AEROSPACE DATA EXCHANGE PROGRAM TRANSMITTAL

PROBLEM ADVISORY

1. TITLE SMD# 5962-17212 ADC AND I2C FUNCTIONAL MANUAL ERRATA	**2. DOCUMENT NUMBER** SPO-2021-PA-0004A
	3. DATE *(Year, Month, Date)* 2021, SEP, 21
4. MANUFACTURER NAME AND ADDRESS CAES 4350 CENTENNIAL BOULEVARD COLORADO SPRINGS, COLORADO 80907-3486	**5. MANUFACTURER POINT OF CONTACT NAME** Owen Watry
	6. MANUFACTURER POINT OF CONTACT TELEPHONE 719-594-8293
	7. MANUFACTURER POINT OF CONTACT EMAIL Owen.Watry@caes.com

8. CAGE CODE 65342	**9. LDC START** ALL	**10. LDC END** ALL	**11. PRODUCT IDENTIFICATION CODE** QS30	**12. BASE PART** TABLE 1
13. BLANK			**14. SMD NUMBER** 17212	**15. DEVICE TYPE DESIGNATOR** 01, 02
			16. RHA LEVELS ALL	**17. QML LEVEL** ALL
			18. NON QML LEVEL ALL	**19. GIDEP NUMBER** GB4-P-21-04A

20. PROBLEM DESCRIPTION / DISCUSSION / EFFECT

CAES has issued Amendment A to add a note to block 21 to address the anomaly encountered in some web browsers. The original document, SPO-2021-PA-0004, was published August 30, 2021.

The Functional Manual for SMD# 5962-17212 has three new errata and one modified errata concerning the ADC peripheral, and one new errata concerning the I2C peripheral. All changes made to the Functional Manual can be found in the Revision History section of the document.

The updated errata is:

- ADC Continuous Mode Wrap Around Hardware Inaccurate Delay Workarounds
  - This update is made to the workaround subsection, changing it to accommodate the new ADC External vs Internal Oscillator CONV_COMPL Conflict errata

The new errata are:

- ADC External vs Internal Oscillator CONV_COMPL Conflict
- ADC COI3_OVER Low Voltage False Flag
- ADC Even/Odd Offset Error
- I2C RD_REQ Interrupt Re-Asserts Itself After a Slave Address Register Change

The errata are listed on the following sheets as they are published in the Functional Manual v1.2.0.

Continued on page 2.

21. ACTION TAKEN / PLANNED

CAES has published this errata with workarounds in the Functional Manual, and released this notification. The Functional Manual can be found on the caes.com product page: https://caes.com/product/ut32m0r500#downloads

(NOTE: Some browsers will modify the above link. If you encounter a 404 webpage, check the url is correct)

22. DISPOSITIONARY RECOMMENDATION:	**CHECK & USE AS IS** ☐	**CONTACT MANUFACTURER** ☐	**REMOVE & REPLACE** ☐	**CORRECT & USE AS SPECIFIED** ☒

Table 1: Affected Part Numbers

UT32M0R500-ZPC
UT32M0R500-ZFC
UT32M0R500LZLC
UT32M0R500-SPF
UT32M0R500-SFF
UT32M0R500LSLF
UT32M0R500LCLF
UT32M0R500-CPA
5962L1721201QXC
5962L1721201QYF
5962L1721202QXC
5962L1721202QYF

ADC Continuous Mode Wrap Around Hardware Inaccurate Delay

In Continuous mode, the ADC performs continuous conversions sequentially on all enabled channels. From one channel to the next, the hardware has a predefined delay for setup time that allows for an accurate ADC conversion. Once the last channel finishes an ADC conversion, the hardware wraps around to the first enabled channel in the sequence to begin a new conversion. When the hardware wraps around from last to first channel in the sequence, the predefined delay for setup time is reduced and causes an inaccurate ADC conversion reading in the first enabled channel.

Note: the “first channel” is not physical channel #1, but the first enabled channel in the sequence, i.e., channel #2 in enabled channel 2-3, channel #5 in enabled channel 5-10, channel #1 in enabled channel 1-15 or any other sequence combination.

Workarounds

1) Run the ADC in Single Sweep mode, and add a software delay of 20us minimum between each sweep completion (CONV_COMPL_COMB and INTR_PEND bit of highest enabled channel) and the triggering of the next sweep

2) In continuous mode, configure the first enabled channel as a "dummy" channel, and only keep data from the second enabled channel and beyond.

ADC External vs Internal Oscillator CONV_COMPL Conflict

When the UT32M0R500 uses an external clock as its system clock (PCLK), the ADC peripheral still uses the internal 50MHz oscillator to perform conversions (at a clock divided rate determined by the ADC_OSCDIV[1:0] bits in the ADC.TIM_CTRL register). When using an external system clock, the ADC_CONV_COMPL bit, which is supposed to set at the end of a conversion, will sometimes not set. The DATA_VALUE_STALE bit located in the ADC->DATA register, bit 15, used the ADC_CONV_COMPL bit to determine if data was stale. Both bits have been marked as having this chance for error. This is due to a hardware error that occurs a small fraction of the time caused by using two different clocks. When using the internal 50MHz oscillator as the system clock (PCLK), this is not an issue.

Workarounds

Bits 31 and [24:0] found in ADC->INT_STATUS are still accurate and can be used to determine which channels have completed a conversion. Once an enabled channel has been sampled, its respective INTR_PEND bit will set along with the CONV_COMPL_COMB bit. Once the INTR_PEND bit for the highest enabled channel is set along with the CONV_COMPL_COMB, the full ADC sweep is complete. This method does not require users to enable ADC interrupts, regardless of the bit name.

There is no workaround for the DATA_VALUE_STALE bit.

ADC COI3_OVER Low Voltage False Flag

The COI3_OVER flag signals to the user when the input voltage from the most recent measurement is out of range. Although it is called COI3_OVER, the flag signals both over-voltage and under-voltage measurements. When the input into a Single-Ended channel is at or below 0.003V (3mV), the COI3_OVER flag has a chance to set. This chance increases as the input voltage approaches 0V, to the point where the COI3_OVER flag is always set. This false flag only occurs when all of the data bits in the respective channel's ADC.DATA register are equal to zero (both the DATA_OUT[11:0] bits and the DATA_OUT_LSB[3:0] bits. If any of these data bits are set in conjunction with the COI3_OVER flag, the flag is operating as intended.

Note that the 3mV value was measured using a high-precision voltage supply and noise-reducing practices. Noise on the voltage input will cause the COI3_OVER false flags to occur at a higher voltage threshold.

Workarounds

If this flag is set and the DATA_OUT and DATA_OUT_LSB bits of the ADC.DATA register are zero, users can ignore the COI3_OVER flag.

ADC Single-Ended Even/Odd Channel Offset Error

When ADC channels are configured as single-ended inputs, even channels (ADC_SE_CHAN_[0,2,4,6,8,10,12,14]) and odd channels (ADC_SE_CHAN_[1,3,5,7,9,11,13,15]) have an offset error that becomes noticeable when the channels are measuring the same signal, but is always there in single-ended mode. The offset error affects the singe-ended channels by the same magnitude, but opposite polarity. Even channel measurements are increased by the offset, odd channel measurements are decreased by the offset. Even channel codes are typically 20 counts higher than the odd channel codes (~7.3mV with a gain of 1V/V). The offset is uniform when measured on a single part, but may vary between parts.

When a channel is configured as a differential input, the offset cancels itself out, and has not been observed on any measurements or characterization data.

Workarounds

Users that intend to use ADC channels as Single-Ended inputs should characterize the even/odd offset by comparing the output codes of an even and odd channel measuring the same voltage, and then addressing that offset in their software. For additional accuracy, users can screen individual parts to find a more accurate part-specific offset for their software.

I2C RD_REQ Interrupt Re-Asserts Itself After a Slave Address Register Change

Some I2C systems will require the I2C slaves to change or modify their slave address. The UT32M0R500 does this by changing the Slave Address Register (I2C.SAR). To change a peripherals' SAR, first clear the I2C.ENABLE register's Enable bit (bit 0). Then write the new Slave Address to the I2C.SAR register. Finally, set the I2C.ENABLE Enable bit. When an I2C master wants to read data from an I2C slave, the RD_REQ interrupt will set when the I2C slave detects an I2C read condition that targets its current SAR. Directly after this interrupt signal is cleared and the read transaction completed, if the slave device attempts to change its Slave Address, upon re-enabling the I2C peripheral the RD_REQ interrupt will set itself, even though there is no such transaction on the I2C bus for the new Slave Address Register value.

Workarounds

To avoid this flag from setting, the most reliable option is to implement a small delay before changing the SAR register after a RD_REQ flag was set.