**
**5.8.1 Effect**

Generating consistent FIFO underflows and overflows.

**5.8.2 Conditions**

This happens when input or output is set to automatic mode.

**5.8.3 Technical Description**

False edge detection of incoming sync signal during startup (high level on sync), results in incorrect resampling ratio being acquired during startup.

**5.8.4 Workaround**

1. Monitor under/overflows and reset the SRC manually if needed. To do this, disable then enable the SRC continually until under/overflows are no longer generated.
2. Synchronize the startup of the SRC with the sample events, by enabling the SRC in a sample-based interrupt (using the input or output side that is in automatic mode). This will prevent an incorrect resampling ratio being acquired. Note: This is only applicable to A2M or M2A.
3. Avoid auto mode, use manual with external control instead. The external SW loop can use the PHASE\_INFO register to adjust the FS.

**5.9  $\Sigma\Delta$  ADC, gain error related to OSR**
**5.9.1 Effect**

Increased gain error

**5.9.2 Conditions**

Always

**5.9.3 Technical Description**

The digital filter integrates M samples, but decimates for only M-1 samples. This causes a gain error that varies with M.  $M = \text{OSR}$  (Over Sampling Rate).

As a result, the gain error of the ADC varies with control settings and typically calibration needs to be redone every time a different OSR is used. Calibration needs an external reference and therefore most likely cannot be done on chip at runtime.

**5.9.4 Workaround**

The following sequence would sketch a workaround:

- Measure the ADC gain error with maximum OSR (1024)
- then subtract the theoretical gain error (0.20% for OSR=1024) to define a base gain error.
- Finally add the theoretical gain error for the configured OSR (0.20% for OSR=1024, 0.40% for OSR=512, 0.81% for OSR=256, 1.69% for OSR=128)

## 5.10 SRC Generates False Overflows During PDM Mode

### 5.10.1 Effect

FIFO overflows when in PDM input mode

### 5.10.2 Conditions

Always

### 5.10.3 Technical Description

There is a mistake in the comparison circuit that will always assert the overflow bit when the SRC is in PDM or DSD modes.

### 5.10.4 Workaround

Ignore the overflow bit when in SRC or DSD modes.

## 5.11 OTP Leakage from VDD to VDD2

### 5.11.1 Effect

Measuring 0.7 mA extra current at PDOWN state versus DSTBY state, while the opposite would be expected.

### 5.11.2 Conditions

Always

### 5.11.3 Technical Description

The OPT cell has two supplies: namely, VDD and VDD2. VDD is connected to the core voltage. The V30 supplies VDD2 through a power switch. The power switch is controlled by a signal coming from the OTP Controller (otp\_vdd2\_en). The VDD2 is disconnected from V30 when the mode in the controller is PDOWN, but VDD is still powered. Under these conditions, there is a current leaking from VDD to VDD2.

### 5.11.4 Workaround

The PDOWN mode of the controller is the same with the DSTBY, with the only difference that the VDD2 is not powered at the PDOWN mode. The suggested workaround is to use the DSTBY instead of the (default) PDOWN.

```
#define OTPC_MODE_DSTBY 0x1
volatile int i;
// Enable the OTP clock
SetBits32 (CLK_AMBA_REG, OTP_ENABLE, 1);
// Wait for a while, until the clock to be enabled
for (i=0;i<10;i++);
// Set the mode to the DSTBY
SetWord32 (OTPC_MODE_REG, OTPC_MODE_DSTBY );
// Wait until the mode to be enabled
```

**DA1469x Errata**

```
while (GetBits32(OTPC_STAT_REG, OTPC_STAT_MRDY) != 1);
// Disable the OTP clock
SetBits32 (CLK_AMBA_REG, OTP_ENABLE, 0);
```

**5.12 Charger Detect Circuit Erroneous if V30 Setting is not at 3.3 V**
**5.12.1 Effect**

Charge detection operation not working as expected.

**5.12.2 Conditions**

If V30 setting is 3 V (POWER\_CTRL\_REG[V30\_LEVEL]=0x0, default)

**5.12.3 Technical Description**

In this block, there is one 0.6 V and one 2.4 V reference. They are derived from V30 supply using a resistive divider, assuming the supply be 3.3 V. However, the default supply is 3 V. This causes the reference voltage to be roughly 10% off the target, when using a 3 V supply.

**5.12.4 Workaround**

Use the V30 with a setting of 3.3 V (POWER\_CTRL\_REG[V30\_LEVEL]=0x2)

**5.13 ADC pstrb[0] not Used for Generating adc\_start\_i**
**5.13.1 Effect**

Byte access of the GP\_ADC\_START or SDADC\_START bitfields might destroy running ADC conversions.

**5.13.2 Conditions**

Always

**5.13.3 Technical Description**

A separate process is used to detect an APB write towards GP\_ADC\_CTRL\_REG[GP\_ADC\_START] (GPADC) and SDADC\_CTRL\_REG[SDADC\_START] (SDADC). This will go wrong when the lower byte isn't addressed (pstrb[0]=0) during the APB write. In that case the internal start\_i updates when it should not. This can destroy an ongoing conversion, or unintendedly start a new one.

**5.13.4 Workaround**

Address all bytes when writing towards the GP\_ADC\_CTRL\_REG and SDADC\_CTRL\_REG.

**5.14 Motor Controller's Read Pointer is Reset by HW Reset Instead of POR**
**5.14.1 Effect**

Pointer to the commands sequence in RAM which controls the Motor Pattern Generators is lost after a HW reset.

**DA1469x Errata**
**5.14.2 Conditions**

Always

**5.14.3 Technical Description**

The motor controller's read pointer should only be reset by POR, but this also happens for HW reset. The register has a separate reset, but is connected to the peripheral domain's retention reset signal, which is also triggered by HW reset.

**5.14.4 Workaround**

None

**5.15 ΣΔADC, First Conversion is Erroneous after Switching SE <-> DIF**
**5.15.1 Effect**

First ΣΔ ADC conversion is wrong.

**5.15.2 Conditions**

When switching from Single Ended to Differential mode and vice-versa.

**5.15.3 Technical Description**

When switching from Differential to Single-Ended and vice versa, the SE bit from Digital to Analogue is changed at the same time as releasing the Analogue from reset. This causes spikes/glitches and settling behavior. As a result, the analogue modulator makes a few incorrect decisions during the first few cycles, causing an error to be integrated. A higher OSR will correct the error more than a lower OSR. But for OSR=1024x (max OSR), the error can still be as large as 50 LSB (for OSR=128x this is about 400 LSB).

**5.15.4 Workaround**

Every time the input mode is changed from single-ended (SE) to differential (DIFF) or vice-versa, the ΣΔ ADC block should be disabled and re-enabled by toggling SDADC\_CTRL\_REG[SDADC\_EN]

**5.16 LCD\_SPI\_DC not Aligned Correctly with the LCD\_SPI\_CLK**
**5.16.1 Effect**

Visual artifacts on the LDC display.

**5.16.2 Conditions**

This issue happens under the following conditions:

- The LCD interface selected is SPI4
- A number of commands and command parameters are registered in the command/data FIFO of the DBIB module (through the LCDC\_DBIB\_CMD\_REG register)
- The commands/parameters are registered in the FIFO, without waiting for the transmission of each byte, but only by checking that there is enough room in the FIFO (LCDC\_STATUS\_REG[LCDC\_DBIB\_CMD\_FIFO\_FULL]). Thus, FIFO may contain commands and command parameters interleaved between them



**DA1469x Errata**
**5.16.3 Technical Description**

In this way, the information that was pushed to the FIFO, is transferred by the controller performing a long burst access in the LCD SPI interface. For each byte transferred, the LCDC provides the corresponding value in the LCD\_SPI\_DC line, according to what has been registered in the FIFO.

Under the above conditions, the LCD Controller updates the value of the LCD\_SPI\_DC line, and applies the value that belongs to the next byte, at the 8-th rising edge of the LCD\_SPI\_CLK clock, which is the last clock of the current byte. However, there are LCD devices with controllers like the ST7789V and the ST7735S that capture the level of the LCD\_SPI\_DC line at this specific clock edge (8-th rising clock edge of each byte). As a result, the LCDC\_SPI\_DC line cannot be captured correctly by the external LCD device due to setup/hold violations. The commands / command parameters cannot be recognized correctly with impact in the visualized image in the LCD.

**5.16.4 Workaround**

This issue can be avoided if the command/data FIFO of the DBIB is not filled with more than one element. The software snippet should be as follows:

```
//make sure command is idle
while(GetWord32(LCDC_STATUS_REG) & LCDC_DBIB_CMD_PENDING);
SetWord32(LCDC_DBIB_CMD_REG, element);
```

In this way the LCD\_SPI\_DC line will stay stable at the 8th clock cycle. Side effect will be a slower execution of the software, due to waiting for the transmission of each byte in the LCD SPI, before sending the next.

**5.17 SRC Output Underflow for 1.5 MHz PDM Input with High PCM Output Sample Rates**
**5.17.1 Effect**

Output underflow on PDM input.

**5.17.2 Conditions**

When sample rates are varying slightly from the typical values as illustrated in the datasheet.

**5.17.3 Technical Description**

Wrong comparator implementation will assert output underflow signal.

**5.17.4 Workaround**

Since the data quality is unaffected, ignore the output underflow.

**5.18 LCDC: RGB111-4 when Used in DBI-SPI I/F**
**5.18.1 Effect**

Visual artifacts on the LCD.

**5.18.2 Conditions**

A necessary condition to see the issue, is not to have a command enabled in the transfer, which means that a command transfer should not be combined with the pixel data transfer (LCDC\_DBIB\_CFG\_REG[LCDC\_DBIB\_SPI\_HOLD]=0).

**DA1469x Errata**
**5.18.3 Technical Description**

The RGB111-4 monochrome color format is first introduced into the LCDC to support the JDI-SPI LCD devices. However, this format is also available for use with non-JDI SPI LCD devices (DBI-SPI).

The issue takes place at the first byte that is transferred on the SPI3 or SPI4 interface, when a full or partial frame update is performed. The LSB of the first byte does not have the correct value, but the value of the last pixel data transfer from the earlier update.

The register `fmt_serial` is loaded with the pixel data; it converts the pixels raw to SPI bytes, according to the selected output format. Register `fmt_serial` is copied to the serializer register `spi_shift` that shifts the data to the SPI output. In the case of the RGB111-4 and if there is no combined command transfer, the first copy of the `fmt_serial` to the `spi_shift` is performed just before the last bit is updated.

If a combined command is to be transferred, the `fmt_serial` is prepared correctly during the time interval that is needed for the transferring the command over the SPI. So, in this case the first copy of the `fmt_serial` to the `spi_shift` is performed when the `fmt_serial` is correct.

In the case of the JDI-SPI LCD devices, there is always a combined command (LCD\_CDBIB\_CFG\_REG[LCD\_CDBIB\_SPI\_HOLD]=1 and a command in the corresponding CMD FIFO) at the beginning of a frame update, so the issue cannot take place. For the non-JDI SPI devices the issue can occur, because a command might not necessarily be combined with the pixel transfers.

**5.18.4 Workaround**

For non-JDI SPI devices, normally a command is necessary before the actual pixel data being transferred. To resolve the issue, set the `LCD_CDBIB_CFG_REG[LCD_CDBIB_SPI_HOLD]=1` and write the command in the corresponding CMD FIFO. In this way the command will be combined with the pixel data transfers and the first pixel byte will be ready before being used.

**5.19 XTAL32M Oscillator: Lowest Capbank Setting**
**5.19.1 Effect**

XTAL oscillator capacitance more than expected.

**5.19.2 Conditions**

Always

**5.19.3 Technical Description**

The capacitance in the lowest capbank setting is too high to meet the datasheet specification value. In the datasheet 4pF is stated, but in reality, the lowest setting for the capbank is 4.9pF (this includes board parasitic).

**5.19.4 Workaround**

Specification change

**5.20 Timer on System or TIM PD Counts +2 Every Time System Wakes up and -2 when Goes to Extend Sleep**
**5.20.1 Effect**

Timers count +2/-2 when switching from fast to slow clocks and vice versa. The overall count is correct. If the capture feature is used, it might not be counting correctly when events happen during sleep

**DA1469x Errata**
**5.20.2 Conditions**

Always

**5.20.3 Technical Description**

The *timer* uses system clock as a clock source. When the timer counts based on LP clock, the system clock is gated based on the rate of the LP clock.

If the system wakes up from extended sleep, the timer counts plus two. When the system clock switches from LP to high speed clock, two false LP clock events are generated.

When the system goes to extended sleep, the timer counts minus two. When the system clock switches from high speed to LP clock, one LP clock is missed, causing two missing counts.

In total, the timer counts correctly.

The capture value will be wrong if the event happens during deep sleep. There is no information if the event happens during sleep, so capture values have more jitter.

**5.20.4 Workaround**

None

**5.21 RFMON will Capture an Extra Word under Specific Conditions**
**5.21.1 Effect**

If data in the Test bus is produced at the same rate as the AHB clock, the RFMON will store one extra word in the memory.

**5.21.2 Conditions**

The previous description applies where the circular mode is not active. If the circular mode is active, the side effect is that the RFMON\_CRV\_LEN\_REG will not count the last write. However, the RFMON\_CRV\_ADDR\_REG will still be valid.

**5.21.3 Technical Description**

The root cause of the issue is that the pack\_en signal, which stops data capturing, will be disabled one clock cycle later than required, due to pipelining. As a result, the capturing logic will be disabled one clock cycle later and an extra sample will be introduced.

**5.21.4 Workaround**

For non-circular mode, the software should allocate an extra word for the dma buffer that will be used by RFMON for captured data storage. This prevents the allocation of this extra position for other usage and avoids data corruption.

For circular mode, the software should only use the value of the RFMON\_CRV\_ADDR\_REG in the calculations, instead of the value of the RFMON\_CRV\_LEN\_REG.

**5.22 CMAC, BS Stop During Sampling Point Calculation**
**5.22.1 Effect**

Radio operation breaks.

**DA1469x Errata**
**5.22.2 Conditions**

Always

**5.22.3 Technical Description**

If during the final stage of the HW calculations for the optimal sampling point a BS\_STOP event is received, then the logic will not be stopped.

**5.22.4 Workaround**

Whenever a forced BS\_STOP is generated, SW should generate a second BS\_STOP at least 1.5  $\mu$ s later.

In this way, SW will make sure that the timeout of the HW optimal sampling point is completed, and then the second BS\_STOP makes sure the request to stop is completed.

**5.23 Overload on V30 Rail can Cause a System Lock Up**
**5.23.1 Effect**

System locks and needs a POR to restart.

**5.23.2 Conditions**

Only observed when an overload (>160 mA) on V30 takes place.

**5.23.3 Technical Description**

An overload on the V30 rail can cause the system to get into a stable lock up state from which it cannot recover/reboot, even if the overload / error situation is gone. To restart, an external event is needed, like battery reconnect, VBUS supply, or triggering the RSTN pin for > 8 seconds (POR).

If the V30 gets too low, the bandgap collapses, the bias currents will go to 0, the reference levels will drop. The system switches to the (unpowered) VBUS supply, and then keeps hanging on the clamps until an external event triggers a POR.

**5.23.4 Workaround**

None

**5.24 CACHE Fetch During a 'Miss' is not Stalling the Sleep Process**
**5.24.1 Effect**

Miss-aligned DATA and TAG RAM content, which in turn leads to program execution faults.

**5.24.2 Conditions**

Always

**5.24.3 Technical Description**

Program execution from QSPI Flash cached after a power-up can result in faulty behavior.

This problem can happen when CACHE is retained due to:

- PMU\_CTRL\_REG[SYS\_SLEEP] = '1' setting over the APB-bridge takes time. New instructions might be fetched meanwhile from QSPI Flash memory, due to a CACHE miss on the next

## DA1469x Errata

instruction. Fetching in progress (and updating of the Cache memories) can be hard-interrupted by the system going to sleep

- The hardware problem here, is that the CACHE fetch (during a cache miss) is not stalling the sleep process, so starting *sleep* might happen during the fetch described above
- Ideally, the CACHE should set the TAG ram to invalid before the first update, and valid again after the last fetch
- Apparently, this does not happen with the current CACHE IP, leading to a miss-aligned DATA and TAG RAM content, which in turn leads to program execution faults

### 5.24.4 Workaround

The software workarounds are:

1. Either disable the Cache Controller just before going to sleep (and enable it again as soon as possible), or,
2. Link this part of code to non-cacheable area.

## 5.25 Charger: $I_{charge}$ Increases at Lower VBUS

### 5.25.1 Effect

Increased charge current by 4 to 20 %, depending on the (VBUS-VBAT) headroom (in the range of 700 down to 200 mV).

### 5.25.2 Conditions

When charging headroom is <700 mV.

### 5.25.3 Technical Description

When the headroom (VBUS - VBAT) of the charger is reduced, the charge current is increased. This effect becomes severe when the headroom becomes lower than 700 mV.

Lower charge currents (example, below 100 mA) are slightly less affected by this issue, because of the lower IR drop for these settings.

The root cause of the specific issue is not yet clear, but there is a strong suspicion towards a large mismatch between the output transistors and the current sensing device.

### 5.25.4 Workaround

None

## 5.26 V18F Switch Resistance too High

### 5.26.1 Effect

This has impact on the V18F level which supplies the QSPI FLASH chip externally, when the FLASH chip draws ~20 mA during program or erase. The V18F voltage can become too low.

### 5.26.2 Conditions

Always

**DA1469x Errata**
**5.26.3 Technical Description**

The measured switch resistance of the V18F switch is about 2.6 Ohms, so 1.6 Ohm higher than simulated.

Most of this was traced back to the connection between the switch and the QSPI supply pad. This connection is made in Metal 5 and is very long.

**5.26.4 Workaround**

Short V18F and V18P on PCB. Do not use the switch at all. Datasheet changed accordingly.

**5.27 CMAC, BS DMA doesn't support big endian**
**5.27.1 Effect**

Big endian format protocols will not be processed correctly.

**5.27.2 Conditions**

Always

**5.27.3 Technical Description**

Using the Bit Stream DMA in big endian format is not fully functional.

If the first byte is not word aligned, then the DMA will also transfer the remaining bytes of the given word.

**5.27.4 Workaround**

None. However, this mode is not used in BLE.

**5.28 ΣΔADC: Insufficient Settling Time when VBAT Scaler is Used**
**5.28.1 Effect**

When VBAT input is being sampled by the ΣΔADC, large errors occur (order of 500 mV).

**5.28.2 Conditions**
**5.28.3 Technical Description**

The 4x attenuator was designed as high-ohmic as possible to minimize unnecessary current flowing. Based on a 150 fF input capacitance of the modulator, the resistors were chosen such that there's ample settling time within 500 μs (one half cycle of 1 MHz). After layout, empty space was filled with additional decap, in order to improve noise performance 1.8 pF was added to the input of the modulator without considering the settling time of the VBAT attenuator. Result: when using the VBAT channel, there's way too less settling time for correct conversions. Effectively, the function becomes useless.

**5.28.4 Workaround**

The clock frequency of the SDADC can be reduced by a factor of four (set SDADC\_TEST\_REG[SDADC\_CLK\_FREQ]=0), but this also increases the conversion time by a factor of four.

**DA1469x Errata**
**5.29 LCDC : Wrong Partial Refresh**
**5.29.1 Effect**

Wrong LCD output. The issue affects the following color modes: L1, L4, L8, RGB332, RGBA5551, RGB565.

**5.29.2 Conditions**

In a partial refresh the LCD can request a refresh of a small number of LCD columns. If the number of LCD columns is less than four columns, the output of the LCDC is wrong.

**5.29.3 Technical Description**

The f\_ack is not produced correctly when the FSM is at the state L\_VAR and data is not retrieved correctly from the local sync FIFO. When the number of columns is less than four (size\_x < 4) the conditions relying on the value of endX\_diff (and affect the fsm\_ack) are not fulfilled. At the specific case when endX\_diff == 2 there is no active w\_ack or c\_ack and the code of the state is not selected at all. As a result, next data is not retrieved from the local sync FIFO at the right time.

**5.29.4 Workaround**

Restrict the number of columns of a partial LCD refresh to >= 4. In a case of request for the refreshing of less than four columns, the number of columns should be increased to four in the software driver of the LCDC.

This fix should be applied for all the mentioned color formats in the frame buffer.

**5.30 BOD doesn't function at low supply**
**5.30.1 Effect**

BOD block crashes if V30 < 1.9 V.

**5.30.2 Conditions**

Always

**5.30.3 Technical Description**

Comparator in BOD stops working on some samples at low supply voltages (<1.7 V). After MC sims, the expected minimum supply level is 1.9 V. Given that BOD is supplied by V30 rail and that the voltage dropout of the VBAT LDO might reach 250 mV, minimum VBAT voltage is 2.25 V or 2.35 V over temperature

**5.30.4 Workaround**

Change Datasheet minimum VBAT to 2.4 V. The system can still operate down to VBAT=1.8 V but without BOD

**5.31 XTAL32M oscillator: Track and Hold timing**
**5.31.1 Effect**

XTAL Oscillator clock output becomes unstable.

**DA1469x Errata**
**5.31.2 Conditions**

The track-and-hold is not allowed to toggle repetitively. As this is controlled by HW (radio\_enable), it cannot be guaranteed. If the track time is ~100  $\mu$ s and the hold time ~1 ms (repeated), the loop will become unstable.

**5.31.3 Technical Description**

To reduce phase-noise, the amplitude regulator of the XTAL32M can be put in HOLD mode. When the HOLD signal (i.e radio-enable) is frequently switched ON and OFF (with an OFF time <300  $\mu$ s) the amplitude regulator may become unstable. This is caused by (repetitive) amplification of a disturbance that is caused by switching the hold signal. This disturbance causes a (slight) overshoot in the amplitude regulation loop, which can be amplified when the signal is switched back to HOLD mode at the peak of the overshoot.

**5.31.4 Workaround**

- M33, SNC should never change XTAL mode
- When CMAC goes to sleep:
  - sets XTAL to TRACKING mode (so that XTAL will always power up in TRACKING mode)
  - starts a timer (XTAL\_tracking\_mode\_timer)
- When CMAC wakes up:
  - sets XTAL to HOLD mode before RF activity starts (so that no switching between TRACKING & HOLD takes place during RF activity)
  - checks time elapsed since XTAL was set to TRACKING mode
  - XTAL\_tracking\_mode\_timer  $\geq$  300 uSec: No action
  - XTAL\_tracking\_mode\_timer < 300 uSec: Before going to sleep, switch to TRACKING mode and wait for 300  $\mu$ s (this delay should obviously be taken into consideration when calculating the time of next wakeup)

**5.32 XTAL32M oscillator: Minimum OFF Time**
**5.32.1 Effect**

Indication that XTAL32M is settled when it is not.

**5.32.2 Conditions**

The crystal oscillator is not allowed to be OFF for a shorter time than ~1 ms. Low OFF times will result in the subsequent startup being (much) faster. This will cause the analog comparator (which detects steady-state) to trip earlier than normal. With very low OFF times, the comparator will not function at all.

**5.32.3 Technical Description**

The crystal oscillator includes an analog comparator that detects steady-state of the crystal. It is intended to be used to calibrate the XTAL\_IRQ timer and to switch the capacitor bank from the XTAL32M\_START to the XTAL32M\_VALUE setting. When the oscillator is OFF for a short time, the crystal resonance is not dampened before the subsequent startup. This causes the comparator to detect a faulty calibration result and cause a faulty setting of the capacitor bank.

**5.32.4 Workaround**

Do the following:



**DA1469x Errata**

1. Do not update IRQ counter (use fixed timing).
2. Always force the comparator to trip (by SW).

**5.33 Charger CC-CV Comparator Hysteresis**
**5.33.1 Effect**

Too many interrupts are generated by the digital FSM, as the latter is switching from Constant Current (CC) to Constant Voltage (CV) state and vice-versa, until the Charger's analog loop is stabilized in CV mode.

**5.33.2 Conditions**

Always applicable while charging.

**5.33.3 Technical Description**

There is not enough hysteresis built in the comparators that define the CC or CV mode in the analog loop. So when the battery voltage reaches a threshold from which the transition from CC to CV mode begins, the respective signals ("cc\_mode", "cv\_mode") driven by the Charger's analog block towards the digital FSM, are toggling.

Since these signals are complementary to each other, this makes the digital FSM transitioning between the CC and CV states, until the battery voltage is sufficiently increased, stabilizing the analog loop in CV mode, and the digital FSM in CV state.

**5.33.4 Workaround**

The interrupts signaling the transition of the digital FSM from CC to CV state and vice versa can be masked. This is possible by disabling the respective interrupt enable bit-fields, as follows:

- `CHARGER_STATE_IRQ_MASK_REG[CC_TO_CV_IRQ_EN] = '0'`
- `CHARGER_STATE_IRQ_MASK_REG[CV_TO_CC_IRQ_EN] = '0'`

This only masks the interrupts due to the specific FSM transitions. This means that any other Charger interrupt that may happen due to normal charging or because of an error condition during charging is not affected and will be captured, provided that the respective interrupt mask register bit is enabled.

**5.34 RF Variable Vref Decreases with Low VDDD**
**5.34.1 Effect**

Decreased Radio performance. Radio expects a stable voltage reference of 1.2 V. However, the reference depends on VDDD which is usually 0.9 V unless the PLL is running. Radio operates out of specification leading to degraded performance.

**5.34.2 Conditions**

Always, when the system runs at 32 MHz.

**5.34.3 Technical Description**

RF variable Vref is implemented with a resistive ladder. The ladder resistors are connected with a PMOS switch to Vout, when  $V_{out} > V_{DDD}$  for a certain value, the PMOS switches cannot remain open, and will pull the output voltage toward VDDD.

---

**DA1469x Errata****5.34.4 Workaround**

Radio operation will be enveloped with programming the core voltage (VDDD) at 1.2 V from 0.9 V, so the radio operates with an expected reference.

**5.35 PLL calibration does not work properly****5.35.1 Effect**

The PLL will not always lock.

**5.35.2 Conditions**

Silicon dependent.

**5.35.3 Technical Description**

The automatic calibration ends, in some cases, in the wrong band. The VCO voltage is < 150mV while it should be between 150mV and 620mV.

**5.35.4 Workaround**

The SW workaround is to measure  $V_{tune}$  with the GP-ADC and compare it with the ADC referred PLL comparator levels. When  $V_{tune}$  is outside the comparator range – even though if the PLL is locked – the internally determined VCO current by the calibration algorithm is going to be overruled by the value programmed in the PLL's system register.

## DA1469x Errata

## Revision History

Revision	Date	Description
1.2	22-Feb-2022	Updated logo, disclaimer, copyright.
1.1	15-Mar-2019	Clarifications, description improvements <ul style="list-style-type: none"><li>• Errata #276 description improved</li><li>• Errata #273, #279, #282, #284, #296, clarification regarding which ADC they address</li><li>• Errata #279 workaround concept added</li><li>• Errata #284 workaround concept updated</li></ul>
1.0	21-Feb-2019	Initial release

---

**DA1469x Errata****Status Definitions**

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.