Decawave Is Now qorvo

DW1000 Errata

Version 1.4

This document is subject to change without notice.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

DOCUMENT INFORMATION

Disclaimer

Decawave reserves the right to change product specifications without notice. As far as possible changes to functionality and specifications will be issued in product-specific errata sheets or in new versions of this document. Customers are advised to check with Decawave for the most recent updates on this product.

LIFE SUPPORT POLICY

Decawave products are not authorized for use in safety-critical applications (such as life support) where a failure of the Decawave product would reasonably be expected to cause severe personal injury or death. Decawave customers using or selling Decawave products in such a manner do so entirely at their own risk and agree to fully indemnify Decawave and its representatives against any damages arising out of the use of Decawave products in such safety-critical applications.

Caution! ESD sensitive device. Precaution should be used when handling the device in order to prevent permanent damage.

REGULATORY APPROVALS

The DW1000 or DW1000 based products, as supplied from Decawave, have not been certified for use in any particular geographic region by the appropriate regulatory body governing radio emissions in that region although it is capable of such certification depending on the region and the manner in which it is used.

All products developed by the user incorporating the DW1000 must be approved by the relevant authority governing radio emissions in any given jurisdiction prior to the marketing or sale of such products in that jurisdiction and user bears all responsibility for obtaining such approval as needed from the appropriate authorities.

1 INTRODUCTION

This errata document details known issues with the DW1000 product. Where available workarounds are presented.

1.1 *Package Marking Definitions*

Package marking details for the DW1000 can be found on the top of the IC as per Figure 1.

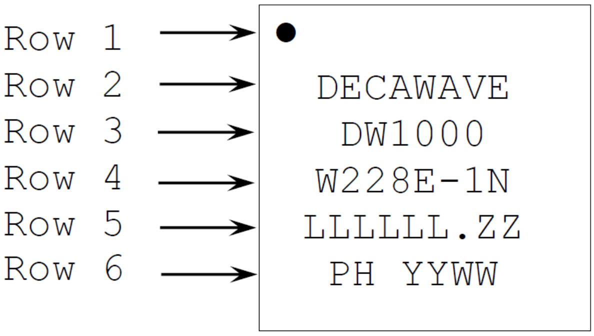

Figure 1 Device Package Markings

Row 1: Pin 1 Designator	Used for device orientation
Row 2: Company Name	**DECAWAVE**
Row 3: Part name	**DW1000**
Row 4: DW1000 part number	**W228E*-1N**

* DW1000 silicon revision identifier. In this example the revision of the IC is “E”. Devices marked with silicon revision identifier prior to 'E' (for example ‘C’) should not be used for DW1000 designs.

Row 5: Manufacturer lot number	**LLLLLL.ZZ** **LLLLLL** is root wafer fab lot number. **ZZ** is the lot split number from fab.
Row 6: Package assembly information.	**PH**: Manufacturing Site Code **YYWW**: Date Code (**YY** =Year; **WW** = Week)

2 Errata Overview

Functional Problem	Short Description	Silicon Revision Identifier	Detailed Description
TX-1	DW1000 may fail to indicate timeout events	E	3.1
RX-1	The data in the received buffer can be corrupted if a frame is transmitted before the data is read out.	E	3.2
TX-2	Writing to the TX buffer while transmission is in progress can corrupt the data to be transmitted	E	3.3
IRQ-1	Spurious IRQ events can occur if double buffer operational mode is used.	E	3.4
PMSC-1	DW1000 may fail to wake up from DEEPSLEEP mode with a 500 µs wakeup event.	E	3.5
SYSSTAT-1	The PLL clock status bits may indicate the PLL failed to lock even though the PLL locked successfully.	E	3.6

Devices marked with silicon revision identifier prior to 'E' (for example ‘C’) should not be used for DW1000 designs.

3 FUNCTIONAL PROBLEMS DETAIL

3.1 TX-1

3.1.1 Introduction

For delayed transmission, the transmit time is programmed into the DW1000. One of the design objectives of delayed transmission is that the specified transmission time would be predictable and aligned with the transmit timestamp. See the section on Delayed Transmission in the DW1000 User manual.

3.1.2 Problem

A DW1000 may fail to indicate timeout events when issuing a delayed TX command under certain conditions. This could result in scheduled transmissions not completing.

When setting up delayed transmission (programming the delayed send time and then issuing the delayed transmit command), the following will happen: If the send time is imminent or the send time has passed - then the HPDWARN is flagged all the time (time NOK), as the time is increased then TXPUTE will be flagged (time TXPUTE). If the send time is far enough in the future (time OK) then no flags will be flagged and the frame is sent at the programmed time.

However, there is a window when the programmed time is greater than time TXPUTE but less than time OK at which there are no flags but the transmission does not happen. In this case there will be no TX done event generated.

3.1.3 Workaround

Check if the delayed TX has been sent in time, if there is no TX done event within the period, then issue TRXOFF and enter RX or handle as timeout.

If the TX clock is enabled (PMSC_CTRL0 bits 5,4 set to 1,0 - force TX clock on) before the delayed TX command is issued than the problem does not happen.

Further information on PMSC_CTRL0 can be found in the DW1000 user manual.

3.2 RX-1

3.2.1 Introduction

The DW1000 has a double buffered receiver allowing reception of a new frame to proceed in one buffer while the host processor is in the process of reading the last frame received into the other buffer of the buffer pair.

3.2.2 Problem

The 129th octet (i.e. buffer offset index[128]) of the second RX buffer (i.e. the one accessed when HSRBP = 1) gets corrupted when the user writes TX data at offsets greater than index 127, and issues a TX send command, before reading the received frame.

3.2.3 Workaround

There are three workaround options:

(1) In this double buffered use case, when receiving long messages, do not send responses that fill beyond 127 octets.

(2) Only use short messages when using double buffer mode.

(3) Only use single buffering when using long messages (>127 bytes).

3.3 TX-2

3.3.1 Introduction

For applications that require high speed transmission the DW1000 has specific features implemented to support maximum utilisation of the transmitter.

One of the features is the fast turnaround, whereby it is possible to initiate preamble sending before writing the frame data to the TX_BUFFER or writing the frame length to the length as specified in the TFLEN and TFLE fields. See section on High Speed Transmission in the DW1000 user manual.

3.3.2 Problem

When preparing the transmission of the next frame, by writing to a part of the TX buffer which is not used for the current transmission, the new data written is written erroneously at offset 0, thus corrupting the data currently being transmitted.

The issue is that when the transmitter is enabled, the address to the TX buffer index is reset to 0.

3.3.3 Workaround

There are two workarounds:

(1) Delay the writing of the next transmission data until the TXFRB ISR of the current transmission is triggered.

(2) The second workaround is to carry out a dummy write of 12* bytes to location 0x3F after TXSTRT and before writing the frame data to ensure that the TX enable has asserted and reset the address before the data is written. See section TX events in the debugging application note, see APS022, debugging DW1000 based products and systems.

*Note: The number of dummy bytes to be written to location 0x3F might differ for different designs as it could depend on the LDO power up time and when the corruption of data happens.

3.4 IRQ-1

3.4.1 Introduction

The DW1000 supports a double buffered receiver allowing reception of a new frame to proceed in one buffer while the host processor is in the process of reading the last frame received from the other buffer.

3.4.2 Problem

In double buffer operational mode the DW1000 may produce a short pulse (glitch) on the IRQ line with a duration of approximately 8 ns, this pulse could be detected by the connected MCU and trigger an IRQ event. The MCU's *irq_pin()* interrupt service routine will subsequently execute the *dwt_isr()* driver function, which will produce an additional reading of the DW1000 STATUS register which indicates there are no events. Although there are no problem if *dwt_isr()* is used, the additional read causes unnecessary spend of host MCU time to read the DW1000 STATUS register.

3.4.3 Workaround

There is a software workaround which will help prevent reading the DW1000 STATUS register unnecessarily during double buffer operational mode. The workaround is to read the IRQ_PIN line status and process `dwt_isr()` only if the IRQ_PIN is set.

```
void irq_pin(void)
{
    while ( IRQ_LINE_IO == HIGH )
    {
        dwt_isr();
    }
}
```

3.5 PMSC-1

3.5.1 Introduction

The DW1000 supports a low-power DEEPSLEEP mode. In this mode all internal circuitry is powered down with the exception of the always-on memory which can be used to hold the device configuration for restoration on wakeup. Using low-power DEEPSLEEP mode will cause the 38.4 MHz reference oscillator to be stopped.

3.5.2 Problem

The DW1000 User Manual and Datasheet specify a wakeup event, configurable to be either pulling low of the SPICSn line or driving high of the WAKEUP line, for a duration of 500 µs, will wake up the DW1000. In fact, the configured wakeup line needs to be asserted for at least the time it takes for the internal digital clock to start. This time depends on the start-up time of the 38.4 MHz reference oscillator and can be more than 500 µs.

3.5.3 Workaround

It is recommended to enable the PLL lock event (MCPLOCK bit) or SLEEP to INIT event (MSLP2INIT bit) in the System Event Mask (SYS_MASK) register and to assert the configured wake up source until either event occurs. Using the PLL lock event has the added benefit of allowing the use of high-speed SPI without requiring any additional wait for the PLL clock to lock. See INIT state description in section 2.3.2 of the DW1000 user manual for more information.

Alternatively, the RSTn line can be monitored to detect the DW1000 transitioning from WAKEUP to INIT mode. The DW1000 will pull down the RSTn pin while in WAKEUP mode and pull it high with an internal pull up resistor once INIT mode is reached. On the host side, the RSTn pin can be configured as a high-impedance input and the configured wake up line can be asserted until the RSTn transitions from a low to a high state. Note that an additional wait should be added before performing SPI transactions when automatic transitioning from INIT to IDLE mode is enabled.

The simplest solution is to increase the time of the wakeup event. The minimum time can be found empirically by performing multiple DEEPSLEEP – WAKEUP cycles, increasing the wakeup event time until no wakeup failures occur. It is advisable to test this at the maximum operating temperature since the clock start-up time is often temperature dependant. Ensure to use a sleep time equal to or longer than the intended sleep time or to ensure internal voltage rails have time to discharge. The minimum time of the wakeup event can also be calculated based on the start-up time of the reference clock. For TCXOs and other integrated oscillators this time is often documented. A safety margin of at least 200 µs should be added to the clock start-up time to account for part variances.

3.6 SYSSTAT -1

3.6.1 Introduction

The DW1000 system clock runs off a 125 MHz PLL clock for high speed SPI and RF operation. This PLL clock needs to successfully lock before it can be used. The PLL lock status can be monitored using status bits in the system status and RF status registers, allowing the host controller to know when the PLL clock is ready to be used.

3.6.2 Problem

The PLL lock status bits can indicate the PLL failed to lock, even if the PLL locked successfully. The affected status bits are:

- The CPLOCK and CLKPLL_LL bits in the system status register, described in section 7.2.17 of the DW1000 user manual.
- The CPLLLOCK in the RF status register, described in section 7.2.41.7 of the DW1000 user manual.

This problem can occur due to temperature, voltage or part variation (PVT). Originally the PLL calibration was found to work reliably over a wide number of devices, over the full working voltage and temperature range, but it has now been found some devices can indicate PLL locking failures, even at room temperature. The PLL was found to still lock successfully on affected devices, the status bits are found to be more sensitive to PVT than the actual PLL.

3.6.3 Workaround

Since the PLL will typically lock successfully and only the status bits are incorrect, the PLL lock status bits can be ignored and instead a conservative wait period can be used when locking the PLL (i.e when switching from INIT to IDLE mode). However, this prevents detecting issues with the PLL and can cause problems if the wait period is too short. This method is only advisable for devices with a known good oscillator circuit and use cases that do not require minimizing startup times.
It is recommended to test the PLL lock (e.g using continuous frame mode) across the working temperature and voltage range on all devices to ensure the PLL status bits can be safely ignored.

To ensure the PLL lock bits operate correctly, the PLL must be calibrated to account for PVT. If the temperature changes significantly (typically ~20°C) the PLL lock bits may indicate the PLL is losing lock, requiring the PLL to be recalibrated.

The following steps will recalibrate the PLL:

- Ensure the DW1000 is in the INIT state, i.e it is clocked of the XTI clock and automatic transitioning to the IDLE state is disabled.
- Trim the oscillator frequency using the xtal trim register if needed.
- Write 0xE7950448 to the FS_RES1 register (address 0x2B, offset 0x00).
- Wait for the PLL lock bit to be set

If the DW1000 is in IDLE state, i.e if the PLL clock is locked, the DW1000 needs to be reset before the PLL clock can be recalibrated.

Example code to calibrate the PLL clock in the dwt_initialise function:

```
int dwt_initialise(uint16 config)
{
        …
        _dwt_enableclocks(FORCE_SYS_XTI); // NOTE: set system clock to
        XTI - this is necessary to make sure the values read by
        _dwt_otpread are reliable
        …
        // Configure XTAL trim
        dwt_xtaltrim(dw1000local.xtrim);
        …
        // Calibrate the PLL clock
        dwt_write32bitoffsetreg(FS_CTRL_ID, FS_RES1_OFFSET,
        0xE7950448);
        // Wait for PLL lock bit to be set
        …
}
```

4 DOCUMENT HISTORY

Table 1: Document History

Revision	Date	Description
1.0	8-Feb-2018	Initial release
1.1	22-Feb-2018	Logo update, added note on older silicon revisions
1.2	5-Oct-2018	Added erratum IRQ-1
1.3	16-Dec-2020	Logo update, added erratum PMSC-1
1.4	15-Apr-2021	Added erratum SYSSTAT-1

5 FURTHER INFORMATION

Decawave develops semiconductors solutions, software, modules, reference designs - that enable real-time, ultra-accurate, ultra-reliable local area micro-location services. Decawave's technology enables an entirely new class of easy to implement, highly secure, intelligent location functionality and services for IoT and smart consumer products and applications.

For further information on this or any other Decawave product, please refer to our website www.decawave.com.